

LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

Simulasi Sistem Traffic Light dengan ESP32 pada Wokwi

Salsabilla Permata Dewi – 233140701111016

Fakultas Vokasi, Universitas Brawijaya

Email: salsabillapermata02@gmail.com

Abstract (Abstrak)

Praktikum ini bertujuan untuk memahami dasar-dasar penggunaan LED pada mikrokontroler melalui simulasi sistem lampu lalu lintas menggunakan ESP32 di platform Wokwi. Simulasi ini melibatkan tiga LED berwarna merah, kuning, dan hijau yang menggambarkan lampu lalu lintas. LED dikendalikan oleh program sederhana pada ESP32 untuk menyala secara bergantian sesuai urutan yang umum digunakan pada sistem lampu lalu lintas. Proses simulasi dilakukan dengan menyusun rangkaian elektronik virtual di Wokwi dan mengupload kode program ke ESP32. Program ini mengatur waktu menyala dan mati setiap LED berdasarkan waktu yang telah ditentukan. Melalui simulasi ini, kita dapat memahami cara mengontrol LED menggunakan mikrokontroler, konsep dasar delay, serta logika pengaturan lampu lalu lintas sederhana. Hasil simulasi menunjukkan bahwa ESP32 dapat mengontrol LED dengan baik sesuai urutan yang diinginkan. Penggunaan Wokwi juga memudahkan pengujian program tanpa memerlukan perangkat keras. Dengan demikian, praktikum ini membantu memahami dasar-dasar pemrograman mikrokontroler dan cara kerja LED dalam aplikasi sederhana seperti lampu lalu lintas.

Kata kunci—ESP32, LED, Traffic Light, Wokwi, Simulasi.

1. Pendahuluan

1.1. Latar Belakang

Teknologi Internet of Things (IoT) telah menjadi bagian penting dalam kehidupan sehari-hari. IoT memungkinkan perangkat elektronik untuk berkomunikasi dan berinteraksi dengan lingkungan sekitar, sehingga meningkatkan efisiensi dan kenyamanan. Salah satu contoh aplikasi IoT adalah sistem kendali lalu lintas, yang menggunakan sensor dan aktuator untuk mengatur aliran lalu lintas dan mengurangi kemacetan.

Pengendalian perangkat elektronik seperti LED menggunakan mikrokontroler merupakan konsep dasar yang penting dalam IoT. Mikrokontroler adalah komponen elektronik yang dapat diprogram untuk mengendalikan perangkat lain, seperti LED, motor, dan sensor. Dalam sistem kendali lalu lintas, mikrokontroler digunakan untuk mengendalikan lampu lalu lintas, sehingga memastikan aliran lalu lintas yang lancar dan aman.

Praktikum ini bertujuan untuk memahami bagaimana mikrokontroler ESP32 mengontrol LED yang merepresentasikan sistem traffic light sederhana. Dengan melakukan simulasi pada platform Wokwi, mahasiswa dapat memahami dasar pemrograman mikrokontroler tanpa harus menggunakan perangkat keras fisik. Simulasi ini membantu memahami konsep pengendalian LED berbasis waktu dan logika pengaturan lampu lalu lintas sederhana.

1. 2. Tujuan eksperimen

Praktikum ini memiliki beberapa tujuan utama:

- Memahami cara kerja ESP32 dalam mengendalikan LED.
- Mempelajari konsep dasar pengaturan waktu (delay) dalam pemrograman mikrokontroler.
- Menyusun dan menguji kode program untuk mengontrol LED sebagai simulasi traffic light.
- Menggunakan platform Wokwi sebagai media simulasi untuk memahami konsep tanpa perangkat keras fisik.

2. Metodologi

2. 1. Alat dan Bahan

Simulasi ini dilakukan secara virtual menggunakan Wokwi, sehingga tidak memerlukan perangkat keras. Alat dan bahan virtual yang digunakan dalam simulasi traffic light ini meliputi:

a) Mikrokontroler Virtual ESP32

Pada praktikum ini saya menggunakan ESP32 sebagai mikrokontroler yang mengontrol nyala LED dalam simulasi. ESP32 dipilih karena memiliki banyak pin GPIO yang memungkinkan kita mengendalikan beberapa LED secara bersamaan.

b) Komponen Elektronik Virtual

– LED Virtual (Merah, Kuning, Hijau)

LED ini digunakan sebagai indikator dalam simulasi sistem traffic light. Merah berarti kendaraan harus berhenti, kuning berarti peringatan untuk bersiap, dan hijau berarti kendaraan boleh berjalan.

– Kabel Virtual

Kabel virtual digunakan untuk menghubungkan pin GPIO ESP32 ke masing-masing LED.

c) Perangkat Lunak (*Software*)

– Wokwi Simulator

Wokwi Simulator digunakan untuk merancang sirkuit dan menjalankan simulasi sistem traffic light tanpa perangkat keras fisik.

– PlatformIO

PlatformIO digunakan sebagai lingkungan pengembangan kode untuk menulis dan mengelola kode program berbasis ESP32. PlatformIO memiliki integrasi yang lebih baik dengan Visual Studio Code dan fitur build, upload, dan debugging yang lebih canggih.

– Bahasa Pemrograman C++

Bahasa pemrograman C++ digunakan untuk mengontrol LED dalam simulasi dengan perintah `digitalWrite()`, `delay()`, dan struktur `loop`.

