

# **LAPORAN PRAKTIKUM ARSIKOM**

(Arsitektur dan Organisasi Komputer)



**Disusun Oleh :**

**Nama : Zinniarethie Andari Kostiene**  
**NIM : 0903058226007**  
**Prodi : Teknik Komputer**  
**Dosen : Adi Hermansyah, S.Kom., M.T**

**LABORATORIUM PERANGKAT KERAS**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS SRIWIJAYA**  
**PALEMBANG 2023**

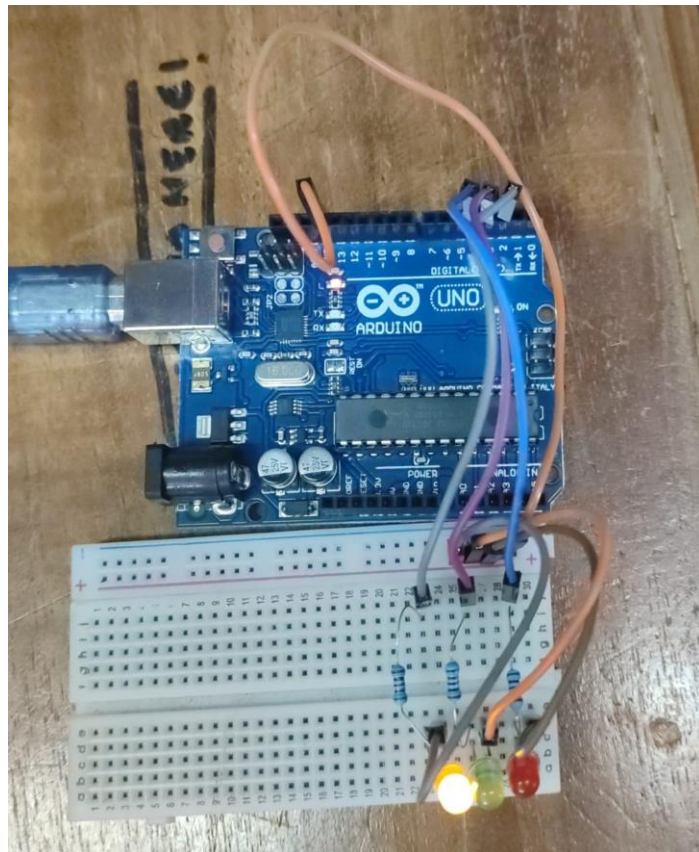
## I. Pendahuluan

Setelah laporan praktik kemarin membuat 5 kondisi dengan arduino di wokwi. Di laporan praktik ini, saya akan melaporkan praktik langsung 5 kondisi lampu tersebut dengan arduino yang real dan dihubungkan ke laptop juga disupport dengan aplikasi arduino itu sendiri. Pada laporan praktikum ini juga, saya juga akan menampilkan praktik dengan arduino asli yang menggunakan servo.

## II. 5 Kondisi Lampu

### a. Gambar

Di bawah ini adalah gambar hasil praktik praktikum yang berupa lima kondisi lampu dengan 3 LED di mana berbeda dengan simulasi kemarin yang menggunakan 7 LED.



## b. Codingan dan tampilan aplikasi arduino

Setelahnya, dengan aplikasi arduino dan codingan untuk 5 kondisi lampu. Inilah tampilan 5 kondisi codingan dan aplikasi arduino itu sendiri.

```
int ledPins[] = {2, 3, 4}; // Pin LED yang akan digunakan
int numLeds = 3; // Jumlah LED yang terdapat
int delayTime = 1000; // Waktu jeda
int brightness = 255; // Kecerahan untuk kondisi ke-5
int fadeAmount = 5; // Pemudaran untuk kondii ke-5

void setup() {
  for (int i = 0; i < numLeds; i++) {
    pinMode(ledPins[i], OUTPUT);
  }
}

void loop() {
  //1. lampu LED akan menyala secara kelap-kelip
  blinkled();
  //2. lampu LED akan menyala bersamaan
  allLedOn();
  //3. lampu LED akan menyala dengan delay yang berbeda-beda
  ledDelay();
  //4. lampu LED akan menyala bergantian dari Kiri ke Kanan dan sebaliknya
  rightToLeftLeds();
  //5. lampu LED akan menyala bergantian dari redup ke terang
  fadeToBrightLeds();
}
```

