

Arduino -LCD 4x20 dengan I2C

I2C LCD Basics

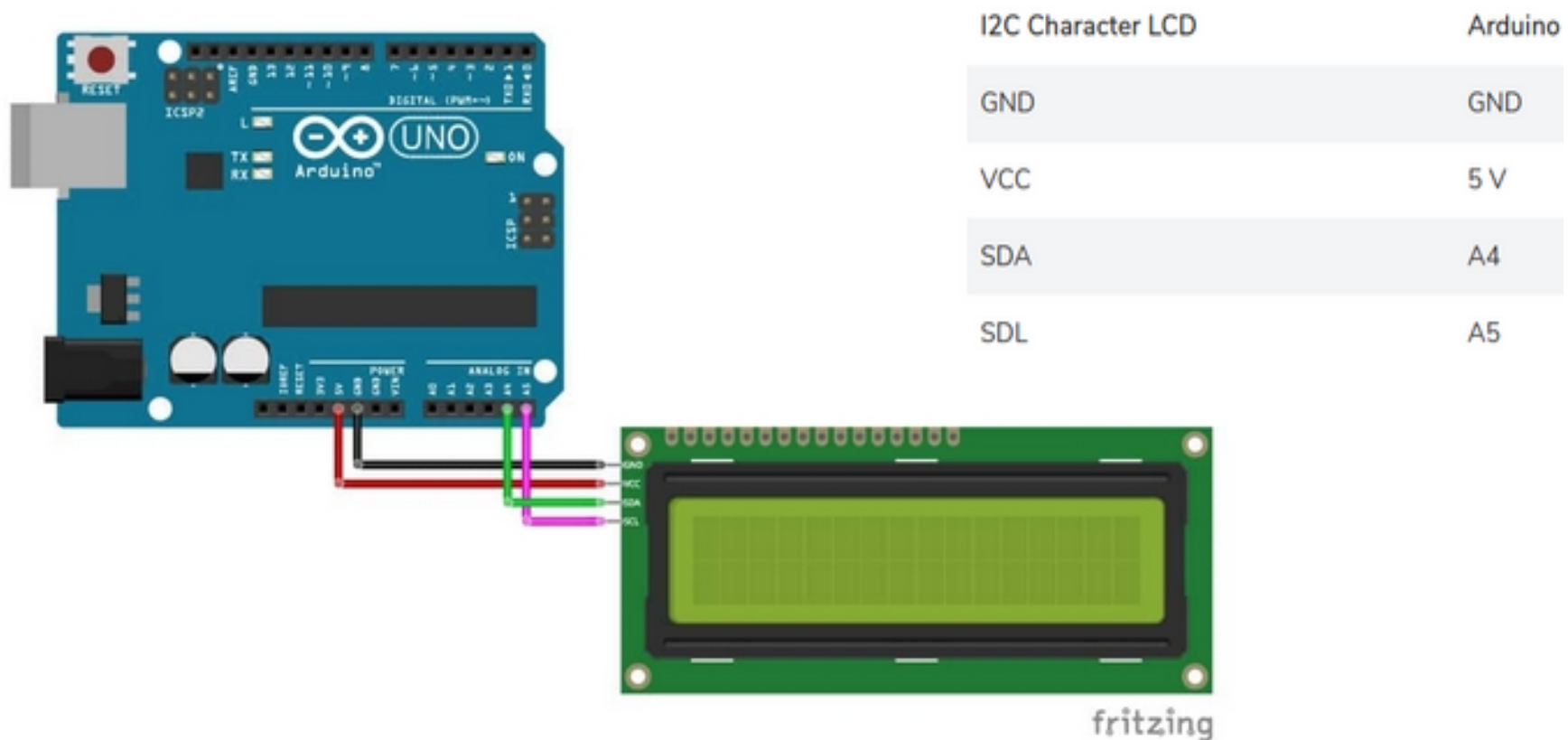
- This type of LCD is ideal for displaying text and numbers, hence the name 'character LCD'. The I2C LCDs that we are using in this tutorial come with a small add-on circuit mounted on the back of the module. This module features a PCF8574 chip (for I2C communication) and a potentiometer to adjust the LED backlight. The advantage of an I2C LCD is that the wiring is very simple. You only need two data pins to control the LCD.
- Standard LCDs typically require around 12 connections, which can be a problem if you do not have many GPIO pins available. Luckily, you can also buy the I2C add-on circuit separately on Amazon, so you can easily upgrade a standard LCD as well.

