

LAPORAN PROJECT AKHIR

ALGORITMA & PEMROGRAMAN

(SISTEM TRAFFIC LIGHT 4 JALUR MENGGUNAKAN ARDUINO UNO LEONADO)

DOSEN PENGAMPU:

ANDI NUR FAIZAL, S.Pd., M.T



ANGGOTA KELOMPOK 2:

MUH. IKRAM (230204502022)

GANIYYIR RAHMAN ABDULLAH JR (230204500018)

MUHAMMAD AGUNG SAHPUTRA (230204501016)

DEPARTEMEN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI

UNIVERSITAS NEGERI MAKASSAR

JURUSAN PENDIDIKAN TEKNIK ELEKTRO

PROGRAM STUDI PENDIDIKAN TEKNIK ELEKTRO S1

MAKASSAR

2024

1. JUDUL

SISTEM TRAFFIC LIGHT 4 JALUR MENGGUNAKAN ARDUINO UNO LEONARDO

2. TUJUAN

1. Simulasi Pengaturan Lalu Lintas
2. Pemrograman Mikrokontroler
3. Pengelolaan Waktu dan Logika Kontrol
4. Penerapan Prinsip IoT (Opsional)
5. Efisiensi Energi dan Kinerja
6. Pemahaman Komponen Elektronik
7. Pengembangan Solusi Smart City (Opsional)

3. TEORI DASAR PRAKTIKUM

Sistem traffic light atau lampu lalu lintas adalah perangkat elektronik yang digunakan untuk mengatur alur lalu lintas pada persimpangan jalan. Pada sistem 4 jalur, lampu lalu lintas harus mengelola pergerakan kendaraan dari empat arah berbeda dengan memperhatikan prinsip keselamatan dan efisiensi lalu lintas. Traffic light bekerja berdasarkan siklus waktu yang bergantian antara warna lampu:

1. Merah: Kendaraan harus berhenti.
2. Kuning: Memberikan peringatan untuk berhati-hati atau bersiap-siap berubah.
3. Hijau: Kendaraan boleh bergerak.

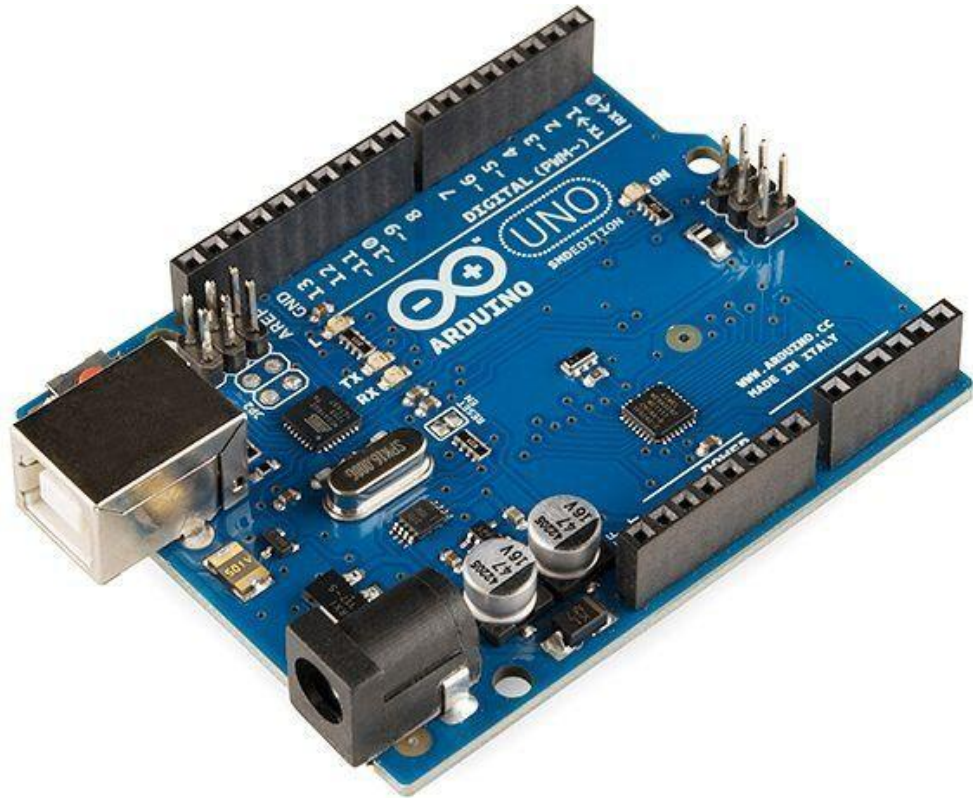
Komponen Utama:

1. Arduino

Arduino Uno dan Arduino Leonardo adalah dua jenis board dari platform Arduino, yang digunakan untuk pengembangan proyek elektronik dan pemrograman. Keduanya memiliki keunggulan dan fitur yang berbeda meskipun tujuan utamanya sama, yaitu memudahkan pembuatan prototipe perangkat keras dan perangkat lunak. Dimana dua papan pengembangan dari Arduino yang menggunakan mikrokontroler AVR.

Arduino Leonardo memiliki lebih banyak pin analog, di mana A0-A5 didedikasikan untuk pin analog dan A6-A11 pada pin digital. Dalam hal pin input

dan output: Arduino Leonardo memungkinkan kita untuk menggunakan 20 pin dibandingkan dengan 14 pin di UNO, karena dapat menggunakan pin analog sebagai I/O.



a. Arduino Uno

Arduino Uno adalah salah satu board yang paling populer di kalangan penggemar elektronika dan pengembang. Board ini berbasis mikrokontroler ATmega328P. Uno sering digunakan untuk prototipe dan proyek sederhana hingga menengah, berkat kemudahan penggunaan dan komunitas yang besar.

b. Arduino Leonardo

Arduino Leonardo, di sisi lain, berbasis mikrokontroler ATmega32U4. Salah satu fitur unik dari Leonardo adalah kemampuannya untuk bertindak sebagai perangkat USB, seperti mouse atau keyboard, yang tidak dimiliki oleh Uno. Hal ini membuat Leonardo lebih fleksibel untuk proyek yang memerlukan interaksi langsung dengan komputer.

Prinsip Kerja

Kedua board bekerja pada prinsip yang sama, yaitu menggunakan mikrokontroler untuk mengontrol input dan output. Berikut adalah cara kerja umumnya:

a) Pendaftaran Input

Board menerima sinyal dari berbagai sensor (misalnya, tombol, sensor gerak, sensor suhu) melalui pin input.

b) Proses Program

Kode (sketch) yang ditulis di lingkungan pengembangan Arduino (IDE) diterima dan dieksekusi oleh mikrokontroler. Kode ini menginstruksikan board tentang bagaimana merespons berbagai input.

c) Output

Setelah pemrosesan, board dapat mengeluarkan sinyal ke perangkat lain (seperti LED, motor, atau perangkat lain) melalui pin output.

Perbedaan Arduino Uno dan Leonardo

Fitur	Arduino Uno	Arduino Leonardo
Mikrokontroler	ATmega328P	ATmega32U4
Digital I/O	14	20
Analog Input	6	12
USB-to-Serial	ATmega16U2	Tidak diperlukan
Fitur USB bawaan	Tidak	Ya (keyboard/mouse)
Tipe USB	USB Type-B	Micro-USB

Aplikasi

1. Robotika dan otomasi.
2. Sistem keamanan dan pengawasan.
3. Proyek IoT (smart home, sensor lingkungan).
4. Pengembangan perangkat wearable.
5. Sistem kontrol industri.

2. LED (Light Emitting Diode)

LED (Light Emitting Diode) adalah perangkat semikonduktor yang memancarkan cahaya ketika arus listrik mengalir melaluinya. Dioda ini terbuat dari lapisan tipis bahan semikonduktor, dan saat bias maju diterapkan, elektron dalam semikonduktor bergabung dengan lubang, menghasilkan cahaya.

