

# LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

## Praktik Pembuatan Sistem Lampu Lalu Lintas Menggunakan Simulator Wokwi ESP 32

*M Bimo Amarulloh*

Fakultas Vokasi, Universitas Brawijaya

Email: bimoamar@gmail.com

### Abstrak

Praktik ini bertujuan untuk membuat sistem lampu lalu lintas menggunakan simulator Wokwi dengan mikrokontroler ESP32 dan code editor Visual Studio Code. Proyek ini dirancang tanpa menggunakan resistor, dengan mengandalkan kontrol langsung dari ESP32 untuk mengatur nyala dan pergantian warna lampu lalu lintas. Implementasi program dilakukan menggunakan bahasa pemrograman Arduino (C++) dengan logika pergantian lampu berbasis waktu (delay) dan kondisi tertentu. Hasil dari praktik ini menunjukkan bahwa sistem lampu lalu lintas dapat berfungsi dengan baik, dengan pergantian warna lampu yang sesuai dengan siklus yang telah diprogram. Lampu merah, kuning, dan hijau menyala bergantian dalam durasi yang telah ditentukan tanpa mengalami kendala. Penggunaan simulator Wokwi mempermudah pengujian tanpa memerlukan perangkat keras fisik, serta memungkinkan debugging yang lebih efisien. Keberhasilan proyek ini menunjukkan bahwa ESP32 dapat mengontrol sistem lampu lalu lintas secara langsung tanpa resistor, meskipun pada implementasi nyata penggunaan resistor disarankan untuk melindungi komponen elektronik. Dengan adanya simulasi ini, pembelajaran tentang sistem kendali berbasis mikrokontroler dapat dilakukan dengan mudah dan efisien sebelum diterapkan pada perangkat fisik.

**Kata Kunci:** ESP32, Wokwi, Lampu Lalu Lintas, Simulasi, Visual Studio Code

### Abstract

This practice aims to develop a traffic light system using the Wokwi simulator with the ESP32 microcontroller and Visual Studio Code as the code editor. The project is designed without using resistors, relying on direct control from the ESP32 to manage the illumination and color transitions of the traffic lights. The program is implemented using the Arduino programming language (C++) with a time-based (delay) and conditional logic approach for light transitions. The results of this practice demonstrate that the traffic light system functions properly, with color transitions occurring according to the programmed cycle. The red, yellow, and green lights switch sequentially within the specified duration without any issues. The use of the Wokwi simulator facilitates testing without requiring physical hardware and allows for more efficient debugging. The success of this project indicates that the ESP32 can directly control the traffic light system without resistors, although resistor usage is recommended in real-world implementations to protect electronic components. This simulation provides an effective and efficient learning experience for understanding microcontroller-based control systems before applying them to physical devices.

**Keywords:** ESP32, Wokwi, Traffic Light, Simulation, Visual Studio Code

## 1. Introduction (Pendahuluan)

### 1.1 Latar belakang

Lampu lalu lintas merupakan sistem kendali yang penting dalam mengatur arus kendaraan di persimpangan jalan guna mencegah kemacetan dan kecelakaan. Dengan perkembangan teknologi, simulasi sistem lampu lalu lintas menggunakan mikrokontroler semakin banyak digunakan sebagai metode pembelajaran dan pengujian sebelum implementasi nyata. Salah satu perangkat yang dapat digunakan untuk tujuan ini adalah ESP32, sebuah mikrokontroler yang memiliki fitur konektivitas dan pemrosesan yang mumpuni. Pada praktik ini, simulator Wokwi digunakan untuk mensimulasikan sistem lampu lalu lintas berbasis ESP32. Wokwi memungkinkan pengguna untuk menguji kode dan desain rangkaian elektronik tanpa perlu menggunakan perangkat keras fisik. Selain itu, Visual Studio Code digunakan sebagai code editor untuk menulis dan mengembangkan program dengan bahasa pemrograman Arduino (C++). Keunikan dari praktik ini adalah penerapan sistem lampu lalu lintas tanpa menggunakan resistor, yang biasanya digunakan untuk membatasi arus pada komponen elektronik. Eksperimen ini bertujuan untuk menguji apakah ESP32 mampu mengontrol langsung lampu lalu lintas tanpa perlindungan tambahan. Hasil yang diperoleh akan memberikan wawasan tentang efektivitas simulasi sistem lampu lalu lintas menggunakan Wokwi serta potensi penerapannya dalam dunia nyata. Praktikum ini diharapkan dapat memberikan manfaat bagi mahasiswa dan praktisi dalam memahami prinsip kerja lampu lalu lintas berbasis mikrokontroler, serta memberikan alternatif simulasi yang efisien sebelum penerapan fisik.