- If you look closely at the LCD, you can see the small rectangles that form the individual characters of the LCD. Each rectangle is made up of a grid of 5×8 pixels. Later in this tutorial, I will show you how you can control the individual pixels to display custom characters on the LCD.
- LCD



Bagaimana Koneksi I2C ke Arduino Uno

- Diagram kabel di bawah ini menunjukan koneksi dari I2C LCD ke Arduino. Wiring an I2C LCD is a lot easier than connecting a standard LCD.
- You only need to connect 4 pins instead of 12



Jika Tidak Pakai Arduino Uno

- Posisi I pin SDA and SCL bisa berbeda beda. SDA (data line) and SCL (clock line)

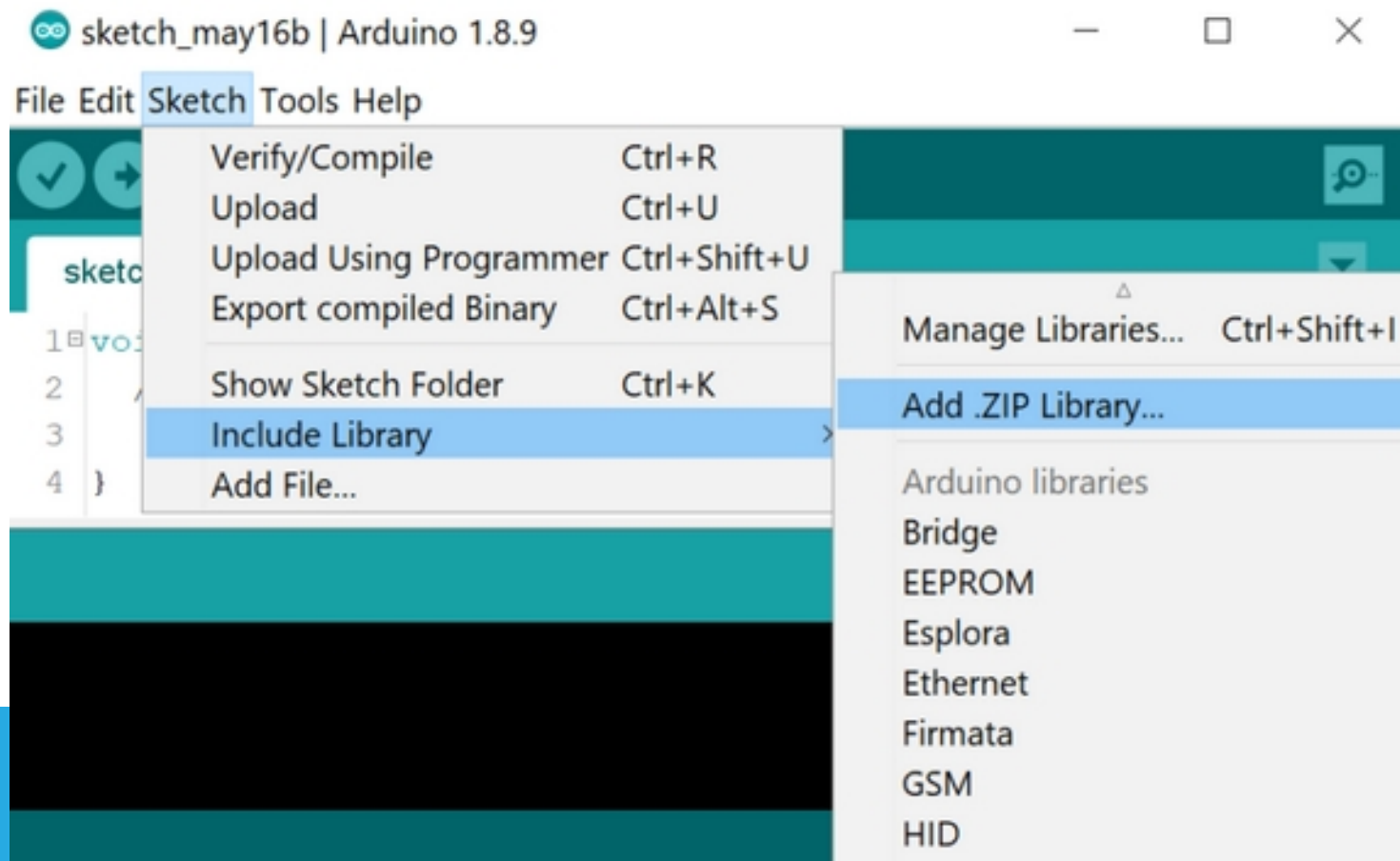
Board	SDA	SCL
Arduino Uno	A4	A5
Arduino Nano	A4	A5
Arduino Micro	2	3
Arduino Mega 2560	20	21
Arduino Leonardo	2	3
Arduino Due	20	21