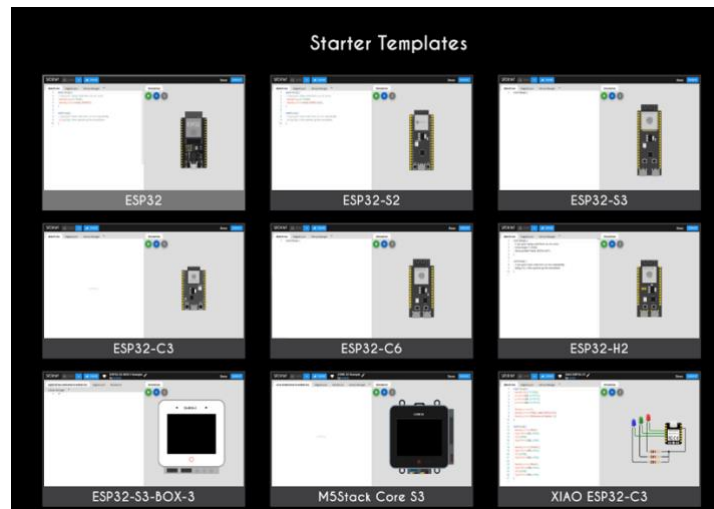
2. 2. Langkah Implementasi

a) Menyiapkan Simulasi Wokwi

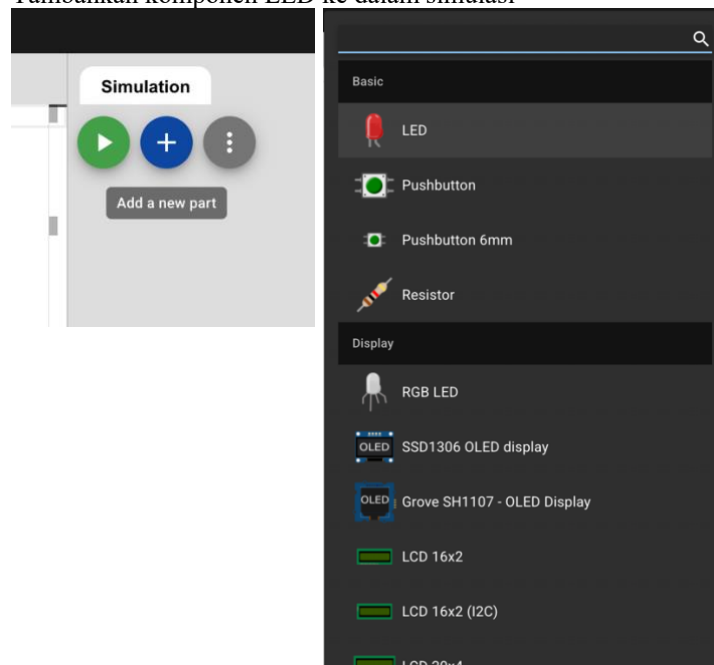
- Membuka situs Wokwi dan pilih proyek ESP32.



- Pilih starter template ESP32.



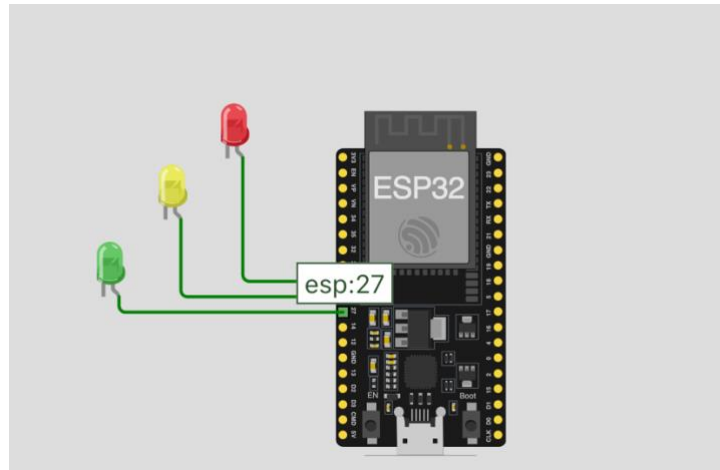
- Tambahkan komponen LED ke dalam simulasi



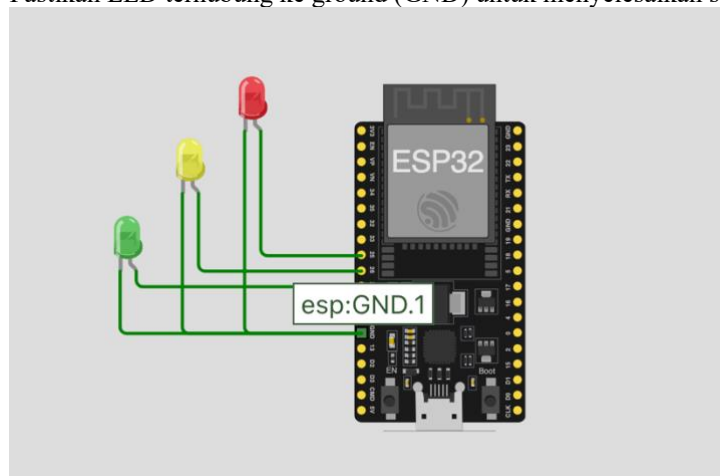
- Klik LED untuk mengubah warna merah, kuning dan hijau.



