

# **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

# **3. KONTROL MULTI LED MENGUNAKAN ARDUINO DAN PROCESSING**

# TUJUAN

- Mengontrol multi LED pada Board Arduino menggunakan Processing.
- Membuat visual kontroler dengan Processing.

# DASAR TEORI

Sebelumnya telah dibahas, menyalakan LED menggunakan kendali *mouse* komputer/laptop.

dibuat GUI (*Graphical User Interface*) menggunakan Processing untuk mengendalikan multi LED.

membuat tampilan dengan menggunakan software Processing, dapat membuat sistem interfacing untuk mengontrol arah kondisi menyala dan mati dari beberapa LED dengan tampilan yang lebih menarik dan mudah digunakan.

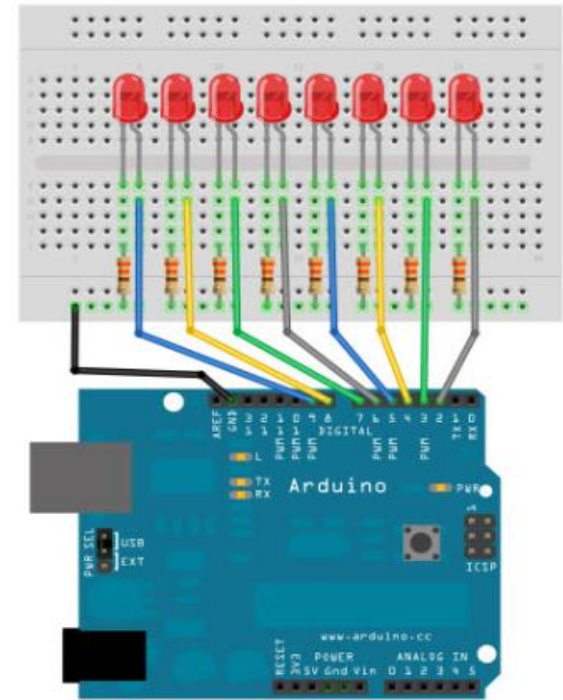


# DASAR TEORI

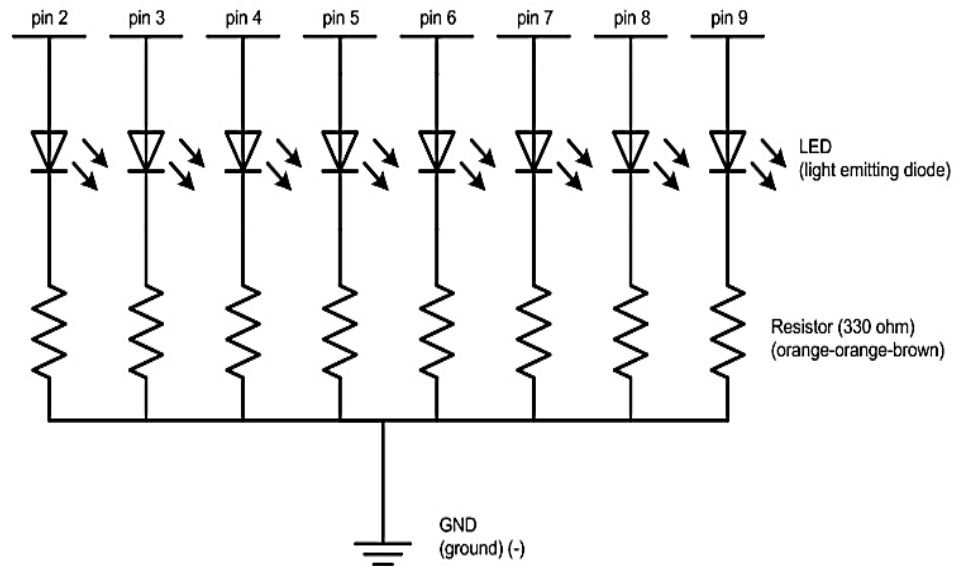
- Untuk membuat interface dengan menggunakan program komunikasi serial sebagaimana telah dibahas pada bab mengakses komunikasi PC dan Arduino.
- Kemudian dibuat pengiriman perintah berupa karakter tertentu yang akan diterjemahkan oleh mikrokontroler untuk menyalakan ataupun mematikan LED

## Hardware :

- Arduino Uno Board
- 8 buah LED
- 8 buah resistor 330 ohm
- Kabel jumper
- Software Processing



## Rangkaian :



# PROGRAM-1:

Arduino Uno

```
int ledPins[] = {2,3,4,5,6,7,8,9};
void setup()
{
  Serial.begin(9600);
  for(int i = 0; i < 8; i++){
    pinMode(ledPins[i],OUTPUT);
  }
}
void loop() {
  byte byteRead;
  if (Serial.available()) {
    byteRead = Serial.read();
    byteRead=byteRead-'0';
    for(int i=2; i<10; i++){
      digitalWrite(i, LOW);
    }
  }
}
```

```
if(byteRead>0){
  for(int i=1; i<(byteRead+1); i++){
    digitalWrite(i+1, HIGH);
  }
}
}
```

