

Workshop Aplikasi Mikroprosesor & Antarmuka

**PROGRAM STUDI
TEKNIK TELKOMUNIKASI**

Akuwan Saleh, MT

PENILAIAN

⇒ Laporan + Tugas + Presentasi = **60%**

eval-1(Lap.1-5) = 20%

eval-2(Lap.6-10) = 20%

eval-3(Lap.11-15 & (PPT+ presentasi)) = 20%

⇒ TPS = **40%**

eval-4 (TPS = Tugas Proyek Semester)

REFERENSI

- Rui Santos & Sara Santos, “ESP32 Web Server With Arduino IDE.pdf: Step By Step Project Guide”, <https://randomnerdtutorials.com/esp32-web-server-arduino-ide/>, juni 2020.
- Julien Bayle, “C Programming for Arduino”, Packt Publishing Ltd, Birmingham, May 2013
- Fabian Winkler, “Arduino/Processing Communication Workshop”, Fall, 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”, Springer, New York, 2011.
- Casey Reas and Ben Fry, “Getting Started with Processing”, O’Reilly Media, Inc., June 2010.
- Joshua Noble, “Programming Interactivity: A Designer’s Guide to Processing, Arduino, and openFrameworks”, O’Reilly Media, Inc., July 2009.
- Peter Hoddie, Lizzie Prader, “IoT Development for ESP32 and ESP8266 with JavaScript; A Practical Guide to XS and Moddable SDK”, Apress, Menlo Park, CA,USA, 2020.

MATERI

PENDAHULUAN

1. KOMUNIKASI MIKROKONTROLER DENGAN SOFTWARE PROCESSING
2. ANALOG INPUT DAN AUDIO PROCESSING
3. KONTROL MULTI LED MENGGUNAKAN ARDUINO DAN PROCESSING
4. PENCAMPUR WARNA VIRTUAL MENGGUNAKAN ARDUINO DAN PROCESSING
5. **MONITORING SUHU DENGAN ARDUINO DAN PROCESSING**
6. MONITORING INTENSITAS CAHAYA DENGAN ARDUINO DAN PROCESSING
7. KONTROL MOTOR DC MENGGUNAKAN ARDUINO DAN PROCESSING
8. APLIKASI SENSOR ULTRASONIC MENGGUNAKAN ARDUINO DAN PROCESSING

MATERI

9. KONTROL LAMPU AC 220 V BERBASIS ARDUINO DAN PROCESSING
10. MODUL WiFi ESP32 DENGAN ARDUINO IDE
11. KOMUNIKASI NIRKABEL MENGGUNAKAN MODUL RF 434 MHz DAN PROCESSING
12. ESP32 WEB SERVER UNTUK KONTROL LED DAN MENAMPILKAN GAMBAR
13. ANTARMUKA MODUL GPS DENGAN MIKROKONTROLER DAN PROCESSING
14. ESP32 WEB SERVER UNTUK PENGUKURAN SUHU DAN KELEMBABAN
15. KOMUNIKASI DATA BERBASIS BLUETOOTH DAN HP

5. MONITORING SUHU DENGAN ARDUINO DAN PROCESSING

TUJUAN

- Membaca nilai Suhu menggunakan sensor LM35 dan arduino
- Memantau perubahan nilai suhu dan menampilkan dalam bentuk grafis menggunakan processing

DASAR TEORI

- Membaca nilai suhu dan mengkonversikan kedalam nilai digital
- Menghitung nilai temperatur/suhu :

Cara 1: → **Suhu = input analog x 0.488**

Cara 2:

Suhu = (5.0 * analogRead(tempPin) * 100.0) / 1024;

1 C = 10 mV / 4.88 mV

1 C = 2.0491

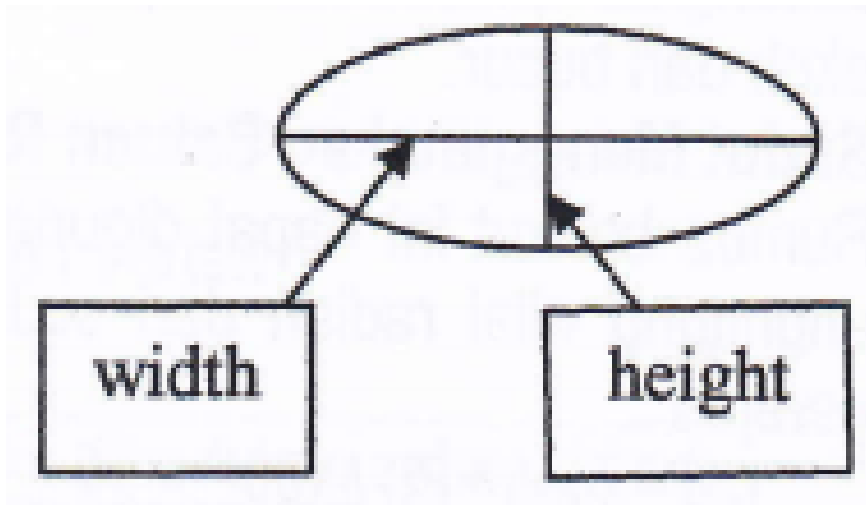


Cara 3:

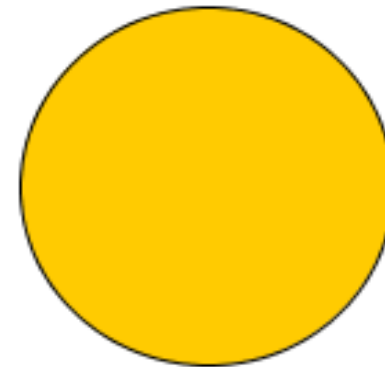
Suhu = input analog / 2.0491

- Menampilkan setiap perubahan nilai suhu derajat Celcius ke bentuk grafis
- Sehingga dapat memantau setiap perubahan nilai suhu melalui perubahan dari grafis tersebut
- Salah satu bentuk grafis adalah obyek elips (lingkaran/oval)
- Untuk membuat lingkaran processing menggunakan perintah : ***ellipse(x,y,width, height)***
- Parameter **x** dan **y** menentukan **lokasi ellips**, Parameter **width** menyatakan **lebar elips** dan parameter **height** menyatakan **tinggi elips**,

- Apabila width dan height bernilai sama maka akan diperoleh sebuah lingkaran.



