

# MOBILE PROGRAMMING FOR STUDENT

MOBILE PROGRAMMING FOR STUDENT



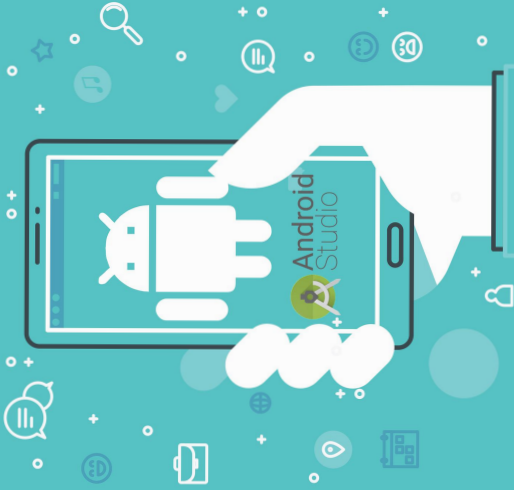
**EDY BUDIMAN** adalah staff pengajar di Program Studi Teknik Informatika Universitas Mulawarman Samarinda sejak tahun 2012. Lahir di Sidenreng Rappang (Sidrap) Sulawesi Selatan dan menyelesaikan studi Magister di Universitas Hasanuddin Makassar.

Pengalaman mengajar di matakuliah komputer sejak tahun 2002 di Universitas Negeri Makassar; tahun 2008 mengajar di Universitas Sulawesi Barat dan Universitas Muslim Indonesia tahun 2011. Matakuliah yang diampu lebih pada matakuliah pemrograman, dari pemrograman prosedural, non procedural, object, web hingga pemrograman mobile-penulis.

Saat ini penulis lebih banyak melakukan riset-riset terkait dengan implikasi dan performance perangkat-aplikasi mobile dalam jaringan telekomunikasi yang sampai saat ini perkembangan teknologinya sangat pesat yang berdampak pada perilaku gaya hidup masyarakat generasi **Zaman Now**.

"Kata **TIDAK BISA**, akan membuatmu berhenti berusaha. Terus Berkarya dan Tetap Semangat" adalah motto penulis yang selalu disampaikan kepada seluruh mahasiswa agar mereka senantiasa mampu melewati segala tantangan dan persaingan yang ada.

# MOBILE PROGRAMMING FOR STUDENT



EDY BUDIMAN

EDY BUDIMAN

# MOBILE PRGRAMING FOR STUDENT

Penulis : Edy Budiman

Editor & Cover Design: Andi Hafitz Khanz

ISBN : 978-602-6834-88-1© 2018. Mulawarman University Press

Edisi : 2019

Hak Cipta Dilindungi Undang-Undang Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apapun tanpa izin tertulis dari penerbit

Isi diluar tanggung jawab percetakan.

Edy Budiman. 2018. Mobile Programing for Student.

Mulawarman University Press. Samarinda

# KATA PENGANTAR



Saat ini institusi pendidikan di Indonesia mulai menyadari pentingnya pemanfaatan teknologi komunikasi dan informasi untuk meningkatkan kualitas aktivitas belajar dan mengajar. Teknologi selular yang menggunakan konsep pembelajaran virtual melalui aplikasi dan akses internet memungkinkan dosen dan mahasiswa saling berinteraksi secara *mobile* kapan saja dan di mana saja.

Buku **Mobile Programming for Student with Android App Development** disusun dan hadir sebagai upaya meningkatkan kompetensi (learning outcome) mahasiswa dan keunggulan kompetitif yang menyediakan solusi untuk mengatasi kebutuhan IT, yang akan terus berinovasi untuk menciptakan solusi yang tepat untuk mahasiswa ditingkat universitas di Indonesia.

Kami berharap buku *Mobile Programming for Student with Android App Development* ini dapat menjadi salah satu buku pegangan pembelajaran untuk mahasiswa di matakuliah pemrograman mobile, mobile computing, dan atau pemrograman visual lanjutan dengan perangkat bergerak (mobile) bagi universitas di Indonesia.

Pada buku ini penulis akan membahas cakupan materi-materi perkuliahan pada matakuliah pemrograman mobile berbasis aplikasi Android beserta contoh latihan disetiap materi.

## STRUKTUR PENULISAN BUKU

Buku *Mobile Programming for Student with Android App Development* ini membahas tentang cakupan materi perkuliahan yang kami ajarkan di Program Studi Teknik Informatika (Buku Ajar) untuk mendukung pembelajaran Mahasiswa pada matakuliah pemrograman mobile di tingkat universitas. Untuk memudahkan memahami materi yang disampaikan buku ini dibagi menjadi 8 bab, yaitu:

**Bab 1** : Pengantar Pemrograman Mobile

Bab ini membahas konsep dasar pemrograman mobile, sistem operasi, kelemahan dan kelebihan dari sistem operasi mobile, arsitektur dan perangkat serta penggunaan umum perangkat mobile

**Bab 2** : Pengantar Aplikasi Android Studio

Bab ini membahas perkembangan android, versi android, dasar pengembangan, pemasangan/installasi, konfigurasi, modul dan pembuatan project aplikasi pertama menggunakan android studio. Selain itu juga membahas struktur file, direktori dan pengujian project menggunakan perangkat secara nyata di emulator android atau smartphone Di akhir bab akan memberikan latihan pemasangan/installasi, dan latihan pembuatan project helloworld dan me-running menggunakan tool emulator android studio.

**Bab 3** : Layout User Interface

Bab ini membahas konsep user interface (UI) pada Tool Android Studio, tata letak (layout) dalam xml maupun desain, mengenal atribut layout, parameter, constraint, dan jenis-jenis layout. Menggunakan editor layout di Android Studio, memposisikan tampilan di jenis-jenis layout. Diakhir bab akan memberikan latihan dan tugas cara penggunaan layout dalam project menggunakan Android Studio.

**Bab 4** : Kontrol User Interface

Bab ini membahas kontrol user interface, komponen masukan (widget), jenis-jenis kontrol, penggunaan scrollview, menambahkan beberapa elemen TextView ke layout XML, membuka aplikasi maps, menambahkan ikon untuk item-menu, dan item yang terkait dengan menu dan opsi. Diakhir bab akan memberikan latihan dan tugas penerapan dari materi dalam bab ini.

**Bab 5** : Activity dan Intent

Bab ini membahas tentang activity, membuat activity, tentang intent, activity dengan intent eksplisit, meneruskan data antara activity dengan intent, mendapatkan data kembali dari activity, dan navigasi activity. Diakhir bab akan memberikan latihan dan tugas membuat aplikasi baru untuk mengirimkan intent implisit dan mengimplementasikan dua intent implisit yang membuka halaman web dan membuka lokasi pada peta.

**Bab 6** : Shared Preferences, File, dan Penyimpanan Data

Bab ini membahas preferensi bersama (shared preference), penyimpanan data internal dan eksternal, database SQLite, bahasa kueri dan struktur, memproses cursor, model data, mencadangkan database dan opsi lain seperti koneksi jaringan dan firebase. Diakhir bab akan memberikan latihan dan tugas Menambahkan kemampuan untuk menyimpan,



mengambil, dan menyetel ulang preferensi bersama ke aplikasi dan lainnya yang terkait dengan preferensi bersama

**Bab 7** : Aplikasi Create, Update, Insert, dan Delete (CRUID) Database

Bab ini membahas SQLiteDatabase kerangka kerja Android dan kelas SQLiteOpenHelper yang akan mengeksekusi permintaan dan mengelola database untuk Anda. Secara khusus dengan penggunaan database SQLite, Cursor dan ContentValues. Selain itu bab ini membahas opsi storage database yang lain seperti persistensi Room SQLite dan set up Firebase realtime database yang akan disajikan contoh implementasi pada aplikasi operasi create, update, insert, dan delete (CRUID). Di akhir bab akan memberikan latihan pembuatan aplikasi CRUID.

**Bab 8** : Pengembangan Aplikasi Android dengan Kotlin

Bab ini membahas konsep dasar bahasa pemrograman kotlin pada android (Android Kotlin), konfigurasi kotlin ke android studio, kotlin extensions, kotlin dasar: Packages dan Function dasar, dan pembuatan project aplikasi sederhana menggunakan Kotlin. Di akhir bab akan memberikan contoh latihan pembuatan aplikasi sederhana menggunakan kotlin.

## CAPAIAN PEMBELAJARAN (LEARNING OUTCOME)

1. Mengetahui perkembangan teknologi perangkat lunak dan perangkat keras pemrograman mobile
2. Mengetahui perkembangan teknologi setiap versi Android dan cara installasi perangkat pendukungnya.
3. Mampu menerapkan dan menggunakan jenis-jenis layout user interface dalam pengembangan project aplikasi android.
4. Mampu menerapkan dan menggunakan jenis-jenis komponen kontrol user interface dalam pengembangan project aplikasi android.
5. Mampu menerapkan penggunaan activity dan intent dalam pengembangan project aplikasi android
6. Memahami dan mengetahui cara penggunaan preferensi bersama dengan API yang bisa diakses/dibagikan semua komponen internal aplikasi yang dikembangkan.
7. Mengetahui dan menguasai penggunaan database dan operasinya dalam pengembangan aplikasi sistem manajemen data berbasis android.
8. Mengetahui Bahasa pemrograman Kotlin pada android (android-kotlin) dalam pengembangan aplikasi mobile.



# DAFTAR ISI



<b>Kata Pengantar</b> .....	3
<b>Daftar Isi</b> .....	7
<b>Daftar Gambar</b> .....	11
<b>Daftar Tabel</b> .....	15
<b>Daftar Latihan</b> .....	16
<b>Bab 1: Pengantar Pemrograman Mobile</b> .....	20
1.1.    Konsep Dasar .....	20
1.2.    Sejarah Perkembangan.....	21
1.3.    Sistem Operasi .....	23
a.    Android OS .....	23
b.    BlackBerry OS .....	26
c.    iOS Apple Inc.....	27
d.    Symbian OS .....	28
e.    Windows Phone OS .....	30
f.    Palm OS .....	31
1.4.    Kelemahan dan Kelebihan Sistem Operasi Mobile.....	36
a.    Sistem Operasi Windows Mobile.....	36
b.    Sistem Operasi Symbian .....	36
c.    Sistem Operasi iOS .....	37
d.    Sistem Operasi Android .....	37
e.    Sistem Operasi Blackberry.....	38
f.    Pengembangan Aplikasi Mobile (WEB APP) .....	38
1.5.    Arsitektur dan Perangkat Mobile .....	40
a.    Arsitektur iOS .....	40
b.    Arsitektur Android .....	42

c.	Arsitektur Palm OS .....	45
d.	Arsitektur Symbian OS .....	47
1.6.	Perangkat Mobile dan Penggunaan Umum.....	49
a.	Perangkat Mobile .....	49
b.	Penggunaan Umum Perangkat Mobile.....	53
<b>Bab 2:</b>	<b>Pengantar Aplikasi Android .....</b>	<b>59</b>
2.1.	Sejarah Perkembangan.....	59
2.2.	Versi Android.....	65
2.3.	Pembuatan Aplikasi Android .....	84
2.4.	Modul, Struktur File dan Pengujian Project Android .....	95
2.5.	Project dan Struktur File Android .....	96
2.6.	Instalasi Android Studio.....	104
2.7.	Menjelajahi proyek .....	108
2.8.	Menjalankan aplikasi pada emulator atau perangkat .....	119
2.9.	Pemecahan masalah koneksi perangkat .....	122
Latihan Bab 2 .....		125
<b>Bab 3:</b>	<b>Layout User Interface (UI).....</b>	<b>146</b>
3.1.	User Interface.....	146
3.2.	Layout User Interface.....	150
3.3.	Jenis-Jenis Layout .....	161
a.	Linier Layout.....	161
b.	Relative Layout .....	167
c.	Tabel Layout .....	171
d.	Frame Layout .....	175
e.	Constraint Layout.....	177
f.	List View .....	183
Latihan Bab 3 .....		190
<b>Bab 4:</b>	<b>Kontrol User Interface.....</b>	<b>213</b>
4.1.	Kontrol Input.....	213
4.2.	Jenis-Jenis Kontrol .....	216
a.	TextView.....	216
b.	Button (Tombol) .....	230

c.	Kotak centang (CheckBox).....	245
d.	Tombol radio (RadioButton).....	247
e.	Switch dan tombol beralih (ToggleButton).....	249
f.	Spinner .....	252
g.	Picker Tanggal Dan Waktu .....	262
h.	Mengenali Isyarat.....	269
Latihan Bab 4 .....		272
<b>Bab 5: Activity dan Intent .....</b>		<b>348</b>
5.1.	Tentang Activity .....	348
5.2.	Tentang Intent .....	352
a.	Navigasi Activity .....	362
b.	Tentang Intent Implisit.....	367
c.	Mode Peluncuran Activy .....	378
Latihan Bab 5 .....		381
<b>Bab 6: Shared Preference, File dan Penyimpanan Data.....</b>		<b>410</b>
6.1.	Shared Preference .....	410
6.2.	Penyimpanan Internal dan Eksternal.....	417
a.	Penyimpanan Internal .....	417
b.	Penyimpanan Eksternal.....	420
6.3.	Database SQLite .....	424
6.4.	Cursor dan Model Data .....	429
a.	Cursor.....	429
b.	Model Data.....	431
c.	Mencadangkan Database dan Aplikasi .....	434
d.	Opsi Storage Lain.....	435
e.	Mencadangkan Data Aplikasi .....	435
Latihan Bab 6 .....		437
<b>Bab 7: Aplikasi CRUD Database.....</b>		<b>462</b>
7.1.	Aplikasi Database SQLite .....	462
7.2.	SQLiteOpenHelper.....	463
7.3.	Mengimplementasikan Database SQLite .....	464
7.4.	Operasi Database.....	468

Latihan Bab 7 .....	576
<b>Bab 8: Pengembangan Aplikasi Adroid dengan Kotlin</b> .....	601
8.1. Kotlin - Android.....	601
8.2. Konfigurasi Kotlin ke Android Studio .....	604
8.3. Kotlin Extensions .....	607
8.4. Kotlin dari Dasar: Packages dan Function Dasar.....	610
a. Packages.....	610
b. Functions.....	611
Latihan Bab 8 .....	620
<b>PENUTUP</b> .....	665

# DAFTAR GAMBAR



Gambar 1. Arsitektur Sistem Operasi iOS .....	40
Gambar 2. Arsitektur Sistem Operasi Android .....	43
Gambar 3. Arsitektur Palm OS .....	46
Gambar 4. Arsitektur Symbian OS.....	47
Gambar 5. Kacamata digital.....	50
Gambar 6. Notebook – Laptop.....	50
Gambar 7. Handphone dan Smartphone.....	51
Gambar 8. Tablet.....	51
Gambar 9. Jam Tangan Digital .....	52
Gambar 10. GPS.....	52
Gambar 11. Antarmuka Android.....	59
Gambar 12. Ikon aplikasi pada layar utama .....	60
Gambar 13. Arsitektur Framework Android Development.....	62
Gambar 14. Diagram proses development high-level.....	84
Gambar 15. Screenshot antarmuka welcome to android studio .....	85
Gambar 16. Antarmuka IDE aplikasi.....	86
Gambar 17. Proses Pembangunan Project Android .....	88
Gambar 18. Struktur Project Modul Android.....	91
Gambar 19. Menyinkronkan project dengan file konfigurasi .....	93
Gambar 20. Project File Android Studio.....	97
Gambar 21. Antarmuka pemilihan deployment target .....	100
Gambar 22. Tampilan konfigurasi Android Virtual Device.....	103
Gambar 23. SDK Platforms .....	103
Gambar 24. Screenshot Target Android Studio device.....	105
Gambar 25. Screenshot layar Customize Activity .....	106
Gambar 26. Screenshot jendela utama Android Studio.....	107

Gambar 27. Menjelajahi project yang telah di-create .....	108
Gambar 28. Tampilan desain layout.....	114
Gambar 29. Folder gradle.....	115
Gambar 30. Create virtual device.....	119
Gambar 31. Pemilihan device definition.....	120
Gambar 32. Running emulator .....	121
Gambar 33. Pesan log.....	124
Gambar 34. Download Android Studio.....	126
Gambar 35. Pilihan Pemasangan Instalasi .....	127
Gambar 36. Proses Instalasi .....	127
Gambar 37. Proses Instalasi Lanjutan .....	127
Gambar 38. Antarmuka Android SDK Manager.....	128
Gambar 39. Tampilan Toolbar Icon SDK Manager .....	129
Gambar 40. Tampilan Setting SDK Paltform .....	129
Gambar 41. Jendela Android Studio .....	134
Gambar 42 Level log default.....	139
Gambar 43. Tampilan pernyataan log .....	141
Gambar 44. Interface Android Studio .....	147
Gambar 45. Menu Bar .....	147
Gambar46. Toolbar .....	147
Gambar 47. Struktur Project.....	148
Gambar 48. Palette .....	148
Gambar 49. Desain Android.....	149
Gambar 50. Component Tree .....	149
Gambar 51. Properties.....	150
Gambar 52. Status bar .....	150
Gambar 53. Hierarki View Layout Parameter.....	158
Gambar 54. Contoh tampilan linear layout vertical .....	162
Gambar 55. Layout horizontal dan vertikal.....	164
Gambar 56. Layout_gravity center dan layout_gravity right .....	165
Gambar 57. Weight pada orientation horizontal .....	166
Gambar 58. Hasil tampilan relative layout.....	170



Gambar 59. Tabel Layout.....	174
Gambar 60. FrameLayout .....	176
Gambar 61. Penyesuaian ukuran tampilan .....	181
Gambar 62. TextView pada item ListView.....	184
Gambar 63. Item pada Custom ListView .....	185
Gambar 64. Folder drawable-hdpi.....	186
Gambar 65. ArrayListview .....	189
Gambar 66. Sketsa UI sederhana .....	192
Gambar 67. Layout Editor.....	193
Gambar 68. Kontrol user interface .....	213
Gambar 69. Hello world.....	217
Gambar 70. Scrollview.....	220
Gambar 71. Scrollview orientasi vertikal.....	221
Gambar 72. ScrollView dengan teks artikel.....	222
Gambar 73. Grup tampilan: subjudul dan artikel .....	223
Gambar 74. Tombol Return pada Keyboard .....	225
Gambar 75. Masukan pesan .....	226
Gambar 76. Tombol .....	230
Gambar 77. Tiga keadaan: normal, dinonaktifkan, dan ditekan.....	230
Gambar 78. Dialog Configure Image Asset .....	233
Gambar 79. Clipart.....	234
Gambar 80. Tombol datar .....	237
Gambar 81. ImageView .....	238
Gambar 82. Tombol aksi mengambang .....	240
Gambar 83. OnClickListener .....	242
Gambar 84. Kotak centang muncul dalam daftar vertikal.....	245
Gambar 85. Tombol radio (RadioButton).....	247
Gambar 86. ToggleButton.....	249
Gambar 87. Switch on-off.....	249
Gambar 88. Switch.....	251
Gambar 89. Spinner.....	252
Gambar 90. Pola spinner-adapter .....	253

Gambar 91. Dialog .....	257
Gambar 92. Picker .....	258
Gambar 93. Subkelas AlertDialog .....	259
Gambar 94. Picker Tanggal Dan Waktu .....	262
Gambar 95. Activity .....	348
Gambar 96. Siklus Activity .....	363
Gambar 97. Tombol Kembali (BackStack) .....	364
Gambar 98. Up Navigation .....	365
Gambar 99. Intent Implicit .....	367
Gambar 100. Tampilkan Pemilihan Aplikasi .....	371
Gambar 101. Mengelola Task dan Activity .....	376
Gambar 102. Launch Activity .....	377
Gambar 103. Alur Databas SQLite .....	464
Gambar 104. Cara Kerja Database SQLite .....	464
Gambar 105. Operasi Database .....	468
Gambar 106. SQLiteDatabase .....	473
Gambar 107. Room Database .....	505
Gambar 108. Install Plugin Kotlin Pada Android Studio .....	605

# DAFTAR TABEL



Tabel 1. Perkembangan Versi Android .....	65
Tabel 2. Modul Aplikasi Android .....	97
Tabel 3. Struktur Project Android .....	98
Tabel 4. Original Equipment Manufacturer (OEM).....	101
Tabel 5. Atribut Layout.....	154
Tabel 6. Contant .....	155
Tabel 7. Linear Layout Attributes .....	161
Tabel 8. Tag- tag pada Relative Layout .....	167
Tabel 9. Kontrol dan deskripsi penggunaan .....	213
Tabel 10. Atribut TextView .....	217
Tabel 11. Atribut button yang umum disetting: .....	231
Tabel 12. Preferensi bersama vs. keadaan instance tersimpan .....	411
Tabel 13. Database_Name.....	424
Tabel 14. Kueri contoh.....	426

# DAFTAR LATIHAN



Latihan 2.1. Memasang Android Studio .....	126
Latihan 2.2. Membuat Aplikasi "Hello World" .....	133
Latihan 2.3. Jelajahi Struktur Proyek .....	135
Latihan 2.4. Membuat perangkat virtual (emulator) .....	137
Latihan 2.5. Menambahkan pernyataan log ke aplikasi Anda.....	139
Latihan 3.1. Membuat proyek "Hello Toast": .....	191
Latihan 3.2. Menambahkan tampilan ke "Hello Toast" .....	192
Latihan 3.3. Menjelajahi layout editor: .....	193
Latihan 3.4. Menambahkan ke Linear Layout dalam Layout Editor ....	196
Latihan 3.5. Mengedit Layout "Hello Toast" dalam XML .....	197
Latihan 3.6. Gravitasi dan berat: .....	201
Latihan 3. 7. Menampilkan toast saat tombol Toast diklik .....	205
Latihan 4.1. Menambahkan beberapa tampilan teks	274
Latihan 4.2. Menambahkan tautan Web aktif dan ScrollView .....	279
Latihan 4.3. Menggulir beberapa elemen .....	284
Latihan 4.4. Bereksperimen dengan atribut keyboard entri teks .....	287
Latihan 4.5. Mengubah tipe keyboard.....	292
Latihan 4.6. Menambahkan Kontrol Spinner .....	293
Latihan 4.7. Menggunakan dialog untuk peringatan .....	303
Latihan 4.8. Menggunakan picker untuk masukan pengguna .....	309
Latihan 4.9. Buat fragmen baru untuk picker tanggal .....	313
Latihan 4.10. Menggunakan tampilan gambar sebagai tombol.....	318
Latihan 4.11. Menggunakan tombol radio .....	325

Latihan 4.12. Menambahkan item ke menu opsi.....	330
Latihan 4.13. Menangani item menu yang dipilih.....	341
Latihan 5.1. Membuat project TwoActivities .....	384
Latihan 5.2. Membuat dan meluncurkan activity kedua .....	387
Latihan 5.3. Mengirim data dari activity utama ke activity kedua.....	391
Latihan 5.4. Mengembalikan data ke activity utama.....	397
Latihan 5.5. Mendapatkan balasan dari ekstra intent .....	404
Latihan 6.1. Menjelajahi HelloSharedPreferences .....	439
Latihan 6.2. Menyimpan data ke file preferensi bersama .....	441
Latihan 6.3. Penyimpanan Internal: .....	451
Latihan 6.4. Database SQLite .....	455
Latihan 7.1. Aplikasi Pendataan data Mahasiswa .....	474
Latihan 7.2. SQLite Room Database.....	504
Latihan 7.3. Membuat Read Detail Data pada Room Database .....	535
Latihan 7.4. Set Up Firebase Realtime Database .....	542
Latihan 7.5. Latihan Membuat Login dengan SQLite Database .....	578
Latihan 8.1. Aplikasi pertama dengan Kotlin.....	622
Latihan 8.2. Membuat Aplikasi Input dan Output Kotlin Android .....	625
Latihan 8.3. Membuat aplikasi konversi suhu sederhana.....	634



## MATERI KOMPETENSI

Materi bab ini akan membahas:

- 1) Konsep dasar dan sejarah perkembangan pemrograman mobile
- 2) Sistem operasi pemrograman mobile
- 3) Arsitektur sistem operasi mobile
- 4) Jenis perangkat dan penggunaan umum perangkat mobile.

## YANG HARUS SUDAH ANDA KETAHUI

Untuk materi dan praktik ini Anda harus bisa:

- 1) Memahami arsitektur organisasi komputer dan sistem operasi.
- 2) Memiliki pengalaman telah mengikuti pembelajaran Perkembangan Teknologi Informatika (PTI), Arsitektur dan Organisasi Komputer, dan Sistem Operasi (OS).

## YANG AKAN ANDA BUTUHKAN

Untuk materi dan praktik ini, Anda akan membutuhkan:

- 1) Akses internet-cara alternatif untuk mengeksplorasi materi perkembangan mobile.
- 2) Perangkat akses materi seperti laptop, PC maupun Smartphone.

## YANG AKAN ANDA PELAJARI

Anda akan belajar:

- 1) Memahami dan mengetahui sejarah perkembangan pemrograman mobile.
- 2) Memahami berbagai jenis sistem operasi pemrograman mobile.
- 3) Memahami dan mengetahui arsitektur sistem operasi mobile.

## YANG HARUS ANDA LAKUKAN

- 1) Mencari referensi terkait materi perkembangan dan sistem operasi mobile.
- 2) Membuat rangkuman materi perkembangan perangkat lunak dan keras mobile.



## Bab 1: Pengantar Pemrograman Mobile

### 1.1. Konsep Dasar

Pemrograman Mobile berasal dari 2 buah kata, yaitu mobile dan program. Mobile adalah perangkat yang bergerak atau bisa dibawa ke mana saja dengan mudah, contohnya HP atau laptop sedangkan pemrogramang mempunyai arti program atau programing. Jadi pemrograman mobile adalah pemrograman yang digunakan untuk perangkat bergerak.

Sejarah mengenai mobile programming, program game snake pada Mobile Phone Nokia sekitar pertengahan tahun 1990-an. Game ini tampilannya hanya berbentuk sebuah field yang terdapat seekor ular berbentuk kotak memanjang dan kita harus mengejar target untuk mendapat point, sangat sederhana. Tapi ini merupakan suatu milestone tersendiri bagi sejarah Mobile Phone.

Game snake adalah aplikasi pertama yang ada pada sebuah Mobile Phone diluar dari fungsi utamanya sebagai alat berkomunikasi. Penciptaan game ini mengilhami produsen-produsen Mobile Phone untuk membuat berbagai aplikasi pendukung. Terciptalah berbagai fungsi lain pada Mobile Phone, dari yang sederhana seperti kalkulator, alarm, kalender, stopwatch, converter, notes, to-do list, dan lain lain. Sampai pada aplikasi yang lebih kompleks seperti GPS, wifi, kamera recorder, aplikasi multimedia, dan lain-lain. Kemudian didukung lagi oleh konektivitas nirkabel seperti IrDa dan Bluetooth. Kemudian diluncurkannya sistem operasi Symbian, yang membuat kitabebas untuk mengembangkan aplikasi sendiri dengan bahasa Java atau Python.

Sangat cepat perkembangan mobile phone ini, dari pertama kali hanya berfungsi sebagai alat telekomunikasi, sekarang menjadi sebuah gadget yang multifungsi. Tidak dapat dipungkiri pada masa mendatang sebuah mobile phone dapat menjadi gadget dengan berbagai macam fungsi, sehingga dapat menjadi gadget tunggal untuk manusia.



## 1.2. Sejarah Perkembangan

Perkembangan mobile komputer hingga kini sangatlah luar biasa. Hal ini tidak lepas dari perkembangan-perkembangan mobile komputer dari waktu ke waktu. Berikut dibahas sedikit mengenai sejarah perkembangan mobile komputer, yakni:

- **Dynabook (1968)**

Alan kay mengembangkan sebuah komputer portabel yang nantinya menawarkan sebuah media elektronik yang modern yang diperuntukan untuk anak anak. Graphical control interface dengan icon pada Komputer ini merupakan cikal bakal dari system operasi yang ada saat ini.

- **Gir Compass 1100(1982)**

Teknologi ini di rancang dan di kembangkan oleh gird sangat terdepan. Cover nya dapat dilipat dan terintegrasi sebuah flatscreen serta casing dari magnesium yang ultralight. Namun ini di teruskan dan dikembangkan oleh bil moggridge ini terlalu mahal dengan harga mencapai dengan US\$ 10.000. hanya kalangan militer AS dan NASA saja yang dapat membelinya.

- **Oseborne 1 (1981)**

IBM 1311 diperkenalkan pada tanggal 11 oktober 1962 hardisk ini bisa menyimpan sampai dengan 2 juta karakter pada diskpack yang dapat di ganti (1316). Ketebalan hardisk ini mencapai 4 inci dan berat 4,5 kg dan memiliki 6 disk yang berukuran 6 disk yang berukuran 14 inci dan permukaan yang dapat ditulis.

- **Compaq Portable (1983)**

Laptop pertama yang kompetibel dengan IBM ditawarkan oleh Compaq. Berat dan tanpa baterai lantaran masih menggunakan system operasi MS DOS laptop ini tidak populer.

- **Gavilan SC (1984)**

Touchpad pertama ini ditawarkan oleh laptop Gavilan dan berada di atas keyboard. Model komputer ini sudah dilipat dan menyediakan layar LCD serta processor intel 80168 yang kompetibel dengan harga MS DOS.

- **Bondwell 2 (1985)**

Walaupun era CP /M yang sudah lewat tetapi kehadiran ini tetap sukses dengan RAM sebesar 64 Kb floppy drive 3.5 inchi dan layer LCD laptop ini menjadi jawara di eranya.

- **COMPAQ SLT 286 (1988)**

Laptop pertama yan sudah dilengkapi dengan batrai hardisk dan LCD. Layarnya menawarkan resolusi VGA penuh. Notebook ini ditujukan untuk kalangan bisnis. Tidak heran harganya mencapai harga mobil pada saat itu.

- **Sharp Multicolor 386 PC 8041(1990)**

Dengan spesifikasi:

- Prosesor: Intel 80386
- Frekuensi clock 20 mhz
- Ram 2MB
- Harddrive 40 mb

- **USB interface (1997)**

Setelah penganalan port interface USB hamper semua perangkat terhubung dengan notebook computer portable mendadak menjadi universal.

- **WLAN untuk semua (1999)**

Berkat penggunaan WLAN notebook kini tidak lagi statis hotspot menghubungkan perangkat ini ke internet dan jaringan office serta dapat diakses dimana saja.

### 1.3. Sistem Operasi

Sistem operasi perangkat mobile (bahasa Inggris: mobile operating system, disingkat Mobile OS) adalah suatu sistem operasi yang mengontrol sistem dan kinerja pada barang elektronik yang mobile, mirip dengan fungsi Windows, Mac OS X, dan Linux pada Desktop PC atau Laptop/Notebook tetapi lebih sederhana. Biasanya penggunaannya hadir di ponsel pintar, PDA, tablet komputer, dan PMP.

Pada tahun 1993 dirilis smartphone yang pertama, “IBM Simon” yang memiliki fitur layar sentuh, email serta PDA. Selang 3 tahun kemudian, tahun 1996 Palm Pilot 1000 memperkenalkan personal digital assistant (PDA) untuk pertama kalinya dengan sistem operasi Palm OS. Masih dalam tahun yang sama PC Handled untuk pertama kalinya diperkenalkan dengan sistem Windows CE. 4 tahun, kemudian, tahun 2000 Symbian menjadi sistem operasi mobile modern pertama pada smartphone yang diluncurkan oleh Ericsson R380. Tahun 2002, Microsoft pertama Windows CE (Pocket PC) pada smartphone. Masih pada tahun 2002, Blackberry merilis smartphone pertamanya. 3 tahun kemudian, tahun 2005 Nokia memperkenalkan Maemo OS pada tablet internet pertama Nokia N770. Lalu tahun 2007, Apple iPhone dengan iOS diperkenalkan sebagai iPhone “mobile phone” dan “internet communicator”. Setahun kemudian, tahun 2008 OHA yang dibentuk oleh Google merilis Android 1.0 dengan HTC Dream (T-Mobile G1) sebagai ponsel Android yang pertama. Sistem operasi yang umum saat ini:

#### a. Android OS

Android dari Google Inc. (open source, Apache) Sistem operasi yang berbasis linux dan banyak digunakan oleh para pembuat/pabrikasi telepon genggam sekarang ini (Samsung, HTC, Motorola, Sony Ericsson, LG, Huawei, dll). Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008.

Antarmuka pengguna Android umumnya berupa manipulasi langsung, menggunakan gerakan sentuh yang serupa dengan tindakan nyata, misalnya menggeser, mengetuk, dan mencubit untuk memanipulasi objek di layar, serta papan ketik virtual untuk menulis teks. Selain perangkat layar sentuh, Google juga telah mengembangkan Android TV untuk televisi, Android Auto untuk mobil, dan Android Wear untuk jam tangan, masing-masingnya memiliki antarmuka pengguna yang berbeda. Varian Android juga digunakan pada komputer jinjing, konsol permainan, kamera digital, dan peralatan elektronik lainnya.

Android adalah sistem operasi dengan sumber terbuka, dan Google merilis kodenya di bawah Lisensi Apache. Kode dengan sumber terbuka dan lisensi perizinan pada Android memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi. Selain itu, Android memiliki sejumlah besar komunitas pengembang aplikasi (apps) yang memperluas fungsionalitas perangkat, umumnya ditulis dalam versi kustomisasi bahasa pemrograman Java. Pada bulan Oktober 2013, ada lebih dari satu juta aplikasi yang tersedia untuk Android, dan sekitar 50 miliar aplikasi telah diunduh dari Google Play, toko aplikasi utama Android. Sebuah survei pada bulan April-Mei 2013 menemukan bahwa Android adalah platform paling populer bagi para pengembang, digunakan oleh 71% pengembang aplikasi bergerak. Di Google I/O 2014, Google melaporkan terdapat lebih dari satu miliar pengguna aktif bulanan Android, meningkat dari 583 juta pada bulan Juni 2013.

Faktor-faktor di atas telah memberikan kontribusi terhadap perkembangan android, menjadikannya sebagai sistem operasi telepon pintar yang paling banyak digunakan di dunia, mengalahkan Symbian pada tahun 2010. Android juga menjadi pilihan bagi perusahaan teknologi yang menginginkan sistem operasi berbiaya rendah, bisa dikustomisasi, dan ringan untuk perangkat berteknologi tinggi tanpa harus mengembangkannya dari awal. Sifat Android yang terbuka juga telah mendorong munculnya sejumlah besar komunitas pengembang aplikasi untuk menggunakan kode sumber terbuka sebagai dasar proyek pembuatan aplikasi, dengan menambahkan fitur-fitur baru bagi pengguna tingkat lanjut atau mengoperasikan Android pada perangkat yang secara resmi dirilis dengan menggunakan sistem operasi lain.

Pada November 2013, Android menguasai pangsa pasar telepon pintar global, yang dipimpin oleh produk-produk Samsung, dengan persentase 64% pada bulan Maret 2013. Pada Juli 2013, terdapat 11.868 perangkat Android berbeda dengan beragam versi. Keberhasilan sistem operasi ini juga menjadikannya sebagai target litigasi paten "perang telepon pintar" antar

perusahaan-perusahaan teknologi. Hingga bulan Mei 2013, total 900 juta perangkat Android telah diaktifkan di seluruh dunia, dan 48 miliar aplikasi telah dipasang dari Google Play.

Antarmuka pengguna pada Android didasarkan pada manipulasi langsung, menggunakan masukan sentuh yang serupa dengan tindakan di dunia nyata, misalnya menggesek (*swiping*), mengetuk (*tapping*), dan mencubit (*pinching*), untuk memanipulasi objek di layar. Masukan pengguna direspon dengan cepat dan juga tersedia antarmuka sentuh layaknya permukaan air, seringkali menggunakan kemampuan getaran perangkat untuk memberikan umpan balik haptik kepada pengguna. Perangkat keras internal seperti akselerometer, giroskop, dan sensor proksimitas digunakan oleh beberapa aplikasi untuk merespon tindakan pengguna, misalnya untuk menyesuaikan posisi layar dari potret ke lanskap, tergantung pada bagaimana perangkat diposisikan, atau memungkinkan pengguna untuk mengarahkan kendaraan saat bermain balapan dengan memutar perangkat sebagai simulasi kendali kontrol.

Ketika dihidupkan, perangkat Android akan memuat pada layar depan (*homescreen*), yakni navigasi utama dan pusat informasi pada perangkat, serupa dengan desktop pada komputer pribadi. Layar depan Android biasanya terdiri dari ikon aplikasi dan widget; ikon aplikasi berfungsi untuk menjalankan aplikasi terkait, sedangkan widget menampilkan konten secara langsung dan terbaru otomatis, misalnya prakiraan cuaca, kotak masuk surel pengguna, atau menampilkan tiker berita secara langsung dari layar depan. Layar depan bisa terdiri dari beberapa halaman, pengguna dapat menggeser bolak balik antara satu halaman ke halaman lainnya, yang memungkinkan pengguna Android untuk mengatur tampilan perangkat sesuai dengan selera mereka. Beberapa aplikasi pihak ketiga yang tersedia di Google Play dan di toko aplikasi lainnya secara ekstensif mampu mengatur kembali tema layar depan Android, dan bahkan bisa meniru tampilan sistem operasi lain, misalnya Windows Phone. Kebanyakan produsen telepon seluler dan operator nirkabel menyesuaikan tampilan perangkat Android buatan mereka untuk membedakannya dari pesaing mereka.

Di bagian atas layar terdapat status bar, yang menampilkan informasi tentang perangkat dan konektivitasnya. Status bar ini bisa "ditarik" ke bawah untuk membuka layar notifikasi yang menampilkan informasi penting atau pembaruan aplikasi, misalnya surel diterima atau SMS masuk, dengan cara tidak mengganggu kegiatan pengguna pada perangkat. Pada versi awal android, layar notifikasi ini bisa digunakan untuk membuka aplikasi yang relevan, namun setelah diperbarui, fungsi ini semakin disempurnakan, misalnya kemampuan untuk memanggil kembali nomor telepon dari notifikasi panggilan tak terjawab

tanpa harus membuka aplikasi utama. Notifikasi ini akan tetap ada sampai pengguna melihatnya, atau dihapus dan di nonaktifkan oleh pengguna.

## b. BlackBerry OS

BlackBerry menggunakan BlackBerry OS sebagai sistem operasinya. BlackBerry OS, dikembangkan oleh Reseach in Motion (RIM), untuk BlackBerry, perusahaan telekomunikasi asal Kanada ini. Diawal kemunculannya BlackBerry booming dengan layanan push-email dan sebagai smartphone yang sukses dipasaran. Versi terakhir dari BlackBerry OS adalah BlackBerry OS 7. Sistem operasi menyediakan multitasking dan mendukung perangkat input khusus yang telah diadopsi oleh RIM untuk digunakan dalam handheld, khususnya trackwheel, trackball, dan yang paling baru, yang trackpad dan touchscreen.

Platform BlackBerry mungkin paling dikenal karena dukungan asli untuk email perusahaan, melalui MIDP 1.0 dan, baru-baru ini, sebuah subset dari MIDP 2.0, yang memungkinkan aktivasi nirkabel lengkap dan sinkronisasi dengan Microsoft Exchange, Lotus Domino, atau Novell GroupWise email, kalender, tugas, catatan, dan kontak. Sistem operasi juga mendukung WAP 1.2.

Pembaruan sistem operasi dapat secara otomatis tersedia dari operator nirkabel yang mendukung BlackBerry atas memuat perangkat lunak udara (OTASL) layanan. Pengembang pihak ketiga dapat menulis perangkat lunak menggunakan BlackBerry yang tersedia API kelas, meskipun aplikasi yang menggunakan fungsi tertentu harus digital ditandatangani. Sementara RIM mengembangkan dan merilis versi update dari sistem operasinya untuk mendukung masing-masing perangkat, terserah kepada operator masing-masing untuk memutuskan jika dan ketika versi dilepaskan untuk para penggunanya. Pada April 2010 RIM mengumumkan BlackBerry OS baru versi 6.0, yang dirilis pada kuartal 3 2010.

Desas-desus dari 6,1 OS mulai tahun 2011 bersama dengan gambar perangkat lunak baru. Pada 2 Mei 2011, RIM kemudian mengumumkan sebagai BlackBerry OS 7. Ini akan dirilis pada musim panas 2011. RIM mengumumkan bahwa perangkat saat ini tidak akan diperbarui untuk BlackBerry OS 7 (Tidak Dukungan Perangkat Legacy). RIM juga mengumumkan rilis Touch BlackBerry Bold (Blackberry Bold 9930 & 9900) yang akan menjalankan Blackberry OS.

### c. iOS Apple Inc.

Apple Inc. (sebelumnya bernama Apple Computer, Inc.) adalah sebuah perusahaan multinasional yang berpusat di Silicon Valley, Cupertino, California dan bergerak dalam bidang perancangan, pengembangan, dan penjualan barang-barang yang meliputi elektronik konsumen, perangkat lunak komputer, serta komputer pribadi. Apple Inc. Didirikan pada tanggal 1 April 1976 dan diinkorporasikan menjadi Apple Computer, Inc. pada tanggal 3 Januari, 1977. Pada 9 Januari, 2007, kata "Computer" dihapus untuk mencerminkan fokus Apple terhadap bidang elektronik konsumen pascapeluncuran iPhone.

Apple dikenal akan jajaran produk perangkat lunak diantaranya sistem operasi OS X dan iOS, pemutar musik iTunes, serta peramban web Safari, dan perangkat keras diantaranya komputer meja iMac, komputer jinjing MacBook Pro, pemutar lagu iPod, serta telepon genggam iPhone dan jam tangan pintar Apple Watch.

Sebelum membantu mendirikan Apple, Steve Wozniak adalah seorang hacker elektronik. Pada tahun 1975, dia bekerja di Hewlett-Packard dan membantu temannya Steve Jobs mendisain video game untuk Atari. Pada saat itu, Wozniak telah membeli computer time pada bermacam jenis minicomputer yang di-host oleh Call Computer, sebuah perusahaan timesharing. Terminal komputer pada saat itu mayoritas berbasis kertas; thermal printer seperti Texas Instruments Silent 700 adalah terminal yang paling maju. Wozniak melihat sebuah artikel di majalah bernama Popular Electronics edisi 1975 yang berisikan bagaimana cara untuk membuat terminal komputer sendiri. Menggunakan alat-alat yang dapat dibeli di toko, Wozniak merancang dan membuat komputer bernama Computer Conversor, sebuah video teletype yang bisa dipakai untuk mengakses minicomputer di Call Computers. Disainnya kemudian dibeli oleh Call Computers dan beberapa unit terjual.

Pada tahun 1975 Wozniak mulai menghadiri pertemuan sebuah klub bernama Homebrew Computer Club. Microcomputer baru seperti Altair 8800 dan IMSAI menginspirasinya untuk memasang sebuah mikroprosesor pada video teletype-nya untuk menjadikannya sebuah unit komputer yang lengkap.

Di awal tahun 2002, Apple mengeluarkan iMac G4. Komputer ini memiliki bagian dasar yang separuh bulat dan layar datar yang ditopang dengan leher yang bisa digerakkan. Model ini diberhentikan di musim panas 2004. Sebuah model yang baru yang menggunakan prosesor prosesor G5 dilepas pada tanggal 31 Agustus 2004 dan mulai dijual di pertengahan September pada tahun yang sama. Model ini meniadakan bagian dasar dan mengemas seluruh unit proses pusatnya (central processing unit) di belakang layar komputer yang ditopang oleh sebuah kaki aluminium yang ramping. Komputer yang dijuluki iMac G5

ini adalah komputer meja tertipis di dunia, dengan ketebalan yang hanya dua inci atau 5,1cm.

Di pertengahan 2003, Apple mengeluarkan prosesor PowerPC generasi kelima yang menggunakan seri prosesor PowerPC 970 yang dikembangkan oleh IBM. Ini adalah komputer 64bit yang pertama kali tersedia untuk umum. Pada tahun 2001, Apple memperkenalkan Mac OS X, sebuah sistem operasi yang didasarkan dari NeXTstep yang menggabungkan stabilitas, kehandalan dan keamanan yang dimiliki UNIX dengan kemudahan dalam menggunakan Macintosh ke dalam sebuah piranti lunak yang bisa dipakai baik bagi mereka yang bergerak di bidang profesional maupun perseorangan. Mac OS X juga memiliki sebuah program bernama Classic Environment yang bisa menjalankan piranti lunak yang dirancang untuk sistem operasi 9.1-9.2.2 yang lama. Apple juga memberikan pilihan bagi pengembang piranti lunak yang lama untuk menggunakan teknologi Carbon agar program mereka bisa diadaptasi secara langsung ke dalam Mac OS X. Dengan cara ini, sebuah program bisa memanfaatkan semua fitur baru yang tersedia dalam Mac OS X.

Pada Januari 2007, Apple memasuki pasar telepon genggam dengan memperkenalkan iPhone, yang dirilis pada tanggal 29 Juni 2007 di AS. Pada saat yang sama nama perusahaan juga akan dipangkas dengan menanggalkan kata "Computer" untuk mewakili diversifikasi produk perusahaan tersebut. Pada tahun berikutnya Apple memperkenalkan iPhone 5 dengan layar yang sedikit lebih besar dari pendahulunya, iPhone 4S. Walau banyak orang meragui akan masa depan Apple pasca di tinggalkan Steve Jobs, namun dengan percaya diri Tim Cook mulai menunjukkan kehebatan dirinya dengan menaikkan harga saham perusahaan yang mencapai \$768 per lembar. Dan pada tahun 2013, Apple memperkenalkan dua unit iPhone sekaligus, iPhone 5S dengan fitur sidik jari dan iPhone 5C dengan berbagai warna-warni. kemudian pada di ajang WWDC 2014 Tim Cook memperkenalkan 2 buah unit iPhone dengan layar jumbo yang di sebut sebagai iPhone 6 (4.7inc) & iPhone 6 plus (5.5inc). selain duo iPhone, Tim Cook juga meluncurkan Apple Watch sebagai jam tangan pintar dan sistem pembayaran digital yang dinamakan Apple Pay.

#### d. Symbian OS

Symbian adalah sistem operasi yang dikembangkan oleh Symbian Ltd. yang dirancang untuk peralatan bergerak (mobile). Sebelum Nokia mengumumkan pembelian seluruh sisa saham Symbian Ltd. yang tidak dimilikinya pada 24 Juni 2008,[6] Symbian dimiliki Nokia (47,9%), Ericsson (15,6%), Panasonic (10,5%), Samsung (4,5%), Siemens/BenQ (8,4%), Sony Ericsson (13,1%). Versi Symbian yang terbaru adalah Symbian ^3 dengan PR



02 yang disebut Symbian belle. Sedangkan ponsel yang paling banyak beredar saat ini menggunakan Symbian OS v6.1s, v7.0s, RV 47 75, v8. OS, v9.1s dan s60 v5. Nokia Nseries rata-rata menggunakan Symbian OS v9.1s, kecuali Nokia N95 yang menggunakan Symbian OS v9.2s. Ponsel Symbian s60 v5 dan Symbian ^3 rata rata menggunakan layar sentuh kecuali Nokia e6-00 dan Nokia e7. Ponsel Symbian ^3 yang pertama meluncur adalah Nokia n8 dengan kamera 12 megapiksel. Symbian ^3 mendapat pembaruan pertama yang disebut Symbian anna yang Digunakan di ponsel Nokia x7-00 dan e6-00. Sedangkan pembaruan terbaru disebut Symbian belle. Semua ponsel Symbian 3 dapat diperbaharui ke Symbian belle.

Sekarang keseluruhan saham di Symbian limited telah dimiliki oleh Nokia dan posisi sistem operasi Symbian sebagai penguasa pasar sistem operasi bergerak telah direbut oleh sistem operasi buatan Google yaitu sistem operasi android yang meluncur pada akhir tahun 2009 dengan ponsel pertamanya HTC dream g1. Vendor ponsel seperti Samsung, LG, Sony Ericsson sekarang tidak lagi menggunakan sistem operasi Symbian dan menyatakan berhenti mendukung sistem operasi ini.

Saat ini Symbian OS banyak telah banyak digunakan oleh berbagai vendor produk peralatan komunikasi mobile pada berbagai jenis produk mereka yang bervariasi. Variasi dari sisi hardware ini di mana Symbian OS diimplementasi dapat dimungkinkan karena sistem operasi ini memiliki antarmuka pemrograman aplikasi (Application Programming Interface; API). API mendukung terhadap komunikasi dan tingkah laku yang umum pada hardware yang dapat digunakan oleh objek aplikasi lain. Hal ini dimungkinkan karena API merupakan objek antarmuka yang didefinisikan pada level aplikasi, yang berisikan prosedur dan fungsi (dan juga variabel serta struktur data) yang mengelola/memanggil kernel di mana sebagai penghubung antara software dan hardware. Dengan adanya standar API ini membantu pihak pengembang untuk melakukan penyesuaian atas aplikasi yang dibuatnya agar dapat diinstal pada produk telepon bergerak yang bermacam-macam.

Mirip seperti sistem operasi desktop, Symbian OS mampu melakukan operasi secara multithreading, multitasking dan pengamanan terhadap memori. Dan semua pemrograman pada Symbian dilakukan secara event-based, artinya hardware CPU menjadi tidak aktif ketika tidak ada inputan berupa aktivitas tertentu. Namun perlu dipahami sistem operasi ini memang ditujukan untuk diinstal pada peralatan mobile dengan keterbatasan sumber daya. Multithread dan multitasking memberikan kemampuan Symbian OS untuk menjalankan lebih dari satu aplikasi sekaligus. Namun khusus ini, adanya preemptive multitasking kernel akan memberi tiap-tiap program suatu pembagian waktu

pemrosesan yang dilakukan bergantian dengan cepat sehingga tampak bagi pemakai seolah-olah proses ini dieksekusi secara bersamaan. Untuk itu telah didefinisikan penjadwalan berdasar prioritas tertentu untuk menentukan proses mana yang berjalan terlebih dahulu dan proses apa berikutnya serta berapa banyak waktu akan jadi diberi.

Symbian OS sendiri bukanlah software yang sifatnya open source secara penuh karena meskipun terdapat ketersediaan API dan dokumentasinya, yang banyak membantu pihak pengembang aplikasi untuk membuat software yang berjalan di atas sistem operasi ini, dipublikasi untuk umum namun tidak untuk kode source sendiri.

#### e. Windows Phone OS

Windows Phone adalah keluarga sistem operasi perangkat bergerak yang dikembangkan oleh Microsoft dan merupakan pengganti platform Windows Mobile, meski tidak kompatibel lagi. Tidak seperti pendahulunya, Windows Phone ditujukan pada pasar konsumen alih alih perusahaan. Sistem operasi ini pertama diluncurkan bulan Oktober 2010 dan diluncurkan di Asia pada awal 2011.

Dengan Windows Phone, Microsoft menciptakan antarmuka pengguna baru menggunakan bahasa desainnya yang diberi nama Modern Style UI. Selain itu, perangkat lunak ini terintegrasikan dengan aplikasi pihak ketiga dan berbagai layanan Microsoft, serta menetapkan persyaratan minimal untuk perangkat keras yang memakai SO ini.

Pada tanggal 11 Februari 2011 di London, CEO Microsoft Steve Ballmer dan CEO Nokia Stephen Elop mengumumkan kerja sama di antara kedua perusahaan, di mana Windows Phone akan menjadi OS utama pada ponsel genggam buatan Nokia dimasa mendatang (sebelumnya Nokia menggunakan Symbian).

Windows Phone 8 adalah generasi kedua dari sistem operasi perangkat bergerak Windows Phone dari Microsoft. Sistem operasi ini diluncurkan pada 29 Oktober 2012. Pada tahun 2013, Microsoft merilis 3 update minor (general distribution release/GDR) dari Windows Phone 8, yaitu GDR 1, GDR 2 dan GDR 3 dengan memberikan beberapa tambahan fitur dan perbaikan penggunaan. Pada tanggal 2 April 2014, di acara BUILD yang diselenggarakan di San Fransisco, Microsoft mengumumkan pembaharuan terbaru dari Windows Phone 8, yaitu Windows Phone 8.1. Pada update tersebut, Microsoft memperkenalkan fitur asisten pribadi yang diberi nama Cortana.

## f. Palm OS

Palm OS (juga dikenal sebagai Garnet OS) adalah suatu sistem operasi mobile awalnya dikembangkan oleh Palm, Inc untuk asisten digital pribadi (PDA) pada tahun 1996. Palm OS dirancang untuk kemudahan penggunaan dengan user interface touchscreen-berbasis grafis. Hal ini disediakan dengan suite aplikasi dasar untuk manajemen informasi pribadi. Kemudian versi dari OS telah diperluas untuk mendukung smartphone. Beberapa lisensi yang lain diproduksi perangkat didukung oleh Palm OS.

Setelah membeli Palm merek dagang Palm, versi saat ini berlisensi dari ACCESS dinamai Garnet OS. Pada tahun 2007, ACCESS memperkenalkan penerus ke Garnet OS, yang disebut Access Linux Platform dan pada tahun 2009, pemegang lisensi utama Palm OS, Palm, Inc, beralih dari Palm OS untuk WebOS untuk perangkat yang akan datang mereka.

### ▪ **Pencipta dan Kepemilikan**

Palm OS awalnya dikembangkan di bawah arahan Jeff Hawkins di Palm Computing, Inc Palm kemudian diakuisisi oleh US Robotics Corp, yang pada gilirannya kemudian dibeli oleh 3Com, yang membuat anak Palm sebuah perusahaan publik independen pada tanggal 2 Maret 2000. Pada bulan Januari 2002, Palm mendirikan anak perusahaan yang sepenuhnya dimiliki untuk mengembangkan dan lisensi Palm OS, yang bernama PalmSource. PalmSource kemudian dipisahkan dari Palm sebagai perusahaan independen pada tanggal 28 Oktober 2003 Palm (kemudian disebut palmOne). Menjadi lisensi reguler Palm OS, tidak lagi dalam kontrol sistem operasi.

Pada bulan September 2005, PalmSource mengumumkan bahwa mereka sedang diakuisisi oleh ACCESS. Pada bulan Desember 2006, Palm memperoleh hak abadi ke kode sumber Palm OS dari ACCESS. Dengan ini Palm dapat memodifikasi sistem operasi berlisensi yang diperlukan tanpa membayar royalti lebih lanjut kepada AKSES. Bersama dengan perolehan 2005 Mei hak penuh untuk nama merek Palm, hanya Palm dapat mempublikasikan rilis sistem operasi dengan nama 'Palm OS'.

Sebagai konsekuensinya, pada 25 Januari 2007 AKSES mengumumkan perubahan nama untuk saat ini sistem operasi mereka Palm OS, sekarang berjudul Garnet OS.

### ▪ **OS Overview**

Palm OS adalah sistem operasi proprietary mobile. Dirancang pada tahun 1996 untuk Palm Computing, Inc 's Pilot baru PDA, telah diimplementasikan pada beragam perangkat mobile, termasuk smartphone, jam tangan, game

konsol genggam, pembaca barcode dan perangkat GPS. Palm OS versi sebelumnya dari 5,0 berjalan di Motorola / Freescale prosesor Dragonball. Dari versi 5.0 dan seterusnya, Palm OS berjalan pada arsitektur prosesor berbasis ARM.

Fitur utama dari Palm OS Garnet saat ini adalah:

- Sederhana, lingkungan single-tasking untuk memungkinkan peluncuran aplikasi layar penuh dengan dasar, GUI umum mengatur
- monokrom atau layar warna dengan resolusi hingga 480×320 pixel
- handwriting recognition sistem input disebut Graffiti 2
- HotSync teknologi untuk sinkronisasi data dengan komputer desktop
- Sound playback dan kemampuan merekam
- Wikipedia model keamanan: Device dapat dikunci dengan password, catatan aplikasi sewenang-wenang dapat dilakukan swasta
- TCP / IP jaringan akses
- Serial port / USB, inframerah, Bluetooth dan koneksi Wi-Fi
- Dukungan kartu memori ekspansi
- Ditetapkan standar format data untuk aplikasi manajemen informasi pribadi untuk menyimpan, alamat tugas kalender, dan entri catatan, diakses oleh aplikasi pihak ketiga. Termasuk dengan OS juga satu set aplikasi standar, dengan yang paling relevan untuk empat operasi PIM.

## ▪ **Versi sejarah dan latar belakang teknis**

Produsen bebas untuk mengimplementasikan fitur yang berbeda dari OS dalam perangkat mereka atau bahkan menambah fitur baru. Ini sejarah versi menggambarkan versi lisensi resmi dari Palm / PalmSource / ACCESS.

### a. *Palm OS 1.0*

Palm OS 1.0 adalah versi asli hadir pada Pilot 1000 dan 5000. Versi dan semua versi sebelumnya ke Palm OS 5 didasarkan di atas kernel 68000 AMX berlisensi dari KADAK Products Ltd Meskipun kernel ini secara teknis mampu multitasking, yang “persyaratan dan kondisi lisensi yang secara khusus menyatakan bahwa mungkin Palm Lindungi API untuk membuat / memanipulasi tugas dalam OS “.

Palm OS tidak membedakan antara RAM dan penyimpanan file sistem. Aplikasi diinstal langsung ke dalam RAM dan dieksekusi di tempat. Karena tidak ada sistem berkas yang didedikasikan didukung, sistem operasi tergantung pada siklus refresh RAM konstan untuk menyimpan memori. OS mendukung 160×160 menampilkan output monokrom. User input yang dihasilkan melalui

sistem pengenalan tulisan tangan Graffiti atau opsional melalui keyboard virtual. Sistem ini mendukung sinkronisasi data ke PC lain melalui teknologi HotSync dengan sendirinya beberapa antarmuka serial. Rilis bugfix terbaru adalah versi 1.0.7. Versi 1.0 memiliki fitur aplikasi PIM klasik Alamat, Date Book, Memo Pad, dan To Do List. Juga dilengkapi kalkulator dan alat keamanan untuk menyembunyikan catatan untuk penggunaan pribadi.

b. *Palm OS 2.0*

Palm OS 2.0 diperkenalkan pada 10 Maret 1997 dengan PalmPilot Personal dan Profesional. Versi ini menambahkan jaringan TCP / IP, jaringan HotSync, dan dukungan tampilan backlight. Rilis bugfix terakhir adalah versi 2.0.5. Dua aplikasi baru, Mail dan Beban ditambahkan, dan standar aplikasi PIM telah ditingkatkan.

c. *Palm OS 3.0*

Palm OS 3.0 diperkenalkan pada tanggal 9 Maret 1998 dengan peluncuran seri Palm III. Versi ini menambahkan IrDA inframerah dan ditingkatkan dukungan font. Versi ini juga memiliki fitur update aplikasi PIM dan sebuah update untuk peluncur aplikasi.

d. *Palm OS 4.0*

Palm OS 4.0 dirilis dengan seri Palm M500 baru pada tanggal 19 Maret 2001. Versi ini menambahkan interface standar untuk sistem file akses eksternal (seperti kartu SD). Sistem file eksternal adalah perubahan radikal untuk eksekusi sebelumnya di-tempat sistem operasi. Opsional sebelumnya Mobile Internet Kit sekarang merupakan bagian dari sistem operasi. Versi 4.0 menambahkan manajer perhatian untuk mengkoordinasikan informasi dari aplikasi yang berbeda, dengan beberapa kemungkinan untuk menarik perhatian pengguna, termasuk suara, LED berkedip atau getaran. Layar warna 16-bit dan zona waktu yang berbeda yang didukung. Versi ini juga memiliki peningkatan keamanan dan UI.

e. *Palm OS 5.0*

Palm OS 5 (tidak disebut 5.0) diresmikan oleh anak perusahaan PalmSource Sawit pada bulan Juni 2002 dan pertama kali diimplementasikan pada Palm Tungsten T. Ini adalah versi pertama dirilis untuk mendukung perangkat ARM, dengan dukungan untuk aplikasi Dragonball melalui Aplikasi Palm Kompatibilitas Lingkungan (PACE) emulator. Bahkan dengan tambahan overhead PACE, aplikasi Palm biasanya berjalan lebih cepat pada perangkat

ARM dari pada hardware generasi sebelumnya. Perangkat lunak baru dapat mengambil keuntungan dari prosesor ARM dengan unit kecil kode ARM, disebut sebagai ARMlets.

Dengan dasar hardware lebih kuat, Palm OS secara substansial telah ditingkatkan untuk kemampuan multimedia. Densitas tinggi 320×320 layar didukung bersama-sama dengan pemutaran suara penuh digital dan merekam API. Palm Bluetooth terpisah stack ditambahkan bersama-sama dengan IEEE 802.11b Wi-Fi stack. Koneksi jaringan aman melalui SSL yang didukung. OS dapat disesuaikan dengan skema warna yang berbeda. Untuk Palm OS 5 PalmSource dikembangkan dan berlisensi browser web yang disebut PalmSource Web Browser, yang didasarkan pada browser NetFront 3.0 AKSES.

f. *Palm OS Cobalt*

Palm OS Cobalt (6,0) adalah penerus yang ditujukan untuk Palm OS 5. Hal ini diperkenalkan pada tanggal 10 Februari 2004, tetapi tidak ditawarkan lagi dari ACCESS (lihat bagian berikutnya). Palm OS 6.0 kemudian diganti untuk Palm OS Cobalt untuk membuat jelas bahwa versi ini awalnya tidak ditujukan untuk menggantikan Palm OS 5, yang mengadopsi nama Palm OS Garnet pada waktu yang sama. Palm OS Cobalt memperkenalkan fitur-fitur sistem operasi modern untuk sebuah sistem operasi embedded berdasarkan kernel baru dengan multitasking dan perlindungan memori, multimedia modern dan kerangka grafis (berasal dari BeOS Palm diperoleh), fitur keamanan baru, dan penyesuaian dari format file PIM untuk lebih baik bekerjasama dengan Microsoft Outlook.

g. *Third party OS enhancements*

Beberapa lisensi telah membuat modifikasi khusus ke sistem operasi. Ini bukan bagian dari versi berlisensi resmi.

- Palm mengembangkan sebuah API Bluetooth untuk eksternal Bluetooth SDIO Kartu untuk Palm OS 4.0 perangkat. Bluetooth stack kemudian dimasukkan di Palm OS 5.
- Palm ditambahkan masukan sebuah grafiti virtual area API terutama untuk perangkat Tungsten T3 mereka. API ini kemudian digantikan oleh pejabat Dynamic Input Area API di Palm OS 5.3.
- Palm ditambahkan ke Palm OS 5.4 Non-Volatile File System, dan digunakan Flash untuk penyimpanan daripada DRAM, mencegah kehilangan data dalam hal menguras baterai. Namun, ini secara fundamental mengubah cara program dieksekusi dari Jalankan-di-Tempat Palm OS sistem yang digunakan secara tradisional, dan telah menjadi sumber banyak masalah kompatibilitas, membutuhkan banyak

aplikasi untuk memiliki eksplisit NVFS menambahkan dukungan bagi mereka untuk menjadi stabil.

- Untuk perangkat mereka dilengkapi kamera Palm menambahkan API CameraLib.
- Sony ditambahkan perpustakaan untuk mendukung input JogDial tersedia pada penyelenggara Client mereka.

#### h. *Modernisasi*

Selama beberapa tahun PalmSource telah berusaha untuk menciptakan penerus modern untuk Palm OS 5 dan memiliki lisensi menerapkannya. Meskipun PalmSource dikirim Palm OS Cobalt 6.0 untuk lisensi pada bulan Januari 2004, tidak menggunakannya untuk perangkat rilis. PalmSource membuat perbaikan besar untuk Palm OS Cobalt dengan merilis Palm OS Cobalt 6.1 pada September 2004 untuk menyenangkan pemegang lisensi, tapi bahkan versi baru tidak mengarah ke perangkat produksi. Pada bulan Desember 2004, PalmSource mengumumkan strategi OS baru. Dengan akuisisi perusahaan perangkat lunak telepon selular Cina Mobilesoft, PalmSource direncanakan ke port Palm OS di atas kernel Linux, sementara masih menawarkan kedua Palm OS Garnet dan Palm OS Cobalt. PalmSource mengumumkan menghentikan upaya pengembangan semua pada setiap produk tidak berkaitan langsung dengan platform masa depan berbasis Linux-nya.

Dengan akuisisi PalmSource oleh ACCESS, Palm OS untuk Linux berubah menjadi ACCESS Platform Linux yang pertama kali diumumkan pada bulan Februari 2006. Versi awal dari platform dan kit pengembangan perangkat lunak untuk Platform Linux ACCESS secara resmi dirilis pada Februari 2007. Pada bulan November 2007, ACCESS Linux Platform belum kapal pada perangkat, namun pengembangan kit ada dan demonstrasi publik telah dipamerkan. Smartphone pertama yang menggunakan Access Linux Platform adalah perangkat Edelweiss oleh Emblaze Mobile yang dijadwalkan untuk pertengahan 2009. Palm, Inc pemegang lisensi utama dari Palm OS Garnet tidak lisensi AKSES Linux Platform untuk perangkat mereka sendiri. Sebaliknya, Palm mengembangkan sistem operasi lain berbasis Linux yang disebut Palm WebOS. Pada 11 Februari 2009 CEO Palm Ed Colligan mengatakan tidak akan ada tambahan Palm OS perangkat (kecuali Centro yang dirilis ke operator lain). Palm berfokus pada Palm WebOS dan Windows Mobile perangkat. Pada April 1, 2009 Palm mengumumkan ketersediaan Palm OS emulator untuk WebOS-nya.

## 1.4. Kelemahan dan Kelebihan Sistem Operasi Mobile

Sistem operasi pada mobile/handphone bermacam – macam, antara lain:

### a. Sistem Operasi Windows Mobile

Kelebihan Windows Mobile:

- Fitur kantor yang sangat lengkap ditambah integrasi dengan layanan Microsoft Outlook dan Exchange menjadikan Windows Mobile perangkat smartphone paling powerful bagi para pebisnis,
- Kemudahan tweaking,
- Kapabilitas multitasking menjadi andalan OS ini

Kekurangan Windows Mobile:

- Kurang user friendly sehingga membutuhkan pemahaman lebih dalam,
- Sistem operasi terkesan lambat,
- Kurang responsif serta agak boros baterai,
- Lisensi Windows Mobile adalah berbayar sehingga smartphone WinMo terkesan lebih mahal.

### b. Sistem Operasi Symbian

Kelebihan Symbian OS:

- Dirancang sebagai gabungan kemampuan komputer dengan ponsel.
- Bersifat open application environment yang memungkinkan ponsel menjadi se-buah platform bagi pengembangan berbagai aplikasi dan layanan (bahasa dan konten).
- Open standard dan interoperability, modul-modulnya mudah diimplementasikan secara fleksibel karena mendukung berbagai aplikasi teknologi dan pemrograman interfaces (APIs).
- Multitasking, yang menekankan pada object oriented dan component based.
- Flexible user interface design sehingga mudah dikembangkan dengan inovasi yang diinginkan.

Kekurangan Symbian OS:

- Interface yang statis.
- Mudah terserang malware.



### c. Sistem Operasi iOS

#### Kelebihan iOS:

- Pada konsepnya Sistem operasi ini dirancang memiliki keunggulan untuk game, koleksi game yang tersedia sangat banyak dengan kualitas hampir mendekati konsol game yang sesungguhnya serta ditunjang kualitas grafis yang diatas standar level,
- Sistem operasi stabil dan mudah digunakan bagi orang awam sekalipun,
- Browser Safari yang dimiliki sistem operasi ini juga merupakan yang terbaik sekarang.
- Responsif dan intuitif,
- Browser web terbaik saat ini,
- Pemutar musik sangat bagus dari segi antarmuka maupun kualitas suara,
- Adanya toko aplikasi online App Store yang menyediakan ribuan aplikasi dan game yang siap didownload secara langsung.

#### Kekurangan iOS:

- Tidak multitasking,
- Sistem operasi sangat tertutup dan tidak bisa dipersonalisasi,
- Proses sertifikasi untuk aplikasi baru sangat ketat,
- Aplikasi dan game hanya bisa didownload dari App Store,
- Bluetooth tidak bisa untuk proses pengiriman file,
- Entry teks kurang nyaman.

### d. Sistem Operasi Android

#### Kelebihan Android OS:

- Open Source, alias gratis, sehingga para developer aplikasi Android OS tidak perlu bayar untuk membuat aplikasi untuk Android. Tidak seperti iPhone atau Windows Mobile yg mengharuskan pada developer membayar untuk membuat aplikasinya.
- Karena pengembangannya gratis, maka aplikasinya pun rata2 gratis, sehingga tidak perlu bayar untuk men-download dan menginstall aplikasi di Android.
- User Friendly
- Stabil dan aman, tidak seperti windows yg terkadang hang atau bahkan terkena virus.
- Didukung penuh oleh google
- Harga relatif murah.
- Mudah dikustomisasi dan dimodifikasi

#### Kekurangan Android OS:

- Bagi orang yg belum pernah memakainya mungkin akan sedikit membingungkan
- Memiliki proses kerja sistem yang cukup berat, sehingga memakan RAM cukup banyak
- Terkadang apabila disandingkan dengan spesifikasi hardware yang buruk, menjadi kurang responsive
- Sistem operasi yang di custom biasanya sering tidak stabil dan kurang optimal
- Untuk melakukan update system tidaklah mudah, harus menunggu dari masing-masing vendor untuk merilis Update Versi yang terbaru atau pun kalo device lama tidak ada update untuk ke OS versi terbaru
- Batterai cukup boros

#### e. Sistem Operasi Blackberry

##### Kelebihan Blackberry OS:

- Merupakan multiple Push Email yang terbaik saat ini,
- Sinkronisasi nirkabel (tanpa kabel seperti Bluetooth dan infra red) yang lengkap,
- Sistem keamanan berlapis yang menjamin keselamatan data penting kita.

##### Kekurangan Blackberry OS:

- Browser Web masih kurang bagus,
- Ada biaya tambahan untuk menikmati layanan BlackBerry yg kita pakai, sistem operasi sangat tertutup dan tidak bisa dipersonalisasi

#### f. Pengembangan Aplikasi Mobile (WEB APP)

Dalam kaitannya dengan teknologi apa saja yang dapat digunakan untuk mengembangkan aplikasi mobile, melihat berbagai perkembangan teknologi yang ada termasuk pengumuman terbaru dari Microsoft, bahwa kini mereka menyediakan dukungan pengembangan aplikasi di Windows 10 menggunakan basis kode yang sama yang pernah dibuat di Android dan iOS.

Secara garis besar didasarkan pada dua pilihan apakah menggunakan native app (aplikasi yang diinstall di dalam device) atau web (menggunakan browser). Masing-masing memiliki keunggulan.

### a. **Web App.**

Keunggulan:

- Ada banyak tool yang dapat dipergunakan.
- Menggunakan kemampuan design web dan skill yang telah dimiliki
- Aplikasi kita berjalan di banyak device seperti halnya web browser.
- Siklus development yang terbilang cepat.

Kekurangan:

- Tidak dapat mengakses semua fitur yang disediakan oleh hardware device.
- Secara umum tampilan kurang responsif dibandingkan dengan native.

### b. **Native App**

Keunggulan:

- Dapat mengatur atau memanfaatkan fitur yang disediakan oleh hardware device.
- Lebih slick

Kekurangan:

- Proses pengembangan memerlukan pengetahuan bahasa pemrograman misalnya Java, atau C++.
- Siklus pengembangan lambat (menulis program, kompilasi, pengujian: baik menggunakan emulator atau langsung diujikan pada device (lama), ulangi lagi jika terjadi kesalahan.
- Dapat mengambil keuntungan dari teknologi array yang luas di perangkat mobile; compass, gyroscope, storage (offline dan cloud), Near Field Communications (NFC), kamera, dsb.

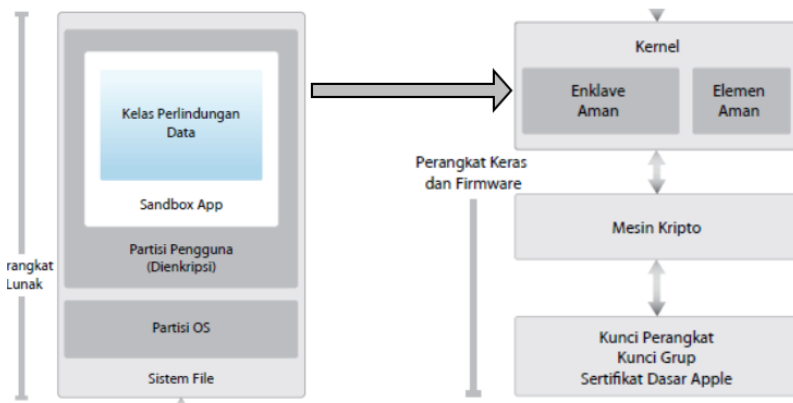
Aplikasi native tidak lintas platform, artinya untuk pengembangan ke dalam platform lain diperlukan usaha dan biaya.

## 1.5. Arsitektur dan Perangkat Mobile

Platform adalah arsitektur hardware/fondasi/standar bagaimana sebuah sistem dimana aplikasi/program dapat berjalan; atau bisa juga dikatakan Platform adalah dasar dari teknologi dimana teknologi yang lain atau proses-proses dibuat. Sebuah Platform terdiri dari sistem operasi yaitu program sistem koordinasi komputer yang memberikan perintah-perintah kepada prosesor dan hardware untuk melakukan operasi-operasi logis dan mengatur pergerakan data di komputer. Banyak orang beranggapan bahwa Platform dan Sistem Operasi adalah sama, namun pada kenyataannya tidak. Platform merupakan dasar atau tempat dimana sistem operasi bekerja, tanpa platform sistem operasi tidak akan bisa berjalan.

### a. Arsitektur iOS

Arsitektur IOS mirip dengan arsitektur dasar yang ditemukan di Mac OS X. Pada tingkat tertinggi IOS bertindak sebagai perantara antara hardware yang mendasari dan aplikasi yang muncul di layar. Penerapan teknologi IOS dapat dipandang sebagai seperangkat lapisan, seperti yang terlihat pada Gambar 1 di bawah ini. Lapisan bawah sistem ini adalah pelayanan dasar dan teknologi di mana semua aplikasi yang mengandalkan; tingkat yang lebih tinggi layer berisi layanan yang lebih canggih dan teknologi.



Gambar 1. Arsitektur Sistem Operasi iOS

Apple merancang platform iOS dengan mengutamakan keamanan, memutuskan untuk membuat platform bergerak terbaik, mengkaji bahaya keamanan lingkungan desktop, dan membangun pendekatan baru terhadap keamanan dalam rancangan iOS. Mengembangkan dan menggabungkan fitur

inovatif yang memperketat keamanan bergerak dan melindungi keseluruhan sistem secara default. Hasilnya, iOS menjadi terobosan besar di bidang keamanan untuk perangkat bergerak.

Setiap perangkat iOS menggabungkan perangkat lunak, perangkat keras, dan layanan yang dirancang untuk berfungsi secara berdampingan untuk keamanan maksimum dan pengalaman pengguna yang transparan. iOS tidak hanya melindungi perangkat dan datanya, tapi juga keseluruhan ekosistem, meliputi semua hal yang dilakukan pengguna di perangkat, jaringan, dan dengan layanan Internet utama.

iOS dan perangkat iOS menyediakan fitur keamanan lanjutan, tapi mudah digunakan. Banyak dari fitur ini yang diaktifkan secara default, sehingga bagian TI tidak perlu melakukan konfigurasi ekstensif. Selain itu, fitur keamanan utama seperti enkripsi perangkat tidak dapat dikonfigurasi, sehingga pengguna tidak dapat menonaktifkannya secara tidak sengaja. Fitur lainnya, seperti Touch ID, meningkatkan pengalaman pengguna dengan membuat pengamanan perangkat lebih mudah dan intuitif.

- 1) **Keamanan sistem:** Perangkat lunak dan perangkat keras terpadu dan aman yang merupakan platform bagi iPhone, iPad, dan iPod touch.
- 2) **Enkripsi dan perlindungan data:** Arsitektur dan rancangan yang melindungi data pengguna jika perangkat hilang atau dicuri, atau jika orang yang tidak berwenang mencoba menggunakan atau memodifikasinya.
- 3) **Keamanan app:** Sistem yang memungkinkan app untuk dijalankan dengan aman dan tanpa merusak integritas platform.
- 4) **Keamanan jaringan:** Protokol jaringan standar industri yang menyediakan pengesahan dan enkripsi data yang aman pada saat transmisi.
- 5) **Apple Pay:** Penerapan pembayaran aman oleh Apple.
- 6) **Layanan internet:** Infrastruktur berbasis jaringan milik Apple untuk pemrosesan pesan, penyelarasan, dan pencadangan.
- 7) **Kontrol perangkat:** Metode yang memungkinkan pengelolaan perangkat iOS, mencegah penggunaan yang tidak sah, dan mengaktifkan penghapusan jarak jauh jika perangkat hilang atau dicuri.
- 8) **Kontrol privasi:** Fitur iOS yang dapat digunakan untuk mengontrol akses ke Layanan Lokasi dan data pengguna.

## ▪ iOS Management Data

### ▪ Core Data

Data Core menyediakan kerangka model data yang fleksibel dan kuat untuk membangun sumur-faktor aplikasi Kakao berdasarkan Model View Controller (MVC) pola. Data Core menyediakan data untuk keperluan umum solusimanajemen dikembangkan untuk menangani data model kebutuhan setiap jenis aplikasi, besar atau kecil. Anda dapat membangun apa saja dari aplikasi menghubungi manajemen untuk program ilustrasi vector-art di atasnya. Langit adalah batas. Data Core memungkinkan Anda dengan cepat menentukan model data aplikasi anda dengan cara grafis dan mudah mengaksesnya dari kode Anda. Ini menyediakan infrastruktur untuk menangani fungsi umum seperti menyimpan, memulihkan, undo dan redo, yang memungkinkan Anda untuk mendapatkan dengan tugas membangun inovasi dalam aplikasi Anda. Karena Core data menggunakan built-in data perpustakaan SQLite tidak perlu untuk menginstal sistem database yang terpisah.

### ▪ SQLite

iOS termasuk perpustakaan SQLite populer, mesin ringan namun kuat database relasional yang mudah tertanam ke aplikasi. Digunakan dalam aplikasi yang tak terhitung di banyak platform, SQLite dianggap sebagai standar industri de facto untuk pemrograman database SQL ringan tertanam. Berbeda dengan kerangka Data berorientasi objek Core, SQLite menggunakan, prosedural SQL-API terfokus untuk memanipulasi tabel data secara langsung.

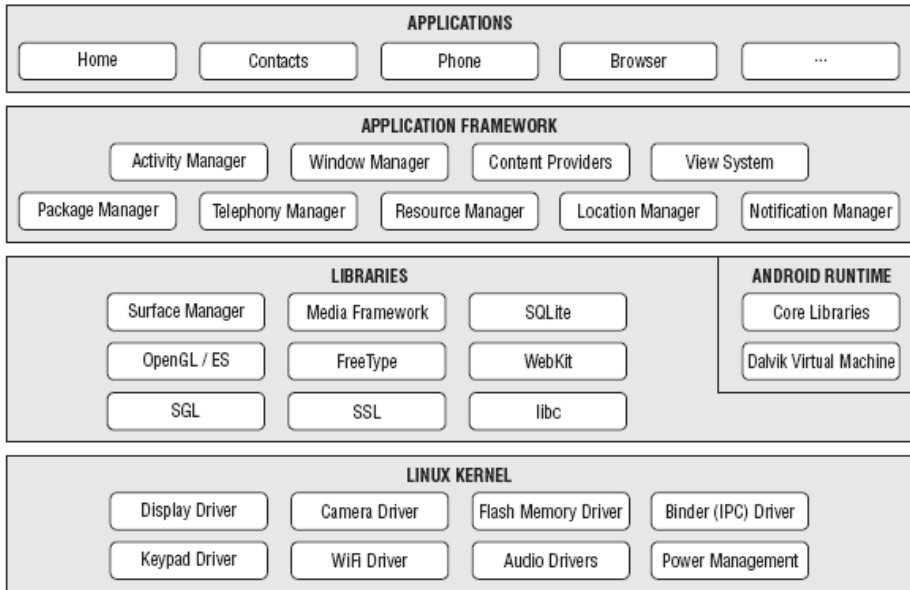
### ▪ XML File

File XML menyediakan format yang terstruktur ringan bahwa aplikasi Anda dengan mudah dapat membaca dan menulis. Plus file XML dengan mudah masuk ke dalam sistem file IOS. Simpan aplikasi pengaturan dan preferensi pengguna di built in database Default User. Ini menyimpan data berbasis XML termasuk API sederhana dengan fitur canggih, termasuk kemampuan untuk serialize dan mengembalikan objek kompleks pada permintaan

### b. Arsitektur Android

Dalam paket sistem operasi android terdiri dari beberapa unsur seperti tampak pada gambar di bawah. Secara sederhana arsitektur android merupakan sebuah kernel Linux dan sekumpulan pustaka C/C++ dalam suatu framework yang menyediakan dan mengatur alur proses aplikasi. Sistem operasi Android berupa sekumpulan software yang terdiri dari: application, runtime

environment, middleware, service dan libraries seperti yang ditunjukkan oleh gambar diagram arsitektur Android dibawah ini:



Gambar 2. Arsitektur Sistem Operasi Android

### 1) Linux Kernel

Android dibangun di atas kernel Linux 2.6. Namun secara keseluruhan android bukanlah linux, karena dalam android tidak terdapat paket standar yang dimiliki oleh linux lainnya. Linux merupakan sistem operasi terbuka yang handal dalam manajemen memori dan proses. Oleh karenanya pada android hanya terdapat beberapa servis yang diperlukan seperti keamanan, manajemen memori, manajemen proses, jaringan dan driver. Kernel linux menyediakan driver layar, kamera, *keypad*, WiFi, *Flash Memory*, audio, dan IPC (*Interprocess Communication*) untuk mengatur aplikasi dan lubang keamanan.

### 2) Libraries

Android menggunakan beberapa paket pustaka yang terdapat pada C/C++ dengan standar *Berkeley Software Distribution* (BSD) hanya setengah dari yang aslinya untuk tertanam pada kernel Linux. Beberapa pustaka di antaranya:

- Media Library untuk memutar dan merekam berbagai macam format audio dan video.

- Surface Manager untuk mengatur hak akses layer dari berbagai aplikasi.
- Graphic Library termasuk di dalamnya SGL dan OpenGL, untuk tampilan 2D dan 3D.
- SQLite untuk mengatur relasi database yang digunakan pada aplikasi.
- SSL dan WebKit untuk browser dan keamanan internet.

Pustaka-pustaka tersebut bukanlah aplikasi yang berjalan sendiri, namun hanya dapat digunakan oleh program yang berada di level atasnya. Sejak versi Android 1.5, pengembang dapat membuat dan menggunakan pustaka sendiri menggunakan Native Development Toolkit (NDK).

### 3) Android Runtime

Pada android tertanam paket pustaka inti yang menyediakan sebagian besar fungsi android. Inilah yang membedakan Android dibandingkan dengan sistem operasi lain yang juga mengimplementasikan Linux. *Android Runtime* merupakan mesin virtual yang membuat aplikasi android menjadi lebih tangguh dengan paket pustaka yang telah ada. Dalam Android Runtime terdapat 2 bagian utama, di antaranya:

- Pustaka Inti, android dikembangkan melalui bahasa pemrograman Java, tapi *Android Runtime* bukanlah mesin virtual Java. Pustaka inti android menyediakan hampir semua fungsi yang terdapat pada pustaka Java serta beberapa pustaka khusus android.
- Mesin Virtual Dalvik, Dalvik merupakan sebuah mesin virtual yang dikembangkan oleh Dan Bornstein yang terinspirasi dari nama sebuah perkampungan yang berada di Iceland. Dalvik hanyalah *interpreter* mesin virtual yang mengeksekusi file dalam format *Dalvik Executable* (\*.dex). Dengan format ini Dalvik akan mengoptimalkan efisiensi penyimpanan dan pengalamatan memori pada file yang dieksekusi. Dalvik berjalan di atas kernel Linux 2.6, dengan fungsi dasar seperti *threading* dan manajemen memori yang terbatas.\

### 4) Application Framework

Kerangka aplikasi menyediakan kelas-kelas yang dapat digunakan untuk mengembangkan aplikasi android. Selain itu, juga menyediakan abstraksi generik untuk mengakses perangkat, serta mengatur tampilan *user interface* dan sumber daya aplikasi. Bagian terpenting dalam kerangka aplikasi android adalah sebagai berikut:

- Activity Manager, berfungsi untuk mengontrol siklus hidup aplikasi dan menjaga keadaan "Backstack" untuk navigasi penggunaan.



- Content Providers, berfungsi untuk merangkum data yang memungkinkan digunakan oleh aplikasi lainnya, seperti daftar nama.
- Resource Manager, untuk mengatur sumber daya yang ada dalam program. Serta menyediakan akses sumber daya di luar kode program, seperti karakter, grafik, dan file layout.
- Location Manager, berfungsi untuk memberikan informasi detail mengenai lokasi perangkat android berada.
- Notification Manager, mencakup berbagai macam peringatan seperti, pesan masuk, janji, dan lain sebagainya yang akan ditampilkan pada status bar.

## 5) Application Layer

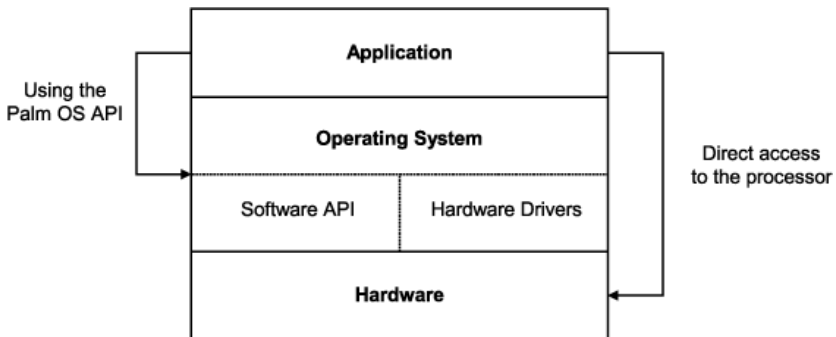
Puncak dari diagram arsitektur android adalah lapisan aplikasi dan *widget*. Lapisan aplikasi merupakan lapisan yang paling tampak pada pengguna ketika menjalankan program. Pengguna hanya akan melihat program ketika digunakan tanpa mengetahui proses yang terjadi dibalik lapisan aplikasi. Lapisan ini berjalan dalam Android *runtime* dengan menggunakan kelas dan *service* yang tersedia pada *framework* aplikasi. Lapisan aplikasi android sangat berbeda dibandingkan dengan sistem operasi lainnya. Pada android semua aplikasi, baik aplikasi inti (native) maupun aplikasi pihak ketiga berjalan di atas lapisan aplikasi dengan menggunakan pustaka API (Application Programming Interface) yang sama.

### c. Arsitektur Palm OS

Pada tingkat tertinggi, arsitektur perangkat Palm OS, dan sebagian besar PDA lainnya, dapat dipecah menjadi tiga lapisan seperti pada Gambar 4: Aplikasi, Sistem Operasi, dan Perangkat Keras. Penggunaan Antarmuka Pemrograman Aplikasi Palm OS (API) menyediakan pengembang aplikasi dengan gagasan independensi perangkat keras dan menyediakan lapisan abstraksi. Jika API digunakan dengan benar, mengkompilasi ulang aplikasi adalah semua yang diperlukan agar dapat dijalankan di perangkat Palm OS berdasarkan perangkat keras yang berbeda. Oleh karena itu, penting untuk memeriksa kelemahan dan menyerang vektor yang dapat ditemukan di antarmuka pemrograman ke sistem operasi.

Mengakses prosesor secara langsung dengan menghindari antarmuka yang dikemukakan oleh sistem operasi memungkinkan pengembang untuk lebih mengontrol prosesor dan fungsinya. Risiko penggunaan sah akses prosesor langsung adalah hilangnya kompatibilitas untuk model masa depan. Sebagai contoh, perangkat OS Palm yang lebih tua tidak mendukung palet LCD grayscale melalui API Palm OS, meskipun perangkat keras yang mendasarinya

memiliki kemampuan ini. Memintas antarmuka ini dan memanfaatkan fungsionalitas prosesor secara langsung akan memperbaiki ini. Idealnya, untuk menyediakan beberapa kemiripan akses kontrol dan keamanan, hanya sistem operasi yang harus memiliki akses ke perangkat keras yang mendasarinya. Mengizinkan aplikasi untuk mengakses perangkat keras secara langsung memberikan jalan untuk serangan jahat.



Gambar 3. Arsitektur Palm OS

Palm OS dirancang untuk terbuka dan modular untuk mendukung pengembangan aplikasi oleh pihak ketiga. Gagasan kontrol akses layer atau berbasis file terutama tidak ada. Tidak mengherankan bahwa semua kode program dan data dapat diakses dan dimodifikasi oleh pengguna atau aplikasi lain. Dalam skenario akses memori yang seragam seperti itu, sulit untuk membedakan antara aplikasi yang sah dan berbahaya hanya dari memori baca / tulis dan panggilan sistem. menawarkan ikhtisar berikut pada sistem file dan struktur aplikasi:

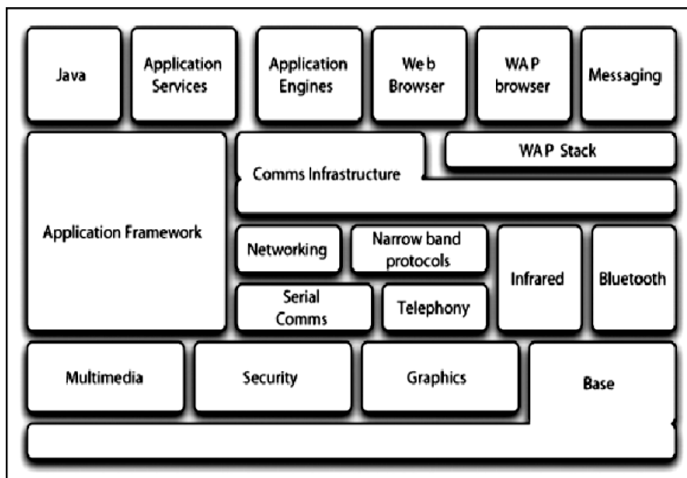
- 1) Palm OS tidak menggunakan sistem file flat tradisional. Data disimpan dalam potongan memori yang disebut "catatan", yang dikelompokkan ke dalam "basis data". Basis data analog dengan file. Perbedaannya adalah bahwa data dipecah menjadi beberapa catatan daripada disimpan dalam satu potongan yang berdekatan.
- 2) Aplikasi Palm OS umumnya merupakan program tunggal, yang digerakkan oleh peristiwa. Hanya satu program yang berjalan pada satu waktu. Setiap aplikasi memiliki fungsi PilotMain yang setara dengan program C utama. Untuk meluncurkan aplikasi, Palm OS memanggil PilotMain dan mengirimkannya kode peluncuran. Kode peluncuran dapat menentukan bahwa aplikasi akan menjadi aktif dan menampilkan antarmuka pengguna (disebut "peluncuran normal"), atau mungkin menetapkan bahwa aplikasi hanya harus melakukan tugas kecil dan

keluar tanpa menampilkan antarmuka penggunanya. Satu-satunya tujuan dari fungsi PilotMain adalah untuk menerima kode peluncuran dan menanggunginya. Versi mendatang dari Palm OS dapat memungkinkan aplikasi pihak ketiga untuk multi-threaded.

Aplikasi dapat mengirim kode peluncuran satu sama lain, sehingga aplikasi mungkin diluncurkan dari aplikasi lain atau mungkin diluncurkan dari sistem. Aplikasi dapat menggunakan kode peluncuran untuk meminta agar aplikasi lain melakukan tindakan atau memodifikasi datanya.

#### d. Arsitektur Symbian OS

Secara umum arsitektur Symbian OS sendiri dapat dilihat pada Gambar 3 yang dibagi menjadi empat lapisan atau grup berdasarkan penggunaan API yang tersedia, yaitu: lapisan pendukung aplikasi (application utility layer), lapisan layanan dan framework antarmuka grafis (GUI framework), lapisan komunikasi, dan system API dasar.



Gambar 4. Arsitektur Symbian OS

- Lapisan pendukung aplikasi (Application Utility Layer):  
Terdiri dari berbagai pendukung yang berorientasi pada aplikasi. Hal ini memungkinkan aplikasi lain (diluar sistem operasi) untuk berintegrasi dengan aplikasi dasar yang tersedia pada sistem operasi. Bentuk layanan lain termasuk proses pertukaran data dan manajemen data.
- Lapisan Layanan dan Framework Antarmuka Grafis (GUI Framework):

Merupakan framework API yang tersedia untuk memberi dukungan terhadap penanganan input user secara grafis maupun suara yang dapat digunakan oleh aplikasi lain.

c. Lapisan Komunikasi (Communication Layer):

Tentu saja sebagai sistem operasi yang fokus diimplementasi pada peralatan komunikasi mobile, Symbian OS memiliki kumpulan API yang fokus pada lapisan komunikasi. Bagian teratas pada lapisan ini terdapat dukungan pencarian dan pengiriman pesan teks. Berikutnya adalah antarmuka yang memberi dukungan komunikasi seperti Bluetooth dan infrared (IrDA) serta USB. Yang terakhir pada lapisan ini adalah protocol komunikasi berupa TCP/IP, HTTP, WAP dan layanan telepon.

d. Lapisan Aplikasi Program Interface dasar (API Layer):

Merupakan kumpulan API yang mendukung pengaksesan data memori, tanggal dan waktu, serta sistem dasar lainnya.

## 1.6. Perangkat Mobile dan Penggunaan Umum

### a. Perangkat Mobile

Perkembangan teknologi semakin cepat. Produk-produk baru berbasis teknologi terus menerus berimprovisasi dengan ragamnya yang semakin bervariasi. Produk berbasis teknologi memberikan kemudahan dan multifungsi yang beragam. Tak terkecuali dengan perangkat mobile, perkembangan perangkat ini lebih signifikan dibanding dengan perangkat lain. Perangkat mobile dimana pengguna dapat membawa perangkat mobile tersebut dengan praktis dan leluasa karena ukurannya yang kecil juga dapat digunakan dimanapun dan kapanpun.

Perangkat mobile adalah istilah umum untuk semua jenis komputer yang dapat digenggam. Perangkat ini dirancang untuk mudah dibawa kemanapun (portabel) dan dapat digenggam dengan tangan. Penggunaan dan kebutuhan akan perangkat mobile juga semakin tinggi, hal ini disebabkan karena perangkat mobile telah mengadopsi teknologi yang mampu mempermudah pengguna mendapatkan dan mengerjakan apa yang mereka inginkan. Sangat nyaman sekali jika pengguna dapat mengerjakan pekerjaannya sekaligus mendapatkan sisi hiburan hanya dengan menggunakan sebuah gadget atau perangkat saja, misalnya membuat laporan sembari mendengarkan musik serta berselancar internet yang bisa dilakukan dimana saja dan kapan saja dengan bantuan perangkat mobile.

Perangkat mobile kini bukan hanya sebatas alat komunikasi seperti handphone saja, dengan penerapan teknologi komputasi, sebuah perangkat mobile dapat melakukan fungsi-fungsi yang berbeda bahkan dalam waktu bersamaan. Apalagi dengan kemunculan beberapa sistem operasi yang dikhususkan untuk perangkat mobile seperti iOS, Android, Blackberry, Windows, dan Ubuntu. Penerapan sistem operasi dalam perangkat mobile tersebut menjadikan perangkat tersebut mampu melakukan kegiatan komputasi yang sama dengan perangkat komputasi pada umumnya, bahkan dapat melebihi dan sangat praktis. Perangkat komputasi mobile juga mengancam keberlangsungan perangkat komputasi konvensional seperti PC (Personal Komputer) hingga ada juga yang meramalkan akan terjadinya kiamat PC. Beberapa perangkat komputasi mobile yang populer tinggi penggunaannya saat ini adalah:

## 1) Kacamata

Adalah Google sebagai pelopor inovasi menghadirkan kacamata yang mampu melakukan kegiatan komputasi digital. Google Glass merupakan kacamata digital yang dikembangkan Google yang memiliki fungsi yang cukup kompleks.



Gambar 5. Kacamata digital

## 2) Notebook/Netbook/Laptop

Notebook pada dasarnya adalah PC yang dikemas secara ringkas dengan ukuran yang jauh lebih kecil dari PC. Notebook memiliki kemampuan yang sama dengan PC dan mengadopsi sistem operasi yang sama. Komponen penyusunnya juga sama dengan PC, namun komponen-komponen tersebut didesain agar mampu berjalan secara lebih cepat dan sederhana. Sayangnya harga perangkat ini lebih mahal dibandingkan dengan harga PC dengan spesifikasi yang hampir sama, bahkan secara performa dengan spesifikasi yang hampir sama PC jauh lebih unggul.



Gambar 6. Notebook – Laptop

### 3) Handphone dan Smartphone

Smartphone ini yang banyak memberikan perubahan yang signifikan dalam perkembangan perangkat mobile. Berkat pengembangan perangkat lunak untuk handphone, perangkat ini berubah menjadi smartphone yang bukan hanya sebagai alat komunikasi saja, tapi juga mampu melakukan tugas komputasi yang sama dengan Notebook atau PC.



Gambar 7. Handphone dan Smartphone

### 4) Tablet

Tablet atau banyak disebut dengan Tab pada dasarnya sama dengan smartphone hanya dikemas dengan ukuran yang lebih lapang. Jika smartphone hanya berukuran 3 hingga 6 inch saja, maka Tablet dapat berukuran hingga 12 inch. Dengan ukuran yang sebesar itu, dapat dibenamkan komponen dengan spesifikasi yang lebih tinggi dibandingkan smartphone pada umumnya sehingga performa juga lebih baik, display dan kenyamanan penggunaan juga pasti lebih baik.



Gambar 8. Tablet

## 5) Jam Tangan

Jam tangan atau arloji kini bukan hanya sebagai penunjuk waktu saja. Jam tangan kini juga mampu berperan sebagai alat komunikasi seperti SMS dan telephone. Bahkan sistem operasi kini sudah mampu ditanamkan dalam arloji, beberapa vendor mampu menerapkan Android dalam jam tangan. Apple juga sedang mengembangkan jam tangan dengan sistem operasi iOS-nya yang dikabarkan dinamai dengan iWatch.



Gambar 9. Jam Tangan Digital

## 6) GPS

Global Positioning System (GPS) merupakan sebuah perangkat yang berguna untuk menunjukkan koordinat lokasi pada peta digital. Perkembangan GPS memang tidak secepat perkembangan perangkat lain, namun perangkat ini semakin banyak diperlukan dan digunakan. GPS sekarang dilengkapi dengan penunjuk arah dan panduan perjalanan agar pengguna dapat lebih mudah menemukan lokasi dan tidak tersesat. Tidak hanya berkuat pada lokasi saja, GPS juga dilengkapi dengan wi-fi bahkan simcard sehingga pengguna dapat berinternet dan menjadikannya sebagai alat komunikasi. Dengan adanya konektivitas internet pada GPS, peta dapat diupdate dengan mudah dan otomatis. Media player juga banyak ditambahkan dalam GPS sehingga pengguna juga dapat memutar lagu dan video.



Gambar 10. GPS



Makin banyak variasi dan ragam perangkat mobile yang mampu melakukan tugas komputasi digital yang luar biasa. Hal ini tentunya berkat adanya perkembangan teknologi yang semakin pesat. Bukan tidak mungkin kelak semua didominasi perangkat digital yang mampu memudahkan manusia dalam mengerjakan sesuatu. Perangkat-perangkat canggih yang biasanya hanya ada dalam Film mungkin dapat diwujudkan keberadaannya. Dunia memang membutuhkan perkembangan teknologi yang cepat agar didapat cara-cara yang jauh lebih efisien untuk mengerjakan suatu tugas namun tetap bersahabat dengan lingkungan.

#### b. Penggunaan Umum Perangkat Mobile

Perangkat mobile memiliki banyak jenis dalam hal ukuran, desain dan layout, tetapi mereka memiliki kesamaan karakteristik yang sangat berbeda dari sistem desktop.

- Ukuran yang kecil perangkat mobile memiliki ukuran yang kecil, Konsumen menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.
- Memory yang terbatas, Perangkat mobile juga memiliki memory yang kecil, yaitu primary (RAM) dan secondary (disk). Pembatasan ini adalah salah satu faktor yang mempengaruhi penulisan program untuk berbagai jenis dari perangkat ini. Dengan pembatasan jumlah dari memory, pertimbangan - pertimbangan khusus harus diambil untuk memelihara pemakaian dari sumber daya yang mahal ini.
- Daya proses yang terbatas, Sistem mobile tidaklah setangguh rekan mereka yaitu desktop. Ukuran, teknologi dan biaya adalah beberapa faktor yang mempengaruhi status dari sumber daya ini. Seperti hardisk dan RAM, Anda dapat menemukan mereka dalam ukuran yang pas dengan sebuah kemasan kecil.
- Mengkonsumsi daya yang rendah, Perangkat mobile menghabiskan sedikit daya dibandingkan dengan mesin desktop. Perangkat ini harus menghemat daya karena mereka berjalan pada keadaan dimana daya yang disediakan dibatasi oleh baterai - baterai.
- Kuat dan dapat diandalkan, karena perangkat mobile selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapi benturan - benturan, gerakan, dan sesekali tetesan air.
- konektivitas yang terbatas, perangkat mobile memiliki bandwidth rendah, beberapa dari mereka bahkan tidak tersambung. Kebanyakan dari mereka menggunakan koneksi wireless.

- Masa hidup yang pendek, Perangkat - perangkat konsumen ini menyala dalam hitungan detik kebanyakan dari mereka selalu menyala. Coba ambil kasus sebuah handphone, mereka booting dalam hitungan detik dan kebanyakan orang tidak mematikan handphone mereka bahkan ketika malam hari. PDA akan menyala jika anda menekan tombol power mereka.

Untuk penggunaan perangkat mobile, kita dapat klasifikasikan ke dalam beberapa macam penggunaan yang umum digunakan oleh perangkat mobile. Beberapa macam penggunaan tersebut antara lain:

#### 1) Telepon dan Messaging (SMS)

Penggunaan ini sebenarnya merupakan fungsi dasar dan awal dari diciptakannya perangkat mobile. Karena memang fungsinya adalah sebagai alat komunikasi, sehingga cukup dengan penggunaan voice dan layanan pesan (Short Message Service), kita bisa berkomunikasi jarak jauh dengan orang lain.

#### 2) Multimedia dan Game

Merupakan penggunaan tambahan dari sisi hiburan (entertainment) yang ditunjukkan bagi pengguna perangkat mobile. Sekarang, fasilitas multimedia sudah sangat umum dijumpai pada perangkat mobile. Fitur-fitur seperti radio, mp3 player, sampai video player adalah beberapa contoh penggunaan multimedia yang sekarang banyak dijumpai pada perangkat mobile. Tak kalah pentingnya, game juga menjadi bagian yang tak terpisahkan dari perangkat mobile saat ini. Rasanya hamper semua perangkat mobile memiliki game yang sudah terinstall didalamnya.

#### 3) Internet Browsing

Dengan adanya fasilitas ini, sangat memungkinkan bagi kita mengakses internet dengan sangat mudah dimanapun dan kapanpun. Hal ini sudah menjadi kebutuhan dasar untuk penggunaan perangkat mobile.

#### 4) Kamera

Ada tidaknya fitur kamera digital nampaknya saat ini telah menjadi salah satu kriteria gadgeters dalam memilih perangkat mobile.

#### 5) Pertukaran Data

Kapasitas memori yang semakin besar dapat memungkinkan pengguna perangkat mobile untuk melakukan penyimpanan data dalam jumlah besar. Dengan adanya fitur pertukaran data seperti bluetooth, infrared, MMS (Multimedia Message Service), dan sebagainya memungkinkan pengguna untuk saling bertukar data.

## 6) Transaksi Mobile

Kemudahan dalam bertransaksi kini dapat dilakukan secara online via perangkat mobile. Pengguna dapat melakukan transaksi melalui SMS banking, atau mobile banking.

## 7) Penggunaan Lainnya

Serta penggunaan lainnya yang menjadikan penggunaan perangkat mobile telah begitu membudaya dikalangan masyarakat. Sebagai contoh, penggunaan perangkat mobile untuk memenuhi hasrat style hidup tak kalah pentingnya bagi sebagian orang.

## RANGKUMAN

- 1) Pemrograman Mobile adalah pemrograman yang ditujukan untuk pembuatan aplikasi diperangkat mobile. Banyak sekali platform mobile yang dapat kita coba, diantaranya iOS, Palm OS, Windows Mobile, Symbian, dan Android. Pemrograman mobile bukan hanya tentang membangun aplikasi telepon. Tetapi pemrograman mobile adalah semua pengembangan untuk segala jenis perangkat mobile.
- 2) Perangkat komputasi bergerak seperti pada Notebook/Netbook/Laptop, kacamata digital, Handphone dan Smartphone, Tablet, Jama tangan digital (iWatch), tidak bisa dilepaskan dari adanya sistem operasi (operating system). Sebagaimana sistem operasi pada komputer, sistem operasi pada perangkat mobile berfungsi untuk mengelola sumber daya perangkat bergerak dan menyediakan layanan (servis) yang bersifat umum agar bisa diakses oleh aplikasi yang berjalan di atasnya. Sistem operasi pada perangkat mobile memadukan kemampuan sistem operasi pada komputer dengan kemampuan untuk mengelola komponen yang biasa ada dalam perangkat mobile seperti layar sentuh, jaringan seluler, bluetooth, WiFi, navigasi GPS, NFC, dan fitur standar lainnya. Jadi sistem operasi perangkat bergerak memiliki kelebihan dalam mengelola secara langsung hardware yang disertakan.
- 3) Dengan adanya sistem operasi pada perangkat mobile ini, memungkinkan perangkat mobile untuk dipasang berbagai jenis aplikasi yang berjalan di atas sistem operasi yang bersangkutan. Sepanjang aplikasinya sesuai (kompatibel) dengan sistem operasinya, aplikasi tersebut bisa ditambahkan. Sehingga pengguna perangkat bisa dengan bebas menambahkan aplikasinya sendiri.

## REFERENSI

- Stojmenovic, I. ed., 2003. *Handbook of wireless networks and mobile computing* (Vol. 27). John Wiley & Sons.
- Perkins, C.E., Alpert, S.R. and Woolf, B., 1997. *Mobile IP; Design Principles and Practices*. Addison-Wesley Longman Publishing Co., Inc.
- V.Jeyasri Arokiamary, 2009. *Mobile Computing*. Technical Publications, 2009
- Wolman, A., Lane, N.D., Chu, D., Moscibroda, T. and Seo, J., Microsoft Technology Licensing LLC, 2018. *Mobile computing device application sharing*. U.S. Patent 9,936,335.

## MATERI KOMPETENSI

Materi bab ini akan membahas:

- 1) Sejarah perkembangan dan versi Android.
- 2) Pemasangan/instalasi dan konfigurasi tool Android Studio
- 3) Struktur direktori file project dan modul Android Studio.
- 4) Pengujian perangkat emulator Android Studio.

## YANG HARUS SUDAH ANDA KETAHUI

Untuk materi dan praktik ini Anda harus bisa:

- 1) Memahami proses development perangkat lunak umum untuk aplikasi berorientasi objek menggunakan IDE (Integrated Development Environment).
- 2) Memiliki pengalaman telah mengikuti pembelajaran pemrograman berorientasi objek dan pemrograman visual dengan java.

## YANG AKAN ANDA BUTUHKAN

Untuk materi dan praktik ini, Anda akan membutuhkan:

- 1) Komputer Mac, Windows, atau Linux. Tool Android Studio untuk persyaratan sistem terbaru.
- 2) Akses internet atau cara alternatif untuk memuat Android Studio dan pemasangan Java terbaru ke komputer Anda.
- 3) Smartphone sebagai perangkat untuk pengujian secara nyata aplikasi yang telah dibuat.

## YANG AKAN ANDA PELAJARI

Anda akan belajar:

- 1) Memasang dan menggunakan Android IDE.
- 2) Memahami proses development untuk membangun aplikasi Android.
- 3) Membuat proyek Android dari template aplikasi dasar.



## Bab 2: Pengantar Aplikasi Android

### 2.1. Sejarah Perkembangan

Android adalah sistem operasi berbasis kernel Linux. Android dikembangkan di Proyek Android Open Source (AOSP). Proyek ini dipimpin oleh Google. Sebagai platform perangkat seluler paling populer di dunia, Android mendukung ratusan juta perangkat seluler di lebih dari 190 negara di seluruh dunia. Android memiliki basis pemasangan terbesar dari platform seluler apa pun dan masih tumbuh dengan cepat. Setiap hari, jutaan pengguna lain menyalakan perangkat Android mereka untuk pertama kalinya dan mulai mencari aplikasi, game, dan materi digital lainnya.



Gambar 11. Antarmuka Android

Android menyediakan antarmuka pengguna (UI) layar sentuh untuk berinteraksi dengan aplikasi. Antarmuka pengguna Android sebagian besar berdasarkan pada manipulasi langsung, menggunakan isyarat sentuhan seperti

menggesek, mengetuk, dan mencubit untuk memanipulasi objek di layar. Selain keyboard, ada keyboard virtual yang bisa disesuaikan untuk masukan teks. Android juga bisa mendukung pengontrol game dan keyboard fisik berukuran penuh yang dihubungkan dengan Bluetooth atau USB.

Layar utama Android bisa berisi sejumlah laman *ikon aplikasi*, yang akan meluncurkan aplikasi terkait, dan *widget*, dengan menampilkan materi langsung yang diperbarui secara otomatis seperti cuaca, kotak masuk email pengguna, atau ticker berita. Android juga bisa memutar materi multimedia seperti musik, animasi, dan video. Gambar 2 menampilkan ikon aplikasi pada layar utama (kiri), musik yang diputar (tengah), dan widget yang ditampilkan (kanan). Sepanjang bagian atas layar terdapat bilah status, yang menampilkan informasi tentang perangkat dan konektivitasnya. Layar utama Android bisa terdiri dari sejumlah laman, yang bisa digesek mundur dan maju oleh pengguna.



Gambar 12. Ikon aplikasi pada layar utama

Android didesain untuk menyediakan respons cepat terhadap masukan pengguna. Selain antarmuka sentuh yang berubah-ubah, kemampuan getaran perangkat Android bisa menyediakan umpan balik sentuhan. Perangkat keras internal seperti akselerometer, giroskop, dan sensor kedekatan, digunakan oleh banyak aplikasi untuk merespons tindakan pengguna tambahan. Sensor tersebut bisa mendeteksi rotasi layar dari potret ke lanskap untuk tampilan yang lebih lebar atau sensor bisa memungkinkan pengguna untuk menyetir kendaraan virtual dengan memutar perangkat seolah-olah setir mobil.

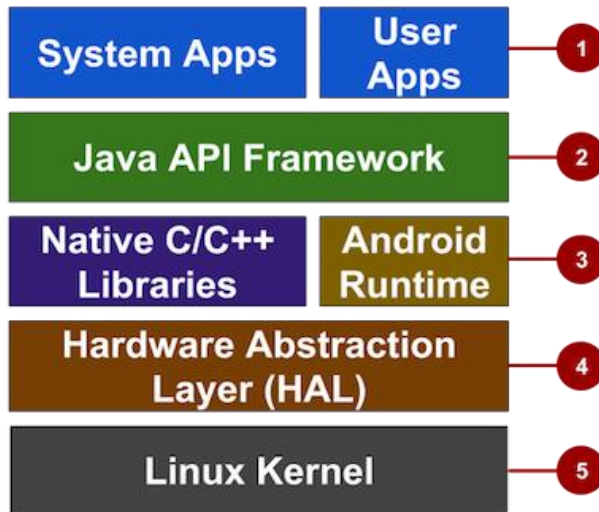


Platform Android, berdasarkan pada kernel Linux, terutama didesain untuk perangkat seluler layar sentuh seperti ponsel cerdas dan tablet. Karena perangkat Android biasanya bertenaga baterai, Android didesain untuk mengelola proses guna menjaga konsumsi daya tetap minimum, sehingga menyediakan penggunaan baterai lebih lama. Gunakan Android Software Development Kit (SDK) Android untuk mengembangkan aplikasi yang memanfaatkan UI dan sistem operasi Android. SDK terdiri dari serangkaian alat development menyeluruh yang menyertakan debugger, pustaka perangkat lunak kode praatulis, emulator perangkat, dokumentasi, kode contoh, dan tutorial. Gunakan alat-alat ini untuk membuat aplikasi yang terlihat hebat dan memanfaatkan kemampuan perangkat keras yang tersedia di setiap perangkat.

Untuk mengembangkan aplikasi menggunakan SDK, gunakan bahasa pemrograman Java untuk mengembangkan aplikasi dan file Extensible Markup Language (XML) untuk menjelaskan resource data. Dengan menulis kode di Java dan membuat biner aplikasi tunggal, Anda akan memiliki aplikasi yang bisa berjalan pada faktor bentuk ponsel dan tablet. Anda bisa mendeklarasikan UI dalam rangkaian resource XML ringan, satu rangkaian untuk bagian UI yang umum bagi semua faktor bentuk, dan rangkaian lain untuk fitur yang khusus bagi ponsel atau tablet. Pada waktu proses, Android menerapkan rangkaian resource yang tepat berdasarkan ukuran layar, kepadatan, lokal, dan sebagainya.

Untuk membantu Anda mengembangkan aplikasi secara efisien, Google menawarkan Lingkungan Development Terintegrasi (IDE) Java lengkap yang disebut Android Studio, dengan fitur lanjutan untuk pengembangan, debug, dan pemaketan aplikasi Android. Dengan menggunakan Android Studio, Anda bisa mengembangkan perangkat Android yang tersedia, atau membuat perangkat virtual yang mengemulasikan konfigurasi perangkat keras apa pun.

Android menyediakan arsitektur development yang kaya. Anda tidak perlu mengetahui banyak tentang komponen arsitektur ini, namun perlu mengetahui apa yang tersedia dalam sistem yang digunakan untuk aplikasi Anda. Diagram berikut menampilkan komponen utama sistem tumpukan Android — sistem operasi dan arsitektur development. Sistem operasi Android dapat dibagi menjadi empat area seperti yang digambarkan dalam grafik berikut. Pengembang aplikasi Android biasanya bekerja dengan dua lapisan di atas untuk membuat aplikasi Android baru.



Gambar 13. Arsitektur Framework Android Development

Pada Gambar 13, dijelaskan sebagai berikut:

1. **Aplikasi:** Aplikasi berada pada tingkat ini, bersama dengan aplikasi sistem inti untuk email, perpesanan SMS, kalender, penjelajahan Internet, atau kontak.
2. **Kerangka Kerja API Java:** Semua fitur Android tersedia untuk developer melalui antarmuka pemrograman aplikasi (API) yang ditulis dalam bahasa Java. Anda tidak perlu mengetahui detail semua API untuk mempelajari cara mengembangkan aplikasi Android, namun Anda bisa mengetahui selengkapnya tentang API berikut ini, yang berguna untuk membuat aplikasi:
  - Sistem Tampilan digunakan untuk membangun UI aplikasi, termasuk daftar, tombol, dan menu.
  - Pengelola Referensi digunakan untuk mengakses resource non-kode seperti string, grafik, dan file layout yang dilokalkan.
  - Pengelola Notifikasi digunakan untuk menampilkan peringatan khusus di bilah status.
  - Pengelola Activity yang mengelola daur hidup aplikasi.
  - Penyedia Materi yang memungkinkan aplikasi untuk mengakses data dari aplikasi lain.
  - Semua API kerangka kerja yang digunakan aplikasi sistem Android.

- 3. Pustaka dan Waktu Proses Android:** Setiap aplikasi berjalan dalam prosesnya sendiri dan dengan instance Android Runtime sendiri, yang memungkinkan beberapa mesin sekaligus virtual pada perangkat bermemori rendah. Android juga menyertakan rangkaian pustaka waktu proses inti yang menyediakan sebagian besar fungsionalitas bahasa pemrograman Java, termasuk beberapa fitur bahasa Java 8 yang digunakan kerangka kerja Java API. Banyak layanan dan komponen sistem Android inti dibangun dari kode asli yang memerlukan pustaka asli yang ditulis dalam C dan C++. Pustaka asli tersebut tersedia untuk aplikasi melalui kerangka kerja Java API.
- 4. Hardware Abstraction Layer (HAL):** Lapisan ini menyediakan antarmuka standar yang menunjukkan kemampuan perangkat keras di perangkat ke kerangka kerja Java API yang lebih tinggi. HAL terdiri atas beberapa modul pustaka, masing-masing mengimplementasikan antarmuka untuk komponen perangkat keras tertentu, seperti modul kamera atau bluetooth.
- 5. Kernel Linux:** Fondasi platform Android adalah kernel Linux. Lapisan di atas mengandalkan kernel Linux untuk fungsionalitas pokok seperti threading dan manajemen memori tingkat rendah. Menggunakan kernel Linux memungkinkan Android memanfaatkan fitur keamanan utama dan memungkinkan produsen perangkat mengembangkan driver perangkat keras untuk kernel yang cukup dikenal

Antarmuka pengguna Android umumnya berupa manipulasi langsung, menggunakan gerakan sentuh yang serupa dengan tindakan nyata, misalnya menggeser, mengetuk, dan mencubit untuk memanipulasi objek di layar, serta papan ketik virtual untuk menulis teks. Selain perangkat layar sentuh, Google juga telah mengembangkan Android TV untuk televisi, Android Auto untuk mobil, dan Android Wear untuk jam tangan, masing-masingnya memiliki antarmuka pengguna yang berbeda. Varian Android juga digunakan pada komputer jinjing, konsol permainan, kamera digital, dan peralatan elektronik lainnya.

Android adalah sistem operasi dengan sumber terbuka, dan Google merilis kodenya di bawah Lisensi Apache. Kode dengan sumber terbuka dan lisensi perizinan pada Android memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi. Selain itu, Android memiliki sejumlah besar komunitas pengembang aplikasi (apps) yang memperluas fungsionalitas perangkat, umumnya ditulis dalam versi kustomisasi bahasa pemrograman Java. Pada bulan Oktober 2013, ada lebih dari satu juta aplikasi yang tersedia untuk

Android, dan sekitar 50 miliar aplikasi telah diunduh dari Google Play, toko aplikasi utama Android. Sebuah survei pada bulan April-Mei 2013 menemukan bahwa Android adalah platform paling populer bagi para pengembang, digunakan oleh 71% pengembang aplikasi bergerak. Di Google I/O 2014, Google melaporkan terdapat lebih dari satu miliar pengguna aktif bulanan Android, meningkat dari 583 juta pada bulan Juni 2013.

Faktor-faktor di atas telah memberikan kontribusi terhadap perkembangan Android, menjadikannya sebagai sistem operasi telepon pintar yang paling banyak digunakan di dunia, mengalahkan Symbian pada tahun 2010. Android juga menjadi pilihan bagi perusahaan teknologi yang menginginkan sistem operasi berbiaya rendah, bisa dikustomisasi, dan ringan untuk perangkat berteknologi tinggi tanpa harus mengembangkannya dari awal. Sifat Android yang terbuka juga telah mendorong munculnya sejumlah besar komunitas pengembang aplikasi untuk menggunakan kode sumber terbuka sebagai dasar proyek pembuatan aplikasi, dengan menambahkan fitur-fitur baru bagi pengguna tingkat lanjut atau mengoperasikan Android pada perangkat yang secara resmi dirilis dengan menggunakan sistem operasi lain.

Pada November 2013, Android menguasai pangsa pasar telepon pintar global, yang dipimpin oleh produk-produk Samsung, dengan persentase 64% pada bulan Maret 2013. Pada Juli 2013, terdapat 11.868 perangkat Android berbeda dengan beragam versi. Keberhasilan sistem operasi ini juga menjadikannya sebagai target litigasi paten "perang telepon pintar" antar perusahaan-perusahaan teknologi. Hingga bulan Mei 2013, total 900 juta perangkat Android telah diaktifkan di seluruh dunia, dan 48 miliar aplikasi telah dipasang dari Google Play.

## 2.2. Versi Android

Versi Android diawali dengan dirilisnya Android beta pada bulan November 2007. Versi komersial pertama, Android 1.0, dirilis pada September 2008. Android dikembangkan secara berkelanjutan oleh Google dan Open Handset Alliance (OHA), yang telah merilis sejumlah pembaruan sistem operasi ini sejak dirilisnya versi awal.

Sejak April 2009, versi Android dikembangkan dengan nama kode yang dinamai berdasarkan makanan pencuci mulut dan penganan manis. Masing-masing versi dirilis sesuai urutan alfabet, yakni Cupcake (1.5), Donut (1.6), Eclair (2.0–2.1), Froyo (2.2–2.2.3), Gingerbread (2.3–2.3.7), Honeycomb (3.0–3.2.6), Ice Cream Sandwich (4.0–4.0.4), Jelly Bean (4.1–4.3), KitKat (4.4+), Lollipop (5.0+), Marshmallow (6.0+), Nougat (7.0+) dan selanjutnya versi android terbaru adalah Android Oreo (8.0+). Sistem operasi Android diterbitkan dalam berbagai versi. Tabel di bawah ini menunjukkan tanggal rilis dan fitur kunci dari semua versi OS Android, diurutkan secara kronologis berdasarkan level antarmuka pemrograman aplikasi (API). Sejarah versi Android menurut level API

Tabel 1. Perkembangan Versi Android

Android 1.0 (API level 1)		
Android 1.0, versi komersial pertama Android, dirilis pada 23 September 2008. Perangkat Android pertama yang tersedia secara komersial adalah <u>HTC Dream</u> . Android 1.0 memiliki fitur sebagai berikut:		
Versi	Tanggal Rilis	Fitur
1.0	23 September 2008	<ul style="list-style-type: none"><li>▪ <u>Android Market</u>, untuk mengunduh dan memperbarui aplikasi melalui toko aplikasi resmi Android.</li><li>▪ <u>Penjelajah web</u>, untuk menampilkan, memperbesar dan melihat dalam layar penuh halaman web <u>HTML</u> dan <u>XHTML</u>.</li><li>▪ Dukungan kamera, versi ini tidak memiliki pilihan untuk mengubah resolusi kamera, kejernihan, kualitas foto, dan sebagainya.</li><li>▪ Memungkinkan pengelompokan sejumlah ikon aplikasi ke dalam satu folder di layar depan (<i>homescreen</i>).</li><li>▪ Akses ke server <u>surel</u> web, mendukung <u>POP3</u>, <u>IMAP4</u>, dan <u>SMTP</u></li><li>▪ Sinkronisasi <u>Gmail</u> dengan aplikasi Gmail.</li><li>▪ Sinkronisasi <u>Google Contacts</u> dengan aplikasi People</li><li>▪ Sinkronisasi <u>Google Calendar</u> dengan aplikasi Calendar</li></ul>

	<ul style="list-style-type: none"> <li>▪ <u>Google Maps</u>, dengan <u>Latitude</u> dan <u>Street View</u> untuk melihat peta dan citra satelit, serta menemukan lokasi bisnis dan petunjuk arah mengemudi dengan menggunakan <u>GPS</u>.</li> <li>▪ <u>Google Sync</u>, memungkinkan pengelolaan sinkronisasi pada aplikasi Gmail, People, dan Calendar.</li> <li>▪ <u>Google Search</u>, memungkinkan pengguna untuk mencari sesuatu di Internet.</li> <li>▪ <u>Google Talk</u>, aplikasi pesan instan.</li> <li>▪ <u>Pesan instan</u>, <u>pesan teks</u> (SMS), dan <u>MMS</u>.</li> <li>▪ <u>Pemutar media</u>, untuk mengelola, mengimpor, dan memutar berkas media, namun versi ini tidak menyediakan dukungan video dan <u>Bluetooth</u> stereo.</li> <li>▪ Notifikasi muncul pada status bar, dengan pilihan untuk mengatur nada dering, LED, atau nada getar.</li> <li>▪ <u>Voice Dialer</u>, memungkinkan pengguna untuk memanggil kontak tanpa harus menyetik nama atau nomor telepon.</li> <li>▪ <u>Wallpaper</u>, memungkinkan pengguna untuk mengatur gambar latar belakang di layar depan.</li> <li>▪ Pemutar video <u>YouTube</u>.</li> <li>▪ Aplikasi lainnya seperti: Jam Alarm, Kalkulator, Panggilan, Home screen (Launcher), Galeri, dan Pengaturan.</li> <li>▪ Dukungan <u>Wi-Fi</u> dan Bluetooth.</li> </ul>
--	--

## Android 1.1 (API level 2)

Pada 9 Februari 2009, pemutakhiran Android 1.1 dirilis, awalnya hanya untuk HTC Dream. Android 1.1 juga dikenal dengan "Petit Four", meskipun nama ini tidak digunakan secara resmi. Versi ini memperbaiki beberapa bug, mengubah API Android, dan menambahkan beberapa fitur:

Versi	Tanggal Rilis	Fitur
1.1	9 Februari 2009	Rincian dan tinjauan tersedia saat pengguna mencari lokasi bisnis pada Peta.
		Kemampuan untuk menampilkan/meenyembunyikan tombol panggilan.
		Kemampuan untuk menyimpan lampiran pada pesan.
		Menambah dukungan marquee pada tata ruang sistem.

## Android 1.5 Cupcake (API level 3)

Pada 27 April 2009, Android 1.5 dirilis, menggunakan kernel Linux 2.6.27. Versi ini adalah rilis pertama yang secara resmi menggunakan nama kode berdasarkan nama-nama makanan pencuci mulut ("Cupcake"), nama yang kemudian digunakan untuk semua versi rilis selanjutnya. Pembaruan pada versi ini termasuk beberapa fitur baru dan perubahan UI:

Versi	Tanggal Rilis	Fitur
1.5	30 April 2009	<ul style="list-style-type: none"><li>▪ Dukungan papan ketik virtual pihak ketiga dengan prediksi teks dan kamus pengguna</li><li>▪ Dukungan Widget – tampilan aplikasi miniatur yang tertanam dalam aplikasi lain dan menerima pembaruan secara periodik[28]</li><li>▪ Kemampuan merekam dan memutar video berformat MPEG-4 dan 3GP</li><li>▪ Kemampuan memasang (pairing) dan dukungan stereo bagi Bluetooth (A2DP dan AVRCP)</li><li>▪ Fitur salin dan tempel pada penjelajah web</li><li>▪ Foto pengguna ditampilkan pada kontak favorit</li><li>▪ Tanggal/waktu ditampilkan pada log panggilan, dan akses satu sentuhan ke nomor kontak dari log panggilan</li><li>▪ Transisi layar animasi</li><li>▪ Opsi memutar-otomatis</li><li>▪ Animasi boot baru</li><li>▪ Kemampuan untuk mengunggah video ke YouTube</li><li>▪ Kemampuan untuk mengunggah foto ke <u>Picasa</u></li></ul>

## Android 1.6 Donut (API level 4)

Pada 15 September 2009, SDK Android 1.6 – dinamai Donut – dirilis, berdasarkan kernel Linux 2.6.29.

Fitur-fitur barunya adalah sebagai berikut:

Versi	Tanggal Rilis	Fitur
1.6	15 September 2009	<ul style="list-style-type: none"><li>▪ Entri pencarian teks dan suara diperluas, termasuk menyertakan riwayat bookmark, kontak, dan web</li><li>▪ Kemampuan bagi para pengembang untuk menyertakan konten mereka pada hasil pencarian</li></ul>

	<ul style="list-style-type: none"> <li>▪ Mesin sintesis pengucapan multibahasa yang memungkinkan aplikasi Android tertentu mampu mengucapkan teks</li> <li>▪ Pencarian yang lebih mudah dan kemampuan untuk melihat cuplikan aplikasi di Android Market</li> <li>▪ Galeri, kamera, dan perekam video yang lebih terintegrasi, dengan akses kamera yang lebih cepat</li> <li>▪ Kemampuan memilih banyak foto untuk dihapus</li> <li>▪ Pembaruan dukungan teknologi bagi CDMA/EVDO, 802.1x, VPN, dan mesin pengucap teks</li> <li>▪ Dukungan bagi resolusi layar WVGA</li> <li>▪ Peningkatan kecepatan dalam pencarian dan aplikasi kamera</li> <li>▪ Perluasan kerangka kerja Gestur dan penambahan perangkat pengembangan GestureBuilder</li> </ul>
--	---

## Android 2.0 Eclair (API level 5)

Pada 26 Oktober 2009, SDK Android 2.0 – dinamai Eclair – dirilis, berbasis kernel Linux 2.6.29. Perubahan pada versi ini meliputi:

Versi	Tanggal Rilis	Fitur
2.0	26 Oktober 2009	<ul style="list-style-type: none"> <li>▪ Sinkronisasi akun diperluas, yang memungkinkan pengguna menambahkan beberapa akun untuk sinkronisasi surel dan kontak</li> <li>▪ Dukungan surel Microsoft Exchange, dengan kemampuan menjelajah surel dari beberapa akun dalam satu halaman</li> <li>▪ Dukungan Bluetooth 2.1</li> <li>▪ Kemampuan untuk memilih foto kontak dan opsi untuk memanggil, mengirim SMS atau surel kepada kontak yang bersangkutan</li> <li>▪ Kemampuan untuk mencari semua SMS dan MMS tersimpan, pesan terlama akan dihapus jika batas yang ditentukan sudah tercapai.</li> <li>▪ Menambahkan sejumlah fitur pada kamera, termasuk dukungan kilat (flash), perbesaran digital, mode skin, kejernihan, efek warna, dan fokus makro.</li> <li>▪ Peningkatan kecepatan mengetik pada papan ketik virtual, dengan dukungan kamus yang mempelajari penggunaan kata-kata, termasuk nama kontak sebagai saran</li> <li>▪ UI penjelajah web yang baru, dengan fitur bookmark thumbnail, double-tap zoom, dan dukungan bagi HTML5</li> </ul>



		<ul style="list-style-type: none"> <li>▪ Penyempurnaan tampilan agenda kalender; menampilkan status menghadiri untuk setiap undangan, dan kemampuan untuk mengundang tamu baru ke acara tertentu</li> <li>▪ Mengoptimalkan kecepatan perangkat lunak dan perubahan UI.</li> <li>▪ Dukungan bagi lebih banyak resolusi dan ukuran layar, dengan rasio kecerahan yang lebih baik</li> <li>▪ Peningkatan Google Maps 3.1.2</li> <li>▪ MotionEvent ditingkatkan untuk melacak aktivitas multitouch</li> <li>▪ Penambahan live wallpaper, yang menampilkan animasi pada latar belakang layar depan.</li> </ul>
--	--	---

Android 2.0.1 Eclair (API level 6)		
Versi	Tanggal Rilis	Fitur
2.0.1	3 Desember 2009	<ul style="list-style-type: none"> <li>▪ Perubahan API minor, perbaikan bug, dan perubahan kerangka kerja.</li> </ul>

Android 2.1 Eclair (API level 7)		
Versi	Tanggal Rilis	Fitur
2.1	12 Januari 2010	<ul style="list-style-type: none"> <li>▪ Perubahan kecil pada API dan perbaikan bug</li> </ul>

Android 2.2–2.2.3 Froyo (API level 8)		
<p>Pada 20 Mei 2010, SDK Android 2.2 (Froyo, singkatan untuk frozen yogurt) dirilis, berbasis kernel Linux 2.6.32.</p>		
Versi	Tanggal Rilis	Fitur
2.2	20 Mei 2010	<ul style="list-style-type: none"> <li>▪ Peningkatan kecepatan, penyimpanan, dan pengoptimalan kinerja</li> <li>▪ Peningkatan kecepatan aplikasi melalui kompilasi JIT</li> <li>▪ Integrasi mesin JavaScript V8 Chrome pada aplikasi penjelajah web</li> </ul>

		<ul style="list-style-type: none"> <li>▪ Dukungan bagi layanan Android Cloud to Device Messaging (C2DM)</li> <li>▪ Peningkatan dukungan Microsoft Exchange, termasuk kebijakan keamanan, pencarian otomatis, GAL, sinkronisasi kalender, dan pembersihan jarak jauh</li> <li>▪ Peningkatan peluncur aplikasi dengan jalan pintas ke Telepon/aplikasi penjelajah web</li> <li>▪ Meningkatkan fungsionalitas USB tethering dan hotspot Wi-Fi</li> <li>▪ Opsi untuk mematikan akses data pada jaringan seluler</li> <li>▪ Pembaruan aplikasi Market dengan menambahkan fitur pembaruan otomatis</li> <li>▪ Kontak dan panggilan suara bisa dibagikan melalui Bluetooth</li> <li>▪ Dukungan bagi Bluetooth-enabled car dan desk docks</li> <li>▪ Dukungan bagi sejumlah kata sandi alfanumerik</li> <li>▪ Dukungan pengunggahan berkas pada aplikasi penjelajah web</li> <li>▪ Penjelajah web bisa menampilkan animasi GIF</li> <li>▪ Dukungan pemasangan aplikasi pada penyimpanan eksternal</li> <li>▪ Dukungan Adobe Flash</li> <li>▪ Dukungan tampilan PPI (hingga 320 ppi), misalnya layar 4" 720p</li> <li>▪ Gestur pembesaran pada Galeri</li> </ul>
2.2.1	18 Januari 2011	<ul style="list-style-type: none"> <li>▪ Perbaiki bug, pembaruan keamanan, dan peningkatan kinerja</li> </ul>
2.2.2	22 Januari 2011	<ul style="list-style-type: none"> <li>▪ Perbaiki bug minor, termasuk bug SMS pada Nexus One</li> </ul>
2.2.3	21 November 2011	<ul style="list-style-type: none"> <li>▪ Penambahan dua patch keamanan</li> </ul>

### Android 2.3–2.3.2 Gingerbread (API level 9)

Pada tanggal 6 Desember 2010, SDK Android 2.3 (Gingerbread) dirilis, berbasis kernel Linux 2.6.35.

