

# **LAPORAN PRAKTIKUM SISTEM TERTANAM**

## **PEMROGRAMAN SISTEM TERTANAM**



### **KELOMPOK 5**

**Agus Pranata Marpaung – 13323033**

**Arizona Damanik – 13323029**

**Jeyshen Siallagan - 13323012**

**DIII TEKNOLOGI KOMPUTER**

**INSTITUT TEKNOLOGI DEL  
FAKULTAS VOKASI**

## Judul Praktikum

---

<b>Minggu/Sesi</b>	:	IV/3
<b>Kode Mata Kuliah</b>	:	4332104
<b>Nama Mata Kuliah</b>	:	DASAR ELEKTRONIKA
<b>Setoran</b>	:	Jawaban dalam bentuk <i>softcopy</i>
<b>Batas Waktu Setoran</b>	:	3 Oktober jam 21:15
<b>Tujuan</b>	:	Memahami konsep sinyal analog dan digital. Menerapkan konversi Analog-to-Digital (ADC) dan Digital-to-Analog (DAC) pada Arduino Mega. Mengolah data sensor analog dan mengendalikan aktuator digital. Membuat proyek sederhana yang menggabungkan komponen analog dan digital.

## Petunjuk

## **1. Alat dan Bahan**

- Arduino Mega
- Breadboard
- Kabel jumper
- Sensor analog (misal: potensiometer, LDR, sensor suhu)
- Aktuator digital (misal: LED, buzzer)
- Resistor
- Arduino IDE

## **2. Materi Pokok**

### **a. Sinyal analog dan digital**

Bayangkan sebuah gelombang suara. Ketika Anda berbicara, suara Anda menciptakan gelombang udara yang terus berubah-ubah dalam amplitudo (tingkat kerasnya) dan frekuensi (tingginya nada). Gelombang suara ini adalah contoh dari sinyal analog. Sinyal analog memiliki nilai yang dapat bervariasi secara kontinu dalam rentang tertentu. Sebaliknya, sinyal digital hanya memiliki dua nilai: 0 dan 1. Ini seperti menyalakan atau mematikan sebuah lampu. Komputer dan perangkat elektronik digital menggunakan sinyal digital untuk memproses informasi.

Sinyal Analog: Nilai kontinu, bisa memiliki nilai apa saja dalam rentang tertentu.

Sinyal Digital: Nilai diskrit, hanya memiliki dua nilai (0 atau 1).

Sinyal Analog: Digambarkan sebagai gelombang yang mulus.

Sinyal Digital: Digambarkan sebagai serangkaian pulsa (tinggi atau rendah).

### **b. Konversi ADC**

#### **Mengapa kita perlu ADC?**

Perangkat elektronik digital seperti Arduino tidak dapat langsung memproses sinyal analog. Untuk itu, kita perlu mengubah sinyal analog menjadi sinyal digital terlebih dahulu. Proses ini disebut konversi Analog-to-Digital (ADC).

### **Bagaimana cara kerjanya?**

**Fungsi `analogRead()`:** Fungsi ini digunakan pada Arduino untuk membaca nilai analog dari pin analog.

**Resolusi ADC:** Ini menentukan seberapa detail nilai analog dapat diwakili dalam bentuk digital. Semakin tinggi resolusi, semakin akurat hasil konversi.

**Pemetaan nilai ADC ke nilai fisik:** Nilai yang didapatkan dari `analogRead()` biasanya berupa bilangan bulat. Untuk mendapatkan nilai fisik (misalnya, tegangan, suhu), kita perlu melakukan perhitungan berdasarkan karakteristik sensor yang digunakan.

### **Contoh:**

Misalkan kita ingin mengukur suhu menggunakan sensor suhu yang terhubung ke pin analog A0. Nilai yang didapat dari `analogRead(A0)` akan kita konversi menjadi nilai suhu dalam derajat Celcius menggunakan rumus yang sesuai dengan datasheet sensor.

## **c. Konversi DAC**

### **Mengapa kita perlu DAC?**

Setelah memproses data secara digital, kita mungkin ingin mengontrol perangkat analog seperti motor, LED, atau speaker. Untuk itu, kita perlu mengubah sinyal digital kembali menjadi sinyal analog. Proses ini disebut konversi Digital-to-Analog (DAC).

### **Bagaimana cara kerjanya?**

**Fungsi `analogWrite()`:** Fungsi ini digunakan pada Arduino untuk menghasilkan sinyal PWM (Pulse Width Modulation). PWM adalah teknik untuk menghasilkan sinyal analog dengan mengatur lebar pulsa dari sinyal digital.

**Prinsip Kerja PWM:** Dengan mengubah lebar pulsa, kita dapat mengatur rata-rata tegangan output. Semakin lebar pulsa, semakin tinggi tegangan rata-rata.

**Pemetaan nilai digital ke tegangan analog:** Nilai yang diberikan pada `analogWrite()` akan menentukan lebar pulsa.

**Contoh:**

Misalkan kita ingin mengontrol kecerahan LED. Dengan mengatur nilai yang diberikan pada `analogWrite()`, kita dapat mengubah lebar pulsa, sehingga kecerahan LED akan berubah.

**3. Praktikum 1: Membaca Nilai Potensiometer (ADC)**

1. **Rangkaian:** Hubungkan potensiometer ke analog pin Arduino
2. **Code Program:**

```
C++

int potPin = A0;
int nilaiPot;

void setup() {
  Serial.begin(9600);
}

void loop() {
  nilaiPot = analogRead(potPin);
  Serial.println(nilaiPot);
  delay(100);
}
```

3. **Analisis:** Seperti apa hasil dari program di atas, jelaskan.

***Penjelasan hasil:***

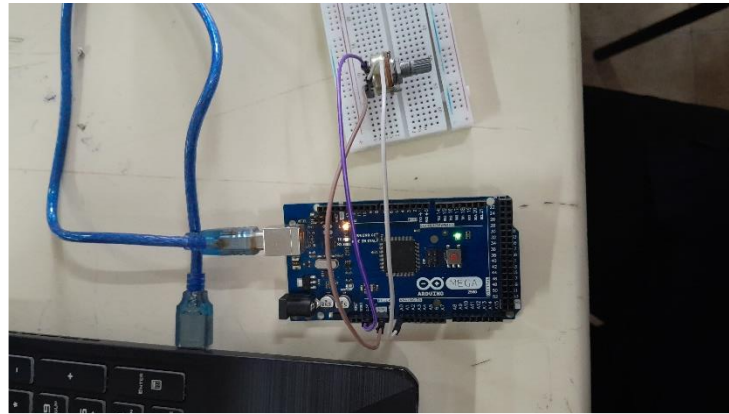
Arduino akan membaca nilai tegangan analog dari potensiometer yang terhubung ke pin A0. Tegangan analog tersebut dikonversi menjadi nilai digital oleh ADC dengan rentang 0 hingga 1023 (karena resolusi ADC 10-bit pada Arduino Mega 2560). Nilai yang terbaca akan bergantung pada posisi knob potensiometer:

- Nilai 0: Jika knob berada pada posisi minimal (tegangan = 0V).
- Nilai 1023: Jika knob berada pada posisi maksimal (tegangan = 5V).

