

MODUL PRAKTIKUM
SISTEM LINIER



DISUSUN OLEH:
Ir. Muslimin, S.T., M.T., IPM., ASEAN Eng.

PROGRAM STUDI S1 TEKNIK ELEKTRO
FAKULTAS TEKNIK

UNIVERSITAS MULAWARMAN
TAHUN 2021

KATA PENGANTAR

Puji Syukur penulis panjatkan ke hadirat Allah SWT yang telah memberikan kekuatan lahir dan batin sehingga penulis dapat menyelesaikan penulisan modul Praktikum Sistem Linier. Modul ini merupakan modul praktikum Sistem Linier pada mata kuliah yang sama berbasis *Problem Solving* yang ditujukan untuk membantu pemahaman mahasiswa dalam menempuh mata kuliah praktikum Sistem Linier.

Susunan setiap judul praktikum memuat tentang tujuan percobaan, alat percobaan, dasar teori, prosedur percobaan, laporan akhir percobaan, tugas pendahuluan sebelum praktikum, dan lembar pengamatan. Setelah menggunakan modul ini diharapkan pemahaman tentang materi Sistem Linier dapat diterima dengan mudah dan baik.

Penulis mengucapkan terima kasih kepada semua pihak yang telah membantu penyusunan modul ini. Namun, modul ini masih jauh dari kata sempurna dan masih terdapat kekurangan. Oleh karena itu, penulis menerima segala kritik dan saran yang sifatnya konstruktif dan bertujuan untuk memperbaiki modul ini pada masa yang akan datang.

Penulis,



Ir. Muslimin, S.T., M.T., IPM., ASEAN Eng.

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI	ii
MODUL I MACAM-MCAM SINYAL DAN PEMBANGKITANNYA.....	1
1.1 Dasar Teori.....	1
1.2 Peralatan.....	12
1.3 Langkah Percobaan.....	12
1.4 Data dan Analisis	21
1.5 Tugas.....	22
MODUL II OPERASI DASAR PADA SINYAL.....	23
2.1 Dasar Teori.....	23
2.2 Peralatan.....	28
2.3 Langkah Percobaan.....	28
2.4 Data dan Analisis	36
2.5 Tugas.....	36
MODUL III SAMPLING DAN ALIASING.....	37
3.1 Dasar Teori	37
3.2 Peralatan.....	42
3.3 Langkah Percobaan.....	42
3.4 Data dan Analisis	49
3.5 Tugas.....	49
MODUL IV OPERASI KONVOLUSI	50
4.1 Dasar Teori	50
4.2 Peralatan.....	53
4.3 Langkah Percobaan.....	53
4.4 Data dan Analisis	62
4.5 Tugas.....	62
DAFTAR PUSTAKA.....	63

MODUL I

MACAM-MACAM SINYAL DAN PEMBANGKITANNYA

Tujuan Praktikum:

1. Mahasiswa mengetahui dan memahami tentang bermacam-macam sinyal dasar yang digunakan dalam pengolahan sinyal digital.
2. Mahasiswa dapat merancang program untuk membangkitkan beberapa jenis sinyal dasar yang banyak digunakan dalam analisis Sinyal dan Sistem.

1.1. Dasar Teori

Dalam Akuisisi Data dan Pengolahan Sinyal, beberapa jenis sinyal banyak dimanfaatkan sebagai pengujian suatu sistem untuk mengetahui tanggapannya. Beberapa jenis sinyal dasar ini dapat dibangkitkan melalui sebuah program.

1.1.1 Sinyal

Sinyal adalah suatu isyarat atau pemberitahuan yang dapat ditangkap oleh indera untuk kepentingan penyampaian peringatan, petunjuk, atau informasi. Sinyal merupakan sebuah fungsi yang berisi informasi mengenai keadaan tingkah laku dari sebuah sistem secara fisik. Meskipun sinyal dapat diwujudkan dalam beberapa cara, dalam berbagai kasus, informasi terdiri dari sebuah pola dari beberapa bentuk yang bervariasi. Sebagai contoh sinyal mungkin berbentuk sebuah pola dari banyak variasi waktu atau sebagian saja.

Secara matematis, sinyal merupakan fungsi dari satu atau lebih variable yang tidak bergantung (*independent variable*). Sebagai contoh, sinyal wicara akan dinyatakan secara matematis oleh tekanan akustik sebagai fungsi waktu dan sebuah gambar dinyatakan sebagai fungsi ke-tajaman-an (*brightness*) dari dua variable ruang (*spatial*).

Untuk analisis, sebuah sinyal dapat didefinisikan sebagai sebuah fungsi matematika yang secara umum dapat ditulis sebagai berikut:

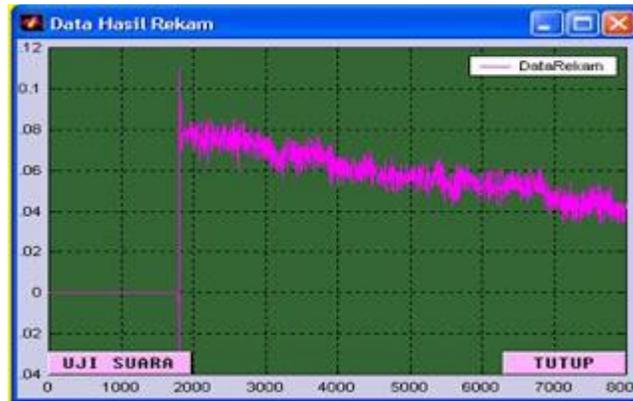
$$y = f(x)$$

dengan x adalah variabel atau peubah yang independen (nilainya tidak bergantung pada nilai peubah lain) dan y (sinyal) merupakan peubah yang tidak independen (dalam hal ini nilai y

bergantung pada nilai x . Peubah independen menentukan **domain** (daerah asal) dari sinyal, misalnya:

1. $y = \sin(\omega t)$ adalah suatu fungsi dengan variabel dalam domain waktu (*time-domain*) t sehingga merupakan sinyal yang berubah terhadap waktu (*time-signal*).
2. $x(\omega) = 1/(-m\omega^2 + ic\omega + k)$ adalah sinyal yang mempunyai domain frekuensi yaitu ω atau disebut *frequency-domain signal*.
3. Intensitas citra (*image*) $I_{(x,y)}$ merupakan sinyal yang mempunyai domain spasial atau disebut *spasial-domain signal*.

Pada Gambar 2.1, ditunjukkan contoh sinyal fungsi waktu yang diperoleh dari sebuah rekaman audio.



Gambar 1.1. Contoh Sinyal Audio

Secara umum, variable yang tidak bergantung (*independent*) secara matematis diwujudkan dalam fungsi waktu, meskipun sebenarnya tidak menunjukkan waktu. Terdapat 2 tipe dasar sinyal, yaitu:

1. Sinyal waktu kontinyu (*continous-time signal*)
2. Sinyal waktu diskrit (*discrete-time signal*)

Pada sinyal kontinyu, variable independen terjadi terus-menerus dan kemudian sinyal dinyatakan sebagai sebuah kesatuan nilai dari variable independen. Sebaliknya, sinyal diskrit hanya menyatakan waktu diskrit dan mengakibatkan variabel independen hanya merupakan himpunan nilai diskrit.

Fungsi sinyal dinyatakan sebagai x dengan menyertakan variable dalam tanda (.). Untuk membedakan antara sinyal waktu kontinu dengan sinyal waktu diskrit digunakan symbol t untuk menyatakan variable kontinu dan simbol n untuk menyatakan variable diskrit. Sebagai contoh sinyal waktu kontinu dinyatakan dengan fungsi $x(t)$ dan sinyal waktu diskrit dinyatakan dengan fungsi $x(n)$. Sinyal waktu diskrit hanya menyatakan nilai integer dari variable independen.

1.1.2. Sinyal Waktu Kontinu

Suatu sinyal $x(t)$ dikatakan sebagai sinyal waktu-kontinu atau sinyal analog ketika memiliki nilai riil pada keseluruhan rentang waktu t yang ditematinya. Sinyal waktu kontinu dapat didefinisikan dengan persamaan matematis sebagai berikut.

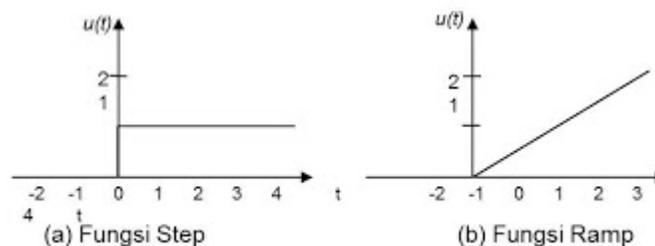
$$f(t) \approx (-\infty, \infty) (1)$$

Fungsi Step (undak) dan Fungsi Ramp (tanjak)

Berikut ini ditunjukkan dua contoh sederhana pada sinyal kontinu yang memiliki fungsi *step* (undak) dan fungsi *ramp* (tanjak). Sebuah fungsi *step* seperti pada Gambar 2.2a, dapat diwakili dengan suatu bentuk matematis sebagai:

$$u(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases}$$

Di sini fungsi undak (*step*) memiliki arti bahwa amplitudo pada $u(t)$ bernilai nol pada $t < 0$ dan bernilai satu untuk semua $t \geq 0$.



Gambar 1.2. Fungsi *Step* dan Fungsi *Ramp* Sinyal Waktu Kontinu

Untuk suatu sinyal waktu-kontinu $x(t)$, hasil kali $x(t)u(t)$ sebanding dengan $x(t)$ untuk $t > 0$ dan sebanding dengan nol untuk $t < 0$. Perkalian pada sinyal $x(t)$ dengan sinyal $u(t)$ mengeliminasi suatu nilai *non-zero* (bukan nol) pada $x(t)$ untuk nilai $t < 0$.

Fungsi *ramp* (tanjak) $r(t)$ didefinisikan secara matematik sebagai:

$$r(t) = \begin{cases} t, & t \geq 0 \\ 0, & t < 0 \end{cases}$$

Perhatikan bahwa untuk $t > 0$, *slope* (kemiringan) pada $r(t)$ adalah senilai 1. Sehingga pada kasus ini $r(t)$ merupakan “*unit slope*”, yang mana merupakan alasan bagi $r(t)$ untuk dapat disebut sebagai *unit-ramp function*. Jika ada variable K sedemikian hingga membentuk $Kr(t)$, maka *slope* yang dimilikinya adalah K untuk $t > 0$. Contoh bentuk gelombang fungsi *ramp* ditunjukkan pada Gambar 2b.

Sinyal Periodik

Sinyal-sinyal periodik muncul secara alamiah dalam sistem-sistem dimana terjadi rotasi (putaran), misalnya roda gigi atau *gearboxes*. Pada pembahasan lain akan diperlihatkan bagaimana sinyal-sinyal periodik dapat digunakan untuk mempermudah pengujian sebuah sistem; biasanya respon sistem juga berupa sinyal periodik (untuk sistem linier). Dengan demikian sinyal deterministik dapat dibagi menjadi dua jenis sinyal yaitu sinyal periodik dan sinyal aperiodik. **Sinyal aperiodik** merupakan sinyal yang tidak memenuhi sifat periodik.



Gambar 1.3 Klasifikasi sinyal deterministik; periodik dan aperiodik

Ditetapkan T sebagai perioda berupa nilai real positif. Suatu sinyal waktu kontinyu $x(t)$ dikatakan periodik terhadap waktu dengan periode T jika

$$x(t + T) = x(t) \text{ untuk semua nilai } t, -\infty < t < \infty$$

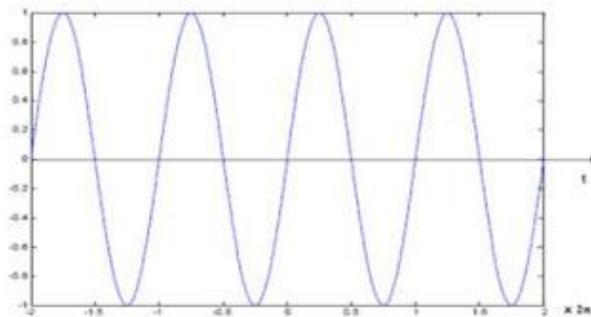
Sebagai catatan, jika $x(t)$ merupakan periodik pada periode T , ini juga periodik dengan qT , dimana q merupakan nilai integer positif. Periode fundamental merupakan nilai positif terkecil T untuk persamaan (5). Suatu contoh, sinyal periodik memiliki persamaan seperti berikut:

$$x(t) = A \cos(\omega t + \theta)$$

Disini A adalah amplitudo, ω adalah frekuensi dalam radian per detik (rad/detik), dan θ adalah fase dalam radian. Frekuensi f dalam hertz (Hz) atau siklus per detik adalah sebesar $f = \omega/2\pi$. Untuk melihat bahwa suatu fungsi sinusoida yang diberikan dalam persamaan berikut adalah fungsi periodik, untuk nilai pada variable waktu t , maka:

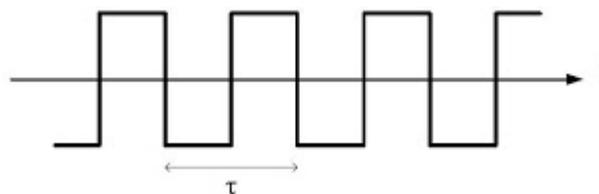
$$A \cos \left[\omega \left(t + \frac{2\pi}{\omega} \right) + \theta \right] = A \cos(\omega t + 2\pi + \theta) = A \cos(\omega t + \theta)$$

Sedemikian hingga fungsi sinusoida merupakan fungsi periodik dengan periode $2\pi/\omega$, nilai ini selanjutnya dikenal sebagai periode fundamentalnya. Sebuah sinyal dengan fungsi sinusoida $x(t) = A \cos(\omega t + \theta)$ diberikan pada Gambar 2.4, untuk nilai $\theta = -\pi/2$, dan $f = 1$ Hz.



Gambar 1.4. Sinyal Periodik Sinusida

Perlu diperhatikan bahwa periodisitas tidak berarti harus berbentuk sinusoidal, perhatikan gelombang kotak berikut ini yang juga merupakan sinyal periodik.



Gambar 1.5. Gelombang kotak; sebuah sinyal periodik

Dalam pembahasan yang lanjut nantinya akan dijelaskan bahwa gelombang kotak mempunyai bentuk fungsional (yaitu deret Fourier) sebagai berikut.

$$S(t) = \sum_{k=1}^{\infty} a_k \sin\left(\frac{2\pi k}{\tau} t\right)$$

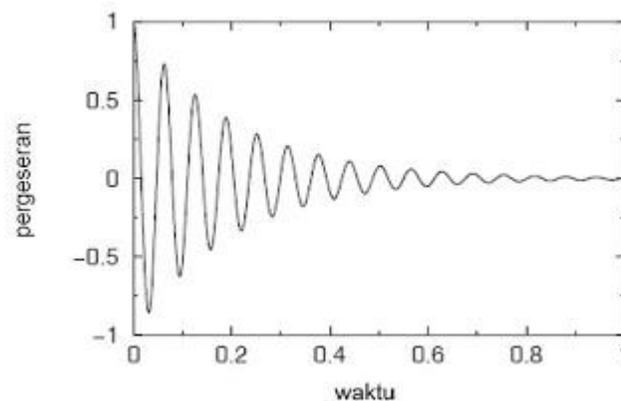
yaitu bahwa gelombang kotak tersusun atas superposisi (jumlahan) tak terhingga gelombang sinus dengan periode τ , $\tau/2$, $\tau/3$, Hal terpenting yang terlihat di sini adalah bahwa komponen dengan periode $\tau/2$, $\tau/3$, dst terus berulang setelah waktu τ yaitu $S(t) = S(t + \tau)$ dan gelombang kotak pada persamaan (1.7) merupakan sinyal periodik. Dari kenyataan ini sinyal periodik dapat dibagi menjadi dua jenis yaitu **periodik frekuensi tunggal (monofrequency)** dan **multifrekuensi (multifrequency)**.



