BUKU

AJAR

MATA KULIAH MIKROKONTROLER

HAPPY NUGROHO, S.T., M.T.

LABORATORIUM KOMPUTASI & TEKNOLOGI INFORMASI

UNIVERSITAS MULAWARMAN

FAKULTAS TEKNIK

PROGRAM STUDI TEKNIK ELEKTRO

TAHUN 2022

LAB 1. **Manipulasi Display LCD**

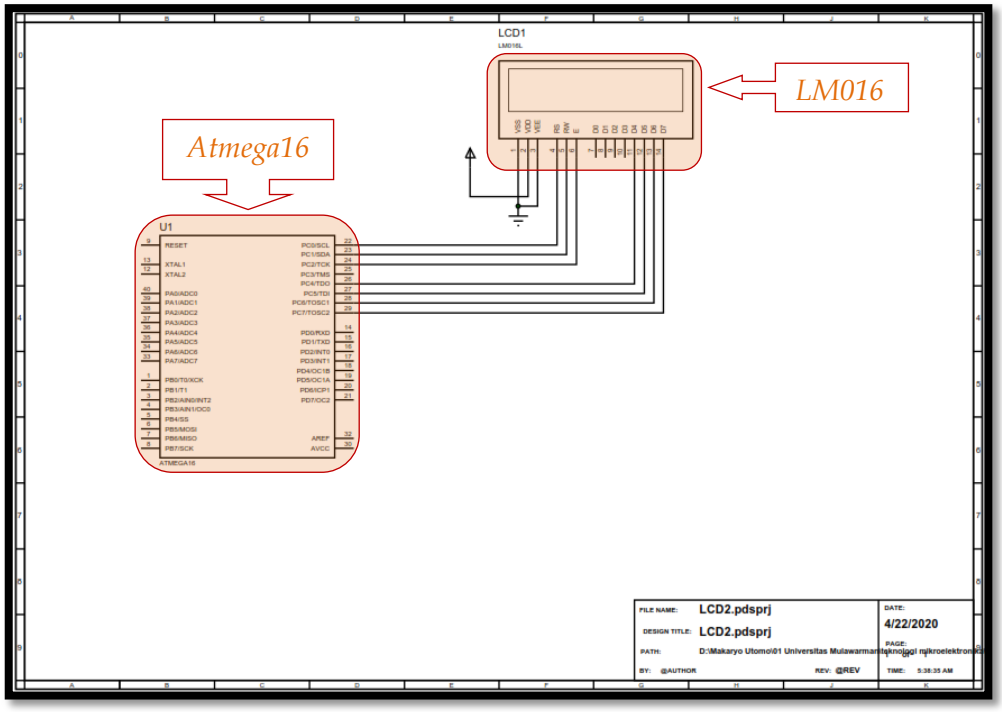
Asisten Percobaan :

NIM :

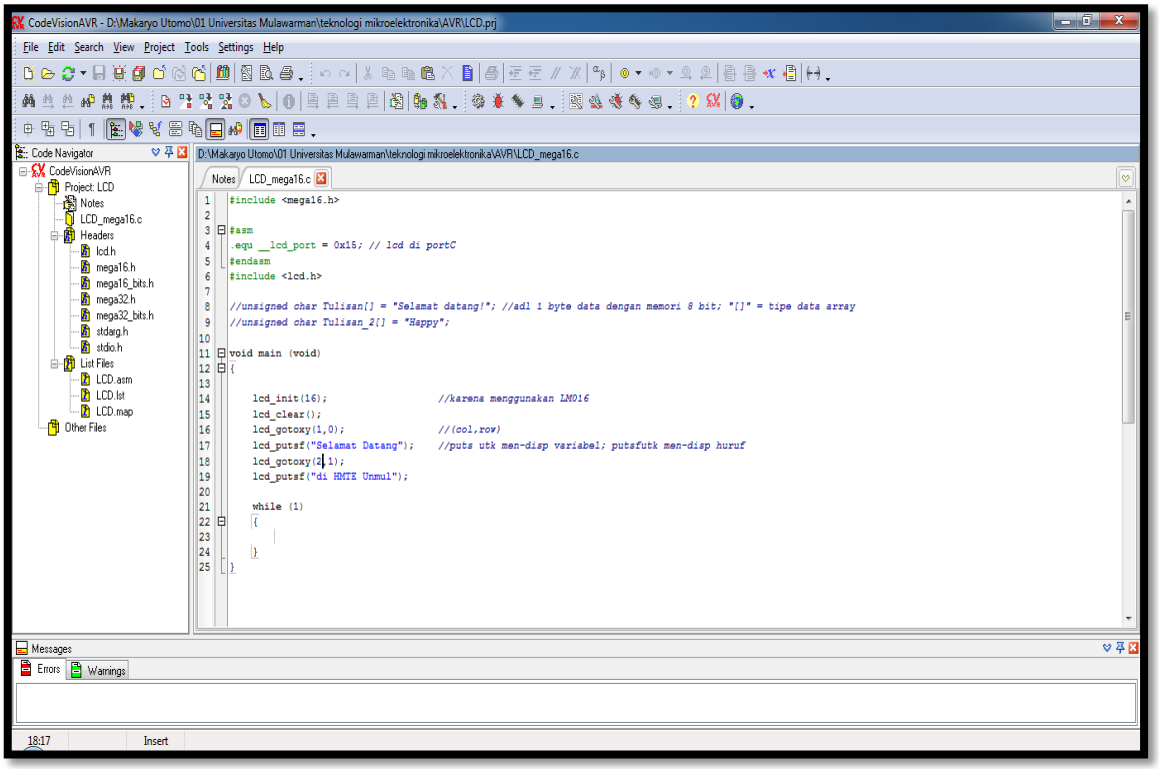
Tanggal Percobaan :

Lab 1. Display LCD

Latihan menampilkan kalimat ke dalam layar LCD



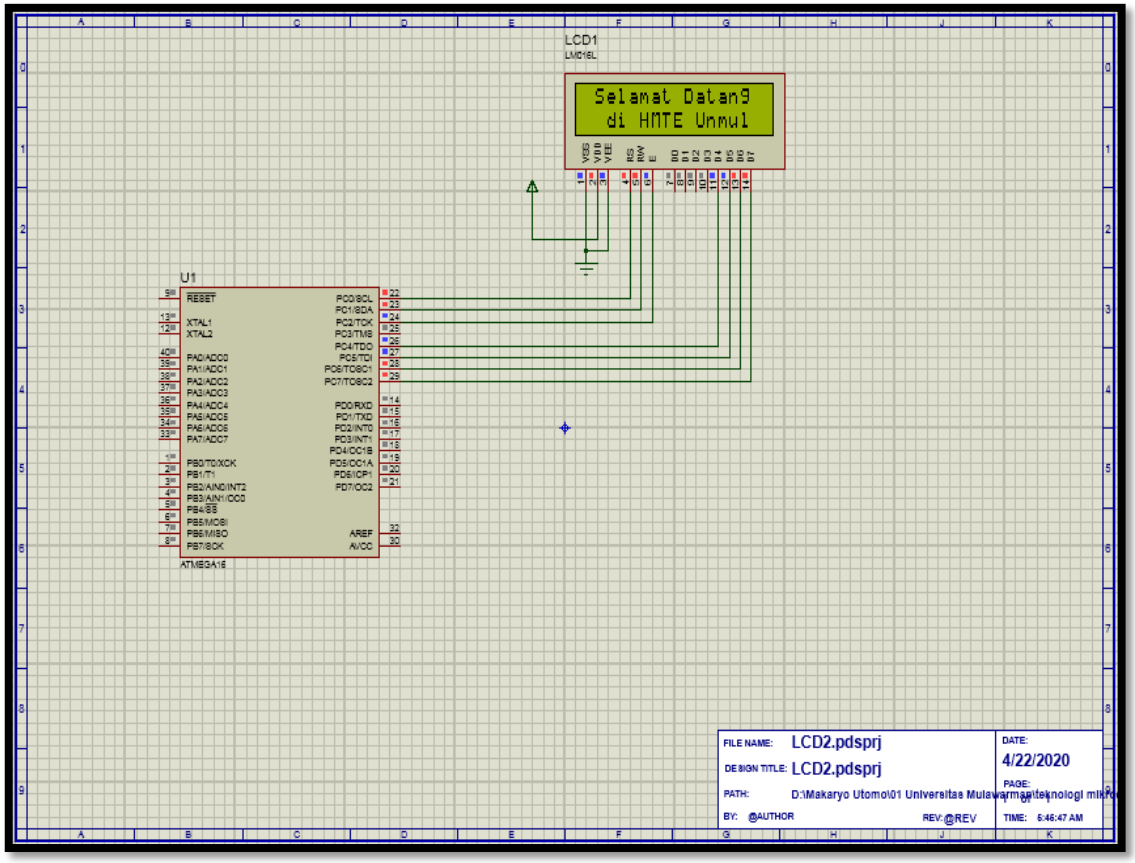
Gambar 1 Rangkaian skematik display LCD



Gambar 2. *Coding* untuk menuliskan instruksi pada *Code Vision AVR*

Langkah mempersiapkan Alat dan Bahan, yakni:

1. Laptop yang telah ter-*install* aplikasi *Proteus 8 Professional* dan *Code Vision* AVR!
2. Merancang Rangkaian Skematik seperti terlihat pada Gambar 1, menggunakan aplikasi *Proteus 8 Professional* pada laptop masing-masing!
3. Menuliskan "*Coding*" seperti terlihat pada Gambar 2 ke dalam aplikasi *Code Vision* AVR!
4. Meng-*Upload* "*Coding*" yang telah dituliskan, ke dalam desain rancangan Rangkaian Skematik kalian masing-masing!
5. *Screenshoot* atau *Capture* hasilnya yang tampak di *LCD LM016*! seperti tampak pada Gambar 3 di bawah.



Gambar 3. Contoh hasil percobaan Lab. 1

6. Meng-*Upload* file (*fullscreen*!) dalam bentuk format ***.pdf** ke Google Classroom, antara lain:
   a. Rangkaian Skematik
   b. *Coding*
   c. Hasil *Capture*

LAB 2. **Merancang Jam Digital DS1302 dan LM016**

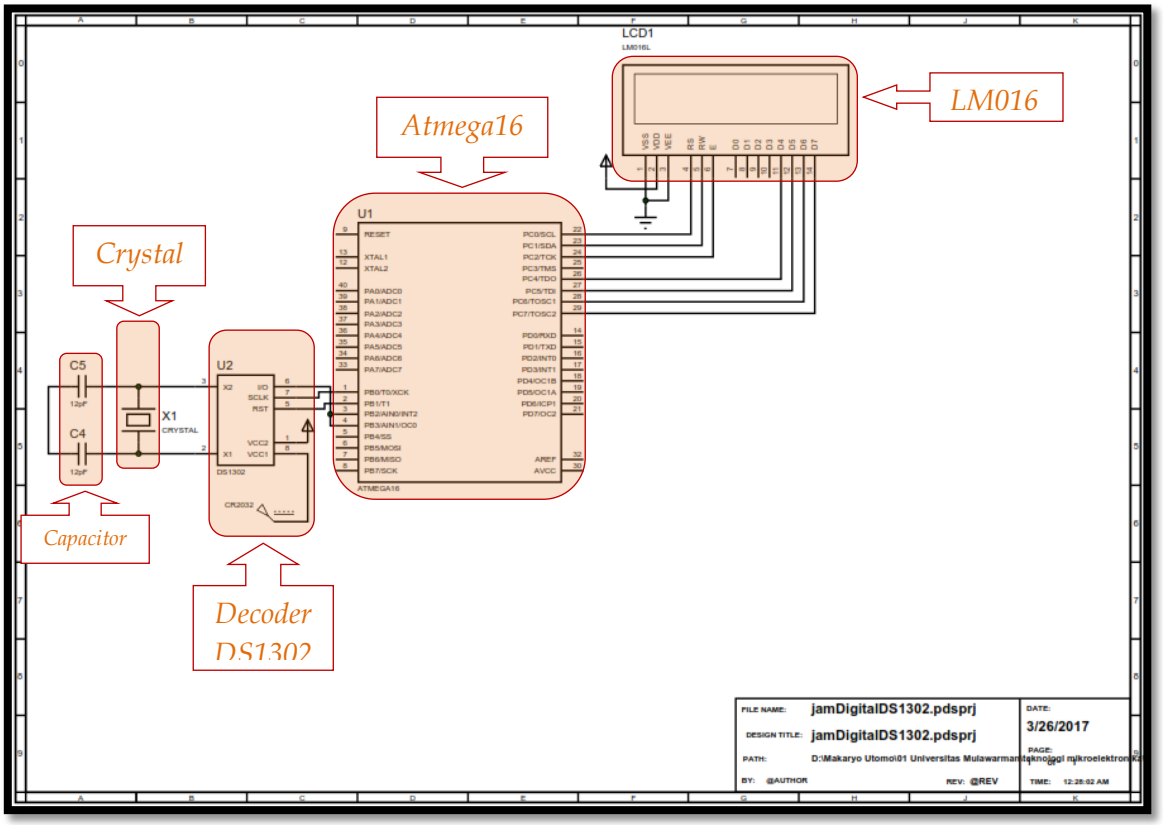
Asisten Percobaan :

NIM :

Tanggal Percobaan :

Lab 2. Merancang Jam Digital

Latihan merancang sebuah jam digital dengan dimensi jam:menit:detik, menggunakan *chip Real Time Clock* (RTC) DS1302 dan menampilkannya ke LCD LM016.

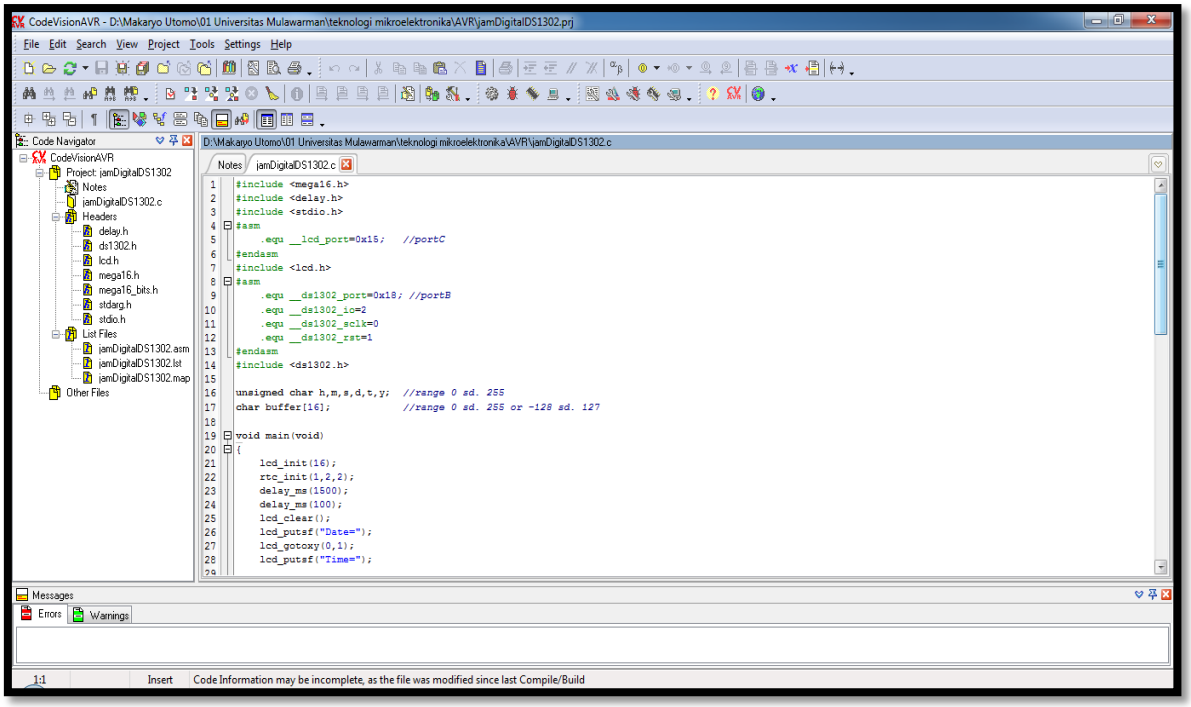


Gambar 4 Rangkaian skematik Jam Digital

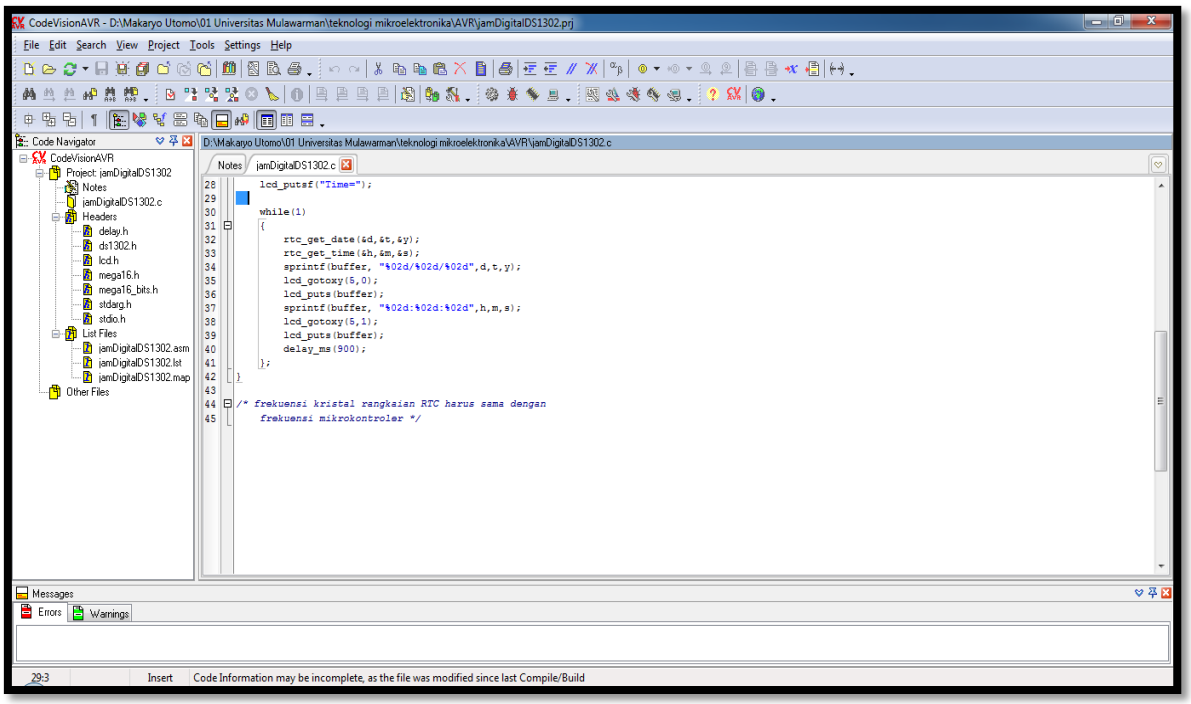
Langkah mempersiapkan Alat dan Bahan, yakni:

1. Laptop yang telah ter-*install* aplikasi *Proteus 8 Professional* dan *Code Vision* AVR!
2. Merancang Rangkaian Skematik seperti terlihat pada Gambar 4, menggunakan aplikasi *Proteus 8 Professional* pada laptop masing-masing!
3. Menuliskan "*Coding*" seperti terlihat pada Gambar 5 ke dalam aplikasi *Code Vision* AVR!
4. Meng-*Upload* "*Coding*" yang telah dituliskan, ke dalam desain rancangan Rangkaian Skematik kalian masing-masing!
5. *Screenshoot* atau *Capture* hasilnya! seperti tampak pada Gambar 6.
6. Meng-*Upload* file (*fullscreen*!) dalam bentuk format ***.pdf** ke Google Classroom, antara lain:
   a. Rangkaian Skematik
   b. *Coding*
   c. Hasil *Capture*

Adapun *Coding* yang digunakan dalam Modul Lab 2 adalah:



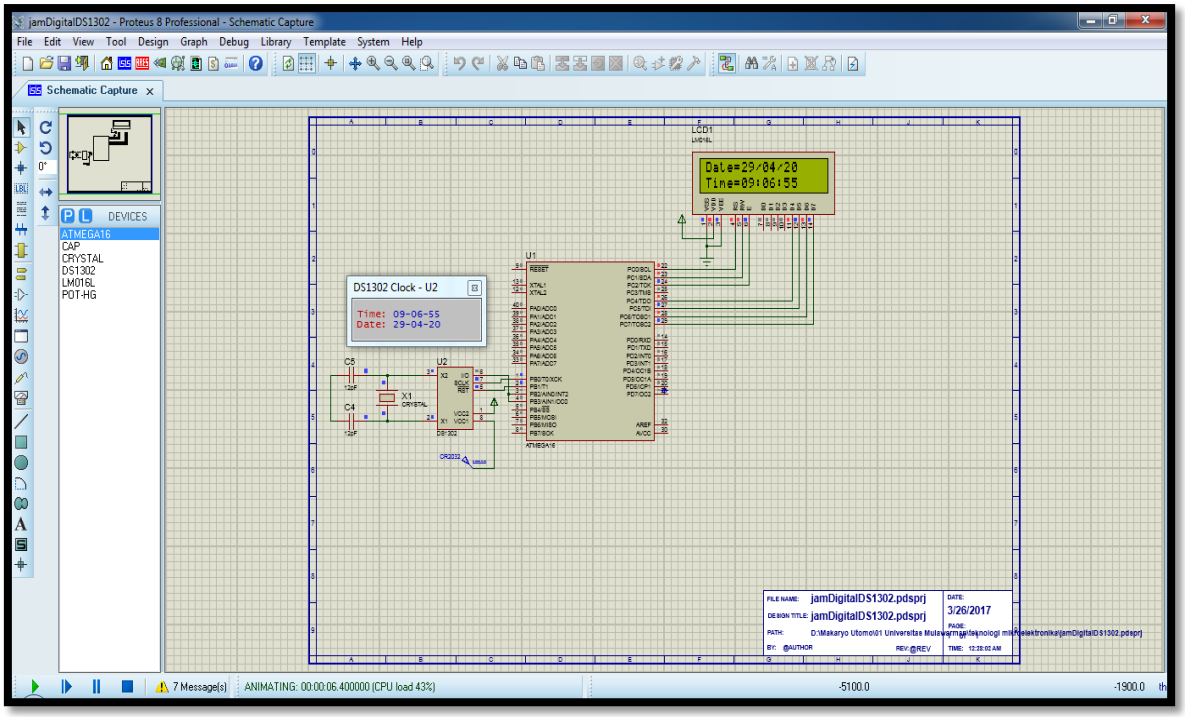
(a)



(b)

Gambar 5.(a) (b). *Coding* untuk menuliskan instruksi pada *Code Vision AVR*

Hasil *Capture* Percobaan Lab 2 (merancang jam digital), yakni:



Gambar 6. Contoh hasil percobaan Lab. 2

LAB 3.

Manipulasi Data I/O (Part I)

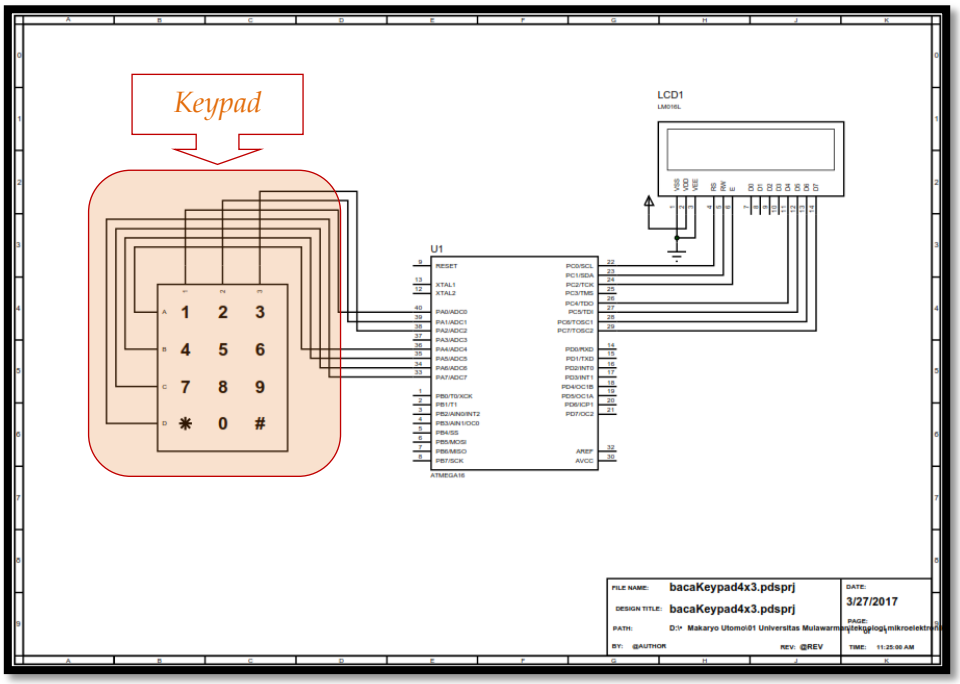
Asisten Percobaan :

NIM :

Tanggal Percobaan :

Lab 3. Manipulasi Data I/O (Part I)

Latihan menggunakan *Matrix Push Buttons* (*keypad*) berdimensi 4x3 dan Display LM016.