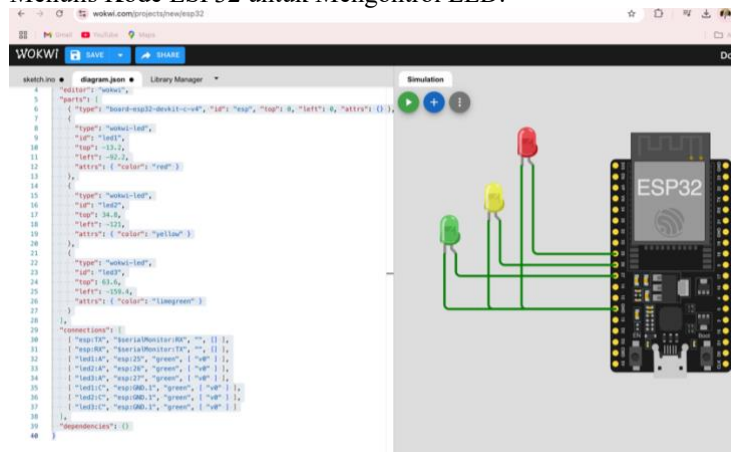
- Hubungkan LED ke pin GPIO ESP32 sesuai rancangan rangkaian.



- Pastikan LED terhubung ke ground (GND) untuk menyelesaikan sirkuit.



- Menulis Kode ESP32 untuk Mengontrol LED.



Berikut adalah kodenya :

```
#include <Arduino.h>

// Deklarasi pin lampu
int lampu = 25; // Lampu merah
int lampu2 = 26; // Lampu kuning
int lampu3 = 27; // Lampu hijau

void setup() {
  Serial.begin(115200);
  Serial.println("Hello, ESP32!");

  // Mengatur pin sebagai output
  pinMode(lampu, OUTPUT);
  pinMode(lampu2, OUTPUT);
  pinMode(lampu3, OUTPUT);
}

void loop() {
  // Lampu merah ON
  digitalWrite(lampu, HIGH);
  Serial.println("Lampu Merah ON");
  delay(45000);
  digitalWrite(lampu, LOW);
  Serial.println("Lampu Merah OFF");
  delay(100);

  // Lampu kuning pertama ON (sebelum hijau)
  digitalWrite(lampu2, HIGH);
  Serial.println("Lampu Kuning ON");
  delay(2000);
  digitalWrite(lampu2, LOW);
  Serial.println("Lampu Kuning OFF");
  delay(100);

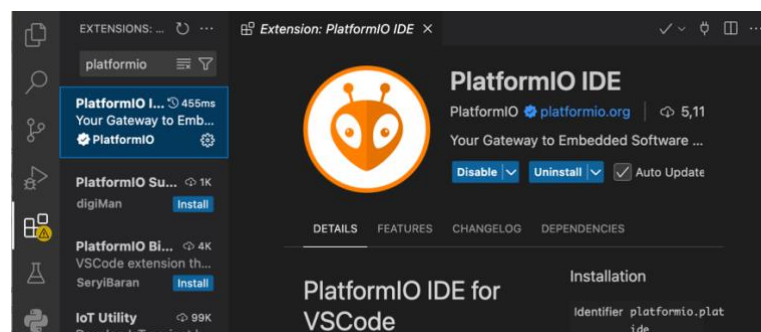
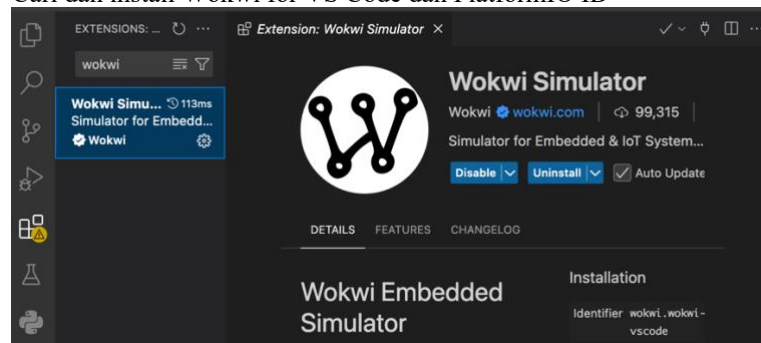
  // Lampu hijau ON
  digitalWrite(lampu3, HIGH);
  Serial.println("Lampu Hijau ON");
  delay(11000);
  digitalWrite(lampu3, LOW);
  Serial.println("Lampu Hijau OFF");
  delay(100);

  // Lampu kuning kedua ON (sebelum merah)
  digitalWrite(lampu2, HIGH);
  Serial.println("Lampu Kuning ON (Sebelum Merah)");
  delay(2000);
  digitalWrite(lampu2, LOW);
  Serial.println("Lampu Kuning OFF");

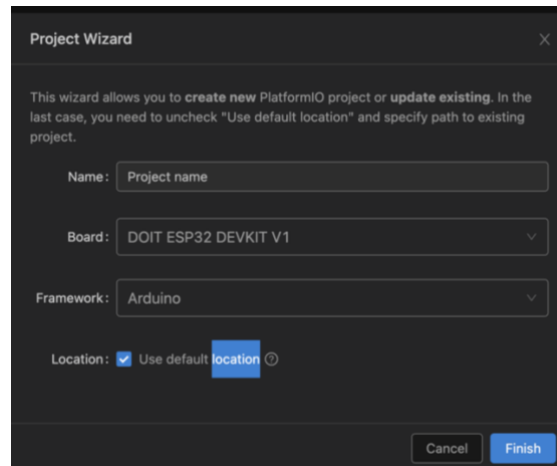
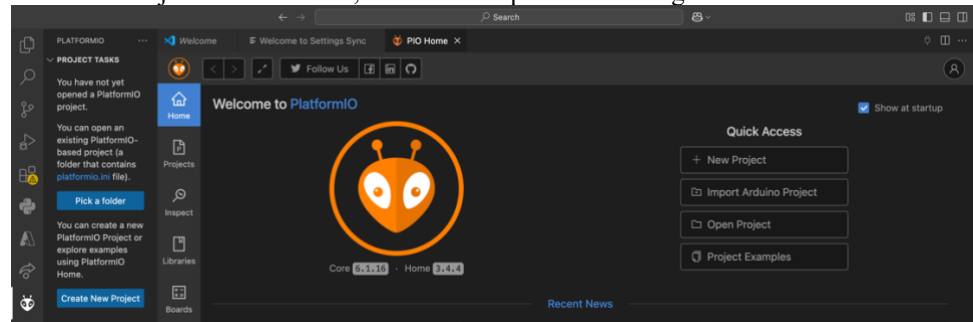
  // Kembali ke awal (Merah lagi)
}
```