- 4 LED Merah: Digunakan untuk menunjukkan lampu merah pada setiap jalur.
- 4 LED Kuning: Menunjukkan lampu kuning sebagai tanda peringatan.
- 4 LED Hijau: Menunjukkan lampu hijau yang menandakan bahwa jalur tersebut aman untuk dilalui.



3. Resistor

12 Resistor 330 Ohm digunakan untuk membatasi arus yang mengalir ke LED, mencegah kerusakan pada LED akibat arus berlebih.



4. Power Supply

Sumber daya yang diperlukan untuk memberikan energi pada Arduino dan LED, biasanya menggunakan adaptor atau baterai 9V DC

5. Breadboard dan Kabel Jumper

Digunakan untuk merangkai komponen elektronik tanpa perlu menyolder, sehingga memudahkan dalam pengujian dan modifikasi rangkaian

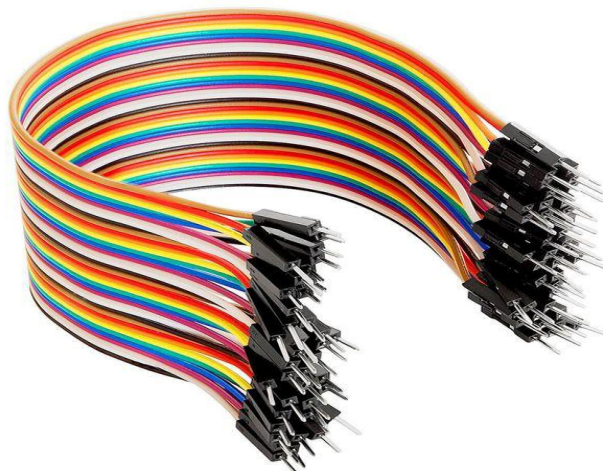


Diagram Rangkaian

Rangkaian sistem traffic light dapat dirancang dengan menghubungkan setiap LED ke pin digital pada Arduino. Setiap LED memiliki anoda yang terhubung ke pin digital dan katoda yang terhubung ke ground melalui resistor. Contoh pengkabelan adalah sebagai berikut:

- ❖ Pin digital 2: LED Merah Jalur 1
- ❖ Pin digital 3: LED Kuning Jalur 1
- ❖ Pin digital 4: LED Hijau Jalur 1

Sistem traffic light 4 jalur menggunakan Arduino untuk mengontrol urutan dan durasi menyala setiap lampu LED. Berikut adalah langkah-langkah kerja sistem:

1. Inisialisasi: Arduino mengatur pin input dan output serta variabel yang diperlukan.
2. Logika Siklus: Arduino menjalankan logika kontrol sesuai urutan:
 - Jalur 1 lampu hijau menyala, jalur lainnya lampu merah.
 - Setelah durasi tertentu, lampu hijau berubah menjadi kuning, kemudian merah.
 - Siklus berlanjut ke jalur berikutnya secara bergantian.
3. Pengaturan Waktu: Penggunaan fungsi waktu (seperti `delay()` atau timer interrupt) untuk mengatur durasi setiap lampu.
4. Keamanan: Pastikan ada jeda antarperubahan (lampu kuning) untuk menghindari konflik antarjalur.

Manfaat Arduino dalam Sistem Traffic Light

1. Fleksibilitas: Pengguna dapat memodifikasi durasi lampu dengan mudah menggunakan kode pemrograman.
2. Efisiensi Biaya: Arduino adalah mikrokontroler yang ekonomis namun cukup andal.
3. Pengembangan Lebih Lanjut: Sistem dapat dikembangkan lebih jauh, seperti integrasi dengan sensor kendaraan untuk menyesuaikan durasi lampu hijau secara dinamis.

