# LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Rangkaian Lampu Lalu Lintas *(Traffic Light)* dengan Simulasi ESP32**

*Eva Latifah*

*Fakultas Vokasi, Universitas Brawijaya*

*Email: ev.tifa@gmail.com*

**Abstrak**

Praktikum ini bertujuan untuk merancang dan mengimplementasikan sistem lampu lalu lintas menggunakan mikrokontroler ESP32. Sistem ini dikembangkan dengan pemrograman menggunakan Arduino IDE dan melibatkan penggunaan LED untuk merepresentasikan sinyal lalu lintas merah, kuning, dan hijau. Praktikum dilakukan dengan mengatur waktu nyala masing-masing lampu sesuai dengan aturan lalu lintas standar. Hasil utama menunjukkan bahwa ESP32 mampu mengontrol lampu lalu lintas secara otomatis dengan durasi yang telah diprogram. Selain itu, sistem dapat dikembangkan lebih lanjut dengan menambahkan sensor atau komunikasi nirkabel untuk meningkatkan efisiensi pengaturan lalu lintas. Kesimpulannya, eksperimen ini berhasil menunjukkan bahwa ESP32 dapat digunakan sebagai pengontrol utama dalam sistem lampu lalu lintas dengan implementasi yang sederhana namun efektif.

*Kata Kunci—ESP32, Lampu Lalu Lintas, Mikrokontroler, Kontrol Otomatis, Internet of Things*

***Abstract***

*This practicum aims to design and implement a traffic light system using the ESP32 microcontroller. The system is developed using Arduino IDE programming and involves the use of LEDs to represent red, yellow, and green traffic signals. The practicum is conducted by setting the lighting duration of each light according to standard traffic regulations. The main results show that the ESP32 can automatically control traffic lights with pre-programmed durations. Furthermore, the system can be further developed by adding sensors or wireless communication to improve traffic management efficiency. In conclusion, this experiment successfully demonstrates that the ESP32 can be used as the main controller in a traffic light system with a simple yet effective implementation.*

*Keywords—ESP32, Traffic Light, Microcontroller, Automatic Control, Internet of Things*

**1. Introduction**

**1.1 Latar belakang**

Lampu lalu lintas merupakan sistem pengendalian lalu lintas yang berfungsi untuk mengatur pergerakan kendaraan dan pejalan kaki di persimpangan jalan. Penggunaan teknologi dalam sistem ini terus berkembang untuk meningkatkan efisiensi dan keamanan. Salah satu teknologi yang dapat digunakan untuk mengontrol lampu lalu lintas adalah mikrokontroler ESP32.

ESP32 merupakan mikrokontroler dengan konektivitas Wi-Fi dan Bluetooth yang memiliki kemampuan pemrosesan tinggi serta efisiensi daya yang baik. Dengan memanfaatkan ESP32, sistem lampu lalu lintas dapat dikendalikan secara otomatis tanpa memerlukan perangkat tambahan seperti Arduino. Pemrograman langsung pada ESP32 memungkinkan implementasi yang lebih fleksibel dan dapat dikembangkan lebih lanjut, misalnya dengan sensor atau sistem berbasis Internet of Things (IoT).

**1.2 Tujuan eksperimen**

Praktikum ini bertujuan untuk merancang dan mengimplementasikan sistem lampu lalu lintas berbasis ESP32. Eksperimen dilakukan dengan memprogram mikrokontroler untuk mengontrol nyala lampu merah, kuning, dan hijau sesuai dengan aturan lalu lintas standar. Hasil eksperimen diharapkan dapat menunjukkan bahwa ESP32 mampu menjadi pengontrol utama dalam sistem lampu lalu lintas yang sederhana dan efektif.

**2. Methodology**

**2.1 Tools & Materials**

1. Mikrokontroler ESP32
2. LED merah, kuning, hijau
3. Resistor 220Ω
4. Breadboard dan jumper kabel
5. Software Arduino IDE
6. Web Wokwi
7. Vscode

**2.2 Implementation Steps**

1. Menghubungkan LED ke ESP32 dengan resistor sebagai pembatas arus.
2. Menulis kode program di Arduino IDE untuk mengatur durasi nyala lampu merah, kuning, dan hijau.
3. Mengunggah kode ke ESP32 dan menjalankan simulasi.
4. Mengamati perilaku lampu lalu lintas sesuai dengan kode yang telah diprogram.
5. Mengevaluasi dan menyempurnakan sistem jika diperlukan.