Perubahan pada versi ini termasuk:

Versi	Tanggal Rilis	Fitur
2.3	6 Desember 2010	<ul style="list-style-type: none"> <li>▪ Memperbarui desain antarmuka pengguna dengan meningkatkan kecepatan dan kesederhanaan</li> <li>▪ Dukungan bagi resolusi dan ukuran layar ekstra-besar (WXGA dan yang lebih tinggi)</li> <li>▪ Dukungan bagi telepon internet SIP VoIP</li> <li>▪ Masukan teks yang lebih cepat dan lebih intuitif pada papan ketik virtual, dengan meningkatkan akurasi, saran teks yang lebih baik, dan modus input suara</li> <li>▪ Peningkatan fungsi salin/tempel, memungkinkan pengguna untuk memilih kata dengan menekan dan menahan layar</li> <li>▪ Dukungan bagi Near Field Communication (NFC), memungkinkan pengguna untuk membaca tag NFC yang tertanam dalam poster, stiker, atau iklan</li> <li>▪ Efek audio baru seperti reverb, equalizer, virtualisasi penyuar kuping, dan bass boost</li> <li>▪ Download Manager baru, memudahkan pengguna untuk mengakses berkas yang diunduh dari penjelajah web, surel, ataupun dari aplikasi lainnya</li> <li>▪ Dukungan multi kamera pada perangkat, termasuk kamera depan, jika tersedia</li> <li>▪ Dukungan bagi pemutar video WebM/VP8, dan audio AAC</li> <li>▪ Peningkatan manajemen daya dengan peran lebih aktif dalam mengelola aplikasi yang beroperasi terlalu lama</li> <li>▪ Peningkatan dukungan bagi pengembangan kode asli</li> <li>▪ Peralihan dari YAFFS ke ext4 pada perangkat yang lebih baru</li> <li>▪ Peningkatan kualitas audio, grafis, dan masukan bagi pengembang permainan</li> <li>▪ Dukungan sensor yang lebih banyak (seperti giroskop dan barometer)</li> </ul>
2.3.1	Desember 2010	<ul style="list-style-type: none"> <li>▪ Perbaiki bug pada Google Nexus S</li> </ul>
2.3.2	Januari 2011	

### Android 2.3.3–2.3.7 Gingerbread (API level 10)

Versi	Tanggal rilis	Fitur
-------	---------------	-------

2.3.3	9 Februari 2011	<ul style="list-style-type: none"> <li>▪ Beberapa peningkatan dan perbaikan API</li> </ul>
2.3.4	28 April 2011	<ul style="list-style-type: none"> <li>▪ Dukungan bagi obrolan video atau suara menggunakan <u>Google Talk</u></li> <li>▪ Dukungan Open Accessory Library</li> </ul>
2.3.5	25 Juli 2011	<ul style="list-style-type: none"> <li>▪ Peningkatan kinerja jaringan pada Nexus S 4G, serta perbaikan lainnya</li> <li>▪ Perbaikan bug Bluetooth pada Samsung Galaxy S</li> <li>▪ Peningkatan kinerja aplikasi Gmail</li> <li>▪ Tampilan animasi bayangan untuk daftar bergulir</li> <li>▪ Peningkatan kinerja aplikasi kamera</li> <li>▪ Peningkatan daya tahan baterai</li> </ul>
2.3.6	2 September 2011	<ul style="list-style-type: none"> <li>▪ Perbaikan bug pada pencarian suara</li> <li>▪ (Pembaruan pada versi 2.3.6 menyebabkan bermasalahnya sambungan ke hotspot Wi-Fi pada sebagian besar perangkat Nexus S di Kanada. Google mengakui masalah ini dan memperbaikinya pada akhir September.)</li> </ul>
2.3.7	21 September 2011	<ul style="list-style-type: none"> <li>▪ Dukungan Google Wallet bagi Nexus S 4G</li> </ul>

## Android 3.0 Honeycomb (API level 11)

Pada 22 Februari 2011, SDK Android 3.0 (Honeycomb) – pembaruan pertama Android yang ditujukan hanya untuk komputer tablet – dirilis, berdasarkan kernel Linux 2.6.36. Perangkat pertama yang menggunakan versi ini adalah tablet Motorola Xoom, yang dirilis pada 24 Februari 2011. Fitur-fitur yang diperbarui antara lain:

Versi	Tanggal Rilis	Fitur
3.0	22 Februari 2011	<ul style="list-style-type: none"> <li>▪ Pengotimalan dukungan tablet dengan antarmuka pengguna virtual baru dan "holografis"</li> <li>▪ Menambahkan System Bar; memberi akses cepat pada notifikasi, status, dan tombol navigasi lunak, yang terdapat di bagian bawah layar</li> <li>▪ Menambahkan Action Bar; memberi akses pada opsi kontekstual, navigasi, widget, ataupun konten-konten lainnya di bagian atas layar</li> <li>▪ Multi-tugas sederhana – mengetuk Recent Apps pada System Bar akan memungkinkan pengguna untuk melihat cuplikan aplikasi yang sedang berjalan, dan secara cepat bisa berpindah dari suatu aplikasi ke aplikasi lainnya.</li> </ul>

	<ul style="list-style-type: none"> <li>▪ Mendesain ulang papan ketik, yang membuat mengetik jadi lebih cepat, efisien, dan akurat pada layar besar</li> <li>▪ Fungsi salin/tempel yang lebih sederhana</li> <li>▪ Beberapa tab penjelajah web mengganti jendela peramban, serta penambahan mode "incognito" baru yang memungkinkan pengguna anonim bisa menjelajah</li> <li>▪ Akses cepat pada eksposur, fokus, flash, zoom, kamera depan, dan fitur-fitur lainnya pada kamera</li> <li>▪ Kemampuan untuk melihat album dan koleksi lainnya dalam mode layar penuh pada galeri</li> <li>▪ UI kontak dua panel dan gulir cepat yang memudahkan pengguna dalam mengelola dan mencari kontak</li> <li>▪ UI surel dua panel yang menampilkan dan mengelola pesan dengan lebih mudah, memungkinkan pengguna untuk memilih satu atau lebih pesan</li> <li>▪ Dukungan obrolan video dengan menggunakan <u>Google Talk</u></li> <li>▪ Akselerasi <u>perangkat</u> keras</li> <li>▪ Dukungan prosesor multi-core</li> <li>▪ Kemampuan untuk mengenskripsikan semua data pengguna</li> <li>▪ Stak HTTPS ditingkatkan dengan <u>Indikasi Nama Server (SNI)</u></li> <li>▪ <u>Filesystem in User space (FUSE; modul kernel)</u></li> </ul>
--	--

### Android 3.1 Honeycomb (API level 12)

Versi	Tanggal Rilis	Fitur
3.1	10 Mei 2011	<ul style="list-style-type: none"> <li>▪ Perbaikan UI</li> <li>▪ Konektifitas bagi aksesoris USB</li> <li>▪ Memperluas daftar Recent Apps</li> <li>▪ Widget layar depan bisa diganti ukurannya</li> <li>▪ Dukungan bagi papan ketik eksternal dan tetikus</li> <li>▪ Dukungan bagi joystick dan gamepad</li> <li>▪ Dukungan bagi pemutar audio FLAC</li> <li>▪ Peningkatan kinerja Wi-Fi, koneksi Wi-Fi akan dipertahankan saat layar perangkat dalam keadaan mati</li> <li>▪ Dukungan bagi proksi HTTP untuk setiap titik akses Wi-Fi</li> </ul>

### Android 3.2 Honeycomb (API level 13)

Google TV generasi pertama dan kedua menggunakan Honeycomb 3.2.

Versi	Tanggal Rilis	Fitur
3.2	15 Juli 2011	<ul style="list-style-type: none"> <li>▪ Peningkatan dukungan perangkat keras, termasuk pengoptimalan lebih lanjut pada tablet</li> <li>▪ Peningkatan kemampuan aplikasi untuk mengakses berkas pada kartu SD, misalnya untuk sinkronisasi</li> <li>▪ Mode tampilan komparabilitas bagi aplikasi yang tidak dioptimalkan untuk resolusi layar tablet</li> <li>▪ Fungsi dukungan tampilan yang baru, memberikan pengembang lebih banyak kontrol atas tampilan layar pada perangkat Android yang berbeda<sup>[67]</sup></li> </ul>
3.2.1	20 September 2011	<ul style="list-style-type: none"> <li>▪ Perbaikan bug dan keamanan, stabilitas, dan peningkatan kinerja Wi-Fi</li> <li>▪ Pembaruan <u>Google Books</u></li> <li>▪ Peningkatan dukungan <u>Adobe Flash</u> pada penjelajah web</li> <li>▪ Peningkatan dukungan prediksi tulisan <u>Tionghoa</u></li> </ul>
3.2.2	30 Agustus 2011	<ul style="list-style-type: none"> <li>▪ Perbaikan bug dan perubahan minor lainnya pada Motorola Xoom 4G</li> </ul>
3.2.3		<ul style="list-style-type: none"> <li>▪ Perbaikan bug dan perubahan minor lainnya pada Motorola Xoom dan Xoom 4G</li> </ul>
3.2.4	December 2011	<ul style="list-style-type: none"> <li>▪ Dukungan "Pay as You Go" bagi tablet 3G dan 4G</li> </ul>
3.2.5	Januari 2012	<ul style="list-style-type: none"> <li>▪ Perbaikan bug dan perubahan minor lainnya pada Motorola Xoom dan Xoom 4G</li> </ul>
3.2.6	Februari 2012	<ul style="list-style-type: none"> <li>▪ Perbaikan masalah konektivitas data saat dalam mode penerbangan pada Motorola Xoom 4G edisi Amerika Serikat</li> </ul>

### Android 4.0–4.0.2 Ice Cream Sandwich (API level 14)

SDK Android 4.0.1 (Ice Cream Sandwich), berdasarkan kernel Linux 3.0.1, dirilis pada 19 Oktober 2011. Petinggi Google, Gabe Cohen, menyatakan bahwa Android 4.0 "secara teoretis kompatibel" dengan perangkat Android 2.3x yang diproduksi pada saat itu. Kode sumber untuk Android 4.0 tersedia pada tanggal 14 November 2011. Pembaruan pada versi ini antara lain:

Versi	Tanggal Rilis	Fitur
4.0	19	<ul style="list-style-type: none"> <li>▪ Tombol lunak tablet Android 3.x tersedia bagi penggunaan di <u>telepon pintar</u></li> </ul>

<p>Oktober 2011</p>	<ul style="list-style-type: none"> <li>▪ Pemisahan widget di tab baru, terletak pada layar yang bersebelahan dengan aplikasi</li> <li>▪ Pembuatan folder yang lebih mudah, dengan gaya drag-and-drop</li> <li>▪ Launcher yang bisa dikustomisasi</li> <li>▪ Peningkatan fitur pesan suara visual, dengan kemampuan untuk mempercepat atau memperlambat kecepatan pesan suara</li> <li>▪ Fungsi 'cubit untuk memperbesar' pada kalender</li> <li>▪ Pengintegrasian fungsi cuplikan layar (screenshot) dengan menekan dan menahan tombol daya dan volume-turun secara bersamaan</li> <li>▪ Perbaikan kesalahan koreksi pada papan ketik</li> <li>▪ Kemampuan untuk mengakses aplikasi secara langsung dari layar kunci (<i>lock screen</i>)</li> <li>▪ Perbaikan fungsi salin dan tempel</li> <li>▪ Integrasi suara yang lebih baik dan berkesinambungan</li> <li>▪ Mode buka kunci identifikasi wajah, fitur yang memungkinkan pengguna untuk membuka perangkat menggunakan perangkat lunak pengenalan wajah</li> <li>▪ Penambahan penjelajah web bawaan Chrome, mampu membuka page hingga 16 tab</li> <li>▪ Sinkronisasi otomatis pada penjelajah web dengan bookmark Chrome pengguna</li> <li>▪ Penambahan jenis huruf baru, Roboto</li> <li>▪ Penggunaan data bisa dibatasi, pengguna akan diperingatkan jika penggunaan data sudah mendekati batas tertentu, dan menonaktifkan data yang digunakan ketika batas tersebut terlampaui</li> <li>▪ Kemampuan untuk mematikan aplikasi yang menggunakan data di latar belakang</li> <li>▪ Peningkatan fungsi aplikasi kamera dengan fitur-fitur seperti zero shutter lag, time lapse settings, mode panorama, dan kemampuan memperbesar saat merekam video</li> <li>▪ Penambahan aplikasi pengedit foto bawaan</li> <li>▪ Tata letak galeri yang baru, bisa dikelola berdasarkan lokasi dan orang</li> <li>▪ Pemutakhiran aplikasi "People" dengan integrasi pada jejaring sosial</li> <li>▪ Android Beam, fitur komunikasi area dekat yang memungkinkan dilakukannya pertukaran jarak pendek bookmark web, info kontak, arah, video YouTube, dan lainnya</li> <li>▪ Dukungan format gambar WebP</li> </ul>
-------------------------	---

		<ul style="list-style-type: none"> <li>▪ Akselerasi perangkat keras UI</li> <li>▪ Wi-Fi Direct</li> <li>▪ Merekam video 1080p bagi perangkat Android tertentu</li> <li>▪ Modul kernel Android VPN Framework (AVF) dan TUN (bukan TAP). Sebelum versi 4.0, perangkat lunak VPN membutuhkan <i>rooting</i>.</li> </ul>
4.0.1	21 Oktober 2011	<ul style="list-style-type: none"> <li>▪ Perbaiki bug minor pada Samsung Galaxy Nexus</li> </ul>
4.0.2	28 November 2011	<ul style="list-style-type: none"> <li>▪ Perbaiki bug minor pada Verizon Galaxy Nexus, peluncurannya di AS ditunda hingga Desember 2011.</li> </ul>

### Android 4.0.3–4.0.4 Ice Cream Sandwich (API level 15)

Ice Cream Sandwich adalah versi terakhir yang mendukung [Flash player Adobe Systems](#).

Versi	Tanggal Rilis	Fitur
4.0.3	16 Desember 2011	<ul style="list-style-type: none"> <li>▪ Sejumlah perbaikan bug dan optimisasi</li> <li>▪ Peningkatan fungsionalitas grafis, basis data, pemeriksa ejaan, dan Bluetooth</li> <li>▪ API baru bagi pengembang, termasuk API arus sosial di penyedia Kontak</li> <li>▪ Peningkatan fitur kalender</li> <li>▪ Fitur baru pada aplikasi kamera yang meningkatkan stabilisasi video dan resolusi QVGA</li> <li>▪ Perbaikan aksesibilitas seperti peningkatan akses konten bagi pembaca layar</li> </ul>
4.0.4	29 Maret 2012	<ul style="list-style-type: none"> <li>▪ Peningkatan stabilitas</li> <li>▪ Kinerja kamera yang lebih baik</li> <li>▪ Rotasi layar yang lebih halus</li> <li>▪ Peningkatan fitur pengidentifikasi nomor telepon</li> </ul>

### Android 4.1 Jelly Bean (API level 16)

Google mengumumkan Android 4.1 (Jelly Bean) dalam konferensi [Google I/O](#) pada tanggal 27 Juni 2012. Berdasarkan kernel Linux 3.0.31, Jelly Bean adalah pembaruan penting yang bertujuan untuk meningkatkan fungsi dan kinerja antarmuka pengguna



(UI). Pembaruan ini diwujudkan dalam "Proyek Butter", perbaikan ini termasukantisipasi sentuh, triple buffering, perpanjangan waktu vsync, dan peningkatan frame rate hingga 60 fps untuk menciptakan UI yang lebih halus. Android 4.1 Jelly Bean dirilis untuk Android Open Source Project pada tanggal 9 Juli 2012. Perangkat pertama yang menggunakan sistem operasi ini adalah tablet Nexus 7, yang dirilis pada 13 Juli 2012.

Versi	Tanggal Rilis	Fitur
4.1	9 Juli 2012	<ul style="list-style-type: none"> <li>▪ Antarmuka pengguna yang lebih halus:               <ul style="list-style-type: none"> <li>• Waktu vsync pada animasi UI dikelola oleh kerangka kerja Android, termasuk reaksi aplikasi, efek sentuh, komposisi layar, dan penyegaran tampilan</li> <li>• Triple buffering pada grafis</li> </ul> </li> <li>▪ Peningkatan aksesibilitas</li> <li>▪ Teks dua bahasa dan dukungan bahasa lainnya</li> <li>▪ Papan ketik yang bisa dimodifikasi oleh pengguna</li> <li>▪ Perluasan notifikasi</li> <li>▪ Kemampuan untuk mematikan notifikasi pada aplikasi tertentu</li> <li>▪ Shortcut dan widget secara otomatis bisa disusun ulang atau diatur ukurannya</li> <li>▪ Transfer data Bluetooth bagi Android Beam</li> <li>▪ Diktasi suara luring</li> <li>▪ Tablet dengan layar kecil bisa menyesuaikan tata letak antarmuka dan layar depan seperti pada telepon pintar</li> <li>▪ Peningkatan pencarian suara</li> <li>▪ Peningkatan aplikasi kamera</li> <li>▪ Google Wallet (pada Nexus 7)</li> <li>▪ Foto kontak Google+ resolusi tinggi</li> <li>▪ Aplikasi pencarian Google Now</li> <li>▪ Audio multi-saluran</li> <li>▪ Audio USB (bagi suara eksternal DACs)</li> <li>▪ Audio chaining</li> <li>▪ Penjelajah web bawaan Android diganti dengan Google Chrome pada perangkat Android pra-instal</li> <li>▪ Kemampuan untuk menambahkan widget aplikasi tanpa akses root</li> </ul>
4.1.1	23 Juli 2012	<ul style="list-style-type: none"> <li>▪ Perbaikan bug pada Nexus 7 terkait dengan ketidakmampuan untuk mengubah orientasi layar pada aplikasi apapun</li> </ul>
4.1.2	9	<ul style="list-style-type: none"> <li>▪ Dukungan rotasi layar depan/layar kunci pada Nexus 7</li> <li>▪ Gerakan satu jari untuk menampilkan/menyembunyikan notifikasi</li> </ul>

Oktober 2012	<ul style="list-style-type: none"> <li>▪ Perbaiki bug dan peningkatan kinerja</li> </ul>
--------------	--

## Android 4.2 Jelly Bean (API level 17)

Google berencana merilis Jelly Bean 4.2 pada sebuah acara di New York City pada 29 Oktober 2012, tapi dibatalkan karena Badai Sandy. Jelly Bean 4.2 didasarkan pada kernel Linux 3.4.0, dan pertama kali digunakan pada Nexus 4 LG dan Nexus 10 Samsung, yang dirilis pada 13 November 2012.

Versi	Tanggal Rilis	Fitur
4.2	13 November 2012	<ul style="list-style-type: none"> <li>▪ Foto panorama "Photo Sphere"</li> <li>▪ Papan kunci dengan gestur ketikan</li> <li>▪ Peningkatan kinerja layar kunci, termasuk dukungan widget dan kemampuan untuk membuka aplikasi kamera secara langsung</li> <li>▪ Kontrol daya notifikasi ("Pengaturan Cepat")</li> <li>▪ Screensaver "Daydream"</li> <li>▪ Akun pengguna lebih dari satu (hanya tablet)</li> <li>▪ Dukungan tampilan nirkabel (Miracast)</li> <li>▪ Perbaiki aksesibilitas: ketuk-tiga kali untuk untuk memperbesar seluruh layar, pan and zoom dengan dua jari, serta keluaran suara dan navigasi Gesture Mode bagi pengguna tunanetra</li> <li>▪ Aplikasi jam baru, disertai dengan jam dunia, stop watch, dan penghitung waktu mundur</li> <li>▪ Seluruh perangkat menggunakan tata letak antarmuka yang sama, terlepas dari ukuran layar</li> <li>▪ SELinux dan VPN yang selalu terhubung</li> <li>▪ Konfirmasi SMS premium dan Group Messaging</li> </ul>
4.2.1	27 November 2012	<ul style="list-style-type: none"> <li>▪ Perbaiki bug pada aplikasi "People"</li> <li>▪ Penambahan gamepad dan joysticks Bluetooth yang mendukung HID</li> </ul>
4.2.2	11 Februari 2013	<ul style="list-style-type: none"> <li>▪ Perbaiki bug pada streaming audio Bluetooth</li> <li>▪ Notifikasi unduhan yang menampilkan persentase dan waktu perkiraan unduhan</li> <li>▪ Nada baru bagi pengisian nirkabel dan baterai lemah</li> <li>▪ Animasi galeri baru yang memungkinkan pemuatan jadi lebih cepat</li> <li>▪ Perbaiki USB debug</li> <li>▪ Perbaiki bug dan peningkatan kinerja</li> </ul>

## Android 4.3 Jelly Bean (API level 18)

Google merilis Jelly Bean 4.3 pada 24 Juli 2013 di [San Francisco](#). Kebanyakan perangkat Nexus menerima pembaruan dengan segera. [Nexus 7 generasi kedua](#) adalah perangkat pertama yang menggunakan sistem operasi ini. Sebuah pembaruan minor dirilis pada tanggal 22 Agustus 2013.

Versi	Tanggal Rilis	Fitur
4.3	24 Juli 2013	Bagian ini memerlukan pengembangan

## Android 4.4 KitKat (API level 19)

Google mengumumkan Android 4.4 [KitKat](#) (dinamai dengan izin dari [Nestlé](#) dan [Hershey](#)) pada 3 September 2013, dengan tanggal rilis 31 Oktober 2013. Sebelumnya, rilis berikutnya setelah Jelly Bean diperkirakan akan diberi nomor 5.0 dan dinamai 'Key Lime Pie'.

Ver si	Tanggal Rilis	Fitur
4.4	31 Oktober 2013	<ul style="list-style-type: none"><li>▪ Pembaruan antarmuka dengan bar status dan navigasi transparan pada layar depan.</li><li>▪ Optimasi kinerja pada perangkat dengan spesifikasi yang lebih rendah</li><li>▪ Kerangka kerja pencetakan</li><li>▪ NFC Host Card Emulation sebagai emulator kartu pintar</li><li>▪ WebViews berbasis Chromium</li><li>▪ Perluasan fungsionalitas bagi layanan pendengar notifikasi</li><li>▪ API umum untuk mengembangkan dan mengelola klien pesan teks, kemampuan untuk menentukan aplikasi SMS standar.</li><li>▪ Kerangka kerja baru untuk transisi UI</li><li>▪ Kerangka kerja akses penyimpanan untuk mengambil konten dan dokumen dari sumber lain</li><li>▪ Sensor batching, Step Detector, dan Counter API</li><li>▪ Peningkatan tampilan mode layar penuh, tombol perangkat lunak dan status bar bisa diakses dari tepi dengan cara menggesek.</li><li>▪ Penyeimbang audio, pemantauan audio, dan peningkatan suara audio</li></ul>

	<ul style="list-style-type: none"> <li>▪ Perekam aktivitas layar yang terintegrasi</li> <li>▪ Inframerah</li> <li>▪ Peningkatan aksesibilitas API</li> <li>▪ Mesin virtual eksperimental baru, ART</li> <li>▪ Dukungan Bluetooth Message Access Profile (MAP)</li> </ul>
--	--

### Android 5.0 Lollipop (API level 21)

Versi	Tanggal Rilis	Fitur
5.0		<p>Android 5.0 pertama kali diperkenalkan di bawah codename "Android L" pada 25 Juni 2014 selama presentasi keynote pada konferensi pengembang Google I/O. Di samping Lollipop, presentasi difokuskan pada sejumlah platform Android yang berorientasi dan teknologi baru, termasuk Android TV, pada platform Android Auto, dpt dipakai pada platform komputasi Android Wear, dan platform pelacakan kesehatan Google Fit.</p> <p>Bagian dari presentasi didedikasikan untuk bahasa desain cross-platform baru yang disebut sebagai "material design". Memperluas pada "kartu" motif pertama kali terlihat di Google Now, adalah desain dengan peningkatan penggunaan layout berbasis grid, animasi dan transisi responsif, padding, dan efek kedalaman seperti pencahayaan dan bayangan.</p> <ul style="list-style-type: none"> <li>▪ Desain antarmuka (tampilan) yang dinamakan "Material Design".</li> <li>▪ 64-bit ART compiler</li> <li>▪ Project volta, yang berguna untuk meningkatkan daya hidup baterai 30% lebih tahan lama.</li> <li>▪ 'factory reset protection'. Fitur ini berguna ketika smartphone hilang, ia tidak bisa direset ulang tanpa memasukkan id google dan kata sandi (password).</li> </ul>

### Android 6.0 Marshmallow (API level 23)

Versi	Tanggal Rilis	Fitur
6.0		<p>Android Marshmallow memperkenalkan model izin yang didesain ulang: sekarang ada hanya delapan kategori izin, dan aplikasi yang tidak lagi secara otomatis diberikan semua hak akses mereka ditentukan pada waktu instalasi. Sebuah sistem opt-in sekarang digunakan, di mana pengguna akan diminta</p>

		<p>untuk memberikan atau menolak izin individu (seperti kemampuan untuk mengakses kamera atau mikrofon) untuk aplikasi ketika mereka dibutuhkan. Aplikasi mengingat hibah izin mereka, dan mereka dapat disesuaikan oleh pengguna setiap saat. Model izin baru akan digunakan hanya oleh aplikasi yang dikompilasi untuk Marshmallow menggunakan kit pengembangan perangkat lunak (SDK) tersebut, sementara semua aplikasi lainnya akan terus menggunakan model izin sebelumnya.</p> <p>Marshmallow juga memiliki skema manajemen daya baru bernama Doze yang mengurangi tingkat aktivitas aplikasi latar belakang saat perangkat menentukan bahwa itu tidak sedang aktif ditangani oleh pengguna, yang, menurut Google, mengganggakan pemakaian baterai perangkat. Hal ini juga memperkenalkan pilihan untuk mengatur ulang semua pengaturan jaringan, tersedia untuk pertama kalinya pada Android, yang membersihkan pengaturan terkait jaringan untuk WI-FI, Bluetooth dan koneksi seluler.</p> <p>Android Marshmallow memberikan dukungan asli untuk pengenalan sidik jari, memungkinkan penggunaan sidik jari untuk membuka perangkat dan otentikasi Play Store dan pembelian Android Pay; API standar juga tersedia untuk melaksanakan otentikasi berbasis sidik jari dalam aplikasi lain.</p> <p>Android Marshmallow mendukung USB Type-C, termasuk kemampuan untuk menginstruksikan perangkat untuk mengisi daya perangkat lain melalui USB. Marshmallow juga memperkenalkan "pranala yang diverifikasi" yang dapat dikonfigurasi untuk membuka langsung dalam aplikasi tertentu mereka tanpa petunjuk pengguna lanjut.</p> <p>Versi API Android yang disediakan oleh Marshmallow adalah 23. Alat pengembang Android Marshmallow tersedia di Pengelola SDK di bawah tingkat API "MNC".</p>
--	--	--

## Android 7.0 Nougat (API 24)

Android "Nougat" (kode nama N dalam pengembangan) adalah rilis 7.0 besar dari sistem operasi Android. Ini pertama kali dirilis sebagai pratinjau pengembang pada tanggal 9 Maret 2016, dengan gambar pabrik untuk perangkat Nexus saat ini, serta dengan "Program Beta Beta" baru yang memungkinkan perangkat yang didukung ditingkatkan versinya ke versi Android Nougat melalui over-the-air update. Rilis terakhir adalah pada tanggal 22 Agustus 2016. Pratinjau akhir pembuatannya dirilis pada tanggal 18 Juli 2016, dengan nomor bangunan NPD90G.

Versi	Tanggal Rilis	Fitur
7.0	22 Agustus, 2016	<ul style="list-style-type: none"> <li>▪ Unicode 9.0 emoji dan skin modifier support (memperlihatkan subset API ICU4J).</li> <li>▪ Kemampuan untuk menampilkan kalibrasi warna</li> <li>▪ Kemampuan untuk layar zoom</li> <li>▪ Kemampuan untuk beralih aplikasi dengan menekan ganda dalam tombol ikhtisar</li> <li>▪ Bagian Informasi Darurat Ditambahkan</li> <li>▪ Menambahkan tombol "Hapus Semua" di layar Ikhtisar</li> <li>▪ Partisi sistem lain, yang diperbarui saat tidak digunakan, memungkinkan pembaruan sistem tanpa sensor</li> <li>▪ Platform virtual reality dunia maya (antarmuka VR)</li> <li>▪ Peningkatan fungsionalitas Doze, yang bertujuan untuk memperpanjang masa pakai baterai</li> <li>▪ Perbaikan file browser dan Pilihan Quick Settings lagi</li> <li>▪ Dukungan multi-window, yang mendukung aplikasi apung di tata letak desktop</li> <li>▪ Mode Data Saver baru, yang dapat memaksa aplikasi mengurangi pemakaian bandwidth</li> <li>▪ New JIT Compiler, membuat instalasi aplikasi 75 persen lebih cepat dan pengurangan 50 persen dalam ukuran kode yang dikompilasi</li> <li>▪ Hanya dalam Time (JIT) kompilator dengan kode profil untuk ART, yang memungkinkannya terus meningkatkan kinerja aplikasi Android saat berjalan</li> <li>▪ Dukungan gambar dalam gambar untuk Android TV</li> <li>▪ Debut pemberitahuan didesain ulang, yang menampilkan akses instan ke setting tertentu</li> <li>▪ Layar Ikhtisar yang didesain ulang</li> <li>▪ Kartu pemberitahuan yang diganti dengan lembar pemberitahuan</li> <li>▪ Setelan laci navigasi aplikasi</li> <li>▪ Vulkan 3D rendering API</li> <li>▪ Beberapa Device Locales</li> </ul>

## Android 7.0 Nougat (API 24)

Android "Nougat" (kode nama N dalam pengembangan) adalah rilis 7.0 besar dari sistem operasi Android. Ini pertama kali dirilis sebagai pratinjau pengembang pada tanggal 9 Maret 2016, dengan gambar pabrik untuk perangkat Nexus saat ini, serta

dengan "Program Beta Beta" baru yang memungkinkan perangkat yang didukung ditingkatkan versinya ke versi Android Nougat melalui over-the-air update. Rilis terakhir adalah pada tanggal 22 Agustus 2016. Pratinjau akhir pembuatannya dirilis pada tanggal 18 Juli 2016, dengan nomor bangunan NPD90G.

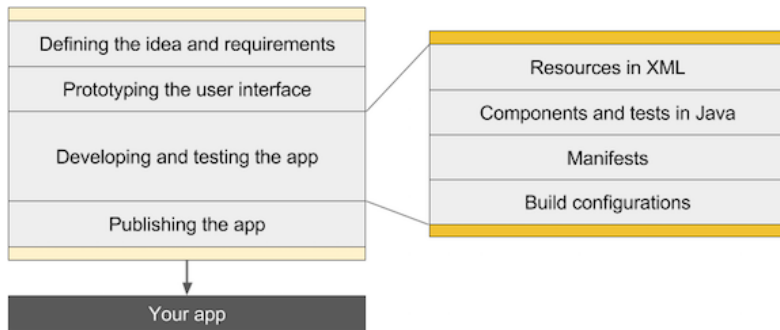
Versi	Tanggal Rilis	Fitur
7.0	22 Agustus 2016	<ul style="list-style-type: none"> <li>▪ Unicode 9.0 emoji dan skin modifier support (dan memperlihatkan subset API ICU4J).</li> <li>▪ Kemampuan untuk menampilkan kalibrasi warna</li> <li>▪ Kemampuan untuk layar zoom</li> <li>▪ Kemampuan untuk beralih aplikasi dengan menekan ganda dalam tombol ikhtisar</li> <li>▪ Bagian Informasi Darurat Ditambahkan</li> <li>▪ Menambahkan tombol "Hapus Semua" di layar Ikhtisar</li> <li>▪ Partisi sistem lain, yang diperbarui saat tidak digunakan, memungkinkan pembaruan sistem tanpa sensor</li> <li>▪ Platform virtual reality dunia maya (antarmuka VR)</li> <li>▪ Peningkatan fungsionalitas Doze, yang bertujuan untuk memperpanjang masa pakai baterai</li> <li>▪ Perbaikan file browser</li> <li>▪ Pilihan Quick Settings lagi</li> <li>▪ Dukungan multi-window, yang mendukung aplikasi apung di tata letak desktop</li> <li>▪ Mode Data Saver baru, yang dapat memaksa aplikasi mengurangi pemakaian bandwidth</li> <li>▪ New JIT Compiler, membuat instalasi aplikasi 75 persen lebih cepat dan pengurangan 50 persen dalam ukuran kode yang dikompilasi</li> <li>▪ Hanya dalam Time (JIT) kompilator dengan kode profil untuk ART, yang memungkinkannya terus meningkatkan kinerja aplikasi Android saat berjalan</li> <li>▪ Dukungan gambar dalam gambar untuk Android TV</li> <li>▪ Debut pemberitahuan didesain ulang, yang menampilkan akses instan ke setting tertentu</li> <li>▪ Layar Ikhtisar yang didesain ulang</li> <li>▪ Kartu pemberitahuan yang diganti dengan lembar pemberitahuan</li> <li>▪ Setelan laci navigasi aplikasi</li> <li>▪ Vulkan 3D rendering API</li> </ul> <p>Beberapa Device Locales</p>

## 2.3. Pembuatan Aplikasi Android

Materi ini menjelaskan cara mengembangkan aplikasi menggunakan Android Studio Integrated Development Environment (IDE).

### a. Proses development

Proyek aplikasi Android dimulai dengan ide dan definisi persyaratan yang diperlukan untuk mewujudkan ide tersebut. Saat proyek berjalan, maka akan melalui desain, development, dan pengujian.



Gambar 14. Diagram proses development high-level

Diagram di atas adalah gambar proses development tingkat tinggi, dengan langkah-langkah berikut:

- 1) *Mendefinisikan ide dan persyaratannya*: Sebagian besar aplikasi dimulai dengan ide tentang hal yang harus dilakukan, yang didukung pasar dan penelitian pengguna. Selama tahap ini, persyaratan aplikasi didefinisikan.
- 2) *Membuat prototipe antarmuka pengguna*: Gunakan gambar, rekaan, dan prototipe untuk menampilkan tampilan antarmuka pengguna nantinya, dan cara kerjanya.
- 3) *Mengembangkan dan menguji aplikasi*: Aplikasi terdiri dari satu atau beberapa activity. Untuk setiap activity, Anda bisa menggunakan Android Studio untuk melakukan hal berikut, tidak dalam urutan tertentu:
  - o *Buat layout*: Tempatkan elemen UI pada layar di layout, dan tetapkan resource string serta item menu, menggunakan Extensible Markup Language (XML).
  - o *Tulis kode Java*: Buat referensi kode sumber untuk komponen dan pengujian, serta gunakan alat pengujian dan debug.
  - o *Daftarkan activity*: Deklarasikan activity dalam file manifest.



- *Definisikan versi:* Gunakan konfigurasi pembangunan default atau buat pembangunan khusus untuk versi aplikasi yang berbeda.
- 4) *Mempublikasikan aplikasi:* Rakit APK (file paket) final dan distribusikan melalui saluran seperti Google Play.

## b. Menggunakan Android Studio

Android Studio menyediakan alat untuk pengujian, dan mempublikasikan tahap proses development, serta lingkungan development terpadu untuk membuat aplikasi bagi semua perangkat Android. Lingkungan development menyertakan kode template dengan kode contoh untuk fitur aplikasi umum, alat pengujian dan kerangka kerja yang banyak, dan sistem pembangunan yang fleksibel. Android Studio menyediakan alat untuk pengujian, dan mempublikasikan tahap proses development, serta lingkungan development terpadu untuk membuat aplikasi bagi semua perangkat Android. Lingkungan development menyertakan kode template dengan kode contoh untuk fitur aplikasi umum, alat pengujian dan kerangka kerja yang banyak, dan sistem pembangunan yang fleksibel.

- ⇒ Di jendela **Welcome to Android Studio**, klik **Start a new Android Studio project**. Atau bila Anda sudah membuka project, pilih **File > New Project**.



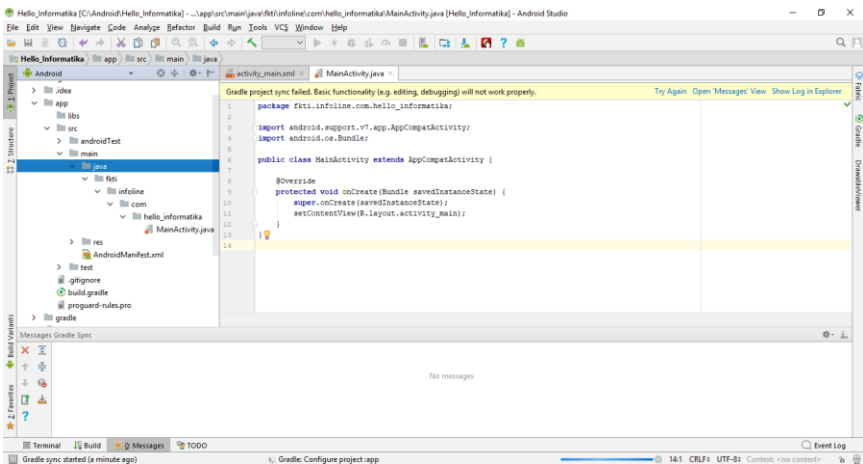
Gambar 15. Screenshot antarmuka welcome to android studio

- ⇒ Di layar **New Project** masukkan nilai-nilai berikut:

- **Application Name:** "My First App"
- **Company Domain:** "example.com"

Anda mungkin ingin mengubah lokasi project, tetapi jangan ubah opsi yang lain.

- ⇒ Klik **Next**.
  - ⇒ Di layar **Target Android Devices**, biarkan nilai-nilai default dan klik **Next**.
  - ⇒ Di layar **Add an Activity to Mobile**, pilih **Empty Activity** dan klik **Next**.
  - ⇒ Di layar **Configure Activity**, biarkan nilai-nilai default dan klik **Finish**.
- Setelah beberapa pemrosesan, Android Studio akan membuka IDE.



Gambar 16. Antarmuka IDE aplikasi

Pertama-tama, pastikan jendela **Project** terbuka (pilih **View > Tool Windows > Project**) dan tampilan **Android** dipilih dari menu drop-down di bagian atas jendela tersebut. Kemudian Anda bisa melihat file berikut:

- **app > java > com.example.myfirstapp > MainActivity.java**

Ini adalah activity utama (titik masuk aplikasi Anda). Saat Anda membangun dan menjalankan aplikasi, sistem akan meluncurkan sebuah instance dari Activity dan memuat layout-nya.

- **app > res > layout > activity\_main.xml**

File XML ini mendefinisikan layout untuk UI activity. File ini mengandung elemen TextView dengan teks "Hello world!".

- **app > manifests > AndroidManifest.xml**

File manifest menjelaskan karakteristik dasar aplikasi dan mendefinisikan masing-masing komponennya.

- **Gradle Scripts > build.gradle**

Anda akan melihat dua file dengan nama ini: satu untuk project dan satu lagi untuk modul "aplikasi". Setiap modul memiliki file `build.gradle` sendiri, tetapi project ini sekarang hanya memiliki satu modul. Anda akan sering bekerja dengan file `build.gradle` modul untuk mengonfigurasi cara alat Gradle mengompilasi dan membangun aplikasi Anda. Untuk informasi selengkapnya tentang file ini.

### c. Mengonfigurasi Pembangunan

Sistem pembangunan Android mengompilasi resource aplikasi serta kode sumber, dan mengemasnya ke APK sehingga bisa Anda uji, terapkan, tanda tangani, dan distribusikan. Android Studio menggunakan Gradle, sebuah toolkit pembangunan lanjutan, untuk mengotomatisasi dan mengelola proses pembangunan, sekaligus memungkinkan Anda untuk menentukan konfigurasi pembangunan khusus yang fleksibel. Setiap konfigurasi pembangunan bisa mendefinisikan set kode dan resource-nya sendiri, sekaligus menggunakan kembali bagian-bagian yang sama untuk semua versi aplikasi Anda. Android Plugin untuk Gradle bekerja dengan toolkit pembangunan agar proses dan setelan bisa dikonfigurasi yang spesifik untuk membuat dan menguji aplikasi Android.

Plugin Android dan Gradle berjalan secara independen dari Android Studio. Ini berarti bahwa Anda bisa membuat aplikasi Android dari dalam Android Studio, baris perintah pada mesin, atau pada mesin yang tidak dipasang Android Studio (seperti server integrasi berkesinambungan). Jika Anda tidak menggunakan Android Studio.



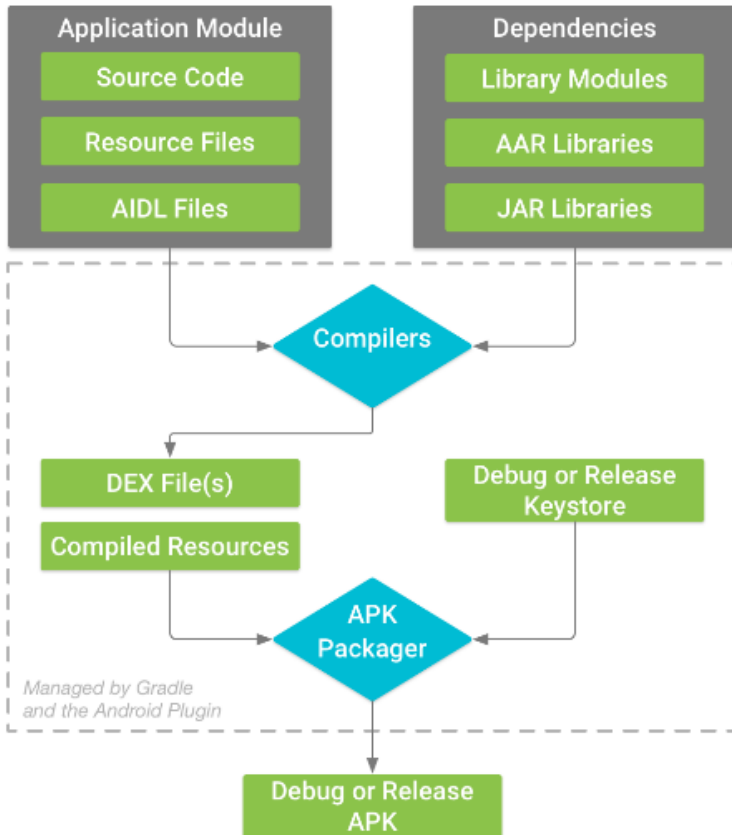
**Catatan:** Karena plugin Android dan Gradle berjalan secara independen dari Android Studio, Anda perlu memperbarui alat pembangunan secara terpisah.

Fleksibilitas sistem pembangunan Android memungkinkan Anda melakukan konfigurasi pembangunan khusus tanpa memodifikasi file sumber inti aplikasi. Bagian ini membantu Anda memahami bagaimana sistem

pembangunan Android bekerja, dan bagaimana itu bisa membantu Anda menyesuaikan dan mengotomatisasi beberapa konfigurasi pembangunan. Jika Anda hanya ingin mengetahui selengkapnya tentang menerapkan aplikasi.

#### d. Proses Pembangunan

Proses pembangunan melibatkan banyak alat dan pemrosesan yang mengonversi proyek Anda menjadi Android Application Package (APK). Proses pembangunan sangat fleksibel, sehingga berguna untuk memahami beberapa kejadian dalam detail pemrosesannya.



Gambar 17. Proses Pembangunan Project Android

Proses pembangunan untuk modul aplikasi Android khusus, seperti yang ditunjukkan pada gambar diatas, ikuti langkah-langkah berikut:

- Compiler mengonversi kode sumber menjadi file DEX (Dalvik Executable), yang menyertakan bytecode yang berjalan pada perangkat Android, serta lainnya menjadi resource yang dikompilasi.
- APK Packager menggabungkan file DEX dan resource yang dikompilasi ke dalam APK tunggal. Namun, sebelum aplikasi Anda bisa diterapkan dan digunakan pada perangkat Android, APK harus ditandatangani.
- APK Packager menandatangani APK Anda menggunakan keystore debug atau rilis:
  - a. Jika Anda membuat sebuah versi debug aplikasi, yaitu, aplikasi yang Anda tujukan hanya untuk pengujian dan pembuatan profil, packager menandatangani aplikasi dengan keystore debug. Android Studio secara otomatis mengonfigurasi proyek baru dengan keystore debug.
  - b. Jika Anda membangun versi rilis aplikasi yang ditujukan untuk dirilis keluar, packager menandatangani aplikasi Anda dengan keystore rilis. Untuk membuat keystore rilis.
- Sebelum menghasilkan APK final, packager menggunakan alat [zipalign](#) untuk mengoptimalkan aplikasi Anda agar hemat memori ketika berjalan pada perangkat.

Pada akhir dari proses pembangunan, Anda akan mendapatkan sebuah APK debug atau APK rilis dari aplikasi yang bisa Anda gunakan untuk menerapkan, menguji, atau merilis untuk pengguna eksternal.

### 1) Konfigurasi Pembangunan Khusus

Plugin Android dan Gradle membantu Anda mengonfigurasi aspek-aspek berikut dari pembangunan Anda:

#### *Tipe Pembangunan*

Tipe pembangunan mendefinisikan properti tertentu yang digunakan Gradle ketika membuat serta memaket aplikasi Anda, dan biasanya dikonfigurasi untuk berbagai daur siklus development Anda. Misalnya, tipe pembangunan versi debug mengaktifkan opsi debug dan menandatangani APK dengan kunci debug, sedangkan tipe pembangunan versi rilis bisa menciutkan, menyamarkan, dan menandatangani APK dengan kunci rilis untuk distribusi. Anda harus menentukan setidaknya satu tipe pembangunan untuk membuat aplikasi—Android Studio membuat tipe pembangunan versi rilis dan debug secara default. Untuk mulai menyesuaikan setelah pemaketan untuk aplikasi Anda.

## *Ragam Produk*

Ragam produk merepresentasikan versi berbeda dari aplikasi yang mungkin Anda rilis untuk pengguna, seperti versi aplikasi gratis dan berbayar. Anda bisa menyesuaikan ragam produk agar menggunakan kode dan resource yang berbeda, sekaligus berbagi dan menggunakan kembali bagian-bagian yang umum untuk semua versi aplikasi. Ragam produk bersifat opsional dan Anda harus membuatnya secara manual. Untuk mulai membuat versi aplikasi yang berbeda.

## *Varian Pembangunan*

Varian pembangunan adalah produk persilangan dari tipe pembangunan dan ragam produk, dan merupakan konfigurasi yang digunakan Gradle untuk membuat aplikasi Anda. Dengan menggunakan varian pembangunan, Anda bisa membuat versi debug dari ragam produk selama development, atau menandatangani versi rilis dari ragam produk untuk distribusi. Meskipun tidak mengonfigurasi varian pembangunan secara langsung, Anda mengonfigurasi tipe pembangunan dan ragam produk yang membentuknya. Membuat tipe pembangunan atau ragam produk tambahan juga akan menciptakan varian pembangunan tambahan. Untuk mempelajari cara membuat dan mengelola varian pembangunan.

*Entri Manifes* : Anda bisa menentukan nilai untuk beberapa properti dari file manifes dalam konfigurasi varian pembangunan. Nilai pembangunan ini menggantikan nilai yang ada dalam file manifes. Ini berguna jika Anda ingin menghasilkan beberapa APK untuk modul dan masing-masing file apk memiliki nama aplikasi yang berbeda, versi SDK minimum, atau versi SDK target. Bila terdapat beberapa manifes, Gradle akan menggabungkan setelan manifes.

*Dependensi* : Sistem pembangunan mengelola dependensi proyek dari sistem file lokal dan dari repositori jarak jauh. Hal ini akan menjaga Anda dari keharusan menelusuri, mengunduh, dan menyalin secara manual paket biner dependensi ke dalam direktori proyek Anda. Untuk mengetahui selengkapnya, pelajari cara Mendeklarasikan Dependensi.

*Penandatanganan* : Sistem pembangunan memungkinkan Anda untuk menentukan setelan penandatanganan dalam konfigurasi pembangunan, dan bisa secara otomatis menandatangani APK selama proses pembangunan. Sistem pembangunan menandatangani versi debug dengan kunci dan sertifikat default menggunakan kredensial yang dikenal untuk menghindari permintaan sandi pada waktu pembangunan. Sistem pembangunan tidak menandatangani versi rilis kecuali Anda secara

eksplisit menentukan konfigurasi penandatanganan untuk pembangunan ini. Jika tidak memiliki kunci rilis, Anda bisa menghasilkannya seperti yang dijelaskan dalam Menandatangani Aplikasi Anda.

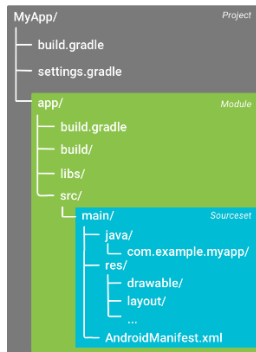
*ProGuard* : Sistem pembangunan memungkinkan Anda untuk menentukan file aturan ProGuard yang berbeda untuk setiap varian pembangunan. Sistem pembangunan bisa menjalankan ProGuard untuk menciutkan dan menyamarkan kelas-kelas Anda selama proses pembangunan.

*Pemisahan APK* : Sistem pembangunan memungkinkan Anda untuk secara otomatis membuat APK berbeda yang masing-masing hanya berisi kode dan resource yang dibutuhkan untuk kepadatan layar tertentu atau Application Binary Interface (ABI).

#### e. File Konfigurasi Pembangunan

Pembuatan konfigurasi pembangunan khusus mengharuskan Anda untuk membuat perubahan terhadap satu atau beberapa file konfigurasi pembangunan, atau file `build.gradle`. File-file teks biasa ini menggunakan Domain Specific Language (DSL) untuk menggambarkan dan memanipulasi logika pembangunan menggunakan Groovy, yang merupakan bahasa dinamis untuk Java Virtual Machine (JVM). Anda tidak perlu mengetahui Groovy untuk mulai mengonfigurasi pembangunan karena Android Plugin untuk Gradle memperkenalkan sebagian besar elemen DSL yang Anda butuhkan. Untuk mengetahui selengkapnya tentang DSL plugin Android.

Ketika memulai sebuah proyek baru, Android Studio secara otomatis membuat beberapa file ini untuk Anda, seperti yang ditunjukkan pada gambar, dan akan mengisinya berdasarkan *default yang logis*.



Gambar 18. Struktur Project Modul Android.

Ada beberapa file konfigurasi pembangunan Gradle yang merupakan bagian dari struktur proyek standar untuk aplikasi Android. Sebelum Anda bisa mulai mengonfigurasi pembangunan, penting untuk memahami cakupan dan tujuan dari setiap file, dan elemen DSL dasar yang harus ditetapkan.

#### a) File Setelan Gradle

File `settings.gradle`, terletak di direktori akar proyek, yang memberi tahu Gradle modul yang harus disertakan ketika membuat aplikasi Anda. Untuk kebanyakan proyek, file-nya sederhana dan hanya berisi:

```
include ':app'
```

Namun, proyek-proyek multi-modul harus menentukan setiap modul yang perlu dimasukkan ke pembangunan final.

#### b) File Pembangunan Tingkat-Atas

File `build.gradle` tingkat atas, terletak di direktori akar proyek, mendefinisikan konfigurasi pembangunan yang berlaku untuk semua modul dalam proyek Anda. Secara default, file pembangunan tingkat-atas menggunakan blok `buildscript {}` untuk mendefinisikan *repositori* dan *dependensi* Gradle yang familier untuk semua modul dalam proyek. Contoh kode berikut menjelaskan elemen DSL dan setelan default yang bisa Anda temukan di `build.gradle` tingkat-atas setelah membuat proyek baru.

#### c) File Pembangunan Tingkat-Modul

File `build.gradle` tingkat-modul, terletak di setiap direktori `<project>/<module>/`, yang memungkinkan Anda untuk mengonfigurasi setelan pembangunan untuk modul tertentu tempatnya berada. Mengonfigurasi setelan pembangunan ini memungkinkan Anda untuk menyediakan opsi pemaketan khusus, seperti tipe pembangunan dan ragam produk tambahan, serta menimpa setelan dalam manifes aplikasi `main/` atau file `build.gradle` tingkat-atas. Contoh file `build.gradle` modul aplikasi Android ini menggarisbawahi beberapa elemen DSL dan setelan dasar yang harus Anda ketahui.

#### d) File Properti Gradle

Gradle juga berisi dua file properti, terletak di direktori akar proyek, yang bisa Anda gunakan untuk menentukan setelan bagi toolkit build Gradle itu sendiri:



`gradle.properties`

Ini adalah tempat di mana Anda bisa mengonfigurasi setelan Gradle bermacam-proyek, seperti ukuran heap maksimum daemon Gradle. Untuk informasi selengkapnya.


`local.properties`

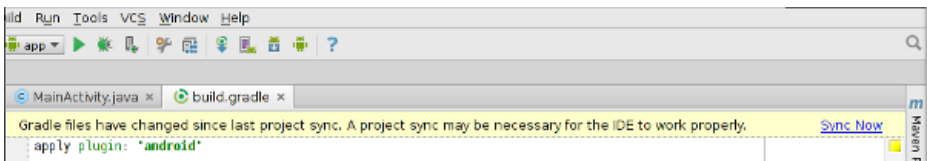
Mengonfigurasi properti lingkungan lokal untuk sistem pembangunan, seperti jalur ke pemasangan SDK. Karena materi dari file ini secara otomatis dihasilkan oleh Android Studio dan dikhususkan untuk lingkungan developer lokal, Anda sebaiknya tidak memodifikasi file ini secara manual atau memeriksanya dalam sistem kontrol versi.

#### e) Menyinkronkan Proyek dengan File Gradle

Bila Anda membuat perubahan pada file konfigurasi pembangunan dalam proyek, Android Studio akan mengharuskan Anda untuk melakukan sinkronisasi file proyek sehingga Android Studio bisa mengimpor perubahan konfigurasi pembangunan dan menjalankan beberapa pemeriksaan untuk memastikan konfigurasi Anda tidak akan menimbulkan kesalahan pembangunan.

Untuk melakukan sinkronisasi file proyek, klik **Sync Now** dalam bilah notifikasi yang muncul saat Anda membuat perubahan, seperti yang ditunjukkan

pada gambar 3, atau klik **Sync Project**  dari bilah menu. Jika Android Studio menemukan kesalahan dalam konfigurasi Anda, misalnya, kode sumber menggunakan fitur API yang hanya tersedia di API level yang lebih tinggi dari `compileSdkVersion`, jendela **Messages** akan muncul untuk menjelaskan masalah ini.



Gambar 19. Menyinkronkan project dengan file konfigurasi

#### f) Set Sumber

Android Studio secara logis mengelompokkan kode sumber dan resource untuk setiap modul ke dalam *set sumber*. Set sumber `main/` modul berisi kode dan resource yang digunakan oleh semua varian pembangunannya. Direktori set

sumber tambahan bersifat opsional, dan Android Studio tidak secara otomatis membuatnya ketika Anda mengonfigurasi varian pembangunan baru. Namun, membuat set sumber, serupa dengan `main/`, akan membantu mengelola file dan resource yang hanya boleh digunakan Gradle ketika membangun versi aplikasi tertentu:

- `src/main/`

Set sumber ini berisi kode dan resource yang biasa dipakai oleh semua varian pembangunan.

- `src/<buildType>/`

Buat set sumber ini untuk memuat kode dan resource khusus untuk tipe pembangunan tertentu.

- `src/<productFlavor>/`

Buat set sumber ini untuk memuat kode dan resource khusus untuk ragam produk tertentu.

- `src/<productFlavorBuildType>/`

Buat set sumber ini untuk memuat kode dan resource khusus untuk varian pembangunan tertentu.

Misalnya, untuk menghasilkan versi aplikasi "fullDebug", sistem pembangunan akan menggabungkan kode, setelan, dan resource dari set sumber berikut:

- `src/fullDebug/` (set sumber varian pembangunan)
- `src/debug/` (set sumber tipe pembangunan)
- `src/full/` (set sumber ragam produk)
- `src/main/` (set sumber utama)

## 2.4. Modul, Struktur File dan Pengujian Project Android

### a. Modul

Menyediakan penampung untuk kode sumber aplikasi Anda, file resource, dan setelan tingkat aplikasi seperti file build tingkat modul dan file Manifest Android. Saat Anda membuat proyek baru, nama modul default adalah "app". Di jendela Buat Modul Baru, Android Studio menawarkan modul aplikasi berikut:

- Modul Telepon & Tablet
- Pakai Modul OS
- Modul TV Android
- Modul Kaca

Setiap modul masing-masing menyediakan file-file penting dan beberapa templat kode yang sesuai untuk jenis aplikasi atau perangkat terkait. Modul menyediakan penampung untuk kode sumber aplikasi, resource file, dan pengaturan tingkat aplikasi, seperti file build tingkat modul dan file manifest Android. Setiap modul dapat dibuat secara independen, built, tested, and di debugger.

Android Studio menggunakan modul untuk mempermudah menambahkan perangkat baru ke proyek. Dengan mengikuti beberapa langkah mudah di Android Studio, dapat membuat modul untuk memuat kode yang khusus untuk jenis perangkat, seperti Wear OS atau Android TV. Android Studio secara otomatis membuat direktori modul, seperti direktori sumber dan resource, dan file build.gradle default yang sesuai untuk jenis perangkat. Selain itu, Android Studio membuat modul perangkat dengan konfigurasi pembuatan yang disarankan, seperti menggunakan pustaka Leanback untuk modul Android TV.

Halaman ini menjelaskan cara menambahkan modul baru untuk perangkat tertentu. Android Studio juga memudahkan untuk menambahkan pustaka atau modul Google Cloud ke proyek Anda. Untuk detail tentang membuat modul perpustakaan.

#### 1) Modul Perpustakaan

Menyediakan wadah untuk kode yang dapat digunakan kembali, yang dapat Anda gunakan sebagai ketergantungan dalam modul aplikasi lain atau diimpor ke proyek lain. Secara struktural, modul perpustakaan sama dengan modul aplikasi, tetapi ketika dibuat, ia menciptakan file arsip kode bukan APK, sehingga tidak dapat dipasang di perangkat. Di jendela Buat Modul Baru, Android Studio menawarkan modul perpustakaan berikut:

- Perpustakaan Android: Jenis pustaka ini dapat berisi semua jenis file yang didukung dalam proyek Android, termasuk kode sumber, resource, dan file manifest. Hasil build adalah file Android Archive (AAR) yang dapat Anda tambahkan sebagai ketergantungan untuk modul aplikasi Android Anda.
- Perpustakaan Java: Jenis pustaka ini hanya dapat berisi file sumber Java. Hasil build adalah file Java Archive (JAR) yang dapat Anda tambahkan sebagai dependensi untuk modul aplikasi Android Anda atau proyek Java lainnya.

## 2) Modul Google Cloud

Menyediakan penampung untuk kode backend Google Cloud Anda. Modul ini memiliki kode dan dependensi yang diperlukan untuk backend Java App Engine yang menggunakan HTTP sederhana, Cloud Endpoint, dan Cloud Messaging untuk terhubung ke aplikasi Anda. Anda dapat mengembangkan backend Anda untuk menyediakan layanan cloud yang dibutuhkan aplikasi Anda. Menggunakan Android Studio untuk mengembangkan modul Google Cloud Anda memungkinkan Anda mengelola kode aplikasi dan kode backend dalam proyek yang sama. Anda juga dapat menjalankan dan menguji kode backend Anda secara lokal, dan menggunakan Android Studio untuk menyebarkan modul Google Cloud Anda.

## 2.5. Project dan Struktur File Android

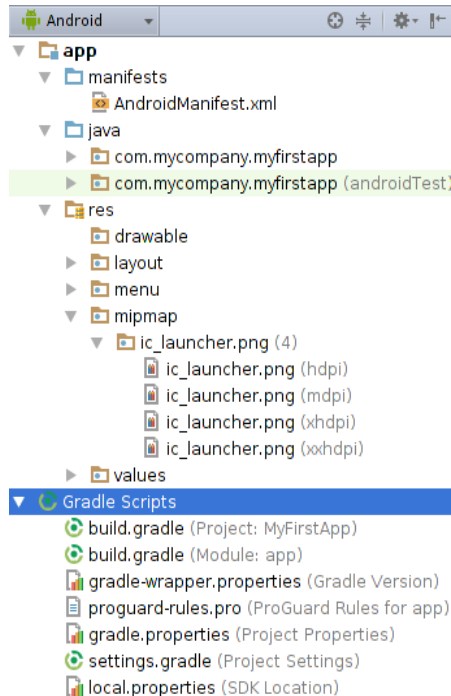
### a. Project file

Android menyediakan kerangka kerja aplikasi yang kaya dan memungkinkan Anda membangun aplikasi dan game inovatif untuk perangkat seluler di lingkungan bahasa pemrograman Java. Dokumen yang tercantum di navigasi sebelah kiri menyediakan detail tentang cara membangun aplikasi menggunakan beragam Android API.

Secara default, Android Studio menampilkan file proyek Anda di tampilan Android. Pandangan ini tidak mencerminkan hierarki file sebenarnya pada disk, tetapi diatur oleh modul dan jenis file untuk menyederhanakan navigasi antara file sumber utama proyek Anda, menyembunyikan file atau direktori tertentu yang tidak umum digunakan. Beberapa perubahan struktural dibandingkan dengan struktur pada disk termasuk yang berikut:

- Menampilkan semua file konfigurasi yang terkait dengan proyek dalam grup Gradle Script tingkat atas.

- Menampilkan semua file manifes untuk setiap modul dalam kelompok tingkat modul (bila Anda memiliki file manifes yang berbeda untuk berbagai jenis produk dan jenis pembuatan).
- Menampilkan semua file resource alternatif dalam satu grup, bukan di folder terpisah per kualifikasi resource. Misalnya, semua versi kerapatan ikon peluncur Anda terlihat berdampingan.



Gambar 20. Project File Android Studio

Dalam setiap modul aplikasi Android, file ditampilkan dalam grup berikut:

Tabel 2. Modul Aplikasi Android

Modul	Deskripsi / Keterangan
manifests	Berisi file AndroidManifest.xml.
java	Berisi file kode sumber Java, dipisahkan oleh nama paket, termasuk kode pengujian JUnit.
res	Berisi semua resource non-kode, seperti tata letak XML, string UI, dan gambar bitmap, dibagi menjadi sub-direktori yang sesuai.

## b. Struktur Project Android

Untuk melihat struktur file proyek yang sebenarnya termasuk semua file yang disembunyikan dari tampilan Android, pilih Project dari dropdown di bagian atas jendela Project. Saat Anda memilih tampilan Proyek, Anda dapat melihat lebih banyak file dan direktori. Yang paling penting adalah sebagai berikut:


Tabel 3. Struktur Project Android

Nama Module	Deskripsi / Keterangan
build/	Berisi build output
libs/	Berisi private libraries
src/	Berisi semua kode dan resource file untuk modul dalam subdirektori berikut
androidTest/	Berisi kode untuk pengujian instrumentasi yang berjalan di perangkat Android.
main/	Berisi file source "utama": kode dan resource Android yang dibagikan oleh semua varian build (file untuk varian build lainnya berada di direktori, seperti src / debug / untuk tipe build debug).
AndroidManifest.xml	Menjelaskan sifat dari aplikasi dan masing-masing komponennya. Untuk informasi lebih lanjut.
java/	Berisi sumber kode Java
jni/	Berisi kode asli menggunakan Java Native Interface (JNI).
gen/	Berisi file Java yang dihasilkan oleh Android Studio, seperti file R.java Anda dan antarmuka yang dibuat dari file AIDL.
res/	Berisi resource aplikasi, seperti file yang dapat digambar, file layout, dan string UI
assets/	Berisi file yang harus dikompilasi menjadi file .apk apa adanya. Anda dapat menavigasi direktori ini dengan cara yang sama seperti sistem file yang umum menggunakan URI dan membaca file sebagai aliran byte menggunakan AssetManager.
test/	Berisi kode untuk tes lokal yang berjalan di JVM host

build.gradle/module	Ini mendefinisikan konfigurasi build modul-spesifik.
build.gradle (project)	Ini mendefinisikan konfigurasi build Anda yang berlaku untuk semua modul. File ini merupakan bagian integral dari proyek, jadi harus mempertahankannya dalam kontrol revisi dengan semua kode sumber lainnya.

### c. Menggunakan Perangkat Android Secara Nyata Untuk Pengujian

Untuk menggunakan perangkat Android normal, hubungkan perangkat ke mesin pengembangan Anda (melalui USB). Di perangkat Android hidupkan USB Debugging. Pilih Pengaturan ► Opsi Pengembangan, kemudian aktifkan opsi USB-Debugging. Anda mungkin juga perlu menginstal driver untuk ponsel Anda. Linux dan Mac OS biasanya bekerja di peralatan luar ketika MS Windows biasanya membutuhkan instalasi driver.

	Versi Android minimum dari aplikasi Android Anda harus sesuai dengan versi Android di perangkat Anda.
---	---

Jika memiliki beberapa perangkat yang terhubung ke komputer, Anda dapat memilih perangkat mana yang harus digunakan. Jika hanya satu perangkat yang terhubung, aplikasi secara otomatis digunakan pada perangkat ini.

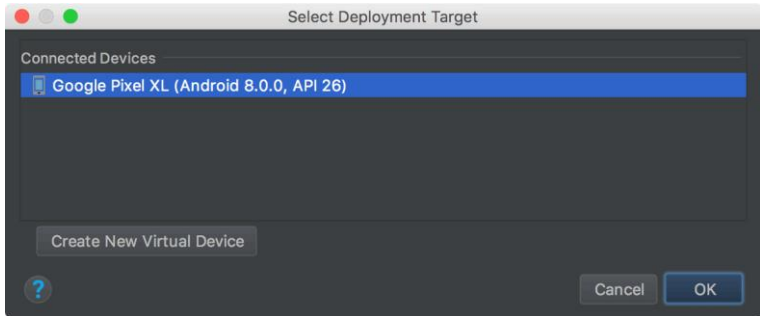
Persiapkan perangkat Anda sebagai berikut:

1. Hubungkan perangkat Anda ke mesin development dengan kabel USB. Jika Anda men-develop dengan menggunakan Windows, Anda mungkin harus menginstal driver USB yang tepat untuk perangkat Anda.
2. Aktifkan **USB debugging** di **Developer options** sebagai berikut.

Buka **Developer options**, kemudian scroll ke bawah untuk mencari dan mengaktifkan **USB debugging**.

Jalankan aplikasi di perangkat Anda sebagai berikut:

1. Di Android Studio, klik modul **app** di jendela **Project** dan pilih **Run > Run** (atau klik **Run** ► di toolbar).
2. Di jendela **Select Deployment Target**, pilih perangkat Anda, dan klik **OK**.



Gambar 21. Antarmuka pemilihan deployment target

Android Studio akan menginstal aplikasi itu di perangkat yang terhubung dan menjalankannya.

a. Memasang Driver USB OEM

Jika Anda tengah melakukan pengembangan pada Windows dan ingin menghubungkan perangkat untuk melakukan pengujian, maka Anda harus memasang driver USB yang sesuai. Laman ini memberikan tautan ke situs web untuk beberapa Original Equipment Manufacturer (OEM), tempat Anda bisa mengunduh driver USB yang sesuai untuk perangkat Anda.

Jika Anda melakukan pengembangan pada Mac OS X atau Linux, maka Anda tidak memerlukan driver USB. Sebagai gantinya. Untuk menghubungkan dan melakukan debug perangkat Google Nexus menggunakan Windows, Anda harus memasang driver USB Google. Satu-satunya pengecualian adalah Galaxy Nexus: driver untuk Galaxy Nexus didistribusikan oleh Samsung (terdaftar sebagai model GT-I9250TSGGEN).

b. *Memasang Driver USB*

Pertama, cari driver yang sesuai untuk perangkat Anda dari tabel driver OEM di bawah ini.

Setelah mengunduh driver USB, ikuti petunjuk di bawah ini untuk memasang atau meningkatkan versi driver, berdasarkan versi Windows dan apakah Anda memasang driver untuk pertama kalinya atau meningkatkan versi yang sudah ada.

- Windows 7 dan versi yang lebih tinggi  
Untuk memasang driver USB Android pada Windows 7 dan versi yang lebih tinggi untuk pertama kalinya:
  1. Hubungkan perangkat Android Anda ke port USB komputer.



2. Klik kanan pada **Computer** dari desktop atau Windows Explorer, dan pilih **Manage**.
3. Pilih **Devices** di panel kiri.
4. Cari dan luaskan **Other device** di panel kanan.
5. Klik kanan pada nama perangkat (seperti **Nexus S**) dan pilih **Update Driver Software**. Ini akan menjalankan Wizard Pembaruan Perangkat Keras.
6. Pilih **Browse my computer for driver software** dan klik **Next**.
7. Klik **Browse** dan arahkan ke folder driver USB. (Driver USB Google terletak di <sdk>\extras\google\usb\_driver\.)
8. Klik **Next** untuk memasang driver. Atau, untuk meningkatkan versi driver USB Android yang ada pada Windows 7 dan versi yang lebih tinggi dengan driver baru.

c. Mendapatkan Driver OEM


Tabel 4. Original Equipment Manufacturer (OEM)

OEM	URL Driver
Acer	<a href="http://www.acer.com/worldwide/support/mobile.html">http://www.acer.com/worldwide/support/mobile.html</a>
ALCATEL	<a href="http://www.alcatelonetouch.com/global-en/support/">http://www.alcatelonetouch.com/global-en/support/</a>
Asus	<a href="http://support.asus.com/download/">http://support.asus.com/download/</a>
Blackberry	<a href="https://swdownloads.blackberry.com/Downloads/entry.do?code=4EE0932F46276313B51570F46266A608">https://swdownloads.blackberry.com/Downloads/entry.do?code=4EE0932F46276313B51570F46266A608</a>
Dell	<a href="http://support.dell.com/support/downloads/index.aspx?c=us&amp;cs=19&amp;l=en&amp;s=dhs&amp;~ck=anavml">http://support.dell.com/support/downloads/index.aspx?c=us&amp;cs=19&amp;l=en&amp;s=dhs&amp;~ck=anavml</a>
Fujitsu	<a href="http://www.fmworld.net/product/phone/sp/android/develop/">http://www.fmworld.net/product/phone/sp/android/develop/</a>
Hisense	<a href="http://app.hismarttv.com/dss/resourcecontent.do?method=viewResourceDetail&amp;resourceId=16&amp;type=5">http://app.hismarttv.com/dss/resourcecontent.do?method=viewResourceDetail&amp;resourceId=16&amp;type=5</a>
HTC	<a href="http://www.htc.com">http://www.htc.com</a>
Huawei	<a href="http://consumer.huawei.com/en/support/index.htm">http://consumer.huawei.com/en/support/index.htm</a>
Intel	<a href="http://www.intel.com/software/android">http://www.intel.com/software/android</a>
Kyocera	<a href="http://www.kyocera-wireless.com/support/phone_drivers.htm">http://www.kyocera-wireless.com/support/phone_drivers.htm</a>
Lenovo	<a href="http://support.lenovo.com/us/en/GlobalProductSelector">http://support.lenovo.com/us/en/GlobalProductSelector</a>
LGE	<a href="http://www.lg.com/us/support/software-firmware">http://www.lg.com/us/support/software-firmware</a>
Motorola	<a href="https://motorola-global-portal.custhelp.com/app/answers/detail/a_id/88481/">https://motorola-global-portal.custhelp.com/app/answers/detail/a_id/88481/</a>
MTK	<a href="http://online.mediatek.com/Public%20Documents/MTK_Android_USB_Driver.zip">http://online.mediatek.com/Public%20Documents/MTK_Android_USB_Driver.zip</a> (Unduhan ZIP)
Oppo	<a href="http://www.oppo.com/index.php?q=software/view&amp;sw_id=631">http://www.oppo.com/index.php?q=software/view&amp;sw_id=631</a>
Pegatron	<a href="http://www.pegatroncorp.com/download/New_Duke_PC_Driver_0705.zip">http://www.pegatroncorp.com/download/New_Duke_PC_Driver_0705.zip</a> (Unduhan ZIP)

Samsung	<a href="http://www.samsung.com/us/support/downloads">http://www.samsung.com/us/support/downloads</a>
Sharp	<a href="http://k-tai.sharp.co.jp/support/">http://k-tai.sharp.co.jp/support/</a>
Sony	<a href="http://developer.sonymobile.com/downloads/drivers/">http://developer.sonymobile.com/downloads/drivers/</a>
Toshiba	<a href="http://support.toshiba.com/sscontent?docId=4001814">http://support.toshiba.com/sscontent?docId=4001814</a>
Xiaomi	<a href="http://www.xiaomi.com/c/driver/index.html">http://www.xiaomi.com/c/driver/index.html</a>
ZTE	<a href="http://support.zte.com.cn/support/news/NewsDetail.aspx?newsId=1000442">http://support.zte.com.cn/support/news/NewsDetail.aspx?newsId=1000442</a>

d. Menggunakan Emulator Android dan Android Virtual Device (AVD)

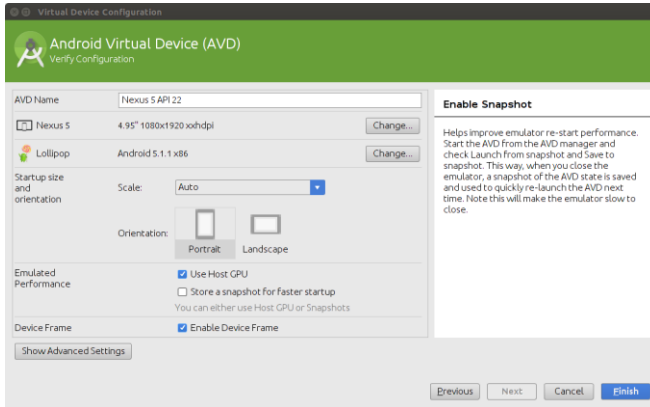
Android Tools berisi Android Device Emulator. Emulator ini dapat digunakan untuk menjalankan Android Virtual Device (AVD). AVD ini memungkinkan untuk menguji aplikasi di perangkat Android yang dipilih tanpa akses ke perangkat keras yang sebenarnya. Perangkat virtual memberi kemungkinan untuk menguji aplikasi untuk versi Android yang dipilih dan dengan konfigurasi tertentu. Bahkan jika memiliki perangkat fisik Android yang tersedia, harus terbiasa dengan pembuatan dan penggunaan AVD. Selama pembuatan AVD, Kita menentukan konfigurasi untuk perangkat virtual, termasuk, misalnya, resolusi, versi API Android, dan densitas display. Dapat menentukan beberapa AVD dengan konfigurasi berbeda dan memulainya secara paralel.

	<p>Jika Anda berhenti dan AVD selama proses startup, AVD mungkin rusak. Awal pertama mungkin membutuhkan waktu lama, misalnya, beberapa menit pada mesin yang lebih lama. Pada mesin modern biasanya membutuhkan kurang dari satu menit untuk memulai suatu AVD baru.</p>
---	---

Setelah AVD dimulai, dapat mengontrol GUI dengan mouse. Emulator juga menyediakan akses ke tombol telepon melalui menu di sisi kanan emulator. Setelah dimulai, jangan hentikan AVD selama pembuatan. Jika mengubah aplikasi dan ingin menguji versi baru, cukup menerapkan kembali aplikasi pada AVD. AVD yang dibuat untuk Android berisi program dari Proyek Open Source Android. AVD yang dibuat untuk Google API mengandung kode khusus Google tambahan. AVD yang dibuat untuk Google API memungkinkan untuk menguji aplikasi yang menggunakan layanan Google Play, misalnya, Google Maps API baru atau layanan lokasi baru.

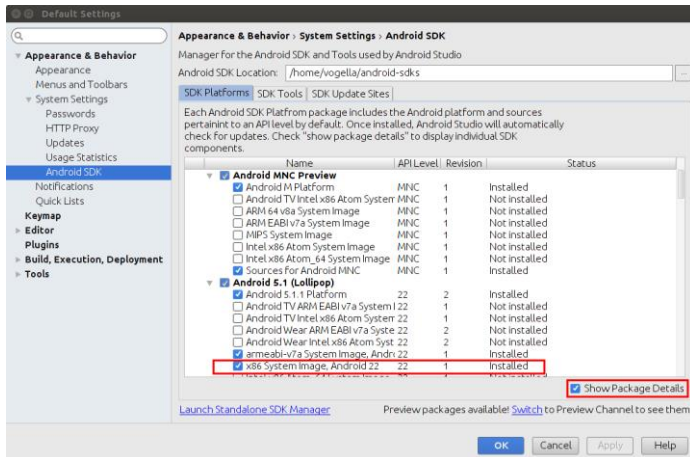
Selama pembuatan emulator dapat memilih apakah ingin Snapshot atau Gunakan Host GPU diaktifkan. Jika memilih opsi Snapshot, kedua kalinya Anda memulai perangkat, itu dimulai dengan sangat cepat karena AVD menyimpan statusnya jika Anda menutupnya. Jika pilih Gunakan Host GPU, AVD

menggunakan kartu grafis dari komputer parent secara langsung yang membuat rendering pada perangkat yang diemulasikan jauh lebih cepat.



Gambar 22. Tampilan konfigurasi Android Virtual Device

Menggunakan ARM CPU atau arsitektur CPU Intel. Jika memungkinkan, pilih AVD dengan gambar berdasarkan arsitektur CPU ARM atau berdasarkan arsitektur CPU Intel. Perangkat virtual Android yang menggunakan citra sistem Intel jauh lebih cepat dalam eksekusi. Gambar berbasis Intel terkadang tidak berfungsi pada mesin tertentu, dalam hal ini coba gambar sistem berbasis ARM. Gambar sistem berbasis ARM cenderung bekerja pada lebih banyak komputer.




Gambar 23. SDK Platforms

## 2.6. Instalasi Android Studio

Android Software Development Kit (Android SDK) (Android SDK) dan perangkat Gradle berisi alat yang diperlukan untuk Compile, package, deploy, dan memulai aplikasi Android. Google juga menyediakan IDE bernama Android Studio untuk melakukan tugas pengembangan. SDK berisi Android debug bridge (adb). adb adalah alat yang memungkinkan kita terhubung ke perangkat Android virtual atau nyata. Ini memungkinkan pengelolaan perangkat atau debugging pada aplikasi.

### Persyaratan sistem

Pengembangan untuk Android dapat dilakukan pada komputer yang berukuran cukup. Untuk pengalaman yang menyenangkan komputer modern direkomendasikan, misalnya, CPU 2,6 GHz dengan setidaknya 8 GB memori. Sebuah SSD mempercepat dimulainya emulator Android secara signifikan.

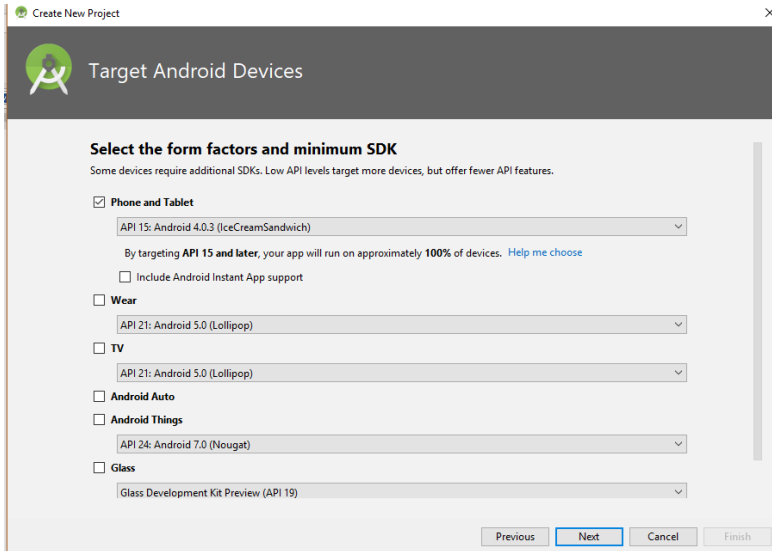
	<p>SDK Android adalah 32-bit, oleh karena itu pada sistem Linux 64-bit Anda harus menginstal paket-paket ia32-libs. Untuk Ubuntu Anda dapat melakukan ini di console melalui perintah berikut:</p> <pre style="text-align: center;"><b>apt-get install ia32-libs</b></pre> <p>Silakan periksa dokumentasi distribusi Anda, jika Anda menggunakan rasa Linux yang berbeda</p>
---	--

### Memulai proyek Android Studio

Setelah berhasil memasang Android Studio IDE, klik dua kali ikon aplikasi Android Studio untuk memulainya.

- ⇒ Pilih **Start a new Android Studio project** di jendela Welcome, dan beri nama proyek dengan nama yang sama yang ingin Anda gunakan untuk aplikasi.
- ⇒ Saat memilih Domain Perusahaan unik, ingat bahwa aplikasi yang dipublikasikan ke Google Play harus memiliki nama paket unik. Karena domain bersifat unik, yang mengawali nama aplikasi dengan nama Anda, atau nama domain perusahaan, sebaiknya sediakan nama paket unik yang memadai. Jika tidak berencana untuk mempublikasikan aplikasi, Anda bisa menerima domain contoh default. Ketahuilah bahwa mengubah nama paket di lain waktu memerlukan kerja tambahan.
- ⇒ Memilih perangkat target dan SDK minimum

Saat memilih Target Android Devices, Phone and Tablet dipilih secara default, seperti yang ditampilkan dalam gambar di bawah ini. Pilihan yang ditampilkan dalam gambar untuk Minimum SDK — **API 15: Android 4.0.3 (IceCreamSandwich)** — membuat aplikasi Anda kompatibel dengan 100% perangkat Android yang aktif di Google Play Store.



Gambar 24. Screenshot Target Android Studio device

Perangkat berbeda menjalankan versi sistem Android yang berbeda, seperti Android 4.0.3 atau Android 4.4. Setiap versi yang berurutan umumnya menambahkan API baru yang tidak tersedia di versi sebelumnya. Untuk menunjukkan rangkaian API yang tersedia, setiap versi menetapkan API level. Misalnya, Android 1.0 adalah API level 1 dan Android 4.0.3 adalah API level 15.

Minimum SDK mendeklarasikan versi Android minimum untuk aplikasi Anda. Setiap versi Android yang berurutan menyediakan kompatibilitas untuk aplikasi yang dibangun menggunakan API dari versi sebelumnya, sehingga aplikasi Anda akan *selalu* kompatibel dengan versi Android mendatang sambil menggunakan Android API yang didokumentasikan.

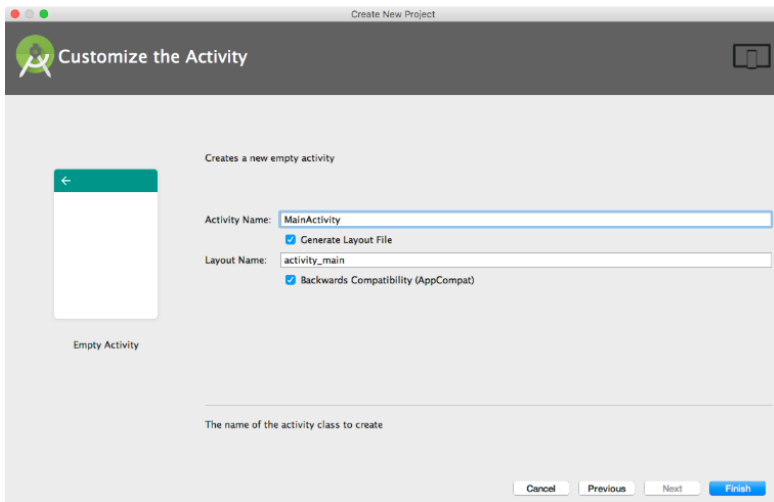
⇒ Memilih template

Android Studio mengumpulkan terlebih dulu proyek Anda dengan kode minimum untuk activity dan layout layar berdasarkan *template*. Tersedia

berbagai macam template, mulai dari template kosong virtual (Add No Activity) hingga beragam tipe activity.

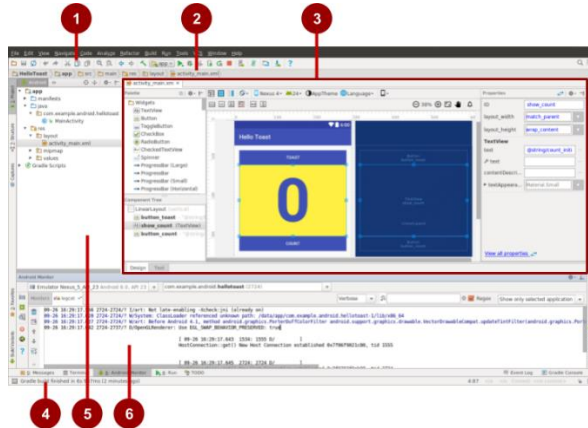
Anda bisa menyesuaikan activity setelah memilih template. Misalnya, template Empty Activity menyediakan satu activity yang disertai satu resource layout untuk layar. Anda bisa memilih untuk menerima nama yang biasa digunakan untuk activity (seperti **MainActivity**) atau mengubah nama di layar Customize Activity. Selain itu, jika Anda menggunakan template Empty Activity, pastikan memeriksa yang berikut ini jika belum diperiksa:

- **Generate Layout File:** Biarkan ini dicentang untuk membuat resource layout yang terhubung dengan activity ini, yang biasanya diberi nama **activity\_main.xml**. Layout mendefinisikan antarmuka pengguna untuk activity.
- **Backwards Compatibility (AppCompat):** Biarkan ini dicentang untuk menyertakan pustaka AppCompat sehingga aplikasi kompatibel dengan Android versi sebelumnya bahkan jika menggunakan fitur yang hanya ditemukan di versi yang lebih baru.



Gambar 25. Screenshot layar Customize Activity

Android Studio membuat folder untuk proyek yang baru saja dibuat di folder AndroidStudioProjects pada komputer Anda.



Gambar 26. Screenshot jendela utama Android Studio

- Panel jendela Android Studio

Jendela utama Android Studio terdiri dari sejumlah area logis, atau *panel*, seperti yang ditampilkan dalam gambar 26.

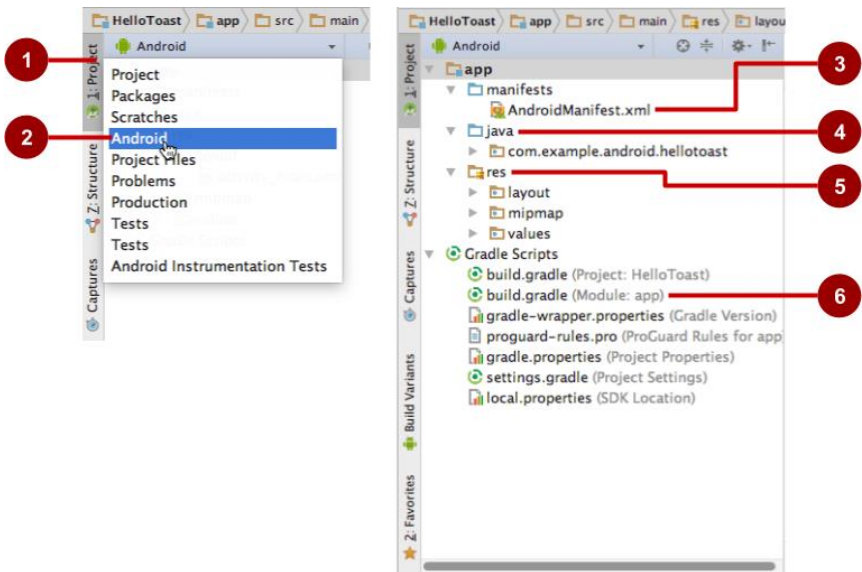
Dalam gambar 26:

- 1. Toolbar.** Toolbar menjalankan beragam tindakan, termasuk menjalankan aplikasi Android dan meluncurkan alat Android.
- 2. Bilah Navigasi.** Bilah navigasi memungkinkan navigasi melalui proyek dan membuka file untuk pengeditan. Bilah navigasi menyediakan tampilan struktur proyek yang lebih ringkas.
- 3. Panel Editor.** Panel ini menampilkan materi file yang dipilih dalam proyek. Misalnya, setelah memilih layout (seperti yang ditampilkan dalam gambar), panel ini menampilkan editor layout dengan alat untuk mengedit layout. Setelah memilih file kode Java, panel ini menampilkan kode dengan alat untuk mengedit kode.
- 4. Bilah Status.** Bilah status menampilkan status proyek dan Android Studio itu sendiri, serta peringatan atau pesan apa pun. Anda bisa mengamati kemajuan pembangunan di bilah status.
- 5. Panel Project.** Panel proyek menampilkan file proyek dan hierarki proyek.
- 6. Panel Monitor.** Panel monitor menawarkan akses ke daftar TODO untuk mengelola tugas, Android Monitor untuk memantau eksekusi aplikasi (ditampilkan dalam gambar), logcat untuk menampilkan pesan log, dan aplikasi Terminal untuk melakukan aktivitas Terminal.

## 2.7. Menjelajahi proyek

Setiap proyek di Android Studio berisi file `AndroidManifest.xml`, file kode sumber komponen, dan file resource terkait. Secara default, Android Studio mengatur file proyek Anda berdasarkan jenis file, dan menampilkannya dalam proyek: Tampilan Android di panel alat (bantu) kiri, seperti yang ditampilkan di bawah ini. Tampilan menyediakan akses cepat ke file kunci proyek Anda.

Untuk beralih kembali ke tampilan ini dari tampilan lainnya, klik tab **Project** vertikal di kolom paling kiri panel Project, dan pilih **Android** dari menu munculan di bagian atas panel Project, seperti yang ditampilkan dalam gambar di bawah ini.



Gambar 27. Menjelajahi project yang telah di-create

Dalam gambar di atas:

1. Tab **Project**. Klik untuk menampilkan tampilan proyek.
2. Pilihan **Android** di menu tarik-turun proyek.
3. File **AndroidManifest.xml**. Digunakan untuk menetapkan informasi tentang aplikasi bagi lingkungan waktu proses Android. Template yang Anda pilih membuat file ini.
4. Folder **java**. Folder ini menyertakan aktivitas, pengujian, dan komponen lain di kode sumber Java. Setiap aktivitas, layanan, dan komponen lain didefinisikan sebagai kelas Java, biasanya dalam file miliknya. Nama



aktiviti pertama (layar) yang pengguna lihat, yang juga melakukan inisialisasi resource seluruh aplikasi, biasanya adalah MainActivity.

5. Folder **res**. Folder ini menyimpan resource, seperti layout XML, string UI, dan gambar. Aktiviti biasanya dikaitkan dengan file resource XML yang menetapkan layout tampilannya. File ini biasanya diberi nama setelah aktiviti atau fungsinya.
6. File **\*\*build.gradle (Module: App)**
  - \*. *File ini menetapkan konfigurasi pembangunan modul. Template yang Anda pilih membuat file ini, yang mendefinisikan konfigurasi pembangunan, termasuk atribut `minSdkVersion` yang mendeklarasikan versi minimum untuk aplikasi, dan atribut `targetSdkVersion` yang mendeklarasikan versi tertinggi (terbaru) untuk aplikasi yang dioptimalkan. File ini juga menyertakan daftar dependensi\*, yaitu pustaka yang diperlukan oleh kode — seperti pustaka `AppCompat` untuk mendukung beragam versi Android.*

### Menampilkan Manifes Android

Sebelum sistem Android bisa memulai komponen aplikasi, sistem harus mengetahui bahwa komponen ada dengan membaca file `AndroidManifest.xml` aplikasi. Aplikasi harus mendeklarasikan semua komponennya dalam file ini, yang harus berada pada akar direktori proyek aplikasi.

Untuk menampilkan file ini, luaskan folder manifes di Project: Tampilan Android, dan klik dua kali file (**AndroidManifest.xml**). Isinya muncul di panel pengeditan seperti yang ditampilkan dalam scriptcode di bawah ini.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.myapplication"
    android:versionCode="1"
    android:versionName="1.0" >
    ...
</manifest>
```

#### ▪ Namespace dan tag aplikasi Android

Manifes Android dikodekan di XML dan selalu menggunakan namespace Android:

```
xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.android.helloworld">
```

Ekspresi `package` menampilkan nama paket unik aplikasi baru. Jangan mengubah ini setelah aplikasi dipublikasikan.

```
<application
  ...
</application>
```

Tag `<application`, dengan tag `</application>` penutupnya, mendefinisikan setelan manifes untuk keseluruhan aplikasi.

- Cadangan otomatis

Atribut `android:allowBackup` memungkinkan pencadangan data aplikasi otomatis:

```
...
  android:allowBackup="true"
...
```

Menyetel atribut `android:allowBackup` ke `true` memungkinkan aplikasi dicadangkan secara otomatis dan dipulihkan jika perlu. Pengguna menggunakan waktu dan upaya untuk mengonfigurasi aplikasi. Beralih ke perangkat baru bisa membatalkan seluruh konfigurasi yang dibuat dengan susah payah. Sistem melakukan cadangan otomatis ini untuk hampir seluruh data aplikasi secara default, dan melakukannya tanpa mengharuskan developer menulis kode aplikasi tambahan lainnya.

Untuk aplikasi dengan versi SDK target berupa Android 6.0 (API level 23) dan yang lebih tinggi, perangkat yang menjalankan Android 6.0 dan yang lebih tinggi secara otomatis membuat cadangan data aplikasi ke awan karena atribut `android:allowBackup` default ke `true` jika dihilangkan. Untuk aplikasi < API level 22, Anda harus secara eksplisit menambahkan atribut `android:allowBackup` dan menyetelnya ke `true`.

- Ikon aplikasi

Metode `android:icon` menyetel ikon untuk aplikasi:


```
...
  android:allowBackup="true"
  android:icon="@mipmap/ic_launcher"
...
```

Atribut `android:icon` menetapkan ikon di folder **mipmap** (di dalam folder **res** di Project: tampilan Android) untuk aplikasi. Ikon muncul di Peluncur untuk meluncurkan aplikasi. Ikon juga digunakan sebagai ikon default untuk komponen aplikasi.

- Resource string dan label aplikasi

Seperti yang bisa Anda lihat di gambar sebelumnya, atribut `android:label` menampilkan string "Hello World" yang disorot. Jika Anda mengklik string ini, maka string berubah untuk menampilkan resource string `@string/app_name`:

```
...  
    android:label="@string/app_name"  
...
```

	<p><b>Tip:</b> Ctrl - klik atau klik-kanan <code>app_name</code> di panel edit untuk melihat menu konteks. Pilih <b>Go To &gt; Declaration</b> untuk melihat lokasi resource string dideklarasikan: dalam file <code>strings.xml</code>. Bila Anda memilih <b>Go To &gt; Declaration</b> atau membuka file dengan mengklik dua kali <code>strings.xml</code> di Project: Tampilan Android (di dalam folder <b>values</b>), isinya muncul di panel pengeditan</p>
---	--

Setelah membuka file `strings.xml`, Anda bisa melihat bahwa nama string `app_name` disetel ke `Hello World`. Anda bisa mengubah nama aplikasi dengan mengubah string `Hello World` ke hal lain. Resource string dijelaskan dalam pelajaran terpisah.

- Tema aplikasi

Atribut `android:theme` menyetel tema aplikasi, yang mendefinisikan penampilan elemen antarmuka pengguna seperti teks:

```
...  
    android:theme="@style/AppTheme">  
...
```

Atribut `theme` disetel ke tema standar `AppTheme`. Tema dijelaskan dalam pelajaran terpisah.

- Mendeklarasikan versi Android

Perangkat yang berbeda mungkin menjalankan versi sistem Android yang berbeda, seperti Android 4.0 atau Android 4.4. Setiap versi yang berurutan bisa

menambahkan API baru yang tidak tersedia di versi sebelumnya. Untuk menunjukkan rangkaian API yang tersedia, setiap versi menetapkan API level. Misalnya, Android 1.0 adalah API level 1 dan Android 4.4 adalah API level 19.

API level memungkinkan developer untuk mendeklarasikan versi minimum dengan aplikasi yang kompatibel, menggunakan tag manifes `<uses-sdk>` dan atribut `minSdkVersion`. Misalnya, Calendar Provider API telah ditambahkan di Android 4.0 (API level 14). Jika aplikasi Anda tidak berfungsi tanpa API tersebut, deklarasikan API level 14 sebagai versi yang didukung minimum aplikasi seperti ini:

```
<manifest ... >
  <uses-sdk android:minSdkVersion="14"
    android:targetSdkVersion="19" />
  ...
</manifest>
```

Atribut `minSdkVersion` mendeklarasikan versi minimum untuk aplikasi, dan atribut `targetSdkVersion` mendeklarasikan versi tertinggi (terbaru) yang telah dioptimalkan dalam aplikasi. Setiap versi Android yang berurutan menyediakan kompatibilitas untuk aplikasi yang dibangun menggunakan API dari versi sebelumnya, sehingga aplikasi akan *selalu* kompatibel dengan versi Android mendatang sambil menggunakan Android API yang didokumentasikan.

Atribut `targetSdkVersion` *tidak* mencegah aplikasi dipasang pada versi Android yang lebih tinggi (yang lebih baru) dibandingkan nilai yang ditetapkan, namun hal ini penting karena menunjukkan pada sistem apakah aplikasi harus mewarisi perubahan perilaku di versi yang baru. Jika Anda tidak memperbarui `targetSdkVersion` ke versi terbaru, sistem akan menganggap bahwa aplikasi memerlukan perilaku kompatibilitas mundur saat dijalankan pada versi terbaru. Misalnya, di antara perubahan perilaku di Android 4.4, alarm yang dibuat dengan AlarmManager API saat ini tidak tepat secara default sehingga sistem bisa mengumpulkan alarm aplikasi dan mempertahankan kekuatan sistem, namun sistem akan menahan perilaku API sebelumnya untuk aplikasi jika API level target Anda lebih rendah daripada "19".

- Menampilkan dan mengedit kode Java

Komponen ditulis di Java dan dicantumkan dalam folder modul di folder **java** dalam Project: Tampilan Android. Setiap nama modul diawali dengan nama domain (seperti **com.example.android**) dan menyertakan nama aplikasi.

Contoh berikut menampilkan komponen *activity*:

1. Klik folder modul untuk meluaskan dan menampilkan file MainActivity untuk activity yang ditulis di Java (kelas MainActivity).
2. Klik dua kali **MainActivity** untuk melihat file sumber di panel pengeditan, seperti yang ditampilkan dalam scriptcode di bawah ini.

```
package com.example.hello.helloworld;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d("MainActivity", "Hello World");
    }
}
```

Bagian paling atas file MainActivity.java adalah pernyataan package yang mendefinisikan paket aplikasi. Ini diikuti dengan blok import yang diringkas dalam gambar di atas, dengan "...". Klik titik untuk meluaskan blok guna menampilkannya.

Pernyataan import akan mengimpor pustaka yang diperlukan untuk aplikasi, seperti berikut ini, yang mengimpor pustaka AppCompatActivity:

```
import android.support.v7.app.AppCompatActivity
```

Setiap activity dalam aplikasi diimplementasikan sebagai kelas Java. Deklarasi kelas berikut memperluas kelas AppCompatActivity untuk mengimplementasikan fitur dengan cara yang kompatibel-mundur dengan Android versi sebelumnya:

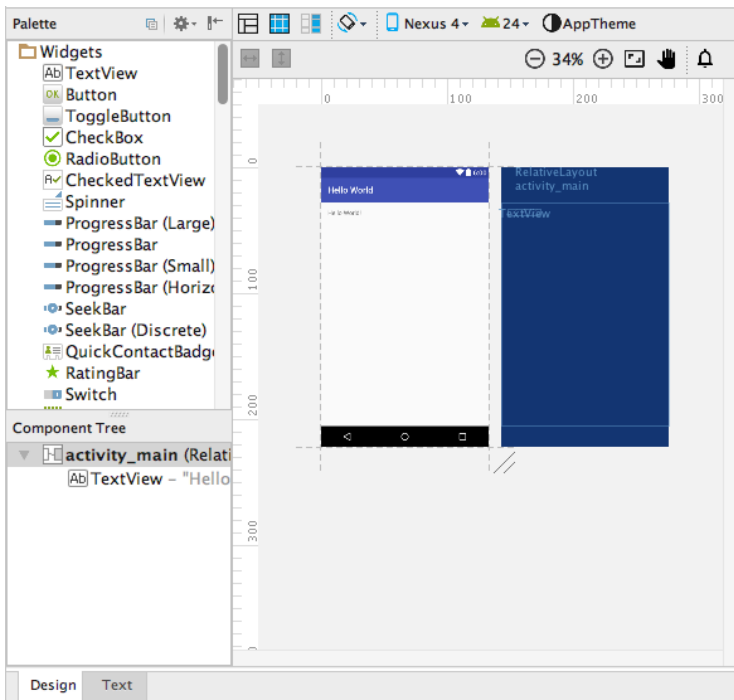
```
MainActivity kelas publik memperluas AppCompatActivity {
    ...
}
```

Seperti yang Anda pelajari terdahulu, sebelum sistem Android bisa memulai komponen aplikasi seperti activity, sistem harus mengetahui bahwa activity ada dengan membaca file `AndroidManifest.xml` aplikasi. Oleh karena itu, setiap activity harus dicantumkan di file `AndroidManifest.xml`.

- Menampilkan dan mengedit layout

Resource layout ditulis dalam XML dan dicantumkan dalam folder **layout** di folder **res** dalam Project: Tampilan Android. Klik **res > layout**, kemudian klik dua kali **activity\_main.xml** untuk melihat file layout di panel pengeditan.

Android Studio menampilkan tampilan Desain layout, seperti yang ditampilkan dalam gambar di bawah ini. Tampilan ini menyediakan panel Palette elemen antarmuka pengguna, dan petak yang menampilkan layout layar.

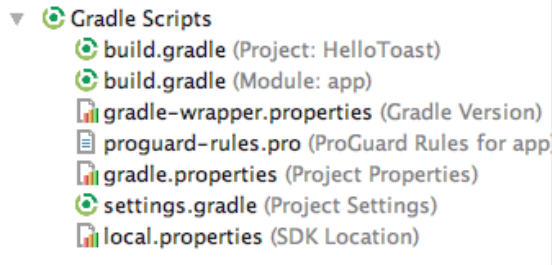


Gambar 28. Tampilan desain layout

- Memahami file build.gradle

Bila Anda membuat proyek, Android Studio secara otomatis menghasilkan file pembangunan penting di folder **Gradle Scripts** dalam Project: Tampilan

Android. File pembangunan Android Studio diberi nama **build.gradle** seperti yang ditampilkan di bawah ini:



Gambar 29. Folder gradle

Setiap proyek memiliki hal berikut:

- **build.gradle (Project: *apptitle*)**

Ini adalah file pembangunan tingkat atas untuk keseluruhan proyek, berada di akar direktori proyek, yang mendefinisikan konfigurasi pembangunan yang berlaku untuk semua modul di proyek Anda. File ini, yang dihasilkan oleh Android Studio, tidak boleh diedit untuk menyertakan dependensi aplikasi.

- **build.gradle (Module: *app*)**

Android Studio membuat file `build.gradle (Module: app)` terpisah untuk setiap modul. Anda bisa mengedit setelan pembangunan guna menyediakan opsi pemaketan khusus untuk setiap modul, seperti tipe pembangunan tambahan dan ragam produk, dan untuk menggantikan setelan di file manifest atau `build.gradle` tingkat atas. File ini paling sering adalah file untuk mengedit ketika mengubah konfigurasi tingkat aplikasi, seperti mendeklarasikan dependensi di bagian `dependencies`. Yang berikut ini menampilkan isi file `build.gradle (Module: app)` proyek:

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 24
    buildToolsVersion "24.0.1"
    defaultConfig {
        applicationId "com.example.android.helloworld2"
        minSdkVersion 15
        targetSdkVersion 24
        versionCode 1
        versionName "1.0"
    }
}
```

```

        testInstrumentationRunner
"android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        rilis {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-
android.txt'), 'proguard-rules.pro' } }
    }
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])

androidTestCompile('com.android.support.test.espresso:espresso
-core:2.2.2', {
    exclude group: 'com.android.support', module: 'support-
annotations'
})
    compile 'com.android.support:appcompat-v7:24.2.1'
    testCompile 'junit:junit:4.12'
}

```

File `build.gradle` menggunakan sintaks Gradle. Gradle adalah Domain Specific Language (DSL) untuk menjelaskan dan memanipulasi logika pembangunan dengan menggunakan Groovy, yang merupakan bahasa dinamis untuk Java Virtual Machine (JVM). Anda tidak perlu mempelajari Groovy untuk melakukan perubahan, karena Android Plugin for Gradle memperkenalkan sebagian besar elemen DSL yang Anda perlukan.

- Blok Plugin dan Android.

Dalam file `build.gradle` (`Module: app`) di atas, pernyataan pertama menerapkan tugas pembangunan plug-in Gradle khusus Android:

```

apply plugin: 'com.android.application'
android {
    ...
}

```

Blok `android { }` menetapkan hal berikut ini untuk versi:

- Versi SDK target untuk mengompilasi kode:

```
compileSdkVersion 24
```
- Versi alat pembangunan untuk digunakan membangun aplikasi:



```
buildToolsVersion "24.0.1"
```

- Blok defaultConfig

Setelan inti dan entri untuk aplikasi ditetapkan di blok defaultConfig { } dalam android { } block:

```
...
defaultConfig {
    applicationId "com.example.hello.helloworld"
    minSdkVersion 15
    targetSdkVersion 23
    versionCode 1
    versionName "1.0"
    testInstrumentationRunner
"android.support.test.runner.AndroidJUnitRunner" }
...
```

Setelan minSdkVersion dan targetSdkVersion menggantikan setelan AndroidManifest.xml untuk versi SDK minimum dan versi SDK target.

Pernyataan testInstrumentationRunner ini menambahkan dukungan instrumentasi untuk menguji antarmuka pengguna dengan Espresso dan UIAutomator. Hal ini dijelaskan dalam pelajaran terpisah.

- Tipe pembangunan

Tipe pembangunan untuk aplikasi ditetapkan di blok buildTypes{ }, yang mengontrol cara aplikasi dibangun dan dipaketkan.

```
...
buildTypes {
    rilis {
        minifyEnabled false proguardFiles
getDefaultProguardFile('proguard-android.txt'),
                        'proguard-rules.pro' }
}
...
```

Tipe pembangunan yang ditetapkan adalah release untuk rilis aplikasi. Tipe pembangunan umum lainnya adalah debug. Mengonfigurasi tipe pembangunan dijelaskan dalam pelajaran terpisah.

- Dependensi

Dependensi untuk aplikasi didefinisikan di blok dependencies { }, yang merupakan bagian file build.gradle yang paling mungkin berubah saat Anda mulai mengembangkan kode yang bergantung pada pustaka. Blok adalah bagian dari Gradle API standar dan termasuk *di luar* blok android { }.

```
...
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso
-core:2.2.2', {
    exclude group: 'com.android.support', module: 'support-
annotations'
    })
    compile 'com.android.support:appcompat-v7:24.2.0'
    testCompile 'junit:junit:4.12'}
}
```

Dalam cuplikan di atas, pernyataan `compile fileTree(dir: 'libs', include: ['*.jar'])` menambahkan dependensi semua file ".jar" di dalam direktori `libs`. Konfigurasi `compile` akan mengompilasi aplikasi utama — segala sesuatu di dalamnya ditambahkan ke `classpath` kompilasi, dan juga dipaketkan ke dalam APK final.

- Menyinkronkan proyek Anda

Bila Anda membuat perubahan pada file konfigurasi pembangunan dalam proyek, Android Studio akan mengharuskan Anda untuk melakukan *sinkronisasi* file proyek sehingga Android Studio bisa mengimpor perubahan konfigurasi pembangunan dan menjalankan beberapa pemeriksaan untuk memastikan konfigurasi tidak akan menimbulkan kesalahan pembangunan.


Untuk menyinkronkan file proyek, klik **Sync Now** di bilah notifikasi yang muncul saat membuat perubahan, atau klik **Sync Project** dari bilah menu. Jika Android Studio memperlihatkan kesalahan apa pun dengan konfigurasi — misalnya, jika kode sumber menggunakan fitur API yang hanya tersedia di API level yang lebih tinggi dari `compileSdkVersion` — jendela `Messages` muncul untuk menjelaskan masalah.

## 2.8. Menjalankan aplikasi pada emulator atau perangkat

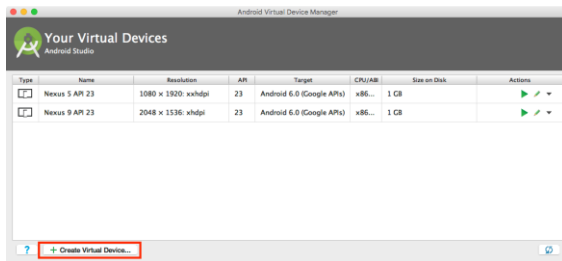
Dengan emulator perangkat virtual, Anda bisa menguji aplikasi pada perangkat yang berbeda seperti tablet atau ponsel cerdas — dengan API level yang berbeda untuk versi Android yang berbeda — untuk memastikan aplikasi terlihat bagus dan berfungsi untuk sebagian besar pengguna. Meskipun ini adalah ide yang bagus, Anda tidak harus bergantung pada memiliki perangkat fisik yang tersedia untuk development aplikasi.

Pengelola Android Virtual Device (AVD) membuat perangkat virtual atau emulator yang menyimulasikan konfigurasi untuk tipe perangkat Android tertentu. Gunakan AVD Manager untuk mendefinisikan karakteristik perangkat keras perangkat dan API levelnya, dan untuk menyimpannya sebagai konfigurasi perangkat virtual. Ketika Anda memulai emulator Android, emulator akan membaca konfigurasi yang ditetapkan dan membuat perangkat emulasi pada komputer yang berperilaku sama persis dengan versi fisik perangkat.

- Membuat perangkat virtual

Untuk menjalankan emulator pada komputer, gunakan AVD Manager untuk membuat konfigurasi yang menjelaskan perangkat virtual. Pilih **Tools > Android > AVD Manager**, atau klik ikon AVD Manager  di bilah alat.

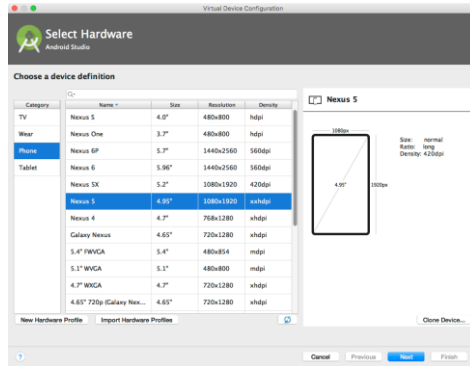
Layar "Your Virtual Devices" muncul menampilkan semua perangkat virtual yang dibuat sebelumnya. Klik tombol **+Create Virtual Device** untuk membuat perangkat virtual baru.



Gambar 30. Create virtual device

Anda bisa memilih perangkat dari daftar perangkat keras yang telah didefinisikan sebelumnya. Untuk setiap perangkat, tabel menampilkan ukuran tampilan diagonal (Size), resolusi layar dalam piksel (Resolution), dan kepadatan piksel (Density). Misalnya, kepadatan piksel perangkat Nexus 5 adalah xxhdpi, artinya aplikasi menggunakan ikon di folder xxhdpi dari folder

**mipmap**. Selain itu, aplikasi akan menggunakan layout dan resource dapat digambar dari folder yang didefinisikan untuk kepadatan tersebut.




Gambar 31. Pemilihan device definition

Anda juga memilih versi sistem Android untuk perangkat. Tab **Recommended** menampilkan sistem yang disarankan untuk perangkat. Lebih banyak versi tersedia di dalam tab **x86 Images** dan **Other Images**.

- Menjalankan aplikasi pada perangkat virtual

Untuk menjalankan aplikasi pada perangkat virtual yang Anda buat di bagian sebelumnya, ikuti langkah-langkah ini:

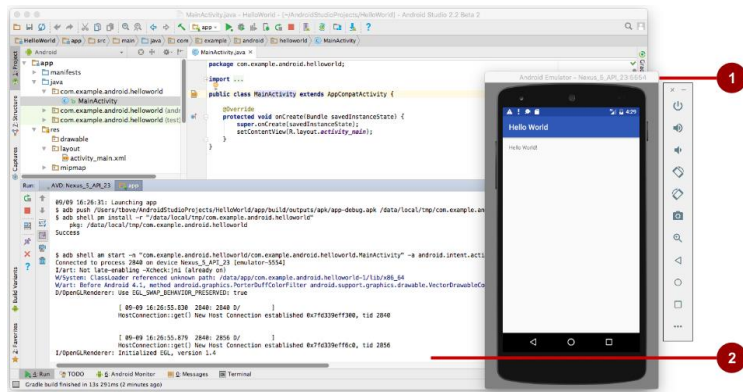
- Di Android Studio, pilih **Run > Run app** atau klik **Run icon**  di bilah alat.
- Di jendela Select Deployment Target, pada Emulator yang tersedia, pilih perangkat virtual yang Anda buat, dan klik **OK**.

Emulator memulai dan mem-booting seperti perangkat fisik. Ini memerlukan waktu beberapa saat, bergantung pada kecepatan komputer Anda. Aplikasi membangun, dan setelah emulator siap, Android Studio mengunggah aplikasi ke emulator dan menjalankannya.

Anda harus melihat aplikasi yang dibuat dari template Empty Activity ("Hello World") seperti yang ditampilkan dalam gambar berikut ini, yang juga menampilkan panel Run Android Studio yang menampilkan tindakan yang dilakukan untuk menjalankan aplikasi di emulator.



**Catatan:** Saat menguji pada emulator, praktik yang baik adalah memulainya sekali pada awal sesi, dan jangan menutupnya hingga selesai sehingga tidak perlu melalui proses booting kembali.



Gambar 32. Running emulator

Dalam gambar di atas:

1. **Emulator** menjalankan aplikasi.
2. **Run Pane.** Ini menampilkan tindakan yang dilakukan untuk memasang dan menjalankan aplikasi.

### Menjalankan aplikasi pada perangkat fisik

Selalu uji aplikasi Anda pada perangkat fisik, karena pengguna akan menggunakan aplikasi pada perangkat fisik. Meskipun emulator cukup bagus, emulator tidak bisa menampilkan semua kemungkinan keadaan perangkat, seperti yang terjadi jika ada panggilan masuk sewaktu aplikasi berjalan. Untuk menjalankan aplikasi pada perangkat fisik, Anda memerlukan hal berikut ini:

- Perangkat Android seperti ponsel cerdas atau tablet.
- Kabel data untuk menghubungkan perangkat Android ke komputer melalui porta USB.
- Jika Anda menggunakan Linux atau Windows, mungkin perlu melakukan langkah tambahan untuk menjalankan aplikasi pada perangkat keras.

Untuk memungkinkan Android Studio berkomunikasi dengan perangkat, aktifkan USB Debugging pada perangkat Android. Pada versi Android 4.2 dan yang lebih baru, layar opsi Developer disembunyikan secara default.

## 2.9. Pemecahan masalah koneksi perangkat

Jika Android Studio tidak mengenali perangkat, coba hal berikut:

1. Putuskan koneksi perangkat dari komputer, kemudian hubungkan kembali.
2. Mulai ulang Android Studio.
3. Jika komputer masih tidak menemukan perangkat atau mendeklarasikannya "tidak sah":
  - a. Putuskan koneksi perangkat dari komputer.
  - b. Di perangkat, pilih **Settings > Developer Options**.
  - c. Ketuk **Revoke USB Debugging authorizations**.
  - d. Hubungkan kembali perangkat ke komputer.
  - e. Bila dikonfirmasi, berikan otorisasi.
4. Anda mungkin perlu memasang driver USB yang sesuai untuk perangkat.
5. Periksa dokumentasi terakhir, forum pemrograman, atau dapatkan bantuan dari instruktur Anda.

### Menggunakan log

Log merupakan alat (bantu) debug andal yang bisa Anda gunakan untuk melihat nilai, jalur eksekusi, dan pengecualian. Setelah menambahkan pernyataan log ke aplikasi, pesan log Anda akan muncul bersama pesan log umum di tab logcat panel Android Monitor di Android Studio.

Untuk melihat panel Android Monitor, klik tombol **Android Monitor** di bagian bawah jendela utama Android Studio. Android Monitor menawarkan dua tab:

- Tab **logcat**. Tab **logcat** menampilkan pesan log tentang aplikasi saat aplikasi berjalan. Jika menambahkan pernyataan logging ke aplikasi, pesan log Anda dari pernyataan tersebut muncul dengan pesan log lain di bawah tab ini.
- Tab **Monitors**. Tab **Monitors** memantau kinerja aplikasi, yang bisa digunakan untuk men-debug dan menyesuaikan kode Anda.

### Menambahkan pernyataan log ke aplikasi Anda

Pernyataan log menambahkan pesan apa pun yang Anda tetapkan ke log. Menambahkan pernyataan log di titik tertentu dalam kode memungkinkan developer melihat nilai, jalur eksekusi, dan pengecualian.

Misalnya, pernyataan log berikut menambahkan "MainActivity" dan "Hello World" ke log:

```
Log.d("MainActivity", "Hello World");
```

Berikut adalah elemen pernyataan ini:

- `Log`: Kelas `Log` adalah API untuk mengirim pesan log.
- `d`: Anda menetapkan *log level* sehingga Anda bisa memfilter pesan log menggunakan menu tarik-turun di bagian tengah panel tab **logcat**. Berikut adalah tingkat log yang bisa Anda tetapkan:
  - `d`: Pilih **Debug** atau **Verbose** untuk melihat pesan ini.
  - `e`: Pilih **Error** atau **Verbose** untuk melihat pesan ini.
  - `w`: Pilih **Warning** atau **Verbose** untuk melihat pesan ini.
  - `i`: Pilih **Info** atau **Verbose** untuk melihat pesan ini.
- `"MainActivity"`: Argumen pertama adalah *log tag* yang bisa digunakan untuk memfilter pesan pada tab **logcat**. Ini biasanya merupakan nama aktivitas tempat pesan berasal. Akan tetapi, Anda bisa membuatnya menjadi apa saja yang berguna bagi Anda untuk men-debug aplikasi. Praktik terbaik adalah untuk menggunakan konstanta sebagai tag log, seperti berikut:
  - 1) Definisikan tag log sebagai konstanta sebelum menggunakannya dalam pernyataan log:

```
private static final String LOG_TAG =  
MainActivity.class.getSimpleName();
```

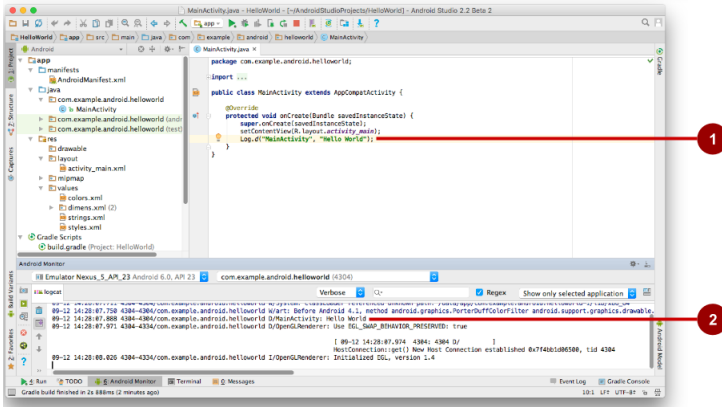
- 2) Gunakan konstanta dalam pernyataan log:

```
Log.d(LOG_TAG, "Hello World")
```

`"Hello World"`: Argumen kedua adalah pesan sebenarnya yang muncul setelah log level dan tag log pada tab **logcat**.

### Menampilkan pesan log Anda

Panel Run muncul di tempat panel Android Monitor bila Anda menjalankan aplikasi di emulator atau perangkat. Setelah mulai menjalankan aplikasi, klik tombol **Android Monitor** di bagian bawah jendela utama, kemudian klik tab **logcat** di panel Android Monitor jika belum dipilih.



Gambar 33. Pesan log

Dalam gambar di atas:

1. Pernyataan log di metode onCreate() dari MainActivity.
2. Panel Android Monitor yang menampilkan pesan log logcat, termasuk pesan dari pernyataan log.

Secara default, tampilan log disetel ke **Verbose** di menu tarik-turun di bagian atas tampilan **logcat** untuk menampilkan semua pesan. Anda bisa mengubahnya menjadi **Debug** untuk melihat pesan yang dimulai dengan Log.d, atau mengubahnya menjadi **Error** untuk melihat pesan yang dimulai dengan Log.e, dan seterusnya





## YANG HARUS SUDAH ANDA KETAHUI

Untuk praktik ini Anda harus bisa:

- 1) Memahami proses development perangkat lunak umum untuk aplikasi berorientasi objek menggunakan IDE (Integrated Development Environment).
- 2) Memiliki pengalaman telah mengikuti pembelajaran pemrograman berorientasi objek dan pemrograman visual dengan java.

## YANG AKAN ANDA BUTUHKAN

Untuk materi dan praktik ini, Anda akan membutuhkan:

- 1) Komputer Mac, Windows, atau Linux. Tool Android Studio untuk persyaratan sistem terbaru.
- 2) Akses internet atau cara alternatif untuk memuat Android Studio dan pemasangan Java terbaru ke komputer Anda.
- 3) Smartphone sebagai perangkat untuk pengujian secara nyata aplikasi yang telah dibuat.

## YANG AKAN ANDA PELAJARI

Anda akan belajar:

- 1) Memasang dan menggunakan Android IDE.
- 2) Memahami proses development untuk membangun aplikasi Android.
- 3) Membuat proyek Android dari template aplikasi dasar.

## YANG HARUS ANDA LAKUKAN

- 1) Memasang lingkungan development Android Studio.
- 2) Membuat emulator (perangkat virtual) untuk menjalankan aplikasi Anda di komputer.
- 3) Membuat dan menjalankan aplikasi Hello World di perangkat virtual dan fisik.
- 4) Menjelajahi layout proyek.
- 5) Membuat dan menampilkan pernyataan log dari aplikasi Anda.
- 6) Menjelajahi file AndroidManifest.xml.

## Latihan 2.1. Memasang Android Studio

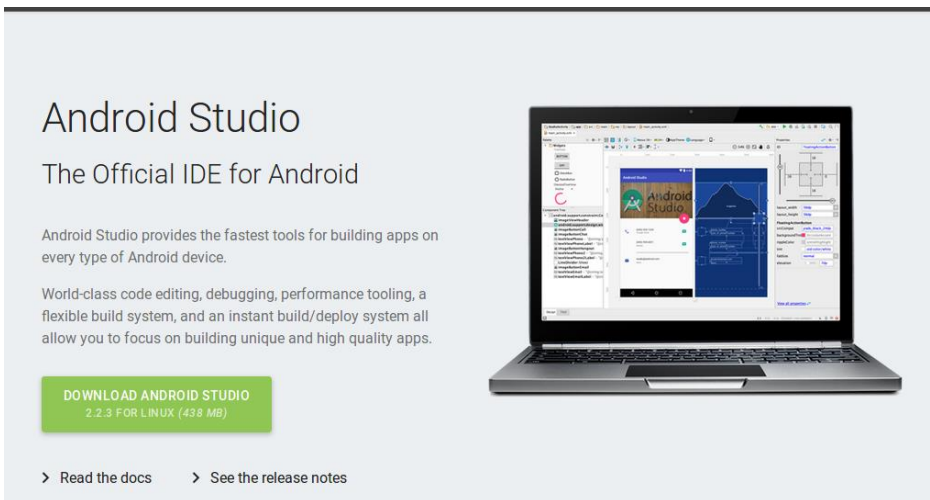
### Unduh Android Studio:

Android Studio adalah IDE Google untuk aplikasi Android. Android Studio menyediakan editor kode tingkat lanjut dan serangkaian template aplikasi. Selain itu, Android Studio juga memiliki alat untuk development, men-debug, menguji, dan meningkatkan kinerja yang membuat pengembangan aplikasi lebih cepat dan mudah. Anda bisa menguji aplikasi dengan berbagai macam emulator yang telah dikonfigurasi sebelumnya atau di perangkat seluler Anda sendiri, dan membangun APK produksi untuk publikasi.

Unduh Android Studio dari Situs Android Studio:

<http://developer.android.com/sdk/index.html>

Unduhan ini hadir dalam dua pilihan, hanya SDK Tools dan Paket Android Studio. Ingin mengunduh Paket Android Studio untuk sistem operasi Anda.

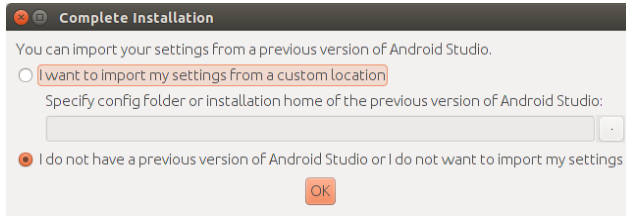


Gambar 34. Download Android Studio

Instalasi untuk Windows sederhana, cukup di launch .exe yang diunduh. Pada Max OSX, drag dan drop Android Studio ke dalam folder Aplikasi.

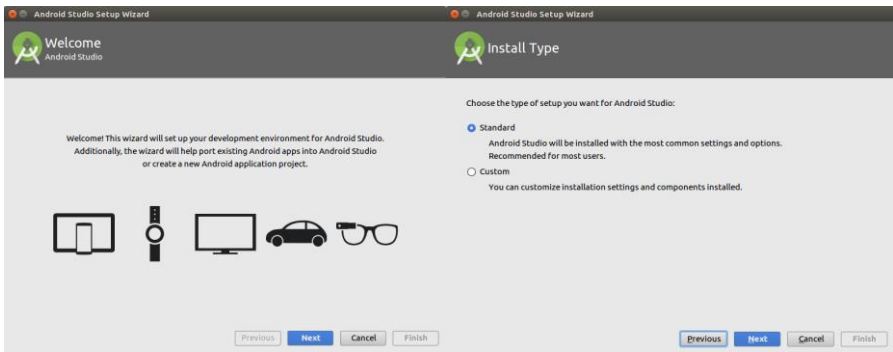
Di Linux, ekstrak file ZIP yang diunduh ke lokasi yang tepat untuk aplikasi. Untuk launch Android Studio, buka direktori android-studio/bin/ di terminal dan jalankan studio.sh.

- Saat pertama kali memulai pemasangan Android Studio baru, Anda memiliki opsi untuk mengimpor pengaturan yang ada.



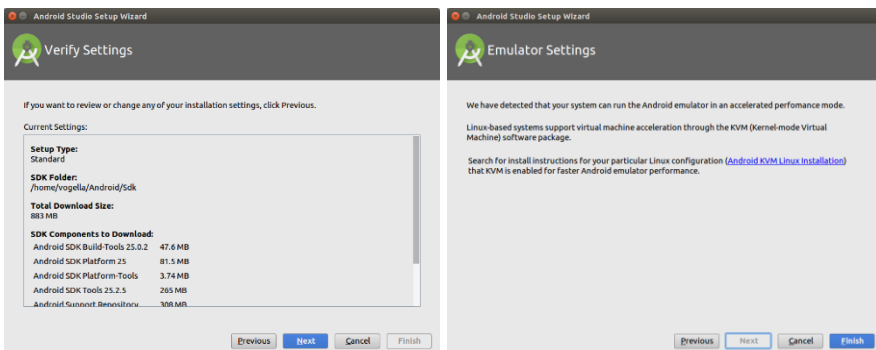
Gambar 35. Pilihan Pemasangan Instalasi

- Setelah itu klik panduan konfigurasi.



Gambar 36. Proses Instalasi

- Klik next untuk tampilan welcome, instal type (pilih standard), verify setting.

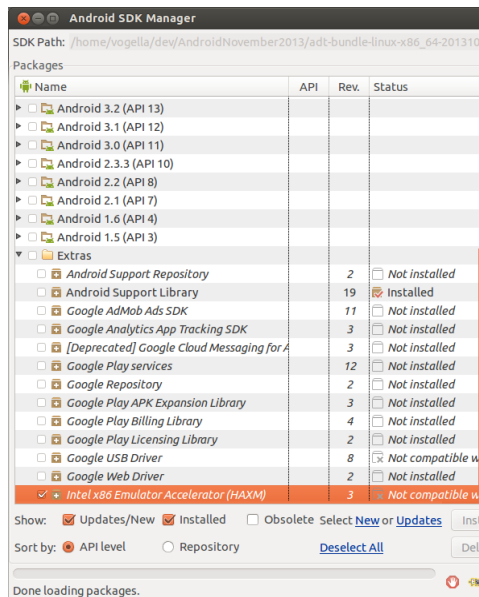


Gambar 37. Proses Instalasi Lanjutan

- Setelah Anda mencapai halaman terakhir, tekan tombol Selesai.

## Pemasangan Emulator Intel di Windows

Pada saat penulisan ini juga perlu mengunduh dan menginstal driver tambahan untuk MS windows.



Gambar 38. Antarmuka Android SDK Manager

Setelah mengunduh, Anda menemukan driver di direktori instalasi Android Anda di folder ekstra / intel. Anda perlu menginstal driver dengan menjalankan file .exe.

Setelah instalasi, Anda dapat membuat AVD baru berdasarkan emulator Intel. Memulai Emulator lebih cepat dan selama aplikasi Android di eksekusi.

## **Pemasangan emulator Intel di Linux**

Linux membutuhkan pengaturan yang lebih kompleks. Untuk instalasi terperinci, lihat Intel emulator installation guide (<https://software.intel.com/en-us/android/articles/>) yang juga termasuk petunjuk terperinci untuk Windows.

## **Pemasangan SDK Android**

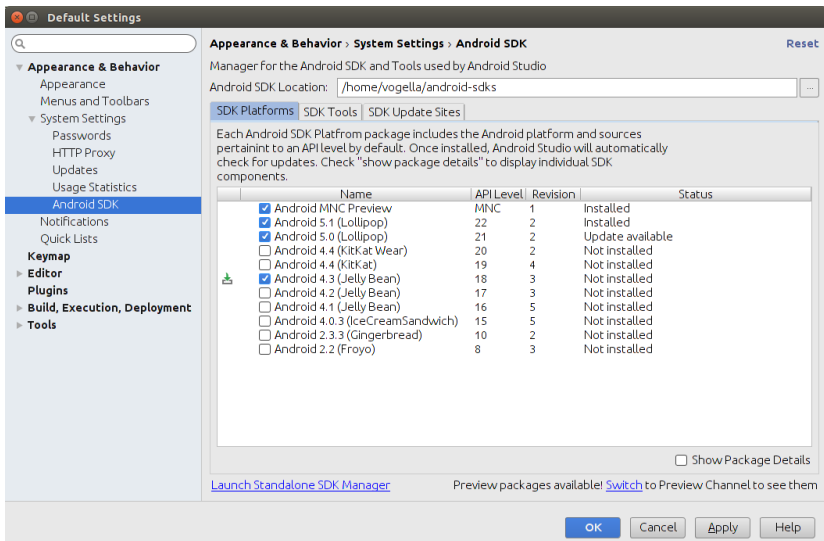
Android SDK Manajer memungkinkan Anda menginstal dan menghapus versi OS Android tertentu.

Select tools ► Android ► SDK Manajer atau icon SDK Manajer di Toolbar Android Studio untuk membuka Android SDK Manager.



Gambar 39. Tampilan Toolbar Icon SDK Manager

Pada Android SDK manager, pilih versi Android yang diinginkan pada dropdown tree and tekan tombol Install. screenshot gambar berikut menunjukkan pilihan untuk Android versi API 18.



Gambar 40. Tampilan Setting SDK Platform

Tekan tombol OK untuk memulai instalasi.

Tab SDK Platforms digunakan untuk menginstal versi API, dan SDK yang digunakan untuk menginstal development tools.

Untuk menyiapkan dan mengoperasikan Android Studio:

- ⇒ Anda mungkin perlu memasang Java Development Kit - Java 7 atau versi yang lebih baik.
- ⇒ Memasang Android Studio

Android Studio dapat digunakan di komputer Windows, Mac, dan Linux. Pemasangannya sama untuk semua platform. Perbedaannya akan dijelaskan di bawah ini.

### Memasang Java Development Kit:

- ⇒ Buka jendela terminal di komputer.
- ⇒ Ketikkan `java -version`

Output termasuk baris:

```
Java(TM) SE Runtime Environment (build1.X.0_05-b13)
```

**X** adalah nomor versi yang perlu dilihat.

- Jika nomornya 7 atau lebih besar, Anda bisa lanjut memasang Android Studio.
- Jika versi Java SE di bawah 7 atau belum dipasang, Anda perlu memasang versi terbaru paket development Java SE sebelum memasang Android Studio.

Untuk mengunduh Java Standard Edition () Development Kit (JDK):

- ⇒ Buka laman unduhan Oracle Java SE.
- ⇒ Klik ikon Java SE Downloads untuk membuka [laman Java SE Development Kit 8 Downloads](#).
- ⇒ Di kotak paket Java SE Development terbaru, Anda perlu menerima Perjanjian Lisensi agar dapat melanjutkan. Lalu unduh versi yang sesuai untuk komputer yang Anda gunakan untuk mengembangkan.
- ⇒ Memasang paket development. Setelah pemasangan JDK selesai, yang seharusnya hanya memakan waktu beberapa menit, Anda bisa mengonfirmasi bahwa pemasangan sudah benar dengan memeriksa versi Java dari baris perintah.
- ⇒ Buka jendela terminal dan ketikkan `java -version` lagi untuk memverifikasi bahwa pemasangan telah berhasil.
- ⇒ Setel variabel lingkungan `JAVA_HOME` ke direktori pemasangan JDK.

### ▪ **Windows:**

1. Setel `JAVA_HOME` ke lokasi pemasangan.

2. Start > Control Panel > System > Advanced System Settings > Environment Variables System Variables > New
  - o Nama variabel: JAVA\_HOME
  - o Nilai variabel: C:\Program Files\Java\jdk1.7.0\_80 (atau apa pun versi pemasangan Anda)
3. Jika variabel sudah ada, perbarui dengan versi JDK ini.
4. Verifikasi variabel JAVA\_HOME dari terminal cmd.exe: `echo %JAVA_HOME%`

Lihat juga:

[https://docs.oracle.com/cd/E19182-01/820-7851/inst\\_cli\\_jdk\\_javahome\\_t/](https://docs.oracle.com/cd/E19182-01/820-7851/inst_cli_jdk_javahome_t/)

▪ **Mac:**

1. Buka Terminal.
2. Konfirmasi bahwa Anda memiliki JDK dengan mengetik "which java".
3. Periksa bahwa Anda memiliki versi Java yang diperlukan dengan mengetik "java -version".
4. Setel JAVA\_HOME menggunakan perintah ini di Terminal: `ekspor JAVA_HOME=`which java``
5. masukkan `echo $JAVA_HOME` untuk mengonfirmasi jalurnya.

▪ **Linux:**

Lihat:

[https://docs.oracle.com/cd/E19182-01/820-7851/inst\\_cli\\_jdk\\_javahome\\_t/](https://docs.oracle.com/cd/E19182-01/820-7851/inst_cli_jdk_javahome_t/)

Pemecahan Masalah Dasar:

- a. Tidak ada ikon UI, Control Panel, atau Startup yang terkait dengan JDK.
- b. Verifikasi bahwa Anda telah memasang JDK dengan benar dengan membuka direktori tempat Anda memasangnya. Untuk mengidentifikasi tempat JDK, lihat variabel PATH dan/atau cari direktori "jdk" atau "java" atau "javac" yang dapat dijalankan di komputer Anda.

Memasang Android Studio

- ⇒ Buka situs developer Android dan ikuti petunjuk untuk mengunduh dan memasang Android Studio.

- Terima konfigurasi default untuk semua langkah.
- Pastikan semua komponen dipilih untuk pemasangan.
- ⇒ Setelah selesai memasang. Wizard Persiapan akan mengunduh dan memasang beberapa komponen tambahan. Bersabarlah karena ini mungkin memakan waktu bergantung kecepatan internet Anda, dan beberapa langkah mungkin tampak berulang.
- ⇒ Setelah unduhan selesai, Android Studio akan dimulai, dan Anda siap membuat proyek pertama.

### Pemecahan masalah:

Jika menemui masalah dengan pemasangan, periksa dokumentasi terbaru, forum pemrograman, atau minta bantuan instruktur Anda.



## Latihan 2.2. Membuat Aplikasi "Hello World"

Dalam tugas ini, Anda akan mengimplementasikan aplikasi "Hello World" untuk memverifikasi bahwa Android Studio sudah dipasang dengan benar dan mempelajari dasar-dasar pengembangan dengan Android Studio.

### Membuat aplikasi "Hello World"

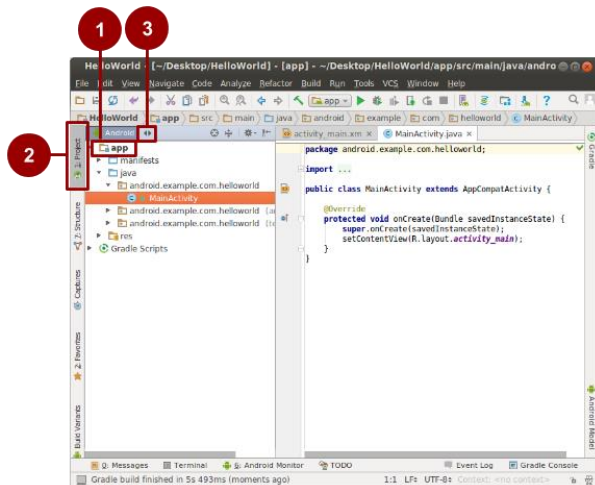
1. Buka Android Studio jika belum terbuka.
2. Di jendela utama **Welcome to Android Studio**, klik "Start a new Android Studio project".
3. Di jendela **New Project**, berikan aplikasi Anda **Application Name**, seperti "Hello World".
4. Verifikasi lokasi Proyek, atau pilih direktori yang berbeda untuk menyimpan proyek.
5. Pilih **Company Domain** yang unik.
  - o Aplikasi yang dipublikasikan di Google Play Store harus memiliki nama paket yang unik. Karena domain unik, mengawali nama aplikasi dengan nama atau nama domain perusahaan Anda akan menghasilkan nama paket yang unik.
6. Verifikasi bahwa **Project Location** default adalah tempat Anda menyimpan aplikasi Hello World dan proyek Android Studio lainnya, atau ubah lokasi ke direktori yang diinginkan. Klik Next.
7. Di layar **Target Android Devices**, "Phone and Tablet" harus dipilih. Dan Anda harus memastikan bahwa API 15: Android 4.0.3 IceCreamSandwich disetel sebagai **Minimum SDK**. (Perbaiki jika perlu.)
  - o Saat buku ini ditulis, memilih tingkat API ini menjadikan aplikasi "Hello World" kompatibel dengan 97% perangkat Android yang aktif di Google Play store.
  - o Ini adalah setelan yang digunakan oleh contoh di buku ini.
8. Klik **Next**.
9. Jika proyek Anda memerlukan komponen tambahan untuk SDK target terpilih, Android Studio akan memasangnya secara otomatis. Klik **Next**.
10. Jendela **Customize the Activity**. Setiap aplikasi membutuhkan paling tidak satu activity. Activity mewakili satu layar dengan satu antarmuka pengguna dan Android Studio menyediakan template untuk membantu Anda memulai. Untuk proyek Hello World, pilih template termudah (saat ini ditulis, template proyek "Empty Activity" adalah template paling sederhana) yang tersedia.
11. Adalah praktik yang biasa untuk menyebut activity utama Anda MainActivity. Ini bukan persyaratan.

12. Pastikan kotak **Generate Layout file** dicentang (jika terlihat).
13. Pastikan kotak **Backwards Compatibility (App Compat)** dicentang.
14. Biarkan **Layout Name** sebagai `activity_main`. Layout biasanya diberi nama sesuai dengan aktivitasnya. Terima defaultnya dan klik **Finish**.

Setelah langkah-langkah ini, Android Studio:

- ⇒ Membuat folder untuk Proyek Android Studio Anda.
- ⇒ Bangun proyek Anda dengan Gradle (ini mungkin memerlukan beberapa waktu). Android Studio menggunakan Gradle sebagai sistem build-nya.
- ⇒ Membuka editor kode dengan proyek Anda.
- ⇒ Menampilkan tip hari ini.
  - Android Studio menawarkan banyak pintasan keyboard, dan membaca tip adalah cara yang bagus untuk mempelajarinya dari waktu ke waktu.