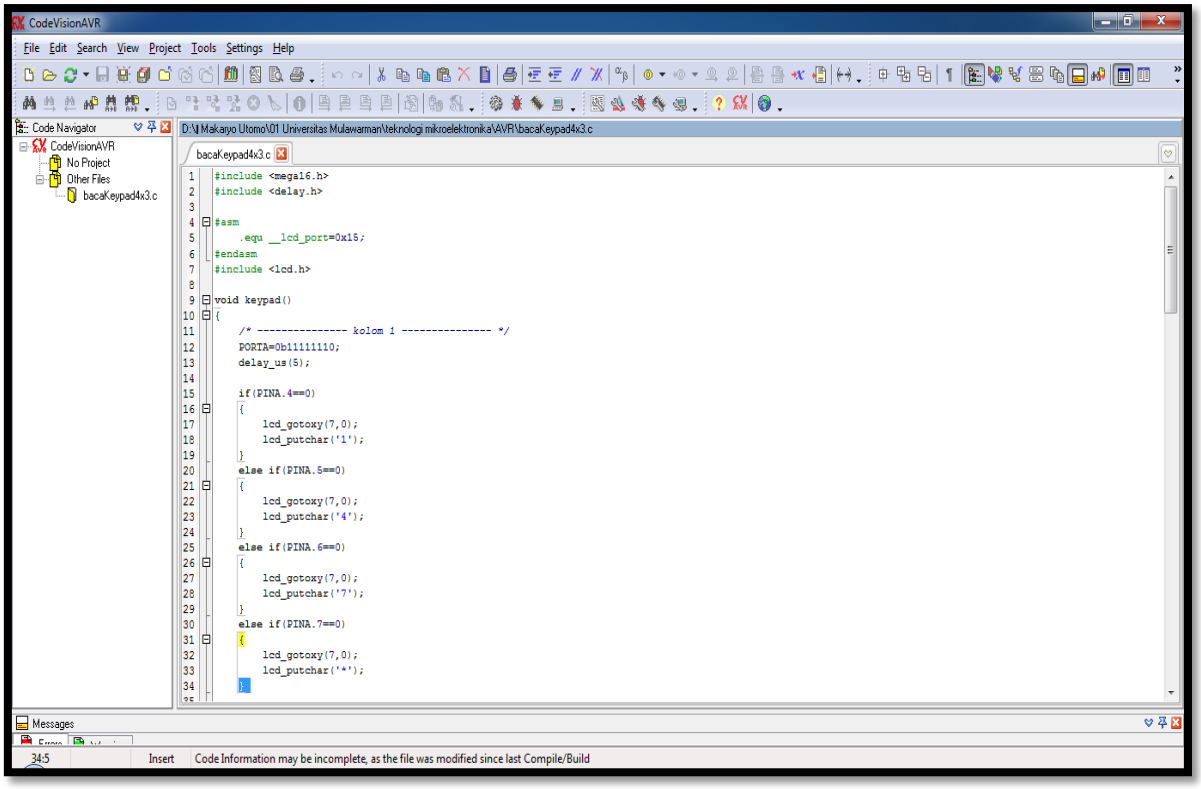


Gambar 7. Rangkaian skematik kontrol keypad

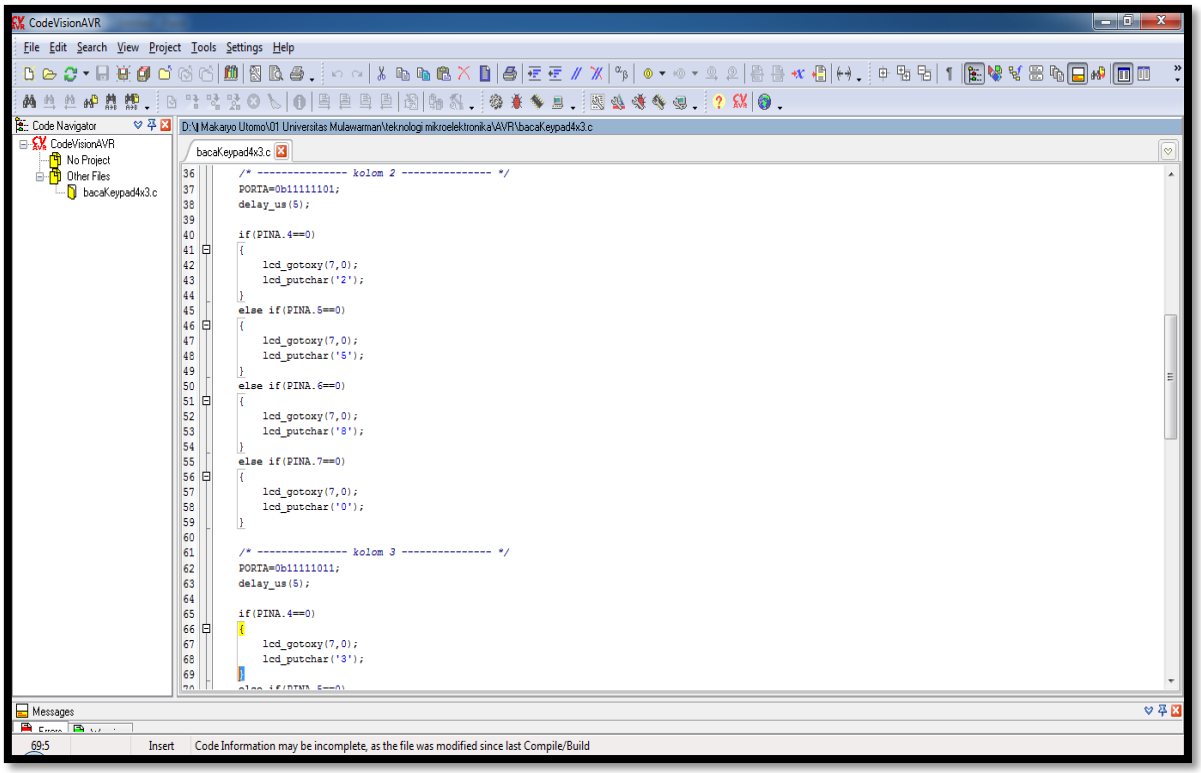
Langkah mempersiapkan Alat dan Bahan, yakni:

1. Laptop yang telah ter-*install* aplikasi *Proteus 8 Professional* dan *Code Vision* AVR!
2. Merancang Rangkaian Skematik seperti terlihat pada Gambar 7, menggunakan aplikasi *Proteus 8 Professional* pada laptop masing-masing!
3. Menuliskan "*Coding*" seperti terlihat pada Gambar 8 ke dalam aplikasi *Code Vision* AVR!
4. Meng-*Upload* "*Coding*" yang telah dituliskan, ke dalam desain rancangan Rangkaian Skematik kalian masing-masing!
5. *Screenshoot* atau *Capture* hasilnya! seperti tampak pada Gambar 9.
6. Meng-*Upload* file (*fullscreen*!) dalam bentuk format ***.pdf** ke Google Classroom, antara lain:
   a. Rangkaian Skematik
   b. *Coding*
   c. Hasil *Capture*

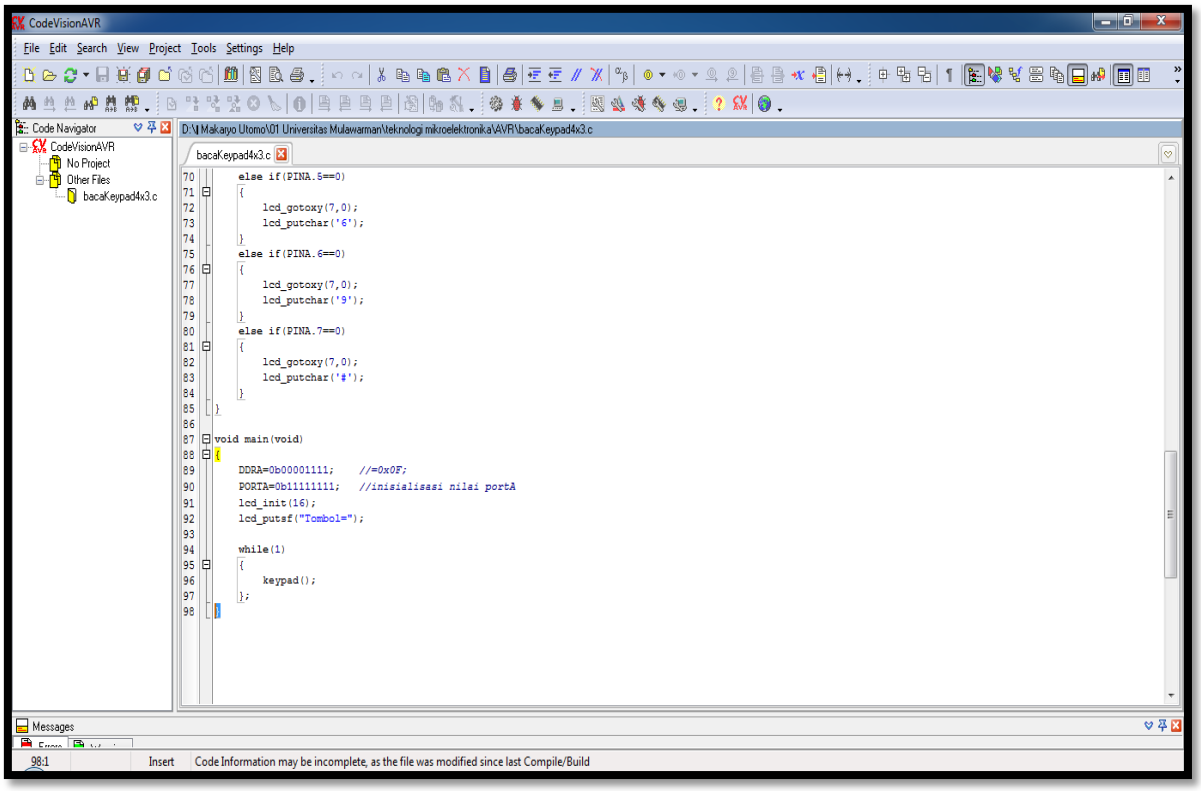
Adapun *Coding* yang digunakan dalam Modul Lab 3 (kontrol keypad) adalah:



(a)



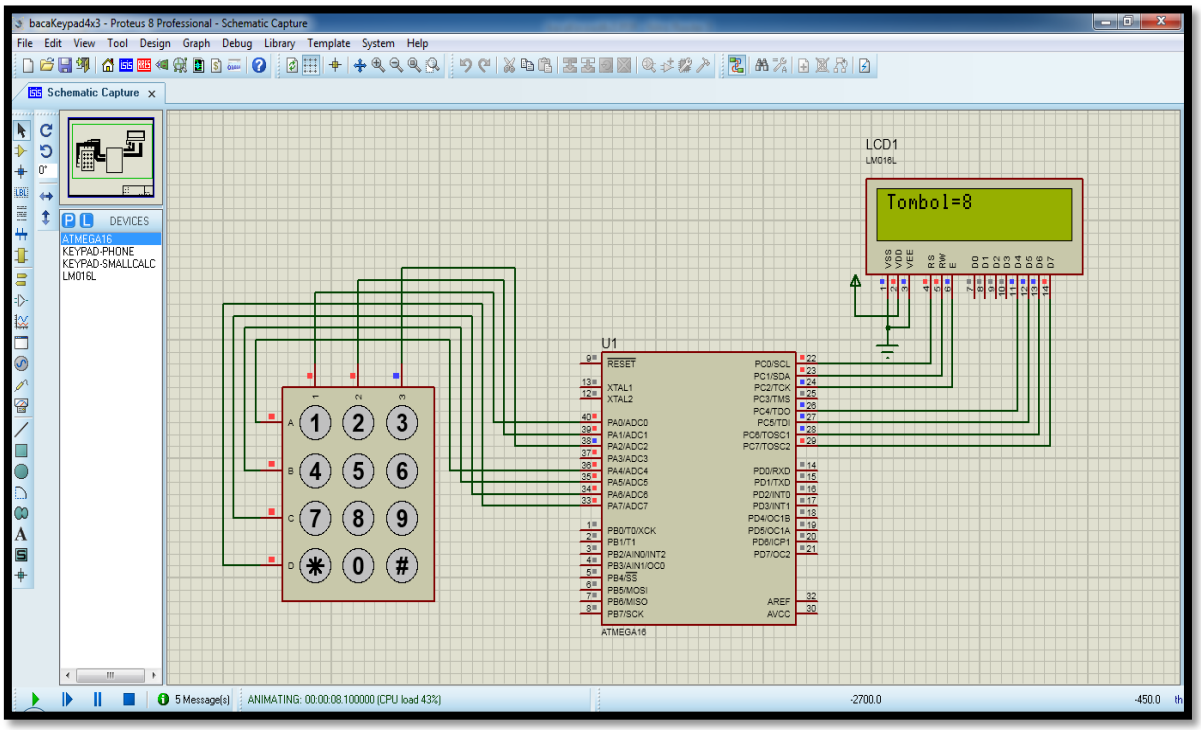
(b)



(c)

Gambar 8 (a) (b) (c). *Coding* untuk menuliskan instruksi pada *Code Vision AVR*

Hasil *Capture* Percobaan Lab 3 (kontrol *keypad*), yakni:



Gambar 9. Contoh hasil percobaan Lab 3

LAB 3.

Manipulasi Data I/O (Part II)

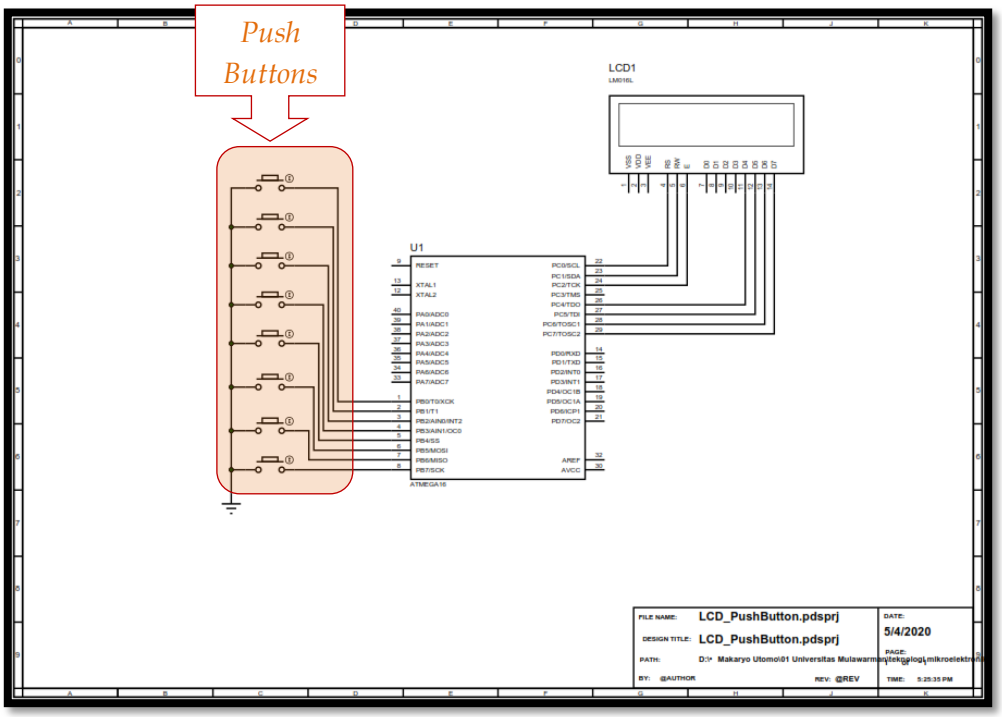
Asisten Percobaan :

NIM :

Tanggal Percobaan :

Lab 3. Manipulasi Data I/O (Part II)

Latihan menggunakan *Push Buttons* 8 unit sebagai input, untuk menampilkan karakter bertipe data *char* pada Display LM016.

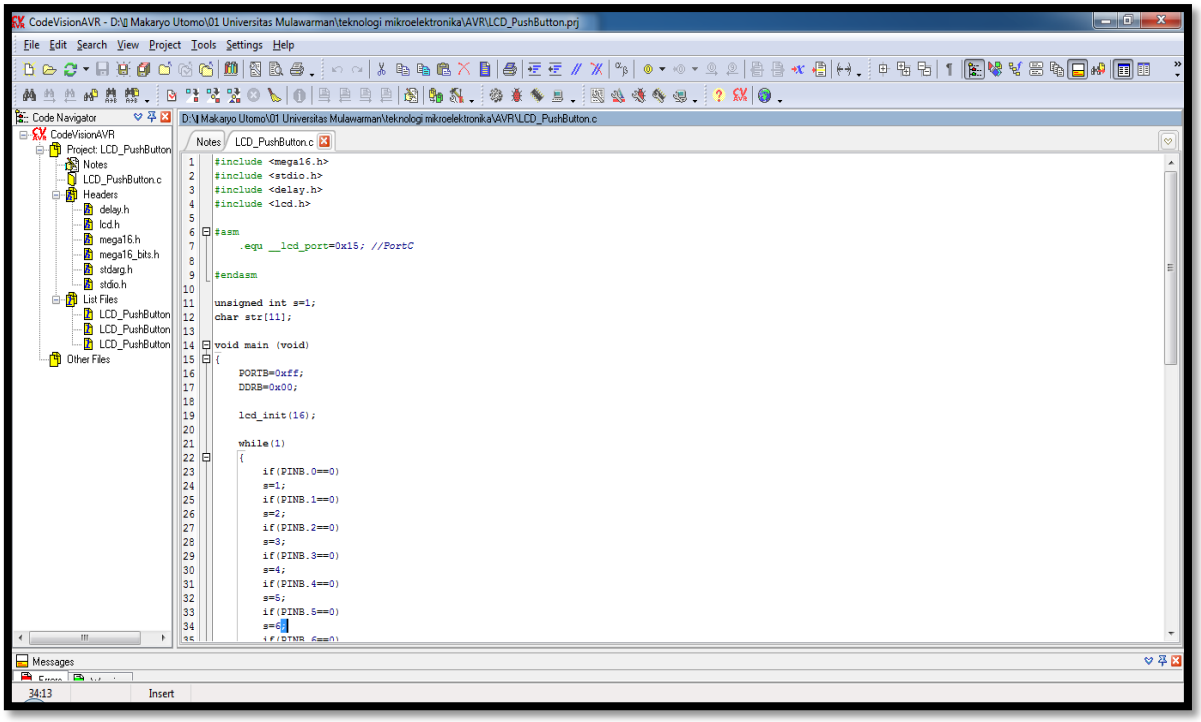


Gambar 10. Rangkaian skematik kontrol *push buttons*

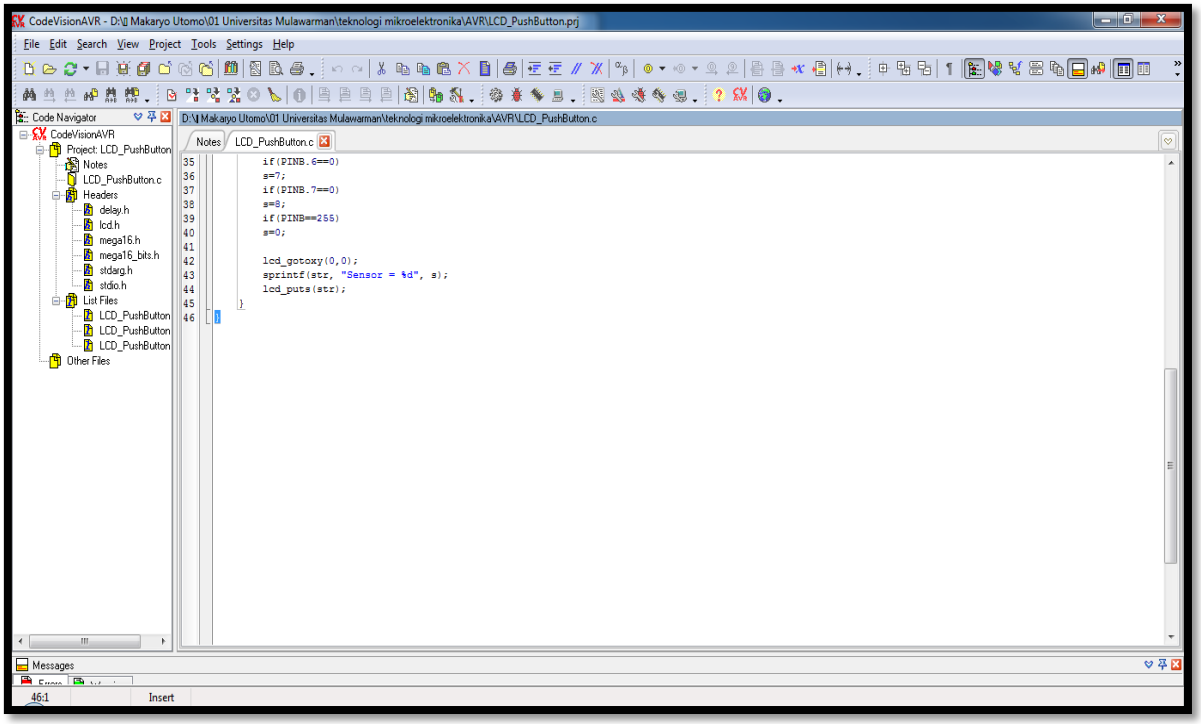
Langkah mempersiapkan Alat dan Bahan, yakni:

1. Laptop yang telah ter-*install* aplikasi *Proteus 8 Professional* dan *Code Vision* AVR!
2. Merancang Rangkaian Skematik seperti terlihat pada Gambar 10, menggunakan aplikasi *Proteus 8 Professional* pada laptop masing-masing!
3. Menuliskan "*Coding*" seperti terlihat pada Gambar 11 ke dalam aplikasi *Code Vision* AVR!
4. Meng-*Upload* "*Coding*" yang telah ditulis, ke dalam desain rancangan Rangkaian Skematik kalian masing-masing!
5. *Screenshoot* atau *Capture* hasilnya! seperti tampak pada Gambar 12.
6. Meng-*Upload* file (*fullscreen*!) dalam bentuk format ***.pdf** ke Google Classroom, antara lain:
   a. Rangkaian Skematik
   b. *Coding*
   c. Hasil *Capture*

Adapun *Coding* yang digunakan dalam Modul Lab 3 (kontrol *push buttons*) adalah:



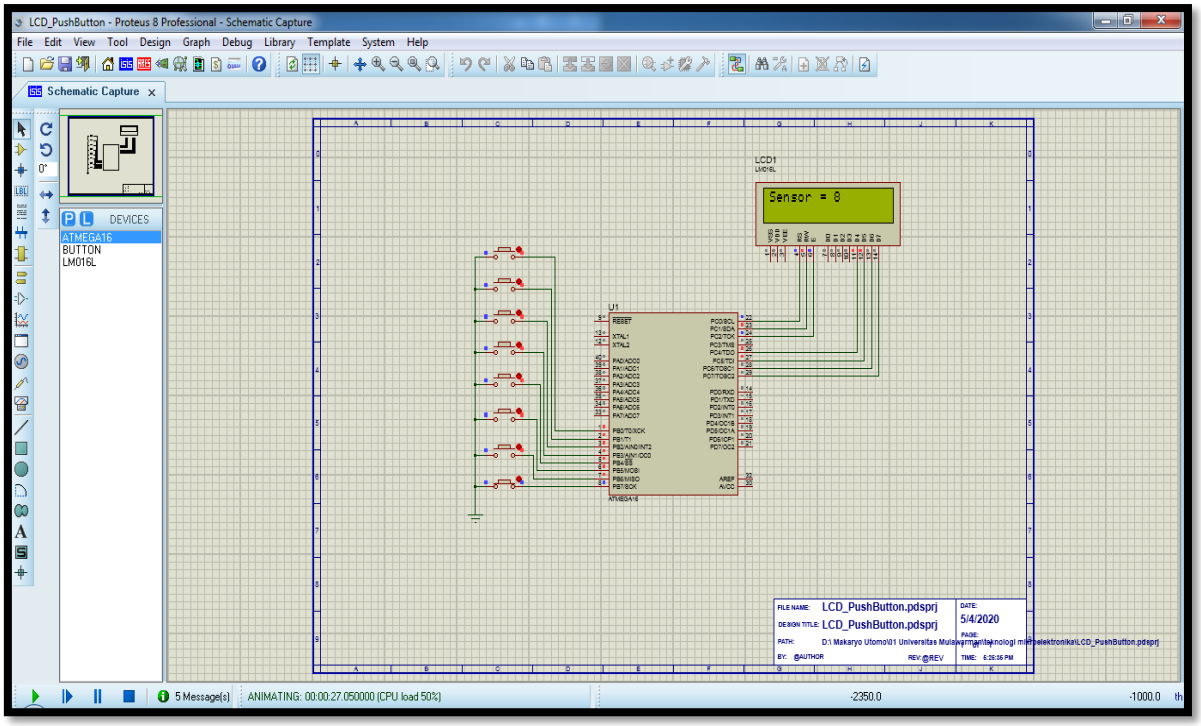
(a)



(b)

Gambar 11 (a) (b). *Coding* pada *Code Vision AVR*

Hasil *Capture* Percobaan Lab 3 (kontrol *push buttons*), yakni:



Gambar 12. Contoh hasil percobaan Lab 3

LAB 4.