Parameter ellipse



Ellipse/lingkaran

- Parameter x dan y menentukan lokasi elips dipengaruhi oleh perintah ***ellipseMode()***, Perintah ***ellipseMode()*** mempunyai sintak :

ellipseMode(mode)

- Mode berisi CORNER yang berarti parameter (x,y) menyatakan lokasi pojok kiri atas dari ellipse dan mode berisi CENTER berarti parameter (x,y) menyatakan lokasi titik tengah dari ellipsis.
- Processing secara baku menggunakan mode CENTER

Perintah noStroke() dan Stroke()

- Perintah **noStroke()** dapat diberikan sebelum perintah ellipse() untuk menggambar elips tanpa garis pinggir.
- Perintah **stroke()** juga dapat digunakan mengatur besarnya garis pinggir.

Mengisi ellips

- Elips dapat diisi dengan warna menggunakan perintah `fill()`. Perintah `fill()` mempunyai sintak sebagai berikut:

`fill(gray)`

`fill(gray,alpha)`

`fill(color)`

`fill(value1, value2, value3)`

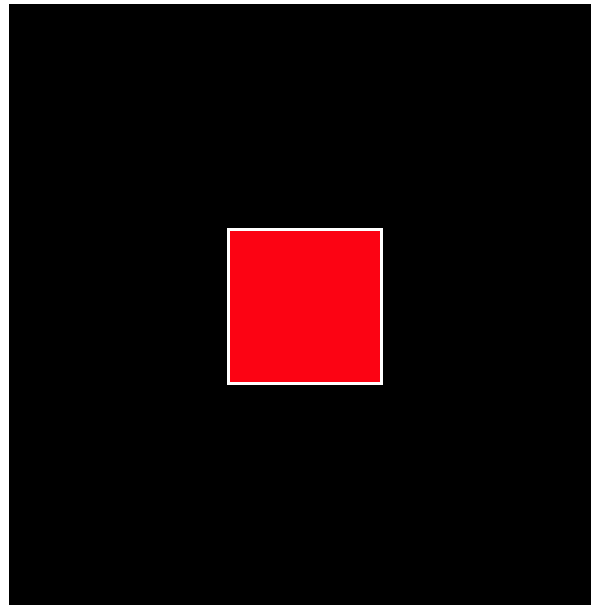
`fill(value1, value2, value3,alpha)`

- Untuk membatalkan perintah `fill()` dengan perintah `noFill()`.

- Selain lingkaran grafis yang lain adalah **Persegi Panjang**
- Perintah untuk menggambar persegi panjang adalah : ***rect(x1, y1, width,height)***
- Dengan x1 dan y1 menyatakan lokasi dari persegi panjang, width menyatakan lebar (mendatar) dan height menyatakan tinggi (vertikal).
- Lokasi persegi panjang diatur menggunakan perintah : ***rectMode(mode)*** dengan mode berisi CENTER berarti x1 dan y1 menyatakan titik tengah persegi panjang dan mode berisi CORNER berarti x1 dan y1 menyatakan titik sudut kiri atas dari persegi panjang dan mode berisi CORNERS berarti

x1, y1 menyatakan salah satu titik sudut dan x2, y2 menyatakan lokasi sudut yang berlawanan.

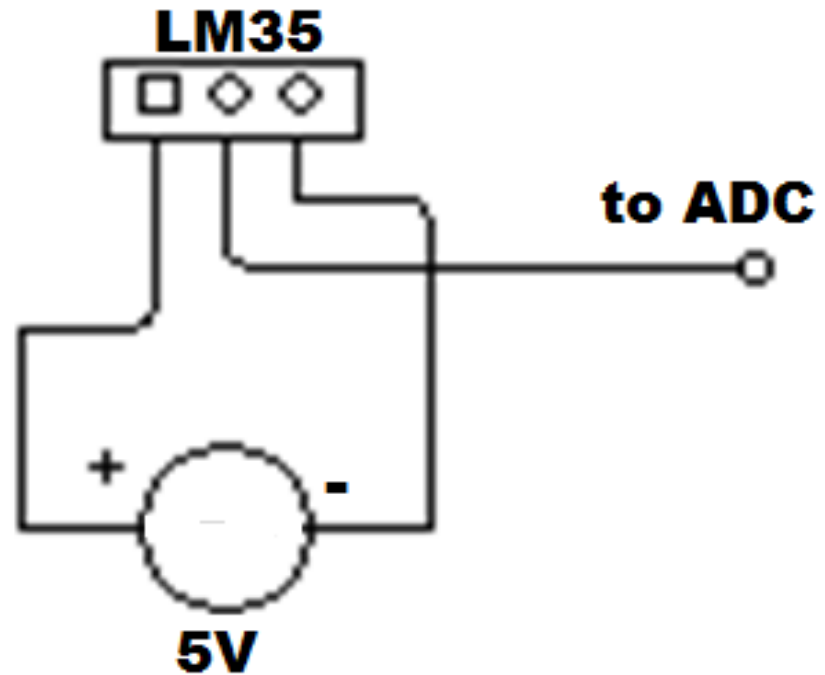
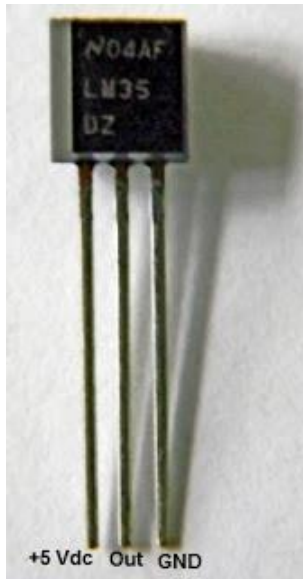
- Contoh: `background (0);`
`stroke (255);`
`fill (252, 3, 19);`
`rectMode (CENTER);`
`rect (width/2, height/2, 50, 50);`