b) Instalasi Software Pendukung

- Buka vscode kemudian masuk ke Extensions Marketplace
- Cari dan install Wokwi for VS Code dan PlatformIO ID



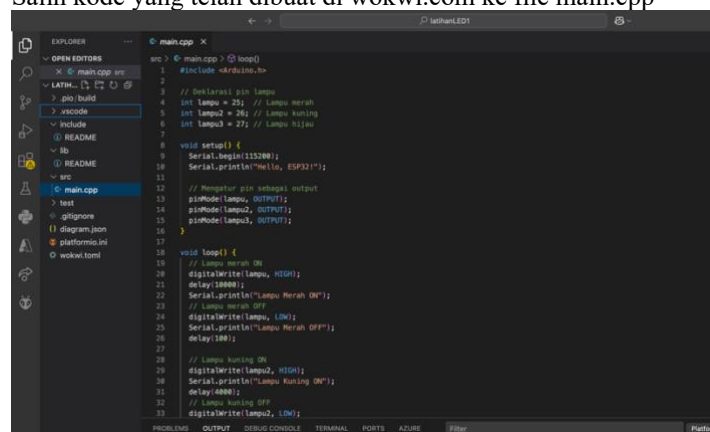
- Klik New Project di PlatformIO, kemudian isi parameter sebagai berikut :



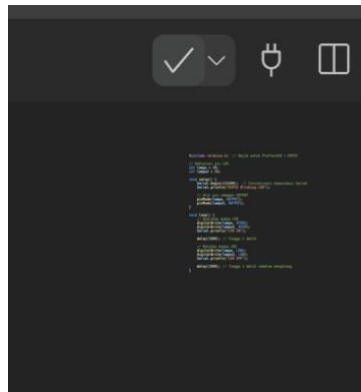
Tunggu hingga PlatformIO selesai mengonfigurasi proyek.

c) Penulisan dan Kompilasi Kode

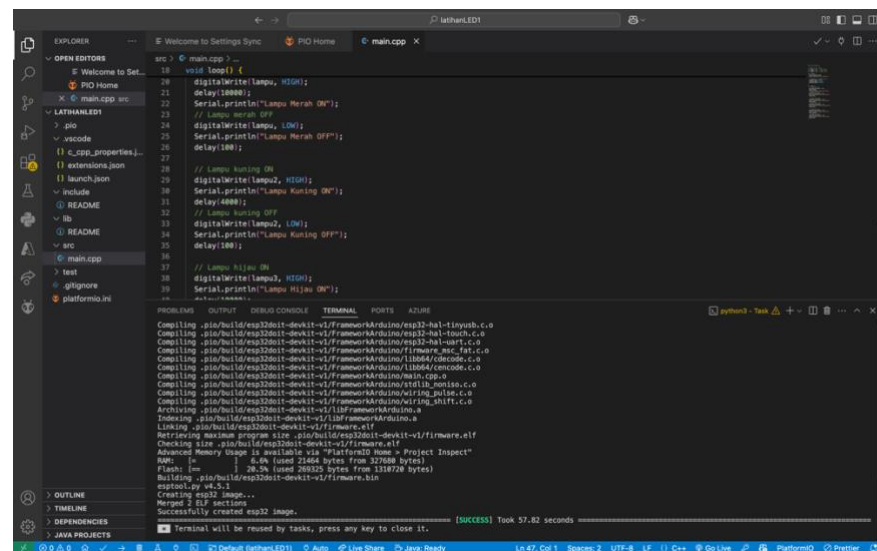
- Salin kode yang telah dibuat di wokwi.com ke file main.cpp



