

# **MIKROKONTROLER**

**PROGRAM STUDI**  
**TEKNIK TELKOMUNIKASI**  
Semester 3

Akuwan Saleh, MT

# REFERENSI

- Julien Bayle, “C Programming for Arduino”, Packt Publishing Ltd, Birmingham, May 2013.
- James Floyd K & Harold T , “Arduino Adventure Escape from Gemini Station”, Apress, 2013.
- Famosa Studio Arduino Starter Kit Manual – V1.0, Famosa Studio, 2013.
- Martin E, Joshua N, & Jordan H, “Arduino in Action“,Manning Publications.Co, USA, 2013.
- Jack Purdum, “Beginning C for Arduino, Learn C Programming for the Arduino and Compatible Microcontrollers”, Apress, 2012.
- John-David Warren, Josh Adams, and Harald Molle, “Arduino Robotics”, Springe, New York, 2011.
- \_\_\_\_\_, Sistem minimum Arduino Uno/ATmega328, Instruction Manual, 2010.

# MATERI

1. PENDAHULUAN
2. ANTARMUKA MIKROKONTROLER DENGAN LIGHT EMITTING DIODE (LED)
3. ANTARMUKA MIKROKONTROLER DENGAN SAKLAR
4. ANTARMUKA MIKROKONTROLER DENGAN LED DOT MATRIK
5. ANTARMUKA MIKROKONTROLER DENGAN KEYPAD
6. ANTARMUKA MIKROKONTROLER DENGAN SEVEN SEGMENT (7-S)
7. ANTARMUKA MIKROKONTROLER DENGAN LCD 2x16
8. PEMROGRAMAN MELODY
9. ANALOG INPUT (ADC)
10. KOMUNIKASI SERIAL
11. ANTARMUKA MIKROKONTROLER DENGAN LM 35
12. ANTARMUKA MIKROKONTROLER DENGAN LDR
13. ANTARMUKA MIKROKONTROLER DENGAN LAMPU AC 220V
14. ANTARMUKA MIKROKONTROLER DENGAN MOTOR DC
15. ANTARMUKA MIKROKONTROLER DENGAN SENSOR ULTRASONIC
16. Demo Tugas Proyek Semester

# **7. ANTARMUKA MIKROKONTROLER DENGAN LCD**

**(Liquid Cristal Display) 2x16**

# TUJUAN

- Mengetahui dan memahami cara antarmuka mikrokontroler dengan modul penampil LCD.
- Membuat program untuk menampilkan karakter pada LCD.

# DASAR TEORI

## LCD :

- Sebagai tampilan suatu data, baik karakter, huruf ataupun grafik.

## LCD → umum:

- Media tampilan yang menggunakan kristal cair sebagai penampil utama.
- LCD sudah digunakan di berbagai bidang misalnya dalam alat-alat elektronik seperti *televisi*, *kalkulator* ataupun *layar komputer*.

## Cara Kerja LCD :

- LCD berwarna seperti monitor, mempunyai banyak sekali titik cahaya (piksel).
- Piksel ini terdiri dari satu buah kristal cair sebagai sebuah titik cahaya.
- kristal cair ini tidak memancarkan cahaya sendiri. Sumber cahaya di dalam sebuah perangkat LCD adalah lampu neon berwarna putih di bagian belakang kristal cair.

# Type LCD



20x4 LCD



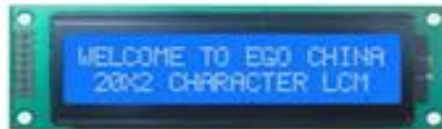
16x2 LCD



128x64 LCD



16x4 LCD



20x2 LCD



40x4 LCD



20x4 LCD



16x2 LCD Keypad Shield

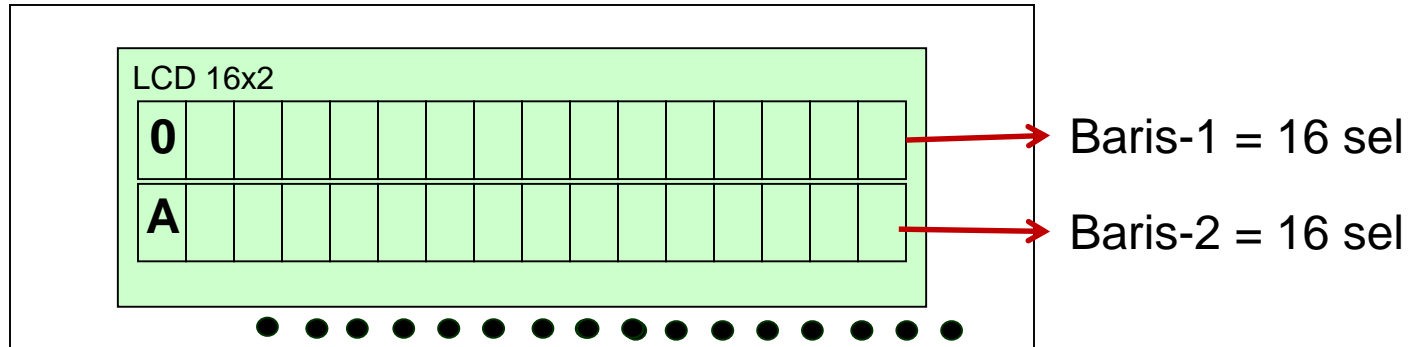


# Tipe LCD

## □ Tipe LCD : 16 X 2.

- Modul LCD Hitachi HD44780U Dot Matrix LCD.
- Menampilkan data karakter alphanumeric yang dibangkitkan dari generator RAM
- Generator RAM sebagai driver dot-matrik LCD.
- Karakter dalam bentuk ukuran 5x8 / 5x11 dot matrik.
- Jumlah karakter tiap baris = 16
- jumlah barisnya = 2

# LCD 16x2 :



## Pengendali / Kontroler LCD

- Dalam modul LCD terdapat microcontroller yang berfungsi sebagai pengendali tampilan karakter LCD.
- Microntroner pada LCD dilengkapi dengan memori dan register.

# Memori microcontroler internal LCD

- **DDRAM** (*Display Data Random Access Memory*) : tempat karakter yang akan ditampilkan
- **CGRAM** (*Character Generator Random Access Memory*) : memori untuk menggambarkan pola sebuah karakter dimana bentuk dari karakter dapat diubah-ubah sesuai dengan keinginan.
- **CGROM** (*Character Generator Read Only Memory*) : memori untuk menggambarkan pola sebuah karakter dasar yang sudah ditentukan secara permanen oleh pabrikan pembuat LCD.

# DDRAM

## (Display Data RAM)

- ❑ Konfigurasi **alamat DDRAM** untuk dua baris :
  - Display baris I : 00h – 27h
  - Display baris II : 40h – 67h
- ❑ **LCD 16 x 2**, baris I alamat DDRAM 00h – 0Fh dan baris II adalah alamat 40h – 4Fh.