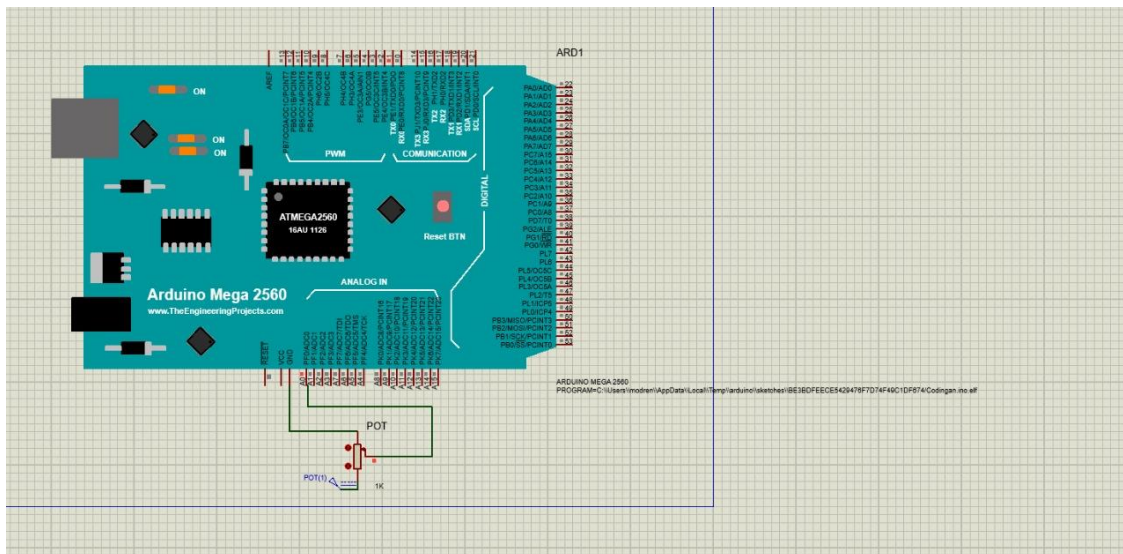
- Nilai antara 0 hingga 1023: Jika knob berada di posisi tengah, nilai akan bervariasi sesuai dengan perubahan tegangan.

### Tugas membuat Laporan:

1. Silahkan membuat rangkaian seperti yang diminta pada langkah 1 di atas kemudian foto atau dokumentasikan pada laporan.

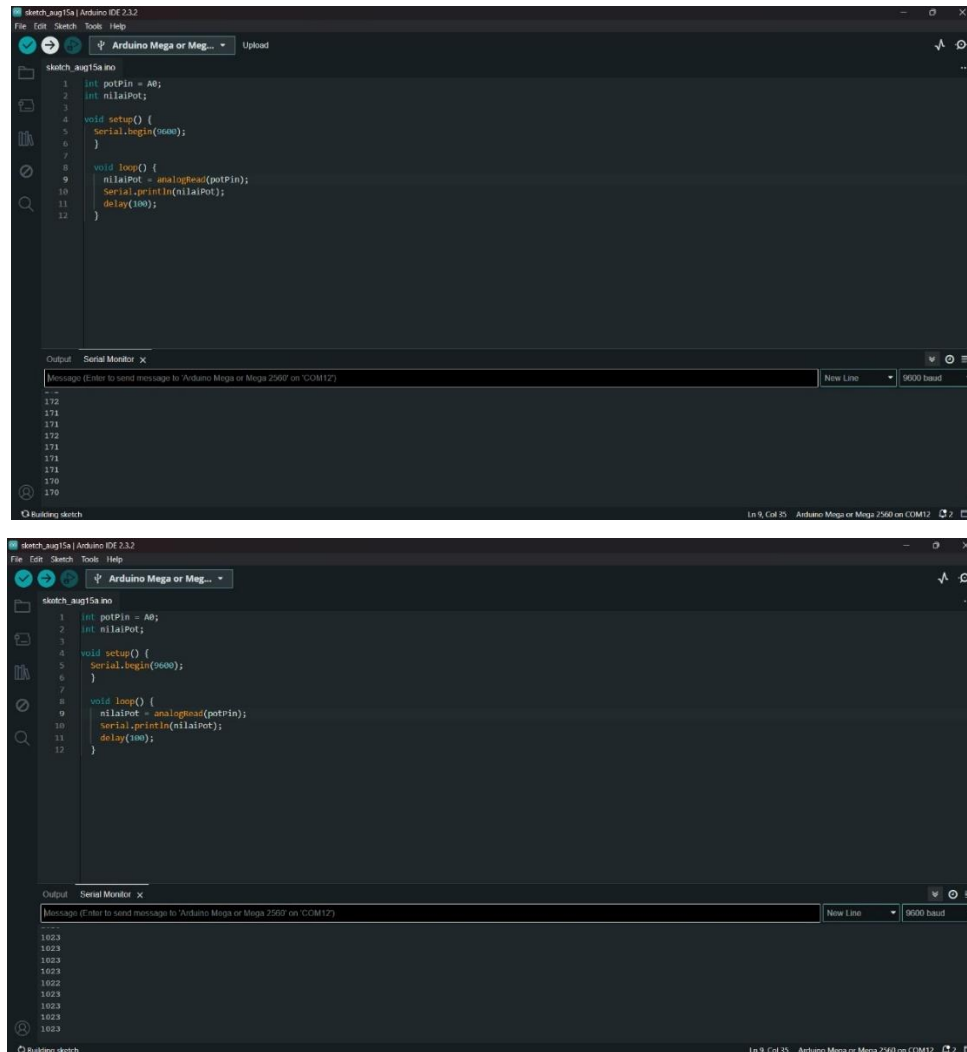


Gambar Rangkaian Asli



Gambar Simulasi Rangkaian

2. Tulis code program pada Arduino IDE anda, kemudian run program dan lampirkan hasil program dari rangkaian dan serial monitor anda.



The image contains two screenshots of the Arduino IDE interface. The top screenshot shows the code editor with the following code:

```
1 int potPin = A0;
2 int nilaiPot;
3
4 void setup() {
5   Serial.begin(9600);
6 }
7
8 void loop() {
9   nilaiPot = analogRead(potPin);
10  Serial.println(nilaiPot);
11  delay(100);
12 }
```

The serial monitor at the bottom shows the output of the program, displaying the values 172, 171, 172, 171, 171, 171, 170, and 170.

The bottom screenshot shows the same code editor with the following code:

```
1 int potPin = A0;
2 int nilaiPot;
3
4 void setup() {
5   Serial.begin(9600);
6 }
7
8 void loop() {
9   nilaiPot = analogRead(potPin);
10  Serial.println(nilaiPot);
11  delay(100);
12 }
```

The serial monitor at the bottom shows the output of the program, displaying the values 1023, 1023, 1023, 1023, 1023, 1023, 1023, and 1023.

### Rangkaian dengan Arduino Mega 2560:

Komponen yang Diperlukan:

- Arduino Mega 2560
- Potensiometer 10k $\Omega$
- Kabel jumper
- Breadboard (opsional)

Koneksi pada Arduino Mega 2560:

Pin tengah potensiometer tetap dihubungkan ke pin analog A0. Salah satu sisi potensiometer dihubungkan ke GND pada Arduino Mega 2560. Sisi lain potensiometer dihubungkan ke 5V pada Arduino Mega 2560.

Penjelasan Koneksi pada Arduino Mega 2560:

- Analog Pin A0 (ADC): Pada Mega 2560, pin analog berada di sisi kanan board dan diidentifikasi sebagai A0 hingga A15. Dalam hal ini, kita menggunakan A0 untuk membaca nilai potensiometer.
- GND & 5V: Seperti pada board lain, 5V akan memberikan tegangan referensi ke potensiometer, dan GND adalah ground.