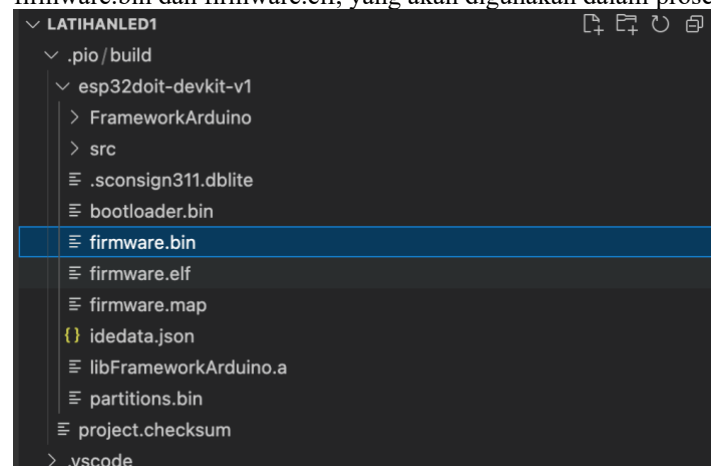
- Tekan tombol centang di pojok kanan atas PlatformIO untuk melakukan proses kompilasi hingga selesai tanpa error.



Vscode akan melakukan proses kompilasi seperti tampilan berikut.tunggu hingga sukses.

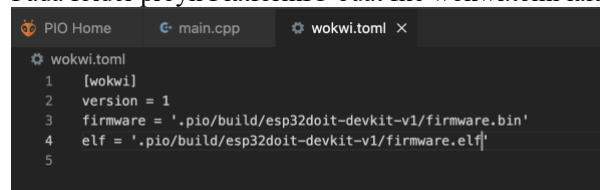


- Setelah proses kompilasi berhasil, maka akan menemukan dua file penting, yaitu firmware.bin dan firmware.elf, yang akan digunakan dalam proses simulasi.



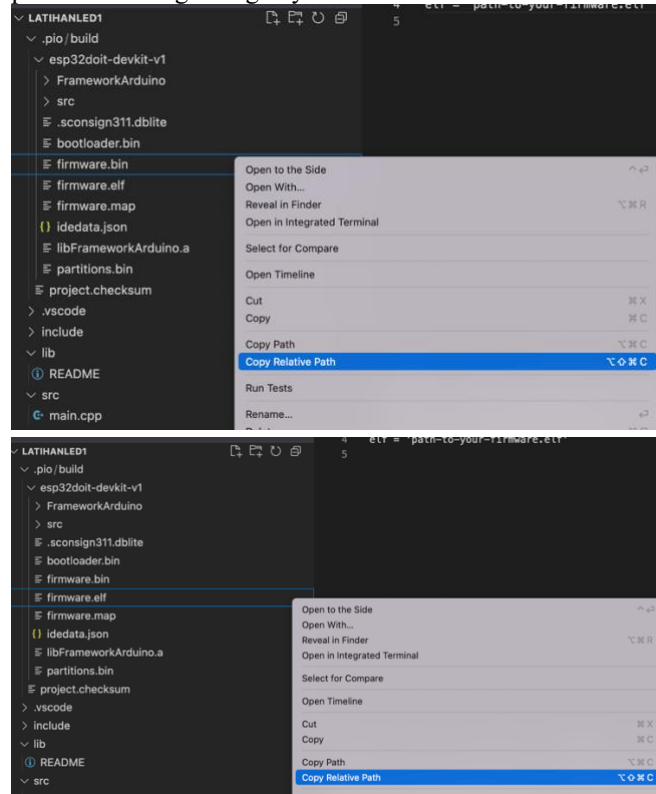
d) Integrasi dengan Wokwi

- Pada folder proyek PlatformIO buat file wokwi.toml lalu tambahkan :

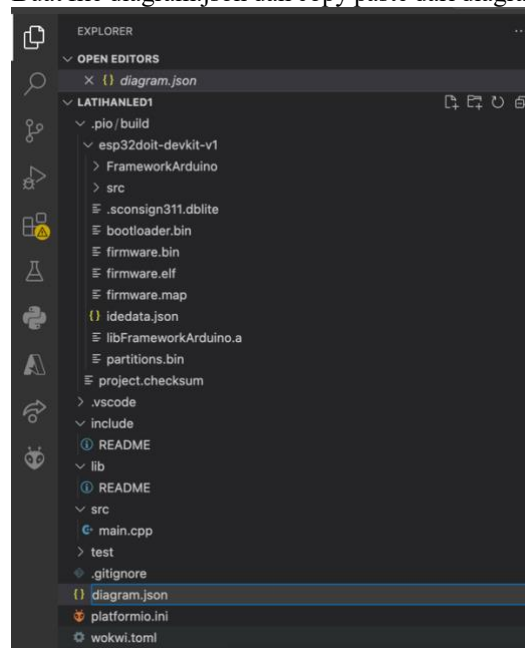


```
wokwi.toml
1 [wokwi]
2 version = 1
3 firmware = '.pio/build/esp32doit-devkit-v1/firmware.bin'
4 elf = '.pio/build/esp32doit-devkit-v1/firmware.elf'
5
```

- Kemudian copy relative path dari file firmware.bin dan firmware.elf dan isikan relative path sesuai dengan bagianya.



