

LAPORAN PRAKTIKUM 4

TRAFFIC LIGHT 4 ARAH

Dosen Pengampu:

Erik Haritman, S.Pd. M.T.



Disusun oleh:

Muhammad Ramdan

(NIM: 1904637)

DEPARTEMEN PENDIDIKAN TEKNIK ELEKTRO
FAKULTAS PENDIDIKAN TEKNIK DAN KEJURUAN
UNIVERSITAS PENDIDIKAN INDONESIA

2021

DAFTAR ISI

DAFTAR ISI	1
PENDAHULUAN	2
Tujuan Praktikum	2
Kajian Teori	2
PROSEDUR PRAKTIKUM	3
Daftar Komponen dan Alat	3
Prosedur Praktikum.....	3
HASIL PRAKTIKUM	7
Hasil Praktikum	7
Pembahasan	8
Kesimpulan	8

PENDAHULUAN

Tujuan Praktikum

Setelah membaca modul diharapkan siswa dapat:

1. Membuat program Traffic Light menggunakan Arduino dengan benar sesuai rancangannya.

Kajian Teori

1. Lampu Lalu Lintas

Lampu lalu lintas atau dalam Bahasa Inggris disebut Traffic Light merupakan alat pemberi isyarat lalu lintas (APILL) yang berupa lampu dan digunakan untuk mengendalikan arus lalu lintas yang terpasang di persimpangan jalan, tempat penyeberangan pejalan kaki/zebra cross, dan tempat arus lalu lintas lainnya. Lampu ini yang menandakan kapan kendaraan harus berjalan dan berhenti secara bergantian dari berbagai arah. Pengaturan lalu lintas di persimpangan jalan dimaksudkan untuk mengatur pergerakan kendaraan pada masing-masing kelompok pergerakan kendaraan agar dapat bergerak secara bergantian sehingga tidak saling mengganggu antar-arus yang ada.

Lampu lalu lintas telah diadopsi di hampir semua kota di dunia ini. Lampu ini menggunakan warna yang diakui secara universal; untuk menandakan berhenti adalah warna merah, hati-hati ditandai dengan warna kuning, dan hijau yang berarti boleh maju.

2. Mikrokontroler

Mikrokontroler adalah sistem komputer yang dikemas dalam sebuah Integrated Circuit (IC). Dimana didalam IC terdapat komponen-komponen penting yang ada pada komputer pada umumnya seperti komputer Central Processing Unit (CPU), RAM, ROM, Port IO. Berbeda dengan PC yang umumnya dirancang untuk digunakan secara umum, mikrokontroler sendiri biasanya dirancang hanya untuk mengerjakan tugas atau fungsi yang khusus saja (special purpose) yaitu mengontrol sistem tertentu.

Orang-orang juga menyebut Mikrokontroler sebagai Embedded Mikrokontroler, hal ini tidak terlepas dari posisi mikrokontroler yang embedded system atau menjadi satu bagian dengan perangkat sistem atau suatu sistem yang lebih besar.

Secara sederhana Mikrokontroler dapat diartikan sebagai suatu sistem komputer yang dikemas dalam IC, dimana sebelum digunakan harus diisi suatu program atau perintah terlebih dahulu sehingga mikrokontroler hanya dapat berjalan bila telah diisi suatu perintah atau program terlebih dahulu.

Salah satu mikrokontroller sederhana yang paling dikenal adalah Arduino, mikrokontroller yang sudah siap pakai dan memiliki aplikasinya tersendiri sehingga tidak perlu diprogram terlalu rumit.

3. Arduino IDE

Untuk memprogram board Arduino, kita butuh aplikasi IDE (Integrated Development Environment) bawaan dari Arduino. Aplikasi ini berguna untuk membuat, membuka, dan mengedit source code Arduino (Sketches, para programmer menyebut source code arduino dengan istilah "sketches"). Selanjutnya, jika kita menyebut source code yang ditulis untuk Arduino, kita sebut "sketch" juga. Sketch merupakan source code yang berisi logika dan algoritma yang akan diupload ke dalam IC mikrokontroller (Arduino).