#### 4. Praktikum 2: Mengendalikan Kecerahan LED dengan Potensiometer (DAC)

1. **Rangkaian:** Hubungkan potensiometer ke analog pin, LED ke digital pin melalui resistor.
2. **Kode Program**

```
C++

int potPin = A0;
int ledPin = 9;
int nilaiPot;

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  nilaiPot = analogRead(potPin);
  analogWrite(ledPin, nilaiPot / 4);
}
```

3. **Analisis:** Silahkan run program tersebut, seperti apa hasil dari program tersebut, jelaskan.

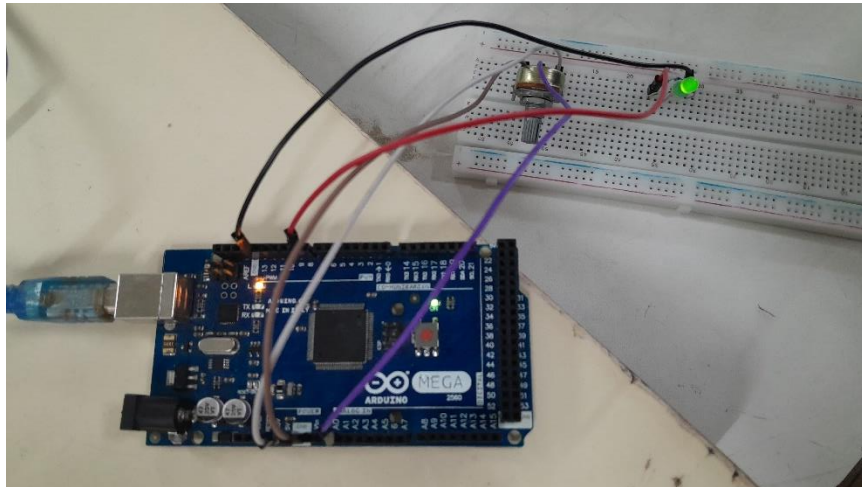


Kecerahan LED akan berubah berdasarkan nilai yang dibaca dari potensiometer. Saat potensiometer diputar, tegangan yang dibaca oleh pin analog (A0) akan berubah dari 0 hingga 1023. Nilai ini kemudian dibagi dengan 4 untuk mendapatkan nilai antara 0 hingga 255, yang kemudian dikirim ke pin PWM (9) untuk mengontrol kecerahan LED.

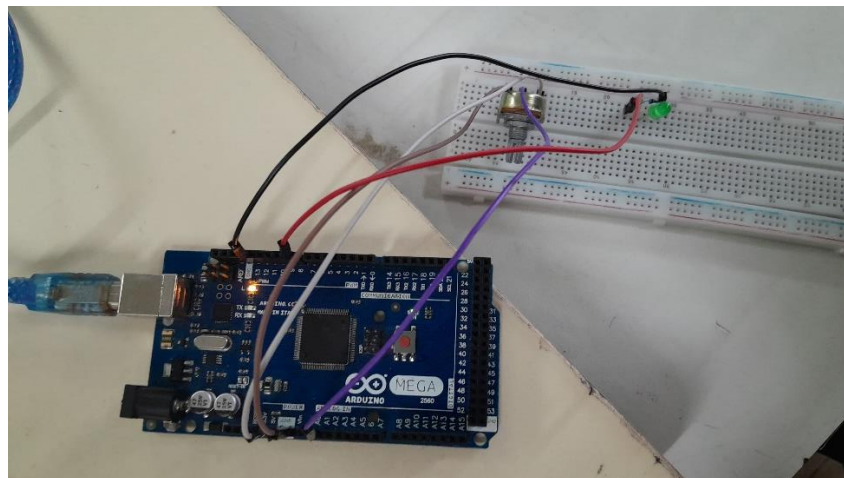
Potensiometer sepenuhnya ke salah satu sisi (misalnya, posisi minimum) akan menyebabkan LED mati (kecerahan 0), dan ketika diputar sepenuhnya ke sisi lain (posisi maksimum), LED akan menyala dengan kecerahan maksimum.

#### 4. Tugas membuat Laporan:

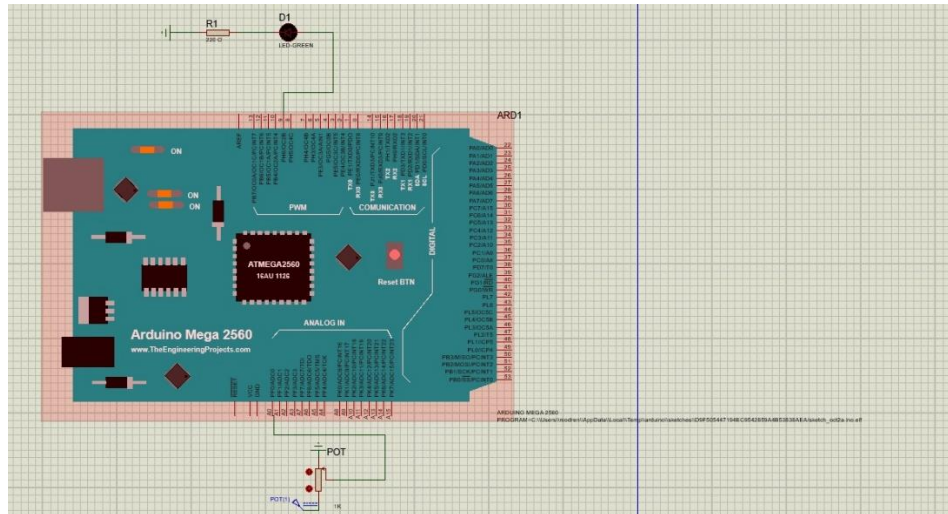
1. Silahkan membuat rangkaian seperti yang diminta pada langkah 1 di atas kemudian foto atau dokumentasikan pada laporan.



Gambar Rangkaian Asli (Lampu Cerah)



Gambar Rangkaian Asli (Lampu Redup)



Gambar Simulasi Rangkaian

2. Tulis code program pada Arduino IDE anda, lampirkan hasil run program dari serial.monitor dan rangkaian.

```

sketch_aug15a.ino
1  int potPin = A0;
2  int ledPin = 9;
3  int nilaiPot;
4
5  void setup() {
6    pinMode(ledPin, OUTPUT);
7  }
8
9  void loop() {
10     nilaiPot = analogRead(potPin);
11     analogWrite(ledPin, nilaiPot / 4);
12 }

```

Output Serial Monitor

Sketch uses 1894 bytes (6%) of program storage space. Maximum is 253952 bytes.  
Global variables use 9 bytes (0%) of dynamic memory, leaving 8183 bytes for local variables. Maximum is 8192 bytes.

Potensiometer:

- Pin 1: Terhubung ke 5V (tegangan suplai dari Arduino).
- Pin 2: Terhubung ke A0 (pin analog Arduino) untuk membaca tegangan keluaran potensiometer.
- Pin 3: Terhubung ke Ground (GND).

LED:

- Anoda (kaki panjang): Terhubung ke pin 9 (pin PWM pada Arduino).
- Katoda (kaki pendek): Terhubung ke resistor 220 ohm, yang kemudian terhubung ke Ground (GND).