Manipulasi Data I/O (Part III)

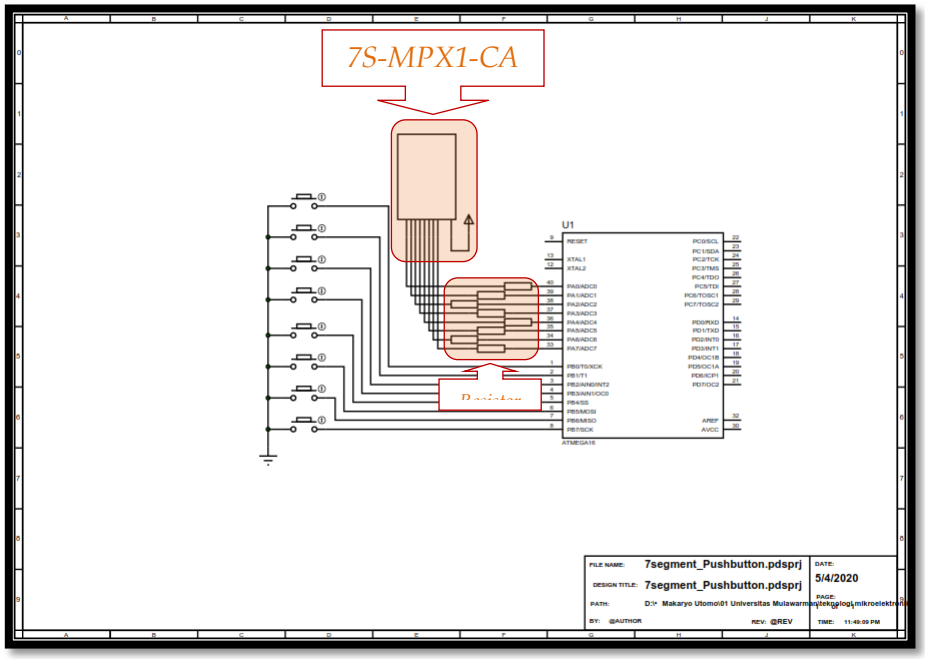
Asisten Percobaan :

NIM :

Tanggal Percobaan :

Lab 4. Manipulasi Data I/O (Part I)

Latihan menggunakan *Push Buttons* 8 unit sebagai input, untuk menampilkan angka atau bilangan ke dalam Display *seven segment* 7S-MPX1-CA.

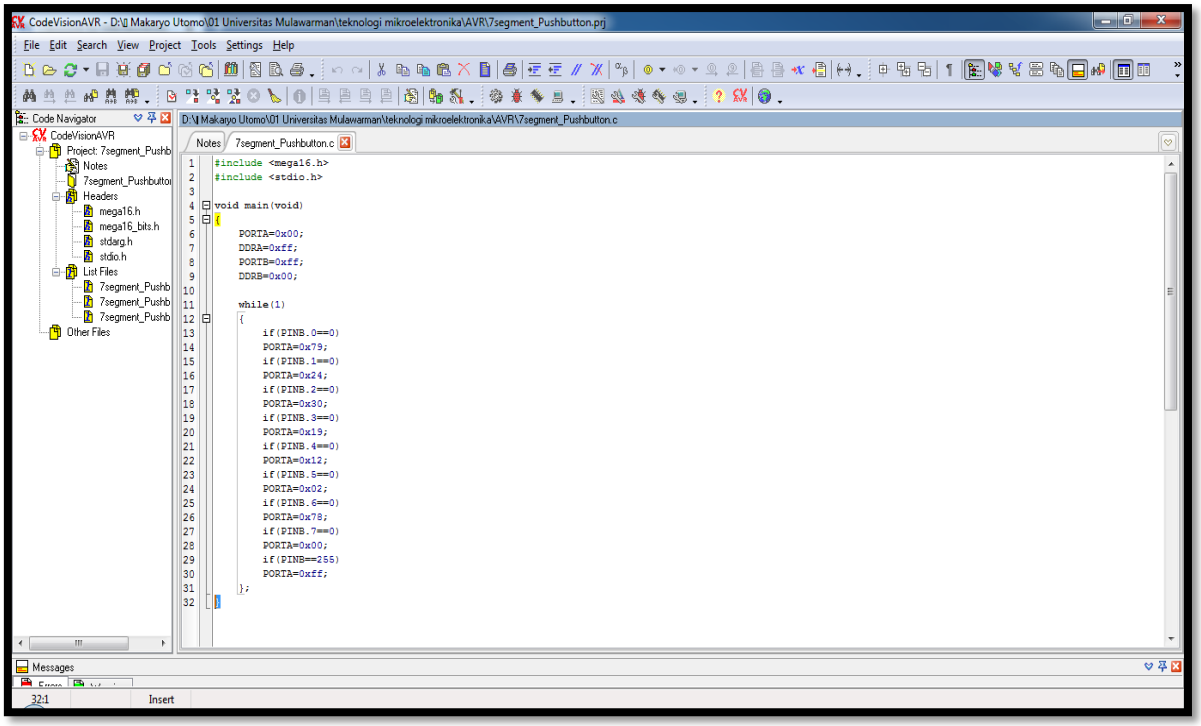


Gambar 13. Rangkaian skematik kontrol *push buttons*

Langkah mempersiapkan Alat dan Bahan, yakni:

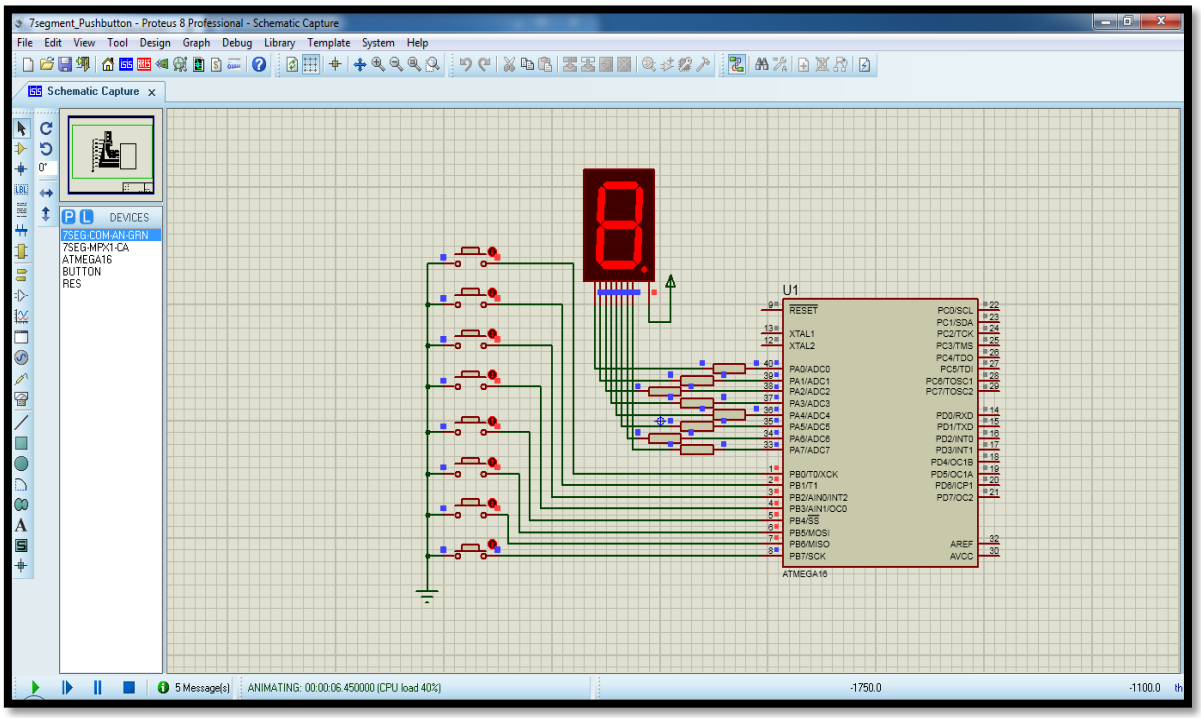
1. Laptop yang telah ter-*install* aplikasi *Proteus 8 Professional* dan *Code Vision* AVR!
2. Merancang Rangkaian Skematik seperti terlihat pada Gambar 13, menggunakan aplikasi *Proteus 8 Professional* pada laptop masing-masing!
3. Menuliskan "*Coding*" seperti terlihat pada Gambar 14 ke dalam aplikasi *Code Vision* AVR!
4. Meng-*Upload* "*Coding*" yang telah ditulis, ke dalam desain rancangan Rangkaian Skematik kalian masing-masing!
5. *Screenshoot* atau *Capture* hasilnya! seperti tampak pada Gambar 15.
6. Meng-*Upload* file (*fullscreen*!) dalam bentuk format ***.pdf** ke Google Classroom, antara lain:
   a. Rangkaian Skematik
   b. *Coding*
   c. Hasil *Capture*

Adapun *Coding* yang digunakan dalam Modul Lab 4 (kontrol *push buttons*) adalah:



Gambar 14. *Coding* pada *Code Vision AVR*

Hasil *Capture* Percobaan Lab 4 (kontrol *push buttons*), yakni:



Gambar 15. Contoh hasil percobaan Lab 4

LAB 4.

Manipulasi Data I/O (Part IV)

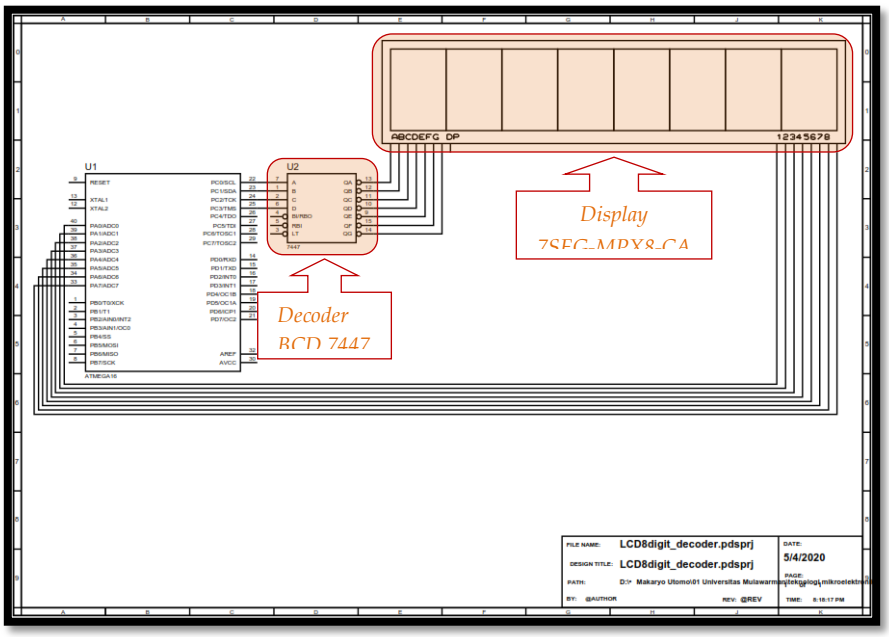
Asisten Percobaan :

NIM :

Tanggal Percobaan :

Lab 4. Manipulasi Data I/O (Part IV)

Latihan mengonversi bilangan digital ke bentuk bilangan *Binary Coded Decimal* (BCD) menggunakan 1 unit *decoder* BCD7447, dan menampilkannya ke dalam Display *seven segment* 7SEG-MPX8-CA.

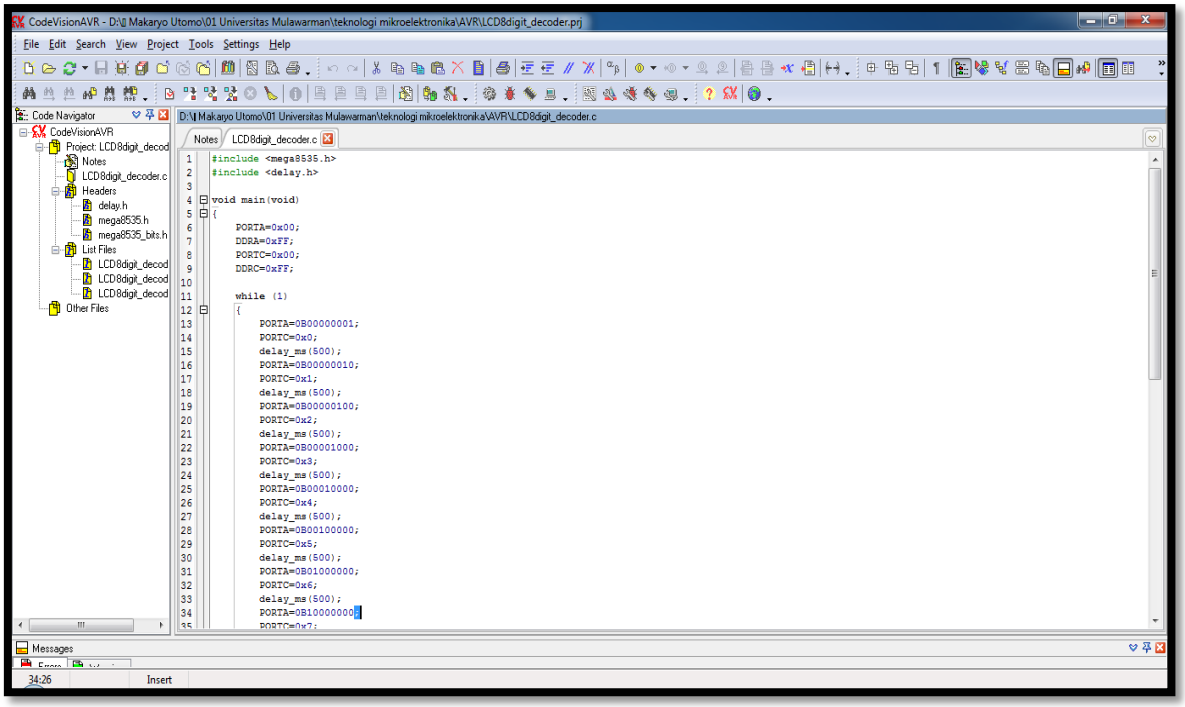


Gambar 16. Rangkaian skematik untuk menampilkan bilangan BCD

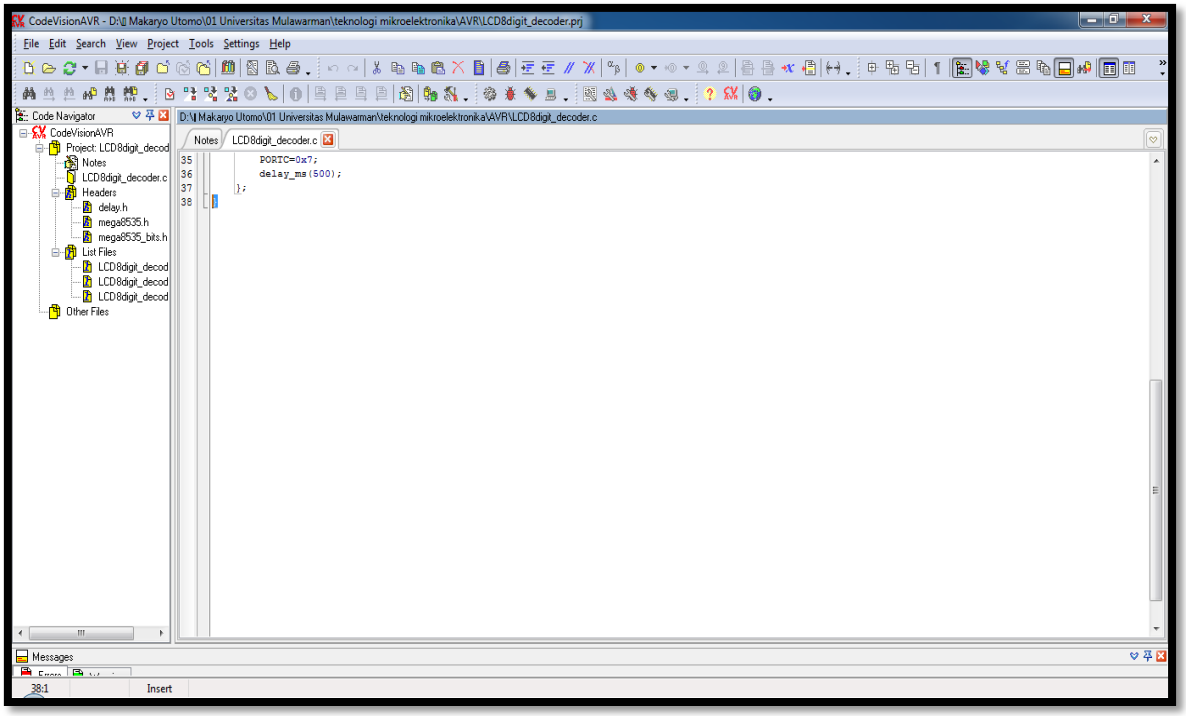
Langkah mempersiapkan Alat dan Bahan, yakni:

1. Laptop yang telah ter-*install* aplikasi *Proteus 8 Professional* dan *Code Vision* AVR!
2. Merancang Rangkaian Skematik seperti terlihat pada Gambar 16, menggunakan aplikasi *Proteus 8 Professional* pada laptop masing-masing!
3. Menuliskan "*Coding*" seperti terlihat pada Gambar 17 ke dalam aplikasi *Code Vision* AVR!
4. Meng-*Upload* "*Coding*" yang telah ditulis, ke dalam desain rancangan Rangkaian Skematik kalian masing-masing!
5. *Screenshoot* atau *Capture* hasilnya! seperti tampak pada Gambar 18.
6. Meng-*Upload* file (*fullscreen*!) dalam bentuk format ***.pdf** ke Google Classroom, antara lain:
   a. Rangkaian Skematik
   b. *Coding*
   c. Hasil *Capture*

Adapun *Coding* yang digunakan dalam Modul Lab 4 (*display* bilangan/angka) adalah:



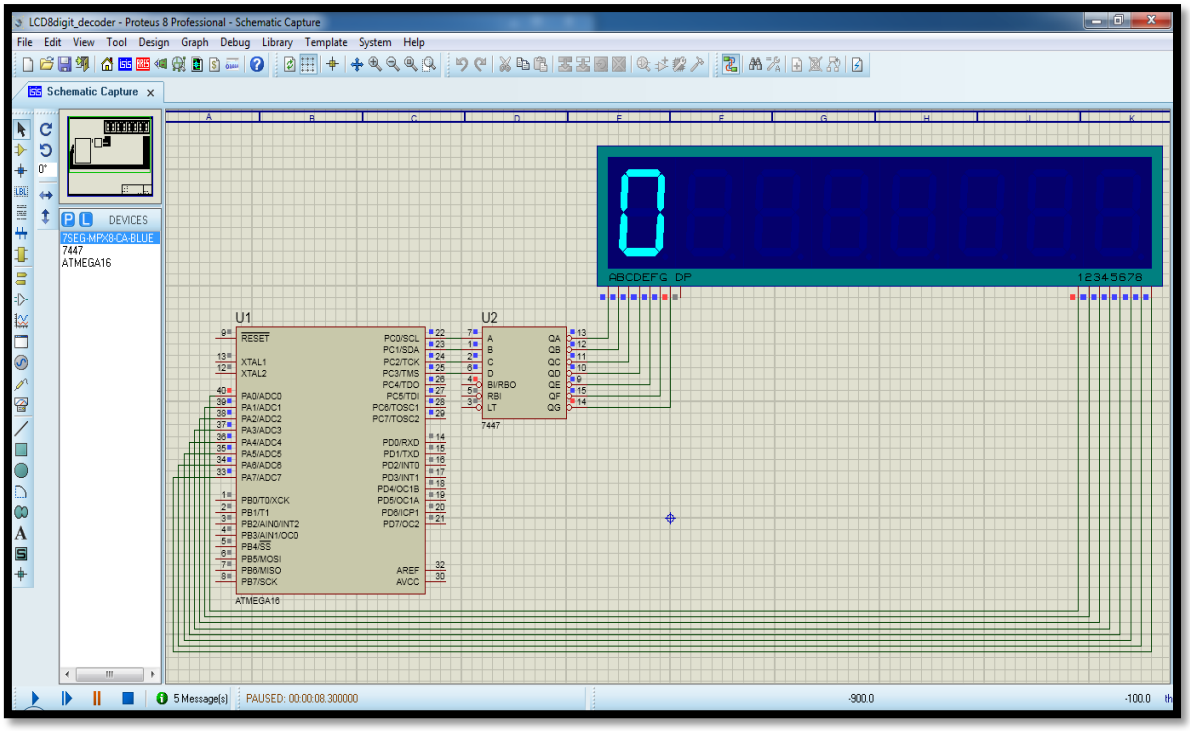
(a)



(b)

Gambar 17 (a) (b). *Coding* pada *Code Vision AVR*

Hasil *Capture* Percobaan Lab 4 (*display* bilangan/angka), yakni:



Gambar 18. Contoh hasil percobaan Lab 4

LAB 5.

Manipulasi Fitur *Pulse Width Modulation* (PWM)

Asisten Percobaan :

NIM :

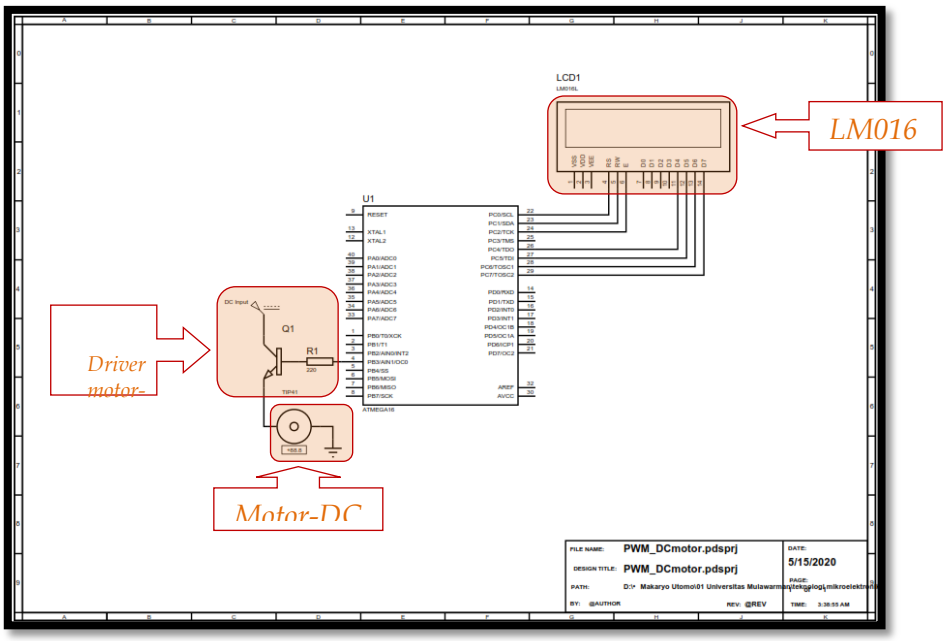
Tanggal Percobaan :

Lab 5. Manipulasi Sinyal PWM

Latihan menggunakan fitur pada mikrokontroler, khususnya PWM, untuk mengendalikan kecepatan motor-DC. Fitur PWM ini umumnya digunakan untuk mengendalikan motor-DC, Servo, intensitas terang-redup cahaya pada lampu *Light Emitting Diode* (LED), maupun aplikasi lain yang memanfaatkan *Timer*/*Counter* sebagai Pembangkit Gelombang (*Wave Generator*), yang dalam kasus ini adalah PWM.