### Contoh:

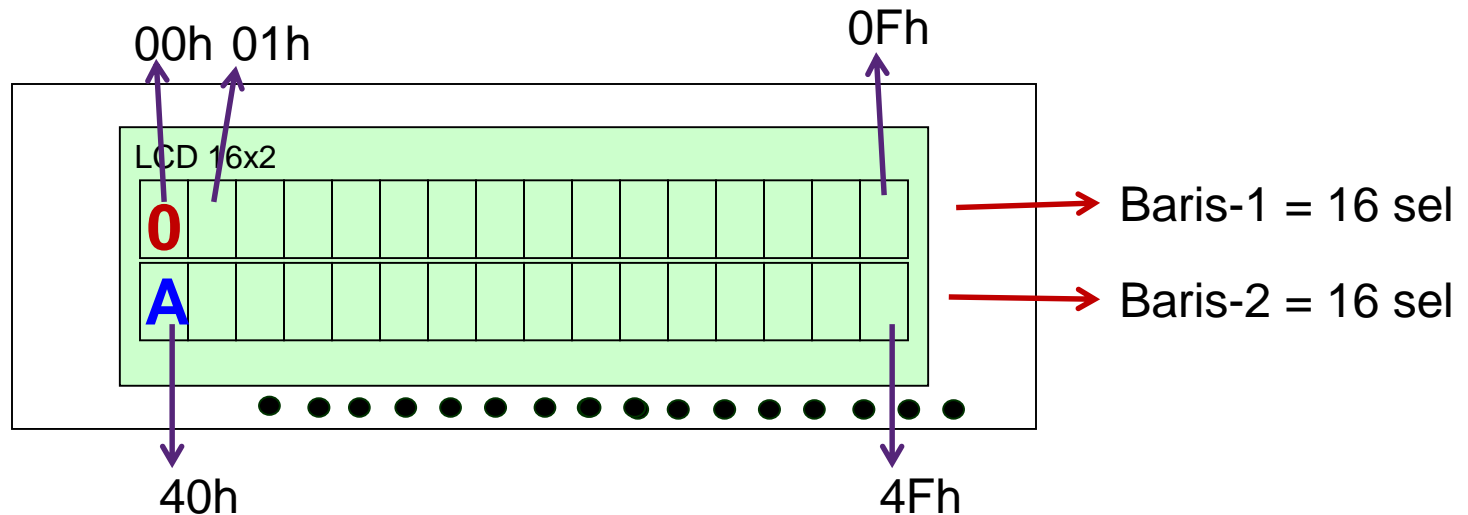
Karakter 'A' atau 41H ditulis pada **alamat 00**, maka karakter tersebut akan tampil pada **baris pertama** dan **kolom pertama** dari LCD.

Apabila ditulis di **alamat 40**, maka karakter tersebut akan tampil pada **baris kedua**, **kolom pertama** dari LCD

# DDRAM

## (Display Data RAM)

- Jika display digeser ke kanan satu kali, maka yang tampil pada baris I adalah alamat 01h – 10h dan baris II adalah alamat 41h – 50h.



# Register control dalam LCD

- ❑ **Register perintah** : berisi perintah-perintah dari mikrokontroler ke panel LCD pada saat proses *penulisan data* atau tempat status dari panel LCD dapat dibaca pada saat *pembacaan data*.
- ❑ **Register data** : untuk menuliskan atau membaca data dari atau ke DDRAM. Penulisan data pada register akan menempatkan data tersebut ke DDRAM sesuai dengan alamat yang telah diatur sebelumnya.

# Instruksi control

No.	Instruction	Hex	Decimal
1	Function Set: 8-bit, 1 Line, 5x7 Dots	0x30	48
2	Function Set: 8-bit, 2 Line, 5x7 Dots	0x38	56
3	Function Set: 4-bit, 1 Line, 5x7 Dots	0x20	32
4	Function Set: 4-bit, 2 Line, 5x7 Dots	0x28	40
5	Entry Mode	0x06	6
6	Display off Cursor off (clearing display without clearing DDRAM content)	0x08	8
7	Display on Cursor on	0x0E	14
8	Display on Cursor off	0x0C	12
9	Display on Cursor blinking	0x0F	15
10	Shift entire display left	0x18	24
12	Shift entire display right	0x1C	30
13	Move cursor left by one character	0x10	16
14	Move cursor right by one character	0x14	20
15	Clear Display (also clear DDRAM content)	0x01	1
16	Set DDRAM address or cursor position on display	0x80+add	128+add

## 1. Function Set

Menentukan panjang data (8-bit/4-bit), jumlah baris (1-line/2-line) dengan font type (5x7 dots)

**FS=30H/48** → panjang data 8-bit, 1 baris, font 5x7 dots.

**FS=38H/56** → panjang data 8-bit, 2 baris, font 5x7 dots.

**FS=20H/32** → panjang data 4-bit, 1 baris, font 5x7 dots

**FS=28H/40** → panjang data 4-bit, 2 baris, font 5x7 dots

## 2. Entry Mode

Mengaktifkan pergeseran kursor dan display, ketika selesai membaca dari atau menulis ke CGRAM.

Nilai data **EM=06**

# Instruksi control

No.	Instruction	Hex	Decimal
1	Function Set: 8-bit, 1 Line, 5x7 Dots	0x30	48
2	Function Set: 8-bit, 2 Line, 5x7 Dots	0x38	56
3	Function Set: 4-bit, 1 Line, 5x7 Dots	0x20	32
4	Function Set: 4-bit, 2 Line, 5x7 Dots	0x28	40
5	Entry Mode	0x06	6
6	Display off Cursor off (clearing display without clearing DDRAM content)	0x08	8
7	Display on Cursor on	0x0E	14
8	Display on Cursor off	0x0C	12
9	Display on Cursor blinking	0x0F	15
10	Shift entire display left	0x18	24
12	Shift entire display right	0x1C	30
13	Move cursor left by one character	0x10	16
14	Move cursor right by one character	0x14	20
15	Clear Display (also clear DDRAM content)	0x01	1
16	Set DDRAM address or cursor position on display	0x80+add	128+add

## 3. Display off Cursor off

Menghapus tampilan dan kursor tanpa menghapus isi data di DDRAM

Nilai data **Doff/Coff=08**

## 4. Display on Cursor on

Mengaktifkan tampilan dan kursor Nilai data **Don/Con=0EH/14**

## 5. Display on Cursor off

Mengaktifkan tampilan dan menghapus kursor. Nilai data **Don/Coff=0CH/12**

## 6. Display on Cursor blinking

Mengaktifkan tampilan dan kursor blink. Nilai data **Don/Cb=0FH/15**

## 7. Shift entire display left

Mengaktifkan tampilan dan geser data ke kiri. Nilai data **Sh/Dl=18H/24**

## 8. Shift entire display right

Mengaktifkan tampilan dan geser data ke kanan. Nilai data **Sh/Dr=1CH/30**



# Instruksi control

No.	Instruction	Hex	Decimal
1	Function Set: 8-bit, 1 Line, 5x7 Dots	0x30	48
2	Function Set: 8-bit, 2 Line, 5x7 Dots	0x38	56
3	Function Set: 4-bit, 1 Line, 5x7 Dots	0x20	32
4	Function Set: 4-bit, 2 Line, 5x7 Dots	0x28	40
5	Entry Mode	0x06	6
6	Display off Cursor off (clearing display without clearing DDRAM content)	0x08	8
7	Display on Cursor on	0x0E	14
8	Display on Cursor off	0x0C	12
9	Display on Cursor blinking	0x0F	15
10	Shift entire display left	0x18	24
12	Shift entire display right	0x1C	30
13	Move cursor left by one character	0x10	16
14	Move cursor right by one character	0x14	20
15	Clear Display (also clear DDRAM content)	0x01	1
16	Set DDRAM address or cursor position on display	0x80+add	128+add

## 9. Move cursor left by one character

Menggeser kursor kekiri satu karakter.

