

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

4. PENCAMPUR WARNA VIRTUAL MENGGUNAKAN ARDUINO DAN PROCESSING

TUJUAN

- Mengontrol Warna LED RGB dengan Arduino dan Processing.
- Membuat antarmuka pencampuran warna (merah, hijau dan biru) menggunakan potensiometer.
- Mengendalikan Warna LED RGB menggunakan keyboard, dengan mengendalikan tegangan masukan analog input Arduino.

DASAR TEORI

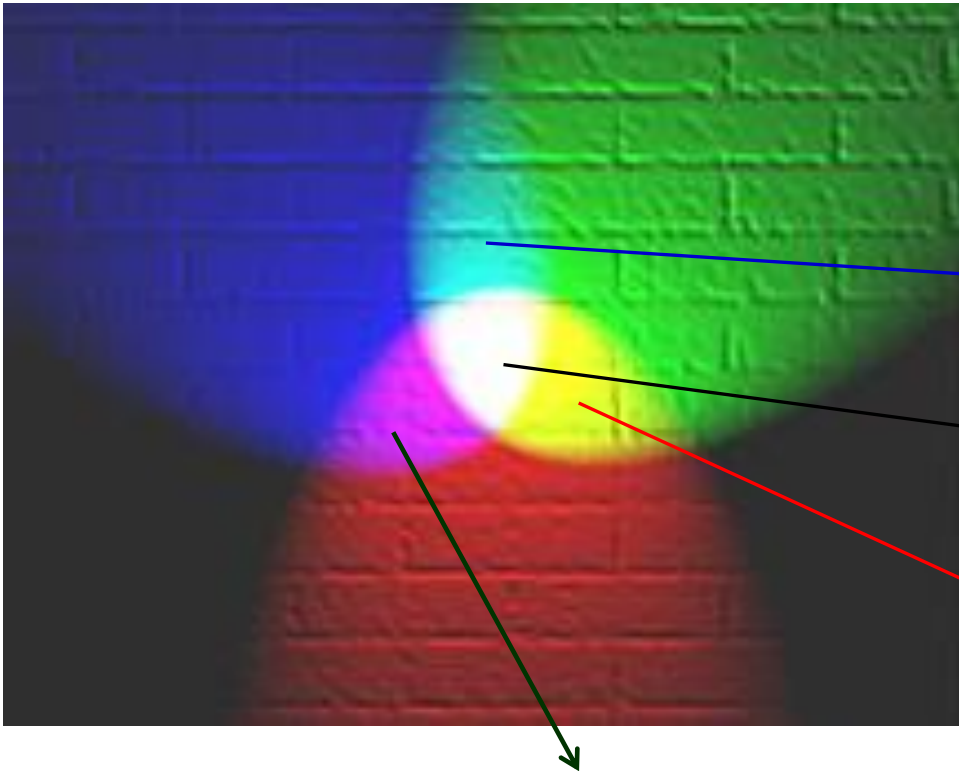
- **RGB Value** (Red, Green, Blue Value).
- **RGB Value** : semua warna bisa dibuat dengan mencampur antara komponen warna merah, biru, dan hijau dengan berbagai variasi proporsi.
- Ketiga nilai ini memiliki rentang dari 0 - 255, dengan 0 = tidak ada sama sekali dari warna tersebut, dan 255 = penuh dengan komponen warna tersebut.

Contoh: merah RGB Value= [255 0 0] dan putih = [255 255 255], hitam = [0 0 0].

DASAR TEORI

- Kombinasi LED RGB terlihat begitu menarik sehingga dibutuhkan kemampuan untuk mencampur warna dan membuat warna pixel.
- Pencahayaan yang berbeda dari setiap LED RGB akan menyulitkan proses pembuatan warna yang diinginkan.
- Sebagai contoh bagaimana untuk menghasilkan warna magenta dengan memberikan nilai pada pin biru dan merah.

DASAR TEORI



Kombinasi LED RGB

Warna cyan:

→ RGB=[0 255 255]

Warna putih:

→ RGB=[255 255 255]

Warna kuning:

→ RGB=[255 255 0]

Warna magenta :

RGB=[255 0 255]

DASAR TEORI

- Penggunaan Arduino dan Processing untuk mengendalikan input tegangan masuk ke RGB sehingga bisa mencampur jumlah warna merah, hijau dan biru yang sempurna untuk mendapatkan warna yang diinginkan.
- Mengendalikan menggunakan keyboard, dengan mengendalikan tegangan masukan dengan tepat (tegangan menuju ke Arduino).