- 1) Kode ini dirancang untuk mengendalikan sejumlah LED yang terhubung ke pin-pin tertentu pada papan Arduino. Di bawah ini adalah penjelasannya:

1. `int ledPins[] = {2, 3, 4};` : Ini adalah deklarasi array yang berisi nomor pin yang terhubung ke LED. Dalam kasus ini, ada 3 LED yang terhubung ke pin 2 hingga 4.

2. `int numLeds = 3;` : Variabel ini menentukan jumlah LED yang terhubung, yaitu 3 LED.

3. `int delayTime = 1000;` : Variabel ini menentukan waktu jeda dalam milidetik antara setiap perubahan keadaan LED. Dalam contoh ini, waktu jeda adalah 1000 ms atau 1 detik.

4. `int brightness = 255;` : Variabel ini menentukan kecerahan maksimum untuk kondisi ke-5 (yang mungkin akan ditentukan nanti dalam kode, tetapi belum ada dalam potongan kode yang Anda berikan).

5. `int fadeAmount = 5;` : Ini adalah variabel yang menentukan seberapa cepat LED akan memudar atau berubah ke keadaan berikutnya (mungkin juga akan digunakan nanti dalam kode selanjutnya).

6. Dalam fungsi `setup()`, ada loop `for` yang menginisialisasi setiap pin yang terhubung ke LED sebagai OUTPUT (keluaran) menggunakan perintah `pinMode`. Ini mempersiapkan pin-pin tersebut untuk mengontrol LED.

Lalu ada void setup () yang mengatur berbagai fungsi yang akan dijalankan secara berulang dalam fungsi loop(). Setiap fungsi memiliki tujuan khusus dalam mengendalikan lampu LED yang terhubung ke pin-pin yang telah ditentukan dalam kode sebelumnya. Berikut adalah penjelasan tentang setiap fungsi yang dipanggil dalam loop:

1. `blinkled()`: Fungsi ini mengatur lampu LED agar berkedip secara berulang
2. `allLedOn()`: Fungsi ini mengatur semua lampu LED agar menyala secara bersamaan.
3. `LedDelay()`: Fungsi ini mengatur lampu LED agar menyala satu per satu dengan delay (waktu jeda) yang berbeda-beda antara satu LED dengan yang lainnya.
4. `rightToleftLeds()`: Fungsi ini mengatur lampu LED agar menyala secara bergantian dari kiri ke kanan dan sebaliknya.
5. `fadeTobrightLeds()`: Fungsi ini mengatur lampu LED agar menyala bergantian dari redup ke terang.

```
//1. lampu LED akan menyala secara kelap-kelip

void blinkled() {
  for (int i = 0; i < numLeds; i++) {
    digitalWrite(ledPins[i], HIGH);
  }
  delay(500);

  for (int i = 0; i < numLeds; i++) {
    digitalWrite(ledPins[i], LOW);
  }
  delay(500);
}
```

- 2) Fungsi ``blinkled()`` dalam kode Arduino yang Anda berikan mengendalikan lampu LED yang terhubung ke pin-pin yang telah ditentukan dalam array ``ledPins``. Fungsi ini digunakan untuk membuat lampu LED berkedip secara bergantian. Berikut adalah penjelasan rinci tentang cara kerja fungsi ``blinkled()``:

1. Loop Pertama:

- Pada awalnya, fungsi ``for`` pertama akan memulai pengulangan untuk semua LED dalam array ``ledPins``.
- Dengan menggunakan perintah ``digitalWrite(ledPins[i], HIGH);``, lampu LED pada pin ke-i akan diatur menjadi HIGH (menyala), sehingga menyalakan semua LED yang ada dalam array.

2. ``delay(500);``:

- Setelah semua LED dinyalakan, perintah ``delay(500);`` akan menghentikan eksekusi program selama 500 milidetik (setengah detik). Ini menciptakan jeda selama setengah detik sebelum langkah berikutnya.

