

SIMULASI *TRAFFIC LIGHT* BERBASIS MIKROKONTROLER ESP32

Feby Kurnia Putri

*Program Studi Teknologi Informasi, Fakultas Vokasi, Universitas Brawijaya Jl. Veteran No.10-11,
Ketawanggede, Kec. Lowokwaru, Kota Malang, Jawa Timur 65145*

e-mail : febystudentub@gmail.com

Abstrack -- The development of science and technology is getting faster which can be seen from various kinds of tools that are made automatically and regularly. With this technological advancement, it can also be used as a benchmark for the level of progress of a nation to create a tool. The number of tools that The number of tools created with various applications that aim to save energy, time, and money for users or the public. This report is entitled "Simulation of Microcontroller-Based Traffic Light". As for The function of this tool is to break down congestion or density of traffic vehicles at road intersections based on traffic density. In the design of this tool, several circuits are used, namely a microcontroller circuit. This experiment must have an implementation of the developed system before a final system is made.

Keyword : Simulation of Microcontroller-Based Traffic Light Settings.

*Abstrak -- Perkembangan ilmu pengetahuan dan teknologi makin pesat saja dimana dapat terlihat dari berbagai macam alat yang dibuat secara otomatis dan teratur. Dengan kemajuan teknologi ini pula dapat dijadikan salah satu tolak ukur tingkat kemajuan suatu bangsa untuk menciptakan sebuah alat. Banyaknya alat yang diciptakan dengan berbagai aplikasi yang bertujuan dapat menghemat tenaga, waktu, dan uang bagi penggunaanya atau khalayak. Laporan ini berjudul " Simulasi *Traffic Light* Berbasis Mikrokontroler ESP32". Adapun fungsi dari alat ini adalah untuk mengurai kemacetan atau kepadatan kendaraan lalu lintas pada persimpangan jalan berdasarkan kepadatan lalu lintas. Didalam Perancangan alat ini, digunakan beberapa rangkaian yaitu rangkaian mikrokontroler. Percobaan ini harus ada implementasi sistem yang dikembangkan sebelum dibuat sebuah sistem final.*

Kata Kunci : : Simulasi Pengaturan Lampu Lalu Lintas (Traffic Light) Berbasis Mikrokontroler.

I. PENDAHULUAN

Sekarang ini, Perkembangan Teknologi di bidang transportasi semakin pesat. Hal ini menyebabkan banyaknya kendaraan yang digunakan untuk mempermudah sebagian besar orang. Hal ini pun kebutuhan kendaraan pribadi ikut meningkat. Kondisi ini pun semakin menyebabkan kepadatan terutama di daerah perkotaan. Demi mencegah padatnya kendaraan tersebut dan mengurangi tingkat kecelakaan di jalan raya dibutuhkan lah *traffic light* atau biasa disebut lampu lalu lintas. *Traffic Light* adalah lampu yang mengendalikan arus lalu lintas yang terpasang pada persimpangan jalan, seperti tempat penyeberangan untuk penjalan kaki (zebra cross), dan tempat arus lalu lintas lainnya. Lampu ini lah yang menandakan kapan suatu kendaraan harus berjalan dan berhenti secara bergantian dari berbagai arah. Saat ini, pengaturan system lampu lalu lintas berbasis mikrokontroler dapat digunakan sebagai simulasi untuk mengatur penyalan lampu. Sistem ini disebut sebagai

actuared controller. Tujuan simulasi kali ini untuk mengembangkan *Traffic Light*).

II. METODOLOGI

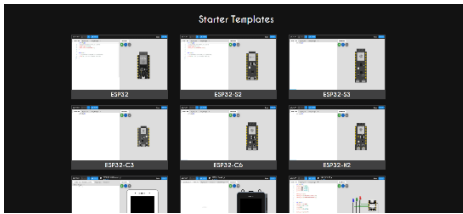
A. ALAT & BAHAN

Mikrokontroler (ESP32), Resistor, Software (Visual Studio Code, PlatformIO IDE, Arduino IDE).

