**LAPORAN PRAKTIKUM INTERNET OF THINGS**

**(IoT)**

**Praktik Simulasi Pembuatan Sistem Lampu Lalu Lintas Berbasis IoT**

*Salma Salsabila*

*Fakultas Vokasi Universitas Brawijaya*

*Email:salmasalsabila@student.ub.ac.id*

Abstrak: Praktikum simulasi ini bertujuan untuk merancang dan mensimulasikan sistem lampu lalu lintas berbasis IoT menggunakan Wokwi, sebuah platform simulasi mikrokontroler. Sistem ini dikembangkan menggunakan mikrokontroler ESP32, dan diuji melalui simulasi di Wokwi. Setelah simulasi berjalan dengan baik, kode program dipindahkan ke Visual Studio Code (VS Code) untuk pengujian kompatibilitas serta pengembangan lebih lanjut. Hasil eksperimen menunjukkan bahwa sistem lampu lalu lintas berfungsi sesuai dengan aturan yang telah ditetapkan, dengan pergantian warna merah, kuning, dan hijau dalam interval waktu tertentu. Penggunaan Wokwi memungkinkan proses perancangan dan pengujian dilakukan secara efisien tanpa memerlukan perangkat keras, sedangkan VS Code membantu dalam pengelolaan serta penyempurnaan kode program. Pendekatan ini memungkinkan perancangan sistem lalu lintas berbasis IoT dilakukan dengan lebih fleksibel sebelum diterapkan di dunia nyata.

***Kata kunci****—Internet of Things, Sistem Lampu Lalu Lintas, Wokwi, Mikrokontroler, ESP32*

1. Pendahuluan
   1. Latar Belakang

Perkembangan teknologi Internet of Things (IoT) telah mendorong inovasi di berbagai sektor, termasuk dalam pengelolaan transportasi cerdas. Salah satu penerapan yang berpotensi memberikan dampak signifikan adalah sistem lampu lalu lintas berbasis IoT, yang dirancang untuk meningkatkan efisiensi pengaturan lalu lintas serta mengurangi tingkat kemacetan.

Pada umumnya, pengujian sistem lampu lalu lintas masih bergantung pada perangkat keras fisik, yang membutuhkan biaya dan waktu yang tidak sedikit. Sebagai alternatif yang lebih efisien, penggunaan simulasi berbasis perangkat lunak menjadi solusi dalam proses perancangan dan pengujian awal. Wokwi, sebagai platform simulasi mikrokontroler, memungkinkan pengembang untuk menguji sistem tanpa perlu menggunakan perangkat fisik. Setelah tahap simulasi selesai, kode dapat dipindahkan ke Visual Studio Code (VS Code) untuk pengembangan lebih lanjut dengan PlatformIO, yang mendukung pengelolaan proyek berbasis ESP32 secara lebih optimal.

Dengan menggunakan ESP32 sebagai mikrokontroler utama, sistem ini memiliki potensi untuk dikembangkan lebih lanjut dengan berbagai fitur IoT, seperti kontrol jarak jauh atau sensor kendaraan, guna menciptakan sistem lalu lintas yang lebih responsif dan adaptif. Oleh karena itu, penelitian ini berfokus pada perancangan dan simulasi sistem lampu lalu lintas berbasis ESP32, yang nantinya dapat diterapkan di dunia nyata setelah melalui tahap pengujian yang komprehensif.