### 3. Loop Kedua:

- Setelah jeda, fungsi `for` kedua akan dijalankan untuk semua LED dalam array `ledPins`.
- Dengan menggunakan perintah `digitalWrite(ledPins[i], LOW);`, lampu LED pada pin ke-i akan diatur menjadi LOW (mati), sehingga mematikan semua LED yang telah dinyalakan sebelumnya.

### 4. `delay(500);`:

- Setelah semua LED dimatikan, perintah `delay(500);` akan kembali memberikan jeda selama setengah detik.

```
//2. lampu LED akan menyala bersamaan

void allLedOn() {
  for (int i = 0; i < numLeds; i++) {
    digitalWrite(ledPins[i], HIGH);
  }
  delay(2000);
  for (int i = 0; i < numLeds; i++) {
    digitalWrite(ledPins[i], LOW);
  }
  delay(1000);
}
```

- 3) Fungsi `allLedOn()` dalam kode Arduino Anda mengendalikan lampu LED yang terhubung ke pin-pin yang telah ditentukan dalam array `ledPins`. Fungsi ini digunakan untuk mengaktifkan semua lampu LED secara bersamaan, menjaga mereka menyala selama dua detik, dan kemudian memmatikannya selama satu detik. Berikut adalah penjelasan rinci tentang cara kerja fungsi `allLedOn()`:

### 1. Loop Pertama:

- Pada awalnya, fungsi `for` pertama akan memulai pengulangan untuk semua LED dalam array `ledPins`.
- Dengan menggunakan perintah `digitalWrite(ledPins[i], HIGH);`, lampu LED pada pin ke-i akan diatur menjadi HIGH (menyala), sehingga menyalakan semua LED yang ada dalam array.

### 2. `delay(2000);`:

- Setelah semua LED dinyalakan, perintah `delay(2000);` akan menghentikan eksekusi program selama 2000 milidetik (2 detik). Ini menciptakan jeda selama dua detik sebelum langkah berikutnya.

### 3. Loop Kedua:

- Setelah jeda, fungsi `for` kedua akan dijalankan untuk semua LED dalam array `ledPins`.
- Dengan menggunakan perintah `digitalWrite(ledPins[i], LOW);`, lampu LED pada pin ke-i akan diatur menjadi LOW (mati), sehingga mematikan semua LED yang telah dinyalakan sebelumnya.

### 4. `delay(1000);`:

- Setelah semua LED dimatikan, perintah `delay(1000);` akan kembali memberikan jeda selama 1000 milidetik (1 detik).

Hasil dari fungsi `allLedOn()` adalah bahwa semua lampu LED akan dinyalakan secara bersamaan, selanjutnya, mereka akan tetap menyala selama dua detik, dan kemudian akan dimatikan selama satu detik. Proses ini akan diulangi terus menerus saat berada dalam fungsi `loop()`. Ini menciptakan efek lampu LED yang menyala bersamaan, dengan interval yang telah ditentukan, dan mematikan sesuai dengan interval yang berbeda. Anda dapat mengganti nilai dalam perintah `delay` sesuai keinginan untuk mengubah durasi penyalaaan dan pemadaman lampu LED.

```
//3. lampu LED akan menyala dengan delay yang berbeda-beda

void LedDelay() {
  int delays[] = {500, 1000, 1500, 2000, 2500, 3000, 3500};
  for (int i = 0; i < numLeds; i++) {
    digitalWrite(ledPins[i], HIGH);
    delay(delays[i]);
    digitalWrite(ledPins[i], LOW);
  }
}
```

- 4) Fungsi `LedDelay()` dalam kode Arduino Anda mengendalikan lampu LED yang terhubung ke pin-pin yang telah ditentukan dalam array `ledPins`. Fungsi ini digunakan untuk mengatur lampu LED agar menyala dengan delay (waktu jeda) yang berbeda-beda antara satu LED dengan yang lainnya. Berikut adalah penjelasan rinci tentang cara kerja fungsi `LedDelay()`:

1. Array `delays[]`:
  - Array ini berisi nilai-nilai waktu jeda (delay) dalam milidetik untuk masing-masing LED. Terdapat 7 nilai yang sesuai dengan jumlah LED yang Anda miliki.
2. Loop:
  - Fungsi `for` akan memulai pengulangan sebanyak jumlah LED dalam array `ledPins`.
3. `digitalWrite(ledPins[i], HIGH);`:
  - Pada awal setiap iterasi, LED pada pin ke-i akan diatur menjadi HIGH (menyala) menggunakan perintah ini.
4. `delay(delays[i]);`:
  - Setelah mengaktifkan LED, ada perintah `delay(delays[i]);` yang akan menghentikan eksekusi program selama jangka waktu yang sesuai dengan nilai yang ada dalam array `delays[]`. Ini menciptakan jeda waktu yang berbeda-beda untuk setiap LED sebelum langkah selanjutnya.
5. `digitalWrite(ledPins[i], LOW);`:
  - Setelah jeda, perintah ini akan mematikan LED pada pin ke-i dengan mengaturnya menjadi LOW (mati).

Hasil dari fungsi `LedDelay()` adalah bahwa lampu LED akan dinyalakan satu per satu dengan waktu jeda yang berbeda antara satu LED dengan yang lainnya sesuai dengan nilai-nilai dalam array `delays[]`. Ini menciptakan efek LED yang menyala dengan delay yang berbeda, sehingga tampak seperti berkedip dengan pola jeda yang bervariasi. Proses ini akan diulangi terus menerus saat berada dalam fungsi `loop()`. Anda dapat mengubah nilai-nilai

dalam array `delays[]` sesuai keinginan untuk mengatur pola berkedip dan waktu jeda yang sesuai dengan kebutuhan Anda.

```
//4. lampu LED akan menyala bergantian dari Kiri ke Kanan dan sebaliknya

void rightToleftLeds(){
  for (int i = 0; i < numLeds; i++) {
    digitalWrite(ledPins[i], HIGH);
    delay(delayTime);
    digitalWrite(ledPins[i], LOW);
  }
  for (int i = numLeds - 1; i >= 0; i--) {
    digitalWrite(ledPins[i], HIGH);
    delay(delayTime);
    digitalWrite(ledPins[i], LOW);
  }
}
```

- 5) Fungsi `rightToleftLeds()` dalam kode Arduino mengendalikan lampu LED yang terhubung ke pin-pin yang telah ditentukan dalam array `ledPins`. Fungsi ini menciptakan efek di mana lampu LED menyala secara bergantian dari kiri ke kanan dan sebaliknya. Berikut adalah penjelasan rinci tentang cara kerja fungsi `rightToleftLeds()`:

1. Loop Pertama:

- Fungsi `for` pertama akan memulai pengulangan dengan mengatur `i` ke 0. Ini digunakan untuk mengaktifkan lampu LED dari kiri ke kanan.
- Dalam setiap iterasi loop, lampu LED pada pin ke-i akan diatur menjadi HIGH (menyala) menggunakan perintah `digitalWrite(ledPins[i], HIGH);`.
- Kemudian, perintah `delay(delayTime);` memberikan jeda selama jumlah waktu yang ditentukan dalam variabel `delayTime`.
- Setelah jeda, lampu LED pada pin ke-i akan dimatikan dengan mengatur menjadi LOW (mati) menggunakan perintah `digitalWrite(ledPins[i], LOW);`.

2. Loop Kedua:

- Setelah loop pertama selesai, fungsi `for` kedua akan dijalankan, tetapi kali ini dengan mengatur `i` ke nilai `numLeds - 1`. Ini digunakan untuk mengaktifkan lampu LED dari kanan ke kiri.
- Dalam setiap iterasi loop, lampu LED pada pin ke-i akan diatur menjadi HIGH (menyala).
- Setelah itu, ada perintah `delay(delayTime);` yang memberikan jeda selama jumlah waktu yang sama seperti sebelumnya.
- Setelah jeda, lampu LED pada pin ke-i akan dimatikan dengan mengatur menjadi LOW.