### 1.2 Tujuan eksperimen

1. **Memahami Konsep Sistem Lampu Lalu Lintas**  
Mempelajari cara kerja sistem lampu lalu lintas serta prinsip pengaturan waktu pergantian lampu merah, kuning, dan hijau menggunakan mikrokontroler ESP32.
2. **Mengimplementasikan Simulasi Lampu Lalu Lintas**  
Menggunakan simulator Wokwi untuk mengembangkan dan menguji sistem lampu lalu lintas berbasis ESP32 tanpa perlu menggunakan perangkat keras fisik.
3. **Mengembangkan Program Kontrol Lampu Lalu Lintas**  
Menulis dan mengimplementasikan kode program dalam bahasa pemrograman Arduino (C++) menggunakan Visual Studio Code sebagai code editor.
4. **Mengujicoba Sistem Tanpa Resistor**  
Mengamati efektivitas dan kinerja ESP32 dalam mengontrol langsung lampu lalu lintas tanpa penggunaan resistor, serta memahami dampaknya terhadap komponen yang digunakan.
5. **Meningkatkan Pemahaman Tentang Mikrokontroler**  
Memberikan wawasan lebih dalam mengenai penggunaan mikrokontroler ESP32 dalam sistem kendali otomatis dan pentingnya simulasi dalam proses pengembangan perangkat berbasis elektronik.

## 2. Methodology (Metodologi)

### 2.1 Tools & Materials (Alat dan Bahan)

#### Alat

1. **Komputer/Laptop** – Digunakan untuk menulis, mengedit, dan menjalankan program simulasi.
2. **Visual Studio Code (VS Code)** – Code editor yang digunakan untuk menulis dan mengembangkan program berbasis Arduino (C++).
3. **Wokwi Simulator** – Simulator berbasis web yang digunakan untuk mensimulasikan sistem lampu lalu lintas tanpa perlu menggunakan perangkat keras fisik.

#### Bahan (Komponen Virtual di Wokwi)

1. **ESP32** – Mikrokontroler yang digunakan sebagai pusat kendali untuk sistem lampu lalu lintas.

2. **LED (Merah, Kuning, Hijau)** – Digunakan untuk mensimulasikan lampu lalu lintas dengan pergantian warna sesuai dengan program yang dibuat.
3. **Breadboard (Virtual)** – Digunakan sebagai tempat untuk merangkai komponen elektronik dalam simulasi.
4. **Jumper Wire (Virtual)** – Digunakan untuk menghubungkan ESP32 dengan LED dalam rangkaian simulasi.

Dalam praktik ini, resistor tidak digunakan untuk membatasi arus pada LED. Hal ini dilakukan untuk menguji bagaimana ESP32 dapat mengontrol langsung sistem lampu lalu lintas tanpa resistor serta memahami dampaknya terhadap rangkaian elektronik dalam simulasi.

## 2.2 Implementation Steps (Langkah Implementasi)

Berikut adalah langkah-langkah implementasi dalam pembuatan sistem lampu lalu lintas menggunakan simulator Wokwi dan mikrokontroler ESP32:

### 1. Persiapan Alat dan Bahan

- Pastikan komputer/laptop sudah terinstall Visual Studio Code (VS Code).
- Akses Wokwi Simulator melalui browser.
- Siapkan komponen virtual di Wokwi, yaitu ESP32, LED (merah, kuning, hijau), dan jumper wire.

### 2. Membuat Rangkaian di Wokwi

- Buka Wokwi Simulator dan pilih proyek baru dengan ESP32.
- Tambahkan tiga LED (merah, kuning, hijau) sebagai lampu lalu lintas.
- Hubungkan masing-masing LED ke pin digital ESP32 tanpa menggunakan resistor.
- Pastikan satu sisi LED terhubung ke GND (ground) dan sisi lainnya ke pin GPIO ESP32.

### 3. Menulis Kode Program di Visual Studio Code

- Buka Visual Studio Code dan buat file baru dengan ekstensi .ino atau langsung edit di Wokwi.
- Tulis kode untuk mengontrol LED menggunakan logika pergantian lampu lalu lintas.