* 1. Tujuan Eksperimen

1. Merancang dan mensimulasikan sistem lampu lalu lintas berbasis ESP32 menggunakan platform Wokwi.
2. Menganalisis kinerja sistem dalam mengatur pergantian lampu merah, kuning, dan hijau sesuai dengan interval waktu yang ditentukan.
3. Menguji kompatibilitas kode dengan Visual Studio Code (VS Code) dan PlatformIO untuk pengembangan lebih lanjut.
4. Mengoptimalkan proses pengujian tanpa perangkat keras fisik, sehingga desain sistem dapat diuji dengan lebih efisien sebelum diimplementasikan di dunia nyata.
5. Metodologi
   1. Alat dan Bahan
6. Mikrokontroler: ESP32
7. Komponen Elektronik: LED merah, kuning, hijau; resistor.
8. Software: Wokwi, Visual Studio Code (VS Code), PlatformIO.
   1. Langkah Implementasi
9. Perancangan dan Simulasi di Wokwi
10. Buka Wokwi dan buat proyek baru.
11. Pilih Mikrokontroler dengan menggunakan ESP32 sebagai board utama untuk simulasi.
12. Tambahkan Komponen berupa LED merah, kuning, hijau dan resistor untuk membatasi arus listrik.
13. Hubungkan Komponen dengan menyambungkan setiap LED ke pin GPIO ESP32 menggunakan resistor sebagai pembatas arus.
14. Buatlah kode program dalam bahasa Arduino/C++ untuk mengatur perubahan warna lampu lalu lintas sesuai interval waktu yang telah ditentukan.
15. Jalankan simulasi untuk menguji apakah sistem berfungsi dengan baik.Lakukan Debugging jika terdapat kesalahan dalam kode atau koneksi.
16. Memindahkan Proyek ke Visual Studio Code (VS Code) dengan DOIT ESP32 DevKit V1
    1. Salin Kode dari Wokwi setelah simulasi berhasil. Buka Visual Studio Code dan pastikan PlatformIO telah terinstal.
17. Buat Proyek Baru di PlatformIO dengan memilih DOIT ESP32 DevKit V1 sebagai board dan Arduino Framework sebagai basis pemrograman.
18. Tempelkan Kode Program yang telah disalin dari Wokwi ke dalam file utama proyek.
19. Tambahkan File Konfigurasi berikut untuk memastikan kompatibilitas proyek:

-File platformio.ini untuk mengatur board, platform, dan komunikasi serial.

-File wokwi.toml untuk menghubungkan proyek dengan Wokwi.

-File diagram.json untuk mendokumentasikan koneksi perangkat keras.

1. Periksa Kompatibilitas Kode dengan melakukan pengecekan sintaks dan konfigurasi.
2. Pengujian dan Evaluasi
   1. Jalankan Program di VS Code untuk memastikan kode berjalan dengan baik.
   2. Gunakan Monitor Serial untuk memantau sistem jika terhubung dengan DOIT ESP32 DevKit V1 secara fisik.
   3. Analisis Hasil Simulasi dengan memastikan pergantian lampu lalu lintas sesuai aturan waktu yang telah ditentukan.
3. Hasil dan Pembahasan

Berikut adalah kode utama proyek yang menghasilkan output nyala lampu sesuai dengan timer yang telah ditaur pada kode di bawah ini:

#include <Arduino.h>

#define RED\_LED 23

#define GREEN\_LED 22

#define YELLOW\_LED 21

void setup() {

    Serial.begin(115200);

    Serial.println("ESP32 Traffic Light Simulation");

    pinMode(RED\_LED, OUTPUT);

    pinMode(GREEN\_LED, OUTPUT);

    pinMode(YELLOW\_LED, OUTPUT);

}

void loop() {

    // Lampu Merah menyala selama 30 detik

    digitalWrite(RED\_LED, HIGH);

    digitalWrite(GREEN\_LED, LOW);

    digitalWrite(YELLOW\_LED, LOW);

    Serial.println("Lampu Merah ON");

    delay(30000);

    // Lampu Hijau menyala selama 20 detik

    digitalWrite(RED\_LED, LOW);

    digitalWrite(GREEN\_LED, HIGH);

    digitalWrite(YELLOW\_LED, LOW);

    Serial.println("Lampu Hijau ON");

    delay(20000);

    // Lampu Kuning menyala selama 5 detik

    digitalWrite(RED\_LED, LOW);