Hasil dari fungsi `rightToleftLeds()` adalah bahwa lampu LED akan menyala bergantian dari kiri ke kanan dan kemudian sebaliknya, menciptakan efek bergerak atau melintasi deretan lampu LED. Proses ini akan diulangi terus menerus saat berada dalam fungsi `loop()`. Anda dapat mengubah nilai dalam variabel `delayTime` sesuai keinginan untuk mengatur kecepatan perpindahan lampu LED.

```
//5. lampu LED akan menyala bergantian dari redup ke terang

void fadeTobrightLeds(){
  for (int i = 0; i < numLeds; i++){
    analogWrite(ledPins[i], brightness);
    brightness = brightness + fadeAmount;
    if (brightness <= 0 || brightness >= 255) {
      fadeAmount = -fadeAmount;
    }
    delay(delayTime);
  }
  if (brightness <= 255 || brightness >= 0) {
    fadeAmount = +fadeAmount;
  }
  delay(delayTime);
}
}
```

- 6) Fungsi `fadeTobrightLeds()` dalam kode Arduino mengendalikan lampu LED yang terhubung ke pin-pin yang telah ditentukan dalam array `ledPins`. Fungsi ini menciptakan efek fading pada lampu LED, yaitu lampu LED akan menyala dari redup ke terang dan kemudian kembali dari terang ke redup. Berikut adalah penjelasan rinci tentang cara kerja fungsi `fadeTobrightLeds()`:

1. Loop:

- Fungsi `for` akan memulai pengulangan sebanyak jumlah LED dalam array `ledPins`. Dalam setiap iterasi loop, kode berikut dijalankan.

2. `analogWrite(ledPins[i], brightness);`:

- Perintah ini mengatur tingkat kecerahan (`brightness`) lampu LED pada pin ke-*i* menggunakan modulasi lebar pulsa (PWM) dengan nilai kecerahan yang disimpan dalam variabel `brightness`.

3. `brightness = brightness + fadeAmount;`:

- Nilai `brightness` ditingkatkan atau dikurangi sebesar `fadeAmount`. Ini akan mengubah tingkat kecerahan lampu LED secara bertahap.

4. `if (brightness <= 0 || brightness >= 255) { fadeAmount = -fadeAmount; }`:

- Pemeriksaan kondisi ini akan mengubah arah peningkatan atau penurunan kecerahan (`fadeAmount`) saat `brightness` mencapai batas tertentu. Jika `brightness` mencapai 0 atau 255, maka nilai `fadeAmount` akan diubah menjadi nilai negatifnya, sehingga kecerahan akan berubah dari redup ke terang atau sebaliknya.

5. `delay(delayTime);`:

- Perintah ini memberikan jeda waktu selama jumlah waktu yang ditentukan dalam variabel `delayTime`.

6. `if (brightness <= 255 || brightness >= 0) { fadeAmount = +fadeAmount; delay(delayTime); }`:

- Ini adalah pemeriksaan kondisi tambahan yang membalikkan arah peningkatan atau penurunan `fadeAmount` saat `brightness` mencapai batas tertentu. Ini menciptakan efek fading yang terus berlanjut dari redup ke terang dan sebaliknya.