Gambar 1.6 Klasifikasi Sinyal Periodik; Frekuensi Tunggal dan Multifrekuensi

Periodisitas yang dinyatakan oleh persamaan (1.5) dan (1.6) mengimplikasikan bahwa sinyal periodik berulang sepanjang waktu, sehingga dengan demikian jenis sinyal aperiodik merupakan sinyal yang hanya ada pada suatu waktu yang tertentu saja. Misalkan tanggapan pendulum teredam yang dilepaskan dari posisi setimbang $\theta = \theta_0$ pada saat $t = 0$ yang dinyatakan dengan persamaan berikut.

$$y(t) = \theta_0 e^{-ct} \cos(\omega t)$$



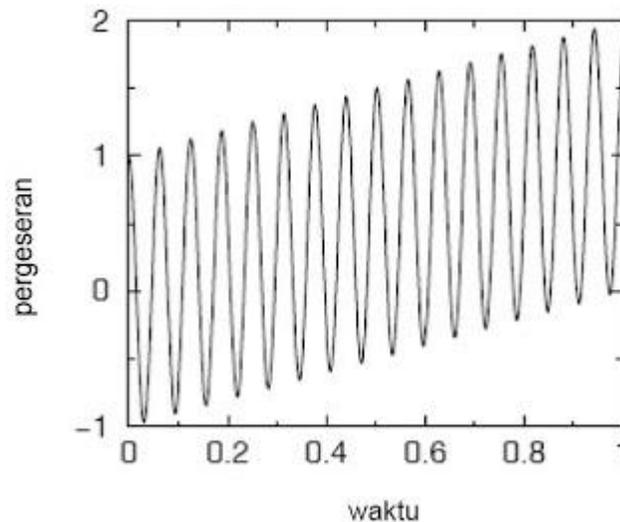
Gambar 1.7 Respon Transien

Sinyal ini hanya ada untuk waktu yang terhingga saja. Meskipun nilai $e^{-ct} > 0$, untuk sebarang waktu t , sistem riil akan dibawa ke titik setimbang oleh efek pangkat-tinggi (*higher order*) daripada redaman. Respon sistem ini diperlihatkan pada Gambar 2.7.

Sinyal-sinyal yang hanya ada pada suatu jangkauan waktu yang tertentu saja disebut **transien**. Beberapa contoh untuk sinyal semacam ini adalah respon sistem terhadap eksitasi berupa impuls, sebuah ledakan, atau petir (halilintar). Kebalikan dari sebuah transien adalah sejenis **fungsi aperiodik yang tak berhingga**. Dalam hal ini periodisitas gagal tercapai karena satu atau lebih parameter penting sinyal berubah. Misalnya rerata sinyal berubah.

$$y(t) = at + b \cos(\omega t)$$

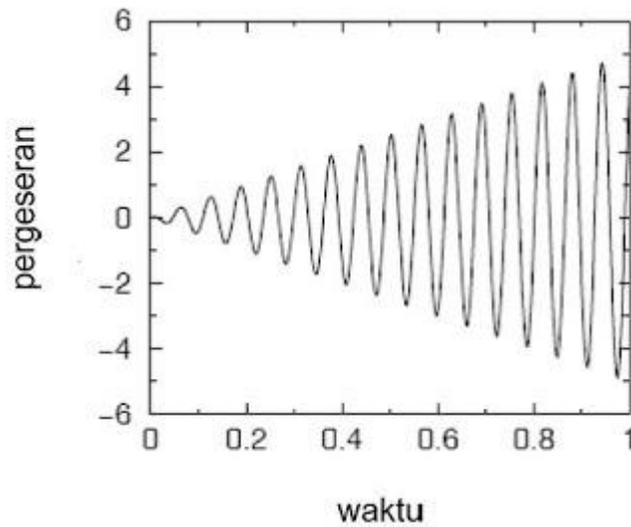
Pada Gambar 2.8 diperlihatkan satu contoh sinyal dengan rerata yang ubah waktu.



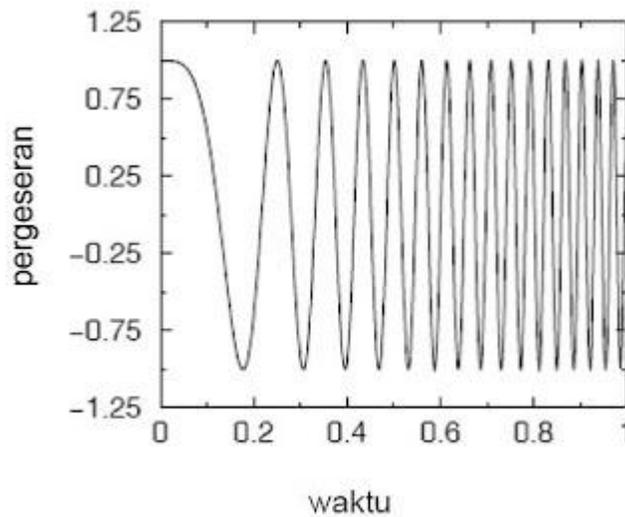
Gambar 1.8 Sebuah Sinyal dengan Rerata Ubah Waktu

Sinyal seperti ini dapat terjadi pada sebuah mesin yang hampir (mendekati) rusak, sehingga karakter yang demikian sangat penting dalam usaha *monitoring* kondisi mesin. Jenis sinyal lain yang juga penting adalah sinyal dengan amplitud yang ubah waktu yang dapat dinyatakan dengan persamaan

$$y(t) = at \cos(\omega t)$$



Gambar 1.9 Sebuah Sinyal dengan Amplitude Ubah Waktu

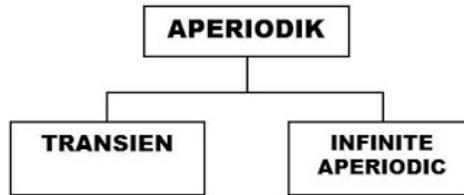


Gambar 1.10 Sebuah Sinyal dengan Frekuensi Ubah Waktu

Sinyal juga dapat bervariasi dalam frekuensinya, misalnya pada persamaan berikut:

$$y(t) = a \cos (\omega t^2)$$

Dengan demikian sinyal aperiodik dapat dibagi menjadi sinyal yang transien dan sinyal aperiodik tak berhingga (*infinite aperiodic*).



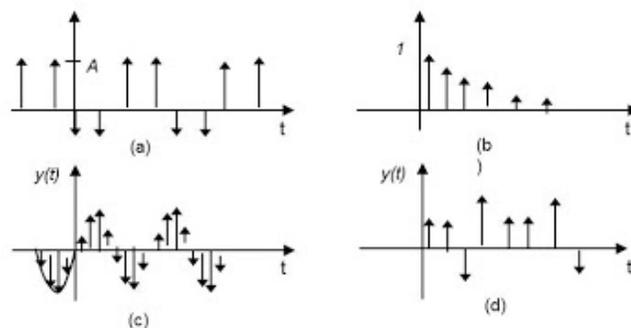
Gambar 1.11. Klasifikasi Sinyal Aperiodik; Transien dan *Infinite Aperiodic*

1.1.3 Sinyal Diskrit

Pada teori system diskrit, lebih ditekankan pada pemrosesan sinyal yang berderetan. Pada sejumlah nilai x , dimana nilai yang $ke-x$ pada deret $x(n)$ akan dituliskan secara formal sebagai:

$$x = \{x(n)\}; -\infty < n < \infty$$

Dalam hal ini $x(n)$ menyatakan nilai yang ke- n dari suatu deret, persamaan di atas biasanya tidak disarankan untuk dipakai dan selanjutnya sinyal diskrit diberikan seperti Gambar (4) Meskipun absis digambar sebagai garis yang kontinyu, sangat penting untuk menyatakan bahwa $x(n)$ hanya merupakan nilai dari n . Fungsi $x(n)$ tidak bernilai nol untuk n yang bukan integer; $x(n)$ secara sederhana bukan merupakan bilangan selain integer dari n .



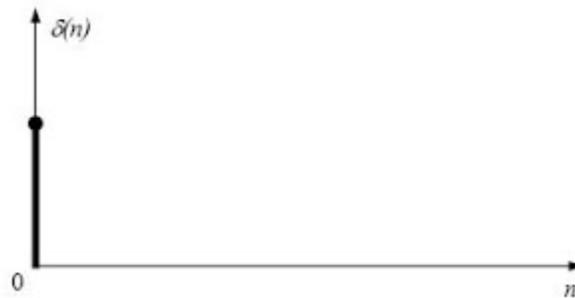
Gambar 1.12 Contoh *discrete time signal* (a) Digital Periodik (b) n positif (c) *discrete time* (d) digital non periodik.

Sekuen Impuls

Deret unit sample (unit-sampel sequence), $\delta(n)$, dinyatakan sebagai deret dengan nilai:

$$\delta(n) = \begin{cases} 0, & n \neq 0 \\ 1, & n = 0 \end{cases}$$

Deret unit sample mempunyai aturan yang sama untuk sinyal diskrit dan system dengan fungsi impuls pada sinyal kontinyu dan system. Deret unit sample biasanya disebut dengan impuls diskrit (discrete-time impuls), atau disingkat impuls (impulse).



Gambar 1.13. Sinyal Impuls

Sekuen Step

Deret unit step (unit-step sequence), $u(n)$, mempunyai persamaan sebagai berikut:

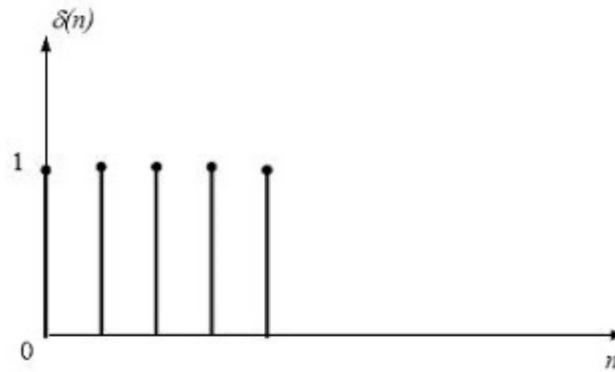
$$u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

Unit step dihubungkan dengan unit sample sebagai:

$$u(n) = \sum_{k=-\infty}^n \delta(k)$$

Unit sample juga dapat dihubungkan dengan unit step sebagai:

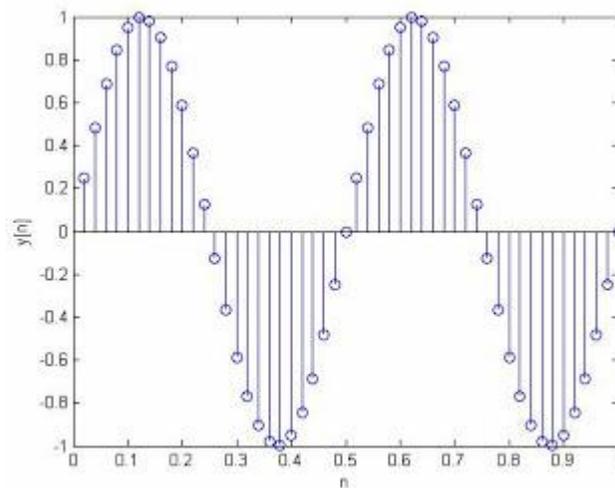
$$\delta(n) = u(n) - u(n-1)$$



Gambar 1.14 Sekuen Step

Sinus Diskrit

Deret eksponensial real adalah deret yang nilainya berbentuk an , dimana a adalah nilai real. Deret sinusoidal mempunyai nilai berbentuk $A\sin(\omega_0 n + \phi)$.



Gambar 1.15 Sinyal Sinus Diskrit

Deret $y(n)$ dinyatakan berkala (periodik) dengan nilai periode N apabila $y(n) = y(n+N)$ untuk semua n . Deret sinusoidal mempunyai periode $2\pi/\omega_0$ hanya pada saat nilai real ini berupa bilangan integer. Parameter ω_0 akan dinyatakan sebagai frekuensi dari sinusoidal atau eksponensial kompleks meskipun deret ini periodik atau tidak. Frekuensi ω_0 dapat dipilih dari nilai jangkauan kontinyu. Sehingga jangkauannya adalah $0 < \omega_0 < 2\pi$ (atau $-\pi < \omega_0 < \pi$) karena deret sinusoidal atau eksponensial kompleks didapatkan dari nilai ω_0 yang bervariasi

dalam jangkauan $2\pi k < \omega_0 < 2\pi(k+1)$ identik untuk semua k sehingga didapatkan ω_0 yang bervariasi dalam jangkauan $0 < \omega_0 < 2\pi$.

1.2 Perangkat

1. Satu buah PC lengkap *sound card* dan OS Windows dan telah di-*install* perangkat lunak program aplikasi MATLAB.
2. Satu set *speaker* aktif.

1.3 Langkah-langkah Percobaan

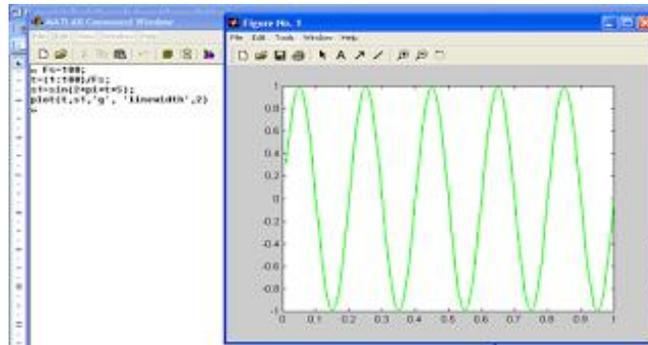
Dalam percobaan ini, mahasiswa harus melakukan langkah-langkah sesuai petunjuk di bawah. Langkah-langkah tersebut harus dibuktikan dalam laporan dengan cara melampirkan *PrtScr* yang telah di *crop* untuk penyajian yang lebih bagus dalam laporan. Jawablah setiap pertanyaan dalam langkah-langkah percobaan !.

1.3.1 Pembangkitan Sinyal Waktu Kontinu Sinusoida

1. Di sini kita mencoba membangkitkan sinyal sinusoida untuk itu coba anda buat program seperti berikut:

```
%-----  
% Nama File : Pembangkik_Sinyal_Sinus.m  
% Oleh      :  
%-----  
Fs=100;  
t=(1:100)/Fs;  
s1=sin(2*pi*t*5);  
plot(t,s1,'r', 'linewidth',2) % Tekan Enter
```

maka hasilnya adalah sebagai berikut:



Gambar 1.16. Contoh Sinyal Sinus

Sinyal yang terbangkit adalah sebuah sinus dengan amplitudo $Amp = 1$, frekuensi $f = 5\text{Hz}$ dan fase awal $\theta = 0$. Diharapkan sudah dipahami tiga parameter dasar pada sinyal sinus ini. Untuk lebih memahami coba lanjutkan dengan langkah berikut.