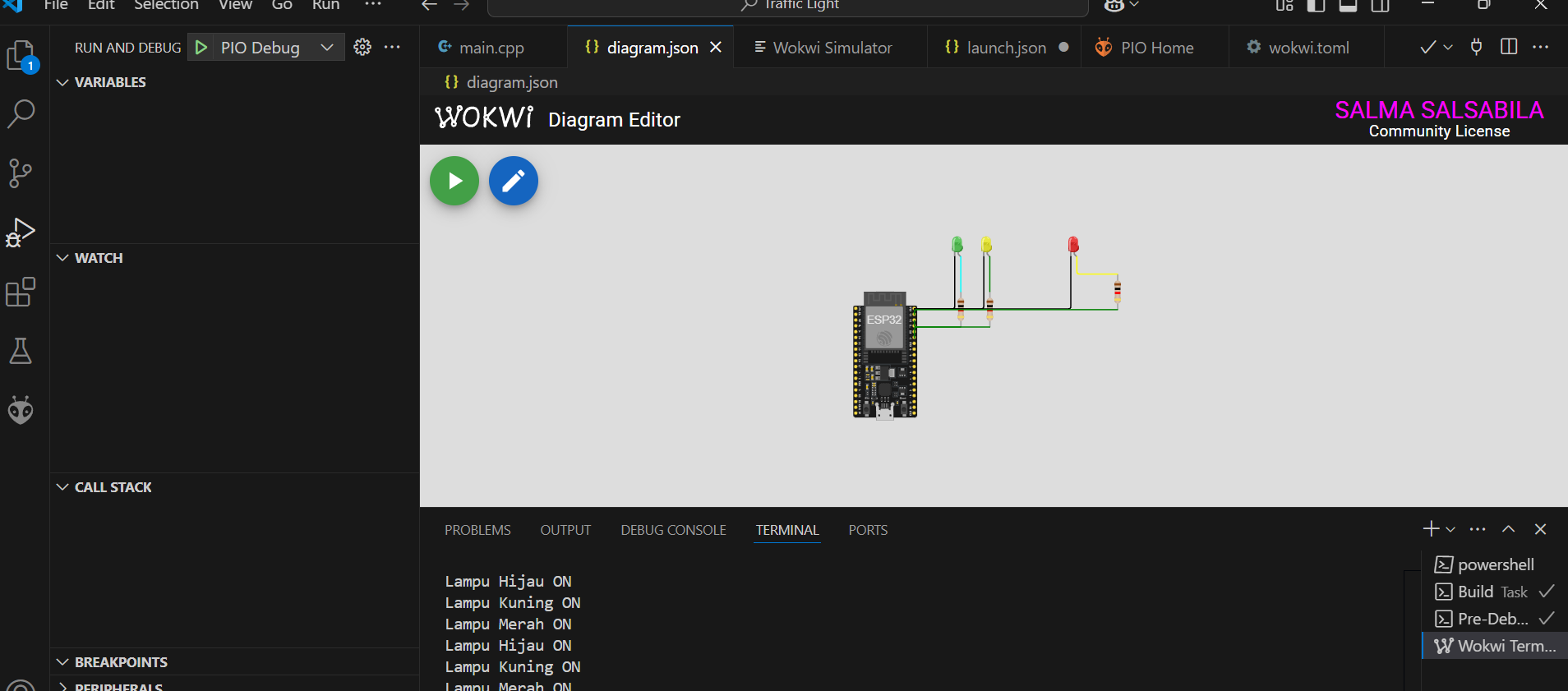
    digitalWrite(GREEN\_LED, LOW);

    digitalWrite(YELLOW\_LED, HIGH);

    Serial.println("Lampu Kuning ON");

   delay(5000);

}



Berdasarkan hasil simulasi di Wokwi dan pengujian di Visual Studio Code (VS Code) menggunakan DOIT ESP32 DevKit V1, sistem lampu lalu lintas berjalan sesuai yang diharapkan.

1. Lampu merah menyala terlebih dahulu selama 30 detik, menunjukkan kendaraan harus berhenti.
2. Lampu hijau menyala selama 20 detik, memberikan kesempatan kendaraan untuk melaju.
3. Lampu kuning menyala selama 5 detik, sebagai tanda peringatan sebelum lampu kembali ke merah.

Pada Wokwi, konfigurasi perangkat keras disimpan dalam file diagram.json, yang berisi informasi mengenai koneksi ESP32 dengan LED merah, kuning, dan hijau. File ini membantu dalam dokumentasi dan memungkinkan integrasi lebih mudah saat berpindah ke VS Code.

Berikut adalah kode program pada diagram sebelum menggunakan format diagram.json yang akan menampilkan gambar.

