

PRAKTIKUM SIMULASI SISTEM LAMPU LALU LINTAS BERBASIS ESP32 DI WOKWI DENGAN PLATFORMIO DI VISUAL STUDIO CODE

TB Rangga Gilang Yanuari
Fakultas Vokasi, Universitas Brawijaya
gilangyanuarirangga@gmail.com

ABSTRAK

Sistem otomatisasi berbasis Internet of Things (IoT) semakin banyak diterapkan, termasuk dalam pengaturan lalu lintas. Praktikum ini mensimulasikan sistem lampu lalu lintas menggunakan mikrokontroler ESP32, platform Wokwi, serta pemrograman dengan Visual Studio Code dan ekstensi PlatformIO IDE. Sistem ini terdiri dari tiga LED berwarna merah, kuning, dan hijau yang menyala secara bergantian sesuai pola yang ditentukan. Hasil simulasi menunjukkan bahwa sistem bekerja sesuai rancangan, dengan urutan penyalan LED merah selama 20 detik, kuning selama 5 detik, dan hijau selama 30 detik sebelum siklus berulang. Penggunaan Wokwi memungkinkan pengujian tanpa perangkat keras fisik serta mempermudah visualisasi implementasi. Pengaturan waktu pada ESP32 memastikan sistem bekerja secara otomatis sesuai durasi yang ditentukan. Penerapan sistem ini menunjukkan bagaimana IoT digunakan dalam otomasi sederhana, khususnya pada simulasi lampu lalu lintas. Pengembangannya dapat diperluas dengan integrasi sensor atau kontrol berbasis jaringan untuk meningkatkan efisiensi dan fleksibilitas sistem.

Kata Kunci: IoT, ESP32, lampu lalu lintas, Wokwi, otomasi.

ABSTRACT

The use of Internet of Things (IoT)-based automation systems is increasingly applied, including in traffic management. This practicum simulates a traffic light system using an ESP32 microcontroller, the Wokwi platform, and programming with Visual Studio Code and the PlatformIO IDE. The system consists of three LEDs—red, yellow, and green—that light up alternately following a predetermined pattern. Simulation results indicate that the system operates as designed, with the LEDs lighting up in sequence: red for 20 seconds, yellow for 5 seconds, and green for 30 seconds before repeating the cycle. Wokwi enables testing without physical hardware and facilitates implementation visualization. The timing configuration on the ESP32 ensures the system operates automatically according to the specified duration. This system demonstrates how IoT can be utilized in simple automation, specifically in traffic light simulations. Future development may include integrating sensors or network-based control to enhance system efficiency and flexibility.

Keywords: IoT, ESP32, traffic light, Wokwi, automation.

1. Pendahuluan

1.1 Latar Belakang

Teknologi **Internet of Things (IoT)** telah berkembang pesat dan diterapkan dalam berbagai bidang, termasuk industri, kesehatan, pertanian, dan transportasi. Salah satu implementasi IoT yang semakin banyak digunakan adalah dalam sistem transportasi cerdas, yang memungkinkan pengelolaan lalu lintas menjadi lebih efisien dan terotomatisasi. Salah

satu contohnya adalah sistem **lampu lalu lintas otomatis**, yang dikendalikan oleh mikrokontroler dan diprogram untuk bekerja secara mandiri sesuai dengan skenario yang telah ditentukan.

Lampu lalu lintas memiliki peran yang sangat penting dalam pengaturan arus kendaraan, terutama di kawasan perkotaan yang memiliki tingkat mobilitas tinggi. Dengan pengaturan yang baik, lampu lalu lintas dapat membantu mengurangi kemacetan, meningkatkan keselamatan pengendara dan pejalan kaki, serta mengoptimalkan penggunaan jalan. Namun, dalam sistem konvensional, lampu lalu lintas masih dioperasikan dengan metode statis yang tidak menyesuaikan dengan kondisi lalu lintas secara real-time. Oleh karena itu, pemanfaatan teknologi IoT dalam sistem ini dapat meningkatkan efisiensi dengan memungkinkan otomatisasi dan pengendalian yang lebih cerdas.