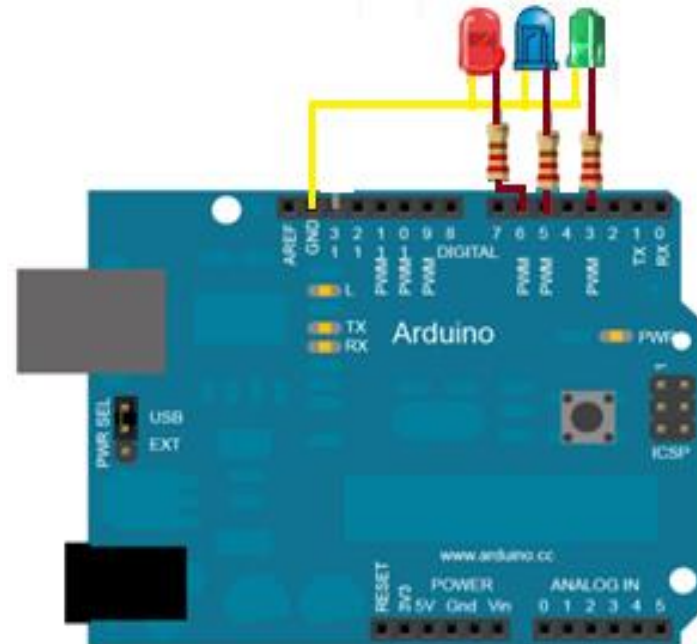
DASAR TEORI

- Skema sangat sederhana yaitu dengan menyambungkan setiap nilai pin LED RGB ke keluaran PWM Arduino (pin PWM: 3, 5, 6, 9, 10, 11).
- Memasang beberapa resistor antara pin LED dan pin Arduino. Resistor 180 ohm untuk pin Merah dan 100 ohm untuk pin Hijau dan Biru.
- Menggunakan panah atas dan bawah untuk warna **Merah**; panah kanan, kiri untuk warna **hijau**; dan huruf b dan huruf c untuk warna **biru**.

Hardware :

- Arduino Uno Board
- 3 buah LED (Merah, Hijau dan Biru)
- 3 buah resistor (180 Ω , 100 Ω =2x)
- Kabel jumper
- Software Processing

Rangkaian -1:



PROGRAM-1:

Arduino Uno

```
int incomingByte = 0;
int g;
int b;
int r;
void setup() {
  Serial.begin(9600);
}

void loop() {
  analogWrite (3, g);
  analogWrite (5, b);
  analogWrite (6, r);
  if (Serial.available() > 0) {
    incomingByte = Serial.read();
    if (incomingByte == 'R') {
      r += 1;
    }
  }
}
```

```
if (incomingByte == 'S') {
  r -= 1;
}
if (incomingByte == 'G') {
  g += 1;
}
if (incomingByte == 'H') {
  g -= 1;
}
if (incomingByte == 'B') {
  b += 1;
}
if (incomingByte == 'C') {
  b -= 1;
}
}
```

PROGRAM-1:

Processing

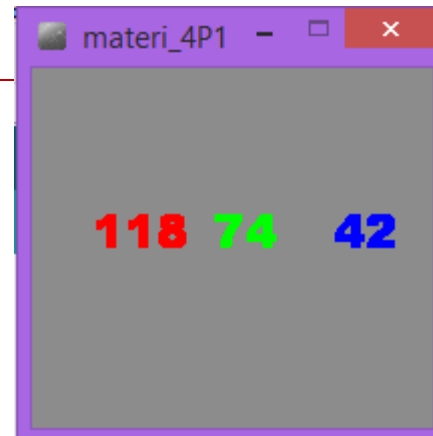
```
import processing.serial.*;
int r; int g; int b;
Serial port;
PFont font;
void setup() {
  size(200, 180);
  background (140);
  println(Serial.list());
  port = new Serial(this,  Serial.list()[0], 9600);
  font = loadFont("Arial-Black-24.vlw");
  textFont(font);
  smooth();
}
void draw() {
  fill (255, 0, 0); text (r, 30, 90);
  fill (0, 255, 0); text (g, 90, 90);
  fill (0, 0, 255); text (b, 150, 90);
  println ("red " + r + "green " + g + "blue " + b);
}
```

PROGRAM-1: (lanjutan)