**3. Results and Discussion**

**3.1 Experimental Results**

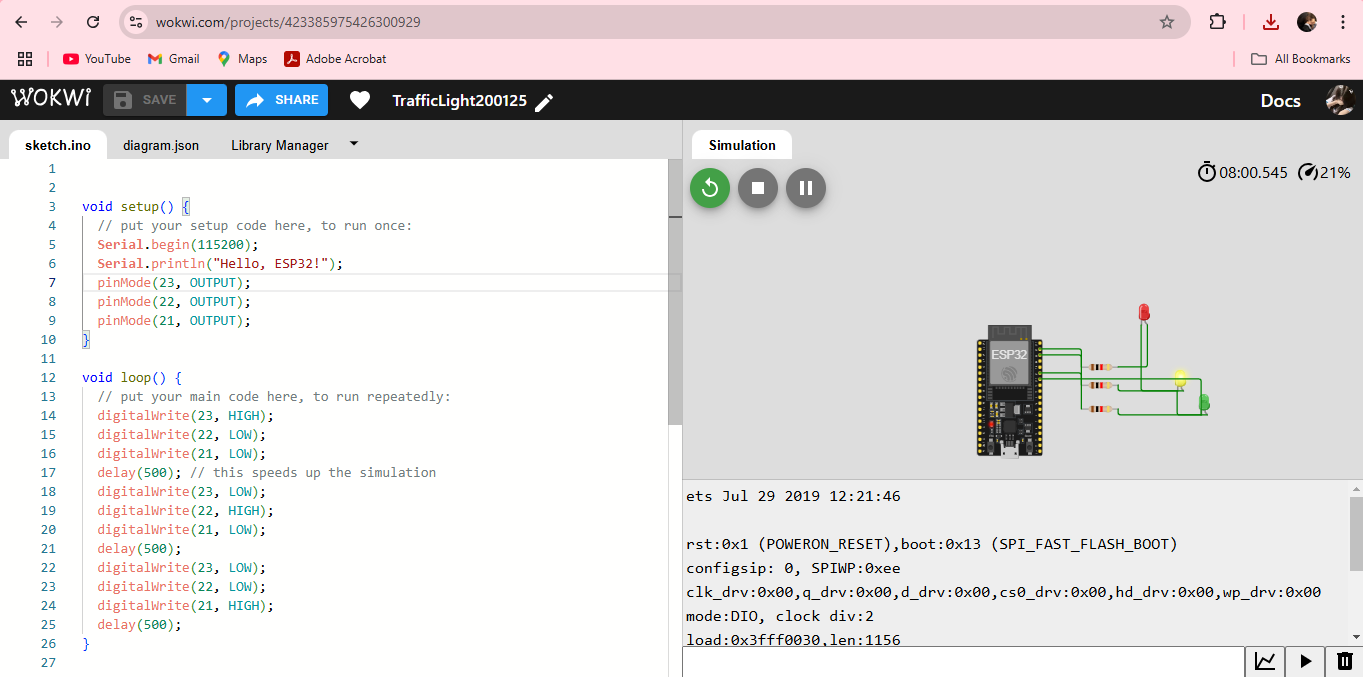
Sistem lampu lalu lintas berbasis ESP32 yang telah diimplementasikan menunjukkan hasil yang sesuai dengan skenario yang telah dirancang. Lampu merah, kuning, dan hijau menyala dalam durasi yang telah diprogram, sesuai dengan aturan lalu lintas standar. Hal ini membuktikan bahwa ESP32 dapat digunakan sebagai pengontrol utama dalam sistem lampu lalu lintas.

Keunggulan dari penggunaan ESP32 dalam sistem ini adalah kemampuannya untuk dikembangkan lebih lanjut. Dengan menambahkan sensor seperti sensor cahaya atau sensor keberadaan kendaraan, sistem dapat menjadi lebih adaptif terhadap kondisi lalu lintas yang sebenarnya. Selain itu, integrasi dengan komunikasi nirkabel memungkinkan sistem ini dikontrol dan dipantau secara real-time melalui jaringan IoT.