2. Lakukan perubahan pada nilai s1:

$$s1=\sin(2*\pi*t*10);$$

Dan perhatikan apa yang terjadi, kemudian ulangi untuk mengganti angka 10 dengan 15, dan 20. Perhatikan apa yang terjadi!.

3. Coba anda edit kembali program anda sehingga bentuknya persis seperti pada langkah 1, kemudian lanjutkan dengan melakukan perubahan pada nilai amplitudo, sehingga bentuk perintah pada s1 menjadi:

$$s1=2*\sin(2*\pi*t*5);$$

Coba perhatikan apa yang terjadi (perhatikan angka-angka sekala pada koordinat)? Lanjutkan dengan merubah nilai amplitudo menjadi 4, 5, 6,... sampai 20. Apa pengaruh perubahan amplitudo pada bentuk sinyal sinus?

4. Kembalikan program anda sehingga menjadi seperti pada langkah pertama. Sekarang coba anda lakukan sedikit perubahan sehingga perintah pada s1 menjadi:

$$s1=2*\sin(2*\pi*t*5 + \pi/2);$$

Coba anda perhatikan, apa yang terjadi? Apa yang baru saja anda lakukan adalah merubah nilai fase awal sebuah sinyal dalam hal ini nilai $\theta = \pi/2 = 90^\circ$. Sekarang lanjutkan langkah anda dengan merubah nilai fase awal menjadi 45° , 120° , 180° , dan 225° . Amati bentuk sinyal sinus terbangkit, dan catat hasilnya.

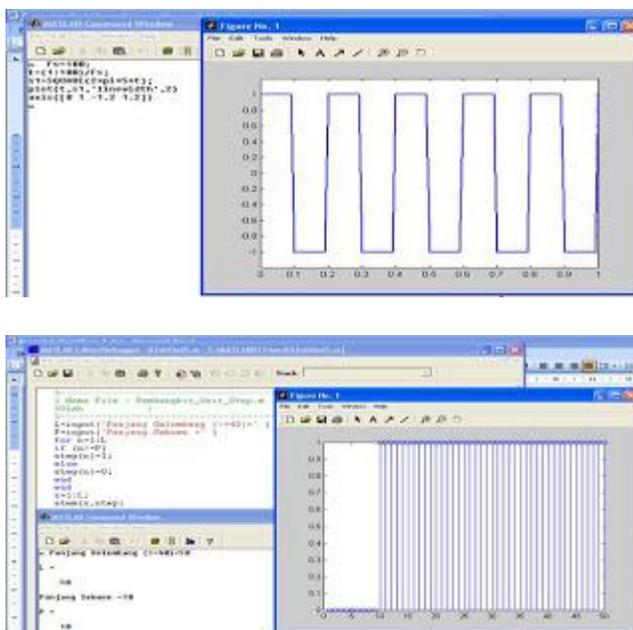
1.3.2. Pembangkitan Sinyal Persegi

Di sini akan dibangkitkan sebuah sinyal persegi dengan karakteristik frekuensi dan amplitudo yang sama dengan sinyal sinus. Untuk melakukannya ikuti langkah berikut ini:

1. Buat sebuah file baru dan beri nama coba_kotak.m kemudian buat program seperti berikut ini.

```
%-----  
% Nama File : Pembangki_Sinyal_Persegi.m  
% Oleh      :  
%-----  
Fs=100;  
t=(1:100)/Fs;  
s1=SQUARE(2*pi*5*t);  
plot(t,s1,'linewidth',2)  
axis([0 1 -1.2 1.2])
```

maka hasilnya adalah sebagai berikut:



Gambar 1.17. Contoh sinyal persegi terbangkit

Dari gambar 2.17 anda dapat melihat sebuah sinyal persegi dengan amplitudo bernilai 1 dan frekuensinya sebesar 5 Hz.

2. Coba anda lakukan satu perubahan dalam hal ini nilai frekuensinya anda rubah menjadi 10 Hz, 15 Hz, dan 20 Hz. Apa yang anda dapatkan?
3. Kembalikan bentuk program menjadi seperti pada langkah pertama. Sekarang coba anda rubah nilai fase awal menjadi 45o, 120o, 180o, dan 225o. Amati dan catat apa yang terjadi dengan sinyal persegi terbangkit.

1.3.3 Pembangkitan Sinyal Waktu Diskrit, Sekuen Konstan

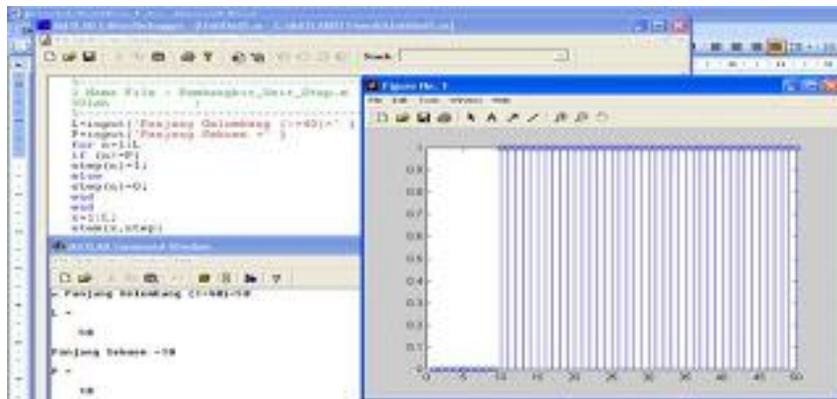
Di sini akan kita lakukan pembangkitan sinyal waktu diskrit. Sebagai langkah awal kita mulai dengan membangkitkan sebuah sekuen unit step. Sesuai dengan namanya, unit step berarti nilainya adalah satu satuan. Untuk itu anda ikuti langkah berikut ini.

1. Berikut ini akan dibuat program baru untuk membentuk gelombang *unit step*. Silahkan anda ketikkan perintah seperti program berikut ini dengan nama: `Pembangkit_Unit_Step.m`

```

%-----
% Nama File : Pembangkit_Unit_Step.m
%Oleh      :
%-----
L=input('Panjang Gelombang (>=40)=')
P=input('Panjang Sekuen =')
for n=1:L
if (n>=P)
step(n)=1;
else
step(n)=0;
end
end
end
x=1:L;
stem(x,step)

```



Gambar 1.18 Contoh Sekuen Step Terbangkit

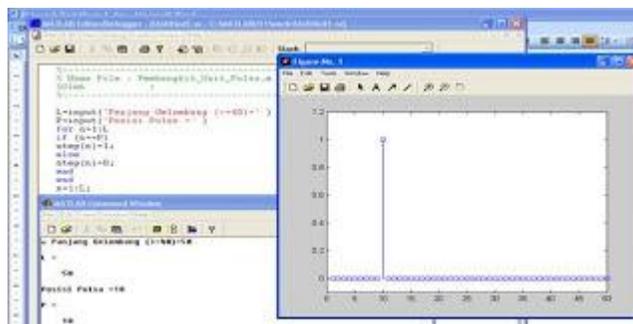
2. Anda ulangi langkah pertama dengan cara menjalankan program anda dan masukan nilai untuk panjang gelombang dan panjang sekuen yang berbeda-beda. Catat apa yang terjadi?

1.3.4 Pembangkitan Sinyal Waktu Diskrit, Sekuen Pulsa

Di sini akan kita bangkitkan sebuah sinyal waktu diskrit berbentuk sekuen pulsa, untuk itu ikuti langkah berikut ini:

1. Buat program baru dengan perintah berikut ini.

```
%-----  
% Nama File : Pembangkit_Unit_Pulsa.m  
%Oleh      :  
%-----  
L=input('Panjang Gelombang (>=40)= ' )  
P=input('Posisi Pulsa = ' )  
for n=1:L  
if (n==P)  
step(n)=1;  
else  
step(n)=0;  
end  
end  
x=1:L;  
stem(x,step)  
axis([0 L -.1 1.2])
```



Gambar 1.19 Contoh sekuen pulsa terbangkit

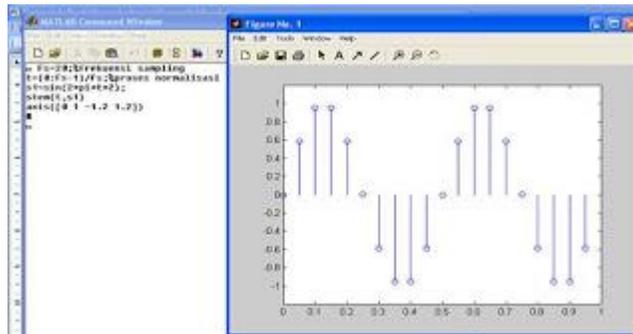
2. Jalankan program diatas berulang-ulang dengan catatan nilai L dan P dirubah-subah sesuai kehendak anda, perhatikan apa yang terjadi? Catat apa yang anda lihat.

1.3.5 Pembentukan Sinyal Sinus waktu Diskrit

Pada bagian ini kita akan dicoba untuk membuat sebuah sinyal sinus diskrit. Secara umum sifat dasarnya memiliki kemiripan dengan sinus waktu kontinyu. Untuk itu ikuti langkah berikut

1. Buat program baru dengan perintah seperti berikut.

```
%-----
% Nama File : Sinyal_Diskrit1.m
%Oleh      :
%-----
Fs=20;%frekuensi sampling
t=(0:Fs-1)/Fs;%proses normalisasi
s1=sin(2*pi*t*2);
stem(t,s1)
axis([0 1 -1.2 1.2])
```



Gambar 1.20 Contoh Sinus Diskrit

2. Lakukan perubahan pada nilai Fs, sehingga bernilai 30, 40, 50, 60, 70, dan 80. Catat apa yang terjadi ?
3. Lakukan perubahan pada nilai Fs, sehingga bernilai 18, 15, 12, 10, dan 8. Catat apa yang terjadi?

1.3.6 Pembangkitan Sirine

Di sini akan kita bangkitkan sebuah sinyal yang dapat memberikan keluaran berupa suara sirine. Karena keluarannya berupa suara maka selayaknya perlu dipasang speaker aktif yang kompatibel dengan komputer anda. Ikuti langkah-langkah berikut ini:

1. Buat program baru dengan perintah seperti berikut dan jalankan!. Apa yang diperoleh.

```
%-----  
% Nama File : Sinyal_Sirine.m  
%Oleh      :  
%-----  
Fs=8000;  
  
dt=1/Fs;  
dur=2.8;  
t=0:dt:dur;  
psi=2*pi*(100 + 200*t + 500*t.*t);  
xx= 7.7*sin(psi);  
sound(xx,fs);  
psi=sirine;  
save sirine;
```

2. Tulis sirine pada command window MATLAB. Apa yang terjadi?
3. Lakukan perubahan pada nilai Fs, sehingga bernilai 4000. Catat apa yang terjadi ?
4. Lakukan perubahan pada nilai Fs, sehingga bernilai 16000. Catat apa yang terjadi?

1.3.7 Pembangkitan Nada DTMF

Di sini akan dibangkitkan sebuah sinyal yang dapat memberikan keluaran berupa nada DTMF. Ikuti langkah-langkah berikut ini:

1. Buat program baru dengan perintah seperti berikut dan jalankan!. Apa yang diperoleh.

```
%-----  
% Nama File : Sinyal_Nada_DTMF.m  
%Oleh      :  
%-----  
% Freq : 1209 | 1336 | 1477  
%-----  
% 697 : 1 | 2 | 3
```

```

%-----
% 770 : 4 | 5 | 6
%-----
% 852 : 7 | 8 | 9
%-----
% 941 : * | 0 | #
%-----
Fs=8000;
t=0:0.001:1.5;
y1=sin(2*pi*852*t)+sin(2*pi*1209*t);
y2=sin(2*pi*770*t)+sin(2*pi*1477*t);
y3=sin(2*pi*770*t)+sin(2*pi*1477*t);
y4=sin(2*pi*697*t)+sin(2*pi*1209*t);
y5=sin(2*pi*697*t)+sin(2*pi*1336*t);
y6=sin(2*pi*697*t)+sin(2*pi*1209*t);
y7=sin(2*pi*941*t)+sin(2*pi*1477*t);
wavplay(y1,Fs)
wavplay(y2,Fs)
wavplay(y3,Fs)
wavplay(y4,Fs)
wavplay(y5,Fs)
wavplay(y6,Fs)
wavplay(y7,Fs)

```

2. Lakukan perubahan pada nilai Fs, sehingga bernilai 4000. Catat apa yang terjadi ?
3. Lakukan perubahan pada nilai Fs, sehingga bernilai 16000. Catat apa yang terjadi?

1.3.8. Memanggil Contoh Suara yang Terdapat pada MATLAB

```

%-----
% Nama File : Membaca_dan_Memainkan _File_wav.m
% Oleh      :
%-----
clear all;
load gong %memanggil audio data (MAT files).
sound(y,Fs)

```

Jalankan program anda, dan anda akan mendengarkan orang tertawa. Coba anda gantikan kata gong dengan *chirp*, *gong*, *handel*, *laughter*, *splat*, dan *train*

1.3.9. Pembangkitan Sinyal Dengan memanfaatkan file *.wav

Kita mulai bermain dengan file *.wav. Dalam hal ini dilakukan pemanggilan sinyal audio yang ada dalam hardisk. Langkah yang harus dilakukan adalah seperti berikut:

1. Simpanlah sebuah lagu (misalnya: How_can_I_tell_her.wav) dalam format "wav" pada folder: work dari MATLAB. Jika formatnya bukan "wav", maka perlu dikonversi terlebih dahulu ke dalam format "wav" menggunakan program yang lain. Lalu buat file dengan nama: Membaca_dan_Memainkan_File_wav.m seperti berikut:

```
%-----  
% Nama File : Membaca_dan_Memainkan_File_wav.m  
% Oleh      :  
%-----  
Fs=16000;  
  
y1=wavread(' How_can_I_tell_her.wav');  
wavplay(y1,Fs,'async') % Memainkan audio sinyal asli
```

2. Cobalah untuk menampilkan file audio yang telah anda panggil dalam bentuk grafik sebagai fungsi waktu. Perhatikan bentuk tampilan yang anda lihat. Apa yang anda catat dari hasil yang telah anda dapatkan tersebut?

1.4. Data dan Analisis

Anda telah melakukan berbagai langkah untuk percobaan pembangkitan sinyal baik diskrit maupun kontinyu dan anda juga sudah mempelajari bagaimana membaca *audio file .wav*. dan mengaktifkan speaker melalui perintah dalam MATLAB. Yang harus anda lakukan adalah: mengujicoba setiap program di atas, memodifikasi sebagian untuk mempengaruhi pengaruhnya, mencatat dan menjawab setiap pertanyaan yang ada pada setiap langkah percobaan diatas.

1.5 Tugas

Jawablah pertanyaan-pertanyaan berikut ini:

1. Coba anda buat sebuah sinyal sinus dan anda simpan menjadi file *.wav.
2. Buat sebuah program yang terdiri dari beberapa gelombang sinus dengan frekuensi yang berbeda, kemudian jumlahkan, hasilnya simpan dengan menggunakan file *.wav. Pindahkan dari work Matlab, ke - file lain. Kemudian click dua kali. Apa yang terjadi?.
3. Buat sebuah program yang dapat membangkitkan sebuah bunyi yang mirip *alarm*.
4. Buatlah sebuah program sendiri yang mirip dengan program yang terdapat pada program MATLAB.
5. Buatlah program yang dapat membangkitkan gelombang diskrit eksponensial menaik dan teredam.
6. Buatlah program yang dapat membangkitkan gelombang kontinyu dan diskrit parabolik.