PROSEDUR PRAKTIKUM

Daftar Komponen dan Alat

Komponen:

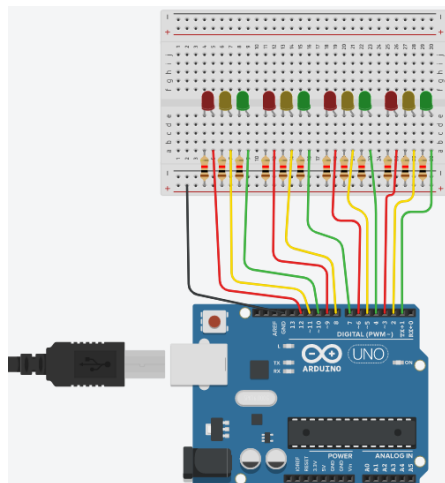
1. LED merah 4 buah
2. LED hijau 4 buah
3. LED kuning 4 buah
4. Resistor 12 buah

Alat:

1. Arduino UNO
2. Laptop
3. TinkerCAD

Prosedur Praktikum

1. Siapkan komponen dan alat.
2. Buat rangkaian seperti berikut



Gambar 1 rangkaian arduino

3. Masukkan script sebagai berikut

```
// C++ code
//
void setup()
{
  pinMode(1, OUTPUT);
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
}
```

```

}

void loop()
{
    digitalWrite(1, HIGH);
    digitalWrite(2, LOW);
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
    digitalWrite(5, LOW);
    digitalWrite(6, HIGH);
    digitalWrite(7, LOW);
    digitalWrite(8, LOW);
    digitalWrite(9, HIGH);
    digitalWrite(10, LOW);
    digitalWrite(11, LOW);
    digitalWrite(12, HIGH);
    delay(1000);

    digitalWrite(1, LOW);
    digitalWrite(2, LOW);
    digitalWrite(3, HIGH);
    digitalWrite(4, LOW);
    digitalWrite(5, HIGH);
    digitalWrite(6, LOW);
    digitalWrite(7, LOW);
    digitalWrite(8, LOW);
    digitalWrite(9, HIGH);
    digitalWrite(10, LOW);
    digitalWrite(11, LOW);
    digitalWrite(12, HIGH);
    delay(1000);

    digitalWrite(1, LOW);
    digitalWrite(2, LOW);
    digitalWrite(3, HIGH);
    digitalWrite(4, HIGH);
    digitalWrite(5, LOW);
    digitalWrite(6, LOW);
    digitalWrite(7, LOW);
    digitalWrite(8, LOW);
    digitalWrite(9, HIGH);
    digitalWrite(10, LOW);
    digitalWrite(11, LOW);
    digitalWrite(12, HIGH);
    delay(1000);

    digitalWrite(1, LOW);
    digitalWrite(2, LOW);
    digitalWrite(3, HIGH);
    digitalWrite(4, LOW);
    digitalWrite(5, LOW);
    digitalWrite(6, HIGH);
    digitalWrite(7, LOW);
    digitalWrite(8, HIGH);
    digitalWrite(9, LOW);

```

```
digitalWrite(10, LOW);  
digitalWrite(11, LOW);  
digitalWrite(12, HIGH);  
delay(1000);
```

```
digitalWrite(1, LOW);  
digitalWrite(2, LOW);  
digitalWrite(3, HIGH);  
digitalWrite(4, LOW);  
digitalWrite(5, LOW);  
digitalWrite(6, HIGH);  
digitalWrite(7, HIGH);  
digitalWrite(8, LOW);  
digitalWrite(9, LOW);  
digitalWrite(10, LOW);  
digitalWrite(11, LOW);  
digitalWrite(12, HIGH);  
delay(1000);
```

```
digitalWrite(1, LOW);  
digitalWrite(2, LOW);  
digitalWrite(3, HIGH);  
digitalWrite(4, LOW);  
digitalWrite(5, LOW);  
digitalWrite(6, HIGH);  
digitalWrite(7, LOW);  
digitalWrite(8, LOW);  
digitalWrite(9, HIGH);  
digitalWrite(10, LOW);  
digitalWrite(11, HIGH);  
digitalWrite(12, LOW);  
delay(1000);
```

```
digitalWrite(1, LOW);  
digitalWrite(2, LOW);  
digitalWrite(3, HIGH);  
digitalWrite(4, LOW);  
digitalWrite(5, LOW);  
digitalWrite(6, HIGH);  
digitalWrite(7, LOW);  
digitalWrite(8, LOW);  
digitalWrite(9, HIGH);  
digitalWrite(10, HIGH);  
digitalWrite(11, LOW);  
digitalWrite(12, LOW);  
delay(1000);
```

