

PRAKTIK PEMBUATAN SISTEM TRAFFIC LIGHT MENGUNAKAN ESP32

Ray Nafi Salsabila Arindivah

Fakultas Vokasi, Universitas Brawijaya

Email: salsarindivah@student.ub.ac.id

ABSTRACT

Lampu lalu lintas adalah perangkat elektronik yang mengendalikan arus lalu lintas kendaraan dan pejalan kaki menggunakan serangkaian lampu berwarna. Lampu lalu lintas yang paling umum terdiri dari tiga lampu: merah, kuning, dan hijau. Lampu merah menunjukkan bahwa kendaraan dan pejalan kaki harus berhenti, lampu kuning menunjukkan bahwa lampu merah akan segera menyala, dan lampu hijau menunjukkan bahwa kendaraan dan pejalan kaki dapat melanjutkan perjalanan.

Sistem lampu lalu lintas dengan mikrokontroler ESP32 akan melibatkan penyambungan ESP32 ke serangkaian LED (merah, kuning, dan hijau) untuk mensimulasikan lampu lalu lintas. Dalam proyek ini kita akan membuat lampu lalu lintas yang dikontrol oleh kartu ESP32 di mana lampu LED merah menyala selama 30 detik, kemudian padam dan lampu LED hijau menyala selama 20 detik dan kemudian padam sehingga lampu LED kuning menyala selama 5 detik. Kemudian program akan dimulai lagi dan dimulai lagi. Kesimpulan dari penelitian ini adalah bahwa penggunaan ESP32 dalam sistem traffic light memungkinkan otomatisasi yang lebih cerdas dan fleksibel.

Kata Kunci- ESP32, LED, Traffic Light

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kemacetan lalu lintas jadi masalah besar di banyak kota, terutama di persimpangan jalan yang padat. Sistem traffic light konvensional umumnya menggunakan timer tetap, sehingga tidak bisa menyesuaikan kondisi lalu lintas secara real-time. Akibatnya, ada waktu tunggu yang tidak efisien—misalnya, lampu merah tetap menyala lama meskipun jalan sedang sepi.

Dengan perkembangan teknologi, sistem lalu lintas pintar mulai dikembangkan untuk mengatasi masalah ini. Salah satu solusi yang bisa diterapkan adalah menggunakan ESP32, mikrokontroler yang mendukung konektivitas Wi-Fi dan memiliki kemampuan pemrosesan yang cukup baik. Dengan bantuan sensor, sistem ini bisa mendeteksi kepadatan kendaraan dan mengatur durasi lampu lalu lintas secara dinamis, sehingga arus kendaraan bisa lebih lancar.

Penelitian ini bertujuan untuk merancang dan menguji sistem traffic light berbasis ESP32 yang mampu menyesuaikan waktu nyala lampu berdasarkan kondisi lalu lintas. Selain itu, sistem ini juga bisa dikendalikan secara nirkabel, sehingga memudahkan pemantauan dan pengelolaan dari jarak jauh. Dengan penerapan teknologi ini, diharapkan efisiensi lalu lintas bisa meningkat dan waktu tunggu pengendara bisa dikurangi.

1.2 Tujuan Eksperimen

Eksperimen ini bertujuan untuk merancang dan mengembangkan sistem traffic light berbasis **ESP32** yang bisa menyesuaikan waktu nyala lampu berdasarkan kepadatan kendaraan. Dengan bantuan sensor, sistem ini diharapkan mampu mengoptimalkan aliran lalu lintas, mengurangi waktu tunggu yang tidak perlu, dan meningkatkan efisiensi di persimpangan jalan. Selain itu, eksperimen ini juga bertujuan untuk menguji konektivitas dan kontrol sistem secara nirkabel, sehingga bisa dipantau atau diatur dari jarak jauh. Hasil dari eksperimen ini diharapkan dapat menjadi langkah awal menuju sistem lalu lintas yang lebih cerdas dan adaptif.