## 5. **Praktikum 3: Membaca Sensor Ultrasonik dan Menampilkan pada Serial Monitor**

1. **Rangkaian:** Hubungkan sensor suhu (misal LM35) ke analog pin

2. **Code Program:**

```
1  #define trigPin 9
2  #define echoPin 10
3  #define buzzerPin 8 // Pin untuk buzzer
4  float jarak;
5
6  void setup() {
7      Serial.begin(9600);
8      pinMode(trigPin, OUTPUT);
9      pinMode(echoPin, INPUT);
10     pinMode(buzzerPin, OUTPUT); // Inisialisasi buzzer sebagai OUTPUT
11 }
12
13 void loop() {
14     // Mengirimkan sinyal ultrasonik
15     digitalWrite(trigPin, LOW);
16     delayMicroseconds(2);
17     digitalWrite(trigPin, HIGH);
18     delayMicroseconds(10);
19     digitalWrite(trigPin, LOW);
20
21     // Menerima sinyal pantulan dan menghitung durasi echo
22     long durasi = pulseIn(echoPin, HIGH);
23
24     // Menghitung jarak (cm)
25     jarak = durasi * 0.034 / 2;
26
27     // Menampilkan hasil pada Serial Monitor
28     Serial.print("Jarak: ");
29     Serial.print(jarak);
30     Serial.println(" cm");
31
32     // Mengeluarkan suara jika jarak kurang dari 10 cm
33     if (jarak < 10) {
```

```
34     tone(buzzerPin, 1000); // Mengeluarkan nada 1000 Hz
35 } else {
36     noTone(buzzerPin);    // Matikan suara
37 }
38
39 delay(1000); // Tunggu 1 detik sebelum pembacaan berikutnya
40 }
```

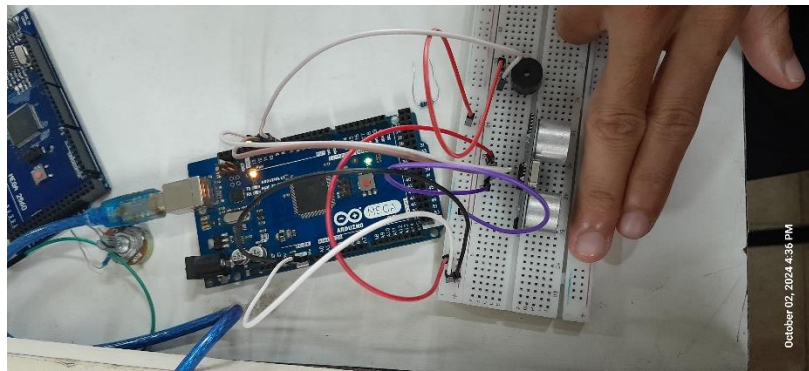
3. **Analisis:** Silahkan run program tersebut, seperti apa hasil dari program tersebut, jelaskan.

Program akan terus-menerus membaca jarak dari sensor ultrasonik HC-SR04.

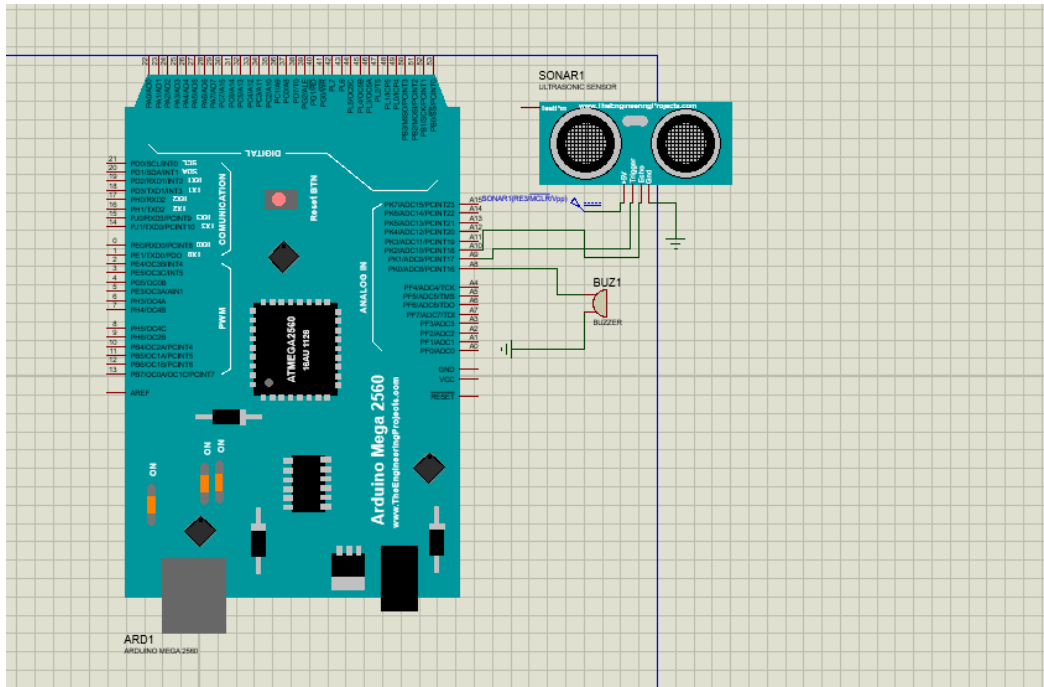
Jika jarak yang terdeteksi kurang dari 10 cm, buzzer dua kaki akan mengeluarkan suara dengan frekuensi 1000 Hz. Jika jarak lebih dari 10 cm, buzzer akan diam. Setiap pembacaan jarak akan ditampilkan di Serial Monitor dalam satuan centimeter (cm) setiap 1 detik.

4. **Tugas membuat Laporan:**

1. Silahkan membuat rangkaian seperti yang diminta pada langkah 1 di atas kemudian foto atau dokumentasikan pada laporan.



Gambar Rangkaian Asli



Gambar Rangkaian di Proteus

Komponen yang Diperlukan:

- Buzzer pasif (dua kaki).
- Sensor ultrasonik HC-SR04.
- Arduino Mega 2560.

Gambaran Koneksi:

- Hubungkan salah satu kaki buzzer ke pin digital 8 Arduino (atau pin lainnya yang tidak digunakan).
- Hubungkan kaki buzzer lainnya ke GND Arduino.
- VCC sensor ultrasonik dihubungkan ke 5V Arduino.
- GND sensor ultrasonik dihubungkan ke GND Arduino.
- Trig sensor ultrasonik dihubungkan ke pin 9 Arduino.
- Echo sensor ultrasonik dihubungkan ke pin 10 Arduino.