Seiring dengan perkembangan teknologi simulasi, kini sistem lampu lalu lintas dapat diuji dan dianalisis tanpa perlu menggunakan perangkat keras fisik. Dengan memanfaatkan **platform Wokwi**, sistem dapat disimulasikan menggunakan **ESP32**, sebuah mikrokontroler yang banyak digunakan dalam pengembangan sistem berbasis IoT. Pemrograman sistem ini dilakukan melalui **Visual Studio Code**, yang didukung oleh ekstensi **PlatformIO IDE** untuk pengelolaan kode, kompilasi, serta debugging. Simulasi ini memungkinkan pengguna untuk memahami bagaimana sistem lampu lalu lintas bekerja tanpa perlu melakukan pengujian langsung pada perangkat keras.

Melalui praktikum ini, mahasiswa dapat mempelajari konsep dasar pemrograman mikrokontroler serta cara mengintegrasikan perangkat lunak dalam pengembangan sistem otomatisasi berbasis IoT. Dengan adanya simulasi, mahasiswa juga dapat melakukan eksperimen dan pengujian tanpa batasan perangkat keras, sehingga lebih fleksibel dalam mengevaluasi dan meningkatkan sistem yang dikembangkan. Selain itu, praktikum ini juga memberikan wawasan tentang bagaimana teknologi IoT dapat diterapkan dalam berbagai sistem otomatisasi lain, seperti smart city, rumah pintar, dan sistem transportasi berbasis kecerdasan buatan.

1.2 Tujuan Eksperimen

Eksperimen ini bertujuan untuk:

1. Memahami konsep dasar Internet of Things (IoT) dalam sistem otomatisasi.
2. Mensimulasikan sistem lampu lalu lintas menggunakan ESP32 pada platform Wokwi.
3. Mengimplementasikan pemrograman mikrokontroler dengan Visual Studio Code dan PlatformIO IDE.
4. Menganalisis kinerja simulasi serta mengevaluasi potensi pengembangan sistem.
- 5.

2. Methodology (Metodologi)

2.1 Tools & Materials (Alat dan Bahan)

Eksperimen ini dilakukan sepenuhnya melalui simulasi, sehingga tidak memerlukan perangkat keras fisik. Sistem dikembangkan menggunakan berbagai alat dan perangkat lunak yang mendukung pemrograman mikrokontroler dan simulasi sistem otomatisasi berbasis IoT. Berikut adalah alat dan bahan yang digunakan:

- Mikrokontroler (Simulasi ESP32)
ESP32 digunakan sebagai komponen utama dalam simulasi untuk mengontrol nyala-mati LED sesuai dengan pola waktu yang telah ditentukan. Mikrokontroler ini dipilih karena memiliki fitur yang mendukung pemrograman berbasis IoT dan kompatibel dengan berbagai platform simulasi, termasuk Wokwi.
- Simulator Wokwi
Wokwi merupakan platform berbasis web yang digunakan untuk mensimulasikan rangkaian elektronik tanpa

perlu menggunakan perangkat keras fisik. Wokwi menyediakan berbagai komponen virtual, seperti ESP32 dan LED, yang dapat dikonfigurasi sesuai dengan kebutuhan eksperimen. Simulator ini memungkinkan pengguna untuk menulis, mengunggah, dan menjalankan kode secara langsung serta memberikan visualisasi real-time dari rangkaian yang telah dibuat.

- **Visual Studio Code (VS Code)**
VS Code adalah editor kode yang digunakan untuk menulis, mengedit, dan mengelola program mikrokontroler. Dengan antarmuka yang fleksibel dan dukungan untuk berbagai ekstensi, VS Code memberikan pengalaman pengembangan yang lebih efisien dibandingkan dengan editor lainnya.
- **PlatformIO IDE**
PlatformIO IDE adalah ekstensi yang ditambahkan ke VS Code untuk mengembangkan proyek berbasis mikrokontroler. Platform ini menyediakan alat bantu untuk kompilasi kode, debugging, dan pengunggahan program ke dalam simulator Wokwi. Dengan menggunakan PlatformIO, pengembang dapat dengan mudah mengatur lingkungan pemrograman dan memastikan kode dapat berjalan dengan optimal pada ESP32.
- **Bahasa Pemrograman C++**
Bahasa pemrograman yang digunakan dalam eksperimen ini adalah C++, yang merupakan bahasa standar untuk pemrograman mikrokontroler. C++ digunakan untuk menulis logika pengendalian lampu lalu lintas, mengatur pin input/output, serta mengontrol durasi penyalan LED.