BAB II

METODOLOGI

2.1 Alat dan Bahan

- Visual Studio
- Wokwi Simulator
- ESP32 – Mikrokontroler utama untuk mengontrol sistem traffic light.
- LED Merah, Kuning, Hijau – Sebagai indikator lampu lalu lintas.
- Resistor – Untuk membatasi arus ke LED.
- Kabel Jumper – Menghubungkan komponen elektronik pada rangkaian.
- Adaptor atau Power Bank (5V/9V) – Sebagai sumber daya untuk ESP32.

2.2 Langkah Implementasi

1. Perancangan Sistem

- Menentukan skema rangkaian dengan ESP32 sebagai pusat kendali.
- Susun komponen seperti LED dan resistor.
- Pastikan koneksi kabel jumper sesuai dengan kebutuhan sistem.

2. Pengkodean Program

- Gunakan Arduino IDE untuk menulis kode ESP32.
- Membuat program untuk:
 - a. Mengontrol nyala LED sesuai urutan lampu lalu lintas.
 - b. Mengatur durasi lampu merah (30 detik), kuning (5 detik) dan hijau (20 detik).
- Unggah kode ke ESP32 dan lakukan debugging jika ada kesalahan.

3. Pengujian Sistem

- Uji apakah lampu lalu lintas menyala sesuai urutan yang diinginkan.
- Pastikan durasi lampu sesuai dengan yang sudah ditentukan.
- Lakukan penyempurnaan jika ada kendala atau kesalahan dalam sistem.

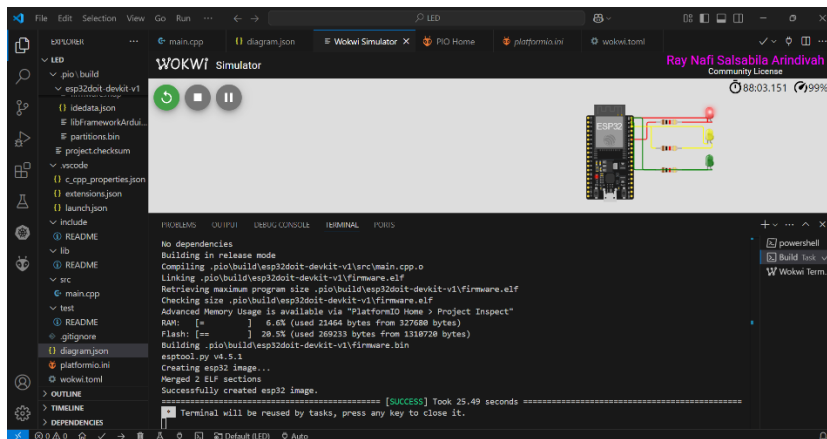
BAB III

Hasil dan Pembahasan

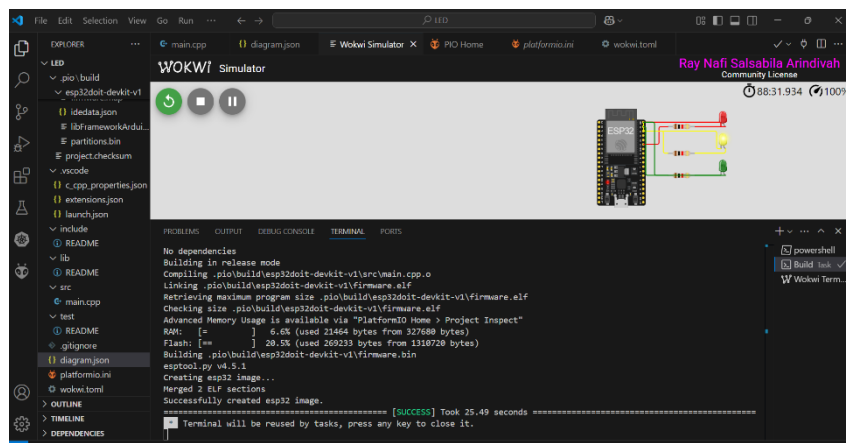
3.1 Hasil Eksperimen

No.	Aspek Pengujian	Hasil	Keterangan
1.	Urutan Lampu	Berhasil	Lampu menyala sesuai urutan (Merah → Kuning → Hijau) tanpa kesalahan.
2.	Penyesuaian Durasi Lampu	Berhasil	Lampu menyala sesuai dengan durasi yang sudah ditentukan tanpa kesalahan (Merah 30 detik, Kuning 5 detik, dan Hijau 20 detik).