Jendela Android Studio harus terlihat mirip dengan diagram berikut:



Gambar 41. Jendela Android Studio

Anda bisa melihat hierarki file untuk aplikasi Anda dengan beberapa cara.

1. Klik di folder Hello World folder untuk meluaskan hierarki file (1),
2. Klik di **Project** (2).
3. Klik di menu **Android** (3).
4. Jelajahi opsi tampilan yang berbeda untuk proyek Anda.

## Latihan 2.3. Jelajahi Struktur Proyek

Dalam praktik ini, Anda akan menjelajahi cara file proyek disusun di Android Studio.

Langkah-langkah ini mengasumsikan bahwa proyek Hello World Anda dimulai seperti yang ditampilkan pada diagram di atas.

### Menjelajahi struktur proyek dan layout

Di tampilan Project > Android tugas sebelumnya, ada tiga folder level teratas di bawah folder **app** Anda: **manifests**, **java**, dan **res**.

- Luaskan folder manifests.

Folder ini berisi **AndroidManifest.xml**. File ini menjelaskan semua komponen aplikasi Android Anda dan dibaca oleh sistem waktu proses Android saat program dijalankan.

- Luaskan folder java. **Semua file bahasa Java dikelola di folder ini. Folder java berisi tiga subfolder:**

- **com.example.hello.helloworld (atau nama domain yang telah Anda tetapkan):** Semua file untuk paket ada di folder yang bernama sama dengan paket tersebut. Untuk aplikasi Hello World, ada satu paket dan paket berisi MainActivity.java (ekstensi file dapat dihilangkan di tampilan Proyek).
- **com.example.hello.helloworld(androidTest):** Folder ini untuk pengujian berinstrumen Anda, dan memulai dengan file pengujian kerangka.
- **com.example.hello.helloworld(test):** Folder ini untuk pengujian unit dan memulai dengan file pengujian unit kerangka yang secara otomatis dibuat.

- Luaskan folder res. **Folder ini berisi semua resource untuk aplikasi Anda, termasuk gambar, file layout, string, ikon, dan penataan gaya. Folder ini berisi subfolder berikut:**

- **drawable.** Simpan semua gambar aplikasi Anda di folder ini.
- **layout.** Setiap activity memiliki paling tidak satu file layout yang menjelaskan UI di XML. Untuk Hello World, folder ini berisi activity\_main.xml.
- **mipmap.** Simpan ikon peluncur Anda di folder ini. Ada subfolder untuk setiap kepadatan layar yang didukung. Android menggunakan kepadatan layar, yaitu, jumlah piksel per inci untuk menentukan resolusi gambar yang diperlukan. Android

mengelompokkan kepadatan layar yang sebenarnya ke dalam kepadatan umum seperti medium (mdpi), tinggi (hdpi), atau ekstra-ekstra-ekstra-tinggi (xxxhdpi). Folder `ic_launcher.png` berisi ikon peluncur default untuk semua kepadatan yang didukung oleh aplikasi Anda.

- **values**. Sebagai ganti melakukan hardcode nilai-nilai seperti string, dimensi, dan warna di file XML dan Java, praktik terbaiknya adalah mendefinisikannya menurut file nilai. Ini membuatnya lebih mudah diubah dan konsisten di seluruh aplikasi Anda.

- **Luaskan subfolder values dalam folder res. Folder ini berisi subfolder berikut:**

- **colors.xml**. Tunjukkan warna default untuk tema pilihan Anda, dan Anda bisa menambahkan warna sendiri atau mengubahnya berdasarkan persyaratan aplikasi Anda.
- **dimens.xml**. Simpan ukuran tampilan dan objek untuk resolusi yang berbeda.
- **strings.xml**. Buat resource untuk semua string Anda. Ini memudahkan penerjemahan string ke bahasa lain.
- **styles.xml**. Semua gaya untuk aplikasi dan tema Anda ada di sini. Gaya membantu memberikan aplikasi Anda tampilan yang konsisten untuk semua elemen UI.

### Sistem pembangunan Gradle

Android Studio menggunakan Gradle sebagai sistem pembangunan. Seiring dengan kemajuan Anda sepanjang praktik ini, Anda akan mempelajari lebih banyak tentang gradle dan apa yang dibutuhkan untuk membangun dan menjalankan aplikasi Anda.

- Luaskan folder **\*Gradle Scripts**. Folder ini berisi semua file yang dibutuhkan oleh sistem pembangunan.
- Cari file `build.gradle (Module:app)`. Saat Anda menambahkan dependensi khusus aplikasi, seperti menggunakan pustaka tambahan, dependensi masuk ke dalam file ini

## Latihan 2.4. Membuat perangkat virtual (emulator)

Pada tugas ini, Anda akan menggunakan Android Virtual Device (AVD) manager untuk membuat perangkat virtual atau emulator yang menyimulasikan konfigurasi untuk jenis perangkat Android tertentu.

Menggunakan AVD Manager, Anda mendefinisikan karakteristik perangkat keras sebuah perangkat dan API level-nya, serta menyimpannya sebagai konfigurasi perangkat virtual. Saat Anda memulai emulator Android, emulator membaca konfigurasi yang ditetapkan dan membuat perangkat emulasi yang bertindak persis seperti versi fisik perangkat tersebut, namun berada di komputer Anda.

### 1) Membuat perangkat virtual

Untuk menjalankan emulator di komputer, Anda perlu membuat konfigurasi yang menjelaskan perangkat virtual.

- Di Android Studio, pilih Tools > Android > AVD Manager atau klik



ikon AVD Manager di bilah alat.

- Klik +Create Virtual Device.... (Jika Anda telah membuat perangkat virtual sebelumnya, jendela akan menunjukkan semua perangkat yang ada dan tombolnya ada di bagian bawah.)

Layar Select Hardware muncul menunjukkan daftar perangkat keras yang telah dikonfigurasi sebelumnya. Untuk setiap perangkat, tabel menunjukkan ukuran tampilan (Size), resolusi layar dalam piksel (Resolution), dan kepadatan piksel (Density).

Untuk perangkat Nexus 5, kepadatan pikselnya adalah xxhdpi, yang berarti aplikasi Anda menggunakan ikon peluncur di folder xxhdpi dari folder mipmap. Aplikasi Anda juga akan menggunakan layout dan drawable dari folder yang didefinisikan untuk kepadatan itu juga.


- Pilih perangkat keras Nexus 5 dan klik Next.
- Pada layar System Image, dari tab Recommended, pilih versi sistem Android yang akan dijalankan pada perangkat virtual. Anda bisa memilih gambar sistem terbaru.

Ada lebih banyak versi yang tersedia dari yang ditunjukkan di tab Recommended. Lihat tab x86 Images dan Other Images untuk melihatnya.

- Jika tautan Download terlihat di samping versi gambar sistem, berarti versi tersebut belum terpasang dan Anda perlu mengunduhnya. Jika perlu, klik tautan untuk mulai mengunduh dan klik Finish setelah selesai.
- Pada layar System Image, pilih gambar sistem dan klik Next.
- Verifikasi konfigurasi Anda dan klik Finish. (Jika jendela AVD Manager Your Android Devices tetap terbuka, Anda bisa melanjutkan dan menutupnya).

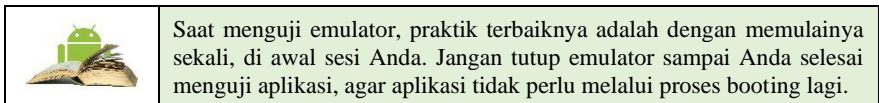
### Latihan 2.1. Menjalankan aplikasi Anda di emulator

Pada tugas ini Anda akhirnya akan menjalankan aplikasi Hello World.

- Jalankan aplikasi Anda di emulator
- Di Android Studio, pilih Run > Run app atau klik ikon Run  di bilah alat.
- Di jendela Select Deployment Target, di bawah Available Emulators, pilih Nexus 5 API 23 dan klik OK.

Emulator dimulai dan di-boot, seperti perangkat fisik. Tergantung kecepatan komputer Anda, ini mungkin akan memerlukan beberapa saat. Aplikasi Anda dibangun, dan setelah emulator siap, Android Studio akan mengunggah aplikasi ke emulator dan menjalankannya.

Anda seharusnya melihat aplikasi Hello World saat aplikasi di RUN.



### Tantangan penyusunan kode

Semua tantangan penyusunan kode bersifat opsional dan tidak diwajibkan untuk praktik selanjutnya. Anda bisa sepenuhnya menyesuaikan perangkat virtual.

- Pelajari dokumentasi AVD Manager.
- Buat satu atau beberapa perangkat virtual khusus.

## Latihan 2.5. Menambahkan pernyataan log ke aplikasi Anda

Dalam praktik ini, Anda akan menambahkan pernyataan log ke aplikasi, yang ditampilkan di jendela pencatatan log Android Monitor.

**Mengapa:** Pesan log adalah alat debug bermanfaat yang bisa Anda gunakan untuk memeriksa nilai, jalur eksekusi, dan melaporkan pengecualian.

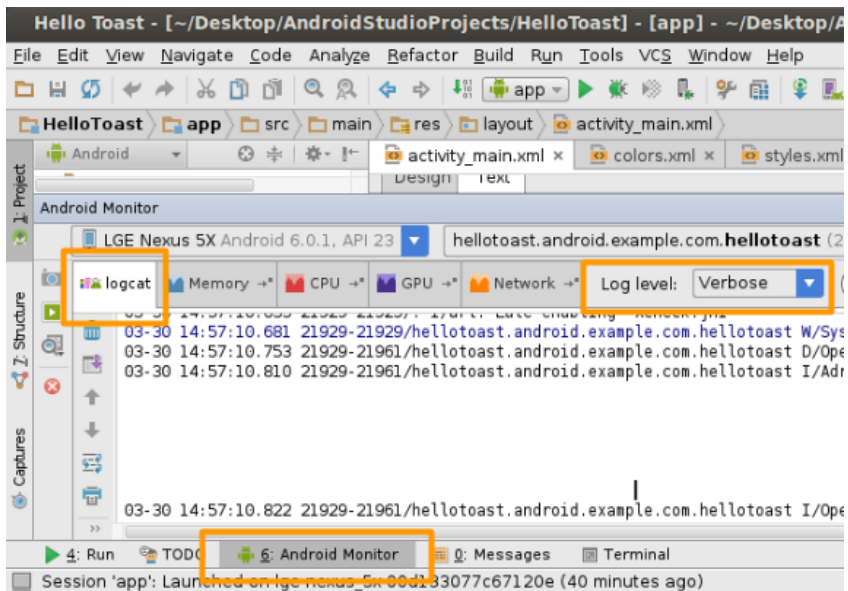
**Android Monitor** menampilkan informasi tentang aplikasi Anda.

1. Klik tombol **Android Monitor** di bawah Android Studio untuk membuka Android Monitor.

Secara default, ini membuka tab **logcat**, yang menampilkan informasi tentang aplikasi Anda saat sedang dijalankan. Jika Anda menambahkan pernyataan log ke aplikasi, pernyataan dicetak di sini juga.

Anda juga bisa memantau Memori, CPU, GPU, dan kinerja jaringan aplikasi dari tab lain Android Monitor. Ini dapat membantu debug dan kinerja menyempurnakan kode.

2. Level log default adalah **Verbose**. Di menu tarik-turun, ubah level log ke **Debug**.



Gambar 42 Level log default

Pernyataan log yang Anda tambahkan ke kode aplikasi mencetak pesan yang ditentukan oleh Anda di tab logcat Android Monitor. Misalnya:

```
Log.d("MainActivity", "Hello World");
```

Bagian dari pesan tersebut adalah:

- Log – Kelas Log. API untuk mengirim pesan log.
- d – Tingkat Log. Digunakan untuk memfilter tampilan pesan di logcat. "d" untuk debug. Tingkat log lainnya adalah "e" untuk error, "w" untuk warning, dan "i" untuk info.
- "MainActivity" – Argumen pertama adalah tag yang dapat digunakan untuk memfilter pesan di logcat. Ini biasanya adalah nama activity asal pesan. Tetapi, Anda dapat menjadikannya apa saja yang berguna untuk men-debug.

Sesuai ketentuan, tag log didefinisikan sebagai konstanta:

```
private static final String LOG_TAG =  
MainActivity.class.getSimpleName()
```

- "Hello world" – Argumen kedua adalah pesan yang sebenarnya.

### Menambahkan pernyataan log ke aplikasi Anda:

1. Buka aplikasi Hello World di Android studio, dan buka file MainActivity.
2. **File > Settings > Editor > General >Auto Import** (Mac: **Android Studio > Preferences > Editor > General >Auto Import**). Pilih semua kotak centang dan setel **Insert imports on paste** ke **All**. Impor yang jelas sekarang ditambahkan secara otomatis ke file Anda. Catat bahwa opsi "add unambiguous imports on the fly" penting bagi beberapa fitur Android seperti NumberFormat. Jika tidak dicentang, NumberFormat menampilkan kesalahan. Klik 'Apply' lalu klik tombol 'Ok'.
3. Dalam metode onCreate, tambahkan pernyataan log berikut:
4. `Log.d("MainActivity", "Hello World");`
5. Jika Android Monitor belum dibuka, klik tab Android Monitor di bawah Android Studio untuk membukanya. (Lihat tangkapan layar.)
6. Pastikan level Log di logcat Android Monitor disetel ke Debug atau Verbose (default).
7. Jalankan aplikasi Anda.



## Kode Solusi:

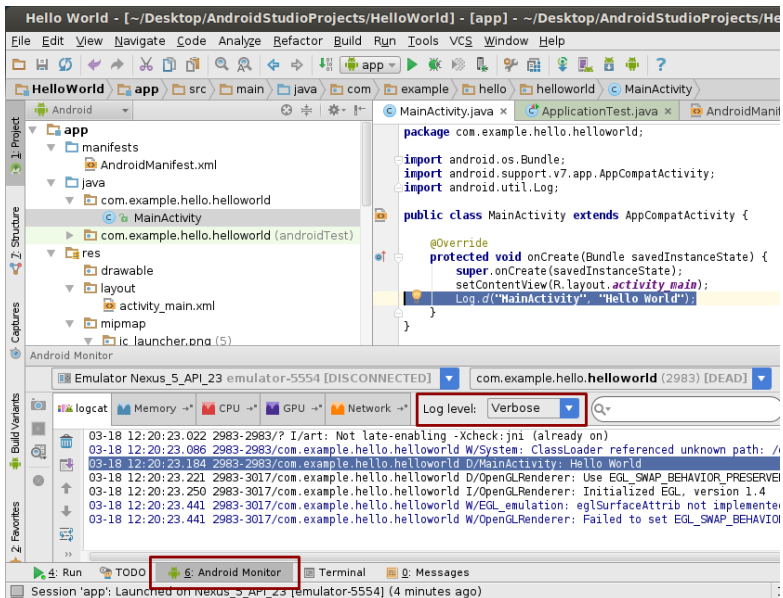
```
package com.example.hello.helloworld;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d("MainActivity", "Hello World");
    }
}
```

## Pesan Log Output

```
03-18 12:20:23.184 2983-
2983/com.example.hello.helloworld D/MainActivity:
Hello World
```



Gambar 43. Tampilan pernyataan log

## RANGKUMAN

Pada bab ini, Anda telah mempelajari cara:

1. Memasang Android Studio
  - Mendapatkan pemahaman dasar tentang alur kerja development setelah membukanya di Android Studio.
  - Mendapatkan pemahaman dasar tentang struktur aplikasi Android di lingkungan pembangunannya.
  - Mendapatkan pemahaman dasar tentang Manifes Android dan kegunaannya.
  - Menambahkan pernyataan log ke kode yang memberi Anda alat dasar untuk proses debug.
2. Menerapkan aplikasi Hello World pada emulator Android dan [opsional] perangkat seluler.

## REFERENSI

- Allan, A., 2013. Learning iOS Programming: From Xcode to App Store. " O'Reilly Media, Inc."
- Campbell, Iain. Symbian OS communications programming. Vol. 20. John Wiley & Sons, 2007.
- Fling, Brian. 2009, : Mobile Design and Development. O'Reilly Media.
- Firtman, Maximiliano. Programming the Mobile Web: Reaching Users on iPhone, Android, BlackBerry, Windows Phone, and more. " O'Reilly Media, Inc.", 2013.
- Harris, N., 2014. Beginning IOS Programming: Building and Deploying IOS Applications. John Wiley & Sons.
- Morris, Ben. The Symbian Os Architecture Sourcebook: Design And Solution Of A Mobile Phone Os. John Wiley & Sons, 2007.
- Petzold, Charles. Microsoft XNA framework edition: programming Windows Phone 7. Microsoft Press, 2010.
- Safaat N.H. 2014. Android: Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android. Bandung: Informatika.
- Stark, Jonathan. 2010. "Building Android Apps with HTML, CSS, and JavaScript". O'Reilly. Sebastopol. California. Dannen, Chris and White, Christopher. (2011). "Beginning iOS Apps with Facebook and Twitter APIsfor iPhone, iPad, and iPod touch". Apress. New York.
- Svennerberg, Gabriel. 2010. "Beginning Google Maps API 3". Apress. New York

## MATERI KOMPETENSI

Materi bab ini akan membahas:

- 1) Konsep user interface (UI) dan tool Android Studio
- 2) Tata letak (Layout) user interface
- 3) Atribut, parameter dan Jenis-jenis Layout

## YANG HARUS SUDAH ANDA KETAHUI

Untuk materi dan praktik ini Anda harus bisa:

- 1) Cara membuat aplikasi Hello World dengan Android Studio.
- 2) Memahami struktur direktori file project Android Studio.
- 3) Menjalankan aplikasi di emulator atau perangkat.

## YANG AKAN ANDA BUTUHKAN

Untuk materi dan praktik ini, Anda akan membutuhkan:

- 1) Komputer Mac, Windows, atau Linux. Tool Android Studio untuk persyaratan sistem terbaru.
- 2) Smartphone sebagai perangkat untuk pengujian secara nyata aplikasi yang telah dibuat.

## YANG AKAN ANDA PELAJARI

Anda akan belajar:

- 1) Cara membuat antarmuka pengguna interaktif dalam Layout Editor, dalam XML, dan secara terprogram.
- 2) Banyak terminologi baru.
- 3) Menggunakan editor layout di Android Studio.

# BAB 3

## LAYOUT USER INTERFACE



### Bab 3: Layout User Interface (UI)

#### 3.1. User Interface

User interface (UI) adalah sebuah desain yang digunakan sebagai penghubung antara manusia sebagai pengguna dengan software pada perangkat sehingga akan memudahkan pengoperasian dari perangkat tersebut. User interface merupakan salah satu komponen yang penting dalam sebuah aplikasi mobile seperti Android, karena user interface merupakan cara user untuk berkomunikasi dengan program. Sebuah user interface yang baik akan membuat user nyaman dengan aplikasi yang kita buat. Mudahnya, UI merupakan desain tampilan pada layar perangkat seperti yang dimiliki oleh smartphone, tablet dan berbagai jenis perangkat lainnya. Atau sebagian orang awam lebih mengenalnya sebagai tema.

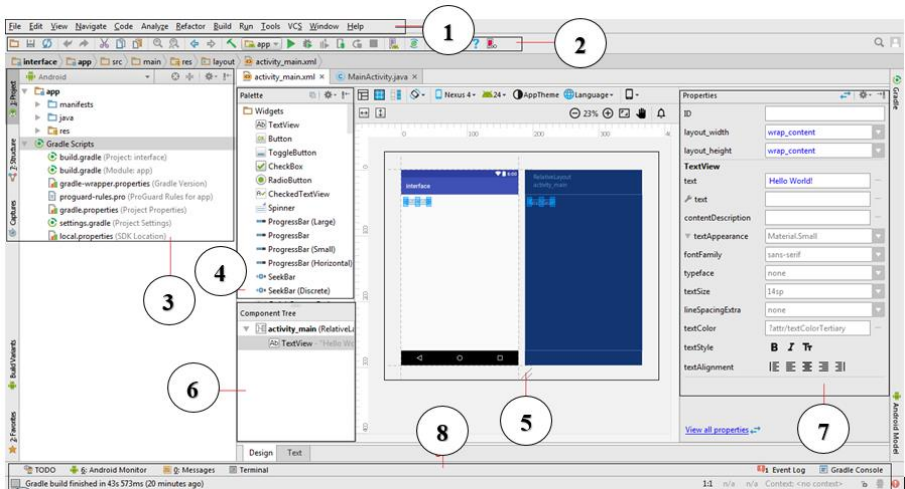
Sekarang ini UI bukan hanya untuk memudahkan interaksi antar manusia (pengguna) dengan perangkatnya saja, tetapi juga untuk mempercantik perangkat ini sendiri. Secara umum arsitek user interface (UI) pada aplikasi android adalah user interface yang meliputi Activity dan User Interface yang terdiri dari komponen. User interface aplikasi android akan mempengaruhi tingkat kemudahan menggunakan aplikasi tersebut. Ketika kita menggunakan sebuah aplikasi biasanya yang pertama kali dilihat yaitu tampilan aplikasinya. Baik itu layout tombol, gambar, teks dan lainnya. user interface yang menarik dan simple akan lebih disukai pengguna dibandingkan dengan user interface yang terlihat rumit.

Semua elemen antarmuka pengguna dalam aplikasi Android dibangun menggunakan objek View dan ViewGroup. View adalah objek yang menarik sesuatu di layar dan dapat berinteraksi dengan pengguna. ViewGroup merupakan sebuah objek yang menyimpan objek View lainnya (dan

ViewGroup) untuk mendefinisikan layout antarmuka. Android menyediakan sekumpulan subkelas View dan ViewGroup yang menawarkan kontrol masukan umum (seperti tombol dan bidang teks) serta berbagai model layout (seperti layout linier atau relatif).

User Interface Aplikasi Android Studio:

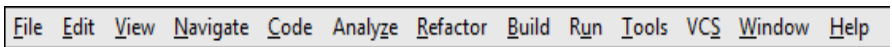
Berikut ini merupakan bagian-bagian penting dari Antar Muka (Interface) Android Studio.



Gambar 44. Interface Android Studio

### 1. Menu Bar

Seperti pada aplikasi lain Menu bar merupakan bagian antar muka (interface) pengguna yang berisi perintah dan opsi yang dapat dipilih untuk mengeksekusi suatu perintah.



Gambar 45. Menu Bar

### 2. Toolbar

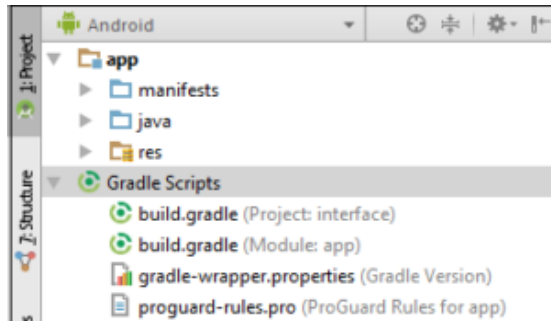
Dengan toolbar kita bisa mempercepat perintah pada sebuah aplikasi.



Gambar 46. Toolbar

### 3. Struktur Project

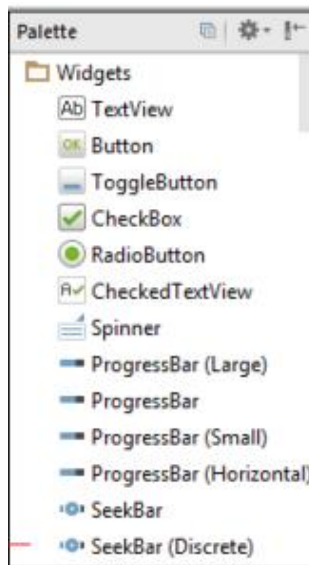
Pada bagian ini akan ditampilkan folder-folder dari sebuah project aplikasi android yang dibuat menggunakan android studio. Lebih jelasnya di sub Struktur Folder Project Android Studio



Gambar 47. Struktur Project

#### 4. Pallette

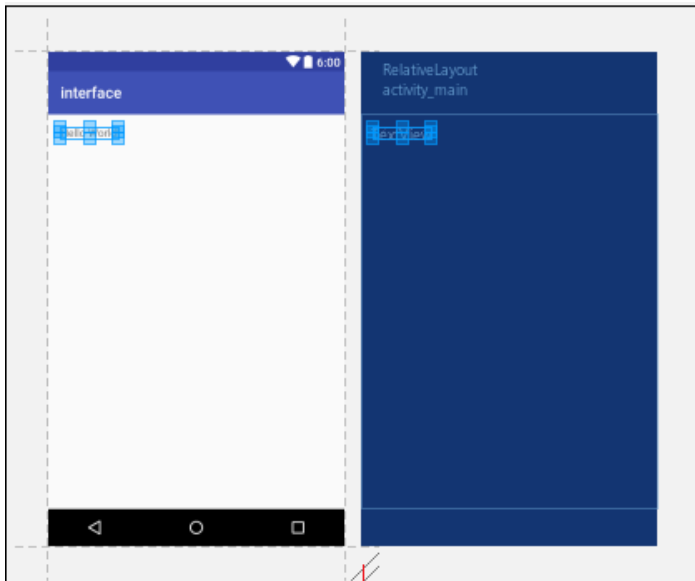
Di pallette tersedia semua tools untuk membuat aplikasi android dan enaknyalagi untuk menggunakannya cukup dengan mendrag and drop ke design android.



Gambar 48. Palette

## 5. Design Android

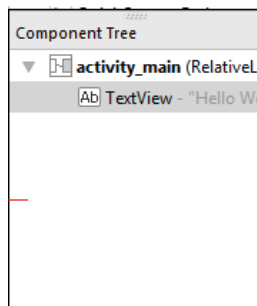
Tempat ini digunakan untuk mendesign layout aplikasi dengan cara drag and drop (tidak mengetikkan script xml)



Gambar 49. Desain Android

## 6. Component Tree

Setelah kita menggunakan tools yang ada di pallete maka akan ditampilkan pada component tree misal kita mendrag TextView (widget untuk membuat teks) ke design android. Maka nanti ditampilkan di component tree TextView.

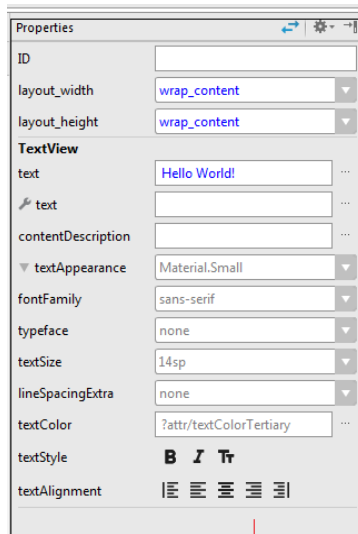


Gambar 50. Component Tree



## 7. Properties

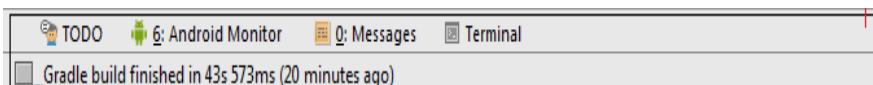
Disini akan ditampilkan pengaturan-pengaturan dari komponen yang digunakan untuk design aplikasi. Jika TextView maka akan ditampilkan untuk mengatur warna, size dan lainnya.



Gambar 51. Properties

## 8. Status Bar

Menampilkan proses pada Android Studio. Proses Loading, Error, dan lainnya.



Gambar 52. Status bar

### 3.2. Layout User Interface

User interface untuk setiap komponen aplikasi Anda didefinisikan menggunakan hierarki objek View dan ViewGroup. Setiap kelompok tampilan merupakan kontainer tak terlihat yang mengelola tampilan anak, sementara tampilan anak ini dapat berupa kontrol masukan atau widget lain yang menggambar sebagian dari UI. Pohon hierarki ini bisa sederhana atau bisa juga kompleks sesuai kebutuhan (namun yang sederhana paling baik untuk kinerja).

Untuk mendeklarasikan layout, Anda dapat menyediakan objek View dalam kode dan mulai membangun pohon. Pada android studio pembuatan User Interface dibagi menjadi dua, yaitu:

- 1) Procedural  
UI procedural yaitu menggunakan bahasa pemrograman java.
- 2) Declarative

UI declarative yaitu menggunakan bahasa xml. Agar lebih mudah dipahami pembahasan user interface kali ini lebih menekankan menggunakan bahasa xml. Pada android terdapat dua jenis user interface yaitu:

- View, Merupakan komponen yang ditampilkan pada layar yang biasanya bisa digunakan untuk berinteraksi dengan pengguna.
- ViewGroups, Merupakan komponen yang berperan mengatur komponen-komponen view lainnya.

Anda tidak harus membangun semua UI menggunakan objek View dan ViewGroup. Android menyediakan beberapa komponen aplikasi yang menawarkan layout UI standar yang tinggal didefinisikan materinya. Komponen UI ini masing-masing memiliki set API unik yang dijelaskan dalam masing-masing dokumennya, seperti Menambahkan Bilah Aplikasi, Dialog, dan Notifikasi Status.

Layout mendefinisikan struktur visual untuk antarmuka pengguna, seperti UI sebuah activity atau widget\_aplikasi. Anda dapat mendeklarasikan layout dengan dua cara:

- Deklarasikan elemen UI dalam XML. Android menyediakan sebuah kosakata XML sederhana yang sesuai dengan kelas dan subkelas View, seperti halnya untuk widget dan layout.
- Buat instance elemen layout saat waktu proses. Aplikasi Anda bisa membuat objek View dan ViewGroup (dan memanipulasi propertinya) lewat program.

Kerangka kerja Android memberi Anda fleksibilitas untuk menggunakan salah satu atau kedua metode ini guna mendeklarasikan dan mengelola UI aplikasi Anda. Misalnya, Anda bisa mendeklarasikan layout default aplikasi Anda dalam XML, termasuk elemen-elemen layar yang akan muncul di dalamnya dan di propertinya. Anda nanti bisa menambahkan kode dalam aplikasi yang akan memodifikasi status objek layar, termasuk yang dideklarasikan dalam XML, saat waktu proses.

- Anda juga harus mencoba alat Hierarchy Viewer, untuk men-debug layout — alat ini akan menampilkan nilai-nilai properti layout,

menggambar bentuk kerangka dengan indikator pengisi/margin, dan tampilan yang dirender penuh selagi Anda merunut pada emulator atau perangkat.

- Alat layoutopt memungkinkan Anda menganalisis layout dan hierarki dengan untuk mengetahui ketidakefisienan atau masalah lainnya.

Keuntungan mendeklarasikan UI dalam XML adalah karena hal ini memungkinkan Anda memisahkan penampilan aplikasi dari kode yang mengontrol perilakunya dengan lebih baik. Keterangan UI Anda bersifat eksternal bagi kode aplikasi Anda, yang berarti bahwa Anda bisa memodifikasi atau menyesuaikannya tanpa harus memodifikasi dan mengompilasi ulang kode sumber. Misalnya, Anda bisa membuat layout XML untuk berbagai orientasi layar, berbagai ukuran layar perangkat, dan berbagai bahasa. Selain itu, mendeklarasikan layout dalam XML akan mempermudah Anda memvisualisasikan struktur UI, sehingga lebih mudah men-debug masalahnya. Karena itu, dokumen ini berfokus pada upaya mengajarkan Anda cara mendeklarasikan layout dalam XML. Jika Anda tertarik dalam membuat instance objek View saat waktu proses, lihat referensi kelas ViewGroup dan View.

Secara umum, kosakata XML untuk mendeklarasikan elemen UI mengikuti dengan sangat mirip struktur serta penamaan kelas dan metode, dengan nama elemen dipadankan dengan nama kelas dan nama atribut dipadankan dengan metode. Sebenarnya, pemadanan ini kerap kali begitu jelas sehingga Anda bisa menebak atribut XML yang berpadanan dengan sebuah metode kelas, atau menebak kelas yang berpadanan dengan sebuah elemen XML. Akan tetapi, perhatikan bahwa tidak semua kosakata identik. Dalam beberapa kasus, ada sedikit perbedaan penamaan. Misalnya, elemen `EditText` memiliki atribut `text` yang berpadanan dengan `EditText.setText()`.

- Tulis XML :

Dengan menggunakan kosakata XML Android, Anda bisa mendesain secara cepat layout UI dan elemen layar yang dimuatnya, sama dengan cara membuat laman web dalam HTML — dengan serangkaian elemen tersarang.

Tiap file layout harus berisi persis satu elemen akar, yang harus berupa sebuah objek View atau ViewGroup. Setelah mendefinisikan elemen akar, Anda bisa menambahkan objek atau widget layout tambahan sebagai elemen anak untuk membangun hierarki View yang mendefinisikan layout Anda secara bertahap. Misalnya, inilah layout XML yang menggunakan LinearLayout vertikal untuk menyimpan TextView dan Button.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView
        android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>

```

Setelah Anda mendeklarasikan layout dalam XML, simpanlah file dengan ekstensi `.xml`, dalam direktori `res/layout/` proyek Android, sehingga nanti bisa dikompilasi dengan benar.

- Muat Resource XML:

Saat mengompilasi aplikasi, masing-masing file layout XML akan dikompilasi dalam sebuah resource View. Anda harus memuat resource layout dari kode aplikasi, dalam implementasi callback Activity.onCreate(). Lakukan dengan memanggil setContentView(), dengan meneruskan acuan ke resource layout berupa: R.layout.layout\_file\_name. Misalnya, jika XML layout Anda disimpan sebagai `main_layout.xml`, Anda akan memuatnya untuk Activity seperti ini:

```

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main_layout);
}

```

Metode callback `onCreate ()` dalam Activity dipanggil oleh kerangka kerja Android saat Activity dijalankan.

- Atribut Layout :

Setiap layout memiliki seperangkat atribut yang menentukan properti visual layout itu. Ada beberapa atribut umum di antara semua layout dan atribut lainnya yang spesifik untuk layout itu. Berikut ini adalah atribut umum dan akan diterapkan ke semua layout. Setiap objek View dan ViewGroup mendukung variasi atribut XML-nya sendiri. Sebagian atribut bersifat spesifik untuk objek View (misalnya, TextView mendukung atribut `textSize`), namun atribut ini juga diwarisi oleh sembarang objek View yang dapat memperluas kelas ini.

Sebagian atribut bersifat umum untuk semua objek View, karena diwarisi dari kelas root view (seperti atribut `id`). Dan, atribut lain dianggap sebagai "parameter layout" yaitu atribut yang menjelaskan orientasi layout tertentu dari objek View, seperti yang didefinisikan oleh objek ViewGroup parent dari objek itu.

Tabel 5. Atribut Layout

Attribute	Deskripsi
<code>android:id</code>	ID yang secara unik mengidentifikasi tampilan
<code>android:layout_width</code>	Lebar layout.
<code>android:layout_height</code>	Tinggi layoutnya
<code>android:layout_marginTop</code>	Ruang ekstra di sisi atas layout
<code>android:layout_marginBottom</code>	Ruang ekstra di sisi bawah layout
<code>android:layout_marginLeft</code>	Ruang ekstra di sisi kiri layout
<code>android:layout_marginRight</code>	Ruang ekstra di sisi kanan layout
<code>android:layout_gravity</code>	menentukan bagaimana child Views diposisikan
<code>android:layout_weight</code>	Menentukan berapa banyak ruang ekstra di layout harus dialokasikan ke View.
<code>android:layout_x</code>	Menentukan x-koordinat layout
<code>android:layout_y</code>	Menentukan y-koordinat layout
<code>android:layout_width</code>	Lebar layout
<code>android:paddingLeft</code>	Padding kiri yang diisi untuk layout
<code>android:paddingRight</code>	Padding kanan yang diisi untuk layout
<code>android:paddingTop</code>	Padding atas yang diisi untuk layout
<code>android:paddingBottom</code>	Padding bawah yang diisi untuk layout

Di sini lebar dan tinggi adalah dimensi layout / tampilan yang dapat ditentukan dalam hal dp (Piksel Density-independen), sp (Skala-independen)

Piksel), **pt** (Poin yang 1/72 inci), **px** (Piksel), **mm** (Milimeter) dan akhirnya dalam (inci). Dapat menentukan lebar dan tinggi dengan pengukuran yang tepat tetapi lebih sering, menggunakan salah satu dari konstanta ini untuk mengatur lebar atau tinggi.

- `android:layout_width = wrap_content` info view untuk mengukur sendiri ke dimensi yang diperlukan oleh isinya.
- `android:layout_width = fill_parent` info view untuk menjadi sebesar tampilan parentnya.

Atribut graviti memainkan peran penting dalam memposisikan objek tampilan dan dapat mengambil satu atau lebih (dipisahkan oleh '|') dari nilai konstan berikut.

Tabel 6. Contant

Constant	Value	Deskripsi
<code>top</code>	<code>0x30</code>	Dorong objek ke bagian atas container, tanpa mengubah ukurannya.
<code>bottom</code>	<code>0x50</code>	Dorong objek ke bagian bawah container, tanpa mengubah ukurannya..
<code>left</code>	<code>0x03</code>	Dorong objek ke bagian kiri container, tanpa mengubah ukurannya.
<code>right</code>	<code>0x05</code>	Dorong objek ke bagian kanan container, tanpa mengubah ukurannya.
<code>center_vertical</code>	<code>0x10</code>	Tempatkan objek di center-vertikal container, tanpa mengubah ukurannya.
<code>fill_vertical</code>	<code>0x70</code>	Tambahkan ukuran vertikal objek jika diperlukan sehingga benar-benar mengisi container.
<code>center_horizontal</code>	<code>0x01</code>	Tempatkan objek di tengah horizontal container, tanpa mengubah ukurannya..
<code>fill_horizontal</code>	<code>0x07</code>	Tambahkan ukuran horizontal objek jika diperlukan sehingga benar-benar mengisi container.
<code>center</code>	<code>0x11</code>	Tempatkan objek di tengah container di sumbu vertikal dan horizontal, tidak mengubah ukurannya..

fill	0x77	Tambahkan ukuran horizontal dan vertikal dari objek jika diperlukan sehingga benar-benar mengisi wadahnya..
clip_vertical	0x80	Opsi tambahan yang dapat diatur untuk memiliki tepi atas dan / atau bawah dari child yang dipotong ke batas containernya.  Klip akan didasarkan pada gravitasi vertikal: gravitasi atas akan memotong tepi bawah, gravitasi bawah akan memotong tepi atas, dan tidak akan memotong kedua sisi.
clip_horizontal	0x08	Opsi tambahan yang dapat diatur untuk memiliki tepi kiri dan / atau kanan child clip ke batas containernya.  Klip akan didasarkan pada gravitasi horizontal: gravitasi kiri akan memotong tepi kanan, gravitasi kanan akan memotong tepi kiri, dan tidak akan memotong kedua sisi..
start	0x0080 0003	Dorong objek ke awal container, tanpa mengubah ukurannya.
end	0x0080 0005	Dorong objek ke akhir container, tanpa mengubah ukurannya

- ID

Objek View apa saja dapat memiliki ID integer yang dikaitkan dengannya, untuk mengidentifikasi secara unik View dalam pohon. Bila aplikasi dikompilasi, ID ini akan diacu sebagai integer, namun ID biasanya ditetapkan dalam file XML layout sebagai string, dalam atribut `id`. Ini atribut XML yang umum untuk semua objek View (yang didefinisikan oleh kelas `View`) dan Anda akan sering sekali menggunakannya. Sintaks untuk ID dalam tag XML adalah:

```
android:id="@+id/my_button"
```

Berikut ini adalah deskripsi singkat dari tanda @ dan + -

- Simbol-at (@) di awal string menunjukkan bahwa parser XML harus menguraikan dan memperluas sisa string ID dan mengidentifikasinya sebagai sumber ID.
- Simbol plus (+) berarti ini adalah nama resource baru yang harus dibuat dan ditambahkan ke resource kami. Untuk membuat instance objek tampilan dan mengambilnya dari layout, gunakan yang berikut.

```
Button myButton = (Button) findViewById(R.id.my_button);
```

Saat mengacu sebuah ID resource Android, Anda tidak memerlukan simbol tanda tambah, namun harus menambahkan namespace paket android, sehingga:

```
android:id="@android:id/empty"
```

Dengan namespace paket android yang tersedia, kita sekarang mengacu ID dari kelas resource android.R. daripada kelas resource lokal.

Untuk membuat tampilan dan mengacunya dari aplikasi, pola yang umum adalah:

- Mendefinisikan tampilan/widget dalam file layout dan memberinya ID unik:

```
<Button android:id="@+id/my_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/my_button_text"/>
```

- Kemudian buat instance objek tampilan dan tangkap instance itu dari layout (biasanya dalam metode onCreate()):

```
Button myButton = (Button) findViewById(R.id.my_button);
```

Mendefinisikan ID untuk objek tampilan adalah penting saat membuat RelativeLayout. Dalam layout relatif, tampilan saudara bisa mendefinisikan layout secara relatif terhadap tampilan saudara lainnya, yang diacu melalui ID unik. ID tidak perlu unik di seluruh pohon, namun harus unik di bagian pohon yang Anda cari (yang mungkin sering kali seluruh pohon, jadi lebih baik benar-benar unik bila memungkinkan).



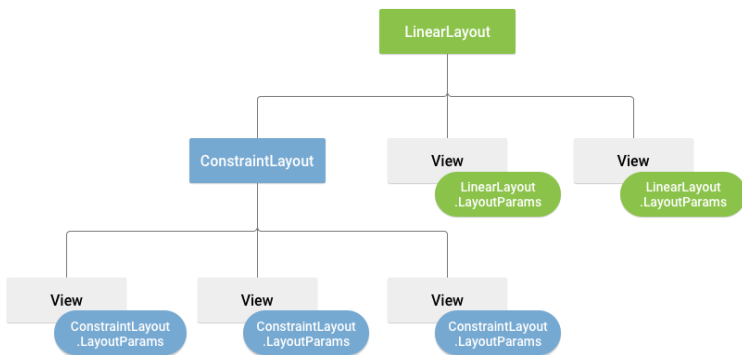
## ▪ Parameter Layout

Atribut layout XML bernama `layout_something` mendefinisikan parameter layout View yang cocok untuk ViewGroup tempatnya berada.

Setiap kelas ViewGroup mengimplementasikan kelas tersarang yang memperluas `ViewGroup.LayoutParams`. Subkelas ini berisi tipe properti yang mendefinisikan ukuran dan posisi masing-masing tampilan anak, sebagaimana mestinya untuk grup tampilan. Seperti yang bisa Anda lihat dalam gambar 1, grup tampilan parent mendefinisikan parameter layout untuk masing-masing tampilan anak (termasuk grup child view).

Perhatikan bahwa setiap subkelas `LayoutParams` memiliki sintaksnya sendiri untuk menetapkan nilai-nilai. Tiap elemen anak harus mendefinisikan `LayoutParams` yang semestinya bagi parentnya, meskipun elemen itu bisa juga mendefinisikan `LayoutParams` untuk anak-anaknya sendiri. Semua grup tampilan berisi lebar dan tinggi (`layout_width` dan `layout_height`), dan masing-masing tampilan harus mendefinisikannya. Banyak `LayoutParams` yang juga menyertakan margin dan border opsional. Anda bisa menetapkan lebar dan tinggi dengan ukuran persis, meskipun Anda mungkin tidak ingin sering-sering melakukannya. Lebih sering, Anda akan menggunakan salah satu konstanta ini untuk mengatur lebar atau tinggi:

- `wrap_content` memberi tahu tampilan Anda agar menyesuaikan sendiri ukurannya dengan dimensi yang dibutuhkan oleh materinya.
- `match_parent` memberi tahu tampilan Anda agar menjadi sebesar yang akan diperbolehkan oleh kelompok tampilan parentnya.



Gambar 53. Hierarki View Layout Parameter

Secara umum, menetapkan lebar dan tinggi layout dengan satuan mutlak seperti piksel tidaklah disarankan. Sebagai gantinya, menggunakan pengukuran

relatif seperti satuan piksel yang tidak bergantung pada kerapatan (dp), `wrap_content`, atau `match_parent`, adalah sebuah pendekatan yang lebih baik, karena membantu memastikan bahwa aplikasi Anda akan ditampilkan dengan benar pada berbagai ukuran layar perangkat.

- Posisi Layout

Geometri tampilan adalah persegi panjang. Sebuah tampilan memiliki lokasi, yang dinyatakan berupa pasangan koordinat kiri dan atas, dan dua dimensi, yang dinyatakan berupa lebar dan tinggi. Satuan untuk lokasi dan dimensi adalah piksel.

Lokasi tampilan dapat diambil dengan memanggil metode `getLeft()` dan `getTop()`. Metode terdahulu mengembalikan koordinat kiri, atau X, persegi panjang yang mewakili tampilan. Metode selanjutnya mengembalikan koordinat atas, atau Y, persegi panjang yang mewakili tampilan. Kedua metode ini mengembalikan lokasi tampilan relatif terhadap parentnya. Misalnya, bila `getLeft()` mengembalikan 20, berarti tampilan berlokasi 20 piksel ke kanan dari tepi kiri parent langsungnya.

Selain itu, beberapa metode praktis ditawarkan untuk menghindari komputasi yang tidak perlu, yakni `getRight()` dan `getBottom()`. Kedua metode ini mengembalikan koordinat tepi kanan dan bawah persegi panjang yang mewakili tampilan. Misalnya, memanggil `getRight()` serupa dengan komputasi berikut: `getLeft() + getWidth()`.

- Ukuran, Pengisi, dan Margin

Ukuran tampilan dinyatakan dengan lebar dan tinggi. Tampilan sebenarnya memiliki dua pasang nilai lebar dan tinggi. Pasangan pertama disebut lebar terukur dan tinggi terukur. Dimensi ini mendefinisikan seberapa besar tampilan yang diinginkan dalam parentnya. Dimensi terukur bisa diperoleh dengan memanggil `getMeasuredWidth()` dan `getMeasuredHeight()`.

Pasangan kedua cukup disebut dengan lebar dan tinggi, atau kadang-kadang lebar penggambaran dan tinggi penggambaran. Dimensi-dimensi ini mendefinisikan ukuran tampilan sebenarnya pada layar, saat digambar dan setelah layout. Nilai-nilai ini mungkin, namun tidak harus, berbeda dengan lebar dan tinggi terukur. Lebar dan tinggi bisa diperoleh dengan memanggil `getWidth()` dan `getHeight()`.

Untuk mengukur dimensinya, tampilan akan memperhitungkan pengisinya (padding). Pengisi dinyatakan dalam piksel untuk bagian kiri, atas, kanan, dan bawah tampilan. Pengisi bisa digunakan untuk meng-offset isi tampilan dengan

piksel dalam jumlah tertentu. Misalnya, pengisi kiri sebesar 2 akan mendorong isi tampilan sebanyak 2 piksel ke kanan dari tepi kiri. Pengisi bisa diatur menggunakan metode `setPadding(int, int, int, int)` dan diketahui dengan memanggil `getPaddingLeft()`, `getPaddingTop()`, `getPaddingRight()`, dan `getPaddingBottom()`.

Meskipun bisa mendefinisikan pengisi, tampilan tidak menyediakan dukungan untuk margin. Akan tetapi, grup tampilan menyediakan dukungan tersebut. Lihat `ViewGroup` dan `ViewGroup.MarginLayoutParams` untuk informasi lebih jauh.

- Layout Umum

Tiap subkelas dari kelas `ViewGroup` menyediakan cara unik untuk menampilkan tampilan yang Anda sarangkan di dalamnya. Di bawah ini adalah beberapa tipe layout lebih umum yang dibuat ke dalam platform Android.



Walaupun Anda bisa menyarangkan satu atau beberapa layout dalam layout lain untuk mendapatkan desain UI, Anda harus berusaha menjaga hierarki layout sedangkal mungkin. Layout Anda akan digambar lebih cepat jika memiliki layout tersarang yang lebih sedikit (hierarki tampilan yang melebar lebih baik daripada hierarki tampilan yang dalam).

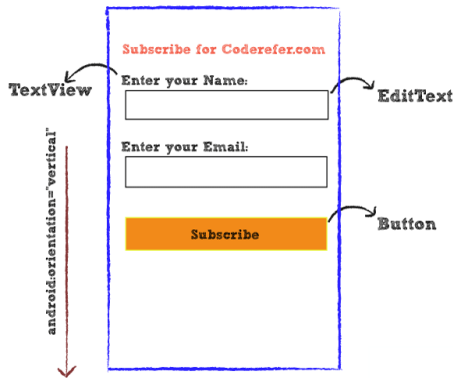
### 3.3. Jenis-Jenis Layout

#### a. Linier Layout

Linear layout adalah layout yang terstruktur. linear layout akan meletakkan elemen yang ada di dalamnya secara berurutan tergantung orientationnya. jika orientationnya vertical, maka peletakan elemen akan terurut kebawah. Sedangkan jika orientationnya horizontal (secara default, orientation horizontal tidak ditulis di dalam XML), maka peletakan elemennya akan terurut ke samping. Kesederhanaan Linear Layout memudahkan penggunaan namun membatasi fleksibilitasnya. Untuk aplikasi yang memiliki design tidak terlalu kompleks maka linear layout sangat cocok untuk digunakan.

Tabel 7. Linear Layout Attributes

Attribute	Deskripsi
<code>android:id</code>	ID yang secara unik mengidentifikasi layout.
<code>android:baselineAligned</code>	harus berupa nilai boolean, baik "benar" atau "salah" dan mencegah layout dari menyejajarkan garis dasar anak-anaknya.
<code>android:baselineAlignedChildIndex</code>	Ketika layout linier merupakan bagian dari layout lain baseline align, itu dapat menentukan mana dari anak-anak ke baseline align.
<code>android:divider</code>	Ini dapat digambar untuk digunakan sebagai pembagi vertikal antara tombol. Anda menggunakan nilai warna, dalam bentuk "#rgb", "#argb", "#rrggbb", atau "#aarrggbb".
<code>android:gravity</code>	Ini menentukan bagaimana suatu objek harus memposisikan isinya, pada sumbu X dan Y. Nilai yang mungkin adalah atas, bawah, kiri, kanan, tengah, center_vertical, center_horizontal dll.
<code>android:orientation</code>	Ini menentukan arah pengaturan dan Anda akan menggunakan "horizontal" untuk sebuah baris, "vertikal" untuk kolom. Standarnya horisontal.
<code>android:weightSum</code>	Sum child weight



Gambar 54. Contoh tampilan linear layout vertical

- Tag LinearLayout

Pada saat kita membuat project baru menggunakan android studio default layout yang digunakan adalah RelativeLayout. Maka kita harus merubahnya menggunakan Tag LinearLayout yaitu tag pembuka dari Linear Layout, juga harus dilengkapi dengan layout\_width dan height jika tidak maka akan terjadi error. Untuk orientationnya sendiri defaultnya horisontal.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
```

- Tag Orientation

Kita dapat mengatur orientasi untuk ditampilkan secara horisontal atau vertikal, sehingga item muncul dalam satu kolom atau satu baris. Lengkapi kode diatas dengan menambahkan tag:

```
android:orientation="horizontal"
```

sehingga menjadi :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
```

## Contoh pengguna tag orientation horizontal yaitu:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal" >
    <Button
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:text="Button 1"
        android:background="@color/colorAccent"
        android:layout_margin="2dp"
        android:textColor="#fff"
        android:textStyle="bold"/>
    <Button
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:text="Button 2"
        android:background="@color/colorAccent"
        android:layout_margin="2dp"
        android:textColor="#fff"
        android:textStyle="bold"/>
    <Button
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:text="Button 3"
        android:background="@color/colorAccent"
        android:layout_margin="2dp"
        android:textColor="#fff"
        android:textStyle="bold" />
</LinearLayout>
```

## Contoh pengguna tag orientation vertical yaitu:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical" >

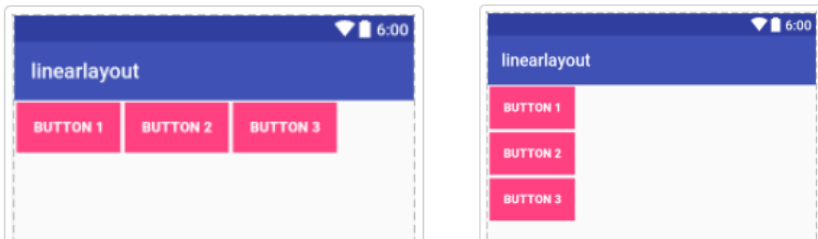
    <Button
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:text="Button 1"
        android:background="@color/colorAccent"
```

```

        android:layout_margin="2dp"
        android:textColor="#fff"
        android:textStyle="bold"/>
<Button
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:text="Button 2"
    android:background="@color/colorAccent"
    android:layout_margin="2dp"
    android:textColor="#fff"
    android:textStyle="bold"/>
<Button
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:text="Button 3"
    android:background="@color/colorAccent"
    android:layout_margin="2dp"
    android:textColor="#fff"
    android:textStyle="bold"/>
</LinearLayout>

```

Maka hasilnya:



Gambar 55. Layout horizontal dan vertikal

- Tag Layout Gravity

Tag gravity digunakan untuk mengatur layout komponen yang ada pada linear layout. Contohnya:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:orientation="vertical" >

    <Button
        android:layout_width="100dp"

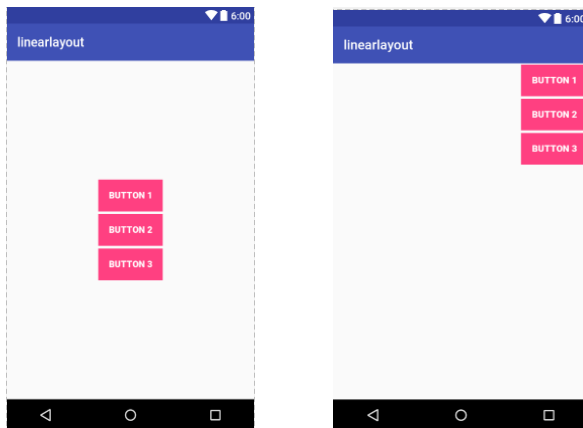
```

```

        android:layout_height="wrap_content"
        android:text="Button 1"
        android:background="@color/colorAccent"
        android:layout_margin="2dp"
        android:textColor="#fff"
        android:textStyle="bold"/>
<Button
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:text="Button 2"
    android:background="@color/colorAccent"
    android:layout_margin="2dp"
    android:textColor="#fff"
    android:textStyle="bold"/>
<Button
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:text="Button 3"
    android:background="@color/colorAccent"
    android:layout_margin="2dp"
    android:textColor="#fff"
    android:textStyle="bold"/>
</LinearLayout>

```

Maka hasilnya:



Gambar 56. Layout\_gravity center dan layout\_gravity right

- Tag Weight

Weight digunakan untuk memberikan bobot pada komponen layout. Contohnya berikut:



```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal" >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:text="teks 1"
        android:background="@color/colorAccent"
        android:layout_weight="5"/>
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:text="teks 2"
        android:background="@color/colorPrimary"
        android:layout_weight="5"/>

<TextView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:text="teks 3"
    android:background="@color/colorPrimaryDark"
    android:layout_weight="5"/>

</LinearLayout>

```

Maka hasilnya akan seperti gambar dibawah ini:



Gambar 57. Weight pada orientation horizontal

## b. Relative Layout

Relative Layout merupakan layout yang berfungsi untuk mengatur layout komponen atau widget aplikasi android dengan cara relative (secara bebas) tidak hanya vertikal atau horisontal saja seperti pada Linear Layout.

Relative Layout memungkinkan kita untuk memposisikan komponen mana saja yang kita inginkan, sehingga dianggap sebagai layout yang paling fleksibel. Karena alasan tersebut Relative layout menjadi layout yang paling banyak digunakan setelah Linear Layout di Android.

Adapun beberapa **Tag xml** penting yang sering digunakan pada **Relative Layout** yaitu:

Tabel 8. Tag- tag pada Relative Layout

Tag	Keterangan
<code>android:layout_alignParentBottom</code>	Digunakan untuk menata letak komponen di tepi paling bawah layar.
<code>android:layout_alignParentTop</code>	Digunakan untuk menata letak komponen di tepi paling atas layar.
<code>android:layout_alignParentLeft</code>	Digunakan untuk menata letak komponen di tepi paling kiri layar.
<code>android:layout_alignParentRight</code>	Digunakan untuk menata letak komponen di tepi paling kanan layar.
<code>android:layout_below</code>	Digunakan untuk memposisikan komponen di bawah komponen (widget) yang lainnya.
<code>android:layout_above</code>	Digunakan untuk memposisikan komponen di atas komponen yang lainnya.
<code>android:layout_toLeftOf</code>	Digunakan untuk memposisikan komponen di sebelah kiri komponen yang lainnya.
<code>android:layout_toRightOf</code>	Digunakan untuk memposisikan komponen di sebelah kanan komponen yang lainnya.

Contoh penggunaan Relative Layout:

## Buat project baru lalu ketikkan kode-kode berikut pada Activity\_main.xml

Activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#ef0000" >

    <EditText
        android:id="@+id/name"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:background="#f8eff8f3"
        android:hint="Nama "
        android:padding="16dp" />

    <EditText
        android:id="@+id/email"
        android:layout_width="180dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="5dp"
        android:layout_toRightOf="@+id/name"
        android:background="#f8eff8f3"
        android:hint="Alamat "
        android:padding="16dp" />

    <EditText
        android:id="@+id/subject"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/name"
        android:layout_marginTop="16dp"
        android:background="#f8eff8f3"
        android:hint="Subjek"
        android:padding="16dp" />

    <EditText
        android:id="@+id/message"
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:layout_below="@+id/subject"
        android:layout_marginTop="16dp"
        android:background="#f8eff8f3"
        android:gravity="top"
        android:hint="Pesan"
        android:padding="16dp" />
```

```

<Button
    android:id="@+id/submit"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/message"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="16dp"
    android:padding="16dp"
    android:text="KIRIM" />
</RelativeLayout>

```

**Contoh lainnya yaitu:**

```

<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <TextView
        android:id="@+id/lblComments"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Komentar"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:textSize="20dp"
        android:padding="10dp"/>

    <EditText
        android:id="@+id/txtComments"
        android:layout_width="match_parent"
        android:layout_height="170dp"
        android:textSize="18sp"
        android:layout_alignLeft="@+id/lblComments"
        android:layout_below="@+id/lblComments"
        android:layout_centerHorizontal="true"
        android:background="#3222"
        android:layout_margin="3dp"/>

    <Button
        android:id="@+id/btnSave"
        android:layout_width="125dp"
        android:layout_height="wrap_content"
        android:text="Simpan"
        android:layout_below="@+id/txtComments"
        android:layout_alignRight="@+id/txtComments"
        android:background="#f124"
        android:textColor="#fff"
        android:layout_margin="3dp"

```

```

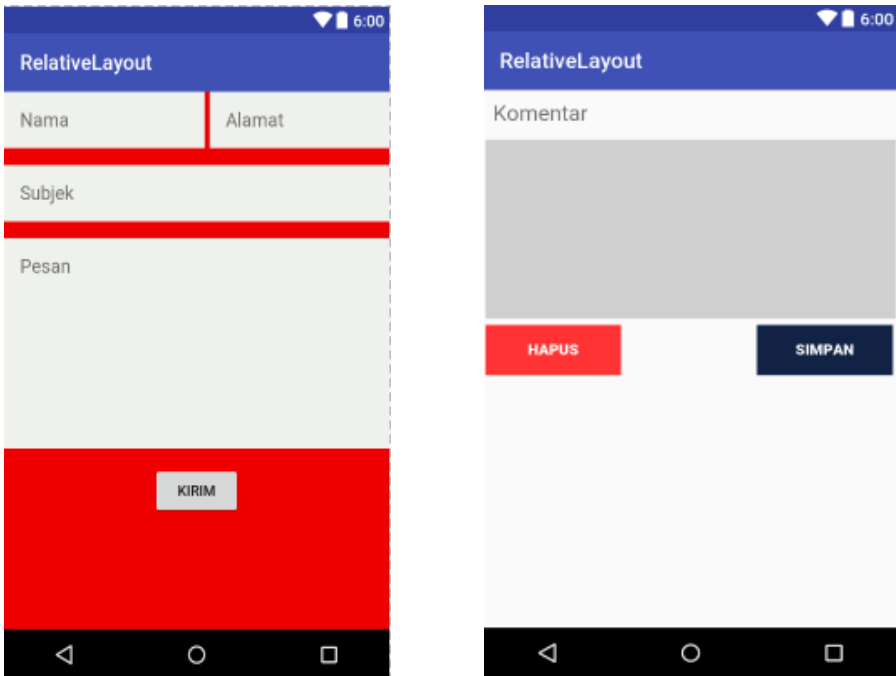
        android:textStyle="bold"/>

<Button
    android:id="@+id/button1"
    android:layout_width="125dp"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/btnSave"
    android:layout_alignBottom="@+id/btnSave"
    android:layout_alignParentLeft="true"
    android:text="Hapus"
    android:background="#ff33"
    android:textColor="#fff"
    android:layout_margin="3dp"
    android:textStyle="bold"/>

</RelativeLayout>

```

Hasil dari tag diatas yaitu seperti gambar dibawah ini:



Gambar 58. Hasil tampilan relative layout

### c. Tabel Layout

Tabel Layout adalah sebuah layout yang terdiri dari kolom dan baris, setiap objek atau view yang berada didalam Table Layout, akan tersusun seperti tabel, yaitu ada kolom dan juga barisnya, Table Layout digunakan untuk mempermudah kita dalam merancang User Interface pada Activity.

Tabel Layout terdiri dari:

- Row/ baris pada dasarnya digunakan untuk menyimpan satu jenis record, hanya satu informasi yang dapat disimpan.
- Kolom adalah sub bagian terbagi dari setiap baris dan satu baris dapat menampung beberapa jenis kolom. Setiap kolom terdiri dari jenis informasi yang berbeda mengenai baris tersebut.

#### Tag Tabel Layout:

Beberapa tag pada Table Layout:

- `TableLayout`: Tag pembuka untuk menggunakan `TableLayout`
- `TableRow`: Tag untuk membuat Baris

Contoh:

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="#FFF5F5F5"
android:gravity="center"

tools:context="android.cianjur.developer.net.basicandroid.Ma
inActivity">

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center">

        <Button
            android:id="@+id/tes1"
            android:layout_width="150dp"
            android:layout_height="150dp"
            android:text="Baris 1 Kolom 1" />

        <Button
            android:id="@+id/tes2"
```

```

        android:layout_width="150dp"
        android:layout_height="150dp"
        android:text="Baris 1 Kolom 2" />
</TableRow>
<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center">

    <Button
        android:id="@+id/tes3"
        android:layout_width="160dp"
        android:layout_height="150dp"
        android:text="Baris 2 Kolom 1" />

    <Button
        android:id="@+id/tes4"
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:text="Baris 2 Kolom 2" />
</TableRow>
</TableLayout>

```

Untuk menambahkan baris baru pada Table Layout, kita harus menambahkan tag `<TableRow>`. Selain itu juga, kita dapat membuat menu seperti, menu login menggunakan Table Layout, disini kita akan menggunakan beberapa TextView, EditText dan Button untuk membuat design tersebut:

Kode xmlnya seperti berikut ini;

```

<?xml version="1.0" encoding="utf-8"?>
<TableLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="8dp"
    android:background="#FFF5F5F5"
    android:gravity="center"

    tools:context="android.cianjur.developer.net.basicandroid.MainActivity">

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center">

```

```

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Email: " />

<EditText
    android:id="@+id/editText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textEmailAddress" />
</TableRow>
<TableRow>
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center">

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Password: " />

    <EditText
        android:id="@+id/editText3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPassword" />
</TableRow>
<TableRow>
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center">

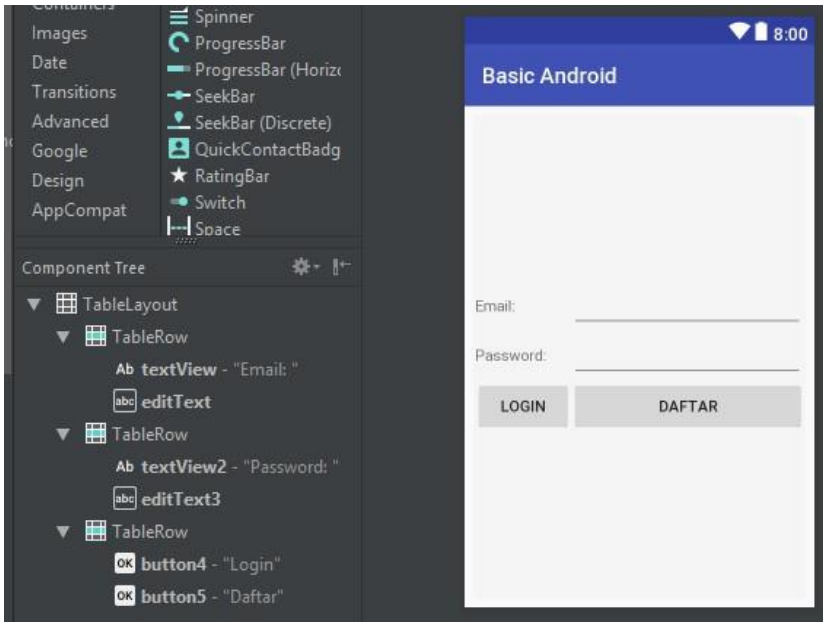
    <Button
        android:id="@+id/button4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Login" />

    <Button
        android:id="@+id/button5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Daftar" />
</TableRow>
</TableLayout>

```

Maka hasilnya akan seperti ini:





Gambar 59. Tabel Layout

#### d. Frame Layout

FrameLayout adalah layout yang paling tepat untuk membuat beberapa UI saling tumpang tindih. Contohnya kita dapat menyisipkan widget (tombol) ke dalam widget lainnya. Agar elemen UI didalam FrameLayout terlihat rapih serta tataletaknya bisa kalian atur sendiri, kalian dapat menggunakan beberapa atribut berikut ini:

- `android:layout_marginLeft`: Untuk mengatur jarak (batasan) sudut kiri elemen UI pada elemen UI lainnya.
- `android:layout_marginRight`: Untuk mengatur jarak (batasan) sudut kanan elemen UI pada elemen UI lainnya.
- `android:layout_marginTop`: Untuk mengatur jarak (batasan) atas elemen UI pada elemen UI lainnya.
- `android:layout_marginBottom`: Untuk mengatur jarak (batasan) bawah elemen UI pada elemen UI lainnya.
- `android:layout_gravity`: Untuk mengatur posisi rata kanan, kiri, atas, bawah, center, center horizontal dan center vertical.

Contoh :

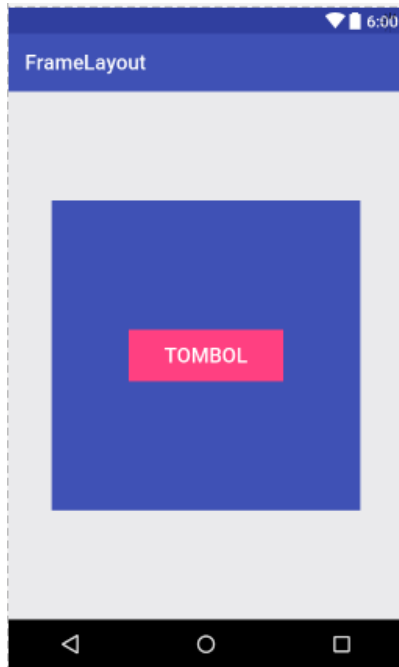
Activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#1113">

    <ImageView
        android:layout_width="300dp"
        android:layout_height="300dp"
        android:layout_gravity="center"
        android:background="@color/colorPrimary" />

    <Button
        android:layout_width="300px"
        android:layout_height="100px"
        android:layout_gravity="center"
        android:text="Tombol"
        android:background="@color/colorAccent"
        android:textColor="#fff"
        android:textSize="20sp" />
</FrameLayout>
```

Hasilnya adalah sebagai berikut:



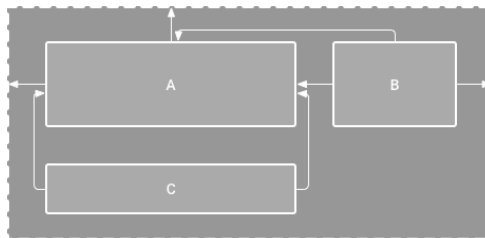
Gambar 60. FrameLayout

### e. Constraint Layout

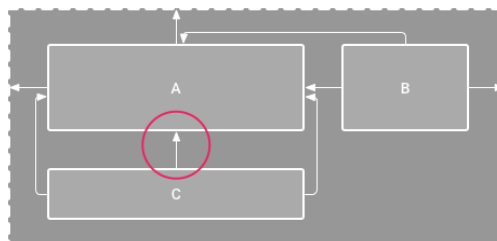
Sebenarnya ConstraintLayout mirip dengan RelativeLayout, karena letak View bergantung pada View lain dalam satu Layout ataupun dengan Parent Layoutnya. Contohnya di RelativeLayout kita bisa meletakkan sebuah View di atas, bawah, atau samping View lain. Kita juga dapat mengatur posisinya berdasarkan Parent Layout menggunakan tag seperti centerVertical, alignParentBottom, dll. tapi ConstraintLayout jauh lebih Fleksibel dan lebih mudah digunakan pada Layout Editor.

Pada ConstraintLayout, setiap View memiliki tali (Constraint) yang menarik tiap sisinya, yang mana tali tersebut bisa kita atur Elastisitas, Margin, dsb. Tali tersebut wajib kita “ikatkan” kepada anchor point atau suatu titik yang dapat berupa sisi dari Parent Layout, View lain, ataupun titik bantu (helper) yang bisa kita buat sendiri.

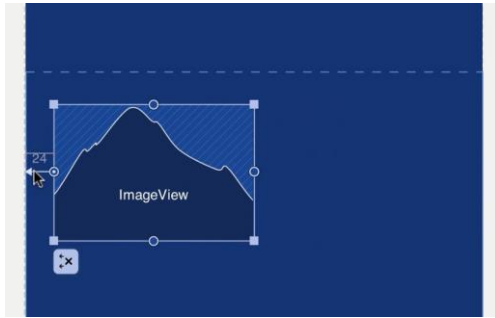
Tanpa constraint tersebut, sebuah Layout secara default akan terletak di posisi [0,0] (sebelah kiri-atas), sehingga sebuah View paling tidak membutuhkan 2 Constraint untuk sisi X dan sisi Y.



Pada layout editor, Layout terlihat normal. Namun jika diperhatikan, View C tidak memiliki Vertical Constraint. Jika di render, View C akan sesuai Horizontal Constraintnya (sisi kiri dan kanan View A), tapi terletak di bagian atas Parent Layout (titik 0).

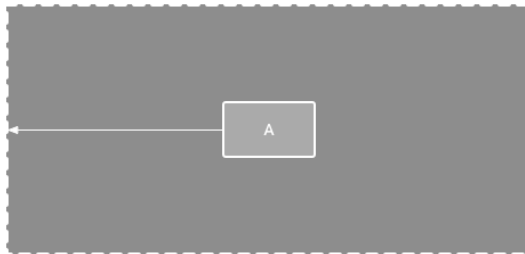


Untuk yang satu ini, View C sudah diberi Vertical Constraint, sehingga ketika di render, Posisinya akan sesuai seperti pada Layout Editor.



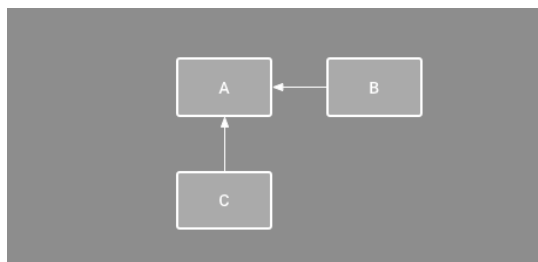
Untuk membuat constraint, setelah menambahkan View kedalam ConstraintLayout, akan muncul 4 titik constraint yang bisa dikaitkan pada Anchor Point, baik Vertical maupun Horizontal. Gambar diatas adalah contoh mengaitkan horizontal constraint bagian kiri kepada Parent Layout. Menggunakan constraint tersebut, kita bisa membuat berbagai macam “Teknik Pengaitan” sesuai kebutuhan, antara lain:

### 1) Parent



Kita bisa mengaitkan constraint pada Parent Layout, kemudian mengatur jaraknya menggunakan Margin. Pada gambar di samping, Bagian kiri View A dikaitkan pada sisi kiri Parent Layout.

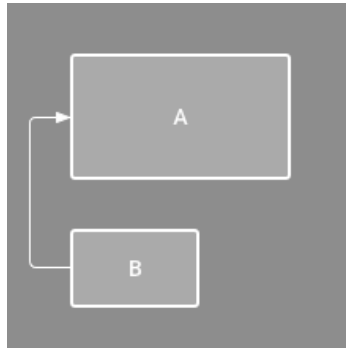
### 2) Posisi View



Kita bisa mengatur urutan posisi dari View yang dibuat. Contoh pada gambar disamping, posisi View B akan selalu berada di sebelah kanan A,

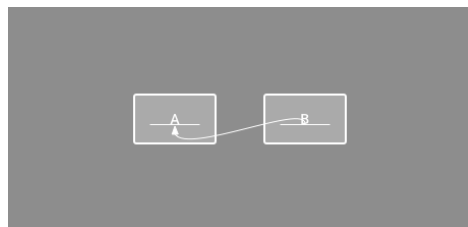
demikian pula View C, akan selalu berada di bawah View A. Namun karena tidak terdapat Alignment (atau constraint lawannya), View B masih bisa berpindah ke atas dan ke bawah, dan View C juga masih bisa berpindah ke Samping kiri dan kanan.

### 3) Alignment



Kita bisa membuat semacam Alignment pada View. Contohnya pada gambar disamping, sisi kiri View B segaris dengan View A. Untuk membuat Align Center, cukup buat dua buah Constraint (kiri dan kanan) kemudian kaitkan dengan masing-masing sisinya. Sebenarnya kita bisa membuat ini secara otomatis dengan cara memilih semua View yang ingin diatur Alignmentnya, kemudian klik menu **Align** pada **Toolbar**, lalu pilih Alignment yang diinginkan.

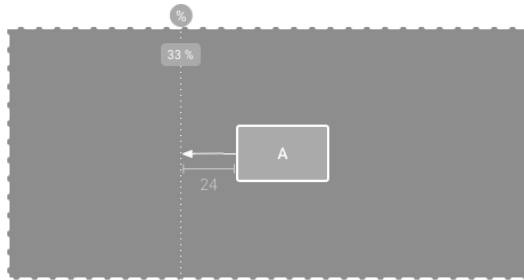
### 4) Baseline Alignment



Misal kita ingin mengatur agar Sebuah TextView segaris dengan TextView lain di Layout yang berbeda, kita bisa menggunakan Baseline Alignment. Untuk membuatnya, pilih TextView yang ingin diatur baselinanya, kemudian klik tombol **Edit Baseline** yang muncul dibawahnya. Setelah itu tinggal tarik saja Baseline nya ke tujuan. Pada gambar di samping, TextView B dibuat segaris dengan TextView A menggunakan Baseline Alignment.

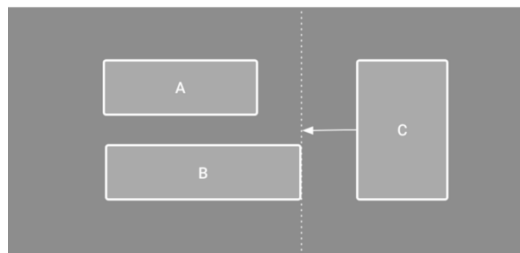
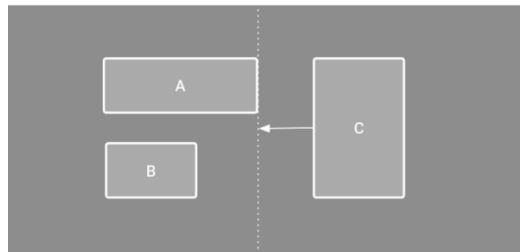
### 5) Guideline

Guideline adalah semacam Garis bayang yang tidak terlihat oleh User. Untuk membuatnya, klik tombol Guidelines pada Toolbar, kemudian pilih Add Vertical Guideline atau Add Horizontal Guideline sesuai Kebutuhan. Kita bisa mengatur posisinya berdasarkan Dimension (dp) ataupun dengan Persentase (berdasarkan sisi layout). Kemudian kita bisa mengaitkan Constraint View kepada Guideline tersebut.



#### 6) Barrier

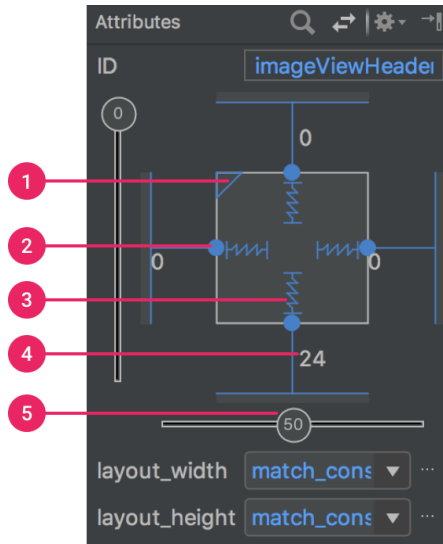
Barrier hampir mirip dengan Guideline, bedanya Barrier tidak berdiri sendiri. Posisi barrier bergantung pada View didalamnya. Contoh kasusnya adalah apabila kita ingin membuat posisi View relatif terhadap beberapa (lebih dari satu) View lain.



Untuk membuat Barrier, klik tombol Guidelines pada Toolbar, kemudian pilih Add Vertical Barrier atau Add Horizontal Barrier sesuai Kebutuhan. Pada

Component Tree, pilih View yang ingin dimasukkan kedalam Barrier tersebut. Kita bisa mengatur opsi dari Barrier seperti `barrierDirection` pada `Window Attribute`. Pada gambar di samping, barrier di atur posisinya menjadi “end” atau “right”, sehingga menyesuaikan sisi paling kanan dari View yang ada di dalamnya.

## 7) Penyesuaian ukuran tampilan



Gambar 61. Penyesuaian ukuran tampilan




Jendela Attributes meliputi kontrol untuk:

- 1 size ratio
- 2 delete constraint,
- 3 height/width mode
- 4 margins, dan
- 5 bias constraint.

Anda dapat menggunakan pegangan sudut untuk mengubah ukuran tampilan, tetapi kode keras ini ukurannya sehingga tampilan tidak akan mengubah ukuran untuk konten atau ukuran layar yang berbeda. Untuk memilih mode penentuan ukuran yang berbeda, klik sebuah tampilan dan buka jendela Attributes di sebelah kanan editor. Di dekat bagian atas jendela Atribut adalah inspektur tampilan, yang mencakup kontrol untuk beberapa atribut layout, seperti yang ditunjukkan pada gambar (ini hanya tersedia untuk tampilan dalam layout constraint). Anda dapat mengubah cara tinggi dan lebar dihitung dengan



mengklik simbol yang ditunjukkan dengan callout 3 pada gambar. Simbol-simbol ini mewakili mode ukuran sebagai berikut (klik simbol untuk beralih di antara pengaturan ini):

- ⇒  Fixed: Anda menetapkan dimensi tertentu dalam kotak teks di bawah atau dengan mengubah ukuran tampilan di editor.
- ⇒  Wrap Content: Tampilan hanya meluas sebanyak yang diperlukan agar sesuai dengan kontennya.
- ⇒  Match Constraints: Tampilan meluas sebanyak mungkin untuk memenuhi Constraints di setiap sisi (setelah memperhitungkan margin tampilan). Namun, Anda dapat memodifikasi perilaku tersebut dengan atribut dan nilai berikut (atribut ini berlaku hanya ketika Anda mengatur lebar tampilan untuk mencocokkan Constraints):
  - ⇒ `layout_constraintWidth_default`
    - `spread`: Perluas tampilan sebanyak mungkin untuk memenuhi Constraints di setiap sisi. Ini adalah perilaku default.
    - `wrap`: Perluas tampilan hanya sebanyak yang diperlukan agar sesuai dengan isinya, tetapi masih memungkinkan tampilan menjadi lebih kecil dari itu jika Constraints membutuhkannya. Jadi perbedaan antara ini dan menggunakan Wrap Content (di atas), adalah pengaturan lebar untuk Wrap Content memaksa lebar untuk selalu sama persis dengan lebar konten; sedangkan menggunakan Match Constraints dengan `layout_constraintWidth_default` set untuk wrap juga memungkinkan tampilan menjadi lebih kecil dari lebar konten.
  - ⇒ `layout_constraintWidth_min`
- ⇒ Ini membutuhkan dimensi dp untuk lebar minimum tampilan.
  - `layout_constraintWidth_max`
- ⇒ Ini membutuhkan dimensi dp untuk lebar maksimum tampilan.
- ⇒ Namun, jika dimensi yang diberikan hanya memiliki satu constraint, maka tampilan diperluas agar sesuai dengan kontennya. Menggunakan mode ini baik pada tinggi maupun lebar juga memungkinkan Anda mengatur rasio ukuran.

## f. List View

ListView adalah user interface pada Android yang menampilkan item-item dari sekumpulan daftar yang tersusun berbaris ke bawah atau ke samping dengan tampilan yang dapat scroll. ListView menampilkan item-item dari suatu Array atau List atau Query Database yang dijadikan data model sebagai item dari ListView.

ListView dikatakan sederhana atau kompleks ditentukan oleh kompleksitas item dari ListView tersebut. Jika masing-masing item dari ListView menampilkan sebuah kata atau kalimat saja, maka ListView tersebut sederhana. Jika masing-masing item dari ListView memiliki isi yang kompleks, maka ListView tersebut adalah ListView kompleks. Pembuatan ListView kompleks dilakukan dengan Custom ListView.

Adapter adalah class yang mengatur item-item pada ListView. Adapter mengatur resource view pada setiap item dari ListView. Resource view pada ListView yang ada pada sebuah tampilan layar sebuah aplikasi memiliki jumlah resource view yang tetap sesuai dengan item yang tampak pada layar. Pada ListView dengan tampilan scroll, resource view akan digunakan secara berulang (*reusable*) dengan mengatur item yang tampak dan yang tersembunyi pada ListView. Adapter juga mengatur data model dari setiap item ListView. Sebuah data model akan diatur menjadi sebuah item dari ListView.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <ListView
        android:id="@+id/id_list"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```

Kemudian kita buat class MainActivity (class ini biasanya sudah dibuat otomatis pada project). Hapus isi class tersebut.

```
import android.os.Bundle;
import android.app.Activity;
import android.widget.ArrayAdapter;
import android.widget.ListView;
```

```

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.layout);

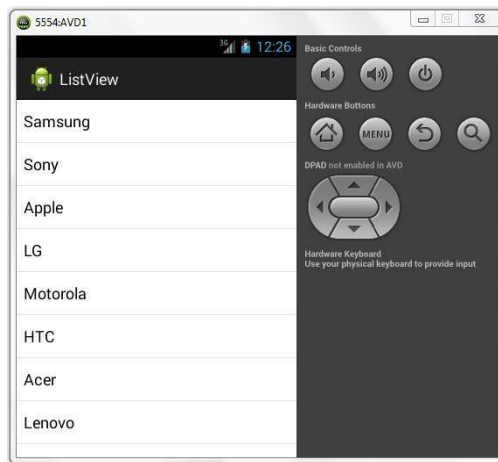
        String[] arrName = {"Samsung", "Sony", "Apple", "LG", "Motorola",
            "HTC", "Acer", "Lenovo", "Oppo", "Nokia", "BlackBerry", "Hua
            wei", "ZTE", "Meizu", "HP", "Asus", "Panasonic"};

        ArrayAdapter adapter = new ArrayAdapter(this,
            android.R.layout.simple_list_item_1, arrName);
        ListView listView = (ListView) findViewById(R.id.id_list);
        listView.setAdapter(adapter);
    }
}

```

Array `String[] arrName` merupakan kumpulan data `String` yang akan ditampilkan pada `ListView`. Array ini dimasukkan ke dalam object dari `ArrayAdapter` yang bernama `adapter`. Adapter ini merupakan adapter sederhana yang hanya menampilkan sebuah `TextView` pada item `ListView`. Code `ListView listView = (ListView) findViewById(R.id.id_list);` membuat object dari class `ListView` dengan menginisiasi object tersebut dengan `ListView` yang kita buat pada file `layout.xml`.

Jalankan project, maka `ListView` tampil seperti berikut ini.



Gambar 62. `TextView` pada item `ListView`

## ⇒ Membuat Custom ListView

Custom ListView merupakan ListView dengan item yang bisa kita buat sesuai selera, misalnya item yang memiliki atribut foto, nama, dan keterangan. Item yang custom seperti itu dapat dibuat dan diatur oleh class Adapter. Contoh Item pada Custom ListView:



Gambar 63. Item pada Custom ListView

Untuk membuat Custom ListView, kita buat terlebih dahulu layout-layoutnya. Kita membutuhkan 2 layout, yaitu layout halaman yang menampilkan ListView dan layout item untuk Adapter. Layout halaman kita buat dengan membuat file **layout.xml** di dalam folder layout dengan isi seperti ini:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <ListView
        android:id="@+id/id_list"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```

Layout item untuk Adapter kita buat dengan membuat file **item.xml** di dalam folder layout dengan isi seperti ini:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    android:padding="8dp" >

    <ImageView
        android:id="@+id/item_icon"
```

```

        android:layout_width="48dp"
        android:layout_height="48dp"
        android:layout_marginRight="8dp" />

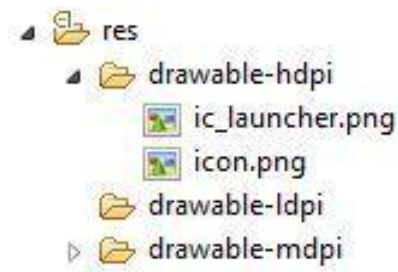
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/item_nama"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Galaxy S4"
        android:textSize="18sp" />

    <TextView
        android:id="@+id/item_keterangan"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Samsung"
        android:textSize="12sp" />
</LinearLayout>
</LinearLayout>

```

Kemudian kita siapkan gambar yang menjadi icon dari item ListView. Saya pakai gambar Android seperti ini dengan ukuran 48 x 48 px (panjang x lebar harus sama). Kita simpan gambar pada folder drawable-hdpi dengan nama icon.png.



Gambar 64. Folder drawable-hdpi

Kemudian kita buat sebuah class data model dari item ListView yang bernama class Phone. Object dari class ini akan menyimpan data-data yang akan ditampilkan pada item dari ListView. Sebuah object dari class Phone akan ditampilkan pada sebuah item dari ListView.

Kita buat class yang bernama Phone.java dengan isi class sebagai berikut.

```
public class Phone {  
  
    private String name;  
    private String brand;  
  
    public Phone(String name, String brand) {  
        setName(name);  
        setBrand(brand); }  
    public String getName() {  
        return name; }  
    public void setName(String name) {  
        this.name = name; }  
    public String getBrand() {  
        return brand; }  
    public void setBrand(String brand) {  
        this.brand = brand; }  
}
```

Kemudian kita buat class Adapter. Class Adapter adalah class yang akan membuat dan mengatur item-item dari ListView. Class ini mengkonversi data dari object Phone menjadi item dari ListView. Kita buat class yang bernama ListAdapter.java dengan isi class sebagai berikut:

```
import java.util.ArrayList;  
  
import android.app.Activity;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ArrayAdapter;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
public class ListAdapter extends ArrayAdapter {  
  
    private ArrayList list;  
    private Activity act;  
    public ListAdapter(Activity context, ArrayList objects)  
{  
        super(context, R.layout.item, objects);  
        this.list = objects;  
}
```

```

        this.act = context;    }

    static class ViewHolder {
        protected ImageView icon;
        protected TextView nama;
        protected TextView keterangan;    }

    @Override
    public View getView(int position, View convertView,
    ViewGroup parent) {
        View view = convertView;
        if (view == null) {
            LayoutInflater inflater = act.getLayoutInflater();
            view = inflater.inflate(R.layout.item, null);

            ViewHolder holder = new ViewHolder();
            holder.icon = (ImageView)
            view.findViewById(R.id.item_icon);
            holder.nama = (TextView)
            view.findViewById(R.id.item_nama);
            holder.keterangan = (TextView)
            view.findViewById(R.id.item_keterangan);
            view.setTag(holder);
        }
        ViewHolder holder = (ViewHolder) view.getTag();
        Phone phone = list.get(position);
        holder.icon.setImageResource(R.drawable.icon);
        holder.nama.setText(phone.getName());
        holder.keterangan.setText(phone.getBrand());
        return view;    }
}

```

Kemudian kita buat class **MainActivity** (class ini biasanya sudah dibuat otomatis pada project) sebagai class controller yang mengelola pembuatan Custom ListView. Buat class MainActivity.java dengan isi sebagai berikut:

```

import java.util.ArrayList;

import android.os.Bundle;
import android.app.Activity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ListView;
import android.widget.Toast;

public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.layout);
    }
}

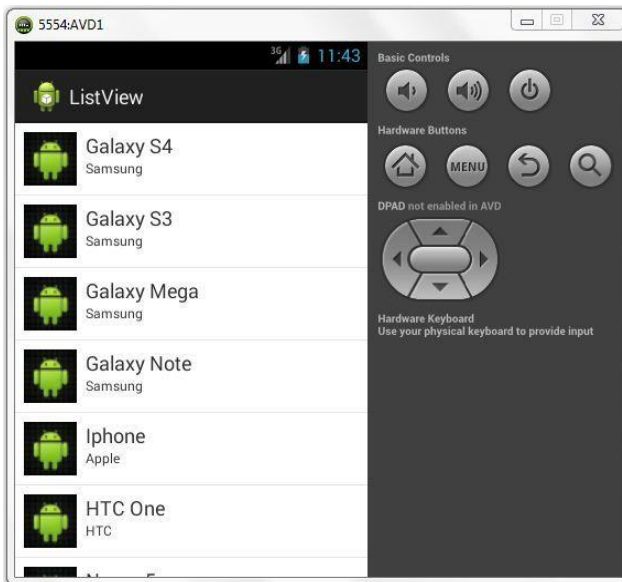
```

```

        ArrayList list = new ArrayList();
list.add(new Phone("Galaxy S4", "Samsung"));
list.add(new Phone("Galaxy S3", "Samsung"));
list.add(new Phone("Galaxy Mega", "Samsung"));
list.add(new Phone("Galaxy Note", "Samsung"));
list.add(new Phone("Iphone", "Apple"));
list.add(new Phone("HTC One", "HTC"));
list.add(new Phone("Nexus 5", "LG"));
list.add(new Phone("Nexus 4", "LG"));
list.add(new Phone("LG G2", "LG"));
list.add(new Phone("Moto x", "Motorola"));
ListAdapter adapter = new ListAdapter(this, list);
ListView listView = (ListView) findViewById(R.id.id_list);
listView.setAdapter(adapter);    }
}

```

Pada class tersebut kita membuat ArrayList yang berisi object-object dari Phone. ArrayList ini menjadi kumpulan data yang akan ditampilkan pada ListView. Di bawahnya ada object dari class ListAdapter yang bernama adapter. Object ini yang akan dimasukkan pada object dari ListView pada fungsi setAdapter(adapter). Jika project dijalankan, maka ListView akan tampil seperti ini.



Gambar 65. ArrayListview





## YANG HARUS ANDA KETAHUI

Untuk praktik ini Anda harus bisa:

- 1) Cara membuat aplikasi Hello World dengan Android Studio.
- 2) Menjalankan aplikasi di emulator atau perangkat.
- 3) Membuat dan menggunakan resource.

## YANG AKAN ANDA BUTUHKAN

Untuk materi dan praktik ini, Anda akan membutuhkan:

- 1) Komputer Mac, Windows, atau Linux. Tool Android Studio untuk persyaratan sistem terbaru.
- 2) Akses internet atau cara alternatif untuk memuat Android Studio dan pemasangan Java terbaru ke komputer Anda.
- 3) Smartphone sebagai perangkat untuk pengujian secara nyata aplikasi yang telah dibuat.

## YANG AKAN ANDA PELAJARI

Anda akan belajar:

- 1) Cara membuat antarmuka pengguna interaktif dalam Layout Editor, dalam XML, dan secara terprogram.
- 2) Menggunakan editor layout di Android Studio
- 3) Memposisikan tampilan di jenis-jenis layout

## YANG HARUS ANDA LAKUKAN

- 1) Memasang lingkungan development Android Studio.
- 2) Membuat aplikasi dan menggunakan elemen antarmuka pengguna seperti tombol dalam Layout Editor.
- 3) Mengedit layout aplikasi dalam XML.
- 4) Menambahkan tombol ke aplikasi. Menggunakan sumber daya string untuk label.
- 5) Mengimplementasikan metode handler klik untuk tombol agar bisa menampilkan pesan pada layar saat pengguna mengekliknya.

## RINGKASAN APLIKASI

Aplikasi "Hello Toast" akan berisi dua tombol dan satu tampilan teks. Saat Anda mengeklik tombol pertama, pesan pendek atau toast akan ditampilkan pada layar. Mengeklik tombol kedua akan meningkatkan penghitung klik; jumlah total klik mouse akan ditampilkan pada tampilan teks.


Latihan 3.1. Membuat project "Hello Toast" yang baru

Dalam praktik ini Anda akan mendesain dan mengimplementasikan project untuk aplikasi "Hello Toast".

### Latihan 3.1. Membuat proyek "Hello Toast":

⇒ Mulailah Android Studio dan buat proyek baru dengan parameter berikut:

Atribut	Nilai
Application Name	Hello Toast
Company Name	com.fkti.unmul or your own domain
Phone/Tablet Minimum SDK	API15: Android 4.0.3 IceCreamSandwich
Template	Empty Activity
Generate Layout file box	Dicentang
Backwards Compatibility box	Dicentang

⇒ Pilih **Run > Run app** atau klik **ikon Run**  pada toolbar untuk membangun dan mengeksekusi aplikasi di emulator atau perangkat.

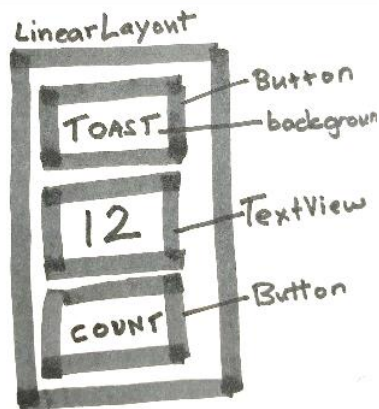
### Latihan 3.2. Menambahkan tampilan ke "Hello Toast"

Dalam tugas ini, Anda akan membuat dan mengonfigurasi antarmuka pengguna untuk aplikasi "Hello Toast" dengan mengatur komponen UI tampilan pada layar.

**Mengapa:** Setiap aplikasi harus dimulai dengan pengalaman pengguna, bahkan jika implementasi awalnya sangat dasar. Tampilan yang digunakan untuk Hello Toast adalah:

- TextView - Tampilan yang menampilkan teks.
- Tombol - Tombol dengan label yang biasanya dikaitkan dengan handler klik.
- LinearLayout - Tampilan yang berfungsi sebagai kontainer untuk mengatur tampilan lainnya. Jenis tampilan ini memperluas kelas ViewGroup dan juga disebut grup tampilan. LinearLayout adalah grup tampilan dasar yang mengatur kumpulan tampilan dalam baris horizontal atau vertikal.

Ini adalah sketsa kasar UI yang akan Anda bangun dalam latihan ini. Sketsa UI sederhana akan sangat berguna untuk memutuskan tampilan mana yang akan digunakan dan bagaimana mengaturnya, terutama saat layout menjadi lebih rumit.



Gambar 66. Sketsa UI sederhana

### Latihan 3.3. Menjelajahi layout editor:

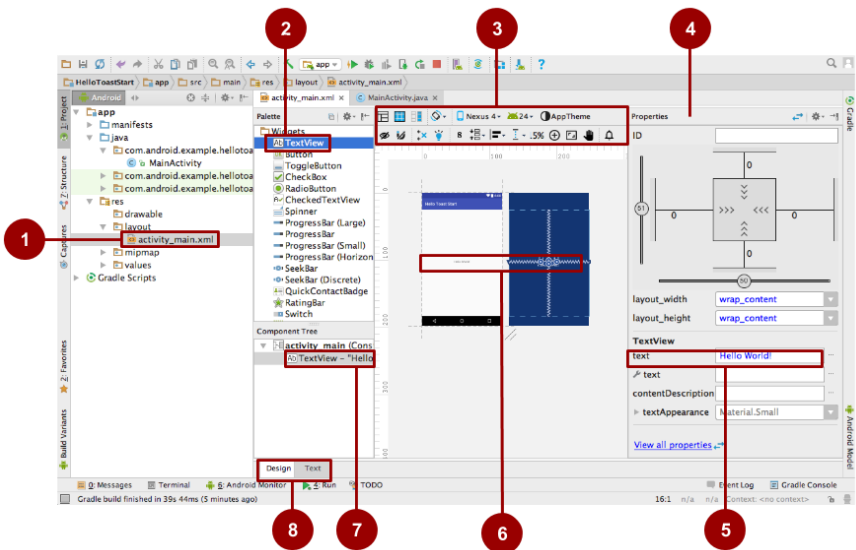
Gunakan Layout Editor untuk membuat layout elemen antarmuka pengguna dan mempratinjau aplikasi menggunakan perangkat yang berbeda serta tema, resolusi, dan orientasi aplikasi.

Lihat tangkapan layar bawah untuk mencocokkan

1. Dalam folder **app > res > layout**, buka file **activity\_main.xml**(1).

Layar Android Studio harus terlihat mirip dengan tangkapan layar di bawah ini. Jika Anda melihat kode XML untuk layout UI, klik tab **Design** di bawah Component Tree (8).

Menggunakan tangkapan layar beranotasi di bawah ini sebagai panduan, jelajahi Layout Editor.



Gambar 67. Layout Editor

2. Temukan berbagai cara yang berbeda untuk merepresentasikan elemen UI string "Hello World", atau TextView.
  - o Dalam Palette elemen UI (2) developer dapat membuat tampilan teks dengan menyeretnya ke dalam panel design.
  - o Secara visual, panel Design (6).
  - o Dalam **Component Tree** (7), sebagai komponen dalam hierarki tampilan UI yang disebut Hierarki Tampilan. Yakni, tampilan yang

diatur ke dalam hierarki pohon parent dan anak, di mana anak mewarisi properti parentnya.

- Dalam **panel Properties (4)**, sebagai daftar propertinya, di mana "Hello Toast" adalah nilai dari properti teks TextView (5).
3. Gunakan selector di atas perangkat virtual (3) untuk melakukan hal berikut ini:
    - Mengubah tema aplikasi.
    - Mengubah rotasi lanskap.
    - Menggunakan versi berbeda dari SDK.
    - Melakukan pratinjau pada gaya penulisan kanan ke kiri.
    - Memilih item UI dan masuk ke kode sumbernya.

Menggunakan keterangan alat pada ikon untuk membantu Anda menemukan fungsinya.

4. Berpindah di antara tab **Design** dan **Text** (8). Beberapa perubahan UI hanya bisa dibuat dalam kode, dan beberapa lebih cepat dilakukan pada perangkat virtual.
5. Setelah selesai, batalkan perubahan (untuk perubahan UI, gunakan **Edit** > **Undo** atau pintasan keyboard untuk sistem operasi).

**Catatan:** Jika Anda mendapatkan kesalahan tentang Tema Aplikasi yang hilang, coba **File** > **Invalidate Caches** / **Restart** atau pilih tema yang tidak menghasilkan kesalahan.

Latihan 3.2. Mengubah grup tampilan ke LinearLayout:

Root hierarki tampilan adalah grup tampilan, seperti yang tersirat dari namanya, adalah tampilan yang berisi tampilan lain.

Secara default, Template Kosong menggunakan grup tampilan RelativeLayout. Layout ini menawarkan banyak fleksibilitas dalam menempatkan tampilan di grup tampilan. Layout linear vertikal adalah salah satu layout yang paling umum. Layout ini sederhana, cepat, dan selalu merupakan titik permulaan yang baik. Ubah grup tampilan ke layout linear vertikal seperti berikut:

1. Dalam panel **Component Tree** (7 dalam tangkapan layar sebelumnya), temukan tampilan atas atau root di bawah Layar Perangkat, yang seharusnya RelativeLayout.
2. Klik tab **Text** (8) untuk berpindah ke tampilan kode layout.
3. Dalam baris kedua kode, cari **RelativeLayout** dan ganti ke **LinearLayout**.

4. Pastikan tag penutup di bagian akhir kode sudah diubah menjadi `</LinearLayout>`. Jika belum berubah secara otomatis, ubah secara manual.
5. `android:layout_height` ditetapkan sebagai bagian dari template. Orientasi layout default adalah baris horizontal. Untuk mengubah layout menjadi vertikal, tambahkan kode berikut di bawah `android:layout_height`.  
`android:orientation="vertical"`
6. Dari bilah menu, pilih: **Code > Reformat Code...**  
Pesan "no lines changed: code is already properly formatted" mungkin muncul.
7. Jalankan kode untuk memastikan bahwa kode masih berfungsi.
8. Kembali ke **Design**.
9. Verifikasi dalam panel **Component Tree** bahwa elemen teratas sekarang adalah `LinearLayout` dan atribut orientasinya disetel ke "vertikal".

#### Kode Solusi:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"

  android:paddingBottom="@dimen/activity_vertical_margin"
  android:paddingLeft="@dimen/activity_horizontal_margin"
  android:paddingRight="@dimen/activity_horizontal_margin"
  android:paddingTop="@dimen/activity_vertical_margin"

  tools:context="hellotoast.android.example.com.hellotoast.MainActivity">

  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!" />

</LinearLayout>
```

### Latihan 3.4. Menambahkan ke Linear Layout dalam Layout Editor

Pada tugas ini, Anda akan menghapus TextView saat ini (untuk latihan), dan menambahkan TextView baru dan dua tombol ke LinearLayout seperti yang ditampilkan dalam sketsa UI untuk tugas ini. Lihat diagram UI di atas, jika perlu.

#### Tambahkan Elemen UI

1. Klik tab **Design** (8) untuk menunjukkan layout perangkat virtual.
2. Klik TextView yang nilai teksnya "Hello World" dalam layout perangkat virtual atau panel **Component Tree** (7).
3. Tekan tombol DELETE untuk menghapus TextView.
4. Dari panel **Palette** (2), seret dan jatuhkan elemen Tombol, TextView, dan elemen Tombol lainnya, dalam urutan tersebut, satu di bawah yang lainnya ke dalam layout perangkat virtual.

#### Sesuaikan Elemen UI

1. Untuk mengidentifikasi setiap tampilan secara unik di dalam activity, setiap tampilan memerlukan id unik. Dan agar ada gunanya, tombol memerlukan label dan tampilan teks perlu menunjukkan beberapa teks. Klik dua kali setiap elemen dalam Layout Manager untuk melihat propertinya dan mengubah string **text** dan **id** seperti berikut:

Elemen	Teks	ID
Top button	Toast	button_toast
Text view	0	show_count
Bottom button	Count	button_count

1. Jalankan aplikasi Anda.

### Latihan 3.5. Mengedit Layout "Hello Toast" dalam XML

Dalam praktik ini Anda akan mengedit kode XML untuk layout UI aplikasi Hello Toast. Anda juga akan mengedit properti tampilan yang sudah Anda buat.

*Mengapa:\** Meskipun Layout Editor adalah alat yang hebat, beberapa perubahan lebih mudah dibuat langsung di kode sumber XML. Menggunakan LayoutEditor grafis atau mengedit file XML secara langsung adalah pilihan personal.

1. Buka **res/layout/activity\_main.xml** dalam mode Text.
2. Dalam bilah menu pilih **Code > Reformat Code**
3. Periksa kode yang dibuat oleh Layout Editor.

Perhatikan bahwa kode Anda mungkin bukan kecocokan eksak, tergantung pada perubahan yang Anda buat dalam Layout Editor. Gunakan solusi sederhana sebagai panduan.

Memeriksa properti LinearLayout

LinearLayout diperlukan untuk memiliki properti ini:

- `layout_width`
- `layout_height`
- `orientation`

`layout_width` dan `layout_height` dapat mengambil satu dari tiga nilai ini:

- Atribut **match\_parent** meluaskan tampilan untuk mengisi parentnya dengan lebar atau tinggi. Saat LinearLayout adalah tampilan rootnya, layout diperluas hingga ke ukuran tampilan parent.
- Atribut **wrap\_content** menyusutkan dimensi tampilan, hanya cukup besar untuk membungkus kontennya. (Jika tidak ada konten, tampilan tidak akan terlihat.)
- Gunakan **angka tetap dp** (piksel yang tidak tergantung perangkat) untuk menentukan ukuran yang tetap, yang disesuaikan untuk ukuran layar perangkat. Misalnya, "16dp" berarti 16 piksel yang tidak tergantung perangkat.

Orientasinya bisa berupa:

- **horizontal:** tampilan diatur dari kiri ke kanan.
- **vertikal:** tampilan diatur dari atas ke bawah.



Ubah LinearLayout "Hello Toast" seperti berikut:

Properti	Nilai
layout_width	match_parent (untuk mengisi layar)
layout_height	match_parent (untuk mengisi layar)
orientasi	vertikal

Membuat resource string:

Daripada melakukan hard-code string ke dalam kode XML, sebaiknya gunakan resource string, yang mewakili string tersebut.

**Mengapa:** String yang ada di file yang terpisah lebih mudah dikelola, terutama jika Anda menggunakan string ini lebih dari sekali. Selain itu, resource string wajib ada untuk menerjemahkan dan melokalisasi aplikasi karena Anda akan membuat file resource string untuk setiap bahasa.

1. Letakkan kursor pada kata "Toast".
2. Tekan **Alt-Enter (Option-Enter di Mac)**.
3. Pilih **Extract string resources**.
4. Setel **nama Resource** ke `button_label_toast` dan klik **OK**. (Jika Anda membuat kesalahan, batalkan perubahan dengan Ctrl-Z.)

Proses ini membuat resource string dalam file `values/res/string.xml`, dan string dalam kode Anda digantikan dengan referensi ke resource string tersebut.

```
@string/button_label_toast
```

5. Ekstrak dan beri nama string yang tersisa dari tampilan sebagai berikut:

Tampilan	Nilai Resource / String	Nama resource
Button	Hello Toast!	<code>button_label_toast</code>
TextView	0	<code>count_initial_value</code>
Button	Count	<code>button_label_count</code>

6. Dalam tampilan Proyek, buka **values/strings.xml** untuk menemukan string Anda. Sekarang Anda bisa mengedit semua string di satu tempat.

Mengubah ukuran:

Sama dengan string, sebaiknya ekstrak dimensi tampilan dari file XML layout utama ke dalam resource dimensi yang terletak di dalam file.

**Mengapa:** Hal ini akan memudahkan pengelolaan dimensi, terutama jika Anda perlu menyesuaikan layout untuk resolusi perangkat yang berbeda. Ini juga akan memudahkan pengaturan ukuran yang konsisten, dan mengubah ukuran beberapa objek dengan mengubah satu properti.

Lakukan hal berikut:

1. Lihat file resource `dimens.xml`. Harus ada nilai untuk margin layar default yang ditetapkan. Untuk dimensi tampilan, sebaiknya jangan gunakan nilai-nilai hard-code, karena nilai tersebut akan menghalangi tampilan untuk menyesuaikan ukuran dengan layar.
2. Kalau perlu, ubah `layout_width` semua elemen di dalam `LinearLayout` ke `"match_parent"`.

Jika Anda ingin menggunakan Layout Editor grafis, klik tab **Design**, pilih setiap elemen dalam panel **Component Tree** dan ubah properti `layout:width` dalam panel **Properties**. Jika Anda ingin langsung mengedit file XML, klik tab **Text**, ubah `android:layout_width` untuk tombol pertama, `TextView`, dan Tombol terakhir.

3. Kalau perlu, ubah `layout_height` semua elemen di dalam `LinearLayout` ke `"wrap_content"`.

Menyetel warna dan latar belakang:

Gaya dan warna adalah properti tambahan yang dapat diekstrak ke dalam properti. Semua tampilan dapat memiliki latar belakang yang bisa berupa warna atau gambar.

Lakukan eksperimen dengan perubahan berikut:

1. Ubah ukuran teks `TextView` `show_count`. `"sp"` adalah singkatan dari piksel yang tidak tergantung pada skala, dan seperti `dp`, ini adalah unit yang skalanya menyesuaikan dengan kepadatan layar dan preferensi ukuran font pengguna. Sebaiknya Anda menggunakan unit ini saat menetapkan ukuran font, agar disesuaikan dengan kepadatan layar dan preferensi pengguna.
2. `android:textSize="160sp"`
3. Ekstrak ukuran teks `TextView` sebagai resource dimensi bernama `count_text_size`, sebagai berikut:

⇒ Klik tab **Text** untuk menunjukkan kode XML, jika belum.

⇒ Letakkan kursor pada `"160sp"`.

⇒ Tekan **Alt-Enter** (**Option-Enter** di Mac).

⇒ Klik **Extract dimension resource**.

- ⇒ Setel **Resource name** ke `count_text_size` dan klik **OK**. (Jika Anda membuat kesalahan, batalkan perubahan dengan Ctrl-Z).
  - ⇒ Dalam tampilan Proyek, buka **values/dimens.xml** untuk menemukan dimensi Anda. File `dimens.xml` file berlaku untuk semua perangkat. File `dimens.xml` file untuk `w820dp` hanya berlaku untuk perangkat yang lebih lebar dari `820dp`.
4. Ubah `textStyle` dari `TextView` `show_count` ke cetak tebal.
  5. `android:textStyle="bold"`
  6. Ubah warna teks dalam tampilan teks `show_count` ke warna utama tema. Lihat file resource `colors.xml` untuk melihat bagaimana file ditetapkan.

`colorPrimary` adalah salah satu dari warna dasar tema yang telah ditetapkan dan digunakan untuk bilah aplikasi. Misalnya, dalam aplikasi produksi Anda bisa menyesuaikan ini agar cocok dengan merek Anda. Menggunakan warna dasar untuk elemen UI membuat UI yang seragam.

```
android:textColor="@color/colorPrimary"
```

7. Ubah warna kedua tombol ke warna utama tema.  
`android:background="@color/colorPrimary"`
8. Ubah warna teks dalam kedua tombol ke warna putih. Putih adalah salah satu warna yang disediakan sebagai Resource Platform Android.

```
android:textColor="@android:color/white"
```

9. Tambahkan warna latar belakang ke `TextView`.

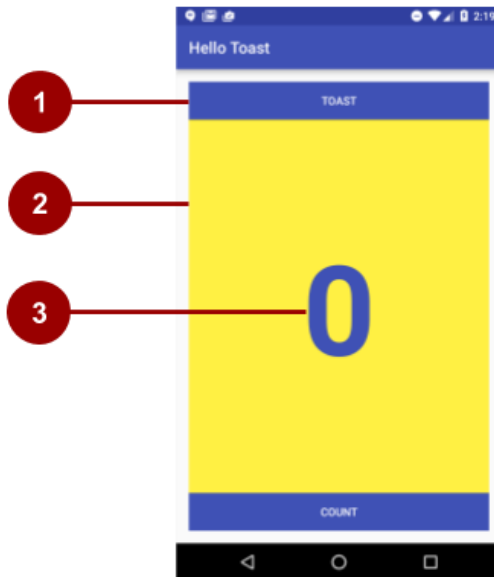
```
android:background="#FFFF00"
```

10. Dalam Layout Editor (tab Text), arahkan kursor ke warna ini dan tekan **Alt-Enter** (**Option-Enter** di Mac).
11. Pilih **Choose color**, yang akan memunculkan pemilih warna, pilih warna yang Anda sukai, atau pilih warna kuning yang sudah dipakai sekarang, lalu klik **Choose**.
12. Buka **values/colors.xml**. Perhatikan bahwa `colorPrimary` yang Anda gunakan sebelumnya ditetapkan di sini.
13. Menggunakan warna dalam **values/colors.xml** sebagai contoh, tambahkan resource bernama `myBackgroundColor` untuk warna latar belakang, lalu gunakan untuk menyetel latar belakang tampilan teks.

```
<color name="myBackgroundColor">#FFF043</color>
```

### Latihan 3.6. Gravitasi dan berat:

Menentukan properti gravitasi dan berat memberi Anda kontrol tambahan untuk mengatur tampilan dan konten dalam layout linear.



1. Android : atribut `layout_gravity` menentukan bagaimana tampilan diratakan dalam *Tampilan parent*. Karena lebar tampilan cocok dengan parentnya, menyetelnya secara eksplisit tidak diperlukan. Anda dapat memusatkan tampilan sempit secara horizontal dalam parentnya:  
`android:layout_gravity="center_horizontal"`
2. `android:atribut layout_weight` menunjukkan berapa banyak ruang ekstra dalam `LinearLayout` yang akan dialokasikan ke tampilan yang sudah menyetel parameter ini. Jika hanya satu tampilan yang memiliki atribut ini, tampilan itu akan mendapatkan semua ruang layar ekstra. Untuk beberapa tampilan, ruangan dipro-rata. Misalnya, jika kedua tombol memiliki berat 1 dan tampilan teks memiliki berat 2, sehingga totalnya 4, masing-masing tombol mendapatkan  $\frac{1}{4}$  ruang, dan tampilan teks mendapatkan setengah.
3. `Android:atribut gravity` menentukan perataan *konten Tampilan di dalam Tampilan itu sendiri*. Kounter dipusatkan dalam tampilannya dengan: `android:gravity="center"`

Lakukan hal berikut:

1. Pusatkan teks dalam TextView `show_count` secara horizontal dan vertikal: `android:gravity="center"`
2. Sesuaikan ukuran TextView `show_count` dengan ukuran layar:
3. `android:layout_weight="2"`

### Contoh Solusi: strings.xml

```
<resources>
    <string name="app_name">Hello Toast</string>
    <string name="button_label_count">Count</string>
    <string name="button_label_toast">Toast</string>
    <string name="count_initial_value">0</string>
</resources>
```

### Contoh Solusi: dims.xml

```
<resources>
    <!-- Default screen margins, per the Android Design
    guidelines. -->
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
    <dimen name="count_text_size">160sp</dimen>
</resources>
```

### Contoh Solusi: colors.xml

```
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
    <color name="myBackgroundColor">#FFF043</color>
</resources>
```

### Contoh Solusi: activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"

    tools:context="hellotoast.android.example.com.hellotoast.MainActivity">
```

```

<Button
    android:id="@+id/button_toast"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/button_label_toast"
    android:background="@color/colorPrimary"
    android:textColor="@android:color/white" />

<TextView
    android:id="@+id/show_count"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="@string/count_initial_value"
    android:textSize="@dimen/count_text_size"
    android:textStyle="bold"
    android:textColor="@color/colorPrimary"
    android:background="@color/myBackgroundColor"
    android:layout_weight="2" />

<Button
    android:id="@+id/button_count"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/button_label_count"
    android:background="@color/colorPrimary"
    android:textColor="@android:color/white" />
</LinearLayout>

```

## Tantangan penyusunan kode

- Membuat proyek baru dengan 5 tampilan. Buat agar satu tampilan menggunakan setengah atas layar, dan tampilan 4 lainnya berbagi setengah bawah layar. Hanya gunakan LinearLayout, gravitasi, dan berat untuk melakukannya.
- Gunakan gambar sebagai latar belakang aplikasi Hello Toast. Tambahkan gambar ke folder drawable, lalu setel sebagai latar belakang tampilan root. Untuk keterangan lebih lanjut mengenai drawable.

### Latihan 3.3. Menambahkan handler onclick untuk tombol

Pada tugas ini Anda akan menambahkan metode ke MainActivity yang mengeksekusi saat pengguna mengeklik setiap tombol.

*\*Mengapa:* Aplikasi interaktif harus merespons ke masukan pengguna.

Untuk menghubungkan tindakan pengguna dalam tampilan ke kode aplikasi, Anda perlu melakukan dua hal:

- Tulis metode yang melakukan tindakan spesifik saat pengguna mengeklik tombol di layar.
- Kaitkan metode ini ke tampilan, agar metode ini dipanggil saat pengguna berinteraksi dengan tampilan.

#### Menambahkan properti onClick ke tombol

Handler klik adalah metode yang dipanggil saat pengguna mengeklik elemen antarmuka pengguna. Dalam Android, Anda bisa menetapkan nama metode handler klik untuk setiap tampilan dalam file layout XML dengan properti `android:onClick`.

1. Buka `res/layout/activity_main.xml`.
2. Tambahkan properti berikut ini ke tombol `button_toast`.  
`android:onClick="showToast"`
3. Tambahkan atribut berikut ke tombol `button_count`:  
`android:onClick="countUp"`
4. Dalam `activity_main.xml`, letakkan kursor mouse Anda di atas setiap nama metode ini.
5. Tekan **Alt-Enter** (**Option-Enter** di Mac), dan pilih **Create onClick event handler**.
6. Pilih **MainActivity** dan klik **OK**.

Tindakan ini akan membuat stub metode placeholder untuk metode `onClick` dalam `MainActivity.java`.

**Catatan:** Anda juga bisa menambahkan handler klik ke tampilan secara terprogram, yang akan Anda lakukan pada praktik berikutnya. Apakah Anda menambahkan handler klik di XML atau secara terprogram sepenuhnya adalah pilihan pribadi; meskipun, ada situasi saat Anda hanya bisa melakukannya secara terprogram.

#### `MainActivity.java` Solusi

```
package hellotoast.android.example.com.hellotoast;
```

```

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void countUp(View view) {
        // What happens when user clicks on the button_count
        Button goes here.
    }

    public void showToast(View view) {
        // What happens when user clicks on the
        button_toast Button goes here.
    }
}

```

### Latihan 3. 7. Menampilkan toast saat tombol Toast diklik

Toast menyediakan feedback sederhana tentang operasi dalam popup kecil. Popup ini hanya akan memenuhi ruang yang diperlukan untuk pesan dan aktivitas saat ini tetap terlihat dan interaktif. Toast menyediakan cara lain untuk menguji interaktivitas aplikasi Anda.

Dalam `MainActivity.java`, tambahkan kode ke metode `showToast()` untuk menampilkan pesan toast.

Untuk membuat instance [Toast](#), panggil metode pabrik `makeText()` pada kelas `Toast`, yang menyediakan konteks (lihat di bawah ini), pesan yang akan ditampilkan, dan durasi tampilan. Anda menampilkan toast yang memanggil `show()`. Ini adalah pola umum di mana Anda bisa menggunakan kembali kode yang akan ditulis.

#### 1. Dapatkan konteks aplikasi.

Menampilkan pesan Toast memerlukan konteks. [Konteks](#) aplikasi berisi informasi global tentang lingkungan aplikasi. Karena toast ditampilkan di atas UI yang terlihat, sistem memerlukan informasi tentang aktivitas saat ini.

```
Context context = getApplicationContext()
```



Saat sudah berada dalam konteks activity yang konteksnya diperlukan, Anda juga bisa menggunakan `this` sebagai pintasan ke konteks.

2. Panjang string toast bisa panjang atau pendek, dan Anda menetapkan yang mana dengan konstanta Toast.
  - `Toast.LENGTH_LONG`
  - `Toast.LENGTH_SHORT`

Panjang sebenarnya adalah 3,5d untuk toast panjang dan 2d untuk toast pendek. Nilai ditetapkan dalam kode sumber Android.

3. Buat instance kelas Toast dengan konteks, pesan, dan durasi.
  - Konteksnya adalah konteks aplikasi yang kita dapatkan sebelumnya.
  - Pesannya adalah string yang ingin Anda tampilkan.
  - Durasinya adalah salah satu konstanta yang sudah ditetapkan sebelumnya `Toast.LENGTH_LONG` atau `Toast.LENGTH_SHORT`.
  - `Toast toast = Toast.makeText(context, "Hello Toast", Toast.LENGTH_LONG);`
4. Ekstrak string "Hello Toast" ke dalam resource string dan beri nama `toast_message`.
  - ⇒ Letakkan kursor pada string "Hello Toast".
  - ⇒ Tekan **Alt-Enter** (**Option-Enter** di Mac).
  - ⇒ Pilih **Extract string resources**.
  - ⇒ Setel **Resource name** ke `toast_message` dan klik **OK**.

Tindakan ini akan menyimpan "Hello World" sebagai nama resource string `toast_message` dalam file resource string `res/values/string.xml`. Parameter string dalam panggilan metode Anda akan digantikan dengan referensi ke resource.

- ⇒ `R`. mengidentifikasi parameter sebagai resource.
- ⇒ string mereferensikan nama file XML tempat resource ditetapkan.
- ⇒ `toast_message` adalah nama resource.

```
Toast toast = Toast.makeText(context,
R.string.toast_message, duration);
```

5. Tampilkan toast.
6. `toast.show();`
7. Jalankan aplikasi dan verifikasi bahwa toast ditampilkan saat tombol **Toast** diketuk.

**Solusi:**

```

/*
 * When the TOAST button is clicked, show a toast.
 *
 * @param view The view that triggers this onClick handler.
 *             Since a toast always shows on the top, view is
not used.
 * */
public void showToast(View view) {
    // Create a toast show it.
    Toast toast = Toast.makeText(this,
R.string.toast_message, Toast.LENGTH_LONG);
    toast.show();
}

```

## Meningkatkan hitungan dalam tampilan teks saat tombol Count di klik

Untuk menampilkan hitungan saat ini dalam tampilan teks:

- Awasi hitungan saat berubah.
- Kirimkan hitungan terbaru ke tampilan teks untuk menampilkannya di antarmuka pengguna.

Implementasikan ini dengan cara berikut:

- ⇒ Dalam MainActivity.java, tambahkan variabel anggota privat `mCount` untuk melacak hitungan dan memulainya dari 0.
- ⇒ Dalam MainActivity.java, tambahkan variabel anggota privat `mShowCount` untuk mendapatkan referensi dari `TextView` `show_count`.
- ⇒ Dalam metode `countUp()`, tingkatkan nilai variabel hitungan setiap kali tombol diklik.
- ⇒ Dapatkan referensi ke tampilan teks menggunakan id yang Anda setel dalam file layout.

Tampilan, seperti string dan dimensi, adalah resource yang dapat memiliki id. Panggilan `findViewById()` mengambil id tampilan sebagai parameternya dan mengembalikan tampilan. Karena metode ini mengembalikan Tampilan, Anda harus menampilkan hasilnya ke jenis tampilan yang diharapkan, dalam kasus ini (`TextView`).

Untuk mendapatkan resource ini sekali saja, gunakan variabel anggota dan setel di dalam `onCreate()`.

```
mShowCount = (TextView) findViewById(R.id.show_count)
```

1. Setel teks dalam tampilan teks ke nilai variabel `count`.

```
if (mShowCount != null)
    mShowCount.setText(Integer.toString(mCount));
```

2. Jalankan aplikasi untuk memverifikasi bahwa hitungan meningkat saat tombol **Count** ditekan.

### Solusi:

Definisi kelas dan menginisialisasi variabel count:

```
public class MainActivity extends AppCompatActivity {
    private int mCount = 0;
    private TextView mShowCount;
```

dalam onCreate():

```
mShowCount = (TextView) findViewById(R.id.show_count);

countUp Method:
    public void countUp(View view) {
        mCount++;
        if (mShowCount != null)
            mShowCount.setText(Integer.toString(mCount));
    }
```

### Tantangan penyusunan kode

Bahkan aplikasi sederhana seperti Hello Toast bisa menjadi dasar dari aplikasi pemberian skor atau pemesanan produk. Tulis satu aplikasi yang akan berguna bagi Anda, atau coba salah satu dari contoh berikut:

- Buat aplikasi pemesanan kopi. Tambahkan tombol untuk mengubah jumlah kopi yang dipesan. Hitung dan tampilkan harga total.
- Buat aplikasi pemberian skor untuk olahraga tim favorit Anda. Buat gambar latar belakang yang mewakili olahraga tersebut. Buat tombol untuk menghitung skor setiap tim.

## RANGKUMAN

Pada bab ini, Anda telah mempelajari cara:

- 1) Menambahkan elemen UI ke aplikasi dalam Layout Editor dan menggunakan XML.
- 2) Membuat interaktif UI dengan tombol dan mengeklik listener.
- 3) Menambahkan listener klik yang memperbarui tampilan teks untuk merespons masukan pengguna.
- 4) Informasi yang ditampilkan ke pengguna menggunakan toast.

## REFERENSI

- Buku Kursus Dasar-Dasar Developer Android – Konsep, Belajar mengembangkan aplikasi Android, Dikembangkan oleh Tim Pelatihan Developer Google.
- Buku Kerja Praktik, Kursus Dasar-Dasar Developer Android — Praktik, Belajar mengembangkan aplikasi Android, Dikembangkan oleh Tim Pelatihan Developer Google
- Safaat N.H. 2014. Android: Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android. Bandung: Informatika.



## MATERI KOMPETENSI

Materi bab ini akan membahas:

- 1) Kontrol user interface (UI).
- 2) Jenis-jenis kontrol user interface pada tool Android Studio
- 3) Komponen kontrol masukan user interface

## YANG HARUS SUDAH ANDA KETAHUI

Untuk materi dan praktik di bab ini Anda harus bisa:

- 1) Membuat dan menjalankan aplikasi di Android Studio.
- 2) Menjalankan aplikasi di emulator atau perangkat.
- 3) Mengimplementasikan TextView dalam layout untuk aplikasi.
- 4) Membuat dan mengedit elemen UI menggunakan Layout Editor, memasukkan kode XML secara langsung, dan mengakses elemen dari kode Java Anda.

## YANG AKAN ANDA BUTUHKAN

Untuk materi dan praktik ini, Anda akan membutuhkan:

- 1) Komputer Mac, Windows, atau Linux. Tool Android Studio untuk persyaratan sistem terbaru.
- 2) Akses internet atau cara alternatif untuk memuat Android Studio dan pemasangan Java terbaru ke komputer Anda.
- 3) Smartphone sebagai perangkat untuk pengujian secara nyata aplikasi yang telah dibuat.

## YANG AKAN ANDA PELAJARI

Anda akan belajar:

- 1) Menambahkan item menu ke menu opsi dan menambahkan ikon di menu opsi.
- 2) Menyetel item menu untuk ditampilkan di bilah tindakan.
- 3) Menambahkan handler kejadian untuk klik item menu.

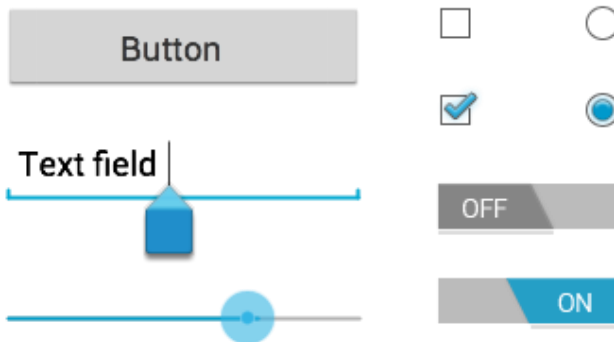
# BAB 4 KONTROL USER INTERFACE



## Bab 4: Kontrol User Interface

### 4.1. Kontrol Input

Kontrol input adalah komponen interaktif di antarmuka pengguna aplikasi Anda. Android menyediakan berbagai macam kontrol yang dapat Anda gunakan di UI Anda, seperti buttons, text fields, seek bars, check box, zoom buttons, toggle buttons, dan banyak lainnya.



Gambar 68. Kontrol user interface

Ada banyak kontrol user interface yang disediakan oleh android, yang mana anda diizinkan untuk membangun grafis user interface pada aplikasi anda.

Tabel 9. Kontrol dan deskripsi penggunaan

Kontrol	Deskripsi User Interface
TextView	Kontrol ini digunakan untuk menampilkan teks ke pengguna.
EditText	EditText adalah subclass yang telah ditetapkan dari TextView yang mencakup kemampuan.



AutoCompleteTextView	AutoCompleteTextView adalah tampilan yang mirip dengan EditText, kecuali bahwa menunjukkan daftar saran penyelesaian secara otomatis saat pengguna mengetik.
Button	Button bisa ditekan atau di klik oleh pengguna untuk melakukan suatu tindakan.
ImageButton	ImageButton adalah sebuah AbsoluteLayout yang memungkinkan anda untuk menentukan lokasi yang tepat dari anak-anaknya. Ini menunjukkan sebuah button berupa gambar bisa ditekan atau di klik oleh pengguna.
CheckBox	Switch on/off dapat diubah oleh pengguna. Anda harus menggunakan indicato ketika presentasi kepada pengguna.
ToggleButton	Sebuah button on/off dengan lampu 214ndicator.
RadioButton	RadioButton memiliki dua sifat yaitu dicentang atau tidak dicentang.
RadioGroup	RadioGroup digunakan untuk kelompok bersama RadioButton.
ProgressBar	ProgressBar memberikan umpan balik visual tentang beberapa tugas yang sedang berlangsung.
Spinner	Sebuah daftar drop-down yang memungkinkan pengguna untuk memilih salah satu nilai dari satu set.
TimePicker	TimePicker memungkinkan pengguna untuk memilih waktu, baik dalam mode 24 jam atau mode AM/PM.
DatePicker	DatePicker memungkinkan pengguna untuk memilih tanggal.

### Membuat User Interface:

Kontrol input adalah komponen interaktif di user interface aplikasi anda. Android menyediakan berbagai macam kontrol yang dapat anda gunakan dalam user interface anda, seperti button, text field, seek bar, check box, zoom button, toggle button, dan masih banyak lagi.

Sebagaimana yang sudah dijelaskan bahwa obyek view memiliki ID unik yang ditugaskan untuk mengidentifikasi. Sintaks untuk ID, dalam sebuah tag XML adalah

```
android:id="@+id/teks"
```

Untuk membuat sebuah kontrol user interface/view/widget anda harus menentukan view/widget dalam file dan menetapkan ID unik sbb.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:gravity="center"
tools:context="com.example.rizaova.project1.MainActivity">

<TextView
    android:id="@+id/teks"
    android:text="Ini Teks"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
</LinearLayout>
```

Setelah membuat contoh dari obyek kontrol, dan untuk memanggilnya dari layout anda bisa gunakan ini.