Atur Intesitas Cahaya

- Setelah semua kawat terpasang ke LCD, Cahaya dapat diatur lewat modul I2C dibelakang, disitu ada potensiometer. Harus pakai obeng kembang kecil
- Pasangkan USB connector ke Arduino untuk menyalakan LCD.
- Akan dilihat cahaya menyala di LCD. Putar-putar potentiometer sampai dapat cahaya yg tepat.
- Once that is done, we can start programming the LCD

Install LiquidCrystal_I2C Arduino library

- Download library dari:
- https://github.com/johnrickman/LiquidCrystal_I2C
- Install dengan menu di Arduino IDE(Sketch):
- Sketch > Include Library > Add .ZIP Library... in the Arduino IDE.



Setiap I2C Punya Alamat LCD

- Kebanyakan I2C LCD mempunyai (alamat)address '0x27'
- Tapi bisa berbeda tergantung pabrik pembuatnya.
- Harus tahu alamat yang benar sebelum menggunakannya.

Program Mencari Address I2C

```
/*I2C_scanner
  This sketch tests standard 7-bit addresses.
  Devices with higher bit address might not be seen properly.*/

#include <Wire.h>
void setup() {
  Wire.begin();
  Serial.begin(9600);
  while (!Serial);
  Serial.println("\nI2C Scanner");
}
void loop() {
  byte error, address;
  int nDevices;
  Serial.println("Scanning...");
  nDevices = 0;
```