Nilai data **Mcl=10H/16**

## 10. Move cursor right by one character

Menggeser kursor kekanan satu karakter. Nilai data **Mcr=14H/20**

## 11. Clear Display

Menghapus tampilan sekaligus menghapus isi DDRAM. Nilai data **CD=1**

## 12. Set DDRAM address or cursor position on display

Menghapus tampilan sekaligus menghapus isi DDRAM. Nilai data **CD=1**

# Deskripsi Pin LCD 16x2

Pin	Nama	Keterangan
1	VSS	GND
2	VCC	+5V
3	VEE	Tegangan Kontras LCD
4	RS	Register Select; H: Data Input; L: Instruction Input
5	R/W	1 = Read, 0 = Write
6	E	Enable Clock LCD
7	D0	Data Bus 0
8	D1	Data Bus 1
9	D2	Data Bus 2
10	D3	Data Bus 3
11	D4	Data Bus 4
12	D5	Data Bus 5
13	D6	Data Bus 6
14	D7	Data Bus 7
15	Anoda(V+BL)	Tegangan Positif Backlight
16	Katoda(V-BL)	Tegangan Negatif Backlight



# Jalur input dan kontrol LCD

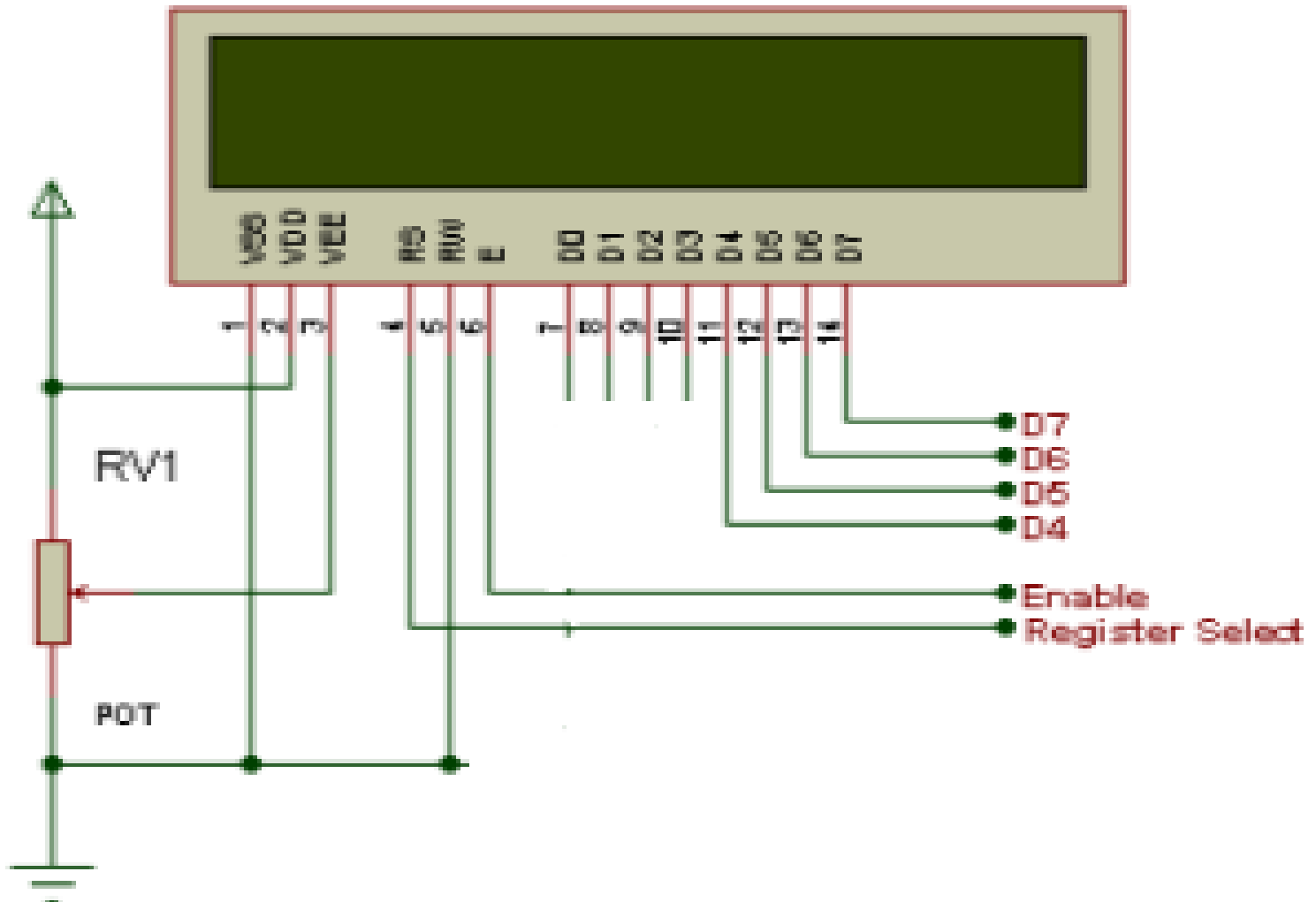
- ❑ **Pin data** : jalur untuk memberikan data karakter yang ingin ditampilkan pada LCD, dapat dihubungkan dengan bus data dari rangkaian lain seperti mikrokontroler dengan lebar data 8 bit.
- ❑ **Pin RS (Register Select)** : berfungsi sebagai indikator atau yang menentukan jenis data yang masuk, apakah data atau perintah. Logika **low** menunjukkan yang masuk adalah **perintah**, sedangkan logika **high** menunjukkan **data**.

# Jalur input dan kontrol LCD

- **Pin R/W (Read Write)** : berfungsi sebagai instruksi pada modul jika **low tulis data**, sedangkan **high baca data**.
- **Pin E (Enable)** : mengaktifkan LCD pada proses penulisan data ke Register Kontrol dan Register Data LCD.
- **Pin VLCD** : berfungsi mengatur kecerahan tampilan (kontras) dimana pin ini dihubungkan dengan trimpot 5 Kohm, jika tidak digunakan dihubungkan ke ground, sedangkan tegangan catu daya ke LCD sebesar 5 Volt.

# Rangkaian LCD 2x16

LCD



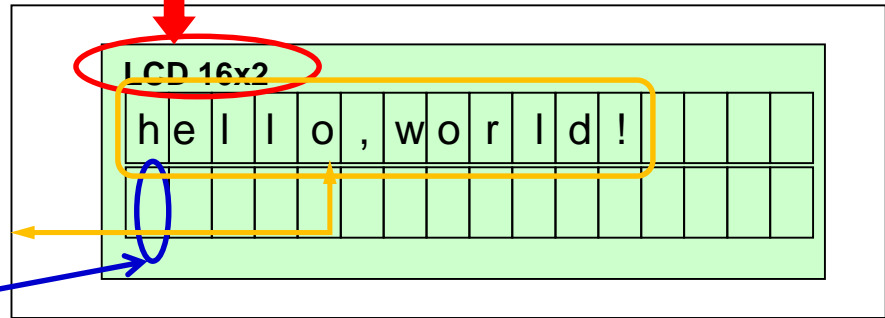
# Penulisan program :

```
Icd.begin(16, 2);
```

```
Icd.print("hello, world!");
```