# PROGRAM-1:

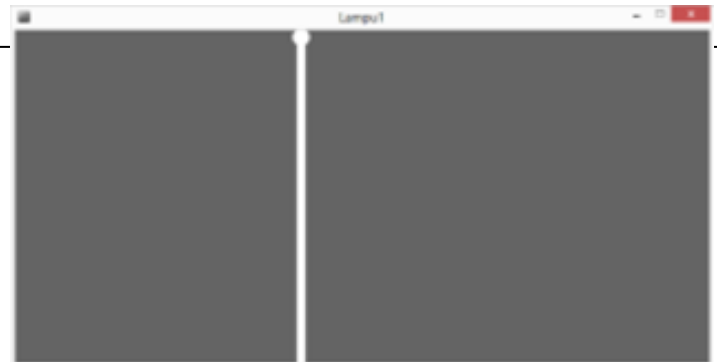
Processing

```
import processing.serial.*;
int new_sX, old_sX;
int nX, nY;
Serial myPort;
void setup(){
  size( 800, 400 );
  strokeWeight( 10 );
  myPort = new Serial(this,
"COM4", 9600);
  myPort.bufferUntil('\n');
}
void draw(){
  background( 100 );
  stroke(255);
  ellipse( nX, nY, 10, 10 );
  line(nX,0,nX,height);
}
```

```
void mouseMoved(){
  nX = mouseX;
  nY = mouseY;

  new_sX=(int)map(nX,0,800,0,10);

  if(new_sX==old_sX){
  } else {
    old_sX = new_sX;
    myPort.write(""+new_sX);
  }
}
```



# PROGRAM-2:

Arduino Uno

```
int incomingByte = 0;

void setup() {
  Serial.begin(9600);
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
}

void loop() {
  if (Serial.available() < 1) {
    return;
  }
  incomingByte = Serial.read();
  // LED-1
  if (incomingByte == 65){
    digitalWrite(2, HIGH);
  }

  if (incomingByte == 66) {
    digitalWrite(2, LOW);
  }
  //LED-2
  if (incomingByte == 67) {
    digitalWrite(3, HIGH);
  }
  if (incomingByte == 68) {
    digitalWrite(3, LOW);
  }
  //LED-3
  if (incomingByte == 69) {
    digitalWrite(4, HIGH);
  }
  if (incomingByte == 70) {
    digitalWrite(4, LOW);
  }
}
```

# ***PROGRAM-2: (Lanjutan)***

```
//LED-4
if (incomingByte == 71) {
    digitalWrite(5, HIGH);
}
if (incomingByte == 72) {
    digitalWrite(5, LOW);
}
}
```

# PROGRAM-2:

Processing

```
import processing.serial.*;
```

```
Serial myPort;
```

```
int val;
```

```
int ButtonValueY = 0;
```

```
int ButtonValueR = 0;
```

```
int ButtonValueG = 0;
```

```
int ButtonValueB = 0;
```

```
int clickflag = 0;
```

```
void setup() {
```

```
  size(480, 500);
```

```
  smooth();
```

```
  String portName = Serial.list()[0];
```

```
  myPort = new Serial(this, portName, 9600);
```

```
  fill(150,150,0);
```

```
  rect(100, 100, 50, 50);
```

```
  fill(150,0,0);
```

```
  rect(100,200, 50,50);
```

```
  fill(0,150,0);
```

```
  rect(100,300,50,50);
```

```
  fill(0,0,150);
```

```
  rect(100,400,50,50);
```

```
}
```

```
void draw() {
```

```
  if (mousePressed) {
```

```
    clickflag = 0;
```

```
    return;
```

```
  }
```

```
  if (clickflag == 1)
```

```
  {
```

```
    return;
```

```
  }
```

```
  clickflag = 1;
```

# PROGRAM-2:

Processing

```
if ((mouseX > 100) && (mouseX < 150) && (mouseY > 100) && (mouseY < 150))
{
  if (ButtonValueY == 0) {
    ButtonValueY = 1;
    fill(255,255,0);
    myPort.write('A');
    rect(100, 100, 50, 50);
  } else {
    fill(150,150,0);
    ButtonValueY = 0;
    myPort.write('B');
    rect(100, 100, 50, 50);
  }
}
if ((mouseX > 100) && (mouseX < 150) && (mouseY > 200) && (mouseY < 250))
{
  if (ButtonValueR == 0) {
    ButtonValueR = 1;
    fill(255,0,0);
```