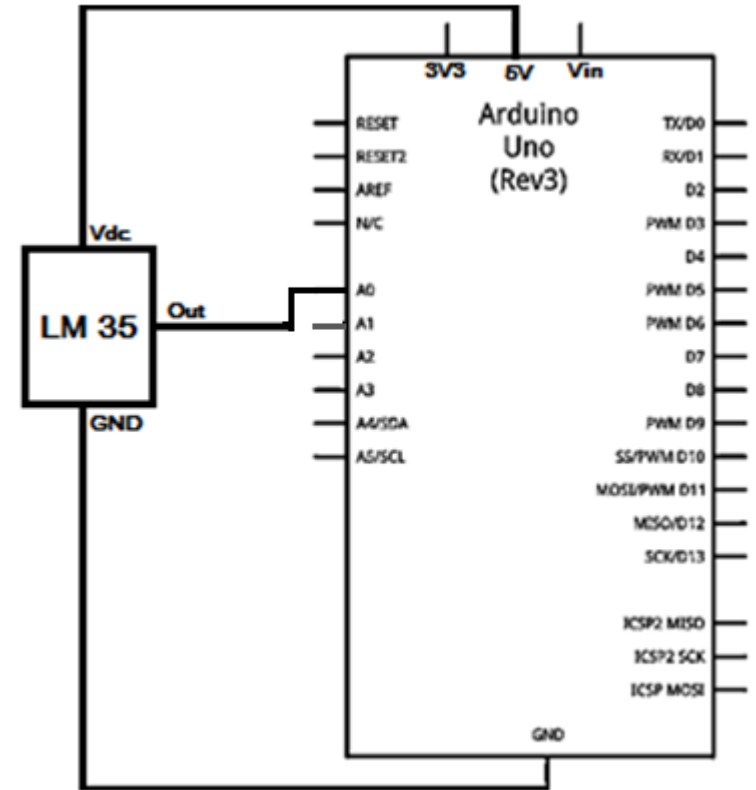
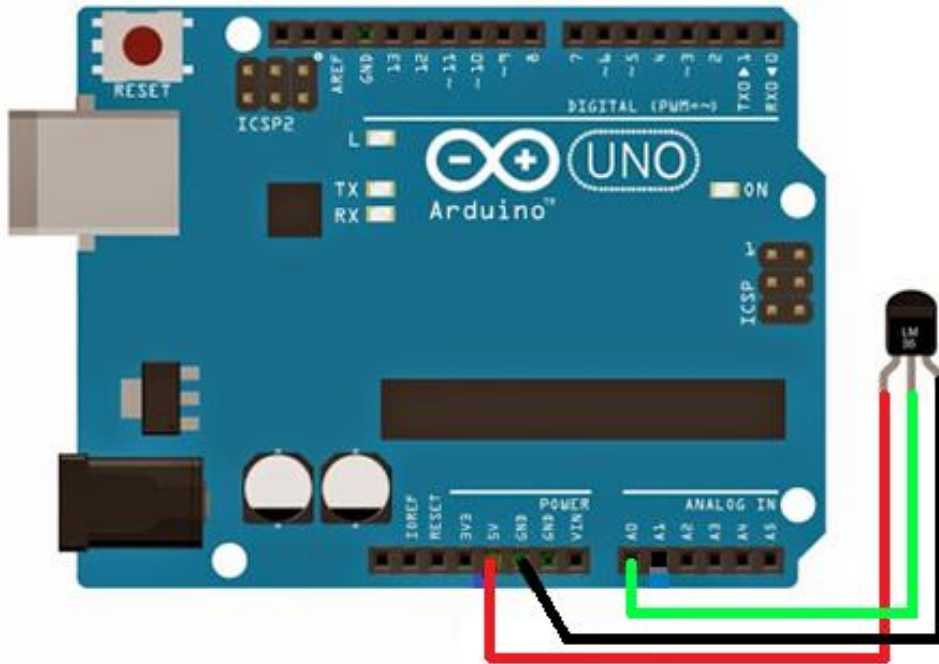
Alat & Bahan :

- Arduino Uno Board
- 1x LM35 sensor suhu
- Kabel jumper
- Software Processing

Skema Rangkaian :



Rangkaian :



PROGRAM-1:

Arduino Uno

```
float tempC;  
int sensorPin = 0;  
void setup()  
{  
  Serial.begin(9600);  
}  
void loop()  
{  
  tempC = analogRead(sensorPin);  
  tempC = (5.0 * tempC * 100.0)/1024.0;  
  Serial.println();  
  Serial.print(tempC);  
  delay(1000);  
}
```

PROGRAM-1:

Processing

```
import processing.serial.*;
Serial commPort;
float tempC;
float tempF;
int yDist;
float[] tempHistory = new float[100];

void setup()
{
  size(360,320);
  commPort = new Serial(this, Serial.list()[0], 9600);
  for(int index = 0; index<100; index++)
    tempHistory[index] = 0;
}

void draw()
{
  while (commPort.available() > 0)
  {
    tempC = commPort.read();
```

PROGRAM-1: (lanjutan)

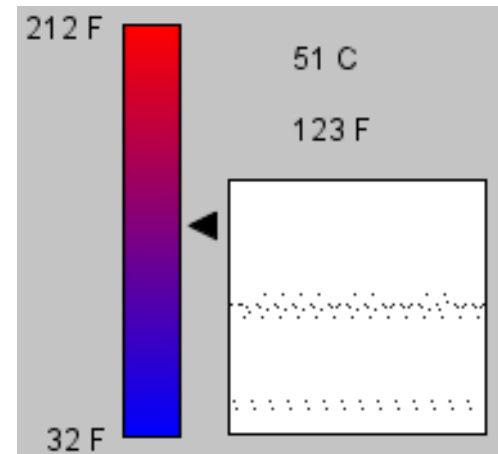
Processing

```
background(123);
colorMode(RGB, 160); //use color mode sized for fading
stroke (0);
rect (49,19,22,162);
for (int colorIndex = 0; colorIndex <= 160; colorIndex++)
{
  stroke(160 - colorIndex, 0, colorIndex);
  line(50, colorIndex + 20, 70, colorIndex + 20);
}
stroke(0);
fill(255,255,255);
rect(90,80,100,100);
for (int index = 0; index<100; index++)
{
  if(index == 99)
    tempHistory[index] = tempC;
  else
    tempHistory[index] = tempHistory[index + 1];
}
```

PROGRAM-1: (lanjutan)

Processing

```
    point(90 + index, 180 - tempHistory[index]);
}
fill(0,0,0);
textAlign(RIGHT);
text("212 F", 45, 25);
text("32 F", 45, 187);
yDist = int(160 - (160 * (tempC * 0.01)));
stroke(0);
triangle(75, yDist + 20, 85, yDist + 15, 85, yDist + 25);
fill(0,0,0);
textAlign(LEFT);
text(str(int(tempC)) + " C", 115, 37);
tempF = ((tempC*9)/5) + 32;
text(str(int(tempF)) + " F", 115, 65);
}
}
```



PROGRAM-2:

Arduino Uno

```
float temp;  
int tempPin = 0;  
void setup()  
{  
  Serial.begin(9600);  
}  
void loop()  
{  
  temp = analogRead(tempPin);  
  temp = temp * 0.48828125;  
  Serial.println(temp);  
  delay(1000);  
}
```

PROGRAM-2:

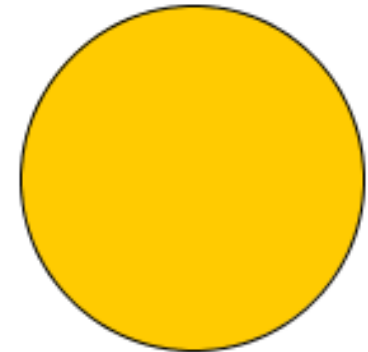
Processing

```
import processing.serial.*;  
int lf = 10;  
String myString = null;  
Serial myPort;  
float num;  
  
void setup() {  
  background(255,255,255);  
  size(320,240);  
  frameRate(120);  
  myPort = new Serial(this, Serial.list()[0], 9600);  
  myPort.clear();  
}
```

PROGRAM-2: (lanjutan)

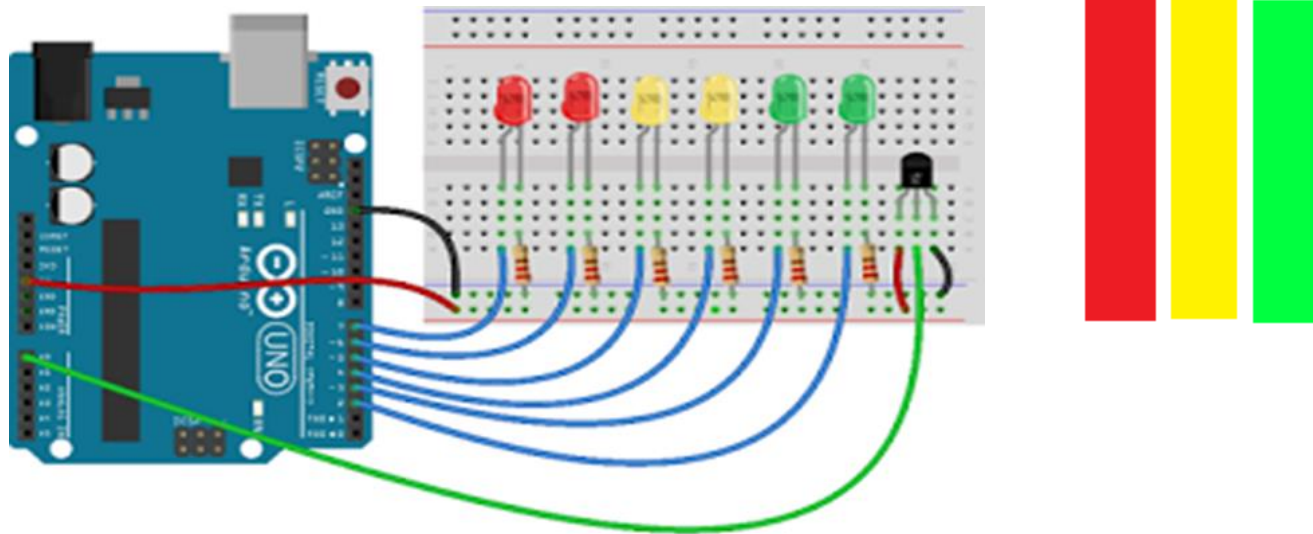
Processing

```
void draw() {  
  while (myPort.available() > 0) {  
    myString = myPort.readStringUntil(lf);  
    if (myString != null) {  
      print(myString);  
      num=float(myString);  
      println(num);  
      println(frameCount);  
      background(255,255,255);  
      fill(#ffcb01);  
      ellipse(160,120,num*4,num*4);  
    }  
  }  
  myPort.clear();  
}
```