### 4. Kode Program Arduino (C++)

Berikut adalah contoh kode untuk mengatur nyala lampu lalu lintas:

```
#include <Arduino.h>
```

```
int ledm = 27; // Lampu Merah di GPIO 27
```

```
int ledk = 25; // Lampu Kuning di GPIO 25
```

```
int ledh = 32; // Lampu Hijau di GPIO 32
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600);
```

```
  pinMode(ledm, OUTPUT);
```

```
pinMode(ledk, OUTPUT);
pinMode(ledh, OUTPUT);
}

void loop()
{
  // Lampu Merah Menyala
  digitalWrite(ledm, HIGH);
  digitalWrite(ledk, LOW);
  digitalWrite(ledh, LOW);
  Serial.println("Merah Menyala");
  Serial.println("Kuning Mati");
  Serial.println("Hijau Mati");
  delay(3000);

  // Lampu Kuning Menyala
  digitalWrite(ledm, LOW);
  digitalWrite(ledk, HIGH);
  digitalWrite(ledh, LOW);
  Serial.println("Merah Mati");
  Serial.println("Kuning Menyala");
  Serial.println("Hijau Mati");
  delay(1000);

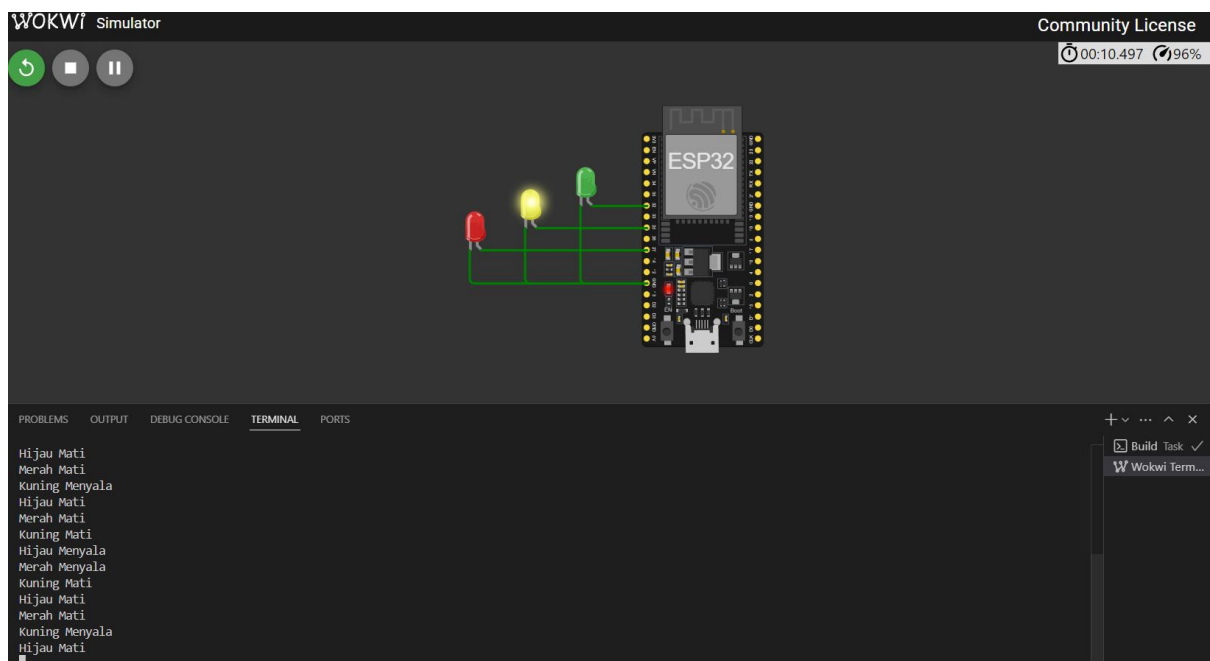
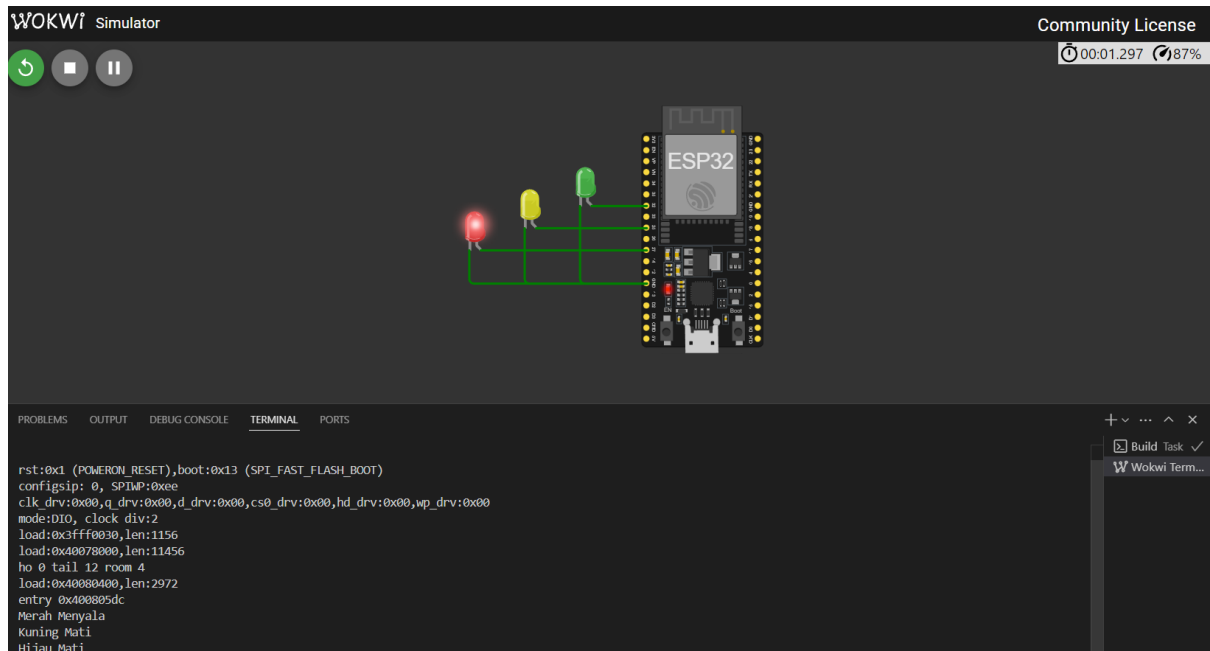
  // Lampu Hijau Menyala
  digitalWrite(ledm, LOW);
  digitalWrite(ledk, LOW);
  digitalWrite(ledh, HIGH);
  Serial.println("Merah Mati");
  Serial.println("Kuning Mati");
  Serial.println("Hijau Menyala");
  delay(2000);
}
```