## 6. **Praktikum 4: Membuat Buzzer Berbunyi dengan Frekuensi Variabel**

1. **Rangkaian:** Hubungkan buzzer ke digital pin.
2. **Code Program**

C++

```
int buzzerPin = 9;
int frekuensi;

void setup() {
  pinMode(buzzerPin, OUTPUT);
}

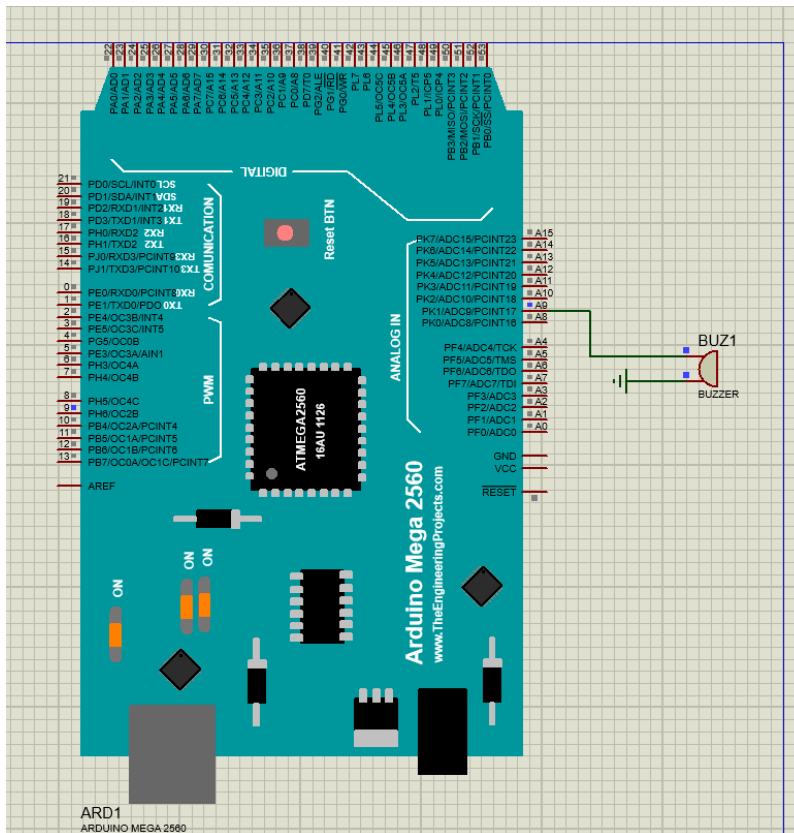
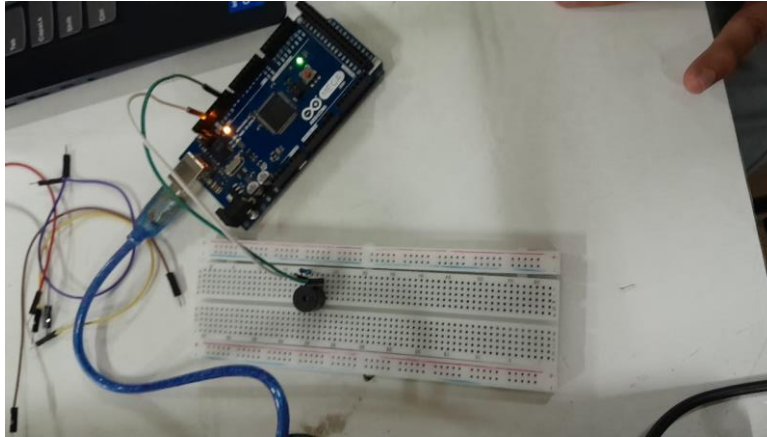
void loop() {
  for (frekuensi = 200; frekuensi < 2000; frekuensi += 50) {
    tone(buzzerPin, frekuensi);
    delay(100);
    noTone(buzzerPin);
    delay(100);
  }
}
```

3. **Analisis:** Silahkan run program tersebut, seperti apa hasil dari program tersebut, jelaskan.

Ketika program ini dijalankan, buzzer akan mengeluarkan suara dengan frekuensi yang meningkat dari 200 Hz hingga 2000 Hz. Setiap frekuensi akan terdengar selama 100 milidetik, dengan jeda 100 milidetik di antara setiap perubahan frekuensi. Hasilnya adalah rangkaian suara yang berubah secara bertahap, membuat suara yang monoton namun menonjol.

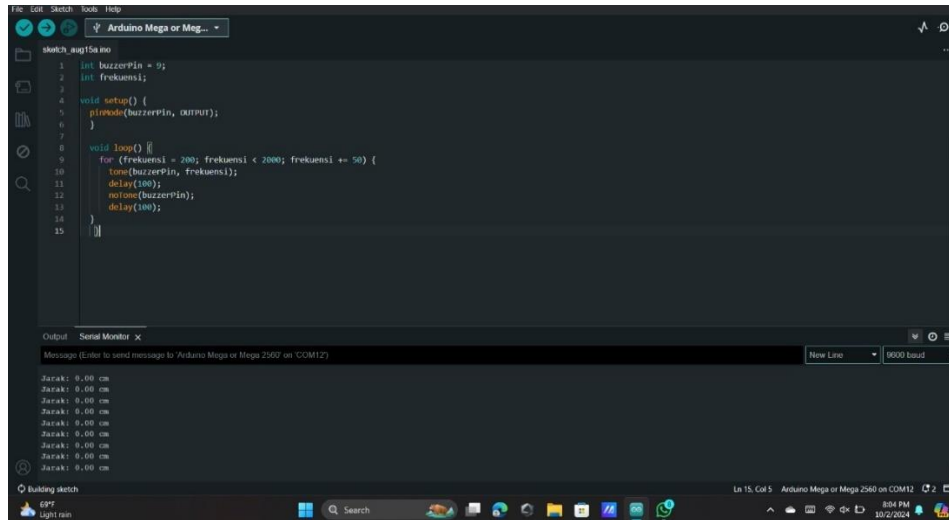
#### 4. Tugas membuat Laporan:

1. Silahkan membuat rangkaian seperti yang diminta pada langkah 1 diatas kemudian foto atau dokumentasikan pada laporan.





2. Tulis code program pada Arduino IDE anda, lampirkan hasil run program dari serial.monitor dan rangkaian.



#### Komponen yang Diperlukan:

- Arduino (misalnya Arduino Uno atau Mega)
- Buzzer
- Resistor (opsional, jika diperlukan untuk perlindungan)
- Breadboard (opsional)
- Jumper wire

#### Cara Menghubungkan:

##### Buzzer:

- Sambungkan kaki positif (yang lebih panjang) dari buzzer ke pin 9 pada Arduino.
- Sambungkan kaki negatif (yang lebih pendek) dari buzzer ke GND (ground) pada Arduino.

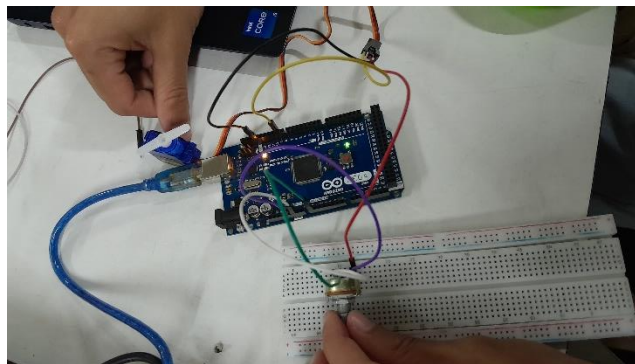


Jika Menggunakan Resistor:

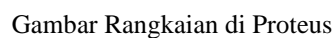
Anda dapat menambahkan resistor (misalnya  $220\Omega$ ) dalam seri dengan buzzer untuk membatasi arus dan melindungi komponen.

**5. Tugas tambahan membuat laporan [Rangkaian, Code Program dan Hasil program]**

- a. Buat program untuk mengontrol kecepatan motor servo berdasarkan nilai potensiometer.