- Buat file diagram.json dan copy paste dari diagram json yang ada di wokwi.com platform



← → ↻ wokwi.com/projects/new/esp32

WOKWI SAVE SHARE

sketch.ino ● diagram.json ● Library Manager ▾

```

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-led",
9       "id": "led1",
10      "top": -13.2,
11      "left": -92.2,
12      "attrs": { "color": "red" }
13    },
14    {
15      "type": "wokwi-led",
16      "id": "led2",
17      "top": 34.8,
18      "left": -121,
19      "attrs": { "color": "yellow" }
20    },
21    {
22      "type": "wokwi-led",
23      "id": "led3",
24      "top": 63.6,
25      "left": -159.4,
26      "attrs": { "color": "limegreen" }
27    }
28  ],
29  "connections": [
30    [ "esp:TX", "$SerialMonitor:RX", "", [ ] ],
31    [ "esp:RX", "$SerialMonitor:TX", "", [ ] ],
32    [ "led1:A", "esp:25", "green", [ "v0" ] ],
33    [ "led2:A", "esp:26", "green", [ "v0" ] ],
34    [ "led3:A", "esp:27", "green", [ "v0" ] ],
35    [ "led1:C", "esp:GND.1", "green", [ "v0" ] ],
36    [ "led2:C", "esp:GND.1", "green", [ "v0" ] ],
37    [ "led3:C", "esp:GND.1", "green", [ "v0" ] ]
38  ],
39  "dependencies": {}
40 }

```

← → ↻ lathaned1

EXPLORER

PIO Home ● main.cpp ● wokwi.toml {} diagram.json ●

OPEN EDITORS 1 unsaved

- PIO Home
- main.cpp src
- wokwi.toml
- {} diagram.json

LATHANED1

- pio/build
- esp32doit-devkit-v1
 - FrameworkArduino
 - src
 - .sconsign311.dblite
 - bootloader.bin
 - firmware.bin
 - firmware.elf
 - firmware.map
 - {} idedata.json
 - libFrameworkArduino.a
 - partitions.bin
 - project.checksum
 - .vscode
 - include
 - lib
 - README
 - src
 - main.cpp
 - test
 - .gitignore
 - {} diagram.json
 - platformio.ini
 - wokwi.toml
- OUTLINE