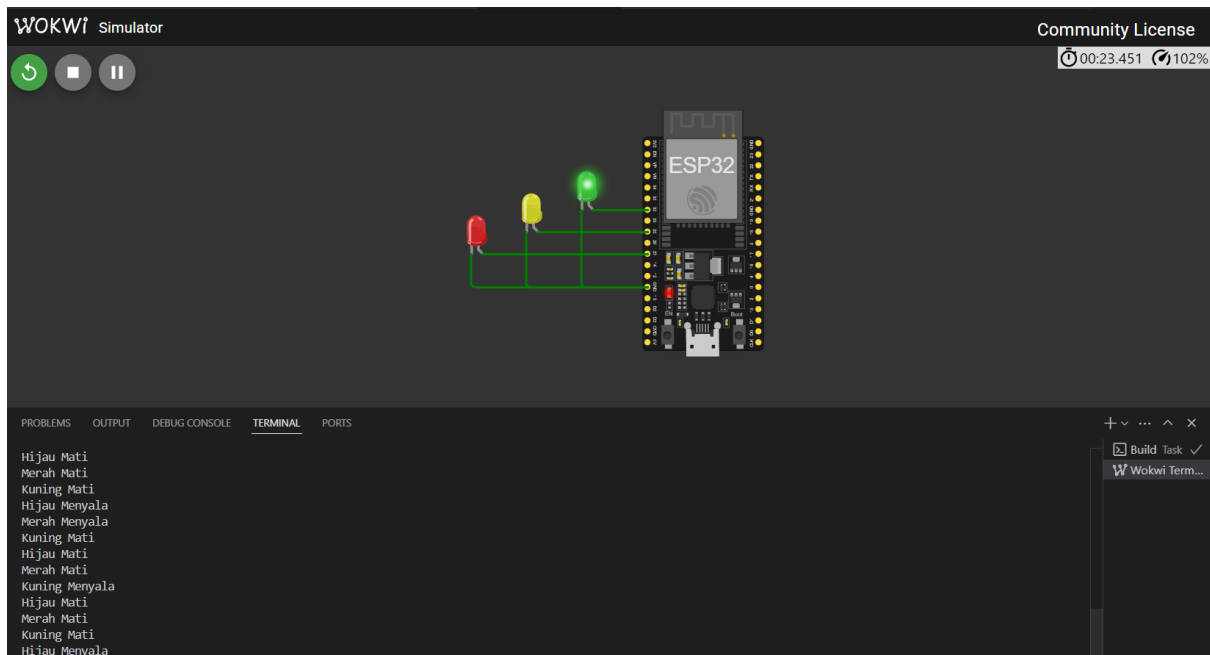
## 5. Mengunggah dan Menjalankan Kode di Wokwi