Nilai Input PWM akan menentukan seberapa cepat putaran motor-DC. Semakin besar nilai Input PWM, maka semakin cepat pula putaran motor-DC yang digunakan, begitupun sebaliknya. Nilai Input PWM memiliki rentang antara nilai 0 s.d. 255 berdasarkan jumlah data yang digunakan, yaitu 8 bit data. Untuk mengendalikan fitur PWM, ada dua *register* yang berperan signifikan dalam mengontrol kecepatan motor-DC pada modul Lab 5 ini, antara lain:

a. TCCR0; dan
b. OCR0

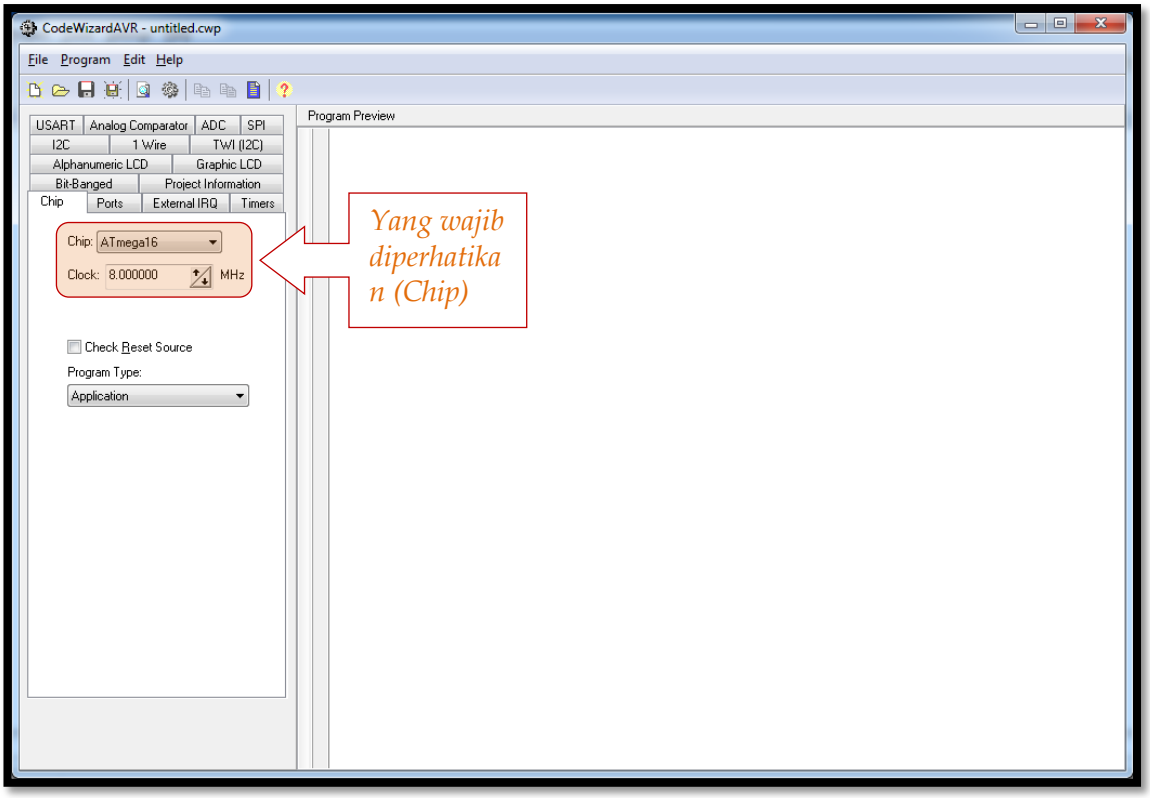


Gambar 19. Rangkaian skematik kontrol kecepatan pada motor-DC

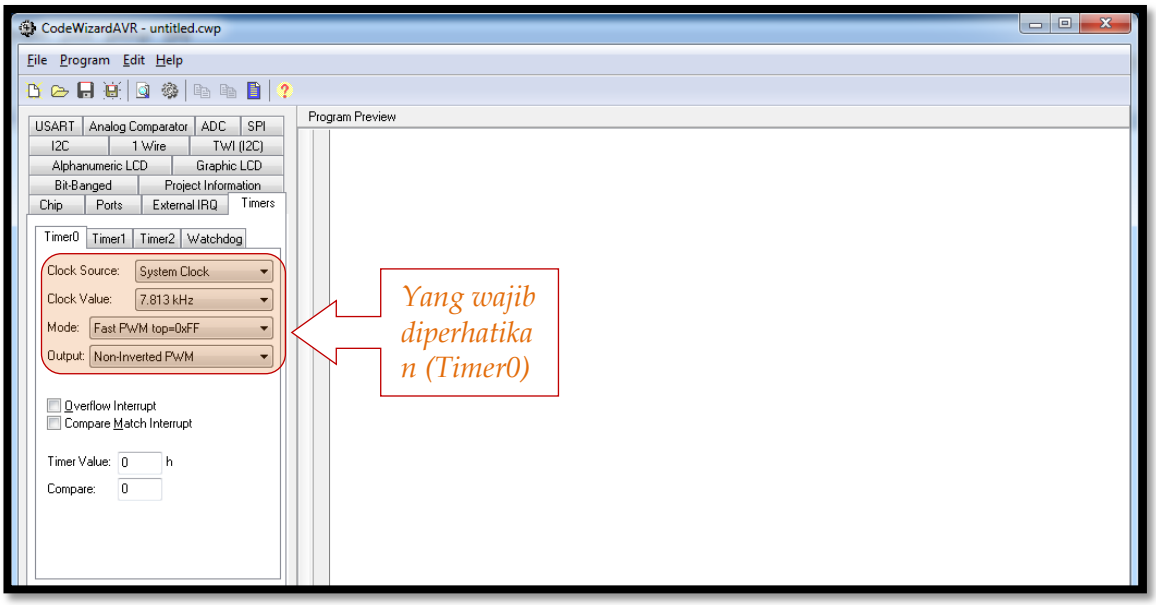
Langkah mempersiapkan Alat dan Bahan, yakni:

1. Laptop yang telah ter-*install* aplikasi *Proteus 8 Professional* dan *Code Vision* AVR!
2. Merancang Rangkaian Skematik seperti terlihat pada Gambar 19, menggunakan aplikasi *Proteus 8 Professional* pada laptop masing-masing!
3. Merancang *Project* awal menggunakan fasilitas *CodeWizardAVR*, sesuai dengan *setting* pada Gambar 20 (a), (b), dan (c)!

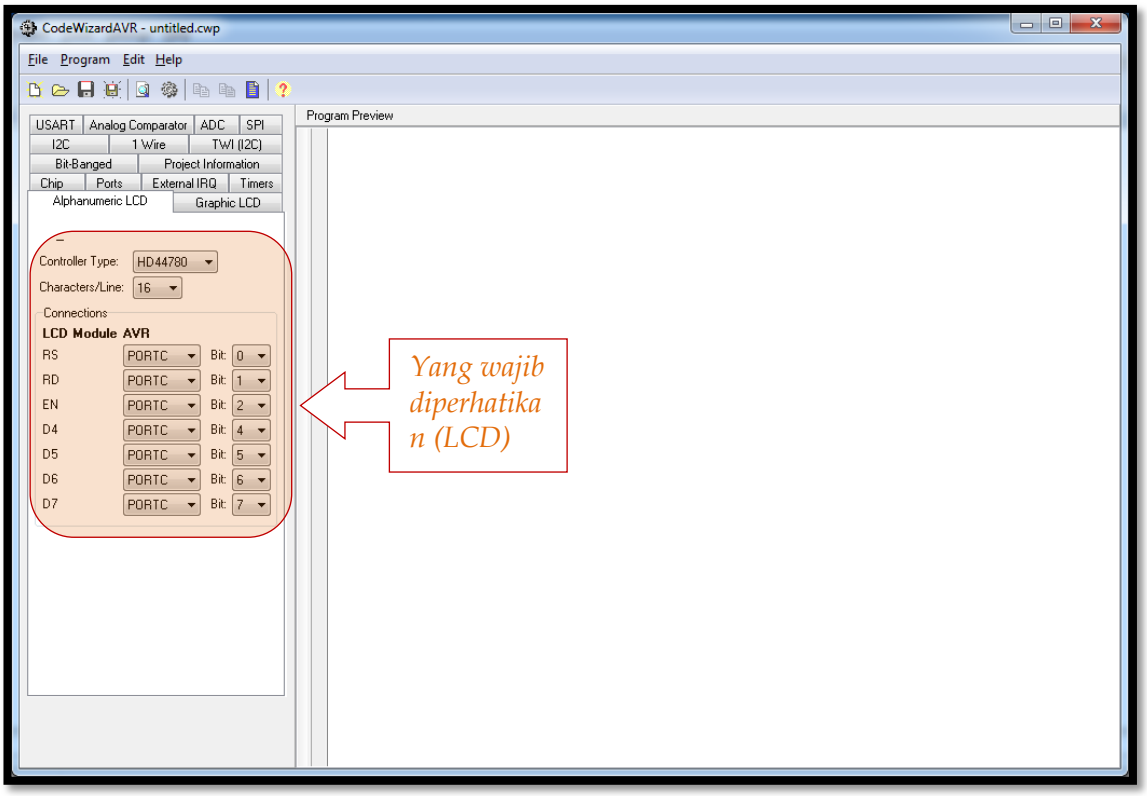
Setting konfigurasi *CodeWizardAVR*, yakni:



(a)



(b)

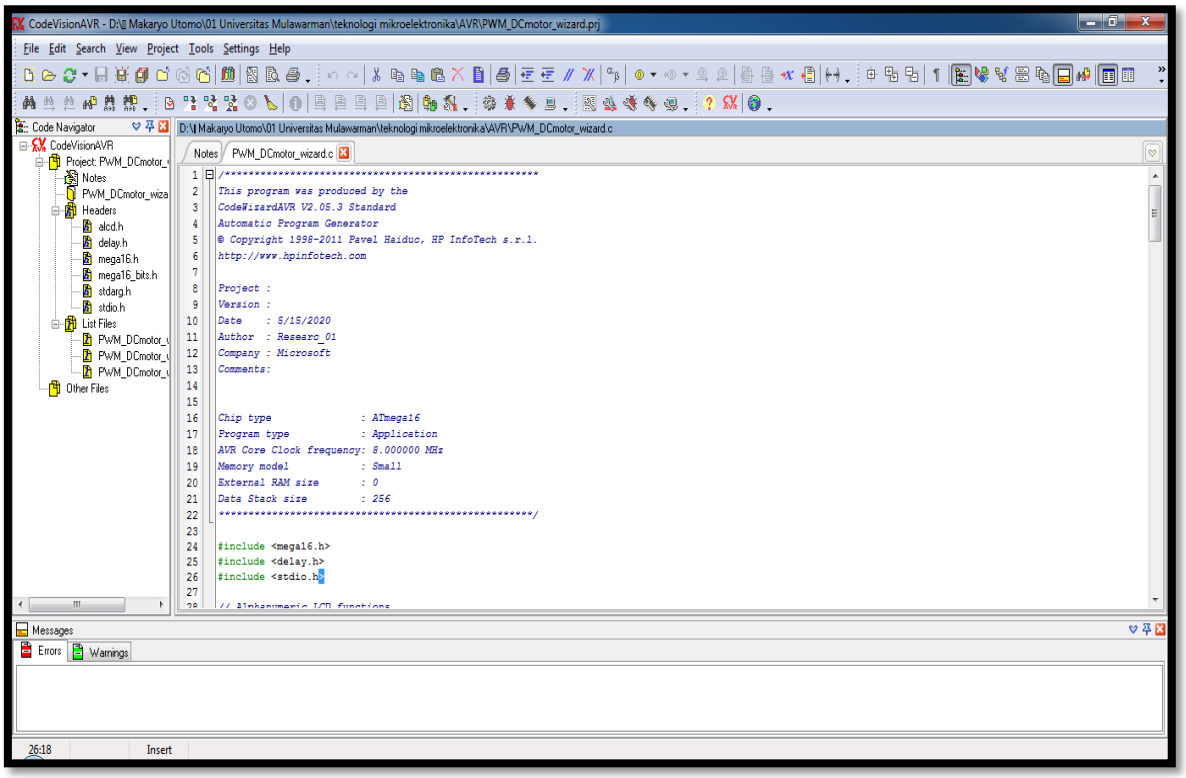


(b)

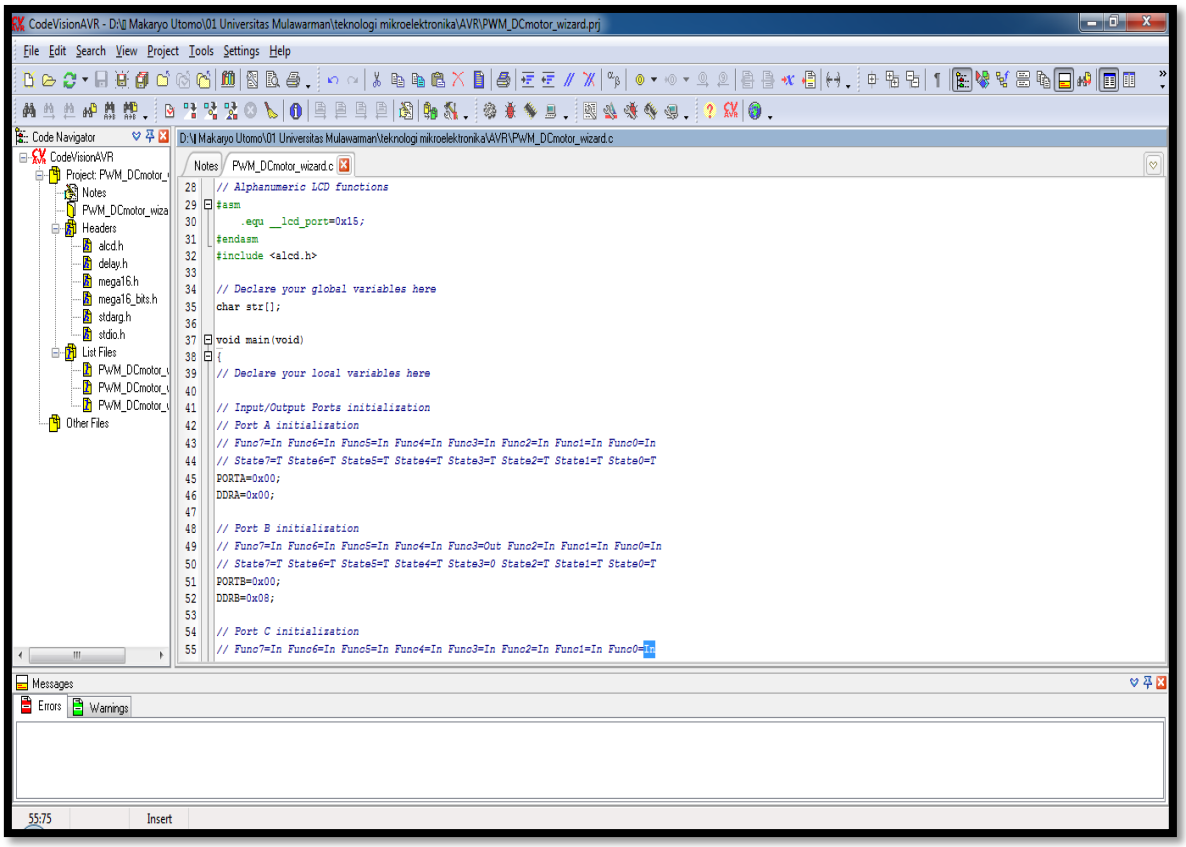
Gambar 20 (a) (b) (c). Konfigurasi *CodeWizardAVR*

4. Menuliskan "*Coding*" seperti terlihat pada Gambar 21 ke dalam aplikasi *Code Vision* AVR!
5. Meng-*Upload* "*Coding*" yang telah ditulis, ke dalam desain rancangan Rangkaian Skematik kalian masing-masing!
6. *Screenshoot* atau *Capture* hasilnya! seperti tampak pada Gambar 22.
7. Meng-*Upload* file (*fullscreen*!) dalam bentuk format ***.pdf** ke Google Classroom, antara lain:
   a. Rangkaian Skematik
   b. *Coding*
   c. Hasil *Capture*

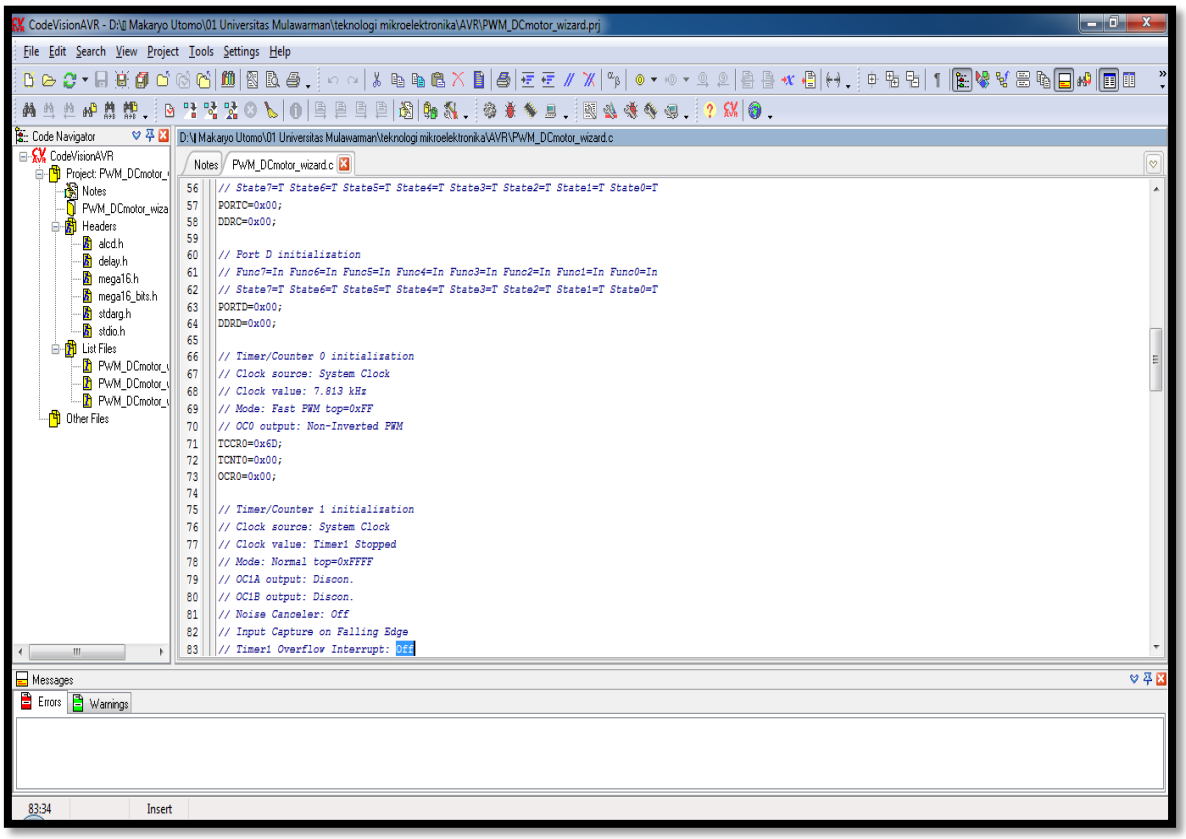
Adapun *Coding* yang digunakan dalam Modul Lab 5 (kontrol kecepatan motor-DC) adalah:



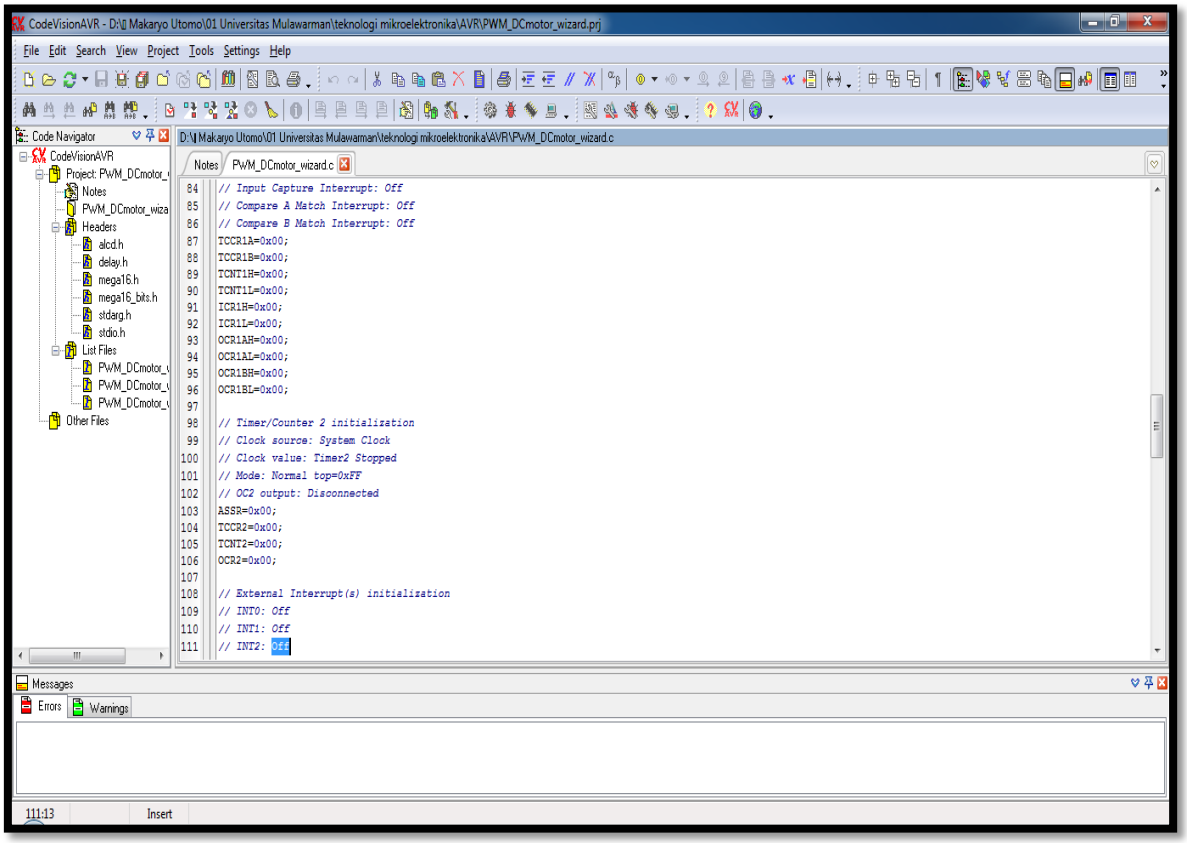
(a)



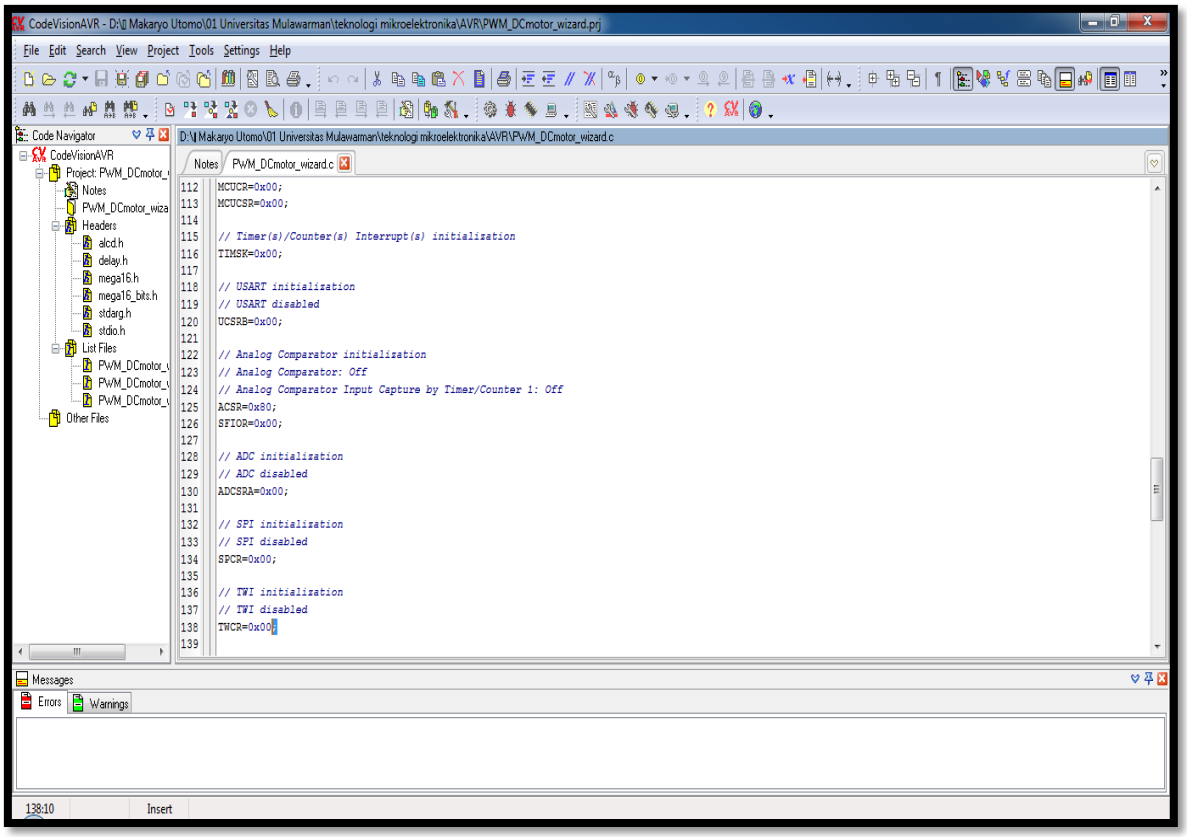
(b)