Menampilkan teks

```
Icd.setCursor(0,1); Posisi teks  
kolom 1, baris 2
```



## Membuat Teks berjalan:

```
static char scrollMessage[] = {"Lcd Dengan Arduino "};  
void loop() {  
  lcd.setCursor(0, 1);  
  if (++position > strlen(scrollMessage) - 16) position = 0;  
  lcd.print(&scrollMessage[position]);  
  delay(150);  
}
```

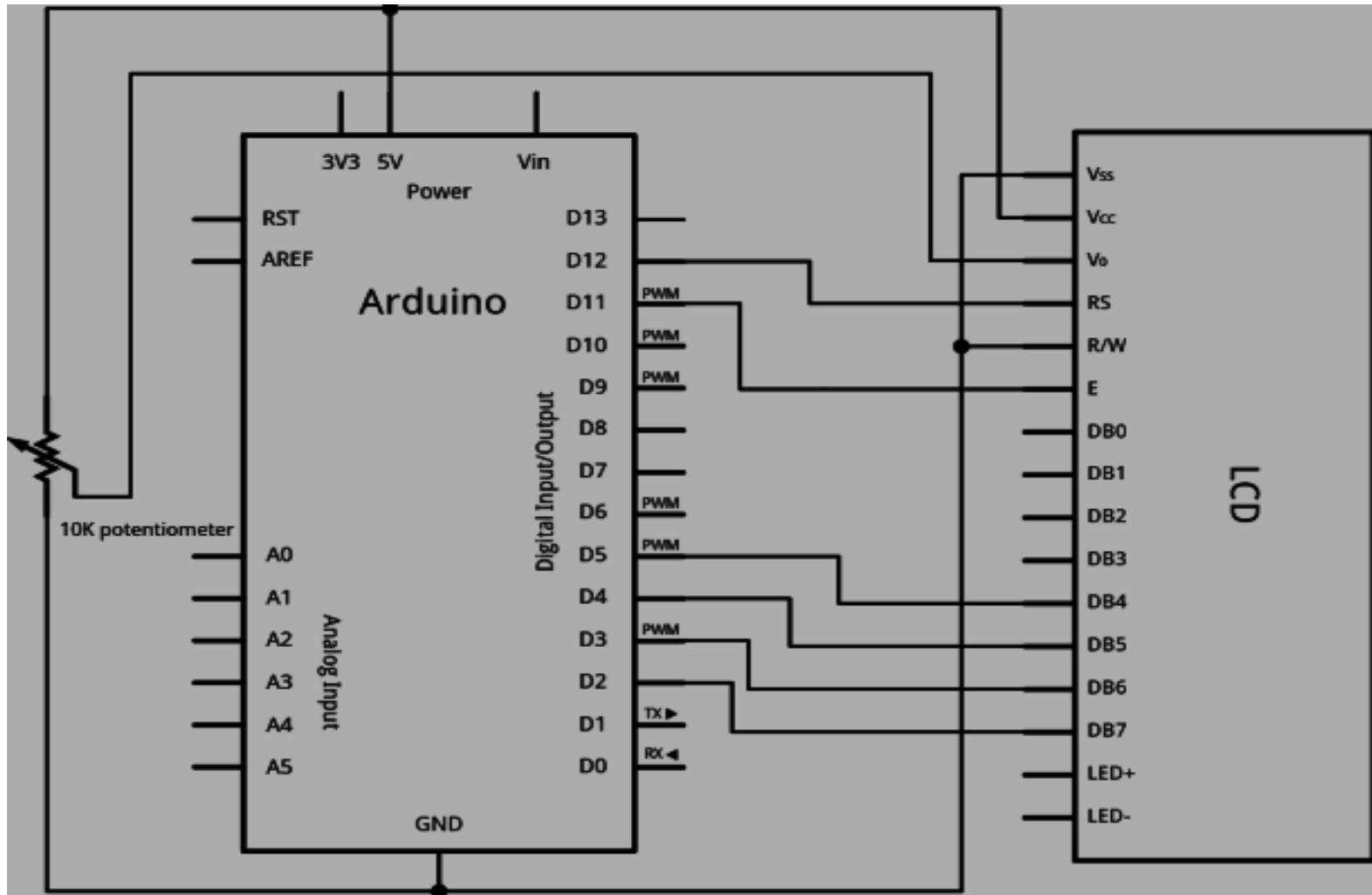
## Membuat Teks berjalan:

```
lcd.setCursor(0, 0);  
lcd.print("Pemrograman LCD 16x2");  
for (i = 0 ; i < 16; i ++ ) {  
  lcd.scrollDisplayLeft();  
  delay(400);  
}
```

# Hardware :

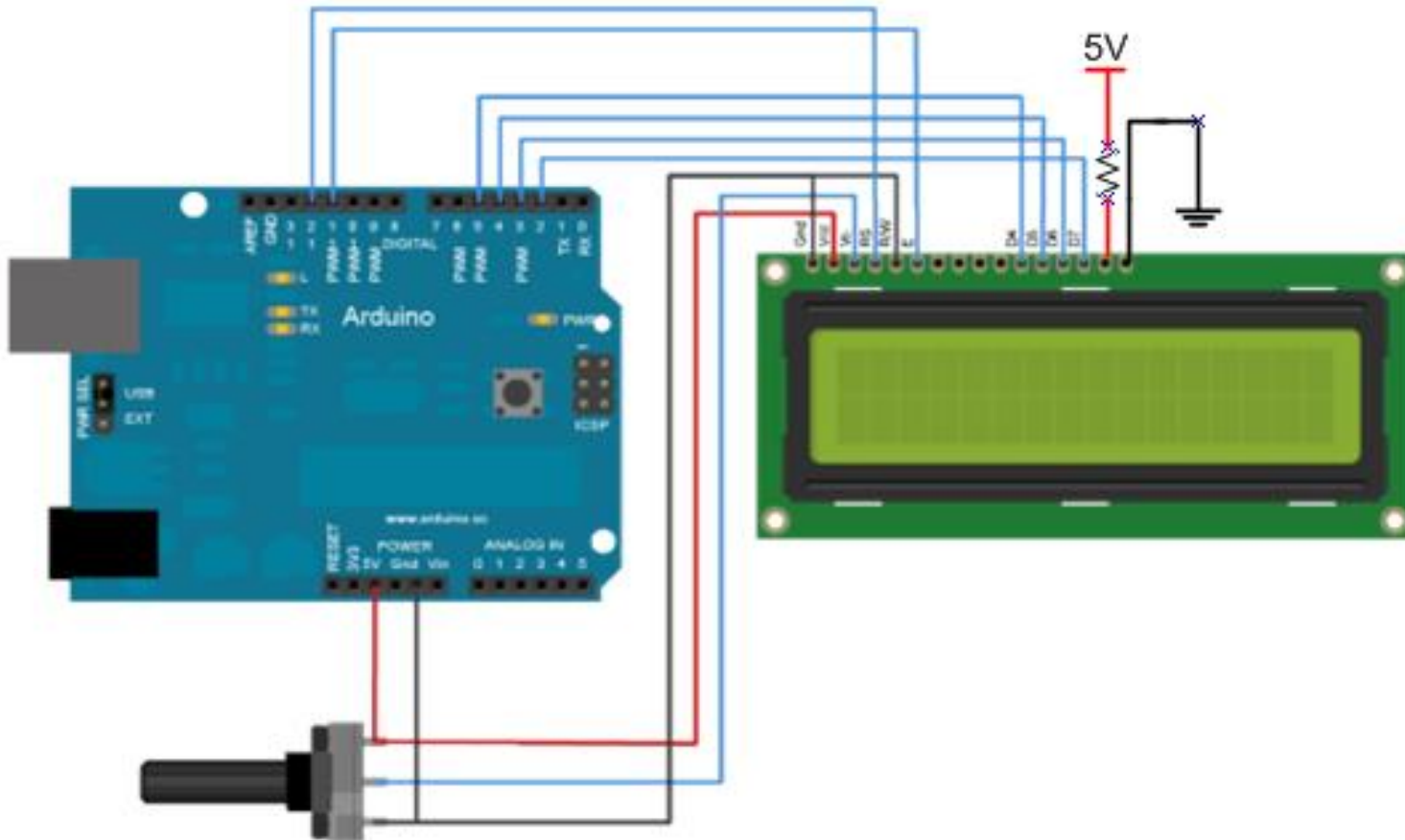
- Arduino Uno Board
- 1 buah LCD 16x2 Hitachi HD44780
- 1 buah potensiometer / variabel resistor 10K
- Kabel jumper