Prinsip Dasar Pemrograman Arduino

- Input dan Output Digital: Lampu LED dihubungkan ke pin output digital Arduino.
- Logika Pemrograman: Menggunakan struktur kontrol seperti *if-else* dan *loop* untuk mengatur urutan lampu.

- Manajemen Waktu: Fungsi seperti `(millis)` atau `(delay)` digunakan untuk pengaturan durasi.

Skema Sederhana Sistem:

1. Jalur 1: Hijau → Kuning → Merah.
2. Jalur 2: Hijau → Kuning → Merah.
3. Jalur 3: Hijau → Kuning → Merah.
4. Jalur 4: Hijau → Kuning → Merah.

Setiap jalur mendapatkan giliran menyala lampu hijau berdasarkan urutan yang telah ditentukan, dan siklus berulang terus menerus. Pengembangan Lebih Lanjut

- Integrasi Sensor: Menggunakan sensor ultrasonik atau inframerah untuk mendeteksi kendaraan.
- Koneksi IoT: Menghubungkan sistem ke jaringan untuk pengelolaan jarak jauh.
- Backup Daya: Menggunakan baterai cadangan untuk memastikan sistem tetap bekerja saat listrik padam.

Dengan dasar ini, sistem traffic light 4 jalur berbasis Arduino dapat diimplementasikan dengan efektif untuk membantu pengaturan lalu lintas.

Percobaan	Kendaraan Jalur 1+3	Kendaraan Jalur 2+4	Lampu Hijau
1	16	20	Jalur 2+4
2	18	15	Jalur 1+3
3	10	25	Jalur 2+4
4	22	19	Jalur 1+3
5	14	14	Normal
6	17	22	Jalur 2+4
7	12	30	Jalur 2+4
8	20	19	Jalur 1+3
9	15	16	Jalur 1+3
10	13	18	Jalur 2+4

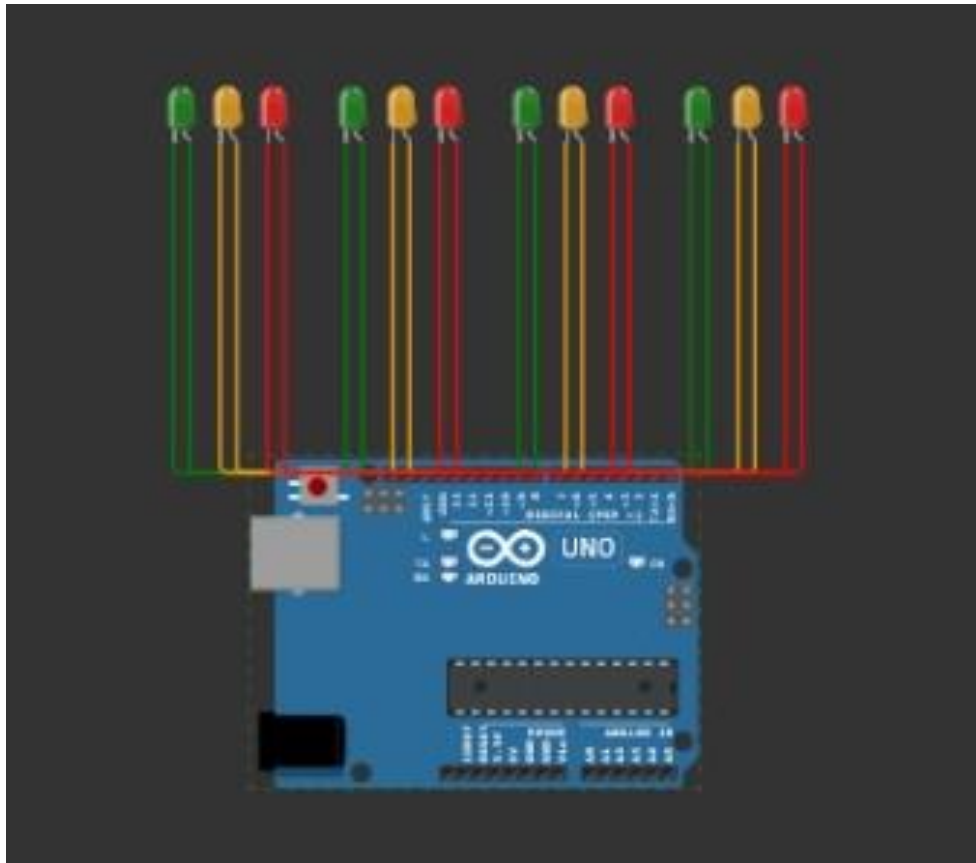
(Tabel Percobaan)