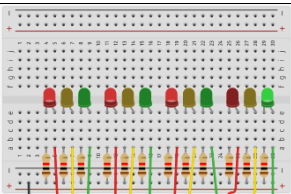
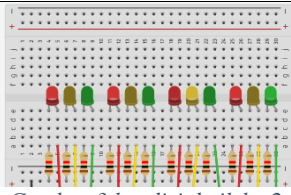
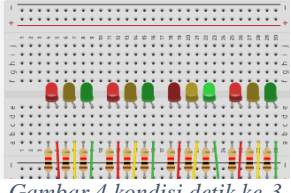
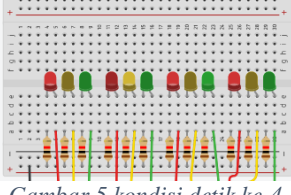
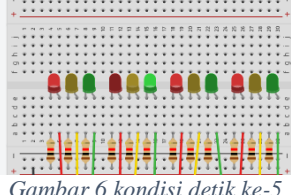
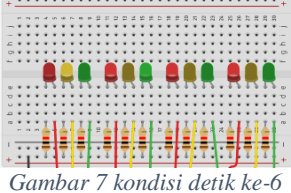
```
digitalWrite(1, LOW);  
digitalWrite(2, HIGH);  
digitalWrite(3, LOW);  
digitalWrite(4, LOW);  
digitalWrite(5, LOW);  
digitalWrite(6, HIGH);  
digitalWrite(7, LOW);  
digitalWrite(8, LOW);
```

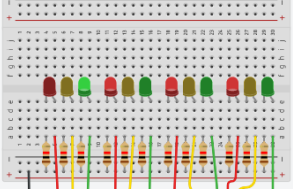
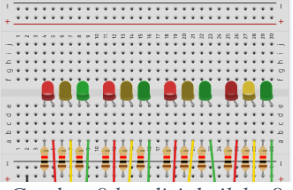
```
digitalWrite(9, HIGH);  
digitalWrite(10, LOW);  
digitalWrite(11, LOW);  
digitalWrite(12, HIGH);  
delay(1000);  
}
```

Tabel 1 script program

HASIL PRAKTIKUM

Hasil Praktikum

Waktu (detik)	Kondisi
1	 <p><i>Gambar 2 kondisi detik ke-1</i></p>
2	 <p><i>Gambar 3 kondisi detik ke-2</i></p>
3	 <p><i>Gambar 4 kondisi detik ke-3</i></p>
4	 <p><i>Gambar 5 kondisi detik ke-4</i></p>
5	 <p><i>Gambar 6 kondisi detik ke-5</i></p>
6	 <p><i>Gambar 7 kondisi detik ke-6</i></p>

7	 <p><i>Gambar 8 kondisi detik ke-7</i></p>
8	 <p><i>Gambar 9 kondisi detik ke-8</i></p>

Tabel 2 hasil percobaan

Pembahasan

Urutan lampu dari kanan ke kiri adalah: Utara, Timur, Selatan, Barat

Awalnya lampu hijau dari arah utara akan menyala, sedangkan yang lainnya merah. Kemudian setelah jeda waktu tertentu, lampu kuning timur akan menyala dan yang lainnya merah. Lalu lampu hijau timur akan menyala, sedangkan yang lainnya merah. Begitu terus hingga lampu utara menyala kuning.

Dalam percobaan ini jeda waktu adalah 1 detik, karena memudahkan untuk mengambil tangkapan layar. Namun pada dunia nyata, fase ketika lampu berwarna hijau diberi jeda waktu 90 detik dan fase ketika lampu berwarna kuning diberi waktu jeda 2 detik.

Kesimpulan

Arduino dapat digunakan untuk membuat lampu jalan berdasarkan waktu. Namun akan timbul masalah ketika jalanan sepi yang memungkinkan lampu berwarna merah padahal semua jalan kosong. Oleh karena itu, aplikasi ini masih berdasarkan waktu, belum bisa mendeteksi kosongnya jalan.