Lanjutan

```
for (address = 1; address < 127; address++ ) {
  Wire.beginTransaction(address);
  error = Wire.endTransmission();
  if (error == 0) {
    Serial.print("I2C device found at address 0x");
    if (address < 16)
      Serial.print("0");
    Serial.print(address, HEX);
    Serial.println("  !");
    nDevices++;
  }
  else if (error == 4) {
    Serial.print("Unknown error at address 0x");
    if (address < 16)
      Serial.print("0");
    Serial.println(address, HEX);
  }
}

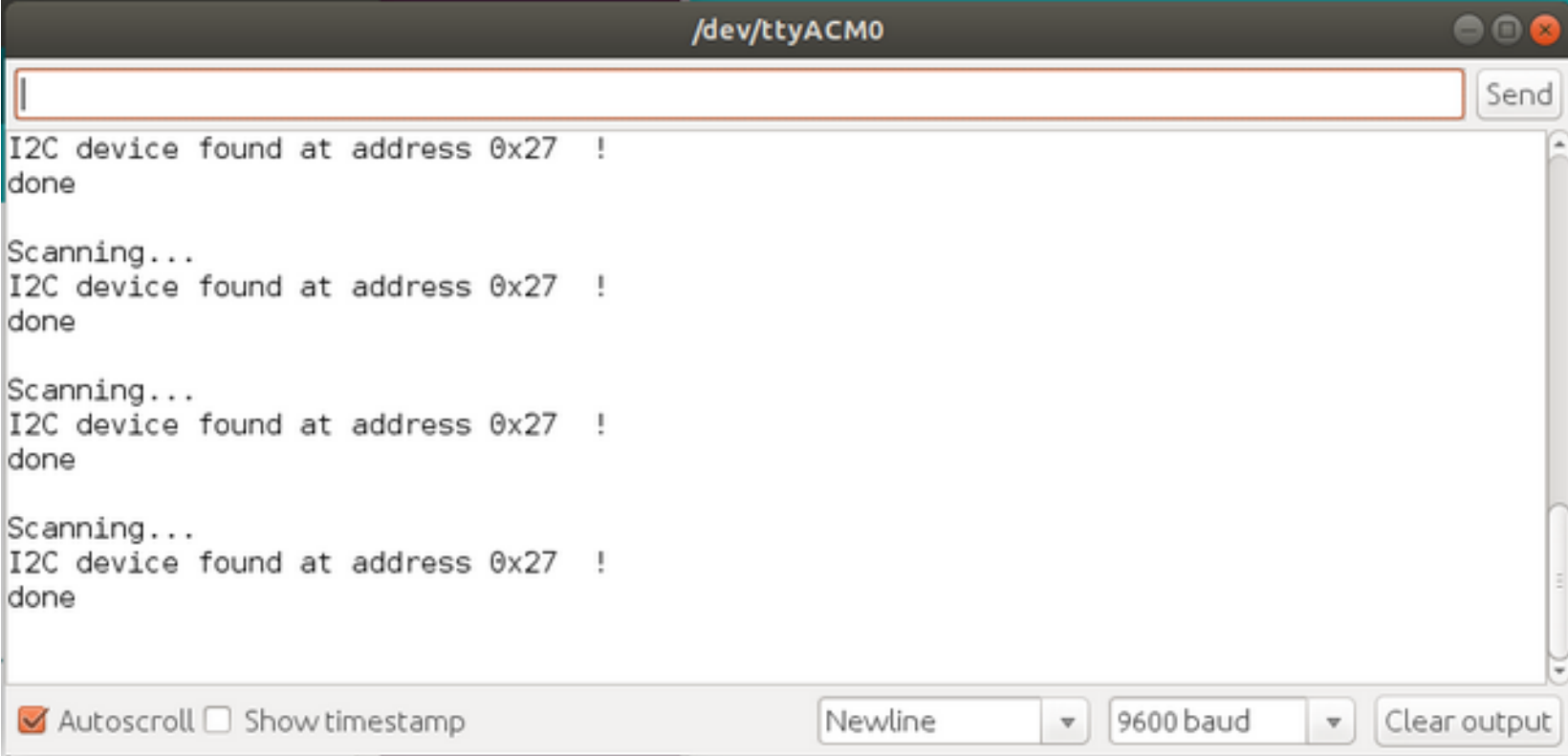
if (nDevices == 0)
  Serial.println("No I2C devices found\n");
else
  Serial.println("done\n");
delay(5000);
}
```

Bagaimana Eksekusi Program

- Jalankan serial monitor
- Execute programnya



Hasil Scanning Address LCD



```
/dev/ttyACM0
I2C device found at address 0x27 !
done

Scanning...
I2C device found at address 0x27 !
done

Scanning...
I2C device found at address 0x27 !
done

Scanning...
I2C device found at address 0x27 !
done

Autoscroll Show timestamp Newline 9600 baud Clear output
```

Ayo Coba Program Pertama

```
/* I2C LCD with Arduino example code. More info: https://www.makerguides.com */
// Include the libraries:
// LiquidCrystal_I2C.h: https://github.com/johnrickman/LiquidCrystal\_I2C
#include <Wire.h> // Library for I2C communication
#include <LiquidCrystal_I2C.h> // Library for LCD
// Wiring: SDA pin is connected to A4 and SCL pin to A5.
// Connect to LCD via I2C, default address 0x27 (A0-A2 not jumpered)
LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 20, 4); // Change to (0x27,16,2) for 16x2 LCD.
void setup() {
    // Initiate the LCD:
    lcd.init();
    lcd.backlight();
}
void loop() {
    // Print 'Hello World!' on the first line of the LCD:
    lcd.setCursor(0, 0); // Set the cursor on the first column and first row.
    lcd.print("Hello World!"); // Print the string "Hello World!"
    lcd.setCursor(2, 1); //Set cursor on third column and the second row (counting starts at 0!).
    lcd.print("LCD tutorial");
}
```