MODUL II

OPERASI DASAR PADA SINYAL

Tujuan Praktikum

1. Mahasiswa dapat memperlihatkan proses-proses aritmatika sinyal dan menerapkan sebagai proses dasar dari pengolah sinyal audio.
2. Mahasiswa dapat menjumlahkan dua sinyal audio dalam format "wav" dan menampilkan grafik dan menguji dalam bunyi.

2.1. Dasar Teori

Bisa jadi sebuah sinyal belum sesuai dengan yang dibutuhkan, mungkin amplitudonya tidak sesuai, atau mungkin perlu adanya penjumlahan dua sinyal, atau operasi yang lainnya. Berikut ini akan dipelajari memanipulasi sinyal.

2.1.1 Operasi Aritmatika Sinyal

Pada analisa sistem pemrosesan sinyal diskrit, deretnya dapat dimanipulasi dalam beberapa cara. Perkalian (*product*) dan penambahan (*sum*) dari dua deret x dan y dinyatakan sebagai *sample* perkalian dan pembagian dimana:

$$x \cdot y = \{x(n) y(n)\} \quad (\text{product})$$

$$x + y = \{x(n) + y(n)\} \quad (\text{sum})$$

Perkalian dari deret x dengan sebuah nilai α dinyatakan sebagai

$$\alpha \cdot x = x(n - n_0)$$

dimana n_0 adalah bilangan integer.

Dalam realita kehidupan sehari-hari, khususnya dalam dunia *electronic communication engineering*, kita mengenal proses aritmatika pada sinyal yang meliputi:

- penguatan sinyal
- pelemahan sinyal
- penjumlahan dua buah sinyal
- perkalian dua buah sinyal

2.1.2 Penguatan Sinyal

Peristiwa penguatan sinyal seringkali kita jumpai pada perangkat audio seperti radio, tape, dsb. Fenomena ini dapat juga direpresentasikan secara sederhana sebagai sebuah operasi matematika sebagai berikut:

$$y(t) = amp \ x(t)$$

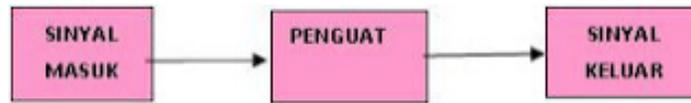
keterangan:

$$y(t) = \text{sinyal output}$$

$$amp = \text{konstanta penguatan sinyal}$$

$$x(t) = \text{sinyal input}$$

Bentuk diagram blok dari sebuah operasi penguatan sinyal dapat diberikan pada gambar 2.1 berikut ini.

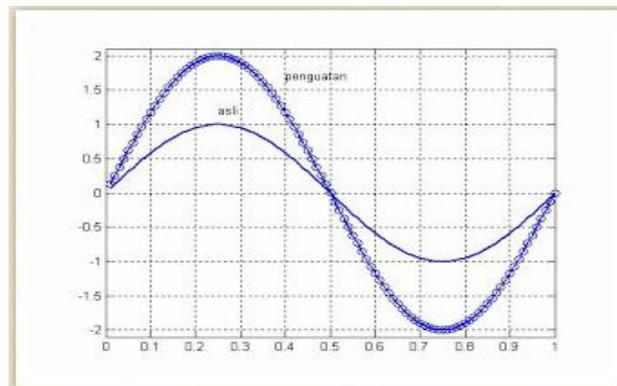


Gambar 2.1. Diagram Blok Penguatan Suatu Sinyal

Besarnya nilai konstanta sinyal $amp > 1$, dan penguatan sinyal seringkali dinyatakan dalam besaran deci Bell (dB), yang didefinisikan sebagai:

$$amp_{dB} = 10 \log(\text{output}/\text{input})$$

Dalam domain waktu, bentuk sinyal asli dan setelah mengalami penguatan adalah seperti gambar 2.2.



Gambar 2.2. Penguatan Sinyal

2.1.3 Pelemahan Sinyal

Apabila sebuah sinyal dilewatkan suatu medium seringkali mengalami berbagai perlakuan dari medium (kanal) yang dilaluinya. Ada satu mekanisme dimana sinyal yang melewati suatu medium mengalami pelemahan energi yang selanjutnya dikenal sebagai atenuasi (pelemahan atau redaman) sinyal.

Bentuk diagram blok dari sebuah operasi pelemahan sinyal dapat diberikan pada gambar 2.3 berikut ini.

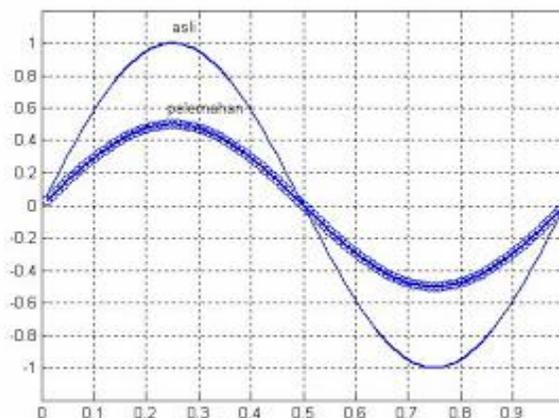


Gambar 2.3 Operasi Pelemahan Suatu Sinyal

Dalam bentuk operasi matematik sebagai pendekatannya, peristiwa ini dapat diberikan sebagai berikut:

$$y(t) = att x(t)$$

Dalam hal ini nilai $att < 1$, yang merupakan konstanta pelemahan yang terjadi. Kejadian ini sering muncul pada sistem transmisi, dan munculnya konstanta pelemahan ini dihasilkan oleh berbagai proses yang cukup kompleks dalam suatu media transmisi.



Gambar 2.4. Pelemahan Sinyal

Dari gambar tersebut dapat dilihat bahwa proses penguatan dan pelemahan sinyal merupakan dua hal yang hampir sama. Dalam penguatan sinyal amplitudo sinyal output lebih tinggi dibanding sinyal input, sementara pada pelemahan sinyal amplitudo sinyal output lebih rendah dibanding sinyal input. Tetapi pada kedua proses operasi ini bentuk dasar sinyal tidak mengalami perubahan.

2.1.4 Penguatan Terkontrol pada Sinyal

Jika terpat beberapa sinyal yang akan dibandingkan untuk keperluan identifikasi, haruslah mempunyai amplitudo yang sama. Salah satu cara untuk menyamakan amplitudo adalah dengan menggunakan penguatan terkontrol (AGC: *Automatic Gain Control*), dengan ide dasar sebagai berikut:

$$AGC = \frac{amp}{Amplitudo\ Maksimum}$$

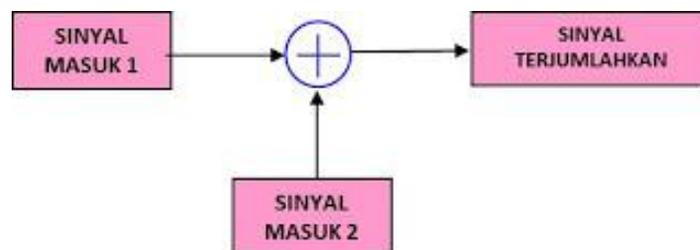
Sehingga sinyal yang baru adalah:

$$x(t)_{baru} = AGC \cdot x(t)$$

Dengan demikian semua sinyal akan mempunyai amplitudo yang sama, sesuai dengan nilai "amp"

2.1.5 Penjumlahan Dua Buah Sinyal

Proses penjumlahan sinyal seringkali terjadi pada peristiwa transmisi sinyal melalui suatu medium. Sinyal yang dikirimkan oleh pemancar setelah melewati medium tertentu misalnya udara akan mendapat pengaruh kanal, dapat menaikkan level tegangan atau menurunkan level tegangannya tergantung komponen yang dijumlahkan. Sehingga pada bagian penerima akan mendapatkan sinyal sebagai hasil jumlahan sinyal asli dari pemancar dengan sinyal yang terdapat pada kanal tersebut.

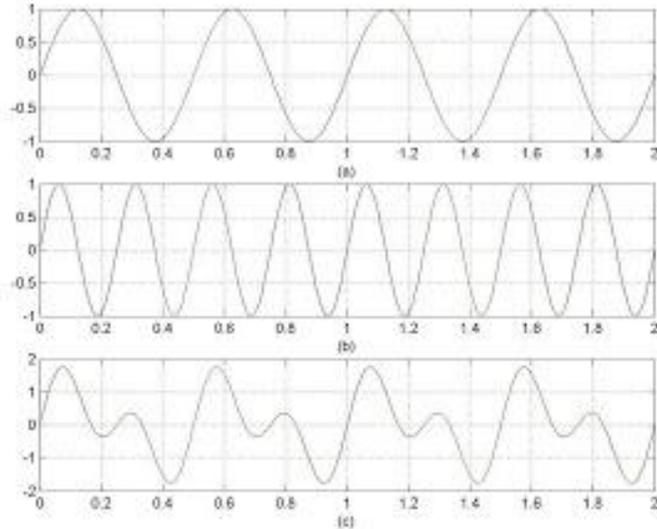


Gambar 2.5. Diagram Blok Operasi Penjumlahan Dua Sinyal.

Secara matematis dapat diberikan sebagai berikut:

$$y(t) = x_1(t) + x_2(t)$$

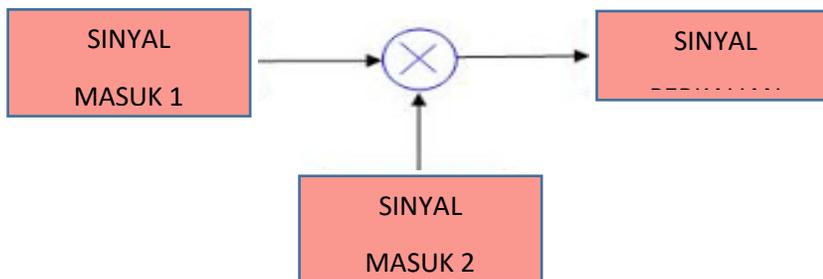
Dalam hal ini, setiap komponen sinyal pertama dijumlahkan dengan komponen sinyal kedua.



Gambar 2.6. Contoh Penjumlahan pada Sinyal Sinus
(a) Sinyal Masuk 1
(b) Sinyal Masuk 2
(c) Sinyal Hasil Penjumlahan

2.1.6 Perkalian Dua Buah Sinyal

Perkalian merupakan bentuk operasi yang sering anda jumpai dalam kondisi real. Pada rangkaian *mixer*, rangkaian *product modulator* dan *frequency multiplier*, operasi perkalian merupakan bentuk standar yang seringkali dijumpai. Bentuk diagram blok operasi perkalian dua buah sinyal dapat diberikan seperti pada Gambar 2.7 berikut.



Gambar 2.7. Diagram blok operasi perkalian dua sinyal.

2.2. Peralatan

Satu buah PC multimedia OS Windows yang telah diinstall dengan perangkat lunak Matlab.

2.3. Langkah Percobaan

Dalam percobaan ini, mahasiswa harus melakukan langkah-langkah sesuai petunjuk di bawah. Langkah-langkah tersebut harus dibuktikan dalam laporan dengan cara melampirkan *PrtScr* yang telah di *crop* untuk penyajian yang lebih bagus dalam laporan. Jawab setiap pertanyaan dalam langkah-langkah percobaan!.

2.3. 1. Penguatan Sinyal

1. Bangkitkan gelombang pertama dengan langkah berikut:

```
%-----  
% Nama File : Penguatan_Sinyal.m  
% Oleh      :  
%-----  
T=100;  
t=0:1/T:2;  
f1=1;  
y1=sin(2*pi*t);  
subplot(2,1,1)  
plot(t,y1)
```

2. Lanjutkan dengan langkah berikut ini

```
a=input('nilai pengali yang anda gunakan (> 0): ');  
y1_kuat=a*sin(2*pi*t);  
subplot(2,1,2)  
plot(t,y1_kuat)
```

Jangan lupa anda masukkan sebuah nilai untuk 'a', misalnya 1.5 atau yang lain. Apa yang anda dapatkan? Apa perbedaan dari kedua gambar sinyal tersebut? Nilai penguatan sinyal

juga seringkali dituliskan dalam dB (dB), untuk penguatan 1.5 kali berapa nilainya dalam dB?

3. Ulangi langkah 1 dan 2, tetapi dengan nilai a berbeda misalnya 1.7, 2.5, 3.0 atau yang lain. Dan jangan lupa anda simpan gambarnya dan buatlah analisa dari apa yang anda amati dari gambar tersebut? Jangan lupa dalam setiap penggambaran anda cantumkan nilai dB setiap percobaan.

2.3.2 Pelemahan Sinyal

Seperti yang kita ketahui bahwa pelemahan merupakan penguatan negatif, atau dalam hal ini konstanta penguatan bernilai <1 . Berdasar pemahaman ini coba anda susun sebuah program pelemahan sinyal dengan memanfaatkan contoh program yang sudah anda buat pada langkah 4.1.

2.3.3 Mengubah Amplitudo Maksimum Menjadi Satu Satuan

Mintalah kepada instruktur sebuah *soft copy* sinyal dalam format *wav*, dan beri nama JantungNormal.wav. Kopikan pada *work* dari Matlab komputer yang digunakan. Kemudian tulis program demikian:

```
%-----  
% Nama File : Amplitudo_Sinyal_Satuan.m  
% Oleh      :  
%-----  
B=wavread('JantungNormal.wav');  
AGC=max(B);  
A=B/AGC;  
subplot(2,1,1);  
plot(B);set(gca,'color',[1 .2 .2]);%grid on;  
subplot(2,1,2);  
plot(A);set(gca,'color',[.2 1 .2]);%grid on;
```

2.3.4 Penjumlahan Dua Sinyal

Dua buah sinyal dapat dijumlahkan jika mempunyai dimensi matrik yang sama. Untuk mengetahui dimensi matrik, gunakan perintah:

```
length (nama_sinyal) % tekan Enter
```

dalam *command window*, maka akan ditampilkan panjang matrik. Dengan mengacu pada penjelasan yang ada di dasar teori bab 2, operasi penjumlahan dua buah sinyal dapat dilakukan dengan mengikuti langkah-langkah berikut ini.

1. Buat sebuah program baru dengan perintah:

```
%-----  
% Nama File : Penjumlahan_Dua_Sinyal.m  
% Oleh      :  
%-----  
T=100;  
t=0:1/T:2;  
f1=1;  
y1=sin(2*pi*t);  
subplot(3,1,1)  
plot(t,y1)
```

2. Bangkitkan gelombang kedua dengan langkah tambahan berikut ini:

```
f2=2;  
pha2=pi/2;  
y2=sin(2*pi*t+pi);  
subplot(3,1,2)  
plot(t,y2)
```

3. Lakukan proses penjumlahan pada kedua sinyal y_1 dan y_2 di atas. Selengkapnya bentuk programnya adalah seperti berikut:

```
T=100;  
t=0:1/T:2;  
f1=1;  
f2=2;  
pha2=pi/2;  
y1=sin(f1*pi*t);
```

```

subplot(3,1,1)
plot(t,y1)
y2=sin(f2*pi*t+ pha2);
subplot(3,1,2)
plot(t,y2)
y3=y1+y2;
subplot(3,1,3)
plot(t,y3)

```

5. Coba anda rubah nilai f2 menjadi 3, 4, 5,.....10. Perhatikan apa yang terjadi dan catat hasilnya.
6. Lakukan perubahan pada pha2 sehingga nilainya menjadi 0.1π , 0.25π , 0.5π , dan 1.5π . Apa yang anda dapatkan dari langkah ini?