{

  "version": 1,

  "author": "Anonymous maker",

  "editor": "wokwi",

  "parts": [

    { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 0, "left": -206.36, "attrs": {} },

    {

      "type": "wokwi-led",

      "id": "led1",

      "top": -99.6,

      "left": -53.8,

      "attrs": { "color": "limegreen" }

    },

    {

      "type": "wokwi-led",

      "id": "led2",

      "top": -99.6,

      "left": -5.8,

      "attrs": { "color": "yellow" }

    },

    {

      "type": "wokwi-led",

      "id": "led3",

      "top": -99.6,

      "left": 138.2,

      "attrs": { "color": "red" }

    },

    {

      "type": "wokwi-resistor",

      "id": "r1",

      "top": -4.8,

      "left": 201.05,

      "rotate": 90,

      "attrs": { "value": "1000" }

    },

    {

      "type": "wokwi-resistor",

      "id": "r2",

      "top": 24,

      "left": -58.15,

      "rotate": 90,

      "attrs": { "value": "1000" }

    },

    {

      "type": "wokwi-resistor",

      "id": "r3",

      "top": 24,

      "left": -10.15,

      "rotate": 90,

      "attrs": { "value": "1000" }

    }

  ],

  "connections": [

    [ "esp:TX", "$serialMonitor:RX", "", [] ],

    [ "esp:RX", "$serialMonitor:TX", "", [] ],

    [ "led1:C", "esp:GND.2", "black", [ "v0" ] ],

    [ "led2:C", "esp:GND.2", "black", [ "v86.4", "h-76.4" ] ],

    [ "led3:C", "esp:GND.2", "black", [ "v0" ] ],

    [ "led1:A", "r2:1", "cyan", [ "v0" ] ],

    [ "led3:A", "r1:1", "yellow", [ "v0" ] ],

    [ "led2:A", "r3:1", "green", [ "v0" ] ],

    [ "r1:2", "esp:23", "green", [ "h0" ] ],

    [ "r2:2", "esp:22", "green", [ "h0" ] ],

    [ "r3:2", "esp:21", "green", [ "h0" ] ]