```
TextView myTeks = (TextView) findViewById(R.id.teks);
```

## 4.2. Jenis-Jenis Kontrol

### a. TextView

Satu tampilan yang mungkin sering Anda gunakan adalah kelas `TextView`, yang merupakan subkelas dari kelas `View` yang menampilkan teks pada layar. Anda bisa menggunakan `TextView` untuk tampilan berukuran apa pun, mulai dari karakter atau kata tunggal hingga teks selayar penuh. Anda bisa menambahkan `id` resource ke `TextView` dan mengontrol cara teks muncul dengan menggunakan atribut di file layout XML.

Anda bisa merujuk tampilan `TextView` di kode Java dengan menggunakan `id` resource-nya, dan memperbarui teks dari kode Anda. Jika Anda ingin memungkinkan pengguna mengedit teks, gunakan `EditText`, subkelas `TextView` yang memungkinkan pengeditan dan masukan teks. Anda mempelajari semua tentang `EditText` di bab lainnya.

#### ▪ Atribut TextView

Anda bisa menggunakan atribut XML untuk mengontrol:

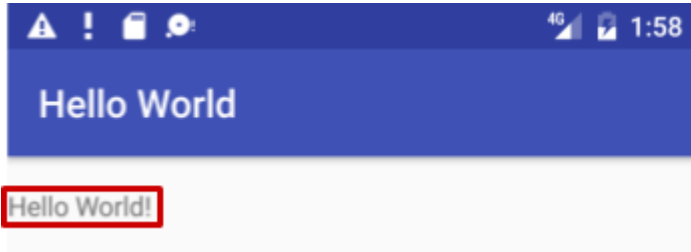
- ⇒ Tempat memosisikan `TextView` di layout (seperti tampilan lainnya)
- ⇒ Cara munculnya tampilan itu sendiri, misalnya dengan warna latar belakang
- ⇒ Seperti apa teks terlihat dalam tampilan, misalnya teks awal dengan gaya, ukuran, dan warnanya

Misalnya, untuk menyetel lebar, tinggi, dan posisi dalam `LinearLayout`:

```
<TextView
    ...
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    ... />
```

Untuk menyetel nilai teks awal tampilan, gunakan atribut `android:text`:

```
android:text="Hello World!"
```



Gambar 69. Hello world

Anda bisa mengekstrak string teks ke dalam resource string (mungkin disebut `hello_world`) yang lebih mudah dikelola untuk versi aplikasi multibahasa, atau jika Anda perlu mengubah string di lain waktu. Setelah mengekstrak string, gunakan nama resource string dengan `@string/` untuk menetapkan teks:

```
android:text="@string/hello_world"
```

Atribut yang paling sering digunakan bersama `TextView` adalah sebagai berikut:

Tabel 10. Atribute `TextView`

Atribut	Penggunaan
<code>android:text</code>	Menyetel teks yang akan ditampilkan
<code>android:textColor</code>	Menyetel warna teks. Anda bisa menyetel atribut ke suatu nilai warna, resource yang telah didefinisikan sebelumnya, atau tema
<code>android:textAppearance</code>	Penampilan teks, termasuk warna, jenis huruf, gaya, dan ukurannya. Anda menyetel atribut ini untuk resource gaya yang telah didefinisikan sebelumnya atau tema yang sudah mendefinisikan nilai-nilai ini
<code>android:textSize</code>	Menyetel ukuran teks (jika belum disetel melalui <code>android:textAppearance</code> ). Gunakan ukuran <code>sp</code> (piksel yang diskalakan) seperti <code>20sp</code> atau <code>14.5sp</code> , atau menyetel atribut untuk resource atau tema yang telah didefinisikan sebelumnya

<code>android:textSize</code>	Menyetel gaya teks (jika belum disetel melalui <code>android:textAppearance</code> ). Gunakan <code>normal</code> , <code>bold</code> , <code>italic</code> , atau <code>bold italic</code>
<code>Android:typeface</code>	Menyetel jenis huruf teks (jika belum disetel melalui <code>android:textAppearance</code> ). Gunakan <code>normal</code> , <code>sans</code> , <code>serif</code> , atau <code>monospace</code>
<code>android:lineSpacingExtra</code>	Menyetel spasi tambahan antar baris teks. Gunakan ukuran <code>sp</code> (piksel yang diskalakan) atau <code>dp</code> (piksel yang tidak tergantung perangkat), atau setel atribut ke resource atau tema yang telah didefinisikan sebelumnya
<code>android:autoLink</code>	Mengontrol apakah tautan seperti URL dan alamat email secara otomatis ditemukan dan diubah ke tautan yang dapat diklik (dapat disentuh). Gunakan salah satu dari berikut ini: <ul style="list-style-type: none"> <li>▪ <code>none</code>: Tidak ada pola yang cocok (default).</li> <li>▪ <code>web</code>: Cocok dengan URL web.</li> <li>▪ <code>email</code>: Cocok dengan alamat email.</li> <li>▪ <code>phone</code>: Cocok dengan nomor telepon.</li> <li>▪ <code>map</code>: Cocok dengan alamat peta.</li> <li>▪ <code>all</code>: Cocok dengan semua pola (setara dengan <code>web email telepon peta</code>).</li> </ul> <code>android:autoLink="web"</code> Misalnya, untuk menyetel atribut agar cocok dengan URL web, gunakan ini

- Menggunakan tag yang disematkan dalam teks

Dalam aplikasi yang mengakses artikel majalah atau koran, artikel yang muncul mungkin berasal dari sumber online atau mungkin disimpan sebelumnya dalam database di perangkat. Anda juga bisa membuat teks sebagai satu string panjang di resource strings.xml.


Dalam kasus lain, teks bisa berisi tag HTML yang disematkan atau kode pemformatan teks lainnya. Agar tampil dengan benar, tampilan teks harus diformat dengan aturan berikut:

- ⇒ Jika ada tanda apostrof (') di teks, Anda harus *meloloskannya* dengan mengawalinya dengan backslash (\'). Jika menggunakan tanda kutip ganda dalam teks, Anda juga harus meloloskannya (\''). Anda juga harus meloloskan karakter non-ASCII lainnya.
- ⇒ TextView mengabaikan semua tag HTML kecuali yang berikut ini:
  - Penggunaan tag HTML dan `</b>` di sekitar kata yang harus ditulis tebal.
  - Penggunaan tag HTML *dan* di sekitar kata yang harus ditulis miring. Akan tetapi perhatikan, jika menggunakan apostrof keriting dalam frasa miring, Anda harus menggantinya dengan apostrof tegak.
  - Anda bisa mengombinasikan tebal dan miring dengan mengombinasikan tag, seperti dalam ... kata...`</i></b>`.

Untuk membuat string teks panjang dalam file **strings.xml**, masukkan keseluruhan teks dalam `<string name="your_string_name"></string>` di file **strings.xml** (*your\_string\_name* adalah nama yang Anda berikan untuk resource string, misalnya *article\_text*).

Baris teks dalam file **strings.xml** tidak digulung ke baris berikutnya — melainkan memanjang melebihi margin kanan. Ini adalah perilaku yang benar. Setiap baris teks baru yang dimulai dari margin kiri menyatakan paragraf lengkap.

Masukkan `\n` untuk menyatakan akhir baris, dan `\n` lainnya untuk menyatakan baris kosong. Jika Anda tidak menambahkan karakter penutup baris, paragraf akan saling bertabrakan saat ditampilkan di layar.

	<p><b>Tip:</b> Jika ingin melihat teks yang dibungkus dalam <b>strings.xml</b>, Anda bisa menekan Return atau Enter untuk memasukkan akhiran ganti baris, atau memformat teks terlebih dahulu dalam editor teks dengan akhiran ganti baris. Akhiran tidak akan ditampilkan di layar.</p>
--	--

- Merujuk TextView di kode

Untuk merujuk TextView di kode Java Anda, gunakan *id* resourcenya. Misalnya, untuk memperbarui TextView dengan teks baru, Anda perlu:

- 1) Mencari TextView (dengan *id* `show_count`) dan menetakannya ke variabel. Anda menggunakan metode `findViewById()` kelas View, dan merujuk ke tampilan yang ingin dicari dengan menggunakan format ini:
- 2) `R.id.view_id`

Dalam hal ini, `view_id` adalah identifier resource untuk tampilan:

```
mShowCount = (TextView) findViewById(R.id.show_count)
```

- 3) Setelah mengambil tampilan sebagai variabel anggota `TextView`, selanjutnya Anda bisa menyetel teks untuk tampilan teks ke teks baru dengan menggunakan metode

```
setText() kelas TextView:  
mShowCount.setText(mCount_text);
```

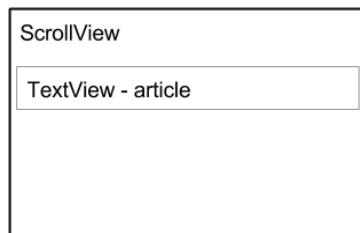
- Tampilan bergulir (*ScrollView*)

Jika informasi yang ingin ditampilkan di aplikasi Anda lebih besar dari tampilan perangkat, Anda bisa membuat *tampilan bergulir* yang bisa digulir pengguna secara vertikal dengan menggesek ke atas atau ke bawah, atau secara horizontal dengan menggesek ke kanan atau ke kiri.

Anda biasanya akan menggunakan tampilan bergulir untuk kabar berita, artikel, atau teks panjang lainnya yang tidak muat pada tampilan. Anda juga bisa menggunakan tampilan bergulir untuk mengombinasikan tampilan (seperti `TextView` dan `Button`) dalam tampilan bergulir.

Membuat layout dengan `ScrollView`

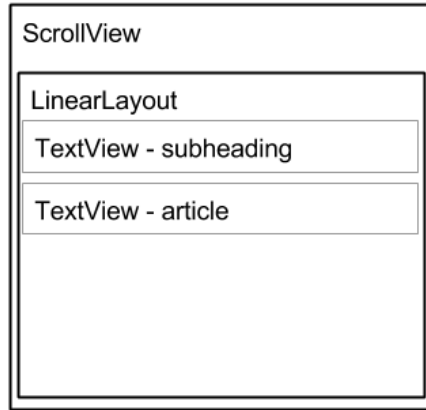
Kelas `ScrollView` menyediakan layout untuk tampilan yang bergulir vertikal. (Untuk pengguliran horizontal, Anda perlu menggunakan `HorizontalScrollView`.) `ScrollView` adalah subkelas `FrameLayout`, yang berarti Anda hanya bisa menempatkan *satu* tampilan sebagai anak; anak tersebut berisi keseluruhan materi yang akan digulir.



Gambar 70. Scrollview

Meskipun Anda hanya bisa menempatkan satu tampilan anak di dalam `ScrollView`, tampilan anak bisa menjadi grup tampilan dengan hierarki tampilan

anak, seperti `LinearLayout`. Pilihan yang bagus untuk tampilan dalam `ScrollView` adalah `LinearLayout` yang disusun dalam orientasi vertikal.



Gambar 71. Scrollview orientasi vertikal

a. `ScrollView` dan kinerja

Dengan `ScrollView`, semua tampilan Anda berada dalam memori dan dalam hierarki tampilan sekalipun tidak ditampilkan di layar. Ini membuat `ScrollView` berguna untuk menggulir laman teks berbentuk bebas dengan lancar, karena teks sudah ada di memori. Akan tetapi, `ScrollView` bisa menggunakan banyak memori, yang bisa memengaruhi kinerja aplikasi lainnya.

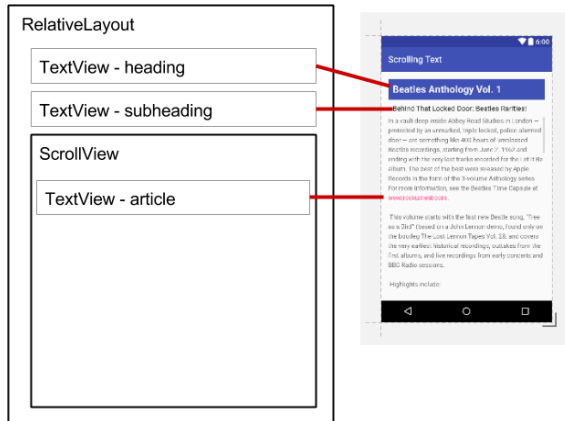
Menggunakan instance `LinearLayout` yang disarangkan juga bisa mengakibatkan hierarki tampilan yang terlalu dalam, sehingga bisa memperlambat kinerja. Menyarangkan sejumlah instance `LinearLayout` yang menggunakan atribut `android:layout_weight` bisa menjadi sangat mahal karena setiap tampilan anak perlu diukur dua kali. Pertimbangkan penggunaan layout yang lebih datar seperti `RelativeLayout` atau `GridLayout` untuk meningkatkan kinerja.

Layout kompleks dengan `ScrollView` mungkin mengalami masalah kinerja, khususnya dengan tampilan anak seperti gambar. Kami menyarankan agar Anda *tidak* menggunakan gambar bersama `ScrollView`. Untuk menampilkan daftar panjang item, atau gambar, pertimbangkan untuk menggunakan `RecyclerView`. Selain itu, menggunakan `AsyncTask` akan menyediakan cara sederhana untuk melakukan pekerjaan di luar thread utama, seperti memuat gambar dalam thread latar belakang, kemudian menerapkannya ke UI setelah selesai.



## b. ScrollView dengan TextView

Untuk menampilkan artikel majalah yang dapat digulir pada layar, Anda dapat menggunakan RelativeLayout bagi layar yang menyertakan TextView terpisah untuk judul artikel, lainnya untuk subjudul artikel, dan TextView ketiga untuk teks artikel bergulir (lihat gambar di bawah ini), yang disetel dalam ScrollView. Satu-satunya bagian layar yang bergulir adalah ScrollView dengan teks artikel.

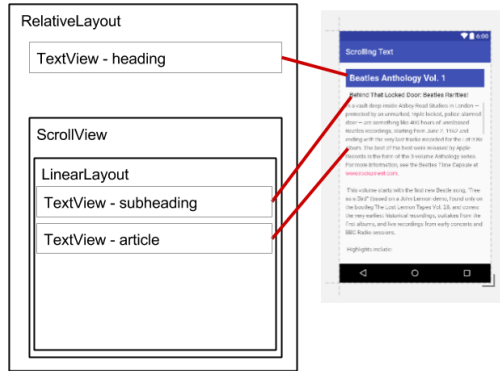


Gambar 72. ScrollView dengan teks artikel

## c. ScrollView dengan LinearLayout

Grup tampilan ScrollView hanya bisa berisi satu tampilan; akan tetapi, tampilan tersebut bisa berupa grup tampilan yang berisi beberapa tampilan, misalnya LinearLayout. Anda bisa menyarangkan grup tampilan seperti LinearLayout *dalam* grup tampilan ScrollView, sehingga menggulir apa pun yang ada di dalam LinearLayout.

Misalnya, jika Anda ingin subjudul artikel digulir bersama dengan artikel meskipun berupa TextView terpisah, Anda bisa menambahkan LinearLayout dalam ScrollView, dan memindahkan TextView subjudul, bersama TextView artikel, ke dalam LinearLayout.



Gambar 73. Grup tampilan: subjudul dan artikel

Grup tampilan `LinearLayout` menjadi tampilan anak tunggal dalam `ScrollView` seperti yang ditampilkan dalam gambar di bawah ini, dan pengguna bisa menggulir keseluruhan grup tampilan tersebut: subjudul dan artikel.

Saat menambahkan `LinearLayout` di dalam `ScrollView`, gunakan `match_parent` untuk atribut `android:layout_width` `LinearLayout` guna mencocokkan dengan lebar grup tampilan parent (`ScrollView`), dan gunakan `wrap_content` untuk atribut `android:layout_height` `LinearLayout` guna membuat grup tampilan cukup besar untuk menampung materi dan pengisinya.

Karena `ScrollView` hanya mendukung pengguliran vertikal, Anda harus menyetel orientasi `LinearLayout` ke vertikal (dengan menggunakan atribut `android:orientation="vertical"` ), sehingga keseluruhan `LinearLayout` akan bergulir secara vertikal. Misalnya, layout XML berikut menggulir `TextView` `article` bersama `article_subheading` `TextView`:

```
<ScrollView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/article_heading">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <TextView
            android:id="@+id/article_subheading"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
```

```

        android:padding="@dimen/padding_regular"
        android:text="@string/article_subtitle"

android:textAppearance="@android:style/TextAppearance" />

<TextView
    android:id="@+id/article"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:autoLink="web"
    android:lineSpacingExtra="@dimen/line_spacing"
    android:text="@string/article_text" />

</LinearLayout>
</ScrollView>

```

## ▪ Masukan Teks

Gunakan kelas `EditText` untuk mendapatkan masukan pengguna yang terdiri dari karakter tekstual, termasuk angka dan simbol. `EditText` memperluas kelas `TextView`, agar `TextView` dapat diedit.

Menyesuaikan objek `EditText` untuk masukan pengguna:

Dalam `Layout Manager Android Studio`, buat tampilan `EditText` dengan menambahkan sebuah `EditText` ke layout Anda dengan XML berikut:

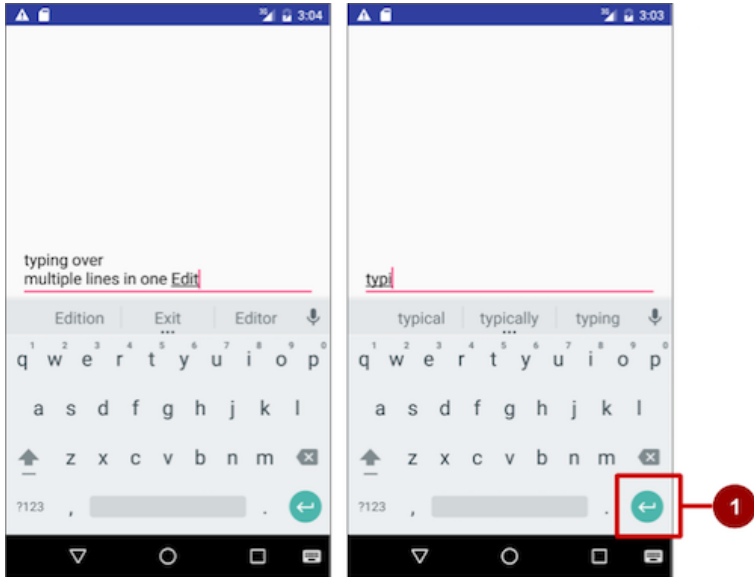
```

<EditText
    android:id="@+id/edit_simple"
    android:layout_height="wrap_content"
    android:layout_width="match_parent">
</EditText>

```

Mengaktifkan beberapa baris masukan

Secara default, tampilan `EditText` memungkinkan beberapa baris masukan seperti yang ditampilkan dalam gambar di bawah ini, dan menyarankan koreksi ejaan. Mengetuk tombol `Return` (disebut juga dengan `Enter`) pada keyboard di layar mengakhiri suatu baris dan memulai baris baru dalam tampilan `EditText` yang sama.



Gambar 74. Tombol Return pada Keyboard

**Catatan:**

Dalam gambar di atas, #1 adalah tombol Return (disebut juga dengan Enter).

- Memungkinkan Return untuk lanjut ke tampilan berikutnya

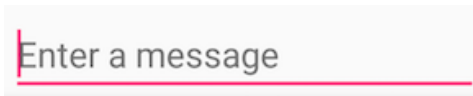
Jika Anda menambahkan atribut `android:inputType` ke tampilan `EditText` dengan nilai seperti `"textCapCharacters"` (untuk mengubah masukan menjadi huruf besar semua) atau `"textAutoComplete"` (untuk mengaktifkan saran ejaan saat pengguna mengetik), menyetuk tombol Return akan menutup keyboard di layar dan melanjutkan fokus ke tampilan berikutnya. Perilaku ini berguna jika Anda ingin pengguna mengisi suatu formulir yang terdiri dari bidang `EditText`, sehingga pengguna bisa melanjutkan dengan cepat ke tampilan `EditText` berikutnya.

Atribut untuk menyesuaikan tampilan `EditText`

Gunakan atribut untuk menyesuaikan tampilan `EditText` untuk masukan. Misalnya:

- ⇒ `android:maxLines="1"`: Setel entri teks untuk menampilkan satu baris saja.

- ⇒ `android:lines="2"`: Setel entri teks untuk menampilkan 2 baris, bahkan jika panjang teksnya kurang.
- ⇒ `android:maxLength="5"`: Setel jumlah karakter masukan maksimum ke 5.
- ⇒ `android:inputType="number"`: Batasi entri teks untuk angka.
- ⇒ `android:digits="01"`: Batasi digit yang dimasukkan hanya "0" dan "1".
- ⇒ `android:textColorHighlight="#7cff88"`: Setel warna latar belakang teks yang dipilih (disorot).
- ⇒ `android:hint="@string/my_hint"`: Setel teks untuk muncul dalam bidang yang menyediakan petunjuk untuk pengguna, seperti "Masukkan pesan".



Gambar 75. Masukan pesan

### Mendapatkan masukan pengguna

Memungkinkan pengguna untuk memasukkan teks hanya berguna jika Anda bisa menggunakan teks itu dalam beberapa cara dalam aplikasi. Untuk menggunakan masukan, Anda terlebih dahulu harus mendapatkannya dari tampilan `EditText` dalam layout XML. Langkah-langkah yang bisa Anda ikuti untuk mempersiapkan tampilan `EditText` dan mendapatkan masukan pengguna adalah:

1. Buat elemen tampilan `EditText` di layout XML untuk suatu aktivitas. Pastikan untuk mengidentifikasi elemen ini dengan sebuah `android:id` sehingga Anda bisa merujuknya dengan `id`:

```
android:id="@+id/editText_main"
```

2. Dalam kode Java untuk aktivitas yang sama, buat sebuah metode dengan parameter `View` yang mengambil objek `EditText` (dalam contoh di bawah, `editText`) untuk tampilan `EditText`, dengan menggunakan metode `findViewById()` kelas `View` untuk menemukan tampilan melalui ID-nya

```
(editText_main):
```

```
EditText editText = (EditText)
findViewById(R.id.editText_main);
```

- Gunakan metode `getText()` dari kelas `EditText` (diwarisi dari kelas `TextView`) untuk mendapatkan teks sebagai urutan karakter (`CharSequence`). Anda bisa mengonversi urutan karakter menjadi string dengan menggunakan metode `toString()` dari kelas `CharSequence`, yang mengembalikan string yang menyatakan data dalam urutan karakter.

```
String showString = editText.getText().toString()
```



**Tip:** Anda bisa menggunakan metode `valueOf()` dari kelas `Integer` untuk mengonversi string menjadi integer jika masukan adalah sebuah integer.

### Mengubah perilaku keyboard dan masukan

Sistem Android menampilkan keyboard di layar—disebut dengan metode masukan *lunak*—jika sebuah bidang teks dalam UI menerima fokus. Untuk menyediakan pengalaman pengguna terbaik, Anda bisa menetapkan karakteristik tipe masukan yang diharapkan aplikasi, misalnya apakah berupa nomor telepon atau alamat email. Anda juga bisa menetapkan perilaku metode masukan, misalnya apakah menampilkan saran ejaan atau menyediakan huruf besar di awal kalimat. Anda bisa mengubah metode masukan lunak menjadi keypad numerik untuk memasukkan angka saja, atau bahkan keypad telepon untuk nomor telepon.

Android juga menyediakan kerangka kerja yang bisa diperluas bagi programmer tingkat lanjut untuk mengembangkan dan memasang Input Method Editor (IME) mereka sendiri untuk masukan ucapan, entri keyboard tipe khusus, dan aplikasi lain.


Deklarasikan metode masukan dengan menambahkan atribut `android:inputType` ke tampilan `EditText`. Misalnya, atribut berikut menyetel keyboard di layar menjadi keypad telepon:

```
android:inputType="phone"
```

Gunakan atribut `android:inputType` dengan nilai berikut:

Atribut	Penggunaan
<code>textCapSentences</code>	Setel keyboard untuk menjadikan huruf besar di awal kalimat

textAutoCorrect	Aktifkan saran ejaan saat pengguna mengetik
textPassword	Ubah setiap karakter yang dimasukkan pengguna menjadi titik untuk menyembunyikan sandi yang dimasukkan
textEmailAddress	Untuk entri email, tampilkan keyboard email dengan simbol "@" yang ditempatkan agar mudah digunakan di sebelah tombol spasi
phone	Untuk entri nomor telepon, tampilkan keypad telepon numerik

	<p><b>Tip:</b> Anda bisa menggunakan karakter pipa ( ) (Java bitwise OR) untuk mengombinasikan nilai atribut <code>android:inputType</code>:</p> <pre>android:inputType="textAutoCorrect textCapSentences"</pre>
---	--

- Mengubah tombol "aksi" pada keyboard

Pada perangkat Android, tombol "aksi" adalah tombol **Return**. Tombol ini umumnya digunakan untuk memasukkan baris teks lain untuk elemen `EditText` yang memungkinkan beberapa baris. Jika Anda menyetel atribut `android:inputType` untuk tampilan `EditText` dengan nilai seperti `"textCapCharacters"` (untuk mengubah masukan menjadi huruf besar semua) atau `"textAutoComplete"` (untuk mengaktifkan saran ejaan saat pengguna mengetik), tombol **Return** menutup keyboard di layar dan melanjutkan fokus ke tampilan berikutnya.

Jika Anda ingin pengguna memasukkan sesuatu selain teks, seperti nomor telepon, Anda mungkin ingin mengubah tombol "aksi" menjadi ikon untuk tombol **Send**, dan mengubah tindakan menjadi menekan nomor telepon. Ikuti langkah-langkah ini:

1. Gunakan atribut `android:inputType` untuk menyetel tipe masukan keyboard:

```
<EditText
    android:id="@+id/phone_number"
    android:inputType="phone"
    ... >
</EditText>
```

Metode `android:inputType` dalam contoh di atas, menyetel tipe keyboard ke `phone`, yang memaksa satu baris masukan (untuk nomor telepon).

2. Gunakan `setOnEditorActionListener()` untuk menyetel listener tampilan `EditText` dalam merespons penggunaan tombol "aksi":

```
EditText editText = (EditText)
findViewById(R.id.phone_number);

editText.setOnEditorActionListener(new

TextView.OnEditorActionListener() {
    // Add onEditorAction() method
}
```

3. Gunakan konstanta `IME_ACTION_SEND` dalam kelas `EditorInfo` bagi `actionId` untuk menampilkan tombol **Send** sebagai tombol "aksi", dan membuat metode untuk merespons tombol **Send** yang ditekan (dalam hal ini, `dialNumber` untuk memanggil nomor telepon):

```
@Override

public boolean onEditorAction(TextView textView,
    int actionId, KeyEvent keyEvent) {
    boolean handled = false;
    if (actionId == EditorInfo.IME_ACTION_SEND) {
        dialNumber();
        handled = true;
    }
    return handled;
});
```



## b. Button (Tombol)

Button atau Tombol, juga salah satu komponen sebuah interface yang sudah biasa digunakan. Dengan sebuah button pengguna dapat memerintahkan aplikasi untuk melakukan aksi yang telah ditentukan. Anda bisa membuat Tombol menggunakan:

- Hanya teks, seperti yang ditampilkan pada sisi kiri gambar di bawah ini.
- Hanya ikon, seperti yang ditampilkan di tengah gambar di bawah ini.
- Teks dan ikon sekaligus, seperti yang ditampilkan pada sisi kanan gambar di bawah ini.

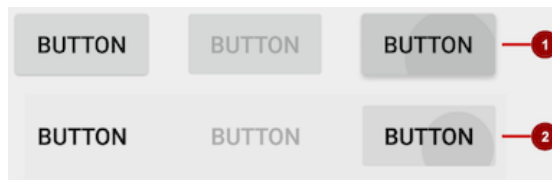
Bila disentuh atau diklik, tombol akan melakukan suatu aksi. Teks dan/atau ikon menyediakan petunjuk mengenai aksi itu. Ini disebut "tombol-tekan" dalam Android.



Gambar 76. Tombol

Tombol adalah suatu persegi atau persegi bersudut tumpul, dengan alas lebih lebar daripada tingginya, dengan teks deskriptif di tengahnya.

Android menawarkan sejumlah tipe tombol, termasuk tombol timbul dan tombol datar seperti yang ditampilkan dalam gambar di bawah ini. Tombol-tombol ini memiliki tiga keadaan: normal, dinonaktifkan, dan ditekan.



Gambar 77. Tiga keadaan: normal, dinonaktifkan, dan ditekan

Dalam gambar di atas:

1. Tombol timbul dalam tiga keadaan: normal, dinonaktifkan, dan ditekan.
2. Tombol datar dalam tiga keadaan: normal, dinonaktifkan, dan ditekan.

Tabel 11. Atribut button yang umum disetting:

Atribut	Deskripsi
<code>android: id</code>	ID yang digunakan untuk mengidentifikasi UI Control secara unique.
<code>android:textAppearance</code>	atribut yang digunakan untuk menampilkan ukuran text
<code>android:gravity</code>	Digunakan untuk mengatur bagaimana suatu content dalam sebuah object ditempatkan secara X dan Y axis di dalam object itu sendiri. Value ini bisa diisi satuatau lebih (dipisahkan dengan “ ”) dari konstanta-konstanta yang disediakan untuk gravity
<code>android:layout_width</code>	Ukuran lebar Layout dari UI Control ( <code>wrap_content</code> , <code>wrap_parent</code> , nilai lebar)
<code>android:width</code>	Ukuran lebar EditText dalam satuan dp (device pixels). Contoh: 15dp
<code>android:layout_height</code>	Ukuran tinggi Layout dari UI Control ( <code>wrap_content</code> , <code>wrap_parent</code> , nilai lebar)
<code>android:teks</code>	berisi text yang akan tampil di layar

#### ▪ Mendesain tombol timbul

Tombol timbul adalah persegi atau persegi bersudut tumpul yang tampak terangkat dari layar—bayangan di sekitarnya menunjukkan bahwa tombol ini bisa disentuh atau diklik. Tombol timbul bisa menampilkan teks atau ikon, atau menampilkan teks dan ikon sekaligus.

Untuk menggunakan tombol timbul yang sesuai dengan Spesifikasi Desain Material, ikuti langkah-langkah ini:

- Dalam file **build.gradle (Modul: app)** Anda, tambahkan pustaka `appcompat` terbaru ke bagian `dependencies` :

```
compile 'com.android.support:appcompat v7:x.x.x.'
```

Di atas, `x.x.x` adalah nomor versi. Jika nomor versi yang Anda tetapkan lebih rendah dari nomor versi pustaka yang tersedia saat ini, Android Studio akan memperingatkan Anda ("a newer version is available"). Perbarui nomor versi ke versi yang disebutkan Android Studio untuk digunakan.

Buat activity Anda memperluas

```
android.support.v7.app.AppCompatActivity:
public class MainActivity extends AppCompatActivity {
    ...
}
```

- Gunakan elemen `Button` dalam file layout. Tidak perlu atribut tambahan, karena tombol timbul adalah gaya default.

```
<Button
android:layout_width="wrap_content"
android:layout_height="wrap_content"
... />
```

Gunakan tombol timbul untuk lebih menonjolkan tindakan di layout dengan berbagai macam materi. Tombol timbul menambahkan dimensi pada layout datar—tombol tersebut menekankan fungsi pada ruang yang penuh atau longgar. Tombol timbul menampilkan bayangan latar belakang saat disentuh (ditekan) atau diklik, seperti yang ditampilkan di bawah ini.



Dalam gambar di atas:

1. Keadaan normal: Dalam keadaan normal, tombol tampak seperti tombol timbul.
  2. Keadaan dinonaktifkan: Bila dinonaktifkan, tombol akan berwarna abu-abu dan tidak aktif dalam konteks aplikasi. Di sebagian besar kasus, Anda akan menyembunyikan tombol nonaktif, namun kadang Anda ingin menampilkannya sebagai dinonaktifkan.
  3. Keadaan ditekan: Keadaan ditekan, dengan bayangan latar belakang lebih besar, menunjukkan bahwa tombol sedang disentuh atau diklik. Bila Anda memasang callback ke tombol (seperti atribut `OnClick`), callback akan dipanggil saat tombol dalam keadaan ini.
- **Membuat tombol timbul dengan teks**

Beberapa tombol timbul paling baik didesain sebagai teks, tanpa ikon, seperti tombol "Save", karena ikon saja tidak menyampaikan makna yang jelas. Untuk membuat tombol timbul dengan teks, gunakan kelas `Button`, yang memperluas kelas `TextView`.

Untuk membuat tombol timbul dengan teks saja, gunakan kelas `Button` di layout XML Anda seperti berikut:

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_text"
    ... />
```

Praktik terbaik terkait tombol teks adalah mendefinisikan satu kata yang sangat pendek sebagai resource string (`button_text` dalam contoh di atas), sehingga string bisa diterjemahkan. Misalnya, "Save" bisa diterjemahkan ke dalam bahasa Prancis sebagai "Enregister" tanpa mengubah kode.

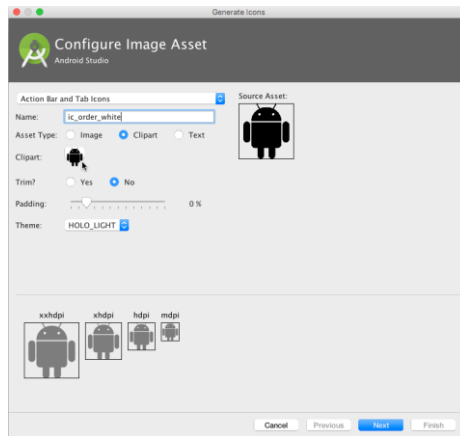
### ▪ Membuat tombol timbul dengan ikon dan teks

Walaupun tombol biasanya menampilkan teks yang memberi tahu pengguna fungsi tombol tersebut, tombol timbul juga bisa menampilkan ikon beserta teks.

#### Memilih ikon:

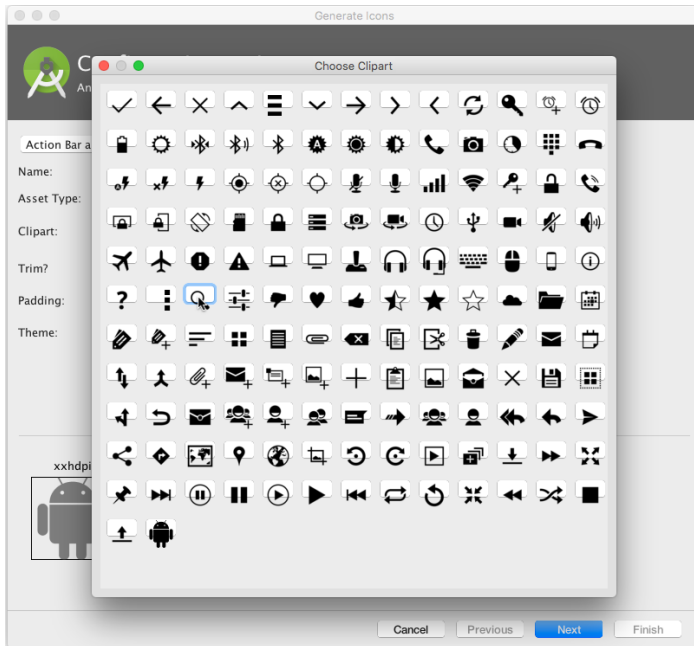
Untuk memilih gambar ikon standar yang diubah ukurannya untuk berbagai tampilan, ikuti langkah-langkah ini:

- ⇒ Luaskan **app > res** dalam tampilan Project, dan klik-kanan (atau klik Command) **drawable**.
- ⇒ Pilih **New > Image Asset**. Dialog Configure Image Asset akan muncul.



Gambar 78. Dialog Configure Image Asset

- ⇒ Pilih **Action Bar and Tab Items** dalam menu tarik-turun dialog Configure Image Asset.
- ⇒ Klik gambar Clipart: (logo Android) untuk memilih gambar clipart sebagai ikon. Laman ikon akan muncul seperti yang ditampilkan di bawah ini. Klik ikon yang ingin Anda gunakan.



Gambar 79. Clipart


- ⇒ Anda mungkin perlu membuat penyesuaian berikut:
  - Pilih **HOLO\_DARK** dari menu tarik-turun Theme untuk menyetel ikon menjadi putih dengan latar belakang berwarna gelap (atau hitam).
  - Bergantung pada bentuk ikon, Anda mungkin ingin menambahkan pengisi pada ikon sehingga ikon tidak memenuhi teks. Seret slider Padding ke kanan untuk menambahkan lapisan lebih banyak.
- ⇒ Klik **Next**, kemudian klik **Finish** dalam dialog Confirm Icon Path. Nama ikon seharusnya sekarang muncul di folder **app > res > drawable**.

Gambar vektor ikon standar diubah ukurannya secara otomatis untuk ukuran tampilan perangkat yang berbeda. Untuk memilih gambar vektor, ikuti langkah-langkah ini:

- ⇒ Luaskan **app > res** pada tampilan Project, dan klik-kanan (atau klik Command) **drawable**.
- ⇒ Pilih **New > Vector Asset** untuk ikon yang mengubah ukurannya sendiri secara otomatis untuk setiap tampilan.
- ⇒ Dialog Vector Asset Studio muncul untuk aset vektor. Klik tombol radio **Material Icon**, kemudian klik tombol **Choose** untuk memilih ikon dari spesifikasi Desain Material.
- ⇒ Klik **Next** setelah memilih ikon, dan klik **Finish** untuk menyelesaikan. Nama ikon seharusnya sekarang muncul di folder **app > res > drawable**.

#### ▪ Menambahkan tombol dengan teks dan ikon pada layout

Untuk membuat tombol dengan teks dan ikon seperti yang ditampilkan dalam gambar di bawah ini, gunakan `Button` dalam layout XML Anda. Tambahkan atribut `android:drawableLeft` untuk menggambar ikon di sebelah kiri teks tombol, seperti yang ditampilkan dalam gambar di bawah ini:

<pre>&lt;Button     android:layout_width="wrap_content"     android:layout_height="wrap_content"     android:text="@string/button_text"      android:drawableLeft="@drawable/button_i con"     ... /&gt;</pre>	
--	--

#### ▪ Membuat tombol timbul dengan ikon saja

Jika ikon dipahami secara universal, Anda mungkin ingin menggunakannya sebagai ganti teks. Untuk membuat tombol timbul hanya dengan ikon atau gambar (tanpa teks), gunakan kelas `ImageButton`, yang memperluas kelas `ImageView`. Anda bisa menambahkan sebuah `ImageButton` ke layout XML Anda seperti berikut:

```
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/button_icon"
    ...
/>
```

## ▪ Mengubah gaya dan penampilan tombol timbul

Cara termudah untuk menampilkan tombol timbul yang lebih menonjol adalah menggunakan warna latar belakang yang berbeda untuk tombol tersebut. Anda bisa menetapkan atribut `android:background` dengan resource warna atau resource dapat digambar:

```
android:background="@color/colorPrimary"
```

Penampilan tombol Anda—warna latar belakang dan font—mungkin bervariasi antara satu perangkat dengan perangkat lain, karena perangkat dari produsen yang berbeda sering kali memiliki gaya default berbeda untuk kontrol masukan. Anda bisa mengontrol dengan tepat gaya tombol dan kontrol masukan lainnya menggunakan *tema* yang Anda terapkan untuk seluruh aplikasi.

Misalnya, untuk memastikan bahwa semua perangkat yang bisa menjalankan tema Holo akan menggunakan tema Holo untuk aplikasi Anda, deklarasikan yang berikut ini dalam elemen `<application>` file `AndroidManifest.xml`:

```
android:theme="@android:style/Theme.Holo"
```

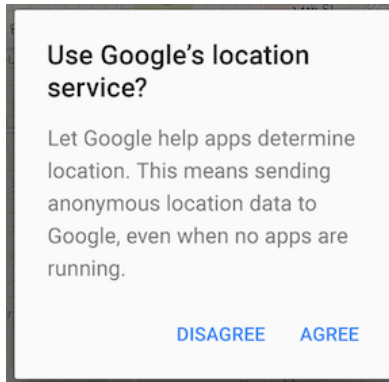
Setelah menambahkan deklarasi di atas, aplikasi akan ditampilkan menggunakan tema. Aplikasi yang didesain untuk Android 4.0 dan yang lebih tinggi juga bisa menggunakan keluarga tema publik `DeviceDefault`. Tema `DeviceDefault` adalah nama lain untuk tampilan dan cara kerja perangkat. Keluarga tema dan keluarga gaya widget `DeviceDefault` menawarkan berbagai cara bagi developer untuk menargetkan tema asli perangkat dengan semua intact penyesuaian.

Untuk aplikasi Android yang menjalankan versi 4.0 dan yang lebih baru, Anda memiliki opsi berikut:

- Gunakan sebuah tema, misalnya salah satu tema Holo, sehingga aplikasi Anda memiliki penampilan yang sama di semua perangkat Android yang menjalankan versi 4.0 atau yang lebih baru. Dalam kasus ini, penampilan aplikasi tidak berubah saat dijalankan pada perangkat dengan kulit default atau kulit khusus yang berbeda.
- Gunakan salah satu tema `DeviceDefault` sehingga aplikasi Anda tampil dengan kulit default perangkat.
- Jangan gunakan tema, tetapi Anda mungkin akan mendapatkan hasil yang tidak dapat diprediksi pada beberapa perangkat.

## ▪ Mendesain tombol datar

Tombol datar, disebut juga dengan tombol tanpa bingkai, adalah tombol teks saja yang muncul datar di layar tanpa bayangan. Manfaat utama tombol datar adalah kemudahan — tombol ini meminimalkan gangguan dari materi. Tombol datar berguna saat Anda memiliki sebuah dialog, seperti yang ditampilkan dalam gambar di bawah ini, yang memerlukan interaksi atau masukan pengguna. Dalam hal ini, Anda mungkin perlu memiliki gaya dan font yang sama dengan teks sekitar tombol. Tombol ini menjaga tampilan dan cara kerja tetap sama di semua elemen dalam dialog.



Gambar 80. Tombol datar

Tombol datar, yang ditampilkan di bawah ini, menyerupai tombol dasar kecuali dalam hal tidak adanya bingkai atau latar belakang, tetapi tetap mengubah penampilan dalam keadaan yang berbeda. Sebuah tombol datar menampilkan bayangan tinta di sekitarnya bila ditekan (disentuh atau diklik).



**Catatan:** Jika Anda menggunakan tombol datar dalam sebuah layout, pastikan untuk menggunakan lapisan untuk membedakannya dari teks yang mengelilinginya, agar pengguna mudah melihatnya.

Untuk membuat tombol datar, gunakan kelas [Button](#). Tambahkan sebuah `Button` ke layout XML Anda, dan terapkan:

`"?android:attr/borderlessButtonStyle"` sebagai atribut `style`:

```
<Button
```



```

android:id="@+id/button_send"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/button_send"
android:onClick="sendMessage"
style="?android:attr/borderlessButtonStyle" />

```

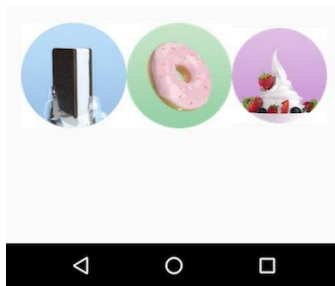
### ▪ Mendesain gambar sebagai tombol

Anda bisa mengubah Tampilan apa pun, seperti `ImageView`, menjadi sebuah tombol dengan menambahkan atribut `android:onClick` dalam layout XML. Gambar untuk `ImageView` harus sudah disimpan dalam folder **drawables** proyek Anda.



**Catatan:** Untuk memasukkan gambar ke dalam proyek Android Studio Anda, buat atau simpan gambar dalam format JPEG, lalu salin file gambar ke dalam folder **app > src > main > res > drawables** proyek Anda.

Jika Anda menggunakan beberapa gambar sebagai tombol, atur gambar dalam sebuah viewgroup sehingga dikelompokkan bersama. Misalnya, gambar berikut dalam folder `drawable` (`icecream_circle.jpg`, `donut_circle.jpg`, dan `froyo_circle.jpg`) didefinisikan untuk `ImageView` yang dikelompokkan dalam `LinearLayout` dan disetel ke orientasi horizontal, sehingga gambar-gambar tersebut muncul bersebelahan:



Gambar 81. `ImageView`

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    android:layout_marginTop="260dp">
    <ImageView

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/icecream_circle"
        android:onClick="orderIcecream"/>
<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/donut_circle"
    android:onClick="orderDonut"/>
<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/froyo_circle"
    android:onClick="orderFroyo"/>
</LinearLayout>

```

### ▪ Mendesain tombol aksi mengambang

Tombol aksi mengambang, yang ditampilkan di bawah ini sebagai #1 dalam gambar di bawah, adalah tombol bundar yang muncul mengambang di atas layout.

Anda harus menggunakan tombol aksi mengambang hanya untuk menyatakan aksi utama suatu layar. Misalnya, aksi utama untuk layar utama aplikasi Kontak adalah menambahkan kontak, seperti yang ditampilkan dalam gambar di atas. Tombol aksi mengambang adalah pilihan tepat jika aplikasi Anda mengharuskan suatu aksi yang persisten dan siap tersedia di layar. Hanya satu tombol aksi mengambang yang disarankan untuk setiap layar.

Tombol aksi mengambang menggunakan tipe ikon yang sama yang Anda gunakan untuk tombol dengan ikon, atau untuk tindakan pada bilah aplikasi di bagian atas layar.

Untuk menggunakan tombol aksi mengambang di proyek Android Studio, Anda harus menambahkan pernyataan berikut ke file **build.gradle (Modul: app)** di bagian dependencies :

```
compile 'com.android.support:design:23.4.0'
```

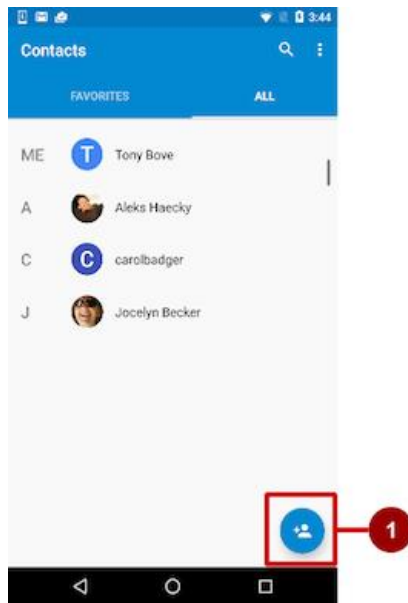
Untuk membuat tombol aksi mengambang, gunakan kelas [FloatingActionButton](#), yang memperluas kelas [ImageButton](#). Anda bisa menambahkan tombol aksi mengambang ke layout XML Anda seperti berikut:

```

<android.support.design.widget.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"

```

```
android:layout_height="wrap_content"  
android:layout_gravity="bottom|end"  
android:layout_margin="@dimen/fab_margin"  
android:src="@drawable/ic_fab_chat_button_white" />
```



Gambar 82. Tombol aksi mengambang

Tombol aksi mengambang secara default berukuran 56 x 56 dp. Ini adalah ukuran default terbaik untuk digunakan, kecuali jika Anda membutuhkan versi yang lebih kecil untuk membuat kontinuitas visual dengan elemen layar lain.

Anda bisa menyetel ukuran *mini* (30 x 40) dengan atribut `app:fabSize` :

```
app:fabSize="mini"
```

Untuk menyetelnya kembali ke ukuran default (56 x 56 dp):

```
app:fabSize="normal"
```

### ▪ Merespons kejadian klik-tombol

Gunakan *event listener* yang disebut `OnClickListener`, yaitu antarmuka kelas `View`, untuk merespons kejadian klik yang timbul saat pengguna mengetuk atau mengklik objek dapat diklik, seperti `Button`, `ImageButton`, atau `FloatingActionButton`. Untuk informasi selengkapnya mengenai event listener,

atau tipe kejadian UI lainnya, baca bagian Kejadian Masukan pada Dokumentasi Developer Android.

### **Menambahkan onClick pada elemen layout**

Untuk mempersiapkan OnClickListener bagi objek yang dapat diklik dalam kode Activity Anda dan menempatkan sebuah metode callback, gunakan atribut `android:onClick` dengan elemen objek yang dapat diklik dalam layout XML. Misalnya:

```
<Button
    android:id="@+id/button_send"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_send"
    android:onClick="sendMessage" />
```

Dalam hal ini, bila pengguna mengeklik tombol, sistem Android akan memanggil metode `sendMessage()` Activity:

```
public void sendMessage(View view) {
    // Do something in response to button click}
```

Metode yang Anda deklarasikan sebagai atribut `android:onClick` harus `public`, mengembalikan `void`, dan mendefinisikan sebuah `View` sebagai satu-satunya parameter (yaitu tampilan yang diklik). Gunakan metode untuk menjalankan tugas atau memanggil metode lainnya sebagai respons terhadap klik tombol.

### **Menggunakan pola desain listener tombol**

Anda juga bisa menangani kejadian klik lewat program dengan menggunakan pola desain tombol-listener (lihat gambar di bawah ini). Untuk informasi selengkapnya mengenai pola desain "listener".

Gunakan event listener `View.OnClickListener`, yaitu antarmuka pada kelas `View` yang terdiri dari metode callback tunggal, `onClick()`. Metode ini dipanggil oleh kerangka kerja Android jika tampilan dipicu oleh interaksi pengguna.

Event listener harus sudah didaftarkan pada tampilan agar dipanggil jika ada kejadian. Ikuti langkah-langkah ini untuk mendaftarkan listener dan menggunakannya (lihat gambar di bawah langkah-langkah):

1. Gunakan metode `findViewById()` dari kelas `View` untuk menemukan tombol dalam file layout XML:

```
Button button = (Button) findViewById(R.id.button_send);
```

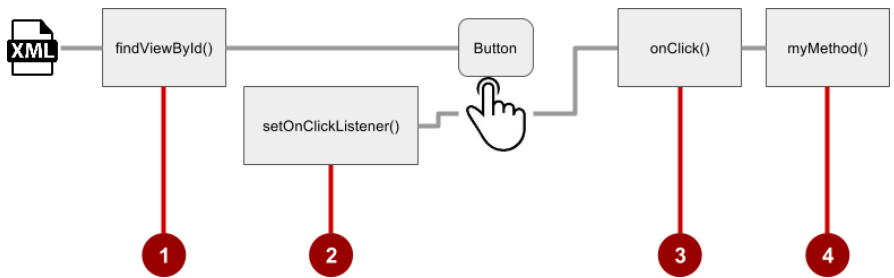
2. Dapatkan objek `View.OnClickListener` yang baru dan daftarkan ke tombol dengan memanggil metode `setOnClickListener()`. Argumen untuk `setOnClickListener()` memerlukan objek yang mengimplementasikan antarmuka `View.OnClickListener`, yang memiliki satu metode: `onClick()`.

```
button.setOnClickListener(new View.OnClickListener() {  
    ...  
}
```

3. Definisikan metode `onClick()` sebagai `public`, kembalikan `void`, dan definisikan `View` sebagai parameternya satu-satunya:

```
public void onClick(View v) {  
    // Do something in response to button click  
}
```

4. Buat metode untuk melakukan sesuatu sebagai respons terhadap klik tombol, misalnya melakukan suatu aksi.



Gambar 83. OnClickListener

Untuk menyetel listener klik lewat program, sebagai ganti dengan atribut `onClick`, sesuaikan kelas `View.OnClickListener` dan ganti penanganan `onClick()` untuk melakukan beberapa aksi, seperti yang ditampilkan di bawah ini:

```
fab.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        } });
```

- Menggunakan antarmuka event listener untuk kejadian lain

Kejadian lainnya bisa terjadi dengan elemen UI, dan Anda bisa menggunakan metode callback yang sudah didefinisikan dalam antarmuka event listener untuk menanganinya. Metode tersebut dipanggil oleh kerangka kerja Android jika tampilan—yang sudah didaftarkan listener—dipicu oleh interaksi pengguna. Oleh karena itu, Anda harus menyetel listener yang sesuai untuk menggunakan metode tersebut. Berikut ini adalah beberapa listener yang tersedia di kerangka kerja Android dan metode callback yang terkait:

- `onClick()` dari `View.OnClickListener`: Menangani kejadian klik di mana pengguna menyentuh dan kemudian melepas area tampilan perangkat yang dikuasai oleh suatu tampilan. Callback `onClick()` tidak memiliki nilai kembalian.
- `onLongClick()` dari `View.OnLongClickListener`: Menangani suatu kejadian di mana pengguna menahan sentuhan pada sebuah tampilan untuk waktu yang lama. Ini akan mengembalikan boolean untuk menunjukkan apakah Anda telah menggunakan kejadian dan tidak boleh dilakukan lebih jauh. Yaitu, kembalikan `true` untuk menunjukkan bahwa Anda telah menangani kejadian dan harus berhenti di sini; kembalikan `false` jika Anda belum menanganinya dan/atau kejadian harus berlanjut ke on-click listener lainnya.
- `onTouchEvent()` dari `View.OnTouchListener`: Menangani segala bentuk kontak sentuh dengan layar termasuk satu atau beberapa gerakan sentuh dan gerakan isyarat, termasuk isyarat menekan, melepas, atau gerakan lain pada layar (dalam batas elemen UI). Suatu `MotionEvent` diteruskan sebagai argumen, yang menyertakan informasi arah, dan mengembalikan boolean untuk menunjukkan apakah listener Anda menggunakan kejadian ini.
- `onFocusChange()` dari `View.OnFocusChangeListener`: Menangani saat fokus menjauh dari tampilan saat ini sebagai hasil interaksi dengan trackball atau tombol navigasi.
- `onKey()` dari `View.OnKeyListener`: Menangani bila sebuah tombol perangkat keras ditekan saat tampilan memiliki fokus.

- Menggunakan kontrol masukan untuk membuat pilihan

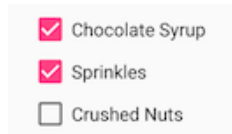
Android menawarkan kontrol masukan siap-buat bagi pengguna untuk memilih satu atau beberapa pilihan:

- Kotak centang: Pilih satu atau beberapa nilai dari serangkaian nilai dengan mengeklik setiap kotak centang nilai.
- Tombol radio: Pilih satu nilai saja dari serangkaian nilai dengan mengeklik tombol "radio" bundar nilai. Jika hanya menyediakan dua atau tiga pilihan, Anda mungkin ingin menggunakan tombol radio untuk pilihan jika memiliki ruang di layout untuk tombol-tombol itu.
- Tombol beralih: Pilih salah satu dari dua atau beberapa keadaan: Tombol beralih biasanya menawarkan dua keadaan yang tampak, seperti "aktif" dan "nonaktif".
- Spinner: Pilih satu nilai dari serangkaian nilai di menu tarik-turun. Hanya satu nilai yang bisa dipilih. Spinner berguna untuk tiga atau beberapa pilihan, dan membutuhkan sedikit ruang di layout Anda.

### c. Kotak centang (CheckBox)

Gunakan kotak centang jika Anda memiliki daftar pilihan dan pengguna boleh memilih berapa pun pilihan, termasuk tanpa pilihan. Setiap kotak centang bersifat independen dari kotak centang lain pada daftar tersebut, sehingga mencentang satu kotak tidak akan menghapus centang pada kotak lainnya. (Jika Anda ingin membatasi pilihan pengguna menjadi satu item saja dari satu rangkaian, gunakan tombol radio.) Pengguna bisa juga menghapus centang pada kotak centang.

Pengguna mengharapkan kotak centang muncul dalam daftar vertikal, seperti daftar kerja, atau bersebelahan secara horizontal jika labelnya pendek.



Gambar 84. Kotak centang muncul dalam daftar vertikal

Setiap kotak centang adalah instance tersendiri dari kelas `CheckBox`. Anda membuat setiap kotak centang menggunakan elemen `CheckBox` dalam layout XML Anda. Untuk membuat beberapa kotak centang dengan orientasi vertikal, gunakan `LinearLayout` vertikal:

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <CheckBox android:id="@+id/checkbox1_chocolate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/chocolate_syrup" />
    <CheckBox android:id="@+id/checkbox2_sprinkles"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/sprinkles" />
    <CheckBox android:id="@+id/checkbox3_nuts"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/crushed_nuts" />

</LinearLayout>
```



Umumnya, program mengambil keadaan kotak centang saat pengguna menyentuh atau mengklik tombol **Submit** atau **Done** pada activity yang sama, yang menggunakan atribut `android:onClick` untuk memanggil metode seperti `onSubmit()` :

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/submit"
    android:onClick="onSubmit"/>
```

Metode callback—`onSubmit()` dalam contoh `Button` di atas—harus `public`, mengembalikan `void`, dan mendefinisikan sebuah `View` sebagai parameter (tampilan yang diklik). Dalam metode ini, Anda bisa menentukan apakah kotak centang dipilih dengan menggunakan metode `isChecked()` (diwarisi dari `CompoundButton`). Metode `isChecked()` akan mengembalikan (boolean) `true` jika ada tanda centang dalam kotak. Misalnya, pernyataan berikut menetapkan nilai boolean dari `true` atau `false` ke `checked` bergantung pada apakah kotak centang itu dicentang:

```
boolean checked = ((CheckBox) view).isChecked()
```

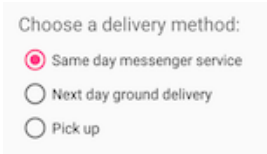
Cuplikan kode berikut menampilkan bagaimana metode `onSubmit()` akan memeriksa untuk mengetahui kotak centang mana yang dipilih, dengan menggunakan `id resource` untuk elemen kotak centang:

```
public void onSubmit(View view) {
    StringBuffer toppings = new
    StringBuffer().append(getString(R.string.toppings_label));
    if (((CheckBox)
    findViewById(R.id.checkbox1_chocolate)).isChecked()) {

    toppings.append(getString(R.string.chocolate_syrup_text));
    }
    if (((CheckBox)
    findViewById(R.id.checkbox2_sprinkles)).isChecked()) {
        toppings.append(getString(R.string.sprinkles_text));
    }
    if (((CheckBox)
    findViewById(R.id.checkbox3_nuts)).isChecked()) {
        toppings.append(getString(R.string.crushed_nuts_text));
    }
    ...
}
```

#### d. Tombol radio (RadioButton)

Gunakan tombol radio jika Anda menggunakan dua atau beberapa opsi yang saling lepas—pengguna harus memilih salah satu di antaranya. (Jika Anda ingin mengaktifkan lebih dari satu pilihan dari rangkaian tersebut, gunakan kotak centang.)



Gambar 85. Tombol radio (RadioButton)

Pengguna mengharapkan tombol radio muncul dalam bentuk daftar vertikal, atau bersebelahan secara horizontal jika labelnya pendek. Setiap tombol radio adalah instance kelas `RadioButton`. Tombol radio umumnya digunakan bersama dalam `RadioGroup`. Bila sejumlah tombol radio berada dalam suatu grup radio, mencentang satu tombol radio akan menghapus semua centang lainnya. Buat setiap tombol radio menggunakan elemen `RadioButton` pada layout XML Anda dalam grup tampilan `RadioGroup`:

```
<RadioGroup
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:layout_below="@id/orderintrotext">
    <RadioButton
        android:id="@+id/sameday"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/same_day_messenger_service"
        android:onClick="onRadioButtonClicked"/>
    <RadioButton
        android:id="@+id/nextday"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/next_day_ground_delivery"
        android:onClick="onRadioButtonClicked"/>
    <RadioButton
        android:id="@+id/pickup"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/pick_up"
        android:onClick="onRadioButtonClicked"/>
</RadioGroup>
```

Gunakan atribut `android:onClick` setiap tombol radio untuk mendeklarasikan metode penanganan kejadian klik bagi tombol radio, yang harus didefinisikan dalam aktivitas yang menjadi host layout ini. Dalam layout di atas, mengeklik tombol radio akan memanggil metode `onRadioButtonClicked()` yang sama dalam aktivitas tersebut, namun Anda bisa membuat metode tersendiri dalam aktivitas dan mendeklarasikannya dalam setiap atribut `android:onClick` tombol radio.

Metode penanganan kejadian klik harus `public`, mengembalikan `void`, dan mendefinisikan sebuah `View` sebagai satu-satunya parameter (tampilan yang diklik). Yang berikut ini menampilkan satu metode, `onRadioButtonClicked()`, untuk semua tombol radio, dengan menggunakan pernyataan `switch case` untuk memeriksa `id` resource bagi elemen tombol radio untuk menentukan tombol radio yang dicentang:

```
public void onRadioButtonClicked(View view) {
    // Check to see if a button has been clicked.
    boolean checked = ((RadioButton) view).isChecked();

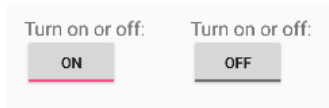
    // Check which radio button was clicked.
    switch(view.getId()) {
        case R.id.sameday:
            if (checked)
                // Same day service
                break;
        case R.id.nextday:
            if (checked)
                // Next day delivery
                break;
        case R.id.pickup:
            if (checked)
                // Pick up
                break;    }
    }
```



**Tip:** Untuk memberi kesempatan pengguna meninjau pilihan tombol radio mereka sebelum aplikasi merespons, Anda bisa mengimplementasikan tombol **Submit** atau **Done** seperti yang ditampilkan sebelumnya bersama kotak centang, dan membuang atribut `android:onClick` dari tombol radio. Kemudian, tambahkan metode `onRadioButtonClicked()` ke atribut `android:onClick` untuk tombol **Submit** atau **Done**.

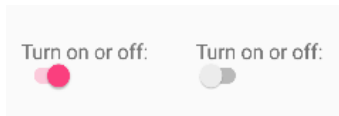
e. Switch dan tombol beralih (ToggleButton)

Kontrol masukan beralih memungkinkan pengguna mengubah setelan di antara dua keadaan. Android menyediakan kelas `ToggleButton`, yang menampilkan tombol timbul dengan "OFF" dan "ON".



Gambar 86. ToggleButton

Contoh beralih menyertakan switch Aktif/Nonaktif untuk Wi-Fi, Bluetooth, dan opsi lain dalam aplikasi Settings. Android juga menyediakan kelas `Switch`, yaitu slider pendek yang terlihat seperti sakelar bulat yang menawarkan dua keadaan (hidup dan mati). Keduanya adalah ekstensi kelas `CompoundButton`.



Gambar 87. Switch on-off

### Menggunakan tombol beralih (`ToggleButton`)

Membuat tombol beralih dengan menggunakan elemen `ToggleButton` dalam layout XML Anda:

```
<ToggleButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/my_toggle"
    android:text="
    android:onClick="onToggleClick"/>
```

**Tip:** Atribut `android:text` tidak menyediakan label teks untuk tombol beralih—tombol beralih selalu menampilkan baik "ON" atau "OFF". Untuk menyediakan label teks di sebelah (atau di atas) tombol beralih, gunakan `TextView` tersendiri.

Untuk merespons ketukan beralih, deklarasikan metode callback `android:onClick` bagi `ToggleButton`. Metode harus didefinisikan dalam aktivitas yang melakukan hosting layout, dan itu harus berupa `public`, mengembalikan `void`, dan mendefinisikan sebuah `View` sebagai satu-satunya parameter (yaitu tampilan yang diklik). Gunakan

CompoundButton.OnCheckedChangeListener() untuk mendeteksi perubahan keadaan beralih. Buat objek CompoundButton.OnCheckedChangeListener dan tetapkan ke tombol dengan memanggil setOnCheckedChangeListener(). Misalnya, metode onToggleClick() akan memeriksa apakah peralihan aktif atau nonaktif, dan menampilkan pesan toast:

```
public void onToggleClick(View view) {
    ToggleButton toggle = (ToggleButton)
    findViewById(R.id.my_toggle);
    toggle.setOnCheckedChangeListener(new

    CompoundButton.OnCheckedChangeListener() {
    public void onCheckedChanged(CompoundButton buttonView,
        boolean isChecked) {
    StringBuffer onOff = new StringBuffer().append("On or off? ");
        if (isChecked) { // The toggle is enabled
            onOff.append("ON ");
        } else { // The toggle is disabled
            onOff.append("OFF ");
        }
        Toast.makeText(getApplicationContext(),
onOff.toString(),
        Toast.LENGTH_SHORT).show();    }
    });
}
```



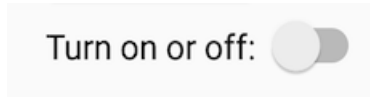
**Tip:** Anda juga bisa mengubah keadaan ToggleButton lewat program dengan menggunakan metode setChecked(boolean). Akan tetapi, perhatikan, metode yang ditetapkan oleh atribut android:onClick() tidak akan dieksekusi dalam kasus ini.

## Menggunakan switch

Switch adalah instance tersendiri dari kelas [Switch](#), yang memperluas kelas CompoundButton seperti ToggleButton. Buat switch beralih dengan menggunakan elemen Switch dalam layout XML Anda:

```
<Switch
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/my_switch"
    android:text="@string/turn_on_or_off"
    android:onClick="onSwitchClick"/>
```

Atribut `android:text` mendefinisikan sebuah string yang muncul di sebelah kiri switch, seperti yang ditampilkan di bawah ini:



Gambar 88. Switch

Untuk merespons ketukan switch, deklarasikan metode callback `android:onClick` bagi Switch—kode pada dasarnya sama dengan kode untuk `ToggleButton`. Metode harus didefinisikan dalam aktivitas yang melakukan hosting layout, dan itu harus berupa `public`, mengembalikan `void`, dan mendefinisikan sebuah `View` sebagai satu-satunya parameter (yaitu tampilan yang diklik).

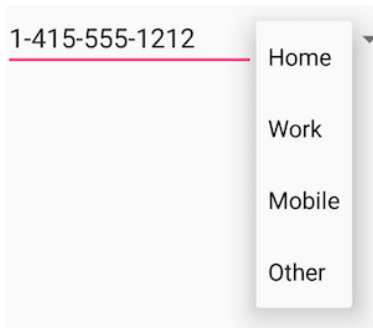
Gunakan `CompoundButton.OnCheckedChangeListener()` untuk mendeteksi perubahan keadaan switch. Buat objek `CompoundButton.OnCheckedChangeListener` dan tetapkan ke tombol dengan memanggil `setOnCheckedChangeListener()`. Misalnya, metode `onSwitchClick()` akan memeriksa apakah switch aktif atau nonaktif, dan menampilkan pesan toast:

```
public void onSwitchClick(View view) {
    Switch aSwitch = (Switch) findViewById(R.id.my_switch);
    aSwitch.setOnCheckedChangeListener(new

CompoundButton.OnCheckedChangeListener() {
    public void onCheckedChanged(CompoundButton buttonView,
                                boolean isChecked) {
        StringBuffer onOff = new StringBuffer().append("On or
off? ");
        if (isChecked) { // The switch is enabled
            onOff.append("ON ");
        } else { // The switch is disabled
            onOff.append("OFF ");
        }
        Toast.makeText(getApplicationContext(),
onOff.toString(),
                                Toast.LENGTH_SHORT).show();
    }
});
}
```

## f. Spinner

Spinner menyediakan cara cepat untuk memilih salah satu dari serangkaian nilai. Menyentuh spinner akan menampilkan daftar tarik-turun dengan semua nilai yang tersedia, yang bisa dipilih oleh pengguna.



Gambar 89. Spinner

Jika Anda memiliki daftar pilihan yang panjang, spinner bisa memanjang melebihi layout, sehingga mengharuskan pengguna untuk menggulirnya. Spinner menggulir secara otomatis, tanpa memerlukan tambahan kode. Akan tetapi, menggulir daftar panjang (seperti daftar negara) tidak disarankan karena akan menyulitkan untuk memilih sebuah item.

Untuk membuat spinner, gunakan kelas `Spinner`, yang membuat tampilan yang menampilkan nilai spinner individual sebagai tampilan anak, dan memungkinkan pengguna memilih salah satu. Ikuti langkah-langkah ini:

- 1) Buat elemen Spinner dalam layout XML Anda, dan tetapkan nilainya menggunakan larik serta `ArrayAdapter`.
- 2) Buat spinner dan adapternya menggunakan kelas `SpinnerAdapter`.
- 3) Untuk mendefinisikan callback pilihan bagi spinner, perbarui `Activity` yang menggunakan spinner untuk mengimplementasikan antarmuka `.AdapterView.OnItemClickListener`.

### Buat elemen UI spinner

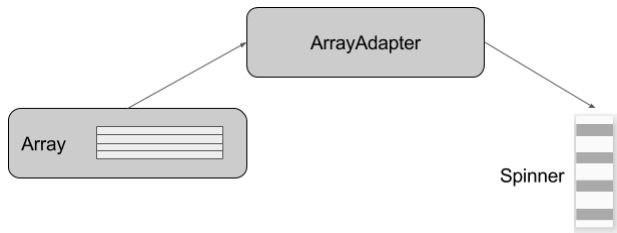
Untuk membuat spinner dalam layout XML Anda, tambahkan elemen `Spinner`, yang menyediakan daftar tarik-turun:

```
<Spinner
    android:id="@+id/label_spinner"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
</Spinner>
```

## Tetapkan nilai spinner

Tambahkan sebuah adapter yang mengisi daftar spinner dengan nilai-nilai. Adapter bagaikan jembatan, atau perantara, antara dua antarmuka yang tidak kompatibel. Misalnya, pembaca kartu memori berfungsi sebagai adapter antara kartu memori dan laptop. Anda memasang kartu memori ke dalam pembaca kartu, dan memasang pembaca kartu ke dalam laptop, sehingga laptop bisa membaca kartu memori.

Pola spinner-adapter mengambil serangkaian data yang Anda tetapkan dan membuat tampilan untuk setiap item dalam rangkaian data tersebut, seperti yang ditampilkan dalam gambar di bawah ini.



Gambar 90. Pola spinner-adapter

Kelas `SpinnerAdapter`, yang mengimplementasikan kelas `Adapter`, memungkinkan Anda mendefinisikan dua tampilan yang berbeda: yang menampilkan nilai data dalam spinner itu sendiri, dan yang menampilkan data dalam daftar tarik-turun saat spinner disentuh atau diklik.

Nilai yang Anda sediakan untuk spinner bisa berasal dari sumber apa pun, tetapi harus disediakan melalui `SpinnerAdapter`, seperti `ArrayAdapter` jika nilainya tersedia dalam larik. Contoh berikut menampilkan larik sederhana yang disebut `labels_array` nilai pra-determinan dalam file `strings.xml`:

```
<string-array name="labels_array">
  <item>Home</item>
  <item>Work</item>
  <item>Mobile</item>
  <item>Other</item>
</string-array>
```



**Tip:** Anda bisa menggunakan `CursorAdapter` jika nilai berasal dari sebuah sumber seperti file tersimpan atau database. Anda akan mengetahui tentang data tersimpan dalam bab lain.



## Buat spinner dan adaptornya

Buat spinner, dan setel listener-nya ke activity yang mengimplementasikan metode callback. Tempat terbaik untuk melakukannya adalah saat tampilan dibuat dalam metode `onCreate()`. Ikuti langkah-langkah ini (lihat metode `onCreate()` lengkap di akhir langkah):

- 1) Tambahkan kode di bawah ke metode `onCreate()`, yang melakukan hal-hal berikut:
- 2) Mendapatkan objek spinner yang Anda tambahkan pada layout menggunakan `findViewById()` untuk menemukannya dengan id-nya (`label_spinner`).
- 3) Menyetel `onItemSelectedListener` ke activity mana pun yang mengimplementasikan callback (`this`) dengan menggunakan metode `setOnItemSelectedListener()`.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // Create the spinner.
    Spinner spinner = (Spinner)
    findViewById(R.id.label_spinner);
    if (spinner != null) {
        spinner.setOnItemSelectedListener(this);
    }
}
```

- 4) Juga di metode `onCreate()`, tambahkan pernyataan yang membuat `ArrayAdapter` dengan larik string:

```
// Create ArrayAdapter using the string array and default
spinner layout.
ArrayAdapter<CharSequence> adapter =
ArrayAdapter.createFromResource(this,
    R.array.labels_array,
    android.R.layout.simple_spinner_item);
```

Seperti yang ditampilkan di atas, Anda menggunakan metode `createFromResource()`, yang memerlukan argumen:

- 5) Activity yang mengimplementasikan callback untuk memproses hasil spinner (`this`)

- 6) Larik (`labels_array`)
- 7) Layout untuk setiap item spinner (`layout.simple_spinner_item`).



**Tip:** Anda harus menggunakan layout default `simple_spinner_item`, kecuali jika Anda ingin mendefinisikan layout sendiri untuk item di spinner.

Tetapkan layout yang harus digunakan adapter untuk menampilkan pilihan daftar spinner dengan memanggil metode `setDropDownViewResource()` dari kelas `ArrayAdapter`. Misalnya, Anda bisa menggunakan `simple_spinner_dropdown_item` sebagai layout Anda:

```
// Specify the layout to use when the list of choices
appears.
adapter.setDropDownViewResource
(android.R.layout.simple_spinner_dropdown_item);
```



**Tip:** Anda harus menggunakan layout default `simple_spinner_dropdown_item`, kecuali jika Anda ingin mendefinisikan layout sendiri untuk penampilan spinner.

- 8) Gunakan `setAdapter()` untuk menerapkan adapter ke spinner:

```
// Apply the adapter to the spinner.
spinner.setAdapter(adapter);
```

Kode lengkap untuk metode `onCreate()` ditampilkan di bawah ini:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // Create the spinner.
    Spinner spinner = (Spinner)
    findViewById(R.id.label_spinner);
    if (spinner != null) {
        spinner.setOnItemSelectedListener(this);
    }
    // Create ArrayAdapter using the string array and default
    spinner layout.
```

```

        ArrayAdapter<CharSequence> adapter =
        ArrayAdapter.createFromResource(this,
            R.array.labels_array,
            android.R.layout.simple_spinner_item);

        // Specify the layout to use when the list of choices
        appears.adapter.setDropDownViewResource
        (android.R.layout.simple_spinner_dropdown_item);

        // Apply the adapter to the spinner.
        if (spinner != null) {
            spinner.setAdapter(adapter);
        }
    }
}

```

## Implementasikan antarmuka `OnItemSelectedListener` dalam Activity

Untuk mendefinisikan callback pilihan spinner, perbarui Activity yang menggunakan spinner untuk mengimplementasikan antarmuka `AdapterView.OnItemSelectedListener`.

```

public class MainActivity extends AppCompatActivity implements
    AdapterView.OnItemSelectedListener {

```

Android Studio secara otomatis mengimpor widget `AdapterView`. Implementasikan antarmuka `AdapterView.OnItemSelectedListener` untuk mendapatkan metode callback `onItemSelected()` dan `onNothingSelected()` yang akan digunakan bersama objek spinner.

Saat pengguna memilih satu item dari daftar tarik-turun spinner, inilah yang akan terjadi dan cara mengambil item tersebut:

1. Objek Spinner menerima kejadian `on-item-selected`.
2. Kejadian ini memicu panggilan metode callback `onItemSelected()` dari antarmuka `AdapterView.OnItemSelectedListener`.
3. Ambil item yang dipilih dalam menu spinner dengan menggunakan metode `getItemAtPosition()` kelas `AdapterView`:

```

public void onItemSelected(AdapterView<?> adapterView, View
view, int pos, long id) {
    String spinner_item =
    adapterView.getItemAtPosition(pos).toString();
}

```

Argumen untuk `onItemSelected()` adalah seperti berikut:

<code>parent AdapterView</code>	AdapterView di mana pemilihan terjadi
<code>view View</code>	Tampilan dalam AdapterView yang diklik
<code>int pos</code>	Posisi tampilan dalam adapter
<code>long id</code>	ID baris item yang dipilih

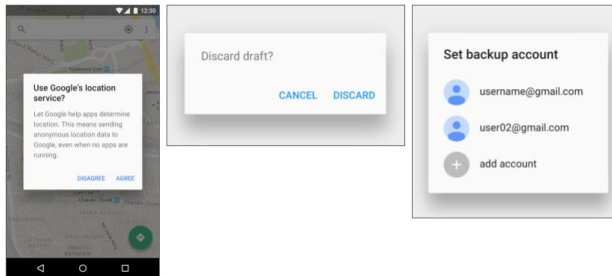
4. Implementasikan/ganti metode callback `onNothingSelected()` dari antarmuka `AdapterView.OnItemSelectedListener` untuk melakukan sesuatu jika tidak ada yang dipilih.

## Menggunakan dialog dan picker

Dialog adalah jendela yang muncul di atas tampilan atau mengisi tampilan, menyela alur activity. Dialog menginformasikan pengguna tentang tugas tertentu dan mungkin mengandung informasi penting, memerlukan keputusan, atau melibatkan beberapa tugas.

Misalnya, Anda ingin secara khusus menggunakan dialog untuk menampilkan peringatan yang mengharuskan pengguna menyetak sebuah tombol untuk membuat keputusan, seperti **OK** atau **Cancel**. Dalam gambar di bawah ini, sisi kiri menampilkan peringatan dengan tombol **Disagree** dan **Agree**, sedangkan bagian tengah menampilkan peringatan dengan tombol **Cancel** dan **Discards**.

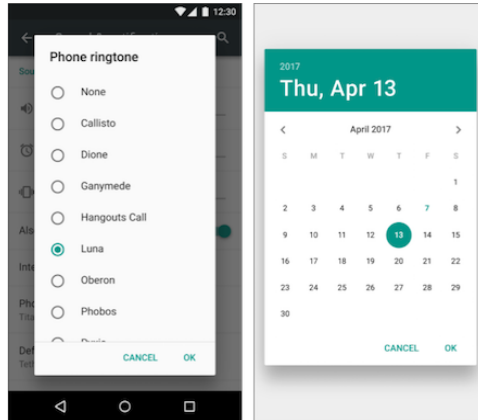
Anda juga bisa menggunakan dialog untuk menyediakan pilihan dalam gaya tombol radio, seperti yang ditampilkan pada sisi kanan gambar di bawah ini.



Gambar 91. Dialog

Kelas dasar untuk semua komponen dialog adalah `Dialog`. Ada sejumlah subkelas `Dialog` yang berguna untuk memperingatkan pengguna atas suatu kondisi, menampilkan status atau perkembangan, menampilkan informasi pada perangkat sekunder, atau memilih atau mengonfirmasi pilihan, seperti yang ditampilkan pada sisi kiri gambar di bawah ini. Android SDK juga menyediakan

subkelas dialog yang siap digunakan seperti "picker" untuk memilih waktu atau tanggal, seperti yang ditampilkan pada sisi kanan gambar di bawah ini. Picker memungkinkan pengguna untuk memasukkan informasi dalam format yang sudah ditentukan dan konsisten, yang mengurangi kemungkinan kesalahan masukan.



Gambar 92. Picker

Dialog selalu menjaga fokus sampai ditutup atau aksi yang diperlukan telah dilakukan.

**Tip:** Praktik terbaik menyarankan untuk menggunakan sedikit dialog karena mengganggu alur kerja pengguna. Kelas Dialog adalah kelas dasar untuk dialog, namun Anda harus menghindari membuat instance Dialog secara langsung, kecuali jika Anda sedang membuat dialog khusus. Untuk dialog Android standar, gunakan salah satu subkelas berikut:

- **AlertDialog:** Dialog yang bisa menampilkan judul, hingga tiga tombol, daftar item yang dapat dipilih, atau layout khusus.
- **DatePickerDialog** atau **TimePickerDialog:** Dialog berisi UI yang sudah didefinisikan, yang memungkinkan pengguna memilih tanggal atau waktu.

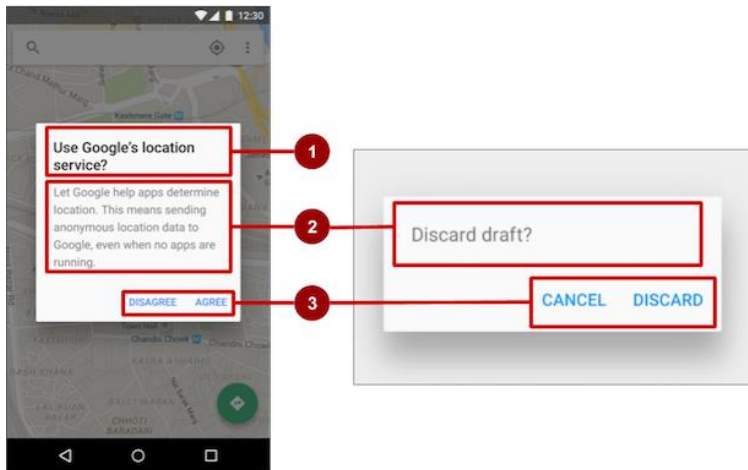


## Menampilkan dialog peringatan

Peringatan adalah interupsi penting, memerlukan balasan atau tindakan, yang memberi tahu pengguna tentang situasi saat peringatan itu terjadi, atau aksi *sebelum* peringatan itu muncul (seperti saat membuang draf). Anda bisa menyediakan tombol dalam peringatan untuk membuat keputusan. Misalnya, sebuah dialog peringatan mungkin mengharuskan pengguna untuk mengeklik **Continue** setelah membacanya, atau memberi pilihan kepada pengguna untuk menyetujui suatu aksi dengan mengeklik tombol positif (seperti **OK** atau

**Accept**), atau untuk tidak menyetujui dengan mengklik tombol negatif (seperti **Disagree** atau **Cancel**).

Gunakan subkelas `AlertDialog` dari kelas `Dialog` untuk menampilkan dialog standar suatu peringatan. Kelas `AlertDialog` memungkinkan Anda untuk membangun berbagai desain dialog. Sebuah dialog peringatan bisa memiliki region berikut (lihat diagram di bawah ini):



Gambar 93. Subkelas `AlertDialog`

1. **Judul:** Judul bersifat opsional. Sebagian besar peringatan tidak memerlukan judul. Jika Anda bisa meringkas keputusan dalam satu atau dua kalimat dengan mengajukan pertanyaan (seperti, "Buang draf?") Atau membuat pernyataan yang berkaitan dengan tombol aksi (seperti, "Klik OK untuk melanjutkan"), tidak perlu repot-repot memikirkan judul. Gunakan judul jika situasinya berisiko-tinggi, seperti potensi hilangnya konektivitas atau data, dan area materi ditempati oleh pesan terperinci, daftar, atau layout khusus.
2. **Area materi:** Area materi bisa menampilkan pesan, daftar, atau layout khusus lainnya.
3. **Tombol aksi:** Anda sebaiknya tidak menggunakan lebih dari tiga tombol aksi dalam dialog, dan kebanyakan hanya memiliki dua tombol.

## Membangun `AlertDialog`

Kelas `AlertDialog.Builder` menggunakan pola desain *builder*, yang mempermudah dalam membangun objek dari kelas yang membutuhkan banyak atribut serta atribut opsional dan karena itu memerlukan banyak parameter untuk

dibangun. Tanpa pola ini, Anda tentunya harus membuat konstruktor untuk kombinasi atribut yang diperlukan dan atribut opsional; dengan pola ini, kode lebih mudah dibaca dan dijaga. Untuk informasi selengkapnya tentang pola desain builder, lihat Pola builder

Gunakan `AlertDialog.Builder` untuk membangun dialog peringatan standar dan menyetel atribut pada dialog. Gunakan `setTitle()` untuk menyetel judulnya, `setMessage()` untuk menyetel pesannya, dan `setPositiveButton()` serta `setNegativeButton()` untuk menyetel tombolnya.

**Catatan:** Jika `AlertDialog.Builder` tidak dikenali saat memasukkannya, Anda mungkin perlu menambahkan pernyataan import berikut ke `MainActivity.java`:

```
import android.content.DialogInterface
import android.support.v7.app.AlertDialog
```

Yang berikut ini akan membuat objek dialog (`myAlertBuilder`) dan menyetel judul (resource string yang disebut `alert_title`) dan pesan (resource string yang disebut `alert_message`):

```
AlertDialog.Builder myAlertBuilder = new
AlertDialog.Builder(MainActivity.this);
myAlertBuilder.setTitle(R.string.alert_title);
myAlertBuilder.setMessage(R.string.alert_message);
```

## Menyetel tombol tindakan untuk dialog peringatan

Gunakan metode `setPositiveButton()` dan `setNegativeButton()` dari kelas `AlertDialog.Builder` untuk menyetel tombol tindakan bagi dialog peringatan. Metode ini memerlukan judul tombol (disediakan oleh resource string) dan kelas `DialogInterface.OnClickListener` yang mendefinisikan aksi yang diambil bila pengguna menekan tombol:

```
myAlertBuilder.setPositiveButton("OK", new
DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog, int which) {
// User clicked OK button.
}
});
myAlertBuilder.setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog, int which) {
```

```
// User clicked the CANCEL button.  
    }  
});
```

Anda bisa menambahkan salah satu saja dari setiap tipe tombol ke AlertDialog. Misalnya, Anda tidak bisa memiliki lebih dari satu tombol "positif".



**Tip:** Anda juga bisa menyetel tombol "netral" dengan `setNeutralButton()`. Tombol netral muncul di antara tombol positif dan tombol negatif. Gunakan tombol netral, seperti "Remind me later" jika Anda ingin pengguna dapat menghilangkan dialog dan memutuskan nanti

## Menampilkan dialog

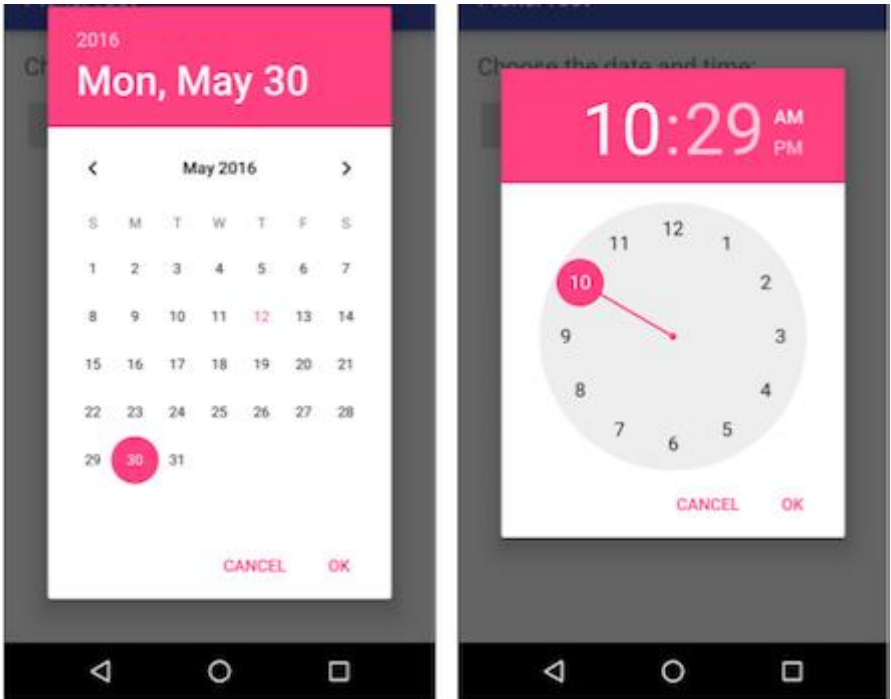
Untuk menampilkan dialog, panggil metode `show()` :

```
AlertDialog.show()
```



### g. Picker Tanggal Dan Waktu

Android menyediakan dialog yang siap digunakan, disebut *picker*, untuk memilih waktu atau tanggal. Gunakan picker untuk memastikan pengguna Anda memilih waktu atau tanggal yang valid, yang diformat dengan benar dan disesuaikan dengan lokal pengguna. Setiap picker menyediakan kontrol untuk memilih setiap bagian waktu (jam, menit, AM/PM) atau tanggal (bulan, tanggal, tahun).



Gambar 94. Picker Tanggal Dan Waktu

Saat menampilkan picker, Anda harus menggunakan instance `DialogFragment`, sebuah subkelas `Fragment`, yang menampilkan jendela dialog yang mengambang di atas jendela aktivitasnya. *Fragmen* adalah perilaku atau bagian antarmuka pengguna dalam suatu aktivitas. *Fragmen* seperti aktivitas mini di dalam aktivitas utama, dengan daur hidup individualnya sendiri. Suatu fragmen menerima kejadian masukannya sendiri, dan Anda bisa menambahkan atau membuangnya saat aktivitas utama berjalan. Anda mungkin mengombinasikan beberapa fragmen dalam suatu aktivitas tunggal untuk membangun antarmuka pengguna multipanel, atau menggunakan kembali suatu

fragmen dalam beberapa activity. Untuk mengetahui tentang fragmen, lihat Fragmen dalam Panduan API.

Salah satu manfaat menggunakan fragmen untuk picker adalah Anda bisa mengisolasi bagian kode untuk mengelola tanggal dan waktu setelah pengguna memilihnya dari picker. Anda juga bisa menggunakan DialogFragment untuk mengelola daur hidup dialog.



**Tip:** Manfaat lain menggunakan fragmen untuk picker adalah Anda bisa mengimplementasikan konfigurasi layout yang berbeda, seperti dialog dasar pada tampilan berukuran handset atau bagian layout yang disematkan pada tampilan yang besar

## Menambahkan fragmen

Untuk menambahkan fragmen bagi picker tanggal, buat fragmen kosong (**DatePickerFragment**) tanpa XML layout, dan tanpa metode pabrik atau callback antarmuka:

1. Luaskan **app > java > com.example.android.DateTimePickers** dan pilih **MainActivity**.
2. Pilih **File > New > Fragment > Fragment (Blank)**, dan beri nama fragmen **DatePickerFragment**. Kosongkan ketiga opsi kotak centang sehingga Anda *tidak* membuat XML layout, *jangan* sertakan metode pabrik fragmen, dan *jangan* sertakan callback antarmuka. Anda tidak perlu membuat layout untuk picker standar. Klik **Finish** untuk membuat fragmen.

## Memperluas DialogFragment untuk picker

Langkah berikutnya adalah membuat picker standar dengan listener. Ikuti langkah-langkah ini:

1. Edit definisi kelas `DatePickerFragment` untuk memperluas `DialogFragment`, dan implementasikan `DatePickerDialog.OnDateSetListener` untuk membuat picker tanggal standar dengan listener:

```
public class DatePickerFragment extends DialogFragment
implements DatePickerDialog.OnDateSetListener {
    ...
}
```

Android Studio secara otomatis menambahkan yang berikut ini dalam blok import di bagian atas:

```
import android.app.DatePickerDialog.OnDateSetListener;
import android.support.v4.app.DialogFragment;
```

2. Android Studio juga menampilkan ikon bola lampu merah di batas kiri, mengarahkan Anda untuk mengimplementasikan metode. Klik ikon dan, dengan **onDateSet** sudah dipilih serta opsi "Insert @Override" sudah dicentang, klik **OK** untuk membuat metode `onDateSet()`.

Kemudian Android Studio secara otomatis menambahkan yang berikut ini dalam blok import di bagian atas:

```
import android.widget.DatePicker
```

## Ganti `onCreateView()` dengan `onCreateDialog()`:

```
@Override
public Dialog onCreateDialog(Bundle savedInstanceState) {
}
```

Saat Anda memperluas `DialogFragment`, Anda harus mengganti metode callback `onCreateDialog()`, daripada `onCreateView`. Gunakan metode callback versi Anda untuk melakukan inisialisasi `year`, `month`, dan `day` untuk picker tanggal.

## Menyetel default dan mengembalikan picker

Untuk menyetel tanggal default dalam picker dan mengembalikannya sebagai objek yang bisa Anda gunakan, ikuti langkah-langkah ini:

1. Tambahkan kode berikut pada metode `onCreateDialog()` untuk menyetel tanggal default bagi picker:

```
// Use the current date as the default date in the picker.
final Calendar c = Calendar.getInstance();
int year = c.get(Calendar.YEAR);
int month = c.get(Calendar.MONTH);
int day = c.get(Calendar.DAY_OF_MONTH);
```

Saat Anda memasukkan `Calendar`, Anda diberi pilihan pustaka Kalender yang akan diimpor. Pilih ini:

```
import java.util.Calendar
```

Kelas Calendar menyetel tanggal default sebagai tanggal saat ini—kelas ini mengonversi antara instan tertentu dan serangkaian bidang kalender seperti YEAR, MONTH, DAY\_OF\_MONTH, HOUR, dan seterusnya. Kalender bersifat sensitif-lokal, dan metode kelasnya getInstance() mengembalikan objek Calendar yang bidang kalendernya telah diinisialisasi dengan tanggal dan waktu saat ini.

2. Tambahkan pernyataan berikut pada akhir metode untuk membuat instance picker tanggal yang baru dan mengembalikannya:

```
return new DatePickerDialog(getActivity(), this, year, month, day)
```

## Menampilkan picker

Pada Aktiviti Utama, Anda perlu membuat metode untuk menampilkan picker tanggal. Ikuti langkah-langkah ini:

1. Buat metode untuk membuat instance fragmen dialog picker tanggal:

```
public void showDatePickerDialog(View v) {
    DialogFragment newFragment = new
    DatePickerFragment();
    newFragment.show(getSupportFragmentManager(),
    "datePicker");
}
```

2. Kemudian Anda bisa menggunakan showDatePickerDialog() dengan atribut android:onClick untuk tombol atau kontrol masukan lainnya:

```
android:id="@+id/button_date"
    ...
    android:onClick="showDatePickerDialog"/>
```

## Memproses pilihan picker pengguna

Metode onDateSet() secara otomatis dipanggil saat pengguna membuat pilihan dalam picker tanggal, sehingga Anda bisa menggunakan metode ini untuk melakukan sesuatu dengan tanggal yang dipilih tersebut. Ikuti langkah-langkah ini:

1. Agar kode lebih terbaca, ubah parameter metode `onDataSet()` dari `int i, int i1, dan int i2` ke `int year, int month, dan int day`:

```
public void onDataSet(DatePicker view, int year, int month,
int day) {
```

2. Buka **MainActivity** dan tambahkan penanda metode `processDatePickerResult()` yang mengambil `year, month, dan day` sebagai argumen:

```
public void processDatePickerResult(int year, int month,
int day) {
}
```

3. Tambahkan kode berikut pada metode `processDatePickerResult()` untuk mengonversi `month, day, dan year` dan untuk memisahkan string:

```
String month_string = Integer.toString(month+1);
String day_string = Integer.toString(day);
String year_string = Integer.toString(year);
```



**Catatan:** Integer `month` yang dikembalikan oleh picker tanggal mulai menghitung dari 0 untuk Januari, sehingga Anda perlu menambahkan 1 untuk mulai menampilkan bulan mulai dari 1.

4. Tambahkan yang berikut ini setelah kode di atas untuk menggabungkan tiga string dan menyertakan garis miring untuk format tanggal A.S.:

```
String dateMessage = (month_string + "/" +
day_string + "/" + year_string);
```

5. Tambahkan yang berikut ini setelah pernyataan di atas untuk menampilkan pesan Toast:

```
Toast.makeText(this, "Date: " + dateMessage,
Toast.LENGTH_SHORT).show();
```

6. Ekstrak string hard-code `"Date: "` menjadi resource string bernama `date`. Hal ini secara otomatis mengganti string hard-code dengan `getString(R.string.date)`. Kode untuk metode `processDatePickerResult()` sekarang harus tampak seperti ini:

```
public void processDatePickerResult(int year, int month,
int day) {
String month_string = Integer.toString(month + 1);
```

```

    String day_string = Integer.toString(day);
    String year_string = Integer.toString(year);
    // Assign the concatenated strings to dateMessage.
    String dateMessage = (month_string + "/" +
                          day_string + "/" +
                          year_string);
    Toast.makeText(this, getString(R.string.date) +
                  dateMessage,
                  Toast.LENGTH_SHORT).show();
}

```

Buka **DatePickerFragment**, dan tambahkan yang berikut ini ke metode `onDataSet()` untuk memanggil metode `processDatePickerResult()` di `MainActivity` dan meneruskan `year, month, dan day` ke sana:

```

public void onDataSet(DatePicker view, int year, int
month, int day) {
    // Set the activity to the Main Activity.
    MainActivity activity = (MainActivity)
getActivity();
    // Invoke Main Activity's
processDatePickerResult() method.
    activity.processDatePickerResult(year, month,
day);
}

```

Gunakan `getActivity()` yang, jika digunakan dalam suatu fragmen, akan mengembalikan `activity` yang saat ini dikaitkan dengan fragmen. Anda memerlukannya karena Anda tidak bisa memanggil suatu metode dalam `MainActivity` tanpa konteks `MainActivity` (Anda seharusnya menggunakan `intent`, seperti yang Anda pelajari pada pembelajaran sebelumnya). `Activity` mewarisi konteks, sehingga Anda bisa menggunakannya sebagai konteks untuk memanggil metode (seperti dalam `activity.processDatePickerResult()`).

## Menggunakan prosedur yang sama untuk picker waktu

Ikuti prosedur yang sama yang diuraikan di atas untuk picker tanggal:

1. Tambahkan fragmen kosong yang disebut **TimePickerFragment** yang memperluas `DialogFragment` dan mengimplementasikan `TimePickerDialog.OnTimeSetListener`:

```
public class TimePickerFragment extends DialogFragment
implements TimePickerDialog.OnTimeSetListener {
```

2. Tambahkan dengan `@Override` metode `onTimeSet()` kosong:

Android Studio juga menampilkan ikon bola lampu merah di batas kiri, yang meminta konfirmasi Anda untuk mengimplementasikan metode. Klik ikon dan, dengan **onTimeSet** sudah dipilih serta opsi "Insert @Override" sudah dicentang, klik **OK** untuk membuat metode `onTimeSet()`. Kemudian Android Studio secara otomatis menambahkan yang berikut ini dalam blok import di bagian atas:

```
import android.widget.TimePicker
```

3. Gunakan `onCreateDialog()` untuk melakukan inisialisasi waktu dan mengembalikan dialog:

```
public Dialog onCreateDialog(Bundle savedInstanceState) {
    // Use the current time as the default values for the
    picker.
    final Calendar c = Calendar.getInstance();
    int hour = c.get(Calendar.HOUR_OF_DAY);
    int minute = c.get(Calendar.MINUTE);

    // Create a new instance of TimePickerDialog and
    return it.
    return new TimePickerDialog(getActivity(), this,
    hour, minute,
        DateFormat.is24HourFormat(getActivity()));
}
```

4. Tampilkan picker: Buka **MainActivity** dan buat metode untuk membuat instance fragmen dialog picker tanggal:

```
public void showDatePickerDialog(View v) {
    DialogFragment newFragment = new DatePickerFragment();
    newFragment.show(getSupportFragmentManager(),
    "datePicker");
}
```

Gunakan `showDatePickerDialog()` dengan atribut `android:onClick` untuk tombol atau kontrol masukan lainnya:

```
android:id="@+id/button_date"
...
android:onClick="showDatePickerDialog"/>
```

5. Buat metode `processTimePickerResult()` di **MainActivity** untuk memproses hasil pemilihan dari picker waktu:

```

public void processTimePickerResult(int hourOfDay, int
minute) {
    // Convert time elements into strings.
    String hour_string = Integer.toString(hourOfDay);
    String minute_string = Integer.toString(minute);
    // Assign the concatenated strings to timeMessage.
    String timeMessage = (hour_string + ":" +
minute_string);
    Toast.makeText(this, getString(R.string.time) +
timeMessage,
Toast.LENGTH_SHORT).show();
}

```

6. Gunakan `onTimeSet()` untuk mendapatkan waktu dan meneruskannya ke metode `processTimePickerResult()` di `MainActivity`:

```

public void onTimeSet(TimePicker view, int hourOfDay, int
minute) {
    // Set the activity to the Main Activity.
    MainActivity activity = (MainActivity) getActivity();
    // Invoke Main Activity's processDatePickerResult()
method.
    activity.processTimePickerResult(hourOfDay, minute);
}

```

#### h. Mengenali Isyarat

Isyarat sentuh terjadi saat seorang pengguna meletakkan satu atau beberapa jari di atas layar sentuh, dan aplikasi Anda akan menerjemahkan pola sentuhan sebagai isyarat khusus, seperti sentuhan lama, ketuk dua kali, gesek cepat, atau guliran.

Android menyediakan berbagai kelas dan metode untuk membantu Anda membuat dan mendeteksi isyarat. Meskipun aplikasi Anda seharusnya tidak bergantung pada isyarat sentuhan untuk perilaku dasar (karena isyarat mungkin tidak tersedia untuk semua pengguna dalam semua konteks), menambahkan interaksi berbasis sentuhan ke aplikasi Anda bisa sangat meningkatkan daya guna dan daya tariknya. Untuk menyediakan pengalaman yang intuitif dan konsisten bagi pengguna, aplikasi Anda seharusnya mengikuti konvensi Android yang diterima untuk isyarat sentuh.

#### **Mendeteksi isyarat umum**

Jika aplikasi Anda menggunakan isyarat umum seperti ketuk dua kali, tekan lama, lempar, dan seterusnya, Anda bisa memanfaatkan kelas `GestureDetector` untuk mendeteksi isyarat umum. Gunakan



GestureDetectorCompat, yang disediakan sebagai implementasi kompatibilitas kelas GestureDetector kerangka kerja, yang menjamin perilaku menggulir titik fokus terbaru dari Jellybean MRI pada semua versi platform. Kelas ini seharusnya hanya digunakan dengan kejadian gerakan yang dilaporkan untuk perangkat sentuh—jangan gunakan untuk trackball atau kejadian perangkat keras lainnya.

GestureDetectorCompat memungkinkan Anda mendeteksi isyarat umum tanpa harus memproses kejadian sentuh individual sendiri. GestureDetectorCompat mendeteksi beragam isyarat dan kejadian dengan menggunakan objek MotionEvent, yang melaporkan gerakan oleh jari (atau mouse, pena, atau trackball).

Cuplikan berikut menampilkan cara Anda menggunakan kelas GestureDetectorCompat dan GestureDetector.SimpleOnGestureListener.

1. Untuk menggunakan GestureDetectorCompat, buat instance (mDetector dalam cuplikan di bawah) dari kelas GestureDetectorCompat, dengan menggunakan metode onCreate() dalam activity (seperti MainActivity):

```
public class MainActivity extends Activity {
    private GestureDetectorCompat mDetector;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mDetector = new GestureDetectorCompat(this, new
            MyGestureListener());
        ...
    }
}
```

Saat Anda membuat instance objek GestureDetectorCompat, salah satu parameter yang diambil adalah kelas yang harus Anda buat—MyGestureListener dalam cuplikan di atas—yang melakukan salah satu hal berikut:

- mengimplementasikan antarmuka GestureDetector.OnGestureListener, untuk mendeteksi semua isyarat standar, atau
- memperluas kelas GestureDetector.SimpleOnGestureListener, yang bisa Anda gunakan untuk memproses sedikit isyarat saja dengan mengganti metode yang Anda perlukan. Beberapa metode yang disediakan antara lain onDown(), onLongPress(), onFling(), onScroll(), dan onSingleTapUp().

1. Buat kelas `MyGestureListener` sebagai activity tersendiri (**MyGestureListener**) untuk memperluas `GestureDetector.SimpleOnGestureListener`, dan mengganti metode `onFling()` serta `onDown()` untuk menampilkan pernyataan log tentang kejadian:

```
class MyGestureListener extends
GestureDetector.SimpleOnGestureListener {
private static final String DEBUG_TAG = "Gestures";
@Override
public boolean onDown(MotionEvent event) {
    Log.d(DEBUG_TAG, "onDown: " + event.toString());
    return true; }
@Override
public boolean onFling(MotionEvent event1, MotionEvent
event2,
float velocityX, float velocityY) {
    Log.d(DEBUG_TAG, "onFling: " +
event1.toString()+event2.toString());
    return true; }
}
```

2. Untuk mencegah kejadian sentuh, gantikan callback `onTouchEvent()` dari kelas `GestureDetectorCompat` di **MainActivity**:

```
@Override
public boolean onTouchEvent(MotionEvent event){
    this.mDetector.onTouchEvent(event);
    return super.onTouchEvent(event);}
```



## YANG HARUS ANDA KETAHUI

Untuk praktik ini Anda harus bisa:

- 1) Membuat aplikasi Hello World dengan Android Studio.
- 2) Menjalankan aplikasi di emulator atau perangkat.
- 3) Mengimplementasikan TextView dalam layout untuk aplikasi.
- 4) Membuat dan menggunakan resource string.
- 5) Mengonversi dimensi layout ke resource.

## YANG AKAN ANDA BUTUHKAN

Untuk materi dan praktik ini, Anda akan membutuhkan:

- 1) Komputer Mac, Windows, atau Linux. Tool Android Studio untuk persyaratan sistem terbaru.
- 2) Akses internet atau cara alternatif untuk memuat Android Studio dan pemasangan Java terbaru ke komputer Anda.
- 3) Smartphone sebagai perangkat untuk pengujian secara nyata aplikasi yang telah dibuat.

## YANG AKAN ANDA PELAJARI

Anda akan belajar:

- 1) Menggunakan kode XML untuk menambahkan beberapa elemen TextView.
- 2) Menggunakan kode XML untuk menentukan tampilan bergulir.
- 3) Menampilkan teks bentuk bebas dengan beberapa tag pemformatan HTML.
- 4) Menata gaya warna latar belakang dan warna teks TextView.
- 5) Menyertakan tautan web di teks.
- 6) Menambahkan item menu ke menu opsi.
- 7) Menambahkan ikon untuk item di menu opsi.
- 8) Menyetel item menu untuk ditampilkan di bilah tindakan.
- 9) Menambahkan handler kejadian untuk klik item menu.

## YANG HARUS ANDA LAKUKAN

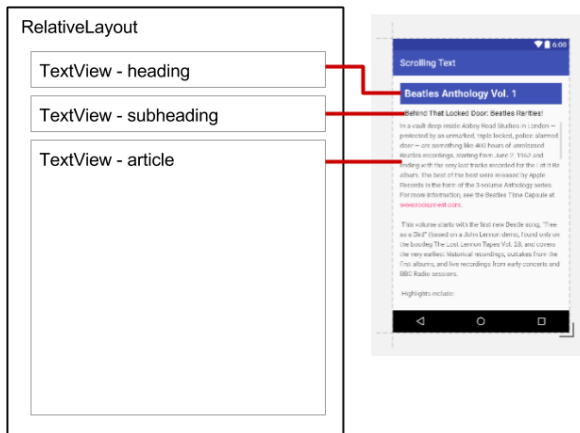
- 1) Membuat aplikasi Scrolling Text.
- 2) Menambahkan dua elemen TextView untuk heading dan subheading artikel.
- 3) Menggunakan gaya dan warna TextAppearance untuk heading dan subheading artikel.
- 4) Menggunakan tag HTML dalam string teks untuk mengontrol pemformatan.
- 5) Menggunakan atribut lineSpacingExtra untuk menambahkan spasi baris agar meningkatkan keterbacaan.
- 6) Menambahkan ScrollView ke layout untuk mengaktifkan pengguliran elemen TextView.
- 7) Menambahkan atribut autoLink untuk mengaktifkan URL di teks agar aktif dan bisa diklik.
- 8) Menambahkan item menu ke menu opsi.
- 9) Menambahkan ikon untuk item menu agar tampil di bilah tindakan.
- 10) Menghubungkan klik item menu ke handler kejadian yang memproses kejadian klik.

## RINGKASAN APLIKASI

Aplikasi Scrolling Text memperagakan komponen UI ScrollView. ScrollView adalah ViewGroup yang dalam contoh ini berisi TextView. ScrollView menunjukkan halaman teks panjang, dalam hal ini ulasan album musik, yang dapat digulir pengguna secara vertikal untuk dibaca dengan mengusap layar ke atas dan ke bawah. Bilah gulir muncul di margin kanan. Aplikasi ini menunjukkan cara menggunakan teks berformat dengan tag HTML minimal untuk mengubah teks menjadi cetak tebal atau miring, dan dengan karakter baris baru untuk memisahkan paragraf. Anda juga bisa menyertakan tautan web aktif dalam teks.

## Latihan 4.1. Menambahkan beberapa tampilan teks

Dalam praktik ini Anda akan membuat proyek Android untuk aplikasi Scrolling Text, menambahkan TextView ke layout untuk judul dan subjudul artikel, dan mengubah elemen TextView "Hello World" yang sudah ada untuk menunjukkan artikel yang panjang. Gambar di bawah ini adalah diagram layout.



Anda akan membuat semua perubahan ini dalam kode XML dan di file `strings.xml`. Anda akan mengedit kode XML untuk layout di panel **Text**, yang ditunjukkan dengan mengklik tab **Text** tab, bukan dengan mengklik tab **Design** untuk panel Design. Beberapa perubahan pada elemen dan atribut UI lebih mudah dibuat langsung pada panel Text menggunakan kode sumber XML.

### Latihan 1: Membuat Project dan Elemen TextView

- Di Android Studio, buat proyek baru dengan parameter berikut:

Atribut	Nilai
Application Name	Scrolling Text
Company Name	android.example.com (atau domain Anda sendiri)
Phone and Tablet Minimum SDK	API15: Android 4.0.3 IceCreamSandwich
Template	Empty Activity
Generate Layout File checkbox	Dicentang

- Pada folder **app > res > layout**, buka file **activity\_main.xml**, dan klik tab **Text** untuk melihat kode XML jika belum dipilih.

Di bagian atas, atau *root*, hierarki tampilan adalah ViewGroup yang disebut RelativeLayout. Seperti ViewGroup lainnya, RelativeLayout adalah tampilan yang berisi tampilan lainnya. Selain itu, tampilan ini juga memungkinkan Anda untuk memosisikan Tampilan anak relatif terhadap satu sama lain atau relatif terhadap RelativeLayout parent itu sendiri. Elemen TextView "Hello World" default yang dibuat oleh template Empty Layout adalah Tampilan anak dalam grup tampilan RelativeLayout. Untuk informasi selengkapnya tentang cara menggunakan RelativeLayout.

- Tambahkan elemen TextView di atas TextView "Hello World". Saat Anda memasukkan `<TextView` to start a TextView, Android Studio automatically adds the ending `>`, which is shorthand for `</TextView>`. Tambahkan atribut-atribut berikut ke TextView:

Atribut #1 TextView	Nilai
android:id	"@+id/article_heading"
layout_width	"match_parent"
layout_height	"wrap_content"
android:background	"@color/colorPrimary"
android:textColor	"@android:color/white"
android:padding	"10dp"
android:textAppearance	"@android:style/TextAppearance.Large"
android:textStyle	"bold"
android:text	"Article Title"

- **Tip:** Atribut untuk menata gaya teks dan latar belakang dirangkul dalam Dokumentasi kelas TextView.
- Ekstrak resource string untuk string yang di-hardcode "Article Title" atribut `android:text` dalam TextView untuk membuat entri untuknya di **strings.xml**
- Letakkan kursor di string yang di-hardcode, tekan Alt-Enter (Option-Enter di Mac), dan pilih **Extract string resource**. Lalu edit nama resource untuk nilai string ke **article\_title**.

- Ekstrak resource dimensi untuk string "10dp" hard-code atribut `android:padding` dalam `TextView` untuk membuat entri di **dimens.xml**.
- Letakkan kursor di string hard-code, tekan Alt-Enter (Option-Enter di Mac), dan pilih **Extract dimension resource**. Lalu edit name Resource ke **padding\_regular**.
- Tambahkan elemen `TextView` `TextView` di atas "Hello World" `TextView` bawah `TextView` yang Anda buat pada langkah sebelumnya. Tambahkan atribut-atribut berikut ke `TextView`:

Atribut #2 <code>TextView</code>	Nilai
<code>android:id</code>	"@id/article_subheading"
<code>layout_width</code>	"match_parent"
<code>layout_height</code>	"wrap_content"
<code>android:layout_below</code>	"@id/article_heading"
<code>android:padding</code>	"@dimen/padding_regular"
<code>android:textAppearance</code>	"@android:style/TextAppearance"
<code>android:text</code>	"Article Subtitle"

- Perhatikan karena Anda mengekstrak resource dimensi untuk string "10dp" ke **padding\_regular** pada `TextView` yang sebelumnya dibuat, Anda bisa menggunakan "@dimen/padding\_regular" untuk atribut `android:padding` dalam `TextView` ini.
- Ekstrak resource string untuk string "Article Subtitle" yang di-hardcode atribut `android:text` dalam `TextView` ke **article\_subtitle**.
- Tambahkan atribut-atribut `TextView` berikut ke elemen `TextView` "Hello World", dan ubah atribut `android:text`:

Atribut <code>TextView</code>	Nilai
<code>android:id</code>	"@id/article"
<code>android:lineSpacingExtra</code>	"5sp"
<code>android:layout_below</code>	"@id/article_subheading"
<code>android:text</code>	Ubah ke "Article text"

- Ekstrak resource string untuk "Article text" ke **article\_text**, dan ekstrak sumber dimensi "5sp" ke **line\_spacing**.
- Format ulang dan ratakan kode dengan memilih **Code > Reformat Code**. Sebaiknya format ulang dan ratakan kode agar memudahkan Anda dan orang lain memahaminya.

### Menambahkan teks artikel:

Pada aplikasi sebenarnya yang mengakses majalah atau artikel koran, artikel yang muncul mungkin berasal dari sumber online melalui penyedia konten, atau mungkin disimpan terlebih dulu dalam database di perangkat.

Untuk praktik ini, Anda akan membuat artikel dengan format satu string panjang pada resource strings.xml.

- Dalam folder **app > res > values**, buka **strings.xml**.
- Masukkan nilai string `article_title` dan `article_subtitle` dengan judul karangan dan subjudul untuk artikel yang Anda tambahkan. Nilai masing-masing string harus berupa teks satu baris tanpa tag HTML atau beberapa baris.
- Masukkan atau salin dan tempel teks untuk string `article_text`.
- Saat Anda memasukkan atau menempel teks di file **strings.xml**, baris teks tidak membungkus ke baris berikutnya, melainkan meluas melebihi margin kanan. Ini adalah perilaku yang benar, setiap baris teks baru yang dimulai di margin kiri mewakili seluruh paragraf.
- Masukkan `\n` untuk mewakili akhir sebuah baris, dan `\n`lainnya untuk mewakili baris kosong.

*Mengapa?\** Anda perlu menambahkan karakter akhir-baris untuk menjaga paragraf agar tidak bertabrakan satu dan lainnya.

**Tips:** Jika Anda ingin melihat teks dibungkus dalam strings.xml, Anda bisa menekan Return untuk memasukkan akhiran baris keras, atau memformat teksnya dulu di editor teks dengan akhiran baris keras.

- 1) Jika ada apostrof (') dalam teks, Anda harus melakukan *escape* dengan mengawalinya dengan backslash (\'). Jika Anda memiliki tanda kutip ganda dalam teks, Anda harus melakukan escape ("). Anda juga harus melakukan escape pada karakter non-ASCII lainnya. Lihat bagian "[Pemformatan dan Penataan gaya.
- 2) Masukkan tag HTML dan `</b>` di sekeliling kata yang harus dicetak tebal.



- 3) Masukkan HTML `</b>` dan tag `****` di sekeliling kata yang harus dicetak miring. Namun perhatikan bahwa jika Anda menggunakan apostrof dengan frasa yang dimiringkan, Anda harus menggantinya dengan apostrof lurus.
- 4) Jika Anda bisa menggabungkan cetak tebal dan miring dengan menggabungkan tag-nya, seperti dalam ... kata-kata...`</b></i>`. Tag HTML lain diabaikan.
- 5) Kurung seluruh teks di dalam `<string name="article_text">` `</string>` pada file `strings.xml`.
- 6) Sertakan tautan web untuk mengujinya, misalnya **www.google.com** (contoh bawah menggunakan `www.rockument.com`). *Jangan* gunakan tag HTML kecuali tag cetak tebal dan miring, karena akan diabaikan dan ditampilkan sebagai teks. Anda tentu tidak menginginkan hal ini.

```

<resources>
  <string name="app_name">Scrolling Text</string>
  <string name="article_title">Beatles Anthology Vol. 1</string>
  <string name="article_subtitle">Behind That Locked Door: Beatles Rarities!</string>
  <string name="article_text">In a vault deep inside Abbey Road Studios in London – pro
    For more information, see the Beatles Time Capsule at www.rockument.com.
  \n\n
  This volume starts with the first new Beatle song, "Free as a Bird" (based on a John Lenn
  \n\n
  <b>Highlights include:</b>
  \n\n
  <b><i>Cry for a Shadow</i></b> – Many a Beatle fanatic started down the bootleg road, lik
  \n\n
  <b><i>My Bonnie</i></b> and <b><i>Ain't She Sweet</i></b> – At the same session, the Beat
  \n\n
  <b><i>Searchin</i></b> – A Jerry Leiber – Mike Stoller comedy song that was a hit for the
  \n\n
  <b><i>Love Me Do</i></b> – An early version of the song, played a bit slower and with mor
  \n\n
  <b><i>She Loves You – Till There Was You – Twist and Shout</i></b> – Live at the Princess
  \n\n
  <b><i>Leave My Kitten Alone</i></b> – One of the lost Beatle songs recorded during the "B
  \n\n
  <b><i>One After 909</i></b> – A song recorded for the <i>Let It Be</i> album was actually
</string>
</resources>

```

- Jalankan aplikasi.

Artikel muncul, dan Anda bahkan bisa menggulirnya, namun penggulirannya tidak lancar dan tidak ada bilah gulir karena Anda belum memasukkan `ScrollView` (yang akan Anda lakukan di tugas berikutnya). Harap diingat bahwa mengetuk tautan web baru saat ini belum memberikan efek apa pun.

## Latihan 4.2. Menambahkan tautan Web aktif dan ScrollView

Pada tugas sebelumnya Anda membuat aplikasi Scrolling Text dengan TextViews untuk judul artikel, subjudul, dan teks artikel panjang. Anda juga menyertakan tautan web, namun tautannya belum aktif. Anda akan menambahkan kode untuk mengaktifkannya. Selain itu, TextView sendiri tidak memungkinkan pengguna untuk menggulir teks artikel untuk melihat semuanya. Anda akan menambahkan grup tampilan baru ScrollView ke layout XML yang akan membuat TextView dapat digulir.

### Menambahkan atribut autoLink untuk tautan web aktif

Tambahkan atribut `android:autoLink="web"` ke TextView `article`. Kode XML untuk TextView ini sekarang harus terlihat seperti berikut:

```
<TextView
    android:id="@+id/article"
    ...
    android:autoLink="web"
    ... />
```

### Menambahkan ScrollView ke layout

Untuk membuat tampilan (seperti TextView) dapat digulir, sematkan tampilan tersebut di dalam ScrollView.

1. Tambahkan ScrollView di antara TextView `article_subheading` dan TextView artikel. Saat Anda memasukkan atribut **<ScrollView>**, Android Studio automatically adds `</ScrollView>` at the end, and presents the `android:layout_width` and `android:layout_height` dengan saran. Pilih **wrap\_content** dari saran untuk kedua atribut. Kode sekarang harus terlihat seperti ini:

```
<TextView
    android:id="@+id/article_subheading"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/article_heading"
    android:padding="10dp"
    android:textAppearance="@android:style/TextAppearance"
    android:text="@string/article_subtitle"/>

<ScrollView
```

```

android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@id/article_subheading"></ScrollView>

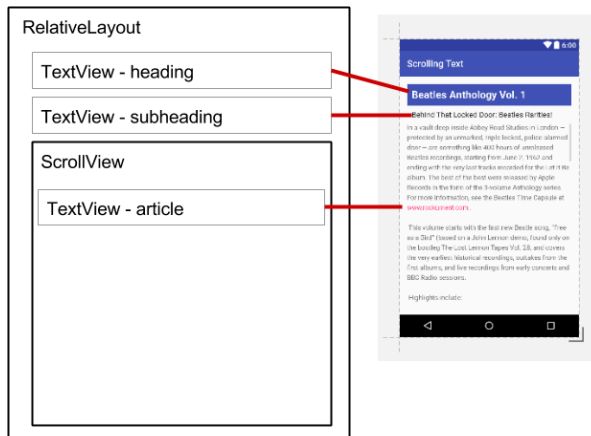
<TextView
    android:id="@+id/article"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/article_subheading"
    android:lineSpacingExtra="5sp"
    android:autoLink="web"
    android:text="@string/article_text"/>

```

2. Pindahkan </ScrollView> code *after* the article TextView so that the article TextView attributes are inside the ScrollView XML element.
3. Pindahkan atribut berikut dari TextView article, karena ScrollView sendiri akan ditempatkan bawah elemen article\_subheading, dan atribut untuk TextView ini akan berkonflik dengan ScrollView:

```
android:layout_below="@id/article_subheading"
```

Layout sekarang harus terlihat seperti ini:



4. Pilih **Code > Reformat Code** untuk memformat ulang kode XML agar **TextView article** sekarang muncul terindentasi di dalam **<ScrollView code**.

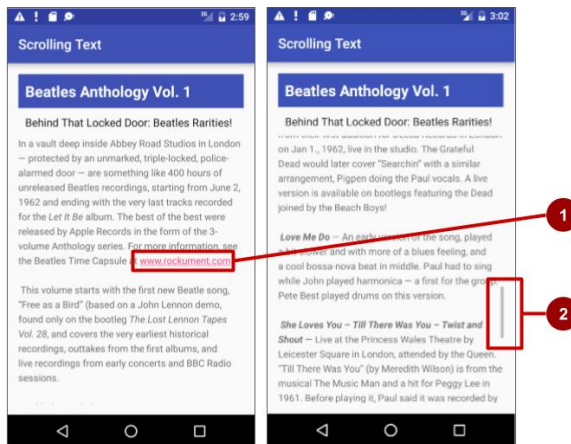
5. Jalankan aplikasi.

Gesek ke atas dan ke bawah untuk menggulir artikel. Bilah gulir muncul di margin kanan saat Anda menggulir.

Ketuk tautan web untuk membuka laman web. Atribut `android:autoLink` menjadikan semua URL yang dapat dikenali di TextView (seperti [www.rockument.com](http://www.rockument.com)) tautan web.

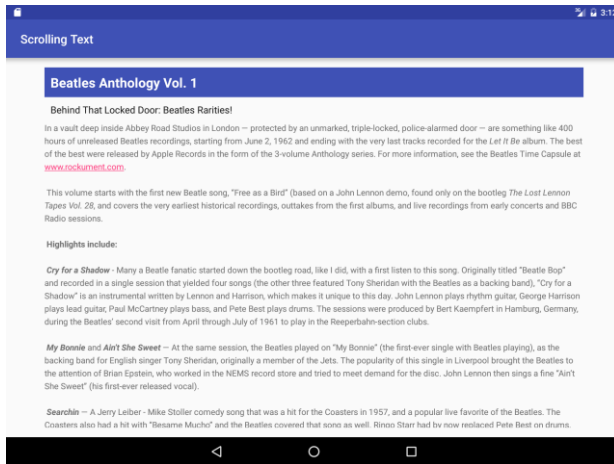
6. Putar perangkat atau emulator saat menjalankan aplikasi. Perhatikan bagaimana tampilan bergulir melebar untuk menggunakan tampilan penuh dan tetap bergulir dengan benar.

7. Jalankan aplikasi di tablet atau emulator tablet. Perhatikan bagaimana tampilan bergulir melebar untuk menggunakan tampilan penuh dan tetap bergulir dengan benar.



Pada gambar di atas, hal berikut muncul:

1. Tautan web aktif dalam teks bentuk bebas
2. Bilah gulir yang muncul saat menggulir teks.



File layout **activity\_main.xml** sekarang harus terlihat seperti ini:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context="com.example.android.scrollingtext.MainActivity">
    <TextView
        android:id="@+id/article_heading"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/colorPrimary"
        android:textColor="@android:color/white"
        android:paddingTop="10dp"
        android:paddingBottom="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:textAppearance="@android:style/TextAppearance.Large"
        android:textStyle="bold"
        android:text="@string/article_title"/>
    <TextView
        android:id="@+id/article_subheading"
```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/article_heading"
        android:paddingTop="10dp"
        android:paddingBottom="10dp"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"

        android:textAppearance="@android:style/TextAppearance"
        android:text="@string/article_subtitle"/>

<ScrollView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/article_subheading">

    <TextView
        android:id="@+id/article"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:lineSpacingExtra="5sp"
        android:autoLink="web"
        android:text="@string/article_text"/>

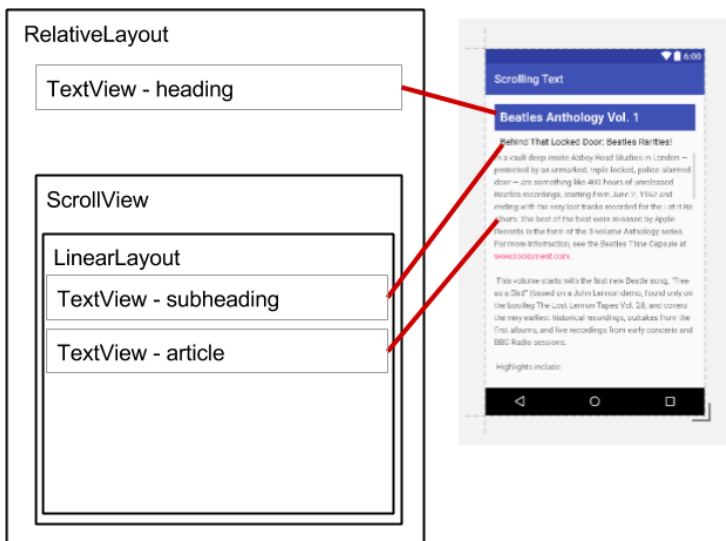
</ScrollView>
</RelativeLayout>

```

### Latihan 4.3. Menggulir beberapa elemen

Seperti yang kita lihat sebelumnya, grup tampilan `ScrollView` bisa hanya berisi satu tampilan anak (seperti `TextView article` yang Anda buat); namun, Tampilan tersebut bisa saja grup tampilan lain yang berisi Tampilan, seperti `LinearLayout`. Anda bisa *melapiskan* grup tampilan seperti `LinearLayout` di dalam grup tampilan `ScrollView`, sehingga menggulir semua yang berada di dalam `LinearLayout`.

Misalnya, jika Anda ingin subheading artikel bergulir bersama artikelnya, tambahkan `LinearLayout` di dalam `ScrollView`, dan pindahkan subheading, bersama artikelnya, ke dalam `LinearLayout`. Grup tampilan `LinearLayout` menjadi Tampilan anak tunggal dalam `ScrollView` seperti yang ditunjukkan pada gambar di bawah ini, dan pengguna dapat menggulir seluruh grup tampilan: subheading dan artikelnya.



### Menambahkan `LinearLayout` ke `ScrollView`

1. Di komputer Anda, buat salinan folder proyek Android Studio untuk `ScrollingText`, dan ganti nama proyek menjadi **`ScrollingText2`**. Untuk menyalin dan mengganti nama proyek, ikuti instruksi "Menyalin dan mengganti nama proyek" di [Appendiks](#).
2. Buka **`ScrollingText2`** di Android Studio, dan buka file `activity_main.xml` untuk mengubah kode layout XML.

3. Tambahkan `LinearLayout` di atas `TextView` `article` dalam `ScrollView`. Saat Anda memasukkan `<LinearLayout>`, Android Studio automatically adds `</LinearLayout>` to the end, and presents the `android:layout_width` and `android:layout_height` attributes with suggestions. Pilih **match\_parent** dan **wrap\_content** dari saran untuk lebar dan tingginya, secara berturut-turut. Kode sekarang harus terlihat seperti ini:

```
<LinearLayout
  android:layout_width="match_parent"
  android:layout_height="wrap_content"></LinearLayou
t>
```

Anda menggunakan `match_parent` untuk mencocokkan lebar grup tampilan parent, dan `wrap_content` membuat grup tampilan cukup besar untuk mengurung konten dan padding-nya.

4. Pindahkan kode `</LinearLayout>` code *after* the article `TextView` but *before* the closing `</ScrollView>` so that the `LinearLayout` includes the article `TextView` dan seluruhnya di dalam `ScrollView`.
5. Tambahkan atribut `android:orientation="vertical"` ke `LinearLayout` untuk menyetel orientasi `LinearLayout` ke vertikal. `LinearLayout` di dalam `ScrollView` seharusnya sekarang terlihat seperti ini (pilih **Code** > **Reformat Code** untuk mengidentasi grup tampilan dengan benar):

```
<ScrollView
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_below="@id/article_subheading">

  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <TextView
      android:id="@+id/article"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
```



```
        android:autoLink="web"
        android:lineSpacingExtra="5sp"
        android:text="@string/article_text" />

    </LinearLayout>
</ScrollView>
```

6. Pindahkan `TextView` `article_subheading` ke posisi di dalam `LinearLayout` di atas `TextView` `article`.
7. Pindahkan atribut `android:layout_below="@id/article_heading"` dari `TextView` `article_subheading`. Karena `TextView` ini sekarang ada di dalam `LinearLayout`, atribut ini akan berkonflik dengan atribut `LinearLayout`.
8. Ubah atribut `layout` `ScrollView` dari :

```
    android:layout_below="@id/article_subheading"
    menjadi
    android:layout_below="@id/article_heading".
```

Karena sekarang `subheading` sudah menjadi bagian dari `LinearLayout`, `ScrollView` harus diletakkan di bawah `heading`, bukan `subheading`.

9. Jalankan aplikasi.  
Gesek ke atas dan ke bawah untuk menggulir artikel, dan perhatikan bahwa `subheading` sekarang bergulir dengan artikelnya sementara `heading` tetap di tempat.

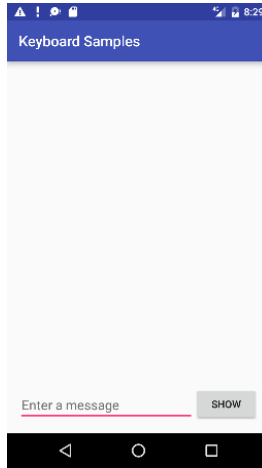
#### Latihan 4.4. Bereksperimen dengan atribut keyboard entri teks

Menyentuh bidang teks EditText yang bisa diedit akan menempatkan kursor pada bidang teks dan otomatis menampilkan keyboard di layar. Anda akan mengubah atribut bidang entri teks sehingga keyboard akan menyarankan pembetulan ejaan selagi Anda mengetik, dan otomatis memulai setiap kalimat baru dengan huruf kapital. Misalnya:

<code>android:inputType="textCapSentences"</code>	Menyetel keyboard untuk menjadikan huruf besar di awal kalimat.
<code>android:inputType="textAutoCorrect"</code>	Menyetel keyboard untuk menampilkan koreksi ejaan otomatis selagi Anda memasukkan karakter.
<code>android:inputType="textMultiLine"</code>	Mengaktifkan kunci Return pada keyboard untuk mengakhiri baris dan membuat baris kosong baru tanpa menutup keyboard
<code>android:inputType="textPassword"</code>	Menyetel karakter yang dimasukkan pengguna agar menjadi titik-titik untuk menyembunyikan sandi yang dimasukkan.

#### Membuat layout utama dan metode showText

Anda akan menambahkan Tombol, dan mengubah elemen TextView menjadi elemen EditText sehingga pengguna bisa memasukkan teks. Layout aplikasi akan tampak seperti gambar berikut.



1. Buat proyek baru yang bernama **Keyboard Samples**, dan pilih template Empty Activity.
2. Buka file layout **activity\_main.xml** untuk mengedit kode XML.
3. Tambahkan Tombol di atas elemen `TextView` yang sudah ada dengan atribut berikut:

Atribut Tombol	Nilai Baru
<code>android:id</code>	<code>"@+id/button_main"</code>
<code>android:layout_width</code>	<code>"wrap_content"</code>
<code>android:layout_height</code>	<code>"wrap_content"</code>
<code>android:layout_alignParentBottom</code>	<code>"true"</code>
<code>android:layout_alignParentRight</code>	<code>"true"</code>
<code>android:onClick</code>	<code>"showText"</code>
<code>android:text</code>	<code>"Show"</code>

4. Ekstrak resource string untuk nilai atribut `android:text` untuk membuat dan memasukkannya di `strings.xml`: Letakkan kursor pada **Show**, tekan Alt-Enter (Option-Enter di Mac), dan pilih **Extract string resource**. Lalu ubah name resource untuk nilai string ke **show**.

Anda mengekstrak resource string karena menjadikan proyek aplikasi lebih fleksibel untuk mengubah string. Penetapan resource string disimpan di file **strings.xml** (pada **app > res > values**). Anda bisa mengedit file ini untuk mengubah penetapan string sehingga aplikasi bisa dilokalkan dengan bahasa berbeda. Misalnya, nilai "Show" untuk resource yang bernama show bisa diubah ke "Montrer" untuk aplikasi dalam versi bahasa Prancis.

5. Ubah elemen `TextView` yang sudah ada seperti berikut:

- ⇒ Hapus atribut `android:text` yang menetapkan "Hello World!".
- ⇒ Ubah tag `TextView` menjadi tag `EditText` dan pastikan tag diakhiri dengan `/>`.
- ⇒ Tambahkan atau ubah atribut berikut:

Atribut EditText	Nilai Lama TextView	Nilai Baru EditText
<code>android:id</code>		<code>"@+id/editText_main"</code>
<code>android:layout_width</code>	<code>"wrap_content"</code>	<code>"match_parent"</code>
<code>android:layout_height</code>	<code>"wrap_content"</code>	<code>"wrap_content"</code>
<code>android:layout_alignParentBottom</code>		<code>"true"</code>
<code>android:layout_toLeftOf</code>		<code>"@id/button_main"</code>
<code>android:hint</code>		<code>"Enter a message"</code>

Anda telah mempelajari tentang atribut `android:layout_toLeftOf` dan `android:layout_alignParentBottom` di pelajaran sebelumnya. Atribut yang terkait layout ini berfungsi dengan grup tampilan `RelativeLayout` untuk memosisikan tampilan turunan relatif terhadap yang lain atau terhadap parent. Atribut `android:hint` menyetel teks agar tampil dalam bidang yang menyediakan petunjuk bagi pengguna untuk menyediakan masukan, misalnya "Enter Enter a message"

6. Ekstrak resource string untuk nilai atribut `android:hint` "Enter a message" ke nama resource **enter**. File layout **activity\_main.xml** sekarang harus terlihat seperti ini:

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/andro
id"
```

```

xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context="com.example.android.keyboardsamples.MainActivity">
    <Button
        android:id="@+id/button_main"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:onClick="showText"
        android:text="@string/show" />
    <EditText
        android:id="@+id/editText_main"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_toLeftOf="@id/button_main"
        android:hint="@string/enter" />
</RelativeLayout>

```

7. Buka MainActivity.java dan masukkan metode showText berikut, yang mengambil informasi yang dimasukkan ke dalam elemen EditText dan menampilkannya di dalam pesan toast


```

public void showText(View view) {
    EditText editText = (EditText)
    findViewById(R.id.editText_main);
    if (editText != null) {
        String showString = editText.getText().toString();
        Toast.makeText(this, showString,
        Toast.LENGTH_SHORT).show();
    }
}

```

8. Buka strings.xml (dalam app > res > values) dan edit nilai app\_name ke "Keyboard Samples" (pastikan untuk menyertakan spasi antara "Keyboard" dan "Samples").
9. Jalankan aplikasi dan periksa cara kerja keyboard.

Mengetuk tombol **Show** akan menampilkan pesan toast entri teks.

Untuk menutup keyboard di layar, ketuk panah yang mengarah ke bawah di baris bawah ikon. Di layout keyboard standar, ikon centang dalam lingkaran hijau, ditampilkan di bawah, muncul di sudut kanan bawah keypad. Ini dikenal sebagai tombol Return (atau Enter) dan digunakan untuk memasukkan baris baru: 

Dengan atribut default untuk elemen EditText, mengetuk tombol Return akan menambahkan baris teks baru. Di bagian berikutnya, Anda akan mengubah keyboard sehingga dapat mengubah huruf besar pada huruf pertama kalimat selagi Anda menetik. Sebagai hasil menyetel atribut `android:inputType`, atribut default untuk tombol Return berubah untuk mengalihkan fokus dari elemen EditText dan menutup keyboard.

#### Setel keyboard untuk menjadikan huruf besar di awal kalimat

1. Tambahkan atribut `android:inputType` ke elemen EditText menggunakan nilai `textCapSentences` untuk menyetel keyboard menjadi huruf besar di awal kalimat, sehingga pengguna bisa otomatis memulai kalimat dengan huruf besar:
2. `android:inputType="textCapSentences"`
3. Jalankan aplikasi Anda.

Huruf besar sekarang akan tampil pada keyboard di awal kalimat. Saat Anda mengetuk tombol Return di keyboard, keyboard ditutup dan entri teks Anda berakhir. Anda masih bisa menetik bidang entri teks untuk menambahkan lebih banyak teks atau mengedit teks. Ketuk **Show** untuk menampilkan teks di pesan toast.

#### Setel keyboard untuk menyembunyikan sandi saat memasukkannya

1. Ubah elemen EditText untuk menggunakan nilai `textPassword` untuk atribut `android:inputType`.
2. `android:inputType="textPassword"`
3. Ubah `android:hint` menjadi **"Enter your password"**.
4. Jalankan aplikasi.

Karakter yang dimasukkan pengguna berubah menjadi titik-titik untuk menyembunyikan sandi yang dimasukkan.

## Latihan 4.5. Mengubah tipe keyboard

Setiap bidang teks mengharapkan tipe masukan teks tertentu, seperti alamat email, nomor telepon, sandi, atau hanya teks biasa. Penting untuk menetapkan tipe masukan untuk setiap bidang teks di aplikasi Anda sehingga sistem akan menampilkan metode masukan virtual yang sesuai, seperti:

- Keyboard di layar standar untuk teks biasa
- Keyboard untuk alamat email yang menyertakan simbol "@" di lokasi penting
- Keypad ponsel untuk nomor telepon

### Menggunakan keyboard email

Ubah elemen `EditText` activity utama agar menampilkan keyboard email, bukan keyboard standar:

1. Pada elemen `EditText` di file layout **activity\_main.xml**, ubah atribut `android:inputType` menjadi yang berikut:  
`android:inputType="textEmailAddress"`
2. Ubah atribut `android:hint` menjadi **"Enter an email address"**.
3. Ekstrak resource string untuk nilai `android:hint` ke `enter_email`.
4. Jalankan aplikasi. Mengetuk bidang akan memunculkan keyboard email di layar dengan simbol "@" yang terletak di sebelah tombol spasi.

### Menggunakan keypad ponsel

Ubah elemen `EditText` activity utama agar menampilkan keypad ponsel, bukan keyboard standar:

1. Pada elemen `EditText` di file layout **activity\_main.xml**, ubah atribut `android:inputType` menjadi yang berikut:  
`android:inputType="phone"`
2. Ubah atribut `android:hint` menjadi **"Enter a phone number"**.
3. Ekstrak resource string untuk nilai `android:hint` ke `enter_phone`.
4. Jalankan aplikasi.

Mengetuk bidang kini akan memunculkan keypad ponsel di layar sebagai ganti keyboard standar.

## Latihan 4.6. Menambahkan Kontrol Spinner

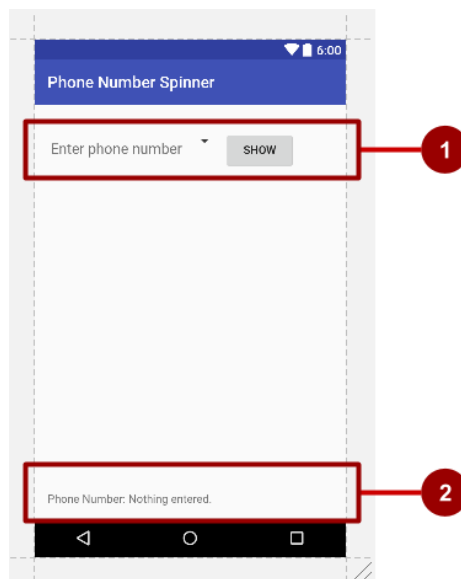
Kontrol masukan adalah komponen interaktif dalam antarmuka pengguna aplikasi Anda. Android menyediakan aneka ragam kontrol yang bisa Anda gunakan dalam UI, seperti tombol, bilah pencarian, kotak centang, tombol zoom, tombol toggle, spinner, dan masih banyak lagi. (Untuk informasi selengkapnya tentang kontrol masukan.

*Spinner* menyediakan cara cepat untuk memilih salah satu dari serangkaian nilai. Menyentuh spinner akan menampilkan daftar tarik-turun dengan semua nilai yang tersedia, yang bisa dipilih oleh pengguna. Jika hanya menyediakan dua atau tiga pilihan, Anda mungkin ingin menggunakan tombol radio untuk pilihan jika memiliki ruang di layout untuk tombol-tombol itu; akan tetapi, jika ada lebih dari tiga pilihan, spinner bekerja sangat baik, menggulir seperlunya untuk menampilkan item, dan hanya membutuhkan sedikit ruang di layout Anda.

Untuk memberikan cara memilih label untuk nomor telepon (misalnya Rumah, Kantor, Seluler, dan Lainnya), Anda bisa menambahkan spinner ke layout agar tampil tepat di sebelah bidang nomor telepon.

### Salin proyek KeyboardSamples dan ubah layout

Gunakan gambar berikut sebagai panduan untuk layout activity utama:



Pada gambar di atas:



1. LinearLayout pertama dengan tampilan EditText, ikon spinner, dan tombol **Show**.
2. LinearLayout kedua dengan dua TextView.

Ikuti langkah-langkah berikut:

1. Salin folder proyek **KeyboardSamples**, ubah namanya menjadi **PhoneNumberSpinner**, dan **optimalkan** kode untuk mengisikan nama baru di seluruh proyek aplikasi. (Lihat [Apendiks](#) untuk melihat petunjuk tentang menyalin proyek.)
2. Setelah mengoptimalkan kode, ubah nilai " di file **strings.xml** (di dalam **app > res > values**) menjadi **Phone Number Spinner** (berikut spasinya) sebagai nama aplikasi.
3. Bukan file layout **activity\_main.xml**.
4. Kurung elemen EditText dan Button yang ada dari pelajaran sebelumnya di dalam LinearLayout dengan orientasi horizontal, dan menempatkan elemen EditText di atas Button:

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="@dimen/activity_vertical_ma
    rgin"
    android:orientation="horizontal">

    <EditText
        ...
    <Button
        ...
</LinearLayout>
```

5. Buat perubahan berikut pada elemen EditText dan Button:
  - ⇒ Hapus atribut berikut ini dari elemen EditText:
    - android:layout\_toLeftOf
    - android:layout\_alignParentBottom
  - ⇒ Hapus atribut berikut dari elemen Button:
    - android:layout\_alignParentRight
    - android:layout\_alignParentBottom
  - ⇒ Ubah tiga atribut elemen EditText lainnya seperti berikut:

Atribut EditText	Nilai
------------------	-------

android:layout_width	"wrap_content"
android:inputType	"phone"
android:hint	"Enter phone number"

6. Tambahkan elemen Spinner di antara elemen EditText dan elemen Button:

```
<Spinner
    android:id="@+id/label_spinner"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
</Spinner>
```

Elemen Spinner menyediakan daftar tarik-turun. Pada tugas selanjutnya, Anda akan menambahkan kode yang akan mengisi daftar spinner dengan nilai. Kode layout untuk elemen EditText, Spinner, dan Button di dalam LinearLayout seharusnya sekarang tampak seperti ini:

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"

    android:layout_marginTop="@dimen/activity_vertical_margin"
    android:orientation="horizontal">

    <EditText
        android:id="@+id/editText_main"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="phone"
        android:hint="Enter phone number" />

    <Spinner
        android:id="@+id/label_spinner"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </Spinner>

    <Button
        android:id="@+id/button_main"
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="showText"
        android:text="Show" />

```

```
</LinearLayout>
```

7. Tambahkan `LinearLayout` lagi di bawah `LinearLayout` yang baru saja Anda buat, dengan orientasi horizontal untuk mengurung dua elemen `TextView` secara bersisian — deskripsi teks dan bidang teks untuk menampilkan nomor telepon dan label telepon — dan meratakan `LinearLayout` ke bawah parentnya (lihat gambar di atas):

```

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_alignParentBottom="true">

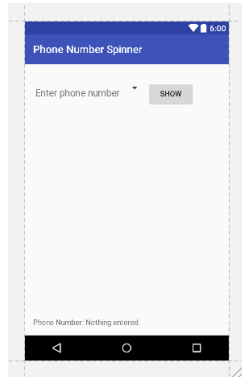
    <TextView
        ...
    <TextView
        ...
</LinearLayout>

```

8. Tambahkan elemen `TextView` berikut ke dalam `LinearLayout`:

<b>Atribut TextView</b>	<b>Nilai</b>
<code>android:id</code>	<code>"@+id/title_phonelabel"</code>
<code>android:layout_width</code>	<code>"wrap_content"</code>
<code>android:layout_height</code>	<code>"wrap_content"</code>
<code>android:text</code>	<code>"Phone Number: "</code>
<b>Atribut TextView</b>	<b>Nilai</b>
<code>android:id</code>	<code>"@+id/text_phonelabel"</code>
<code>android:layout_width</code>	<code>"wrap_content"</code>
<code>android:layout_height</code>	<code>"wrap_content"</code>
<code>android:text</code>	<code>"Nothing entered."</code>

- Periksa layout Anda dengan mengklik tab Preview di sebelah kanan jendela layout.