Fungsi Clear()

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
void setup() {
    lcd.init();
    lcd.backlight();
}
void loop() {
    lcd.clear();
    lcd.print("Monday"); //Cetak string "Hello World!"
    delay(2000);
    lcd.clear();
    lcd.print("13:45");
    delay(2000);
}
```

Menulis Posisi Baris Tertentu

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
void setup() {
    lcd.init();
    lcd.backlight();
}
void loop() {
    lcd.clear();
    lcd.print("Senin"); //Cetak string "Senin!"
    delay(2000);
    lcd.clear();
    lcd.print("Baris Ke 0 ");
    lcd.print("13:45");
    lcd.setCursor(0,1);
    lcd.print("Baris Ke 1");
    lcd.setCursor(0,2);
    lcd.print("Baris Ke 2");
    lcd.setCursor(0,3);
    lcd.print("Baris Ke 3");
    delay(2000);
}
```

Fungsi-fungsi lainnya

- **home()**
- Positions the cursor in the top-left corner of the LCD. Use `clear()` if you also want to clear the display.
- **cursor()**
- Displays the LCD cursor: an underscore (line) at the position of the next character to be printed.
- **noCursor()**
- Hides the LCD cursor.
- **blink()**
- Creates a blinking block style LCD cursor: a blinking rectangle at the position of the next character to be printed.
- **noBlink()**
- Disables the block style LCD cursor

Blinking Text

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 20, 4);
void setup() {
    lcd.init();
    lcd.backlight();
    lcd.print("Blinking text");
}
void loop() {
    lcd.display();
    delay(2000);
    lcd.noDisplay();
    delay(2000);
}
```

Fungsi-fungsi lainnya

- **display()**

This function turns on the LCD screen and displays any text or cursors that have been printed to the display.

- **noDisplay()**

This function turns off any text or cursors printed to the LCD. The text/data is not cleared from the LCD memory. This means it will be shown again when the function `display()` is called.

- **write()**

This function can be used to write a character to the LCD. See the section about creating and displaying custom characters below for more info.

- **scrollDisplayLeft()**

Scrolls the contents of the display (text and cursor) one space to the left. You can use this function in the loop section of the code in combination with `delay(500)`, to create a scrolling text animation.

- **scrollDisplayRight()**

Scrolls the contents of the display (text and cursor) one space to the right.

- **autoscroll()**

This function turns on automatic scrolling of the LCD. This causes each character output to the display to push previous characters over by one space. If the current text direction is left-to-right (the default), the display scrolls to the left, if the current direction is right-to-left, the display scrolls to the right. This has the effect of outputting each new character to the same location on the LCD.

Geser Kiri/Kanan

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 20, 4);
void setup() {
    lcd.init();
    lcd.backlight();
    lcd.print("Kumaha Daramang!");
    lcd.setCursor(0, 1);
    lcd.print("Muter Kamana?");
    lcd.setCursor(0, 2);
    lcd.print("Baris no 2");
}
void loop() {

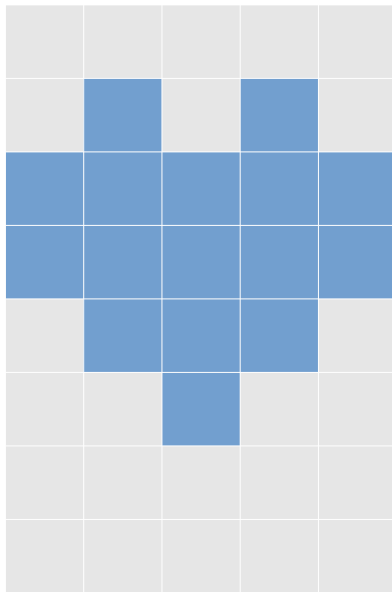
    //lcd.scrollDisplayLeft(); // Geser ke kiri
    lcd.scrollDisplayRight(); // Geser ke kanan
    delay(500);
}
```