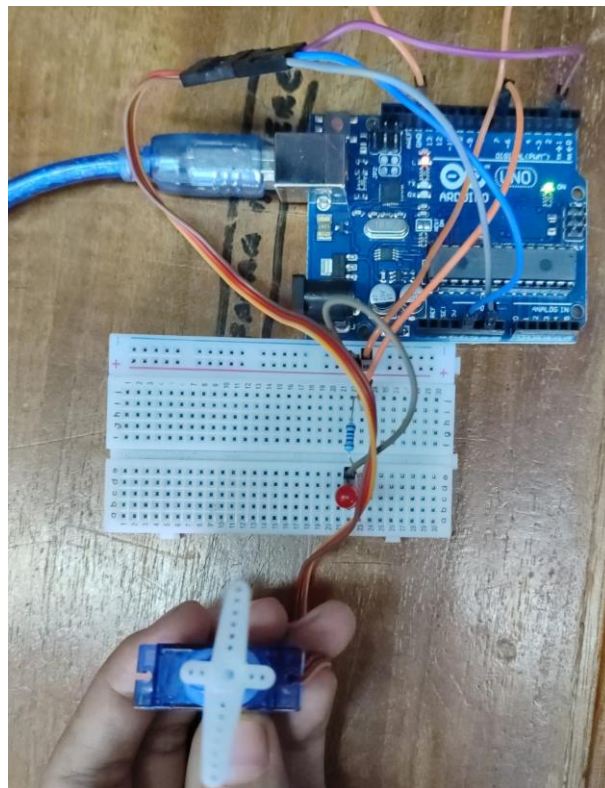


Hasil dari fungsi `fadeTobrightLeds()` adalah efek fading yang mengubah kecerahan lampu LED secara bertahap dari redup ke terang dan sebaliknya. Proses ini akan terus berlanjut selama berada dalam fungsi `loop()`. Anda dapat mengatur nilai awal `brightness`, `fadeAmount`, dan `delayTime` sesuai dengan keinginan Anda untuk mengubah kecepatan dan rentang kecerahan efek fading.

### III. Servo

#### a. Gambar

Di bawah ini adalah gambar hasil praktik praktikum yang berupa arduino yang menggunakan servo dan satu buah lampu LED.



## b. Codingan dan tampilan aplikasi arduino

Berikut adalah codingan untuk arduino yang menggunakan servo:

```
#include <Servo.h>

Servo servo; // Membuat objek Servo

int servoPin = 2; // Pin yang terhubung ke servo motor

void setup() {
  servo.attach(servoPin); // Menghubungkan objek servo dengan pin servo
}

void loop() {
  // Posisi awal servo (0 derajat)
  servo.write(0);
  delay(1000);

  // Posisi akhir servo (90 derajat)
  servo.write(90);
  delay(1000);
}
```

Program ini mengatur servo untuk bergerak dari posisi awal (0 derajat) ke posisi akhir (90 derajat) dengan jeda 1 detik antara pergerakan servo. Berikut adalah penjelasan baris per baris dari kode tersebut:

1. `#include <Servo.h>`: Ini adalah pernyataan preprocessor yang memasukkan perpustakaan Servo untuk digunakan dalam program. Perpustakaan Servo digunakan untuk mengontrol motor servo dengan mudah.
2. `Servo servo;`: Ini adalah deklarasi objek `servo` dari kelas `Servo`. Objek ini akan digunakan untuk mengendalikan motor servo.
3. `int servoPin = 2;`: Ini adalah deklarasi variabel `servoPin` yang menyimpan nomor pin Arduino yang terhubung ke motor servo.
4. `void setup() {`: Ini adalah fungsi `setup()`, yang dieksekusi sekali saat papan Arduino dihidupkan. Di dalamnya:
  - `servo.attach(servoPin);`: Ini menghubungkan objek `servo` dengan pin yang ditentukan dalam `servoPin`. Ini mengatakan kepada Arduino bahwa motor servo akan digunakan pada pin tersebut.
5. `void loop() {`: Ini adalah fungsi `loop()`, yang berjalan terus-menerus setelah `setup()` selesai. Di dalamnya, ada dua blok perintah yang diulang:
  - `servo.write(0);`: Perintah ini mengatur posisi motor servo ke 0 derajat, yang merupakan posisi awal. Servo akan bergerak ke posisi ini.
  - `delay(1000);`: Perintah ini menunda eksekusi program selama 1000

milidetik (1 detik).

- Kemudian, program mengulang perintah yang sama untuk menggerakkan servo ke posisi 90 derajat (`servo.write(90);`) dan menunggu selama 1 detik lagi (`delay(1000);`).

Dengan demikian, program ini menggerakkan motor servo dari posisi awal (0 derajat) ke posisi akhir (90 derajat) dengan jeda 1 detik antara pergerakan. Hal ini dapat digunakan sebagai contoh dasar untuk mengendalikan motor servo dengan menggunakan Arduino.