Processing

```
void keyPressed() {  
  if (key == 'b') {           //blue color  
    port.write('B'); b += 1;  
    background(140);  
  }  
  if (key == 'c') {  
    port.write('C'); b -= 1;  
    background(140);  
  }  
  if (key == CODED) {  
    if (keyCode == UP) {      //red color  
      port.write('R'); r += 1;  
      background(140);  
    }  
    if (keyCode == DOWN) {  
      port.write('S'); r -= 1;  
      background(140);  
    }  
  }  
}
```

```
if (keyCode == RIGHT) { //green color  
  port.write('G');  
  g += 1;  
  background(140);  
}  
if (keyCode == LEFT) {  
  port.write('H');  
  g -= 1;  
  background(140);  
}  
}  
}
```



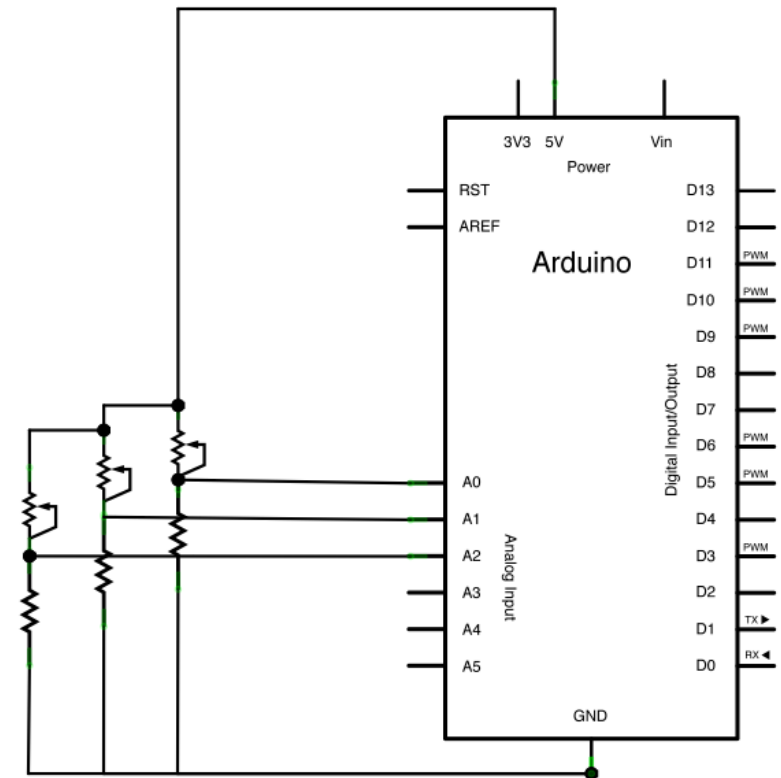
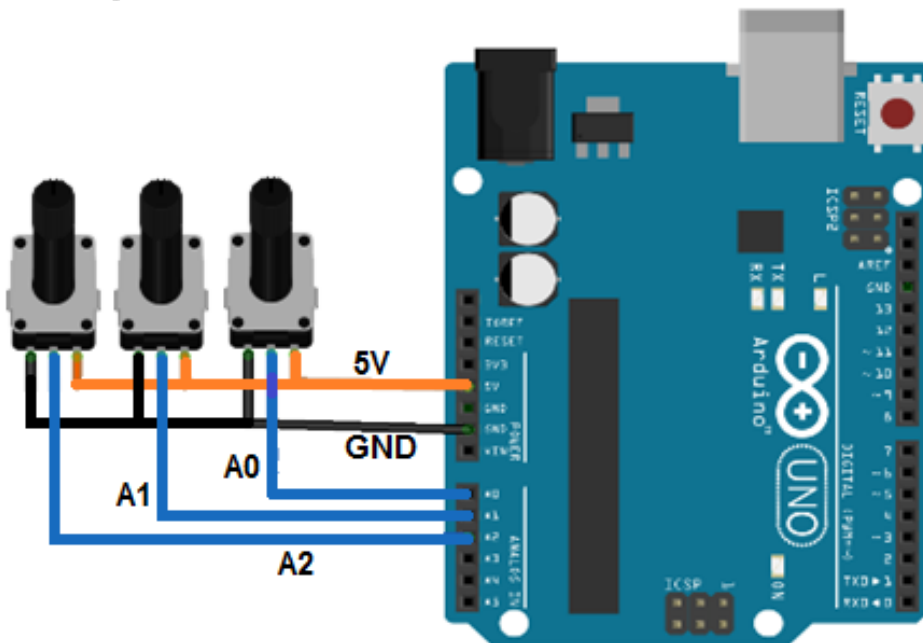
Penjelasan

- Kode di atas menunjukkan nilai pasti dari LED RGB yang dicampur untuk mendapatkan warna yang diinginkan.
- Menunjukkan jumlah nilai yang tepat yang dibutuhkan untuk diteruskan ke pin LED RGB tersebut.
- Jadi, agar mendapatkan warnanya, hanya perlu menuliskan nilai tersebut pada pin output Arduino.