# Rangkaian Antarmuka:





# Rangkaian Antarmuka:



# Rangkaian Antarmuka:

- **Pin 3 VEE LCD** untuk pengaturan kontras karakter dengan menyambungkan **potensio** 10 KOhm ke **+5v** dan **GND**
- **Pin 5 (R/W) LCD** → pin arduino **Ground**
- **Pin 4 RS LCD** → pin arduino digital **pin 12**
- **Pin 6 E LCD** → pin arduino digital **pin 11**
- **Pin 11 D4 LCD** → pin arduino digital **pin 5**
- **Pin 12 D5 LCD** → pin arduino digital **pin 4**
- **Pin 13 D6 LCD** → pin arduino digital **pin 3**
- **Pin 14 D7 LCD** → pin arduino digital **pin 2**
- **Pin 15 Anoda LCD** → pin arduino **+5volt**
- **Pin 16 Katoda LCD** → pin arduino **Ground**
- **Pin 1 VSS (GND) LCD** → pin arduino **Ground**
- **Pin 2 VCC LCD** → pin arduino **+5volt**

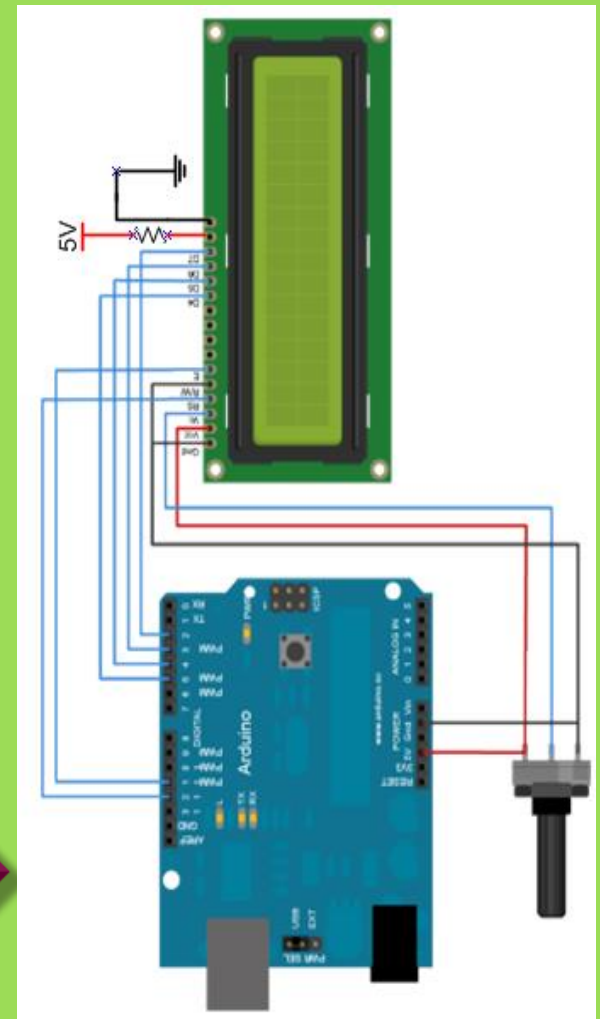
# PROGRAM 1a:

```
#include <LiquidCrystal.h>
```

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

```
void setup() {  
  lcd.begin(16, 2);  
  lcd.print("hello, world!");  
}
```

```
void loop() {  
  lcd.noDisplay();  
  delay(500);  
  lcd.display();  
  delay(500);  
}
```



# PROGRAM 1b dan 1c:

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  lcd.begin(16, 2);
  lcd.print("hello, world!");
}

void loop() {
  lcd.setCursor(0,1);
  lcd.print(millis() / 1000);
}
```

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup()
{
  lcd.begin(16, 2);
  lcd.setCursor(3,0);
  lcd.print("Telkom");
  lcd.setCursor(0,1);
  lcd.print("2-TA dan TB");
}

void loop()
{
}
```



Teks 2 baris di LCD

# PROGRAM 2a:

Teks berjalan

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  lcd.begin(16, 2);
}

void loop() {
  int i;
  lcd.setCursor(0, 0);
  lcd.print("Pemrograman LCD 16x2");
  for (i = 0 ; i < 16; i ++ ) {
    lcd.scrollDisplayLeft();
    delay(400);
  }
}
```

# PROGRAM 2b:

Teks berjalan

```
#include <LiquidCrystal.h>
```

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