2.2 Implementation Steps (Langkah Implementasi)

Eksperimen ini terdiri dari beberapa tahapan yang dilakukan secara sistematis, mulai dari perancangan sistem hingga analisis hasil. Setiap langkah bertujuan untuk memastikan bahwa sistem yang dibuat dapat berfungsi sesuai dengan skenario yang telah ditetapkan. Berikut adalah langkah-langkah implementasi secara rinci:

1. Perancangan Sistem

Sebelum melakukan pemrograman, tahap awal yang dilakukan adalah merancang sistem lampu lalu lintas yang akan disimulasikan. Beberapa langkah yang dilakukan dalam tahap ini meliputi:

- **Menentukan konsep sistem lampu lalu lintas**
Sistem yang dirancang meniru cara kerja lampu lalu lintas di persimpangan jalan, di mana terdapat tiga lampu utama—merah, kuning, dan hijau—yang menyala secara bergantian dengan durasi tertentu.
- **Menentukan durasi penyalan lampu**
Berdasarkan aturan standar lalu lintas, lampu merah menyala selama 5 detik untuk memberi tanda berhenti, lampu kuning menyala selama 2 detik sebagai tanda peringatan, dan lampu hijau menyala selama 5 detik sebagai tanda kendaraan boleh melaju.
- **Memilih platform simulasi yang sesuai**
Untuk mensimulasikan sistem ini tanpa perangkat keras fisik, dipilih Wokwi, yang memiliki dukungan penuh untuk ESP32 dan memungkinkan pengguna untuk merancang serta menguji rangkaian elektronik secara virtual.
- **Menyusun diagram rangkaian secara virtual**
Dalam simulator Wokwi, ESP32 dihubungkan dengan tiga LED (merah, kuning, dan hijau) yang akan dikontrol berdasarkan logika pemrograman. Diagram ini bertujuan untuk memastikan bahwa setiap komponen terhubung dengan benar sebelum implementasi kode program.

2. Penulisan Kode Program

Setelah sistem dirancang, tahap berikutnya adalah menuliskan kode program dalam C++ untuk mengontrol nyala-mati LED sesuai dengan pola yang telah ditentukan. Langkah-langkahnya meliputi:

- Membuka Visual Studio Code dan menginstal PlatformIO IDE
Jika belum tersedia, ekstensi PlatformIO IDE diinstal ke dalam VS Code agar proyek berbasis ESP32 dapat dikembangkan dengan lebih mudah.
- Membuat proyek baru untuk ESP32
Proyek baru dibuat dengan memilih ESP32 sebagai board utama, sehingga seluruh konfigurasi sesuai dengan perangkat yang akan digunakan dalam simulasi.
- Menuliskan kode program dalam bahasa C++
Kode program yang ditulis mencakup:
 - Deklarasi pin untuk LED merah, kuning, dan hijau, yang masing-masing akan dikontrol oleh ESP32.
 - Konfigurasi pin sebagai OUTPUT dalam fungsi setup(), sehingga ESP32 dapat mengendalikan LED dengan benar.
 - Implementasi logika pengendalian lampu lalu lintas dalam fungsi loop(), di mana setiap LED menyala sesuai dengan durasi yang telah ditentukan menggunakan fungsi delay().
- Melakukan pengecekan kesalahan (debugging)
Sebelum menjalankan simulasi, dilakukan pengecekan kode untuk memastikan tidak ada kesalahan sintaks atau logika yang dapat menyebabkan sistem gagal beroperasi.