2.3.5 Perkalian Dua Sinyal

Seperti pada penjumlahan dua sinyal, perkalian dua sinyal juga harus mempunyai panjang matrik yang sama. Pada program di bawah yang menentukan panjang gelombang adalah T dan t. Dimulai dengan membangkitkan dua buah sinyal sinus, kemudian langkah yang harus dilakukan adalah seperti berikut:

1. Kita mulai membangkitkan sebuah gelombang sinus. Pertama dibangkitkan gelombang dengan langkah berikut:

```

%-----
% Nama File : Perkalian_Dua_Sinyal.m
% Oleh      :
%-----
T=100;
t=0:1/T:2;
f1=1;
y1=sin(2*pi*t);
subplot(3,1,1)
plot(t,y1)

```

2. Bangkitkan gelombang kedua dengan langkah tambahan berikut ini:

```
f2=2;  
pha2=pi/2;  
y2=sin(2*pi*t+pi);  
subplot(3,1,2)  
plot(t,y2)
```

3. Lakukan proses perkalian pada kedua sinyal y_1 dan y_2 diatas. Selengkapnya bentuk programnya adalah seperti berikut:

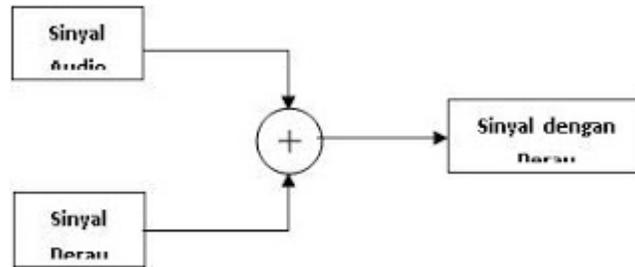
```
T=100;  
t=0:1/T:2;  
f1=1;  
f2=2;  
pha2=pi/2;  
y1=sin(f1*pi*t);  
subplot(3,1,1)  
plot(t,y1)  
y2=sin(f2*pi*t+ pha2);  
subplot(3,1,2)  
plot(t,y2)  
y3=y1.*y2;  
subplot(3,1,3)  
plot(t,y3)
```

4. Coba anda rubah nilai f_2 menjadi 3, 4, 5,.....10. Apa yang terjadi dan catat hasilnya.

4. Lakukan perubahan pada pha_2 sehingga nilainya menjadi 0.1π , 0.25π , dan 1.5π . Apa yang anda dapatkan dari langkah ini?

2.3.6 Penambahan Noise Gaussian pada Sinyal Audio

Mungkin anda sudah bosan melakukan aktifitas dengan sesuatu yang serba ideal teoritis dan serba serius. Sekaranglah saatnya anda belajar sambil bermain. Tentu saja, dalam hal ini PC tempat anda bekerja harus dilengkapi dengan perangkat multimedia, minimal sound card lengkap dengan *speaker active*.



Gambar 2.8. Operasi penjumlahan sinyal audio *.wav dengan noise

Baiklah, kita mulai dengan memanggil sebuah file audio3.wav. Kalau dalam folder dimana anda sekarang bekerja tidak ada file ini, cobalah tanyakan ke dosen yang bersangkutan, atau kalau anda ingin dikatakan sebagai orang yang kreatif, coba anda carai file *.wav apa saja yang ada di PC anda, copykan ke folder dimana Matlab anda bekerja.

1. Untuk contoh kasus ini ikuti langkah pertama dengan membuat file Noise_Gaussian.m, seperti berikut.

```
%-----  
% Nama File : Noise_Gaussian.m  
% Oleh      :  
%-----  
y1=wavread('audio3.wav');  
Fs=8192;  
Fs1 = Fs;  
wavplay(y1,Fs1,'sync') % Sinyal asli dimainkan
```

2. Tambahkan perintah berikut ini setelah langkah satu di atas.

```
N=length(y1); %menghitung dimensi file wav  
var = 0.1;
```

```

noise_1=var*randn(N,1); %membangkitkan noise Gaussian
y_1n=y1 + noise_1;      %menambahkan noise ke file
wavplay(y_1n,Fs1,'sync') % Sinyal bernoise dimainkan

```

3. Apakah anda melihat ada sesuatu yang baru dengan langkah anda?
Coba anda lakukan sekali lagi langkah 2 dengan nilai var 0.2, 0.3, 0.5, dst. Coba amati apa yang terjadi?
4. Cobalah untuk menampilkan file audio yang telah anda panggil dalam bentuk grafik sebagai fungsi waktu, baik untuk sinyal asli atau setelah penambahan noise.

2.3.7 Proses Penguatan pada Sinyal Audio

Sekarang kita lanjutkan permainan kita dengan file *.wav. Dalam hal ini kita lakukan penguatan atau pelemahan sinyal audio yang telah kita panggil. Langkah yang kita lakukan adalah seperti berikut.

1. Buatlah program untuk membangkitkan sebuah sinyal audio dan simpan dalam format "wav" pada WORK dari MATLAB. Lanjutkan dengan membuat sebuah program dengan nama: Penguatan.m seperti berikut

```

%-----
% Nama File : Penguatan.m
% Oleh      :
%-----
y1=wavread('audio3.wav'); % Membaca sinyal audio
Fs=8192;
wavplay(y1,Fs,'async')   % Memainkan audio sinyal asli

```

2. Lakukan penambahan perintah seperti dibawah ini

```

amp =1.5;
y2=amp*y1;
wavplay(y1,Fs,'async') % Memainkan audio

```

3. Apakah anda mengamati sesuatu yang baru pada sinyal audio anda? Kalau belum juga memahami coba rubah nilai amp = 0.1, 0.2, 0.5, dst sampai nilainya 2.0.
4. Cobalah untuk menampilkan file audio yang telah anda panggil dalam bentuk grafik sebagai fungsi waktu, baik untuk sinyal asli atau setelah penguatan dan pelemahan.

2.3.8 Memanggil Sinyal Audio

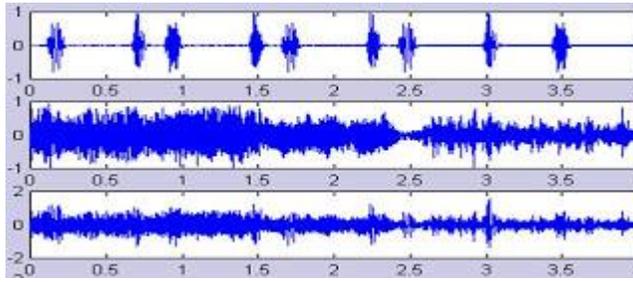
Pertama harus disiapkan sinyal audio yang akan dipanggil dalam format “wav”. Misalnya beri nama sinyal tersebut dengan nama “AsthmaWheeze.wav” dan “JantungNormal.wav”, kemudian simpan dalam work dari Matlab. Kemudian buat program sebagai berikut:

```

%-----
% Nama File : Memanggil_Sinyal_Audio.m
% Oleh      :
%-----
A=wavread('AsthmaWheeze.wav');
B=wavread('JantungNormal.wav');
t=length(A);
A=A(1:1:29400);
B=B(1:1:29400);
Fs=16000;
nfft = 256;
C=A+B';
subplot(3,1,1);
plot(R,'y','Linewidth',2);set(gca,'color',[1 .2 .2]);
%grid on;
subplot(3,1,2);
plot(S,'y','Linewidth',2);set(gca,'color',[.2 1 .2]);
subplot(3,1,3);
plot(T,'y','Linewidth',2);set(gca,'color',[.2 .7 .1]);

```

Hailnya seperti ditunjukkan pada gambar 2.9



Gambar 2.9 Suara dan Jantung Normal, Asthma Wheeze dan Penjumlahannya

2.4. Data dan Analisis

Anda telah melakukan berbagai langkah untuk percobaan operasi dasar sinyal. Yang harus anda lakukan adalah menjawab setiap pertanyaan yang ada pada langkah percobaan. Kemudian lanjutkan dengan mengerjakan tugas berikut ini.

2.5. Tugas

1. Buat program dari penjumlahan tiga buah sinyal sinusoida dengan frekuensi masing-masing: $f_1 = 100$ Hz, $f_2 = 1$ kHz, dan $f_3 = 5$ kHz. Tampilkan masing-masing gelombang dan hasil penjumlahan dalam bentuk grafis dalam satu halaman.
2. Buat program yang digunakan untuk memanggil sebuah gambar dan hasilnya tampilkan dalam sebuah frame secara tepat.

MODUL III

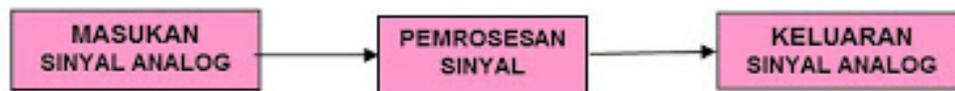
SAMPLING DAN ALIASING

Tujuan Praktikum

1. Mahasiswa memahami dan dapat memilih nilai sampel yang tepat terhadap suatu sinyal.
2. Mahasiswa memahami pengaruhnya sampling, oversampling, dan undersampling pada proses recovery sinyal
3. Mahasiswa dapat melakukan rekontruksi sinyal dari hasil sampling dan menguji hasilnya.
4. Mahasiswa dapat membuat dan menyimpan file dalam format “wav”

3.1 Dasar Teori

Dalam proses pengolahan sinyal analog, sinyal input masuk ke *Analog Signal Processing* (ASP), diberi berbagai perlakuan (misalnya pemfilteran, penguatan, dsb.) dan outputnya berupa sinyal analog.



Gambar 3.1. Sistem Pengolahan Sinyal Analog

Proses pengolahan sinyal secara digital memiliki bentuk sedikit berbeda. Komponen utama sistem ini berupa sebuah processor digital yang mampu bekerja apabila masukannya berupa sinyal digital. Untuk sebuah input berupa sinyal analog perlu proses awal yang bernama digitalisasi melalui perangkat yang bernama *analog-to-digital conversion* (ADC), dimana sinyal analog harus melalui proses *sampling*, *quantizing* dan *coding*. Demikian juga output dari processor digital harus melalui perangkat *digital-to-analog conversion* (DAC) agar outputnya kembali menjadi bentuk analog. Ini bisa kita amati pada perangkat seperti PC, *digital sound system*, dsb. Secara sederhana bentuk diagram bloknnya adalah seperti berikut ini.



Gambar 3.2. Sistem Pengolahan Sinyal Digital

3.1.1. Sinyal Waktu Diskrit

Berdasarkan pada penjelasan di atas kita tahu betapa pentingnya satu proses yang bernama *sampling*. Setelah sinyal waktu kontinyu atau yang juga populer kita kenal sebagai sinyal analog disampel, akan didapatkan bentuk sinyal waktu diskrit. Untuk mendapatkan sinyal waktu diskrit yang mampu mewakili sifat sinyal aslinya, proses *sampling* harus memenuhi syarat Nyquist:

$$f_s > 2 f_i$$

dimana:

f_s = frekuensi sinyal *sampling*

f_i = frekuensi sinyal informasi yang akan disampel

atau dengan kata lain bahwa frekuensi sampel adalah minimal dua kali frekuensi sinyal.

Fenomena *aliasing* akibat proses *sampling* akan muncul pada sinyal hasil *sampling* apabila proses frekuensi sinyal *sampling* tidak memenuhi kriteria di atas. Perhatikan sebuah sinyal sinusoida waktu diskrit yang memiliki bentuk persamaan matematika seperti berikut:

$$x(n) = A \sin(\omega n + \theta) \quad (2)$$

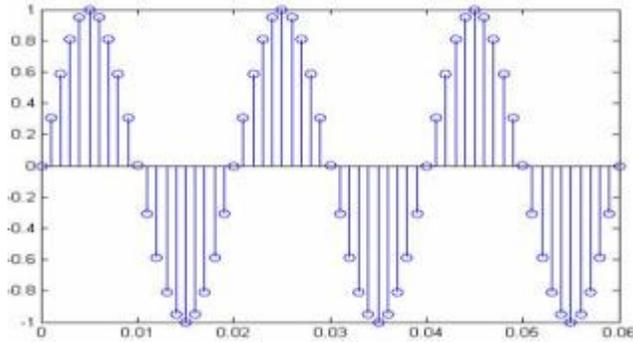
dimana:

A = amplitudo sinyal

ω = frekuensi sudut

θ = fase awal sinyal

Frekuensi dalam sinyal waktu diskrit memiliki satuan radian per indek sample, dan memiliki ekuivalensi dengan $2\pi f$.



Gambar 3.3. Sinyal Sinus Diskrit

Sinyal sinus pada Gambar 3.3 tersusun dari 61 sampel pada sepanjang sinyal, sinyal sinus ini memiliki frekuensi $f = 50$ Hz dan disampel dengan frekuensi sampel $F_s = 1000$ Hz. Sehingga untuk satu siklus sinyal sinus memiliki sample sebanyak $F_s/f = 1000/50 = 20$ sampel. Berbeda dengan sinyal waktu kontinyu (C-T), sifat frekuensi pada sinyal waktu diskrit (D-T) adalah:

1. Sinyal hanya periodik jika f rasional.

Sinyal periodic dengan periode N apabila berlaku untuk semua n bahwa $x(n+N) = x(n)$. Periode fundamental NF adalah nilai N yang terkecil.

Sebagai contoh: agar suatu sinyal periodic maka:

$$\cos(2\pi(N+n) + \theta) = \cos(2\pi n + \theta) = \cos(2\pi + \theta + 2\pi k)$$

$$\Leftrightarrow 2\pi N = 2\pi k \Leftrightarrow f = \frac{k}{N} \Leftrightarrow f \text{ harus rasional}$$

2. Sinyal dengan frekuensi berbeda sejauh $k2\pi$ (dengan k bernilai integer) adalah identik.

Jadi berbeda dengan kasus pada C-T, pada kasus D-T ini sinyal yang memiliki suatu frekuensi unik tidak berarti sinyal nya bersifat unik.