Anda sekarang seharusnya melihat layar (lihat gambar di atas) yang menampilkan bidang entri telepon di atas sebelah kiri, spinner kerangka di sebelah bidang dan tombol **Show** di sebelah kanan. Di bawah, akan muncul text "Phone Number:" yang diikuti oleh "Nothing entered".

- Ekstrak string Anda ke dalam resource string: Letakkan kursor di string hard-code, tekan Alt-Enter (Option-Enter di Mac), dan pilih **Extract string resources**. Lalu edit nama Resource untuk nilai string. Ekstrak seperti berikut:

Elemen	String	Resource string
EditText	"Enter phone number"	"@string/hint_phonenumber"
Button	"Show"	"@string/show_button"
TextView	"Phone Number: "	"@string/phonenummer_label"
TextView	"Nothing entered."	"@string/nothing_entered"

Tambahkan kode untuk mengaktifkan spinner dan listener-nya

Pilihan untuk spinner label telepon ini adalah string statis yang didefinisikan dengan baik ("Home", "Work", dsb), sehingga Anda bisa menggunakan larik teks yang didefinisikan di strings.xml untuk menyimpan nilai untuknya.

Untuk mengaktifkan spinner dan listener-nya, implementasikan antarmuka `AdapterView.OnItemSelectedListener`, yang juga memerlukan

penambahan metode callback `onItemSelected()` dan `onNothingSelected()`.

1. Buka **strings.xml** untuk mendefinisikan nilai yang bisa dipilih (**Home**, **Work**, **Mobile**, dan **Other**) untuk spinner sebagai larik string `labels_array`.

```
<string-array name="labels_array">
    <item>Home</item>
    <item>Work</item>
    <item>Mobile</item>
    <item>Other</item>
</string-array>
```

2. Untuk mendefinisikan callback pilihan untuk spinner, ubah kelas `MainActivity` Anda untuk mengimplementasikan antarmuka.

```
AdapterView.OnItemSelectedListener seperti yang
ditampilkan:

public class MainActivity extends
    AppCompatActivity implements
    AdapterView.OnItemSelectedListener {
```

Selagi Anda mengetik **AdapterView**. di pernyataan di atas, Android Studio otomatis mengimpor widget `AdapterView`. Alasan mengapa Anda memerlukan `AdapterView` adalah Anda memerlukan adaptor—khususnya `ArrayAdapter`—untuk menetapkan larik ke spinner. Sebuah *adaptor*\* menyambungkan data Anda—dalam hal ini, larik item spinner—ke tampilan spinner. Anda akan mempelajari selengkapnya tentang pola penggunaan adaptor untuk menyambungkan data seperti ini di pelajaran lain. Baris ini harus muncul di blok pernyataan impor:

```
import android.widget.AdapterView
```

Setelah mengetikkan **OnItemSelectedListener** di pernyataan di atas, tunggu beberapa detik sampai bola lampu merah muncul di batas kiri.

3. Klik bola lampu dan pilih **Implement methods**. Metode `onItemSelected()` dan `onNothingSelected()` yang

diperlukan untuk `OnItemSelectedListener`, seharusnya sudah disorot, dan opsi "Insert @Override" harus dicentang. Klik **OK**.

Langkah ini otomatis menambahkan metode callback `onItemSelected()` dan `onNothingSelected()` kosong ke bawah kelas `MainActivity`. Kedua metode tersebut menggunakan parameter `AdapterView<?>`. The `<?>` is a Java type wildcard, enabling the method to be flexible enough to accept any type of `AdapterView` sebagai argumen.

4. Buat instance objek spinner di metode `onCreate()` menggunakan elemen `Spinner` di layout (`label_spinner`), dan tetapkan listener-nya (`spinner.setOnItemSelectedListener`) di metode `onCreate()`. Tambahkan kode ke metode `onCreate()`:

```
@Override
protected void onCreate(Bundle
savedInstanceState) {
    ...
    // Create the spinner.
    Spinner spinner = (Spinner)
findViewById(R.id.label_spinner);
    if (spinner != null) {
        spinner.setOnItemSelectedListener(this);
    }
    ...
}
```

5. Lanjutkan mengedit metode `onCreate()`, tambahkan pernyataan yang membuat `ArrayAdapter` dengan larik string (`labels_array`) menggunakan layout spinner sederhana yang disediakan oleh Android untuk setiap item (`layout.simple_spinner_item`):

```
...
// Create ArrayAdapter using the string array
and default spinner layout.
ArrayAdapter<CharSequence> adapter =
ArrayAdapter.createFromResource(this,
    R.array.labels_array,
    android.R.layout.simple_spinner_item);
...
```

Layout `simple_spinner_item` yang digunakan di langkah ini, dan layout `simple_spinner_dropdown_item` yang digunakan di langkah berikutnya, adalah layout default yang ditetapkan sebelumnya yang disediakan oleh Android di kelas `R.layout`. Anda harus menggunakan layout ini kecuali Anda ingin mendefinisikan layout Anda sendiri untuk item di spinner dan tampilan spinner.

6. Tetapkan layout untuk pilihan spinner menjadi `simple_spinner_dropdown_item`, lalu terapkan adaptor ke spinner:

```
...
    // Specify the layout to use when the list of
    // choices appears.
    adapter.setDropDownViewResource

    (android.R.layout.simple_spinner_dropdown_item);
    // Apply the adapter to the spinner.
    if (spinner != null) {
        spinner.setAdapter(adapter);
    }
...

```

### Tambahkan kode untuk merespons pilihan pengguna:

Saat pengguna memilih item di spinner, objek Spinner menerima kejadian `on-item-selected`. Untuk menangani kejadian ini, Anda telah mengimplementasikan `antarmuka AdapterView.OnItemSelectedListener` di langkah sebelumnya, dengan menambahkan metode callback `onItemSelected()` dan `onNothingSelected()` kosong.

Di langkah ini, Anda terlebih dulu harus mendeklarasikan `mSpinnerLabel` sebagai string untuk menampung item yang dipilih. Kemudian Anda mengisikan kode untuk metode `onItemSelected()` untuk mengambil item yang dipilih di spinner, menggunakan `getItemAtPosition()`, dan menetapkan item ke `mSpinnerLabel`.

1. Deklarasikan string `mSpinnerLabel` di awal definisi kelas `MainActivity`:

```

public class MainActivity extends
    AppCompatActivity implements

    AdapterView.OnItemClickListener {
    private String mSpinnerLabel = "";
    ...
}

```

2. Tambahkan kode ke metode callback `onItemSelected()` kosong, seperti yang ditampilkan di bawah ini, untuk mengambil item yang dipilih pengguna menggunakan `getItemAtPosition`, dan tetapkan ke `mSpinnerLabel`. Anda juga bisa menambahkan sebuah panggilan ke metode `showText()` yang sudah Anda tambahkan ke versi aplikasi sebelumnya:

```

public void onItemSelected(AdapterView<?>
    adapterView, View view, int
        i, long l) {
    mSpinnerLabel =
    adapterView.getItemAtPosition(i).toString();
    showText(view);
}

```

**Tip:** Dengan menambahkan metode `showText()` ke metode `onItemSelected()` di atas, Anda telah mengaktifkan listener pilihan spinner untuk menampilkan pilihan spinner bersama dengan nomor telepon, sehingga Anda tidak lagi memerlukan tombol **Show** yang memanggil metode `showText()`.

3. Tambahkan kode ke metode callback `onNothingSelected()` yang kosong, seperti yang ditampilkan di bawah ini, untuk menampilkan pesan logcat jika tidak satu pun yang dipilih:

```

public void onNothingSelected(AdapterView<?>
    adapterView) {
    Log.d(TAG, "onNothingSelected: ");
}

```

TAG di pernyataan di atas berwarna merah karena belum didefinisikan.



4. Ekstrak resource string untuk "onNothingSelected: " ke nothing\_selected.
5. Klik **TAG**, klik bola lampu merah, dan pilih **Create constant field 'TAG'** dari menu munculan. Android Studio menambahkan yang berikut di bawah deklarasi kelas MainActivity:

```
private static final String TAG =
```

- ⇒ Tambahkan MainActivity.class.getSimpleName() agar menggunakan nama sederhana kelas untuk TAG:

```
private static final String TAG =  
MainActivity.class.getSimpleName()
```

- ⇒ Ubah pernyataan String showString di metode showText untuk menampilkan string yang dimasukkan dan item spinner yang dipilih (mSpinnerLabel):

```
String showString =  
(editText.getText().toString() + " - " +  
mSpinnerLabel);
```

6. Jalankan aplikasi.

#### Latihan 4.7. Menggunakan dialog untuk peringatan

Anda bisa menyediakan dialog untuk peringatan yang mengharuskan pengguna membuat keputusan. *Dialog* adalah jendela yang muncul di atas tampilan atau mengisi tampilan, menyela alur aktivitas.

Misalnya, sebuah dialog peringatan mungkin mengharuskan pengguna untuk mengeklik **Continue** setelah membacanya, atau memberi pilihan kepada pengguna untuk menyetujui suatu tindakan dengan mengeklik tombol positif (seperti **OK** atau **Accept**), atau untuk tidak menyetujui dengan mengeklik tombol negatif (seperti **Cancel**). Di Android, gunakan subclass `AlertDialog` dari kelas `Dialog` untuk menampilkan dialog standar untuk peringatan.

Dalam praktik ini Anda akan menggunakan tombol untuk memicu dialog peringatan standar. Di aplikasi nyata, Anda bisa memicu dialog peringatan berdasarkan beberapa ketentuan, atau saat pengguna menyetuk sesuatu.

#### Proyek Android Studio: `AlertSample`

Dalam latihan ini, Anda akan membangun sebuah peringatan dengan tombol `OK` dan `Cancel`, yang akan dipicu oleh klik pengguna pada tombol.

1. Buat proyek baru bernama **Alert Sample** berdasarkan template `Empty Activity`.
2. Bukan layout `activity_main.xml` dan buat perubahan berikut:

Atribut <code>TextView</code>	Nilai
<code>android:id</code>	<code>"@+id/top_message"</code>
<code>android:text</code>	<code>"Tap to test the alert:"</code>

3. Ekstrak string `android:text` di atas ke dalam `tap_test` resource untuk membuatnya lebih mudah diterjemahkan.
4. Tambahkan `Button` dengan atribut berikut:

Atribut Tombol	Nilai
<code>android:id</code>	<code>"@+button1"</code>
<code>android:layout_width</code>	<code>wrap_content</code>
<code>android:layout_height</code>	<code>wrap_content</code>
<code>android:layout_below</code>	<code>"@id/top_message"</code>

android:layout_marginTop	"36dp"
android:text	"Alert"
android:onClick	"onClickShowAlert"

5. Ekstrak string `android:text` di atas ke dalam `alert_button` resource untuk membuatnya lebih mudah diterjemahkan.
6. Ekstrak nilai dimensi untuk `android:layout_marginTop` dengan cara yang sama: Letakkan kursor di `"36dp"`, tekan Alt-Enter (Option-Enter di Mac), dan pilih **Extract dimension resource**. Lalu edit nama Resource untuk nilai ke **button\_top\_margin**.

Penetapan resource dimensi disimpan di file **dimens.xml** (di **app > res > values > dimens**). Anda bisa mengedit file ini untuk mengubah penetapan, sehingga aplikasi bisa diubah untuk ukuran layar yang berbeda.

### Menambahkan dialog peringatan ke activity utama

Pola desain *builder* mempermudah pembuatan objek dari kelas yang membutuhkan banyak atribut wajib serta opsional dan karena itu memerlukan banyak parameter untuk dibangun. Tanpa pola ini, Anda tentunya harus membuat konstruktor untuk kombinasi atribut yang diperlukan dan atribut opsional; dengan pola ini, kode lebih mudah dibaca dan dikelola. Untuk informasi selengkapnya tentang pola desain builder.

Kelas builder biasanya merupakan kelas member statis dari kelas yang dibangunnya. Anda menggunakan `AlertDialog.Builder` untuk membangun dialog peringatan standar, menggunakan `setTitle` untuk menyetel judulnya, `setMessage` untuk menyetel pesannya, dan `setPositiveButton` dan `setNegativeButton` untuk menyetel tombolnya.

Untuk membuat peringatan, Anda perlu membuat objek `AlertDialog.Builder`. Anda akan menambahkan metode `onClickShowAlert()`, yang akan membuat objek ini sebagai urutan pertama bisnis.

**Catatan:** Agar memudahkan contoh ini dipahami, dialog peringatan dibuat di metode `onClickShowAlert()`. Ini hanya terjadi jika metode `onClickShowAlert` dipanggil, yang akan terjadi saat pengguna mengklik tombol. Artinya, aplikasi membangun dialog baru hanya saat tombol diklik, yang berguna jika dialog jarang dilihat (saat pengguna mengambil jalur tertentu di aplikasi). Akan tetapi, jika dialog sering mungkin, sebaiknya Anda

membangun dialog sekali di metode `onCreate()`, lalu menjalankan dialog di metode `onClickShowAlert`.

1. Tambahkan metode `onClickShowAlert()` ke `MainActivity.java` seperti berikut:

```
public void onClickShowAlert(View view) {
    AlertDialog.Builder myAlertBuilder = new
    AlertDialog.Builder(MainActivity.this);
```

**Catatan:** Jika `AlertDialog.Builder` tidak dikenali saat Anda memasukkannya, klik ikon bola lampu merah dan pilih version pustaka dukungan (**android.support.v7.app.AlertDialog**) untuk mengimpornya ke activity Anda.

2. Setel judul dan pesan untuk dialog peringatan dalam `onClickShowAlert()` setel kode di langkah sebelumnya:

```
...
// Set the dialog title.
myAlertBuilder.setTitle("Alert");
// Set the dialog message.
myAlertBuilder.setMessage("Click OK to continue,
or Cancel to stop:");
...
```

3. Ekstrak judul dan pesan ke dalam resource string. Baris kode sebelumnya sekarang seharusnya menjadi:

```
...
// Set the dialog title.
myAlertBuilder.setTitle(R.string.alert_title);
// Set the dialog message.
myAlertBuilder.setMessage(R.string.alert_message);
...
```

4. Tambahkan tombol **OK** pada peringatan dengan `setPositiveButton()` dan menggunakan `onClickListener()`:

```
...
// Add the buttons.
myAlertBuilder.setPositiveButton("OK", new
DialogInterface.OnClickListener() {
```

```

        public void onClick(DialogInterface
dialog, int which) {
            // User clicked OK button.

            Toast.makeText(getApplicationContext(),
"Pressed OK",
Toast.LENGTH_SHORT).show();
        }
    });
    ...

```

Anda menyetel tombol positif (**OK**) dan negatif (**Cancel**) menggunakan metode `setPositiveButton()` dan `setNegativeButton()`. Setelah pengguna menyetel tombol **OK** di peringatan, Anda bisa mengambil pilihan pengguna dan menggunakannya di kode Anda. Dalam contoh ini, Anda menampilkan pesan toast jika tombol **OK** diklik.

5. Ekstrak resource string untuk "OK" dan untuk "Pressed OK". Pernyataan sekarang seharusnya menjadi:

```

...
// Add the buttons.
myAlertBuilder.setPositiveButton(R.string.ok, new
    DialogInterface.OnClickListener() {
        public void onClick(DialogInterface
dialog, int which) {
            // User clicked OK button.

            Toast.makeText(getApplicationContext(),
R.string.pressed_ok,
Toast.LENGTH_SHORT).show();
        }
    });
...

```

6. Tambahkan tombol **Cancel** pada peringatan dengan `setNegativeButton()` dan `onClickListener()`, tampilkan pesan toast jika tombol diklik, lalu batalkan dialog:

```

...
myAlertBuilder.setNegativeButton("Cancel", new
    DialogInterface.OnClickListener() {

```

```

        public void onClick(DialogInterface
dialog, int which) {
            // User cancelled the dialog.

            Toast.makeText(getApplicationContext(),
"Pressed Cancel", Toast.LENGTH_SHORT).show();
        }
    });
    ...

```

7. Ekstrak resource string untuk "Cancel" dan "Pressed Cancel". Pernyataan sekarang seharusnya menjadi:

```

    ...
    myAlertBuilder.setNegativeButton(R.string.cancel, new

    DialogInterface.OnClickListener() {
        public void onClick(DialogInterface
dialog, int which) {
            // User cancelled the dialog.

            Toast.makeText(getApplicationContext(),
R.string.pressed_cancel,

            Toast.LENGTH_SHORT).show();
        }
    });
    ...

```

8. Tambahkan `show()` yang membuat lalu menampilkan dialog peringatan, ke akhir `onClickShowAlert()`:

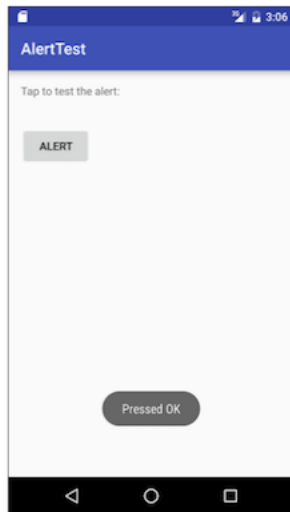
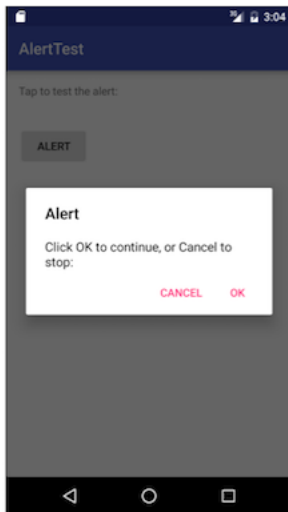
```

    ...
    // Create and show the AlertDialog.
    myAlertBuilder.show();
}

```

9. Jalankan aplikasi.

Anda seharusnya bisa mengetuk tombol **Alert**, yang ditampilkan di sebelah kiri gambar berikut, untuk melihat dialog peringatan, yang ditampilkan di tengah gambar berikut ini. Dialog menampilkan tombol **OK** dan **Cancel** dan pesan toast muncul yang menampilkan mana yang Anda tekan, seperti yang ditampilkan di sebelah kanan gambar berikut ini.



#### Latihan 4.8. Menggunakan picker untuk masukan pengguna

Dalam tugas ini, Anda akan membuat proyek baru, dan menambahkan picker tanggal dan picker waktu. Anda juga akan mempelajari cara menggunakan fragmen. *Fragmen* adalah perilaku atau bagian antarmuka pengguna dalam suatu activity. Fragmen mirip seperti activity mini di dalam activity utama, dengan siklus hidupnya sendiri, dan digunakan untuk membangun picker. Semua pekerjaan dilakukan untuk Anda. Untuk mengetahui tentang fragmen, lihat Fragmen dalam Panduan API.

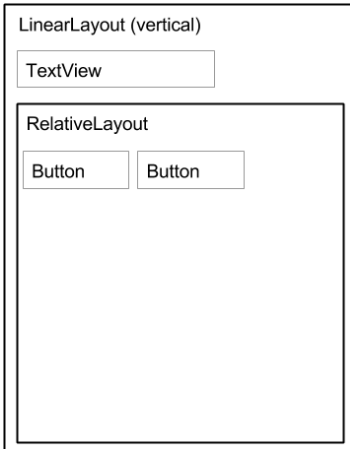
Salah satu manfaat menggunakan fragmen untuk picker adalah Anda bisa mengisolasi bagian kode untuk mengelola tanggal dan waktu untuk berbagai lokal yang menampilkan tanggal dan waktu dengan cara yang berbeda. Praktik terbaik untuk menampilkan picker adalah dengan menggunakan instance `DialogFragment` yang merupakan subkelas `Fragmen`. `DialogFragment` menampilkan jendela dialog yang mengambang di atas jendela activitynya. Dalam latihan ini, Anda akan menambahkan fragmen untuk setiap dialog picker dan menggunakan `DialogFragment` untuk mengelola daur hidup dialog.

##### Membuat layout activity utama:

Untuk memulai tugas ini, buat layout activity utama untuk menyediakan tombol untuk mengakses picker waktu dan tanggal. Periksa kode layout XML berikut:

1. Mulai proyek baru yang bernama **Date Time Pickers** menggunakan template `Empty Activity`.
2. Buka `activity_main.xml` untuk mengedit kode layout.
3. Ubah parent `RelativeLayout` agar menjadi `LinearLayout` dan tambahkan `android:orientation="vertical"` untuk mengatur orientasi layout menjadi vertikal. Tidak perlu mengkhawatirkan penampilan layout saat ini. Sasarannya adalah menggunakan layout yang menyematkan `RelativeLayout` di dalam `LinearLayout`:





- Ubah teks elemen `TextView` pertama ke **"Choose the date and time: "** dan ekstrak teks ke resource string `choose_datetime`.

Atribut <code>TextView</code>	Nilai Lama	Nilai Baru
<code>android:text</code>	"Hello World"	"@string/choose_datetime"

- Tambahkan atribut `android:textSize` dan masukkan ukuran teks **20sp**. Ekstrak dimensi `android:textSize` ke `text_size`.

Atribut <code>TextView</code>	Nilai Lama	Nilai Baru
<code>android:textSize</code>		"@dimen/text_size"

- Tambahkan turunan `RelativeLayout` di dalam `LinearLayout` untuk menampung elemen `Button`, dan terima lebar dan tinggi default `"match parent"`.
- Tambahkan elemen `Button` pertama di dalam `RelativeLayout` dengan atribut berikut:

Atribut Tombol Pertama	Nilai
<code>android:layout_width</code>	"wrap_content"
<code>android:layout_height</code>	"wrap_content"
<code>android:id</code>	"@+id/button_date"
<code>android:layout_marginTop</code>	"12dp"

android:text	"Date"
android:onClick	"showDatePickerDialog"

8. Tidak perlu mengkhawatirkan referensi `showDatePickerDialog` berwarna merah. Metode belum didefinisikan—Anda akan mendefinisikannya nanti.
9. Ekstrak string "Date" ke dalam resource string `date_button`.
10. Ekstrak dimensi "12dp" untuk `android:layout_marginTop` ke `button_top_margin`.
11. Tambahkan elemen `Button` kedua di dalam turunan `RelativeLayout` atribut berikut:

Atribut Tombol Kedua	Nilai
android:layout_width	"wrap_content"
android:layout_height	"wrap_content"
android:id	"@+id/button_time"
android:layout_marginTop	"@dimen/button_top_margin"
android:layout_alignBottom	"@id/button_date"
android:layout_toRightOf	"@id/button_date"
android:text	"Time"
android:onClick	"showTimePickerDialog"

12. Referensi `showTimePickerDialog` berwarna merah. Metode belum didefinisikan — Anda akan mendefinisikannya nanti.
13. Ekstrak string "Time" ke dalam resource string `time_button`.
14. Jika Anda belum melakukannya, klik tab **Preview** untuk menampilkan pratinjau layout. Pratinjau seharusnya tampak seperti kode dan gambar berikut ini.

### Kode solusi untuk layout utama:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:orientation="vertical"
android:layout_width="match_parent"
android:layout_height="match_parent"
```

```

android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"

tools:context="com.example.android.DateTimePickers.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="@dimen/text_size"
        android:text="@string/choose_datetime"/>

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/button_date"

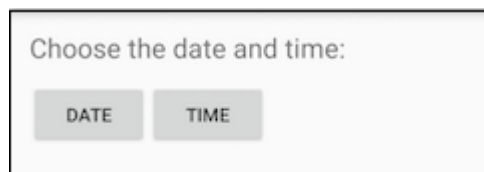
android:layout_marginTop="@dimen/button_top_margin"
            android:text="@string/date_button"
            android:onClick="showDatePickerDialog"/>

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/button_time"

android:layout_marginTop="@dimen/button_top_margin"
            android:layout_alignBottom="@id/button_date"
            android:layout_toRightOf="@id/button_date"
            android:text="@string/time_button"
            android:onClick="showTimePickerDialog"/>

    </RelativeLayout>
</LinearLayout>

```



#### Latihan 4.9. Buat fragmen baru untuk picker tanggal

Dalam latihan ini, Anda akan menambahkan fragmen untuk picker tanggal. Fragmen mirip seperti activity mini di dalam activity utama, dengan daur hidupnya sendiri.

1. Luaskan **app > java > com.example.android.DateTimePickers** dan pilih **MainActivity**.
2. Pilih **File > New > Fragment > Fragment (Blank)**, dan beri nama fragmen **DatePickerFragment**. Kosongkan ketiga opsi kotak centang sehingga Anda *tidak* membuat XML layout, *jangan* sertakan metode pabrik fragmen, dan *jangan* sertakan callback antarmuka. Anda tidak perlu membuat layout untuk picker standar. Klik **Finish** untuk membuat fragmen.
3. Buka **DatePickerFragment** dan edit definisi kelas `DatePickerFragment` untuk memperluas `DialogFragment` dan `DatePickerDialog.OnDateSetListener` dan implementasikan `DatePickerDialog.OnDateSetListener` untuk membuat picker tanggal standar dengan sebuah listener.

```
public class DatePickerFragment extends
    DialogFragment implements
    DatePickerDialog.OnDateSetListener {
```

Selagi Anda mengetik **DialogFragment** dan **DatePickerDialog.OnDateSetListener**, Android Studio otomatis menambahkan yang berikut di blok `import` di atas:

```
import android.app.DatePickerDialog;
import android.support.v4.app.DialogFragment;
```

Sebagai tambahan, ikon bola lampu merah muncul di margin kiri setelah beberapa detik.

4. Klik ikon bola lampu merah dan pilih **Implement methods** dari menu munculan. Sebuah dialog muncul dengan `onDateSet()` telah dipilih dan opsi "Insert @Override" dicentang. Klik **OK** untuk membuat metode `onDateSet()` kosong. Metode ini akan dipanggil saat pengguna menyetel tanggal. Setelah menambahkan metode `onDateSet()` kosong, Android Studio secara otomatis menambahkan yang berikut ini dalam blok `import` di atas.

```
import android.widget.DatePicker
```

5. Parameter metode `onDateSet()` harus `int year`, dan `int month`, dan `int dayOfMonth`. Ubah parameter `dayOfMonth` ke `day` agar ringkas:

```
public void onDateSet(DatePicker view, int  
year, int month, int day)
```

6. Buang konstruktor publik kosong untuk `DatePickerFragment`.
7. Ganti `onCreateView()` dengan `onCreateDialog()` yang mengembalikan `Dialog`, dan anotasikan `onCreateDialog()` dengan `@NonNull` untuk menunjukkan bahwa nilai pengembalian `Dialog` tidak boleh `null`—upaya apa pun untuk mengacu pada nilai pengembalian `Dialog` harus dicentang `null`.

```
@NonNull  
@Override  
public Dialog onCreateDialog(Bundle  
savedInstanceState) {  
    ...  
}
```

8. Tambahkan kode berikut ke `onCreateDialog()` untuk membuat instance `year`, `month`, dan `day` dari `Calendar`, dan mengembalikan dialog dan nilai ini ke activity utama. Saat Anda memasukkan `Calendar`, tetapkan import menjadi `java.util.Calendar`.

```
@NonNull  
@Override  
public Dialog onCreateDialog(Bundle  
savedInstanceState) {  
    // Use the current date as the default date in the picker.  
    final Calendar c = Calendar.getInstance();  
    int year = c.get(Calendar.YEAR);  
    int month = c.get(Calendar.MONTH);  
    int day = c.get(Calendar.DAY_OF_MONTH);  
    // Create a new instance of DatePickerDialog and return  
    it.  
    return new DatePickerDialog(getActivity(), this,  
year, month, day);  
}
```

## Kode solusi untuk DatePickerFragment:

```
public class DatePickerFragment extends DialogFragment
    implements DatePickerDialog.OnDateSetListener {
    @NonNull
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState)
    {
        // Use the current date as the default date in the picker.
        final Calendar c = Calendar.getInstance();
        int year = c.get(Calendar.YEAR);
        int month = c.get(Calendar.MONTH);
        int day = c.get(Calendar.DAY_OF_MONTH);

        // Create a new instance of DatePickerDialog and return it.
        return new DatePickerDialog(getActivity(), this, year,
            month, day);
    }
    public void onDateSet(DatePicker view, int year, int
        month, int day) {
        // Do something with the date chosen by the user.
    }
}
```

## Buat fragmen baru untuk picker waktu

Tambahkan fragmen ke proyek DateTimePickers untuk picker waktu:

1. Pilih **MainActivity** lagi.
2. Pilih **File > New > Fragment > Fragment (Blank)**, dan beri nama fragmen **TimePickerFragment**. Kosongkan ketiga opsi sehingga Anda tidak membuat XML layout, jangan sertakan metode pabrik fragmen, dan jangan sertakan callback antarmuka. Klik **Finish** untuk membuat fragmen.
3. Bukan *\*TimePickerFragment* dan ikuti prosedur yang sama dengan *DatePickerFragment*, mengimplementasikan metode kosong `onTimeSet()`, mengganti `onCreateView()` dengan `onCreateDialog()`, dan menghapus konstruktor publik untuk *TimePickerFragment*. *TimePickerFragment* melakukan tugas yang sama dengan *DatePickerFragment*, tetapi dengan nilai waktu:

- Ini memperluas `DialogFragment` dan mengimplementasikan `TimePickerDialog.OnTimeSetListener` untuk membuat picker waktu standar dengan sebuah listener. Lihat Picker untuk informasi selengkapnya tentang memperluas `DialogFragment` untuk picker waktu:
- Ini menggunakan metode `onCreateDialog()` untuk melakukan inisialisasi `hour` dan `minute` dari `Calendar`, dan mengembalikan dialog dan nilai ini ke aktivitas utama menggunakan format tanggal 24 jam. Saat Anda memasukkan `Calendar`, tetapkan `import` menjadi `java.util.Calendar`.
- Ini juga mendefinisikan metode `onTimeSet()` kosong agar Anda bisa menambahkan kode untuk menggunakan `hourOfDay` dan `minute` yang dipilih pengguna. Metode ini akan dipanggil saat pengguna menyetel waktu:

```
public void onTimeSet(TimePicker view, int
hourOfDay, int minute) {
// Do something with the time chosen by the user.
}
```

**Catatan:** Saat Anda membuat perubahan, Android Studio secara otomatis menambahkan yang berikut ini dalam blok `import` di bagian atas:

```
import android.support.v4.app.DialogFragment;
import android.app.TimePickerDialog;
import android.widget.TimePicker;
import java.util.Calendar;
```

### Modifikasi aktivitas utama

Meskipun kebanyakan kode di aktivitas utama tetap sama, Anda perlu menambahkan metode yang membuat instance `FragmentManager` untuk mengelola setiap fragmen dan menampilkan setiap picker.

1. Buat resource string di **strings.xml**:

```
<string
name="date_picker">datePicker</string>
<string
name="time_picker">timePicker</string>
```

2. Buka **MainActivity**.

3. Tambahkan metode `showDatePickerDialog()` dan `showTimePickerDialog()`, dengan merujuk ke kode di bawah ini. Ini membuat instance `FragmentManager` untuk mengelola fragmen secara otomatis, dan untuk menampilkan picker.

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void showDatePickerDialog(View v) {
        DialogFragment newFragment = new
        DatePickerFragment();

        newFragment.show(getSupportFragmentManager(),
        getString(R.string.date_picker));
    }

    public void showTimePickerDialog(View view) {
        DialogFragment newFragment = new
        TimePickerFragment();

        newFragment.show(getSupportFragmentManager(),
        getString(R.string.time_picker));
    }
}
```

4. Jalankan aplikasi.



## Latihan 4.10. Menggunakan tampilan gambar sebagai tombol

Anda bisa membuat tampilan agar bisa diklik, sebagai tombol, dengan menambahkan atribut `android:onClick` di layout XML. Misalnya, Anda bisa membuat gambar agar berfungsi sebagai tombol dengan menambahkan `android:onClick` ke `ImageView`.

Dalam tugas ini, Anda akan membuat prototipe aplikasi untuk memesan pencuci mulut dari kafe. Setelah memulai proyek baru berdasarkan template Basic Activity, Anda akan memodifikasi `TextView` "Hello World" dengan teks yang sesuai, dan menambahkan gambar yang digunakan untuk tombol "Add to order".

### Mulai proyek baru

1. Mulai proyek Android Studio baru dengan nama aplikasi **Droid Cafe**. Pilih template **Basic Activity**, terima setelan default, dan klik **Finish**. Proyek terbuka dengan dua layout dalam folder **res > layout**: **activity\_main.xml**, dan **content\_main.xml**.
2. Buka **content\_main.xml** dan ekstrak string "Hello World" di `TextView` untuk menggunakan nama resource `intro_text`. Lalu buka **strings.xml** dan ubah definisi resource `intro_text` untuk menggunakan lebih banyak teks deskriptif, seperti **Droid Desserts**:

```
<string id="intro_text">Droid  
Desserts</string>
```

3. Ubah `TextView` di layout agar menggunakan ukuran teks yang lebih besar `24sp` dan `padding 10dp`, dan tambahkan atribut `android:id` dengan id `textintro`.
4. Ekstrak resource dimensi untuk atribut `android:padding` ke nama resource `padding_regular`, dan atribut `android:textSize` ke resource `text_heading`. Anda akan menggunakan nama resource ini di langkah-langkah selanjutnya.
5. Tambahkan `TextView` lagi di bawah `textintro` `TextView` dengan atribut berikut:

Atribut <code>TextView</code>	Nilai
<code>android:layout_width</code>	"wrap_content"
<code>android:layout_height</code>	"wrap_content"

android:padding	"@dimen/padding_regular"
android:id	"@+id/choose_dessert"
android:layout_below	"@id/textintro"
android:text	"Choose a dessert."

- Ekstrak resource string untuk atribut `android:text` ke nama resource `choose_a_dessert`.

Tambahkan gambar

- Gambar bernama `circle.jpg`, `froyo_circle.jpg`, dan `icecream_circle.jpg` atau gambar lainnya yang bisa Anda unzip di komputer. Untuk menyalin gambar ke proyek Anda, ikuti langkah-langkah ini:
  - ⇒ Tutup proyek Anda.
- Salin file gambar ke dalam folder **drawable** proyek Anda. Temukan folder **drawable** di proyek dengan menggunakan jalur ini: `project_name > app > src > main > res > drawable`
  - ⇒ Buka kembali proyek Anda.
- Buka file **content\_main.xml** lagi dan tambahkan sebuah `ImageView` untuk gambar donat ke layout di bawah tampilan `choose_dessert`, menggunakan atribut berikut:

Atribut <code>ImageView</code> untuk donat	Nilai
<code>android:layout_width</code>	"wrap_content"
<code>android:layout_height</code>	"wrap_content"
<code>android:padding</code>	"@dimen/padding_regular"
<code>android:id</code>	"@+id/donut"
<code>android:layout_below</code>	"@id/choose_dessert"
<code>android:contentDescription</code>	"Donuts are glazed and sprinkled with candy."
<code>android:src</code>	"@drawable/donut_circle"

4. Ekstrak nilai atribut `android:contentDescription` ke resource string `donuts`. Anda akan menggunakan resource string ini di langkah berikutnya.
5. Tambahkan `TextView` yang akan muncul di di sebelah gambar donat sebagai deskripsi, atribut berikut:

<b>Atribut TextView</b>	<b>Nilai</b>
<code>android:layout_width</code>	<code>"wrap_content"</code>
<code>android:layout_height</code>	<code>"wrap_content"</code>
<code>android:padding</code>	<code>"35dp"</code>
<code>android:layout_below</code>	<code>"@+id/choose_dessert"</code>
<code>android:layout_toRightOf</code>	<code>"@id/donut"</code>
<code>android:text</code>	<code>"@string/donuts"</code>

6. Ekstrak resource dimensi untuk atribut `android:padding` ke nama resource `padding_wide`. Anda akan menggunakan nama resource ini di langkah-langkah selanjutnya.
7. Tambahkan `ImageView` kedua ke layout untuk sandwich es krim, menggunakan atribut berikut:

<b>Atribut ImageView untuk ice_cream</b>	<b>Nilai</b>
<code>android:layout_width</code>	<code>"wrap_content"</code>
<code>android:layout_height</code>	<code>"wrap_content"</code>
<code>android:padding</code>	<code>"@dimen/padding_regular"</code>
<code>android:id</code>	<code>"@+id/ice_cream"</code>
<code>android:layout_below</code>	<code>"@id/donut"</code>
<code>android:contentDescription</code>	<code>"Ice cream sandwiches have chocolate wafers and vanilla filling."</code>
<code>android:src</code>	<code>"@drawable/icecream_circle"</code>

8. Ekstrak nilai atribut `android:contentDescription` ke resource string `ice_cream_sandwiches`.
9. Tambahkan `TextView` yang akan muncul di di sebelah sandwich es krim sebagai deskripsi, dengan atribut berikut:

Atribut TextView	Nilai
android:layout_width	"wrap_content"
android:layout_height	"wrap_content"
android:padding	"@dimen/padding_wide"
android:layout_below	"@+id/donut"
android:layout_toRightOf	"@id/ice_cream"
android:text	"@string/ice_cream_sandwiches"

10. Tambahkan ImageView ketiga ke layout untuk froyo, menggunakan atribut berikut:

Atribut ImageView untuk ice_cream	Nilai
android:layout_width	"wrap_content"
android:layout_height	"wrap_content"
android:padding	"@dimen/padding_regular"
android:id	"@+id/froyo"
android:layout_below	"@id/ice_cream"
android:contentDescription	"FroYo is premium self-serve frozen yogurt."
android:src	"@drawable/froyo_circle"

11. Ekstrak nilai atribut android:contentDescription ke resource string froyo.
12. Tambahkan TextView yang akan muncul di sebelah gambar froyo sebagai deskripsi, dengan atribut berikut:

Atribut TextView	Nilai
android:layout_width	"wrap_content"
android:layout_height	"wrap_content"
android:padding	"@dimen/padding_wide"
android:layout_below	"@+id/ice_cream"
android:layout_toRightOf	"@id/froyo"
android:text	"@string/froyo"

⇒ Tambahkan metode onClick untuk tampilan gambar

Anda bisa menambahkan atribut `android:onClick` ke View apa pun untuk membuatnya bisa diklik sebagai tombol. Di langkah ini, Anda akan menambahkan `android:onClick` ke gambar di layout **content\_main.xml**. Anda juga perlu menambahkan metode untuk atribut `android:onClick` yang akan dipanggil. Metode ini, untuk tugas ini, menampilkan pesan toast yang menampilkan gambar mana yang diketuk. (Di tugas berikutnya, Anda akan memodifikasi metode untuk meluncurkan activity lain yang disebut `OrderActivity`.)

⇒ Tambahkan resource string berikut ke file **strings.xml** yang akan ditampilkan di pesan toast:

```
<string name="donut_order_message">You ordered a
donut.</string>
<string name="ice_cream_order_message">You ordered an ice
cream sandwich.</string>
<string name="froyo_order_message">You ordered a
FroYo.</string>
```

Tambahkan metode `displayToast()` berikut untuk menampilkan pesan toast:

```
public void displayToast(String message) {
    Toast.makeText(getApplicationContext(), message,
        Toast.LENGTH_SHORT).show();
}
```

1. Tambahkan metode `showFoodOrder()` berikut di akhir *MainActivity*\* (sebelum tanda kurung penutup). Untuk tugas ini, gunakan metode `displayToast()` untuk menampilkan pesan toast:

```
/**
 * Displays a toast message for the food order
 * and starts the OrderActivity activity.
 * @param message    Message to display.
 */
public void showFoodOrder(String message) {
    displayToast(message);
}
```

**Tip:** Empat baris pertama adalah komentar dalam format Javadoc, yang memudahkan pemahaman kode dan juga membantu membuat dokumentasi untuk kode Anda jika Anda menggunakan [Javadoc](#). Menambahkan komentar seperti itu ke setiap metode baru yang Anda buat merupakan praktik terbaik. Untuk informasi selengkapnya tentang cara menulis komentar.

Meskipun Anda mungkin telah menambahkan metode ini di posisi mana pun di dalam **MainActivity**, sebaiknya tempatkan metode Anda sendiri *di bawah* metode yang telah disediakan di dalam **MainActivity** oleh template.

⇒ Tambahkan metode berikut di akhir **MainActivity** (Anda bisa menambahkannya sebelum `showFoodOrder()`):

```
/**
 * Shows a message that the donut image was clicked.
 */
public void showDonutOrder(View view) {

    showFoodOrder(getString(R.string.donut_order_message));
}

/**
 * Shows a message that the ice cream sandwich image was
 * clicked.
 */
public void showIceCreamOrder(View view) {

    showFoodOrder(getString(R.string.ice_cream_order_message)
);
}

/**
 * Shows a message that the froyo image was clicked.
 */
public void showFroyoOrder(View view) {

    showFoodOrder(getString(R.string.froyo_order_message));
}
```

⇒ Tambahkan atribut `android:onClick` ke tiga `ImageView` di `content_main.xml`:

```
<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="10dp"
    android:id="@+id/donut"
    android:layout_below="@id/choose_dessert"
    android:contentDescription="@string/donut"
    android:src="@drawable/donut_circle"
    android:onClick="showDonutOrder"/>
. . .
<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
        android:padding="10dp"
        android:id="@+id/ice_cream"
        android:layout_below="@id/donut"

        android:contentDescription="@string/ice_cream_sandwich"
        android:src="@drawable/icecream_circle"
        android:onClick="showIceCreamOrder"/>
    . . .
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:id="@+id/froyo"
        android:layout_below="@id/ice_cream"
        android:contentDescription="@string/froyo"
        android:src="@drawable/froyo_circle"
        android:onClick="showFroyoOrder"/>
```

⇒ Jalankan aplikasi.

## Latihan 4.11. Menggunakan tombol radio

Tombol radio adalah kontrol masukan yang berguna untuk memilih hanya satu opsi dari serangkaian opsi. Anda harus menggunakan tombol radio jika Anda ingin pengguna melihat semua opsi yang tersedia secara berdampingan. Jika tidak perlu menampilkan semua opsi secara berdampingan, sebaiknya gunakan spinner.

Nanti dalam praktik ini, Anda akan menambahkan *activity* dan *layout* layar lainnya untuk menyetel opsi pengantaran untuk pesanan makanan, dan menggunakan tombol radio untuk pilihan pengantaran.

⇒ Tambahkan *activity* lain

Sesuai yang telah Anda pelajari di pelajaran sebelumnya, sebuah *activity* merepresentasikan satu layar di aplikasi Anda tempat pengguna Anda bisa melakukan satu tugas yang terfokus. Anda sudah memiliki satu *activity*, `MainActivity.java`. Anda sekarang akan menambahkan *activity* lain untuk menyetel opsi pengiriman untuk pesanan, dan menggunakan *intent* eksplisit untuk meluncurkan *activity* kedua.

1. Klik kanan folder **com.example.android.droidcafe** di kolom kiri dan pilih **New > Activity > Empty Activity**. Edit Nama *Activity* agar menjadi **OrderActivity** dan Nama *Layout* menjadi **activity\_order**. Jangan ubah opsi lain, dan klik **Finish**.

Kelas `OrderActivity` sekarang seharusnya dicantumkan di bawah `MainActivity` di folder **java** dan `activity_order.xml` sekarang seharusnya dicantumkan di folder **layout**. *Template Empty Activity* ditambahkan di file berikut.

2. Buka **MainActivity**. Ubah metode `showFoodOrder()` untuk membuat *intent* eksplisit untuk memulai `OrderActivity`:

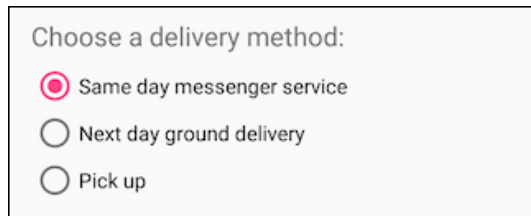
```
public void showFoodOrder(String message) {
    displayToast(message);
    Intent intent = new Intent(this,
    OrderActivity.class);
    startActivity(intent);
}
```

3. Jalankan aplikasi. Mengeklik tombol gambar sekarang akan meluncurkan *activity* kedua, yaitu layar kosong. (Pesan toast muncul sebentar di layar kosong.)

⇒ Tambahkan *layout* untuk tombol radio



Untuk membuat setiap opsi tombol radio, Anda akan membuat elemen `RadioButton` di file layout `activity_order.xml`, yang ditautkan ke `OrderActivity`. Setelah mengedit file layout, layout untuk tombol radio di `OrderActivity` harus tampak seperti gambar di bawah ini.



Karena pilihan tombol radio bersifat eksklusif secara mutual, Anda akan mengelompokkan tombol-tombol ini di dalam `RadioGroup`. Dengan mengelompokkan tombol radio, sistem Android memastikan hanya satu tombol radio yang bisa dipilih dalam satu waktu.

**Catatan:** Urutan cantuman elemen `RadioButton` menentukan urutan tampilannya di layar.

1. Buka `activity_order.xml` dan tambahkan elemen `TextView` dengan id `order_intro_text`:

Atribut <code>TextView</code>	Nilai
<code>android:id</code>	<code>"@+id/order_intro_text"</code>
<code>android:layout_width</code>	<code>"match_parent"</code>
<code>android:layout_height</code>	<code>"wrap_content"</code>
<code>android:layout_marginTop</code>	<code>"24dp"</code>
<code>android:layout_marginBottom</code>	<code>"6dp"</code>
<code>android:textSize</code>	<code>"18sp"</code>
<code>android:text</code>	<code>"Choose a delivery method:"</code>

2. Ekstrak resource string untuk `"Choose a delivery method:"` menjadi `choose_delivery_method`.
3. Ekstrak resource dimensi untuk nilai margin:

```
"24dp" ke text_margin_top
"6dp" ke text_margin_bottom
"18sp" ke intro_text_size
```

- Tambahkan `RadioGroup` ke layout di bawah `TextView` yang baru saja Anda tambahkan:

```
<RadioGroup
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:layout_below="@id/order_intro_text">
</RadioGroup>
```

- Tambahkan tiga elemen `RadioButton` berikut di dalam `RadioGroup` menggunakan atribut berikut. Entri `"onRadioButtonClicked"` untuk atribut `onClick` akan disorot sampai Anda menambahkan metode tersebut di tugas berikutnya.

Atribut #1 <code>RadioButton</code>	Nilai
<code>android:id</code>	<code>"@+id/sameday"</code>
<code>android:layout_width</code>	<code>"wrap_content"</code>
<code>android:layout_height</code>	<code>"wrap_content"</code>
<code>android:text</code>	<code>"Same day messenger service"</code>
<code>android:onClick</code>	<code>"onRadioButtonClicked"</code>
Atribut #2 <code>RadioButton</code>	Nilai
<code>android:id</code>	<code>"@+id/nextday"</code>
<code>android:layout_width</code>	<code>"wrap_content"</code>
<code>android:layout_height</code>	<code>"wrap_content"</code>
<code>android:text</code>	<code>"Next day ground delivery"</code>
<code>android:onClick</code>	<code>"onRadioButtonClicked"</code>
Atribut #3 <code>RadioButton</code>	Nilai
<code>android:id</code>	<code>"@+id/pickup"</code>
<code>android:layout_width</code>	<code>"wrap_content"</code>
<code>android:layout_height</code>	<code>"wrap_content"</code>
<code>android:text</code>	<code>"Pick up"</code>
<code>android:onClick</code>	<code>"onRadioButtonClicked"</code>

- Ekstrak tiga resource string untuk atribut `android:text` ke nama-nama berikut, sehingga string bisa dengan mudah diterjemahkan:

- o same\_day\_messenger\_service
- o next\_day\_ground\_delivery
- o pick\_up

⇒ Tambahkan handler klik tombol radio

Atribut `android:onClick` untuk setiap elemen tombol radio menetapkan metode `onRadioButtonClicked()` untuk menangani kejadian klik. Dengan demikian, Anda perlu menambahkan metode `onRadioButtonClicked()` baru di kelas `OrderActivity`.

Biasanya, aplikasi Anda akan menampilkan pesan mengenai tipe pengantaran seperti apa yang dipilih. Anda akan membuat ini dengan pesan toast dengan membuat metode yang bernama `displayToast()` di `OrderActivity`.

Di metode `onRadioButtonClicked()`, Anda akan menggunakan blok `switch case` untuk memeriksa apakah tombol radio telah diklik. Di akhir blok `switch case`, Anda akan menambahkan pernyataan default yang menampilkan pesan `log` jika tidak satu pun tombol radio yang dicentang.

1. Buka **strings.xml** dan buat resource string berikut:

1. Sebuah resource yang bernama `chosen` untuk string "Chosen: " (sertakan spasi setelah titik dua dan tanda tanya).
2. Sebuah resource yang bernama `nothing_clicked` untuk string "onRadioButtonClicked: Nothing clicked."

2. Buka **OrderActivity** dan tambahkan pernyataan berikut untuk mendefinisikan `TAG_ACTIVITY` untuk pesan `log`:

```
private static final String TAG_ACTIVITY =
    OrderActivity.class.getSimpleName();
```

Tambahkan metode `displayToast` berikut ke `OrderActivity`:

```
public void displayToast(String message) {
    Toast.makeText(getApplicationContext(), message,
        Toast.LENGTH_SHORT).show();
}
```

3. Tambahkan metode `onRadioButtonClicked()` berikut, yang memeriksa untuk melihat apakah tombol radio telah dicentang, dan menggunakan blok `switch case` untuk menentukan item tombol

radio mana yang dipilih, untuk menyetel message yang sesuai untuk item tersebut untuk digunakan bersama `displayToast()`:

```
public void onRadioButtonClicked(View view) {
    // Is the button now checked?
    boolean checked = ((RadioButton)
view).isChecked();
    // Check which radio button was clicked
    switch(view.getId()) {
        case R.id.sameday:
            if (checked)
                // Same day service

displayToast(getString(R.string.chosen) +
getString(R.string.same_day_messenger_service));
            break;
        case R.id.nextday:
            if (checked)
                // Next day delivery

displayToast(getString(R.string.chosen) +
getString(R.string.next_day_ground_delivery));
            break;
        case R.id.pickup:
            if (checked)
                // Pick up

displayToast(getString(R.string.chosen) +
getString(R.string.pick_up));
            break;
        default:
            Log.d(TAG_ACTIVITY,
getString(R.string.nothing_clicked));
            break;
    }
}
```

4. Jalankan aplikasi. Ketuk sebuah pesan untuk melihat activity `OrderActivity`, yang menampilkan pilihan pengantaran. Ketuk pilihan pengantaran dan Anda akan melihat pesan toast di bagian bawah layar dengan pilihan, seperti yang ditampilkan dalam gambar di bawah ini.

## Latihan 4.12. Menambahkan item ke menu opsi

Anda akan membuka proyek Droid Cafe dari praktik sebelumnya, dan menambahkan item menu dari menu opsi di bilah aplikasi di bagian atas layar.

### Periksa kode bilah aplikasi

1. Buka proyek Droid Cafe dari praktik sebelumnya. Proyek menyertakan file layout berikut di folder **res > layout**:
  - ⇒ **activity\_main.xml**: Layout utama untuk MainActivity, layar pertama yang dilihat pengguna.
  - ⇒ **content\_main.xml**: **Layout untuk konten layar MainActivity, yang (seperti akan segera Anda lihat) disertakan di dalam activity\_main.xml\*\*.**
  - ⇒ **activity\_order.xml**: Layout untuk OrderActivity, yang Anda tambahkan di praktik sebelumnya.
2. Buka **content\_main.xml**: Di praktik sebelumnya, Anda telah menambahkan TextView dan ImageView ke grup view root, yang merupakan RelativeLayout.

*Perilaku* layout untuk RelativeLayout disetel ke `@string/appbar_scrolling_view_behavior`, yang mengontrol perilaku pengguliran layar terkait bilah aplikasi di bagian atas.

Klik kanan (Control-klik) resource string ini dan pilih **Go To > Declaration** untuk melihat nilai resource string sebenarnya, yang didefinisikan dalam sebuah file yang bernama "values.xml". File ini dibuat oleh Android Studio, tidak terlihat di Project: Android view dan tidak boleh diedit.

Nilai `@string/appbar_scrolling_view_behavior` sebenarnya di values.xml adalah `"android.support.design.widget.AppBarLayout$ScrollingViewBehavior"`.

3. Buka **activity\_main.xml** untuk melihat layout utama, yang menggunakan layout CoordinatorLayout dengan layout AppBarLayout yang disematkan. Tag CoordinatorLayout dan AppBarLayout memerlukan nama yang sepenuhnya memenuhi kualifikasi yang menetapkan `android.support.design`, yaitu Android Design Support Library.

AppBarLayout adalah LinearLayout yang menggunakan kelas Toolbar di pustaka dukungan, sebagai ganti ActionBar, untuk mengimplementasikan bilah aplikasi. Bilah aplikasi adalah bagian di atas layar yang bisa menampilkan judul aktivitas, navigasi, dan item interaktif lainnya. ActionBar asli berperilaku berbeda bergantung pada versi Android yang berjalan pada perangkat. Oleh karena itu, jika Anda menambahkan menu opsi, Anda harus menggunakan Toolbar pustaka dukungan v7 appcompat sebagai bilah aplikasi. Menggunakan Toolbar akan memudahkan penyiapan bilah aplikasi yang bekerja pada beragam perangkat, serta memberi Anda ruang untuk menyesuaikan bilah aplikasi Anda nanti seiring pengembangan aplikasi. Bilah alat menyertakan berbagai fitur terbaru, dan bekerja pada perangkat apa pun yang bisa menggunakan pustaka dukungan.

Toolbar di dalam layout ini memiliki id `toolbar`, dan juga ditetapkan, seperti AppBarLayout, dengan nama yang berkualifikasi sepenuhnya (`android.support.v7.widget`):

```
<android.support.design.widget.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:theme="@style/AppTheme.AppBarOverlay">

    <android.support.v7.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimary"
        app:popupTheme="@style/AppTheme.PopupOverlay" />

</android.support.design.widget.AppBarLayout>
```

4. Jalankan aplikasi. Perhatikan bilah di bagian atas layar yang menampilkan nama aplikasi (Droid Cafe). Ini juga menampilkan tombol *luapan tindakan* (tiga titik vertikal) di sebelah kanan. Ketuk tombol luapan untuk melihat menu opsi, yang di tahap ini hanya memiliki satu menu opsi, **Settings**.
5. Periksa file **AndroidManifest.xml**. Aktivitas `.MainActivity` disetel untuk menggunakan tema `NoActionBar`:
6. `android:theme="@style/AppTheme.NoActionBar"`

Tema `NoActionBar` didefinisikan di file **styles.xml** (luaskan expand **app > res > values > styles.xml** untuk melihatnya). Gaya dicakup di pelajaran lainnya, tetapi Anda bisa melihat bahwa tema `NoActionBar` menyetel atribut `windowActionBar` ke `false` (tanpa bilah tindakan jendela) dan atribut `windowNoTitle` ke `true` (tanpa judul).

Alasan mengapa nilai ini disetel adalah Anda mendefinisikan bilah aplikasi di layout (`activity_main.xml`) dengan `AppBarLayout`, bukan menggunakan `ActionBar`. Menggunakan salah satu tema `NoActionBar` mencegah aplikasi menggunakan kelas `ActionBar` asli untuk menyediakan bilah aplikasi. Kelas `ActionBar` asli berperilaku berbeda, bergantung pada versi sistem Android yang sedang digunakan perangkat. Sebaliknya, fitur terbaru ditambahkan ke pustaka dukungan versi `Toolbar`, dan tersedia pada setiap perangkat apa pun yang bisa menggunakan pustaka dukungan. Dengan alasan ini, Anda harus menggunakan pustaka dukungan kelas `Toolbar` untuk mengimplementasikan aktivitas bilah aplikasi Anda, bukannya `ActionBar`. Menggunakan Bilah Alat pustaka dukungan memastikan aplikasi Anda akan memiliki perilaku yang konsisten pada rangkaian perangkat yang paling luas.

7. Lihat **MainActivitMainActivity** yang memerlukan `AppCompatActivity` dan dimulai dengan metode `onCreate()`:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Toolbar toolbar = (Toolbar)
    findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
    ...
}
```

Setelah menyetel tampilan konten ke layout `activity_main.xml`, metode `onCreate()` menyetel toolbar menjadi `Toolbar` yang didefinisikan di layout `activity_main.xml`. Ini kemudian memanggil metode `setSupportActionBar()` aktivitas dan meneruskan toolbar padanya, menyetel toolbar yang didefinisikan di **activity\_main.xml** sebagai bilah aplikasi untuk aktivitas.

## Menambahkan lebih banyak item menu ke menu opsi

Anda akan menambahkan item menu berikut ke menu opsi aplikasi Droid Cafe:

- ⇒ **Order:** Masuk ke layar Order Activity untuk melihat pesanan makanan.
- ⇒ **Status:** Memeriksa status pesanan makanan.
- ⇒ **Favorites:** Menampilkan makanan favorit.
- ⇒ **Contact:** Menghubungi restoran. Karena Anda tidak memerlukan item **Settings** yang ada, anda akan mengubah **Settings** ke **Contact**.

Android menyediakan format XML standar untuk mendefinisikan item menu. Sebagai ganti membangun menu di kode activity Anda, Anda bisa mendefinisikan menu dan semua item menunya dalam resource menu XML. Anda kemudian bisa memekarkan resource menu (muat sebagai objek Menu) di activity atau fragmen Anda:

1. Lihat **menu\_main.xml** (luaskan **res > menu** di tampilan Project). Ini mendefinisikan item menu dengan `<item>` `</item>` di dalam `<menu>` `</menu>` block. Satu-satunya item menu yang tersedia dari template adalah 'action\_settings' (pilihan **Settings**) yang didefinisikan sebagai:

```
<item
    android:id="@+id/action_settings"
    android:orderInCategory="100"
    android:title="@string/action_settings"
    app:showAsAction="never" />
```

Di Android Studio, atribut `android:title` menampilkan nilai string "Settings" meskipun string didefinisikan sebagai resource. Android Studio menampilkan nilai sehingga Anda bisa melihat sekilas berapa nilainya tanpa harus membuka file resource strings.xml. Jika Anda mengeklik string ini, string berubah untuk menampilkan resource string `"@string/action_settings"`.

2. Ubah atribut item `action_settings` berikut untuk membuatnya sebagai item `action_contact` (jangan ubah atribut `android:orderInCategory` yang sudah ada):

Atribut	Nilai
<code>android:id</code>	<code>"@+id/action_contact"</code>
<code>android:title</code>	<code>"Contact"</code>
<code>app:showAsAction</code>	<code>"never"</code>



- Ekstrak string "Contact" yang di-hardcode ke dalam resource string `action_contact`.
- Tambahkan item menu baru menggunakan blok `<item> </item>` di dalam `<menu> </menu>`, dan berikan atribut berikut ke item:

Atribut	Nilai
<code>android:id</code>	<code>"@+id/action_order"</code>
<code>android:orderInCategory</code>	<code>"10"</code>
<code>android:title</code>	<code>"Order"</code>
<code>app:showAsAction</code>	<code>"never"</code>

- Atribut `android:orderInCategory` menetapkan urutan kemunculan item menu di menu, dengan angka terkecil muncul lebih tinggi dalam menu. Item Contact disetel ke 100, yang merupakan nomor besar untuk menetapkan bahwa ini akan ditampilkan di bagian bawah, bukan di bagian atas. Anda menyetel item Order ke 10, yang menempatkannya di atas Contact, dan meninggalkan cukup banyak ruang di menu untuk lebih banyak item.
- Ekstrak string "Order" yang di-hardcode ke dalam resource string `action_order`.
- Tambahkan dua item menu lagi dengan cara yang sama dengan atribut berikut:

Atribut Item Status	Nilai
<code>android:id</code>	<code>"@+id/action_status"</code>
<code>android:orderInCategory</code>	<code>"20"</code>
<code>android:title</code>	<code>"Status"</code>
<code>app:showAsAction</code>	<code>"never"</code>
Atribut Item Favorit	Nilai
<code>android:id</code>	<code>"@+id/action_favorites"</code>
<code>android:orderInCategory</code>	<code>"40"</code>
<code>android:title</code>	<code>"Favorites"</code>
<code>app:showAsAction</code>	<code>"never"</code>

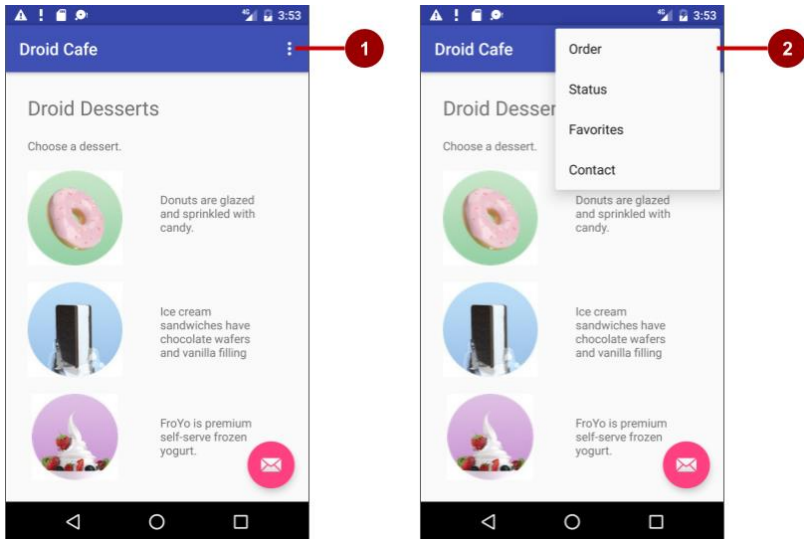
9. Ekstrak "Status" ke dalam resource `action_status` dan "Favorites" ke dalam resource `action_favorites`.
10. Anda akan menampilkan pesan toast dengan pesan tindakan bergantung pada item menu apa yang dipilih pengguna. Tambahkan nama dan nilai string berikut di `strings.xml` untuk pesan ini:

```
<string name="action_order_message">You selected
Order.</string>
<string name="action_status_message">You selected
Status.</string>
<string name="action_favorites_message">You selected
Favorites.</string>
<string name="action_contact_message">You selected
Contact.</string>
```

11. Buka **MainActivity** dan ubah pernyataan `if` di metode `onOptionsItemSelected()` dengan mengganti id `action_settings` dengan id `action_order` baru:

```
if (id == R.id.action_order)
```

Jalankan aplikasi dan ketuk ikon luapan tindakan, yang ditampilkan di sebelah kiri gambar di bawah ini, untuk melihat menu opsi, yang ditampilkan di sebelah kanan gambar di bawah ini. Anda akan segera menambahkan callback untuk merespons item yang dipilih dari menu ini.



Dalam gambar di atas:

1. Ketuk ikon luapan di bilah aplikasi untuk melihat menu opsi.
2. Menu opsi akan ditarik menurun dari bilah aplikasi.

Perhatikan urutan item di menu opsi. Anda menggunakan atribut `android:orderInCategory` untuk menetapkan prioritas item menu di menu: Item Order adalah 10, diikuti oleh Status (20) dan Favorites (40), dan Contact berada di urutan terakhir (100). Tabel berikut menampilkan prioritas item di menu:

Item Menu	Atribut <code>orderInCategory</code>
Order	10
Status	20
Favorites	40
Contact	100

### Menambahkan ikon untuk item menu

Jika memungkinkan, Anda bisa menampilkan tindakan yang paling sering digunakan menggunakan ikon di bilah aplikasi, sehingga pengguna bisa mengkliknya tanpa harus mengklik ikon luapan terlebih dahulu. Dalam tugas ini, Anda akan menambahkan ikon untuk beberapa item menu dan menampilkan beberapa item menu di bilah aplikasi di bagian atas layar sebagai ikon.

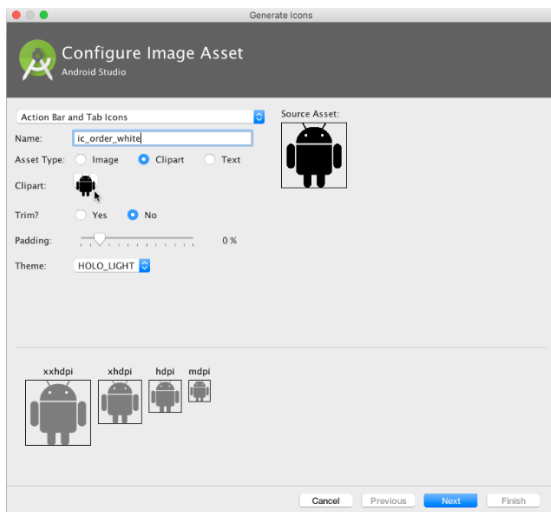
Dalam contoh ini, misalkan tindakan **Order** dan *Status dianggap sebagai paling sering digunakan. Favorites kadang-kadang digunakan, dan Contact\** paling jarang digunakan. Anda bisa menyetel ikon untuk tindakan ini dan menentukan yang berikut:

- **Order** dan **Status** harus selalu ditampilkan di bilah aplikasi.
- **Favorites** harus ditampilkan di bilah aplikasi jika pas ukurannya, jika tidak, harus tampil di menu luapan.
- **Contact** harus tidak muncul di bilah aplikasi, dan *hanya* muncul di menu luapan.

### Menambahkan ikon untuk item menu

Untuk menetapkan ikon untuk tindakan, Anda terlebih dulu perlu menambahkan ikon sebagai aset gambar di folder **drawable**.

1. Luaskan **res** di tampilan Project, dan klik kanan (atau Kontrol-klik) **drawable**.
2. Pilih **New > Image Asset**. Dialog Configure Image Asset akan muncul.
3. Pilih **Action Bar and Tab Items** di menu tarik-turun.
4. Ubah **ic\_action\_name** ke **ic\_order\_white** (untuk tindakan **Order**).



5. Klik gambar clipart (logo Android di sebelah "Clipart") untuk memilih gambar clipart sebagai ikon. Laman ikon akan muncul. Klik ikon yang

ingin Anda gunakan untuk tindakan **Order** (misalnya, ikon keranjang belanja mungkin sesuai).

6. Pilih **HOLO\_DARK** dari menu tarik-turun Theme. Ini akan menyetel ikon menjadi putih dengan latar belakang berwarna gelap (atau hitam). Klik **Next**.
7. Klik **Finish**, dalam dialog Confirm Icon Path.
8. Ulang langkah-langkah di atas untuk ikon **Status** dan **Favorites**, dan beri nama masing-masing **ic\_status\_white** dan **ic\_favorites\_white**. Anda mungkin ingin menggunakan ikon i dalam lingkaran untuk **Status** (umumnya digunakan untuk Info) dan ikon hati untuk **Favorites**.

### Tampilkan item menu sebagai ikon di bilah aplikasi

Untuk menampilkan item menu sebagai ikon di bilah aplikasi, gunakan atribut `app:showAsAction` di **menu\_main.xml**. Nilai berikut untuk atribut akan menetapkan apakah tindakan akan muncul atau tidak di bilah aplikasi sebagai ikon:

- `"always"`: Selalu muncul di bilah aplikasi. (Jika tidak cukup ruang, ikon bisa menutupi ikon menu lainnya.)
- `"ifRoom"`: Muncul di bilah aplikasi jika ada ruang.
- `"never"`: Tidak pernah muncul di bilah aplikasi; teksnya muncul di menu luapan.

Ikuti langkah-langkah berikut untuk menampilkan beberapa item menu sebagai ikon:

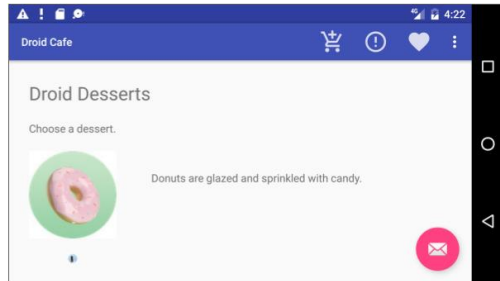
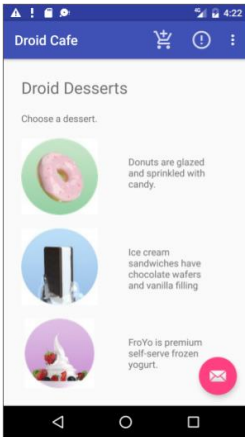
1. Buka **menu\_main.xml** lagi dan tambahkan atribut berikut ke item **Order**, **Status**, dan **Favorites** sehingga dua item pertama (**Order** dan **Status**) selalu muncul, dan item **Favorites** hanya muncul jika terdapat ruang untuknya:

Atribut Item Order	Nilai Lama	Nilai Baru
<code>android:icon</code>		<code>"@drawable/ic_order_white"</code>
<code>app:showAsAction</code>	<code>"never"</code>	<code>"always"</code>
Atribut Item Status	Nilai Lama	Nilai Baru
<code>android:icon</code>		<code>"@drawable/ic_status_white"</code>
<code>app:showAsAction</code>	<code>"never"</code>	<code>"always"</code>
Atribut Item Favorit	Nilai Lama	Nilai Baru

android:icon		"@drawable/ic_favorites_white"
app:showAsAction	"never"	"ifRoom"

- Jalankan aplikasi. Anda sekarang seharusnya melihat paling tidak dua ikon di bilah aplikasi: ikon untuk **Order** dan ikon untuk **Status** seperti yang ditampilkan dalam gambar di bawah ini. Jika perangkat atau emulator Anda menampilkan orientasi vertikal, opsi **Favorites** dan **Contact** muncul di menu luapan.
- Putar perangkat Anda ke orientasi horizontal, atau jika Anda menjalankan emulator, klik ikon **Rotete Left** atau **Rotate Right** untuk memutar layar ke orientasi horizontal. Anda seharusnya melihat ketiga ikon di bilah aplikasi untuk **Order**, **Status**, dan **Favorites**.

**Tip:** Berapa banyak tombol tindakan yang dapat masuk di bilah aplikasi? Bergantung pada orientasi dan ukuran layar perangkat. Lebih sedikit tombol akan muncul di orientasi vertikal, seperti yang ditampilkan di sebelah kiri gambar di bawah, dibandingkan dengan orientasi horizontal seperti yang ditampilkan di sebelah kanan gambar di bawah. Tombol tindakan mungkin tidak menempati lebih dari separuh lebar bilah aplikasi utama.



## **Tantangan:**

Saat Anda mengeklik tombol tindakan mengambang dengan ikon email yang muncul di bagian bawah layar, kode di MainActivity menampilkan panel samping yang membuka dan menutup, yang disebut sebagai **snackbar**. Snackbar memberikan umpan balik mengenai sebuah operasi—yaitu menampilkan pesan singkat di bagian bawah layar ponsel cerdas atau di bagian sudut kiri bawah di perangkat yang lebih besar. Untuk informasi selengkapnya.

Pelajari bagaimana aplikasi lain mengimplementasikan tombol tindakan mengambang. Misalnya, aplikasi Gmail menyediakan tombol tindakan mengambang untuk membuat pesan email baru dan aplikasi Kontak menyediakan tombol untuk membuat kontak baru.

Cara menambahkan ikon untuk item menu, gunakan teknik yang sama untuk menambahkan ikon lain dan menetapkan ikon tersebut pada tombol tindakan mengambang, yang menggantikan ikon email. Misalnya, Anda mungkin ingin tombol aksi mengambang memulai sesi chat, yang dalam hal ini Anda mungkin ingin menggunakan ikon yang menampilkan wajah manusia.

**Petunjuk:** Tombol Aksi Mengambang didefinisikan di **activity\_main.xml**.

Saat menambahkan ikon, ubah juga teks yang muncul di snackbar setelah mengetuk tombol aksi mengambang. Anda akan menemukan teks ini di pernyataan `Snackbar.make` di aktivitas utama. Ekstrak resource string untuk teks ini agar menjadi `snackbar_text`.

### Latihan 4.13. Menangani item menu yang dipilih

Dalam tugas ini, Anda akan menambahkan metode untuk menampilkan pesan tentang item menu mana yang diketuk dan menggunakan metode `onOptionsItemSelected` untuk menentukan item menu mana yang diketuk.

#### Buat metode untuk menampilkan pilihan menu

1. Buka **MainActivity**.
2. Jika Anda belum menambahkan metode berikut (di pelajaran sebelumnya) untuk menampilkan pesan toast, tambahkan sekarang:

```
public void displayToast(String message) {  
    Toast.makeText(getApplicationContext(), message,  
                    Toast.LENGTH_SHORT).show();  
}
```

Metode `displayToast()` mengambil `message` dari string yang sesuai (misalnya `action_contact_message`).

#### Gunakan handler kejadian `onOptionsItemSelected`

Metode `onOptionsItemSelected` menangani pilihan dari menu opsi. Anda akan menambahkan blok `switch case` untuk menentukan item menu mana yang dipilih dan `message` mana yang akan dibuat untuk setiap item yang dipilih. (Sebagai ganti membuat `message` untuk setiap item, Anda bisa mengimplementasikan handler kejadian untuk setiap item yang melakukan sebuah tindakan, misalnya memulai aktivitas lainnya, seperti yang ditampilkan nanti di pelajaran ini.)

1. Temukan metode `onOptionsItemSelected()`. Pernyataan `if` di metode yang disediakan oleh template ini, menentukan apakah item menu tertentu diklik, menggunakan `id` menu item (`action_order` di contoh berikut):

```
@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
    int id = item.getItemId();  
    if (id == R.id.action_order) {  
        return true;  
    }  
    return super.onOptionsItemSelected(item);  
}
```



2. Ganti pernyataan if dan penetapannya ke id dengan blok switch case berikut yang menyetel message yang sesuai berdasarkan id item menu:

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.action_order:

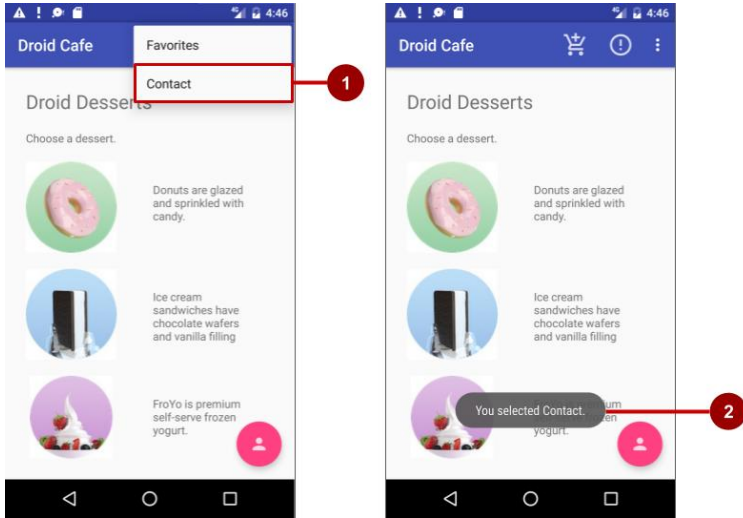
displayToast (getString(R.string.action_order_message));
        return true;
        case R.id.action_status:

displayToast (getString(R.string.action_status_message));
        return true;
        case R.id.action_favorites:

displayToast (getString(R.string.action_favorites_message));
        return true;
        case R.id.action_contact:

displayToast (getString(R.string.action_contact_message));
        return true;
        default:
            // Do nothing
    }
    return super.onOptionsItemSelected(item);
}
```

3. Jalankan aplikasi. Anda sekarang seharusnya melihat pesan toast berbeda di layar, seperti yang ditampilkan di sebelah kanan gambar di bawah, berdasarkan item menu mana yang Anda pilih.



Di gambar di atas:

1. Memilih item **Contact** di menu opsi.
2. Pesan toast yang muncul.

## Tantangan 2

Di tantangan sebelumnya, Anda telah mengubah ikon untuk tombol aksi mengambang yang muncul di bagian bawah layar `MainActivity` di aplikasi Anda.

Untuk tantangan ini:

1. Ubah ikon untuk tombol aksi mengambang lagi, tetapi kali ini menjadi ikon yang sesuai untuk peta, misalnya ikon dunia.
2. Di `MainActivity`, ganti tindakan untuk menampilkan `Snackbar` dengan `intent` implisit untuk meluncurkan aplikasi `Maps` saat tombol aksi mengambang diketuk.
3. Tambahkan koordinat khusus berikut (untuk kantor pusat Google) dan tingkat `zoom` (12) ke sebuah `string` yang bernama `google_mtv_coord_zoom12`:

```
4. <string
    name="google_mtv_coord_zoom12">geo:37.422114,-
    122.086744?z=12</string>
```

Tambahkan metode berikut untuk memulai aplikasi `Maps`, yang meneruskan `string` di atas sebagai data menggunakan `intent` implisit:

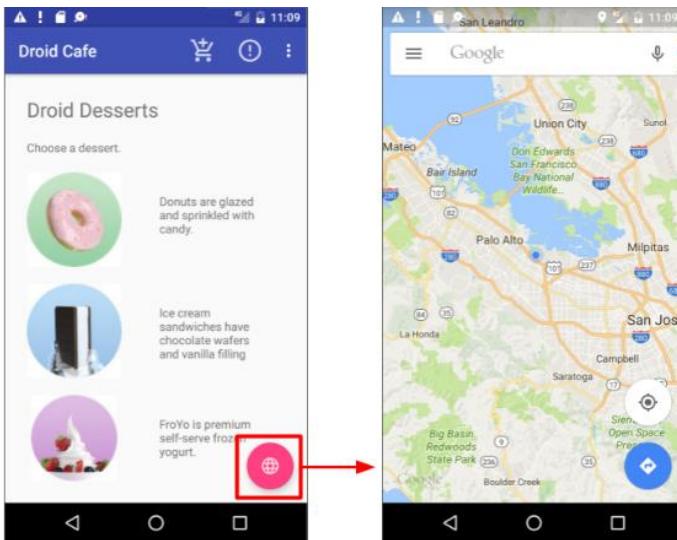
```

public void displayMap() {
    Intent intent = new Intent();
    intent.setAction(Intent.ACTION_VIEW);
    // Using the coordinates for Google headquarters.
    String data =
    getString(R.string.google_mtv_coord_zoom12);
    intent.setData(Uri.parse(data));
    if (intent.resolveActivity(getPackageManager()) !=
    null) {
        startActivity(intent);
    }
}
}

```

Contoh intent implisit, termasuk membuka aplikasi Maps.

Setelah mengetuk tombol aksi mengambang untuk beralih ke aplikasi Maps, seperti yang ditampilkan dalam gambar di bawah ini, pengguna bisa mengetuk tombol **Kembali** di bawah layar untuk kembali ke aplikasi Anda.



Tambahkan elemen UI lainnya— Tombol—ke grup tampilan LinearLayout yang berada di dalam ScrollView. Munculkan Tombol di bawah artikel. Pengguna harus menggulir ke bagian akhir artikel untuk melihat tombolnya. Gunakan teks "Add Comment" untuk Tombol, agar pengguna mengkliknya untuk menambahkan komentar ke artikel. Untuk tantangan ini, tidak perlu membuat metode penanganan tombol untuk bisa menambahkan komentar; cukup letakkan elemen Tombol di tempat yang tepat dalam layout.

## RANGKUMAN

Dalam Praktik ini, Anda telah belajar tentang elemen tampilan Android Studio dan cara menggulir serta melapiskan kode. Anda bekerja untuk:

- 1) Menambahkan beberapa elemen TextView ke layout XML.
  - a. Menampilkan teks bentuk bebas dalam TextView dengan tag pemformatan HTML untuk cetak tebal dan miring.
  - b. Menggunakan `\n` sebagai karakter akhir baris dalam teks bentuk bebas untuk menjaga agar satu paragraf tidak menabrak paragraf berikutnya.
  - c. Gunakan atribut `android:autoLink="web"` untuk membuat tautan web dalam teks dapat diklik.
- 2) Menambahkan grup tampilan ScrollView ke layout untuk mendefinisikan tampilan bergulir dengan salah satu elemen TextView.
- 3) Menambahkan grup tampilan LinearLayout di dalam ScrollView untuk menggulir beberapa elemen TextView bersama-sama.
- 4) Mengekstrak nilai string ke dalam nama string dalam file strings.xml untuk penerjemahan resource string yang lebih mudah.

## REFERENSI

- Buku Kursus Dasar-Dasar Developer Android – Konsep, Belajar mengembangkan aplikasi Android, Dikembangkan oleh Tim Pelatihan Developer Google.
- Buku Kerja Praktik, Kursus Dasar-Dasar Developer Android — Praktik, Belajar mengembangkan aplikasi Android, Dikembangkan oleh Tim Pelatihan Developer Google

## MATERI KOMPETENSI

Materi bab ini akan membahas:

- 1) Tentang activity dan membuat activity
- 2) Tentang intent
- 3) Memulai activity dengan intent eksplisit
- 4) Mendapatkan data kembali dari activity
- 5) Navigasi activity

## YANG HARUS SUDAH ANDA KETAHUI

Untuk materi dan praktik di bab ini Anda harus bisa:

- 1) Membuat dan menjalankan aplikasi dalam Android Studio.
- 2) Membuat dan mengedit elemen UI dengan Layout Editor Grafis atau langsung di file layout XML.
- 3) Menambahkan fungsionalitas onClick ke tombol.

## YANG AKAN ANDA BUTUHKAN

Untuk materi dan praktik ini, Anda akan membutuhkan:

- 1) Komputer Mac, Windows, atau Linux. Tool Android Studio untuk persyaratan sistem terbaru.
- 2) Akses internet atau cara alternatif untuk memuat Android Studio dan pemasangan Java terbaru ke komputer Anda.

## YANG AKAN ANDA PELAJARI

Anda akan belajar:

- 1) Membuat intent implisit dan menggunakan tindakan dan kategorinya.
- 2) Membuat activity baru di Android Studio.
- 3) Mendefinisikan activity induk dan anak untuk navigasi.
- 4) Memulai activity dengan intent eksplisit.

# BAB 5 ACTIVITY DAN INTENT



## Bab 5: Activity dan Intent

### 5.1. Tentang Activity

Activity menyatakan layar tunggal di aplikasi Anda dengan antarmuka yang bisa digunakan pengguna untuk berinteraksi. Misalnya, aplikasi email mungkin memiliki satu activity yang menampilkan daftar email baru, activity lain untuk menulis email, dan activity lainnya lagi untuk membaca pesan satu per satu. Aplikasi Anda adalah koleksi activity yang dibuat sendiri atau yang digunakan kembali dari aplikasi lain.

Meskipun activity di aplikasi Anda bekerja sama membentuk pengalaman pengguna yang kohesif di aplikasi, setiap aplikasi tidak saling bergantung. Proses ini memungkinkan aplikasi memulai activity di aplikasi lainnya, dan aplikasi lainnya bisa memulai activity Anda (jika aplikasi mengizinkannya). Misalnya, aplikasi perpesanan yang Anda tulis bisa memulai activity di aplikasi kamera untuk mengambil gambar, kemudian memulai activity di aplikasi email untuk memungkinkan pengguna berbagi gambar itu di email.



Gambar 95. Activity

Umumnya, satu activity di aplikasi ditetapkan sebagai activity "utama", yang disajikan kepada pengguna saat membuka aplikasi untuk pertama kali. Kemudian setiap activity bisa memulai activity lainnya untuk melakukan tindakan yang berbeda.

Setiap kali activity baru dimulai, activity sebelumnya akan dihentikan, namun sistem mempertahankan activity dalam tumpukan ("back-stack"). Bila pengguna selesai dengan activity saat ini dan menekan tombol Kembali, activity akan muncul dari tumpukan (dan dimusnahkan) dan activity sebelumnya dilanjutkan.

Bila activity dihentikan karena activity baru dimulai, activity pertama akan diberi tahu tentang perubahan tersebut dengan metode callback daur hidup activity. Daur hidup Activity adalah serangkaian keadaan activity, mulai dari pertama kali dibuat, hingga setiap kali dihentikan atau dilanjutkan, hingga bila sistem memusnahkannya. Anda akan mengetahui selengkapnya tentang daur hidup activity di bab berikutnya.

### Membuat Activity

Untuk mengimplementasikan activity di aplikasi Anda, lakukan yang berikut ini:

- Buat kelas Java activity.
- Implementasikan antarmuka pengguna untuk activity itu.
- Deklarasikan activity baru itu di manifes aplikasi.

Bila Anda membuat proyek baru untuk aplikasi atau menambahkan activity baru ke aplikasi, di Android Studio (dengan File > New > Activity), kode template untuk setiap tugas ini akan disediakan untuk Anda.

#### ▪ Buat kelas activity

Activity adalah subkelas dari kelas Activity atau salah satu dari subkelasnya. Jika Anda membuat proyek baru di Android Studio, activity tersebut secara default menjadi subkelas dari kelas AppCompatActivity. Kelas AppCompatActivity adalah subkelas Activity yang memungkinkan Anda menggunakan fitur aplikasi Android terbaru seperti bilah aksi dan desain material, sementara tetap memungkinkan aplikasi tersebut kompatibel dengan perangkat yang menjalankan Android versi lama.

Inilah subkelas kerangka dari activity AppCompatActivity:

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
```



```
super.onCreate(savedInstanceState);  
setContentView(R.layout.activity_main);  
}  
}
```

Tugas pertama Anda di subkelas activity adalah mengimplementasikan metode callback daur hidup activity standar (seperti `onCreate()`) untuk menangani perubahan keadaan activity. Perubahan keadaan ini meliputi hal-hal seperti kapan activity dibuat, dihentikan, dilanjutkan, atau dimusnahkan. Anda akan mengetahui selengkapnya tentang daur hidup activity dan callback daur hidup di bab berikutnya.

Salah satu callback yang diperlukan dan harus diimplementasikan oleh aplikasi Anda adalah metode `onCreate()`. Sistem memanggil metode ini bila membuat activity, dan semua komponen penting activity Anda harus melakukan diinisialisasi di sini. Yang paling penting, metode `onCreate()` memanggil `setContentView()` untuk membuat layout utama activity.

Anda biasanya mendefinisikan antarmuka pengguna untuk activity di satu atau beberapa file layout XML. Bila metode `setContentView()` dipanggil dengan jalur ke file layout, sistem akan membuat semua tampilan awal dari layout yang ditetapkan dan menambahkannya ke activity Anda. Hal ini sering kali disebut sebagai *memekarkan* layout.

Sering kali Anda mungkin juga ingin mengimplementasikan metode `onPause()` di kelas activity. Sistem memanggil metode ini sebagai indikasi pertama bahwa pengguna meninggalkan activity Anda (walaupun tidak selalu berarti activity akan dimusnahkan). Di sinilah biasanya Anda harus mengikat perubahan yang harus dipertahankan di luar sesi pengguna saat ini (karena pengguna mungkin tidak akan kembali). Anda akan mengetahui selengkapnya tentang callback `onPause()` dan semua callback daur hidup lainnya di bab berikutnya. Selain daur hidup callback, Anda juga bisa mengimplementasikan metode di activity untuk menangani perilaku lainnya seperti masukan pengguna atau klik tombol.

### ▪ **Implementasikan antarmuka pengguna**

Antarmuka pengguna untuk activity disediakan menurut hierarki tampilan, yang mengontrol ruang tertentu dalam jendela activity dan bisa merespons interaksi pengguna.

Cara yang paling umum untuk mendefinisikan antarmuka pengguna yang menggunakan tampilan adalah dengan file layout XML yang disimpan sebagai bagian dari resource aplikasi Anda. Mendefinisikan layout di XML

memungkinkan Anda untuk mengelola desain antarmuka pengguna secara terpisah dari kode sumber yang mendefinisikan perilaku activity.

Anda juga bisa membuat tampilan baru secara langsung di kode activity Anda dengan menyisipkan objek tampilan baru ke dalam ViewGroup, kemudian meneruskan ViewGroup akar ke setContentView(). Setelah layout dimekarkan -- apa pun sumbernya -- Anda bisa menambahkan lebih banyak tampilan di Java di mana saja dalam hierarki tampilan.

### ▪ Deklarasikan activity di manifes

Setiap activity di aplikasi Anda harus dideklarasikan di manifes aplikasi Android bersama elemen <activity>, di dalam <application>. Bila Anda membuat proyek baru atau menambahkan activity baru ke proyek di Android Studio, manifes akan dibuat atau diperbarui untuk menyertakan deklarasi activity kerangka bagi setiap activity. Inilah deklarasi untuk activity utama.

```
<activity android:name=".MainActivity" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN"
/>
        <category
android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

Elemen <activity> menyertakan sejumlah atribut untuk mendefinisikan properti activity seperti label, ikon, atau tema. Satu-satunya atribut yang diperlukan adalah android:name, yang menetapkan nama kelas activity (seperti "MainActivity"). Lihat referensi elemen <activity> untuk informasi selengkapnya tentang deklarasi activity.

Elemen <activity> juga bisa menyertakan deklarasi untuk filter intent. Filter intent menetapkan jenis intent yang akan diterima activity Anda.

```
<intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category
android:name="android.intent.category.LAUNCHER" />
</intent-filter>
```

Filter intent harus menyertakan setidaknya satu elemen, dan juga bisa menyertakan sebuah dan opsional. Activity utama untuk aplikasi Anda

memerlukan filter intent yang mendefinisikan aksi "main" dan kategori "launcher" agar sistem bisa meluncurkan aplikasi. Android Studio membuat filter intent ini untuk activity utama di proyek Anda:

Elemen menetapkan bahwa ini adalah titik masuk "utama" ke aplikasi. Elemen menetapkan bahwa activity ini harus tercantum dalam peluncur aplikasi sistem (untuk memungkinkan pengguna meluncurkan activity ini).

Activity lain di aplikasi Anda juga bisa mendeklarasikan filter intent, namun hanya activity utama yang harus menyertakan aksi "utama". Anda akan mengetahui selengkapnya tentang intent implisit dan filter intent dalam bab berikutnya.

### ▪ **Tambahkan lebih banyak activity ke proyek Anda**

Activity utama untuk aplikasi Anda dan file layout terkait disertakan bersama proyek bila Anda membuatnya. Anda bisa menambahkan activity baru ke proyek di Android Studio dengan menu **File > New > Activity**. Pilih template activity yang ingin Anda gunakan, atau buka Gallery untuk melihat semua template yang tersedia.

Bila memilih sebuah template activity, Anda akan melihat serangkaian layar yang sama untuk membuat activity baru yang dilakukan saat membuat proyek di awal. Android Studio menyediakan tiga hal ini untuk setiap activity baru di aplikasi Anda:

- File Java untuk activity baru dengan definisi kelas kerangka dan metode onCreate(). Activity baru, seperti activity utama, adalah subclass AppCompatActivity.
- File XML yang berisi layout untuk activity baru. Perhatikan, metode setContentView() di kelas activity akan memerkan layout baru ini.

Elemen <activity> tambahan di manifest Android yang menetapkan activity baru. Definisi activity kedua tidak menyertakan filter intent apa pun. Jika Anda ingin menggunakan activity ini hanya dalam aplikasi (dan tidak memungkinkan activity tersebut dimulai oleh aplikasi lain), maka tidak perlu menambahkan filter.

## **5.2. Tentang Intent**

Semua activity Android dimulai atau diaktifkan dengan *intent*. Intent adalah objek pesan yang membuat permintaan yang akan digunakan oleh waktu proses Android untuk memulai activity atau komponen aplikasi lainnya di

aplikasi Anda atau di beberapa aplikasi lainnya. Anda tidak bisa memulai activity itu sendiri;

Bila aplikasi pertama kali dimulai dari layar utama perangkat, waktu proses Android akan mengirimkan intent ke aplikasi Anda untuk memulai activity utama aplikasi (yang didefinisikan dengan aksi MAIN dan kategori LAUNCHER di Manifest Android). Untuk memulai activity lain di aplikasi Anda, atau meminta tindakan untuk dilakukan oleh beberapa activity lain yang tersedia di perangkat, bangunlah intent sendiri dengan kelas Intent dan panggil metode `startActivity()` untuk mengirim intent itu.

Selain untuk memulai activity, intent juga digunakan untuk meneruskan data di antara activity. Bila membuat intent untuk memulai activity baru, Anda bisa menyertakan informasi tentang data yang diinginkan untuk mengoperasikan activity baru itu. Jadi, misalnya, activity email yang menampilkan daftar pesan bisa mengirim intent ke activity yang menampilkan pesan itu. Activity tampilan memerlukan data tentang pesan yang akan ditampilkan, dan Anda bisa menyertakan data itu di intent.

Dalam bab ini, Anda akan mempelajari tentang penggunaan intent bersama activity, namun intent juga digunakan untuk memulai layanan dan penerima siaran.

Ada dua tipe intent di Android:

- *Intent eksplisit* menetapkan activity penerima (atau komponen lainnya) melalui nama kelas yang benar-benar memenuhi syarat di activity tersebut. Gunakan intent eksplisit untuk memulai komponen di aplikasi milik sendiri (misalnya, untuk beralih layar di antarmuka pengguna), karena Anda sudah mengetahui paket dan nama kelas komponen itu.
- *Intent implisit* tidak menetapkan activity tertentu atau komponen lainnya untuk menerima intent. Sebagai gantinya, Anda mendeklarasikan aksi umum yang dilakukan di intent tersebut. Sistem Android mencocokkan permintaan Anda dengan activity atau komponen lainnya yang bisa menangani aksi permintaan tersebut. Anda akan mengetahui selengkapnya tentang intent implisit dalam bab berikutnya.

#### ▪ **Objek dan bidang Intent**

- Objek Intent adalah instance kelas Intent. Untuk intent eksplisit, bidang kunci suatu intent menyertakan yang berikut ini:
- *Kelas activity* (untuk intent eksplisit). Ini adalah nama kelas activity atau komponen lainnya yang akan menerima intent, misalnya,

`com.example.SampleActivity.class`. Gunakan konstruktor intent atau metode `setComponent()`, `setComponentName()` atau `setClassName()` intent untuk menetapkan kelas.

- *Data* intent. Bidang data intent berisi referensi ke data yang Anda inginkan untuk mengoperasikan activity penerima, sebagai objek Uri.
- *Ekstra* intent. Ini adalah pasangan nilai-kunci yang membawa informasi yang diperlukan activity penerima untuk melakukan aksi yang diminta.
- *Flag* intent. Ini adalah bit metadata tambahan, yang didefinisikan oleh kelas Intent. Flag dapat menginstruksikan sistem Android tentang cara meluncurkan activity atau cara memperlakukan activity setelah diluncurkan.

Untuk intent implisit, Anda juga mungkin perlu mendefinisikan kategori dan aksi intent.

### ▪ Memulai activity dengan intent eksplisit

Untuk memulai activity tertentu dari activity lain, gunakan intent eksplisit dan metode `startActivity()`. Intent eksplisit menyertakan nama kelas yang benar-benar memenuhi syarat untuk activity atau komponen lainnya di objek Intent. Semua bidang intent lainnya bersifat opsional, dan nol secara default.

Misalnya, jika Anda ingin memulai `ShowMessageActivity` untuk menampilkan pesan tertentu di aplikasi email, gunakan kode seperti ini.

```
Intent messageIntent = new Intent(this,
    ShowMessageActivity.class);
startActivity(messageIntent);
```

Konstruktor Intent menggunakan dua argumen untuk intent eksplisit.

- Konteks aplikasi. Dalam contoh ini, kelas activity menyediakan materi (di sini, `this`).
- Komponen tertentu untuk dimulai (`ShowMessageActivity.class`).

Gunakan metode `startActivity()` bersama objek intent baru sebagai satu-satunya argumen. Metode `startActivity()` mengirim intent ke sistem Android, yang meluncurkan kelas `ShowMessageActivity` atas nama aplikasi Anda. Activity baru muncul pada layar dan activity pembuatnya dihentikan sementara.

Activity yang dimulai tetap pada layar hingga pengguna mengetuk tombol kembali pada perangkat, pada saat itu activity ditutup dan diklaim kembali oleh sistem, dan activity yang menghasilkannya akan dilanjutkan. Anda juga bisa

menutup activity yang dimulai secara manual sebagai respons terhadap aksi pengguna (seperti klik tombol) dengan metode finish():

```
public void closeActivity (View view) {  
    finish();  
}
```

### ▪ Meneruskan data di antara activity dengan intent

Selain untuk memulai satu activity dari activity lain, Anda juga menggunakan intent untuk meneruskan informasi di antara activity. Objek intent yang Anda gunakan untuk memulai activity bisa menyertakan *data* intent (URI objek untuk bertindak), atau *ekstra* intent, yang merupakan bit data tambahan yang mungkin diperlukan activity.

Di activity (pengirim) pertama:

1. Buat objek Intent.
2. Masukkan data atau ekstra ke dalam intent itu.
3. Mulailah activity baru dengan startActivity().

Di activity (penerima) kedua:

1. Dapatkan objek intent yang digunakan memulai activity.
2. Ambil data atau ekstra dari objek Intent.

### ▪ Waktu untuk menggunakan data intent atau ekstra intent

Anda bisa menggunakan data intent dan ekstra intent untuk meneruskan data di antara activity. Ada sejumlah perbedaan utama antara data dan ekstra yang menentukan mana yang harus Anda gunakan.

*Data* intent hanya bisa menyimpan satu bagian informasi. URI yang menyatakan lokasi data yang ingin Anda gunakan untuk mengoperasikan. URI tersebut bisa berupa URL laman web (http://), nomor telepon (tel://), lokasi geografis (geo://), atau URI khusus lainnya yang Anda definisikan.

Gunakan bidang intent data:

- ⇒ Bila Anda hanya memiliki satu bagian informasi yang perlu dikirim ke activity yang telah dimulai.
- ⇒ Bila informasi itu adalah lokasi data yang bisa dinyatakan dengan URI.

*Ekstra* intent adalah untuk data arbitrer lainnya yang ingin Anda teruskan ke activity yang telah dimulai. Ekstra intent disimpan di objek Bundle sebagai pasangan kunci dan nilai. Bundle adalah peta, yang dioptimalkan untuk Android, dengan tombol berupa string, dan nilai-nilainya bisa berupa tipe objek atau primitif apa pun (objek harus mengimplementasikan antarmuka

Parcelable). Untuk memasukkan data ke dalam ekstra intent, Anda bisa menggunakan salah satu metode `putExtra()` kelas Intent, atau membuat bundel sendiri dan memasukkannya ke dalam intent dengan `putExtras()`.

Gunakan ekstra intent:

- Jika Anda ingin meneruskan lebih dari satu bagian informasi ke activity yang telah dimulai.
- Jika informasi yang ingin Anda teruskan tidak bisa dinyatakan melalui URI.

Data dan ekstra intent tidak eksklusif; Anda bisa menggunakan data URI dan ekstra untuk informasi tambahan yang diperlukan activity yang dimulai untuk memproses data dalam URI itu.

### Tambahkan data ke intent

Untuk menambahkan data ke intent eksplisit dari activity pembuatnya, buat objek intent seperti yang Anda lakukan sebelumnya:

```
Intent messageIntent = new Intent(this, ShowMessageActivity.class);
```

Gunakan metode `setData()` bersama objek Uri untuk menambahkan URI itu ke intent. Beberapa contoh penggunaan `setData()` bersama URI:

```
// A web page URL
messageIntent.setData(Uri.parse("http://www.google.com"));
// a Sample file URI
messageIntent.setData(Uri.fromFile(new
File("/sdcard/sample.jpg")));
// A sample content: URI for your app's data model
messageIntent.setData(Uri.parse("content://mysample.provider/d
ata"));
// Custom URI
messageIntent.setData(Uri.parse("custom:" + dataID +
buttonId));
```

Perlu diingat bahwa bidang data hanya bisa berisi URI tunggal; jika Anda memanggil `setData()` beberapa kali, hanya nilai terakhir yang akan digunakan. Gunakan ekstra intent untuk menyertakan informasi tambahan (termasuk URI.)

Setelah menambahkan data, Anda bisa memulai activity dengan intent seperti biasanya.

```
startActivity(messageIntent)
```

## Tambahkan ekstra ke intent

Untuk menambahkan ekstra intent ke intent eksplisit dari activity pembuatnya:

1. Tentukan kunci yang akan digunakan untuk informasi yang ingin dimasukkan ke dalam ekstra, atau definisikan sendiri. Setiap bagian informasi memerlukan kunci unik.
2. Gunakan metode `putExtra()` untuk menambahkan pasangan kunci/nilai ke ekstra intent. Secara opsional Anda bisa membuat objek `Bundle`, menambahkan data ke bundel, kemudian menambahkan bundel ke intent.

Kelas `Intent` menyertakan sejumlah kunci ekstra intent yang bisa digunakan, didefinisikan sebagai konstanta yang dimulai dengan kata `EXTRA_`. Misalnya, Anda bisa menggunakan `Intent.EXTRA_EMAIL` untuk menunjukkan larik alamat email (sebagai string), atau `Intent.EXTRA_REFERRER` untuk menetapkan informasi tentang activity pembuat yang mengirim intent tersebut.

Anda juga bisa mendefinisikan kunci ekstra intent milik sendiri. Secara konvensional Anda mendefinisikan kunci ekstra intent sebagai variabel-variabel statis dengan nama yang dimulai kata `EXTRA_`. Untuk menjamin kunci tersebut unik, nilai string kunci itu sendiri harus diawali dengan nama kelas yang benar-benar memenuhi syarat aplikasi. Misalnya:

```
public final static String EXTRA_MESSAGE =
    "com.example.mysampleapp.MESSAGE";
public final static String EXTRA_POSITION_X =
    "com.example.mysampleapp.X";
public final static String EXTRA_POSITION_Y =
    "com.example.mysampleapp.Y";
```

**Buat objek intent (jika belum ada):**

```
Intent messageIntent = new Intent(this,
    ShowMessageActivity.class);
```

Gunakan metode `putExtra()` bersama kunci untuk memasukkan data ke dalam ekstra intent. Kelas `Intent` mendefinisikan banyak metode `putExtra()` untuk jenis data yang berbeda:

```
messageIntent.putExtra(EXTRA_MESSAGE, "this is my message");
messageIntent.putExtra(EXTRA_POSITION_X, 100);
messageIntent.putExtra(EXTRA_POSITION_Y, 500);
```



Atau, Anda bisa membuat bundel baru dan mengisinya dengan ekstra intent. Bundel mendefinisikan banyak metode "put" untuk jenis data primitif yang berbeda serta objek yang mengimplementasikan antarmuka Parcelable Android atau Serializable Java.

```
Bundle extras = new Bundle();
extras.putString(EXTRA_MESSAGE, "this is my message");
extras.putInt(EXTRA_POSITION_X, 100);
extras.putInt(EXTRA_POSITION_Y, 500);
```

Setelah Anda mengisi bundel tersebut, tambahkan ke intent dengan metode `putExtras()` (perhatikan "s" di `Extras`):

```
messageIntent.putExtras(extras);
```

Mulai activity dengan intent seperti biasa:

```
startActivity(messageIntent);
```

### Ambil data dari intent di activity yang dimulai

Bila Anda memulai activity bersama intent, activity yang telah dimulai akan memiliki akses ke intent dan data yang dimuatnya. Untuk mengambil intent yang digunakan untuk memulai activity (atau komponen lain), gunakan metode `getIntent()`:

```
Intent intent = getIntent();
```

Gunakan `getData()` untuk mendapatkan URI dari intent itu:

```
Uri locationUri = getData()
```

Untuk mendapatkan ekstra dari intent, Anda perlu mengetahui kunci untuk pasangan kunci/nilai. Anda bisa menggunakan ekstra Intent standar jika telah menggunakannya, atau bisa menggunakan kunci yang didefinisikan di activity pembuatnya (jika didefinisikan sebagai publik.)

Gunakan salah satu metode `getExtra()` untuk mengekstrak data ekstra dari objek intent:

```
String message =
intent.getStringExtra(MainActivity.EXTRA_MESSAGE);
int positionX =
intent.getIntExtra(MainActivity.EXTRA_POSITION_X);
int positionY =
intent.getIntExtra(MainActivity.EXTRA_POSITION_Y);
```

Atau Anda bisa mendapatkan seluruh bundle ekstra dari intent dan mengekstrak nilai dengan beragam metode Bundle:

```
Bundle extras = intent.getExtras();  
String message =  
extras.getString(MainActivity.EXTRA_MESSAGE);
```

### Mendapatkan data kembali dari activity

Bila Anda memulai activity bersama sebuah intent, activity pembuatnya akan dihentikan sementara, dan activity baru tetap di layar hingga pengguna mengklik tombol kembali, atau panggil metode finish() di handler klik atau fungsi lainnya yang mengakhiri keterlibatan pengguna pada activity ini.

Kadang-kadang bila mengirim data ke activity bersama sebuah intent, Anda juga ingin mendapatkan kembali data dari intent itu. Misalnya, Anda mungkin memulai activity galeri foto yang memungkinkan pengguna memilih foto. Dalam hal ini activity asli Anda perlu menerima informasi tentang foto yang dipilih pengguna dari activity yang diluncurkan.

Untuk meluncurkan activity baru dan mendapatkan kembali hasilnya, lakukan langkah-langkah berikut di activity pembuatnya:

1. Sebagai ganti meluncurkan activity bersama startActivity(), panggil startActivityForResult() bersama intent dan kode permintaan.
2. Buat intent baru di activity yang diluncurkan dan tambahkan data yang dikembalikan ke intent itu.
3. Implementasikan onActivityResult() di activity pembuatnya untuk memproses data yang dikembalikan.

Anda akan mempelajari tentang setiap langkah di bagian berikut ini.

### Gunakan startActivityForResult() untuk meluncurkan activity

Untuk mendapatkan kembali data dari activity yang diluncurkan, mulailah activity itu bersama metode startActivityForResult() sebagai ganti startActivity().

```
startActivityForResult(messageIntent, TEXT_REQUEST);
```

Metode startActivityForResult(), seperti startActivity(), mengambil argumen intent yang berisi informasi tentang activity yang diluncurkan dan data yang dikirim ke activity itu. Akan tetapi, metode startActivityForResult() juga memerlukan kode permintaan.

Kode permintaan adalah integer yang mengidentifikasi permintaan dan bisa digunakan untuk membedakan hasil bila Anda memproses data yang dikembalikan. Misalnya, jika meluncurkan satu activity untuk mengambil foto

dan activity lain untuk memilih foto dari galeri, Anda akan memerlukan kode permintaan yang berbeda untuk mengidentifikasi permintaan yang menjadi pemilik data yang dikembalikan.

Secara konvensional, Anda mendefinisikan kode permintaan sebagai variabel-variabel integer statis dengan nama yang menyertakan REQUEST. Gunakan integer yang berbeda untuk setiap kode. Misalnya:

```
public static final int PHOTO_REQUEST = 1;
public static final int PHOTO_PICK_REQUEST = 2;
public static final int TEXT_REQUEST = 3;
```

### Kembalikan respons dari activity yang diluncurkan

Data respons dari activity yang diluncurkan kembali ke activity pembuatnya akan dikirim dalam intent, baik dalam data maupun ekstra. Anda membentuk intent yang dikembalikan ini dan memasukkan data ke dalamnya menggunakan cara yang sangat mirip dengan yang Anda lakukan untuk intent yang mengirimnya. Biasanya activity yang diluncurkan akan memiliki metode onClick atau metode callback masukan pengguna lain yang Anda gunakan untuk memproses aksi pengguna dan menutup activity. Di sini juga Anda membentuk respons. Untuk mengembalikan data dari activity yang diluncurkan, buat objek intent kosong yang baru.

```
Intent returnIntent = new Intent();
```



**Catatan:** Untuk menghindari kebingungan atas data yang dikirim dengan data yang dikembalikan, gunakan objek intent baru, bukan menggunakan kembali objek intent pengirim asal.

Intent hasil yang dikembalikan tidak memerlukan kelas atau nama komponen berakhir di tempat yang tepat. Sistem Android akan mengarahkan respons kembali ke activity pembuatnya untuk Anda.

Tambahkan data atau ekstra ke intent dengan cara sama seperti yang Anda lakukan pada intent asal. Anda mungkin perlu mendefinisikan kunci untuk ekstra intent yang dikembalikan pada awal kelas.

```
public final static String EXTRA_RETURN_MESSAGE =
    "com.example.mysampleapp.RETURN_MESSAGE";
```

Selanjutnya masukkan data yang dikembalikan ke intent seperti biasa. Di sini, pesan yang dikembalikan adalah ekstra intent dengan kunci EXTRA\_RETURN\_MESSAGE.

```
messageIntent.putExtra(EXTRA_RETURN_MESSAGE,  
mMessage);
```

Gunakan metode setResult() bersama kode respons dan intent dengan data respons: setResult(RESULT\_OK,replyIntent);

Kode respons didefinisikan oleh kelas Activity, dan bisa berupa

- RESULT\_OK: permintaan berhasil.
- RESULT\_CANCELED: pengguna membatalkan operasi.
- RESULT\_FIRST\_USER: untuk mendefinisikan kode hasil milik Anda.

Anda akan menggunakan kode hasil di activity pembuatnya.

Terakhir, panggil finish() untuk menutup activity dan melanjutkan activity pembuatnya:

```
finish();
```

#### Baca data respons di onActivityResult()

Karena sekarang activity yang diluncurkan telah mengirimkan data kembali ke activity pembuatnya bersama intent, activity pertama itu harus menangani data tersebut. Untuk menangani data yang dikembalikan di activity pembuatnya, implementasikan metode callback onActivityResult(). Inilah sebuah contoh sederhana.

```
public void onActivityResult(int requestCode, int  
resultCode, Intent data) {  
    super.onActivityResult(requestCode, resultCode,  
data);  
  
    if (requestCode == TEXT_REQUEST) {  
        if (resultCode == RESULT_OK) {  
            String reply =  
  
data.getStringExtra(SecondActivity.EXTRA_RETURN_MESSAGE)  
;  
                // process data  
            }  
        }  
    }  
}
```

Tiga argumen untuk `onActivityResult()` berisi semua informasi yang Anda perlukan untuk menangani data yang dikembalikan.

- **Kode permintaan.** Kode permintaan yang Anda setel saat meluncurkan activity bersama `startActivityForResult()`. Jika Anda meluncurkan activity yang berbeda untuk melakukan operasi yang berbeda, gunakan kode ini untuk mengidentifikasi data tertentu yang Anda dapatkan kembali.
- **Kode hasil:** kode hasil yang disetel di activity yang diluncurkan, biasanya salah satu dari `RESULT_OK` atau `RESULT_CANCELED`.
- **Data intent.** intent berisi data yang dikembalikan dari activity peluncur.

Metode contoh yang ditampilkan di atas menampilkan logika tipikal untuk menangani kode permintaan dan kode respons. Pengujian pertama adalah untuk permintaan `TEXT_REQUEST`, dan bahwa hasilnya adalah berhasil. Di dalam isi pengujian itu, ekstrak informasi yang dikembalikan dari intent. Gunakan `getData()` untuk mendapatkan data intent, atau `getStringExtra()` untuk mengambil nilai dari ekstra intent bersama kunci tertentu.


#### a. Navigasi Activity

Setiap aplikasi dengan kompleksitas apa pun yang Anda bangun akan menyertakan beberapa activity, baik yang didesain dan diimplementasikan oleh Anda, maupun yang berpotensi di aplikasi lain. Karena pengguna Anda berpindah-pindah dalam aplikasi dan di antara activity, navigasi yang konsisten menjadi semakin penting untuk pengalaman pengguna aplikasi. Beberapa hal lebih mengganggu pengguna daripada navigasi dasar yang berperilaku tidak konsisten dan tak terduga. Mendesain navigasi aplikasi Anda dengan bijak akan membuat penggunaan aplikasi tersebut bisa diprediksi dan bisa diandalkan pengguna.

Sistem Android mendukung dua bentuk strategi navigasi yang berbeda untuk aplikasi Anda.

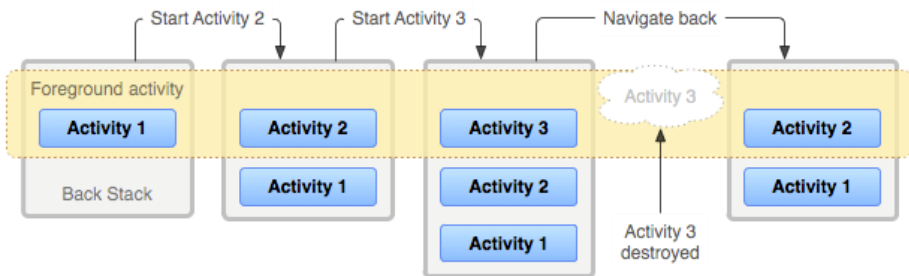
- Navigasi Sementara atau Kembali, yang disediakan melalui tombol kembali di perangkat, dan back-stack.
- Navigasi leluhur, atau Up, yang disediakan oleh Anda sebagai opsi di bilah aksi aplikasi.

- **Navigasi kembali, tugas, dan back-stack**

Navigasi Kembali memungkinkan pengguna Anda untuk kembali ke activity sebelumnya dengan menyetuk tombol kembali di perangkat . Navigasi kembali juga disebut navigasi *sementara* karena tombol kembali menyusuri riwayat layar yang baru saja ditampilkan, dalam urutan kronologi terbalik.


*Back-stack* adalah serangkaian activity yang telah dikunjungi pengguna dan bisa dikunjungi kembali oleh pengguna dengan tombol kembali. Setiap kali activity baru dimulai, activity akan didorong ke back-stack dan mengambil fokus pengguna. Activity sebelumnya akan dihentikan, namun tetap tersedia di back-stack. Back-stack beroperasi berdasarkan mekanisme "masuk terakhir, keluar pertama", jadi bila pengguna selesai dengan activity saat ini dan menekan tombol Kembali, activity tersebut akan dimunculkan dari tumpukan (serta dimusnahkan) dan activity sebelumnya dilanjutkan.

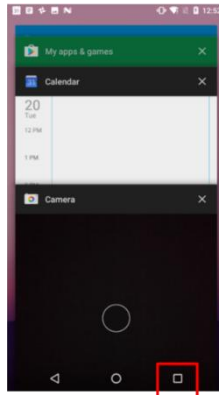
Karena aplikasi bisa memulai activity baik di dalam maupun di luar satu aplikasi, back-stack berisi semua activity yang telah diluncurkan oleh pengguna dalam urutan terbalik. Setiap kali pengguna menekan tombol Kembali, setiap activity di tumpukan akan dimunculkan untuk membuka activity sebelumnya, hingga pengguna kembali ke layar Utama.



Gambar 96. Siklus Activity

Android menyediakan back-stack untuk setiap *tugas*. Tugas adalah konsep penyusunan semua activity yang berinteraksi dengan pengguna saat melakukan operasi, baik di dalam aplikasi maupun di beberapa aplikasi. Sebagian besar tugas dimulai dari layar utama Android, dan menyetuk ikon aplikasi akan memulai tugas (serta back-stack baru) untuk aplikasi itu. Jika pengguna menggunakan aplikasi sebentar, menyetuk beranda, dan memulai aplikasi, aplikasi baru itu akan diluncurkan di tugasnya sendiri dan memiliki back-stack

sendiri. Jika pengguna kembali ke aplikasi pertama, back-stack tugas pertama akan kembali. Menyusuri dengan tombol kembali hanya akan mengembalikan ke activity di tugas saat ini, bukan untuk semua tugas yang berjalan pada perangkat. Android memungkinkan pengguna menyusuri berbagai tugas bersama ringkasan atau layar tugas saat ini, yang bisa diakses dengan tombol segi empat di sudut kanan bawah perangkat 




Gambar 97. Tombol Kembali (BackStack)

Dalam sebagian besar kasus, Anda tidak perlu khawatir tentang pengelolaan tugas atau back-stack untuk aplikasi—sistem terus melacak hal ini untuk Anda, dan tombol kembali selalu tersedia pada perangkat.

Akan tetapi, mungkin ada saatnya Anda ingin mengganti perilaku default untuk tugas atau untuk back-stack. Misalnya, jika layar Anda berisi browser web yang disematkan yang memungkinkan pengguna menyusuri laman web, Anda mungkin ingin menggunakan perilaku kembali default di browser bila pengguna menekan tombol *Kembali* di perangkat, daripada mengembalikan ke activity sebelumnya. Anda mungkin juga perlu mengubah perilaku default aplikasi dalam kasus khusus lainnya seperti pada notifikasi atau widget, sehingga activity yang berada jauh di dalam aplikasi Anda mungkin diluncurkan sebagai tugasnya sendiri, tanpa back-stack sama sekali. Anda akan mengetahui selengkapnya tentang pengelolaan tugas dan back-stack di bagian berikutnya.

### ▪ Up Navigasi

Navigasi Up, kadang-kadang disebut sebagai navigasi leluhur atau logis, digunakan untuk navigasi dalam aplikasi berdasarkan hubungan hierarki eksplisit di antara layar. Dengan navigasi Up, activity Anda disusun dalam suatu hierarki, dan activity "anak" menampilkan panah hadap-kiri di bilah aksi 

yang mengembalikan pengguna ke activity "parent". Activity teratas di hierarki biasanya adalah activity utama Anda, dan pengguna tidak bisa Up lagi dari sana.



Gambar 98. Up Navigation

Misalnya, jika activity utama di aplikasi email adalah daftar semua pesan, memilih sebuah pesan akan meluncurkan activity kedua untuk menampilkan satu email itu. Dalam hal ini, activity pesan akan menyediakan tombol Up yang mengembalikan ke daftar pesan.

Perilaku tombol up didefinisikan oleh Anda di setiap activity berdasarkan cara mendesain navigasi aplikasi. Dalam banyak kasus, navigasi Up dan Kembali mungkin menyediakan perilaku yang sama: cuma mengembalikan ke activity sebelumnya. Misalnya, activity Setelan mungkin tersedia dari activity apa pun di aplikasi Anda, sehingga "Up" sama dengan kembali -- cuma mengembalikan pengguna ke tempat sebelumnya di hierarki. Menyediakan perilaku Up untuk aplikasi Anda adalah hal yang opsional, namun praktik desain yang baik, adalah menyediakan navigasi yang konsisten untuk berbagai activity di aplikasi Anda.

- **Implementasikan navigasi up dengan parent activity**

Dengan proyek template standar di Android Studio, mudah untuk mengimplementasikan navigasi Up. Jika satu activity adalah anak activity lain di hierarki activity aplikasi, tetapkan parent activity itu di Manifest Android.

Mulai di Android 4.1 (API level 16), deklarasikan parent logis setiap activity dengan menetapkan atribut `android:parentActivityName` di elemen `<activity>`. Untuk mendukung Android versi lama, sertakan informasi



<meta-data> untuk mendefinisikan activity parent secara eksplisit. Gunakan kedua metode agar kompatibel mundur dengan semua versi Android. Inilah definisi kerangka untuk kedua activity utama (parent) dan activity kedua:

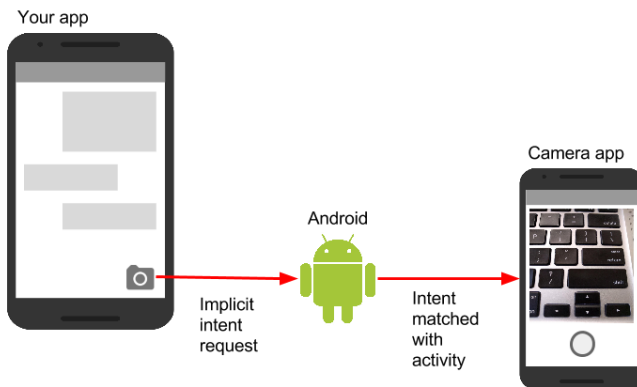
```
<application ... >
  <!-- The main/home activity (it has no parent activity) -->
  <activity
    android:name=".MainActivity" ...>
    <intent-filter>
      <action android:name="android.intent.action.MAIN"
    />
      <category
        android:name="android.intent.category.LAUNCHER" />
    </intent-filter>

    </activity>
    <!-- A child of the main activity -->
    <activity android:name=".SecondActivity"
      android:label="@string/activity2_name"
      android:parentActivityName=".MainActivity">
      <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value="com.example.android.twoactivities.MainActivity"
      />
    </activity>
  </application>
```

## b. Tentang Intent Implisit

Di bab awal, Anda telah mempelajari tentang intent eksplisit, tempat Anda bisa memulai satu activity dari activity lain dengan menetapkan nama kelas activity itu. Inilah cara paling dasar untuk menggunakan intent, untuk memulai activity atau komponen aplikasi lainnya dan meneruskan data ke intent (dan kadang-kadang meneruskan data kembali).

Penggunaan intent yang lebih fleksibel adalah *intent implisit*. Dengan intent implisit, Anda tidak menetapkan activity yang tepat (atau komponen lainnya) untuk dijalankan—sebagai gantinya, Anda menyertakan informasi secukupnya dalam intent tentang tugas yang ingin dilakukan. Sistem Android akan mencocokkan informasi dalam intent permintaan dengan activity yang tersedia di perangkat yang bisa melakukan tugas itu. Jika hanya ada satu activity yang cocok, activity itu akan diluncurkan. Jika ada beberapa activity yang cocok, pengguna disediakan pemilih aplikasi yang memungkinkan mereka memilih aplikasi yang disukai untuk melakukan tugas.



Gambar 99. Intent Implicit

Misalnya, Anda memiliki aplikasi yang mencantumkan cuplikan video yang tersedia. Jika pengguna menyentuh item dalam daftar, Anda akan memutar cuplikan video itu. Daripada mengimplementasikan keseluruhan pemutar video di aplikasi, Anda bisa meluncurkan intent yang menetapkan tugas seperti "putar objek video tipe." Sistem Android selanjutnya akan mencocokkan permintaan dengan activity yang telah mendaftarkan dirinya sendiri untuk memutar objek video tipe.

Activity mendaftarkan dirinya sendiri secara otomatis pada sistem agar dapat menangani intent implisit dengan *filter* intent, yang dideklarasikan dalam manifes Android. Misalnya, activity utama (dan satu-satunya activity utama)

untuk aplikasi Anda memiliki filter intent yang mendeklarasikan activity utama untuk kategori peluncur. Filter intent ini adalah cara sistem Android mengetahui cara memulai activity tertentu di aplikasi Anda bila pengguna mengetuk ikon untuk aplikasi di layar utama perangkat.

- **Aksi intent, kategori, dan data**

Intent implisit, seperti intent eksplisit, merupakan instance kelas Intent. Selain bagian intent yang telah Anda pelajari di bab awal (seperti data intent dan ekstra intent), bidang-bidang ini digunakan oleh intent implisit:

- 1) *Aksi intent*, merupakan aksi umum yang harus dilakukan activity penerima. Aksi intent yang tersedia didefinisikan sebagai konstanta dalam kelas Intent dan dimulai dengan kata ACTION\_. Aksi intent umum adalah ACTION\_VIEW, yang Anda gunakan bila memiliki beberapa informasi yang bisa ditampilkan activity kepada pengguna, seperti foto untuk ditampilkan di aplikasi galeri, atau alamat untuk ditampilkan di aplikasi peta. Anda bisa menetapkan aksi untuk sebuah intent di konstruktor intent, atau dengan metode setAction().
- 2) *Kategori intent*, menyediakan informasi tambahan tentang kategori komponen yang harus menangani intent. Kategori intent bersifat opsional, dan Anda bisa menambahkan lebih dari satu kategori ke intent. Kategori intent juga didefinisikan sebagai konstanta di kelas Intent dan dimulai dengan kata CATEGORY\_. Anda bisa menambahkan kategori ke intent dengan metode addCategory().
- 3) *Tipe data*, yang menunjukkan MIME tipe data yang digunakan untuk mengoperasikan activity. Biasanya, ini diambil dari URI dalam bidang data intent, namun Anda juga bisa secara eksplisit mendefinisikan tipe data dengan metode setType().

Tindakan, kategori, dan tipe data intent keduanya digunakan oleh objek Intent yang Anda buat di activity pengiriman, serta di filter intent yang didefinisikan di manifes Android untuk activity penerimaan. Sistem Android menggunakan informasi ini untuk mencocokkan permintaan intent implisit dengan activity atau komponen lain yang bisa menangani intent itu.

### Mengirim intent implisit

Memulai activity bersama intent implisit, dan meneruskan data di antara activity itu, cara kerjanya sangat mirip dengan yang dilakukan untuk intent eksplisit:

1. Di activity pengirim, buat objek Intent baru.

2. Tambahkan informasi tentang permintaan ke objek Intent, seperti data atau ekstra.
3. Kirim intent bersama `startActivity()` (untuk memulai activity saja) atau `startActivityForResult()` (untuk memulai activity dan mengharapkan hasil kembali).

Bila membuat objek Intent implisit, Anda:

- Jangan menetapkan activity khusus atau komponen lain untuk diluncurkan.
- Menambahkan aksi intent atau kategori intent (atau keduanya).
- Cocokkan intent dengan sistem sebelum memanggil `startActivity()` atau `startActivityForResult()`.
- Menampilkan pemilih aplikasi untuk permintaan (opsional).

### Buat objek Intent implisit

Untuk menggunakan intent implisit, buat objek Intent seperti yang dilakukan untuk intent eksplisit, hanya tanpa nama komponen spesifik.

```
Intent sendIntent = new Intent()
```

Anda juga bisa membuat objek Intent dengan aksi tertentu:

```
Intent sendIntent = new Intent(Intent.ACTION_VIEW)
```

Setelah memiliki objek Intent, Anda bisa menambahkan informasi lain (kategori, data, ekstra) dengan beragam metode Intent. Misalnya, kode ini membuat objek Intent implisit, menyetel aksi intent ke `ACTION_SEND`, mendefinisikan ekstra intent untuk menampung teks, dan menyetel tipe data ke tipe MIME `"text/plain"`.

```
Intent sendIntent = new Intent();
sendIntent.setAction(Intent.ACTION_SEND);
sendIntent.putExtra(Intent.EXTRA_TEXT,
    textMessage);
sendIntent.setType("text/plain");
```

### Cocokkan activity sebelumnya memulainya

Bila Anda mendefinisikan intent implisit dengan aksi dan/atau kategori tertentu, ada kemungkinan bahwa tidak akan ada activity *apa pun* di perangkat yang bisa menangani permintaan Anda. Jika Anda hanya mengirim intent dan tidak ada kecocokan, aplikasi akan mogok.

Untuk memverifikasi apakah activity atau komponen lain tersedia untuk menerima intent Anda, gunakan metode `resolveActivity()` bersama pengelola paket sistem seperti ini:

```
if (sendIntent.resolveActivity(getPackageManager())
    != null) {
    startActivity(chooser);
}
```

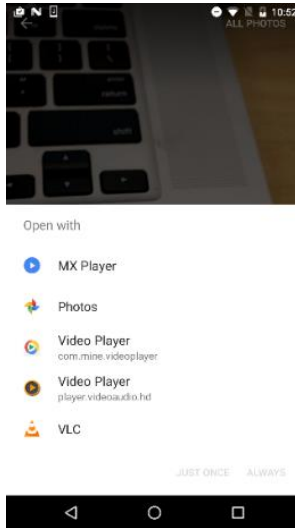
Jika hasil `resolveActivity()` bukan nol, maka setidaknya ada satu aplikasi yang tersedia yang bisa menangani intent, dan aman untuk menggunakan `startActivity()`. Jangan mengirim intent jika hasilnya nol.

Jika Anda memiliki fitur yang bergantung pada activity eksternal yang mungkin atau mungkin tidak tersedia di perangkat, praktik terbaik adalah menguji ketersediaan activity eksternal sebelum pengguna mencoba menggunakannya. Jika tidak ada activity yang bisa menangani permintaan Anda (yakni, `resolveActivity()` mengembalikan nol), nonaktifkan fitur atau sediakan pesan kesalahan untuk fitur tersebut kepada pengguna.

### Tampilkan pemilih aplikasi

Untuk menemukan activity atau komponen lain yang bisa menangani permintaan intent Anda, sistem Android akan mencocokkan intent implisit dengan activity yang filter intentnya menunjukkannya bisa melakukan aksi tersebut. Jika beberapa aplikasi yang telah dipasang memang cocok, pengguna akan disajikan pemilih aplikasi yang memungkinkan mereka memilih aplikasi yang ingin digunakan untuk menangani intent tersebut.

Dalam banyak kasus, pengguna memiliki aplikasi pilihan untuk tugas yang diberikan, dan mereka akan memilih opsi untuk selalu menggunakan aplikasi itu bagi tugas tersebut. Akan tetapi, jika beberapa aplikasi bisa merespons intent dan pengguna setiap kali mungkin ingin menggunakan aplikasi yang berbeda, Anda bisa memilih untuk secara eksplisit menampilkan dialog pemilih kapan saja. Misalnya, bila aplikasi melakukan aksi "bagikan ini" bersama aksi `ACTION_SEND`, pengguna mungkin ingin berbagi menggunakan aplikasi yang berbeda, bergantung pada situasi saat ini.



Gambar 100. Tampilkan Pemilihan Aplikasi

Untuk menampilkan pemilih, buat intent wrapper untuk intent implisit dengan metode `createChooser()`, kemudian cocokkan dan gunakan `startActivity()` dengan intent wrapper. Metode `createChooser()` juga memerlukan argumen string untuk judul yang muncul pada pemilih. Anda bisa menetapkan judul bersama resource string seperti yang dilakukan pada string lain. Misalnya:

```
// The implicit Intent object
Intent sendIntent = new Intent(Intent.ACTION_SEND);
// Always use string resources for UI text.
String title =
getResources().getString(R.string.chooser_title);
// Create the wrapper intent to show the chooser
dialog.
Intent chooser = Intent.createChooser(sendIntent,
title);
// Resolve the intent before starting the activity
if (sendIntent.resolveActivity(getPackageManager()) !=
null) {
startActivity(chooser);
}
```

### Menerima intent implisit

Jika ingin activity di aplikasi merespons intent implisit (dari aplikasi milik Anda atau aplikasi lain), deklarasikan satu atau beberapa filter intent di manifest Android. Setiap filter intent menetapkan tipe intent yang diterimanya berdasarkan aksi, data, dan kategori intent. Sistem akan mengirimkan intent implisit ke komponen aplikasi Anda hanya jika intent itu bisa melewati salah satu filter intent.



**Catatan:** Intent eksplisit selalu dikirimkan ke targetnya, terlepas dari filter intent yang dideklarasikan komponen. Sebaliknya, jika activity tidak mendeklarasikan filter intent apa pun, activity hanya bisa diluncurkan dengan intent eksplisit.

Setelah activity berhasil diluncurkan bersama intent implisit, Anda bisa menangani intent tersebut dan datanya dengan cara sama yang dilakukan pada intent eksplisit:

1. Mendapatkan objek Intent dengan `getIntent()`.
2. Mendapatkan data intent atau ekstra dari intent itu.
3. Melakukan tugas yang diminta intent.
4. Mengembalikan data ke activity pemanggil bersama intent lain, jika diperlukan.

#### ▪ **Filter intent**

Definisikan filter intent dengan satu atau beberapa elemen `<intent-filter>` di file manifest aplikasi, yang disarangkan di elemen `<activity>` yang sesuai. Di dalam `<intent-filter>`, tetapkan tipe intent yang bisa ditangani activity Anda. Sistem Android mencocokkan intent implisit dengan activity atau komponen aplikasi lain hanya jika bidang-bidang dalam objek Intent cocok dengan filter intent untuk komponen itu.

Filter intent dapat berisi elemen ini, sesuai dengan bidang di objek Intent yang dijelaskan di atas:

- `<action>`: Aksi intent.
- `<data>`: Tipe data yang diterima, termasuk tipe MIME atau atribut URI data atribut lainnya (seperti skema, host, porta, jalur, dan sebagainya).
- `<category>`: Kategori intent.

Misalnya, activity utama untuk aplikasi Anda menyertakan elemen `<intent-filter>` ini, yang telah Anda lihat di bab sebelumnya:

```

<intent-filter>
  <action android:name="android.intent.action.MAIN" />
  <category
android:name="android.intent.category.LAUNCHER" />
</intent-filter>

```

Filter intent ini memiliki aksi MAIN dan kategori LAUNCHER. Elemen `<action>` menetapkan bahwa ini adalah titik masuk "utama" ke aplikasi. Elemen `<category>` menetapkan bahwa activity ini harus dicantumkan dalam peluncur aplikasi sistem (untuk memungkinkan pengguna meluncurkan activity ini). Hanya activity utama untuk aplikasi Anda yang harus memiliki filter intent ini.

Inilah contoh lain intent implisit untuk berbagi sedikit teks. Filter intent ini mencocokkan contoh intent implisit dari bagian sebelumnya:

```

<activity android:name="ShareActivity">
  <intent-filter>
    <action
android:name="android.intent.action.SEND"/>
    <category
android:name="android.intent.category.DEFAULT"/>
    <data android:mimeType="text/plain"/>
  </intent-filter>
</activity>

```

Anda bisa menetapkan lebih dari satu aksi, data, atau kategori untuk filter intent yang sama, atau memiliki beberapa filter intent per activity untuk menangani jenis intent yang berbeda.

Sistem Android akan menguji intent implisit terhadap filter intent dengan membandingkan bagian intent tersebut dengan ketiga elemen filter intent (aksi, kategori, dan data). Intent harus lulus ketiga pengujian atau sistem Android tidak akan mengirim intent ke komponen. Akan tetapi, karena sebuah komponen mungkin memiliki beberapa filter intent, intent yang tidak lulus salah satu filter komponen mungkin saja lulus di filter lainnya.

- **Action (Tindakan)**

Filter intent bisa mendeklarasikan nol atau beberapa elemen `<action>` untuk aksi intent. Aksi didefinisikan dalam atribut nama, dan berisi string "android.intent.action." ditambah nama aksi intent, tanpa awalan ACTION\_. Jadi, misalnya, intent implisit dengan aksi ACTION\_VIEW cocok dengan filter intent yang aksinya adalah `android.intent.action.VIEW`.



Misalnya, filter intent ini cocok dengan ACTION\_EDIT dan ACTION\_VIEW:

```
<intent-filter>
  <action android:name="android.intent.action.EDIT"
/>
  <action android:name="android.intent.action.VIEW"
/>
  ...
</intent-filter>
```

Untuk melewati filter ini, aksi yang ditetapkan dalam objek Intent yang masuk harus cocok dengan setidaknya salah satu tindakan. Anda harus menyertakan setidaknya satu aksi intent untuk mencocokkan intent implisit yang masuk.

### ▪ **Kategori**

Filter intent bisa mendeklarasikan nol atau beberapa elemen `<category>` untuk kategori intent. Kategori didefinisikan dalam atribut nama, dan berisi string "android.intent.category." ditambah nama kategori intent, tanpa awalan CATEGORY.

Misalnya, filter intent ini cocok dengan CATEGORY\_DEFAULT dan CATEGORY\_BROWSABLE:

```
<intent-filter>
  <category
android:name="android.intent.category.DEFAULT" />
  <category
android:name="android.intent.category.BROWSABLE" />
  ...
</intent-filter>
```

Perhatikan, semua activity yang Anda inginkan untuk menerima intent implisit harus menyertakan filter intent android.intent.category.DEFAULT. Kategori ini diterapkan ke semua objek Intent implisit oleh sistem Android.

### ▪ **Data**

Filter intent bisa mendeklarasikan nol atau beberapa elemen `<data>` untuk URI yang dimuat dalam data intent. Karena data intent terdiri dari URI dan (opsional) tipe MIME, Anda bisa membuat filter intent untuk beragam aspek data itu, termasuk:

- Skema URI

- Host URI
- Jalur URI
- Tipe MIME

Misalnya, filter intent ini mencocokkan intent data dengan skema URI http dan tipe MIME "video/mpeg" atau "audio/mpeg".

```
<intent-filter>
  <data android:mimeType="video/mpeg"
  android:scheme="http" />
  <data android:mimeType="audio/mpeg"
  android:scheme="http" />
  ...
</intent-filter>
```

### Berbagi data dengan `ShareCompat.IntentBuilder`

Tindakan berbagi adalah cara mudah bagi pengguna untuk berbagi item di aplikasi dengan jaringan sosial dan aplikasi lainnya. Meskipun Anda bisa membangun aksi berbagi di aplikasi sendiri menggunakan intent implisit dengan aksi `ACTION_SEND`. Android menyediakan kelas helper `ShareCompat.IntentBuilder` agar mudah mengimplementasikan berbagi di aplikasi.



**Catatan:** Untuk aplikasi yang menargetkan rilis Android setelah API 14, Anda bisa menggunakan kelas `ShareActionProvider` untuk tindakan berbagi sebagai ganti `ShareCompat.IntentBuilder`. Kelas `ShareCompat` adalah bagian dari pustaka dukungan V4, dan memungkinkan Anda untuk menyediakan tindakan berbagi di aplikasi yang kompatibel-mundur. `ShareCompat` menyediakan API tunggal untuk berbagi pada perangkat Android lama dan baru.