3. Simulasi dan Pengujian

Setelah kode program selesai ditulis, tahap berikutnya adalah menjalankan simulasi dan melakukan pengujian terhadap sistem yang telah dibuat. Langkah-langkahnya meliputi:

- Membuka platform Wokwi
Simulator Wokwi digunakan untuk mengunggah kode ESP32 dan menjalankan simulasi sistem lampu lalu lintas.
- Menjalankan simulasi dan mengamati hasilnya
Saat simulasi berjalan, diamati apakah LED menyala dalam urutan yang benar dengan durasi yang sesuai (merah 5 detik, kuning 2 detik, hijau 5 detik).
- Melakukan debugging jika ditemukan kesalahan
Jika terdapat kesalahan dalam logika atau durasi penyalan LED, dilakukan perbaikan kode dan pengujian ulang hingga sistem berfungsi dengan benar.
- Mengulangi simulasi untuk memastikan kestabilan sistem
Sistem diuji beberapa kali untuk memastikan bahwa siklus lampu lalu lintas berjalan secara konsisten tanpa mengalami kesalahan atau gangguan.

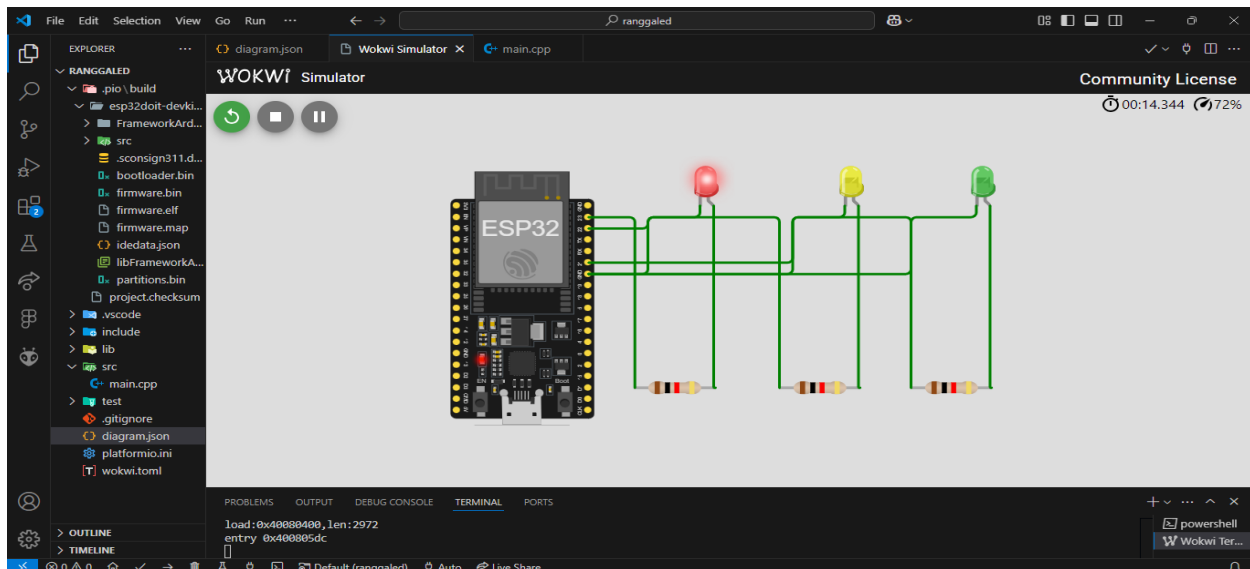
4. Analisis Hasil dan Evaluasi

Tahap terakhir adalah melakukan analisis terhadap hasil simulasi yang telah dijalankan. Beberapa aspek yang dievaluasi meliputi:

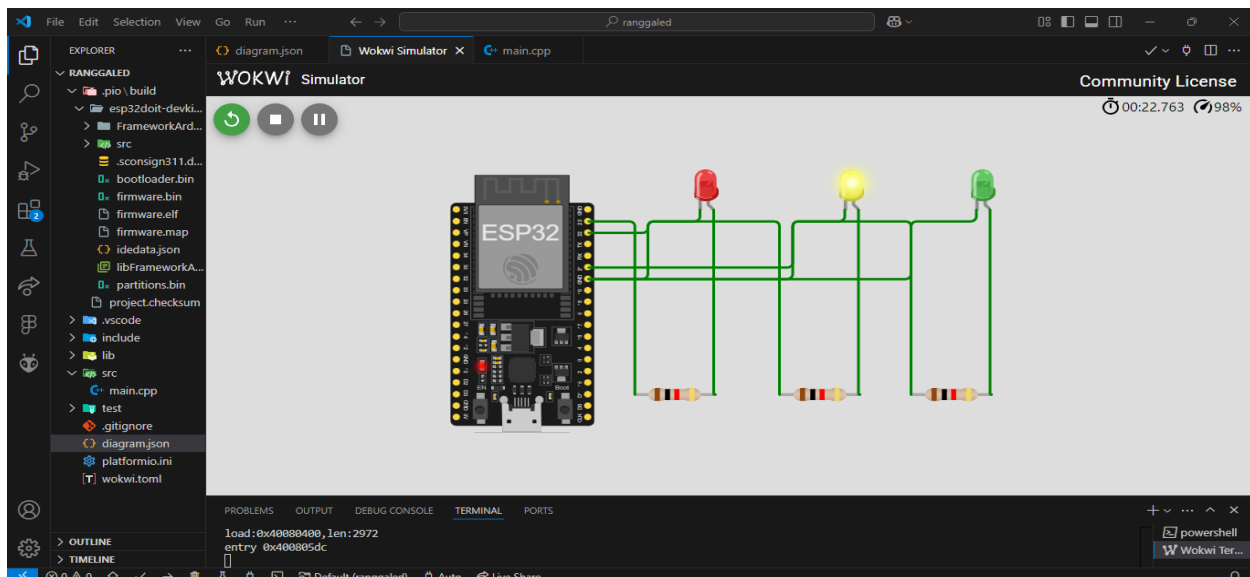
- Menilai apakah sistem berfungsi sesuai dengan rancangan awal
Jika sistem bekerja dengan baik dan mengikuti skenario yang telah dirancang, maka eksperimen dianggap berhasil.
- Mengidentifikasi kelebihan dan kekurangan sistem
Jika terdapat kekurangan dalam sistem, seperti penggunaan waktu yang tidak efisien atau kesalahan dalam logika nyala LED, maka dapat dicari solusi untuk memperbaikinya.
- Menganalisis kemungkinan pengembangan lebih lanjut
Untuk meningkatkan sistem, dapat dilakukan pengembangan lebih lanjut seperti menambahkan sensor untuk mendeteksi kendaraan atau menggunakan kontrol berbasis jaringan agar lampu lalu lintas dapat menyesuaikan dengan kondisi lalu lintas secara real-time.