# PROGRAM-2:

Processing

```
    myPort.write('C');
    rect(100, 200, 50, 50);
  } else {
    fill(150,0,0);
    ButtonValueR = 0;
    myPort.write('D');
    rect(100, 200, 50, 50);
  }
}
}
if ((mouseX > 100) && (mouseX < 150) && (mouseY > 300) && (mouseY < 350))
{
  if (ButtonValueG == 0) {
    ButtonValueG = 1;
    fill(0,255,0);
    myPort.write('E');
    rect(100, 300, 50, 50);
  } else {
    fill(0,150,0);
    ButtonValueG = 0;
    myPort.write('F');
    rect(100, 300, 50, 50);
  }
}
```

# PROGRAM-2:

Processing

```
if ((mouseX > 100) && (mouseX < 150) && (mouseY > 400) && (mouseY < 450))  
{  
  if (ButtonValueB == 0) {  
    ButtonValueB = 1;  
    fill(0,0,255);  
    myPort.write('G');  
    rect(100, 400, 50, 50);  
  } else {  
    fill(0,0,150);  
    ButtonValueB = 0;  
    myPort.write('H');  
    rect(100, 400, 50, 50);  
  }  
}
```

LED pin =2



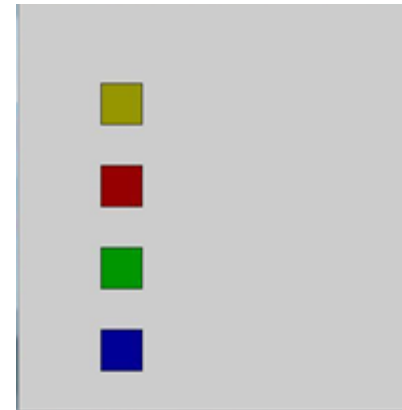
LED pin =3



LED pin =4



LED pin =5



# PROGRAM-3:

Arduino Uno

```
int ledPins[] = {2,3,4,5,6,7,8,9};
void setup()
{
  Serial.begin(9600);
  for(int i = 0; i < 8; i++){
    pinMode(ledPins[i],OUTPUT);
  }
}
void loop() {
  byte byteRead;
  if (Serial.available()) {
    byteRead = Serial.read();
    byteRead=byteRead-'0';
  if(byteRead==0){
    digitalWrite(2, LOW);
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
    digitalWrite(5, LOW);
    digitalWrite(6, LOW);
```

```
digitalWrite(7, LOW);
digitalWrite(8, LOW);
digitalWrite(9, LOW);
}
if(byteRead>0){
  digitalWrite((byteRead+1), HIGH); }
}
}
```

# PROGRAM-3:

Processing

```
import processing.serial.*;
```

```
Serial comPort;
```

```
int counter=0; .
```

```
int numItems=0;
```

```
boolean sendStrings=false;
```

```
StringLoader sLoader;
```

```
void setup(){
```

```
  comPort = new Serial(this, Serial.list()[0], 9600);
```

```
  background(255,0,0);
```

```
}
```

```
void draw(){
```

```
}
```

```
void mousePressed() {
```

```
  sendStrings=!sendStrings;
```

```
  if(sendStrings){
```

```
    background(0,255,0);
```

```
    sLoader=new StringLoader();
```

```
    sLoader.start();
```

```
}else{
```

```
  background(255,0,0);
```

```
  counter=0;
```

```
}
```

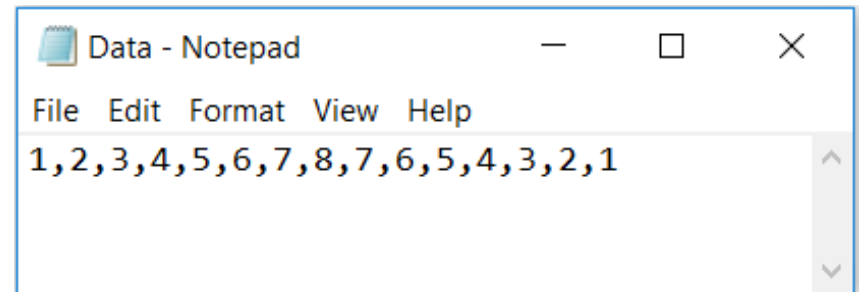
```
}
```

# PROGRAM-3:

Processing lanjutan

```
public StringLoader(){
}

public void run() {
String textFileLines[]=loadStrings("d:/Data.txt");
String lineItems[]=splitTokens(textFileLines[0], ",");
numItems=lineItems.length;
for(int i = counter; i<numItems; i++){
comPort.write(lineItems[i]);
delay(500);
comPort.write("0");
}
counter=numItems;
}
}
```



## ***Latihan :***

1. Buat aplikasi kontrol multi LED dengan ketentuan :
  - ✓ Saat tombol merah ditekan, maka tombol akan berubah warna menjadi hijau dan LED menyala
  - ✓ Sedangkan saat tombol ditekan kembali maka tombol akan berubah menjadi berwarna merah kembali dan LED akan mati.

Karakter yang dikirim PC ke Arduino

No	Kondisi LED	Karakter
1	Nyala	a
2	Mati	b

Visual kontroler pada PC (processing)