CodeVisionAVR - D:\|| Makaryo Utomo\01 Universitas Mulawarman\teknologi mikroelektronika\AVR\PWM_DCmotor_wizard.prj
File Edit Search View Project Tools Settings Help
Code Navigator
CodeVisionAVR
Project: PWM_DCmotor_
Notes
PWM_DCmotor_wiza
Headers
alcd.h
delay.h
mega16.h
mega16_bits.h
stdarg.h
stdio.h
List Files
PWM_DCmotor_
PWM_DCmotor_
PWM_DCmotor_
Other Files
D:\|| Makaryo Utomo\01 Universitas Mulawarman\teknologi mikroelektronika\AVR\PWM_DCmotor_wizard.c
Notes
PWM_DCmotor_wizard.c
56 // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
57 PORTC=0x00;
58 DDRC=0x00;
59
60 // Port D initialization
61 // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
62 // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
63 PORTD=0x00;
64 DDRD=0x00;
65
66 // Timer/Counter 0 initialization
67 // Clock source: System Clock
68 // Clock value: 7.813 kHz
69 // Mode: Fast PWM top=0xFF
70 // OC0 output: Non-Inverted PWM
71 TCCR0=0x6D;
72 TCNT0=0x00;
73 OCR0=0x00;
74
75 // Timer/Counter 1 initialization
76 // Clock source: System Clock
77 // Clock value: Timer1 Stopped
78 // Mode: Normal top=0xFFFF
79 // OC1A output: Discon.
80 // OC1B output: Discon.
81 // Noise Canceler: Off
82 // Input Capture on Falling Edge
83 // Timer1 Overflow Interrupt: Off
Messages
Errors
Warnings
83:34
Insert

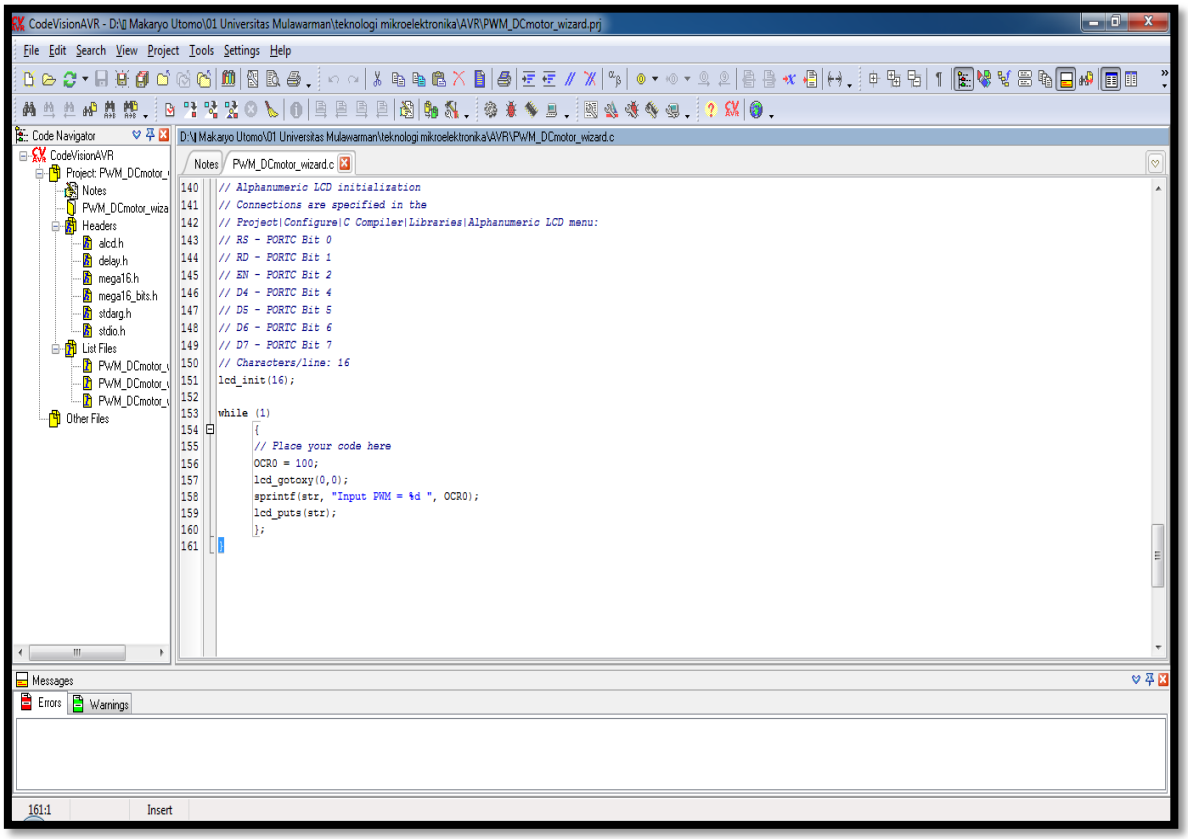
(c)


CodeVisionAVR - D:\|| Makaryo Utomo\01 Universitas Mulawarman\teknologi mikroelektronika\AVR\PWM_DCmotor_wizard.prj
File Edit Search View Project Tools Settings Help
Code Navigator
CodeVisionAVR
Project: PWM_DCmotor_
Notes
PWM_DCmotor_wiza
Headers
alcd.h
delay.h
mega16.h
mega16_bits.h
stdarg.h
stdio.h
List Files
PWM_DCmotor_
PWM_DCmotor_
PWM_DCmotor_
Other Files
D:\|| Makaryo Utomo\01 Universitas Mulawarman\teknologi mikroelektronika\AVR\PWM_DCmotor_wizard.c
Notes
PWM_DCmotor_wizard.c
84 // Input Capture Interrupt: Off
85 // Compare A Match Interrupt: Off
86 // Compare B Match Interrupt: Off
87 TCCR1A=0x00;
88 TCCR1B=0x00;
89 TCNT1H=0x00;
90 TCNT1L=0x00;
91 ICR1H=0x00;
92 ICR1L=0x00;
93 OCR1AH=0x00;
94 OCR1AL=0x00;
95 OCR1BH=0x00;
96 OCR1BL=0x00;
97
98 // Timer/Counter 2 initialization
99 // Clock source: System Clock
100 // Clock value: Timer2 Stopped
101 // Mode: Normal top=0xFF
102 // OC2 output: Disconnected
103 ASSR=0x00;
104 TCCR2=0x00;
105 TCNT2=0x00;
106 OCR2=0x00;
107
108 // External Interrupt(s) initialization
109 // INT0: Off
110 // INT1: Off
111 // INT2: Off
Messages
Errors
Warnings
111:13
Insert

(d)



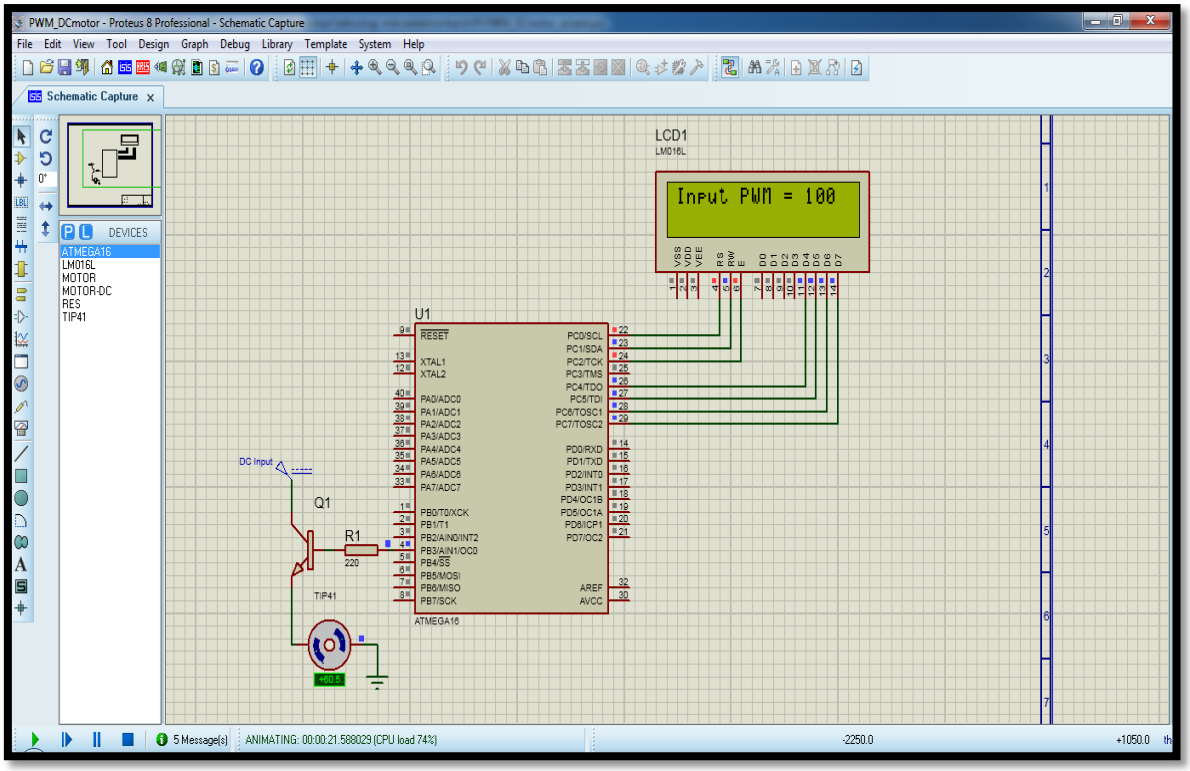
(e)



(f)

Gambar 21. (a) (b) (c) (d) (e) (f). *Coding* pada *Code Vision AVR*

Hasil *Capture* Percobaan Lab 5, yakni:



Gambar 22. Contoh hasil percobaan Lab 5

LAB 6.

Interaksi Data I/O dan *Serial Communication* (Serial Comm)

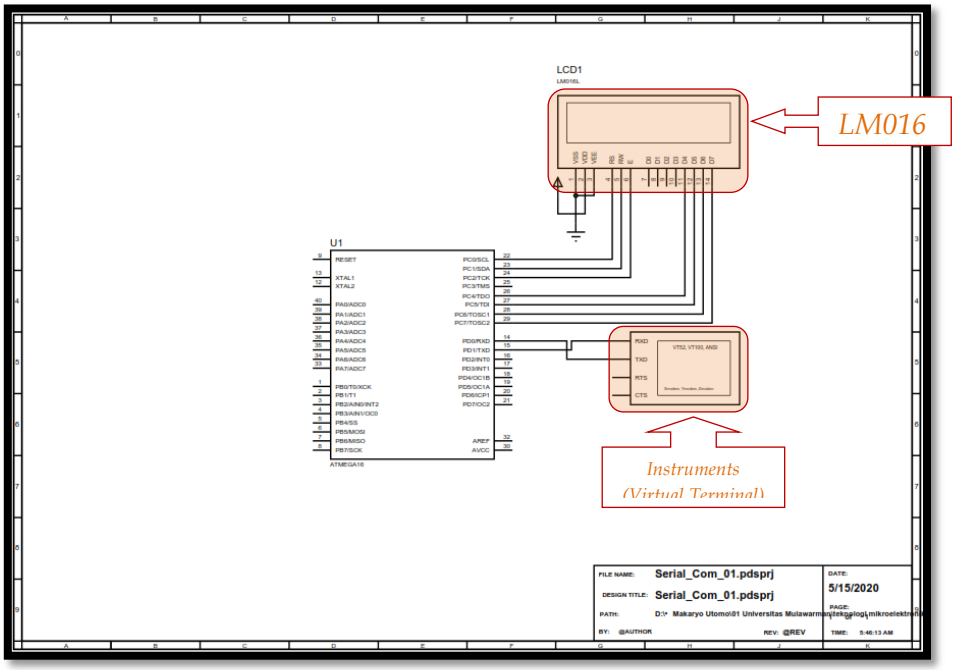
Asisten Percobaan :

NIM :

Tanggal Percobaan :

Lab 6. Komunikasi Serial Data I/O (Part IV)

Latihan berinteraksi antara operator dengan data I/O pada mikrokontroler melalui *Serial Communication* (*Serial Comm*).

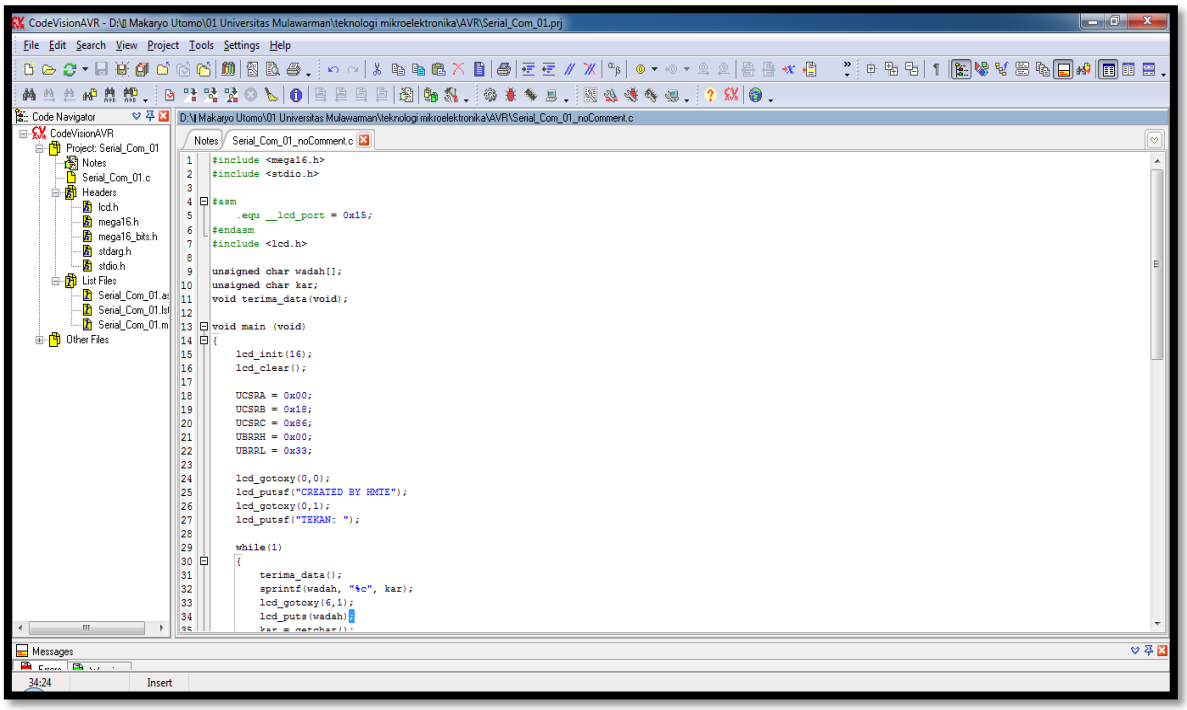


Gambar 23. Rangkaian skematik *Serial Comm*

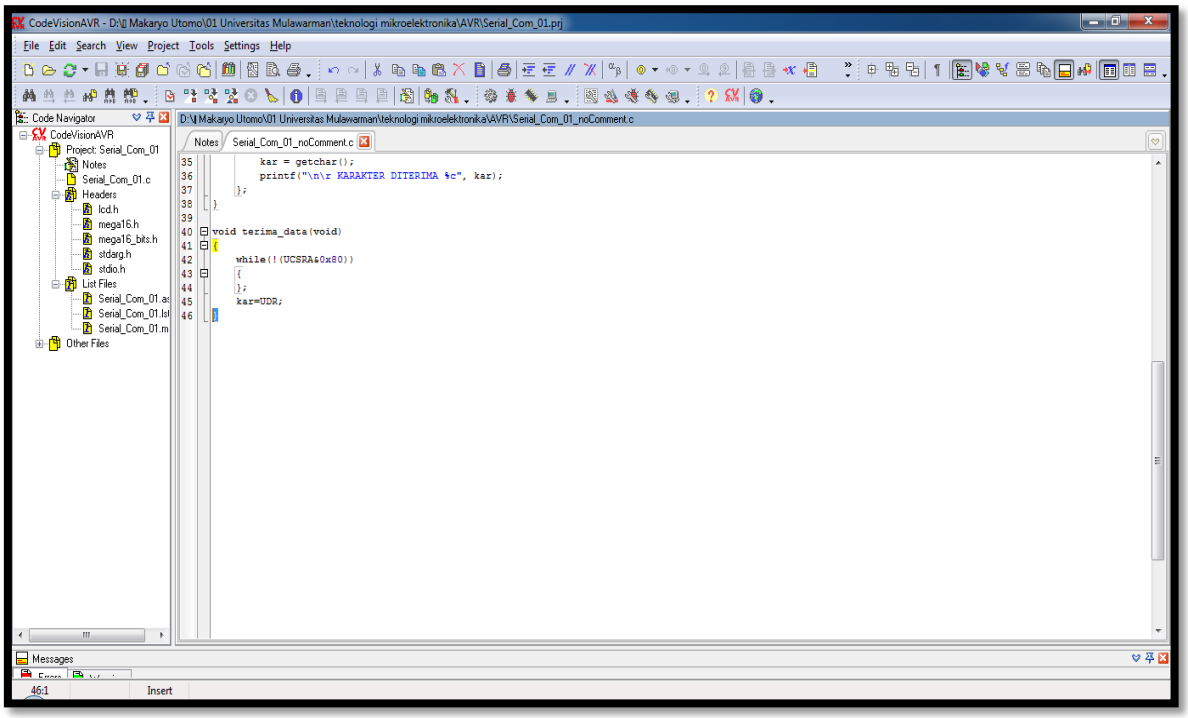
Langkah mempersiapkan Alat dan Bahan, yakni:

1. Laptop yang telah ter-*install* aplikasi *Proteus 8 Professional* dan *Code Vision* AVR!
2. Merancang Rangkaian Skematik seperti terlihat pada Gambar 23, menggunakan aplikasi *Proteus 8 Professional* pada laptop masing-masing!
3. Menuliskan "*Coding*" seperti terlihat pada Gambar 24 ke dalam aplikasi *Code Vision* AVR!
4. Meng-*Upload* "*Coding*" yang telah ditulis, ke dalam desain rancangan Rangkaian Skematik kalian masing-masing!
5. *Screenshoot* atau *Capture* hasilnya! seperti tampak pada Gambar 25.
6. Meng-*Upload* file (*fullscreen*!) dalam bentuk format ***.pdf** ke Google Classroom, antara lain:
   a. Rangkaian Skematik
   b. *Coding*
   c. Hasil *Capture*

Adapun *Coding* yang digunakan dalam Modul Lab 6 (interaksi data I/O dengan *Serial Comm*) adalah:



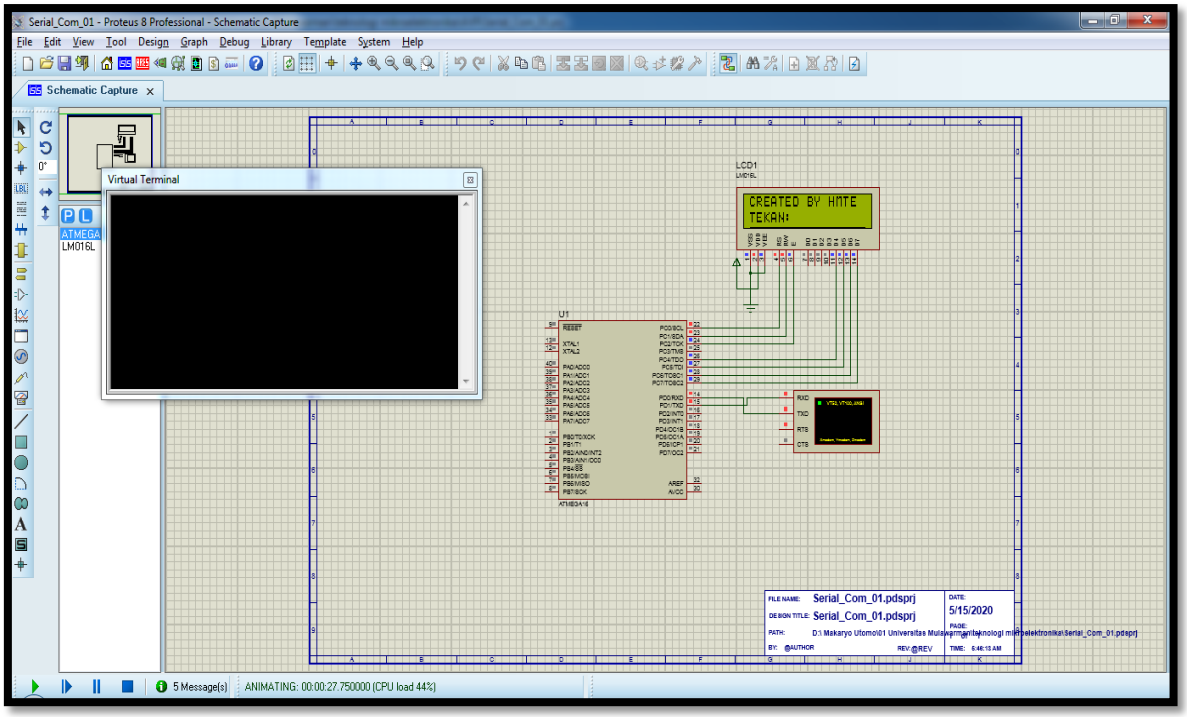
(a)



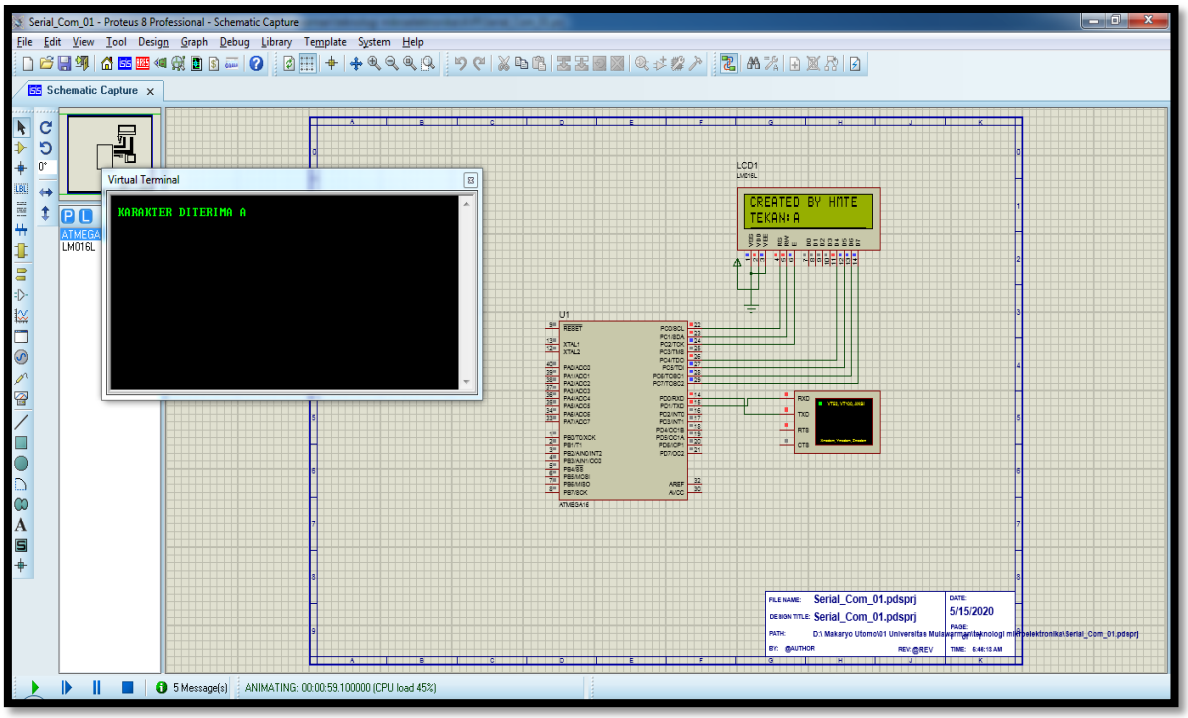
(b)

Gambar 24 (a) (b). *Coding* pada *Code Vision AVR*

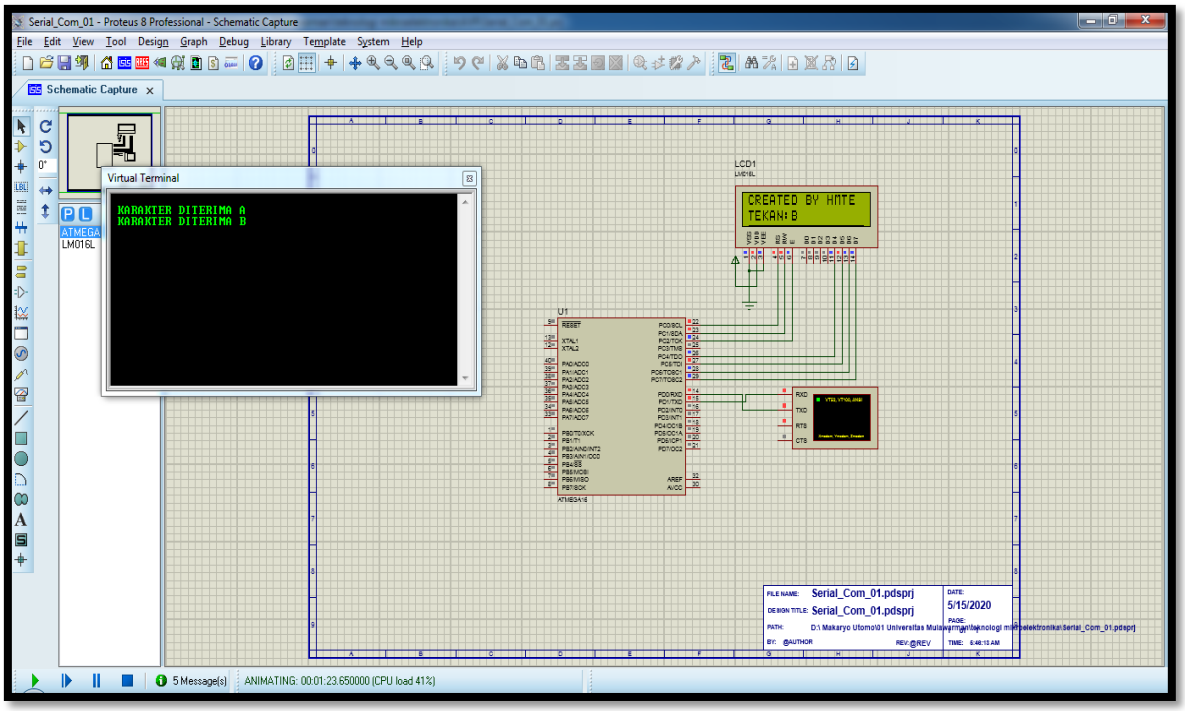
Hasil *Capture* Percobaan Lab 6, yakni:



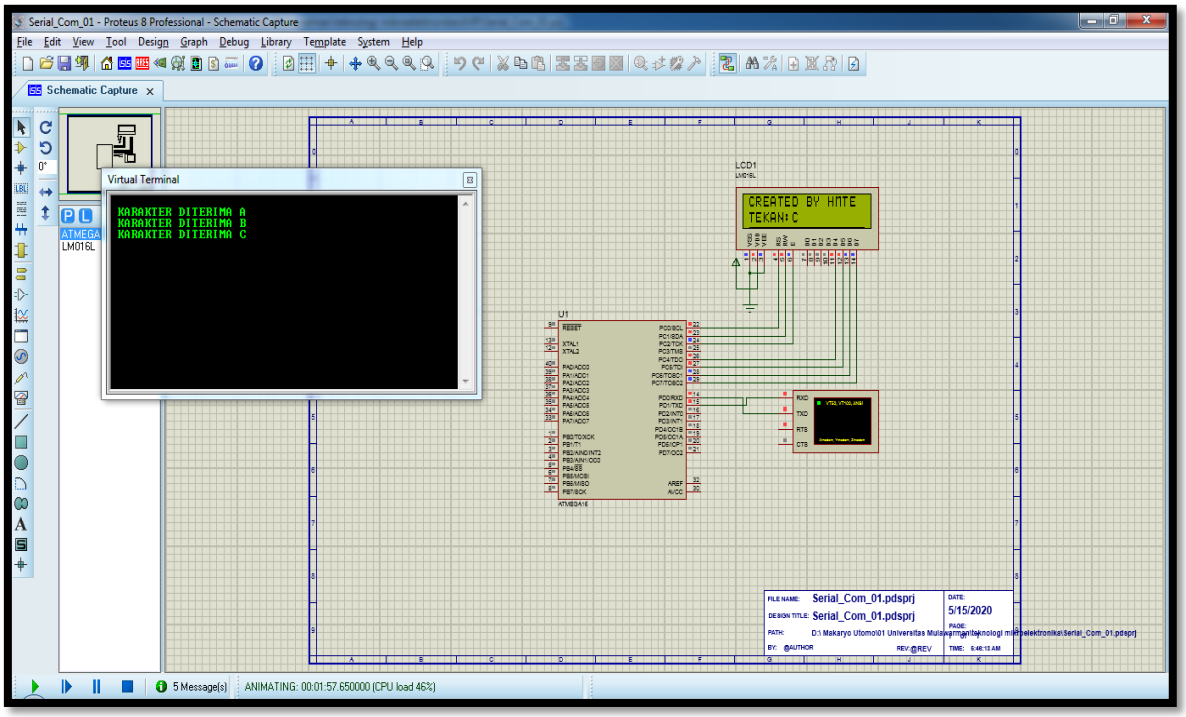
(a)



(b)



(c)



(d)

Gambar 25 (a) (b) (c) (d). Contoh hasil percobaan Lab 6

LAB 7.

Akuisis Data Menggunakan Fitur *Analog to Digital Converter* (ADC)

Asisten Percobaan :

NIM :

Tanggal Percobaan :

Lab 7. Akuisi Data dengan *Analog to Digital Converter* (ADC)

Latihan mengakuisi data menggunakan fitur *Analog to Digital Converter* (ADC) dengan memanfaatkan perubahan intensitas cahaya pada sensor *Light Dependent Resistor* (LDR). Dalam proses untuk memperoleh data analog dengan ADC ini, ada dua *register* yang berperan sangat penting dalam proses akuisisi data, antara lain:

a. ADCSRA; dan
b. ADMUX

ADC merupakan suatu piranti yg berfungsi mengonversi sinyal analog menjadi sinyal digital. Mikrokontroler AVR jenis AtMega16 yang digunakan pada Modul ini sudah *include* ADC di dalam Chip mikrokontrolernya, sehingga tidak perlu lagi membuat rangkaian ADC tambahan. Adapun pin Analog ADC terdapat pada $\boldsymbol{PORTA.0}$ hingga $\boldsymbol{PORTA.7}$.

Fitur ADC yang terdapat dalam mikrokontroler jenis AtMega16 yang digunakan dalam Modul Lab 7 ini menggunakan Resolusi 10 bit ($2^{10} = 1024$), maka hasil konversi **Nilai ADC** dapat ditentukan berdasarkan Persamaan (1) di bawah.

$$\boldsymbol{ADC = \left(\frac{V_{in} \cdot 1024}{V_{reff}}\right) - 1\ LSB}$$

Persamaan (1)

dimana, ADC adalah Nilai ADC hasil konversi, V_{in} $(Volt)$ adalah Nilai tegangan luaran dari rangkaian sensor LDR yg masuk ke pin ADC mikrokontroler, dan V_{reff} $(Volt)$ adalah tegangan referensi yang digunakan mikrokontroler.

Dari Persamaan (1), diperoleh,
apabila tegangan referensi $V_{reff} = 5\,Volt$ dan tegangan input minimal yakni $V_{in} = \mathbf{0}\,\boldsymbol{Volt}$, maka:

$ADC = \mathbf{0},$; dan,

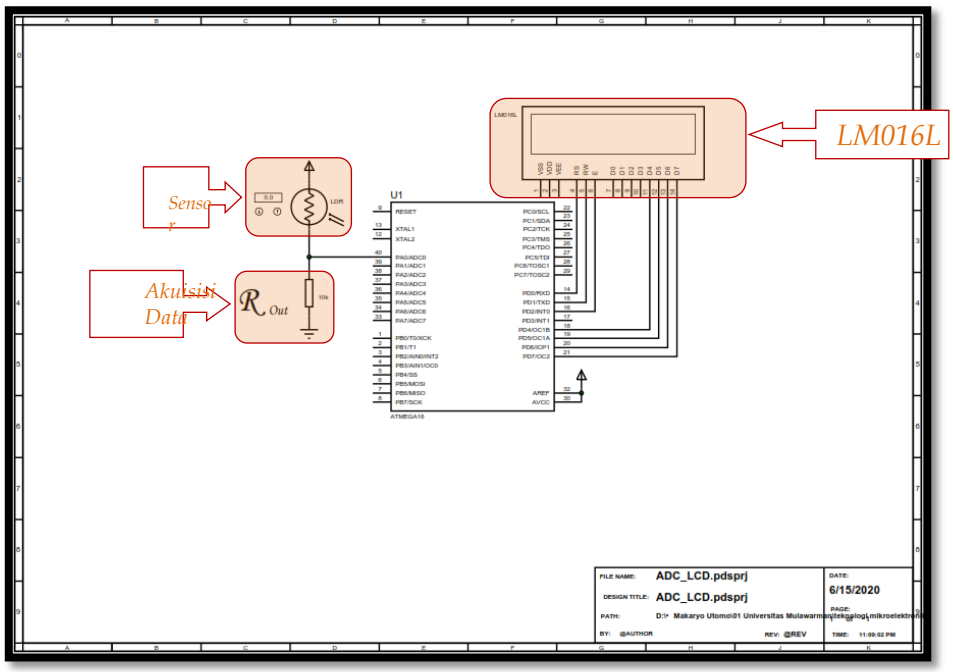
apabila tegangan referensi $V_{reff} = 5\,Volt$ dan tegangan input maksimal yakni $V_{in} = \mathbf{5}\,\boldsymbol{Volt}$, maka:

$$ADC = \left(\frac{V_{in} \cdot 1024}{V_{reff}}\right) - 1\ LSB = \left(\frac{5 \cdot 1024}{5}\right) - 1\ LSB = \mathbf{1023}.$$

Dari Persamaan (1), juga dapat dihitung besarnya nilai kenaikan yang diperoleh setiap perubahan 1 bit ADC, adalah sebesar $V_{in} = 4{,}88\ miliVolt$. Adapun *clock input* ADC yg digunakan dalam Modul Lab 7 ini adalah 62.500 Hz (62,5 kHz).

Light Dependent Resistor (LDR) / *Photo-resistor* / *Photo-conduction* / *Photo-cell* adalah komponen yang hambatannya berubah-ubah tergantung cahaya yang diterima dipermukaan sensor.

Umumnya LDR berkaki dua atau tiga, namun untuk percobaan Modul Lab 7 ini menggunakan jenis dua kaki, dan tidak memiliki polaritas sehingga dipasang bolak-balik sama saja, dimana sensor terhubung pada ***Pin*. 0** pada ***PORTA*** mikrokontroler AtMega16. Satuan untuk intensitas cahaya yg ditunjukkan pada LDR adalah *Illumination* (*Lux*).



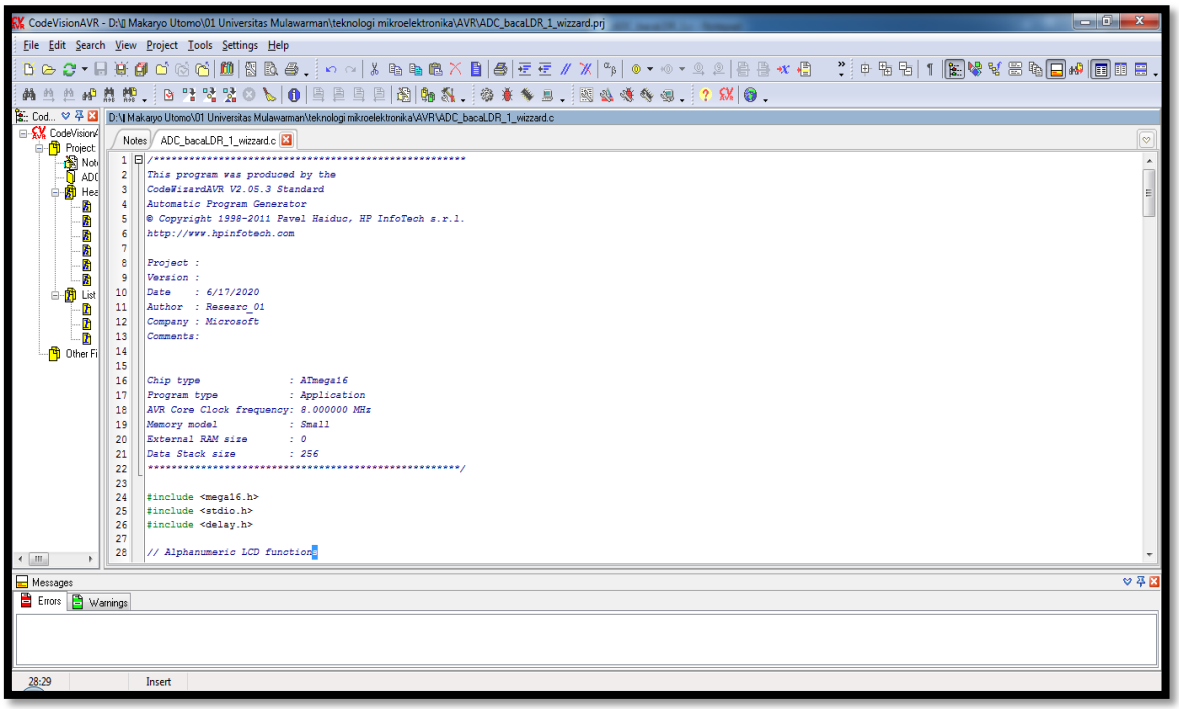
Gambar 26. Rangkaian skematik Akuisisi Data

Langkah mempersiapkan Alat dan Bahan, yakni:

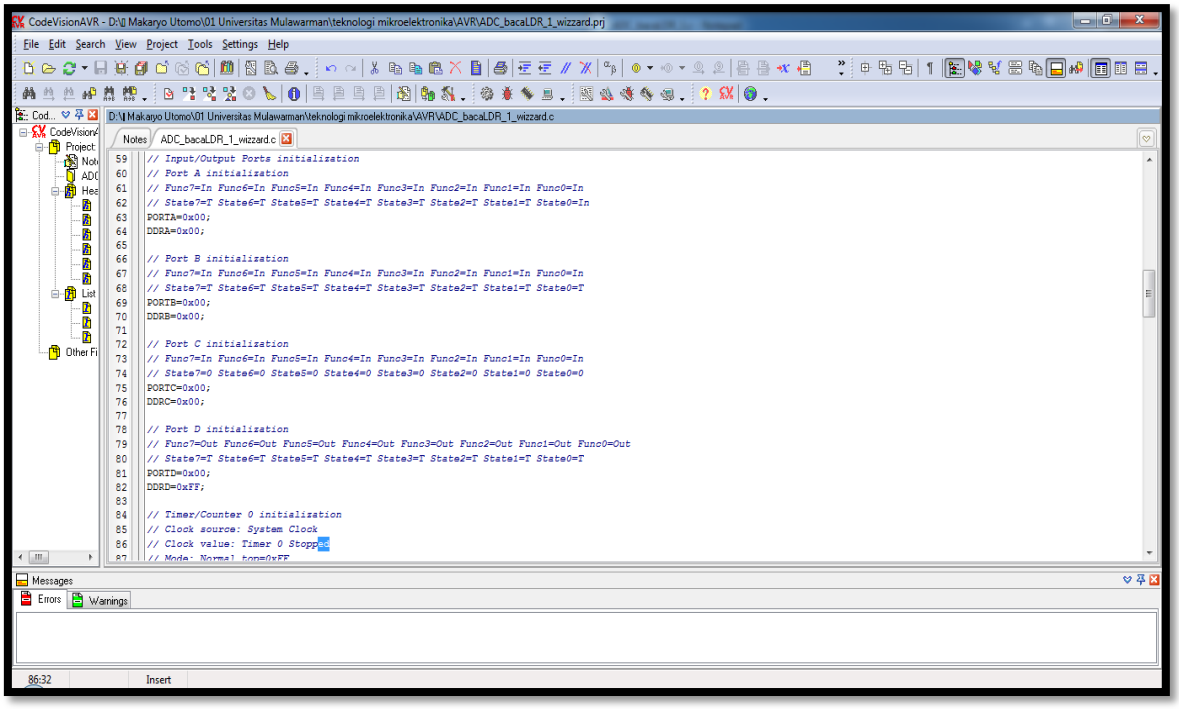
1. Laptop yang telah ter-*install* aplikasi *Proteus 8 Professional* dan *Code Vision* AVR!
2. Merancang Rangkaian Skematik seperti terlihat pada Gambar 26, menggunakan aplikasi *Proteus 8 Professional* pada laptop masing-masing!
3. Menuliskan "*Coding*" seperti terlihat pada Gambar 27 ke dalam aplikasi *Code Vision* AVR!
4. Meng-*Upload* "*Coding*" yang telah ditulis, ke dalam desain rancangan Rangkaian Skematik kalian masing-masing!
5. *Screenshoot* atau *Capture* hasilnya! seperti tampak pada Gambar 28.

6. Meng-*Upload* file (*fullscreen*!) dalam bentuk format ***.pdf** ke Google Classroom, antara lain:
    a. Rangkaian Skematik
    b. *Coding*
    c. Hasil *Capture*

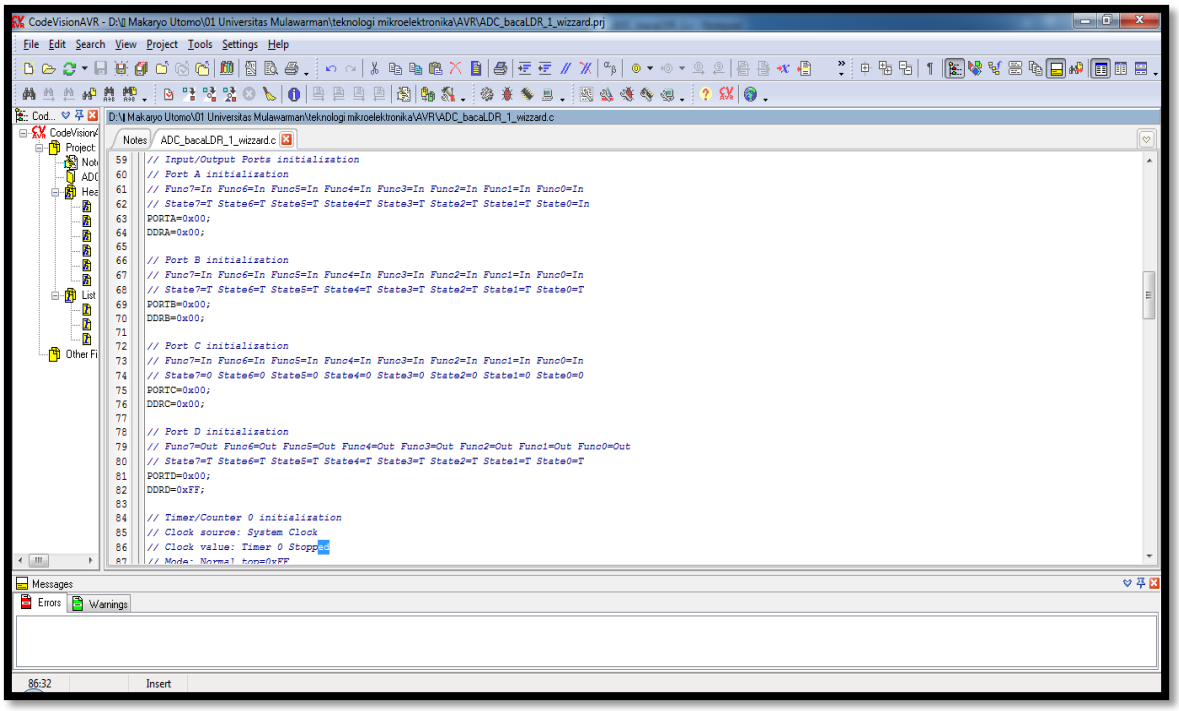
Adapun *Coding* yang digunakan dalam Modul Lab 7 (akusisi data menggunakan ADC) adalah:



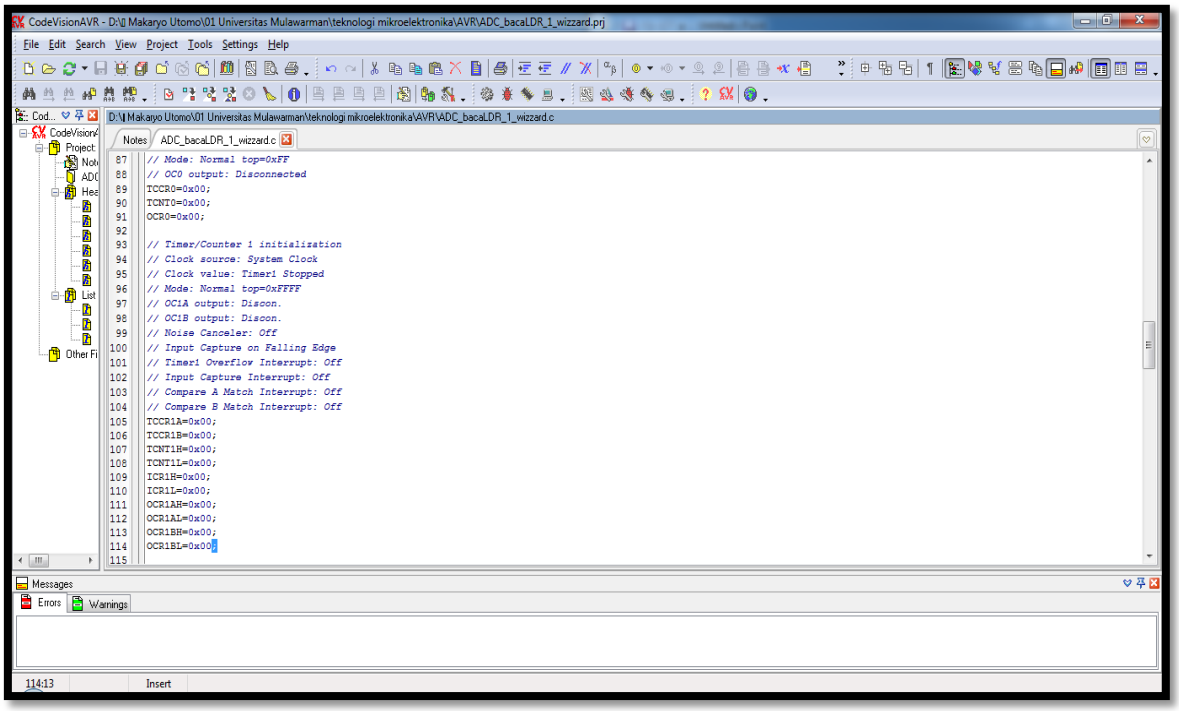
(a)



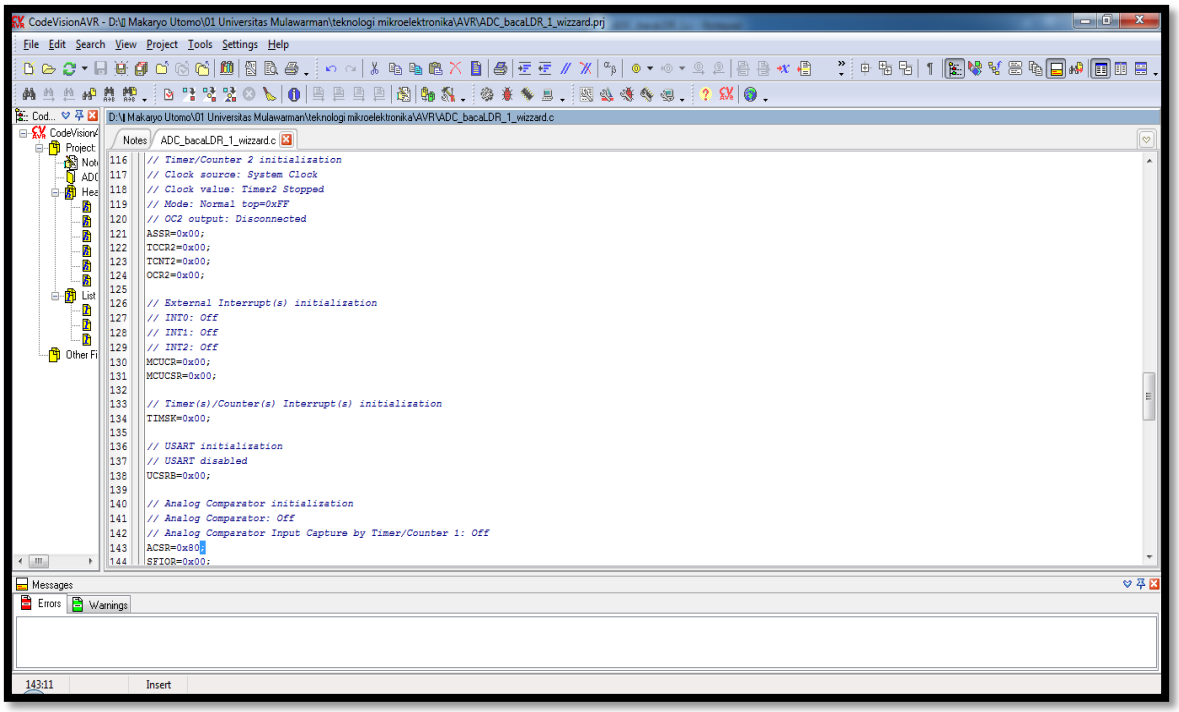
(b)


CodeVisionAVR - D:\[] Makaryo Utomo\01 Universitas Mulawarman\teknologi mikroelektronika\AVR\ADC_bacaLDR_1_wizzard.prj
File Edit Search View Project Tools Settings Help
D:\[] Makaryo Utomo\01 Universitas Mulawarman\teknologi mikroelektronika\AVR\ADC_bacaLDR_1_wizzard.c
Notes
ADC_bacaLDR_1_wizzard.c
// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=In
PORTA=0x00;
DDRA=0x00;
// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0x00;
// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTC=0x00;
DDRC=0x00;
// Port D initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0xFF;
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
Messages
Errors
Warnings
86:32
Insert