3. Results and Discussion (Hasil dan Pembahasan)

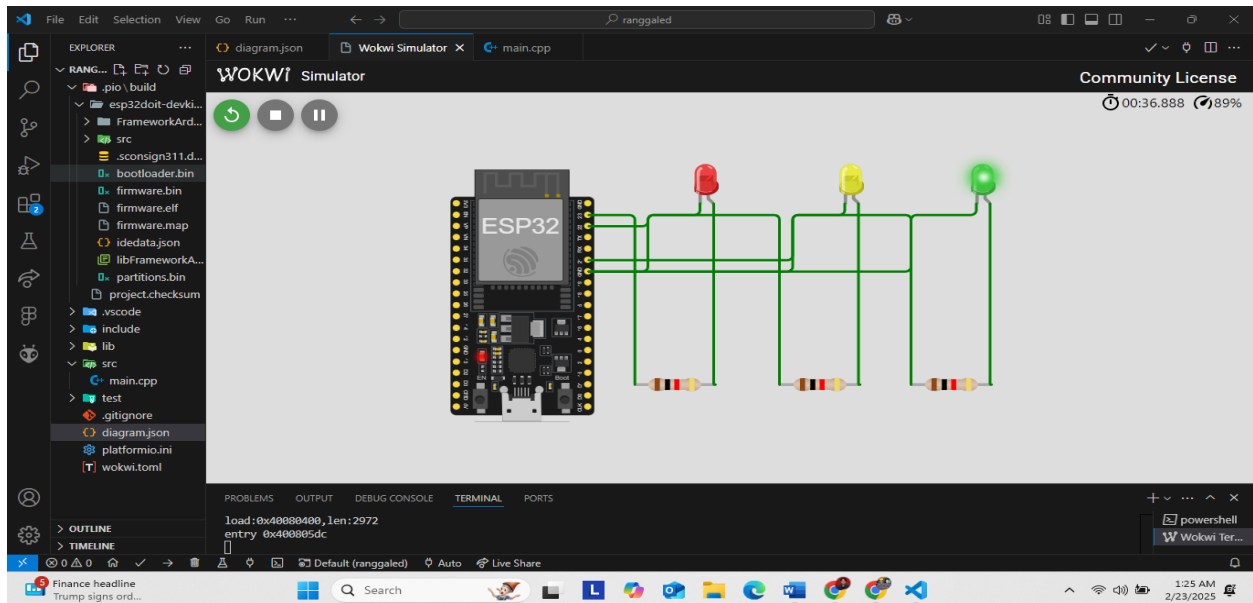
3.1 Lampu Merah Menyala



3.2 Lampu Kuning Menyala



3.3 Lampu Hijau Menyala



4. Appendix (Lampiran)

4.1 kode program untuk traffic light main.cpp

```
src > C++ main.cpp > loop()
1  #include <Arduino.h>
2
3  void setup() {
4      pinMode(23, OUTPUT); // LED Merah
5      pinMode(22, OUTPUT); // LED Kuning
6      pinMode(21, OUTPUT); // LED Hijau
7  }
8
9  void loop() {
10     // Lampu Merah menyala selama 20 detik
11     digitalWrite(23, HIGH);
12     digitalWrite(22, LOW);
13     digitalWrite(21, LOW);
14     delay(20000); // 20 detik
15
16     // Lampu Kuning menyala selama 5 detik
17     digitalWrite(23, LOW);
18     digitalWrite(22, HIGH);
19     digitalWrite(21, LOW);
20     delay(5000); // 5 detik
21
22     // Lampu Hijau menyala selama 30 detik
23     digitalWrite(23, LOW);
24     digitalWrite(22, LOW);
25     digitalWrite(21, HIGH);
26     delay(30000); // 30 detik
27 }
28
```

4.2 kode program untuk diagram traffic light diagram.json

```
1 {
2   "version": 1,
3   "author": "Anonymous maker",
4   "editor": "wokwi",
5   "parts": [
6     { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 0, "left": 0, "attrs": {} },
7     {
8       "type": "wokwi-resistor",
9       "id": "r1",
10      "top": 176.75,
11      "left": 134.4,
12      "attrs": { "value": "1000" }
13    },
14    {
15      "type": "wokwi-resistor",
16      "id": "r2",
17      "top": 176.75,
18      "left": 240,
19      "attrs": { "value": "1000" }
20    },
21    {
22      "type": "wokwi-resistor",
23      "id": "r3",
24      "top": 176.75,
25      "left": 336,
26      "attrs": { "value": "1000" }
27    },
28    { "type": "wokwi-led", "id": "led1", "top": -13.2, "left": 167, "attrs": { "color": "red" } },
29    {
30      "type": "wokwi-led",
31      "id": "led2",
32      "top": -13.2,
33      "left": 368.6,
34      "attrs": { "color": "limegreen" }
35    },
36    {
37      "type": "wokwi-led",
38      "id": "led3",
39      "top": -13.2,
40      "left": 272.6,
41      "attrs": { "color": "yellow" }
42    }
43  ],
44  "connections": [
45    [ "esp:TX", "$serialMonitor:RX", "", [ ] ],
46    [ "esp:RX", "$serialMonitor:TX", "", [ ] ],
47    [ "esp:23", "r1:1", "green", [ "h0" ] ],
48    [ "esp:22", "r2:1", "green", [ "h43.24", "v-9.6", "h86.4" ] ],
49    [ "r3:1", "esp:21", "green", [ "v-144", "h-86.4", "v38.4" ] ],
50    [ "led1:A", "r1:2", "green", [ "v0" ] ],
51    [ "led3:A", "r2:2", "green", [ "v0" ] ],
52    [ "led2:A", "r3:2", "green", [ "v0" ] ],
53    [ "led2:C", "esp:GND.3", "green", [ "v9.6", "h-47.6", "v38.4" ] ],
54    [ "led3:C", "esp:GND.3", "green", [ "v9.6", "h-38", "v48" ] ],
55    [ "led1:C", "esp:GND.3", "green", [ "v9.6", "h-38", "v48" ] ]
56  ],
57  "dependencies": {}
58 }
```