- Copy kode program ke editor kode di Visual Studio Code
- Klik "Start Simulation" untuk menjalankan simulasi.
- Amati apakah lampu lalu lintas berubah sesuai urutan dan waktu yang telah ditentukan.

## 3. Results and Discussion (Hasil dan Pembahasan)

### 3.1 Experimental Results (Hasil Eksperimen)





Kode berjalan dengan baik sesuai dengan keinginan peneliti, ketika lampu kuning menyala maka lampu merah dan hijau akan mati. Ketika lampu hijau menyala maka lampu kuning dan merah akan mati. Ketika lampu merah menyala maka lampu hijau dan kuning akan mati. Delay lampu merah adalah 3000ms, lampu kuning 1000ms, lampu hijau 2000ms

Lampu berjalan dengan urutan Merah → Kuning → Hijau


#### 4. Appendix (Lampiran)

```

1  diagram.json > ...
2  {
3    "version": 1,
4    "author": "Anonymous maker",
5    "editor": "wokwi",
6    "parts": [
7      { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 0, "left": -4.76, "attrs": {} },
8      { "type": "wokwi-led", "id": "led1", "top": 82.8, "left": -169, "attrs": { "color": "red" } },
9      {
10        "type": "wokwi-led",
11        "id": "led2",
12        "top": 63.6,
13        "left": -121,
14        "attrs": { "color": "yellow" }
15      },
16      {
17        "type": "wokwi-led",
18        "id": "led3",
19        "top": 44.4,
20        "left": -73,
21        "attrs": { "color": "limegreen" }
22      }
23    ],
24    "connections": [
25      [ "esp:TX", "$serialMonitor:RX", "", [] ],
26      [ "esp:RX", "$serialMonitor:TX", "", [] ],
27      [ "led3:A", "esp:32", "green", [ "v0" ] ],
28      [ "led3:C", "esp:GND.1", "green", [ "v0" ] ],
29      [ "led2:A", "esp:25", "green", [ "v0" ] ],
30      [ "led2:C", "esp:GND.1", "green", [ "v0" ] ],
31      [ "led1:A", "esp:27", "green", [ "v0" ] ],
32      [ "led1:C", "esp:GND.1", "green", [ "v0" ] ]
33    ],
34    "dependencies": {}
35  }

```

Gambar 1. 1 Syntax Diagram ESP 32



```
1 #include <Arduino.h>
2
3 int ledm = 27; // Lampu Merah di GPIO 27
4 int ledk = 25; // Lampu Kuning di GPIO 25
5 int ledh = 32; // Lampu Hijau di GPIO 32
6
7 void setup()
8 {
9   Serial.begin(9600);
10  pinMode(ledm, OUTPUT);
11  pinMode(ledk, OUTPUT);
12  pinMode(ledh, OUTPUT);
13 }
14
15 void loop()
16 {
17   // Lampu Merah Menyala
18   digitalWrite(ledm, HIGH);
19   digitalWrite(ledk, LOW);
20   digitalWrite(ledh, LOW);
21   Serial.println("Merah Menyala");
22   Serial.println("Kuning Mati");
23   Serial.println("Hijau Mati");
24   delay(3000);
25
26   // Lampu Kuning Menyala
27   digitalWrite(ledm, LOW);
28   digitalWrite(ledk, HIGH);
29   digitalWrite(ledh, LOW);
30   Serial.println("Merah Mati");
31   Serial.println("Kuning Menyala");
32   Serial.println("Hijau Mati");
33   delay(1000);
34
35   // Lampu Hijau Menyala
36   digitalWrite(ledm, LOW);
37   digitalWrite(ledk, LOW);
38   digitalWrite(ledh, HIGH);
39   Serial.println("Merah Mati");
40   Serial.println("Kuning Mati");
41   Serial.println("Hijau Menyala");
42   delay(2000);
43 }
44
```

Gambar 1. 2 Syntax Traffic Light