Gambar Rangkaian



b. Code program

```
#include <Servo.h>

Servo myServo;

int potPin = A0;
int potValue;
int servoValue;

void setup() {
  myServo.attach(9);
  Serial.begin(9600);
}

void loop() {
  potValue = analogRead(potPin);

  // Mengubah nilai potensiometer menjadi sudut servo (0-180 derajat)
  servoValue = map(potValue, 0, 1023, 0, 180);

  myServo.write(servoValue); // Menggerakkan servo ke sudut yang diinginkan

  // Menampilkan nilai potensiometer dan sudut servo ke Serial Monitor
  Serial.print("Nilai Potensiometer: ");
  Serial.print(potValue);
  Serial.print(" | Sudut Servo: ");
  Serial.println(servoValue);

  delay(15); // Menunggu servo untuk mencapai posisi
}
```

Tegangan = 0

The screenshot shows the Arduino IDE interface. The top toolbar includes icons for file operations, running the sketch, and serial monitor. The sketch editor shows the following code:

```

1 #include <Servo.h>
2
3 Servo myServo; // Membuat objek servo
4
5 int potPin = A0; // Pin potensiometer
6 int potValue; // Variabel untuk menyimpan nilai potensiometer
7 int servoValue; // Variabel untuk menyimpan nilai sudut servo
8
9 void setup() {
10   myServo.attach(9); // Menghubungkan objek servo ke pin digital 9
11   Serial.begin(9600); // Memulai komunikasi serial dengan baud rate 9600
12 }
13
14 void loop() {
15   potValue = analogRead(potPin); // Membaca nilai dari potensiometer (0-1023)
16
17   // Mengubah nilai potensiometer menjadi sudut servo (0-180 derajat)
18   servoValue = map(potValue, 0, 1023, 0, 180);
19
20   myServo.write(servoValue); // Menggerakkan servo ke sudut yang diinginkan
21
22   // Menampilkan nilai potensiometer dan sudut servo ke Serial Monitor
23   Serial.print("Nilai Potensiometer: ");

```

The serial monitor at the bottom shows the output of the code, displaying the potentiometer value and the corresponding servo angle:

```

Message (Enter to send message to 'Arduino Mega or Mega 2560' on 'COM12')
New Line 9600 baud

Nilai Potensiometer: 1019 | Sudut Servo: 179
Nilai Potensiometer: 1023 | Sudut Servo: 180
Nilai Potensiometer: 1023 | Sudut Servo: 180
Nilai Potensiometer: 1023 | Sudut Servo: 180
Nilai Potensiometer: 1023 | Sudut Servo: 180
Nilai Potensiometer: 1021 | Sudut Servo: 179
Nilai Potensiometer: 1023 | Sudut Servo: 180
Nilai Potensiometer: 1023 | Sudut Servo: 180
Nilai Potensiometer: 1023 | Sudut Servo: 180

```

$$\text{Tegangan} = \text{maksimum} / 1023$$

### Komponen yang Diperlukan

- Arduino Board
- Motor Servo
- Potensiometer
- Breadboard
- Kabel Jumper
- Skema Koneksi

### Potensiometer:

- Terminal tengah potensiometer ke Analog Pin A0 Arduino.
- Salah satu terminal luar ke GND.
- Terminal luar lainnya ke VCC (5V) pada Arduino.

### Motor Servo:

- Kabel sinyal servo (biasanya kabel kuning atau putih) ke Digital Pin 9 pada Arduino.
- Kabel merah ke VCC (5V) pada Arduino.
- Kabel hitam atau coklat ke GND.

### Hasil Program:

Ketika Anda memutar potensiometer, sudut putar motor servo akan berubah sesuai dengan nilai yang dibaca dari potensiometer. Semakin besar nilai potensiometer, semakin besar sudut putar servo. Ini memberikan kontrol langsung dan responsif atas posisi servo menggunakan potensiometer.

Potensiometer berfungsi sebagai sensor yang memberikan input analog ke Arduino.

Arduino memproses nilai dari potensiometer dan mengontrol motor servo.

Motor Servo menerima sinyal dari Arduino untuk menentukan sudut putarnya berdasarkan nilai potensiometer.

- c. Buat program untuk membuat LED berkedip dengan frekuensi yang diatur oleh potensiometer.

Code Program:

```
int ledPin = 9; // Pin LED

int potPin = A0; // Pin potensiometer

int potValue; // Variabel untuk menyimpan nilai potensiometer

int blinkDelay; // Variabel untuk menyimpan delay berkedip LED

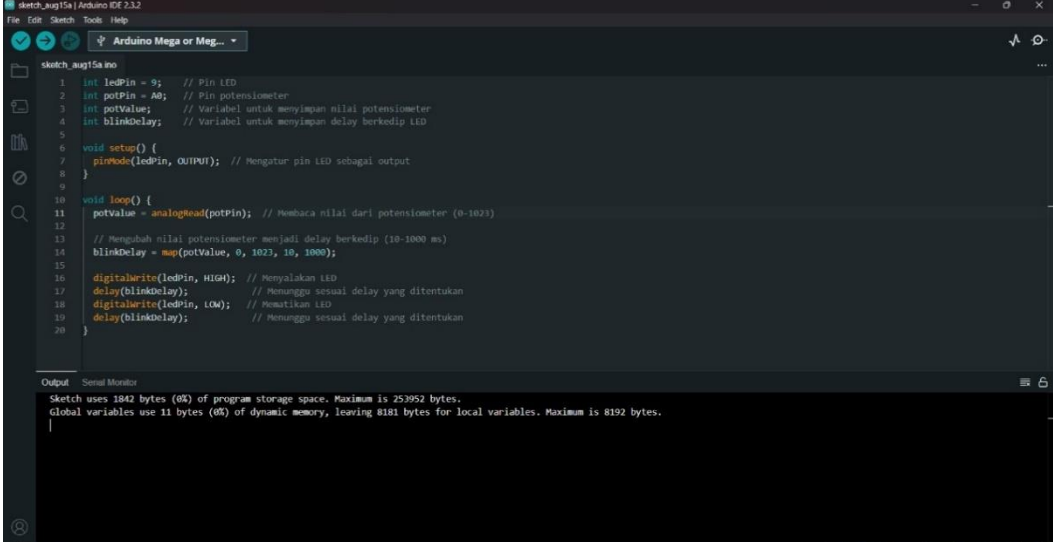

void setup() {
  pinMode(ledPin, OUTPUT); // Mengatur pin LED sebagai output
}


void loop() {
  potValue = analogRead(potPin); // Membaca nilai dari potensiometer (0-1023)

  // Mengubah nilai potensiometer menjadi delay berkedip (10-1000 ms)
  blinkDelay = map(potValue, 0, 1023, 10, 1000);

  digitalWrite(ledPin, HIGH); // Menyalakan LED
  delay(blinkDelay); // Menunggu sesuai delay yang ditentukan
  digitalWrite(ledPin, LOW); // Mematikan LED
  delay(blinkDelay); // Menunggu sesuai delay yang ditentukan
}
```

## Output Program:



```
1 int ledPin = 9; // Pin LED
2 int potPin = A0; // Pin potensiometer
3 int potValue; // Variabel untuk menyimpan nilai potensiometer
4 int blinkDelay; // Variabel untuk menyimpan delay berkedip LED
5
6 void setup() {
7   pinMode(ledPin, OUTPUT); // Mengatur pin LED sebagai output
8 }
9
10 void loop() {
11   potValue = analogRead(potPin); // Membaca nilai dari potensiometer (0-1023)
12
13   // Mengubah nilai potensiometer menjadi delay berkedip (10-1000 ms)
14   blinkDelay = map(potValue, 0, 1023, 10, 1000);
15
16   digitalWrite(ledPin, HIGH); // Menyalakan LED
17   delay(blinkDelay); // Menunggu sesuai delay yang ditentukan
18   digitalWrite(ledPin, LOW); // Mematikan LED
19   delay(blinkDelay); // Menunggu sesuai delay yang ditentukan
20 }
```