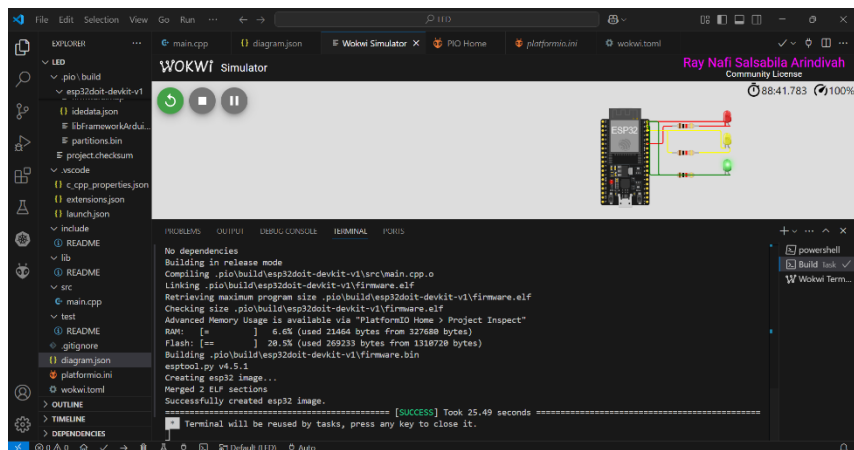
- Screenshoot ESP32 saat lampu merah nyala selama 30 detik