Dengan kelas `ShareCompat.IntentBuilder`, Anda tidak perlu membuat atau mengirim intent implisit untuk aksi berbagi. Gunakan metode di `ShareCompat.IntentBuilder` untuk menunjukkan data yang ingin dibagikan serta informasi tambahan. Mulai dengan metode `from()` untuk membuat builder intent baru, tambahkan metode lain untuk menambahkan lebih banyak data, dan akhiri dengan metode `startChooser()` untuk membuat dan mengirim intent. Anda bisa merangkai metode seperti ini:

```

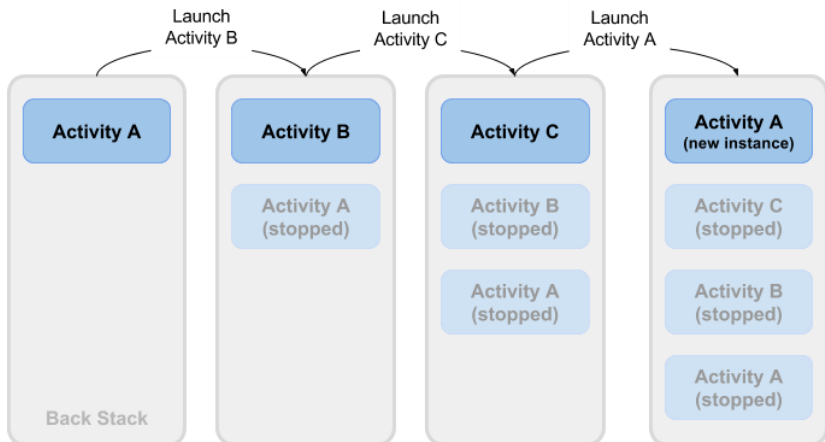
ShareCompat.IntentBuilder
    .from(this)           // information about the calling
activity
    .setType(mimeType)  // mime type for the data
    .setChooserTitle("Share this text with: ") //title
for the app chooser
    .setText(txt)       // intent data
    .startChooser();    // send the intent

```

- **Mengelola tugas dan activity**

Umumnya navigasi pengguna dari activity ke activity dan kembali lagi melalui tumpukan secara langsung. Bergantung pada desain dan navigasi aplikasi Anda, mungkin ada komplikasi, khususnya dengan activity yang dimulai dari aplikasi lain dan tugas lain.

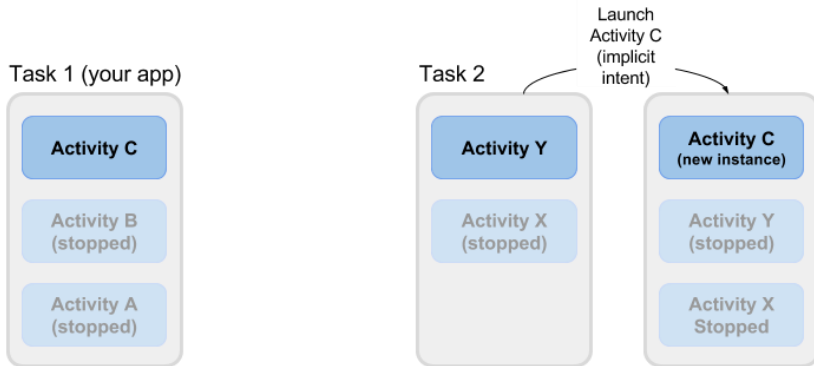
Misalnya, katakanlah Anda memiliki aplikasi dengan tiga activity: A, B, dan C. A meluncurkan B bersama sebuah intent, dan B meluncurkan C. C, sebaliknya mengirimkan intent untuk meluncurkan A. Dalam hal ini, sistem membuat *instance kedua* dari A di bagian atas tumpukan, dibandingkan memberikan instance yang sudah berjalan ke latar depan. Bergantung pada cara mengimplementasikan activity, dua instance dari A bisa menjadi tidak sinkron dan memberikan pengalaman membingungkan pada pengguna saat menyusuri kembali melalui tumpukan.



Gambar 101. Mengelola Task dan Activity

Atau, katakanlah activity C Anda bisa diluncurkan dari aplikasi kedua bersama sebuah intent implisit. Pengguna menjalankan aplikasi kedua, yang

memiliki tugas dan back-stack sendiri. Jika aplikasi itu menggunakan intent implisit untuk meluncurkan activity C Anda, instance C akan baru dibuat dan ditempatkan pada back-stack untuk tugas aplikasi kedua itu. Aplikasi Anda tetap memiliki tugas, back-stack, dan instance C sendiri.



Gambar 102. Launch Activity

Kebanyakan perilaku default Android untuk tugas dan activity berfungsi dengan baik dan Anda tidak perlu khawatir tentang cara activity dikaitkan dengan tugas atau bagaimana activity bisa ada di back-stack. Jika Anda ingin mengubah perilaku normal, Android menyediakan sejumlah cara untuk mengelola tugas dan activity dalam tugas tersebut, termasuk:

- Mode peluncuran activity, untuk menentukan cara meluncurkan aplikasi.
- Afinitas tugas, yang menunjukkan tugas mana yang memiliki activity yang diluncurkan.

### c. Mode Peluncuran Activity

Gunakan mode peluncuran activity untuk menunjukkan cara memperlakukan activity baru bila diluncurkan—yakni jika activity harus ditambahkan ke tugas saat ini, atau diluncurkan ke dalam tugas baru. Definisikan mode peluncuran untuk activity bersama atribut di elemen `<activity>` manifes Android, atau bersama flag yang disetel pada intent yang memulai activity itu.

#### ▪ Atribut activity

Untuk mendefinisikan mode peluncuran activity, tambahkan atribut `android:launchMode` ke elemen `<activity>` di manifes Android. Contoh ini menggunakan mode peluncuran "standar", yang merupakan default.

```
<activity
    android:name=".SecondActivity"
    android:label="@string/activity2_name"
    android:parentActivityName=".MainActivity"
    android:launchMode="standard">
    ...
</activity>
```

Ada empat mode peluncuran yang tersedia sebagai bagian dari elemen `<activity>`:

- 1) "standard" (default): Activity baru diluncurkan dan ditambahkan ke back-stack untuk tugas saat ini. Activity bisa dibuat instance-nya beberapa kali, tugas tunggal bisa memiliki beberapa instance activity yang sama, dan beberapa instance bisa dimiliki oleh tugas yang berbeda.
- 2) "singleTop": Jika instance activity ada di bagian atas back-stack tugas saat ini dan permintaan intent untuk activity tersebut datang, Android akan merutekan intent tersebut ke instance activity yang ada, bukan membuat instance baru. Activity baru tetap dibuat instance-nya jika masih ada activity di back-stack selain di bagian atas.
- 3) "singleTask": Bila activity diluncurkan, sistem akan membuat tugas baru untuk activity itu. Jika tugas lain sudah ada dengan instance activity itu, sistem akan merutekan intent ke activity itu sebagai gantinya.
- 4) "singleInstance": Sama seperti tugas tunggal, hanya saja sistem tidak meluncurkan activity lain ke dalam tugas yang berisi instance activity. Activity akan selalu tunggal dan satu-satunya anggota tugasnya.

Sebagian besar aplikasi hanya akan menggunakan mode peluncuran standar atau `singleTop`. Lihat dokumentasi atribut `launchMode` untuk informasi lebih detail tentang mode peluncuran.

- **Flag intent**

Flag intent merupakan opsi yang menetapkan cara activity (atau komponen aplikasi lainnya) yang menerima intent harus menangani intent tersebut. Flag intent didefinisikan sebagai konstanta di kelas `Intent` dan dimulai dengan kata `FLAG_`. Anda menambahkan flag intent ke objek `Intent` dengan `setFlag()` atau `addFlag()`.

Tiga flag intent spesifik digunakan untuk mengontrol mode peluncuran activity, baik yang berhubungan dengan atribut `launchMode` atau yang menggantikannya. Flag intent selalu diutamakan daripada mode peluncuran jika terjadi konflik.

- 1) `FLAG_ACTIVITY_NEW_TASK`: memulai activity di tugas baru. Ini adalah perilaku yang sama dengan mode peluncuran `singleTask`.
- 2) `FLAG_ACTIVITY_SINGLE_TOP`: jika activity yang diluncurkan berada di bagian atas back-stack, rutekan intent ke instance activity yang ada tersebut. Jika tidak, buat instance activity baru. Ini adalah perilaku yang sama dengan mode peluncuran `singleTop`.
- 3) `FLAG_ACTIVITY_CLEAR_TOP`: Jika instance activity yang akan diluncurkan sudah ada di back-stack, musnahkan activity lain di atasnya dan rutekan intent ke instance yang ada. Bila digunakan bersama-sama dengan `FLAG_ACTIVITY_NEW_TASK`, flag ini berada di instance activity yang ada dalam tugas apa pun dan membawanya ke latar depan.

- **Tangani intent baru**

Bila sistem Android merutekan intent ke instance activity yang ada, sistem akan memanggil metode callback `onNewIntent()` (biasanya persis sebelum metode `onResume()`). Metode `onNewIntent()` menyertakan argumen untuk intent baru yang dirutekan ke activity. Ganti metode `onNewIntent()` dalam kelas Anda untuk menangani informasi dari intent baru tersebut.

Perhatikan, metode `getIntent()`—untuk mendapatkan akses ke intent yang meluncurkan activity—**selalu** mempertahankan intent asal yang meluncurkan instance activity. Panggil `setIntent()` dalam metode `onNewIntent()`:

```

@Override
public void onNewIntent(Intent intent) {
    super.onNewIntent(intent);

    // Use the new intent, not the original one
    setIntent(intent);
}

```

Panggilan ke `getIntent()` setelah ini akan mengembalikan intent baru.

### ▪ Afinitas Tugas

Afinitas tugas menunjukkan tugas yang sebaiknya memiliki activity pilihan bila instance activity itu diluncurkan. Secara default, semua activity dimiliki aplikasi yang meluncurkannya. Activity dari luar aplikasi yang diluncurkan bersama intent implisit dimiliki aplikasi yang mengirim intent implisit.

Untuk mendefinisikan afinitas tugas, tambahkan atribut `android:taskAffinity` ke elemen `<activity>` di manifes Android. Afinitas tugas default adalah nama paket untuk aplikasi (yang dideklarasikan di `. Nama tugas baru harus unik dan berbeda dari nama paket. Contoh ini menggunakan "com.example.android.myapp.newtask" untuk nama afinitas.`

```

<activity
    android:name=".SecondActivity"
    android:label="@string/activity2_name"
    android:parentActivityName=".MainActivity"

    android:taskAffinity="com.example.android.myapp.newtask"
>
    ...
</activity>

```

Afinitas tugas biasanya digunakan bersama-sama dengan mode peluncuran `singleTask` atau flag intent `FLAG_ACTIVITY_NEW_TASK` untuk menempatkan activity baru dalam tugas yang diberi nama sendiri. Jika tugas baru sudah ada, intent akan dirutekan ke tugas dan afinitas itu.

Penggunaan afinitas tugas lainnya adalah pengparentan ulang, yang memungkinkan tugas dipindahkan dari activity yang semula meluncurkannya ke activity yang menjadi afinitasnya. Untuk mengaktifkan pengparentan ulang atas tugas, tambahkan atribut afinitas tugas ke elemen `<activity>` dan setel `android:allowTaskReparenting` ke `true`.



## YANG HARUS SUDAH ANDA KETAHUI

Untuk praktik ini Anda harus bisa:

- 1) Membuat dan menjalankan aplikasi dalam Android Studio.
- 2) Membuat dan mengedit elemen UI dengan Layout Editor Grafis atau langsung di file layout XML.
- 3) Menambahkan fungsionalitas onClick ke tombol.

## YANG AKAN ANDA BUTUHKAN

Untuk materi dan praktik ini, Anda akan membutuhkan:

- 1) Komputer Mac, Windows, atau Linux. Tool Android Studio untuk persyaratan sistem terbaru.
- 2) Akses internet atau cara alternatif untuk memuat Android Studio dan pemasangan Java terbaru ke komputer Anda.
- 3) Smartphone sebagai perangkat untuk pengujian secara nyata aplikasi yang telah dibuat.

## YANG AKAN ANDA PELAJARI

Anda akan belajar:

- 1) Membuat activity baru di Android Studio.
- 2) Mendefinisikan activity induk dan anak untuk navigasi "Up".
- 3) Memulai activity dengan intent eksplisit.
- 4) Meneruskan data di antara activity dengan ekstra intent.

## YANG HARUS ANDA LAKUKAN

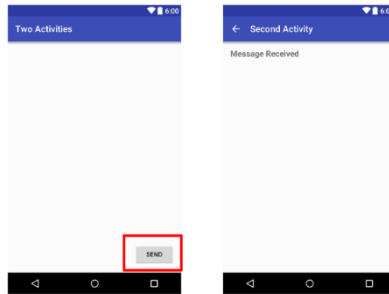
- 1) Membuat aplikasi Android baru dengan dua activity.
- 2) Meneruskan beberapa data (string) dari activity utama ke yang kedua menggunakan intent, dan menampilkan data tersebut dalam activity kedua.
- 3) Mengirimkan bit data kedua yang berbeda kembali ke activity utama, menggunakan intent juga.



## RINGKASAN APLIKASI

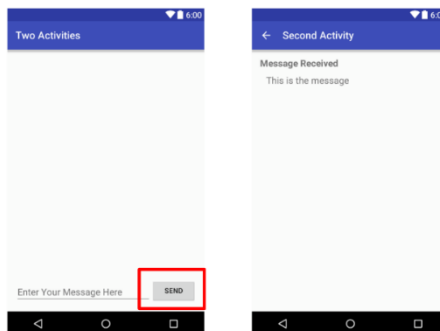
Dalam bab ini Anda akan membuat dan membangun aplikasi bernama TwoActivities yang, ternyata, berisi dua activity. Aplikasi ini akan dibangun dalam tiga tahap.

Pada tahap pertama, buat aplikasi yang activity utamanya hanya berisi satu tombol (Send). Saat pengguna mengklik tombol ini, activity utama menggunakan intent untuk memulai activity kedua.



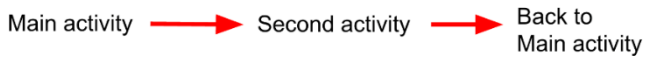
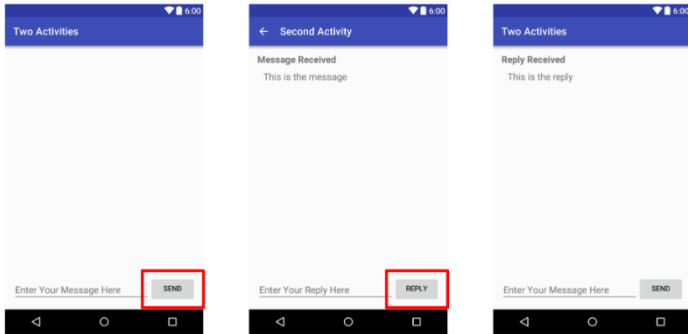
Main activity → Second activity

Dalam tahap kedua, Anda akan menambahkan tampilan EditText ke activity utama. Pengguna memasukkan pesan dan mengklik Send. Activity utama menggunakan intent untuk memulai activity kedua dan untuk mengirimkan pesan pengguna ke activity tersebut. Activity kedua menampilkan pesan yang diterimanya.



Main activity → Second activity

Pada tahap terakhir aplikasi TwoActivities, tambahkan tampilan EditText dan tombol Reply ke activity kedua. Sekarang pengguna bisa mengetik pesan balasan dan mengeklik Reply, dan balasannya ditampilkan di activity utama. Pada titik ini, gunakan intent di sini untuk meneruskan pesan balasan kembali dari activity kedua ke activity utama.



## Latihan 5.1. Membuat project TwoActivities

Dalam tugas ini Anda akan menyiapkan proyek awal dengan activity utama, mendefinisikan layout, dan mendefinisikan metode kerangka untuk peristiwa tombol `onClick`.

### Membuat proyek TwoActivities

1. Mulai Android Studio dan buat proyek Android Studio baru. Panggil aplikasi "TwoActivities" dan ubah domain perusahaan ke "android.example.com." Pilih SDK Minimum yang Anda gunakan di proyek sebelumnya.
2. Pilih **Empty Activity** untuk template proyek. Klik **Next**.
3. Terima nama activity default (MainActivity). Pastikan kotak **Generate Layout file** dicentang. Klik **Finish**.

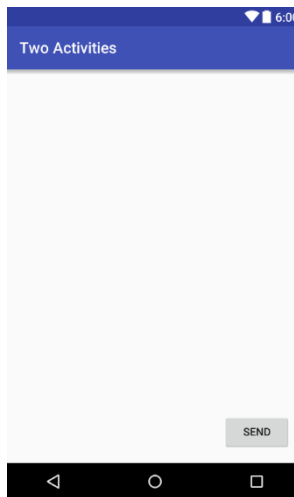
### Definisikan layout untuk activity utama

1. Buka `res/layout/activity_main.xml`. Pada Layout Editor, klik tab Design di bagian bawah layar.
2. Hapus TextView yang berbunyi "Hello World."
3. Tambahkan Tombol ke layout dalam posisi apa pun.
4. Beralih ke XML Editor (klik tab Text) dan modifikasi atribut ini di dalam Tombol:

Atribut	Nilai
<code>android:id</code>	<code>"@+id/button_main"</code>
<code>android:layout_width</code>	<code>wrap_content</code>
<code>android:layout_height</code>	<code>wrap_content</code>
<code>android:layout_alignParentRight</code>	<code>"true"</code>
<code>android:layout_alignParentBottom</code>	<code>"true"</code>
<code>android:layout_alignParentEnd</code>	<code>"true"</code>
<code>android:text</code>	<code>"Send"</code>
<code>android:onClick</code>	<code>"launchSecondActivity"</code>

5. Ini mungkin membuat kesalahan "Method `launchSecondActivity` is missing in MainActivity." Abaikan kesalahan ini sekarang. Masalah ini akan diatasi pada tugas berikutnya.
6. Letakkan kursor pada kata "Send".

7. Tekan **Alt-Enter** (**Option-Enter** di Mac), dan pilih **Extract string resources**.
8. Setel **nama Resource** ke `button_main` dan klik **OK**.  
Proses ini membuat resource string dalam file `values/res/string.xml`, dan string dalam kode Anda digantikan dengan referensi ke resource string tersebut.
9. Pilih **Code > Reformat Code** untuk memformat kode XML, jika perlu.
10. Pratinjau layout activity utama menggunakan Layout Editor. Layout harus terlihat seperti ini:



### Kode Solusi:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"

tools:context="com.example.android.twoactivities.MainActivity"
">
    <Button
        android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:text="@string/button_main"
        android:id="@+id/button_main"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:onClick="launchSecondActivity"/>
</RelativeLayout>

```

### Mendefinisikan tindakan tombol

Dalam tugas ini Anda akan mengimplementasikan metode `onClick` yang didefinisikan dalam layout.

1. Dalam Editor XML, letakkan kursor pada kata `launchSecondActivity`.
2. Tekan **Alt-Enter** (**Option-Enter** di Mac) dan pilih **Create 'launchSecondActivity(View)' dalam 'MainActivity'**.

File `MainActivity.java` terbuka, dan Android Studio membuat metode kerangka untuk handler `onClick`.

3. Dalam `launchSecondActivity`, tambahkan pernyataan log yang bernama `"Button Clicked!"`

```
Log.d(LOG_TAG, "Button clicked!")
```

`LOG_TAG` akan terlihat berwarna merah. Definisi untuk variabel tersebut akan ditambahkan dalam langkah selanjutnya.

4. Tempatkan kursor pada kata `"Log"` dan tekan **Alt-Enter** (**Option-Enter** di Mac). Android Studio menambahkan pernyataan impor untuk `android.util.Log`.
5. Di bagian atas kelas, tambahkan konstanta untuk variabel `LOG_TAG`:

```
private static final String LOG_TAG =
MainActivity.class.getSimpleName();
```

Konstanta ini menggunakan nama kelas itu sendiri sebagai tag-nya.

6. Jalankan aplikasi Anda. Saat mengklik tombol `"Send"`, Anda akan melihat pesan `"Button Clicked!"` di Android Monitor (logcat). Jika ada terlalu banyak keluaran di monitor, ketik `MainActivity` di dalam kotak pencarian dan log hanya akan menunjukkan baris yang cocok dengan tag tersebut.

## Latihan 5.2. Membuat dan meluncurkan activity kedua

Setiap activity baru yang Anda tambahkan ke proyek memiliki layout dan file Java-nya sendiri, terpisah dari milik activity utama. Activity tersebut juga memiliki elemen `<activity>` sendiri dalam manifes Android. Seperti activity utama, activity baru yang Anda buat di Android Studio juga meluas dari kelas `AppCompatActivity`.

Semua activity di aplikasi terhubung secara longgar dengan satu sama lain. Namun, Anda dapat mendefinisikan sebuah activity sebagai parent dari akti lain dalam file `AndroidManifest.xml`. Hubungan parent-anak ini memungkinkan Android untuk menambahkan petunjuk navigasi seperti panah yang menghadap ke kiri pada bilah judul untuk setiap activity.

Activity berkomunikasi dengan satu sama lain (di dalam aplikasi yang sama dan di berbagai aplikasi yang berbeda) dengan *intent*. Ada dua tipe intent, eksplisit dan implisit. Intent eksplisit adalah intent yang targetnya Anda ketahui. Anda sudah mengetahui nama kelas yang sepenuhnya memenuhi syarat dari activity spesifik tersebut. Intent implisit adalah intent yang nama komponen targetnya tidak Anda ketahui, namun memiliki tindakan umum untuk dikerjakan. Anda akan mempelajari tentang intent implisit pada praktik berikutnya.

Pada tugas ini Anda akan menambahkan activity kedua pada aplikasi, dengan layout-nya sendiri. Anda akan memodifikasi manifes Android untuk mendefinisikan activity utama sebagai parent activity kedua. Lalu Anda akan memodifikasi metode kejadian `onClick` dalam activity utama untuk menyertakan intent yang meluncurkan activity kedua saat tombol diklik.

### Membuat activity kedua

1. Klik folder **app** untuk proyek Anda dan pilih **File > New > Activity > Empty Activity**.
2. Namakan activity baru "SecondActivity." Pastikan **Generate Layout File** dicentang, dan nama layout akan diisi dengan `activity_second`.
3. Klik **Finish**. Android Studio menambahkan layout activity baru (`activity_second`) dan file Java baru (`SecondActivity`) ke proyek Anda untuk activity baru tersebut. Ini juga akan memperbarui manifes Android untuk menyertakan activity baru.

### Memodifikasi manifes Android

1. Buka `manifests/AndroidManifest.xml`.

2. Temukan elemen `<activity>` yang dibuat Android Studio untuk activity kedua.

```
<activity  
android:name=".SecondActivity"></activity>
```

3. Tambahkan atribut ini ke elemen `<activity>`:

Atribut	Nilai
<code>android:label</code>	"Second Activity"
<code>android:parentActivityName</code>	".MainActivity"

4. Atribut `label` menambahkan judul activity ke bilah tindakan.
5. Atribut `parentActivityName` menandakan bahwa activity utama adalah parent activity kedua. Hubungan activity parent ini digunakan untuk navigasi "ke atas" di dalam aplikasi Anda. Dengan mendefinisikan atribut ini, bilah tindakan untuk activity kedua akan muncul dengan panah yang menghadap ke kiri untuk memungkinkan pengguna untuk bergerak "ke atas" ke activity utama.
6. Tempatkan kursor di "Second Activity" dan tekan **Alt-Enter** (**Option-Enter** di Mac).
7. Pilih **Extract string resource**, namakan resource "activity2\_name", dan klik **OK**. Android Studio menambahkan resource string untuk label activity.
8. Tambahkan elemen `<meta-data>` element inside the `<activity>` di dalam elemen untuk activity kedua. Gunakan atribut ini:

Atribut	Nilai
<code>android:name</code>	"android.support.PARENT_ACTIVITY"
<code>android:value</code>	"com.example.android.twoactivities.MainActivity"

9. Elemen `<meta-data>` menyediakan informasi arbitrer tambahan tentang activity sebagai pasangan kunci-nilai. Dalam hal ini, atribut ini mencapai hal yang sama dengan atribut `android:parentActivityName`, yakni mendefinisikan hubungan antara dua activity untuk tujuan navigasi ke atas. Atribut ini diperlukan untuk versi Android lebih lama.

`android:parentActivityName` hanya tersedia untuk API level 16 dan yang lebih tinggi.

### Mendefinisikan layout untuk activity utama

1. Buka `res/layout/activity_second.xml`.
2. Tambahkan `TextView` ("Plain Textview" dalam Layout Editor). Berikan atribut berikut pada `TextView`:

Atribut	Nilai
<code>android:id</code>	<code>"@+id/text_header"</code>
<code>android:layout_width</code>	<code>wrap_content</code>
<code>android:layout_height</code>	<code>wrap_content</code>
<code>android:layout_marginBottom</code>	<code>"@dimen/activity_vertical_margin"</code>
<code>android:text</code>	<code>"Message Received"</code>
<code>android:textAppearance</code>	<code>"?android:attr/textAppearanceMedium"</code>
<code>android:textStyle</code>	<code>"bold"</code>

3. Nilai `textAppearance` adalah atribut bertema Android yang mendefinisikan gaya font dasar font kecil, medium, dan besar. Anda akan mempelajari selengkapnya tentang tema pada pelajaran berikutnya.
4. Ekstrak string `"Message Received"` ke resource bernama `text_header`.
5. Pratinjau layout dalam Layout Editor. Layout harus terlihat seperti ini:

### Menambahkan intent ke activity utama

Dalam activity ini Anda akan menambahkan intent eksplisit ke activity utama. Intent ini digunakan untuk mengaktifkan activity kedua saat tombol Send diklik.

1. Buka file Java untuk `MainActivity` (`java/com.example.android.twoactivities/MainActivity`).
2. Buat intent baru dalam metode `launchSecondActivity()`.

Konstruktor intent memerlukan dua argumen untuk intent eksplisit: konteks Aplikasi dan komponen spesifik yang akan menerima intent tersebut.



Di sini Anda harus menggunakan `this` sebagai konteksnya, dan `SecondActivity.class` sebagai kelas spesifiknya.

```
Intent intent = new Intent(this,
SecondActivity.class)
```

3. Tempatkan kursor di `intent` dan tekan **Alt-Enter** (**Option-Enter** di Mac) untuk menambahkan impor ke kelas `intent`.
4. Panggil metode `startActivity()` dengan `intent` baru sebagai argumennya.
5. `startActivity(intent);`
6. Jalankan aplikasi.

Saat mengeklik tombol `Send`, `activity` utama mengirim `intent` dan sistem Android membuka `activity` kedua. `Activity` kedua muncul di layar. Untuk kembali ke `activity` utama, klik tombol `Kembali Android` di kiri bawah layar, Anda bisa menggunakan panah kiri di atas `activity` kedua untuk kembali ke `activity` utama.

- **Tantangan:**

Apa yang terjadi jika Anda menghapus `android:parentActivityName` dan elemen `<meta-data>` dari `manifest`? Buat perubahan ini dan jalankan aplikasi.

### Latihan 5.3. Mengirim data dari activity utama ke activity kedua

Dalam tugas terakhir, Anda akan menambahkan intent eksplisit ke activity utama yang mengaktifkan activity kedua. Anda juga bisa menggunakan intent untuk mengambil data dari activity satu ke activity lainnya.

Dalam tugas ini, Anda akan memodifikasi intent eksplisit dalam activity utama untuk menyertakan data tambahan (dalam kasus ini, string yang dimasukkan pengguna) dalam ekstra intent. Lalu Anda akan memodifikasi activity kedua untuk mendapatkan data kembali dari ekstra intent dan menampilkannya di layar.

#### Menambahkan EditText ke layout activity utama

1. Buka res/layout/activity\_main.xml.
2. Tambahkan tampilan EditText (Plain Text dalam Layout Editor.)  
Berikan EditText atribut berikut:

Atribut	Nilai
android:id	"@+id/editText_main"
android:layout_width	match_parent
android:layout_height	wrap_content
android:layout_toLeftOf	"@+id/button_main"
android:layout_toStartOf	"@+id/button_main"
android:layout_alignParentBottom	"true"
android:hint	"Enter Your Message Here"

3. Hapus android:text attribute.
4. Ekstrak string "Enter Your Message Here" ke resource bernama editText\_main.

#### **Kode Solusi:**

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/androi
d"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
```

```
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
```

```
tools:context="com.example.android.twoactivities.MainActivity"
ivity">
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_main"
    android:id="@+id/button_main"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:onClick="launchSecondActivity"/>
```

```
<EditText
    android:id="@+id/editText_main"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_toLeftOf="@+id/button_main"
    android:layout_toStartOf="@+id/button_main"
    android:hint="@string/editText_main" />
```

```
</RelativeLayout>
```

### Menambahkan string ke ekstra intent activity utama

Objek intent dapat meneruskan data ke activity target dalam dua cara: dalam bidang data, atau dalam *ekstra* intent. Data intent adalah URI yang menandakan data spesifik untuk dikerjakan. Jika informasi yang ingin diteruskan ke activity melalui intent bukan URI, atau jika Anda memiliki lebih dari satu informasi yang ingin dikirimkan, Anda bisa meletakkan informasi tambahan tersebut ke dalam *ekstra* intent.

*Ekstra* intent adalah pasangan kunci/nilai dalam Bundel. *Bundel* adalah kumpulan data, yang disimpan sebagai pasangan kunci/nilai. Untuk meneruskan informasi dari satu activity ke yang lainnya, letakkan kunci dan nilai ke dalam bundel ekstra intent dari activity mengirim, dan keluarkan lagi ke activity menerima.

1. **Buka**  
`java/com.example.android.twoactivities/MainActivity`.
2. Tambahkan konstanta publik di bagian atas kelas untuk mendefinisikan kunci untuk ekstra intent:

```
public static final String EXTRA_MESSAGE =  
    "com.example.android.twoactivities.extra.MESSAGE";
```

3. Tambahkan variabel privat di bagian atas kelas untuk menampung objek `EditText`. Mengimpor kelas `EditText`.

```
private EditText mMessageEditText
```

4. Dalam metode `onCreate()`, gunakan `findViewById` untuk mendapatkan referensi tentang contoh `EditText` dan menetapkannya ke variabel privat tersebut:

```
mMessageEditText = (EditText)  
    findViewById(R.id.editText_main);
```

5. Dalam metode `launchSecondActivity()`, di bawah intent baru, dapatkan teks dari `EditText` sebagai string:

```
String message =  
    mMessageEditText.getText().toString();
```

6. Tambahkan string tersebut ke intent sebagai sebuah ekstra dengan konstanta `EXTRA_MESSAGE` sebagai kunci dan string-nya sebagai nilai:

```
intent.putExtra(EXTRA_MESSAGE, message);
```

### Menambahkan TextView ke activity kedua untuk pesan

1. Buka `res/layout/activity_second.xml`.
2. Tambahkan `TextView` kedua. Berikan atribut berikut pada `TextView`:

Atribut	Nilai
android:id	"@+id/text_message"
android:layout_width	wrap_content
android:layout_height	wrap_content
android:layout_below	"@+id/text_header"
android:layout_marginLeft	"@dimen/activity_horizontal_margin"
android:layout_marginStart	"@dimen/activity_horizontal_margin"
android:textSize	"?android:attr/textAppearanceMedium"

### 3. Hapus atribut android:text (jika ada).

Layout baru untuk activity kedua terlihat sama dengan layout pada tugas sebelumnya karena TextView baru tidak berisi (belum) teks apa pun, sehingga tidak muncul di layar.

#### Kode Solusi:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"

tools:context="com.example.android.twoactivities.SecondActivit
y">

    <TextView
        android:id="@+id/text_header"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/text_header"

        android:layout_marginBottom="@dimen/activity_vertical_margin"

        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textStyle="bold"/>

    <TextView
```

```

        android:id="@+id/text_message"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_header"

        android:layout_marginLeft="@dimen/activity_horizontal_margin"
        android:layout_marginStart="@dimen/activity_horizontal_margin"
        android:textAppearance="?android:attr/textAppearanceMedium" />
</RelativeLayout>

```

## Memodifikasi activity kedua untuk mendapatkan ekstra dan menampilkan pesan

### 1. Buka

```
java/com.example.android.twoactivities/SecondActivity.
```

### 2. Dalam metode onCreate(), dapatkan intent yang mengaktifkan activity ini:

```
Intent intent = getIntent()
```

### 3. Dapatkan string berisi pesan dari ekstra intent menggunakan variabel statis MainActivity.EXTRA\_MESSAGE sebagai kuncinya:

```
String message =
intent.getStringExtra(MainActivity.EXTRA_MESSAGE);
```

### 4. Gunakan findViewById untuk mendapatkan referensi ke TextView untuk pesan dari layout (Anda mungkin perlu mengimpor kelas TextView):

```
TextView textView = (TextView)
findViewById(R.id.text_message);
```

### 5. Setel teks TextView tersebut ke string dari ekstra intent:

```
textView.setText(message)
```

### 6. Jalankan aplikasi. Saat Anda mengetik pesan dalam activity utama dan mengeklik Send, activity kedua terbuka dan menampilkan pesan tersebut.

## Kode Solusi:

```
package com.example.android.twoactivities;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;

public class SecondActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);

        Intent intent = getIntent();
        String message =

intent.getStringExtra(MainActivity.EXTRA_MESSAGE);
        TextView textView = (TextView)
        findViewById(R.id.text_message);
        textView.setText(message);
    }
}
```

## Latihan 5.4. Mengembalikan data ke activity utama

Sekarang karena Anda telah memiliki aplikasi yang meluncurkan activity baru dan mengirimkan data ke aplikasi, langkah terakhir adalah mengembalikan data ke activity utama. Anda juga akan menggunakan intent dan *ekstra* intent untuk tugas ini.

### Menambahkan EditText dan Tombol ke layout activity kedua

1. Salin EditText dan Tombol dari file layout activity utama dan tempel ke layout kedua.
2. Dalam file activity\_second.xml, modifikasi nilai atribut untuk Tombol dan tampilan EditText. Gunakan nilai ini:

Atribut Lama (Tombol)	Atribut Baru (Tombol)
android:id="@+id/button_main"	android:id="@+id/button_second"
android:onClick="launchSecondActivity"	android:onClick="returnReply"
android:text="@string/button_main"	android:text="@string/button_second"
Atribut Lama (EditText)	Atribut Baru (EditText)
android:id="@+id/editText_main"	android:id="@+id/editText_second"
android:layout_toLeftOf="@+id/button_main"	android:layout_toLeftOf="@+id/button_second"
android:layout_toStartOf="@+id/button_main"	android:layout_toStartOf="@+id/button_second"
android:hint="@string/editText_main"	android:hint="@string/editText_second"

3. Buka `res/values/strings.xml` dan tambahkan resource String untuk teks tombol dan petunjuknya di EditText:

```
<string name="button_second">Reply</string>
<string name="editText_second">Enter Your Reply
Here</string>
```

4. Dalam editor layout XML, letakkan kursor pada "returnReply", tekan **Alt-Enter** (**Option-Enter** di Mac) dan pilih Create 'launchSecondActivity(View)' dalam 'SecondActivity'.



File `SecondActivity.java` terbuka, dan Android Studio membuat metode kerangka untuk handler `onClick`. Anda akan mengimplementasikan metode ini dalam tugas berikutnya.

### Kode Solusi:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"

android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"

tools:context="com.example.android.twoactivities.SecondA
ctivity">

    <TextView
        android:id="@+id/text_header"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/text_header"

        android:layout_marginBottom="@dimen/activity_vertical_ma
        rgin"

        android:textAppearance="?android:attr/textAppearanceMedi
        um"
        android:textStyle="bold"/>

    <TextView
        android:id="@+id/text_message"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_header"

        android:layout_marginLeft="@dimen/activity_horizontal_ma
        rgin"

        android:layout_marginStart="@dimen/activity_horizontal_m
        rgin"
```

```

android:textAppearance="?android:attr/textAppearanceMedium" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_second"
        android:id="@+id/button_second"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:onClick="returnReply"/>

    <EditText
        android:id="@+id/editText_second"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_toLeftOf="@+id/button_second"
        android:layout_toStartOf="@+id/button_second"
        android:hint="@string/editText_second" />

</RelativeLayout>

```

## Membuat intent respons dalam activity kedua

### 1. Buka

```

java/com.example.android.twoactivities/SecondActivity.

```

2. Di atas kelas, tambahkan konstanta publik untuk mendefinisikan kunci untuk ekstra intent:

```

public static final String EXTRA_REPLY =
    "com.example.android.twoactivities.extra.REPLY"
;

```

3. Tambahkan variabel privat di bagian atas kelas untuk menampung objek EditText.

```

private EditText mReply

```

4. Dalam metode onCreate(), gunakan findViewById() untuk mendapatkan referensi tentang contoh EditText dan menentukannya ke variabel privat tersebut:

```
mReply = findViewById(R.id.editText_second);
```

5. Dalam metode `returnReply()`, dapatkan teks `EditText` sebagai string:

```
String reply = mReply.getText().toString();
```

Membuat intent baru untuk respons tersebut.

**Catatan:** Jangan gunakan ulang objek intent yang diterima dari permintaan asalnya. Buat intent baru untuk respons tersebut.

```
Intent replyIntent = new Intent();
```

6. Tambahkan string balasan dari `EditText` ke intent baru sebagai *ekstra* intent. Karena *ekstra* adalah pasangan kunci/nilai, di sini kuncinya adalah `EXTRA_REPLY` dan nilainya adalah balasan:

```
replyIntent.putExtra(EXTRA_REPLY, reply);
```

7. Setel hasilnya ke `RESULT_OK` untuk menunjukkan bahwa responsnya berhasil. Kode hasil (termasuk `RESULT_OK` dan `RESULT_CANCELLED`) didefinisikan oleh kelas `Activity`.

```
setResult(RESULT_OK, replyIntent);
```

8. Panggil `finish()` untuk menutup activity dan kembali ke activity utama.

```
finish();
```

**Kode Solusi:**

```
package com.example.android.twoactivities;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
```

```

import android.widget.TextView;

public class SecondActivity extends AppCompatActivity {
    public static final String EXTRA_REPLY =
"com.example.android.twoactivities.extra.REPLY";

    private EditText mReply;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
        mReply = (EditText)
findViewById(R.id.editText_second);

        Intent intent = getIntent();
        String message =

intent.getStringExtra(MainActivity.EXTRA_MESSAGE);
        TextView textView = (TextView)
findViewById(R.id.text_message);
        textView.setText(message);    }

    public void returnReply(View view) {
        String reply = mReply.getText().toString();

        Intent replyIntent = new Intent();
        replyIntent.putExtra(EXTRA_REPLY, reply);
        setResult(RESULT_OK, replyIntent);
        finish();    }
}

```

### Menambahkan TextViews ke layout activity utama untuk menampilkan balasan

Activity utama memerlukan cara untuk menampilkan balasan yang dikirim kembali ke activity kedua. Dalam tugas ini Anda akan menambahkan TextViews ke layout activity utama untuk menampilkan balasan tersebut. Untuk memudahkannya, Anda dapat menyalin TextViews yang digunakan dalam activity kedua.

1. Salin dua TextViews untuk tampilan pesan dari file layout activity kedua dan tempel ke layout utama di atas EditText yang ada dan tampilan Tombol.

- Modifikasi nilai atribut untuk kedua TextViews baru ini. Gunakan nilai ini:

Atribut Lama (header TextView)	Atribut Baru (header TextView)
android:id="@+id/text_header"	android:id="@+id/text_header_reply"
android:text="@string/text_header"	android:text="@string/text_header_reply"
Atribut Lama (TextView pesan)	Atribut Baru (TextView pesan)
android:id="@+id/text_message"	android:id="@+id/text_message_reply"
android:layout_below="@+id/text_header"	android:layout_below="@+id/text_header_reply"

- Tambahkan atribut `android:visibility` ke setiap TextView agar tidak terlihat di awal. (Membuatnya terlihat di layar tanpa konten apa pun dapat membuat pengguna bingung.) Anda akan menjadikan TextView terlihat setelah data respons diteruskan kembali dari activity kedua.

```
android:visibility="invisible"
```

- Buka `res/values/strings.xml` dan tambahkan resource string untuk header balasan:

```
<string name="text_header_reply">Reply  
Received</string>
```

Layout untuk activity utama terlihat sama dengan tugas sebelumnya, walaupun Anda sudah menambahkan dua TextView baru ke layout. Namun, karena Anda menyetel TextView menjadi tidak terlihat, TextView tidak muncul di layar.

### Kode Solusi:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context="com.example.android.twoactivities.MainActivity">
```

```

<TextView
    android:id="@+id/text_header_reply"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/text_header_reply"
    android:visibility="invisible"

android:layout_marginBottom="@dimen/activity_vertical_margin"

android:textAppearance="?android:attr/textAppearanceMedium"
    android:textStyle="bold"/>

<TextView
    android:id="@+id/text_message_reply"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/text_header_reply"
    android:visibility="invisible"

android:layout_marginLeft="@dimen/activity_horizontal_margin"

android:layout_marginStart="@dimen/activity_horizontal_margin"

android:textAppearance="?android:attr/textAppearanceMedium" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_main"
    android:id="@+id/button_main"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:onClick="launchSecondActivity"/>

<EditText
    android:id="@+id/editText_main"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_toLeftOf="@+id/button_main"
    android:layout_toStartOf="@+id/button_main"
    android:hint="@string/editText_main" />
</RelativeLayout>

```

## Latihan 5.5. Mendapatkan balasan dari ekstra intent

Saat Anda menggunakan intent eksplisit untuk memulai activity, Anda mungkin tidak berharap untuk mendapatkan data kembali, Anda hanya mengaktifkan activity tersebut. Dalam kasus seperti ini, gunakan `StartActivity()` untuk memulai activity baru, seperti yang Anda lakukan sebelumnya pada pelajaran ini. Tetapi jika ingin mendapatkan data kembali dari activity yang diaktifkan, Anda perlu memulainya dengan `startActivityForResult()`.

Dalam tugas ini Anda akan memodifikasi aplikasi untuk memulai activity kedua dan mengharapkan hasil, untuk mengekstrak data pengembalian itu dari intent, dan untuk menampilkan data tersebut dalam `TextView` yang dibuat dalam tugas terakhir.

1. Buka `java/com.example.android.twoactivities/MainActivity`.
2. Tambahkan konstanta publik di bagian atas kelas untuk mendefinisikan kunci untuk tipe respons tertentu yang Anda inginkan:

```
public static final int TEXT_REQUEST = 1
```

3. Tambahkan dua variabel privat untuk menampung header balasan dan `TextViews` balasan:

```
private TextView mReplyHeadTextView;  
private TextView mReplyTextView;
```

4. dalam metode `onCreate()`, gunakan `findViewById` untuk mendapatkan referensi dari layout ke header balasan dan `TextView` balasan. Tetapkan dua contoh tampilan ke variabel privat:

```
mReplyHeadTextView = (TextView)  
findViewById(R.id.text_header_reply);  
mReplyTextView = (TextView)  
findViewById(R.id.text_message_reply);
```

5. Dalam metode `launchSecondActivity()`, modifikasi panggilan ke `startActivity()` menjadi `startActivityForResult()`, dan sertakan kunci `TEXT_REQUEST` sebagai argumen:

```
startActivityForResult(intent, TEXT_REQUEST);
```

6. Buat metode callback `onActivityResult()` dengan tanda tangan ini:

```
public void onActivityResult(int requestCode, int  
resultCode, Intent data) {}
```

7. Dalam `onActivityResult()`, panggil `super.onActivityResult()`:

```
super.onActivityResult(requestCode, resultCode, data)
```

8. Tambahkan kode untuk `TEXT_REQUEST` (untuk memproses hasil intent yang benar, jika ada beberapa) dan `RESULT_CODE` (untuk memastikan permintaannya sukses):

```
if (requestCode == TEXT_REQUEST) {  
    if (resultCode == RESULT_OK) {  
    }  
}
```

9. Di bagian dalam blok `if`, dapatkan ekstra intent dari intent respons (`data`). Di sini, kunci untuk ekstra adalah konstanta `EXTRA_REPLY` dari `SecondActivity`:

```
String reply =  
data.getStringExtra(SecondActivity.EXTRA_REPLY);
```

10. Setel; visibilitas header balasan ke benar:

```
mReplyHeadTextView.setVisibility(View.VISIBLE)
```

11. Setel teks `textView` balasan ke balasan, dan setel visibilitasnya ke benar:

```
mReplyTextView.setText(reply);  
mReplyTextView.setVisibility(View.VISIBLE);
```

12. Jalankan aplikasi.

Sekarang, saat Anda mengirimkan pesan ke `activity` kedua dan mendapatkan balasan kembali, `activity` utama diperbarui untuk menampilkan balasan.

### ▪ Tantangan

Buat aplikasi dengan tiga tombol yang diberi label: `Text One`, `Text Two`, dan `Text Three`. Saat tombol mana saja diklik, `activity` kedua terbuka. `Activity` kedua itu harus berisi `ScrollView` yang menampilkan salah satu dari tiga bagian teks (Anda bisa menyertakan pilihan bagian). Gunakan intent untuk membuka `activity` kedua dan ekstra intent untuk menunjukkan mana dari tiga bagian tersebut yang akan ditampilkan.



Dalam Praktik ini, Anda telah belajar :

1. Sebuah Activity adalah komponen aplikasi yang menyediakan layar tunggal yang fokus pada tugas seorang pengguna.
  - a. Setiap activity memiliki file layout antarmuka penggunanya masing-masing.
  - b. Anda bisa menetapkan hubungan parent/anak pada activity untuk mengaktifkan navigasi "ke atas" di dalam aplikasi.
  - c. Untuk mengimplementasikan activity di dalam aplikasi, lakukan hal berikut:
2. Buat kelas Java activity.
3. Implementasikan antarmuka pengguna untuk activity itu.
4. Deklarasikan activity baru itu dalam manifes aplikasi.
5. Saat membuat proyek baru untuk aplikasi, atau menambahkan activity baru ke aplikasi, dalam Android Studio (dengan File > new > Activity), kode template untuk setiap tugas ini disediakan untuk Anda.
6. Intent mengizinkan Anda meminta tindakan dari komponen lain dalam aplikasi, misalnya, untuk memulai satu activity dari lainnya. Intent dapat menjadi eksplisit atau implisit.
  - a. Dengan intent eksplisit Anda mengindikasikan komponen target spesifik untuk menerima data.
  - b. Dengan intent implisit Anda menetapkan fungsionalitas yang Anda inginkan tetapi tidak dengan komponen target.
  - c. Intent dapat menyertakan data tempat melakukan tindakan (sebagai URI) atau informasi tambahan sebagai *ekstra* intent.
  - d. *Ekstra* intent adalah pasangan kunci/nilai dalam bundel yang dikirimkan bersama intent-nya.
7. Tampilan dapat dijadikan terlihat atau tidak terlihat dengan atribut `android:visibility`.

## REFERENSI

- Buku Kursus Dasar-Dasar Developer Android – Konsep, Belajar mengembangkan aplikasi Android, Dikembangkan oleh Tim Pelatihan Developer Google.
- Buku Kerja Praktik, Kursus Dasar-Dasar Developer Android — Praktik, Belajar mengembangkan aplikasi Android, Dikembangkan oleh Tim Pelatihan Developer Google



## MATERI KOMPETENSI

Materi bab ini akan membahas:

- 1) Tentang preferensi bersama (shared preference)
- 2) Penyimpanan internal
- 3) Penyimpanan eksternal
- 4) Database SQLite
- 5) File dan opsi storage lain
- 6) Mencadangkan aplikasi dan koneksi jaringan

## YANG HARUS SUDAH ANDA KETAHUI

Untuk materi dan praktik di bab ini Anda harus bisa:

- 1) Membuat, membangun, dan menjalankan aplikasi di Android Studio.
- 2) Merancang layout dengan tombol dan tampilan teks.
- 3) Menyimpan dan memulihkan keadaan instance aktivitas.

## YANG AKAN ANDA BUTUHKAN

Untuk materi dan praktik ini, Anda akan membutuhkan:

- 1) Komputer Mac, Windows, atau Linux. Tool Android Studio untuk persyaratan sistem terbaru.
- 2) Akses internet atau cara alternatif untuk memuat Android Studio dan pemasangan Java terbaru ke komputer Anda.

## YANG AKAN ANDA PELAJARI

Anda akan belajar:

- 1) Mengenali apa yang dimaksud dengan preferensi bersama.
- 2) Membuat file preferensi bersama untuk aplikasi Anda.
- 3) Menyimpan data di preferensi bersama dan membaca preferensi tersebut kembali.

# BAB 6 SHARED PREFERENCES, FILE DAN PENYIMPANAN DATA



## Bab 6: Shared Preference, File dan Penyimpanan Data

### 6.1. Shared Preference

Preferensi bersama memungkinkan Anda membaca dan menulis data primitif dalam jumlah kecil sebagai pasangan kunci/nilai ke file pada storage perangkat. Kelas `SharedPreferences` menyediakan API untuk mendapatkan handle ke file preferensi dan untuk membaca, menulis, dan mengelola data ini. File preferensi bersama ini sendiri dikelola oleh framework Android, dan bisa diakses oleh (dibagikan dengan) semua komponen aplikasi Anda. Akan tetapi, data tidak dibagikan dengan atau bisa diakses other aplikasi lain.

Data yang Anda simpan ke preferensi bersama berbeda dengan data di status aktivitas yang disimpan yang telah Anda pelajari di bab sebelumnya. Data di keadaan instance aktivitas dipertahankan di instance aktivitas di sesi pengguna yang sama. Preferensi bersama tetap ada di semua sesi pengguna, meskipun aplikasi Anda ditutup dan dimulai ulang atau perangkat dimulai ulang. Gunakan preferensi bersama hanya jika Anda perlu menyimpan data dalam jumlah kecil sebagai pasangan kunci/nilai.


Android menyediakan sejumlah opsi bagi Anda untuk menyimpan data aplikasi yang persisten. Solusi yang dipilih bergantung pada kebutuhan khusus Anda, misalnya apakah data harus bersifat privat untuk aplikasi Anda atau bisa diakses oleh aplikasi lainnya (dan pengguna) serta berapa banyak ruang yang diperlukan data.

Opsi storage data Anda adalah seperti berikut:

Preferensi bersama	Menyimpan data primitif pribadi dalam pasangan nilai-kunci Hal ini akan dibahas di bab berikutnya
--------------------	--

Penyimpanan internal	Menyimpan data privat pada memori perangkat
Penyimpanan eksternal	Menyimpan data publik pada penyimpanan eksternal bersama.
Database SQLite	Menyimpan data terstruktur dalam database privat
Koneksi jaringan	Menyimpan data di web dengan server jaringan sendiri.
Pencadangan Awan	Membuat cadangan data pengguna dan aplikasi di awan
Penyedia materi	Menyimpan data secara privat dan membuatnya tersedia secara publik
Database realtime Firebase	Menyimpan dan menyinkronkan data dengan database awan NoSQL. Data disinkronkan dengan semua klien yang terhubung secara realtime, dan tetap tersedia bila aplikasi Anda offline

Menggunakan preferensi bersama adalah cara untuk membaca dan menulis pasangan nilai-kunci informasi secara persisten ke dan dari suatu file.


	<p><b>Catatan:</b> Secara default, pasangan nilai-kunci ini tidak digunakan bersama, bukan pula preferensi, jadi jangan mencampuradukkannya dengan Preference API.</p>
---	--

Preferensi bersama memungkinkan Anda membaca dan menulis data primitif dalam jumlah kecil sebagai pasangan kunci/nilai ke file pada storage perangkat. Kelas `SharedPreferences` menyediakan API untuk mendapatkan handle ke file preferensi serta untuk membaca, menulis, dan mengelola data ini. File preferensi bersama ini sendiri dikelola oleh kerangka kerja dan bisa diakses (digunakan bersama) oleh semua komponen aplikasi Anda. Data tersebut tidak digunakan bersama atau bisa diakses oleh aplikasi lainnya. Untuk mengelola data dalam jumlah besar, gunakan database SQLite atau opsi storage lainnya yang cocok.

Tabel 12. Preferensi bersama vs. keadaan instance tersimpan

<b>Preferensi Bersama</b>	<b>Keadaan instance tersimpan</b>
Bertahan di semua sesi pengguna, meskipun aplikasi Anda ditutup dan dimulai ulang, atau perangkat di-boot ulang.	Mempertahankan data keadaan di seluruh instance aktivitas dalam sesi pengguna yang sama.

Data yang harus diingat di semua sesi, misalnya setelan pilihan pengguna atau skor game mereka.	Data yang tidak boleh diingat di semua sesi, misalnya tab yang dipilih, atau keadaan aktivitas saat ini.
Sejumlah kecil pasangan kunci/nilai	Sejumlah kecil pasangan kunci/nilai
Data bersifat privat untuk aplikasi.	Data bersifat privat untuk aplikasi.
Penggunaan umumnya adalah untuk menyimpan preferensi pengguna.	Kita biasanya membuat ulang keadaan setelah perangkat diputar.

	<p><b>Catatan:</b> SharedPreference API juga berbeda dari Preference API. Preference API bisa digunakan untuk membangun antarmuka pengguna untuk laman setelan, dan tidak menggunakan preferensi bersama bagi implementasi dasarnya.</p>
---	--

- **Membuat file preferensi bersama file**

Anda hanya memerlukan satu file preferensi bersama untuk aplikasi, dan biasanya diberi nama dengan nama paket aplikasi. Hal ini membuat namanya unik dan mudah dikaitkan dengan aplikasi Anda.

Anda membuat file preferensi bersama dalam metode onCreate() aktivitas utama dan menyimpannya dalam sebuah variabel anggota.

```
private String sharedPrefFile =
    "com.example.android.hellosharedprefs";
mPreferences = getSharedPreferences(sharedPrefFile,
    MODE_PRIVATE);
```

Diperlukan argumen mode, karena Android versi lama memiliki mode lain yang memungkinkan Anda membuat file preferensi bersama yang bisa dibaca dan ditulis oleh umum. Mode ini tidak digunakan lagi di API 17, dan sekarang **sangat tidak dianjurkan** karena alasan keamanan. Jika Anda perlu berbagi data dengan aplikasi lainnya, gunakan layanan atau penyedia materi.

- **Menyimpan Preferensi Bersama**

Anda menyimpan preferensi dalam keadaan onPause() daur hidup aktivitas dengan menggunakan antarmuka SharedPreferences.Editor.

- 1) Dapatkan SharedPreferences.Editor. Editor menangani semua operasi file untuk Anda. Bila dua editor memodifikasi preferensi pada waktu yang sama, maka yang terakhir dipanggil akan menang.

- 2) Tambahkan pasangan kunci/nilai ke editor dengan menggunakan metode `put` yang sesuai untuk tipe data tersebut. Metode `put` akan menimpa nilai yang ada sebelumnya pada kunci yang ada.
- 3) Panggil `apply()` untuk menuliskan perubahan Anda. Metode `apply()` menyimpan preferensi secara asinkron, di luar thread UI. Editor preferensi bersama juga memiliki metode `commit()` untuk menyimpan preferensi secara sinkron. Metode `commit()` tidak disarankan karena bisa memblokir operasi lain. Karena instance `SharedPreferences` adalah singleton dalam suatu proses, maka aman untuk mengganti instance `commit()` dengan `apply()` jika Anda sudah mengabaikan nilai kembalian.

Anda tidak perlu mengkhawatirkan daur hidup komponen Android dan interaksinya dengan penulisan `apply()` ke disk. Kerangka kerja memastikan penulisan disk yang sedang diakses dari `apply()` selesai sebelum mengubah keadaan.

```
@Override
protected void onPause() {
    super.onPause();
    SharedPreferences.Editor preferencesEditor =
mPreferences.edit();
    preferencesEditor.putInt("count", mCount);
    preferencesEditor.putInt("color", mCurrentColor);
    preferencesEditor.apply();}
```

- Memulihkan preferensi bersama

Anda memulihkan preferensi bersama dalam metode `onCreate()` aktivitas. Metode `get()` mengambil dua argumen— satu untuk kunci dan satu lagi untuk nilai default jika kunci tidak bisa ditemukan. Dengan menggunakan argumen default, Anda tidak perlu menguji apakah preferensi ada dalam file.

```
mPreferences = getSharedPreferences(sharedPrefFile,
MODE_PRIVATE);
if (savedInstanceState != null) {
    mCount = mPreferences.getInt("count", 1);
    mShowCount.setText(String.format("%s", mCount));

    mCurrentColor = mPreferences.getInt("color",
mCurrentColor);
    mShowCount.setBackgroundColor(mCurrentColor);
} else { ... }
```



- Mengosongkan preferensi bersama

Untuk mengosongkan semua nilai dalam file preferensi bersama, panggil metode `clear()` pada editor preferensi bersama dan terapkan perubahan.

```
SharedPreferences.Editor preferencesEditor =
mPreferences.edit();
preferencesEditor.putInt("number", 42);
preferencesEditor.clear();
preferencesEditor.apply();
```

Anda bisa mengombinasikan panggilan untuk `put` dan `clear`. Akan tetapi, saat menerapkan preferensi, `clear` selalu dilakukan terlebih dahulu, tanpa menghiraukan apakah Anda memanggil metode `clear` sebelum atau setelah metode `put` pada editor ini.

- Mendengarkan perubahan preferensi

Ada sejumlah alasan yang membuat Anda perlu diberi tahu segera setelah pengguna mengubah salah satu preferensi. Untuk menerima callback saat perubahan terjadi pada salah satu preferensi, implementasikan antarmuka `SharedPreferences.OnSharedPreferenceChangeListener` dan daftarkan listener untuk objek `SharedPreferences` dengan memanggil `registerOnSharedPreferenceChangeListener()`. Antarmuka tersebut hanya memiliki satu metode callback, `onSharedPreferenceChanged()`, dan Anda bisa mengimplementasikan antarmuka sebagai bagian dari aktivitas.

```
public class SettingsActivity extends PreferenceActivity
implements OnSharedPreferenceChangeListener {
    public static final String KEY_PREF_SYNC_CONN =
"pref_syncConnectionType";
    ...
    public void onSharedPreferenceChanged(SharedPreferences
sharedPreferences,
        String key) {
        if (key.equals(KEY_PREF_SYNC_CONN)) {
            Preference connectionPref = findPreference(key);
            // Set summary to be the user-description for the selected value
            connectionPref.setSummary(sharedPreferences.getString(key,
""));
        }
    }
}
```

Dalam contoh ini, metode akan memeriksa apakah setelan yang diubah adalah untuk kunci preferensi yang diketahui. Ini akan memanggil `findPreference()` untuk mendapatkan objek `Preference` yang diubah agar bisa memodifikasi rangkuman item menjadi keterangan pilihan pengguna.

Untuk pengelolaan daur hidup yang tepat dalam aktivitas, daftarkan dan cabut pendaftaran `SharedPreferences.OnSharedPreferenceChangeListener` Anda, masing-masing selama callback `onResume()` dan `onPause()`:

```
@Override
protected void onResume() {
    super.onResume();
    getPreferenceScreen().getSharedPreferences()
        .registerOnSharedPreferenceChangeListener(this);
}

@Override
protected void onPause() {
    super.onPause();
    getPreferenceScreen().getSharedPreferences()
        .unregisterOnSharedPreferenceChangeListener(this);
}
```

- **Menahan referensi ke listener**

Bila Anda memanggil `registerOnSharedPreferenceChangeListener()`, pengelola preferensi saat ini tidak akan langsung menyimpan referensi ke listener. Anda harus menahan referensi ke listener, atau referensi akan rentan terhadap pengumpulan sampah. Pertahankan referensi ke listener dalam data instance suatu objek yang akan ada selama Anda memerlukan listener.

```
SharedPreferences.OnSharedPreferenceChangeListener listener =
    new SharedPreferences.OnSharedPreferenceChangeListener() {
    public void onSharedPreferenceChanged(SharedPreferences
    prefs, String key) {
        // listener implementation
    }
};
prefs.registerOnSharedPreferenceChangeListener(listener);
```

Android menggunakan sistem file yang serupa dengan sistem file berbasis disk pada platform lainnya seperti Linux. Operasi berbasis disk tentunya familier bagi setiap orang yang telah menggunakan I/O file Linux atau paket `java.io`.

Semua perangkat Android memiliki dua area file-storage: `storage` “internal” dan “eksternal”. Nama-nama ini berasal dari masa awal Android, saat

kebanyakan perangkat menawarkan memori bawaan non-volatil (penyimpanan internal), plus media storage lepas-pasang seperti kartu micro-SD (penyimpanan eksternal).

Saat ini, beberapa perangkat membagi ruang storage permanen menjadi partisi "internal" dan "eksternal", sehingga walaupun tanpa media storage lepas-pasang, selalu ada dua ruang storage dan perilaku API sama, ada atau tidak ada penyimpanan eksternal. Daftar berikut merangkum fakta tentang setiap ruang storage.

<b>Penyimpanan internal</b>	<b>Penyimpanan eksternal</b>
Selalu tersedia.	Tidak selalu tersedia, karena pengguna bisa memasang penyimpanan eksternal sebagai storage USB, dan dalam beberapa kasus, bisa melepasnya dari perangkat.
Hanya aplikasi Anda yang bisa mengakses file. Secara khusus, direktori penyimpanan internal aplikasi ditetapkan oleh nama paket aplikasi Anda di lokasi khusus sistem file Android. Aplikasi lain tidak bisa menjelajah direktori internal Anda dan tidak memiliki akses baca atau tulis kecuali jika Anda secara eksplisit menyetel file itu agar bisa dibaca atau bisa ditulis.	Bisa dibaca semua orang. Setiap aplikasi bisa membacanya.
Bila pengguna mencopot pemasangan aplikasi, sistem akan membuang semua file aplikasi Anda dari penyimpanan internal.	Bila pengguna mencopot pemasangan aplikasi, sistem akan membuang file aplikasi dari sini hanya jika Anda menyimpannya dalam direktori dari <code>getExternalFilesDir()</code> .
Penyimpanan internal adalah yang terbaik bila ingin memastikan bahwa pengguna maupun aplikasi lain tidak bisa mengakses file Anda.	Penyimpanan eksternal adalah tempat terbaik untuk file yang tidak memerlukan pembatasan akses dan untuk file yang ingin Anda gunakan bersama aplikasi lain atau memungkinkan pengguna mengaksesnya dengan komputer.

## 6.2. Penyimpanan Internal dan Eksternal

### a. Penyimpanan Internal

Anda tidak memerlukan izin apa pun untuk menyimpan file pada penyimpanan internal. Aplikasi Anda selalu memiliki izin untuk membaca dan menulis file dalam direktori penyimpanan internalnya.

Anda bisa membuat file dalam dua direktori berbeda:

- Storage permanen: `getFilesDir()`
- Storage sementara: `getCacheDir()`. Disarankan untuk file kecil dan sementara dengan total kurang dari 1 MB. Perhatikan, sistem mungkin menghapus file sementara jika kehabisan memori.

Untuk membuat file baru di salah satu direktori ini, Anda bisa menggunakan konstruktor `File()`, meneruskan `File` yang disediakan oleh salah satu metode di atas yang menetapkan direktori penyimpanan internal Anda. Misalnya:

```
File file = new File(context.getFilesDir(), filename);
```

Atau, Anda bisa memanggil `openFileOutput()` untuk mendapatkan `FileOutputStream` yang menulis ke file dalam direktori internal. Misalnya, inilah cara menulis sejumlah teks ke file:

```
String filename = "myfile";
String string = "Hello world!";
FileOutputStream outputStream;

try {
    outputStream = openFileOutput(filename,
Context.MODE_PRIVATE);
    outputStream.write(string.getBytes());
    outputStream.close();
} catch (Exception e) {
    e.printStackTrace();
}
```

Atau, jika Anda perlu menyimpan beberapa file ke cache, lebih baik gunakan `createTempFile()`. Misalnya, metode berikut akan mengekstrak nama file dari URL dan membuat file dengan nama itu dalam direktori cache internal aplikasi Anda:

```
public File getTempFile(Context context, String url) {
    File file;
    try {
```

```

String fileName =
Uri.parse(url).getLastPathSegment();
file = File.createTempFile(fileName, null,
context.getCacheDir());
} catch (IOException e) {
// Error while creating file
}
return file;
}

```

Anda bisa menyimpan file secara langsung pada penyimpanan internal perangkat. Secara default, file yang disimpan ke penyimpanan internal adalah privat untuk aplikasi Anda dan aplikasi lain tidak bisa mengaksesnya (begitu juga penggunaannya). Bila pengguna mencopot pemasangan aplikasi Anda, file-file ini akan dibuang.

Untuk membuat dan menulis file privat ke penyimpanan internal:

1. Panggil `openFileOutput()` dengan nama file dan mode pengoperasian. Proses ini mengembalikan `FileOutputStream`.
2. Tulis ke file dengan `write()`.
3. Tutup aliran dengan `close()`.

Misalnya:

```

String FILENAME = "hello_file";
String string = "hello world!";
FileOutputStream fos = openFileOutput(FILENAME,
Context.MODE_PRIVATE);
fos.write(string.getBytes());
fos.close();

```

`MODE_PRIVATE` akan membuat file (atau mengganti file dengan nama yang sama) dan membuatnya privat untuk aplikasi Anda. Mode lain yang tersedia adalah: `MODE_APPEND`, `MODE_WORLD_READABLE`, dan `MODE_WORLD_WRITEABLE`.



**Catatan:** Konstanta `MODE_WORLD_READABLE` dan `MODE_WORLD_WRITEABLE` tidak digunakan lagi sejak API level 17. Dimulai dari Android N, penggunaannya akan mengakibatkan `SecurityException` dikeluarkan. Ini berarti bahwa aplikasi yang menargetkan Android N dan lebih tinggi tidak dapat berbagi file privat berdasar nama, dan upaya untuk berbagi "file://" URI akan mengakibatkan `FileUriExposedException` dilontarkan. Jika aplikasi Anda perlu berbagi file privat dengan aplikasi lain, aplikasi bisa menggunakan `FileProvider` dengan `FLAG_GRANT_READ_URI_PERMISSION`.

Untuk membaca file dari penyimpanan internal:

1. Panggil `openFileInput()` dan memberikannya nama file yang akan dibaca. Proses ini mengembalikan `FileInputStream`.
2. Baca byte dari file dengan `read()`.
3. Kemudian tutup aliran dengan `close()`.



**Tip:** Jika Anda ingin menyimpan file statis dalam aplikasi Anda pada waktu kompilasi, simpan file tersebut dalam direktori `res/raw/` proyek. Anda bisa membukanya dengan `openRawResource()`, memberikan ID sumber daya `R.raw.<filename>`. Metode ini mengembalikan sebuah `InputStream` yang bisa Anda gunakan untuk membaca file (namun Anda tidak bisa menulis ke file aslinya).

### Menyimpan file cache

Jika Anda ingin membuat cache beberapa data, daripada menyimpannya secara terus-menerus, Anda sebaiknya menggunakan `getCacheDir()` untuk membuka `File` yang mewakili direktori internal tempat aplikasi Anda harus menyimpan file cache sementara. Bila perangkat hampir kehabisan ruang penyimpanan internal, Android bisa menghapus file cache ini untuk memulihkan ruang. Namun, Anda tidak boleh bergantung pada sistem untuk membersihkan file-file ini. Anda harus selalu mengelola sendiri file cache ini dan menjaga ruang yang digunakan dalam batas wajar, seperti 1MB. Bila pengguna mencopot pemasangan aplikasi Anda, file-file ini akan dibuang.

### Metode berguna yang lainnya

```
getFilesDir()
```

Mendapatkan jalur absolut ke direktori filesystem tempat file internal Anda disimpan.

```
getDir()
```

Membuat (atau membuka yang sudah ada) direktori dalam ruang penyimpanan internal.

```
deleteFile()
```

Menghapus file yang disimpan pada penyimpanan internal.

```
fileList()
```

Mengembalikan larik file yang sekarang disimpan oleh aplikasi Anda.

## b. Penyimpanan Eksternal

Setiap perangkat kompatibel-Android mendukung "penyimpanan eksternal" bersama yang bisa Anda gunakan untuk menyimpan file. Ini bisa berupa media storage lepas-pasang (seperti kartu SD) atau penyimpanan internal (tidak lepas-pasang). File yang disimpan ke penyimpanan eksternal bisa dibaca oleh semua dan dapat dimodifikasi oleh pengguna ketika mereka mengaktifkan penyimpanan massal USB untuk mentransfer file pada komputer.

Gunakan penyimpanan eksternal untuk file yang harus disimpan secara permanen, bahkan jika aplikasi Anda dicopot pemasangannya, dan tersedia bebas untuk pengguna dan aplikasi lainnya, seperti foto, gambar, atau dokumen yang dibuat oleh aplikasi.

Beberapa file privat yang tidak memiliki nilai bagi aplikasi lain juga bisa disimpan pada penyimpanan eksternal. File semacam itu bisa berupa sumber daya aplikasi tambahan yang telah diunduh, atau file media sementara. Pastikan menghapusnya bila aplikasi Anda telah dicopot pemasangannya.

- Memperoleh izin untuk penyimpanan eksternal

Untuk menulis ke penyimpanan eksternal, Anda harus meminta izin `WRITE_EXTERNAL_STORAGE` Dalam file manifest.

Hal ini secara implisit menyertakan izin untuk membaca.

```
<manifest ...>
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
...
</manifest>
```

Jika aplikasi Anda perlu membaca penyimpanan eksternal (namun bukan menulisnya), maka Anda perlu mendeklarasikan izin `READ_EXTERNAL_STORAGE`.

```
<manifest ...>
<uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" />
...
</manifest>
```

- Selalu periksa apakah penyimpanan eksternal telah dipasang

Karena penyimpanan eksternal mungkin tidak tersedia—seperti saat pengguna telah memasang storage ke PC atau telah melepas kartu SD yang menyediakan penyimpanan eksternal—Anda harus selalu memverifikasi apakah volume tersedia sebelum mengaksesnya. Anda bisa membuat kueri status penyimpanan eksternal dengan memanggil `getExternalStorageState()`. Jika keadaan yang dikembalikan sama dengan `MEDIA_MOUNTED`, maka Anda bisa membaca dan menulis file. Misalnya, metode berikut ini berguna untuk menentukan ketersediaan storage:

```
/* Checks if external storage is available for read and write
*/
public boolean isExternalStorageWritable() {
    String state = Environment.getExternalStorageState();
    if (Environment.MEDIA_MOUNTED.equals(state)) {
        return true;
    }
    return false;
}

/* Checks if external storage is available to at least read
*/
public boolean isExternalStorageReadable() {
    String state = Environment.getExternalStorageState();
    if (Environment.MEDIA_MOUNTED.equals(state) ||
        Environment.MEDIA_MOUNTED_READ_ONLY.equals(state)) {
        return true;
    }
    return false;
}
```

- **Penyimpanan eksternal privat dan publik**

Penyimpanan eksternal berstruktur sangat spesifik dan digunakan oleh sistem Android. Ada beberapa direktori publik dan direktori privat yang khusus untuk aplikasi Anda. Setiap pohon file ini memiliki tiga direktori yang diidentifikasi melalui konstanta sistem.

Misalnya, setiap file yang Anda simpan ke dalam direktori nada dering public `DIRECTORY_RINGTONES` akan tersedia untuk semua aplikasi nada dering lainnya.



Sebaliknya, setiap file yang Anda simpan dalam direktori nada dering privat DIRECTORY\_RINGTONES, secara default, hanya bisa dilihat oleh aplikasi Anda dan akan dihapus bersama aplikasi.

### Mendapatkan deskriptor file

Untuk mengakses direktori penyimpanan eksternal publik, dapatkan sebuah jalur dan buat file yang memanggil `getExternalStoragePublicDirectory()`

```
File path = Environment.getExternalStoragePublicDirectory(
Environment.DIRECTORY_PICTURES);
File file = new File(path, "DemoPicture.jpg");
```

Untuk mengakses direktori penyimpanan eksternal pribadi, dapatkan sebuah jalur dan file yang memanggil `getExternalFilesDir()`

```
File file = new File(getExternalFilesDir(null),
"DemoFile.jpg");
```

### Membuat kueri ruang storage

Jika sudah mengetahui jumlah data yang disimpan, Anda bisa mengetahui apakah tersedia ruang yang cukup tanpa menyebabkan `IOException` dengan memanggil `getFreeSpace()` atau `getTotalSpace()`. Setiap metode ini memberitahukan ruang yang tersedia saat ini dan total ruang di volume storage.

Anda tidak harus memeriksa jumlah ruang yang tersedia sebelum menyimpan file. Sebagai gantinya, Anda bisa langsung mencoba menulis file, kemudian menangkap `IOException` jika terjadi. Anda mungkin perlu melakukannya jika tidak mengetahui secara persis jumlah ruang yang diperlukan.

### Menghapus file

Anda harus selalu menghapus file yang tidak lagi diperlukan. Cara paling langsung untuk menghapus file adalah membuat referensi file yang telah dibuka memanggil `delete()` pada dirinya sendiri.

```
myFile.delete();
```

Jika file disimpan pada penyimpanan internal, Anda juga bisa meminta `Context` untuk menemukan dan menghapus file dengan memanggil `deleteFile()`:

```
myContext.deleteFile(fileName);
```

Sebagai warga yang baik, Anda juga seharusnya secara teratur menghapus file cache yang dibuat dengan `getCacheDir()`



**Perhatian:** Penyimpanan eksternal bisa menjadi tidak tersedia jika pengguna memasang penyimpanan eksternal pada komputer atau membuang media, dan tidak ada keamanan yang diberlakukan pada file yang Anda simpan pada penyimpanan eksternal. Semua aplikasi bisa membaca dan menulis file yang ditempatkan pada penyimpanan eksternal dan pengguna dapat membuangnya.

- **Menggunakan Scoped Directory Access**

Pada Android 7.0 atau yang lebih baru, jika Anda memerlukan akses ke direktori tertentu pada penyimpanan eksternal, gunakan `scoped directory access`. `Scoped directory access` menyederhanakan cara aplikasi Anda mengakses direktori penyimpanan eksternal standar, seperti direktori `Pictures`, dan menyediakan UI izin sederhana yang menjelaskan dengan detail tentang direktori yang dimintai akses oleh aplikasi. Untuk detail selengkapnya tentang `scoped directory access`.

### 6.3. Database SQLite

Menyimpan data ke database cocok untuk data terstruktur atau berulang, misalnya informasi kontak. Android menyediakan database seperti-SQL untuk keperluan ini.

SQLite pustaka perangkat lunak yang mengimplementasikan mesin database SQL yang:

- mandiri (tidak memerlukan komponen lain)
- tanpa server (tidak memerlukan server-backend)
- konfigurasi-nol (tidak perlu dikonfigurasi untuk aplikasi Anda)
- transaksional (perubahan dalam satu transaksi di SQLite akan diterapkan pada semua bagian atau tidak diterapkan sama sekali)

SQLite adalah mesin database yang paling banyak diterapkan di seluruh dunia. Kode sumber untuk SQLite berada dalam domain publik.

#### Tabel contoh :

SQLite menyimpan data dalam tabel.

Dengan anggapan berikut ini:

- Database DATABASE\_NAME
- Tabel WORD\_LIST\_TABLE
- Kolom untuk `_id`, kata, dan keterangan

Setelah menyisipkan kata "alpha" dan "beta", dalam hal ini alpha memiliki dua definisi, tabelnya akan terlihat seperti ini:

Tabel 13. Database\_Name

<b>WORD_LIST_TABLE</b>		
<b>_id</b>	<b>word</b>	<b>definition</b>
1	"alpha"	"first letter"
2	"beta"	"second letter"
3	"alpha"	"particle"

Anda bisa menemukan apa yang ada dalam baris tertentu menggunakan `_id`, atau bisa mengambil baris dengan merumuskan kueri yang memilih baris dari tabel yang menetapkan pembatas. Gunakan bahasa kueri SQL yang dibahas di bawah ini untuk membuat kueri.

#### ▪ **Transaksi**

Transaksi adalah urutan operasi yang dijalankan sebagai satu unit kerja logis. Unit kerja logis harus menunjukkan empat properti, yaitu properti

atomisitas, konsistensi, isolasi, dan ketahanan (ACID), untuk memenuhi syarat sebagai transaksi.

Semua perubahan dalam satu transaksi di SQLite diterapkan pada semua bagian atau tidak diterapkan sama sekali), bahkan jika tindakan menulis perubahan ke disk terputus karena program failure, sistem operasi failure, atau listrik mati.

Contoh transaksi:

- ⇒ Mengirimkan uang dari rekening tabungan ke rekening giro.
- ⇒ Memasukkan suatu istilah dan definisi ke dalam kamus.
- ⇒ Mengikat changelist ke cabang utama.

## ▪ ACID

- **Atomisitas.** Semua modifikasi data dijalankan, atau tidak ada yang dijalankan.
- **Konsistensi.** Bila selesai, suatu transaksi harus meninggalkan semua data dalam keadaan konsisten.
- **Isolasi.** Modifikasi yang dibuat oleh transaksi yang terjadi bersamaan harus diisolasi dari modifikasi yang dibuat oleh transaksi bersamaan lainnya. Suatu transaksi mengenali data baik dalam keadaan sebelum transaksi bersamaan lainnya memodifikasinya, maupun mengenali data setelah transaksi kedua selesai, namun tidak mengenali keadaan di antara keduanya.
- **Ketahanan.** Setelah transaksi selesai, efeknya diterapkan permanen dalam sistem. Modifikasi tetap ada bahkan seandainya listrik mati.

## ▪ Bahasa Kueri

Anda menggunakan bahasa kueri SQL khusus untuk berinteraksi dengan database. Kueri bisa sangat kompleks, namun operasi dasarnya adalah

- ⇒ menyisipkan baris
- ⇒ menghapus baris
- ⇒ memperbarui nilai dalam baris
- ⇒ mengambil baris yang memenuhi kriteria tertentu

Di Android, objek database menyediakan metode praktis untuk menyisipkan, menghapus, dan memperbarui database. Anda hanya perlu memahami SQL untuk mengambil data.

### Struktur kueri

Kueri SQL sangat terstruktur dan berisi bagian dasar berikut:

⇒ `SELECT kata, keterangan FROM WORD_LIST_TABLE WHERE kata="alpha"`

Versi generik kueri contoh:

⇒ `SELECT kolom FROM tabel WHERE kolom="value"`

Bagian:

- ⇒ **SELECT kolom**—memilih kolom untuk dikembalikan. Gunakan `*` untuk mengembalikan semua kolom.
- ⇒ **FROM tabel**—menetapkan tabel yang dipakai untuk mendapatkan hasil.
- ⇒ **WHERE**—kata kunci untuk ketentuan yang harus terpenuhi.
- ⇒ `kolom="value"`—ketentuan yang harus dipenuhi.
  - operator umum: `=, LIKE, <, >`
- ⇒ **AND, OR**—menghubungkan beberapa ketentuan dengan operator logika.
- ⇒ **ORDER BY**—mengabaikan urutan default, atau menetapkan **ASC** untuk menaik, **DESC** untuk menurun.
- ⇒ **LIMIT** adalah kata kunci yang sangat berguna jika Anda ingin mendapatkan jumlah hasil yang terbatas.

Tabel 14. Kueri contoh

1	<code>SELECT * FROM WORD_LIST TABLE</code>	Mendapatkan seluruh tabel.
2	<code>SELECT word, definition FROM WORD_LIST_TABLE WHERE id &gt; 2</code>	Mengembalikan <code>[["alpha", "particle"]]</code>
3	<code>SELECT id FROM WORD_LIST_TABLE WHERE word="alpha" AND definition LIKE "%art%"</code>	Mengembalikan id kata alpha dengan substring "art" dalam definisi. <code>[["3"]]</code>
4	<code>SELECT * FROM WORD_LIST_TABLE ORDER BY word DESC LIMIT 1</code>	Mengurutkan secara terbalik dan mengambil item pertama. Hal ini akan memberi Anda item terakhir untuk setiap urutan penyortiran. Pengurutan berdasarkan kolom pertama, dalam hal ini, <code>_id</code> . <code>[["3", "alpha", "particle"]]</code>
5	<code>SELECT * FROM WORD_LIST_TABLE LIMIT 2,1</code>	Mengembalikan 1 item mulai posisi 2. Penghitungan posisi dimulai dari 1 (bukan nol!). Mengembalikan <code>[["2", "beta", "second letter"]]</code>

- Kueri untuk Android SQLite

Anda bisa mengirimkan kueri ke database SQLite sistem Android sebagai kueri mentah atau sebagai parameter.

⇒ `rawQuery(String sql, String[] selectionArgs)` menjalankan SQL yang disediakan dan mengembalikan `Cursor` rangkaian hasil.

Tabel berikut menampilkan bagaimana dua kueri pertama dari hal di atas akan terlihat sebagai kueri mentah.

1	<code>String query = "SELECT * FROM WORD_LIST_TABLE"; rawQuery(query, null);</code>
2	<code>query = "SELECT word, definition FROM WORD_LIST_TABLE WHERE _id&gt; ? "; String[] selectionArgs = new String[]{"2"} rawQuery(query, selectionArgs) ;</code>

⇒ `query(String table, String[] columns, String selection, String[] selectionArgs, String groupBy, String having, String orderBy, String limit)` menjalankan kueri untuk tabel tertentu, yang mengembalikan `Cursor` melalui rangkaian hasil.

Inilah kueri yang menampilkan cara mengisi argumen:

```
SELECT * FROM WORD_LIST_TABLE
WHERE word="alpha"
ORDER BY word ASC
LIMIT 2,1;
```

**Mengembalikan:**

```
["alpha", "particle"]
String table = "WORD_LIST_TABLE"
String[] columns = new String[]{"*"};
String selection = "word = ?"
String[] selectionArgs = new String[]{"alpha"};
String groupBy = null;
String having = null;
String orderBy = "word ASC"
String limit = "2,1"

query(table, columns, selection, selectionArgs, groupBy,
having, orderBy, limit);
```

Dari banyak opsi storage yang telah dibahas, menggunakan database SQLite adalah salah satu yang paling serbaguna dan praktis untuk diimplementasikan.

- Database SQLite adalah solusi storage yang baik jika Anda memiliki data terstruktur yang perlu diakses dan disimpan secara persisten serta sering ditelusuri dan diubah.
- Anda bisa menggunakan database sebagai storage utama untuk data aplikasi atau pengguna, atau Anda bisa menggunakannya untuk meng-cache serta menyediakan data yang diambil dari awan.
- Jika Anda bisa menyatakan data berupa baris dan kolom, pertimbangkan database SQLite.
- Penyedia materi, yang akan diperkenalkan dalam bab berikutnya, bekerja dengan bagus pada database SQLite.

Jika Anda menggunakan database SQLite, yang dinyatakan sebagai objek `SQLiteDatabase`, semua interaksi dengan database adalah melalui instance kelas `SQLiteOpenHelper` yang akan mengeksekusi permintaan dan mengelola database untuk Anda. Aplikasi Anda hanya boleh berinteraksi dengan `SQLiteOpenHelper`, yang akan dijelaskan di bawah ini.

Ada dua tipe data yang dikaitkan secara khusus dengan penggunaan database SQLite, `Cursor` dan `ContentValues`.

## 6.4. Cursor dan Model Data

### a. Cursor

SQLiteDatabase selalu menyajikan hasil berupa Cursor dalam format tabel yang menyerupai database SQL. Anda bisa menganggap data sebagai larik baris. Cursor adalah pointer ke dalam satu baris data terstruktur. Kelas Cursor menyediakan metode untuk menggerakkan kursor melalui struktur data, dan metode untuk mendapatkan data dari bidang-bidang di setiap baris.

Kelas Cursor memiliki sejumlah subkelas yang mengimplementasikan kursor untuk tipe data tertentu.

- SQLiteCursor mengekspos hasil kueri dari sebuah SQLiteDatabase. SQLiteCursor tidak disinkronkan secara internal, sehingga kode yang menggunakan SQLiteCursor dari beberapa thread harus melakukan sinkronisasi sendiri saat menggunakan SQLiteCursor.
- MatrixCursor adalah implementasi kursor lengkap dan tidak tetap, yang didukung oleh larik objek yang secara otomatis meluaskan kapasitas internal bila diperlukan.

Beberapa operasi yang umum pada kursor adalah:

- 1) getCount() mengembalikan jumlah baris dalam kursor.
  - 2) getColumnNames() mengembalikan larik string yang berisi nama semua kolom dalam rangkaian hasil dalam urutan pencantumannya dalam hasil.
  - 3) getPosition() mengembalikan posisi kursor saat ini dalam rangkaian baris.
  - 4) Getter tersedia untuk tipe data tertentu, seperti getString(int column) dan getInt(int column).
  - 5) Operasi seperti moveToFirst() dan moveToNext() akan menggerakkan kursor.
  - 6) close() membebaskan semua sumber daya dan membuat kursor menjadi tidak valid sama sekali. Ingat untuk menutup panggilan guna membebaskan sumber daya!
- Memproses kursor

Jika panggilan metode mengembalikan ulang pada hasil, ekstrak data, lakukan sesuatu dengan data, dan terakhir, harus menutup kursor untuk membebaskan memori. Jika tidak dilakukan, aplikasi Anda bisa mogok saat kehabisan memori.

Kursor dimulai sebelum baris hasil pertama, sehingga pada pengulangan pertama gerakkan kursor ke hasil pertama jika ada. Jika kursor kosong, atau



baris terakhir sudah diproses, maka akan keluar dari loop. Jangan lupa menutup kursor bila Anda selesai menggunakannya. (Ini tidak boleh diulang terlalu sering.)

```
/ Perform a query and store the result in a Cursor
Cursor cursor = db.rawQuery(...);
try {
    while (cursor.moveToNext()) {
        // Do something with the data
    }
} finally {
    cursor.close();
}
```

Jika menggunakan database SQL, Anda bisa mengimplementasikan kelas `SQLiteOpenHelper` untuk mengembalikan kursor ke aktivitas pemanggil atau adapter, atau Anda bisa mengonversi data ke format yang lebih cocok untuk adapter. Manfaat dari yang terakhir itu adalah pengelolaan kursor (dan penutupannya) ditangani oleh helper terbuka, dan antarmuka pengguna Anda tidak tergantung pada apa yang terjadi di backend. Lihat praktik Database SQLite untuk contoh implementasi.

- **ContentValues**

Serupa dengan cara ekstra menyimpan data, instance `ContentValues` menyimpan data sebagai pasangan nilai-kunci, dalam ini kuncinya adalah nama kolom dan nilainya adalah nilai untuk sel. Satu instance `ContentValues` menyatakan satu baris tabel.

Metode `insert()` untuk database memerlukan nilai untuk mengisi baris yang diteruskan sebagai instance `ContentValues`.

```
ContentValues values = new ContentValues();
// Insert one row. Use a loop to insert multiple rows.
values.put(KEY_WORD, "Android");
values.put(KEY_DEFINITION, "Mobile operating system.");
db.insert(WORD_LIST_TABLE, null, values);
```

- **Mengimplementasikan database SQLite**

Untuk mengimplementasikan database aplikasi Android, Anda perlu melakukan yang berikut ini.

1. (Disarankan) Buat model data.
2. Jadikan `SQLiteOpenHelper` sebagai subkelas
  - ⇒ Gunakan konstanta untuk nama tabel dan kueri pembuatan database

- ⇒ Implementasikan onCreate untuk membuat SQLiteDatabase bersama tabel untuk data Anda
  - ⇒ Implementasikan onUpgrade()
  - ⇒ Implementasikan metode opsional
3. Implementasikan metode query(), insert(), delete(), update(), count() dalam SQLiteOpenHelper.
  4. Dalam MainActivity Anda, buat instance SQLiteOpenHelper.
  5. Panggil metode SQLiteOpenHelper untuk digunakan bersama database Anda.

#### Keberatan:

- Bila Anda mengimplementasikan metode, selalu masukkan operasi database ke dalam blok try/catch.
- Aplikasi contoh tidak memvalidasi data pengguna. Bila menulis aplikasi untuk dipublikasikan, selalu pastikan data pengguna sesuai harapan untuk menghindari penyuntikan data buruk atau eksekusi perintah SQL yang berbahaya ke dalam database Anda.

#### b. Model Data

Praktik yang baik adalah dengan membuat kelas yang menyatakan data Anda dengan getter dan setter.

Untuk database SQLite, instance kelas ini dapat menyatakan satu catatan, dan untuk database sederhana, satu baris dalam tabel.

```
public class WordItem {
    private int mId;
    private String mWord;
    private String mDefinition;
    // Getters and setters and more
}
```

Jadikan SQLiteOpenHelper sebagai subkelas Open helper apa pun yang Anda buat harus memperluas SQLiteOpenHelper.

```
public class WordListOpenHelper extends SQLiteOpenHelper {

    public WordListOpenHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
        Log.d(TAG, "Construct WordListOpenHelper");
    }
}
```

- Definiskan konstanta untuk nama tabel

Walaupun tidak diwajibkan, sudah biasa mendeklarasikan nama tabel, kolom, dan baris sebagai konstanta. Hal ini akan membuat kode lebih terbaca, lebih mudah mengubah nama, dan kueri akan terlihat lebih mirip SQL. Anda bisa melakukannya dalam kelas open helper, atau dalam kelas publik tersendiri; Anda akan mengetahui selengkapnya tentang hal ini dalam bab tentang penyedia materi.

```
private static final int DATABASE_VERSION = 1;
// has to be 1 first time or app will crash
private static final String WORD_LIST_TABLE =
"word_entries";
private static final String DATABASE_NAME = "wordlist";

// Column names...
public static final String KEY_ID = "_id";
public static final String KEY_WORD = "word";

// ... and a string array of columns.
private static final String[] COLUMNS = {KEY_ID, KEY_WORD};
```

- Definiskan kueri untuk membuat database

Anda memerlukan kueri yang membuat tabel untuk membuat database. Ini biasanya juga didefinisikan sebagai konstanta string. Contoh dasar ini membuat satu tabel dengan satu kolom untuk ID bertambah-otomatis dan kolom untuk menampung kata.

```
private static final String WORD_LIST_TABLE_CREATE =
    "CREATE TABLE " + WORD_LIST_TABLE + " (" +
    KEY_ID + " INTEGER PRIMARY KEY, " +
    // will auto-increment if no value passed
    KEY_WORD + " TEXT );";
```

- Implementasikan onCreate() dan buat database

Metode onCreate hanya dipanggil jika tidak ada database. Buat tabel Anda dalam metode, dan boleh menambahkan data awal.

```
@Override
public void onCreate(SQLiteDatabase db) { // Creates new
database
    db.execSQL(WORD_LIST_TABLE_CREATE); // Create the tables
    fillDatabaseWithData(db); // Add initial data
```

```
// Cannot initialize mWritableDB and mReadableDB here,
because
// this creates an infinite loop of on Create()
// being repeatedly called.
}
```

- Implementasikan `onUpgrade()`

Ini adalah metode yang diperlukan.

Jika database hanya berfungsi sebagai cache untuk data yang juga disimpan online, Anda bisa menghapus tabel dan membuat ulang setelah peningkatan versi selesai.



**Catatan:** Jika database adalah storage utama, Anda **harus** mengamankan data pengguna sebelum melakukannya karena operasi ini akan memusnahkan semua data. Lihat bab mengenai Menyimpan Data.

```
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
    // SAVE USER DATA FIRST!!!
    Log.w(WordListOpenHelper.class.getName(),
        "Upgrading database from version " +
oldVersion + " to "
        + newVersion + ", which will destroy
all old data");
    db.execSQL("DROP TABLE IF EXISTS " + WORD_LIST_TABLE);
    onCreate(db);
}
```

- Metode opsional

Kelas open helper menyediakan metode tambahan yang bisa Anda ganti bila diperlukan.

- ⇒ `onDowngrade()`—Implementasi default menolak penurunan versi.
- ⇒ `onConfigure()`—dipanggil sebelum `onCreate`. Gunakan ini hanya untuk memanggil metode yang mengonfigurasi parameter koneksi database.
- ⇒ `onOpen()`—Pekerjaan apa pun selain konfigurasi yang harus dilakukan sebelum database dibuka.

### c. Mencadangkan Database dan Aplikasi

Ada baiknya Anda mencadangkan database aplikasi.

Anda bisa melakukannya menggunakan opsi Cloud Backup yang dibahas dalam bab Opsi Storage.

#### Mengirim database bersama aplikasi Anda

Kadang-kadang Anda mungkin ingin menyertakan database yang sudah terisi bersama aplikasi. Ada sejumlah cara untuk melakukannya, dan konsekuensi dari setiap cara tersebut.

- Sertakan perintah SQL bersama aplikasi dan perintahkan untuk membuat database dan menyisipkan data pada penggunaan pertama. Inilah yang pada dasarnya akan Anda lakukan dalam praktik storage data. Jika jumlah data yang ingin dimasukkan dalam database kecil, hanya satu contoh agar pengguna bisa melihat sesuatu, Anda bisa menggunakan metode ini.
- Kirimkan data bersama APK sebagai sumber daya, dan bangun database saat pengguna membuka aplikasi untuk pertama kali. Metode ini sama dengan metode pertama, namun sebagai ganti mendefinisikan data dalam kode, Anda memasukkannya dalam sumber daya, misalnya, dalam format CSV. Selanjutnya Anda bisa membaca data dengan aliran masukan dan menambahkannya ke database.
- Bangun dan isi dahulu database SQLite lalu sertakan dalam APK. Dengan metode ini, tulis aplikasi yang akan membuat dan mengisi database. Anda bisa melakukan ini pada emulator. Selanjutnya Anda bisa menyalin file tersebut di tempat penyimpanan database sebenarnya (direktori `"/data/data/YOUR_PACKAGE/databases/"`) dan menyertakannya sebagai aset bersama aplikasi. Saat aplikasi dimulai untuk pertama kali, salin kembali file database ke dalam direktori `"/data/data/YOUR_PACKAGE/databases/"`.

Kelas `SQLiteAssetHelper`, yang bisa Anda unduh dari Github, memperluas `SQLiteOpenHelper` untuk membantu Anda melakukannya. Dan entri blog Stackoverflow akan membahas topik ini lebih detail.

Perhatikan, untuk database yang lebih besar, mengisi database harus dilakukan di latar belakang, dan aplikasi Anda tidak akan mogok jika belum ada database, atau database masih kosong.

#### d. Opsi Storage Lain

Android menyediakan opsi storage tambahan di luar cakupan kursus pengantar ini. Jika Anda ingin mendalaminya, lihatlah sumber daya di bawah ini.

- Koneksi jaringan

Anda bisa menggunakan jaringan (bila tersedia) untuk menyimpan dan mengambil data pada layanan berbasis web sendiri. Untuk melakukan operasi jaringan, gunakan kelas-kelas dalam paket berikut:

- `java.net.*`
- `android.net.*`

#### e. Mencadangkan Data Aplikasi

Pengguna seringkali menghabiskan banyak waktu serta tenaga untuk membuat data dan menyetel preferensi dalam aplikasi. Mempertahankan data itu untuk pengguna jika mereka mengganti perangkat yang rusak atau memutakhirkan ke perangkat baru merupakan bagian penting untuk memastikan pengalaman pengguna yang menyenangkan.

##### Auto Backup untuk Android 6.0 (API level 23) dan yang lebih tinggi

Untuk aplikasi dengan versi SDK target berupa Android 6.0 (API level 23) dan yang lebih tinggi, perangkat yang menjalankan Android 6.0 dan yang lebih tinggi secara otomatis membuat cadangan data aplikasi ke awan. Sistem melakukan pencadangan otomatis ini bagi hampir semua data aplikasi secara default, dan melakukannya tanpa harus menulis kode aplikasi tambahan.

Bila pengguna memasang aplikasi pada perangkat baru, atau memasang ulang aplikasi di satu perangkat (misalnya setelah dikembalikan ke setelan pabrik), sistem secara otomatis memulihkan data aplikasi dari awan. Fitur pencadangan otomatis mempertahankan data aplikasi yang dibuat pada perangkat pengguna dengan mengunggahnya ke akun Google Drive pengguna dan mengenkripsinya. Anda atau pengguna tidak akan dikenakan biaya storage data, dan data yang disimpan tidak dihitung terhadap kuota Google Drive pribadi pengguna. Setiap aplikasi bisa menyimpan hingga 25 MB. Setelah data yang dicadangkan mencapai 25 MB, aplikasi tidak akan lagi mengirim data ke awan. Jika melakukan pemulihan data, sistem akan menggunakan cuplikan data terakhir yang dikirim aplikasi ke awan.

Pencadangan otomatis terjadi bila ketentuan berikut terpenuhi:

- Perangkat sedang tidak digunakan.
- Perangkat sedang diisi dayanya.
- Perangkat terhubung ke jaringan Wi-Fi.

- Paling tidak 24 jam telah berlalu sejak pencadangan terakhir.

Anda bisa menyesuaikan dan mengonfigurasi pencadangan otomatis untuk aplikasi.

### Pencadangan untuk Android 5.1 (API level 22) dan yang lebih rendah

Untuk pengguna Android versi sebelumnya, Anda perlu menggunakan Backup API untuk mengimplementasikan pencadangan data. Singkatnya, hal ini mengharuskan Anda:

1. Mendaftar Android Backup Service untuk mendapatkan Backup Service Key.
2. Mengonfigurasi Manifes untuk menggunakan Backup Service.
3. Membuat agen pencadangan dengan memperluas kelas BackupAgentHelper.
4. Meminta pencadangan bila data berubah.
  - Firebase

Firebase adalah platform seluler yang membantu Anda mengembangkan aplikasi, menumbuhkan basis pengguna, dan menghasilkan uang lebih banyak. Firebase terdiri dari beberapa fitur pelengkap yang bisa dipadupadankan sesuai dengan kebutuhan Anda.

Beberapa fitur tersebut adalah Analytics, Perpesanan Awan, Notifikasi, dan Test Lab. Pada bab selanjutnya akan dibahas lebih lanjut materi ini.

Untuk pengelolaan data, Firebase menawarkan Realtime Database.

- Simpan dan sinkronkan data dengan database awan NoSQL.
- Aplikasi yang terhubung akan berbagi data
- Ditampung di awan
- Data disimpan sebagai JSON
- Data disimpan sebagai JSON dan disinkronkan secara real-time dengan setiap klien yang terhubung.
- Data tetap tersedia saat aplikasi Anda sedang offline



## YANG HARUS ANDA KETAHUI

Untuk praktik ini Anda harus bisa:

- 1) Membuat, membangun, dan menjalankan aplikasi di Android Studio.
- 2) Merancang layout dengan tombol dan tampilan teks.
- 3) Menggunakan Gaya dan tema.
- 4) Menyimpan dan memulihkan keadaan instance aktivitas.

## YANG AKAN ANDA BUTUHKAN

Untuk materi dan praktik ini, Anda akan membutuhkan:

- 1) Komputer Mac, Windows, atau Linux. Tool Android Studio untuk persyaratan sistem terbaru.
- 2) Akses internet atau cara alternatif untuk memuat Android Studio dan pemasangan Java terbaru ke komputer Anda.
- 3) Smartphone sebagai perangkat untuk pengujian secara nyata aplikasi yang telah dibuat.

## YANG AKAN ANDA PELAJARI

Anda akan belajar:

- 1) Mengenali yang dimaksud dengan preferensi bersama.
- 2) Membuat file preferensi bersama untuk aplikasi Anda.
- 3) Menyimpan data di preferensi bersama dan membaca preferensi tersebut kembali.
- 4) Menghapus data di preferensi bersama.

## YANG HARUS ANDA LAKUKAN

- 1) Menambahkan kemampuan untuk menyimpan, mengambil, dan menyetel ulang preferensi bersama ke aplikasi.




Aplikasi HelloSharedPrefs adalah variasi lain dari aplikasi HelloToast yang telah Anda buat di Pelajaran 1. Aplikasi ini menyertakan tombol untuk menambah angka, mengubah warna latar belakang, dan menyetel ulang angka dan warna ke defaultnya. Aplikasi juga menggunakan tema dan gaya untuk mendefinisikan tombol. Anda akan memulai dengan aplikasi awal di praktik ini dan menambahkan preferensi bersama ke kode aktivitas utama. Anda juga akan menambahkan tombol setel ulang yang akan menyetel hitungan dan warna latar belakang ke default dan menghapus file preferensi.

## Latihan 6.1. Menjelajahi HelloSharedPrefs

Proyek aplikasi awal yang lengkap untuk praktik ini tersedia di HelloSharedPrefs-Start. Dalam tugas ini Anda akan memuat proyek ke Android Studio dan menjelajahi beberapa fitur utama aplikasinya.

### Buka dan Jalankan Proyek HelloSharedPrefs

1. Unduh aplikasi HelloSharedPrefs-Start dan buka zip file.
2. Buka aplikasi di Android Studio.
3. Bangun dan jalankan proyek di Android Studio. Coba hal-hal berikut ini:
  - ⇒ Klik tombol Count untuk menambah angka di tampilan teks utama.
  - ⇒ Klik tombol warna apa pun untuk mengubah warna latar belakang tampilan teks utama.
  - ⇒ Putar perangkat dan perhatikan bahwa warna latar belakang dan hitungan dipertahankan.
  - ⇒ Klik tombol Reset untuk menyetel warna dan hitungan kembali ke defaultnya.
4. Paksa aplikasi keluar menggunakan salah satu metode berikut:
  - ⇒ Di Android Studio, pilih Run > Stop 'app' atau klik Ikon Berhenti  di toolbar.
  - ⇒ Di perangkat, klik tombol Recents (tombol kota di sudut kanan bawah). Gesek kartu agar aplikasi HelloSharedPrefs keluar atau klik X di sudut kanan. Jika Anda keluar dari aplikasi dengan cara ini, tunggu beberapa detik sebelum Anda memulainya lagi sehingga sistem bisa melakukan pembersihan.
5. Jalankan kembali aplikasi.

Aplikasi dimulai ulang dengan penampilan default -- hitungan kembali ke 0 dan warna latar belakang abu-abu.

### Jelajahi kode Aktivitas

1. Buka MainActivity  
(`java/com.example.android.simplecalc/MainActivity`).
2. Periksa kodenya dan catat yang berikut ini:
  - ⇒ Hitungan (`mCount`) didefinisikan oleh sebuah integer. Metode handler klik `countUp()` menambah nilai ini dan memperbarui `TextView` utama.
  - ⇒ Warna (`mColor`) juga merupakan integer yang awalnya didefinisikan sebagai abu-abu di file sumber daya `colors.xml` sebagai `default_background`.

- ⇒ Metode handler klik `changeBackground()` mendapatkan warna latar belakang tombol yang diklik, lalu menyetel warna latar belakang `TextView` utama.
- ⇒ Kedua integer hitungan dan warna disimpan ke bundel keadaan instance di `onSaveInstanceState()`, dan dipulihkan di `onCreate()`. Kunci bundel untuk hitungan dan warna didefinisikan oleh variabel pribadi (`COUNT_KEY`) dan (`COLOR_KEY`).

## Latihan 6.2. Menyimpan data ke file preferensi bersama

Dalam tugas ini, Anda akan menyimpan keadaan aplikasi ke file preferensi bersama dan membaca data kembali saat aplikasi dimulai ulang. Karena keadaan data yang Anda simpan di preferensi bersama (hitungan dan warna saat ini) adalah data yang **sama** yang Anda pertahankan di keadaan instance, Anda tidak harus melakukannya dua kali -- Anda bisa mengganti keadaan instance sekaligus dengan keadaan preferensi bersama.

### Inisialisasi preferensi

1. Tambahkan variabel anggota ke kelas MainActivity untuk menampung nama file preferensi bersama dan sebuah referensi ke objek SharedPreferences.

```
private SharedPreferences mPreferences;  
private String sharedPrefFile =  
    "com.example.android.hellosharedprefs";
```

Anda bisa menamai file preferensi bersama dengan nama apa pun yang Anda ingin, tetapi umumnya namanya sama dengan nama paket aplikasi Anda.

2. Di metode onCreate(), inisialisasi preferensi bersama. Pastikan Anda menyalin kode ini sebelum pernyataan if:

```
mPreferences = getSharedPreferences(sharedPrefFile,  
    MODE_PRIVATE);
```

Metode getSharedPreferences() membuka file di nama file yang diberikan (sharedPrefFile) dengan mode MODE\_PRIVATE.

**Catatan:** Versi lama Android memiliki mode lain yang memungkinkan Anda membuat file preferensi bersama yang bisa dibaca dan ditulis oleh umum. Mode ini tidak digunakan lagi di API 17, dan sekarang **sangat tidak dianjurkan** karena alasan keamanan.

### Kode Solusi (Aktivitas Utama - sebagian)

```
public class MainActivity extends AppCompatActivity {  
    private int mCount = 0;  
    private TextView mShowCount;  
    private int mCurrentColor;  
  
    private SharedPreferences mPreferences;  
    private String sharedPrefFile =  
        "com.example.android.hellosharedprefs";  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {
```

```

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mShowCount = (TextView)
        findViewById(R.id.textview);
        mCurrentColor = ContextCompat.getColor(this,
        R.color.default_background);
        mPreferences = getSharedPreferences(sharedPrefFile,
        MODE_PRIVATE);

        // ...
    }
}

```

### Simpan preferensi di onPause()

Menyimpan preferensi sangat mirip dengan menyimpan keadaan instance -- kedua operasi ini menyisihkan data ke objek Bundel sebagai pasangan kunci/nilai. Untuk preferensi bersama, Anda menyimpan data di callback daur hidup onPause() dan Anda perlu objek editor bersama (SharedPreferences.Editor) untuk menulis ke objek preferensi bersama.

1. Klik baris terakhir kelas MainActivity, tepat sebelum tanda kurung penutup.
2. Pilih Code > Generate, lalu pilih Override Methods.
3. Ketikkan "onPause", pilih tanda tangan metode untuk metode onPause() dan klik OK.

Metode onPause() kerangka ditambahkan ke titik penyisipan.

4. Dapatkan sebuah editor untuk objek SharedPreferences:

```

SharedPreferences.Editor preferencesEditor =
mPreferences.edit();

```

Editor preferensi bersama diperlukan untuk menulis ke objek preferensi bersama. Tambahkan baris ini ke onPause() setelah panggilan ke super.onPause().

5. Gunakan metode putInt() untuk menempatkan integer mCount dan mCurrentColor ke dalam preferensi bersama dengan kunci yang sesuai:

```

preferencesEditor.putInt(COUNT_KEY, mCount)
preferencesEditor.putInt(COLOR_KEY, mCurrentColor)

```

Kelas SharedPreferences.Editor menyertakan beberapa metode put untuk berbagai tipe data, termasuk putInt() dan putString().

## 6. Panggil `apply()` untuk menyimpan preferensi:

```
preferencesEditor.apply()
```

Metode `apply()` menyimpan preferensi secara asinkron, di luar thread UI. Editor preferensi bersama juga memiliki metode `commit()` untuk menyimpan preferensi secara sinkron. Metode `commit()` tidak disarankan karena bisa memblokir operasi lain.

7. Hapus seluruh metode `onSaveInstanceState()`. Karena keadaan instance aktivitas berisi data yang sama dengan preferensi bersama, Anda bisa mengganti keadaan instance sekaligus.

## Kode Solusi (MainActivity - metode `onPause()`)

```
@Override
protected void onPause() {
    super.onPause();

    SharedPreferences.Editor preferencesEditor =
mPreferences.edit();
    preferencesEditor.putInt(COUNT_KEY, mCount);
    preferencesEditor.putInt(COLOR_KEY, mColor);
    preferencesEditor.apply();
}
```

### Pulihkan preferensi di `onCreate()`

Seperti pada keadaan instance, aplikasi Anda membaca preferensi bersama apa pun yang disimpan di metode `onCreate()`. Sekali lagi, karena preferensi bersama berisi data yang sama dengan keadaan instance, kita juga bisa mengganti keadaan dengan preferensi di sini. Setiap kali `onCreate()` dipanggil - saat aplikasi dimulai, saat konfigurasi berubah -- preferensi bersama digunakan untuk memulihkan keadaan tampilan.

1. Cari bagian dari metode `onCreate()` yang menguji apakah argumen `savedInstanceState` adalah null dan mengembalikan keadaan instance:

```
if (savedInstanceState != null) {
    mCount = savedInstanceState.getInt(COUNT_KEY);
    if (mCount != 0) {
```

```
mShowCountTextView.setText(String.format("%s",  
mCount));  
}  
mColor = savedInstanceState.getInt(COLOR_KEY);  
mShowCountTextView.setBackgroundColor(mColor);  
}
```

2. Hapus seluruh blok tersebut.
3. Di metode `onCreate()`, di tempat yang sama dengan kode simpan keadaan instance, dapatkan hitungan dari preferensi dengan kunci `COUNT_KEY` dan tetapkan ke variabel `mCount`.

```
mCount = mPreferences.getInt(COUNT_KEY, 0)
```

Ketika Anda membaca data dari preferensi, Anda tidak perlu mendapatkan editor preferensi bersama. Gunakan metode `get` apa pun pada objek preferensi bersama untuk mengambil data preferensi.

Perhatikan bahwa metode `getInt()` mengambil dua argumen: satu untuk kunci dan yang lain untuk nilai default jika kunci tidak bisa ditemukan. Dalam hal ini, nilai default adalah 0, yang sama dengan nilai awal `mCount`.

4. Perbarui nilai tampilan teks utama dengan hitungan baru.

```
mShowCountTextView.setText(String.format("%s", mCount))
```

5. Dapatkan warna dari preferensi dengan kunci `COLOR_KEY` dan tetapkan ke variabel `mColor`.

```
mColor = mPreferences.getInt(COLOR_KEY, mColor)
```

Seperti sebelumnya, argumen kedua ke `getInt()` adalah nilai default yang akan dipakai dalam hal kunci tidak ada di preferensi bersama. Dalam hal ini, Anda bisa cukup menggunakan nilai `mColor`, yang baru diinisialisasi ke latar belakang default nanti di metode.

6. Perbarui warna latar belakang tampilan teks utama.

```
mShowCountTextView.setBackgroundColor(mColor)
```

7. Jalankan aplikasi. Klik tombol Count dan ubah warna latar belakang untuk memperbarui keadaan instance dan preferensi.
8. Putar perangkat atau emulator untuk memverifikasi bahwa hitungan dan warna disimpan di perubahan konfigurasi.
9. Paksa aplikasi keluar menggunakan salah satu metode berikut:
  - ⇒ Di Android Studio, pilih Run > Stop 'app'.
  - ⇒ Di perangkat, klik tombol Recents (tombol kota di sudut kanan bawah). Gesek kartu agar aplikasi HelloSharedPreferences keluar atau klik X di sudut kanan.
10. Jalankan kembali aplikasi. Aplikasi dimulai ulang dan memuat preferensi, dengan mempertahankan keadaan.

### Kode Souisi (Main Activity - onCreate())

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // Initialize views, color, preferences
    mShowCountTextView = (TextView)
    findViewById(R.id.count_textview);
    mColor = ContextCompat.getColor(this,
    R.color.default_background);
    mPreferences = getSharedPreferences(mSharedPrefFile,
    MODE_PRIVATE);

    // Restore preferences
    mCount = mPreferences.getInt(COUNT_KEY, 0);
    mShowCountTextView.setText(String.format("%s", mCount));
    mColor = mPreferences.getInt(COLOR_KEY, mColor);
    mShowCountTextView.setBackgroundColor(mColor);
}
```

#### ▪ Setel ulang preferensi di handler klik reset()

Tombol Reset di aplikasi awal menyetel ulang hitungan dan warna untuk aktivitas ke nilai defaultnya. Karena preferensi menampung keadaan aktivitas, penting juga untuk menghapus preferensi di waktu yang sama.

- 1) Di metode handler klik reset(), setelah warna dan hitungan disetel ulang, dapat editor untuk objek SharedPreferences:
  - 2) `SharedPreferences.Editor preferencesEditor = mPreferences.edit();`
  - 3) Hapus semua preferensi bersama:



- 4) `preferencesEditor.clear();`
- 5) Terapkan perubahan:
- 6) `preferencesEditor.apply();`

### **Tantangan:**

Ubah aplikasi `HelloSharedPrefs` sehingga, sebagai ganti otomatis menyimpan keadaan ke file preferensi, menambahkan aktivitas kedua untuk mengubah, menyetel ulang, dan menyimpan preferensi tersebut. Tambahkan sebuah tombol ke aplikasi yang bernama `Settings` untuk meluncurkan aktivitas tersebut. Sertakan tombol `toggle` dan `spinner` untuk memodifikasi preferensi dan tombol `Save` dan `Reset` untuk menyimpan dan membersihkan preferensi

- Cara Menyimpan Data Menggunakan SharedPreferences

- 1) Buka aplikasi Android Studio kalian, Buat Project Baru.
- 2) Buka layout *activity\_main.xml* kalian, disini kita akan menambahkan Edittext (Untuk Input), TextView (Untuk Output Data) dan Button (Untuk mengeksekusi program).

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="8dp"
    android:orientation="vertical"
    android:background="#FFF5F5F5"

tools:context="android.fkti.unmul.com.basicandroid.MainAct
ivity">

    <EditText
        android:id="@+id/input"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Masukan Data"
        android:inputType="textPersonName" />

    <Button
        android:id="@+id/save"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Simpan" />

    <TextView
        android:id="@+id/output"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:gravity="center"
        android:text="Output Data :"
        android:textStyle="bold" />
</LinearLayout>
```

- 3) Selanjutnya buka file *MainActivity.java*, pada source code berikut ini kita akan menambahkan *SharedPreferences*, serta beberapa fungsi/methodnya.

```
package android.fkti.unmul.com.basicandroid;

import android.content.Context;
import android.content.SharedPreferences;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity{

    //Deklarasi Variable Pendukung
    private TextView Hasil;
    private EditText Masukan;
    private Button Eksekusi;

    //Deklarasi dan Inisialisasi SharedPreferences
    private SharedPreferences preferences;

    //Digunakan Untuk Konfigurasi SharedPreferences
    private SharedPreferences.Editor editor;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Masukan = findViewById(R.id.input);
        Hasil = findViewById(R.id.output);
        Eksekusi = findViewById(R.id.save);

        //Membuat File Baru Beserta Modifiernya
        preferences =
        getSharedPreferences("Belajar_SharedPreferences",
        Context.MODE_PRIVATE);

        Eksekusi.setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View view) {
                getData();
            }
        });
    }
}
```

```

Toast.makeText(getApplicationContext(),"Data
Tersimpan", Toast.LENGTH_SHORT).show();
    }
});
}

private void getData(){
    //Mendapatkan Input dari user
    String getKonten =
Masukan.getText().toString();

    //Digunakan Untuk Pengaturan Konfigurasi
SharedPreferences
    editor = preferences.edit();

    //Memasukan Data Pada Editor SharedPreferences
dengan key (data_saya)
    editor.putString("data_saya", getKonten);

    //Menjalankan Operasi
    editor.apply();

    //Menampilkan Output
    Hasil.setText("Output Data :
"+preferences.getString("data_saya", null));
}
}

```

### Penjelasan:

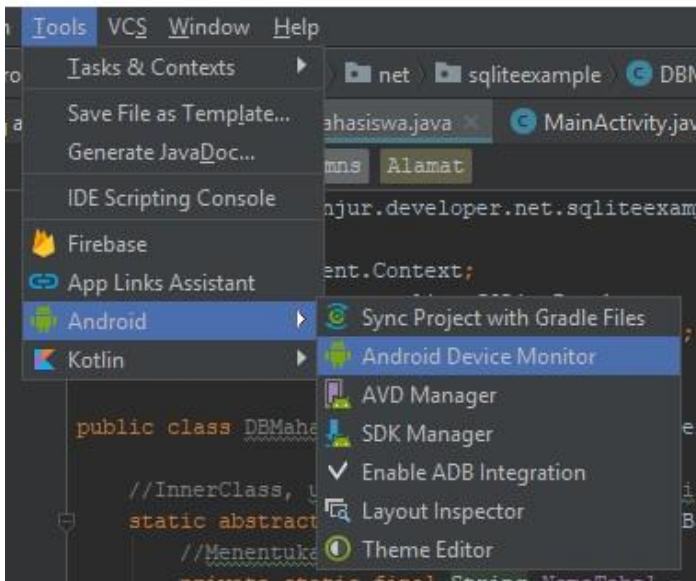
Pertama kita mendeklarasikan beberapa variable pendukung, seperti `TextView`, `Edittext` dan `Button`, lalu kita juga Menginisialisasi `SharedPreferences` dan juga `SharedPreferences,Editor`, yang digunakan konfigurasi data pada `SharedPreferences`.

Fungsi `getSharedPreferences()` gunakan jika kita ingin membuat file `Shared Preference` yang baru dengan parameter yang spesifik dan juga nama filenya spesifik dan gunakan `Context` untuk mengakses file pada aplikasi.

Didalam method `getSharedPreferences()` file yang akan kita buat mempunyai mode `private`, yang artinya, data pada file tersebut hanya untuk aplikasi itu saja.

Untuk memasukan data, kita dapat menggunakan fungsi `putString()`; dengan parameter `key` dan nilai. Dan untuk menampilkan output, kita dapat menggunakan fungsi `getString()`, kita dapat menampilkan data tersebut dengan memanggil keynya saya.

Untuk melihat hasil dari SharedPreferences, caranya jalankan project tersebut menggunakan Emulator, lalu buka *Android Device Emulator*, seperti berikut ini:



Jangan lupa untuk mencentang *Enable ADB Interation*. Selanjutnya buka direktori *data > data > (Nama Package pada Aplikasi Kita) > shared\_prefs*.

### Latihan 6.3. Penyimpanan Internal:

- a. Buka aplikasi Android Studio kalian, Buat Project Baru.
- b. Disini kita akan mendesain layout pada `activity_main.xml`, pada layout tersebut kita akan menambahkan sebuah widget atau view seperti `EditText` dan `Button`.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_margin="8dp"
android:orientation="vertical"
android:background="#FFF5F5F5"

tools:context="android.fkti.unmul.com.basicandroid.Ma
inActivity">

    <EditText
        android:id="@+id/input_data"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Masukan Data"
        android:inputType="textPersonName" />

    <Button
        android:id="@+id/save_internal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Simpan ke Internal" />

    <Button
        android:id="@+id/save_external"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Simpan ke External" />

</LinearLayout>
```

- c. Selanjutnya buka file *MainActivity.java* pada project kalian, pertama kita akan memasukan source code untuk Internal Storage.

```
package android.fkti.unmul.com.basicandroid;

import android.content.Context;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import java.io.FileOutputStream;

public class MainActivity extends AppCompatActivity{

    private Button Internal, External;
    private EditText InputData;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        InputData = findViewById(R.id.input_data);
        Internal = findViewById(R.id.save_internal);
        External = findViewById(R.id.save_external);

        Internal.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String setData =
InputData.getText().toString();
                //Digunakan untuk membuat dan menulis
File/Data pada Penyimpanan
                FileOutputStream fileOutputStream;
                try {
                    //Membuat Berkas Baru dengan mode
Private
                    fileOutputStream =
openFileOutput("DataSaya", Context.MODE_PRIVATE);

                    //Menulis Data Baru dan
Mengkonversinya kedalam bentuk byte
                    fileOutputStream.write(setData.getBytes());

                    //Menutup FileOutputStream
                    fileOutputStream.close();
```

```

Toast.makeText(getApplicationContext(), "Data Disimpan di
Internal", Toast.LENGTH_SHORT).show();
        }catch (Exception ex){
            ex.printStackTrace();
        }
    }
});
}
}

```

### **Penjelasan:**

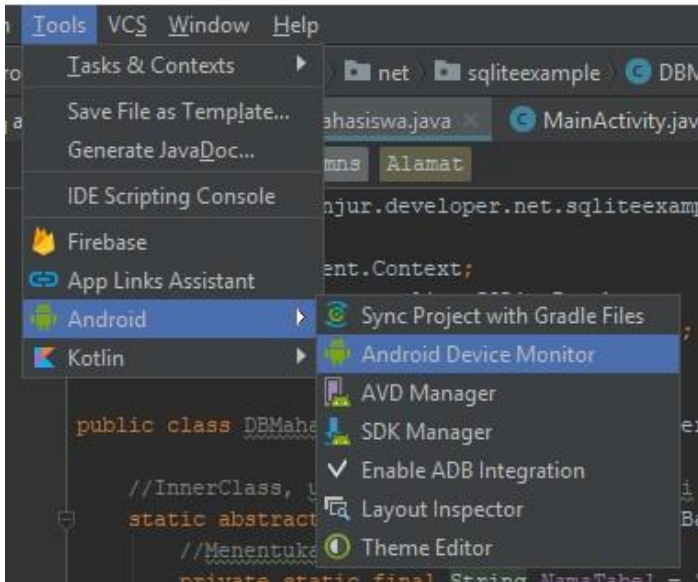
FileOutputStream digunakan untuk membuat dan menulis File atau Data baru pada penyimpanan, dengan menggunakan fungsi `openFileOutput`, pada fungsi tersebut kita membuat berkas baru dengan nama "*DataSaya*" yang mempunyai akses Private, yang artinya file tersebut hanya bisa diakses oleh aplikasi itu sendiri.

Untuk menambahkan data/nilai pada berkas "*DataSaya*", kita dapat menggunakan fungsi `write()`, lalu kita harus mengkonversinya kedalam bentuk Byte, menggunakan fungsi `getBytes()`.

Terakhir, kita menutup `FileOutputStream` menggunakan fungsi `close()`.

Untuk melihat datanya, caranya jalankan project tersebut menggunakan Emulator, lalu buka Android Device Emulator, seperti berikut ini:





Jangan lupa untuk mencentang Enable ADB Interation. Selanjutnya buka direktori *data > data > (Nama Package pada Aplikasi Kita) > files*.

## Latihan 6.4. Database SQLite

- 1) Masukan kedua permission berikut ini, digunakan untuk memperoleh izin, agar aplikasi dapat menulis dan membaca file External Storage. Buka file *androidManifest* (pada direktori main).

```
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

- 2) Buka MainActivity.java kalian, tambahkan source code berikut ini pada method *onCreate()*.

```
External.setOnClickListener(new
View.OnClickListener() {
@Override
    public void onClick(View view) {
        //Digunakan Untuk Mengecek Apakah
Tersebut Penyimpanan External atau Tidak
        String state =
Environment.getExternalStorageState();
        if
(Environment.MEDIA_MOUNTED.equals(state)){
            //Digunakan Untuk Mengecek Lokasi
Penyimpanan Direktori di Memory External
            File dirExternal =
Environment.getExternalStorageDirectory();

            //Membuat Direktori Baru dengan
Nama "ContohDirektori"
            File createDir = new
File(dirExternal.getAbsolutePath()+"/ContohDirektori"
);

            //Jika Direktori "ContohDirektori"
Tidak Ada, Maka akan Membuatnya
            if (!createDir.exists()){
                createDir.mkdir();
                //Membuat File Baru didalam
Direktori "ContohDirektori"
                File file = new File(createDir,
"ContohFile.txt");
                String setData =
InputData.getText().toString();
```

```

        //Digunakan untuk membuat dan
menulis File/Data di Internal
        FileOutputStream
fileOutputStream;
        try {
            //Membuat Berkas Baru
dengan mode Private
            fileOutputStream = new
FileOutputStream(file);
fileOutputStream.write(setData.getBytes());

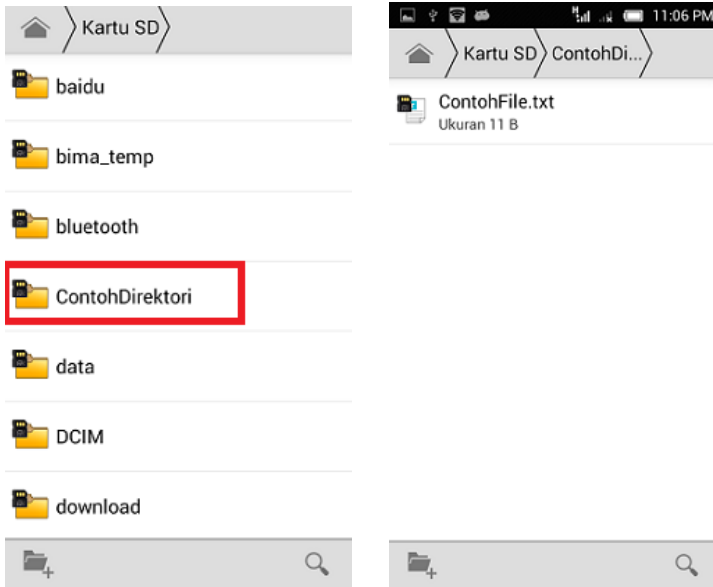
            fileOutputStream.close();

Toast.makeText(getApplicationContext(), "Data
Disimpan di External", Toast.LENGTH_SHORT).show();
        }catch (IOException ex){
            ex.printStackTrace();}
    }
        }else {

Toast.makeText(getApplicationContext(), "Penyimpanan
External Tidak Tersedia", Toast.LENGTH_SHORT).show();
        }
    }
});

```

3) Jalankan project tersebut, coba kalian inputkan nilai/data pada aplikasi tersebut, lalu simpan pada penyimpanan External, maka akan terlihat Direktori Baru pada Penyimpanan External kalian, seperti berikut ini.



Untuk membuat file .txt tersebut kalian memerlukan aplikasi editor.

Dalam Praktik ini, Anda telah belajar :

- 1) Kelas `SharedPreferences` memungkinkan aplikasi untuk menyimpan sejumlah kecil data primitif sebagai pasangan kunci-nilai.
- 2) Preferensi bersama bersifat persisten di semua sesi pengguna yang berbeda di aplikasi yang sama.
- 3) Untuk menulis ke preferensi bersama, dapatkan objek `SharedPreferences.Editor`.
- 4) Gunakan berbagai metode `put*` di objek `SharedPreferences.Editor`, seperti `putInt()` atau `putString()`, untuk menempatkan data ke dalam preferensi bersama dengan kunci dan nilai.
- 5) Gunakan berbagai metode `get*` di objek `SharedPreferences`, seperti `getInt()` atau `getString()`, untuk mendapatkan data dari preferensi bersama dengan sebuah kunci.
- 6) Gunakan metode `clear()` di objek `SharedPreferences.Editor` untuk membuang semua data yang disimpan di preferensi.
- 7) Gunakan metode `apply()` di objek `SharedPreferences.Editor` untuk menyimpan perubahan ke file preferensi.

## REFERENSI

- Buku Kursus Dasar-Dasar Developer Android – Konsep, Belajar mengembangkan aplikasi Android, Dikembangkan oleh Tim Pelatihan Developer Google.
- Buku Kerja Praktik, Kursus Dasar-Dasar Developer Android — Praktik, Belajar mengembangkan aplikasi Android, Dikembangkan oleh Tim Pelatihan Developer Google
- Safaat N.H. 2014. Android: Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android. Bandung: Informatika.



## MATERI PEMBELAJARAN

Materi bab ini akan membahas:

- 1) Database SQLite
- 2) SQLite room Database
- 3) Set up Firebase realtime database
- 4) Operasi database: create, update, insert, delete (CRUID)

## YANG HARUS SUDAH ANDA KETAHUI

Untuk materi dan praktik di bab ini Anda harus bisa:

- 1) Membuat, membangun, dan menjalankan aplikasi di Android Studio.
- 2) Anda sudah familier dengan database secara umum, terutama database SQL, dan bahasa SQL yang digunakan untuk berinteraksi dengannya.
- 3) Membuat menu opsi di bilah aplikasi.
- 4) Menambahkan dan mengedit item menu dan handler kejadian untuk klik item menu.

## YANG AKAN ANDA BUTUHKAN

Untuk materi dan praktik ini, Anda akan membutuhkan:

- 1) Komputer Mac, Windows, atau Linux. Tool Android Studio untuk persyaratan sistem terbaru.
- 2) Smartphone sebagai perangkat untuk pengujian secara nyata aplikasi yang telah dibuat.

## YANG AKAN ANDA PELAJARI

Anda akan belajar:

- 1) Membuat, mengelola, dan berinteraksi dengan database SQLite.
- 2) Membuat Fungsi Create, Read, Update, Delete
- 3) Mengimplementasikan database SQLite dan operasi database
- 4) Set Up Firebase Realtime Database.



# BAB 7 APLIKASI CREATE, UPDATE, INSERT, DELETE (CRUID) DATABASE



## Bab 7: Aplikasi CRUD Database

### 7.1. Aplikasi Database SQLite

Bab ini membahas operasi database create, update, insert, dan delete menggunakan SQLiteDatabase yang merupakan kerangka kerja Android dan kelas SQLiteOpenHelper. Bab ini bukanlah pengantar SQLite atau database SQL. Bab ini beranggapan bahwa Anda sudah familier dengan database SQL secara umum, dan pembangunan kueri SQL dasar. Dari banyak opsi storage yang telah dibahas, menggunakan database SQLite adalah salah satu yang paling serbaguna dan praktis untuk diimplementasikan.

Database SQLite atau bisa juga disebut SQLite adalah:

- a. Perpustakaan dalam proses yang mengimplementasikan mandiri, serverless, nol-konfigurasi, transaksional mesin database SQL. Kode untuk SQLite adalah dalam domain publik dan dengan demikian bebas untuk digunakan untuk tujuan apapun, komersial atau pribadi. Database SQLite adalah yang paling banyak digunakan di dunia dengan aplikasi yang lebih dari yang kita bisa hitung, termasuk beberapa proyek profil tinggi.
- b. SQLite adalah mesin database SQL tertanam. Tidak seperti kebanyakan database SQL lainnya, SQLite tidak memiliki proses server terpisah. SQLite membaca dan menulis langsung ke file disk biasa. Sebuah database SQL lengkap dengan beberapa tabel, indeks, pemicu, dan pandangan, yang terkandung dalam sebuah file disk tunggal. Database format file adalah cross-platform - Anda dapat dengan bebas menyalin database antara 32-bit dan 64-bit sistem atau antara big-endian dan little-endian arsitektur. Fitur-fitur ini membuat SQLite menjadi pilihan populer

sebagai Aplikasi File Format. Pikirkan SQLite bukan sebagai pengganti Oracle tetapi sebagai pengganti fopen ().

- c. Database SQLite adalah solusi penyimpanan yang baik jika anda memiliki data terstruktur yang perlu diakses dan disimpan secara persisten serta sering ditelusuri dan diubah. Anda juga bisa menggunakan SQLite sebagai media penyimpanan utama untuk data aplikasi atau pengguna, atau anda juga bisa menggunakannya untuk proses *caching* serta menyediakan data yang diambil dari *cloud*.

Setiap aplikasi android yang kita rancang memiliki directory sendiri pada aplikasi, dengan mendapatkan Chace data pada sistem operasi android nantinya, pada perancangan SQLite ini selalu disimpan di dalam directory data/data/< name Project > /database/helper. Rancangan ini hampir sama dengan kita menggunakan SQLman pada sistem operasi Ubuntu server umumnya. Penyimpanan semua file database pada SQLite ini ada pada SQLiteOpenHelper.

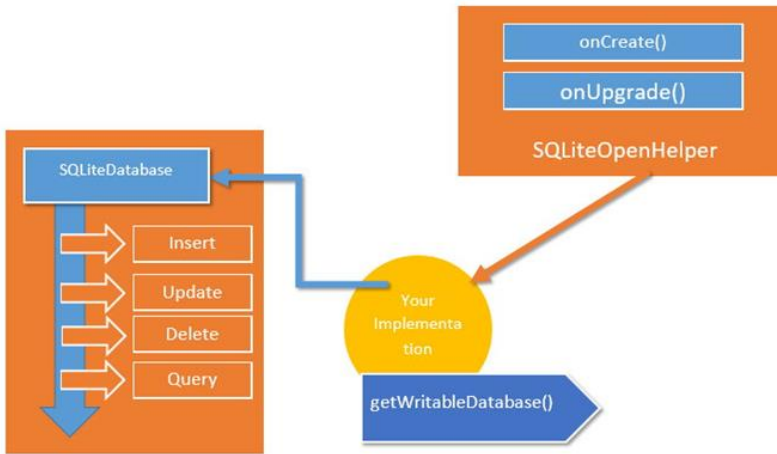
## 7.2. SQLiteOpenHelper

SQLiteOpenHelper adalah cara untuk membuat dan dapat meng-Upgrade database pada veris android selanjutnya, pada Aplikasi android juga menggunakan subclass dari class SQLiteOpenHelper tersebut. pada SQLiteOpenHelper ini kita diminta untuk dapat memahami metode onCreate(), onUpdate() untuk dapat memaksimalkan upgrade selanjutnya. dan ini keterangan dari dua metode ini :

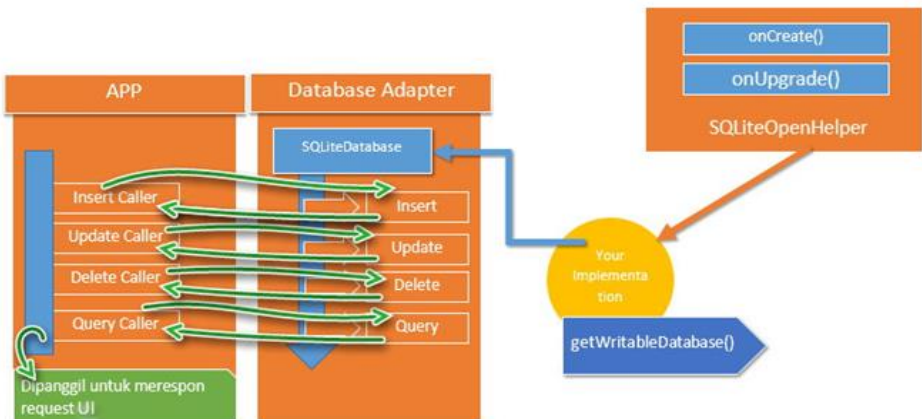
- 1) onCreate(), dipanggil ketika database dapat diakses namun belum dibuat.
- 2) onUpdate(), dipanggil ketika aplikasi diupgrade dan nomor versi telah berubah.

Pada coding android metode ini anak memungkinkan perubahan data, penambahan serta pengembangan selanjutnya, sehingga sistem dapat memperbaharui data ( updat ) skema suatu database atau dapat menghapus database dan membuat tabel databes kembali, disinilah metodr onCreate digunakan.class SQLiteOpenHelper menyediakan method getReadableDatabase () dan getWritableDatabase (), untuk mendapatkan akses ke objek SQLiteDatabase.

Berikut Rancangan dan cara kerja Databas SQLite



Gambar 103. Alur Databas SQLite



Gambar 104. Cara Kerja Database SQLite

### 7.3. Mengimplementasikan Database SQLite

Untuk mengimplementasikan SQLite, anda perlu melakukan ini :

1. Buat model data.
2. Jadikan SQLiteOpenHelper sebagai subkelas
  - Gunakan konstanta untuk nama tabel dan query pembuatan database.

- Implementasikan **onCreate()** untuk membuat SQLiteDatabase bersama tabel untuk data anda.
  - Implementasikan **onUpgrade()**.
  - Implementasikan metode **opsional**.
3. Implementasikan metode **query(), insert(), delete(), update(), count()** dalam SQLiteOpenHelper.
  4. Dalam MainActivity anda, buat *instance* SQLiteOpenHelper.
  5. Panggil metode SQLiteOpenHelper untuk digunakan bersama database anda.

Objection:

- Bila anda mengimplementasikan metode, selalu masukan operasi database ke dalam blok *try/catch*.
- Aplikasi contoh tidak memvalidasi data pengguna. Bila anda menulis aplikasi untuk dipublikasikan, selalu pastikan data pengguna sesuai harapan untuk menghindari penyuntikan data buruk atau eksekusi perintah SQL yang berbahaya ke dalam database Anda.

- Pembuatan Model Data

Praktik yang baik adalah dengan membuat kelas yang menyatakan data Anda dengan *getter* dan *setter*. Untuk database SQLite, *instance* kelas ini dapat menyatakan satu catatan, dan untuk database sederhana, satu baris dalam tabel.

```
public class WordItem {
    private int mId;
    private String mWord;
    private String mDefinition;
    // Getters and setters and more
}
```

Jadikan SQLiteOpenHelper sebagai subkelas

Open helper apa pun yang Anda buat harus meng-*extends* SQLiteOpenHelper.

```
public class WordListOpenHelper extends SQLiteOpenHelper {
    public WordListOpenHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
        Log.d(TAG, "Construct WordListOpenHelper");
    }
}
```

### Definisikan konstanta untuk nama tabel

Walaupun tidak diwajibkan, biasakan untuk mendeklarasikan nama tabel, kolom, dan baris sebagai konstanta. Hal ini akan membuat kode lebih terbaca, lebih mudah saat mengubah nama, dan *query* akan terlihat lebih mirip dengan SQL. Anda bisa melakukannya dalam kelas open helper, atau dalam kelas publik tersendiri.