Autoscroll

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 20, 4);
void setup() {
    lcd.init();
    lcd.backlight();
}
void loop() {
    lcd.autoscroll();
    lcd.setCursor(20, 0);
    for (int x = 0; x < 14; x++) {
        lcd.print(x);
        delay(500);
    }
    lcd.clear();
}
```

Custom Karakter

Seperti menggambar di kotak-kotak 5x8
Kemudian kotak yang hitam (gelap) diberi angka satu(1) dan
Yang terang diberi angka nol (0)



```
byte Heart[] = {  
    B00000,  
    B01010,  
    B11111,  
    B11111,  
    B01110,  
    B00100,  
    B00000,  
    B00000  
};
```

Ini teh Gambar Hati

Custom Karakter

```
/* display custom characters on I2C character LCD. More
info: www.makerguides.com */

// Include the library:

#include <LiquidCrystal_I2C.h>
// Create lcd object of class LiquidCrystal_I2C:
LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 20, 4);
// Make custom characters:
byte Heart[] = {
B00000,
B01010,
B11111,
B11111,
B01110,
B00100,
B00000,
B00000
};
byte Lock[] = {
B01110,
B10001,
B10001,
B11111,
B11011,
B11011,
B11111,
B00000
};
```

```
byte Bell[] = {
B00100,
B01110,
B01110,
B01110,
B11111,
B00000,
B00100,
B00000
};
byte Alien[] = {
B11111,
B10101,
B11111,
B11111,
B01110,
B01010,
B11011,
B00000
};
byte Check[] = {
B00000,
B00001,
B00011,
B10110,
B11100,
B01000,
B00000,
B00000
};

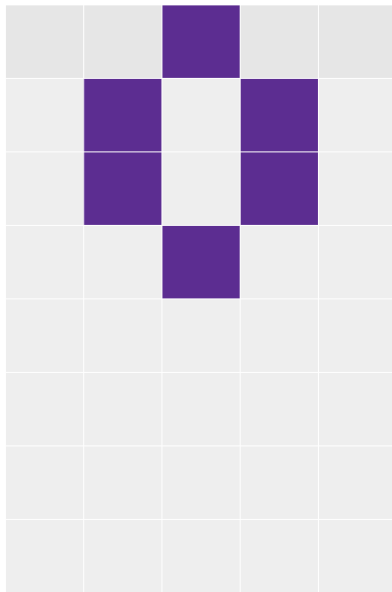
byte Speaker[] = {
B00001,
B00011,
B01111,
B01111,
B01111,
B00011,
B00001,
B00000
};
byte Sound[] = {
B00001,
B00011,
B00101,
B01001,
B01001,
B01011,
B11011,
B11000
};
byte Skull[] = {
B00000,
B01110,
B10101,
B11011,
B01110,
B01110,
B00000,
B00000
};
```

Custom Karakter

```
void setup() {
  // Initialize LCD and turn on the backlight:
  lcd.init();
  lcd.backlight();
  // Create new characters:
  lcd.createChar(0, Heart);
  lcd.createChar(1, Bell);
  lcd.createChar(2, Alien);
  lcd.createChar(3, Check);
  lcd.createChar(4, Speaker);
  lcd.createChar(5, Sound);
  lcd.createChar(6, Skull);
  lcd.createChar(7, Lock);
  // Clear the LCD screen:
  lcd.clear();
  // Print a message to the lcd:
  lcd.print("Custom Character");
}
// Print all the custom characters:
void loop() {
  lcd.setCursor(0, 1);
  lcd.write(0);
  lcd.setCursor(2, 1);
  lcd.write(1);
  lcd.setCursor(4, 1);
  lcd.write(2);
  lcd.setCursor(6, 1);
  lcd.write(3);
  lcd.setCursor(8, 1);
  lcd.write(4);
  lcd.setCursor(10, 1);
  lcd.write(5);
  lcd.setCursor(12, 1);
  lcd.write(6);
  lcd.setCursor(14, 1);
  lcd.write(7);
}
```

Custom Karakter °C

Seperti menggambar di kotak-kotak 5x8
Kemudian kotak yang hitam (gelap) diberi angka satu(1) dan
Yang terang diberi angka nol (0)



```
byte Drajat[] = {  
    B00100,  
    B01010,  
    B01010,  
    B00100,  
    B00000,  
    B00000,  
    B00000,  
    B00000  
};
```

Ini Derajat