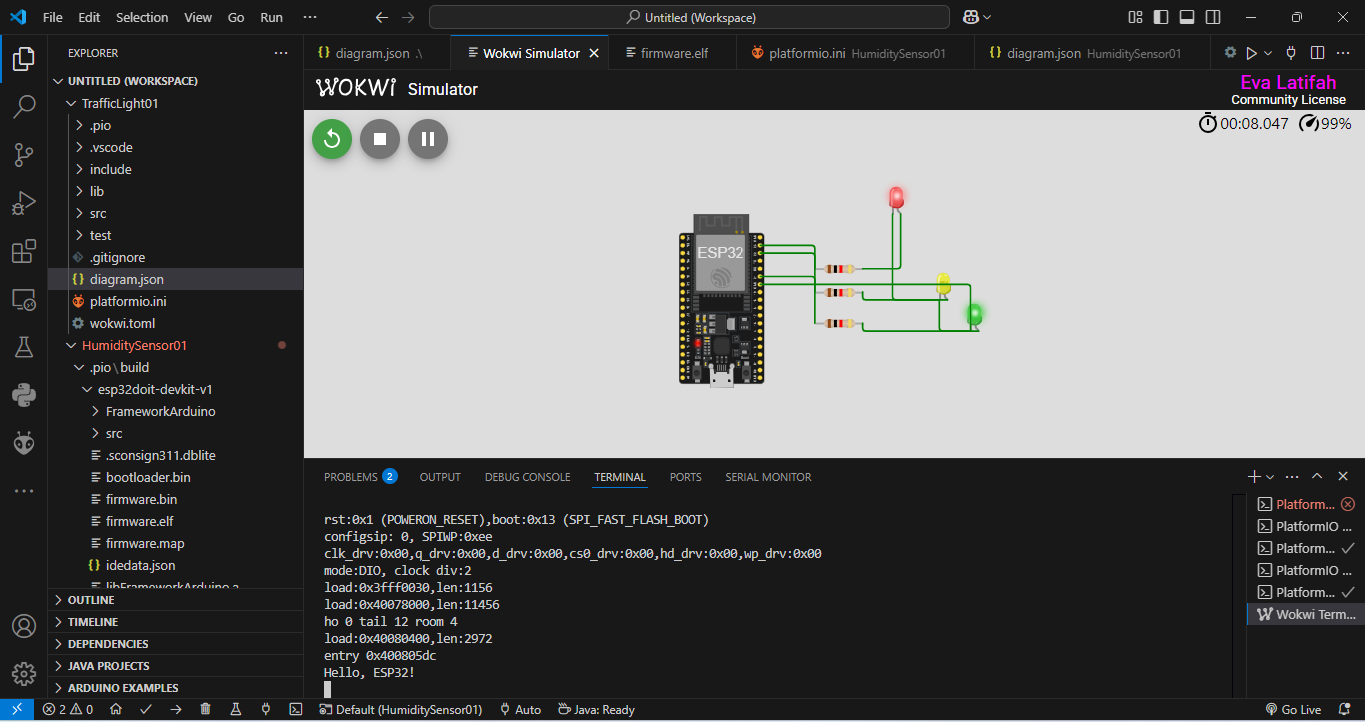
Namun, terdapat beberapa tantangan yang perlu diperhatikan dalam implementasi sistem ini. Salah satunya adalah keterbatasan daya dan kestabilan koneksi jika digunakan dalam skala besar. Selain itu, pemrograman ESP32 memerlukan pemahaman mendalam mengenai algoritma pengendalian waktu untuk memastikan transisi lampu berlangsung secara akurat dan tidak terjadi konflik dalam pengaturan sinyal lalu lintas.

Secara keseluruhan, eksperimen ini berhasil menunjukkan bahwa ESP32 merupakan solusi yang efektif dan efisien dalam pembuatan sistem lampu lalu lintas otomatis. Dengan pengembangan lebih lanjut, sistem ini dapat digunakan dalam implementasi nyata untuk meningkatkan efisiensi lalu lintas dan keselamatan di jalan ray

**Result Wokwi:**



**Result Vscode:**



**4. Appendix**

void setup() {

// put your setup code here, to run once:

**Serial**.begin(115200);

**Serial**.println("Hello, ESP32!");

pinMode(23, OUTPUT);

pinMode(22, OUTPUT);

pinMode(21, OUTPUT);

}

void loop() {

// put your main code here, to run repeatedly:

digitalWrite(23, HIGH);

digitalWrite(22, LOW);

digitalWrite(21, LOW);

delay(500); // this speeds up the simulation

digitalWrite(23, LOW);

digitalWrite(22, HIGH);

digitalWrite(21, LOW);

delay(500);

digitalWrite(23, LOW);

digitalWrite(22, LOW);

digitalWrite(21, HIGH);

delay(500);

}