{} diagram.json > ...

```

1 {
2   "version": 1,
3   "author": "rachmad andri atmoko",
4   "editor": "wokwi",
5   "parts": [
6     { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 0, "left": 0, "attrs": {} },
7     { "type": "wokwi-led", "id": "led1", "top": -32.4, "left": -73, "attrs": {} },
8     {
9       "type": "wokwi-led",
10      "id": "led2",
11      "top": -13.2,
12      "left": -138.6,
13      "attrs": { "color": "yellow" }
14    },
15    {
16      "type": "wokwi-led",
17      "id": "led3",
18      "top": 63.6,
19      "left": -159.4,
20      "attrs": { "color": "limegreen" }
21    }
22  ],
23  "connections": [
24    [ "esp:TX", "$SerialMonitor:RX", "", [ ] ],
25    [ "esp:RX", "$SerialMonitor:TX", "", [ ] ],
26    [ "led1:A", "esp:25", "green", [ "v0" ] ],
27    [ "led2:A", "esp:26", "green", [ "v0" ] ],
28    [ "led3:A", "esp:27", "green", [ "v0" ] ],
29    [ "led1:C", "esp:GND.1", "green", [ "v0" ] ],
30    [ "led2:C", "esp:GND.1", "green", [ "v0" ] ],
31    [ "led3:C", "esp:GND.1", "green", [ "v0" ] ]
32  ],
33  "dependencies": {}
34 }

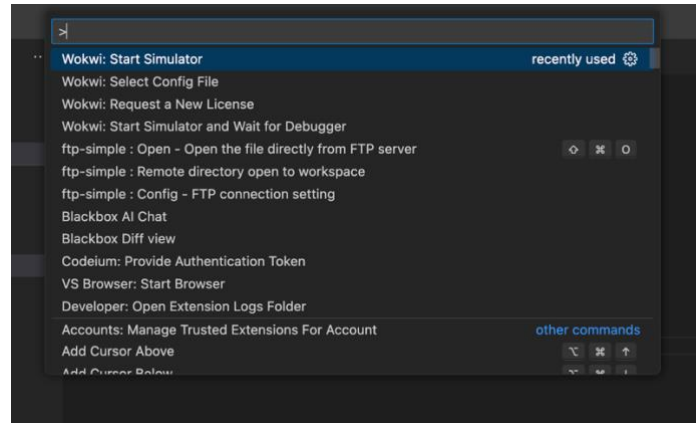
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

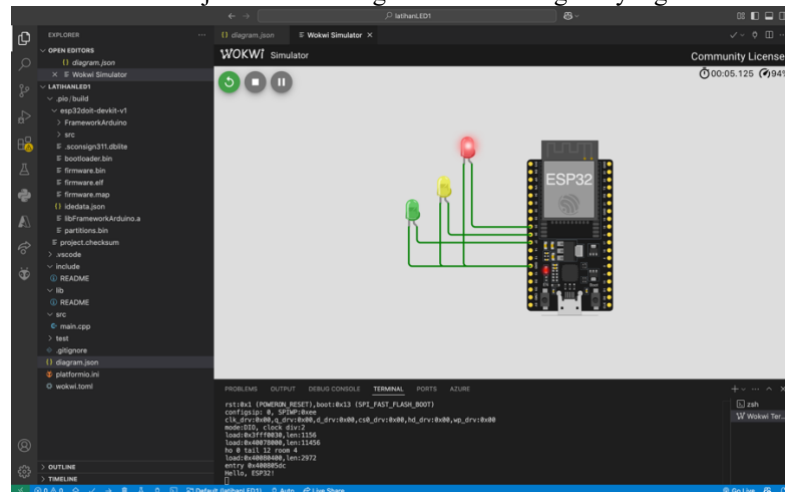
Compiling .pio/build/esp32doit-devkit-v1/FrameworkArduino/esp32-hal-uart.c.o
 Compiling .pio/build/esp32doit-devkit-v1/FrameworkArduino/firmware_esc_fat.c.o
 Compiling .pio/build/esp32doit-devkit-v1/FrameworkArduino/libb64/decode.c.o
 Compiling .pio/build/esp32doit-devkit-v1/FrameworkArduino/main.cpp.o
 Compiling .pio/build/esp32doit-devkit-v1/FrameworkArduino/stdlib_noniso.c.o
 Compiling .pio/build/esp32doit-devkit-v1/FrameworkArduino/wiring_pulse.c.o
 Compiling .pio/build/esp32doit-devkit-v1/FrameworkArduino/wiring_shift.c.o
 Archiving .pio/build/esp32doit-devkit-v1/libFrameworkArduino.a
 Indexing .pio/build/esp32doit-devkit-v1/libFrameworkArduino.a
 Linking .pio/build/esp32doit-devkit-v1/firmware.elf
 Retrieving maximum program size .pio/build/esp32doit-devkit-v1/firmware.elf
 Checking size .pio/build/esp32doit-devkit-v1/firmware.elf
 Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
 RAM: [=====] 6.6% (used 21464 bytes from 327680 bytes)
 Flash: [=====] 20.5% (used 269153 bytes from 1310720 bytes)
 Building .pio/build/esp32doit-devkit-v1/firmware.bin
 esp32doit-v4.5-1
 Creating esp32 image...
 Merged 2 ELF sections
 Successfully created esp32 image. [SUCCESS] Took 26.84 seconds
 Terminal will be reused by tasks, press any key to close it.

e) Pengujian dan Debugging

- Langkah terakhir, Buka Command Palette di VS Code (Ctrl + Shift + P) lalu ketik Wokwi: Start Simulator lalu tekan Enter.