4. ALAT DAN BAHAN

NO	NAMA ALAT DAN BAHAN	JUMLAH
1.	Arduino Uno	1 Buah
2.	Traffic light	4 Buah
3.	Kabel jumper	20 Buah
4.	Kabel USB	1 Buah
5.	Resistor SMD	12 Buah
6.	Power Supply	1 Buah
7.	Kertas karton	1 Buah

5. LANGKAH KERJA HASIL PRAKTIKUM :

❖ R a n g k a i a n



❖ Program

```
// Deklarasi pin untuk setiap lampu di jalur 1, 2, 3, dan 4
int red1 = 13, yellow1 = 12, green1 = 11; // Jalur 1
int red2 = 10, yellow2 = 9, green2 = 8; // Jalur 2
int red3 = 7, yellow3 = 6, green3 = 5; // Jalur 3
int red4 = 4, yellow4 = 3, green4 = 2; // Jalur 4

// Variabel jumlah kendaraan
int vehicles1and3 = 0; // Kendaraan di jalur 1 dan 3
int vehicles2and4 = 0; // Kendaraan di jalur 2 dan 4

// Variabel simulasi
const int testCount = 10; // Jumlah percobaan
int testResults[testCount][3]; // Array untuk menyimpan hasil
[Percobaan][Kendaraan1+3, Kendaraan2+4, Output]

void setup() {
  // Serial Monitor
  Serial.begin(9600);

  // Atur semua pin sebagai output
  pinMode(red1, OUTPUT);
  pinMode(yellow1, OUTPUT);
  pinMode(green1, OUTPUT);

  pinMode(red2, OUTPUT);
  pinMode(yellow2, OUTPUT);
  pinMode(green2, OUTPUT);

  pinMode(red3, OUTPUT);
  pinMode(yellow3, OUTPUT);
  pinMode(green3, OUTPUT);
```

```

pinMode(red4, OUTPUT);
pinMode(yellow4, OUTPUT);
pinMode(green4, OUTPUT);

// Jalankan simulasi
simulateTraffic();
}

void loop() {
    // Setelah simulasi selesai, lampu kembali berjalan normal
    normalTrafficCycle();
}

// Fungsi simulasi lalu lintas
void simulateTraffic() {
    for (int i = 0; i < testCount; i++) {
        // Simulasi jumlah kendaraan acak
        vehicles1and3 = random(10, 30); // Kendaraan di jalur 1 dan 3
        vehicles2and4 = random(10, 30); // Kendaraan di jalur 2 dan 4

        // Tentukan jalur mana yang mendapatkan lampu hijau
        int greenOutput;
        if (abs(vehicles1and3 - vehicles2and4) <= 2) {
            // Jika jumlah kendaraan hampir sama, jalankan logika normal
            normalTrafficCycleOnce();
            greenOutput = 0; // Normal
        } else if (vehicles1and3 > vehicles2and4) {
            greenOutput = 1; // Jalur 1+3 mendapatkan lampu hijau
            activateTraffic(red2, yellow2, green2, red4, yellow4, green4, red1, yellow1,
green1, red3, yellow3, green3);
        } else {
            greenOutput = 2; // Jalur 2+4 mendapatkan lampu hijau
            activateTraffic(red1, yellow1, green1, red3, yellow3, green3, red2, yellow2,
green2, red4, yellow4, green4);