Latihan :

1. Buat aplikasi menggunakan rangkaian di bawah untuk menampilkan nilai suhu pada LED dan grafik berupa 3 persegi panjang dengan warna merah kuning dan hijau.



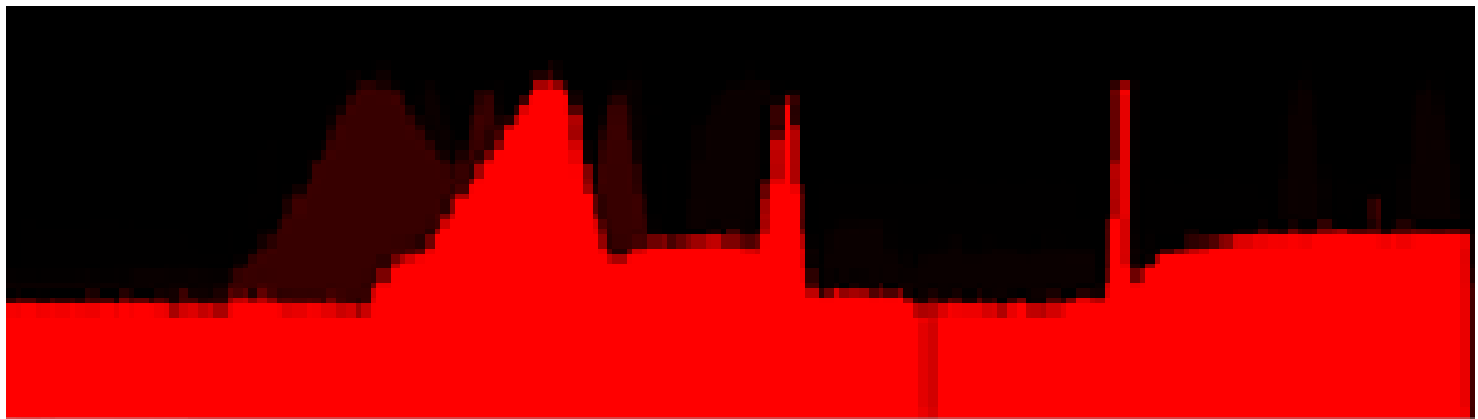
2. Menggunakan rangkaian pada soal no.1 untuk menampilkan nilai suhu pada LED dan grafik seperti di samping.



Latihan :

3. Buat aplikasi menggunakan rangkaian pada soal no.1 untuk menampilkan nilai suhu pada LED dan grafik sinyal.

Data suhu dari Input Analog



APLIKASI

- Membaca perubahan nilai suhu karena adanya hujan
- Mendeteksi perubahan cuaca menggunakan sensor

Kebutuhan Sensor

✓ Sensor Hujan



Rain



Rain Sensor



Lower Output Voltage

✓ Sensor Temperatur



+Vs OUT GND

LM-35



DHT11
temperature-humidity
sensor



Sun



Rain Sensor

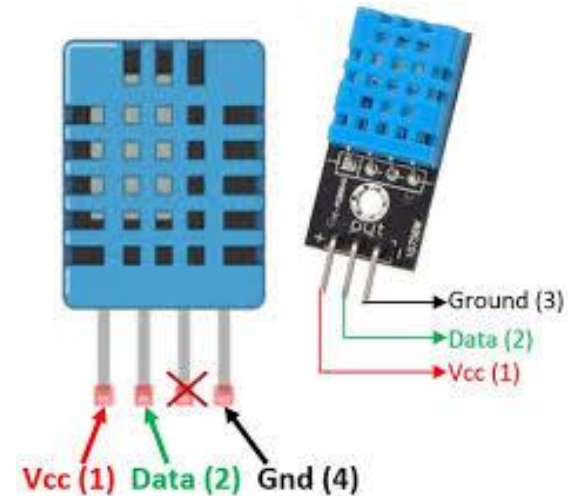


Higher Output Voltage

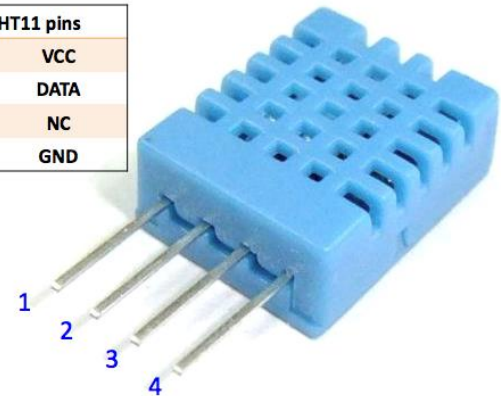
Kebutuhan Sensor

✓ DHT11 temperature-humidity sensor

- Ultra low cost
- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 20-80% humidity readings with 5% accuracy
- Good for 0-50°C temperature readings $\pm 2^{\circ}\text{C}$ accuracy
- No more than 1 Hz sampling rate (once every second)
- Body size 15.5mm x 12mm x 5.5mm
- 4 pins with 0.1" spacing