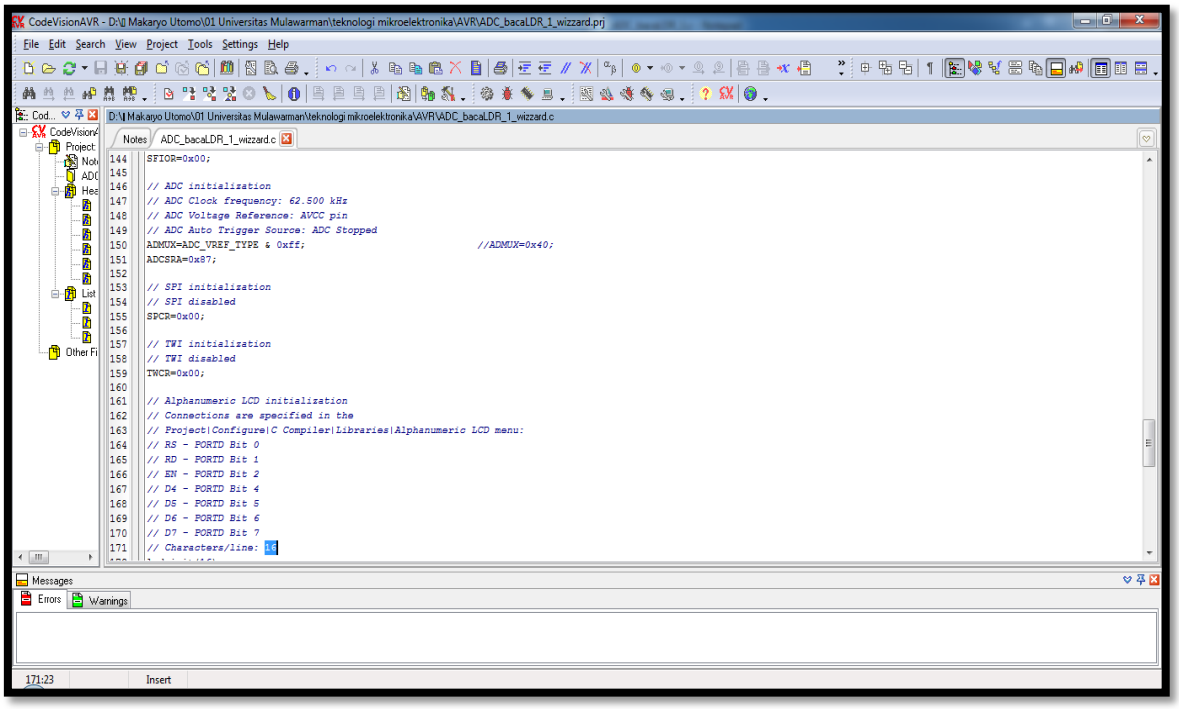
(c)


CodeVisionAVR - D:\[] Makaryo Utomo\01 Universitas Mulawarman\teknologi mikroelektronika\AVR\ADC_bacaLDR_1_wizzard.prj
File Edit Search View Project Tools Settings Help
D:\[] Makaryo Utomo\01 Universitas Mulawarman\teknologi mikroelektronika\AVR\ADC_bacaLDR_1_wizzard.c
Notes
ADC_bacaLDR_1_wizzard.c
// Mode: Normal top=0xFF
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
Messages
Errors
Warnings
114:13
Insert

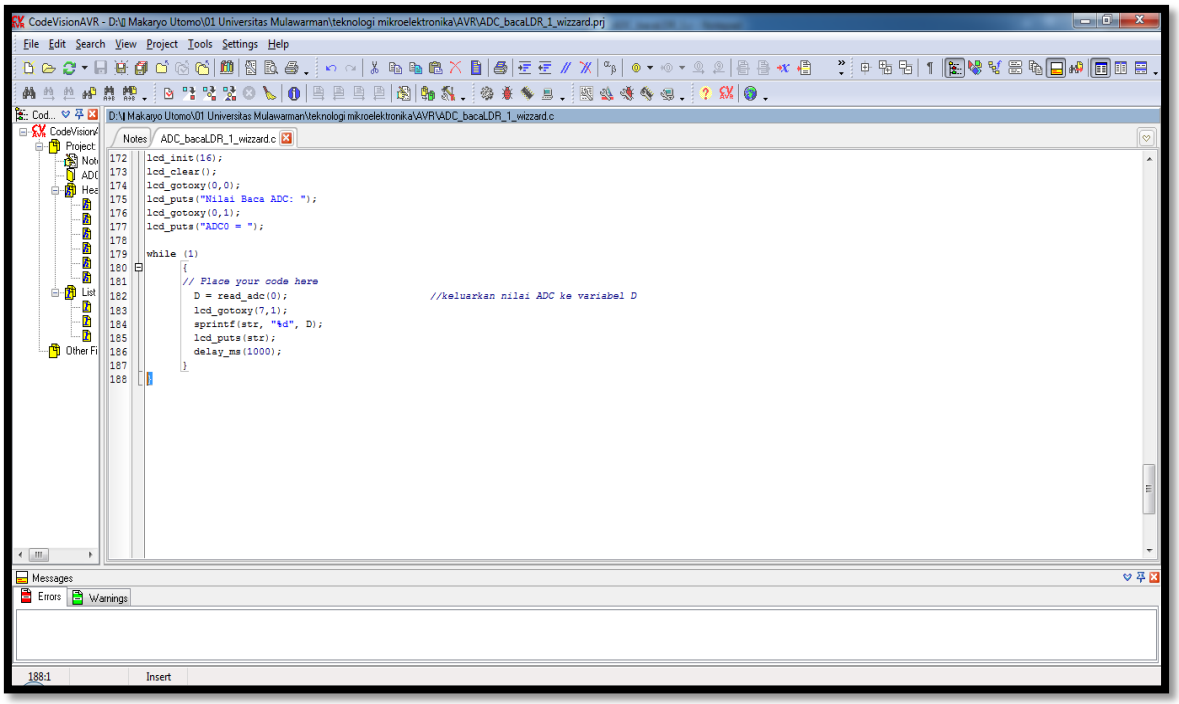
(d)


CodeVisionAVR - D:\[] Makaryo Utomo\01 Universitas Mulawarman\teknologi mikroelektronika\AVR\ADC_bacaLDR_1_wizzard.prj
File Edit Search View Project Tools Settings Help
D:\[] Makaryo Utomo\01 Universitas Mulawarman\teknologi mikroelektronika\AVR\ADC_bacaLDR_1_wizzard.c
Notes
ADC_bacaLDR_1_wizzard.c
116 // Timer/Counter 2 initialization
117 // Clock source: System Clock
118 // Clock value: Timer2 Stopped
119 // Mode: Normal top=0xFF
120 // OC2 output: Disconnected
121 ASSR=0x00;
122 TCCR2=0x00;
123 TCNT2=0x00;
124 OCR2=0x00;
125
126 // External Interrupt(s) initialization
127 // INT0: Off
128 // INT1: Off
129 // INT2: Off
130 MCUCR=0x00;
131 MCUCSR=0x00;
132
133 // Timer(s)/Counter(s) Interrupt(s) initialization
134 TIMSK=0x00;
135
136 // USART initialization
137 // USART disabled
138 UCSRB=0x00;
139
140 // Analog Comparator initialization
141 // Analog Comparator: Off
142 // Analog Comparator Input Capture by Timer/Counter 1: Off
143 ACSR=0x80;
144 SFIOR=0x00;
Messages
Errors
Warnings
143:11
Insert

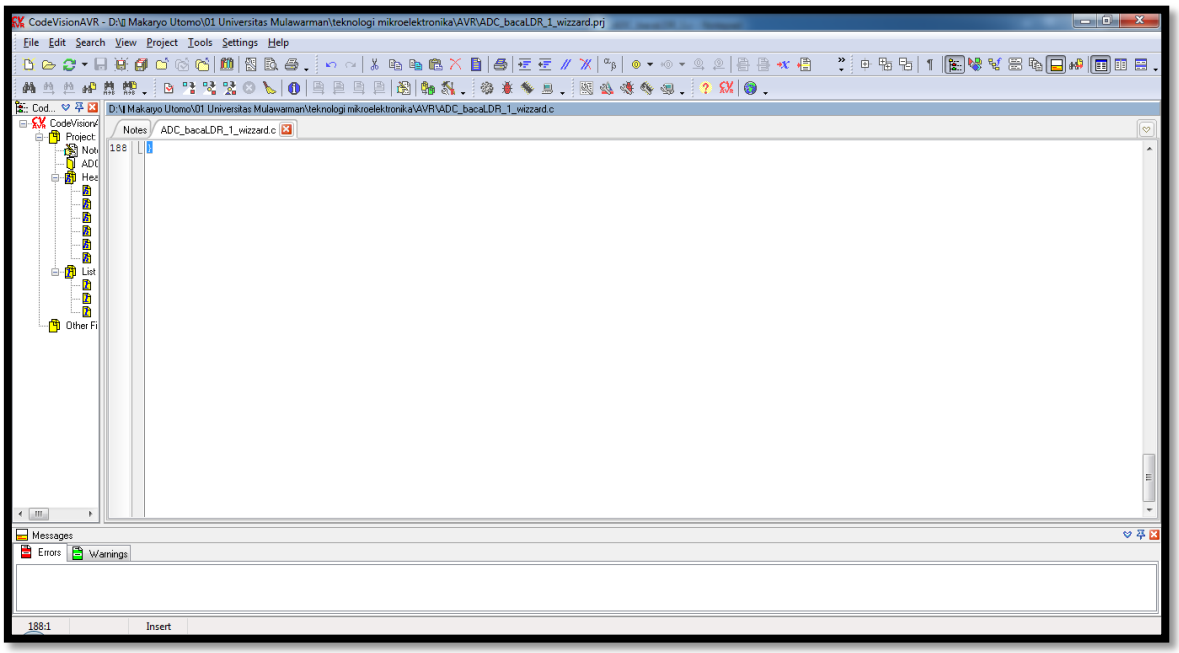
(e)


CodeVisionAVR - D:\[] Makaryo Utomo\01 Universitas Mulawarman\teknologi mikroelektronika\AVR\ADC_bacaLDR_1_wizzard.prj
File Edit Search View Project Tools Settings Help
D:\[] Makaryo Utomo\01 Universitas Mulawarman\teknologi mikroelektronika\AVR\ADC_bacaLDR_1_wizzard.c
Notes
ADC_bacaLDR_1_wizzard.c
144 SFIOR=0x00;
145
146 // ADC initialization
147 // ADC Clock frequency: 62.500 kHz
148 // ADC Voltage Reference: AVCC pin
149 // ADC Auto Trigger Source: ADC Stopped
150 ADMUX=ADC_VREF_TYPE & 0xff; //ADMUX=0x40;
151 ADCSRA=0x87;
152
153 // SPI initialization
154 // SPI disabled
155 SPCR=0x00;
156
157 // TWI initialization
158 // TWI disabled
159 TWCR=0x00;
160
161 // Alphanumeric LCD initialization
162 // Connections are specified in the
163 // Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
164 // RS - PORTD Bit 0
165 // RD - PORTD Bit 1
166 // EN - PORTD Bit 2
167 // D4 - PORTD Bit 4
168 // D5 - PORTD Bit 5
169 // D6 - PORTD Bit 6
170 // D7 - PORTD Bit 7
171 // Characters/line: 16
Messages
Errors
Warnings
171:23
Insert

(f)



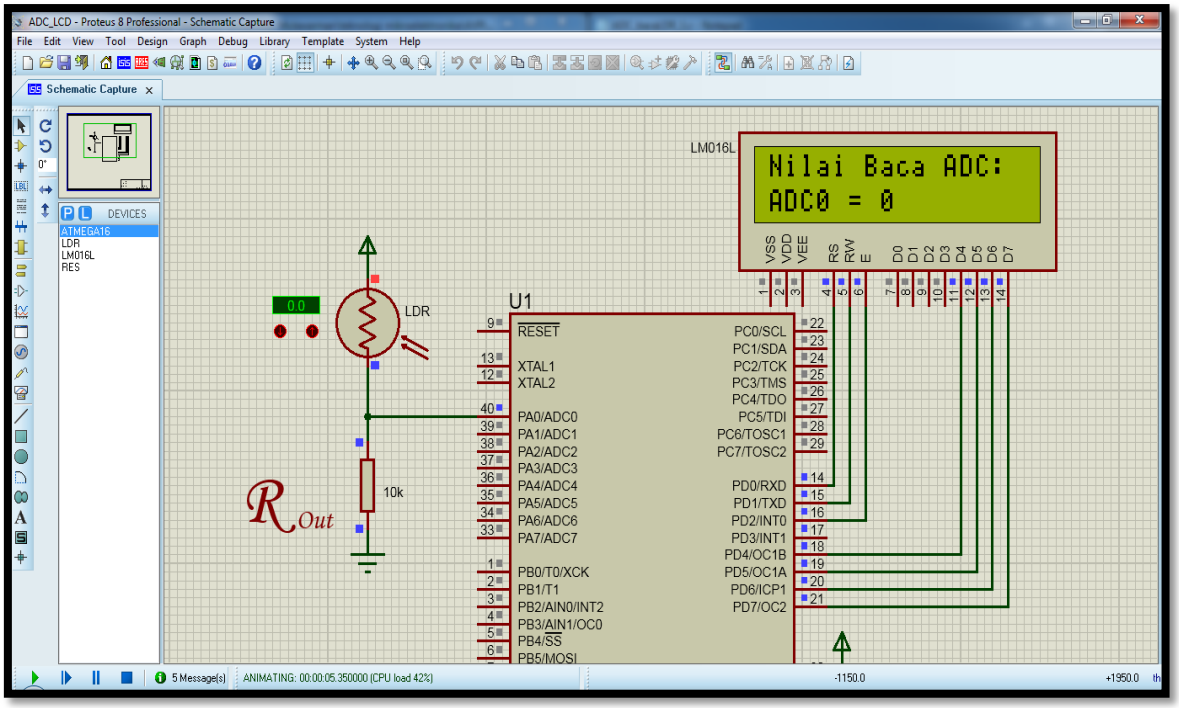
(g)



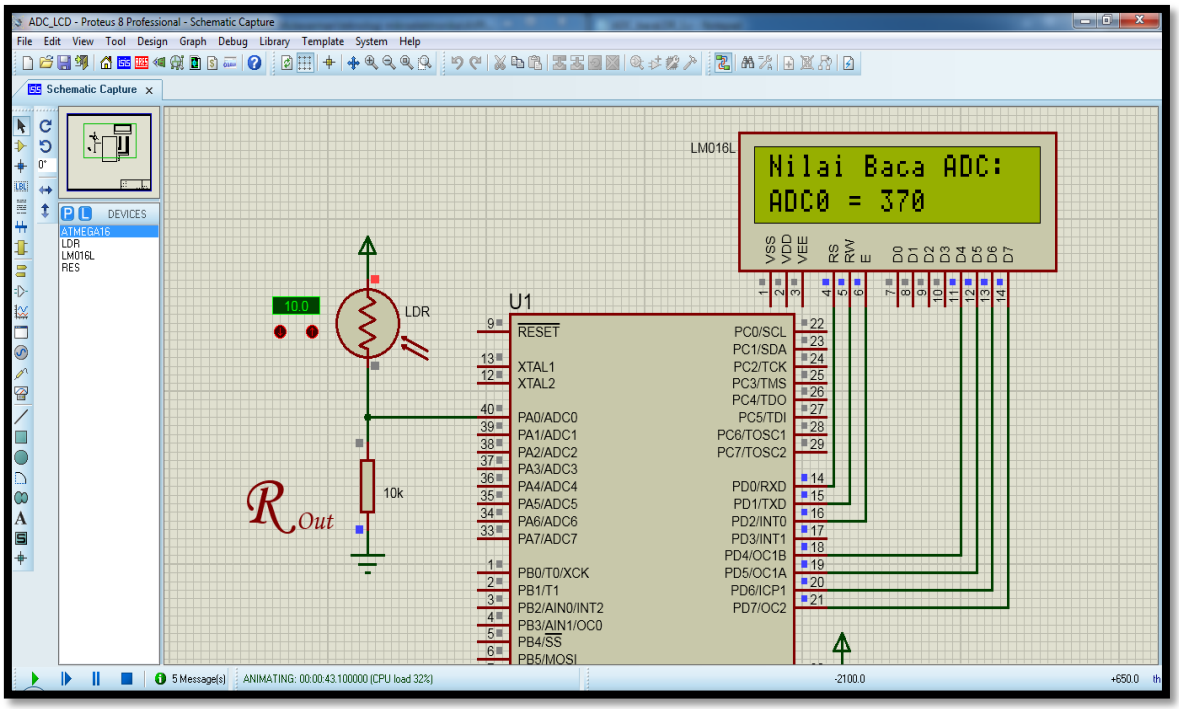
(h)

Gambar 27 (a) (b) (c) (d) (e) (f) (g) (h). *Coding* pada *Code Vision AVR*

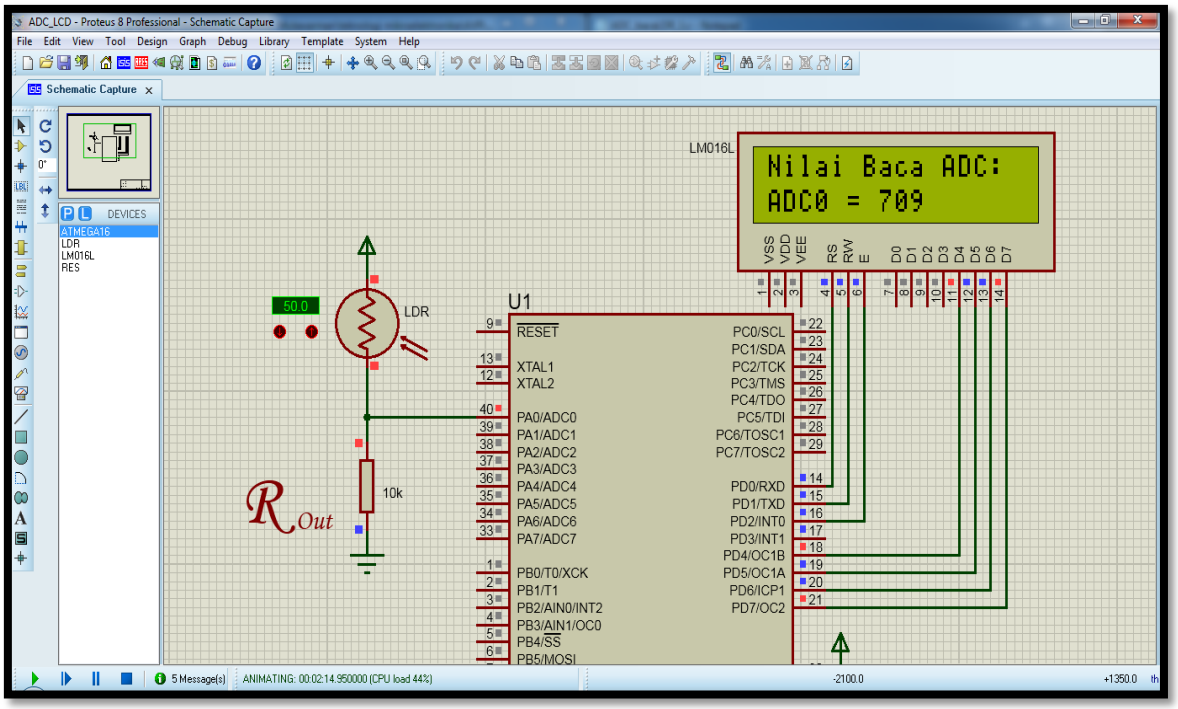
Hasil *Capture* Percobaan Lab 7, yakni:



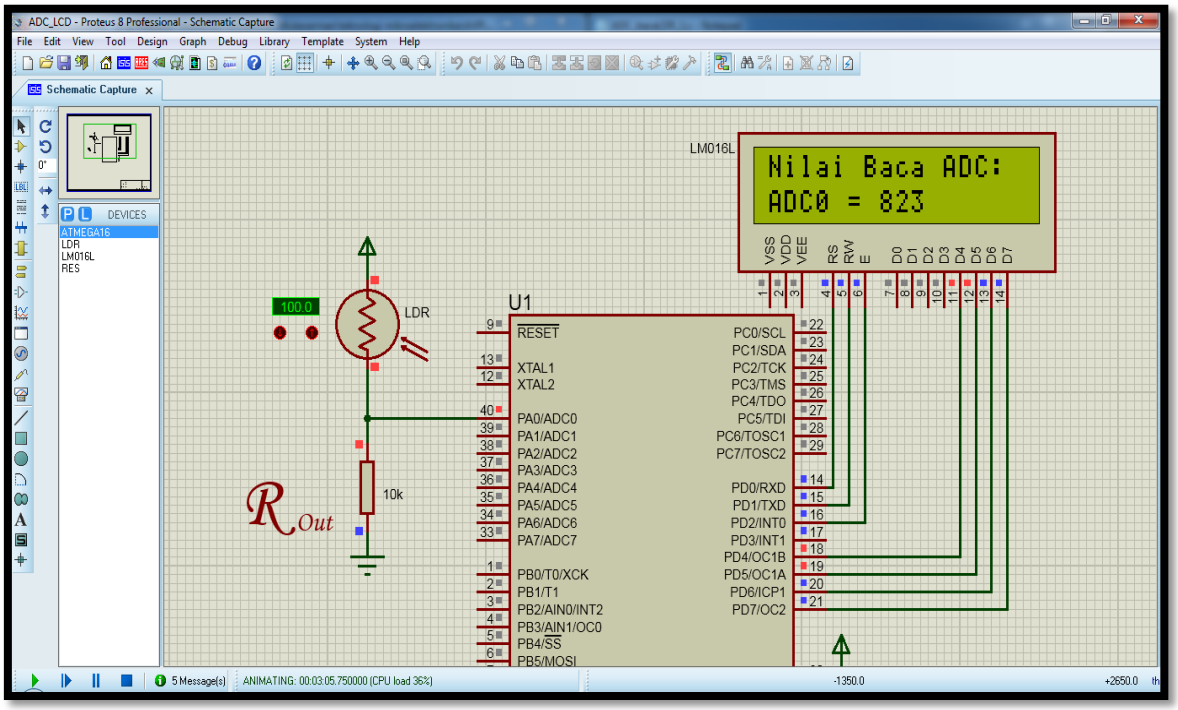
(a)



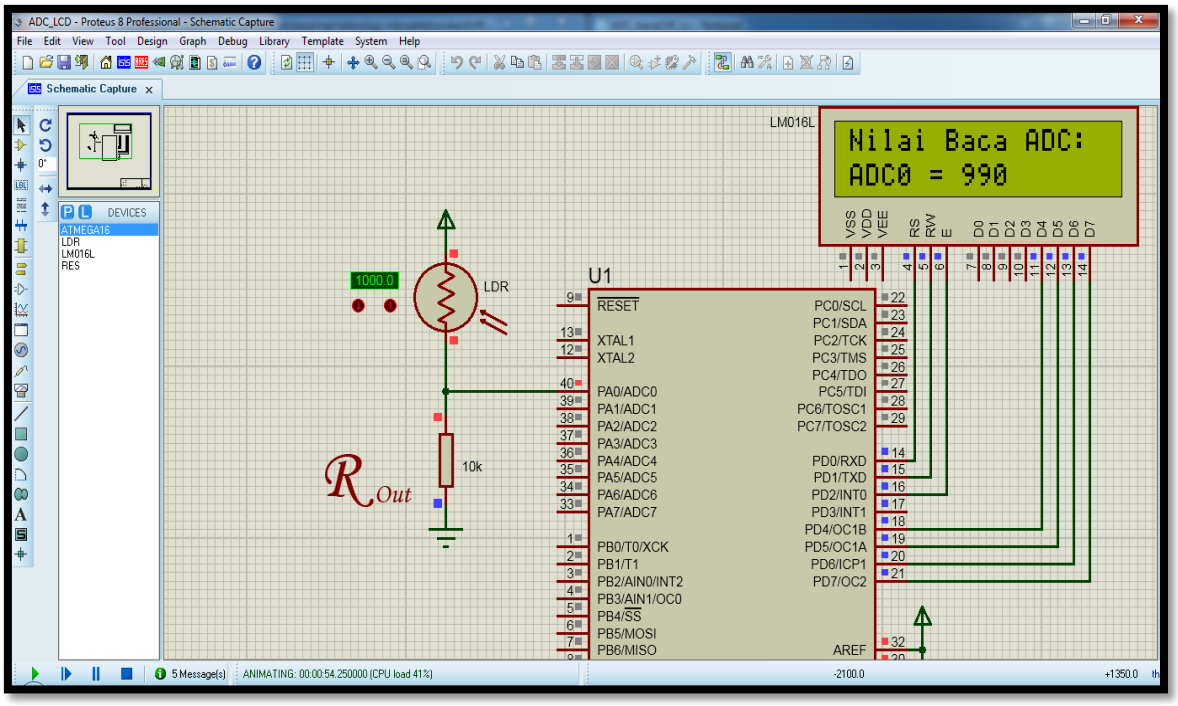
(b)


ADC_LCD - Proteus 8 Professional - Schematic Capture
Nilai Baca ADC:
ADC0 = 709
LM016L
U1
LDR
50.0
10k
ROut
RESET
XTAL1
XTAL2
PA0/ADC0
PA1/ADC1
PA2/ADC2
PA3/ADC3
PA4/ADC4
PA5/ADC5
PA6/ADC6
PA7/ADC7
PB0/T0/XCK
PB1/T1
PB2/AIN0/INT2
PB3/AIN1/OC0
PB4/SS
PB5/MOSI
PC0/SCL
PC1/SDA
PC2/TCK
PC3/TMS
PC4/TDO
PC5/TDI
PC6/TOSC1
PC7/TOSC2
PD0/RXD
PD1/TXD
PD2/INT0
PD3/INT1
PD4/OC1B
PD5/OC1A
PD6/ICP1
PD7/OC2
VSS
VDD
VEE
RS
RW
E
D0
D1
D2
D3
D4
D5
D6
D7
DEVICES
ATMEGA16
LDR
LM016L
RES
5 Message(s)
ANIMATING: 00:02:14.950000 (CPU load 44%)
-2100.0
+1350.0

(c)


ADC_LCD - Proteus 8 Professional - Schematic Capture
Nilai Baca ADC:
ADC0 = 823
LM016L
U1
LDR
100.0
10k
ROut
RESET
XTAL1
XTAL2
PA0/ADC0
PA1/ADC1
PA2/ADC2
PA3/ADC3
PA4/ADC4
PA5/ADC5
PA6/ADC6
PA7/ADC7
PB0/T0/XCK
PB1/T1
PB2/AIN0/INT2
PB3/AIN1/OC0
PB4/SS
PB5/MOSI
PC0/SCL
PC1/SDA
PC2/TCK
PC3/TMS
PC4/TDO
PC5/TDI
PC6/TOSC1
PC7/TOSC2
PD0/RXD
PD1/TXD
PD2/INT0
PD3/INT1
PD4/OC1B
PD5/OC1A
PD6/ICP1
PD7/OC2
VSS
VDD
VEE
RS
RW
E
D0
D1
D2
D3
D4
D5
D6
D7
DEVICES
ATMEGA16
LDR
LM016L
RES
5 Message(s)
ANIMATING: 00:03:05.750000 (CPU load 36%)
-1350.0
+2650.0

(d)



(e)

Gambar 28 (a) (b) (c) (d) (e). Hasil percobaan Lab 7

LAB 8.