```

```
}
```

```
// Simpan hasil percobaan
```

```
testResults[i][0] = vehicles1and3;
```

```
testResults[i][1] = vehicles2and4;
```

```
testResults[i][2] = greenOutput;
```

```
}
```

```
// Tampilkan hasil simulasi
```

```
printResults();
```

```
}
```

```
// Fungsi untuk mengatur lampu lalu lintas
```

```
void activateTraffic(int redOpp1, int yellowOpp1, int greenOpp1,
```

```
int redOpp2, int yellowOpp2, int greenOpp2,
```

```
int redActive1, int yellowActive1, int greenActive1,
```

```
int redActive2, int yellowActive2, int greenActive2) {
```

```
// Jalur aktif (lampu hijau menyala)
```

```
digitalWrite(redActive1, LOW); // Mematikan lampu merah di jalur aktif
```

```
digitalWrite(redActive2, LOW); // Mematikan lampu merah di jalur aktif
```

```
digitalWrite(greenActive1, HIGH); // Menyalakan lampu hijau di jalur aktif
```

```
digitalWrite(greenActive2, HIGH); // Menyalakan lampu hijau di jalur aktif
```

```
// Jalur lawan (lampu merah menyala)
```

```
digitalWrite(redOpp1, HIGH); // Menyalakan lampu merah di jalur lawan
```

```
digitalWrite(redOpp2, HIGH); // Menyalakan lampu merah di jalur lawan
```

```
digitalWrite(greenOpp1, LOW); // Mematikan lampu hijau di jalur lawan
```

```
digitalWrite(greenOpp2, LOW); // Mematikan lampu hijau di jalur lawan
```

```
delay(5000); // Durasi lampu hijau (5 detik)
```

```
// Jalur aktif (lampu kuning menyala)
```

```
digitalWrite(greenActive1, LOW); // Mematikan lampu hijau
```

```

digitalWrite(greenActive2, LOW); // Mematikan lampu hijau
digitalWrite(yellowActive1, HIGH); // Menyalakan lampu kuning
digitalWrite(yellowActive2, HIGH); // Menyalakan lampu kuning

delay(3000); // Durasi lampu kuning (3 detik)

// Jalur aktif (lampu merah menyala)
digitalWrite(yellowActive1, LOW); // Mematikan lampu kuning
digitalWrite(yellowActive2, LOW); // Mematikan lampu kuning
digitalWrite(redActive1, HIGH); // Menyalakan lampu merah
digitalWrite(redActive2, HIGH); // Menyalakan lampu merah
}

// Fungsi untuk satu siklus normal
void normalTrafficCycleOnce() {
    // Jalur 1 dan 3 menyala hijau, jalur 2 dan 4 merah
    activateTraffic(red2, yellow2, green2, red4, yellow4, green4, red1, yellow1, green1,
red3, yellow3, green3);

    // Jalur 2 dan 4 menyala hijau, jalur 1 dan 3 merah
    activateTraffic(red1, yellow1, green1, red3, yellow3, green3, red2, yellow2, green2,
red4, yellow4, green4);
}

// Fungsi untuk siklus normal tanpa batas
void normalTrafficCycle() {
    while (true) {
        normalTrafficCycleOnce();
    }
}

void printResults() {
    Serial.println("Hasil Percobaan:");
    Serial.println("Percobaan\tJalur 1+3\tJalur 2+4\tLampu Hijau di Jalur");
}

```

```

for (int i = 0; i < testCount; i++) {
    Serial.print(i + 1);
    Serial.print("\t\t");
    Serial.print(testResults[i][0]); // Kendaraan Jalur 1+3
    Serial.print("\t\t");
    Serial.print(testResults[i][1]); // Kendaraan Jalur 2+4
    Serial.print("\t\t");
    if (testResults[i][2] == 0) {
        Serial.println("Normal"); // Siklus normal
    } else {
        Serial.println(testResults[i][2]); // Jalur yang mendapatkan lampu hijau
    }
}
}

```

❖ Hasil Output

Simulasi Siklus Lampu Lalu Lintas :

1. Jalur 1: Hijau → Kuning → Merah

- Lampu hijau di jalur 1 menyala selama 15 detik (kendaraan bergerak).
- Lampu kuning menyala selama 3 detik (peringatan transisi).
- Lampu merah menyala (giliran jalur lain).