```
private static final int DATABASE_VERSION = 1;
// has to be 1 first time or app will crash
private static final String WORD_LIST_TABLE = "word_entries";
private static final String DATABASE_NAME = "wordlist";
// Column names...
public static final String KEY_ID = "_id";
public static final String KEY_WORD = "word";
// ... and a string array of columns.
private static final String[] COLUMNS = {KEY_ID, KEY_WORD};
```

### Definisikan query untuk membuat database

Anda memerlukan *query* yang membuat tabel untuk membuat database. Ini biasanya juga didefinisikan sebagai konstanta string. Contoh dasar ini membuat satu tabel dengan satu kolom untuk ID bertambah-otomatis/auto increment dan kolom untuk menampung kata.

```
private static final String WORD_LIST_TABLE_CREATE =
"CREATE TABLE " + WORD_LIST_TABLE + " (" +
KEY_ID + " INTEGER PRIMARY KEY, " +
// will auto-increment if no value passed
KEY_WORD + " TEXT );";
```

### Implementasikan onCreate() dan buat database

Metode **onCreate()** hanya dipanggil jika tidak ada database. Buat tabel Anda dalam metode, dan boleh menambahkan data awal.

```
@Override
public void onCreate(SQLiteDatabase db) { // Creates new
database
db.execSQL(WORD_LIST_TABLE_CREATE); // Create the tables
fillDatabaseWithData(db); // Add initial data
// Cannot initialize mWritableDatabase and mReadableDB here, because
// this creates an infinite loop of on Create()
// being repeatedly called.
}
```

### Implementasikan onUpgrade()

Ini adalah metode yang diperlukan. Jika database hanya berfungsi sebagai cache untuk data yang juga disimpan online, Anda bisa menghapus tabel dan membuat ulang setelah peningkatan versi selesai.



**Catatan:** Jika database adalah *storage* utama, Anda harus mengamankan data pengguna sebelum melakukannya karena operasi ini akan memusnahkan semua data

```
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
// SAVE USER DATA FIRST!!!
Log.w(WordListOpenHelper.class.getName(),
"Upgrading database from version " + oldVersion + " to "
+ newVersion + ", which will destroy all old data");
db.execSQL("DROP TABLE IF EXISTS " + WORD_LIST_TABLE);
onCreate(db);
}
```

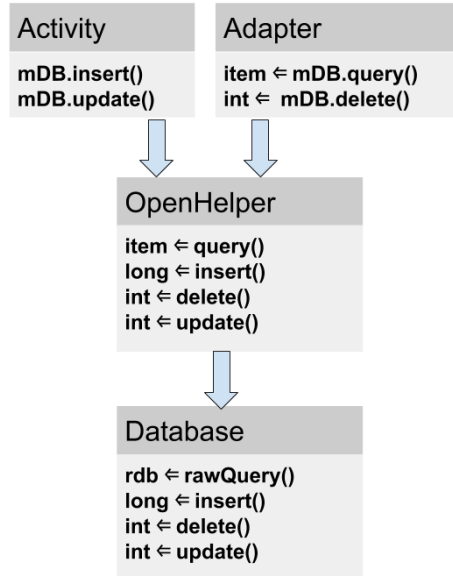
#### ▪ *Metode opsional*

Kelas open helper menyediakan metode tambahan yang bisa Anda ganti bila diperlukan.

- 1) **onDowngrade()**, Implementasi default menolak penurunan versi.
- 2) **onConfigure()**, dipanggil sebelum onCreate. Gunakan ini hanya untuk memanggil metode yang mengonfigurasi parameter koneksi database.
- 3) **onOpen()**, pekerjaan apa pun selain konfigurasi yang harus dilakukan sebelum database dibuka.

## 7.4. Operasi Database

Walaupun bisa memanggil metode dalam open helper yang diinginkan dan mengembalikan yang Anda pilih ke aktivitas pemanggil, lebih baik lanjutkan dengan metode `query()`, `insert()`, `delete()`, `update()`, `count()` standar yang sesuai dengan API database dan penyedia materi. Menggunakan format ini akan mempermudah penambahan penyedia materi atau pemuat di masa mendatang, dan memudahkan orang lain dalam memahami kode Anda. Diagram berikut menampilkan cara mendesain API yang berbeda agar konsisten dan jelas.



Gambar 105. Operasi Database

- **query()**

Metode kueri yang diimplementasikan dalam kelas open helper Anda bisa mengambil dan mengembalikan tipe data apa pun yang diperlukan antarmuka pengguna. Karena open helper menyediakan metode praktir untuk menyisipkan, menghapus, dan memperbarui baris, metode kueri Anda tidak perlu generik dan mendukung operasi ini.

Secara umum, metode kueri Anda hanya boleh mengizinkan kueri yang diperlukan oleh aplikasi dan bukan untuk keperluan umum. Database menyediakan dua metode untuk mengirimkan kueri: `SQLiteDatabase.rawQuery()` dan `SQLiteDatabase.query()`, bersama sejumlah opsi untuk argumen.

## SQLiteDatabase.rawQuery().rawQuery()

Metode kueri open helper bisa membentuk kueri SQL dan mengirimkannya sebagai rawQuery ke database yang mengembalikan kursor. Jika data disediakan oleh aplikasi dan dikontrol penuh, Anda bisa menggunakan rawQuery()

```
rawQuery(String sql, String[] selectionArgs)
```

- Parameter pertama untuk db.rawQuery() adalah string kueri SQLite.
- Parameter kedua berisi argumen.

```
cursor = mReadableDB.rawQuery(queryString,  
selectionArgs);
```

## SQLiteDatabase.query()

Jika Anda memproses data yang disediakan pengguna, bahkan setelah validasi, lebih aman membentuk kueri dan menggunakan versi metode SQLiteDatabase.query() untuk database. Argumen adalah apa yang Anda harapkan dalam SQL dan didokumentasikan dalam dokumentasi SQLiteDatabase.

```
Cursor query (boolean distinct, String table, String[]  
columns, String selection, String[] selectionArgs, String  
groupBy, String having,  
String orderBy, String limit)
```

Inilah contoh dasarnya:

```
String[] columns = new String[]{KEY_WORD};  
String where = KEY_WORD + "LIKE ?";  
searchString = "%" + searchString + "%";  
String[] whereArgs = new String[]{searchString};  
cursor = mReadableDB.query(WORD_LIST_TABLE, columns, where,  
whereArgs, null, null, null);
```

## Contoh lengkap open helper query()

```
public WordItem query(int position) {  
    String query = "SELECT * FROM " + WORD_LIST_TABLE +  
        " ORDER BY " + KEY_WORD + " ASC " + "LIMIT " +  
    position + ",1";  
  
    Cursor cursor = null;  
    WordItem entry = new WordItem();
```



```

try {
    if (mReadableDB == null) {mReadableDB =
getReadableDatabase();}
    cursor = mReadableDB.rawQuery(query, null);
    cursor.moveToFirst();

entry.setId(cursor.getInt(cursor.getColumnIndex(KEY_ID)));

entry.setWord(cursor.getString(cursor.getColumnIndex(KEY_WORD)
));
} catch (Exception e) {
    Log.d(TAG, "EXCEPTION! " + e);
} finally {
    // Must close cursor and db now that we are done with
it.
    cursor.close();
    return entry; }
}

```

## ▪ insert()

Metode insert() open helper memanggil SQLiteDatabase.insert(), yaitu metode SQLiteDatabase praktis yang digunakan untuk menyisipkan baris ke dalam database. (Metode ini praktis, karena Anda tidak perlu menulis kueri SQL sendiri.)

### Format

```

long insert(String table, String nullColumnHack,
ContentValues values)

```

- Argumen pertama adalah nama tabel.
- Argumen kedua adalah sebuah String nullColumnHack. Ini adalah solusi yang memungkinkan Anda untuk menyisipkan baris kosong. Gunakan null.
- Argumen ketiga harus berupa kontainer [ContentValues] bersama nilai-nilai untuk mengisi baris. Contoh ini hanya memiliki satu kolom; untuk tabel dengan banyak kolom, tambahkan
- Metode database mengembalikan ID item yang baru disisipkan, dan Anda harus meneruskannya ke aplikasi.

### Contoh

```

newId = mWritableDatabase.insert(WORD_LIST_TABLE, null,
values);

```

## ▪ delete()

Metode delete dari open helper memanggil metode delete() database, yang praktis digunakan sehingga Anda tidak perlu menulis kueri SQL seluruhnya.

### Format

```
int delete (String table, String whereClause, String[] whereArgs)
```

- ⇒ Argumen pertama adalah nama tabel.
- ⇒ Argumen kedua adalah sebuah klausa WHERE.
- ⇒ Argumen ketiga adalah argumen untuk klausa WHERE.

Anda bisa menghapus menggunakan kriteria apa pun, dan metode akan mengembalikan jumlah item yang sebenarnya dihapus, yang juga harus dikembalikan oleh open helper.

### Contoh

```
deleted = mWritableDatabase.delete(WORD_LIST_TABLE,
    KEY_ID + " =? ", new
String[] {String.valueOf(id)});
```

## ▪ update()

Metode update dari open helper memanggil metode update() database, yang praktis digunakan sehingga Anda tidak perlu menulis kueri SQL seluruhnya. Argumen tersebut sudah familier dari metode sebelumnya, dan onUpdate mengembalikan jumlah baris yang diperbarui.

### Format

```
int update(String table, ContentValues values,
    String whereClause, String[] whereArgs)
```

- Argumen pertama adalah nama tabel.
- Argumen kedua harus berupa ContentValues bersama nilai-nilai baru untuk baris tersebut.
- Argumen ketiga adalah klausa WHERE.
- Argumen keempat adalah argumen untuk klausa WHERE.

### Contoh

```
ContentValues values = new ContentValues();
values.put(KEY_WORD, word);
```

```
mNumberOfRowsUpdated = mWritableDatabase.update(WORD_LIST_TABLE,
values, // new values to insert
KEY_ID + " = ?",
new String[]{String.valueOf(id)});
count()
```

Metode `count()` mengembalikan jumlah entri dalam database. Jika menggunakan `RecyclerView.Adapter`, Anda harus mengimplementasikan `getItemCount()`, yang perlu mendapatkan sejumlah baris dari open helper, yang perlu didapat dari database.

### Di adapter

```
@Override
public int getItemCount() {
    return (int) mDB.count();
}
```

### Di open helper

```
public long count() {
    if (mReadableDB == null) {mReadableDB =
getReadableDatabase();}
    return DatabaseUtils.queryNumEntries(mReadableDB,
WORD_LIST_TABLE);
}
```

`queryNumEntries()` adalah metode di kelas `DatabaseUtils` publik, yang menyediakan banyak metode praktis untuk digunakan bersama kursor, database, dan juga penyedia materi.

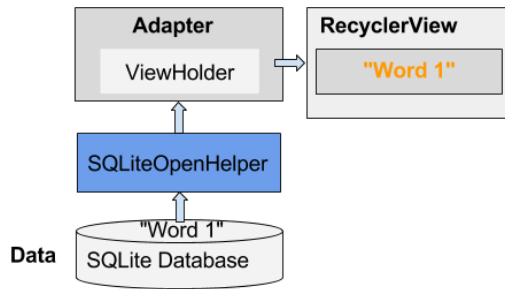
### Buat Instance Open Helper

Untuk menangani database, dalam `MainActivity`, di `onCreate`, panggil:

```
mDB = new WordListOpenHelper(this)
```

### ▪ **Bekerja dengan database**

Mengombinasikan backend `SQLiteDatabase` dengan `RecyclerView` untuk menampilkan data merupakan pola yang umum.



Gambar 106. SQLiteDatabase

Misalnya:

- Menekan FAB bisa memulai aktivitas yang akan mendapatkan masukan dari pengguna dan menyimpannya ke dalam database sebagai item baru atau diperbarui.
- Menggesek suatu item bisa menghapusnya setelah pengguna mengkonfirmasi penghapusan.

### Transaksi

Gunakan transaksi

- ⇒ Saat melakukan beberapa operasi yang semuanya harus diselesaikan agar database konsisten, misalnya, memperbarui penetapan harga item terkait untuk kejadian penjualan.
- ⇒ untuk batch beberapa operasi independen guna meningkatkan kinerja, seperti penyisipan massal.

Transaksi bisa disarangkan, dan kelas SQLiteDatabase menyediakan metode tambahan untuk mengelola transaksi tersarang.

### Idiom transaksi

```

db.beginTransaction();
try {
    ...
    db.setTransactionSuccessful();
} finally {
    db.endTransaction();
}
  
```



## Latihan 7.1. Aplikasi Pendataan data Mahasiswa

Sebelum membuat database tersebut, pertama kita perlu menyiapkan Struktur tabelnya terlebih dahulu, disini kita harus menentukan data apa saja yang harus di Inputkan. Dan berikut ini merupakan Struktur Table yang akan kita terapkan pada SQLite Database:

Field	Tipe Data	key
NIM	Text	Primary Key
Nama_Mahasiswa	Text	Not Null
Jurusan	Text	Not Null
Jenis_Kelamin	Text	Not Null
Tanggal_Lahir	Text	Not Null
Alamat	Text	Not Null

### Membuat Database pada SQLite

- ⇒ Buka Aplikasi Android Studio kalian, Buat Project Baru.
- ⇒ Pertama kita perlu membuat class SQLiteOpenHelper terlebih dahulu, Class ini digunakan untuk mengatur atau mengelola Struktur tabel yang akan dibuat, misalnya menentukan nama Tabel, nama kolom, Tipe Data, Key, dan juga Query.

**Query** digunakan untuk melakukan permintaan terhadap basis data atau database, untuk membuat tabel tersebut, kita perlu memasukan nama Tabel, kolom serta Tipe Data dan juga Key pada Query tersebut, supaya bisa di Create.

Buat Class baru, berinama class tersebut, misalnya "*DBMahasiswa*", pada class ini, kita akan mengextends suatu class, bernama *SQLiteOpenHelper*, dan juga mengimplementasi method-method didalamnya, pertama, buatlah source codenya seperti berikut ini:

```
package android.fkti.unmul.com.sqliteexample;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class DBMahasiswa extends SQLiteOpenHelper{

    private static final String NamaDatabse = "unpi.db";
```

```

private static final int VersiDatabase = 1;

public DBMahasiswa(Context context) {
    super(context, NamaDatabase, null, VersiDatabase);
}

@Override
public void onCreate(SQLiteDatabase db) {

}

@Override
public void onUpgrade(SQLiteDatabase db, int i, int i1) {

}
}

```

### Penjelasan:

Coba kalian perhatikan, kita mengextends class `SQLiteOpenHelper` beserta fungsi-fungsinya, yaitu `onCreate()` dan `onUpgrade()`, kedua fungsi tersebut digunakan untuk Membuat Query dan juga untuk mengupgrade Struktur Tabel.

- ⇒ Pertama kita membuat 2 buah variable, yang digunakan untuk menentukan nama database dan versi database, untuk versi awal, kita set 1, terlebih dahulu.
- ⇒ Lalu kita implementasikan kedua variabel tersebut kedalam konstruktor dari class `DBMahasiswa`.
- ⇒ Selanjutnya kita buat InnerClass, yang digunakan untuk mengatur atribut-atribut seperti Nama Tabel, nama-nama kolom dan Query yang akan digunakan, pada InnerClass tersebut kita akan mengimplementasi class, bernama *BaseColumns*.

```

package android.fkti.unmul.com.sqliteexample;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.provider.BaseColumns;

public class DBMahasiswa extends SQLiteOpenHelper{

    //InnerClass, untuk mengatur atribut seperti Nama
    Tabel, nama-nama kolom dan Query
    static abstract class MyColumns implements
    BaseColumns{
        //Menentukan Nama Table dan Kolom
        static final String NamaTabel = "Mahasiswa";
    }
}

```

```

        static final String NIM = "NIM";
        static final String Nama = "Nama_Mahasiswa";
        static final String Jurusan = "Jurusan";
        static final String JenisKelamin =
"Jenis_Kelamin";
        static final String TanggalLahir =
"Tanggal_Lahir";
        static final String Alamat = "Alamat";
    }

    private static final String NamaDatabase = "unpi.db";
    private static final int VersiDatabase = 1;

    //Query yang digunakan untuk membuat Tabel
    private static final String SQL_CREATE_ENTRIES =
"CREATE TABLE "+MyColumns>NamaTabel+
        "("+MyColumns.NIM+" TEXT PRIMARY KEY,
"+MyColumns>Nama+" TEXT NOT NULL, "+MyColumns.Jurusan+
        " TEXT NOT NULL, "+MyColumns.JenisKelamin+"
TEXT NOT NULL, "+MyColumns.TanggalLahir+
        " TEXT NOT NULL, "+MyColumns.Alatat+" TEXT NOT
NULL)";

    //Query yang digunakan untuk mengupgrade Tabel
    private static final String SQL_DELETE_ENTRIES = "DROP
TABLE IF EXISTS "+MyColumns>NamaTabel;

    DBMahasiswa(Context context) {
        super(context, NamaDatabase, null, VersiDatabase);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(SQL_CREATE_ENTRIES);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int i, int
i1) {
        db.execSQL(SQL_DELETE_ENTRIES);
        onCreate(db);
    }
}

```

### Penjelasan:

Pada Class tersebut, kita menambahkan class Abstract, yaitu class yang tidak mempunyai Implementasinya secara langsung, class ini digunakan untuk menyiapkan atribut atau komponen yang diperlukan.

Lalu kita membuat 2 buah variable, untuk menyimpan Query pada Tabel, yang akan dipanggil pada method onCreate() dan onUpgrade(), pada variable **SQL\_CREATE\_ENTRIES**, digunakan untuk membuat dan menyimpan Query untuk membuat Struktur Tabelnya dan menentukan Tipe Data serta Key pada masing-masing kolom.

- ⇒ Pada kolom *NIM*, kita set Primary Key, karena ini merupakan nomor unik yang tidak bisa di duplikasi atau diduakan, karena masing-masing mahasiswa harus mempunyai nomor NIM yang berbeda.
- ⇒ Selanjutnya pada variable **SQL\_DELETE\_ENTRIES** digunakan untuk mengupgrade Tabel, jika terdapat perubahan pada Struktur tabel tersebut, program secara otomatis akan menghapus Tabel lama dan menggantikannya dengan yang baru.
- ⇒ Berikutnya, kita panggil Database tersebut pada MainActivity kita, untuk mengecek apakah Databsnya berhasil dibuat.

```
package android.fkti.unmul.com.sqlitedexample;

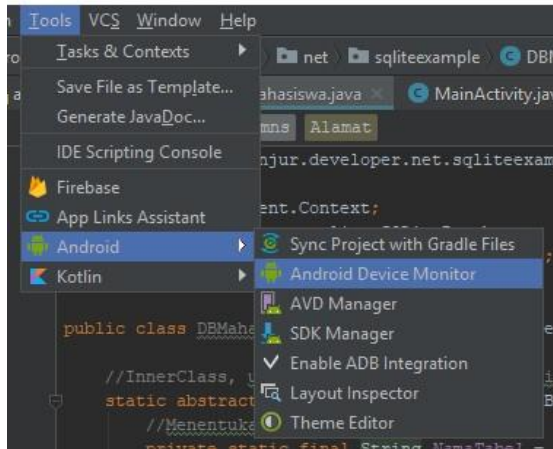
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        DBMahasiswa dbMahasiswa = new
DBMahasiswa(getApplicationContext());
        Toast.makeText(getApplicationContext(),
dbMahasiswa.getDatabaseName(), Toast.LENGTH_SHORT).show();
    }
}
```

Untuk mengecek apakah Database SQLitenya telah berhasil dibuat, kita dapat melakukan pengecekan dengan cara, jalankan project tersebut menggunakan **Emulator**, lalu buka **Android Device Emulator**, seperti berikut ini:



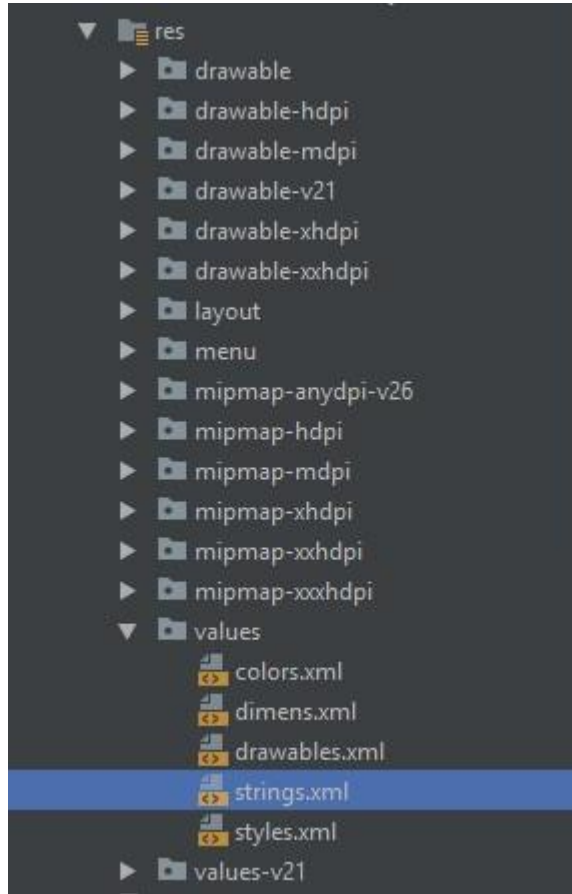


Jangan lupa untuk mencentang **Enable ADB Interation**. Selanjutnya buka direktori `data > data > (Nama Package pada Aplikasi Kita) > Database`.

### Cara Menambahkan Data pada SQLite Database

Sebelumnya kita sudah latihan mengenai Cara Membuat Database pada SQLite, selanjutnya kita akan latihan, bagaimana cara untuk menambahkan sebuah data yang di inputkan user pada SQLite Database

- ⇒ Pertama, buka file `activity_main.xml` pada project kalian, untuk mendapatkan Input dari user, kita membutuhkan bebrapa komponen seperti `EditText`, `RadioButton`, `Spinner` dan sebuah `Button` untuk mengeksekusi serta menyimpan data.  
Untuk input jurusan, kita menggunakan `Spinner` untuk menentukan pilihannya, dan pada input jenis kelamin, kita menggunakan `RadioButton` untuk pilihannya.
- ⇒ Karena kita menggunakan `Spinner` untuk input jurusannya, disini kita akan latihan cara menambahkan item pada `Spinner` tersebut, caranya kalian buka file `strings.xml` yang terdapat pada folder `res/values`.



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_margin="12dp"
android:orientation="vertical"

tools:context="android.fkti.unmul.com.sqliteexample.MainActivity">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```

        android:layout_marginBottom="4dp"
        android:gravity="center"
        android:text="Data Mahasiswa"
        android:textSize="20sp"
        android:textStyle="bold" />

<EditText
    android:id="@+id/nim"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Masukan NIM"
    android:inputType="textPersonName"/>

<EditText
    android:id="@+id/nama"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Masukan Nama"
    android:inputType="textPersonName" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginRight="4dp"
    android:layout_marginTop="6dp"
    android:text="Fakultas/Jurusan:" />

<Spinner
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/jurusan"
    android:entries="@array/daftar_jurusan"
    android:padding="15dp">

</Spinner>

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Jenis Kelamin:" />

<RadioGroup
    android:id="@+id/gender"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <RadioButton

```

```

        android:id="@+id/male"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Laki-Laki" />

        <RadioButton
            android:id="@+id/female"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="18dp"
            android:text="Perempuan" />
    </RadioGroup>

    <EditText
        android:id="@+id/date"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Tanggal Lahir"
        android:inputType="date" />

    <EditText
        android:id="@+id/alamat"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Alamat Lengkap"
        android:inputType="textMultiLine" />

    <Button
        android:id="@+id/save"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Simpan Data baru" />
</LinearLayout>

```

⇒ Lalu kita tambahkan string-array , misalnya kita ingin membuat daftar jurusan, tambahkan baris kode seperti berikut:

```

<string-array name="daftar_jurusan">
    <item>Teknik Informatika</item>
    <item>Manajemen Informatika</item>
    <item>Sastra Inggris</item>
    <item>Fakultas Ilmu Komunikasi</item>
    <item>Ekonomi Manajemen</item>
    <item>Ekonomi Akutansi</item>
</string-array>

```

⇒ Lalu tambahkan atribut entries didalam tag Spinner, seperti berikut ini:

```
<Spinner
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/jurusan"
    android:entries="@array/daftar_jurusan"
    android:padding="15dp">
```

⇒ Buka file MainActivity.java kalian, masukan source code berikut ini:

```
package android.fkti.unmul.com.sqliteexample;

import android.content.ContentValues;
import android.database.sqlite.SQLiteDatabase;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.Spinner;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    private EditText NIM, Nama, TanggalLahir, Alamat;
    private Spinner Jurusan;
    private RadioButton MALE, FEMALE;

    //Variable Untuk Menyimpan Input Dari Ueer
    private String setNIM, setName, setTanggalLahir,
    setAlamat, setJurusan, setJenisKelamin;

    //Variable Untuk Inisialisasi Database DBMahasiswa
    private DBMahasiswa dbMahasiswa;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button simpan = findViewById(R.id.save);
        NIM = findViewById(R.id.nim);
        Nama = findViewById(R.id.nama);
        TanggalLahir = findViewById(R.id.date);
        MALE = findViewById(R.id.male);
        FEMALE = findViewById(R.id.female);
        Jurusan = findViewById(R.id.jurusan);
        Alamat = findViewById(R.id.alamat);
```

```

        //Inisialisasi dan Mendapatkan Konteks dari
DBMahasiswa
        dbMahasiswa = new DBMahasiswa(getApplicationContext());

        simpan.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                setData();
                saveData();
                Toast.makeText(getApplicationContext(), "Data Berhasil
Disimpan", Toast.LENGTH_SHORT).show();
                clearData();
            }
        });

        //Berisi Statement-Statement Untuk Mendapatkan Input Dari
User
        private void setData(){
            setNIM = NIM.getText().toString();
            setName = Nama.getText().toString();
            setJurusan = Jurusan.getSelectedItem().toString();
            if(MALE.isChecked()){
                setJenisKelamin = MALE.getText().toString();
            }else if (FEMALE.isChecked()){
                setJenisKelamin = FEMALE.getText().toString();
            }
            setTanggalLahir = TanggalLahir.getText().toString();
            setAlamat = Alamat.getText().toString();
        }
        //Berisi Statement-Statement Untuk Menyimpan Data Pada
Database
        private void saveData(){
            //Mendapatkan Repository dengan Mode Menulis
            SQLiteDatabase create =
dbMahasiswa.getWritableDatabase();

            //Membuat Map Baru, Yang Berisi Nama Kolom dan Data
Yang Ingin Dimasukan
            ContentValues values = new ContentValues();
            values.put(DBMahasiswa.MyColumns.NIM, setNIM);
            values.put(DBMahasiswa.MyColumns>Nama, setName);
            values.put(DBMahasiswa.MyColumns>Jurusan, setJurusan);
            values.put(DBMahasiswa.MyColumns>JenisKelamin,
setJenisKelamin);
            values.put(DBMahasiswa.MyColumns>TanggalLahir,
setTanggalLahir);
            values.put(DBMahasiswa.MyColumns>Alamat, setAlamat);

```

```

//Menambahkan Baris Baru, Berupa Data Yang Sudah
Diinputkan pada Kolom didalam Database
create.insert(DBMahasiswa.MyColumns>NamaTabel, null,
values);
}
private void clearData(){
    NIM.setText("");
    Nama.setText("");
    TanggalLahir.setText("");
    Alamat.setText("");
}
}

```

### Menampilkan Data SQLite pada ListView

Untuk mengambil data dari SQLite Database, kita dapat menggunakan suatu Class bernama **Cursor**, Cursor di implementasikan sebagai Pointer atau petunjuk, yang memungkinkan kita untuk mengambil data dari Rows (Baris) pada sebuah kolom didalam SQLite Database.

Cursor digunakan untuk mengarahkan atau menentukan data mana yang akan diambil, kita dapat menentukan kolom dan juga no index atau posisi dari suatu data didalam baris.

⇒ Pertama buka project yang sudah kalian buat sebelumnya, buka *activity\_main.xml*, kita akan menambahkan Button baru, untuk berpindah activity pada activity yang menampilkan data.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_margin="12dp"
android:orientation="vertical"
tools:context="android.fkti.unmul.com.sqliteexample.MainActivity">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="4dp"
        android:gravity="center"
        android:text="Data Mahasiswa"
        android:textSize="20sp"
        android:textStyle="bold" />

```

```

<EditText
    android:id="@+id/nim"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Masukan NIM"
    android:inputType="textPersonName"/>

<EditText
    android:id="@+id/nama"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Masukan Nama"
    android:inputType="textPersonName" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginRight="4dp"
    android:layout_marginTop="6dp"
    android:text="Fakultas/Jurusan:" />

<Spinner
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/jurusan"
    android:entries="@array/daftar_jurusan"
    android:padding="15dp">

</Spinner>

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Jenis Kelamin:" />

<RadioGroup
    android:id="@+id/gender"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <RadioButton
        android:id="@+id/male"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:checked="false"
        android:text="Laki-Laki" />

```



```

        <RadioButton
            android:id="@+id/female"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="18dp"
            android:text="Perempuan" />
    </RadioGroup>

    <EditText
        android:id="@+id/date"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Tanggal Lahir"
        android:inputType="date" />

    <EditText
        android:id="@+id/alamat"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Alamat Lengkap"
        android:inputType="textMultiLine" />

    <Button
        android:id="@+id/save"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Simpan Data baru" />

    <Button
        android:id="@+id/readData"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Lihat Data Yang Tersimpan" />
</LinearLayout>

```

Jangan lupa untuk menambahkan *OnClickListener* pada Button tersebut:

```

Button viewData = findViewById(R.id.readData);
viewData.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view) {
        startActivity(new Intent(MainActivity.this,
ViewData.class));
    }
});

```

Buat Activity Baru, berimana, misalnya *"ViewData"*, pada activity tersebut, tambahkan ListView seperti berikut ini:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="8dp"

tools:context="android.fkti.unmul.com.sqliteexample.ViewData
">

    <ListView
        android:id="@+id/list"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true" />

</RelativeLayout>
```

Buka *ViewData.java* pada project kalian, disini kita akan membuat sebuah Cursor untuk mencatat data yang akan diambil, lalu menampilkannya pada ListView.

```
package android.fkti.unmul.com.sqliteexample;

import android.annotation.SuppressLint;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import java.util.ArrayList;

public class ViewData extends AppCompatActivity {

    private ListView listView;
    private DBMahasiswa MyDatabase;
    private ArrayList<String> ListData;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_data);
        getSupportActionBar().setTitle("Daftar Mahasiswa");
    }
}
```

```

        listView = findViewById(R.id.list);
        ListData = new ArrayList<>();
        MyDatabase = new DBMahasiswa(getApplicationContext());
        getData();
        listView.setAdapter(new ArrayAdapter<>(this,
android.R.layout.simple_list_item_1, ListData));
    }

    //Berisi Statement-Statement Untuk Mengambi Data dari
    Database
    @SuppressWarnings("Recycle")
    private void getData(){
        //Mengambil Repository dengan Mode Membaca
        SQLiteDatabase ReadData =
MyDatabase.getReadableDatabase();
        Cursor cursor = ReadData.rawQuery("SELECT * FROM "+
DBMahasiswa.MyColumns>NamaTabel,null);

        cursor.moveToFirst();//Memulai Cursor pada Posisi
    Awal

        //Meloooping Sesuai Dengan Jumlahn Data (Count) pada
    cursor
        for(int count=0; count < cursor.getCount();
    count++){

        cursor.moveToPosition(count);//Berpindah Posisi dari no
    index 0 hingga no index terakhir

                ListData.add(cursor.getString(1));//Menambil
    Data Dari Kolom 1 (Nama)
                //Lalu Memasukan Semua Datanya kedalam
    ArrayList
        }
    }
}

```

### Penjelasan:

Pertama kita membuat beberapa Variable, yaitu ListView (Untuk memberikan fungsi pada ListView), DBMahasiswa (Untuk Menhubungkan Activity dengan Database pada class ini) dan ArrayList (Untuk Menyimpan Data Yang Telah Didapat dari Database Menggunakan Cursor).

Pada Method *getData()*, disana terdapat Statement-Statement untuk mengambil data Dari Database, dalam method tersebut, kita menggunakan Cursor untu mengarahkan data mana yang ingin diambil, karena kita ingin mengambil semua datanya sekaligus, jai kita cukup menggunakan *rawQuery()*

untuk mengambilnya, menggunakan `"SELECT * FROM"`, yang digunakan untuk mengambil semua data dari Tabel.

Supaya datanya bisa terambil semua, kita set posisi cursor tersebut berada di posisi awal, dengan menggunakan fungsi `moveToFirst()`, jadi setiap ada data yang masuk, cursor akan mengambil data tersebut dari awal, hingga akhir.

Pada program tersebut, kita menggunakan for-loops untuk melooping sebanyak data yang terdapat pada database, posisi cursor akan berubah ubah dari baris (row) 0 hingga jumlah yang telah ditentukan, jadi dengan begitu, kita dapat dengan mudah untuk mengambil semua data tersebut dari Database.

Lalu pada ArrayList, kita mengambil data dan memasukkannya kedalam ArrayList dengan menggunakan fungsi `getString(1)`, angka 1 (satu) tersebut artinya, kita mengambil data dari kolom 1, yaitu Nama.

Terakhir kita set ArrayList tersebut pada Adapter, dimana data tersebut akan di tampilkan pada ListView.

## Menampilkan Data SQLite pada RecyclerView

Seperti pada latihan sebelumnya, untuk mengambil data dari SQLite Database, kita dapat menggunakan suatu Class bernama Cursor, Cursor di implementasikan sebagai Pointer atau petunjuk, yang memungkinkan kita untuk mengambil data dari Rows (Baris) pada sebuah kolom didalam SQLite Database.

⇒ Pertama, kalian masukan beberapa library berikut ini, pada *dependencies* didalam file *build.gradle* (didalam direktori app).

```
compile 'com.android.support:appcompat-v7:27.0.2'  
compile 'com.android.support:design:27.0.2'  
compile 'com.android.support:recyclerview-v7:27.0.2'  
compile 'com.android.support:support-v4:27.0.2'
```

⇒ Pertama, yang perlu kita lakukan yaitu membuat design layout untuk setiap item didalam RecyclerView, caranya buat layout baru, bernama `"view_design"`, pada layout tersebut kita akan menggunakan beberapa view seperti TextView dan ImageView.

```
<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    android:id="@+id/item_list"
```

```

android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_margin="8dp"
android:orientation="vertical">

<ImageView
    android:id="@+id/image"
    android:layout_width="70dp"
    android:layout_height="70dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    app:srcCompat="@drawable/unpi" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="70dp"
    android:layout_toLeftOf="@+id/overflow"
    android:layout_toRightOf="@+id/image"
    android:gravity="center"
    android:orientation="vertical">

    <TextView
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="2dp"
        android:text="Nama"
        android:textSize="18sp"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/NIM"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="2dp"
        android:text="NIM"
        android:textSize="14sp" />

    <TextView
        android:id="@+id/jurusan"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="2dp"
        android:text="Jurusan"
        android:textSize="14sp" />
</LinearLayout>

</RelativeLayout>

```

Selanjutnya kita akan membuat sebuah class bernama *RecyclerViewAdapter*, adapter tersebut akan kita gunakan untuk mengatur data model yang akan ditampilkan pada *RecyclerView*, buat class baru bernama *RecyclerViewAdapter*.

```
package android.fkti.unmul.com.sqliteliteexample;

import android.annotation.SuppressLint;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import java.util.ArrayList;

//Class Adapter ini Digunakan Untuk Mengatur Bagaimana Data
akan Ditampilkan
public class RecyclerViewAdapter extends
RecyclerView.Adapter<RecyclerViewAdapter.ViewHolder>{

    private ArrayList namaList; //Digunakan untuk Nama
    private ArrayList jurusanList; //Digunakan untuk Jurusan
    private ArrayList nimList; //Digunakan untuk Jurusan

    //Membuat Konstruktor pada Class RecyclerViewAdapter
    RecyclerViewAdapter(ArrayList namaList, ArrayList
    jurusanList, ArrayList nimList){
        this.namaList = namaList;
        this.jurusanList = jurusanList;
        this.nimList = nimList;
    }

    //ViewHolder Digunakan Untuk Menyimpan Referensi Dari
View-View
    class ViewHolder extends RecyclerView.ViewHolder{

        private TextView Nama, Jurusan, Nim;

        ViewHolder(View itemView) {
            super(itemView);
            //Menginisialisasi View-View untuk kita gunakan
pada RecyclerView
            Nama = itemView.findViewById(R.id.name);
            Jurusan = itemView.findViewById(R.id.jurusan);
            Nim = itemView.findViewById(R.id.NIM);
        }
    }
}
```

```

@Override
public ViewHolder onCreateViewHolder(ViewGroup parent,
int viewType) {
    //Membuat View untuk Menyiapkan dan Memasang Layout
    yang Akan digunakan pada RecyclerView
    View V =
    LayoutInflater.from(parent.getContext()).inflate(R.layout.vie
w_design, parent, false);
    return new ViewHolder(V);
}

@Override
public void onBindViewHolder(ViewHolder holder,
@SuppressLint("RecyclerView") final int position) {
    //Memanggil Nilai/Value Pada View-View Yang Telah
    Dibuat pada Posisi Tertentu
    final String Nama =
    namaList.get(position); //Mengambil data (Nama) sesuai dengan
    posisi yang telah ditentukan
    final String Jurusan =
    jurusanList.get(position); //Mengambil data (Jurusan) sesuai
    dengan posisi yang telah ditentukan
    final String NIM = nimList.get(position); //Mengambil
    data (NIM) sesuai dengan posisi yang telah ditentukan
    holder>Nama.setText(Nama);
    holder.Jurusan.setText(Jurusan);
    holder.Nim.setText(NIM);
}

@Override
public int getItemCount() {
    //Menghitung Ukuran/Jumlah Data Yang Akan Ditampilkan
    Pada RecyclerView
    return nimList.size();
}
}

```

Klik kanan pada direktori, pilih *New > Drawable resource file*, bernimana *"line"*, file ini digunakan untuk membuat ***Underline*** pada setiap list item didalam RecyclerView.

```

<?xml version="1.0" encoding="utf-8"?>
<shape
xmlns:android="http://schemas.android.com/apk/res/android"
android:shape="rectangle">
    <solid android:color="#FFD8D8D8"/>
    <size android:height="0.5dp"/>
</shape>

```

Selanjutnya pada *activity\_view\_data*, kalian ganti Tambahkan atau ganti ListView tersebut dengan RecyclerView.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="8dp"

tools:context="android.fkti.unmul.com.sqliteexample.ViewData">

    <android.support.v7.widget.RecyclerView
        android:id="@+id/recycler"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:scrollbars="vertical">

    </android.support.v7.widget.RecyclerView>

</RelativeLayout>
```

Buka file *ViewData.java*, untuk source codenya, kalian bisa membuatnya seperti berikut ini:

```
package android.fkti.unmul.com.sqliteexample;

import android.annotation.SuppressLint;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.DividerItemDecoration;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;

import java.util.ArrayList;

public class ViewData extends AppCompatActivity {
    private DBMahasiswa MyDatabase;

    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;
    private ArrayList>NamaList;
    private ArrayList>JurusanList;
```



```

private ArrayList NIMList;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_view_data);
    getSupportActionBar().setTitle("Daftar Mahasiswa");
    NamaList = new ArrayList<>();
    JurusanList = new ArrayList<>();
    NIMList = new ArrayList<>();
    MyDatabase = new DBMahasiswa(getApplicationContext());
    recyclerView = findViewById(R.id.recycler);
    getData();
    //Menggunakan Layout Manager, Dan Membuat List Secara
Vertical
    layoutManager = new LinearLayoutManager(this);
    recyclerView.setLayoutManager(layoutManager);
    recyclerView.setHasFixedSize(true);
    adapter = new RecyclerViewAdapter(NamaList,
JurusanList, NIMList);
    //Memasang Adapter pada RecyclerView
    recyclerView.setAdapter(adapter);
    //Membuat Underline pada Setiap Item Didalam List
    DividerItemDecoration itemDecoration = new
DividerItemDecoration(getApplicationContext(),
DividerItemDecoration.VERTICAL);

    itemDecoration.setDrawable(ContextCompat.getDrawable(getApplic
ationContext(), R.drawable.line));
    recyclerView.addItemDecoration(itemDecoration);
}

//Berisi Statement-Statement Untuk Mengambi Data dari
Database
@SuppressLint("Recycle")
protected void getData(){
    //Mengambil Repository dengan Mode Membaca
    SQLiteDatabase ReadData =
MyDatabase.getReadableDatabase();
    Cursor cursor = ReadData.rawQuery("SELECT * FROM "+
DBMahasiswa.MyColumns>NamaTabel,null);

    cursor.moveToFirst();//Memulai Cursor pada Posisi Awal

    //Meloooping Sesuai Dengan Jumlan Data (Count) pada
cursor
    for(int count=0; count < cursor.getCount(); count++){

        cursor.moveToPosition(count);//Berpindah Posisi dari no
index 0 hingga no index terakhir

```

```

    NIMList.add(cursor.getString(0)); //Menambil Data Dari Kolom
0 (NIM)
    NamaList.add(cursor.getString(1)); //Menambil Data Dari Kolom
1 (Nama)
    JurusanList.add(cursor.getString(2)); //Menambil Data Dari
Kolom 2 (Jurusan)
    }
    }
}

```

## Membuat Fungsi Delete dan Update pada SQLite Database

Pada materi yang akan kita pelajari hari ini, kita akan belajar cara menghapus dan juga mengedit data yang tersimpan pada SQLite Database, sebelumnya kita sudah belajar cara menampilkan data SQLite pada RecyclerView, pada tutorial ini, kita dapat menghapus dan update data tersebut melalui RecyclerView.

Untuk itu kita akan menambahkan Menu Navigasi seperti menu Overflow pada layout/list item RecyclerView tersebut. Untuk menampilkan menu pilihan pada Menu Overflow, kita akan menggunakan Menu Popup untuk menampilkan menu pilihan berupa Delete dan Update pada masing-masing item.

Pertama kita perlu membuat menu terlebih dahulu, caranya : Klik kanan pada direktori *res > New > Android resource directory*. lalu berinama file tersebut, misalnya "*popup\_menu*". setelah itu klik Ok.

Selanjutnya buat menu tersebut dengan menggunakan kode xml, untuk membuatnya, kita perlu menambahkan tag *<item>* dan juga beberapa atribut tambahan lainnya, seperti id, title, dsb.

```

<?xml version="1.0" encoding="utf-8"?>
<menu
xmlns:android="http://schemas.android.com/apk/res/android">

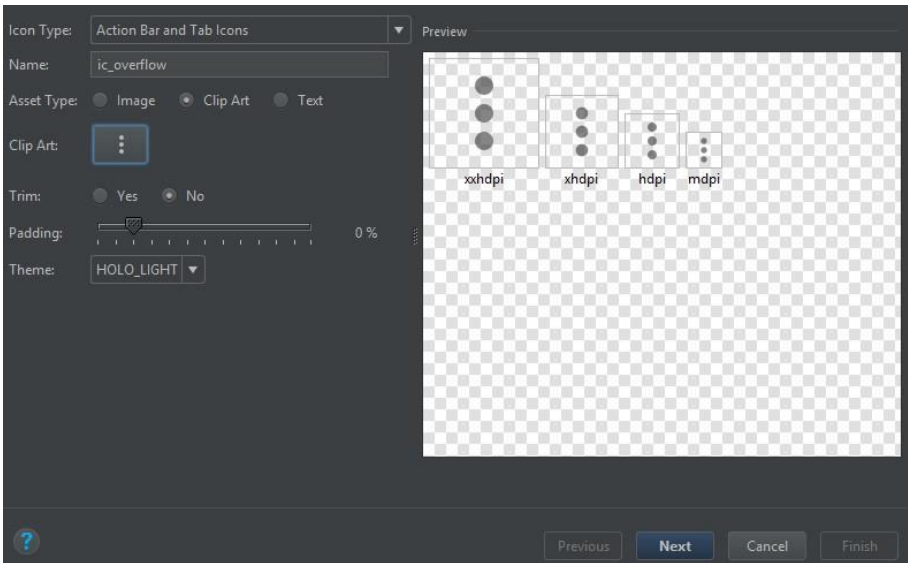
    <item android:id="@+id/delete"
        android:title="Delete"/>

    <item android:id="@+id/update"
        android:title="Update"/>

</menu>

```

Buat Icon Menu Overflownya, klik kanan pada dir *res > New > Image Asset*.



Selanjutnya kita edit layout design pada RecyclerView, buka file *view\_design.xml*, kita akan menambahkan ImageButton sebagai navigasi untuk icon tersebut, kalian edit menjadi seperti berikut ini:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/item_list"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="8dp"
    android:orientation="vertical">

    <ImageView
        android:id="@+id/image"
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:layout_marginEnd="8dp"
        android:layout_marginRight="8dp"
        app:srcCompat="@drawable/unpi" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="70dp"
        android:layout_toLeftOf="@+id/overflow"
        android:layout_toRightOf="@+id/image"
        android:gravity="center">
```

```

        android:orientation="vertical">

        <TextView
            android:id="@+id/name"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="2dp"
            android:text="Nama"
            android:textSize="18sp"
            android:textStyle="bold" />

        <TextView
            android:id="@+id/NIM"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="2dp"
            android:text="NIM"
            android:textSize="14sp" />

        <TextView
            android:id="@+id/jurusan"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="2dp"
            android:text="Jurusan"
            android:textSize="14sp" />
    </LinearLayout>

    <ImageButton
        android:id="@+id/overflow"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_centerVertical="true"
        android:background="#FFF9F9"
        app:srcCompat="@drawable/ic_overflow" />

</RelativeLayout>

```

## Menghapus Data Dari Database

Buka *RecyclerViewAdapter* kalian, lalu tambahkan *Context* dan juga *ImageButton* didalam class *ViewHolder*, seperti ini:

```

public class RecyclerViewAdapter extends
RecyclerView.Adapter{

    private ArrayList namaList; //Digunakan untuk Nama

```

```

private ArrayList jurusanList; //Digunakan untuk Jurusan
private ArrayList nimList; //Digunakan untuk Jurusan
private Context context; //Membuat Variable Context

//Membuat Konstruktor pada Class RecyclerViewAdapter
RecyclerViewAdapter(ArrayList namaList, ArrayList
jurusanList, ArrayList nimList){
    this.namaList = namaList;
    this.jurusanList = jurusanList;
    this.nimList = nimList;
}

//ViewHolder Digunakan Untuk Menyimpan Referensi Dari
View-View
class ViewHolder extends RecyclerView.ViewHolder{

    private TextView Nama, Jurusan, Nim;
    private ImageButton Overflow;

    ViewHolder(View itemView) {
        super(itemView);

        //Mendapatkan Context dari itemView yang
terhubung dengan Activity ViewData
        context = itemView.getContext();

        //Menginisialisasi View-View untuk kita gunakan
pada RecyclerView
        Nama = itemView.findViewById(R.id.name);
        Jurusan = itemView.findViewById(R.id.jurusan);
        Overflow = itemView.findViewById(R.id.overflow);
        Nim = itemView.findViewById(R.id.NIM);
    }
}

```

Lalu didalam method *onBindViewHoler()*, tambahkan *onClickListener* pada *ImageButton* tersebut, didalamnya akan kita tambahkan *Popup* menu pada *onClicListener* tersebut, seperti berikut ini:

```

public void onBindViewHolder(final ViewHolder holder,
@SuppressLint("RecyclerView") final int position) {
    //Memanggil Nilai/Value Pada View-View Yang Telah
Dibuat pada Posisi Tertentu
    final String Nama =
namaList.get(position); //Mengambil data (Nama) sesuai dengan
posisi yang telah ditentukan
    final String Jurusan =
jurusanList.get(position); //Mengambil data (Jurusan) sesuai
dengan posisi yang telah ditentukan
}

```

```

        final String NIM = nimList.get(position); //Mengambil
data (NIM) sesuai dengan posisi yang telah ditentukan
        holder>Nama.setText>Nama);
        holder.Jurusan.setText(Jurusan);
        holder.Nim.setText(NIM);

        //Mengimplementasikan Menu Popup pada Overflow
(ImageButton)
        holder.Overflow.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(final View view) {
                //Membuat Instance/Objek dari PopupMenu
                PopupMenu popupMenu = new PopupMenu(view.getContext(),
view);
                popupMenu.inflate(R.menu.popup_menu);
                popupMenu.setOnMenuItemClickListener(new
PopupMenu.OnMenuItemClickListener() {
                    @Override
                    public boolean onOptionsItemSelected(MenuItem item) {
                        switch (item.getItemId()){
                            case R.id.delete:
                                //Menghapus Data Dari Database
                                DBMahasiswa getDatabase = new
DBMahasiswa(view.getContext());
                                SQLiteDatabase DeleteData =
getDatabase.getWritableDatabase();
                                //Menentukan di mana bagian kueri yang akan dipilih
                                String selection = DBMahasiswa.MyColumns.NIM + " LIKE ?";
                                //Menentukan Nama Dari Data Yang Ingin Dihapus
                                String[] selectionArgs =
{holder.Nim.getText().toString()};

                                DeleteData.delete(DBMahasiswa.MyColumns>NamaTabel, selection,
selectionArgs);

                                //Menghapus Data pada List dari Posisi Tertentu
                                int position =
nimList.indexOf(NIM);
                                nimList.remove(position);
                                notifyItemRemoved(position);

                                Toast.makeText(view.getContext(), "Data
Dihapus", Toast.LENGTH_SHORT).show();
                                break;

                                case R.id.update:
                                    //Statement Update
                                    break;
                        }
                    }
                }
            }
        }
    }
}

```

```

        return true;
    }
    });
    popupMenu.show();
}
});
}

```

### Penjelasan:

Pada menu overflow yang sudah kita buat tadi sebelumnya, menggunakan ImageButton, saat menu tersebut diklik, maka akan muncul 2 buah menu Popup, yaitu Delete dan Update.

Untuk menghapus Data dari Database SQLite, pertama kita hapus membuat Instance/Objek dan juga context dengan class database kita, yaitu *DBMahasiswa*, dengan parameter *view.getContext()*.

Pada variable selection, kita menentukan Bagian Nama Query yang akan dipilih, pada contoh tersebut kita memilih Kolom **NIM**, sebagai acuan untuk menghapus data. Jadi artinya, kita dapat menghapus salah satu data dari database berdasarkan NIM item yang dipilih.

Pada variable Array selectionArgs, digunakan untuk menentukan NIM dari Data yang akan dihapus, NIM tersebut didapat dari masing-masing item, terdantung posisinya. Jadi kita dapat menghapus data tersebut berdasarkan NIM pada item didalam RecyclerView.

Dan yang terakhir, digunakan untuk menghapus Item dari RecyclerView, supaya saat kita mengklik menu Delete, maka secara realtime item yang kita pilih pada RecyclerView akan menghilang.

### Mengedit Data Dari Database

1) Pertama kita akan membuat Activity baru, bernama "*UpdateActivity*", activity tersebut akan kita gunakan untuk mengedit data dari Database, disini kita hanya akan mengupdate atau mengubah Nama, NIM dan Jurusan pada setiap Item yang dipilih. Untuk itu kalian buat activitynya seperti berikut ini:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_margin="8dp"

```

```

        android:orientation="vertical"

tools:context="android.fkti.unmul.com.sqliteexample.UpdateActi
vity">

    <EditText
        android:id="@+id/new_nim"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="NIM Baru"
        android:inputType="textPersonName" />

    <EditText
        android:id="@+id/new_nama"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Nama Baru"
        android:inputType="textPersonName" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="6dp"
        android:text="Fakultas/Jurusan:" />

    <Spinner
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/new_jurusan"
        android:entries="@array/daftar_jurusan"
        android:padding="15dp">

</Spinner>

    <Button
        android:id="@+id/new_data"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Simpan Data Baru" />
</LinearLayout>

```

Buka kembali RecyclerViewAdapter, tambahkan baris kode berikut ini pada Item didalam Popup menu:

```

case R.id.update:
    Intent dataForm = new Intent(view.getContext(),
UpdateActivity.class);

```



```

        dataForm.putExtra("SendNIM",
holder.Nim.getText().toString());
        dataForm.putExtra("SendNama",
holder>Nama.getText().toString());
        context.startActivity(dataForm);
        ((Activity)context).finish();
        break;

```

Disana kita menggunakan Intent untuk berpindah activity, serta Mempassing/Kirim Data, NIM pada activity selanjutnya, untuk kita kirim pada ActivityUpdate, NIM tersebut digunakan untuk menentukan Data mana yang ingin di Update.

Lalu kita panggil variable context yang sudah kita buat tadi sebelumnya, untuk mengeksekusi Intent tersebut.

Buka file *UpdateActivity.java*, masukan source code berikut ini:

```

package android.fkti.unmul.com.sqliteexample;

import android.content.ContentValues;
import android.content.Intent;
import android.database.sqlite.SQLiteDatabase;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.Toast;

public class UpdateActivity extends AppCompatActivity {

    private DBMahasiswa MyDatabase;
    private EditText NewNIM, NewNama;
    private Spinner NewJurusan;
    private String getNewNIM, getNewNama, getNewJurusan;
    private Button Update;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_update);
        getSupportActionBar().setTitle("Masukan Data Baru");
        MyDatabase = new DBMahasiswa(getBaseContext());
        NewNIM = findViewById(R.id.new_nim);
        NewNama = findViewById(R.id.new_nama);

```

```

        NewJurusan = findViewById(R.id.new_jurusan);

        //Menerima Data Nama dan NIM yang telah dipilih Oleh
        User untuk diposes

        NewNama.setText(getIntent().getExtras().getString("SendNama");
        NewNim.setText(getIntent().getExtras().getString("SendNIM");

        Update = findViewById(R.id.new_data);
        Update.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                setUpdateData();
                startActivity(new Intent(UpdateActivity.this,
ViewData.class));
                finish();
            }
        });
    }
    private void setUpdateData(){
        getNewNIM = NewNIM.getText().toString();
        getNewNama = NewNama.getText().toString();
        getNewJurusan =
NewJurusan.getSelectedItem().toString();
        SQLiteDatabase database =
MyDatabase.getReadableDatabase();

        //Memasukan Data baru pada 3 kolom (NIM, Nama dan
Jurusan)
        ContentValues values = new ContentValues();
        values.put(DBMahasiswa.MyColumns>Nama, getNewNama);
        values.put(DBMahasiswa.MyColumns.Jurusan,
getNewJurusan);
        values.put(DBMahasiswa.MyColumns.NIM, getNewNIM);

        //Untuk Menentukan Data/Item yang ingin diubah,
berdasarkan NIM
        String selection = DBMahasiswa.MyColumns.NIM + " LIKE
?";

        String[] selectionArgs = {GetNIM};
        database.update(DBMahasiswa.MyColumns>NamaTabel,
values, selection, selectionArgs);
        Toast.makeText(getApplicationContext(), "Berhasil
Diubah", Toast.LENGTH_SHORT).show();
    }
}

```

## Latihan 7.2. SQLite Room Database

Latihan ini membuat aplikasi CRUD SQLite dengan sedikit berbeda yaitu menggunakan library Room Database. Kita tetap menggunakan SQLite Database, hanya saja ada beberapa perbedaan saat kita membuat app CRUD SQLite dengan Room Database.

Berikut ini perbedaan mendasar antara SQLite dengan Room Database:

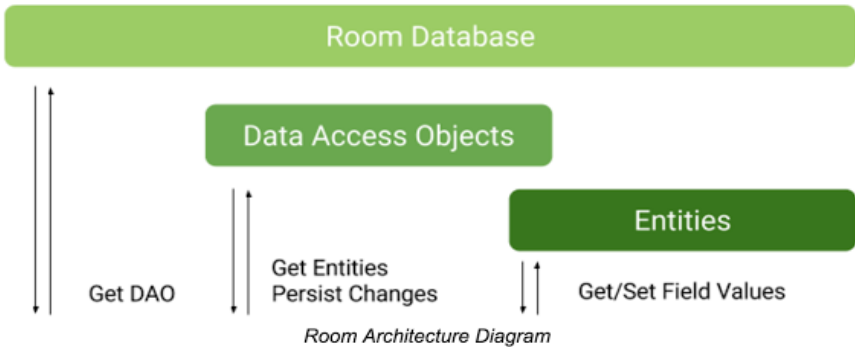
- **SQLite:** Menggunakan class seperti SQLiteOpenHelper dan interface standar lainnya seperti DBHelper, dsb.
- **Room Database:** Full Abstraction menggunakan Room Library yang lebih efisien, karena sudah menyediakan method siap pakai yang lebih mudah untuk digunakan.

Seperti yang disebutkan, Room bukanlah sistem basis data baru. Ini adalah lapisan abstrak yang membungkus database SQLite standar yang diadopsi oleh Android. Namun, Room menambahkan begitu banyak fitur ke SQLite yang hampir tidak mungkin dikenali. Room menyederhanakan semua operasi yang berhubungan dengan database dan juga membuat mereka jauh lebih kuat karena memungkinkan kemungkinan pengembalian yang dapat diamati dan kueri SQL waktu kompilasi-waktu.

Kamar terdiri dari tiga komponen utama: **Database**, **DAO** (Objek Akses Data), dan **Entity**. Setiap komponen memiliki tanggung jawabnya, dan semuanya harus diimplementasikan agar sistem berfungsi. Untungnya, penerapannya cukup sederhana. Berkat anotasi dan kelas abstrak yang disediakan, boiler untuk menerapkan Room dijaga seminimal mungkin.

1. **Entity:** Seperti database pada umumnya, entity digunakan untuk merepresentasikan data pada sebuah tabel, contohnya seperti id, nama, dan lainnya.
2. **DAO (Data Access Object):** Menggambarkan sebuah method-method khusus atau standar untuk mengakses database, termasuk salah satunya CRUD (Create, Read, Update, Delete).
3. **Database:** adalah class holder yang digunakan untuk menampilkan daftar dari entity yang ada dan kelas ini juga berisi daftar dari DAO serta database version.

Gambaranya bisa kalian lihat pada diagram Room Database berikut ini.



Gambar 107. Room Database

Pada latihan ini kita akan belajar membuat aplikasi CRUD sederhana, yaitu aplikasi pendataan Mahasiswa baru menggunakan Room Database, jadi kemungkinan data pada entity yang akan kita buat seperti:

- NIM (Primary Key)
- Nama
- Jurusan
- Tanggal Lahir
- Dan lainnya

Atribut-atribut tersebut yang nantinya akan direpresentasikan ke dalam object Mahasiswa, yang akan digunakan sebagai entity pada Room database di project aplikasi yang akan kita buat.

⇒ Pertama sebelum kita menggunakan Room Database, kita perlu menambahkan kedua library berikut ini kedalam dependencies pada file **build.gradle (app level)**.

```
implementation
"android.arch.persistence.room:runtime:1.1.1"
annotationProcessor
"android.arch.persistence.room:compiler:1.1.1"
```

Setelah itu lakukan sync gradle. Jika sudah selesai maka library Room akan terpasang di project aplikasi Android tersebut.

Latihan ini terbagi menjadi 6 bagian, yaitu :

## 1) Membuat Entity, DAO, dan Database pada Room Database

Buka project Android Studio kalian sebelumnya, pertama kita akan membuat class Entity nya terlebih dahulu, pada class ini kita akan mendefinisikan atribut-atribut yang diperlukan untuk table Mahasiswa seperti NIM (primary key), Nama, Jurusan, Tanggal Lahir dan Jenis Kelamin.

### ▪ Membuat Entity pada Room Database

Buat class baru, bernama Mahasiswa.java, class ini yang akan kita jadikan sebagai Entity. Masukkan source code beserta penjelasannya seperti berikut ini:

```
Mahasiswa.java
package com.fkti.crudroomdatabase.model;

import android.arch.persistence.room.ColumnInfo;
import android.arch.persistence.room.Entity;
import android.arch.persistence.room.PrimaryKey;
import android.support.annotation.NonNull;

import java.io.Serializable;

@Entity(tableName = "tMahasiswa") //Membuat tabel baru dengan
nama "mahasiswa"
public class Mahasiswa implements Serializable{

    //Membuat kolom NIM sebagai Primary Key
    @NonNull
    @PrimaryKey
    @ColumnInfo(name = "nim_mahasiswa")
    private
    String nim;

    //Membuat kolom Nama
    @ColumnInfo(name = "nama_mahasiswa")
    private
    String nama;

    //Membuat kolom Tanggal Lahir
    @ColumnInfo(name = "tanggal_lahir")
    private
    String tanggalLahir;

    //Membuat kolom Jurusan
    @ColumnInfo(name = "jurusan")
    private
    String jurusan;
```

```

//Jenis Kelamin
@ColumnInfo(name = "jenis_kelamin")
private
String jenisKelamin;

//Method untuk mengambil data NIM
@NonNull
public String getNim() {
    return nim;
}

//Method untuk memasukan data NIM
public void setNim(@NonNull String nim) {
    this.nim = nim;
}

//Method untuk mengambil data Nama
public String getNama() {
    return nama;
}

//Method untuk memasukan data Nama
public void setNama(String nama) {
    this.nama = nama;
}

//Method untuk mengambil data Tanggal Lahir
public String getTanggalLahir() {
    return tanggalLahir;
}

//Method untuk memasukan data Tanggal Lahir
public void setTanggalLahir(String tanggalLahir) {
    this.tanggalLahir = tanggalLahir;
}

//Method untuk mengambil data Jurusan
public String getJurusan() {
    return jurusan;
}

//Method untuk memasukan data Jurusan
public void setJurusan(String jurusan) {
    this.jurusan = jurusan;
}

//Method untuk mengambil data Jenis Kelamin
public String getJenisKelamin() {
    return jenisKelamin;
}

```

```
//Method untuk memasukan data Jenis Kelamin
public void setJenisKelamin(String jenisKelamin) {
    this.jenisKelamin = jenisKelamin;
}
}
```

Perhatikan, class tersebut mirip dengan class Objek Model Java pada umumnya, didalam class tersebut kita definisikan beberapa atribut seperti NIM, Nama, Tanggal Lahir, Jurusan dan Jenis Kelamin. Serta mehod Getter dan Setter yang digunakan untuk mengambil dan memasukan data.

Untuk membuat Entity kita perlu menggunakan @ (Annotation) untuk membuat tabel dan atributnya. pada class tersebut kita dapat membuat tabel dengan menggunakan Annotation @Entity, untuk merepresentasikan tabel Mahasiswa pada Database.

Lalu untuk membuat kolom, kita bisa menggunakan Annotation @ColumnInfo, berikutnya pada NIM kita definisikan sebagai @PrimaryKey, karena setiap mahasiswa NIMnya pasti berbeda-beda.

Untuk Primary Key didalam Room Database kita juga dapat membuat autoGenerate id seperti ini @PrimaryKey(autoGenerate = true) yang secara otomatis akan membuat id-nya sendiri

Pada atribut NIM kita definisikan dengan Annotation @NonNull, karena data yang dimasukan tidak boleh Null/Kosong, terakhir kita meng-extends Serializable, untuk mengubah data objek pada class tersebut menjadi Byte, karena nanti data tersebut akan disimpan dalam bentuk File.

Berikutnya kita buat interface DAO atau Data Access Database, dengan nama MahasiswaDAO.java, interface ini berisi method-method khusus yang digunakan untuk mengakses database, seperti Insert, Read, Update, Delete, Select, atau semacamnya.

```
package com.fkti.crudroomdatabase.data;

import android.arch.persistence.room.Dao;
@Dao
public interface MahasiswaDAO {
}
```

Hampir sama seperti pada interface pada umumnya, yang membedakannya, pada interface tersebut kita menambahkan Annotation @DAO dari Library Room

Database. Interface tersebut kita biarkan kosong, karena akan kita bahas pada tutorial berikutnya.

Jika kalian sudah membuat kedua komponen tersebut, terakhir kita buat class Database Room itu sendiri.

### ▪ Membuat Abstract Class pada Room

Buat class baru bernama *AppDatabase.java*. Class ini digunakan untuk membuat objek dari Room Database itu sendiri, didalamnya terdapat class-class DAO yang akan digunakan.

```
package com.fkti.crudroomdatabase.data.factory;

import android.arch.persistence.room.Database;
import android.arch.persistence.room.RoomDatabase;
import com.fkti.crudroomdatabase.data.MahasiswaDAO;
import com.fkti.crudroomdatabase.model.Mahasiswa;

/*
 * Membuat Class Database
 * Entity yang digunakan adalah Mahasiswa.class
 * Version = 1
 */
@Database(entities = {Mahasiswa.class}, version = 1)
public abstract class AppDatabase extends RoomDatabase {

    //Untuk mengakses Database menggunakan method abstract
    public abstract MahasiswaDAO mahasiswaDAO();
}
```

Pertama kita gunakan Annotation `@Database` untuk membuat class Database itu sendiri, lalu kita set Entity-entity yang akan digunakan, karena kita hanya membuat satu entity (Mahasiswa.java) lalu kita definisikan versionnya yaitu 1, jadi kita cukup menggunakan kode seperti ini : `@Database(entities = {Mahasiswa.class}, version = 1)`.

Class tersebut meng-extends Room Database lalu didalamnya kita buat method abstract dari class MahasiswaDAO bernama mahasiswaDAO, method ini hanyalah method kosong yang tidak berisi, nantinya digunakan untuk mengakses Database.



## 2) Membuat Fungsi Insert Data pada Room Database

Pada latihan ini kita akan membahas mengenai cara membuat atau menambahkan data baru pada Room Database dengan menggunakan fungsi *Create/Insert*.

Pada dasarnya Room Database hanyalah abstraction / interface yang memudahkan kita untuk membuat Database dibandingkan dengan menggunakan SQLite secara langsung. Perlu kalian ketahui, di bagian dalam layer Room Database adalah SQLite Database, jadi saya sarankan kalian sudah memahami konsep dasar mengenai SQLite.

Kita lanjutkan project sebelumnya, pada project tersebut kita hanya perlu menambahkan menu Input pada MainActivity, yaitu data-data mahasiswa yang akan dimasukkan serta menambahkan beberapa baris kode pada Komponen Room Database.

⇒ Buka project Android Studio kalian, selanjutnya buka class MahasiswaDAO.java yang sudah kalian buat pada latihan sebelumnya, pada class tersebut kita tambahkan fungsi/method insert seperti ini.

```
package com.fkti.crudroomdatabase.data;

import android.arch.persistence.room.Dao;
import android.arch.persistence.room.Insert;
import android.arch.persistence.room.OnConflictStrategy;

import com.fkti.crudroomdatabase.model.Mahasiswa;

@Dao
public interface MahasiswaDAO {

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    long insertMahasiswa(Mahasiswa mahasiswa);
}
```

Perhatikan, didalam interface tersebut kita tambahkan sebuah method bertipe data long, bernama insertMahasiswa, method ini gunakan untuk menambahkan data baru pada database, untuk itu kita perlu menggunakan Annotation @Insert.

Pada Annotation @Insert kita tambahkan parameter yaitu onConflict, parameter ini digunakan untuk menentukan Strategi saat konflik/penambahan data terjadi, disana kita memilih REPLACE, artinya jika data yang kita tambahkan sudah ada / exists, maka data yang lama / existing akan di-Replace dengan data yang baru ditambahkan.

- ⇒ Selanjutnya kita akan buat desain layout MainActivity-nya, tapi sebelum itu kita tambahkan library material design berikut ini pada dependencies.

```
implementation  
'com.android.support:design:27.1.1'
```

Kita perlu menambahkan library tersebut, karena kita akan menggunakan *TextInputLayout* **TextInputEditText**, untuk mendapatkan input dari User.

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
xmlns:android="http://schemas.android.com/apk/res/android"  
xmlns:tools="http://schemas.android.com/tools"  
android:layout_width="match_parent"  
android:layout_height="match_parent"  
android:layout_margin="16dp"  
android:orientation="vertical"  
tools:context="com.fkti.crudroomdatabase.MainActivity">  
  
    <TextView  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:gravity="center"  
        android:text="Input Data"  
        android:textSize="18sp"  
        android:textStyle="bold" />  
  
    <android.support.design.widget.TextInputLayout  
        android:id="@+id/input_layout1"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content">  
  
        <android.support.design.widget.TextInputEditText  
            android:id="@+id/nim"  
            android:layout_width="match_parent"  
            android:layout_height="wrap_content"  
            android:hint="NIM" />  
  
    </android.support.design.widget.TextInputLayout>  
  
    <android.support.design.widget.TextInputLayout  
        android:id="@+id/input_layout2"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content">  
  
        <android.support.design.widget.TextInputEditText  
            android:id="@+id/nama"
```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Nama" />
</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout
    android:id="@+id/input_layout3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <android.support.design.widget.TextInputEditText
        android:id="@+id/jurusan"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Jurusan" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout
    android:id="@+id/input_layout4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <android.support.design.widget.TextInputEditText
        android:id="@+id/tanggal_lahir"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Tanggal lahir" />

</android.support.design.widget.TextInputLayout>

<RadioGroup
    android:id="@+id/jenis_kelamin"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="4dp"
    android:layout_marginTop="4dp"
    android:orientation="horizontal">

    <RadioButton
        android:id="@+id/laki_laki"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:checked="true"
        android:text="Laki-Laki" />

    <RadioButton
        android:id="@+id/perempuan"
        android:layout_width="wrap_content"

```

```

        android:layout_height="wrap_content"
        android:layout_marginLeft="8dp"
        android:layout_marginStart="8dp"
        android:text="Perempuan" />

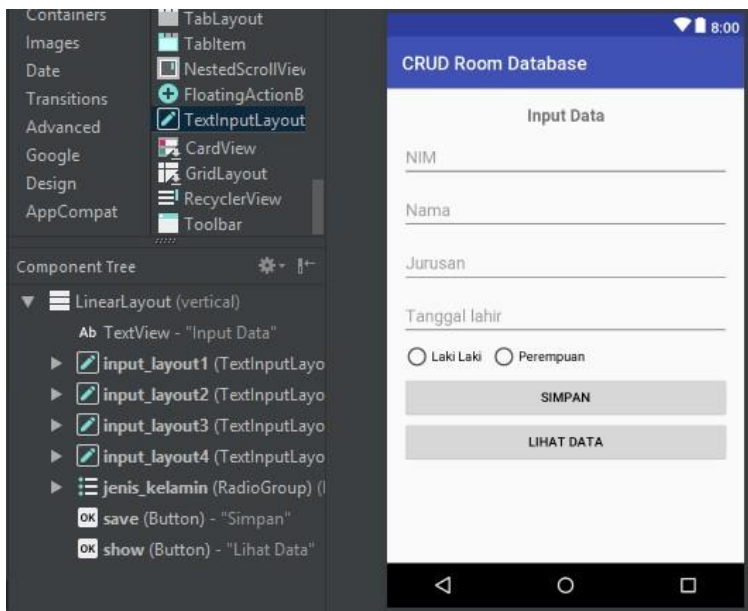
</RadioGroup>
<Button
    android:id="@+id/save"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Simpan"/>

<Button
    android:id="@+id/show"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Lihat Data"/>

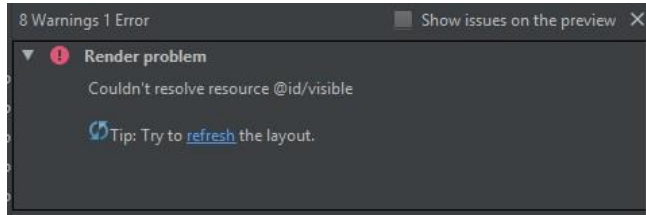
</LinearLayout>

```

## View:



Jika terdapat error seperti pada gambar berikut ini.



Untuk mengatasinya, pada folder *values*, buka file *string.xml*, lalu tambahkan baris kode berikut ini.

```
<item name="visible" type="id"/>
<item name="masked" type="id"/>
```

⇒ Berikutnya buka file *MainActivity.java*, disini kita masukan source code untuk input data kedalam Database menggunakan method insert yang sudah kita buat tadi.

```
package com.fkti.crudroomdatabase.ui;

import android.annotation.SuppressLint;
import android.arch.persistence.room.Room;
import android.content.Intent;
import android.os.AsyncTask;
import android.support.design.widget.TextInputEditText;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.Toast;

import com.fkti.crudroomdatabase.R;
import com.fkti.crudroomdatabase.data.factory.AppDatabase;
import com.fkti.crudroomdatabase.model.Mahasiswa;

public class MainActivity extends AppCompatActivity implements
View.OnClickListener{

    //Deklarasi Variable
    private TextInputEditText NIM, Nama, Jurusan,
tanggalLahir;
    private AppDatabase database;
    private Button bSimpan, bLihatData;
    private RadioButton lakiLaki, perempuan;
    private String myJenisKelamin;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        NIM = findViewById(R.id.nim);
        Nama = findViewById(R.id.nama);
        Jurusan = findViewById(R.id.jurusan);
        tanggalLahir = findViewById(R.id.tanggal_lahir);
        lakiLaki = findViewById(R.id.laki_laki);
        perempuan = findViewById(R.id.perempuan);
        bSimpan = findViewById(R.id.save);
        bSimpan.setOnClickListener(this);
        bLihatData = findViewById(R.id.show);
        bLihatData.setOnClickListener(this);

//Menentukan Jenis Kelamin pada Data Mahasiswa Baru
        if(lakiLaki.isChecked()){
            myJenisKelamin = "Laki-Laki";
        }else if (perempuan.isChecked()){
            myJenisKelamin = "Perempuan";
        }

//Inisialisasi dan memanggil Room Database
        database = Room.databaseBuilder(
            getApplicationContext(),
            AppDatabase.class,
            "dbMahasiswa") //Nama File Database yang akan
disimpan
            .build();
    }
    //Menjalankan method Insert Data
    @SuppressWarnings("StaticFieldLeak")
    private void insertData(final Mahasiswa mahasiswa){
        new AsyncTask<Void, Void, Long>() {
            @Override
            protected Long doInBackground(Void... voids) {
                //Menjalankan proses insert data
                return
database.mahasiswaDAO().insertMahasiswa(mahasiswa);
            }

            @Override
            protected void onPostExecute(Long status) {
                //Menandakan bahwa data berhasil disimpan
                Toast.makeText(MainActivity.this, "Status Row
"+status, Toast.LENGTH_SHORT).show();
            }
        }.execute();
    }

    @Override
    public void onClick(View v) {

```

```

        switch (v.getId()){
            case R.id.save:

                //Mengecek Data NIM dan Nama
                if(NIM.getText().toString().isEmpty() ||
                Nama.getText().toString().isEmpty()){
                    Toast.makeText(MainActivity.this, "NIM
                atau Nama Tidak Boleh Kosong", Toast.LENGTH_SHORT).show();
                }else {
                    //Membuat Instance/Objek Dari Class Entity Mahasisiwa
                    Mahasiswa data = new Mahasiswa();

                    //Memasukan data yang diinputkan user pada database
                    data.setNim(NIM.getText().toString());
                    data.setNama(Nama.getText().toString());
                    data.setJurusan(Jurusan.getText().toString());

                    data.setTanggalLahir(tanggalLahir.getText().toString());
                    data.setJenisKelamin(myJenisKelamin);
                    insertData(data);
                    //Mengembalikan EditText menjadi seperti semula\
                    NIM.setText("");
                    Nama.setText("");
                    Jurusan.setText("");
                    tanggalLahir.setText("");        }
                    break;

            case R.id.show:
                startActivity(new Intent(MainActivity.this,
                ReadDataActivity.class));
                break;        }
        }
    }
}

```

### Penjelasan:

Pertama kita deklarasikan beberapa variable yang dibutuhkan, selanjutnya kita buat instance/objek untuk menghubungkan dengan Room Database, serta membuat database baru bernama "dbMahasiswa".

Selanjutnya pada method `insertData()` kita memanggil method `insertMahasiswa()`, melalui `AppDatabase`, untuk menambahkan data mahasiswa baru yang diinputkan oleh user. Perlu kalian ketahui, Room Database Library adalah fungsi `async` yang tidak bisa dieksekusi pada main thread maka kita menggunakan `AsyncTask` untuk menjalankan fungsi tersebut. `AsyncTask` Memungkinkan kita menjalankan program di latar belakang layar dan mempublikasikan hasilnya.

### 3) Membuat Fungsi Read Data menggunakan Room Database

Pada latihan sebelumnya kita sudah belajar bagaimana cara menambahkan /Insert data pada Database Room. Setelah user menambahkan data, pastinya kita perlu menampilkan hasil dari semua data yang tersimpan pada Database.

Untuk itu, pada latihan ini kita akan belajar bagaimana cara membaca / Read data dari database, data tersebut akan ditampilkan pada RecyclerView dalam bentuk list vertical. Data yang ditampilkan adalah NIM dan Nama, untuk menampilkan detail data seperti Jurusan, Tanggal Lahir dan Jenis Kelamin, akan dibahas pada Tutorial selanjutnya.

Kita akan membuat Layout Activity baru, untuk menampilkan data tersimpan pada Database, serta penggunaan RecyclerViewAdapter untuk menentukan data apa saja yang akan ditampilkan pada RecyclerView.

- Pertama kita tambahkan library berikut ini kedalam dependencies, karena kita akan menggunakan CardView pada item didalam RecyclerView tersebut.

```
implementation 'com.android.support:cardview v7:27.1.1'
```

- Untuk menggunakan fungsi Read Data pada Room Database, didalam interface DAO, kita perlu membuat method baru dengan menambahkan Annotation @Query seperti berikut ini.

```
package com.fkti.crudroomdatabase.data;

import android.arch.persistence.room.Dao;
import android.arch.persistence.room.Insert;
import android.arch.persistence.room.OnConflictStrategy;
import android.arch.persistence.room.Query;

import com.fkti.crudroomdatabase.model.Mahasiswa;

@Dao
public interface MahasiswaDAO {

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    long insertMahasiswa(Mahasiswa mahasiswa);

    @Query("SELECT * FROM tMahasiswa")
    Mahasiswa[] readDataMahasiswa();
}
```



Pada Room Database kita juga dapat menggunakan Query SQLite seperti pada umumnya, pada Query tersebut kita mengambil semua data yang terdapat pada tabel bernama "tMahasiswa".

- Selanjutnya buat activity baru bernama "ReadDataActivity", activity ini digunakan untuk menampilkan data yang terdapat pada RecyclerView. Di dalam layout activity-nya, kita cukup menambahkan RecyclerView saja.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_margin="14dp"

tools:context="com.fkti.crudroomdatabase.ui.ReadDataActi
vity">

    <android.support.v7.widget.RecyclerView
        android:id="@+id/dataItem"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

</RelativeLayout>
```

Jangan lupa pada class MainActivity.java, didalam button tambahkan listener berupa Intent untuk berpindah activity.

```
case R.id.show:
startActivity(new Intent(MainActivity.this,
ReadDataActivity.class));
break;
```

Lalu buat layout baru, yaitu untuk desain RecyclerView-nya, didalam layout tersebut kita menggunakan 1 buah imageView dan 2 buah TextView.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="wrap_content">
```

```

<android.support.v7.widget.CardView
    android:id="@+id/cvMain"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_margin="8dp"
    app:cardCornerRadius="8dp"
    app:cardElevation="4dp">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="10dp"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/gambar"
            android:layout_width="50dp"
            android:layout_height="50dp"
            android:layout_marginEnd="10dp"
            android:layout_marginRight="10dp"
            app:srcCompat="@drawable/unpi" />

        <TextView
            android:id="@+id/nim"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentTop="true"
            android:layout_marginTop="3dp"
            android:layout_toEndOf="@+id/gambar"
            android:layout_toRightOf="@+id/gambar"
            android:textSize="16sp"
            android:textStyle="bold"
            tools:text="NIM" />

        <TextView
            android:id="@+id/nama"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/nim"
            android:layout_toEndOf="@+id/gambar"
            android:layout_toRightOf="@+id/gambar"
            android:text="Nama"
            android:textSize="15sp" />

    </RelativeLayout>
</android.support.v7.widget.CardView>
</RelativeLayout>

```

Jika sudah, buat sebuah adapter untuk RecyclerView-nya, buat class baru bernama "RecyclerMahasiswaAdapter", pada class ini kita mengatur bagaimana data akan tampil didalam RecyclerView, seperti Inisialisasi ArrayList, menentukan view yang akan digunakan dan mengambil data yang akan ditampilkan.

```
package com.fkti.crudroomdatabase.adapter;

import android.arch.persistence.room.Room;
import android.content.Context;
import android.support.annotation.NonNull;
import android.support.v7.widget.CardView;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import com.fkti.crudroomdatabase.R;
import com.fkti.crudroomdatabase.data.factory.AppDatabase;
import com.fkti.crudroomdatabase.model.Mahasiswa;

import java.util.ArrayList;

public class RecyclerMahasiswaAdapter extends
RecyclerView.Adapter<RecyclerMahasiswaAdapter.ViewHolder> {

    //Deklarasi Variable
    private ArrayList<Mahasiswa> daftarMahasiswa;
    private AppDatabase appDatabase;
    private Context context;

    public RecyclerMahasiswaAdapter(ArrayList<Mahasiswa>
daftarMahasiswa, Context context) {

        //Inisialisasi data yang akan digunakan
        this.daftarMahasiswa = daftarMahasiswa;
        this.context = context;
        appDatabase = Room.databaseBuilder(
            context.getApplicationContext(),
            AppDatabase.class,
            "dbMahasiswa").allowMainThreadQueries().build();
    }

    class ViewHolder extends RecyclerView.ViewHolder{

        //Deklarasi View yang akan digunakan
        private TextView Nim, Nama;
        private CardView item;
```

```

ViewHolder(View itemView) {
    super(itemView);
    Nim = itemView.findViewById(R.id.nim);
    Nama = itemView.findViewById(R.id.nama);
    item = itemView.findViewById(R.id.cvMain);
}

}

@NonNull
@Override
public ViewHolder onCreateViewHolder(@NonNull ViewGroup
parent, int viewType) {
    //Inisialisasi Layout Item untuk RecyclerView
    View v =
LayoutInflater.from(parent.getContext()).inflate(R.layout.rec
ycler_layout, parent, false);
    return new ViewHolder(v);
}

@Override
public void onBindViewHolder(@NonNull ViewHolder holder,
int position) {

    //Deklarasi Variable untuk mendapatkan data dari
Database melalui Array
    String getNIM =
daftarMahasiswa.get(position).getNim();
    String getNama =
daftarMahasiswa.get(position).getNama();

    //Menampilkan data berdasarkan posisi Item dari
RecyclerView
    holder.Nim.setText(getNIM);
    holder>Nama.setText(getNama);
}

@Override
public int getItemCount() {
    //Menghitung data / ukuran dari Array
    return daftarMahasiswa.size();
}
}

```

Selanjutnya buka class `ReadDataActivity.java` kalian, disini kita akan memsang atribut-artibut yang diperlukan `RecyclerView` seperti adapter dan `Data Array` dari `Database`.

```
package com.fkti.crudroomdatabase.ui;

import android.arch.persistence.room.Room;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;

import com.fkti.crudroomdatabase.R;
import
com.fkti.crudroomdatabase.adapter.RecyclerMahasiswaAdapter;
import com.fkti.crudroomdatabase.data.factory.AppDatabase;
import com.fkti.crudroomdatabase.model.Mahasiswa;

import java.util.ArrayList;
import java.util.Arrays;

public class ReadDataActivity extends AppCompatActivity {

    //Deklarasi Variable
    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;
    private AppDatabase database;
    private ArrayList<Mahasiswa> daftarMahasiswa;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_read_data);
        getSupportActionBar().setTitle("Daftar Mahasiswa");

        //Inisialisasi ArrayList
        daftarMahasiswa = new ArrayList<>();

        //Inisialisasi RoomDatabase
        database =
Room.databaseBuilder(getApplicationContext(),
        AppDatabase.class,
        "dbMahasiswa").allowMainThreadQueries().build();

        /*
        * Mengambil data Mahasiswa dari Database
        * lalu memasukannya ke kedalam ArrayList
        (daftarMahasiswa)
        */
    }
}
```

```

        */
daftarMahasiswa.addAll(Arrays.asList(database.mahasiswaDAO().readDataMahasiswa()));

        //Agar ukuran RecyclerView tidak berubah
recyclerView.setHasFixedSize(true);

        //Menentukan bagaimana item pada RecyclerView akan
tampil
        layoutManager = new LinearLayoutManager(this);
recyclerView.setLayoutManager(layoutManager);

        //Mamasang adapter pada RecyclerView
        adapter = new
RecyclerMahasiswaAdapter(daftarMahasiswa,
ReadDataActivity.this);
recyclerView.setAdapter(adapter);
    }
}

```

Seperti pada class adapter sebelumnya, pada didalam Room Database kita menggunakan fungsi `allowMainThreadQueries()` supaya kita dapat menjalankan fungsi Read Data di Main Thread.

#### 4) Membuat Fungsi Update Data menggunakan Room Database

Pada latihan ini kita akan belajar cara menggunakan fungsi edit data pada Room Database. Seperti SQLite pada umumnya, didalam Room Databas kita juga dapat mengubah data sebelumnya menjadi data yang baru hanya saja pada Room Database penggunaan Edit data lebih mudah dan fleksibel dibandingkan dengan SQLite secara langsung.

Untuk membuat fungsi Update Data, pada project ini pertama kita akan membuat layout activity baru untuk proses editing data, activity ini akan menerima data yang telah dipilih oleh user lalu user dapat mengubah data tersebut didalam activity tersebut lalu menyimpannya kembali.

Pertama buka kembali interface MahasiswaDAO.java, disini kita tambahkan method baru bernama updateMahasiswa().

```
package com.fkti.crudroomdatabase.data;

import android.arch.persistence.room.Dao;
import android.arch.persistence.room.Insert;
import android.arch.persistence.room.OnConflictStrategy;
import android.arch.persistence.room.Query;
import android.arch.persistence.room.Update;

import com.fkti.crudroomdatabase.model.Mahasiswa;

@Dao
public interface MahasiswaDAO {

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    long insertMahasiswa(Mahasiswa mahasiswa);

    @Query("SELECT * FROM tMahasiswa")
    Mahasiswa[] readDataMahasiswa();

    @Update
    int updateMahasiswa(Mahasiswa mahasiswa);
}
```

Pada method tersebut kita tambahkan Annotation @Update bawaan dari library Room Database, method ini berfungsi untuk mengedit data didalam Database.

Jika sudah, selanjutnya buat activity baru bernama EditActivity, activity ini digunakan sebagai form edit dari data yang akan diubah oleh user.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:layout_margin="16dp"
  android:orientation="vertical"
  tools:context="com.fkti.crudroomdatabase.ui.EditActivity">

  <TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="Edit Data"
    android:textSize="18sp"
    android:textStyle="bold" />

  <android.support.design.widget.TextInputLayout
    android:id="@+id/input_layout2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <android.support.design.widget.TextInputEditText
      android:id="@+id/nama"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="Nama" />

  </android.support.design.widget.TextInputLayout>

  <android.support.design.widget.TextInputLayout
    android:id="@+id/input_layout3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <android.support.design.widget.TextInputEditText
      android:id="@+id/jurusan"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="Jurusan" />

  </android.support.design.widget.TextInputLayout>

  <android.support.design.widget.TextInputLayout
    android:id="@+id/input_layout4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

```



```

        <android.support.design.widget.TextInputEditText
            android:id="@+id/tanggal_lahir"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Tanggal lahir" />

    </android.support.design.widget.TextInputLayout>

    <RadioGroup
        android:id="@+id/jenis_kelamin"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="4dp"
        android:layout_marginTop="4dp"
        android:orientation="horizontal">

        <RadioButton
            android:id="@+id/laki_laki"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Laki Laki" />

        <RadioButton
            android:id="@+id/perempuan"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="8dp"
            android:layout_marginStart="8dp"
            android:text="Perempuan" />

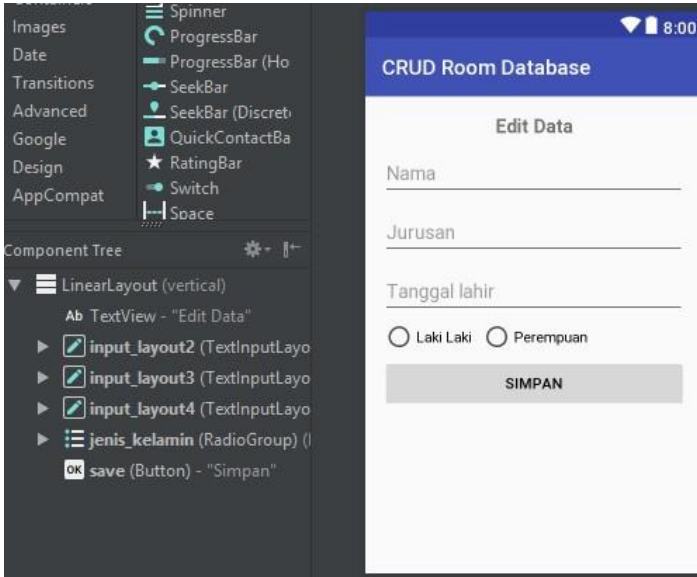
    </RadioGroup>

    <Button
        android:id="@+id/save"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Simpan"/>

</LinearLayout>

```

## View:



Tampilannya mirip dengan Activity untuk input data, hanya saja disini kita menghilangkan NIM, karena NIM sebagai Primary Key tidak akan kita ubah.

Berikutnya buka class adapter kalian bernama `RecyclerMahasiswaAdapter.java`, disini kita tambahkan method baru bernama `onEditData` serta tambahkan `onLongClickListener` pada widget `CardView`, sehingga saat user menekan salah satu item maka akan muncul menu pilihan yaitu Edit dan Delete.

```
@SuppressWarnings("RecyclerView")
@Override
public void onBindViewHolder(@NonNull ViewHolder holder, final
int position) {

    //Deklarasi Variable untuk mendapatkan data dari Database
    melalui Array
    final String getNIM =
daftarMahasiswa.get(position).getNim();
    final String getNama =
daftarMahasiswa.get(position).getNama();

    //Menampilkan data berdasarkan posisi Item dari
RecyclerView
holder.Nim.setText(getNIM);
```

```

holder>Nama.setText(getNama);

holder.item.setOnLongClickListener(new
View.OnLongClickListener() {
    @Override
    public boolean onLongClick(View v) {
        CharSequence[] menuPilihan = {"Edit", "Delete"};
        AlertDialog.Builder dialog = new
AlertDialog.Builder(v.getContext())
            .setTitle("Pilih Aksi")
            .setItems(menuPilihan, new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface
dialog, int which) {
                    switch (which){
                        case 0:
                            /*Menjalankan Perintah Edit Data
Menggunakan Bundle untuk
mengambil data yang akan Diedit
context);
                            */
                            onEditData(position,
context);
                            break;

                        case 1:
                            /* Menjalankan Perintah Delete Data
                            */
                            break;
                    }
                }
            });
        dialog.create();
        dialog.show();
        return true;
    }
});
}

//Mengirim Data yang akan diedit dari ArrayList berdasarkan
posisi item pada RecyclerView
private void onEditData(int position, Context context){
    context.startActivity(new Intent(context,
EditActivity.class).putExtra("data",
daftarMahasiswa.get(position));
    ((Activity)context).finish();
}

```

Didalam metho `onEditData()`, kita akan mengirim data mahasiswa yang dipilih oleh user menggunakan `Intent` pada `EditDataActivity`.

Lalu didalam `onLongClickListener`, disini kita menggunakan `AlertDialog.Builder` untuk menampilkan menu pilihan berupa `Edit` dan `Delete`, saat menu `Edit` di klik maka method `onEditData` akan berjalan dan menuju form activity untuk edit data.

Terakhir buka class `EdtActivity.java`, disini kita akan memanggil method `updateMahasiswa()`, untuk mengedit data yang diubah.

```
package com.fkti.crudroomdatabase.ui;

import android.annotation.SuppressLint;
import android.arch.persistence.room.Room;
import android.content.Intent;
import android.os.AsyncTask;
import android.support.design.widget.TextInputEditText;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Toast;

import com.fkti.crudroomdatabase.R;
import com.fkti.crudroomdatabase.data.factory.AppDatabase;
import com.fkti.crudroomdatabase.model.Mahasiswa;

public class EditActivity extends AppCompatActivity {

    //Deklarasi Variable
    private TextInputEditText Nama, Jurusan, tanggalLahir;
    private AppDatabase database;
    private Button bSimpan;
    private RadioButton lakiLaki, perempuan;
    private RadioGroup jenisKelamin;
    private String myJenisKelamin;
    private Mahasiswa mahasiswa;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_edit);
        Nama = findViewById(R.id.nama);
        Jurusan = findViewById(R.id.jurusan);
        tanggalLahir = findViewById(R.id.tanggal_lahir);
        jenisKelamin = findViewById(R.id.jenis_kelamin);
```

```

lakiLaki = findViewById(R.id.laki_laki);
perempuan = findViewById(R.id.perempuan);
bSimpan = findViewById(R.id.save);

database =
Room.databaseBuilder(getApplicationContext(),
                        AppDatabase.class, "dbMahasiswa").build();

//Menampilkan data mahasiswa yang akan di edit
getDataMahasiswa();

bSimpan.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v) {
        mahasiswa.setName>Nama.getText().toString());
mahasiswa.setJurusan(Jurusan.getText().toString());
mahasiswa.setTanggalLahir(tanggalLahir.getText().toString());
        mahasiswa.setJenisKelamin(myJenisKelamin);
        updateData(mahasiswa);
    }
});

//Method untuk menampilkan data mahasiswa yang akan di edit
private void getDataMahasiswa(){
    //Mendapatkan data dari Intent
    mahasiswa =
(Mahasiswa) getIntent().getSerializableExtra("data");

   >Nama.setText(mahasiswa.getName());
    Jurusan.setText(mahasiswa.getJurusan());
    tanggalLahir.setText(mahasiswa.getTanggalLahir());
    switch (mahasiswa.getJenisKelamin()){
        case "Laki-Laki":
            lakiLaki.setChecked(true);
            myJenisKelamin = "Laki-Laki";
            break;

        case "Perempuan":
            perempuan.setChecked(true);
            myJenisKelamin = "Perempuan";
            break;
    }
}

//Menjalankan method Update Data
@SuppressLint("StaticFieldLeak")

```

```

private void updateData(final Mahasiswa mahasiswa){
    new AsyncTask<Void, Void, Integer>() {
        @Override
        protected Integer doInBackground(Void... voids) {
            //Menjalankan proses insert data
            return
database.mahasiswaDAO().updateMahasiswa(mahasiswa);
        }

        @Override
        protected void onPostExecute(Integer status) {
            //Menandakan bahwa data berhasil disimpan
            Toast.makeText(EditActivity.this, "Data
Berhasil Diubah", Toast.LENGTH_SHORT).show();
            startActivity(new Intent(EditActivity.this,
ReadDataActivity.class));
            finish();
        }
    }.execute();
}

@Override
public void onBackPressed() {
    super.onBackPressed();
    startActivity(new Intent(EditActivity.this,
ReadDataActivity.class));
    finish();
}
}

```

Seperti pada saat input data, pada class ini juga kita menggunakan AsyncTask untuk menjalankan program di latar belakang layar dan mempublikasikan hasilnya. AsyncTask Ini kita gunakan untuk memantau proses saat user menyimpan data baru, jika data berhasil disimpan maka akan muncul pesan Toast.

## 5) Membuat Fungsi Delete Data menggunakan Room Database

Pada latihan sebelumnya kita sudah membahas mengenai Cara Edit Data pada Room Database, kita akan belajar bagaimana cara menghapus data pada Room Database menggunakan fungsi Delete.

Untuk membuat fungsi delete data, caranya kita cukup menambahkan Annotation `@Delete` pada interface DAO, lalu kita tambahkan method baru yang didalamnya terdapat fungsi untuk proses delete data dari Room Database.

Cara Membuat Fungsi Get Detail Data menggunakan Room Database.

Proses untuk menghapus data ini bekerja saat user menekan salah satu item pada RecyclerView menggunakan fungsi `onLongClickListener` maka akan muncul menu dengan 2 pilihan yaitu Edit dan Delete seperti pada tutorial sebelumnya. Saat user memilih menu delete pada item yang dipilih akan dihapus dari database.

Menambahkan fungsi delete untuk menghapus data didalam Room Database

Seperti sebelumnya untuk menambahkan fungsi baru, pertama kita buka interface `MahasiswaDAO.java`, disini kita tambahkan method baru bernama `deleteMahasiswa()`, dengan Annotation `@Delete` untuk menghapus data.

```
package com.fkti.crudroomdatabase.data;

import android.arch.persistence.room.Dao;
import android.arch.persistence.room.Delete;
import android.arch.persistence.room.Insert;
import android.arch.persistence.room.OnConflictStrategy;
import android.arch.persistence.room.Query;
import android.arch.persistence.room.Update;

import com.fkti.crudroomdatabase.model.Mahasiswa;

@Dao
public interface MahasiswaDAO {

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    long insertMahasiswa(Mahasiswa mahasiswa);

    @Query("SELECT * FROM tMahasiswa")
    Mahasiswa[] readDataMahasiswa();

    @Update
    int updateMahasiswa(Mahasiswa mahasiswa);
}
```





Menjalankan Perintah Edit Data Menggunakan Bundle untuk mengambil data yang akan Diedit

```
*/
    onEditData(position, context);
        break;

    case 1:
        onDeleteData(position);
        break;
    }
    });
    dialog.create();
    dialog.show();
    return true;
}
});
}

//Menghapus Data dari Room Database yang dipilih
oleh user
private void onDeleteData(int position){
appDatabase.mahasiswaDAO().deleteMahasiswa(daftarMahasiswa.get(position));
    daftarMahasiswa.remove(position);
    notifyItemRemoved(position);
    notifyItemRangeChanged(position,
daftarMahasiswa.size());
    Toast.makeText(context, "Data Berhasil Dihapus",
Toast.LENGTH_SHORT).show();
}
}
```

Perhatikan, pada method `onDeleteData()`, dengan parameter `position` untuk menentukan posisi dari item yang dipilih. Pada method tersebut pertama Room Database memanggil method `deleteMahasiswa()` dari DAO untuk menentukan data mana yang akan dihapus. Dibawahnya kita juga akan menghapus data dari `ArrayList` tersebut lalu memberikan notify, untuk melihat perubahan setelah user menghapus salah satu item pada Database melalui `RecyclerView`, lalu akan muncul pesan Toas yang menandakan data berhasil dihapus.

### Latihan 7.3. Membuat Read Detail Data pada Room Database

Pada latihan ini kita akan belajar bagaimana cara menampilkan secara detail dari data yang telah dimasukan kedalam Database. Karena aplikasi yang akan kita buat yaitu daftar Mahasiswa, jadi Detail data yang akan ditampilkan berdasarkan NIM dari Mahasiswa tersebut.

Pertama buka kembali interface MahasiswaDAO.java, disini kita tambahkan method baru bernama selectDetailMahasiswa(), pada method tersebut kita menggunakan Annotation @Query dari Room Database, pada Query tersebut kita mengambil data dari tabel mahasiswa berdasarkan NIM-nya.

```
package com.fkti.crudroomdatabase.data;

import android.arch.persistence.room.Dao;
import android.arch.persistence.room.Delete;
import android.arch.persistence.room.Insert;
import android.arch.persistence.room.OnConflictStrategy;
import android.arch.persistence.room.Query;
import android.arch.persistence.room.Update;

import com.fkti.crudroomdatabase.model.Mahasiswa;

@Dao
public interface MahasiswaDAO {

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    long insertMahasiswa(Mahasiswa mahasiswa);

    @Query("SELECT * FROM tMahasiswa")
    Mahasiswa[] readDataMahasiswa();

    @Update
    int updateMahasiswa(Mahasiswa mahasiswa);

    @Delete
    void deleteMahasiswa(Mahasiswa mahasiswa);

    @Query("SELECT * FROM tMahasiswa WHERE nim_mahasiswa = :nim LIMIT 1")
```

```
Mahasiswa selectDetailMahasiswa(String nim);  
}
```

Karena NIM kita menggunakan tipe data String, jadi parameter didalam method tersebut kita gunakan String.

Jika sudah, selanjutnya buat activity baru bernama DetailDataActivity, activity ini digunakan untuk menampilkan detail dari data yang user pilih pada RecyclerView.

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
xmlns:android="http://schemas.android.com/apk/res/android"  
  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:layout_margin="12dp"  
    android:orientation="vertical"  
  
    tools:context="com.fkti.crudroomdatabase.ui.DetailDataAc  
tivity">  
  
    <TextView  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:layout_marginBottom="8dp"  
        android:text="Detail Data"  
        android:textAlignment="center"  
        android:textSize="18sp"  
        android:textStyle="bold" />  
  
    <TextView  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:text="NIM"  
        android:textSize="15sp"  
        android:textStyle="bold" />
```

```
<EditText
    android:id="@+id/myNIM"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:ems="10"
    android:enabled="false"
    android:inputType="textPersonName" />
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Nama"
    android:textSize="15sp"
    android:textStyle="bold" />
```

```
<EditText
    android:id="@+id/myName"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:ems="10"
    android:enabled="false"
    android:inputType="textPersonName" />
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Jurusan"
    android:textSize="15sp"
    android:textStyle="bold" />
```

```
<EditText
    android:id="@+id/myJurusan"
```

```
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_marginBottom="8dp"  
android:ems="10"  
android:enabled="false"  
android:inputType="textPersonName" />
```

```
<TextView
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Tanggal Lahir"  
    android:textSize="15sp"  
    android:textStyle="bold" />
```

```
<EditText
```

```
    android:id="@+id/myTanggalLahir"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginBottom="8dp"  
    android:ems="10"  
    android:enabled="false"  
    android:inputType="textPersonName" />
```

```
<TextView
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Jenis Kelamin"  
    android:textSize="15sp"  
    android:textStyle="bold" />
```

```
<EditText
```

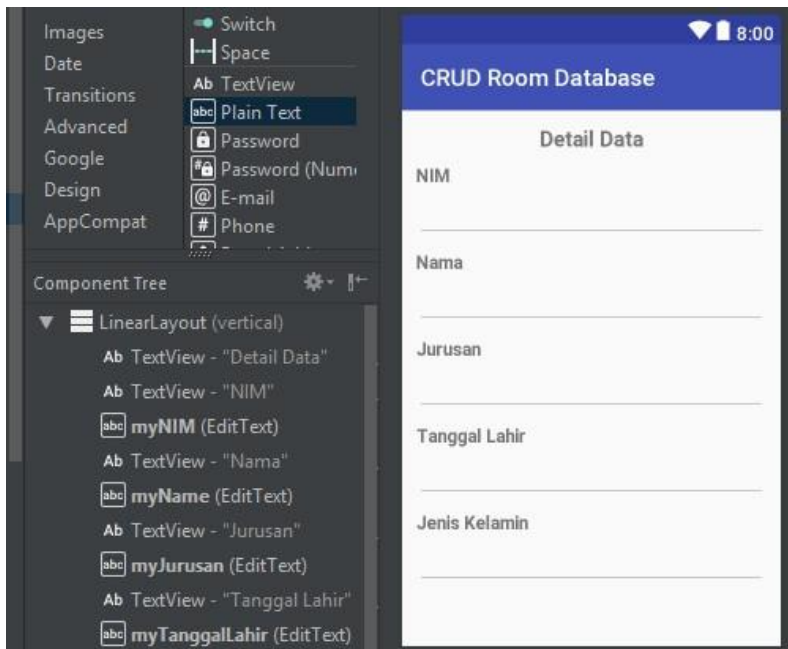
```
    android:id="@+id/myJenisKelamin"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginBottom="8dp"
```

```
android:ems="10"  
android:enabled="false"  
android:inputType="textPersonName" />
```

```
</LinearLayout>
```

Data tersebut akan ditampilkan pada EditText yang sudah kita set menjadi false, tapi kalian dapat menyesuaikan tampilannya sendiri.

View:



Seperti biasa kita akan membuat data flow untuk read detail data pada adapter dari RecyclerView, buka class RecyclerViewMahasiswaAdapter.java, pada method onBindViewHolder() tambahkan onClickListener pada widget CardView (item) kita, untuk memberi aksi saat salah satu item diklik.

```
@SuppressWarnings("RecyclerView")  
@Override  
public void onBindViewHolder(@NonNull ViewHolder holder, final  
int position) {
```

```

//Deklarasi Variable untuk mendapatkan data dari Database
melalui Array
final String getNIM =
daftarMahasiswa.get(position).getNim();
final String getNama =
daftarMahasiswa.get(position).getNama();

//Menampilkan data berdasarkan posisi Item dari
RecyclerView
holder.Nim.setText(getNIM);
holder>Nama.setText(getNama);

holder.item.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v) {
        Mahasiswa mahasiswa = appDatabase.mahasiswaDAO()
.selectDetailMahasiswa(daftarMahasiswa.get(position).getNim())
;
        context.startActivity(new Intent(context,
DetailDataActivity.class).putExtra("detail", mahasiswa));
    }
});

```

Perhatikan didalam method `onClickListener()` tersebut, kita memanggil method `selectDetailMahasiswa()` dari interface DAO, method tersebut akan mengambil data dari item yang kita klik, berdasarkan data NIM-nya, lalu mengimnya pada class `DetailDataActivity` menggunakan Intent.

Terakhir buka class `DetailDataActivity.java`, pada class ini akan menampilkan data secara detail dari item yang dipilih oleh user pada `RecyclerView`.

```

package com.fkti.crudroomdatabase.ui;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.EditText;

import com.fkti.crudroomdatabase.R;
import com.fkti.crudroomdatabase.model.Mahasiswa;

public class DetailDataActivity extends
AppCompatActivity {

```

```

    private EditText getNIM, getName, getJurusan,
    getTanggalLahir, getJenisKelamin;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detail_data);
        getNIM = findViewById(R.id.myNIM);
        getName = findViewById(R.id.myName);
        getJurusan = findViewById(R.id.myJurusan);
        getTanggalLahir =
    findViewById(R.id.myTanggalLahir);
        getJenisKelamin =
    findViewById(R.id.myJenisKelamin);

        getDetailData();
    }

    //Mendapatkan data yang akan ditampilkan secara
    detail
    private void getDetailData(){
        Mahasiswa mahasiswa =
    (Mahasiswa) getIntent().getSerializableExtra("detail");

        //Menampilkan data Mahasiswa pada activity
        if(mahasiswa != null){
            getNIM.setText(mahasiswa.getNim());
            getName.setText(mahasiswa.getNama());
            getJurusan.setText(mahasiswa.getJurusan());

            getTanggalLahir.setText(mahasiswa.getTanggalLahir());

            getJenisKelamin.setText(mahasiswa.getJenisKelamin());
        }
    }
}

```

Didalam method `getDetailData()`, kita menggunakan intent untuk menerima data yang dikirim lalu menampilkannya pada widget-widget EditText yang secara default sudah kita set menjadi false.



## Latihan 7.4. Set Up Firebase Realtime Database

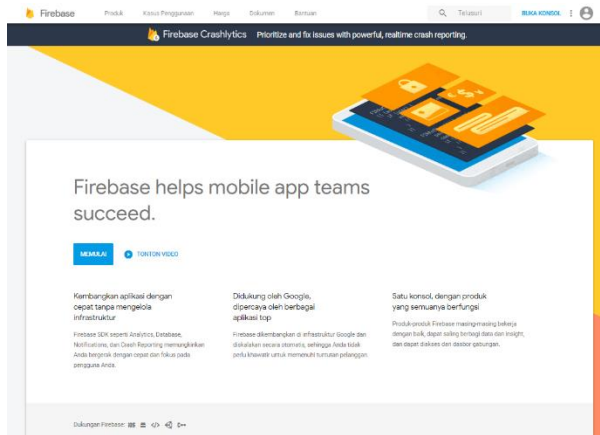
### 1) Menambahkan Firebase ke Project Android

Pada latihan ini akan belajar bagaimana cara membuat aplikasi CRUD (Create, Read, Update, Delete) dengan menggunakan Firebase Realtime Database.

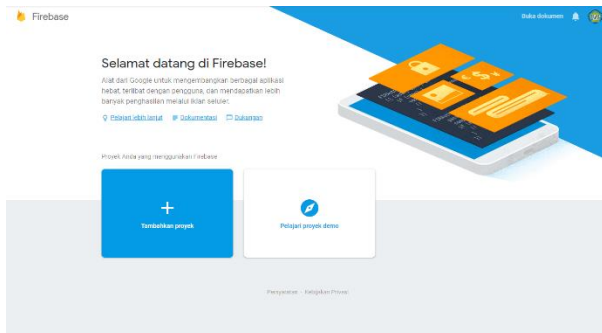
Pada latihan bagian pertama, akan menjelaskan cara menambahkan Firebase kedalam aplikasi Android kita.

Buat project baru pada Android Studio kalian, setelah itu Login/Masuk pada Firebase dengan menggunakan Alamat Gmail kalian, pada link berikut ini <https://firebase.google.com/?hl=id>

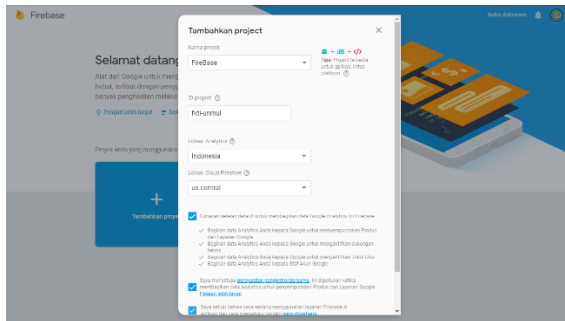
### Klik GO TO CONSOLE / BUKA KONSOL



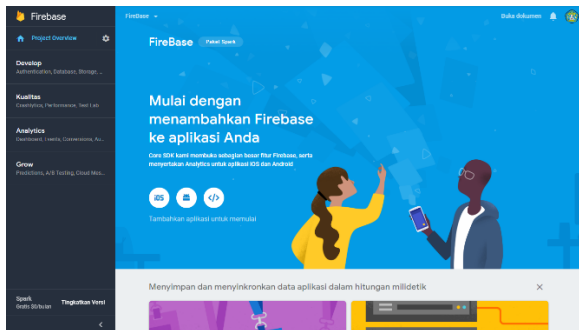
Selanjutnya tambahkan project baru kita kedalam Firebase Console, dengan mengklik Tambahkan proyek(+).



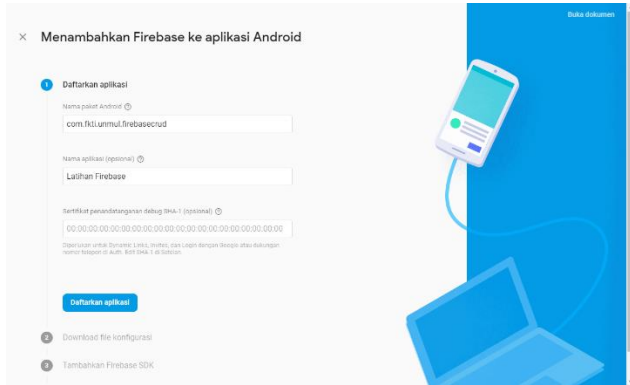
Isi nama proyek, ID project dan Negara/Wilayah kalian, setelah itu klik Buat Proyek.



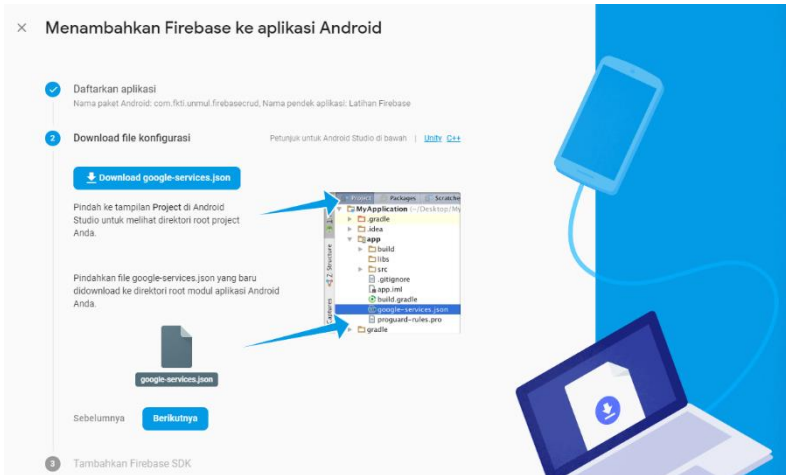
Pada halaman Dashboard Firebase, disini kalian dapat memilih platform yang akan digunakan, klik icon berlogo android, untuk menambahkan Firebase kedalam aplikasi Android.



Isi nama package (wajib), nama pendek aplikasi (jika diperlukan), lalu masukan kode SHA-1 (jika diperlukan), untuk melengkapinya kita akan menambahkan semua data tersebut kedalam Firebase.



Download File Konfigurasi google-services.json, setelah selesai, pindahkan file tersebut pada direktori app/root yang terdapat pada project aplikasi kalian.

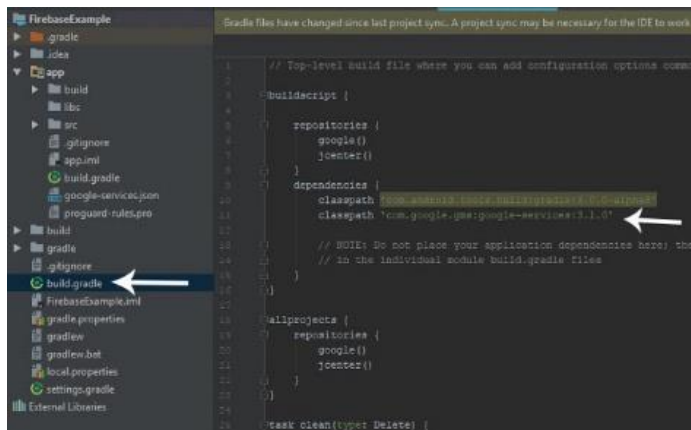


```
<uses-permission
android:name="android.permission.INTERNET"/>
```

Selanjutnya tambahkan Firebase SDK pada file build.gradle(Project-Level), masukan pluign dibawah ini.

```
'com.google.gms:google-services:3.1.0'
```

Perhatikan gambar berikut ini.



Selanjutnya buka build.gradle(app-level) lalu tambahkan library dibawah ini didalam dependencies.

```
'com.google.firebase:firebase-core:11.8.0'
```

Lalu tambahkan baris apply plugin: '**com.google.gms.google-services**' di bagian bawah dependencies. Seperti berikut ini:

```
dependencies {
    implementation 'com.google.firebase:firebase-core:11.8.0'
}
apply plugin: 'com.google.gms.google-services'
```

## 2) Membuat Sistem Autentikasi

Setelah kita Menambahkan Firebase pada Project Aplikasi, yang sudah kita bahas pada postingan sebelumnya, selanjutnya kita akan membuat sistem autentikasi, yaitu metode masuk yang digunakan oleh user agar dapat login/masuk kedalam aplikasi.

Demi keamanan pengguna, secara default, Firebase Realtime Database mempunyai aturan yaitu user diharuskan untuk terautentikasi terlebih dahulu jika ingin menulis atau membaca data didalam Databas, supaya tidak sembarang orang dapat mengakses data dari database.

Kita akan membuat sistem autentikasi menggunakan Library FirebaseUI, library tersebut menyediakan tampilan atau User Interface untuk autentikasi.

Ada banyak pilihan metode masuk dengan menggunakan FirebaseUI, tapi disini kita hanya akan menggunakan Akun Google saja untuk metode masuknya.

Buka Firebase Konsol, arahkan pada menu Authentication, disini kita pilih Metode Masuk dengan menggunakan Akun Google.



Penyedia	Status
Email/Sandi	Dinonaktifkan
Ponsel	Dinonaktifkan
Google	Diaktifkan
Play Game	Dinonaktifkan
Facebook	Dinonaktifkan
Twitter	Dinonaktifkan
GitHub	Dinonaktifkan
Anonim	Dinonaktifkan

Buka project aplikasi Android kalian, untuk autentikasi menggunakan akun Google, kita cukup menambahkan library FirebaseUI-auth dan google-play-service-auth kedalam Dependencies.

```
dependencies {  
    implementation 'com.firebaseui:firebase-ui-auth:3.2.2'  
    implementation 'com.google.android.gms:play-services-auth:11.8.0'  
}
```

Pada activity\_main.xml, disini kita akan membuat layout untuk tampilan User Interfacenya, komponen yang digunakan seperti EditText untuk Input data, dan Beberapa Button. Serta ProgressBar.

```
<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:layout_margin="12dp"  
    tools:context="com.fkti.firebaseio.ui.MainActivity">  
  
    <ProgressBar
```

```

        android:id="@+id/progress"
        style="?android:attr/progressBarStyle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="false"
        android:layout_alignParentLeft="false"
        android:layout_alignParentRight="false"
        android:layout_alignParentStart="false"
        android:layout_alignParentTop="false"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical">

    <EditText
        android:id="@+id/nim"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="NIM"
        android:inputType="textPersonName" />

    <EditText
        android:id="@+id/nama"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Masukan Nama"
        android:inputType="textPersonName" />

    <EditText
        android:id="@+id/jurusan"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Jurusan"
        android:inputType="textPersonName" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:orientation="horizontal">

        <Button
            android:id="@+id/login"

```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Login" />

<Button
    android:id="@+id/save"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="Simpan" />

<Button
    android:id="@+id/logout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="Logout" />

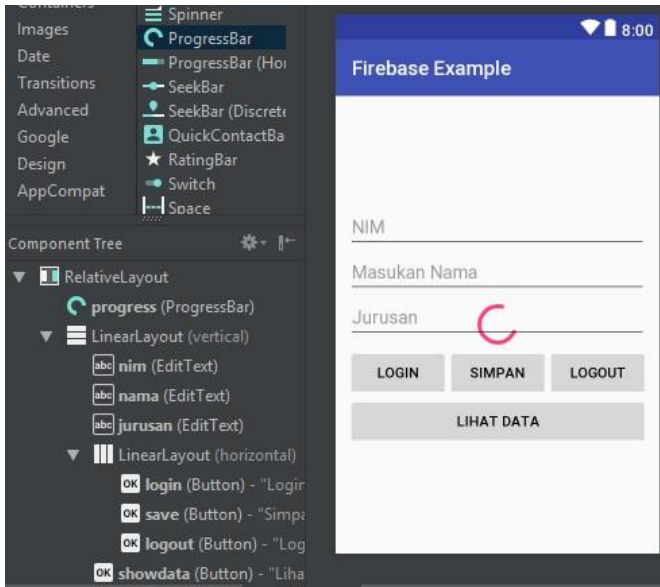
</LinearLayout>

<Button
    android:id="@+id/showdata"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Lihat Data" />

</LinearLayout>
</RelativeLayout>

```

**View:**



Screenshot\_InputDataLayout FirebaseCRUD

Buka file MainActivity.java, pada source code tersebut kita akan menambahkan FirebaseUI untuk autentikasi.

```
package com.fkti.firebasecrud.ui;

import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.firebase.ui.auth.AuthUI;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.fkti.firebasecrud.R;

import java.util.Collections;

public class MainActivity extends AppCompatActivity implements
View.OnClickListener{
```



```

//Deklarasi Variable
private ProgressBar progressBar;
private EditText NIM, Nama, Jurusan;
private FirebaseAuth auth;
private Button Logout, Simpan, Login, ShowData;

//Membuat Kode Permintaan
private int RC_SIGN_IN = 1;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    progressBar = findViewById(R.id.progress);
    progressBar.setVisibility(View.GONE);

    //Inisialisasi ID (Button)
    Logout = findViewById(R.id.logout);
    Logout.setOnClickListener(this);
    Simpan = findViewById(R.id.save);
    Simpan.setOnClickListener(this);
    Login = findViewById(R.id.login);
    Login.setOnClickListener(this);
    ShowData = findViewById(R.id.showdata);
    ShowData.setOnClickListener(this);

    auth = FirebaseAuth.getInstance(); //Mendapatkan
Instance Firebase Autentifikasi

    //Inisialisasi ID (EditText)
    NIM = findViewById(R.id.nim);
    Nama = findViewById(R.id.nama);
    Jurusan = findViewById(R.id.jurusan);

    /*
    * Mendeteksi apakah ada user yang masuk, Jika tidak,
maka setiap komponen UI akan dinonaktifkan
    * Kecuali Tombol Login. Dan jika ada user yang
terautentikasi, semua fungsi/komponen
    * didalam User Interface dapat digunakan, kecuali
tombol Logout
    */
    if(auth.getCurrentUser() == null){
        defaultUI();
    }else {
        updateUI();
    }
}

```

```

//Tampilan Default pada Activity jika user belum
terautentikasi
private void defaultUI(){
    Logout.setEnabled(false);
    Simpan.setEnabled(false);
    ShowData.setEnabled(false);
    Login.setEnabled(true);
    NIM.setEnabled(false);
    Nama.setEnabled(false);
    Jurusan.setEnabled(false);
}

//Tampilan User Interface pada Activity setelah user
Terautentikasi
private void updateUI(){
    Logout.setEnabled(true);
    Simpan.setEnabled(true);
    Login.setEnabled(false);
    ShowData.setEnabled(true);
    NIM.setEnabled(true);
    Nama.setEnabled(true);
    Jurusan.setEnabled(true);
    progressBar.setVisibility(View.GONE);
}

// Mengecek apakah ada data yang kosong, akan digunakan
pada Tutorial Selanjutnya
private boolean isEmpty(String s){
    return TextUtils.isEmpty(s);
}

protected void onActivityResult(int requestCode, int
resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    // RC_SIGN_IN adalah kode permintaan yang Anda berikan
ke startActivityForResult, saat memulai masuknya arus.
    if (requestCode == RC_SIGN_IN) {

        //Berhasil masuk
        if (resultCode == RESULT_OK) {
            Toast.makeText(MainActivity.this, "Login
Berhasil", Toast.LENGTH_SHORT).show();
            updateUI();
        }else {
            progressBar.setVisibility(View.GONE);
            Toast.makeText(MainActivity.this, "Login
Dibatalkan", Toast.LENGTH_SHORT).show();
        }
    }
}
}

```

```

@Override
public void onClick(View v) {
    switch (v.getId()){
        case R.id.login:
            // Statement program untuk login/masuk
            startActivityForResult(AuthUI.getInstance()
                .createSignInIntentBuilder()

                //Memilih Provider atau Method
masuk yang akan kita gunakan

                .setAvailableProviders(Collections.singletonList(new
AuthUI.IdpConfig.GoogleBuilder().build()))
                .setIsSmartLockEnabled(false)
                .build(),
                RC_SIGN_IN);
            progressBar.setVisibility(View.VISIBLE);
            break;

        case R.id.save:
            /*
            Digunakan untuk mendapatkan referensi dan meyimpan data pada
Database
            Akan dibahas pada Tutorial Berikutnya, Mengenai Fungsi
Create
            */
            break;

        case R.id.logout:
            // Statement program untuk logout/keluar
            AuthUI.getInstance()
                .signOut(this)
                .addOnCompleteListener(new
OnCompleteListener() {
                    @Override
                    public void onComplete(@NonNull
Task task) {

                        Toast.makeText(MainActivity.this, "Logout Berhasil",
Toast.LENGTH_SHORT).show();

                            finish();
                        }
                    });
            break;

        case R.id.showdata:
            // Digunakan untuk melihat data yang sudah
tersimpan didalam Database
            // Akan digunakan pada Tutorial Berikutnya,
mengenai penggunaan Fungsi Read

```

```
        break;
    }
}
```

Didalam method `onCreate()`, Firebase mengecek apakah ada user yang terautentikasi, jika tidak ada atau null, maka akan mengarah pada activity yang disediakan FirebaseUI, didalamnya kita menggunakan sebuah provider `AuthUI.IdpConfig.GoogleBuilder().build()`, yang memungkinkan user untuk login menggunakan akun Google.

Method `setAvailableProviders()` digunakan untuk menentukan dan menambahkan metode autentikasi yang digunakan. disini kita hanya menggunakan satu metode masuk, jadi kita gunakan Class dan fungsi `Collections.singletonList()`.

Smart Lock menyimpan kredensial pengguna dan secara otomatis memasukkan pengguna ke aplikasi, tapi pada contoh kali ini kita menonaktifkan fitur SmartLock.

Dan yang terakhir, yaitu `AuthUI.getInstance().signOut(this)`, pada saat Button Logout diklik, digunakan untuk proses keluar/logout.

### 3) Membuat Fungsi Create

Setelah kita Membuat Fungsi Autentikasi untuk metode masuk user kedalam aplikasi, dengan menggunakan Akun Google, selanjutnya kita akan membuat fungsi Create untuk meyimpan data kedalam Database, data tersebut dapat berupa String, Integer, Boolean, dll.

Firebase Realtime Database adalah database NonSQL, yang menggunakan format data JSON, berupa key-value. Dengan menggunakan Firebase, walaupun klien tidak terkoneksi internet, klien dapat menerima setiap perubahan data yang ada, saat klien terkoneksi kembali dengan Internet, peristiwa realtime akan terus berlangsung.

Aplikasi yang akan kita buat yaitu, aplikasi Data Mahasiswa, jadi didalam aplikasi tersebut kita berperan sebagai Admin yang mengelola pendataan Mahasiswa, fungsi pertama yang akan kita buat yaitu fungsi Create, yang memungkinkan Admin/User dapat meyimpan inputannya pada Database. Data yang akan kita Inputkan berupa NIM, Nama dan Jurusan.

Pertama tambahkan library berikut ini kedalam dependencies kalian.

```
implementation 'com.google.firebase:firebase-  
database:11.8.0'
```

Selanjutnya kita akan membuat class model, bernama data\_mahasiswa, class ini digunakan untuk menentukan atribut (seperti NIM, Nama, Jurusan) untuk masing-masing data mahasiswa yang akan disimpan.

Masing-masing atribut menggunakan method Getter dan Setter, yang nantinya digunakan untuk mengambil dan menyimpan data dari Database. Dan juga terdapat variable key, yang nantinya digunakan untuk proses Edit/Update dan Delete

```
package com.fkti.firebaseio.models;  
  
public class data_mahasiswa {  
  
    //Deklarasi Variable  
    private String nim;  
    private String nama;  
    private String jurusan;  
    private String key;  
  
    public String getKey() {  
        return key;  
    }  
  
    public void setKey(String key) {  
        this.key = key;  
    }  
  
    public String getNim() {  
        return nim;  
    }  
  
    public void setNim(String nim) {  
        this.nim = nim;  
    }  
  
    public String getNama() {  
        return nama;  
    }  
  
    public void setNama(String nama) {  
        this.nama = nama;  
    }  
}
```

```

public String getJurusan() {
    return jurusan;
}

public void setJurusan(String jurusan) {
    this.jurusan = jurusan;
}

//Membuat Konstuktur kosong untuk membaca data
snapshot
public data_mahasiswa(){
}

//Konstruktor dengan beberapa parameter, untuk
mendapatkan Input Data dari User
public data_mahasiswa(String nim, String nama,
String jurusan) {
    this.nim = nim;
    this.nama = nama;
    this.jurusan = jurusan;
}
}

```

Buka kembali class MainActivity.java kalian, pada method onClick(), ubah dan tambahkan baris kode berikut ini dialam case R.id. save, yang digunakan untuk meyimpan data pada Database. dan jangan lupa untuk mengimport package-packagenya.

```

case R.id.save:
    //Mendapatkan UserID dari pengguna yang
    Terautentikasi
    String getUserID = auth.getCurrentUser().getUid();

    //Mendapatkan Instance dari Database
    FirebaseDatabase database =
    FirebaseDatabase.getInstance();
    DatabaseReference getReference;

    //Menyimpan Data yang diinputkan User kedalam
    Variable
    String getNIM = NIM.getText().toString();
    String getNama = Nama.getText().toString();

```

```

String getJurusan = Jurusan.getText().toString();

getReference = database.getReference(); //
Mendapatkan Referensi dari Database

// Mengecek apakah ada data yang kosong
if(isEmpty(getNIM) && isEmpty(getNama) &&
isEmpty(getJurusan)){
    //Jika Ada, maka akan menampilkan pesan singkat
    seperti berikut ini.
    Toast.makeText(MainActivity.this, "Data tidak
    boleh ada yang kosong", Toast.LENGTH_SHORT).show();
}else {
    /*
    Jika Tidak, maka data dapat diproses dan
    meyimpannya pada Database
    Menyimpan data referensi pada Database
    berdasarkan User ID dari masing-masing Akun
    */

getReference.child("Admin").child(getUserID).child("Maha
siswa").push()
    .setValue(new data_mahasiswa(getNIM,
getNama, getJurusan)
    .addOnSuccessListener(this, new
OnSuccessListener() {
        @Override
        public void onSuccess(Object o) {
            //Peristiwa ini terjadi saat
            user berhasil menyimpan datanya kedalam Database
            NIM.setText("");
            Nama.setText("");
            Jurusan.setText("");

Toast.makeText(MainActivity.this, "Data Tersimpan",
Toast.LENGTH_SHORT).show();
        }
    });
}
break;

```

Pada source code tersebut, kita membuat referensi database dengan 3 buah Child, yang pertama yaitu child("Admin"), child(getUserID) digunakan untuk

membuat penyimpanan baru berdasarkan ID dari pengguna yang masuk, agar setiap akun yang terautentikasi mempunyai datanya masing-masing dan yang terakhir yaitu child("Mahasiswa") sebagai lokasi data-data mahasiswa yang tersimpan, fungsi push() digunakan untuk membuat kode unik, dari setiap data yang masuk.

#### 4) Membuat Fungsi Read

Kita sudah latihan mengenai penggunaan fungsi Create untuk menambahkan data kedalam database, setelah data didalam firebase tersebut tersimpan, selanjutnya kita akan membuat fungsi Read untuk membaca dan mengambil data dari Database tersebut dan Menampilkannya pada RecyclerView.

Pada latihan ini kita akan menggunakan RecyclerView untuk menampilkan datanya, jadi saya harap kalian sudah memahami penggunaan RecyclerView pada Android Studio, sebelumnya kita sudah menambahkan beberapa data kedalam Database.

Data yang tersimpan menggunakan format JSON yang meliputi key-value, data ini akan kita tampilkan pada RecyclerView secara Realtime, itu artinya jika kita ubah data tersebut langsung dari servernya, secara otomatis data yang ditampilkan pada aplikasi juga ikut berubah.

Tambahkan beberapa library berikut ini kedalam dependencies kalian, disini kita menggunakan Library Material Design untuk RecyclerView.

```
implementation 'com.android.support:appcompat-v7:27.0.2'  
implementation 'com.android.support:design:27.0.2'  
implementation 'com.android.support:recyclerview-v7:27.0.2'  
implementation 'com.android.support:support-v4:27.0.2'
```

Karena menggunakan RecyclerView, pertama kita harus membuat layout untuk mengatur tampilan per-itemnya, klik kanan pada direktori res/layout, buat layout baru, berinam nama view\_design.xml.

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
xmlns:android="http://schemas.android.com/apk/res/android"  
xmlns:app="http://schemas.android.com/apk/res-auto"  
android:id="@+id/list_item"  
android:layout_width="match_parent"
```



```

android:layout_height="wrap_content"
android:background="#FFFFFF"
android:gravity="center"
android:orientation="horizontal">

<ImageView
    android:layout_width="match_parent"
    android:layout_height="90dp"
    android:layout_weight="2.5"
    app:srcCompat="@drawable/graduation_cap" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="6dp"
    android:layout_weight="1"
    android:orientation="vertical"
    android:layout_marginStart="6dp">

    <TextView
        android:id="@+id/nim"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="4dp"
        android:text="NIM"
        android:textSize="15sp"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/nama"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="4dp"
        android:text="Nama" />

    <TextView
        android:id="@+id/jurusan"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="4dp"
        android:text="Jurusan:" />

</LinearLayout>
</LinearLayout>

```

Didalam layout tersebut kita menggunakan 1 buah ImageView dan 3 Buah TextView, untuk gambar saya sudah mempersiapkannya sendiri, atau bisa disesuaikan dengan keinginan kalian. Pada linearLayout yang ber-id "list\_item", nantinya layout tersebut digunakan untuk berinteraksi dengan user, saat klik.

Selanjutnya buat class bernama RecyclerViewAdapter, class ini digunakan untuk mengatur dan mengolah data yang ingin ditampilkan pada RecyclerView.

```
package com.fkti.firebasecrud.adapter;

import android.annotation.SuppressLint;
import android.content.Context;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.LinearLayout;
import android.widget.TextView;

import com.fkti.firebasecrud.R;
import com.fkti.firebasecrud.models.data_mahasiswa;

import java.util.ArrayList;

//Class Adapter ini Digunakan Untuk Mengatur Bagaimana
Data akan Ditampilkan
public class RecyclerViewAdapter extends
RecyclerView.Adapter<RecyclerViewAdapter.ViewHolder>{

    //Deklarasi Variable
    private ArrayList<data_mahasiswa> listMahasiswa;
    private Context context;

    //Membuat Konstruktor, untuk menerima input dari
Database
    public RecyclerViewAdapter(ArrayList<data_mahasiswa>
listMahasiswa, Context context) {
        this.listMahasiswa = listMahasiswa;
        this.context = context;
    }

    //ViewHolder Digunakan Untuk Menyimpan Referensi
Dari View-View
    class ViewHolder extends RecyclerView.ViewHolder{
```

```

private TextView NIM, Nama, Jurusan;
private LinearLayout ListItem;

ViewHolder(View itemView) {
    super(itemView);
    //Menginisialisasi View-View yang terpasang
pada layout RecyclerView kita
    NIM = itemView.findViewById(R.id.nim);
    Nama = itemView.findViewById(R.id.nama);
    Jurusan =
itemView.findViewById(R.id.jurusan);
    ListItem =
itemView.findViewById(R.id.list_item);
}

@Override
public ViewHolder onCreateViewHolder(ViewGroup
parent, int viewType) {
    //Membuat View untuk Menyiapkan dan Memasang
Layout yang Akan digunakan pada RecyclerView
    View V =
LayoutInflater.from(parent.getContext()).inflate(R.layou
t.view_design, parent, false);
    return new ViewHolder(V);
}

@SuppressLint("SetTextI18n")
@Override
public void onBindViewHolder(ViewHolder holder,
final int position) {
    //Menggambil Nilai/Value yang terdapat pada
RecyclerView berdasarkan Posisi Tertentu
    final String NIM =
listMahasiswa.get(position).getNim();
    final String Nama =
listMahasiswa.get(position).getNama();
    final String Jurusan =
listMahasiswa.get(position).getJurusan();

    //Memasukan Nilai/Value kedalam View (TextView:
NIM, Nama, Jurusan)
    holder.NIM.setText("NIM: "+NIM);
    holder>Nama.setText("Nama: "+Nama);

```

```

        holder.Jurusan.setText("Jurusan: "+Jurusan);

        holder.ListItem.setOnLongClickListener(new
View.OnLongClickListener() {
            @Override
            public boolean onLongClick(View v) {
                /*
                    Kodingan untuk membuat fungsi Edit dan
Delete,
                    yang akan dibahas pada Tutorial
Berikutnya.
                */
                return true;
            }
        });
    }
    @Override
    public int getItemCount() {
        //Menghitung Ukuran/Jumlah Data Yang Akan
Ditampilkan Pada RecyclerView
        return listMahasiswa.size();
    }
}

```

Setelah pengaturan RecyclerView selesai, berikutnya kita buat activity baru, bernama **"MyListData"**, lalu kita tambahkan RecyclerView kedalam activity tersebut, yang nantinya digunakan untuk menampilkan data yang terdapat pada Database.

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_margin="8dp"
android:background="#FFFFFF"
tools:context="com.fkti.firebaseio.ui.MyListData">

<android.support.v7.widget.RecyclerView
    android:id="@+id/datalist"
    android:layout_width="match_parent"

```

```
        android:layout_height="match_parent"
        android:scrollbars="vertical" />
</RelativeLayout>
```

Untuk menampilkan Garis bawah (Underline) pada setiap item didalam RecyclerView, Buat resource baru bernama line.xml pada direktori res > drawable > New > Drawable resource file, jika sudah, masukan kode xml berikut ini.

```
<?xml version="1.0" encoding="utf-8"?>
<shape
xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <solid android:color="#FFD8D8D8"/>
    <size android:height="0.5dp"/>
</shape>
```

Jangan lupa, didalam class MainActivity.java, pada method onClick(), tambahkan baris kode berikut ini, didalam case R.id.showdata.

```
case R.id.showdata:
    startActivity(new Intent(MainActivity.this,
MyListData.class));
    break;
```

Buka class MyListData.java, pada class tersebut kita akan memasukan source code untuk menampilkan Data dari Firebase kedalam RecyclerView.

```
package com.fkti.firebaseio.ui;

import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.DividerItemDecoration;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.widget.Toast;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
```

```

import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.fkti.firebaseio.R;
import
com.fkti.firebaseio.adapter.RecyclerViewAdapter;
import com.fkti.firebaseio.models.data_mahasiswa;

import java.util.ArrayList;

public class MyListData extends AppCompatActivity {

    //Deklarasi Variable untuk RecyclerView
    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private RecyclerView.LayoutManager layoutManager;

    //Deklarasi Variable Database Reference dan
    ArrayList dengan Parameter Class Model kita.
    private DatabaseReference reference;
    private ArrayList<data_mahasiswa> dataMahasiswa;

    private FirebaseAuth auth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_my_list_data);
        recyclerView = findViewById(R.id.datalist);
        getSupportActionBar().setTitle("Data
Mahasiswa");
        auth = FirebaseAuth.getInstance();
        MyRecyclerView();
        GetData();
    }

    //Berisi baris kode untuk mengambil data dari
    Database dan menampilkannya kedalam Adapter
    private void GetData(){
        Toast.makeText(getApplicationContext(),"Mohon
Tunggu Sebentar...", Toast.LENGTH_LONG).show();
        //Mendapatkan Referensi Database
        reference =
FirebaseDatabase.getInstance().getReference();

```

```

reference.child("Admin").child(auth.getUid()).child("Mahasiswa")
        .addValueEventListener(new
ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot
dataSnapshot) {
        //Inisialisasi ArrayList
        dataMahasiswa = new ArrayList<>();

        for (DataSnapshot snapshot :
dataSnapshot.getChildren()){
            //Mapping data pada DataSnapshot ke
dalam objek mahasiswa
            data_mahasiswa mahasiswa =
snapshot.getValue(data_mahasiswa.class);

            //Mengambil Primary Key, digunakan
untuk proses Update dan Delete
            mahasiswa.setKey(snapshot.getKey());
            dataMahasiswa.add(mahasiswa);
        }

        //Inisialisasi Adapter dan data
Mahasiswa dalam bentuk Array
        adapter = new
RecyclerViewAdapter(dataMahasiswa, MyListData.this);

        //Memasang Adapter pada RecyclerView
recyclerView.setAdapter(adapter);

Toast.makeText(getApplicationContext(),"Data Berhasil
Dimuat", Toast.LENGTH_LONG).show();
    }

    @Override
    public void onCancelled(DatabaseError
databaseError) {
        /*
        Kode ini akan dijalankan ketika ada
error dan
        pengambilan data error tersebut lalu
memprint error nya

```

```

        ke LogCat
        */

Toast.makeText(getApplicationContext(), "Data Gagal
Dimuat", Toast.LENGTH_LONG).show();
        Log.e("MyListActivity",
databaseError.getDetails()+"
"+databaseError.getMessage());
    }
});
}

//Methode yang berisi kumpulan baris kode untuk
mengatur RecyclerView
private void MyRecyclerView(){
    //Menggunakan Layout Manager, Dan Membuat List
Secara Vertical
    layoutManager = new LinearLayoutManager(this);
    recyclerView.setLayoutManager(layoutManager);
    recyclerView.setHasFixedSize(true);

    //Membuat Underline pada Setiap Item Didalam
List
    DividerItemDecoration itemDecoration = new
DividerItemDecoration(getApplicationContext(),
DividerItemDecoration.VERTICAL);

    itemDecoration.setDrawable(ContextCompat.getDrawable(get
ApplicationContext(), R.drawable.line));
    recyclerView.addItemDecoration(itemDecoration);
}
}

```

Dengan menggunakan for-each, akan me-mapping data kedalam objek mahasiswa, lalu pada fungsi setKey(), digunakan untuk mengambil semua key, berupa kode unik yang dihasilkan oleh push(), nantinya digunakan untuk proses Update dan Delete.