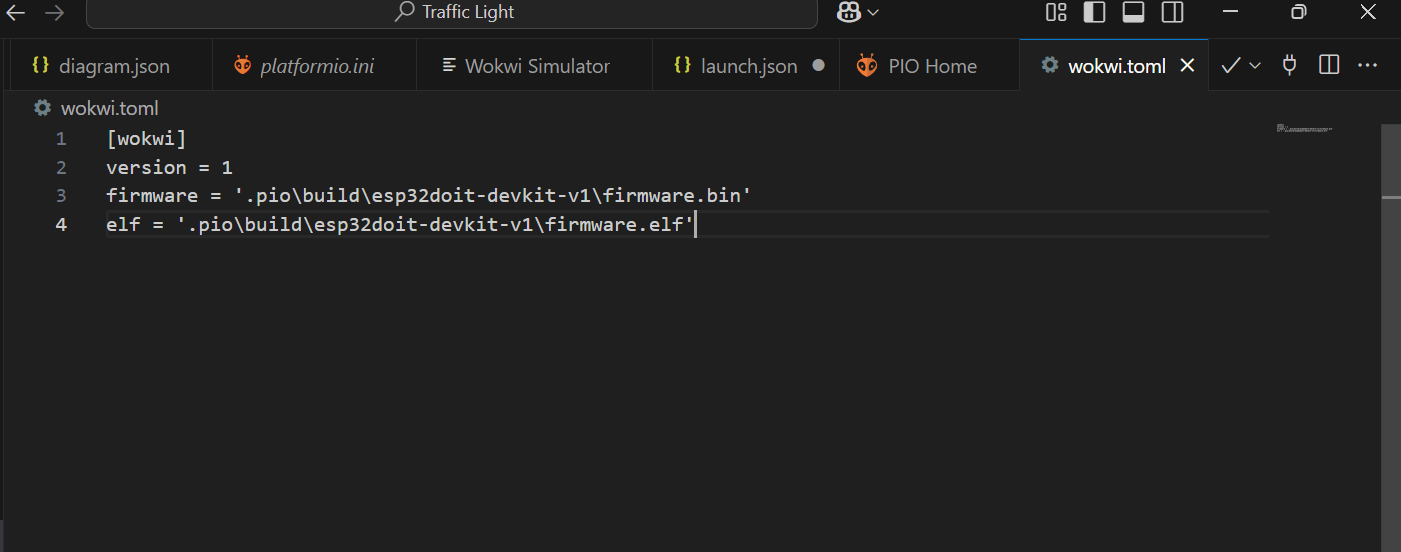
  ],

  "dependencies": {}

}

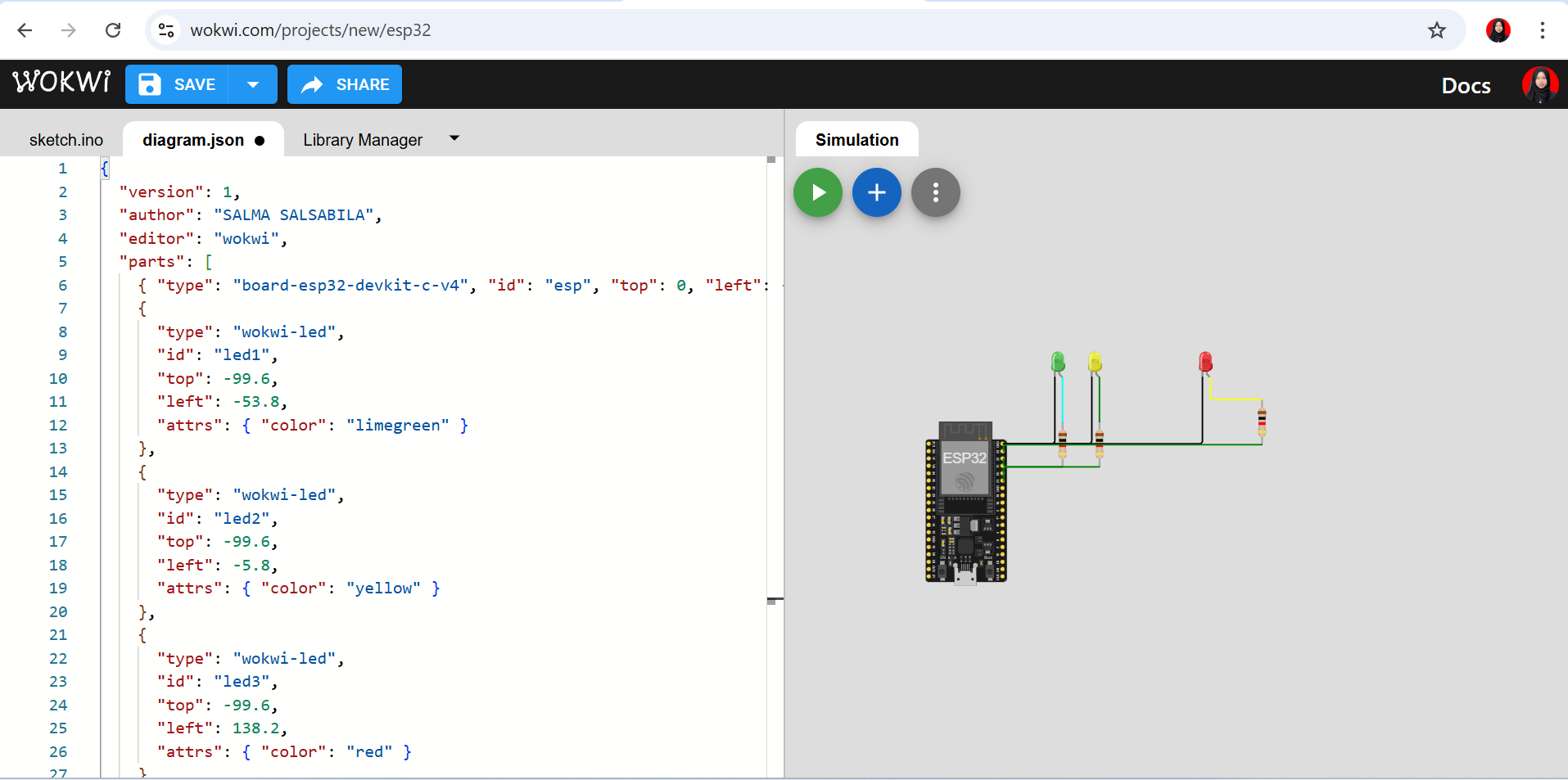
Selain itu, dalam VS Code, digunakan file wokwi.toml untuk menghubungkan proyek dengan Wokwi agar dapat diuji lebih lanjut sebelum diunggah ke perangkat fisik.

Berikut adalah tampilan dari wokwi.toml pada vscode dimana berfungsi sebagai penghubung proyek dengan wokwi.



1. Lampiran

Tampilan json pada website wokwi



Tampilan json pada vscode