Sebagai contoh:

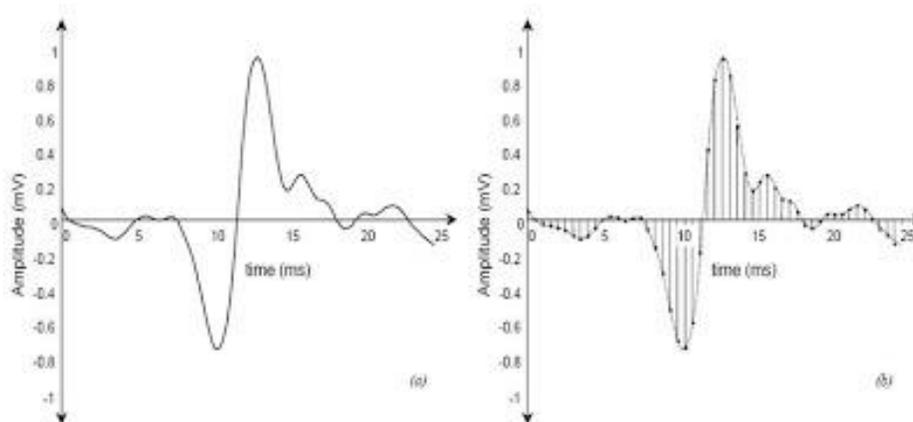
$$\cos[(\omega_0 + 2\pi)n + \theta] = \cos(\omega_0 + 2\pi)$$

karena $\cos(\omega_0 + 2\pi) = \cos(\omega_0)$. Jadi bila $x_k(n) = \cos(\omega_0 n + 2\pi k)$, $k = 0, 1, \dots$ dimana $\omega_k = \omega_0 + 2\pi k$, maka $x_k(n)$ tidak bisa dibedakan satu sama lain. Artinya $x_1(n) = x_2(n) =$

$x_3(n) \dots = x_k(n)$. Sehingga suatu sinyal dengan frekuensi berbeda akan berbeda jika frekuensinya dibatasi pada daerah $-\pi < \omega < \pi$ atau $-1/2 < f < 1/2$. Diluar itu akan terjadi fenomena *aliasing*. Berikut ini akan dijelaskan lebih rinci tentang *sampling* dan *aliasing*.

3.1.2. Digital Sampling

Dalam analisis dan aplikasi sinyal semuanya diterapkan dalam sebuah komputer. Dalam komputer sebuah sinyal dinyatakan sebagai sederetan bilangan. Seperti telah disebutkan bahwa dalam pengolahan sinyal, sinyal yang terdeteksi dikonversi menjadi sederetan bilangan oleh sebuah perangkat elektronik atau komputer disebut: *analog-to-digital conversion*. Sinyal analog adalah sinyal tegangan kontinyu yang dinyatakan secara analog. Amplitudo sinyal ini secara kontinyu bervariasi pada kisarannya. Konversi analog ke digital merupakan proses pembangkitan sederetan bilangan, setiap bilangan menyatakan amplitudo dari sinyal analog pada titik tertentu. Sederetan bilangan yang dihasilkan dinamakan sinyal diskrit atau mungkin sinyal digital, dan diperoleh dari sinyal analog yang disampel. Proses ini ditunjukkan pada Gambar 3.4. Gambar 3.4(a) adalah sinyal analog dan Gambar 3.4(b) adalah sinyal diskrit. Pada sinyal diskrit tampak bahwa sampulnya menyerupai sinyal analog asli yang disampel.

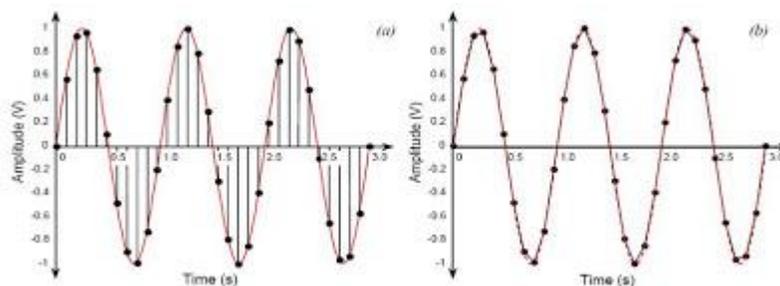


Gambar 3.4 Proses digitalisasi (a) Sinyal Analog, (b) Deretan Digital Hasil Sampling dari Sinyal (a).

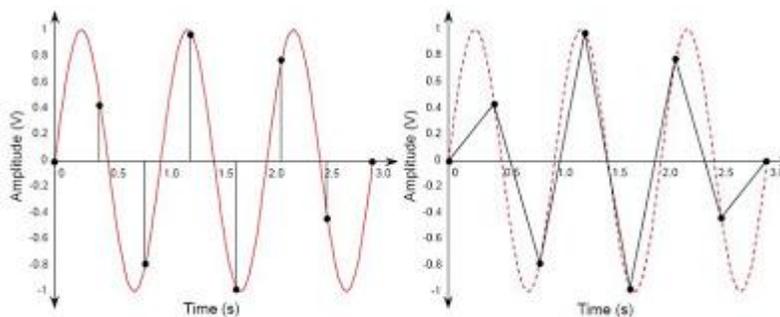
Proses digitalisasi sinyal didefinisikan dengan konsep *sampling*. Pada Gambar 3.4(b), menunjukkan *sampling* sinyal analog pada interval waktu beraturan 0,5 ms. Atau juga dapat dinyatakan bahwa sinyal di-*sampling* pada frekuensi 2000 sample/second. Nilai ini diperoleh

dengan mengambil inverse dari interval waktu, dan secara tipikal dinyatakan dalam Hertz (Hz). Sehingga frekuensi *sampling*-nya menjadi 2 kHz.

Suatu sinyal sinusoida yang dinyatakan dengan amplitudo 1 volt dan frekuensinya 1 Hertz. *Sampling* sinyal ini pada frkuensi 10 Hz menghasilkan sebuah deretan titik data yang melingkupi sinusoida yang asli jika titik-titik tersebut dihubungkan dengan sebuah garis (warna merah pada Gambar 3.5). Ini merupakan dasar yang harus diperhatikan yang menentukan batas frekuensi terendah sinusoida agar sinyal dapat dikembalikan atau direkonstruksi menjadi sinyal yang sesuai dengan aslinya.



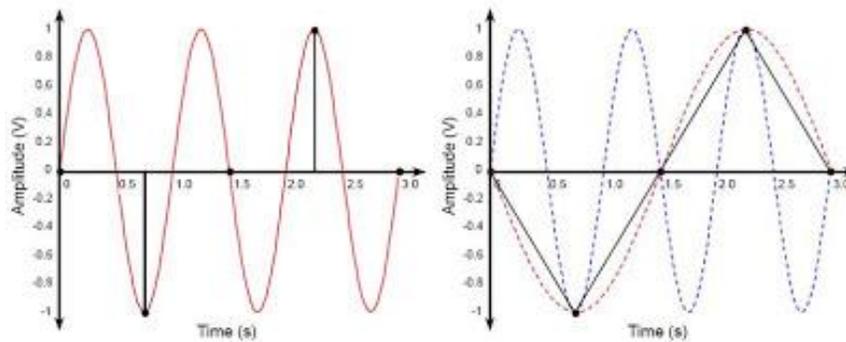
Gambar 3.5 Samping dan Rekonstruksi (a) Sinyal 1 V, 1Hz Sinusoida Di-sampling 10 Hz, (b) Pembentukan Sinusoida Tersampling 10 Hz.



Gambar 3.6 Samping dan Rekonstruksi (a) Sampling Sebuah Sinusoida 1 V, 1 Hz Mendekati 2 Hz (b) Pembentukan Sinusoida pada Sampling 2 Hz.

Perhatikan sinyal yang sama 1 V, 1 Hz sinusoida, tapi sekarang di-*sampling* pada setiap 0,75 detik ($4/3$ Hz). Berbeda dengan dua kejadian sebelumnya, menghasilkan frekuensi sinusoida terendah melalui titik-titik deretan tersebut bukan sinusoida 1 Hz, tapi gelombang sinusoida sekitar $1/3$ Hz. Ini jelas dari contoh tersebut bahwa sinyal original adalah

undersampled dan tidak cukup titik-titik untuk membawa informasi secara tepat. Ingat bahwa kondisi *undersampled* dapat menghasilkan *aliasing*.



Gambar 3.7 Samping dan Rekonstruksi (a) Menyampling Sebuah Sinusoida 1 V, 1 Hz pada 4/3 Hz (b) Pembentukan Sinusoida Tersampling pada 4/3 Hz Menghasilkan Sinyal Merah pada 1/3 Hz. Sinyal 1 Hz yang Asli Disampling Terlalu Rendah.

Dengan contoh di atas, ini menjadi penting untuk realisasi bahwa sinusoida hanya dapat secara tepat tercapai jika di-*sampling* paling kecil dua kali frekuensinya. Hukum ini dikenal sebagai *Nyquist Theorem*.

3.2 Peralatan

- PC yang dilengkapi dengan perangkat multimedia (*sound card, Microphone, active speaker e, atau headset*)
- Sistem Operasi Windows dan Perangkat Lunak MATLAB yang dilengkapi dengan tool box DSP

3.3 Langkah Percobaan

3.3.1 Pengamatan Pengaruh Pemilihan Frekuensi *Sampling* Secara Visual

1. Prosedur yang akan anda lakukan mirip dengan yang ada di modul 2, tetapi di sini lebih ditekankan pada konsep pemahaman fenomena *sampling*. Untuk itu anda mulai dengan membuat program baru dengan perintah seperti berikut.

```
%-----  
%Nama File : Sinyal_Diskrit.m
```

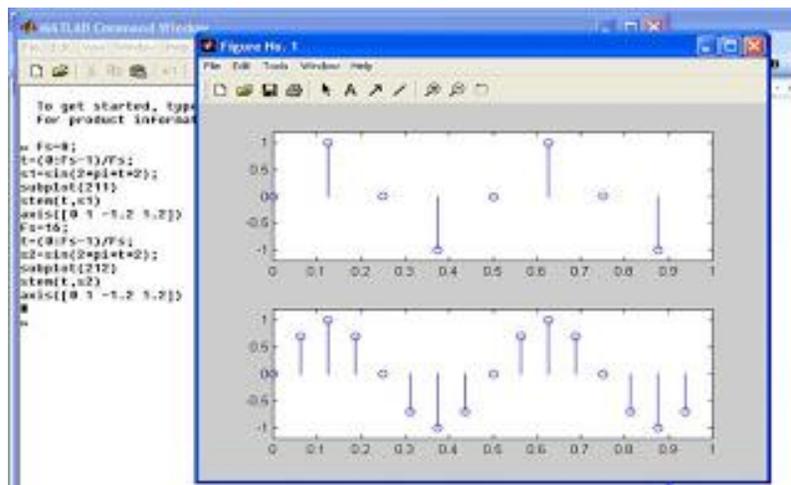
```

%Oleh :
%-----

Fs=8;           %frekuensi sampling
t=(0:Fs-1)/Fs; %proses normalisasi
s1=sin(2*pi*t*2);
subplot(211)
stem(t,s1)
axis([0 1 -1.2 1.2])

Fs=16;         %frekuensi sampling
t=(0:Fs-1)/Fs; %proses normalisasi
s2=sin(2*pi*t*2);
subplot(212)
stem(t,s2)
axis([0 1 -1.2 1.2])

```



Gambar 3.8. Pengaruh Jumlah Sampel Berbeda Terhadap Satu Periode Sinyal Terbangkit

2. Lakukan perubahan pada nilai F_s , pada sinyal s_1 sehingga bernilai 10, 12, 14, 16, 20, dan 30. Catat apa yang terjadi? Apa pengaruh jumlah sampel berbeda untuk satu periode sinyal terbangkit?

Berikut ini diberikan contoh yang harus anda coba dan amati hasilnya.

```

%-----
%Nama File : Sinyal_Diskrit_Sampling_Beda.m

```

```

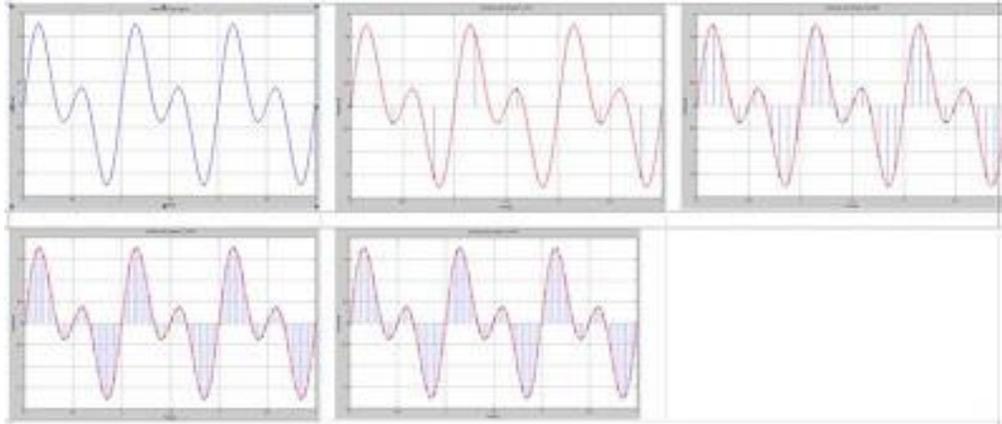
%Oleh      :
%-----
% sinyal kontinyu
t=[0:0.005:3];
x=sin(2*pi*1*t)+sin(2*pi*2*t);
figure(1), clf, plot(t,x), grid on,...
xlabel('Time(s)'); ylabel('Amplitude'),...
title('Continous Time signal')

% sinyal diskrit dengan Ts = 0.4
ts1=[0:0.4:3];
xs1=sin(2*pi*1*ts1)+sin(2*pi*2*ts1);
figure(2),clf,stem(ts1,xs1);
grid on,hold on,plot(t,x,'r:'), hold off,...
xlabel('Time(s)'); ylabel('Amplitude'),...
title('sampling x(t) dengan T_s=0,4')

% sinyal diskrit dengan Ts = 0.08
ts2=[0:0.08:3];
xs2=sin(2*pi*1*ts2)+sin(2*pi*2*ts2);
figure(3),clf,stem(ts2,xs2);
grid on,hold on,plot(t,x,'r:'), hold off,...
xlabel('Time(s)'); ylabel('Amplitude'),...
title('sampling x(t) dengan T_s=0,08')

% sinyal diskrit dengan Ts = 0.04
ts3=[0:0.04:3];
xs3=sin(2*pi*1*ts3)+sin(2*pi*2*ts3);
figure(4),clf,stem(ts3,xs3);
grid on,hold on,plot(t,x,'r:'), hold off,...
xlabel('Time(s)'); ylabel('Amplitude'),...
title('sampling x(t) dengan T_s=0,04')

```



Gambar 3.9. Macam-macam Jumlah Sampel Berbeda Terhadap Satu Sinyal Terbangkit

3.3.2 Pengaruh Pemilihan Frekuensi *Sampling* pada Efek Audio

Misalkan $x_1(t) = \cos 2\pi 10t$ dan $x_2(t) = 2 \pi 50t$. Samplinglah kedua sinyal ini menjadi $x_1(n)$ dan $X_2(n)$ dengan $F_s = 40$ Hz dan bandingkan hasilnya. Jika $F_s = 40$ Hz maka $T = 1/40$, dan

$$x_1(n) = x_1(t)|_{t=nT} = \cos 2\pi 10nT = \cos 2\pi(10/40)nT = \cos \pi n/2$$

dan

$$x_2(n) = x_2(t)|_{t=nT} = \cos 2\pi(50/40)nT = \cos(2\pi)(1+(10/40))n = \cos 2\pi n(10/40) = \cos \pi n/2$$

Perhatikan bahwa $x_1 = x_2n!$ Dapat disimpulkan bahwa untuk $F_1 = 40$ Hz, sinyal $F_2 = 50$ Hz adalah alias dari $F_1 = 10$ Hz. Demikian juga untuk $F_k = 10 + F_{sk}$.