Semua data yang ter-mapping tersebut dimasukan kedalam ArrayList dataMahasiswa, dan terakhir dimasukan kedalam Adapter, untuk ditampilkan pada RecyclerView.



## 5) Membuat Fungsi Update Data

Sebelumnya kita sudah belajar mengenai cara penerapan fungsi Read untuk membaca data dan menampilkannya pada RecyclerView, selanjutnya kita akan menambahkan fungsi update untuk mengubah data yang sudah dimasukan, dan mengubahnya menjadi data yang baru.

Disini kita akan menjelaskan sedikit cara kerja proses update tersebut, pertama kita menentukan item/data mana yang akan dipilih berdasarkan posisi tertentu dari RecyclerView, setiap data didalam list tersebut mempunyai kode uniknya masing-masing, kode unik inilah yang digunakan untuk mengambil data yang akan di update oleh user, lalu menyimpannya kembali kedalam database.

Pertama kita akan membuat class dan activity baru, bernama "updateData" activity ini digunakan untuk memproses data yang ingin kita ubah.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_margin="8dp"
android:orientation="vertical"
tools:context="com.fkti.firebaseio.ui.updateData">

<EditText
    android:id="@+id/new_nim"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="NIM Baru"
    android:inputType="textPersonName" />
<EditText
    android:id="@+id/new_nama"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Nama Baru"
    android:inputType="textPersonName" />
<EditText
    android:id="@+id/new_jurusan"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Jurusan Baru"
    android:inputType="textPersonName" />
<Button
```

```

        android:id="@+id/update"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Update" />

```

```
</LinearLayout>
```

View :



Setelah kita membuat activity tersebut, selanjutnya kita beralih pada class RecyclerViewAdapter, didalam method onLongClickListener(), kita tambahkan baris kode berikut ini.

```

//Menampilkan Menu Update dan Delete saat user melakukan
long klik pada salah satu item
holder.ListItem.setOnLongClickListener(new
View.OnLongClickListener() {
    @Override
    public boolean onLongClick(final View view) {
        final String[] action = {"Update", "Delete"};
        AlertDialog.Builder alert = new
AlertDialog.Builder(view.getContext());
        alert.setItems(action, new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog,
int i) {
                switch (i){
                    case 0:
                        /*

```

```

        Berpindah Activity pada
halaman layout updateData
        dan mengambil data pada
listMahasiswa, berdasarkan posisinya
        untuk dikirim pada activity
selanjutnya
        */
        Bundle bundle = new Bundle();
        bundle.putString("dataNIM",
listMahasiswa.get(position).getNim());
        bundle.putString("dataNama",
listMahasiswa.get(position).getNama());
        bundle.putString("dataJurusan",
listMahasiswa.get(position).getJurusan());

        bundle.putString("getPrimaryKey",
listMahasiswa.get(position).getKey());
        Intent intent = new
Intent(view.getContext(), updateData.class);
        intent.putExtras(bundle);
        context.startActivity(intent);
        break;
        case 1:
//Pembahasan selanjutnya
mengenai fungsi Delete
        break;
    }
}
});
alert.create();
alert.show();
return true;
}
});

```

Saat user menekan salah satu item dari list, maka akan muncul menu Update dan Delete, untuk menu delete akan dibahas pada tutorial berikutnya. Saat user memilih menu Update, maka disana kita menggunakan Class Bundle untuk mengambil dan meyimpan data dari mahasiswa yang kita pilih.

Data yang disimpan pada Bundle berupa NIM, Nama, Jurusan dan Key (kode unik yang dihasilkan Push), key ini digunakan untuk menentukan data mana yang akan diubah didalam database, karena setiap data yang tampil

didalam RecyclerView, mempunyai PrimaryKeynya masing-masing, yang dihasilkan oleh push() pada saat user mengCreate data.

Buka class updateData.java, disini kita akan menambahkan source code untuk memproses update, berdasarkan data yang telah ditentukan.

```
package com.fkti.firebaseio.ui;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.fkti.firebaseio.R;
import com.fkti.firebaseio.models.data_mahasiswa;

public class updateData extends AppCompatActivity {

    //Deklarasi Variable
    private EditText nimBaru, namaBaru, jurusanBaru;
    private Button update;
    private DatabaseReference database;
    private FirebaseAuth auth;
    private String cekNIM, cekNama, cekJurusan;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_update_data);
        getSupportActionBar().setTitle("Update Data");
        nimBaru = findViewById(R.id.new_nim);
        namaBaru = findViewById(R.id.new_nama);
        jurusanBaru = findViewById(R.id.new_jurusan);
        update = findViewById(R.id.update);

        //Mendapatkan Instance autentikasi dan Referensi
        dari Database
        auth = FirebaseAuth.getInstance();
```

```

        database =
        FirebaseDatabase.getInstance().getReference();
        getData();
        update.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {

                //Mendapatkan Data Mahasiswa yang akan
dicek

                cekNIM = nimBaru.getText().toString();
                cekNama = namaBaru.getText().toString();
                cekJurusan =
jurusanBaru.getText().toString();

                //Mengecek agar tidak ada data yang
kosong, saat proses update
                if(isEmpty(cekNIM) || isEmpty(cekNama)
|| isEmpty(cekJurusan)){
                    Toast.makeText(updateData.this,
                    "Data tidak boleh ada yang kosong",
                    Toast.LENGTH_SHORT).show();
                }else {
                    /*
                    Menjalankan proses update data.
                    Method Setter digunakan untuk
mendapatkan data baru yang diinputkan User.
                    */
                    data_mahasiswa setMahasiswa = new
data_mahasiswa();

                    setMahasiswa.setNim(nimBaru.getText().toString());

                    setMahasiswa.setNama(namaBaru.getText().toString());

                    setMahasiswa.setJurusan(jurusanBaru.getText().toString()
);

                    updateMahasiswa(setMahasiswa);
                }
            }
        });

        // Mengecek apakah ada data yang kosong, sebelum
diupdate

```

```

private boolean isEmpty(String s){
    return TextUtils.isEmpty(s);
}

//Menampilkan data yang akan di update
private void getData(){
    //Menampilkan data dari item yang dipilih
    sebelumnya
    final String getNIM =
    getIntent().getExtras().getString("dataNIM");
    final String getNama =
    getIntent().getExtras().getString("dataNama");
    final String getJurusan =
    getIntent().getExtras().getString("dataJurusan");
    nimBaru.setText(getNIM);
    namaBaru.setText(getNama);
    jurusanBaru.setText(getJurusan);
}

//Proses Update data yang sudah ditentukan
private void updateMahasiswa(data_mahasiswa
mahasiswa){
    String userID = auth.getUserId();
    String getKey =
    getIntent().getExtras().getString("getPrimaryKey");
    database.child("Admin")
        .child(userID)
        .child("Mahasiswa")
        .child(getKey)
        .setValue(mahasiswa)
        .addOnSuccessListener(new
OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void aVoid) {
                nimBaru.setText("");
                namaBaru.setText("");
                jurusanBaru.setText("");
                Toast.makeText(updateData.this,
                "Data Berhasil diubah", Toast.LENGTH_SHORT).show();
                finish();
            }
        });
}
}
}

```

Untuk penjelasan singkatnya, kalian perhatikan method `updateMahasiswa()`, dalam method tersebut, pertama kita mengambil `UserID`, yang didapat dari akun pengguna, dan `PrimaryKey`, yaitu kode unik yang dihasilkan `push()` pada saat `create` data.

Didalam database, pertama kita masuk kedalam lokasi Admin, lalu masuk pada lokasi `userID` dari pengguna, setelah itu lokasi dari `chid` mahasiswa, didalamnya terdapat kumpulan data mahasiswa yang sudah dimasukan oleh user, untuk menentukan data mana yang akan dipilih, kita menggunakan `PrimaryKey`. Dan yang terakhir kita memasukan data mahasiswa baru.

## 6) Membuat Fungsi Delete Data

Kita sudah melalui beberapa tahap pembuatan aplikasi CRUD dengan menggunakan `Firestore Realtime Database`, berikut ini merupakan materi terakhir dalam pembuatan aplikasi CRUD, disini kita akan membuat fungsi `Delete` untuk menghapus data Mahasiswa yang dipilih oleh user.

Konsep pada fitur `Delete` hampir sama dengan fitur `Update`, yang telah kita bahas pada tutorial sebelumnya, yaitu dengan menggunakan fungsi `onClick`, untuk menampilkan menu delete pada item didalam list.

Cara kerja dari fitur delete tersebut, pertama kita membuat sebuah interface untuk menghubungkan adapter dengan class **`MyListData`**.java, karena pada adapter tersebut tidak mempunyai referensi database, jadi kita membuat sebuah interface untuk menghubungkannya.

Pertama buka class `RecyclerViewAdapter`, pada class tersebut kita tambahkan interface, yang digunakan untuk menjembatani data yang akan dikirim dari Adapter pada class **`MyListData`**.

```
//Membuat Interfece
public interface dataListener{
    void onDeleteData(data_mahasiswa data, int
position);
}

//Deklarasi objek dari Interfece
dataListener listener;

//Membuat Konstruktor, untuk menerima input dari
Database
public RecyclerViewAdapter(ArrayList listMahasiswa,
Context context) {
    this.listMahasiswa = listMahasiswa;
}
```

```

    this.context = context;
    listener = (MyListData)context;
}

```

Pada class RecyclerViewAdaper, kita menambahkan interface dan menambahkan inisialisasi variabelnya pada constructor, untuk mendapatkan context dari class MyListData.

Jika terjadi error, itu dikarenakan kita belum mengimplementasikan method onDeleteData() pada class MyListData, yang menjadi di contextnya, Solusinya, tekan (alt + enter) pada bagian yang error, lalu implementasikan method tersebut, seperti pada gambar berikut ini.

```

//Membuat Konstruktork, untuk menerima input dari Database
public RecyclerViewAdapter(ArrayList<data_mahasiswa> listMahasiswa, Context context) {
    this.listMahasiswa = listMahasiswa;
    this.context = context;
    listener = (MyListData)context;
}

```

- Cast to 'com.wildan.firebaseio.adapter.RecyclerViewAdapter.dataListener'
- Change field 'listener' type to 'com.wildan.firebaseio.ui.MyListData'
- Make 'MyListData' implement 'com.wildan.firebaseio.adapter.RecyclerViewAdapter.dataListener'
- Migrate 'listener' type to 'com.wildan.firebaseio.ui.MyListData'

Setelah kalian selesai mengimplementasi method onDeleteData(), selanjutnya buka kembali class RecyclerViewAdapter, tambahkan baris kode berikut ini didalam onBindViewHolder(), pada method onLongClickListener(), seperti berikut ini.

```

holder.ListItem.setOnLongClickListener(new
View.OnLongClickListener() {
    @Override
    public boolean onLongClick(final View view) {
        final String[] action = {"Update", "Delete"};
        AlertDialog.Builder alert = new
AlertDialog.Builder(view.getContext());
        alert.setItems(action, new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog,
int i) {
                switch (i){
                    case 0:
                        /*

```



```

        Berpindah Activity pada
halaman layout updateData
        dan mengambil data pada
listMahasiswa, berdasarkan posisinya
        untuk dikirim pada activity
selanjutnya
        */
        Bundle bundle = new Bundle();
        bundle.putString("dataNIM",
listMahasiswa.get(position).getNim());
        bundle.putString("dataNama",
listMahasiswa.get(position).getNama());
        bundle.putString("dataJurusan",
listMahasiswa.get(position).getJurusan());

        bundle.putString("getPrimaryKey",
listMahasiswa.get(position).getKey());
        Intent intent = new
Intent(view.getContext(), updateData.class);
        intent.putExtras(bundle);
        context.startActivity(intent);
        break;
        case 1:
        //Menggunakan interface untuk
mengirim data mahasiswa, yang akan dihapus

        listener.onDeleteData(listMahasiswa.get(position),
position);

        break;

    }
});
alert.create();
alert.show();
return true;
}
});

```

Buka class MyListAdapter, didalam method onDeleteData() yang sudah kalian implementasi tadi, tambahkan baris kode berikut ini.

```
@Override
```

```

public void onDeleteData(data_mahasiswa data, int
position) {
    /*
    * Kode ini akan dipanggil ketika method
onDeleteData
    * dipanggil dari adapter pada RecyclerView melalui
interface.
    * kemudian akan menghapus data berdasarkan primary
key dari data tersebut
    * Jika berhasil, maka akan memunculkan Toast
    */
    String userID = auth.getUid();
    if(reference != null){
        reference.child("Admin")
            .child(userID)
            .child("Mahasiswa")
            .child(data.getKey())
            .removeValue()
            .addOnSuccessListener(new
OnSuccessListener() {
                @Override
                public void onSuccess(Void aVoid) {
                    Toast.makeText(MyListData.this,
                    "Data Berhasil Dihapus", Toast.LENGTH_SHORT).show();
                }
            });
    }
}

```

Aplikasi akan menggunakan key untuk penanda, data mana yang akan dihapus. Jika ada data dengan key yang sama, maka aplikasi akan memanggil fungsi `removeValue()` untuk menghapus data dari Database.



## YANG HARUS SUDAH ANDA KETAHUI

Untuk praktik ini Anda harus bisa:

- 1) Membuat, membangun, dan menjalankan aplikasi di Android Studio.
- 2) Merancang layout dengan tombol dan tampilan teks.
- 3) Menggunakan gaya dan tema.

## YANG AKAN ANDA BUTUHKAN

Untuk materi dan praktik ini, Anda akan membutuhkan:

- 1) Komputer Mac, Windows, atau Linux. Tool Android Studio untuk persyaratan sistem terbaru.
- 2) Akses internet atau cara alternatif untuk memuat Android Studio dan pemasangan Java terbaru ke komputer Anda.
- 3) Smartphone sebagai perangkat untuk pengujian secara nyata aplikasi yang telah dibuat.

## YANG AKAN ANDA PELAJARI

Anda akan belajar:

- 1) Membuat form login android pada android studio dengan menggunakan sqlite database.
- 2) Membuat class DatabaseHelper yang berisi perintah untuk menjalankan query seperti Create Table, Insert Data, Delete Data dan sebagainya ke Database SQLite.
- 3) Membuat class LoginActivity yang berisi perintah untuk login untuk masuk ke MainActivity dan cek session yang ada dalam database.

## YANG HARUS ANDA LAKUKAN

- 1) Mendesain tampilan antarmuka form login dengan menambahkan gambar.
- 2) Membuat aplikasi login pengguna dengan username dan password yang tersimpan di database SQLite

## RINGKASAN APLIKASI

Aplikasi login pengguna sebagai form autentifikasi sebelum masuk ke sistem manajemen user, aplikasi ini memiliki username dan password sebagai login awal saat aplikasi diakses. Seperti umumnya form login user, jika username dan password benar maka akan masuk ke menu yang menampilkan bahwa login sukses. Dan sebaliknya akan muncul login gagal jika username dan password salah.

## Latihan 7.5. Latihan Membuat Login dengan SQLite Database

### 1. Buat Project Baru

Langkah pertama silahkan buat project baru di android studio dengan menggunakan nama project loginsqlite.

### 2. Library Material Design

Tambahkan library material design dengan cara **klik gradle (module:app) >** masukan 1 baris kode berikut pada **dependencies >** **klik sync now (pastikan terkoneksi dengan internet)**

```
implementation 'com.android.support.design:26.1.0'
```

### 3. Edit String

Edit string yang nantinya digunakan untuk nama aplikasi dan beberapa komponen widget. Buka **res > values > string.xml**.

```
<resources>
    <string
name="app_name">LoginRegisterScreenDesign</string>
    <string name="create_account"><font
color="#00ff00">text</font></string>
    <string name="email">Email</string>
    <string name="password">Password</string>
    <string name="login">Login</string>
    <string name="username">Username</string>
    <string name="register">Register</string>
    <string name="back_to_login">Back to Login</string>
</resources>
```

### 4. Edit Color

Edit warna yang dengan cara buka **res > values > colors.xml**.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#607d8b</color>
    <color name="colorPrimaryDark">#455a64</color>
    <color name="colorAccent">#0c0099</color>
</resources>
```

### 5. Drawable

Copykan gambar/ icon pendukung.

### 6. Edit Style

Edit tema pada **res > values > styles.xml**. Dalam aplikasi yang akan kita buat ini kita tidak akan menampilkan **action bar (app bar)**. Sehingga kode-kodenya kita edit menjadi seperti seperti berikut.

```
<resources>
  <!-- Base application theme. -->
  <style name="AppTheme"
parent="Theme.AppCompat.Light.DarkActionBar">
  <!-- Customize your theme here. -->
  <item
name="colorPrimary">@color/colorPrimary</item>
  <item
name="colorPrimaryDark">@color/colorPrimaryDark</item>
  <item name="colorAccent">@color/colorAccent</item>
  </style>

  <style name="AppTheme.NoActionBar">
  <item name="windowActionBar">false</item>
  <item name="windowNoTitle">>true</item>
  </style>
  <style name="AppTheme.AppBarOverlay"
parent="ThemeOverlay.AppCompat.Dark.ActionBar" />

  <style name="AppTheme.PopupOverlay"
parent="ThemeOverlay.AppCompat.Light" />
</resources>
```

## 7. Activity Login

Buat activity login dengan cara **klik folder java > klik kanan nama package > New > Activity > Empty Activity > Isikan Nama LoginActivity**. Sehingga akan terdapat class java **LoginActivity.java** dan layout **activity\_login.xml**.

⇒ **LoginActivity.java**

Masukan kode-kode berikut dibawah nama package.

```
import android.content.Intent;
import android.os.Bundle;
import android.support.design.widget.Snackbar;
import android.support.design.widget.TextInputLayout;
import android.support.v7.app.AppCompatActivity;
import android.text.Html;
import android.text.Spanned;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
```

```

public class LoginActivity extends AppCompatActivity {

    //Declaration EditTexts
    EditText editTextEmail;
    EditText editTextPassword;

    //Declaration TextInputLayout
    TextInputLayout textInputLayoutEmail;
    TextInputLayout textInputLayoutPassword;

    //Declaration Button
    Button buttonLogin;

    //Declaration SqliteHelper
    SqliteHelper sqliteHelper;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        sqliteHelper = new SqliteHelper(this);
        initCreateAccountTextView();
        initView();

        //set click event of login button
        buttonLogin.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {

//Check user input is correct or not
                if (validate()) {

//Get values from EditText fields
                    String Email =
editTextEmail.getText().toString();
                    String Password =
editTextPassword.getText().toString();

//Authenticate user
                    User currentUser =
sqliteHelper.Authenticate(new User(null, null, Email,
Password));

//Check Authentication is successful or not
                    if (currentUser != null) {
                        Snackbar.make(buttonLogin,
"Successfully Logged in!", Snackbar.LENGTH_LONG).show();

```

```

        //User Logged in Successfully
Launch You home screen activity
        Intent intent=new
Intent(LoginActivity.this,MainActivity.class);
        startActivity(intent);
        finish();

    } else {

        //User Logged in Failed
        Snackbar.make(buttonLogin, "Failed
to log in , please try again",
Snackbar.LENGTH_LONG).show();

    }

    }

});

}

//this method used to set Create account TextView text
and click event( multipl colors
// for TextView yet not supported in Xml so i have
done it programmatically)
private void initCreateAccountTextView() {
    TextView textViewCreateAccount = (TextView)
findViewById(R.id.textViewCreateAccount);
    textViewCreateAccount.setText (fromHtml("<font
color='#ffffff'>I don't have account yet. </font><font
color='#0c0099'>create one</font>"));
    textViewCreateAccount.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new
Intent(LoginActivity.this, RegisterActivity.class);
            startActivity(intent);
        }
    });
}

//this method is used to connect XML views to its
Objects
private void initView() {
    editTextEmail = (EditText)
findViewById(R.id.editTextEmail);
    editTextPassword = (EditText)
findViewById(R.id.editTextPassword);
    textInputLayoutEmail = (TextInputLayout)
findViewById(R.id.textInputLayoutEmail);
}

```



```

        textInputLayoutPassword = (TextInputLayout)
findViewById(R.id.textInputLayoutPassword);
        buttonLogin = (Button)
findViewById(R.id.buttonLogin);

    }

    //This method is for handling fromHtml method
deprecation
    @SuppressWarnings("deprecation")
    public static Spanned fromHtml(String html) {
        Spanned result;
        if (android.os.Build.VERSION.SDK_INT >=
android.os.Build.VERSION_CODES.N) {
            result = Html.fromHtml(html,
Html.FROM_HTML_MODE_LEGACY);
        } else {
            result = Html.fromHtml(html);
        }
        return result;
    }

    //This method is used to validate input given by user
    public boolean validate() {
        boolean valid = false;

        //Get values from EditText fields
        String Email = editTextEmail.getText().toString();
        String Password =
editTextPassword.getText().toString();

        //Handling validation for Email field
        if
(!android.util.Patterns.EMAIL_ADDRESS.matcher(Email).match
es()) {
            valid = false;
            textInputLayoutEmail.setError("Please enter
valid email!");
        } else {
            valid = true;
            textInputLayoutEmail.setError(null);
        }

        //Handling validation for Password field
        if (Password.isEmpty()) {
            valid = false;
            textInputLayoutPassword.setError("Please enter
valid password!");
        } else {
            if (Password.length() > 5) {

```

```

        valid = true;
        textInputLayoutPassword.setError (null);
    } else {
        valid = false;
        textInputLayoutPassword.setError ("Password
is to short!");
    }
}

return valid;
}
}

```

## ⇒ activity\_login.xml

Untuk membuat layout halaman login, silahkan ketikkan kode-kode dibawah ini.

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
android:layout_height="match_parent"
xmlns:tools="http://schemas.android.com/tools"
android:background="@color/colorPrimary"
android:fitsSystemWindows="true"
tools:context=".LoginActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical"
        android:layout_marginLeft="16dp"
        android:layout_marginRight="16dp"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/imageView"
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:layout_gravity="center_horizontal"
            android:layout_margin="16dp"
            android:src="@drawable/usericon" />

        <android.support.design.widget.TextInputLayout
            android:id="@+id/textInputLayoutEmail"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            app:errorEnabled="true">

```

```

<android.support.design.widget.TextInputEditText
    android:id="@+id/editTextEmail"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="@string/email"
    android:inputType="textEmailAddress" />
</android.support.design.widget.TextInputLayout>

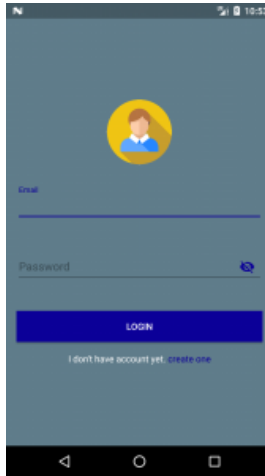
<android.support.design.widget.TextInputLayout
    android:id="@+id/textInputLayoutPassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    app:errorEnabled="true"
    app:passwordToggleEnabled="true"
    app:passwordToggleTint="@color/colorAccent">

<android.support.design.widget.TextInputEditText
    android:id="@+id/editTextPassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="@string/password"
    android:inputType="textPassword" />
</android.support.design.widget.TextInputLayout>

<Button
    android:id="@+id/buttonLogin"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="16dp"
    android:background="@color/colorAccent"
    android:text="@string/login"
    android:textColor="@android:color/white" />

<TextView
    android:id="@+id/textViewCreateAccount"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"
    android:layout_marginTop="16dp"
    android:gravity="center_horizontal" />
</LinearLayout>
</ScrollView>

```



## 8. Activity Register

Buat activity register dengan cara **klik java > klik kanan nama package > New > Activity > Empty Activity > Isikan Nama RegisterActivity**. Sehingga akan terdapat class java **RegisterActivity.java** dan layout **activity\_register.xml**.

⇒ **RegisterActivity.java**

Untuk kode-kode **RegisterActivity.java** (tempatkan dibawah nama package) adalah:

```
import android.os.Bundle;
import android.os.Handler;
import android.support.annotation.Nullable;
import android.support.design.widget.Snackbar;
import android.support.design.widget.TextInputLayout;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class RegisterActivity extends AppCompatActivity {

    //Declaration EditTexts
    EditText editTextUserName;
    EditText editTextEmail;
    EditText editTextPassword;
```

```

//Declaration TextInputLayout
TextInputLayout textInputLayoutUserName;
TextInputLayout textInputLayoutEmail;
TextInputLayout textInputLayoutPassword;

//Declaration Button
Button buttonRegister;

//Declaration SqliteHelper
SqliteHelper sqliteHelper;

@Override
protected void onCreate(@Nullable Bundle
savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_register);
    sqliteHelper = new SqliteHelper(this);
    initViewLogin();
    initView();
    buttonRegister.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if (validate()) {
                String UserName =
editTextUserName.getText().toString();
                String Email =
editTextEmail.getText().toString();
                String Password =
editTextPassword.getText().toString();

                //Check in the database is there any user
associated with this email
                if (!sqliteHelper.isEmailExists(Email)) {

                    //Email does not exist now add new user
to database
                    sqliteHelper.addUser(new User(null,
UserName, Email, Password));
                    Snackbar.make(buttonRegister, "User
created successfully! Please Login ",
Snackbar.LENGTH_LONG).show();
                    new Handler().postDelayed(new
Runnable() {
                        @Override
                        public void run() {
                            finish();
                        }
                    }, Snackbar.LENGTH_LONG);

```

```

        }else {

            //Email exists with email input
            provided so show error user already exist
            Snackbar.make(buttonRegister, "User
            already exists with same email ", Snackbar.LENGTH_LONG).show();
        }
    }
}

});
}

//this method used to set Login TextView click event
private void initViewLogin() {
    TextView textViewLogin = (TextView)
    findViewById(R.id.textViewLogin);
    textViewLogin.setOnClickListener(new
    View.OnClickListener() {
        @Override
        public void onClick(View view) {
            finish();
        }
    });
}

//this method is used to connect XML views to its Objects
private void initView() {
    editTextEmail = (EditText)
    findViewById(R.id.editTextEmail);
    editTextPassword = (EditText)
    findViewById(R.id.editTextPassword);
    editTextUserName = (EditText)
    findViewById(R.id.editTextUserName);
    textInputLayoutEmail = (TextInputLayout)
    findViewById(R.id.textInputLayoutEmail);
    textInputLayoutPassword = (TextInputLayout)
    findViewById(R.id.textInputLayoutPassword);
    textInputLayoutUserName = (TextInputLayout)
    findViewById(R.id.textInputLayoutUserName);
    buttonRegister = (Button)
    findViewById(R.id.buttonRegister);
}

//This method is used to validate input given by user
public boolean validate() {
    boolean valid = false;

    //Get values from EditText fields
    String UserName =
    editTextUserName.getText().toString();
    String Email = editTextEmail.getText().toString();
}

```

```

        String Password =
        editTextPassword.getText().toString();

        //Handling validation for UserName field
        if (UserName.isEmpty()) {
            valid = false;
            textInputLayoutUserName.setError("Please enter
valid username!");
        } else {
            if (UserName.length() > 5) {
                valid = true;
                textInputLayoutUserName.setError(null);
            } else {
                valid = false;
                textInputLayoutUserName.setError("Username is
to short!");
            }
        }

        //Handling validation for Email field
        if
(!android.util.Patterns.EMAIL_ADDRESS.matcher(Email).matches())
{
            valid = false;
            textInputLayoutEmail.setError("Please enter valid
email!");
        } else {
            valid = true;
            textInputLayoutEmail.setError(null);
        }
        //Handling validation for Password field
        if (Password.isEmpty()) {
            valid = false;
            textInputLayoutPassword.setError("Please enter
valid password!");
        } else {
            if (Password.length() > 5) {
                valid = true;
                textInputLayoutPassword.setError(null);
            } else {
                valid = false;
                textInputLayoutPassword.setError("Password is
to short!");
            }
        }
        return valid;
    }
}

```

⇒ activity\_register.xml

## Masukan kode-kode dibawah ini untuk merancang layout register.

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    android:background="@color/colorPrimary"
    android:fitsSystemWindows="true"
    tools:context=".RegisterActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical"
        android:layout_marginLeft="16dp"
        android:layout_marginRight="16dp"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/imageView"
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:layout_gravity="center_horizontal"
            android:layout_margin="10dp"
            android:src="@drawable/usericon" />

        <android.support.design.widget.TextInputLayout
            app:errorEnabled="true"
            android:id="@+id/textInputLayoutUserName"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp">

            <android.support.design.widget.TextInputEditText
                android:id="@+id/editTextUserName"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:hint="@string/username"
                android:inputType="textPersonName" />
        </android.support.design.widget.TextInputLayout>

        <android.support.design.widget.TextInputLayout
            android:id="@+id/textInputLayoutEmail"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp">
```



```

        app:errorEnabled="true">

<android.support.design.widget.TextInputEditText
    android:id="@+id/editTextEmail"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="@string/email"
    android:inputType="textEmailAddress" />
</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout
    android:id="@+id/textInputLayoutPassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    app:errorEnabled="true"
    app:passwordToggleEnabled="true"
    app:passwordToggleTint="@color/colorAccent">

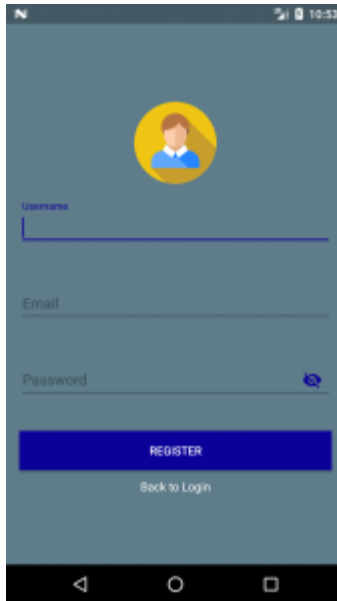
<android.support.design.widget.TextInputEditText
    android:id="@+id/editTextPassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="@string/password"
    android:inputType="textPassword" />
</android.support.design.widget.TextInputLayout>

<Button
    android:id="@+id/buttonRegister"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColor="@android:color/white"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="10dp"
    android:layout_weight="1"
    android:background="@color/colorAccent"
    android:text="@string/register" />

<TextView
    android:id="@+id/textViewLogin"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="10dp"
    android:gravity="center"
    android:text="@string/back_to_login"
    android:textColor="@android:color/white" />
</LinearLayout>

```

```
</ScrollView>
```



## 9. Class SqliteHelper

Tambahkan class **SqliteHelper**. Untuk membuatnya silahkan **klik folder java > klik kanan pada nama package > pilih new > class > beri nama class SqliteHelper**. Masukkan kode-kode berikut ini pada **SqliteHelper.java** (tempatkan dibawah nama package).

```
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class SqliteHelper extends SQLiteOpenHelper {

    //DATABASE NAME
    public static final String DATABASE_NAME =
"loopwiki.com";

    //DATABASE VERSION
    public static final int DATABASE_VERSION = 1;
```

```

//TABLE NAME
public static final String TABLE_USERS = "users";

//TABLE USERS COLUMNS
//ID COLUMN @primaryKey
public static final String KEY_ID = "id";

//COLUMN user name
public static final String KEY_USER_NAME = "username";

//COLUMN email
public static final String KEY_EMAIL = "email";

//COLUMN password
public static final String KEY_PASSWORD = "password";

//SQL for creating users table
public static final String SQL_TABLE_USERS = " CREATE
TABLE " + TABLE_USERS
    + " ( "
    + KEY_ID + " INTEGER PRIMARY KEY, "
    + KEY_USER_NAME + " TEXT, "
    + KEY_EMAIL + " TEXT, "
    + KEY_PASSWORD + " TEXT"
    + " ) ";

    public SQLiteHelper(Context context) {
        super(context, DATABASE_NAME, null,
DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase) {
        //Create Table when oncreate gets called
        sqLiteDatabase.execSQL(SQL_TABLE_USERS);
    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase,
int i, int il) {
        //drop table to create new one if database version
updated
        sqLiteDatabase.execSQL(" DROP TABLE IF EXISTS " +
TABLE_USERS);
    }

    //using this method we can add users to user table
    public void addUser(User user) {

```

```

        //get writable database
        SQLiteDatabase db = this.getWritableDatabase();

        //create content values to insert
        ContentValues values = new ContentValues();

        //Put username in @values
        values.put(KEY_USER_NAME, user.userName);

        //Put email in @values
        values.put(KEY_EMAIL, user.email);

        //Put password in @values
        values.put(KEY_PASSWORD, user.password);

        // insert row
        long todo_id = db.insert(TABLE_USERS, null,
values);
    }

    public User Authenticate(User user) {
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.query(TABLE_USERS, // Selecting
Table
            new String[]{KEY_ID, KEY_USER_NAME,
KEY_EMAIL, KEY_PASSWORD}, //Selecting columns want to query
            KEY_EMAIL + "=?",
            new String[]{user.email}, //Where clause
            null, null, null);

        if (cursor != null && cursor.moveToFirst() &&
cursor.getCount() > 0) {
            //if cursor has value then in user database
            there is user associated with this given email
            User user1 = new User(cursor.getString(0),
cursor.getString(1), cursor.getString(2),
cursor.getString(3));

            //Match both passwords check they are same or
not
            if
            (user.password.equalsIgnoreCase(user1.password)) {
                return user1;
            }
            //if user password does not matches or there is no
record with that email then return @false
            return null;
        }
    }
}

```

```

public boolean isEmailExists(String email) {
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db.query(TABLE_USERS, // Selecting
Table
        new String[]{KEY_ID, KEY_USER_NAME,
KEY_EMAIL, KEY_PASSWORD}, //Selecting columns want to query
        KEY_EMAIL + "=?",
        new String[]{email}, //Where clause
        null, null, null);

    if (cursor != null && cursor.moveToFirst() &&
cursor.getCount() > 0) {
        //if cursor has value then in user database
there is user associated with this given email so return
true
        return true;
    }

    //if email does not exist return false
return false;
}
}

```

Di dalam Class **SqliteHelper** merupakan bagian dari **SQLiteOpenHelper** di dalamnya terdapat: Nama database, Versi database, Tabel, Kolom

## 10. Class User

Tambahkan class **User** yang akan dijadikan **model dari user**. Untuk membuatnya sama seperti point 8, lalu beri nama class **User**. Tambahkan kode-kode dibawah ini pada **User.java**.

```

public class User {
    public String id;
    public String userName;
    public String email;
    public String password;

    public User(String id, String userName, String email,
String password) {
        this.id = id;
        this.userName = userName;
        this.email = email;
        this.password = password;
    }
}

```

## 11. Home Screen

Jika login berhasil maka aplikasi akan menampilkan home screen. Home screen yang kita buat kali ini hanya menampilkan komponen `TextView`. Tambahkan kode-kode dibawah ini pada `activity_main.xml`.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"

tools:context="badoystudio.com.loginsqlite.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:text="Kamu telah berhasil login "
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```



## 12. Android Manifest

Buka **AndroidManifest.xml** yang ada pada folder **app > manifest**. Edit kode-kodenya sehingga menjadi seperti dibawah ini(pakcagenya sesuaikan dengan nama package yang kamu buat).

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
package="badoystudio.com.loginsqlite">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".LoginActivity"
            android:label="@string/app_name"
            android:theme="@style/AppTheme.NoActionBar">
            <intent-filter>
                <action
                    android:name="android.intent.action.MAIN" />

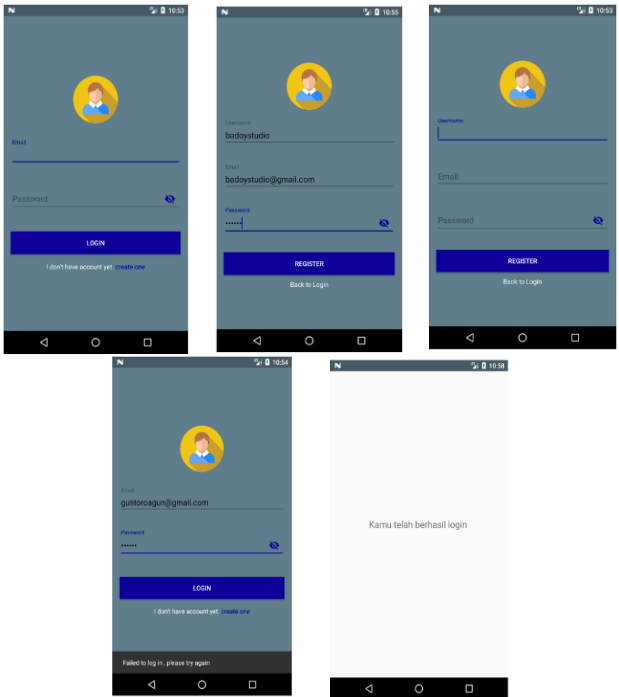
                    <category
                        android:name="android.intent.category.LAUNCHER" />
                </intent-filter>
            </activity>
            <activity
                android:name=".RegisterActivity"
                android:label="@string/app_name"
                android:theme="@style/AppTheme.NoActionBar" />

            <activity
                android:name=".MainActivity"
                android:label="Home Screen"
                android:theme="@style/AppTheme.NoActionBar" />
        </application>

</manifest>
```

## 13. Running Project

Running project menggunakan emulator bawaan anroid, genymotion, atau langsung melalui smartphone android yang kamu miliki. Maka hasil akhirnya adalah seperti gambar dibawah ini.



Tampilan Form



## RANGKUMAN

Dalam Praktik ini, Anda telah belajar:

- 1) Membuat form login android pada android studio dengan menggunakan sqlite database
- 2) Membuat class DatabaseHelper yang berisi perintah untuk menjalankan query seperti Create Table, Insert Data, Delete Data dan sebagainya ke Database SQLite.
- 3) Membuat class LoginActivity yang berisi perintah untuk login untuk masuk ke MainActivity dan cek session yang ada dalam database.

## REFERENSI

- Buku Kursus Dasar-Dasar Developer Android – Konsep, Belajar mengembangkan aplikasi Android, Dikembangkan oleh Tim Pelatihan Developer Google.
- Buku Kerja Praktik, Kursus Dasar-Dasar Developer Android — Praktik, Belajar mengembangkan aplikasi Android, Dikembangkan oleh Tim Pelatihan Developer Google

## MATERI KOMPETENSI

Materi bab ini akan membahas:

- 1) Konsep dasar bahasa pemrograman kotlin.
- 2) Konfigurasi Kotlin ke Android Studio dan Kotlin Extensions
- 3) Kotlin dasar: Packages dan Function Dasar
- 4) Pembuatan project aplikasi sederhana menggunakan Kotlin

## YANG HARUS SUDAH ANDA KETAHUI

Untuk materi dan praktik ini Anda harus bisa:

- 1) Cara membuat project aplikasi dengan Android Studio.
- 2) Memahami struktur direktori file project Android Studio.
- 3) Menjalankan aplikasi di emulator atau perangkat.

## YANG AKAN ANDA BUTUHKAN

Untuk materi dan praktik ini, Anda akan membutuhkan:

- 1) Komputer Mac, Windows, atau Linux. Tool Android Studio untuk persyaratan sistem terbaru.
- 2) Smartphone sebagai perangkat untuk pengujian secara nyata aplikasi yang telah dibuat.

## YANG AKAN ANDA PELAJARI

Anda akan belajar:

- 1) Cara membuat antarmuka pengguna interaktif dalam Layout Editor, dalam XML, secara terprogram menggunakan Kotlin.
- 2) Cara membuat project menampilkan data dengan kotlin.
- 3) Belajar tentang package dan function dasar dalam bahasa pemrograman Kotlin
- 4) Membuat aplikasi sederhana input dan output data.

# BAB 8

## PENGEMBANGAN APLIKASI ANDROID DENGAN KOTLIN



### Bab 8: Pengembangan Aplikasi Adroid dengan Kotlin

#### 8.1. Kotlin - Android

Kotlin merupakan Bahasa Pemrograman modern yang bersifat *statically-typed* yang dapat di jalankan di atas *platform Java Virtual Machine (JVM)*. Kotlin juga dapat di kompilasi (*compile*) ke dalam bentuk JavaScript. Kotlin merupakan bahasa pemrograman yang bisa dibilang relatif baru. Adalah JetBrains yang juga mengembangkan salah satu *tools* pemrograman (IDE) yang saat ini di adopsi oleh *google* untuk di gunakan sebagai *tools* pengembangan aplikasi di Android (Android Studio).

Hingga saat ini Kotlin telah mencapai rilis 1.0. Pengembang menyatakan Kotlin sebagai bahasa pemrograman yang pragmatis. Kotlin merupakan bahasa pemrograman pragmatis untuk JVM dan Android yang mengkombinasikan Object Oriented (OO) dan fitur fungsional dan fokus pada interoperabilitas, keamanan, kejelasan dan dukungan integrasi dengan berbagai tools major. Sebagai bahasa pemrograman dengan tujuan umum, Kotlin bisa berjalan dimanapun layaknya Java. Kotlin bisa digunakan untuk mengembangkan berbagai aplikasi server-side, aplikasi mobile (Android), aplikasi desktop. Dukungan tools major untuk Kotlin:

- IntelliJ IDEA, Android Studio dan Eclipse
- Maven, Gradle dan Ant
- Spring Boot (Dukungan terbaru pada rilis ini)
- Github, Slack dan Minecaft

Salah satu kunci fokus Kotlin sebagai bahasa pemrograman pragmatis adalah interoperabilitas dengan menggabungkan proyek Java + Kotlin. Oleh karena itu, Kotlin membuat adopsinya menjadi lebih mudah untuk koding. Selain itu, Kotlin memiliki library standar yang luas sehingga bisa digunakan

untuk pengerjaan tugas sehari-hari dan tetap menjaga ukuran bytecode tetap rendah, hanya 6 byte. Dukungan lainnya adalah berbagai library Java yang bisa digunakan di Kotlin dan sebaliknya.

Letak pragmatis yang diusung pengembang Kotlin, ada pada desainnya. Keuntungan desainnya, bisa digunakan untuk mencegah bug dan daur ulang kode. Selain interoperabilitas, fokus kedua adalah infrastrukstur. Untuk dukungan infrastukturnya, Kotlin berintegrasi dengan Java, Maven dan Android.

Tidak seperti bahasa pemrograman lainnya yang membangun paket manajer sendiri, Kotlin memanfaatkan berbagai plugin Maven dan Gradle sebagai paket manajernya. Kotlin juga berusaha mengefektifkan waktu pengembangan dengan membuat interface dari koleksi JDK sehingga tak perlu mendesain dari nol. Kotlin menggunakan ukuran bytecode yang rendah, karena banyak orang masih menggunakan Java 6 (Terutama Android).

Kelebihan dari kotlin tersebut dibandingkan dengan bahasa pemrograman lainnya (dalam hal ini Java) Jika kita lihat dari penjelasan yang tertulis pada website [kotlin](#), maka dapat di simpulkan ke dalam poin-poin berikut ini;

- **Concise** : Kotlin mampu mengurangi *boilerplate of code* atau tingkat kerumitan dari kode yang biasa kita tulis, ketika menggunakan bahasa Java
- **Safe** : Kotlin mampu menjamin bahwa setiap *syntax* yang kita tulis secara proses kompilasi dapat mencegah kemungkinan terjadinya *error*, misalnya mampu mencegah terjadinya **NullPointerExceptions** ketika kita *coding* menggunakan bahasa Java
- **Versatile** : Kotlin sejatinya sama seperti Java, karena memang kotlin itu sendiri di turunkan dari bahasa induknya, yaitu Java. Sehingga kotlin juga dapat di pakai dalam pengembangan aplikasi di **Web** maupun **Mobile**
- **Interoperable** : Kotlin **tidak sama** seperti bahasa Java turunan lainnya (misal; Scala ataupun Clojure) yang tidak dapat dijalankan bersamaan dengan kode yang kita tulis menggunakan Java. Kotlin mampu membaca kode lama atau *library* yang kita gunakan atau kita tulis dengan bahasa Java dan begitupun sebaliknya

Lalu, apa hubungannya dengan Android. Tentu saja berhubungan, sebab Android di tulis menggunakan bahasa pemrograman Java, meskipun Android memiliki *lifecycle* sendiri. Karena Android di tulis menggunakan bahasa pemrograman Java, maka otomatis Kotlin pun juga dapat digunakan untuk membuat aplikasi di Android.

Kotlin diresmikan sebagai bahasa kedua yang didukung untuk dipakai membuat aplikasi Android di Android Studio. Saat ini, posisi Java sebagai bahasa utama memang belum digantikan oleh Kotlin. Namun tidak dapat disangkal pula bahwa, perkembangan bahasa Kotlin di komunitas developer Android sangat pesat. Bahkan, komunitas-komunitas developer Android saling menguatkan untuk merilis tutorial baru dalam bahasa Kotlin dan memperbarui tutorial lama untuk menggunakan bahasa ini.

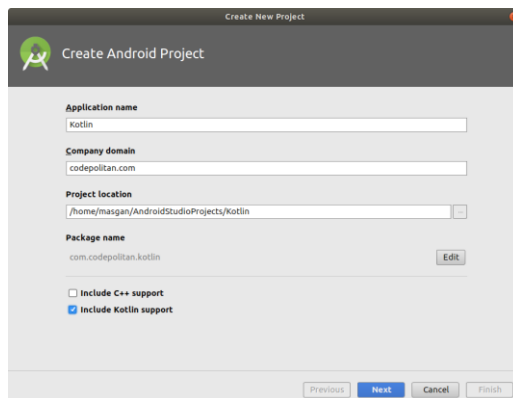
### 1) Memiliki Ekosistem yang Sudah Matang

Kotlin sudah melalui tahapan panjang sebelum memasuki versi stabil 1.0 sejak tahun 2016 yang lalu. Itu artinya, bahasa pemrograman Kotlin adalah bahasa yang sudah sangat matang. Bahasa ini tidak lagi dalam tahap *beta* apalagi *alpha*, tapi sudah *stable* sehingga cocok dipakai untuk *production*.

### 2) Terintegrasi dengan Android studio

Kotlin dikembangkan oleh JetBrains, perusahaan yang mengembangkan IntelliJ, basis awal Android Studio. Tidak mengherankan jika Kotlin terintegrasi dengan sempurna di dalam Android Studio karena memang bisa dibayangkan keduanya dikembangkan oleh perusahaan yang sama.

Di Android Studio versi 2.3 kita masih harus memasang *plugin* tambahan untuk Android Studio agar bisa memakai Kotlin. Tetapi, setelah versi 3.0 Kotlin sudah bisa langsung digunakan tanpa harus memasang *plugin* tambahan.



### 3) Update Java di Android Relatif Lebih Lambat

Kotlin memiliki konsep *lambda*, *extention function*, *functional operation on collection*, *sealed classes*, *data classes*, dan konsep-konsep modern yang lain yang sudah dibungkus siap pakai saat ini juga. Bandingkan dengan Java, dimana kita harus *ngoprek* sendiri agar *project* Android bisa memakai Java 8 dan

menggunakan konsep-konsep modern tadi. Dengan Kotlin, semua itu sudah ada tanpa susah payah.

#### 4) Aman dari NullPointerException

Jika sudah pernah membuat aplikasi Android dengan Java, pasti sudah bersahabat dengan yang namanya **NullPointerException** (NPE). *Exception* ini bisa membuat developer frustrasi karena membuat aplikasi *crash*. NPE memang masih bisa diakali dengan menggunakan blok **try catch** atau menggunakan *annotations*, tapi dengan Kotlin kita mendapatkan pengamanan secara otomatis dari sananya.

Saat kita membuat aplikasi Android dengan Kotlin, *compiler* akan mendeteksi adanya kemungkinan error sebelum membuat APK (atau saat di **run**). Dengan begini kita bisa langsung tahu dimana sumber yang nantinya bisa bermasalah. Jika bagian yang memiliki kemungkinan error tidak diatasi, maka aplikasi tidak akan bisa di run. Berbeda dengan Java, dimana tidak ada pemeriksaan seperti itu dari sananya. Kita akan tahu bahwa suatu bagian menyebabkan NPE saat aplikasi sudah berjalan dan *crash*.

#### 5) Kotlin Tidak Hanya di Android

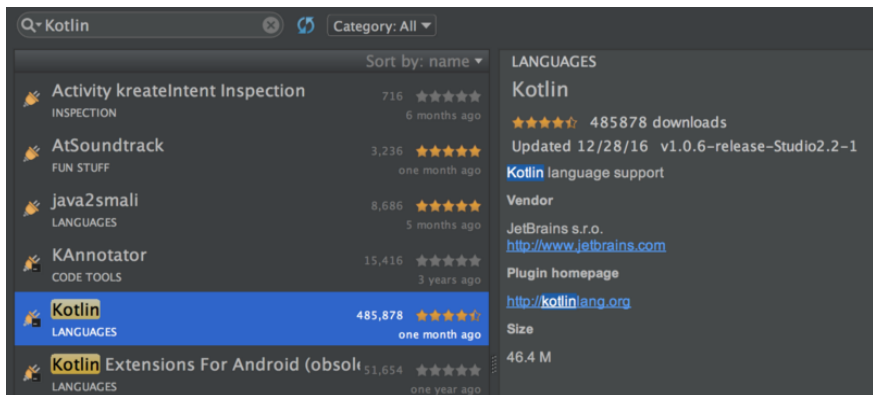
Meski pemakaiannya yang paling *booming* adalah dilingkungan Android, Kotlin nyatanya bisa dipakai diluar itu. Kotlin bisa dipakai untuk membuat aplikasi iOS (melalui Kotlin Native), juga bisa dipakai untuk menulis JavaScript (melalui Kotlin JS). Intinya, karena Kotlin memiliki *interoperability* 100% dengan Java, maka apapun yang bisa menggunakan Java bisa menggunakan Kotlin.

## 8.2. Konfigurasi Kotlin ke Android Studio

Contoh sederhana bagaimana kotlin dapat di implementasikan di Android:

Sebelum kita mulai menulis kode dengan kotlin, kita perlu menambahkan *plugin* dari Android Studio yang tentu saja akan digunakan dalam menulis bahasa pemrograman dengan Kotlin.

- ⇒ Buka Android Studio, lalu pilih menu **configure** -> **plugins**. Kemudian klik tombol **“Browse repositories...”** dan ketikkan kata **“Kotlin”** lalu klik **“Install”**



Gambar 108. Install Plugin Kotlin Pada Android Studio

- ⇒ Setelah terinstall, silahkan di *restart* Android Studio.
- ⇒ Langkah selanjutnya adalah kita perlu menambahkan *plugin* pada file *gradle* kita terlebih dahulu, yaitu di dalam *root project* kita, agar kotlin dapat di kompilasi dan di jalankan di atas *platform* Android.

```
dependencies {
    classpath 'org.jetbrains.kotlin:kotlin-gradle-
plugin:1.0.6'
}
```

- ⇒ Tambahkan pada *module* utama dari *project*

```
apply plugin: 'kotlin-android'

dependencies {
    compile 'org.jetbrains.kotlin:kotlin-stdlib:1.0.6'
}
```

- ⇒ Tampilan utama dari aplikasi kita dengan nama **activity\_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/andr
oid"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/cl_main"
android:layout_width="match_parent"
```



```

android:layout_height="match_parent"

tools:targetApi="LOLLIPOP">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar_main"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_alignParentTop="true"
            android:background="@color/colorAccent"
            android:elevation="4dp"
            app:theme="@style/AppTheme.Toolbar" />

        <Button
            android:id="@+id/btn_main"
            style="?attr/borderlessButtonStyle"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_centerInParent="true"
            android:layout_marginEnd="16dp"
            android:layout_marginStart="16dp"
            android:paddingBottom="16dp"
            android:paddingTop="16dp"
            android:text="@string/btn_click"

            android:textColor="@color/colorPrimaryDark"
            android:textSize="16sp" />
    </RelativeLayout>
</android.support.design.widget.CoordinatorLayout>

```

⇒ Buat implementasi dari *layout* tersebut dengan bahasa pemrograman kotlin seperti berikut.

```

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?)
    {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val root = findViewById(R.id.cl_main) as
        CoordinatorLayout

        val toolbar = findViewById(R.id.toolbar_main)
        as Toolbar
    }
}

```

```

toolbar.title = title
setSupportActionBar(toolbar)

val message =
resources.getString(R.string.message)
val button = findViewById(R.id.btn_main) as
Button
button.setOnClickListener { view ->
Snackbar.make(root, message,
Snackbar.LENGTH_SHORT).show() }
}
}

```

⇒ Sebelum dijalankan, jangan lupa di registrasikan *class activity* yang sudah kita buat pada file *AndroidManifest*

### 8.3. Kotlin Extensions



Kotlin, similar to C# and Gosu, provides the ability to extend a class with new functionality without having to inherit from the class or use any type of design pattern such as Decorator. This is done via special declarations called *extensions*. Kotlin supports *extension functions* and *extension properties*

Sederhananya adalah ada beberapa fungsi yang biasa kita jadikan sebagai *common function* yang tujuannya adalah untuk mempermudah *object* lain dalam menerapkan fungsi tersebut, misalnya ketika kita membuat *object utility*. Pada kotlin, hal tersebut dapat dipermudah dengan bantuan *extension*.

Sebelum kita mulai mempelajari *kotlin extension*, ada baiknya kita mengetahui terlebih dahulu (khususnya pada penerapannya di Android) terdapat 2 jenis *extension* yang dapat diterapkan pada Android;

- ***Kotlin Extensions*** : *Extension* ini dapat digunakan sebagai pengganti dari *utility class* yang biasa kita buat menggunakan bahasa Java
- ***Kotlin Android Extensions*** : *Extension* ini dapat digunakan untuk mempermudah kita dalam memanggil **ID** dari *layout* yang sudah kita buat pada *file xml* kita. Untuk dapat menggunakan *extension* tersebut, maka kita perlu menambahkan *plugin* berikut pada *file gradle* kita

```
apply plugin: 'kotlin-android-extensions'
```

Selanjutnya, kita buat terlebih dahulu *layout* kita, sebut saja dengan nama **activity\_main.xml**

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/cl_main"
android:layout_width="match_parent"
android:layout_height="match_parent"

tools:targetApi="LOLLIPOP">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar_main"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_alignParentTop="true"
            android:background="@color/colorAccent"
            android:elevation="4dp"
            app:theme="@style/AppTheme.Toolbar" />

        <Button
            android:id="@+id/btn_main"
            style="?attr/borderlessButtonStyle"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_centerInParent="true"
            android:layout_marginEnd="16dp"
            android:layout_marginStart="16dp"
            android:paddingBottom="16dp"
            android:paddingTop="16dp"
            android:text="@string/btn_click"
            android:textColor="@color/colorPrimaryDark"
            android:textSize="16sp" />

    </RelativeLayout>
</android.support.design.widget.CoordinatorLayout>

```

Setelah itu, mari kita buat implementasi dengan *class* Kotlin, kita namakan saja dengan **MainActivity.kt**

```

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
    }
}

```

```

        setContentView(R.layout.activity_main)

        toolbar_main.title = title
        setSupportActionBar(toolbar_main)
    }
}

```

Ini tidak melakukan inisialisasi dari *view* yang terdapat pada *layout*, melainkan hanya memanggilnya dengan menggunakan **ID** daripada *view* tersebut, cukup mudah dan sederhana. Jika kita lihat di Java pun ada cara *binding view* seperti ini, salah satunya dengan menggunakan ButterKnife. Inilah yang kita sebut dengan ***Kotlin Android Extensions***, lalu bagaimana dengan ***Kotlin Extensions***.

Pertama-tama, mari kita buat *file* (bukan *class*, melainkan *file*) dengan nama **MainExtensions.kt** dan membuatkan salah satu fungsi untuk mengembalikan nilai *String* seperti pada contoh berikut.

```

fun getHello() = "Hello from Kotlin Extension!"

```

Lalu, coba kita terapkan fungsi tersebut pada *activity class* kita

```

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        toolbar_main.title = title
        setSupportActionBar(toolbar_main)

        btn_main.setOnClickListener { view ->
Snackbar.make(cl_main, getHello(),
Snackbar.LENGTH_SHORT).show() }
    }
}

```

Kita bisa lihat dari contoh tersebut bahwa ***Kotlin Extension*** dapat di panggil dari class manapun tanpa harus melakukan inisialisasi atau membuatkan *pattern singleton*, seperti pada umumnya yang biasa kita lakukan jika menggunakan bahasa Java.

## 8.4. Kotlin dari Dasar: Packages dan Function Dasar

Kita akan belajar bahasa dengan melihat bagaimana mengatur kode menggunakan packages, dan kemudian melanjutkan pengenalan function di Kotlin.

### a. Packages

Jika Anda sudah familiar dengan Java, Anda tahu Java menggunakan packages untuk mengelompokkan class terkait; Sebagai contoh, packages `java.util` memiliki sejumlah class utilitas. Packages dideklarasikan dengan kata kunci `package`, dan file Kotlin dengan deklarasi `package` di awal dapat berisi deklarasi class, function, atau interface.

#### ▪ Deklarasi

Lihatlah kode di bawah ini, kami telah mendeklarasikan package `com.chikekotlin.projectx` menggunakan kata kunci `package`. Kami juga mendeklarasikan class `MyClass` di dalam package ini.

```
package com.chikekotlin.projectx
class MyClass
```

nama yang memenuhi syarat untuk class `MyClass` adalah `com.chikekotlin.projectx.MyClass`.

```
package com.fktikotlin.project
```

```
fun saySomething(): String {
    return "How long?"
}
```

Pada kode di atas, top-level function telah dibuat (kita akan segera melakukannya). Begitu juga dengan `MyClass`, nama yang memenuhi syarat untuk

function `saySomething()` adalah `com.chikekotlin.projectx.saySomething`.

#### ▪ Import

Di Kotlin, kami menggunakan deklarasi `import` untuk mengaktifkan compiler untuk menemukan class, function, interface atau object yang akan diimpor. Di sisi lain, pada Java kita tidak dapat secara langsung mengimpor function atau class hanya method atau interface.

Kami menggunakan `import` untuk mengakses function, interface, class atau object di luar package tempat dideklarasikan.

```
import com.fktikotlin.project.Something

fun main(args: Array<String>) {
    Something() // will print "How long"
}
```

Dalam potongan kode di atas, kita mengimpor function `saySomething()` dari package yang berbeda, dan kemudian kita menjalankan function tersebut.

Kotlin juga mendukung impor wildcard menggunakan operator `*`. Operator akan mengimpor semua class, interface dan function yang dideklarasikan dalam package secara sekaligus. Hal ini tidak disarankan, meskipun-biasanya lebih baik membuat impor Anda eksplisit.

```
import com.fktikotlin.project.*
```

- **Import Aliasing**

Bila Anda memiliki perpustakaan yang memiliki nama kelas atau function yang bertentangan (misalnya masing-masing mendeklarasikan function dengan nama yang sama), Anda dapat menggunakan kata kunci `as` untuk diberikan ke entitas yang diimpor tersebut sebagai nama sementara.

```
import com.chikekotlin.projectx.saySomething
import com.chikekotlin.projecty.Something as
projectSomething

fun main(args: Array<String>) {
    projectSomething()
}
```

Perhatikan bahwa nama sementara hanya digunakan di dalam file yang telah ditetapkan.

## b. Functions

Sebuah kelompok function dan serangkaian pernyataan kode yang melakukan tugas. Rincian pelaksanaan function disembunyikan dari pemanggil.

Di Kotlin, function didefinisikan dengan menggunakan kata kunci `fun`, seperti yang ditunjukkan pada contoh berikut:

```
fun hello(name: String): String {
    return "Hello $name" }
val message = hello("fkti")
print(message) // will print "Hello fkti"
```

Pada kode yang berada di atas, kita mendefinisikan sebuah function sederhana `hello()` dengan satu parameter tunggal `name` dengan tipe `String`. Function ini mengembalikan tipe `String`. The parameter definition format for functions is `name: type`, e.g. `age: Int`, `price: Double`, `student: StudentClass`.

```
fun hello(name: String): Unit {
    print("Hello $name")
}
hello("fkti") // will print "Hello fkti"
```

Function diatas hampir mirip dengan yang sebelumnya, tapi perhatikan bahwa yang satu ini memiliki tipe pengembalian `Unit`. Karena function ini tidak mengembalikan satupun nilai signifikan - hanya menampilkan pesan - secara default tipe pengembaliannya adalah `Unit`. `Unit` adalah sebuah object di Kotlin yang mirip dengan tipe `Void` pada Java dan C.

```
public object Unit {
    override fun toString() = "kotlin.Unit"
}
```

Perhatikan bahwa jika Anda tidak secara eksplisit menyatakan tipe pengembalian menjadi `Unit`, jenisnya akan disimpulkan oleh compiler.

```
fun hello(name: String) { // will still compile
    print("Hello $name")
}
```

- **Single-Line Functions**

Single-line atau one-line functions adalah function yang hanya memiliki satu ekspresi. Pada function ini kita akan mendapatkan In this function, kita menyingkirkan kurung kurawal dan menggunakan simbol `=` sebelum ekspresi.

Dengan kata lain, kita menyingkirkan blok function.

```
fun calCircumference(radius: Double): Double {
    return (2 * Math.PI) * radius
}
```

Function diatas bisa dipersingkat menjadi satu baris:

```
fun calCircumference(radius: Double) = (2 * Math.PI) * radius
```

Lihatlah function yang diperbarui di atas, Anda dapat melihat bahwa kami telah membuat kode menjadi lebih singkat dengan menghapus kurung kurawal {}, kata kunci return, dan juga jenis pengembalian (yang disimpulkan oleh compiler).

Anda masih bisa memasukkan tipe pengembalian agar lebih eksplisit.

```
fun calCircumference(radius: Double): Double = (2 * Math.PI) * radius
```

- **Named Parameters**

Parameter yang diberi nama memungkinkan function lebih mudah dibaca dengan memberi nama parameter yang dilewatkan ke function saat dipanggil.

Pada contoh berikut, kami membuat sebuah function yang mencetak nama lengkap.

```
fun sayMyFullName(firstName: String, lastName: String,
middleName: String): Unit {
    print("My full name is $firstName $middleName $lastName");
}
```

Untuk menjalankan function di atas, kita dapat memanggilnya seperti ini:

```
sayMyFullName("fkti", "unmul", "informatika")
```

Melihat function panggilan di atas, kita tidak tahu argumen tipe String mana yang sesuai dengan parameter function (walaupun beberapa IDE seperti IntelliJ IDEA dapat membantu kita). Pengguna function harus melihat signature function (atau source code) atau dokumentasi untuk mengetahui parameter yang sesuai dengan masing-masing parameter.

```
sayMyFullName(firstName = "fkti", middleName = "unmul",
lastName = "informatika")
```

Pada function kedua di atas, kami memberikan nama parameter sebelum nilai argumen. Anda dapat melihat bahwa pemanggilan function ini lebih jelas dan lebih mudah dibaca daripada yang sebelumnya. Cara memanggil function ini membantu mengurangi kemungkinan adanya bug yang bisa terjadi bila argumen dari tipe yang sama tertukar karena kesalahan.

Pemanggil juga dapat mengubah urutan parameter menggunakan parameter bernama. Sebagai contoh:



```
sayMyFullName(lastName = "informatika", middleName = "unmul",
firstName = "fkti") // will still compile
```

Pada kode di atas, kita menukar posisi argumen dari `firstName` dengan `lastName`. Urutan argumen tidak masalah dengan parameter yang diberi nama karena compiler akan memetakan masing-masing parameter function dengan tepat.

- **Parameter Default**

Di Kotlin, kita bisa memberi nilai default function untuk parameter-parameternya. Nilai default ini digunakan jika tidak ada yang ditugaskan ke argumen selama pemanggilan function. Untuk melakukan ini di Java, kita harus membuat metode kelebihan beban yang berbeda.

Pada method `calCircumference()`, memodifikasi method dengan menambahkan nilai default untuk parameter `pi` — `Math.PI`, constant yang berada dari package `java.lang.Math`

```
fun calCircumference(radius: Double, pi: Double = Math.PI):
Double = (2 * pi) * radius
```

Ketika kita memanggil function ini, kita dapat melewati nilai perkiraan untuk `pi` atau menggunakannya secara default.

```
print(calCircumference(24.0))
// used default value for PI and prints 150.79644737231007
print(calCircumference(24.0, 3.14))
// passed value for PI and prints 150.72
```

Perhatikanlah contoh lainnya:

```
fun printName(firstName: String, middleName: String = "N/A",
lastName: String) {
    println("first name: $firstName - middle name: $middleName
- last name: $lastName")
}
```

Dalam kode berikut, kita mencoba memanggil function tersebut, namun tidak akan di-compile:

```
printName("fkti", "informatika") // won't compile
```

Dalam function panggilan di atas, saya melewati nama depan dan nama belakang saya ke function, dan berharap untuk menggunakan nilai default untuk nama tengah. Tapi ini tidak akan di-compile karena compiler bingung. Tidak tahu untuk apa argumen "informatika" itu — apakah untuk parameter `middleName` atau parameter `lastName`?

Untuk memecahkan masalah tersebut, kita bisa menggabungkan named parameter dan default parameter.

```
printName("fkti", lastName = "informatika") // will now compile
```

- **Java Interoperability**

Mengingat Java tidak mendukung nilai parameter default dalam method, Anda harus menentukan semua nilai parameter secara eksplisit saat Anda memanggil function Kotlin dari Java. Tapi Kotlin memberi kita fungsionalitas untuk mempermudah dalam pemanggilan Java dengan memberi anotasi pada function Kotlin dengan `@JvmOverloads`. Anotasi ini akan menginstruksikan compiler Kotlin untuk menghasilkan overloads function Java untuk kita.

Dalam contoh berikut, kami memberi beberapa function `calCircumference()` dengan `@JvmOverloads`.

```
@JvmOverloads
fun calCircumference(radius: Double, pi: Double = Math.PI):
Double = (2 * pi) * radius
```

Kode berikut dihasilkan oleh compiler Kotlin sehingga pemanggil Java tersebut dapat memilih yang mana yang akan dipanggil.

```
// Java
double calCircumference(double radius, double pi);
double calCircumference(double radius);
```

Dalam definisi metode Java yang terakhir dihasilkan, parameter `pi` dihilangkan. Ini berarti bahwa metode tersebut akan menggunakan nilai `pi` default.

- **Unlimited Arguments**

Di Java, kita dapat membuat metode untuk menerima sejumlah argumen yang tidak ditentukan dengan menyertakan elipsis (`...`) setelah menyetikkan dalam daftar parameter metodenya. Konsep ini juga didukung oleh function Kotlin dengan penggunaan `vararg` modifier yang diikuti dengan nama parameter.

```
fun printInts(vararg ints: Int): Unit {
    for (n in ints) {
        print("$n\t")    }
    }
printInts(1, 2, 3, 4, 5, 6) // will print 1    2  3 4    5    6
```

`Vararg` modifier memungkinkan pemanggil untuk masuk dalam daftar argumen yang dipisahkan koma. Di balik layar, daftar argumen ini akan dibungkus dalam sebuah array.

Bila sebuah function memiliki beberapa parameter, parameter `vararg` biasanya adalah yang terakhir. Hal ini juga memungkinkan untuk memiliki parameter setelah `vararg`, namun Anda harus menggunakan `named` parameter untuk menentukannya saat Anda memanggil function.

```
fun printNumbers(myDouble: Double, myFloat: Float, vararg
ints: Int) {
    println(myDouble)
    println(myFloat)
    for (n in ints) {
        print("$n\t")
    }
}
printNumbers(1.34, 4.4F, 2, 3, 4, 5, 6) // will compile
```

Misalnya, pada kode di atas, parameter dengan `vararg` modifier berada pada posisi terakhir dalam daftar beberapa parameter (inilah yang biasanya kita lakukan). Tapi bagaimana kalau kita tidak menginginkannya di posisi terakhir? Pada contoh berikut, berada di posisi kedua.

```
fun printNumbers(myDouble: Double, vararg ints: Int, myFloat:
Float) {
    println(myDouble)
    println(myFloat)
    for (n in ints) {
        print("$n\t")
    }
}
printNumbers(1.34, 2, 3, 4, 5, 6, myFloat = 4.4F)
// will compile
printNumbers(1.34, ints = 2, 3, 4, 5, 6, myFloat = 4.4F)
// will not compile
printNumbers(myDouble = 1.34, ints = 2, 3, 4, 5, 6, myFloat =
4.4F)
// will also not
compile
```

Seperti yang dapat Anda amati dalam kode yang diperbarui di atas, kami menggunakan `named` argumen pada parameter terakhir untuk menyelesaikan ini.

- Spread Operator

Misalkan kita ingin melewati array bilangan bulat ke function `printNumbers()`. Function ini mengharapakan nilai yang akan dibuka kembali ke dalam daftar parameter. Jika Anda mencoba untuk melewatkan array langsung ke `printNumbers()`, Anda akan melihat bahwa itu tidak akan di-compile.

```
val intsArray: IntArray = intArrayOf(1, 3, 4, 5)
printNumbers(1.34, intsArray, myFloat = 4.4F) // won't compile
```

Untuk mengatasi masalah ini, kita perlu menggunakan operator penyebaran `*`. Operator ini akan membongkar array dan kemudian melewati elemen individual sebagai argumen ke function untuk kita.

```
val intsArray: IntArray = intArrayOf(1, 3, 4, 5)
printNumbers(1.34, *intsArray, myFloat = 4.4F)
// will now compile
```

Dengan memasukkan operator penyebaran `*` di depan `intsArray` dalam daftar argumen function, kode sekarang dapat di-compile dan menghasilkan hasil yang sama seperti jika kita telah melewati elemen `intsArray` sebagai daftar argumen yang dipisahkan koma.

- Return Multiple Values

Terkadang kita ingin mengembalikan beberapa nilai dari suatu function. Salah satu cara adalah dengan menggunakan tipe [Pair](#) di Kotlin untuk membuat `Pair` dan kemudian mengembalikannya. Struktur `Pair` ini membungkus dua nilai yang nantinya bisa diakses. Tipe Kotlin ini bisa menerima tipe yang Anda berikan pada konstruktornya. Dan, terlebih lagi, kedua tipe ini bahkan tidak perlu sama.

```
fun getUsernameAndState(id: Int): Pair<String?, String?> {
    require(id > 0, { "Error: id is less than 0" })
    val usernames: Map<Int, String> = mapOf(101 to "fkti", 102 to
"ilkom", 104 to "komputer")
    val userStates: Map<Int, String> = mapOf(101 to "ft", 102 to
"elektro", 104 to "teknik")
    val userName = usernames[id]
    val userState = userStates[id]
    return Pair(userName, userState)
}
```

Pada function di atas, kami membuat `Pair` baru dengan melewati variabel `userName` dan `userState` sebagai argumen pertama dan kedua masing-masing ke konstruktornya, dan kemudian mengembalikan `Pair` ini ke pemanggil.

Hal lain yang perlu diperhatikan adalah kita menggunakan function yang disebut `require()` pada function `getUserNameAndState()`. Fungsi penolong ini dari library standar yang digunakan untuk memberi function kepada pemanggil sebuah prasyarat untuk terpenuhi, atau `IllegalArgumentException` akan dilemparkan. Argumen kedua opsional untuk `require()` adalah function literal yang mengembalikan pesan yang akan ditampilkan jika `Exceptions` dilempar. Misalnya, memanggil function `getUserNameAndState()` dan melewati `-1` sebagai argumen untuk itu akan memicu:

- Retrieving Data From Pair

```
val userNameAndStatePair: Pair<String?, String?> =
getUserNameAndState(101)
println(userNameAndStatePair.first) // fk1i
println(userNameAndStatePair.second) // ft
```

Pada kode di atas, kita mengakses nilai `first` dan `second` dari tipe `Pair` dengan menggunakan properti `first` dan `second`.

Namun, ada cara yang lebih baik untuk melakukan hal ini: destrukturisasi.

```
val (name, state) = getUserNameAndState(101)
println(name) // fk1i
println(state) // ft
```

Apa yang telah kita lakukan dalam kode yang diperbarui di atas adalah dengan langsung menetapkan nilai pertama dan kedua dari tipe `Pair` yang dikembalikan ke `name` variabel dan `state` masing-masing. Fitur ini disebut *deklarasi destrukturisasi*.

### Advertisement

- Triple Return Values dan Luar

Nah, bagaimana jika Anda ingin mengembalikan tiga nilai sekaligus? Kotlin menyediakan tipe lain yang berguna yang disebut `Triple`.

```
fun getUserNameStateAndAge(id: Int): Triple<String?, String?,
Int> {
    require(id > 0, { "id is less than 0" })
```

```
    val userNames: Map<Int, String> = mapOf(101 to "fkti", 102
to "ilkom", 104 to "komputer")
    val userStates: Map<Int, String> = mapOf(101 to "ft", 102
to "elektro", 104 to "teknik")

    val userName = userNames[id]
    val userState = userStates[id]
    val userAge = 6
    return Triple(userNames[id], userStates[id], userAge)
}
val (name, state, age) = getUserNameStateAndAge(101)
println(name) // fkti
println(state) // ft
println(age) // 6
```



## YANG HARUS ANDA KETAHUI

Untuk praktik ini Anda harus bisa:

- 1) Cara membuat aplikasi Hello World dengan Android Studio.
- 2) Menjalankan aplikasi di emulator atau perangkat.
- 3) Sudah belajar pemrograman sehingga untuk konsep-konsep mendasar seperti percabangan (if), perulangan (for), atau fungsi sudah cukup memahami.

## YANG AKAN ANDA BUTUHKAN

Untuk materi dan praktik ini, Anda akan membutuhkan:

- 1) Komputer Mac, Windows, atau Linux. Tool Android Studio untuk persyaratan sistem terbaru.
- 2) Akses internet atau cara alternatif untuk memuat Android Studio dan pemasangan Java terbaru ke komputer Anda.

## YANG AKAN ANDA PELAJARI

Anda akan belajar:

- 1) Cara membuat project menampilkan data dengan kotlin.
- 2) Belajar tentang package dan function dasar dalam bahasa pemrograman Kotlin
- 3) Membuat aplikasi sederhana input dan output data.

## YANG HARUS ANDA LAKUKAN

- 1) Memasang dan mengkonfigurasi library kotlin pada Android Studio.
- 2) Membuat aplikasi dan menggunakan elemen antarmuka pengguna seperti tombol dalam Layout Editor.
- 3) Mengimplementasikan metode handler klik untuk tombol agar bisa menampilkan pesan pada layar saat pengguna mengkliknya.

## RINGKASAN APLIKASI

Input dan Output adalah suatu proses mengambil atau menangkap suatu data(nilai atau value) ,yang di lakukan oleh user atau pengguna , yang dimana nantinya akan di terima , lalu di proses oleh sistem ,dalam hal ini bahasa pemrograman.Kemudian hasilnya akan ditampilkan, berdasarkan inputan data(nilai atau value) ,yang dimasukan oleh user atau pengguna.



## Latihan 8.1. Aplikasi pertama dengan Kotlin

Kebutuhan yang perlu disiapkan untuk pengembangan aplikasi Android dengan Kotlin adalah sebagai berikut:

- Android Studio 2.2+
  - Gradle 2.2+
  - Plugin Kotlin 1.0.6+
1. Buatlah proyek Android baru seperti biasa, kemudian atur konfigurasi Gradle. Tambahkan kode di bawah ini untuk **project level build.gradle**:

```
buildscript {
    ext.kotlin_version = '1.0.6'
    dependencies {
        classpath 'com.android.tools.build:gradle:2.2.3'
        classpath "org.jetbrains.kotlin:kotlin-gradle-
plugin:$kotlin_version"
    }
}
```

2. Kemudian tambahkan juga kode berikut untuk **module level build.gradle**:

```
apply plugin: 'com.android.application'
apply plugin: 'kotlin-android'
apply plugin: 'kotlin-android-extensions'
///
dependencies {
    compile "org.jetbrains.kotlin:kotlin-
stdlib:$kotlin_version"
}
```

Setelah konfigurasi diatas telah dibuat dan Gradle sudah selesai disinkronisasi, ubahlah layout pada resource folder yang telah dibuat oleh IDE menjadi seperti di bawah ini:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/activity_main"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context="com.bibitnomic.codepolitan.MainActivity"
android:orientation="vertical">
```

```

<TextView
    android:text="Ketik nama anda "
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textPersonName"
    android:text="Nama"
    android:ems="10"
    android:id="@+id/field_nama" />

<Button
    android:text="Sapa Saya!"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/tombol_sapa" />
</LinearLayout>

```

Untuk MainActivity yang juga sudah dibuat otomatis, akan tampak seperti kode di bawah ini:

```

package com.bibitnomic.codepolitan;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(com.putraxor.codepolitan.R.layout.activity_main
        );
    }
}

```

Untuk mengubahnya ke dalam kode Kotlin, silahkan pilih menu:

[Code > Convert Java To Kotlin] atau tekan [Ctrl+Alt+Shift+K]

Kemudian setelah kode tersebut diubah ke dalam bahasa pemrograman Kotlin, lakukan modifikasi seperti di bawah ini:

```

package com.putraxor.codepolitan

```

```

import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import android.view.View
import android.widget.Toast
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity(), View.OnClickListener {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        setContentView(com.putraxor.codepolitan.R.layout.activity_main
        )
        tombol_sapa.setOnClickListener(this)
    }

    override fun onClick(v: View) {
        if (v.id == R.id.tombol_sapa) {
            var nama = field_nama.text.toString()
            Toast.makeText(this, "Halo $nama, selamat belajar
Kotlin", Toast.LENGTH_SHORT)
        }
    }
}

```

## Penjelasan Kode

Pada baris `import kotlinx.android.synthetic.main.activity_main.*` menggunakan fitur Android Extension. Fungsinya untuk menginjeksi otomatis semua elemen kontrol yang ada pada layout `main_activity.xml`; menggantikan `findViewById()` yang selama ini kita tau sulit untuk dibaca dan berpotensi menimbulkan *bugs*.

Selanjutnya variabel `tombol_sapa` dan `field_nama` yang merupakan ekstensi *property* untuk `MainActivity` yang diambil dari `main_activity.xml`, keduanya memiliki tipe yang sama seperti yang dideklarasikan pada layout.

Sedangkan pada String `"Halo $nama, selamat belajar Kotlin"` merupakan fitur interpolasi String, dimana variable nama dapat langsung dimasukkan pada string tersebut tanpa *concatenation* seperti yang sering dilakukan pada Java. Untuk interpolasi cukup menambahkan tanda `$` pada awal variabel, atau tambahkan *mustache*  `${nama.toString()}`  jika ingin menambahkan pemanggilan fungsi ataupun operasi lebih kompleks.

## Latihan 8.2. Membuat Aplikasi Input dan Output Kotlin Android

Input dan Output adalah suatu proses mengambil atau menangkap suatu data(nilai atau value) ,yang di lakukan oleh user atau pengguna , yang dimana nantinya akan di terima , lalu di proses oleh sistem ,dalam hal ini bahasa pemrograman.Kemudian hasilnya akan ditampilkan, berdasarkan inputan data(nilai atau value) ,yang dimasukan oleh user atau pengguna. Pada latihan ini, kita akan menerapkan Konsep Input dan Output Data (nilai atau value):

- **ButterKnife by Jake Wharton**

Library yang satu ini untuk menggantikan *method* findViewById() yang akan sangat merepotkan bila *project* kita sangat kompleks dengan *layout* yang begitu banyak. Dengan cara @BindView di awal akan memudahkan dalam penulisan *functional code* di Java.

Link GitHub ButterKnife : <https://github.com/JakeWharton/butterknife>

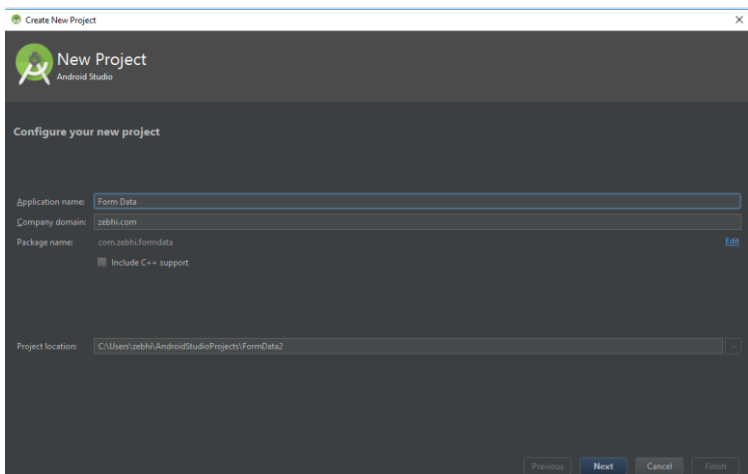
- **MaterialEditText by Kai Zhu (Rengwuxian)**

Library ini juga memudahkan kita dalam membuat *form input* (view EditText). Kita tidak perlu lagi mengatur *coding* satu per satu agar sesuai dengan konsep *material design*. Dengan *library* MaterialEditText, kita bisa mengatur *floating hint*, *icon*, *highlight label*, *max/min* karakter, *helper / error text*, *validation* dan lain sebagainya di view EditText.

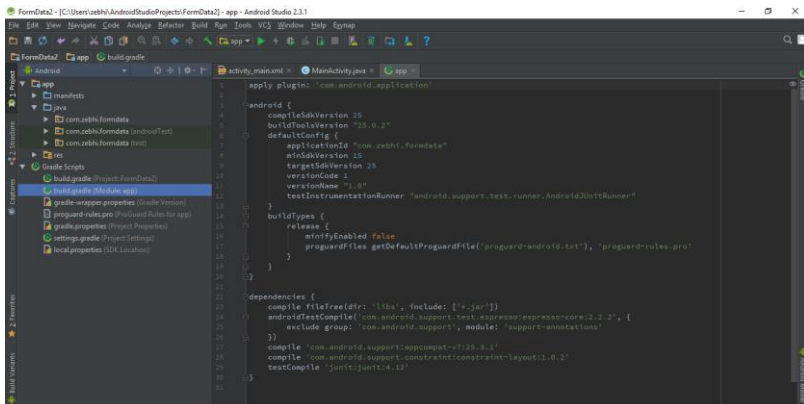
Link GitHub MaterialEditText : <https://github.com/rengwuxian/MaterialEditText>

### 1. Create new project

Pertama-tama kita buat project dahulu. Di sini, saya menggunakan nama project ‘Form Data’. Biarkan yang lainnya default lalu klik next hingga finish.



2. Import ButterKnife dan MaterialEditText library ke dalam Android Studio  
Klik *build.gradle* seperti gambar di bawah ini.

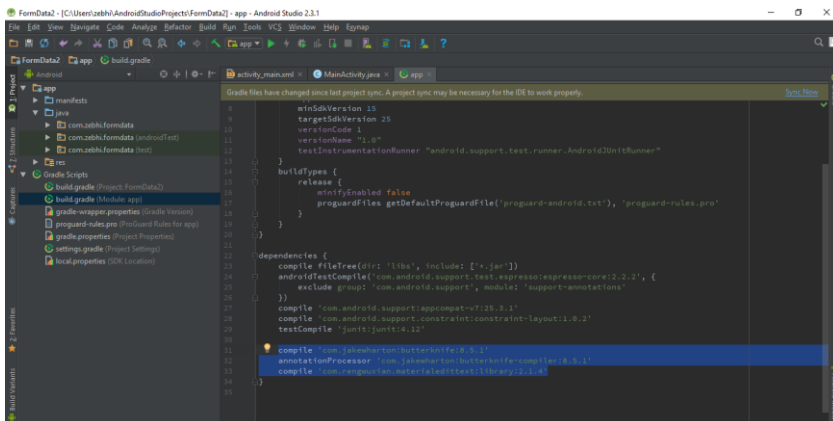


3. Lalu masukkan kode di bawah ke dalam *dependencies* lalu jangan lupa klik 'sync now', tunggu hingga proses *import* berhasil. Sangat disarankan untuk selalu terhubung ke internet karena proses di Android Studio yang sering menggunakan akses internet.

```

compile 'com.jakewharton:butterknife:8.5.1'
annotationProcessor 'com.jakewharton:butterknife-
compiler:8.5.1'
compile
'com.rengwuxian.materialedittext:library:2.1.4'

```



3. Edit layout di *activity\_main.xml*

Langkah selanjutnya kita bisa ke *activity\_main.xml* untuk mengatur tampilan aplikasi Android kita. Di aplikasi *Form Data Input* ini kita akan membuat 5 kolom isian dengan 5 *view* *MaterialEditText*, 1 *view* *Button* dan beberapa *TextView* untuk menampilkan hasil data di bawahnya. *ViewGroup* yang kita gunakan cukup *LinearLayout* dengan sistem *NestedViewGroup* sehingga sesuai dengan keinginan kita dan mudah dalam pengaturannya karena cukup menggunakan attribute *layout\_weight* dengan value 1 (*android:layout\_weight = "1"*) agar terlihat sama rata.

Kemudian Anda bisa mengganti kode *default* yang ada di *activity\_main.xml* dengan kode di bawah ini.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:padding="16dp"
tools:context="com.zebhi.formdata.MainActivity">

<com.rengwuxian.materialedittext.MaterialEditText
    android:id="@+id/input_nama"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Nama"
    android:inputType="textPersonName"
    android:textSize="16sp"
    app:met_floatingLabel="highlight" />
<com.rengwuxian.materialedittext.MaterialEditText
    android:id="@+id/input_nik"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Nomor Induk Kependudukan"
    android:inputType="number"
    android:textSize="16sp"
    app:met_floatingLabel="highlight" />
<com.rengwuxian.materialedittext.MaterialEditText
    android:id="@+id/input_umur"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Usia"
    android:inputType="number"
    android:textSize="16sp"
    app:met_floatingLabel="highlight"
    app:met_maxCharacters="2" />
```

```

<com.rengwuxian.materialEditText.MaterialEditText
    android:id="@+id/input_alamat"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Alamat"
    android:inputType="text"
    android:textSize="16sp"
    app:met_floatingLabel="highlight" />
<com.rengwuxian.materialEditText.MaterialEditText
    android:id="@+id/input_institusi"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Asal Institusi"
    android:inputType="textCapWords"
    android:textSize="16sp"
    app:met_floatingLabel="highlight" />
<Button
    android:id="@+id/simpan_button"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="submit"
    android:text="simpan"
    android:textAllCaps="true"
    android:textSize="16sp" />
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="180dp"
    android:background="#eceff1"
    android:orientation="horizontal">
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:layout_marginLeft="8dp"
        android:layout_marginTop="8dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Nama" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="NIK" />

    <TextView

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Umur" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="Alamat" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="Asal Institusi" />

</LinearLayout>

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="8dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text=":" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text=":" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text=":" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text=":" />

```



```

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text=":" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:layout_marginLeft="8dp"
        android:layout_marginTop="8dp">

        <TextView
            android:id="@+id/hasil_nama"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1" />

        <TextView
            android:id="@+id/hasil_nik"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1" />

        <TextView
            android:id="@+id/hasil_umur"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1" />

        <TextView
            android:id="@+id/hasil_alamat"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1" />

        <TextView
            android:id="@+id/hasil_institusi"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1" />

    </LinearLayout>

</LinearLayout>
</LinearLayout>

```

Bila kita klik *Run* baik menggunakan *emulator* bawaan Android Studio atau *USB Debugging* melalui *real device* maka tampilannya akan seperti ini.

4:08 PM 23%

## Form Data

Nama

Nomor Induk Kependudukan

Usia

Alamat

Asal Institusi

SIMPAN

Nama :  
NIK :  
Umur :  
Alamat :  
Asal Institusi :

#### 4. Memberikan *functional code* di `main_activity.java`

Setelah kita membuat tampilan aplikasi, selanjutnya kita bisa beralih ke `main_activity.java` untuk memberikan *functional code*. Pertama kali kita bisa *import* fungsi ButterKnife atau biarkan saja karena saat *coding* dapat secara otomatis *ter-import*. Lalu kita `@BindView` untuk ID dari EditText di layout

seperti di kode di bawah ini. Lalu jangan lupa tambahkan *ButterKnife.bind(this);* setelah *setContentView*.

Selanjutnya di *@onClick*, bisa meniru kode di bawah dan jangan lupa untuk mengambil setiap data (*getText()*) dari isian dan memunculkan hasilnya (*setText()*).

```
package com.zebhi.formdata;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import butterknife.BindView;
import butterknife.ButterKnife;
import butterknife.OnClick;

public class MainActivity extends AppCompatActivity {

    @BindView(R.id.input_nama)
    EditText inputNama;
    @BindView(R.id.input_nik)
    EditText inputNik;
    @BindView(R.id.input_umur)
    EditText inputUmur;
    @BindView(R.id.input_alamat)
    EditText inputAlamat;
    @BindView(R.id.input_institusi)
    EditText inputInstitusi;

    @BindView(R.id.simpan_button)
    Button simpan_button;

    @BindView(R.id.hasil_nama)
    TextView hasilNama;
    @BindView(R.id.hasil_nik)
    TextView hasilNik;
    @BindView(R.id.hasil_umur)
    TextView hasilUmur;
    @BindView(R.id.hasil_alamat)
    TextView hasilAlamat;
    @BindView(R.id.hasil_institusi)
    TextView hasilInstitusi;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ButterKnife.bind(this);
    }
}
```

```

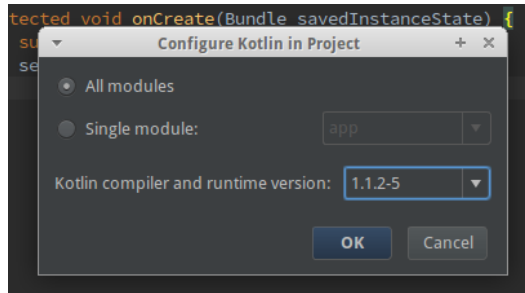
    }
    @OnClick(R.id.simpan_button)
    public void btnClicked() {
        String nama = inputNama.getText().toString();
        hasilNama.setText(nama);
        String nik = inputNik.getText().toString();
        hasilNik.setText(nik);
        String umur = inputUmur.getText().toString();
        hasilUmur.setText(umur);
        String alamat = inputAlamat.getText().toString();
        hasilAlamat.setText(alamat);
        String institusi =
inputInstitusi.getText().toString();
        hasilInstitusi.setText(institusi);
    }
}

```

Bila kita klik *Run* dan aplikasi berjalan, kita dapat mengisi isian di kolom-kolom yang tersedia dan cobalah klik tombol ‘Simpan’ maka akan muncul data di bawahnya.

### Latihan 8.3. Membuat aplikasi konversi suhu sederhana

Silakan buat proyek baru dengan nama **Belajar Kotlin Extension** dan *package name*-nya terserah Anda isi apa. Karena, penulis saat ini pakai Android Studio versi Kotlin yang plugin jadi, setelah proyek selesai dibuat maka, proyek tersebut harus kita *convert* ket proyek Kotlin. Caranya, pilih menu **Tools** kemudian, pilih **Kotlin** dan pilih **Configure Kotlin In Project**. Kemudian, jika muncul *popup* maka, pilih **Android With Gradle**. Dan pilih versi kotlin yang mau Anda pakai.



- Konfigurasi file build.gradle

Pada file **build.gradle(app)**, ketikkan `apply plugin: 'kotlin-android-extensions'` pada bagian paling atas dan pastikan `kotlin-stdlib-jre` punya si Kotlin sudah di masukkan ke *dependencies* dan hasil akhir dari file **build.gradle(app)** akan menjadi seperti berikut.

```
apply plugin: 'com.android.application'
apply plugin: 'kotlin-android'
apply plugin: 'kotlin-android-extensions'

android {
    compileSdkVersion 25
    buildToolsVersion "25.0.2"
    defaultConfig {
        applicationId "com.yesn.belajarkotlinextension"
        minSdkVersion 21
        targetSdkVersion 25
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner
"android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
```

```

        proguardFiles getDefaultProguardFile('proguard-
android.txt'), 'proguard-rules.pro'
    }
}

dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')

androidTestCompile('com.android.support.test.espresso:espresso
-core:2.2.2', {
    exclude group: 'com.android.support', module:
'support-annotations'
})
    compile 'com.android.support:appcompat-v7:25.3.1'
    compile 'com.android.support.constraint:constraint-
layout:1.0.0-beta4'
    testCompile 'junit:junit:4.12'
    compile "org.jetbrains.kotlin:kotlin-stdlib-
jre7:$kotlin_version"
    compile 'com.android.support:design:25.3.1'
    compile 'com.android.support:gridlayout-v7:25.3.1'
}

```

Selain itu, pastikan juga pada file **build.gradle(root)** seperti berikut ini juga.

```

// Top-level build file where you can add configuration
options common to all sub-projects/modules.

buildscript {
    ext.kotlin_version = '1.1.2-3'
    repositories {
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:2.3.1'
        classpath "org.jetbrains.kotlin:kotlin-gradle-
plugin:$kotlin_version"

        // NOTE: Do not place your application dependencies
here; they belong
        // in the individual module build.gradle files
    }
}

allprojects {
    repositories {
        jcenter()
    }
}

```

```

}

task clean(type: Delete) {
    delete rootProject.buildDir
}

```

- Pembuatan Layout

Silakan buka file **activity\_main.xml** dan ubah kode didalamnya menjadi seperti berikut.

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"

tools:context="com.ysn.belajarkotlinextension.MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

android:layout_above="@+id/grid_layout_container_keypad_activi
ty_main"
        android:layout_alignParentTop="true"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_weight="1"
            android:orientation="horizontal"

android:paddingTop="@dimen/activity_vertical_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin">

            <RelativeLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent">

                <Spinner
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"

```

```

android:id="@+id/app_compat_spinner_unit_temperature_from_acti
vity_main"
        android:layout_centerVertical="true"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

android:id="@+id/text_view_value_temperature_from_activity_main"
        android:layout_centerVertical="true"
        android:layout_alignParentEnd="true"
        android:text="0"
        android:textSize="20sp"
        android:textStyle="bold"
        android:textColor="#EF5350"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

android:id="@+id/text_view_unit_temperature_from_activity_main"
        android:text="Celcius"
        android:layout_centerVertical="true"
        android:layout_alignParentEnd="true"

android:layout_below="@+id/text_view_value_temperature_from_ac
tivity_main"/>

    </RelativeLayout>
    </LinearLayout>

    <View
        android:layout_width="match_parent"
        android:layout_height="0.5dp"
        android:background="@android:color/darker_gray"/>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:orientation="horizontal"

android:paddingTop="@dimen/activity_vertical_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin">

```



```

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <Spinner
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"

android:id="@+id/app_compat_spinner_unit_temperature_to_activi
ty_main"android:layout_centerVertical="true"/>

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"

android:id="@+id/text_view_value_temperature_to_activity_main"
                android:layout_centerVertical="true"
                android:text="0"
                android:textSize="20sp"
                android:textStyle="bold"
                android:layout_alignParentEnd="true"
            />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"

android:id="@+id/text_view_unit_temperature_to_activity_main"
                android:text="Celcius"
                android:layout_alignParentEnd="true"

android:layout_below="@+id/text_view_value_temperature_to_acti
vity_main"
            />

        </RelativeLayout>

    </LinearLayout>

</LinearLayout>

<View
    android:layout_width="match_parent"
    android:layout_height="0.5dp"

android:layout_above="@+id/grid_layout_container_keypad_activi
ty_main"
    android:background="#e0e0e0" />

<android.support.v7.widget.GridLayout

```

```

android:id="@+id/grid_layout_container_keypad_activity_main"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    app:alignmentMode="alignBounds"
    app:columnCount="4">

    <!--row 0-->
    <Button
        android:id="@+id/button_keypad_1_activity_main"
        android:layout_width="0dp"
        android:layout_height="wrap_content"

android:background="@drawable/background_button_keypad"
    android:text="1"
    app:layout_column="0"
    app:layout_columnWeight="1"
    app:layout_gravity="fill"
    app:layout_row="0" />

    <Button
        android:id="@+id/button_keypad_2_activity_main"
        android:layout_width="0dp"
        android:layout_height="wrap_content"

android:background="@drawable/background_button_keypad"
    android:text="2"
    app:layout_column="1"
    app:layout_columnWeight="1"
    app:layout_gravity="fill"
    app:layout_row="0" />

    <Button
        android:id="@+id/button_keypad_3_activity_main"
        android:layout_width="0dp"
        android:layout_height="wrap_content"

android:background="@drawable/background_button_keypad"
    android:text="3"
    app:layout_column="2"
    app:layout_columnWeight="1"
    app:layout_gravity="fill"
    app:layout_row="0" />

    <Button
        android:id="@+id/button_keypad_ac_activity_main"
        android:layout_width="0dp"
        android:layout_height="wrap_content"

```

```

android:background="@drawable/background_button_keypad"
    android:text="AC"
    app:layout_column="3"
    app:layout_columnWeight="1"
    app:layout_gravity="fill"
    app:layout_row="0"
    app:layout_rowSpan="2" />

<!--row 1-->
<Button
    android:id="@+id/button_keypad_4_activity_main"
    android:layout_width="0dp"
    android:layout_height="wrap_content"

android:background="@drawable/background_button_keypad"
    android:text="4"
    app:layout_column="0"
    app:layout_columnWeight="1"
    app:layout_gravity="fill"
    app:layout_row="1" />

<Button
    android:id="@+id/button_keypad_5_activity_main"
    android:layout_width="0dp"
    android:layout_height="wrap_content"

android:background="@drawable/background_button_keypad"
    android:text="5"
    app:layout_column="1"
    app:layout_columnWeight="1"
    app:layout_gravity="fill"
    app:layout_row="1" />

<Button
    android:id="@+id/button_keypad_6_activity_main"
    android:layout_width="0dp"
    android:layout_height="wrap_content"

android:background="@drawable/background_button_keypad"
    android:text="6"
    app:layout_column="2"
    app:layout_columnWeight="1"
    app:layout_gravity="fill"
    app:layout_row="1" />

<!--row 2-->
<Button
    android:id="@+id/button_keypad_7_activity_main"
    android:layout_width="0dp"

```

```

        android:layout_height="wrap_content"

android:background="@drawable/background_button_keypad"
        android:text="7"
        app:layout_column="0"
        app:layout_columnWeight="1"
        app:layout_gravity="fill"
        app:layout_row="2" />

<Button
        android:id="@+id/button_keypad_8_activity_main"
        android:layout_width="0dp"
        android:layout_height="wrap_content"

android:background="@drawable/background_button_keypad"
        android:text="8"
        app:layout_column="1"
        app:layout_columnWeight="1"
        app:layout_gravity="fill"
        app:layout_row="2" />

<Button
        android:id="@+id/button_keypad_9_activity_main"
        android:layout_width="0dp"
        android:layout_height="wrap_content"

android:background="@drawable/background_button_keypad"
        android:text="9"
        app:layout_column="2"
        app:layout_columnWeight="1"
        app:layout_gravity="fill"
        app:layout_row="2" />

<Button

android:id="@+id/button_keypad_backspace_activity_main"
        android:layout_width="0dp"
        android:layout_height="wrap_content"

android:background="@drawable/background_button_keypad"
        android:text="X"
        app:layout_column="3"
        app:layout_columnWeight="1"
        app:layout_gravity="fill"
        app:layout_row="2" />

<!--row 3-->
<Button
        android:id="@+id/button_keypad_0_activity_main"
        android:layout_width="0dp"

```

```

        android:layout_height="wrap_content"

android:background="@drawable/background_button_keypad"
        android:text="0"
        app:layout_column="0"
        app:layout_columnSpan="2"
        app:layout_columnWeight="1"
        app:layout_gravity="fill"
        app:layout_row="3" />

<Button
        android:id="@+id/button_keypad_dot_activity_main"
        android:layout_width="0dp"
        android:layout_height="wrap_content"

android:background="@drawable/background_button_keypad"
        android:text="."
        app:layout_column="2"
        app:layout_columnWeight="1"
        app:layout_gravity="fill"
        app:layout_row="3" />

<Button

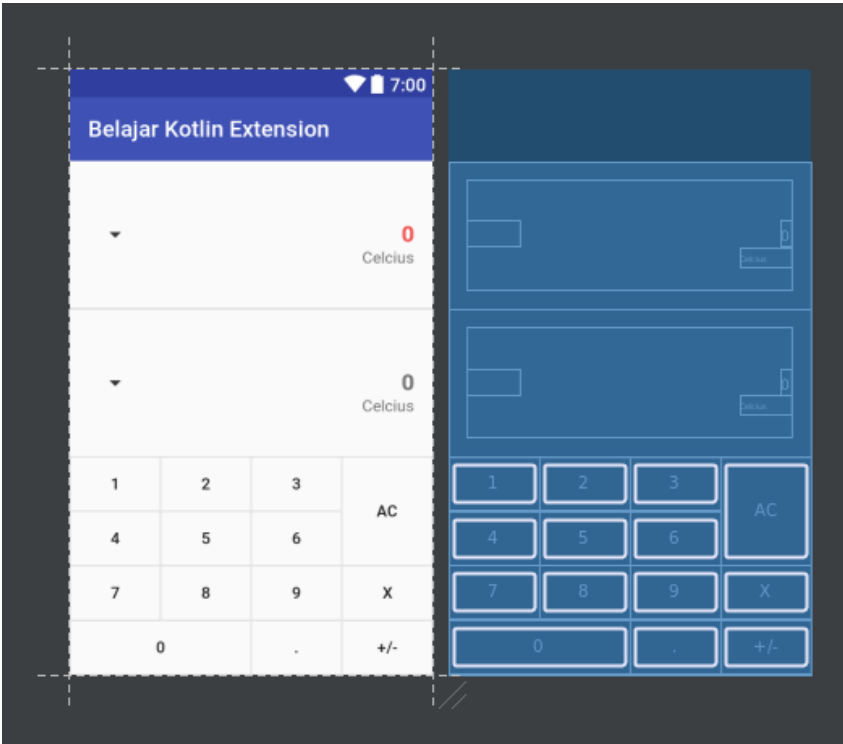
android:id="@+id/button_keypad_plus_minus_activity_main"
        android:layout_width="0dp"
        android:layout_height="wrap_content"

android:background="@drawable/background_button_keypad"
        android:text="+/-"
        app:layout_column="3"
        app:layout_columnWeight="1"
        app:layout_gravity="fill"
        app:layout_row="3" />

</android.support.v7.widget.GridLayout>
</RelativeLayout>

```

Adapun output dari kode layout diatas adalah seperti berikut.



- Pembuatan File Pendukung

Pada file layout diatas, penulis ada menggunakan *custom drawable*. Jadi, silakan buat file *drawable* baru dengan nama **background\_button\_keypad.xml** dan isi dengan *source code* berikut.

```
<?xml version="1.0" encoding="utf-8"?>
<inset
xmlns:android="http://schemas.android.com/apk/res/android">
  <selector>
    <item
      android:state_pressed="true"
      >
      <shape
        android:shape="rectangle"
        >
        <solid
          android:color="#bdbdbd"
          />
        <stroke
          android:width="0.5dp"
```

```

        android:color="#e0e0e0"
    />
</shape>
</item>
<item>
    <shape
        android:shape="rectangle"
    >
        <solid
            android:color="@android:color/transparent"
        />
        <stroke
            android:width="0.5dp"
            android:color="#e0e0e0"
        />
    </shape>
</item>
</selector>
</inset>

```

- **MainActivity.kt**

Di dalam file **MainActivity.kt**, penulis akan menjelaskan beberapa syntax baru yang mungkin pembaca belum tahu apa itu fungsinya. Jadi, saya akan menyampaikannya secara per kode ya.

- **Penulisan Package Name dan Import Library**

```

package com.yasn.belajarkotlinextension

import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import android.util.Log
import android.view.View
import android.widget.AdapterView
import android.widget.AdapterView.OnItemClickListener
import androidx.synthetic.main.activity_main.*

```

Penjelasan: Dalam Kotlin, penulisan *package name* dan *import library* hampir sama bahkan sama persis seperti di Java jadi, tidak ada perbedaannya pada kode diatas ada melakukan import library kotlin extension yang ini import `kotlinx.android.synthetic.main.activity_main.*`

- **Penulisan Nama Class, Extends dan Implements**

```
class MainActivity : AppCompatActivity(),  
View.OnClickListener
```

Penjelasan: Dalam Kotlin, penulisan kelas diawali dengan *keyword class* kemudian dilanjutkan dengan nama kelasnya. Kemudian, untuk melakukan *extends* dan *implements* dari kelas tersebut maka, di Kotlin Anda bisa menggunakan tanda titik dua (:). Seperti yang Anda lihat diatas, bahwa kita ada melakukan *extends* kelas ke **AppCompatActivity** dan *implements* ke **View.OnClickListener**

- Penulisan Nama Variable dan Object

```
private val TAG: String = "MainActivityTAG"  
private var arrayUnitTemperature: Array<String>? = null  
private var temperature: String = "0"
```

Penjelasan:

Dalam Kotlin, penulisan nama variable harus diawali dengan nama variable-nya baru dilanjutkan dengan tipe datanya atau langsung diinisialisasikan juga bisa. Ada 2 tipe penulisan variable di Kotlin yakni, menggunakan *keyword val* dan *var*. Perbedaannya adalah *val* untuk nilai konstan atau *read only* sementara *var* untuk nilai yang dapat diubah. Kemudian, juga bisa Anda lihat ada operator **?** yang mana dalam Kotlin ini ada 2 tipe yakni, **!!** dan **?**. Perbedaannya adalah **!!** untuk mengeksekusi sebuah variable apabila nilainya *non-null* maupun *null* sedangkan, **?** untuk mengeksekusi sebuah variable apabila nilainya *non-null* saja jadi, jelas lebih bagus ini karena bisa mencegah **NPE** atau **Null Pointer Exception**

- Penulisan Method/Function

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.activity_main)  
    initComponents()  
}
```

Penjelasan: Dalam Kotlin, penulisan *method* atau *function* harus menggunakan *keyword fun* kemudian dilanjutkan dengan nama *method* atau *function* nya. Dan apabila *method* atau *function* tersebut mempunyai *return value* maka, setelah tanda tutup kurung diberi tanda titik dua dan dilanjutkan dengan tipe datanya. Contoh:



```

fun hitung(nilai1: Int, nilai2: Int): Int {
    return nilai1.plus(nilai2)
}

```

### ▪ Penulisan Fungsi initComponents

```

private fun initComponents() {
    arrayUnitTemperature = arrayOf("C", "F", "K", "R",
    "Re")
    val arrayUnitDescribeTemperature = arrayOf("Celcius",
    "Fahrenheit", "Kelvin", "Rankine", "Reaumur")
    val arrayAdapterUnitTemperature =
    ArrayAdapter<String>(this,
    android.R.layout.simple_dropdown_item_1line,
    arrayUnitTemperature)

    app_compat_spinner_unit_temperature_from_activity_main.adapter
    = arrayAdapterUnitTemperature

    app_compat_spinner_unit_temperature_to_activity_main.adapter =
    arrayAdapterUnitTemperature

    app_compat_spinner_unit_temperature_to_activity_main.setSelect
    ion(1)
    text_view_unit_temperature_to_activity_main.text =
    "Fahrenheit"

    button_keypad_0_activity_main.setOnClickListener(this)
    button_keypad_1_activity_main.setOnClickListener(this)
    button_keypad_2_activity_main.setOnClickListener(this)
    button_keypad_3_activity_main.setOnClickListener(this)
    button_keypad_4_activity_main.setOnClickListener(this)
    button_keypad_5_activity_main.setOnClickListener(this)
    button_keypad_6_activity_main.setOnClickListener(this)
    button_keypad_7_activity_main.setOnClickListener(this)
    button_keypad_8_activity_main.setOnClickListener(this)
    button_keypad_9_activity_main.setOnClickListener(this)

    button_keypad_backspace_activity_main.setOnClickListener(this)
    button_keypad_ac_activity_main.setOnClickListener(this)
    button_keypad_dot_activity_main.setOnClickListener(this)
    button_keypad_plus_minus_activity_main.setOnClickListener(this
    )

    app_compat_spinner_unit_temperature_from_activity_main.onItems
    electedListener = object : AdapterView.OnItemSelectedListener
    {

```

```

override fun onNothingSelected(p0: AdapterView<*>?) {
    // nothing to do in here
}

override fun onItemSelected(adapterView: AdapterView<*>?,
view: View?, index: Int, id: Long) {

text_view_unit_temperature_from_activity_main.text =
arrayUnitDescribeTemperature.get(index)
    if (temperature != "0" && temperature !=
"0.0") {
        calculateTemperatureConversion()
    }
}

app_compat_spinner_unit_temperature_to_activity_main.onItemSel
ectedListener = object : AdapterView.OnItemSelectedListener {

    override fun onNothingSelected(p0:
AdapterView<*>?) {
        // nothing to do in here
    }

    override fun onItemSelected(adapterView:
AdapterView<*>?, view: View?, index: Int, id: Long) {

text_view_unit_temperature_to_activity_main.text =
arrayUnitDescribeTemperature.get(index)
        if (temperature != "0" && temperature !=
"0.0") {
            calculateTemperatureConversion()
        }
    }
}
}

```

Penjelasan: Dalam kode diatas, kita ada mendeklarasikan sebuah array dengan nilainya String(C, F, R, Re). Kemudian, ada pendeklarasian array lagi untuk kepanjangan dari setiap singkatan suhu yang dideklarasikan sebelumnya pada variable arrayUnitDescribeTemperature. Selain itu, pada kode diatas, penulis ada mendeklarasikan adapter untuk spinner untuk app\_compat\_spinner\_unit\_temperature\_from\_activity\_main dan app\_compat\_spinner\_unit\_temperature\_to\_activity\_main.

Selanjutnya, kita ada memberikan *listener onclick* ke semua **Button** yang ada. Dan *listener onItemSelected* untuk komponen `app_compat_spinner_unit_temperature_from_activity_main` dan `app_compat_spinner_unit_temperature_to_activity_main`.

- **Method Listener OnClick**

```
override fun onClick(view: View?) {
    when (view?.id) {
        R.id.button_keypad_0_activity_main -> {
            updateTemperatureValueFrom("0")
        }
        R.id.button_keypad_1_activity_main -> {
            updateTemperatureValueFrom("1")
        }
        R.id.button_keypad_2_activity_main -> {
            updateTemperatureValueFrom("2")
        }
        R.id.button_keypad_3_activity_main -> {
            updateTemperatureValueFrom("3")
        }
        R.id.button_keypad_4_activity_main -> {
            updateTemperatureValueFrom("4")
        }
        R.id.button_keypad_5_activity_main -> {
            updateTemperatureValueFrom("5")
        }
        R.id.button_keypad_6_activity_main -> {
            updateTemperatureValueFrom("6")
        }
        R.id.button_keypad_7_activity_main -> {
            updateTemperatureValueFrom("7")
        }
        R.id.button_keypad_8_activity_main -> {
            updateTemperatureValueFrom("8")
        }
        R.id.button_keypad_9_activity_main -> {
            updateTemperatureValueFrom("9")
        }
        R.id.button_keypad_backspace_activity_main -> {
            backspaceTemperatureValueFrom()
        }
        R.id.button_keypad_ac_activity_main -> {
            clearTemperatureValueFrom()
        }
        R.id.button_keypad_dot_activity_main -> {
            updateTemperatureValueFrom(".")
        }
    }
}
```

```

        R.id.button_keypad_plus_minus_activity_main -> {
            changePlusMinusValueFrom()
        }
        else -> {
            // nothing to do in here
        }
    }
}

```

Penjelasan: Diatas merupakan method *listener onclick* yang berbentuk implements dari class. Kemudian, didalam method tersebut ada *keyword when* yang mana dalam Java ini merupakan switch. Jadi, `R.id.button_keypad_0_activity_main ->` ini merupakan case untuk setiap item yang ada. Dan `else` pada bagian akhir merupakan default pada switch jika di Java.

## ▪ Pembuatan Method Logic Konversi Suhu

```

private fun changePlusMinusValueFrom() {
    if (temperature.startsWith("-")) {
        temperature = temperature.replace("-", "")
    } else {
        temperature = "-$temperature"
    }
    temperature = if (temperature == "-0") "0" else
    temperature
    text_view_value_temperature_from_activity_main.text =
    temperature
}

private fun clearTemperatureValueFrom() {
    temperature = "0"
    text_view_value_temperature_from_activity_main.text =
    temperature
    text_view_value_temperature_to_activity_main.text =
    "0"
}

private fun backspaceTemperatureValueFrom() {

```

```

        if (temperature != "0") {
            temperature = temperature.substring(0,
temperature.length - 1)
        }
        temperature = if (temperature.isEmpty() || temperature
== "-") "0" else temperature
        text_view_value_temperature_from_activity_main.text =
temperature
        if (temperature == "0" || temperature == "0.0") {
            text_view_value_temperature_to_activity_main.text
= "0"
        }
    }

    private fun updateTemperatureValueFrom(strValue: String) {
        if (strValue == "0") {
            temperature = if (temperature == "0") "0" else
temperature.plus(strValue)
        } else if (strValue == ".") {
            temperature = if (!temperature.contains("."))
temperature.plus(strValue) else temperature
        } else {
            temperature = if (temperature == "0") strValue
else temperature.plus(strValue)
        }
        Log.d(TAG, "temperature: $temperature")
        text_view_value_temperature_from_activity_main.text =
temperature
        if (temperature != "0" && temperature != "0.0") {
            calculateTemperatureConversion()
        } else {
            text_view_value_temperature_to_activity_main.text
= "0"
        }
    }

    private fun calculateTemperatureConversion() {
        val unitTemperatureFrom =
app_compat_spinner_unit_temperature_from_activity_main.selecte
dItem.toString()
        val unitTemperatureTo =
app_compat_spinner_unit_temperature_to_activity_main.selectedI
tem.toString()
        Log.d(TAG, "unitTemperatureTo: $unitTemperatureTo")
        val temperatureFrom = temperature.toDouble()
        var temperatureTo: Double = 0.0
        Log.d(TAG, "temperatureFrom: $temperatureFrom &&
temperatureTo: $temperatureTo && unitTemperatureFrom:
$unitTemperatureFrom && unitTemperatureTo:
$unitTemperatureTo")
    }

```

```

    if (unitTemperatureFrom == "C") {
        when (unitTemperatureTo) {
            "F" -> {
                // C x 1,8 + 32
                temperatureTo =
temperatureFrom.times(1.8).plus(32)
                Log.d(TAG, "temperatureTo new:
$temperatureTo")
            }
            "K" -> {
                // C + 273,15
                temperatureTo =
temperatureFrom.plus(273.15)
            }
            "R" -> {
                // 1,8 x (C + 496,67)
                temperatureTo =
(temperatureFrom.plus(496.67)).times(1.8)
            }
            "Re" -> {
                // C x 0,8
                temperatureTo = temperatureFrom.times(0.8)
            }
            else -> {
                temperatureTo = temperatureFrom
            }
        }
    } else if (unitTemperatureFrom == "F") {
        when (unitTemperatureTo) {
            "C" -> {
                // (F - 32) / 1,8
                temperatureTo =
(temperatureFrom.minus(32)).div(1.8)
            }
            "K" -> {
                // (F + 459,67) / 1,8
                temperatureTo =
(temperatureFrom.plus(459.67)).div(1.8)
            }
            "R" -> {
                // F + 459,67
                temperatureTo =
temperatureFrom.plus(459.67)
            }
            "Re" -> {
                // (F - 32) / 2,25
                temperatureTo =
(temperatureFrom.minus(32)).div(2.25)
            }
        }
    }

```

```

        else -> {
            temperatureTo = temperatureFrom
        }
    }
} else if (unitTemperatureFrom == "K") {
    when (unitTemperatureTo) {
        "C" -> {
            // K - 273,15
            temperatureTo =
temperatureFrom.minus(273.15)
        }
        "F" -> {
            // K x 1,8 - 459,67
            temperatureTo =
temperatureFrom.times(1.8).minus(459.67)
        }
        "R" -> {
            // K x 1,8
            temperatureTo = temperatureFrom.times(1.8)
        }
        "Re" -> {
            // (K - 273,15) x 0,8
            temperatureTo =
(temperatureFrom.minus(273.15)).div(0.8)
        }
        else -> {
            temperatureTo = temperatureFrom
        }
    }
} else if (unitTemperatureFrom == "R") {
    when (unitTemperatureTo) {
        "C" -> {
            // R / 1,8 + 273,15
            temperatureTo =
temperatureFrom.div(1.8).plus(273.15)
        }
        "F" -> {
            // R - 459,67
            temperatureTo =
temperatureFrom.minus(459.67)
        }
        "K" -> {
            // R / 1,8
            temperatureTo = temperatureFrom.div(1.8)
        }
        "Re" -> {
            // (R / 1,8 + 273,15) x 0,8
            temperatureTo =
temperatureFrom.div(1.8).plus(273.15).times(0.8)
        }
    }
}

```

```

        else -> {
            // nothing to do in here
            temperatureTo = temperatureFrom
        }
    }
} else if (unitTemperatureFrom == "Re") {
    when (unitTemperatureTo) {
        "C" -> {
            // Re / 0,8
            temperatureTo = temperatureFrom.div(0.8)
        }
        "F" -> {
            // Re x 2,25 + 32
            temperatureTo =
temperatureFrom.times(2.25).plus(32)
        }
        "K" -> {
            // Re / 0,8 + 273,15
            temperatureTo =
temperatureFrom.div(0.8).plus(273.15)
        }
        "R" -> {
            // Re x 2,25 + 491,67
            temperatureTo =
temperatureFrom.times(2.25).plus(491.67)
        }
        else -> {
            temperatureTo = temperatureFrom
        }
    }
}
text_view_value_temperature_to_activity_main.text =
temperatureTo.toString()
}

```

Penjelasan: Pada method `changePlusMinusValueFrom` berfungsi untuk mengubah nilai negatif menjadi positif dan sebaliknya. Kemudian, pada method `clearTemperatureValueFrom` berfungsi untuk meng-set semua nilai di *view* kembali ke semula atau nol. Method `backspaceTemperatureValueFrom` berfungsi untuk menghapus satu karakter diakhir pada nilai yang mau dikonversi karena, pada app inputan dilakukan menggunakan **button** angka yang tersedia bukan menggunakan *soft keyboard*. Selanjutnya method `updateTemperatureValueFrom` berfungsi untuk melakukan update atau eksekusi lakukan proses konversi suhu apabila nilainya berubah karena, pada app tidak ada button sama dengan jadi,



proses konversi berjalan secara *listener*. Dan yang terakhir adalah method `calculateTemperatureConversion` yang ini berguna untuk melakukan proses konversi dari suatu suhu tertentu menjadi suhu lain.

- Kode Lengkap MainActivity.kt

```
package com.ysn.belajarkotlinextension

import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import android.util.Log
import android.view.View
import android.widget.AdapterView
import android.widget.AdapterView.Adapter
import androidx.core.widget.AdapterView.OnItemClickListener

class MainActivity : AppCompatActivity(), View.OnClickListener {
    private var TAG: String = "MainActivityTAG"
    private var arrayUnitTemperature: Array<String>? = null
    private var temperature: String = "0"

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        initComponents()
    }

    private fun initComponents() {
        arrayUnitTemperature = arrayOf("C", "F", "K", "R",
"Re")
        val arrayUnitDescribeTemperature = arrayOf("Celcius",
"Fahrenheit", "Kelvin", "Rankine", "Reaumur")
        val arrayAdapterUnitTemperature =
ArrayAdapter<String>(this,
android.R.layout.simple_dropdown_item_1line,
arrayUnitTemperature)

app_compat_spinner_unit_temperature_from_activity_main.adapter
= arrayAdapterUnitTemperature

app_compat_spinner_unit_temperature_to_activity_main.adapter =
arrayAdapterUnitTemperature

app_compat_spinner_unit_temperature_to_activity_main.setSelection(1)

        text_view_unit_temperature_to_activity_main.text =
"Fahrenheit"
```

```

        button_keypad_0_activity_main.setOnClickListener(this)
        button_keypad_1_activity_main.setOnClickListener(this)
        button_keypad_2_activity_main.setOnClickListener(this)
        button_keypad_3_activity_main.setOnClickListener(this)
        button_keypad_4_activity_main.setOnClickListener(this)
        button_keypad_5_activity_main.setOnClickListener(this)
        button_keypad_6_activity_main.setOnClickListener(this)
        button_keypad_7_activity_main.setOnClickListener(this)
        button_keypad_8_activity_main.setOnClickListener(this)
        button_keypad_9_activity_main.setOnClickListener(this)

button_keypad_backspace_activity_main.setOnClickListener(this)

button_keypad_ac_activity_main.setOnClickListener(this)

button_keypad_dot_activity_main.setOnClickListener(this)

button_keypad_plus_minus_activity_main.setOnClickListener(this
)

app_compat_spinner_unit_temperature_from_activity_main.onItems
electedListener = object : AdapterView.OnItemClickListener
{
    override fun onNothingSelected(p0: AdapterView<*>?) {
        // nothing to do in here
    }

    override fun onItemClick(adapterView:
AdapterView<*>?, view: View?, index: Int, id: Long) {

text_view_unit_temperature_from_activity_main.text =
arrayUnitDescribeTemperature.get(index)
        if (temperature != "0" && temperature !=
"0.0") {
            calculateTemperatureConversion()
        }
    }
}

app_compat_spinner_unit_temperature_to_activity_main.onItemSel
ectedListener = object : AdapterView.OnItemClickListener {

override fun onNothingSelected(p0: AdapterView<*>?) {
    // nothing to do in here
}

    override fun onItemClick(adapterView:
AdapterView<*>?, view: View?, index: Int, id: Long) {

```

```

text_view_unit_temperature_to_activity_main.text =
arrayUnitDescribeTemperature.get(index)
        if (temperature != "0" && temperature !=
"0.0") {
                calculateTemperatureConversion()
        }
    }
}

override fun onClick(view: View?) {
    when (view?.id) {
        R.id.button_keypad_0_activity_main -> {
            updateTemperatureValueFrom("0")
        }
        R.id.button_keypad_1_activity_main -> {
            updateTemperatureValueFrom("1")
        }
        R.id.button_keypad_2_activity_main -> {
            updateTemperatureValueFrom("2")
        }
        R.id.button_keypad_3_activity_main -> {
            updateTemperatureValueFrom("3")
        }
        R.id.button_keypad_4_activity_main -> {
            updateTemperatureValueFrom("4")
        }
        R.id.button_keypad_5_activity_main -> {
            updateTemperatureValueFrom("5")
        }
        R.id.button_keypad_6_activity_main -> {
            updateTemperatureValueFrom("6")
        }
        R.id.button_keypad_7_activity_main -> {
            updateTemperatureValueFrom("7")
        }
        R.id.button_keypad_8_activity_main -> {
            updateTemperatureValueFrom("8")
        }
        R.id.button_keypad_9_activity_main -> {
            updateTemperatureValueFrom("9")
        }
        R.id.button_keypad_backspace_activity_main -> {
            backspaceTemperatureValueFrom()
        }
        R.id.button_keypad_ac_activity_main -> {
            clearTemperatureValueFrom()
        }
    }
}

```

```

        R.id.button_keypad_dot_activity_main -> {
            updateTemperatureValueFrom(".")
        }
        R.id.button_keypad_plus_minus_activity_main -> {
            changePlusMinusValueFrom()
        }
        else -> {
            // nothing to do in here
        }
    }
}

private fun changePlusMinusValueFrom() {
    // do something
    if (temperature.startsWith("-")) {
        temperature = temperature.replace("-", "")
    } else {
        temperature = "-$temperature"
    }
    temperature = if (temperature == "-0") "0" else
temperature
    text_view_value_temperature_from_activity_main.text =
temperature
}

private fun clearTemperatureValueFrom() {
    temperature = "0"
    text_view_value_temperature_from_activity_main.text =
temperature
    text_view_value_temperature_to_activity_main.text =
"0"
}

private fun backspaceTemperatureValueFrom() {
    if (temperature != "0") {
        temperature = temperature.substring(0,
temperature.length - 1)
    }
    temperature = if (temperature.isEmpty() || temperature
== "-") "0" else temperature
    text_view_value_temperature_from_activity_main.text =
temperature
    if (temperature == "0" || temperature == "0.0") {
        text_view_value_temperature_to_activity_main.text
= "0"
    }
}

private fun updateTemperatureValueFrom(strValue: String) {
    if (strValue == "0") {

```

```

        temperature = if (temperature == "0") "0" else
temperature.plus(strValue)
    } else if (strValue == ".") {
        temperature = if (!temperature.contains("."))
temperature.plus(strValue) else temperature
    } else {
        temperature = if (temperature == "0") strValue
else temperature.plus(strValue)
    }
    Log.d(TAG, "temperature: $temperature")
    text_view_value_temperature_from_activity_main.text =
temperature
    if (temperature != "0" && temperature != "0.0") {
        calculateTemperatureConversion()
    } else {
        text_view_value_temperature_to_activity_main.text
= "0"
    }
}

private fun calculateTemperatureConversion() {
    val unitTemperatureFrom =
app_compat_spinner_unit_temperature_from_activity_main.selecte
dItem.toString()
    val unitTemperatureTo =
app_compat_spinner_unit_temperature_to_activity_main.selectedI
tem.toString()
    Log.d(TAG, "unitTemperatureTo: $unitTemperatureTo")
    val temperatureFrom = temperature.toDouble()
    var temperatureTo: Double = 0.0
    Log.d(TAG, "temperatureFrom: $temperatureFrom &&
temperatureTo: $temperatureTo && unitTemperatureFrom:
$unitTemperatureFrom && unitTemperatureTo:
$unitTemperatureTo")

    if (unitTemperatureFrom == "C") {
        when (unitTemperatureTo) {
            "F" -> {
                // C x 1,8 + 32
                temperatureTo =
temperatureFrom.times(1.8).plus(32)
                Log.d(TAG, "temperatureTo new:
$temperatureTo")
            }
            "K" -> {
                // C + 273,15
                temperatureTo =
temperatureFrom.plus(273.15)
            }
            "R" -> {

```

```

        // 1,8 x (C + 496,67)
        temperatureTo =
    (temperatureFrom.plus(496.67)).times(1.8)
    }
    "Re" -> {
        // C x 0,8
        temperatureTo = temperatureFrom.times(0.8)
    }
    else -> {
        temperatureTo = temperatureFrom
    }
}
} else if (unitTemperatureFrom == "F") {
    when (unitTemperatureTo) {
        "C" -> {
            // (F - 32) / 1,8
            temperatureTo =
    (temperatureFrom.minus(32)).div(1.8)
        }
        "K" -> {
            // (F + 459,67) / 1,8
            temperatureTo =
    (temperatureFrom.plus(459.67)).div(1.8)
        }
        "R" -> {
            // F + 459,67
            temperatureTo =
    temperatureFrom.plus(459.67)
        }
        "Re" -> {
            // (F - 32) / 2,25
            temperatureTo =
    (temperatureFrom.minus(32)).div(2.25)
        }
        else -> {
            temperatureTo = temperatureFrom
        }
    }
} else if (unitTemperatureFrom == "K") {
    when (unitTemperatureTo) {
        "C" -> {
            // K - 273,15
            temperatureTo =
    temperatureFrom.minus(273.15)
        }
        "F" -> {
            // K x 1,8 - 459,67
            temperatureTo =
    temperatureFrom.times(1.8).minus(459.67)
        }
    }
}
}

```

```

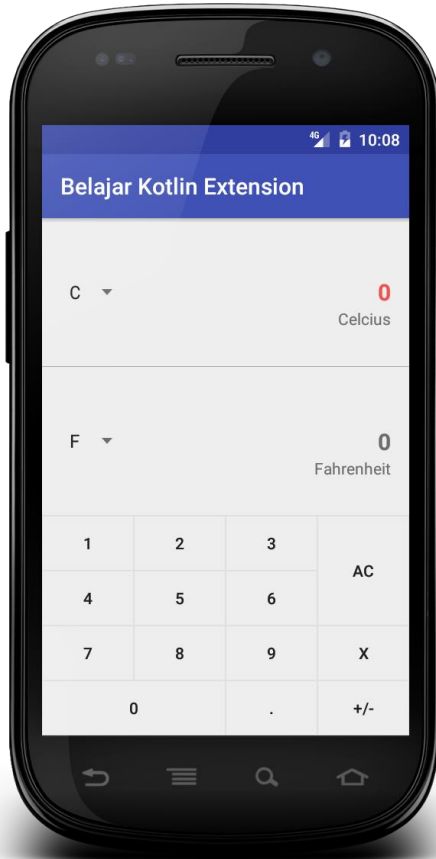
        "R" -> {
            // K x 1,8
            temperatureTo = temperatureFrom.times(1.8)
        }
        "Re" -> {
            // (K - 273,15) x 0,8
            temperatureTo =
temperatureFrom.minus(273.15).div(0.8)
        }
        else -> {
            temperatureTo = temperatureFrom
        }
    }
} else if (unitTemperatureFrom == "R") {
    when (unitTemperatureTo) {
        "C" -> {
            // R / 1,8 + 273,15
            temperatureTo =
temperatureFrom.div(1.8).plus(273.15)
        }
        "F" -> {
            // R - 459,67
            temperatureTo =
temperatureFrom.minus(459.67)
        }
        "K" -> {
            // R / 1,8
            temperatureTo = temperatureFrom.div(1.8)
        }
        "Re" -> {
            // (R / 1,8 + 273,15) x 0,8
            temperatureTo =
temperatureFrom.div(1.8).plus(273.15).times(0.8)
        }
        else -> {
            // nothing to do in here
            temperatureTo = temperatureFrom
        }
    }
} else if (unitTemperatureFrom == "Re") {
    when (unitTemperatureTo) {
        "C" -> {
            // Re / 0,8
            temperatureTo = temperatureFrom.div(0.8)
        }
        "F" -> {
            // Re x 2,25 + 32
            temperatureTo =
temperatureFrom.times(2.25).plus(32)
        }
    }
}

```

```
        "K" -> {
            // Re / 0,8 + 273,15
            temperatureTo =
temperatureFrom.div(0.8).plus(273.15)
        }
        "R" -> {
            // Re x 2,25 + 491,67
            temperatureTo =
temperatureFrom.times(2.25).plus(491.67)
        }
        else -> {
            temperatureTo = temperatureFrom
        }
    }
}
    text_view_value_temperature_to_activity_main.text =
temperatureTo.toString()
}
}
```

## Output





Pada bab ini, Anda telah mempelajari cara:

Menambahkan elemen UI ke aplikasi dalam Layout Editor dan menggunakan XML.

Membuat interaktif UI dengan tombol dan mengeklik listener.

Menambahkan listener klik yang memperbarui tampilan teks untuk merespons masukan pengguna.

Informasi yang ditampilkan ke pengguna menggunakan toast.

## REFERENSI

- Buku Kursus Dasar-Dasar Developer Android – Konsep, Belajar mengembangkan aplikasi Android, Dikembangkan oleh Tim Pelatihan Developer Google.
- Buku Kerja Praktik, Kursus Dasar-Dasar Developer Android — Praktik, Belajar mengembangkan aplikasi Android, Dikembangkan oleh Tim Pelatihan Developer Google
- Safaat N.H. 2014. Android: Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android. Bandung: Informatika.

Meskipun jurusan IT merupakan salah satu jurusan paling favorit di Indonesia nyatanya *output* yang dihasilkan belum bisa memenuhi kebutuhan industri. Di era persaingan global seperti sekarang ini, jika kita tidak berbenah diri maka developer dari luar-lah yang akan masuk dan menggeser posisi kita.

Bagi mahasiswa IT, *coding* tentu sudah menjadi makanan sehari-hari sehingga yang bisa dilakukan saat ini adalah meningkatkan kemampuan dengan *skill* yang dibutuhkan oleh industri. Lalu bagaimana dengan orang yang berasal dari luar dunia IT? Jika pembaca saat ini merupakan orang yang berasal dari dunia IT tapi ingin memasuki industri ini, maka sekaranglah saatnya untuk mulai belajar coding.

Belajar *coding* memang bukan hal yang mudah (kalau *coding* mudah tentu perusahaan startup tidak akan kesulitan mencari talenta yang mereka perlukan). Namun, apabila memiliki motivasi dan semangat belajar yakinlah bahwa pembaca juga dapat bersaing dan sudah cukup banyak contoh orang non IT yang nyatanya bisa sukses.

Bagi pembaca yang sudah pernah kuliah sebelumnya (di jurusan lain) dan saat ini sudah bekerja tentu saja kuliah lagi menjadi pilihan yang kurang pas karena akan memakan waktu lebih lama dan biaya yang lebih besar. Lalu apa solusi lain bagi orang non IT yang ingin masuk ke industri IT dalam durasi waktu lebih pendek dan biaya yang lebih terjangkau?

Kamu dapat belajar coding secara otodidak. Baik melalui buku ini maupun banyak sekali sumber belajar pemrograman yang dapat Anda akses mulai dari yang gratis maupun yang berbayar di internet. Untuk belajar pemrograman mobile android, dalam buku ini telah penulis sajikan mulai dari pengenalan dasar hingga ke pembuatan project Aplikasi Android skala yang siap dikembangkan, dan diharapkan dapat menjadi salahsatu referensi baik untuk modul pembelajaran di perkuliahan maupun pembeajaran umum.

# BIOGRAFI PENULIS



Penulis adalah staff pengajar di Program Studi Teknik Informatika Universitas Mulawarman Samarinda sejak tahun 2012. Lahir di Sidenreng Rappang (Sidrap) Sulawesi Selatan dan menyelesaikan studi Magister di Universitas Hasanuddin Makassar.

Pengalaman mengajar di matakuliah komputer sejak tahun 2002 di Universitas Negeri Makassar, tahun 2008 mengajar di Universitas Sulawesi Barat dan Universitas Muslim Indonesia tahun 2011. Matakuliah yang diampu lebih pada matakuliah pemrograman, dari pemrograman prosedural, non procedural, object, web hingga pemrograman mobile penulis.

Saat ini penulis lebih banyak melakukan riset-riset terkait dengan implikasi dan performance perangkat-aplikasi mobile dalam jaringan telekomunikasi yang sampai saat ini perkembangan teknologinya sangat pesat yang berdampak pada perilaku gaya hidup masyarakat generasi **Zaman Now**.

“Kata **TIDAK BISA**, akan membuatmu berhenti berusaha, Terus Berkarya dan Tetap Semangat” adalah motto penulis yang selalu disampaikan kepada seluruh mahasiswa agar mereka senantiasa mampu melewati segala tantangan dan persaingan yang ada.