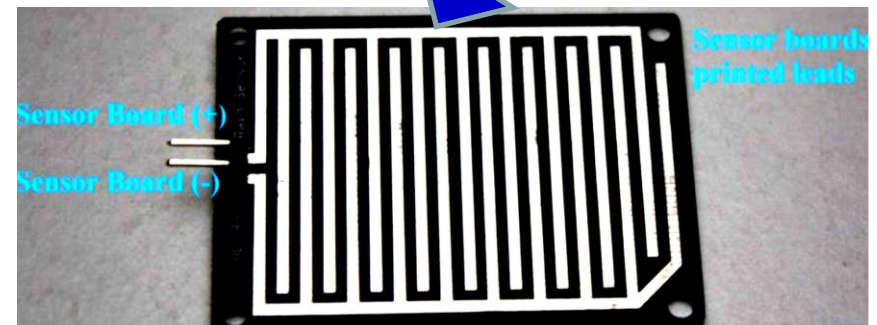
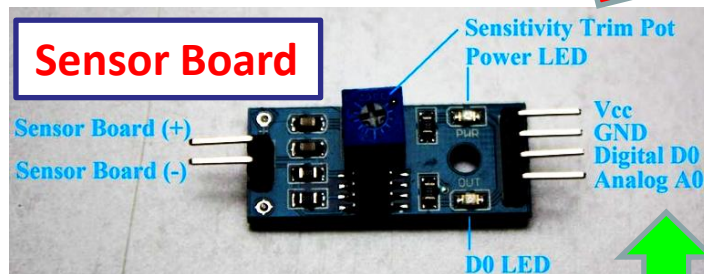
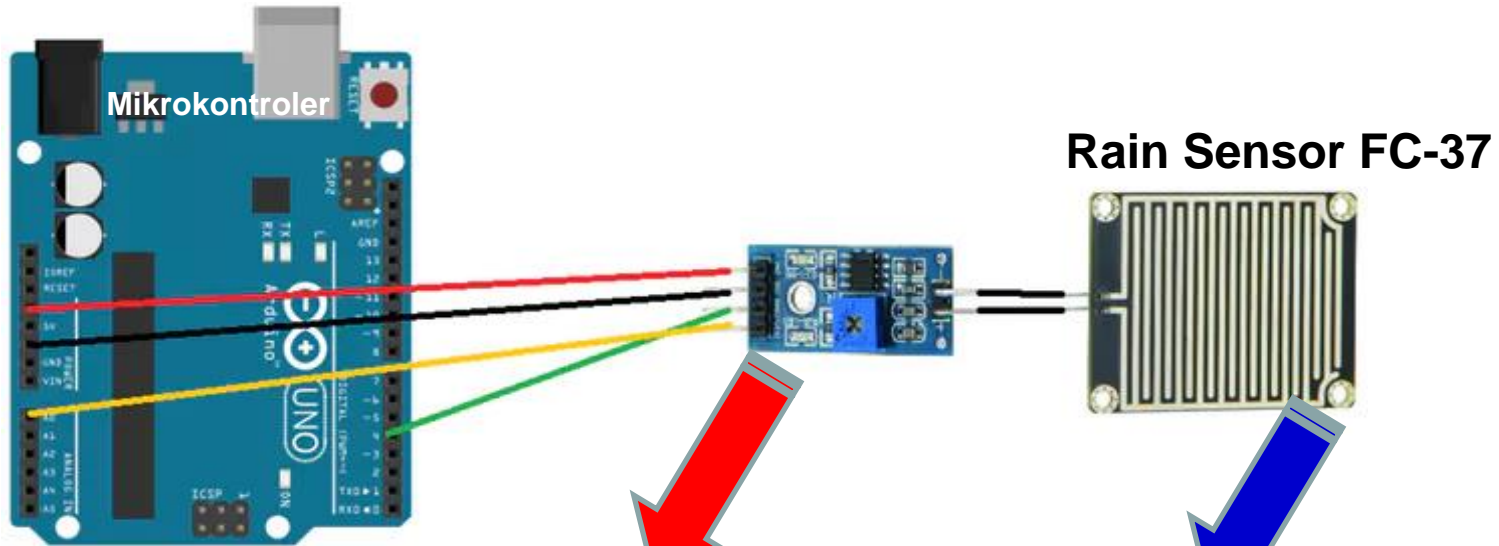