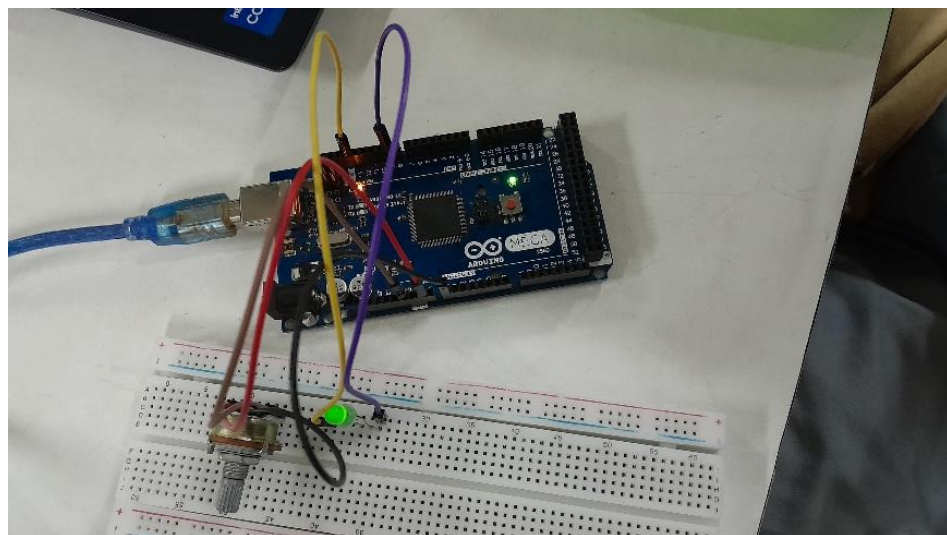
Output Serial Monitor

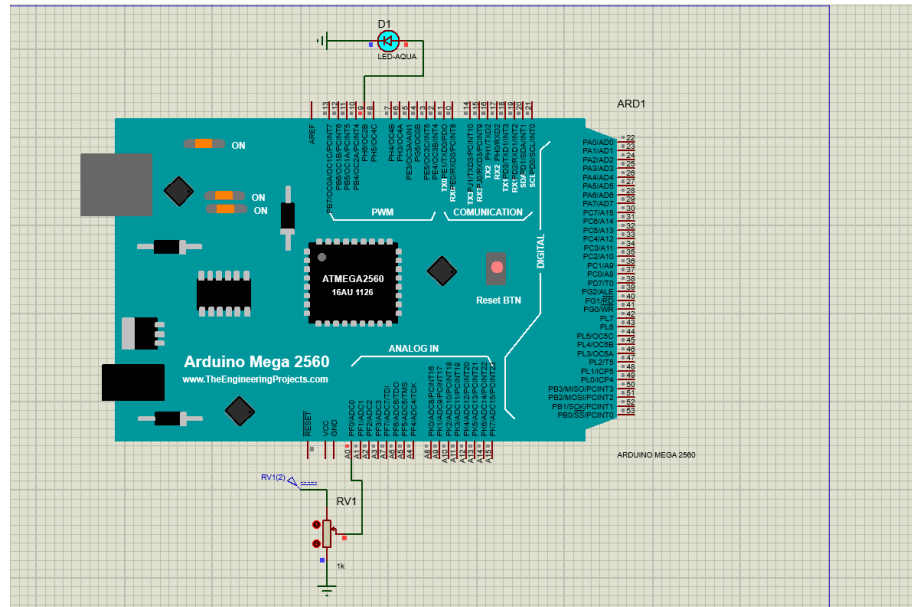
Sketch uses 1842 bytes (6%) of program storage space. Maximum is 253952 bytes.  
Global variables use 11 bytes (0%) of dynamic memory, leaving 8181 bytes for local variables. Maximum is 8192 bytes.

Building sketch

In 11, Col 79 Arduino Mega or Mega 2560 on COM12 8:28 PM 10/2/2024

## Gambar Rangkaian:





Gambar rangkaian di Proteus

### Hasil dari program ini:

**LED Berkedip:** LED akan berkedip dengan kecepatan yang berubah-ubah sesuai dengan posisi potensiometer.

**Kecepatan Berkedip:** Potensiometer mengatur delay antara nyala dan matinya LED. Ketika potensiometer berada pada posisi minimum, LED akan berkedip cepat (delay 10 ms), sedangkan pada posisi maksimum, LED akan berkedip lambat (delay 1000 ms).

### Gambaran Rangkaian:

#### 1. LED:

Katoda (kaki pendek) terhubung ke ground.

Anoda (kaki panjang) terhubung ke resistor (misalnya 220 ohm), kemudian ke pin digital 9 Arduino Mega 2560.

#### 2. Potensiometer:

Salah satu pin luar terhubung ke 5V pada Arduino.

Pin luar lainnya terhubung ke ground (GND).



Pin tengah (wiper) terhubung ke pin analog A0 pada Arduino.

### 3. Cara Kerja Rangkaian:

Potensiometer: Membaca nilai analog (0 hingga 1023) dari potensiometer, yang menunjukkan posisi wiper antara 0V hingga 5V.

LED: Berdasarkan nilai yang dibaca dari potensiometer, program menghitung waktu delay antara keadaan LED menyala dan mati menggunakan fungsi `map()`. LED kemudian akan berkedip dengan frekuensi yang bervariasi, tergantung pada nilai potensiometer.

- d. Buat program untuk membaca nilai LDR dan mengendalikan kecerahan LED secara otomatis.

#### Code Program:

```
int ldrPinAO = A0; // Pin untuk AO pada modul LDR
int ldrPinDO = 2; // Pin untuk DO pada modul LDR
int ledPin = 9;    // Pin untuk LED

int ldrValue;      // Variabel untuk menyimpan nilai analog dari LDR (AO)
int ledBrightness; // Variabel untuk menyimpan kecerahan LED
int digitalState;  // Variabel untuk menyimpan nilai digital dari LDR (DO)

void setup() {
  pinMode(ledPin, OUTPUT); // Mengatur pin LED sebagai output
  pinMode(ldrPinDO, INPUT); // Mengatur pin DO sebagai input
  Serial.begin(9600);      // Memulai komunikasi serial untuk debugging
}

void loop() {
  // Membaca nilai dari AO (Analog Output)
  ldrValue = analogRead(ldrPinAO);
  // Mengubah nilai LDR menjadi kecerahan LED (0-255)
```

```
ledBrightness = map(ldrValue, 0, 1023, 255, 0); // Semakin terang cahayanya,
semakin redup LED
```

```
// Mengontrol kecerahan LED berdasarkan nilai analog dari AO
analogWrite(ledPin, ledBrightness);
```

```
// Membaca nilai dari DO (Digital Output)
digitalState = digitalRead(ldrPinDO);
```

```
// Menyalakan atau mematikan LED berdasarkan nilai digital DO
if (digitalState == LOW) {
    // Jika gelap (nilai DO = LOW), nyalakan LED
    digitalWrite(ledPin, LOW);
} else {
    // Jika terang (nilai DO = HIGH), matikan LED
    digitalWrite(ledPin, HIGH);
}
```

```
// Menampilkan nilai LDR, kecerahan LED, dan status DO di Serial Monitor
Serial.print("Nilai LDR (AO): ");
Serial.print(ldrValue);
Serial.print(" | Kecerahan LED: ");
Serial.print(ledBrightness);
Serial.print(" | Status DO: ");
Serial.println(digitalState);
```

```
delay(100); // Menunggu sejenak sebelum membaca kembali
nilai LDR dan DO
}
```