```
static char scrollMessage[] = {"  LCD Dengan Arduino  "};
```

```
int position = 0;
```

```
void setup() {
```

```
  lcd.begin(16, 2);
```

```
  lcd.print("Belajar Program");
```

```
}
```

```
void loop() {
```

```
  lcd.setCursor(0, 1);
```

```
  if (++position > strlen(scrollMessage) - 16) position = 0;
```

```
  lcd.print(&scrollMessage[position]);
```

```
  delay(150);
```

```
}
```

## ***Hasil :***

- Catat tampilan pada LCD dan foto sebagai dokumentasi.
- Buat laporan hasil dari percobaan

## ***Latihan :***

1. Modifikasi program contoh diatas supaya dapat menampilkan data **hello, world!** di baris-1, **5171** di baris-2 pada LCD dan bergerak kekanan kekiri 3x.



## ***Latihan :***

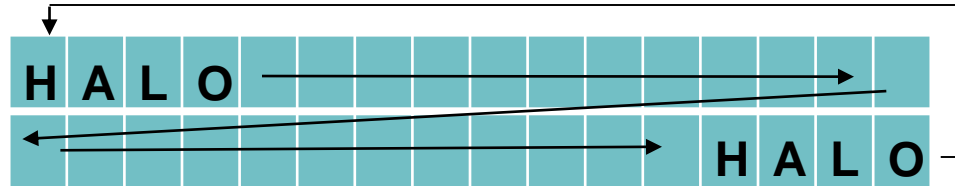
2. Buat program untuk menampilkan tulisan **kelas** anda di baris pertama LCD dan tulisan **prodi** anda di baris kedua LCD
  - tulisan di baris 1 dan 2 berjalan/ bergeser bersama-sama kekanan 3x dan kiri 3x kemudian berkedip 3x.
3. Buat program untuk menampilkan tulisan **kelas** anda ditengah baris pertama LCD kemudian tulisan berjalan berputar seperti pada gambar berikut.





## Latihan :

4. Buat program untuk menampilkan tulisan **HALO** di baris pertama LCD kemudian tulisan berjalan zig-zag seperti pada gambar dibawah ini.



5. Buat program untuk menampilkan tulisan **kelas** anda di baris pertama LCD dan tulisan **prodi** anda di baris kedua LCD . Tulisan baris-1 dan 2 bergerak keatas 5x.



# **8. PEMROGRAMAN MELODY**

# TUJUAN

- Membuat melody, DO, RE, MI, FA, SOL, LA, SI, DO”.
- Membuat program untuk mambangkitan frekuensi nada melody.

# DASAR TEORI

## **NADA :**

- Suatu jenis unsur suprasegmental yang ditandai oleh tinggi-rendahnya arus-ujaran.
- Tinggi rendahnya arus-ujaran terjadi karena frekuensi getaran yang berbeda antar segmen.

## **MELODY :**

- Melodi adalah susunan nada yang diatur tinggi rendahnya, pola, dan harga nada sehingga menjadi kalimat lagu.

# DASAR TEORI

## Melody dalam speaker

- Setiap nada yang di bunyikan oleh Speaker memiliki frekuensi
- Dimana frekuensi itu telah membentuk nada, seperti Do , RE, MI, FA, SO, LA, SI, dan DO tinggi.
- Frekwensi- frekwensi ini, jika disusun dapat dibentuk suatu melody music.

NADA	DO	RE	MI	FA	SOL	LA	SI	DO''
FREKUENSI	262	294	330	349	395	440	494	523

## Tabel Frekuensi

<u>note</u>	<u>frequency</u>	<u>period</u>	<u>timeHigh</u>
c	261 Hz	3830	1915
d	294 Hz	3400	1700
e	329 Hz	3038	1519
f	349 Hz	2864	1432
g	392 Hz	2550	1275
a	440 Hz	2272	1136
b	493 Hz	2028	1014
C	523 Hz	1912	956

$$\text{timeHigh} = \text{period} / 2$$

# Tabel Frekuensi

## Frequencies for equal-tempered scale

This table created using  $A^4 = 440$  Hz

Speed of sound = 345 m/s = 1130 ft/s = 770 miles/hr

("Middle C" is  $C^4$  )

NOTE	FREQUENCY (HZ)	WAVELENGTH (CM)
$C^0$	16.35	2100.
$C_{\#}^0/D_b^0$	17.32	1990.
$D^0$	18.35	1870.
$D_{\#}^0/E_b^0$	19.45	1770.
$E^0$	20.60	1670.
$F^0$	21.83	1580.
$F_{\#}^0/G_b^0$	23.12	1490.
$G^0$	24.50	1400.
$G_{\#}^0/A_b^0$	25.96	1320.

$A^0$	27.50	1250.
$A_{\#}^0/B_b^0$	29.14	1180.
$B^0$	30.87	1110.
$C^1$	32.70	1050.
$C_{\#}^1/D_b^1$	34.65	996.
$D^1$	36.71	940.
$D_{\#}^1/E_b^1$	38.89	887.
$E^1$	41.20	837.
$F^1$	43.65	790.
$F_{\#}^1/G_b^1$	46.25	746.
$G^1$	49.00	704.
$G_{\#}^1/A_b^1$	51.91	665.

# Tabel Frekuensi

NOTE	FREQUENCY (HZ)	WAVELENGTH (CM)			
A <sup>1</sup>	55.00	627.	A <sub>#</sub> <sup>2</sup> /B <sub>b</sub> <sup>2</sup>	116.54	296.
A <sub>#</sub> <sup>1</sup> /B <sub>b</sub> <sup>1</sup>	58.27	592.	B <sup>2</sup>	123.47	279.
B <sup>1</sup>	61.74	559.	C <sup>3</sup>	130.81	264.
C <sup>2</sup>	65.41	527.	C <sub>#</sub> <sup>3</sup> /D <sub>b</sub> <sup>3</sup>	138.59	249.
C <sub>#</sub> <sup>2</sup> /D <sub>b</sub> <sup>2</sup>	69.30	498.	D <sup>3</sup>	146.83	235.
D <sup>2</sup>	73.42	470.	D <sub>#</sub> <sup>3</sup> /E <sub>b</sub> <sup>3</sup>	155.56	222.
D <sub>#</sub> <sup>2</sup> /E <sub>b</sub> <sup>2</sup>	77.78	444.	E <sup>3</sup>	164.81	209.
E <sup>2</sup>	82.41	419.	F <sup>3</sup>	174.61	198.
F <sup>2</sup>	87.31	395.	F <sub>#</sub> <sup>3</sup> /G <sub>b</sub> <sup>3</sup>	185.00	186.
F <sub>#</sub> <sup>2</sup> /G <sub>b</sub> <sup>2</sup>	92.50	373.	G <sup>3</sup>	196.00	176.
G <sup>2</sup>	98.00	352.	G <sub>#</sub> <sup>3</sup> /A <sub>b</sub> <sup>3</sup>	207.65	166.
G <sub>#</sub> <sup>2</sup> /A <sub>b</sub> <sup>2</sup>	103.83	332.	A <sup>3</sup>	220.00	157.
A <sup>2</sup>	110.00	314.	A <sub>#</sub> <sup>3</sup> /B <sub>b</sub> <sup>3</sup>	233.08	148.
			B <sup>3</sup>	246.94	140.
			C <sup>4</sup>	261.63	132.



# Tabel Frekuensi

NOTE	FREQUENCY (HZ)	WAVELENGTH (CM)
$C_{\#}^4/D_b^4$	277.18	124.
$D^4$	293.66	117.
$D_{\#}^4/E_b^4$	311.13	111.
$E^4$	329.63	105.
$F^4$	349.23	98.8
$F_{\#}^4/G_b^4$	369.99	93.2
$G^4$	392.00	88.0
$G_{\#}^4/A_b^4$	415.30	83.1
$A^4$	440.00	78.4
$A_{\#}^4/B_b^4$	466.16	74.0
$B^4$	493.88	69.9
$C^5$	523.25	65.9
$C_{\#}^5/D_b^5$	554.37	62.2
$D^5$	587.33	58.7
$D_{\#}^5/E_b^5$	622.25	55.4
$E^5$	659.26	52.3
$F^5$	698.46	49.4
$F_{\#}^5/G_b^5$	739.99	46.6
$G^5$	783.99	44.0
$G_{\#}^5/A_b^5$	830.61	41.5
$A^5$	880.00	39.2
$A_{\#}^5/B_b^5$	932.33	37.0
$B^5$	987.77	34.9
$C^6$	1046.50	33.0
$C_{\#}^6/D_b^6$	1108.73	31.1
$D^6$	1174.66	29.4
$D_{\#}^6/E_b^6$	1244.51	27.7
$E^6$	1318.51	26.2

# Tabel Frekuensi

NOTE	FREQUENCY (HZ)	WAVELENGTH (CM)
F <sup>6</sup>	1396.91	24.7
F <sub>#</sub> <sup>6</sup> /G <sub>b</sub> <sup>6</sup>	1479.98	23.3
G <sup>6</sup>	1567.98	22.0
G <sub>#</sub> <sup>6</sup> /A <sub>b</sub> <sup>6</sup>	1661.22	20.8
A <sup>6</sup>	1760.00	19.6
A <sub>#</sub> <sup>6</sup> /B <sub>b</sub> <sup>6</sup>	1864.66	18.5
B <sup>6</sup>	1975.53	17.5
C <sup>7</sup>	2093.00	16.5
C <sub>#</sub> <sup>7</sup> /D <sub>b</sub> <sup>7</sup>	2217.46	15.6
D <sup>7</sup>	2349.32	14.7
D <sub>#</sub> <sup>7</sup> /E <sub>b</sub> <sup>7</sup>	2489.02	13.9
E <sup>7</sup>	2637.02	13.1
F <sup>7</sup>	2793.83	12.3
F <sub>#</sub> <sup>7</sup> /G <sub>b</sub> <sup>7</sup>	2959.96	11.7
G <sup>7</sup>	3135.96	11.0
G <sub>#</sub> <sup>7</sup> /A <sub>b</sub> <sup>7</sup>	3322.44	10.4
A <sup>7</sup>	3520.00	9.8
A <sub>#</sub> <sup>7</sup> /B <sub>b</sub> <sup>7</sup>	3729.31	9.3
B <sup>7</sup>	3951.07	8.7
C <sup>8</sup>	4186.01	8.2
C <sub>#</sub> <sup>8</sup> /D <sub>b</sub> <sup>8</sup>	4434.92	7.8
D <sup>8</sup>	4698.64	7.3
D <sub>#</sub> <sup>8</sup> /E <sub>b</sub> <sup>8</sup>	4978.03	6.9

# DASAR TEORI

## Sintak :

- tone(pin, frequency)
- tone(pin, frequency, duration)

## PARAMETERS

- pin: pin untuk menghasilkan nada
- frequency: frekuensi nada (hertz)
- duration: durasi nada dalam milidetik (opsional)

# DASAR TEORI

## Contoh :

**Contoh : tone(pin, frequency)**