Silakan dicoba program MATLAB berikut ini yang menyatakan $x_1(t)$, $x_2(t)$, dan $x_3(t)$, amati dan bandingkan hasilnya. Di samping itu anda harus menjelaskan langkah-langkah dari program tersebut untuk memahami tujuan tentang *sampling* dan *aliasing*.

```
%-----
%Nama File : Sampling_Aliasing.m
%Oleh      :
%-----

Fs=40;
T=1/Fs;
% sinyal asli frekuensi 10 Hz
t=[-pi/2:0.02:pi/2];
x1=cos(2*pi*10*t);
subplot(3,1,1),plot(x1);
```

```

%
t=[-pi/2:T:pi/2];
x2=cos(2*pi*10*t);
subplot(3,1,2),plot(x2);
%
t=[-pi/2:T:pi/2];
x3=cos(2*pi*50*t);
subplot(3,1,3),plot(x3);

```

Di sini kita akan mendengarkan bagaimana pengaruh frekuensi *sampling* melalui sinyal audio. Untuk itu anda harus mempersiapkan PC anda dengan speaker aktif yang sudah terkoneksi dengan *sound card*. Selanjutnya anda ikuti langkah berikut.

1. Buat program baru dengan nama "Sampling.m" menggunakan perintah-perintah seperti berikut ini:

```

%-----
%Nama File : Sampling.m
%Oleh      :
%-----

clear all;
Fs=1000;
t=0:1/Fs:0.25;
f=100;
x=sin(2*pi*f*t);
sound(x,Fs)

```

2. Setelah anda menjalankan program tersebut apa yang anda dapatkan? Selanjutnya coba anda rubah nilai $f = 200, 300, 400, 500, 600, 700, 800,$ dan 900 . Apa yang anda dapatkan? Bentuk suara yang sama dengan frekuensi pembangkitan berbeda itulah yang seringkali disebut orang sebagai efek aliasing. Coba anda catat frekuensi 200 memiliki bunyi yang sama dengan frekuensi berapa? Sehingga frekuensi 200 adalah alias dari frekuensi tsb.

3.3.3 Pengamatan Efek Aliasing pada Audio

Berikut ini sebuah program untuk membangkitkan delapan gelombang sinusoida dengan frekuensi yang berbeda-beda. Dengan memilih frekuensi tertentu akan dibangkitkan sinyal

dengan nada yang berbeda. Program di bawah ini telah diatur frkuensinya agar dapat memberikan nada “do, re, mi, fa, sol, la, si, do”. Agar diketahui perbedaannya, program ini telah dilengkapi dengan perintah untuk mengaktifkan *sound card*. Silakan dicoba. Perlu diingatkan bahwa untuk menghemat waktu, dalam menyalin program ke dalam comman window atau work MATLAB, anda dapat langsung melakukan *copy block* dan *paste*.

```

%-----
%Nama File : Nada.m
%Oleh      :
%-----
clear

clc

Fs=16000;
t=0:1/Fs:0.5;
c=2*sin(2*pi*262*t);
d=2*sin(2*pi*294*t);
e=2*sin(2*pi*330*t);
f=2*sin(2*pi*350*t);
g=2*sin(2*pi*392*t);
a=2*sin(2*pi*440*t);
b=2*sin(2*pi*494*t);
c1=2*sin(2*pi*523*t);
nol = [zeros(size(t))];
nada=[c,d,e,f,g,a,b,c1,nol];
sound(nada)

```

dengan demikian dengan mengatur nada tersebut anda dapat mengatur sedemikian rupa sehingga menjadi suatu lagu yang lengkap.

Berikut ini dipelajari cara menyimpan data dalam format “wav”. Caranya:

1. Susunlah sebuah rangkaian not di atas sehingga membentuk suatu lagu. Berilah nama program sesuai dengan judul lagu. Jalankan dan dengarkan suaranya. Jika sudah benar lakukan langkah berikut:

2. Pada bagian akhir program tambahkan perintah berikut yang digunakan untuk menyimpan program dalam format “wav”.

```
wavwrite(nada,'judul_lagu.wav')
```

3. Coba minimize Matlab, cobalah gunakan Windows Explorer untuk melihat dimana file berada. Kalau sudah terlihat coba click kanan dan jalankan.
4. Coba anda edit program yang telah dibuat, dan lakukan perubahan pada nilai frekuensi sampling $F_s=16000$, menjadi $F_s=10000, 8000, 2000, 1000, 900, 800, 700, 600$, dan 500 . Apa yang anda dapatkan?
5. Dengan $F_s=16000$, lakukan perubahan nilai $t=0:1/F_s:0.1, t=0:1/F_s:0.2, t=0:1/F_s:0.3, t=0:1/F_s:0.7$,

3.3.4 Pengamatan Pembacaan File “wav”

Disini kita akan bermain dengan sebuah lagu yang diambil dari sebuah file *.wav. Untuk itu mulailah dengan langkah.

1. Anda buat program baru seperti berikut ini.

```
%-----  
%Nama File : Baca_Wav.m  
%Oleh      :  
%-----  
% mintalah copy sebuah lagu dengan format wav dan  
% frekuensi sampling 16000 kepada instruktur  
% kemudian buatlah program sebagai berikut:  
  
clear all;  
[Y,Fs]=wavread('How_can_I_tell_her.wav');  
Fs=16000;  
sound(Y,Fs)
```

Copy-kan lagu tersebut dengan nama: “How_can_I_tell_her.wav” dalam *work* dari MATLAB. Jalankan program tersebut? Nikmati beberapa saat. Perhatikan suaranya, apakah normal. Tunggu sampai selesai, karena lagu ini tidak dapat dimatikan sebelum selesai.

2. Lanjutkan langkah anda dengan merubah nilai $F_s = 8000$. Jalankan program anda, dan dengarkan yang terjadi. Suaranya seperti apa?
3. Ulangi lagi dengan merubah nilai $F_s = 22000$, dan 44000 . Apakah kedengaran seperti suara syaitan. Jika ya, beerarti anda bohong, karena anda tidak pernah mendengarkan suara syaitan. Kalau belum puas coba ganti F_s sesuka hati anda. Jangan lupa catat dan buat analisa tentang fenomena yang terjadi dari hasil percobaan.

3.4. Analisis Data

Setelah anda memahami dan puas bermain dengan teorema *sampling*, sekarang saatnya anda melakukan hal yang juga dirasa penting. Apa yang telah anda lakukan pastinya sudah dicatat, buatlah laporan dan analisis mengapa muncul fenomena seperti diatas? Pasti anda dapat menjelaskan secara ilmiah atau matematis, sajikan dalam laporan anda. Fenomena itu lebih dikenal dengan nama apa? Apa yang menyebabkannya? Karena anda telah belajar menyimpan dan memanggil ulang file wav, jelaskan bagaimana suaranya jika frekuensi sampling terlalu rendah atau terlalu tinggi!.

3.5. Tugas

1. Jelaskan. Apa yang dimaksud dengan undersampling dan oversampling? Beri contoh dalam bentuk program!.
2. Buat program yang dapat memanggil sebuah lagu dan sekaligus dapat memainkan lagu tersebut (selain lagu yang diberikan dalam contoh di atas).
3. Buat program yang dapat memanggil sebuah lagu dan sekaligus dapat memainkan lagu tersebut, tapi dengan durasi separuhnya saja (lagu tepotong separuh).
4. Buat program yang dapat memanggil sebuah lagu selain format wav dan sekaligus dapat memainkan lagu tersebut.

MODUL IV

OPERASI KONVOLUSI

Tujuan Praktikum

1. Mahasiswa dapat memahami tujuan dan proses operasi konvolusi pada dua sinyal.
2. Mahasiswa dapat membuat program operasi konvolusi dan mengetahui fungsi konvolusi dari suatu sinyal terhadap sistem.

4.1 DASAR TEORI

4.1.1 Konvolusi dua Sinyal

Berikut ini akan ditunjukkan konvolusi dua sinyal. Konvolusi antara dua sinyal diskrit $x[n]$ dan $v[n]$ dapat dinyatakan sebagai

$$x[n] * v[n] = \sum_{i=-\infty}^{\infty} x[i]v[n-i]$$

Bentuk penjumlahan yang ada di bagian kanan pada persamaan (1) disebut sebagai *convolution sum*. Jika $x[n]$ dan $v[n]$ memiliki nilai 0 untuk semua integer pada $n < 0$, selanjutnya $x[i]=0$ untuk semua integer pada $i < 0$ dan $v[i-n]=0$ untuk semua integer $n-i < 0$ (atau $n < i$). Sehingga jumlahan pada persamaan (1) akan menempati dari nilai $i=0$ sampai dengan $i=n$, dan operasi konvolusi selanjutnya dapat dituliskan sebagai:

$$x[n] * v[n] = \begin{cases} 0 & , n = -1, -2, \dots \\ \sum_{i=0}^n x[i]v[n-i] & , n = 0, 1, 2, \dots \end{cases}$$

4.1.2 Mekanisme Konvolusi

Komputasi pada persamaan (1) dan (2) dapat diselesaikan dengan merubah *discretetime index* n sampai dengan i dalam sinyal $x[n]$ dan $v[n]$. Sinyal yang dihasilkan $x[i]$ dan $v[i]$ selanjutnya menjadi sebuah fungsi *discrete-time index* i . Step berikutnya adalah menentukan $v[n-i]$ dan kemudian membentuk pencerminan terhadap sinyal $v[i]$. Lebih tepatnya $v[-i]$

merupakan pencerminan dari $v[i]$ yang diorientasikan pada sumbu vertikal (axis), dan $v[n-i]$ merupakan $v[-i]$ yang digeser ke kanan dengan step n . Saat pertama kali *product* (hasil kali) $x[i]v[n-i]$ terbentuk, nilai pada konvolusi $x[n]*v[n]$ pada titik n dihitung dengan menjumlahkan nilai $x[i]v[n-i]$ sesuai rentang i pada sederetan nilai integer tertentu.

Untuk lebih jelasnya permasalahan ini akan disajikan dengan suatu contoh penghitung konvolusi pada dua deret nilai integer berikut ini.

Sinyal pertama: $x[i]= 1\ 2\ 3$

Sinyal kedua : $v[i]= 2\ 1\ 3$

- Step pertama adalah pembalikan sinyal kedua, $v[n]$ sehingga didapatkan kondisi seperti berikut:

Sinyal pertama: $x[i] = 1\ 2\ 3$

Sinyal kedua: $v[-i] = 3\ 1\ 2$

- Step ke dua adalah pergeseran dan penjumlahan

Sinyal pertama: $1\ 2\ 3$

Sinyal kedua: $3\ 1\ 2$

----- x

product and sum: $0\ 0\ 2\ 0\ 0 = 2$

- Step ke tiga adalah pergeseran satu step dan penjumlahan

Sinyal pertama: $1\ 2\ 3$

Sinyal kedua: $3\ 1\ 2$

----- x

product and sum: $0\ 1\ 4\ 0 = 5$

- Step ke empat adalah pergeseran satu step dan penjumlahan

Sinyal pertama: $1\ 2\ 3$

Sinyal kedua: $3\ 1\ 2$

----- x

product and sum: $3\ 2\ 6 = 11$

- Step ke lima adalah pergeseran satu step dan penjumlahan

Sinyal pertama: $1\ 2\ 3$

Sinyal kedua: 3 1 2
 ----- x
 product and sum: 0 6 3 0 = 9

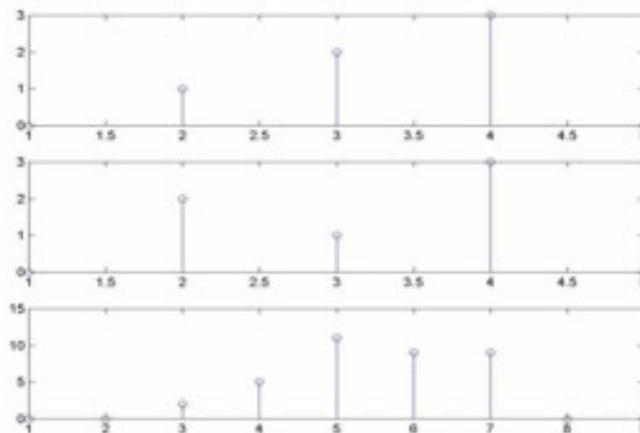
- Step ke enam adalah pergeseran satu step dan penjumlahan

Sinyal pertama: 1 2 3
 Sinyal kedua: 3 1 2
 ----- x
 product and sum: 0 0 9 0 0 = 9

- Step ke tujuh adalah pergeseran satu step dan penjumlahan

Sinyal pertama: 1 2 3
 Sinyal kedua: 3 1 2
 ----- x
 product and sum: 0 0 0 0 0 0 = 0

Dari hasil product and sum tersebut hasilnya dapat kita lihat dalam bentuk deret sebagai berikut: 2 5 11 9 9. Disini hasil penghitungan product and sum sebelum step pertama dan step ke tujuh dan selanjutnya menunjukkan nilai 0, sehingga tidak ditampilkan. Secara grafis dapat dilihat seperti berikut ini:



Gambar 4.1. Mekanisme Konvolusi

Pada Gambar 6.1 bagian atas, menunjukkan sinyal $x[n]$, bagian kedua menunjukkan sinyal $v[n]$, sedangkan bagian ketiga atau yang paling bawah merupakan hasil konvolusi.