DHT11 pins	
1	VCC
2	DATA
3	NC
4	GND



Rangkaian Antarmuka-1

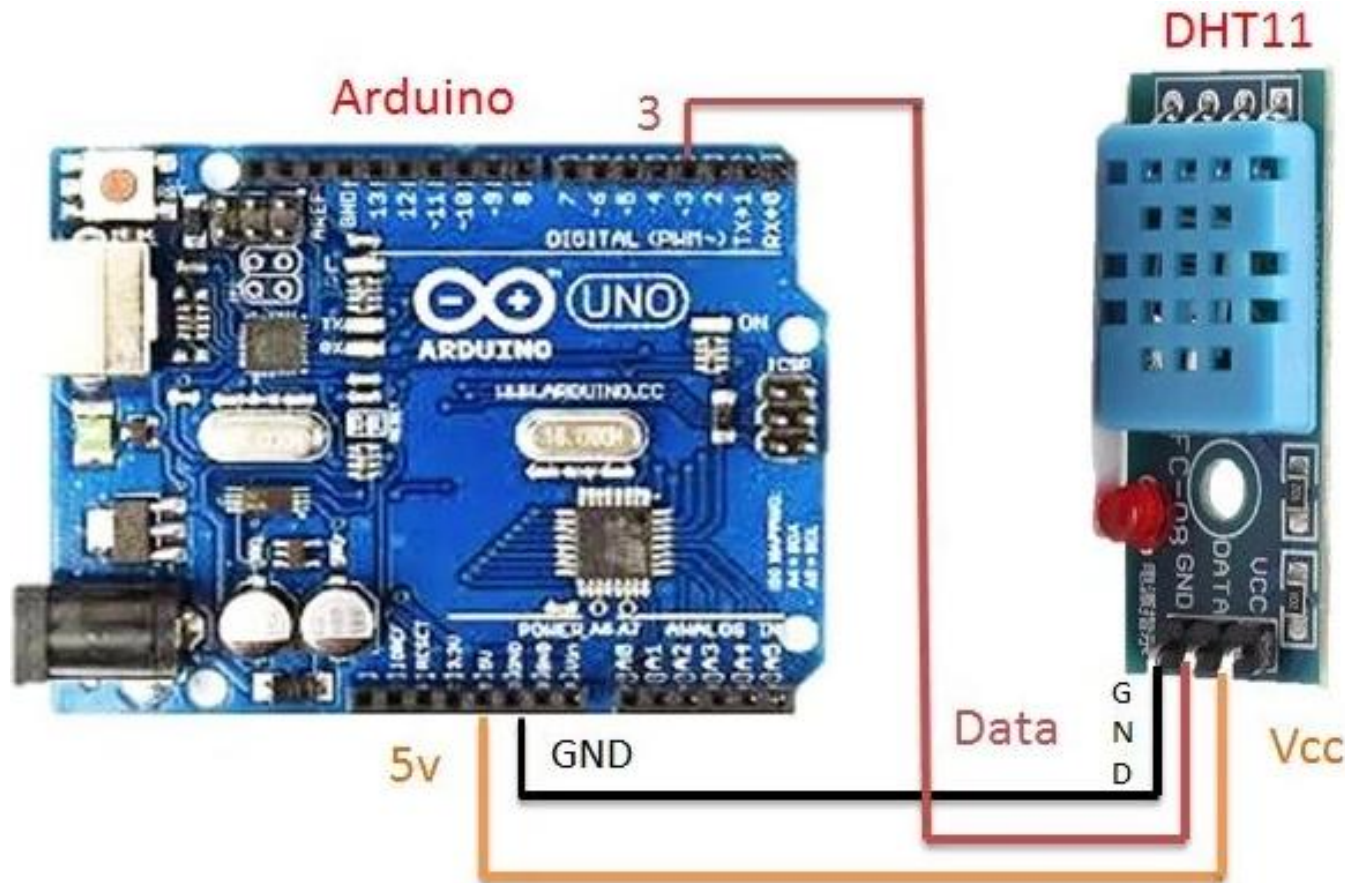
Mikrokontroler - Rain Sensor



Arduino:
5V --> VCC
GND --> GND
D0 --> D4
A0 --> A0

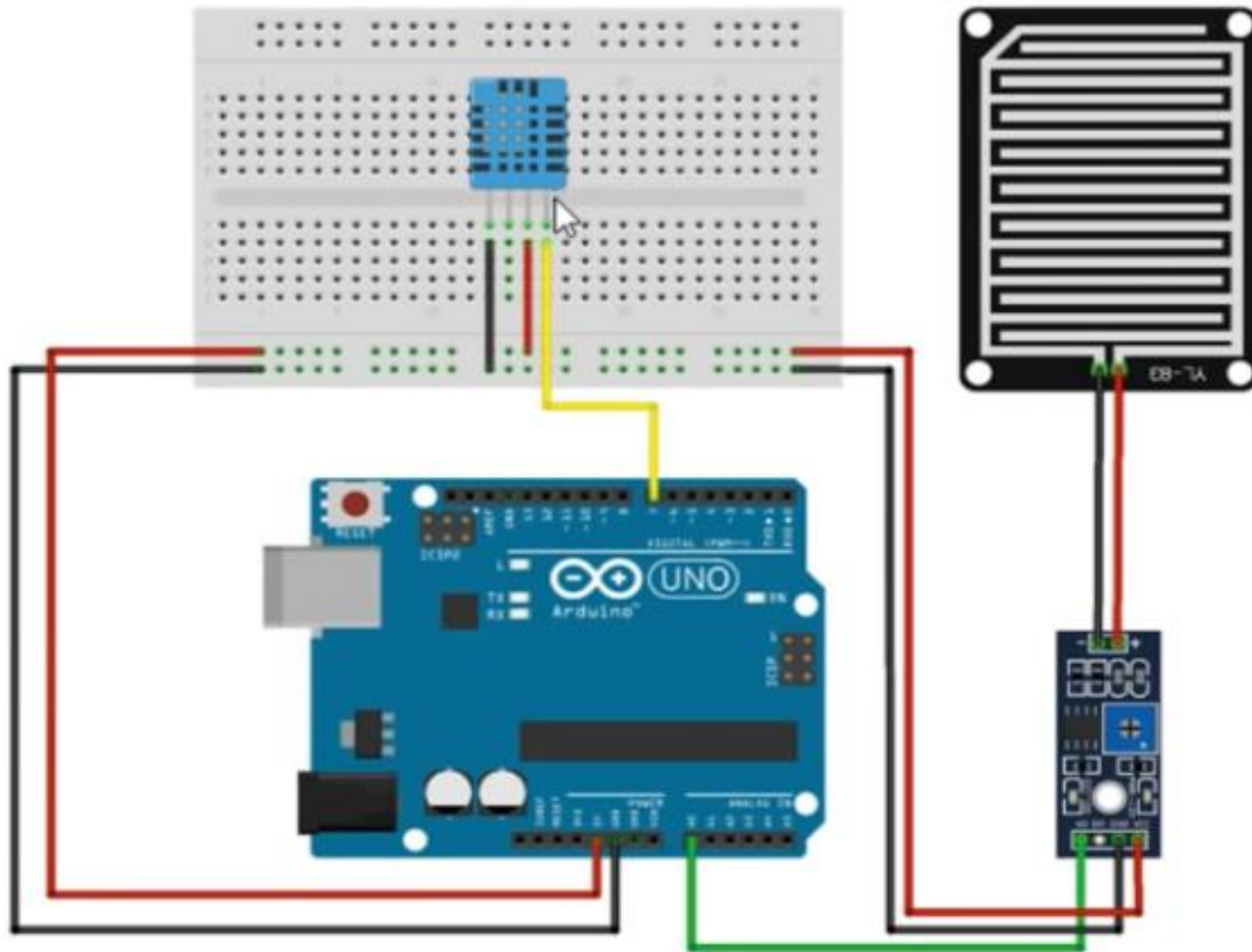
Rangkaian Antarmuka-2

Mikrokontroler – Humidity/temperature Sensor



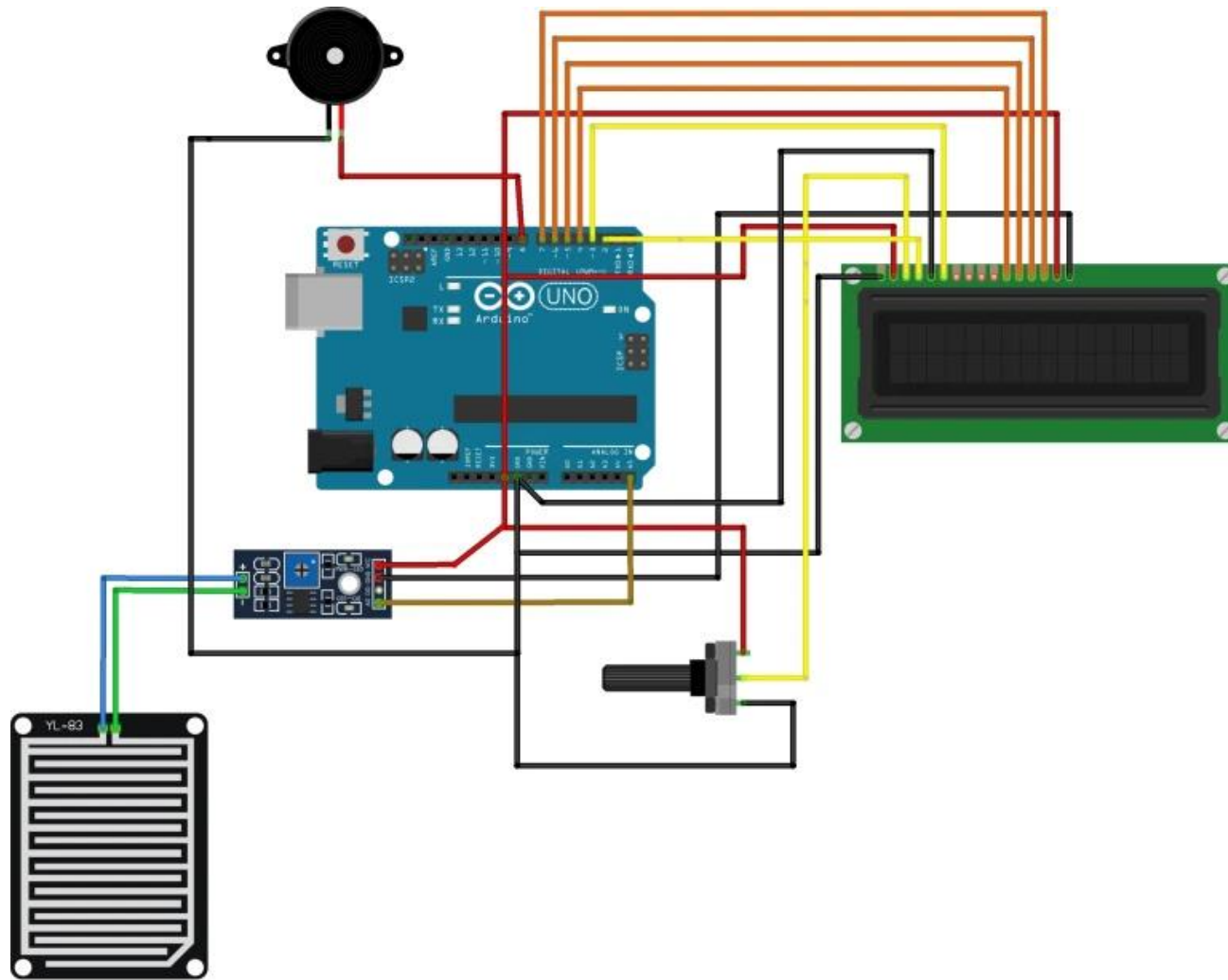
Rangkaian Antarmuka-3

Mendeteksi perubahan cuaca



Rangkaian Antarmuka-4

Mendeteksi perubahan cuaca



Latihan :

1. Buatlah rangkaian antarmuka mikrokontroler Arduino dengan sensor temperature/humidity, sensor hujan dan software processing.
2. Dari rangkaian No.1, buatlah aplikasi monitoring perubahan nilai suhu karena adanya hujan menggunakan sensor temperature dan sensor hujan.