Hardware :

- Arduino Uno Board
- 3 buah potensiometer (5K atau 10K)
- Kabel jumper
- Software Processing

Rangkaian -2:



PROGRAM-2:

Arduino Uno

```
const int redPin = A0;    // sensor untuk kontrol warna merah
const int greenPin = A1; // sensor untuk kontrol warna hijau
const int bluePin = A2;  // sensor untuk kontrol warna biru
```

```
void setup()
```

```
{
  Serial.begin(9600);
}
```

```
void loop()
```

```
{
  Serial.print(analogRead(redPin));
  Serial.print(",");
  Serial.print(analogRead(greenPin));
  Serial.print(",");
  Serial.println(analogRead(bluePin));
}
```

PROGRAM-2:

Processing

```
import processing.serial.*;
float redValue = 0;
float greenValue = 0;
float blueValue = 0;
Serial myPort;

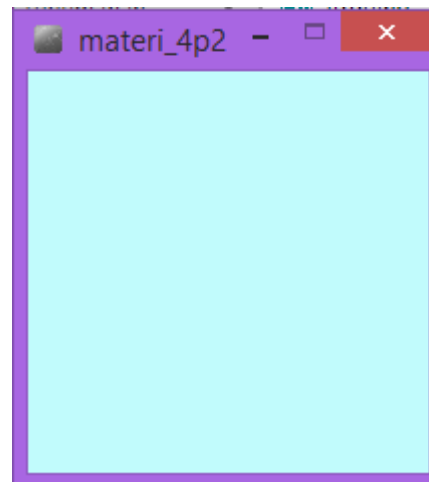
void setup() {
  size(200, 200);
  println(Serial.list());
  myPort = new Serial(this, Serial.list()[0], 9600);
  myPort.bufferUntil('\n');
}

void draw() {
  background(redValue, greenValue, blueValue);
}
```

PROGRAM-2: *(lanjutan)*

Processing

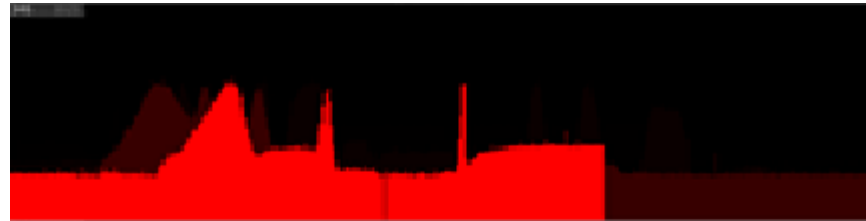
```
void serialEvent(Serial myPort) {  
  String inString = myPort.readStringUntil('\n');  
  if (inString != null) {  
    inString = trim(inString);  
    float[] colors = float(split(inString, ","));  
    if (colors.length >=3) {  
      redValue = map(colors[0], 0, 1023, 0, 255);  
      greenValue = map(colors[1], 0, 1023, 0, 255);  
      blueValue = map(colors[2], 0, 1023, 0, 255);  
    }  
  }  
}
```



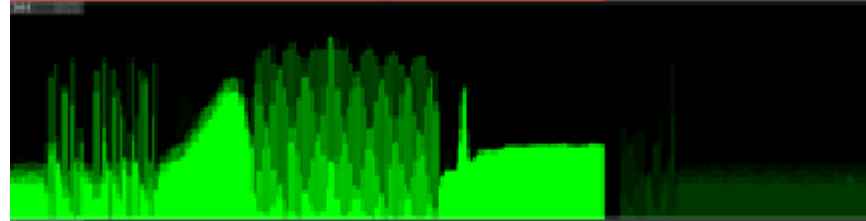
Latihan :

1. Buat aplikasi seperti pada rangkaian-2 untuk menampilkan grafik berwarna dengan tampilan sebagai berikut.

Data dari
Input Analog A0



Data dari
Input Analog A1



Data dari
Input Digital pin 2