2. Jalur 2: Hijau → Kuning → Merah

Setelah jalur 1, giliran jalur 2 menyala dengan siklus yang sama.

3. Jalur 3 dan Jalur 4:

Jalur 3 dan 4 mendapatkan giliran masing-masing secara bergantian, mengikuti siklus yang sama.

4. Loop Berulang:

Siklus diatur terus menerus, memastikan alur lalu lintas teratur.

Representasi Lampu LED

- Hijau : Menyala terang di jalur yang aktif (kendaraan bergerak).
- Kuning : Menyala terang sesaat sebelum perubahan ke merah (peringatan).
- Merah : Menyala terang di jalur non-aktif (kendaraan berhenti).

6. KESIMPULAN

Setelah berhasil menyelesaikan proyek sistem traffic light 4 jalur menggunakan Arduino, dapat disimpulkan bahwa proyek ini merupakan solusi yang efisien dan praktis dalam mengelola lalu lintas pada persimpangan jalan. Dengan menggunakan Arduino sebagai mikrokontroler utama, sistem ini mampu mengatur siklus lampu lalu lintas secara stabil dan konsisten sesuai logika yang telah diprogram. Durasi lampu hijau, kuning, dan merah dapat dengan mudah disesuaikan sesuai kebutuhan, sehingga meningkatkan fleksibilitas dalam pengelolaan lalu lintas.

Sistem ini terbukti mampu meningkatkan keselamatan lalu lintas dengan mengatur jeda waktu melalui lampu kuning, yang memberikan peringatan bagi pengemudi sebelum perubahan lampu. Hal ini memastikan kendaraan dari jalur yang berbeda dapat bergerak secara bergantian tanpa risiko tabrakan. Proyek ini juga menawarkan solusi yang terjangkau, mengingat penggunaan komponen sederhana seperti LED, resistor, dan power supply, menjadikannya alternatif ekonomis dibandingkan sistem lampu lalu lintas komersial yang lebih mahal.

Selain itu, proyek ini memberikan wawasan penting tentang pemrograman Arduino dan teknik dasar elektronik, termasuk pengaturan LED, manajemen daya, dan logika kontrol. Keberhasilan proyek ini membuka peluang untuk pengembangan lebih lanjut, seperti integrasi sensor untuk mendeteksi kepadatan lalu lintas, penghubungan dengan jaringan IoT untuk pengelolaan jarak jauh, dan implementasi cadangan daya agar sistem tetap berfungsi saat terjadi pemadaman listrik.

Secara keseluruhan, sistem traffic light 4 jalur berbasis Arduino adalah solusi yang efektif untuk pengaturan lalu lintas. Dengan pengembangan lebih lanjut, sistem ini dapat menjadi lebih cerdas dan responsif terhadap kebutuhan lalu lintas yang dinamis. Proyek ini juga menunjukkan bagaimana teknologi sederhana dapat memberikan dampak positif dalam kehidupan sehari-hari.

7. DOKUMENTASI



8. REFERENSI

Banzi, M., & Shiloh, M. (2015). *Getting Started with Arduino (3rd Edition)*. O'Reilly Media.

Monk, S. (2020). *Programming Arduino: Getting Started with Sketches (3rd Edition)*. McGraw-Hill Education.

Fried, L. (2012). *Arduino Cookbook: Recipes to Begin, Expand, and Enhance Your Projects*. O'Reilly Media.

Tanenbaum, A. S. (2019). *Modern Operating Systems (4th Edition)*. Pearson Education.