- Screenshoot ESP32 saat lampu kuning nyala selama 5 detik

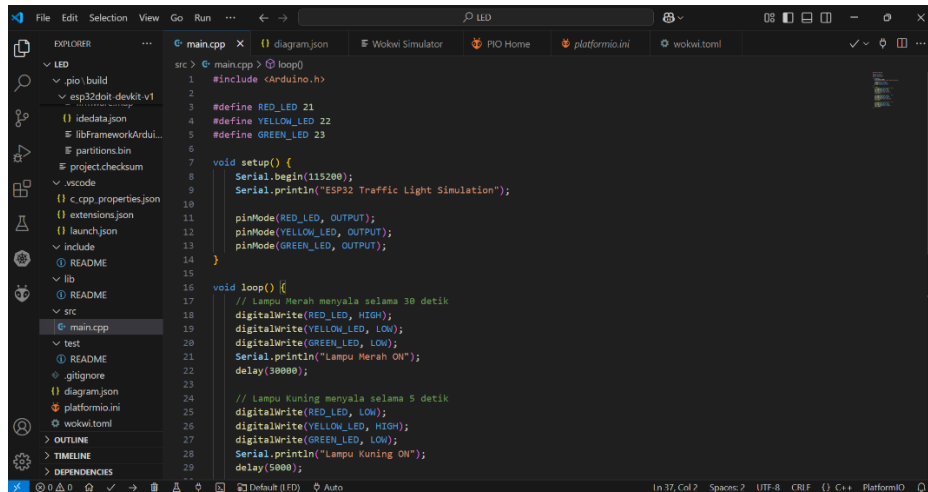


- Screenshoot ESP32 saat lampu hijau nyala selama 20 detik

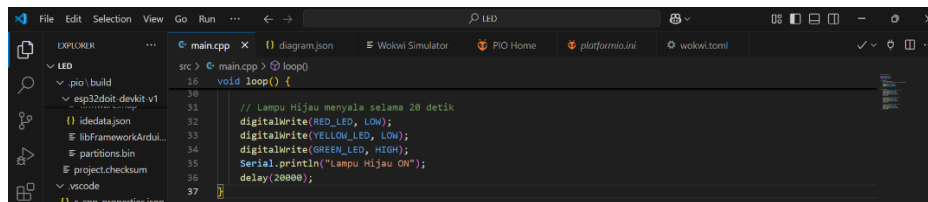


BAB IV

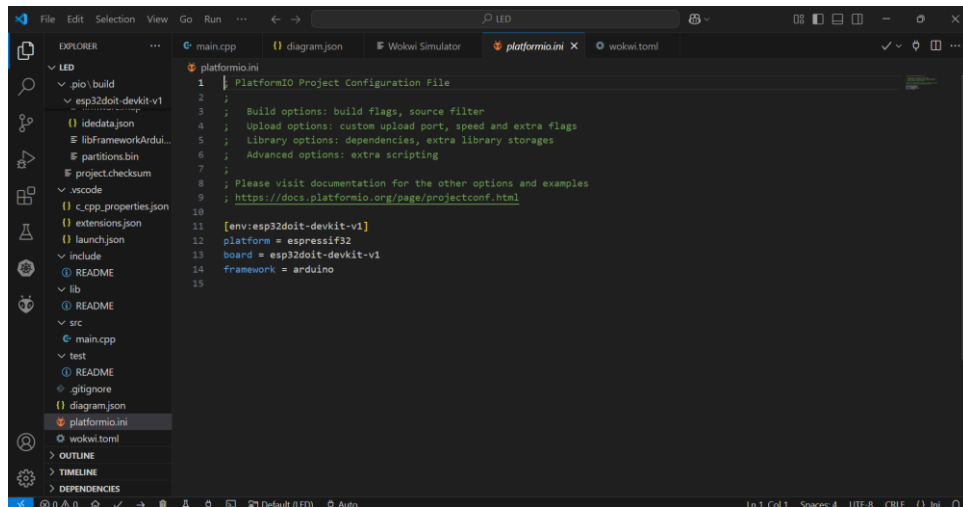
LAMPIRAN



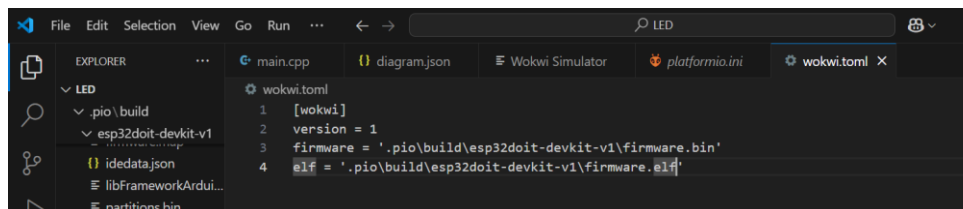
```
1 #include <Arduino.h>
2
3 #define RED_LED 21
4 #define YELLOW_LED 22
5 #define GREEN_LED 23
6
7
8 void setup() {
9     Serial.begin(115200);
10    Serial.println("ESP32 Traffic Light Simulation");
11
12    pinMode(RED_LED, OUTPUT);
13    pinMode(YELLOW_LED, OUTPUT);
14    pinMode(GREEN_LED, OUTPUT);
15 }
16
17 void loop() {
18     // Lampu Merah menyala selama 30 detik
19     digitalWrite(RED_LED, HIGH);
20     digitalWrite(YELLOW_LED, LOW);
21     digitalWrite(GREEN_LED, LOW);
22     Serial.println("Lampu Merah ON");
23     delay(30000);
24
25     // Lampu Kuning menyala selama 5 detik
26     digitalWrite(RED_LED, LOW);
27     digitalWrite(YELLOW_LED, HIGH);
28     digitalWrite(GREEN_LED, LOW);
29     Serial.println("Lampu Kuning ON");
30     delay(5000);
31 }
```