Interaksi I/O antara Respon Akuisisi Data ADC terhadap Kendali Sinyal PWM

Asisten Percobaan :

NIM :

Tanggal Percobaan :

Lab 8. Kontrol sinyal PWM menggunakan Data Input ADC

Latihan mengendalikan sinyal *Pulse Width Modulation* (PWM) yang berupa gelombang kotak , menggunakan kumpulan data yang telah diperoleh dari fitur *Analog to Digital Converter* (ADC). Prinsip kerja dari Modul Lab 8 ini adalah data analog yang diperoleh dan dikumpulkan dari perubahan nilai resistor (sebagai masukan), kemudian digunakan untuk mengendalikan bentuk gelombang pada sinyal PWM (sebagai luaran). Untuk memperoleh data dengan fitur ADC, ada dua *register* yang berperan sangat penting dalam proses Akusisi Data pada Modul Lab 8, antara lain:

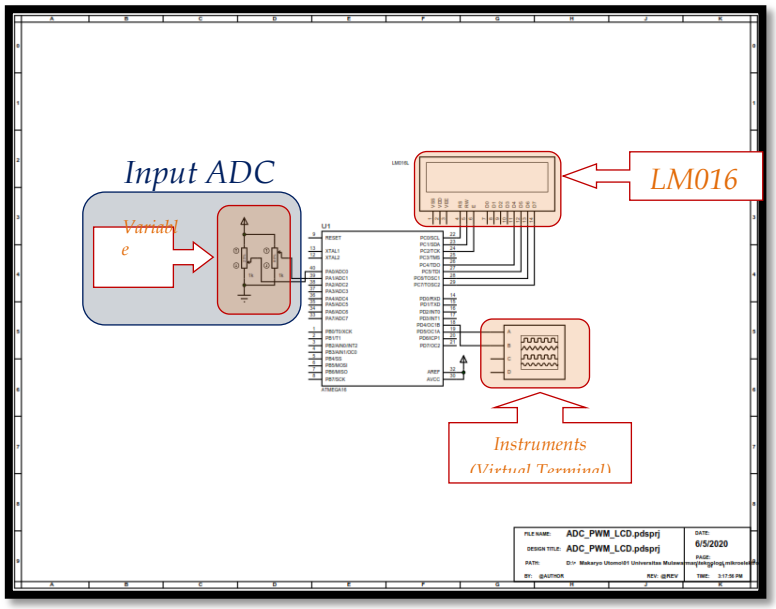
a. ADCSRA; dan,
b. ADMUX

Selain itu, untuk mengendalikan sinyal PWM, ada empat *register* yang berperan sangat penting dalam mengendalikan perubahan sinyal PWM pada percobaan Modul Lab 8 ini, antara lain:

a. TCCR1A;
b. TCCR1B;
c. OCR1AL; dan,
c. OCR1BL

Indikator perubahan sinyal PWM dapat dicari dengan menganalisa perubahan nilai dan siklus *duty cycle* pada gelombang kotak sinyal PWM, seperti tampak pada Gambar 30 (b).

Luaran (output) dari sinyal PWM ini dapat diterapkan untuk mengontrol motor-DC, motor *Servo*, terang-redupnya cahaya LED, dan aplikasi yang lainnya.

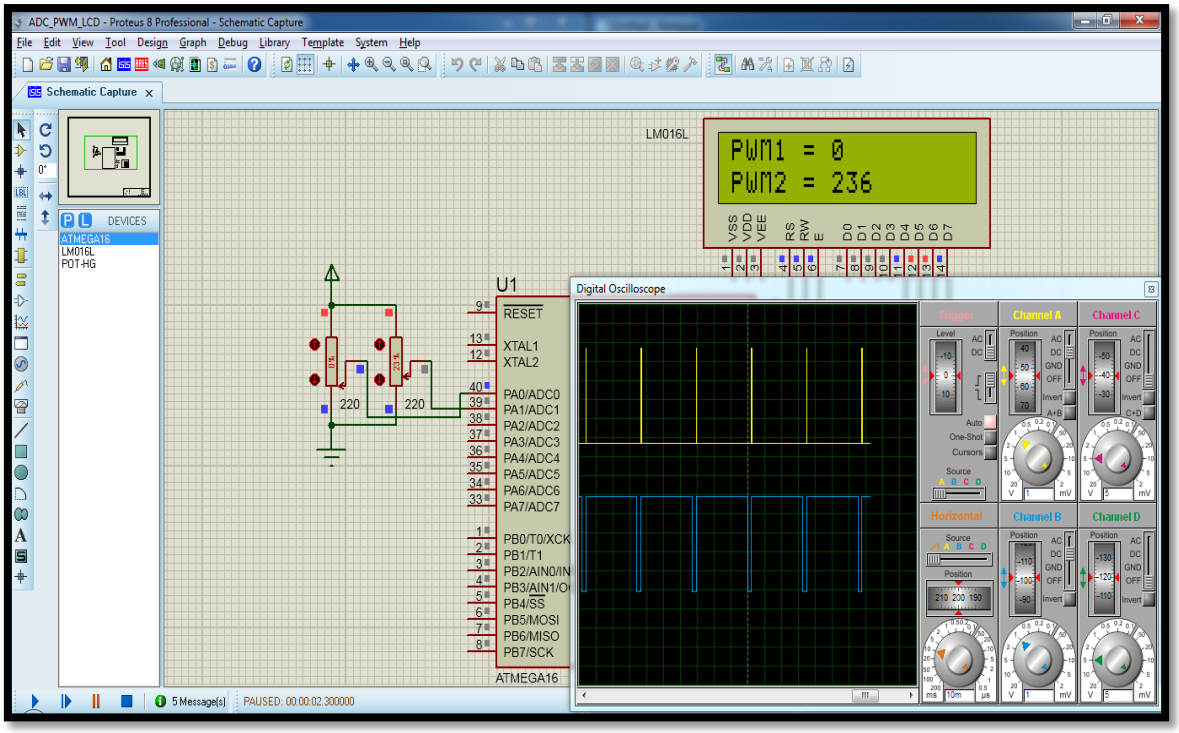


Gambar 29. Rangkaian skematik untuk menampilkan bilangan BCD

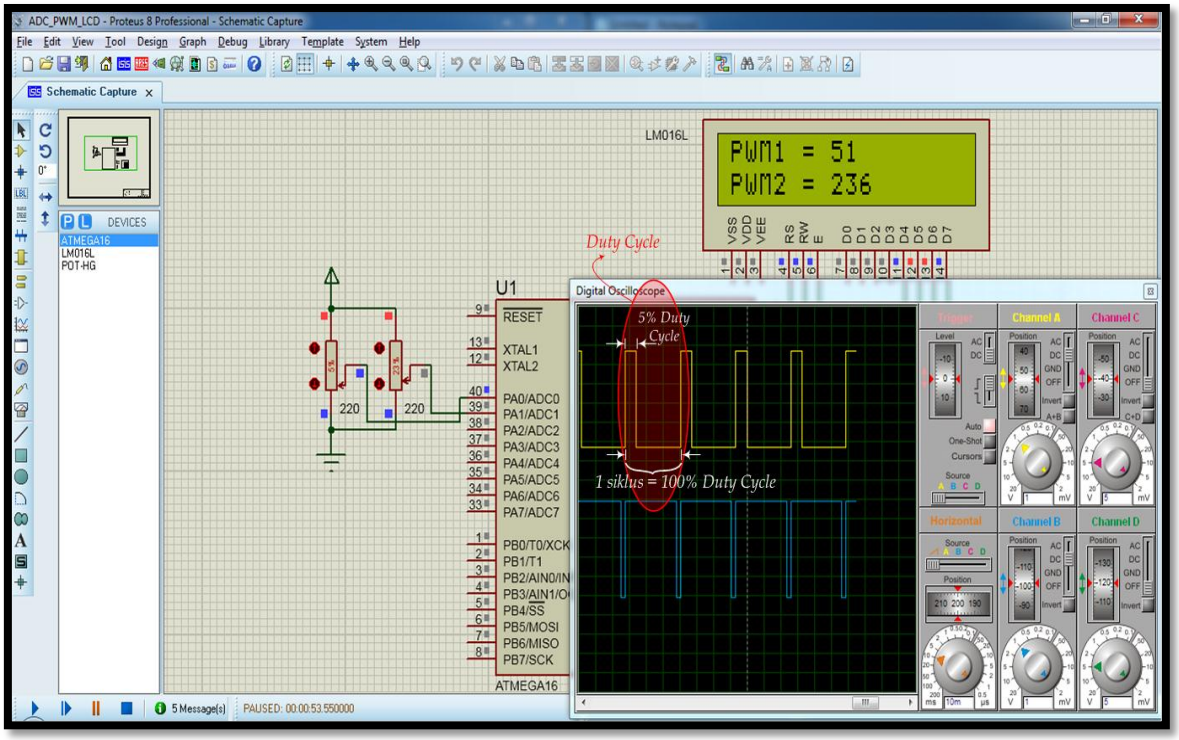
Langkah mempersiapkan Alat dan Bahan, yakni:

1. Laptop yang telah ter-*install* aplikasi *Proteus 8 Professional* dan *Code Vision* AVR!
2. Merancang Rangkaian Skematik seperti terlihat pada Gambar 29, menggunakan aplikasi *Proteus 8 Professional* pada laptop masing-masing!
3. Menuliskan "*Coding*" (**Tugas kalian**) ke dalam aplikasi *Code Vision* AVR!
4. Meng-*Upload* "*Coding*" yang telah ditulis, ke dalam desain rancangan Rangkaian Skematik kalian masing-masing!
5. *Screenshoot* atau *Capture* hasilnya! seperti tampak pada Gambar 30.
6. *Upload* semua file (*fullscreen*!) menjadi satu dalam bentuk format ***.pdf** via Google Classroom, antara lain:
   a. Rangkaian Skematik
   b. *Coding* CVAVR
   c. Hasil *Capture*

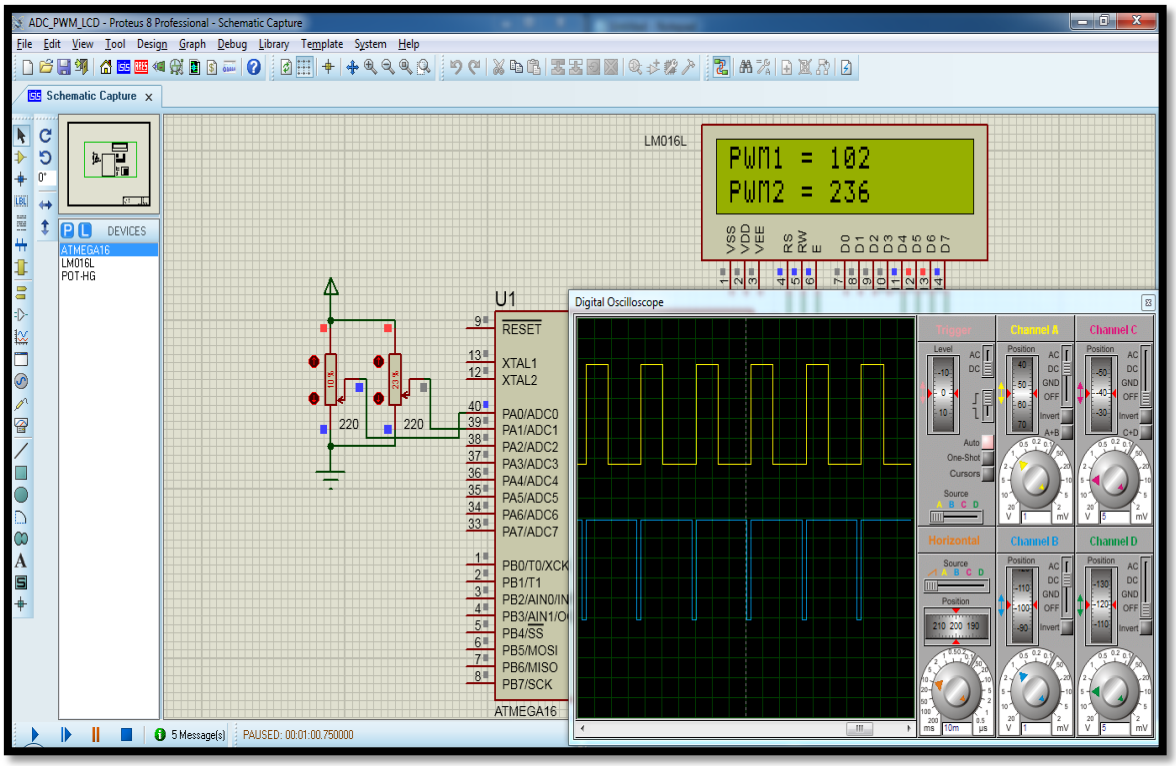
Hasil *Capture* Percobaan Lab 8, yakni:



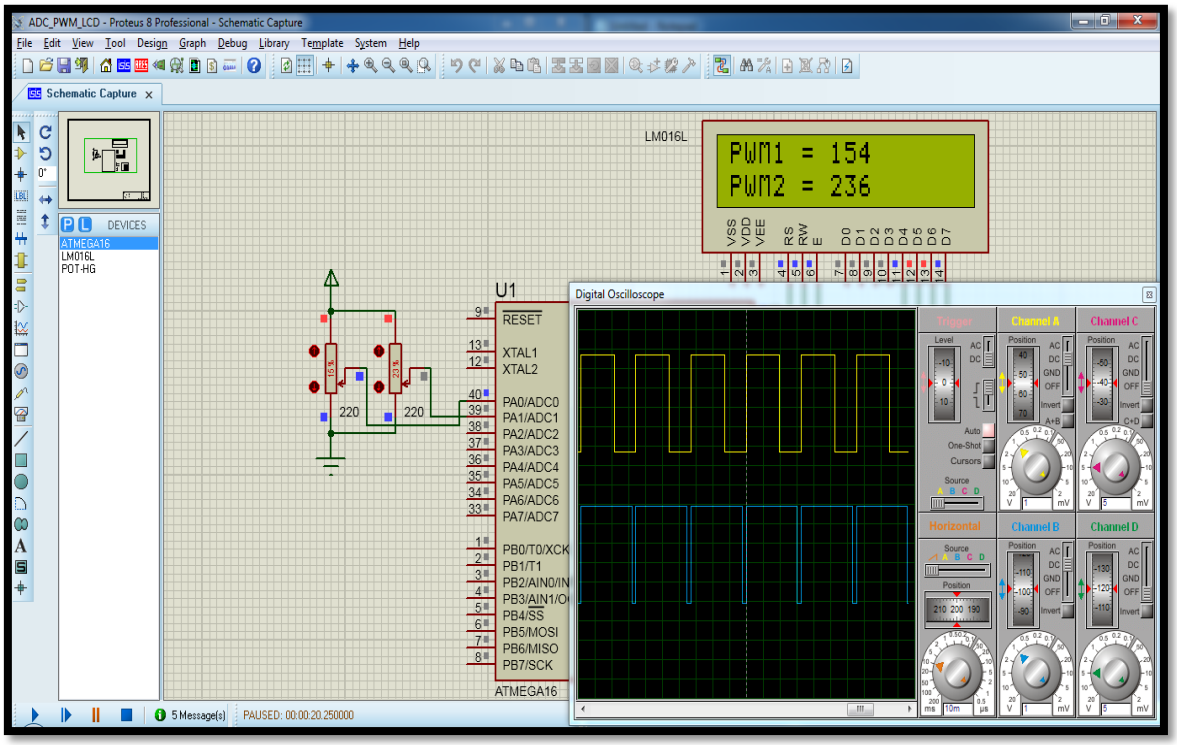
(a)



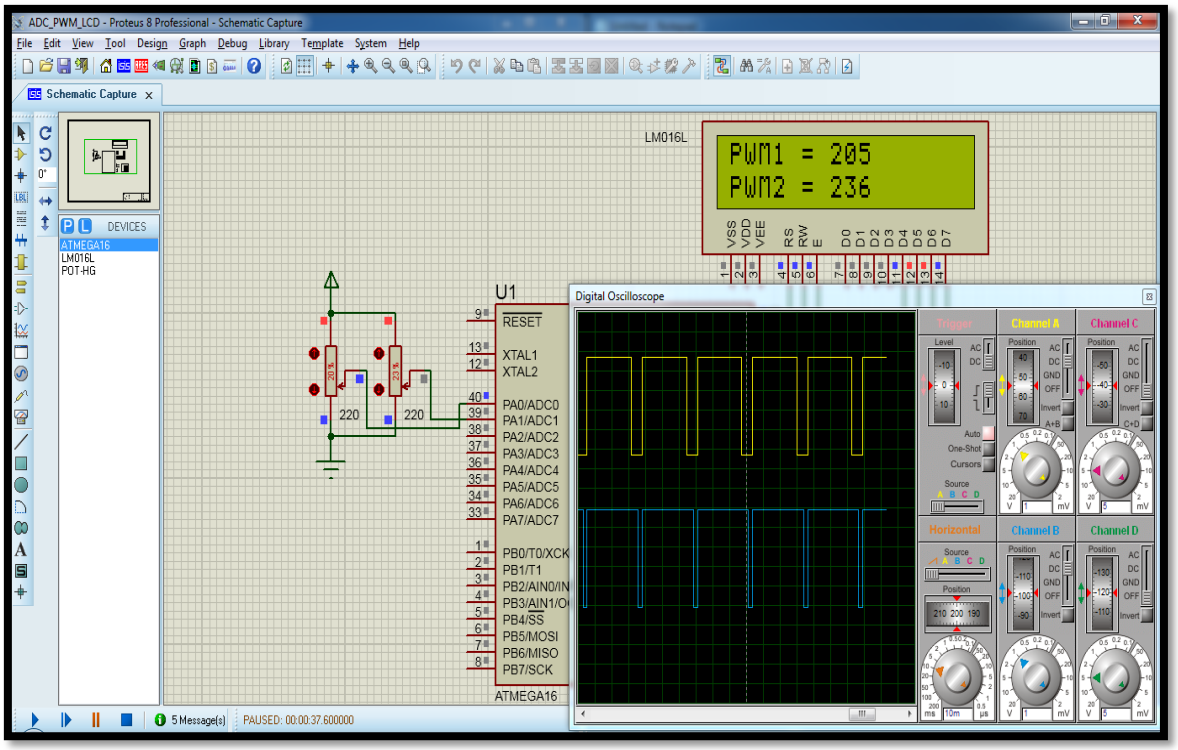
(b)


ADC_PWM_LCD - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
ATMEGA16
LM016L
POT-HG
LM016L
PWM1 = 102
PWM2 = 236
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
U1
RESET
XTAL1
XTAL2
PA0/ADC0
PA1/ADC1
PA2/ADC2
PA3/ADC3
PA4/ADC4
PA5/ADC5
PA6/ADC6
PA7/ADC7
PB0/T0/XCK
PB1/T1
PB2/AIN0/IN
PB3/AIN1/O
PB4/SS
PB5/MOSI
PB6/MISO
PB7/SCK
ATMEGA16
220
220
Digital Oscilloscope
Trigger
Channel A
Channel C
Horizontal
Channel B
Channel D
5 Message(s)
PAUSED: 00:01:00.750000

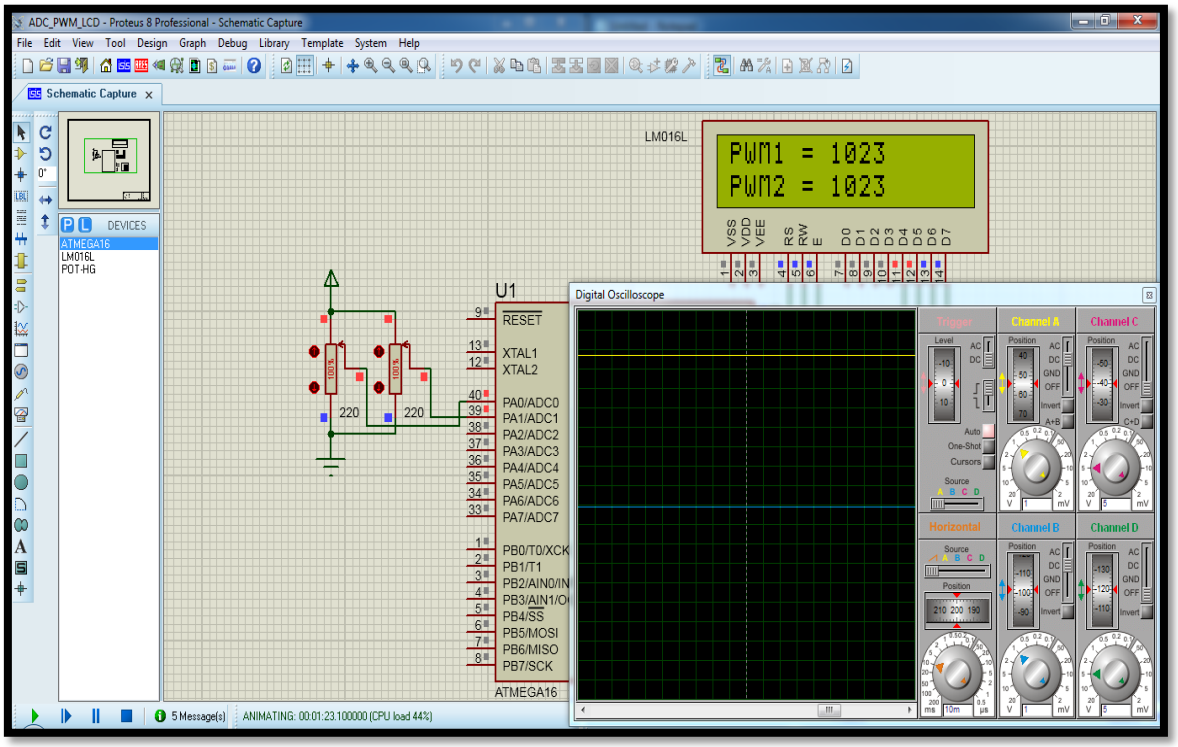
(c)


ADC_PWM_LCD - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
ATMEGA16
LM016L
POT-HG
LM016L
PWM1 = 154
PWM2 = 236
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
U1
RESET
XTAL1
XTAL2
PA0/ADC0
PA1/ADC1
PA2/ADC2
PA3/ADC3
PA4/ADC4
PA5/ADC5
PA6/ADC6
PA7/ADC7
PB0/T0/XCK
PB1/T1
PB2/AIN0/IN
PB3/AIN1/O
PB4/SS
PB5/MOSI
PB6/MISO
PB7/SCK
ATMEGA16
220
220
Digital Oscilloscope
Trigger
Channel A
Channel C
Horizontal
Channel B
Channel D
5 Message(s)
PAUSED: 00:00:20.250000

(d)



(e)



(f)

Gambar 30 (a) (b) (c) (d) (e) (f). Hasil percobaan Lab 8