4.2 Peralatan

Sistem Operasi Windows dan Perangkat Lunak Matlab yang dilengkapi dengan tool box DSP

4.3 Langkah Percobaan

4.3.1 Konvolusi Dua Sinyal

Misalnya ada dua sekuen sebagai berikut:

$$x(n) = [3, 11, 7, 0, -1, 4, 2], \quad -3 \leq n \leq 3$$

dan

$$h(n) = [2, 3, 0, -5, 2, 1] \quad -1 \leq n \leq 4$$

kemudian akan ditentukan konvolusi $y(n) = x(n) * h(n)$. Jika sekuen acak dengan durasi yang terbatas. MATLAB tidak dapat digunakan secara langsung untuk menghitung konvolusi. MATLAB menyediakan fungsi yang sudah jadi, disebut CONV yang menghitung konvolusi antara dua sekuen dimulai pada $n =$ dan perintahnya adalah:

$$y = \text{conv}(x,h);$$

Untuk mencari nilai-nilai $y(n)$ untuk percobaan di atas program MATLAB-nya adalah:

```
%-----  
%Nama File : Konvolusi_1.m  
%Oleh      :Tri Budi Santoso, Miftahul Huda, Hary Octvianto  
%-----  
x(n) = [3, 11, 7, 0, -1, 4, 2];  
h(n) = [2, 3, 0, -5, 2, 1];  
y = conv(x,h);
```

Walupun demikian, fungsi CONV tidak menyediakan dan tidak pula menerima informasi pewaktuan apapun jika sekuen mempunyai support yang sembarang. Apa yang diperlukan adalah titik awal dan akhir dari $y(n)$. Diberikan suatu durasi tertentu $x(n)$ dan $y(n)$ sangatlah mudah untuk menentukan titik-titik ini. Misalnya:

$$\{x(n); n_{xb} \leq n \leq n_{xe}\} \text{ dan } \{h(n); n_{hb} \leq n \leq n_{he}\}$$

Adalah dua durasi tertentu suatu sekuen. Kita lihat bahwa titik awal dan akhir dari $y(n)$ adalah berturut-turut:

$$n_{yb} = n_{xb} + n_{hb} \text{ dan } n_{ye} = n_{xe} + n_{he}$$

Perluasa dan fungsi conv (disebut conv_m.m) yang melakukan konvolusi dengan dukungan sekuen yang acak dapat dirancang. Perhatikan berikut ini:

```
% [y,ny] = conv_m(x,nx,h,nh)
  % [y,ny] = hasil konvolusi
  % [x,nx] = sinyal pertama
  % [h,nh] = sinyal kedua
  nyb = nx(1) + nh(1);
  nye = nx(length(x)) + nh(length(h));
  ny = [nyb;nye];
  y = conv(x,h);
```

Mari kita coba melakukan konvolusi untuk : $x(n) = [3, 11, 7, 0, -1, 4, 2], -3 \leq n \leq 3$ dan $h(n) = [2, 3, 0, -5, 2, 1], -1 \leq n \leq 4$, sebagai berikut:

```
%-----
%Nama File : Konvolusi_1.m
%Oleh
%-----
x = [3, 11, 7, 0, -1, 4, 2];
nx = [-3:3];
h = [2, 3, 0, -5, 2, 1];
nh = [-1:4];
[n,ny]=conv(x,nx,h,nh)
```

Sehingga:

$$Y(n) = \{6, 31, 47, 6, -51, -5, 41, 18, -22, -3, 8, 2\}$$

4.3.2 Konvolusi Dua Sinyal Discrete Unit Step

Disini kita akan membangkitkan sebuah sinyal unit step diskrit yang memiliki nilai seperti berikut:

$$x(n) = v(v) = \begin{cases} 1 & \text{untuk } 0 \leq n \leq 4 \\ 0 & \text{untuk nilai lain} \end{cases}$$

Dan melakukan operasi konvolusi yang secara matematis dapat dituliskan sebagai berikut:

$$x[n]*v[n]$$

Untuk itu langkah yang harus dilakukan adalah sebagai berikut:

1. Bangkitkan sinyal $x[n]$ dengan mengetikkan perintah berikut:

```
L=input('Panjang gelombang(>=10) : ');
```

```
P=input('Lebar pulsa (lebih kecil dari L): ');
```

```
%-----
%Nama File : Konvolusi_2.m
%Oleh      :Tri Budi Santoso, Miftahul Huda, Hary Octvianto
%-----
%Sinyal x(n)
for n=1:L
if n<=P
x(n)=1;
else
x(n)=0;
end
end
t=1:L;
subplot(3,1,1)
stem(t,x)
```

2. Jalankan program dan tetapkan nilai $L=20$ dan $P=10$.
3. Selanjutnya masukkan pembangkitan sekuen unit step ke dua dengan cara menambahkan syntax berikut ini di bawah program anda pada langkah pertama:

```
%Sinyal v(n)
```

```

for n=1:L
if n<=P
v(n)=1;
else
v(n)=0;
end
end
t=1:L;
subplot(3,1,2)
stem(t,v)

```

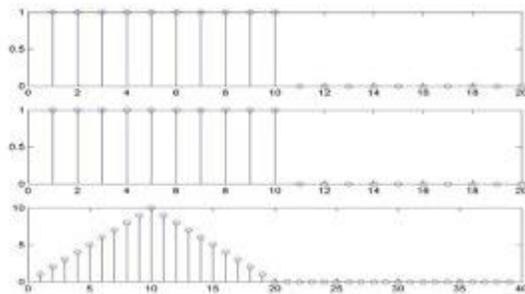
4. Coba jalankan program dan tambahkan perintah berikut:

```

subplot(3,1,3)
stem(conv(x,v)) %Konvolusi sinyal: x(n)*v(n)

```

5. Coba anda jalankan seperti pada langkah kedua, dan apakah hasilnya seperti ini?



Gambar 4.2. Contoh Hasil Konvolusi

6. Ulangi langkah ke 5 dan rubahlah nilai untuk $L=12$, 15 , dan 12 . Sedangkan untuk P masukkan nilai 10 , 5 , dan 12 , apa yang terjadi?

4.3.3 Konvolusi Dua Sinyal Sinus

Di sini kita mencoba untuk membangkitkan dua sinyal sinus dan melakukan operasi konvolusi untuk keduanya. Langkah yang harus anda lakukan adalah sebagai berikut:

1. Buat program untuk membangkitkan dua gelombang sinus seperti berikut:

- L =input('Banyaknya titik sampel(>=20): ');
- f1 =input('Besarnya frekuensi gel 1 adalah Hz: ');
- f2 =input('Besarnya frekuensi gel 2 adalah Hz: ');
- teta1=input('Besarnya fase gel 1(dalam radiant): ');
- teta2=input('Besarnya fase gel 2(dalam radiant): ');
- A1 =input('Besarnya amplitudo gel 1: ');
- A2 =input('Besarnya amplitudo gel 2: ');

```
%-----  
%Nama File : Konvolusi_3.m  
%Oleh      :Tri Budi Santoso, Miftahul Huda, Hary Octvianto  
%-----  
%Sinus pertama  
t=1:L;  
t=2*t/L;  
y1=A1*sin(2*pi*f1*t + teta1*pi);  
subplot(3,1,1)  
stem(y1)  
%Sinus kedua  
t=1:L;  
t=2*t/L;  
y2=A2*sin(2*pi*f2*t + teta2*pi);  
subplot(3,1,2)  
stem(y2)
```

2. Coba anda jalankan program anda dan isikan seperti berikut ini:

Banyaknya titik sampel(>=20): 20
Besarnya frekuensi gel 1 adalah Hz: 1
Besarnya frekuensi gel 2 adalah Hz: 0.5
Besarnya fase gel 1(dalam radiant): 0
Besarnya fase gel 2(dalam radiant): 0.5
Besarnya amplitudo gel 1: 1

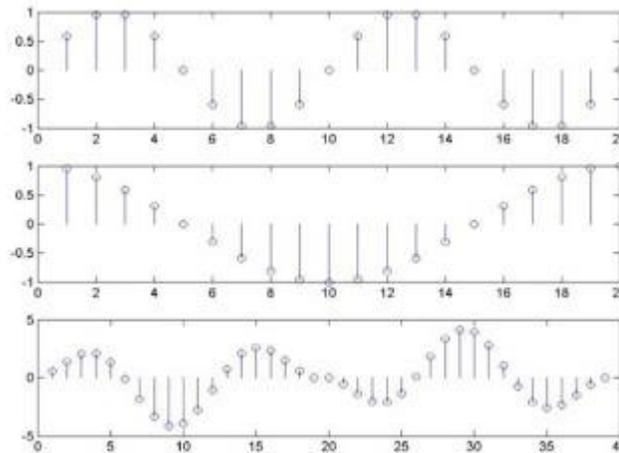
Besarnya amplitudo gel 2: 1

Perhatikan tampilan yang dihasilkan. Apakah ada kesalahan pada program anda?

3. Lanjutkan dengan menambahkan program berikut ini pada bagian bawah program yang anda buat tadi.

```
subplot(3,1,3)  
stem(conv(y1,y2))
```

4. Jalankan program anda, dan kembali lakukan pengisian seperti pada langkah ke 3. Lihat hasilnya apakah anda melihat tampilan seperti berikut?



Gambar 4.3. Contoh Hasil Konvolusi Dua Sinyal Sinus

5. Ulangi langkah ke 4, dengan menetapkan nilai sebagai berikut: $L=50$, $w_1=w_2=2$, $teta_1=1.5$, $teta_2=0.5$, dan $A_1=A_2=1$. Apa yang anda dapatkan? Apakah anda mendapatkan hasil yang berbeda dari program sebelumnya? Mengapa ?

4.3.4 Konvolusi Sinyal Bernoise dengan Raise Cosine

Sekarang kita mulai mencoba utnuk lebih jauh melihat implementasi dari sebuah operasi konvolusi. Untuk itu ikuti langkah-langkah berikut.

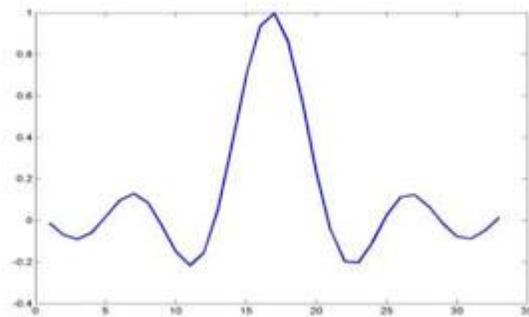
1. Bangkitkan sinyal raise cosine dan sinyal sinus dengan program berikut.

```
%-----
```

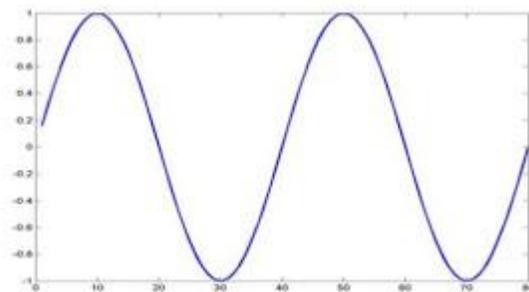
```

%Nama File : Konvolusi_3.m
%Oleh      :Tri Budi Santoso, Miftahul Huda, Hary Octvianto
%-----
%convolusi sinyal sinus bernoise dengan raise cosine;
n=-7.9:.5:8.1;
y=sin(4*pi*n/8)/(4*pi*n/8);
figure(1);
plot(y,'linewidth',2)
t=0.1:.1:8;
x=sin(2*pi*t/4);
figure(2);
plot(x,'linewidth',2)

```



Gambar 4.4. Sinyal Raise Cosine



Gambar 4.5. Sinyal Sinus Asli

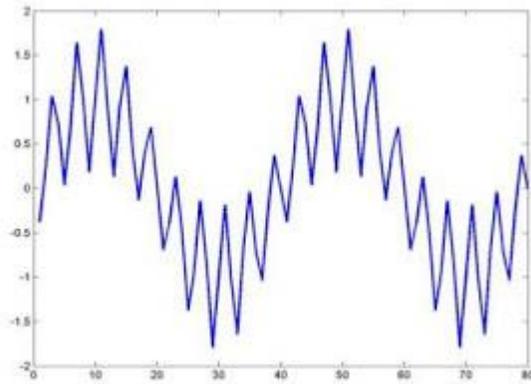
2. Tambahkan noise pada sinyal sinus.

```
t=0.1:.1:8;
```

```

x_n=sin(2*pi*t/4)+0.5*randn*sin(2*pi*10*t/4)+... 0.2*randn*sin(2*pi*12*t/4);
figure(3);
plot(x_n,'linewidth',2)

```



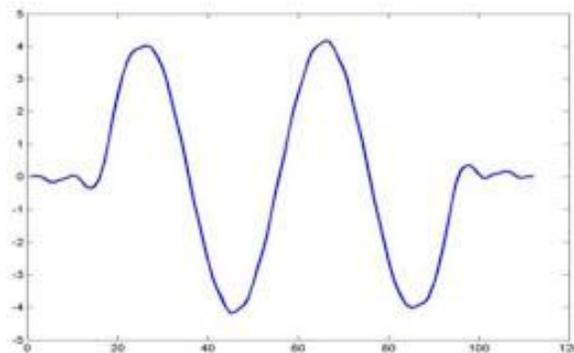
Gambar 4.6. Sinyal Sinus Bernoise

3. Lakukan konvolusi sinyal sinus bernoise dengan raise cosine, perhatikan apa yang terjadi?

```

xy=conv(x_n,y);
figure(4);
plot(xy,'linewidth',2)

```



Gambar 4.7. Hasil Konvolusi

4. Coba anda lakukan perubahan pada nilai sinyal raise cosine dengan mengurangi rentang nilai pada n, bisa anda buat lebih pendek atau lebih panjang, dan ulangi lagi langkah 3, catat apa yang terjadi.

4.3.5 Konvolusi Pada Sinyal Audio

Coba kita lihat bersama bagaimana pengaruh operasi konvolusi pada sinyal audio, dalam hal ini kita ulangi permainan seperti pada modul sebelumnya. Untuk itu ikuti langkah berikut.

1. Buat sebuah program baru sebagai berikut:

```
%-----  
%Nama File : Konvolusi_4.m  
%Oleh      :Tri Budi Santoso, Miftahul Huda, Hary Octvianto  
%-----  
%convolusi_1.m  
  
clear all;  
[Y,Fs] = wavread('lagu_1_potong.wav');  
Fs = 16000;%nilai default Fs=16000  
sound(Y,Fs)
```

Apa yang anda dapatkan?

2. Beri tanda % pada sound(Y,Fs) untuk membuatnya tidak dieksekusi oleh Matlab, sehingga menjadi % sound(Y,Fs). Kemudian tambahkan perintah berikut.

```
nois = randn(length(Y),1);  
Y_noise = Y + 0.08*nois;  
sound(Y_noise,Fs)
```

Coba amati lagi apa yang terjadi?

3. Buat perintah sound tidak aktif, kemudian bangkitkan sebuah sinyal yang bernilai 1 dengan cara seperti berikut.

```
satu = ones(4,1);
```

4. Lakukan operasi konvolusi dan dengarkan hasilnya pada speaker anda

```
Y_c = conv(satu,Y_noise);  
sound(Y_c,Fs)
```

4.4. Analisa Data

Setelah anda melakukan praktikum dengan sukses, ada satu hal yang selalu anda harus lakukan, yaitu membuat dokumentasi dari pekerjaan anda. Untuk itu cobalah anda catat semua yang telah anda lakukan dan jawablah beberapa pertanyaan berikut ini:

1. Bagaimana bentuk dasar dari sebuah operasi konvolusi?
2. Apa pengaruh operasi konvolusi pada sinyal sinus bernoise?
3. Dimana pemakaian operasi konvolusi pada system yang pengolah audio?

4.5. Tugas

1. Demonstrasikan sebah aplikasi *crosscorrelation* menggunakan program MATLAB jika diketahui:

$$X(n) = [3, 11, 7, 0, -1, 4, 2]$$

Adalah prototipe sekuen dan misalkan $y(n)$ adalah versi *noise-corrupted* and *shifted*-nya;

$$Y(n) = x(n-2)-W(n)$$

Dengan $W(n)$ adalah sekuen Gaussian dengan rata-rata (mean) 0 dan varian 1. Hitug *crosscorrelation* antara $y(n)$ dan $x(n)$

2. Misalkan pulsa rectangular $x(n) = u(n) - u(n-10)$ dijadikan sebagai masukan pada sistem LTI dengan impulse respon

$$h(n) = (0,9)^n u(n)$$

Tentukan keluaran $y(n)$ dan plot hasil impulse $x(n)$ fungsi respon ($h(n)$) dan keluaran ($y(n)$) dan diskusikan hasilnya.

3. Diberikan dua sekuen sebagai berikut:

$$x(n) = [3, 11, 7, 0, -1, 4, 2], -3 \leq n \leq 3; \text{ dan } h(n) = [2, 3, 0, -5, 2, 1], -1 \leq n \leq 4$$

tentukan konvolusi $y(n) = x(n)*h(n)$

DAFTAR PUSTAKA

1. Hans J. W., (*penerjemah*), 1996, "*Sinyal dan Sistem Linier*", Edisi ke-3, Erlangga, Jakarta.
2. Robert, M. J., "**Signal and System**", Mc Graw Hill, New York.
3. Simon H., Barry V. V. , 2004, "*Sinyal and Sistem*", John Wiley & Son, New York.
4. Tanudjaja H, 2007, "*Pengolahan Sinyal Digital dan Sistem Pemrosesan Sinyal*", Andi, Yogyakarta.