- Simulasi akan berjalan sesuai dengan kode dan diagram yang telah dibuat.



3. Hasil dan Pembahasan

3.1. Hasil Eksperimen

Dalam eksperimen ini, dilakukan simulasi sistem lampu lalu lintas menggunakan ESP32 pada platform Wokwi, dengan bantuan PlatformIO untuk proses kompilasi. Simulasi ini bertujuan untuk memahami cara kerja pengendalian lampu lalu lintas secara digital menggunakan mikrokontroler. Hasil eksperimen berupa output dari simulasi yang menampilkan nyala LED sesuai urutan lalu lintas yang telah diprogram. Dalam simulasi ini, dibuat skenario lampu lalu lintas di perempatan untuk memastikan kendaraan dapat melintas dengan aman dan menghindari tabrakan. Skenario ini dirancang berdasarkan sistem lampu lalu lintas di dunia nyata, dengan pengaturan waktu sebagai berikut :

- Lampu Merah : Menyala selama 45 detik untuk menghentikan kendaraan.
- Lampu Kuning: Menyala selama 2 detik sebagai peringatan sebelum lampu hijau menyala.
- Lampu Hijau: Menyala selama 11 detik untuk memberi kesempatan kendaraan melaju.
- Lampu Kuning (lagi): Menyala kembali selama 2 detik untuk memperingatkan pengendara sebelum lampu kembali ke merah.

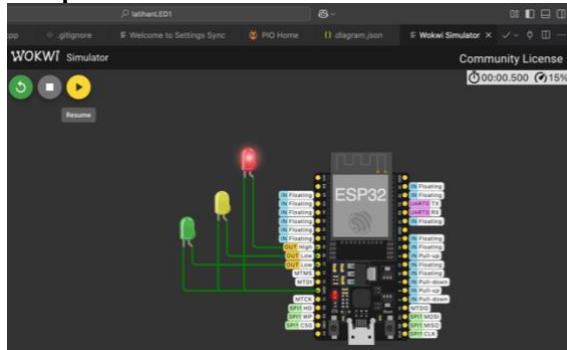
Tabel Hasil Simulasi

No.	Durasi (millisecond)	Status LED Merah	Status LED Kuning	Status LED Hijau	Rentang Waktu	Bukti
1.	45000ms	On	Off	Off	00 : 00,500 – 00 : 45,465	<i>Gambar 4.1</i>
2.	2000ms	Off	On	Off	00 : 45,783 – 00 : 47,50	<i>Gambar 4.2</i>
3.	11000ms	Off	Off	On	00 : 47,719 – 00 : 59,08	<i>Gambar 4.3</i>
4.	2000ms	Off	On	Off	00 : 59,145 – 01 : 00,625	<i>Gambar 4.4</i>
5.	45000ms	On	Off	Off	01 : 00,826 – 01 : 45,850	<i>Gambar 4.5</i>
6.	2000ms	Off	On	Off	01 : 45,850 – 01: 48,08	<i>Gambar 4.6</i>
7.	11000ms	Off	Off	On	01 : 48,165 – 01 : 59,189	<i>Gambar 4.7</i>

Tabel Simulasi menunjukkan bahwa setiap perubahan lampu berjalan sesuai dengan waktu yang telah diprogram tanpa adanya kesalahan.

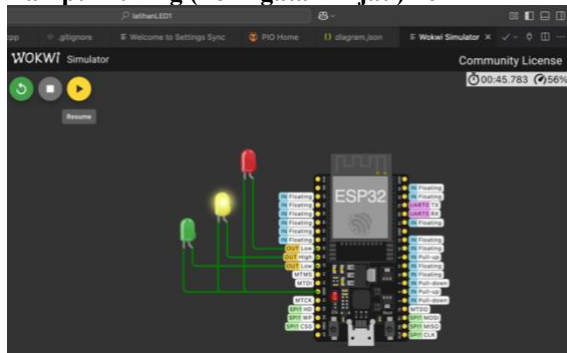
4. Lampiran (jika diperlukan)

4. 1. Lampu Merah ke - 1



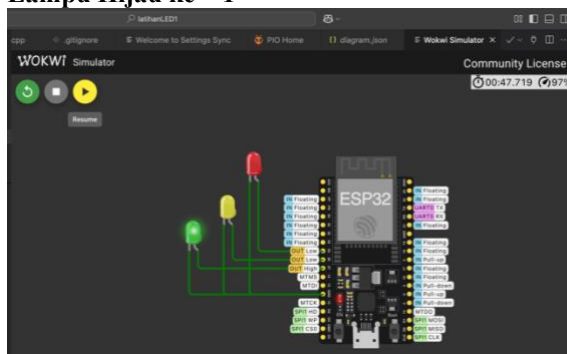
Gambar 4.1 Merah Satu

4. 2. Lampu Kuning (Peringatan Hijau) ke – 1



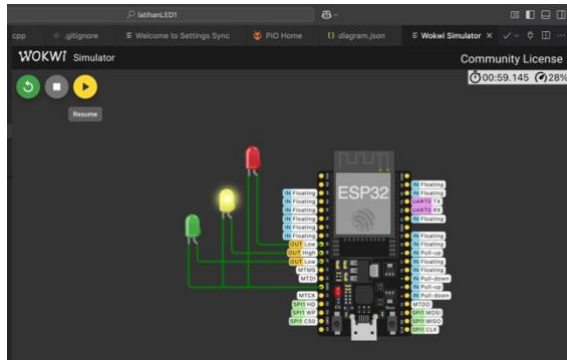
Gambar 4.2 Kuning Hijau Satu

4. 3. Lampu Hijau ke – 1



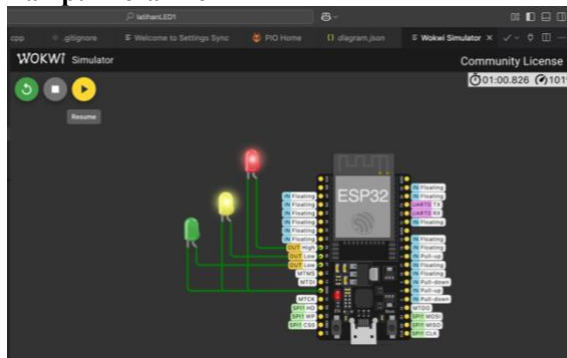
Gambar 4.3 Hijau Satu

4. 4. Lampu Kuning (Peringatan Merah) ke – 1



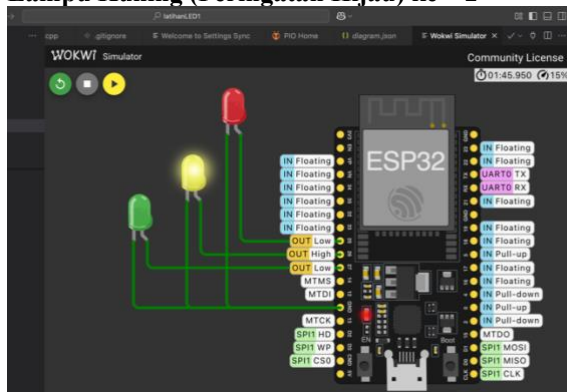
Gambar 4.4 Kuning Merah Satu

4. 5. Lampu Merah ke – 2



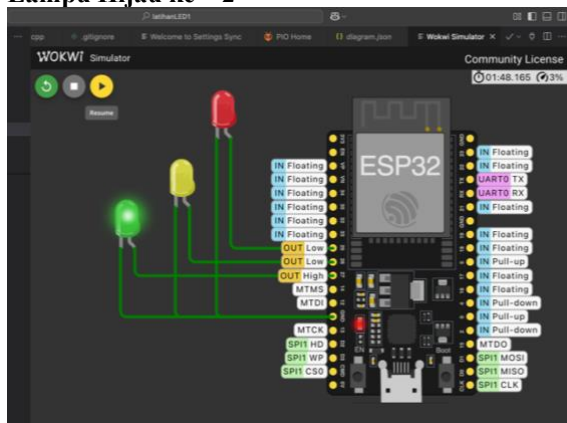
Gambar 4.5 Merah Dua

4. 6. Lampu Kuning (Peringatan Hijau) ke – 2



Gambar 4.6 Kuning Hijau Dua

4. 7. Lampu Hijau ke – 2



Gambar 4.7 Hijau Dua