```
int speaker= 9;  
tone(speaker, 262);
```

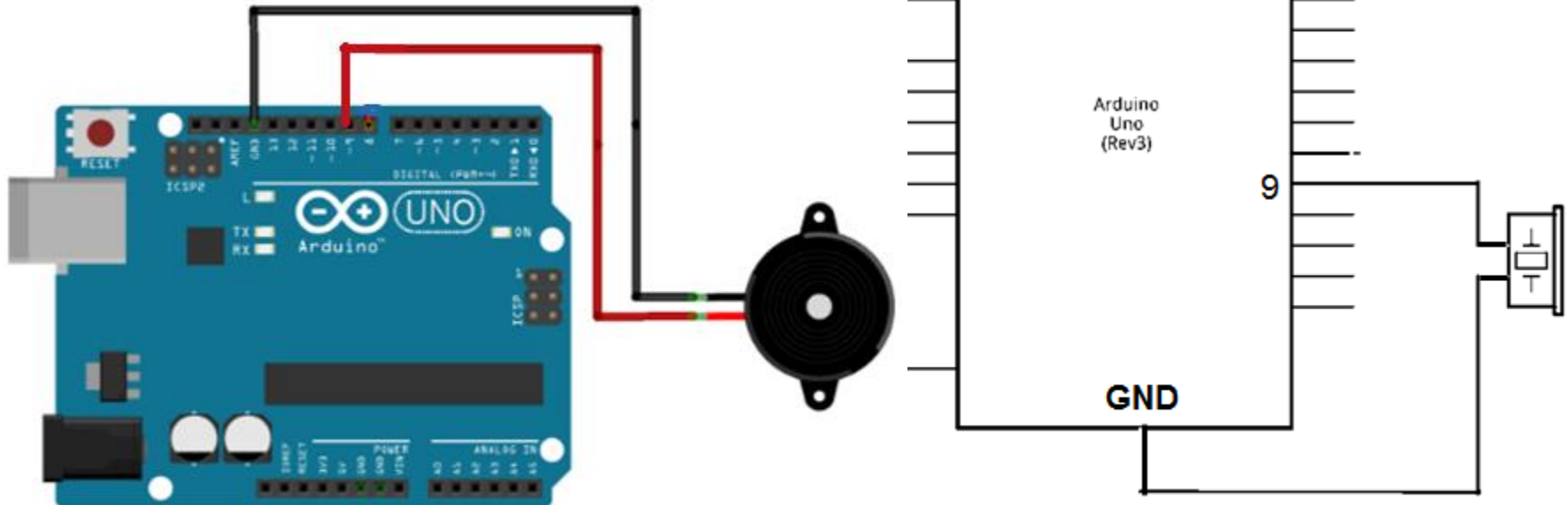
**Contoh : tone(pin, frequency, duration)**

```
int noteDuration = 1000 / 4;  
tone(9, 262, noteDuration);
```

## Hardware :

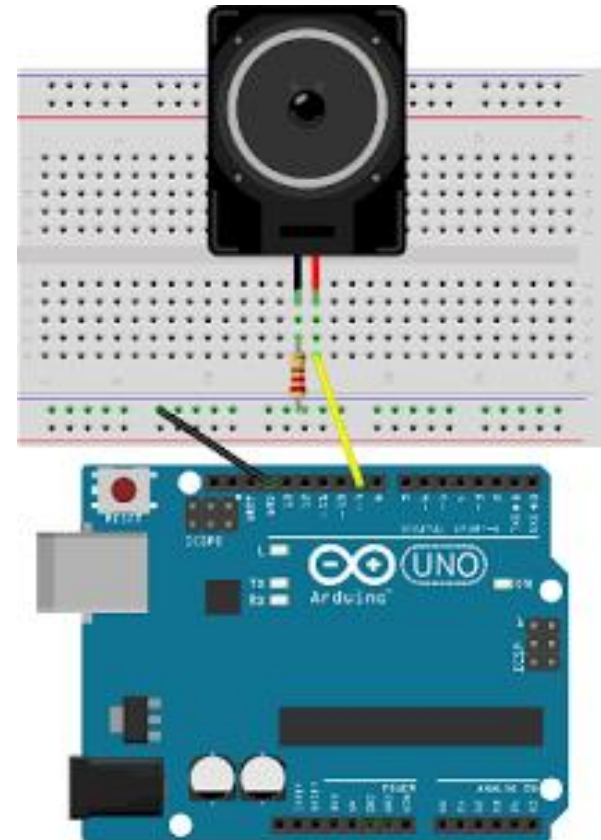
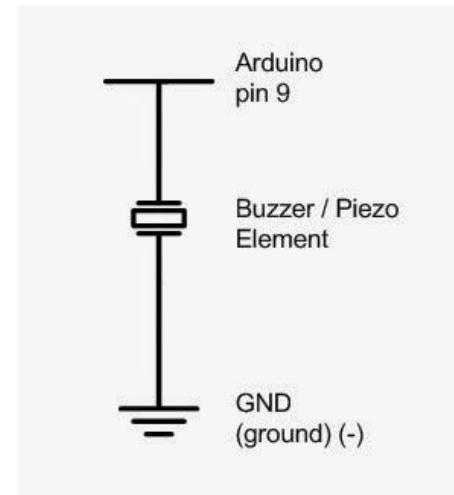
- 1x Arduino ; 1x Breadboard
- 1x Speaker; 1x Resistor 220 ohm
- Kabel jumper

## Rangkaian :



# Rangkaian :

1. Hubungkan GND arduino ke Breadboard.
2. Hubungkan GND breadboard ke kabel kiri Speaker.
3. Hubungkan pin 9 Arduino ke kabel kanan Speaker.
4. Speaker sebenarnya tidak memiliki polaritas. Jadi mau dipasang seperti apapun antara positif dan negatif, Speaker masih tetap akan menyala.



# ***PROGRAM 1:***