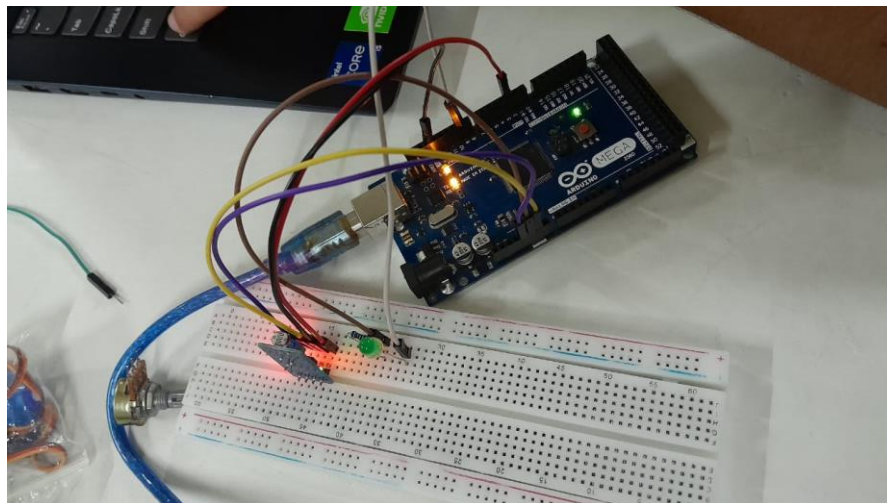
## Output Program:

```
sketch_aug15a.ino
19 ledBrightness = map(ldrValue, 0, 1023, 255, 0); // Semakin terang cahayanya, semakin redup LED
20
21 // Mengontrol kecerahan LED berdasarkan nilai analog dari AO
22 analogWrite(ledPin, ledBrightness);
23
24 // Membaca nilai dari DO (Digital Output)
25 digitalWriteState = digitalRead(ldrPin00);
26
27 // Menyalakan atau mematikan LED berdasarkan nilai digital DO
28 if (digitalState == LOW) {
29   // Jika gelap (nilai DO = LOW), nyalakan LED
30   digitalWrite(ledPin, LOW);
31 } else {
32   // Jika terang (nilai DO = HIGH), matikan LED
33   digitalWrite(ledPin, HIGH);
34 }
35
36 // Menampilkan nilai LDR, kecerahan LED, dan status DO di Serial Monitor
37 Serial.print("Nilai LDR (AO): ");
38 Serial.print(ldrValue);
39 Serial.print(" | Kecerahan LED: ");
40 Serial.print(ledBrightness);
41 Serial.print(" | Status DO: ");
```

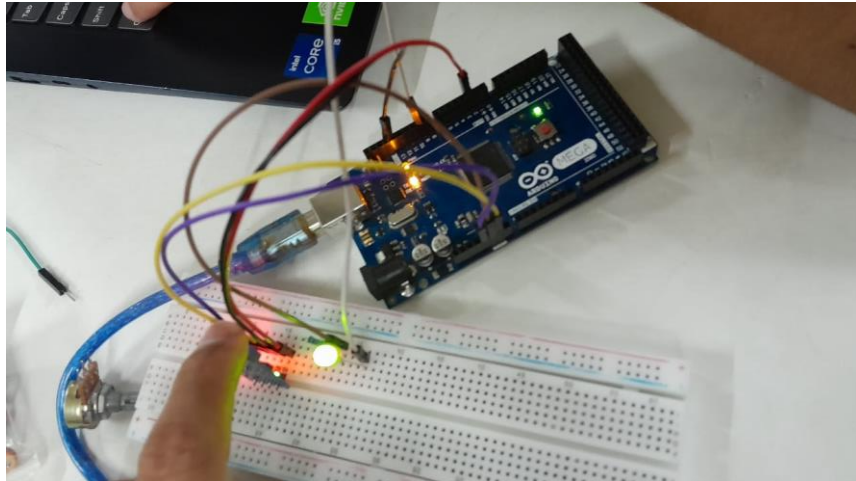
Output Serial Monitor

Message (Enter to send message to 'Arduino Mega or Mega 2560' on 'COM12')	Now Line	9600 baud
Nilai LDR (AO): 451   Kecerahan LED: 143   Status DO: 0		
Nilai LDR (AO): 496   Kecerahan LED: 132   Status DO: 0		
Nilai LDR (AO): 532   Kecerahan LED: 123   Status DO: 0		
Nilai LDR (AO): 572   Kecerahan LED: 113   Status DO: 0		
Nilai LDR (AO): 617   Kecerahan LED: 102   Status DO: 0		
Nilai LDR (AO): 647   Kecerahan LED: 94   Status DO: 0		
Nilai LDR (AO): 669   Kecerahan LED: 89   Status DO: 0		
Nilai LDR (AO): 688   Kecerahan LED: 84   Status DO: 0		
Nilai LDR (AO): 700   Kecerahan LED: 81   Status DO: 0		

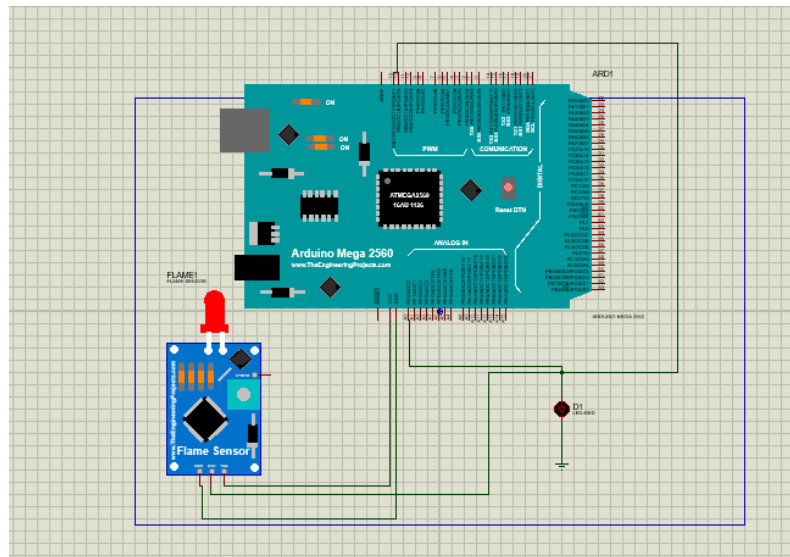
## Gambar Rangkaian:



Sebelum adanya gelap



Setelah sensor mendeteksi gelap



Gambar rangkaian di Proteus

**a) Komponen yang Diperlukan:**

- Arduino Mega 2560
- Modul LDR 4 Kaki
- LED
- Resistor 220 ohm (untuk LED)
- Breadboard
- Kabel Jumper

**b) Fungsi Modul LDR 4 Kaki**

- AO (Analog Output): Menghasilkan nilai analog yang mencerminkan intensitas cahaya (0-1023).
- DO (Digital Output): Menghasilkan nilai digital HIGH atau LOW tergantung pada intensitas cahaya dan ambang batas yang ditetapkan pada modul.

**c) Hubungan Rangkaian**

**Modul LDR:**

- VCC dari modul terhubung ke 5V pada Arduino Mega 2560.
- GND dari modul terhubung ke GND pada Arduino.
- AO (Analog Output) dari modul terhubung ke Analog Pin A0 pada Arduino.
- DO (Digital Output) dari modul terhubung ke Digital Pin 2 pada Arduino.

**LED:**

- Anoda (kaki panjang) LED terhubung ke Digital Pin 9 pada Arduino Mega 2560.
- Katoda (kaki pendek) LED terhubung ke GND melalui resistor 220 ohm.