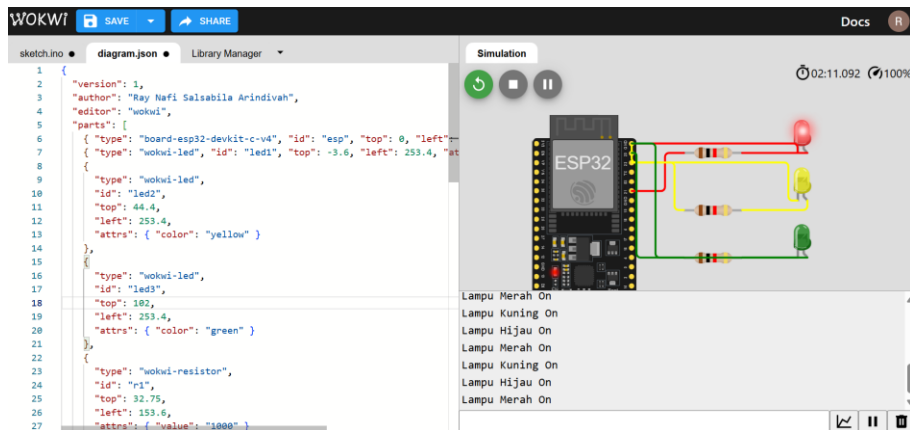
```
16 void loop() {
17
18     // Lampu Hijau menyala selama 20 detik
19     digitalWrite(RED_LED, LOW);
20     digitalWrite(YELLOW_LED, LOW);
21     digitalWrite(GREEN_LED, HIGH);
22     Serial.println("Lampu Hijau ON");
23     delay(20000);
24 }
```



```
1 PlatformIO Project Configuration File
2
3 ; Build options: build flags, source filter
4 ; Upload options: custom upload port, speed and extra flags
5 ; Library options: dependencies, extra library storages
6 ; Advanced options: extra scripting
7
8 ; Please visit documentation for the other options and examples
9 ; https://docs.platformio.org/page/projectconf.html
10
11 [env:esp32doit-devkit-v1]
12 platform = espressif32
13 board = esp32doit-devkit-v1
14 framework = arduino
```



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



```

WOKWI SAVE SHARE
sketch.ino diagram.json Library Manager
1
2 {
3   "version": 1,
4   "author": "Ray Nafi Salsabila Arindivah",
5   "editor": "wokwi",
6   "parts": [
7     { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 0, "left": 0, "attrs": {} },
8     { "type": "wokwi-led", "id": "led1", "top": -3.6, "left": 253.4, "attrs": { "color": "red" } },
9     { "type": "wokwi-led",
10      "id": "led2",
11      "top": 44.4,
12      "left": 253.4,
13      "attrs": { "color": "yellow" } },
14     { "type": "wokwi-led",
15      "id": "led3",
16      "top": 102,
17      "left": 253.4,
18      "attrs": { "color": "green" } },
19     { "type": "wokwi-resistor",
20      "id": "r1",
21      "top": 32.75,
22      "left": 153.6,
23      "attrs": { "value": "1000" } },
24   ],
25   "connections": [
26     [ "esp:TX", "$serialMonitor:RX", "", [ ] ],
27     [ "esp:RX", "$serialMonitor:TX", "", [ ] ],
28     [ "led2:C", "esp:GND.2", "yellow", [ "h-9.2", "v-38.4", "h-158.44" ] ],
29     [ "led1:C", "esp:GND.2", "red", [ "v0" ] ],
30     [ "r1:2", "led1:A", "red", [ "v0" ] ],
31     [ "r1:1", "esp:21", "red", [ "h-19.2", "v38.4" ] ],
32     [ "r2:1", "esp:22", "yellow", [ "h-9.6", "v-48" ] ],
33     [ "r3:1", "esp:23", "green", [ "h-48", "v-144" ] ],
34     [ "r2:2", "led2:A", "yellow", [ "h56.4", "v0", "h28.8" ] ],
35     [ "r3:2", "led3:A", "green", [ "v0" ] ],
36     [ "led3:C", "esp:GND.2", "green", [ "h-143.6", "v-115.2" ] ],
37   ],
38   "dependencies": {}
39 }

```

BAB V

PENUTUP

5.1 Kesimpulan

Eksperimen ini menunjukkan bahwa sistem traffic light berbasis ESP32 bisa bekerja otomatis dan menyesuaikan durasi lampu sesuai dengan waktu yang diatur. Sistem lampu lalu lintas yang dibuat menggunakan ESP32 dapat mengatasi kemacetan lalu lintas. Sistem ini dapat mengatur lampu lalu lintas secara dinamis. Manfaat sistem lampu lalu lintas menggunakan ESP32 Meningkatkan efisiensi lalu lintas, Mengurangi emisi, Memberikan pengalaman berkendara yang lebih baik.