```
int speaker= 9;
```

```
DO,RE,MI,FA,SOL,...
```

```
void setup() { }
```

```
void loop() {
```

```
  tone(speaker, 262);   delay(500); // nada DO  
  tone(speaker, 294);   delay(550); // nada RE  
  tone(speaker, 330);   delay(590); // nada MI  
  tone(speaker, 349);   delay(600); // nada FA  
  tone(speaker, 395);   delay(610); // nada SOL  
  tone(speaker, 440);   delay(620); // nada LA  
  tone(speaker, 494);   delay(630); // nada SI  
  tone(speaker, 523);   delay(700); // nada DO
```

```
}
```

# PROGRAM 2:

Twinkle Twinkle Little Star

```
int speakerPin = 9;

int length = 15;
char notes[] = "ccggaagffeeddc ";
int beats[] = { 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 4 };
int tempo = 300;
void playTone(int tone, int duration) {
  for (long i = 0; i < duration * 1000L; i += tone * 2) {
    digitalWrite(speakerPin, HIGH);
    delayMicroseconds(tone);
    digitalWrite(speakerPin, LOW);
    delayMicroseconds(tone);
  }
}
```



# ***PROGRAM 2 - lanjutan***

```
void playNote(char note, int duration) {  
    char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' };  
    int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014,  
                    956 };  
  
    // play the tone corresponding to the note name  
    for (int i = 0; i < 8; i++) {  
        if (names[i] == note) {  
            playTone(tones[i], duration);  
        }  
    }  
}
```

# PROGRAM 2 - lanjutan

```
void setup() {  
  pinMode(speakerPin, OUTPUT);  
}  
void loop() {  
  for (int i = 0; i < length; i++) {  
    if (notes[i] == ' ') {  
      delay(beats[i] * tempo); // rest  
    } else {  
      playNote(notes[i], beats[i] * tempo);  
    }  
    // pause between notes  
    delay(tempo / 2);  
  }  
}
```

# ***PROGRAM 3:***

```
int speaker= 9;
void setup(){ }
void DO() { tone(speaker, 262); }
void RE() { tone(speaker, 294); }
void MI() { tone(speaker, 330); }
void FA() { tone(speaker, 349); }
void SOL() { tone(speaker, 395); }
void LA() { tone(speaker, 440)
void SI() { tone(speaker, 494); }
void DOO() { tone(speaker, 523); }
void DIAM() { noTone(speaker); }
void loop() {
    SOL(); delay(500); MI(); delay(1000); SOL(); delay(1000); //1
    DOO(); delay(1000); DIAM(); delay(500);
    SOL(); delay(500); DOO(); delay(500); SOL(); delay(500); //2
    FA() ; delay(500); MI(); delay(500); RE(); delay(1000);
    DIAM(); delay(500);
```

Lihat Kebunku

# ***PROGRAM 3 - lanjutan***

```
FA(); delay(500); RE(); delay(1000); FA(); delay(1000); //3
LA(); delay(1000); SOL(); delay(500); DO(); delay(500); //4
MI(); delay(500); RE(); delay(500); DO(); delay(1000);
DIAM(); delay(500);
SOL(); delay(500); MI(); delay(1000); SOL(); delay(1000); //5
DIAM(); delay(500);
SOL(); delay(500); DOO(); delay(500); SOL(); delay(500); //6
FA(); delay(500); MI(); delay(500); RE(); delay(1000);
DIAM(); delay(500);
FA(); delay(500); RE(); delay(1000); FA(); delay(1000); //7
LA(); delay(1000); SOL(); delay(500); DO(); delay(500); //8
MI(); delay(500); RE(); delay(500); DO(); delay(1000);
DIAM(); delay(1000);
}
```

## ***Hasil :***

- Buat laporan hasil dari percobaan

## ***Latihan :***

1. Buatlah eksperimen dengan menggunakan kombinasi nada-nada yang lain disertai dengan tampilan LED.
2. Buatlah eksperimen dengan menggunakan kombinasi nada-nada yang membentuk sebuah lagu.

# Referensi definisi note

```
#define NOTE_B0 31
#define NOTE_C1 33
#define NOTE_CS1 35
#define NOTE_D1 37
#define NOTE_DS1 39
#define NOTE_E1 41
#define NOTE_F1 44
#define NOTE_FS1 46
#define NOTE_G1 49
#define NOTE_GS1 52
#define NOTE_A1 55
#define NOTE_AS1 58
#define NOTE_B1 62
#define NOTE_C2 65
#define NOTE_CS2 69
#define NOTE_D2 73
#define NOTE_DS2 78
#define NOTE_E2 82
#define NOTE_F2 87
#define NOTE_FS2 93
#define NOTE_G2 98
#define NOTE_GS2 104
#define NOTE_A2 110
```

```
#define NOTE_AS2 117
#define NOTE_B2 123
#define NOTE_C3 131
#define NOTE_CS3 139
#define NOTE_D3 147
#define NOTE_DS3 156
#define NOTE_E3 165
#define NOTE_F3 175
#define NOTE_FS3 185
#define NOTE_G3 196
#define NOTE_GS3 208
#define NOTE_A3 220
#define NOTE_AS3 233
#define NOTE_B3 247
#define NOTE_C4 262
#define NOTE_CS4 277
#define NOTE_D4 294
#define NOTE_DS4 311
#define NOTE_E4 330
#define NOTE_F4 349
#define NOTE_FS4 370
#define NOTE_G4 392
#define NOTE_GS4 415
```

## pitches.h

```
#define NOTE_A4 440
#define NOTE_AS4 466
#define NOTE_B4 494
#define NOTE_C5 523
#define NOTE_CS5 554
#define NOTE_D5 587
#define NOTE_DS5 622
#define NOTE_E5 659
#define NOTE_F5 698
#define NOTE_FS5 740
#define NOTE_G5 784
#define NOTE_GS5 831
#define NOTE_A5 880
#define NOTE_AS5 932
#define NOTE_B5 988
#define NOTE_C6 1047
#define NOTE_CS6 1109
#define NOTE_D6 1175
#define NOTE_DS6 1245
#define NOTE_E6 1319
#define NOTE_F6 1397
#define NOTE_FS6 1480
#define NOTE_G6 1568
```

# ***Referensi definisi note***

```
#define NOTE_GS6 1661
#define NOTE_A6 1760
#define NOTE_AS6 1865
#define NOTE_B6 1976
#define NOTE_C7 2093
#define NOTE_CS7 2217
#define NOTE_D7 2349
#define NOTE_DS7 2489
#define NOTE_E7 2637
#define NOTE_F7 2794
#define NOTE_FS7 2960
#define NOTE_G7 3136
#define NOTE_GS7 3322
#define NOTE_A7 3520
#define NOTE_AS7 3729
#define NOTE_B7 3951
#define NOTE_C8 4186
#define NOTE_CS8 4435
#define NOTE_D8 4699
#define NOTE_DS8 4978
```