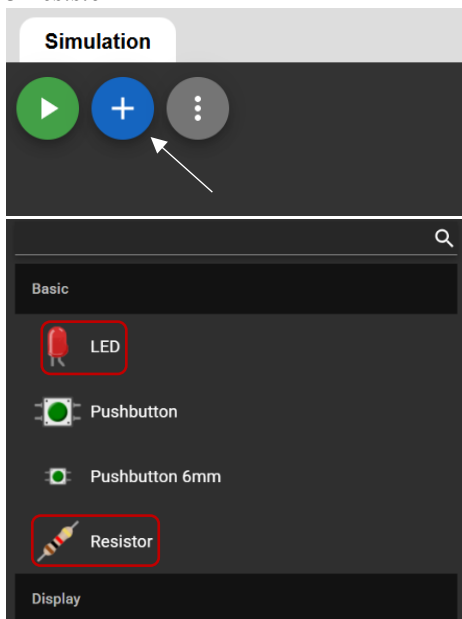
B. LANGKAH IMPLEMENTASI

1. Buka <https://wokwi.com/> pilih ESP32, kemudian pilih *Starter Templates*.

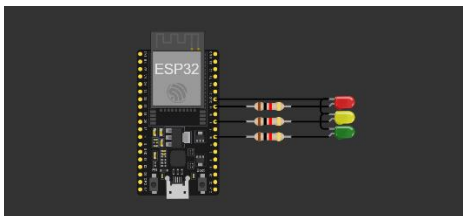




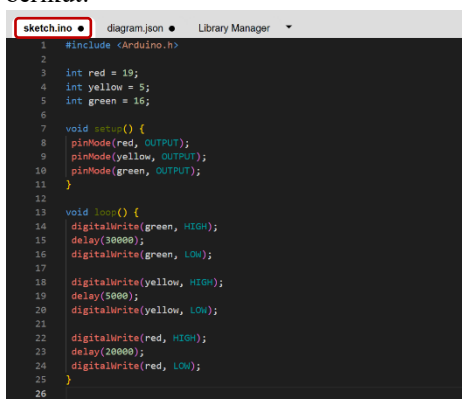
2. Add simulasi untuk memulai rangkaian simulasi Lampu Lalu Lintas. Add 3 LED dan 3 Resistor.



3. Atur rangkain Lampu Lalu Lintas sesuai angka yang disambungkan dengan code nanti.



4. Lakukan pengkodean seperti gambar berikut:



```
#include <Arduino.h>
```

```
int red = 19;
int yellow = 5;
int green = 16;
```

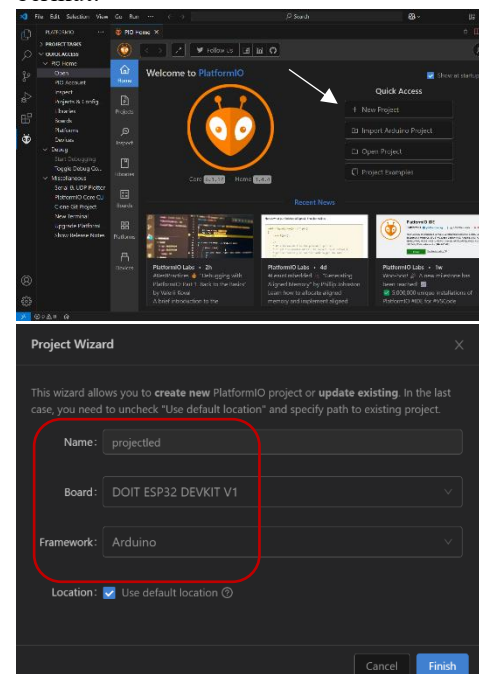
```
void setup() {
  pinMode(red, OUTPUT);
  pinMode(yellow, OUTPUT);
  pinMode(green, OUTPUT);
}
```

```
void loop() {
  digitalWrite(green, HIGH);
  delay(30000);
  digitalWrite(green, LOW);
```

```
  digitalWrite(yellow, HIGH);
  delay(5000);
  digitalWrite(yellow, LOW);
```

```
  digitalWrite(red, HIGH);
  delay(20000);
  digitalWrite(red, LOW);
}
```

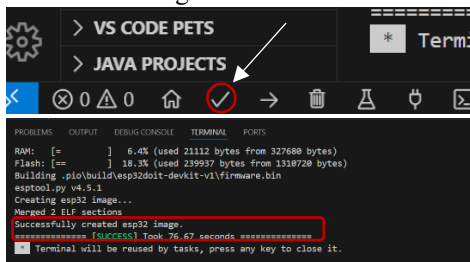
4. Buka VSCode, kemudian install PlatformIO IDE. Buat proyek baru seperti gambar berikut:



5. Setelah itu kembali ke Wokwi untuk meng copy code sketch.io, lalu paste code pada Visual Studio Code pada bagian src-main.cpp.

```
main.cpp X
1 #include <Arduino.h>
2
3 int red = 10;
4 int yellow = 5;
5 int green = 16;
6
7 void setup() {
8   pinMode(red, OUTPUT);
9   pinMode(yellow, OUTPUT);
10  pinMode(green, OUTPUT);
11 }
12
13 void loop() {
14   digitalWrite(green, HIGH);
15   delay(1000);
16   digitalWrite(green, LOW);
17
18   digitalWrite(yellow, HIGH);
19   delay(1000);
20   digitalWrite(yellow, LOW);
21
22   digitalWrite(red, HIGH);
23   delay(1000);
24   digitalWrite(red, LOW);
25 }
26
```

- Jika sudah, lakukan build Wokwi dengan cara klik centang.



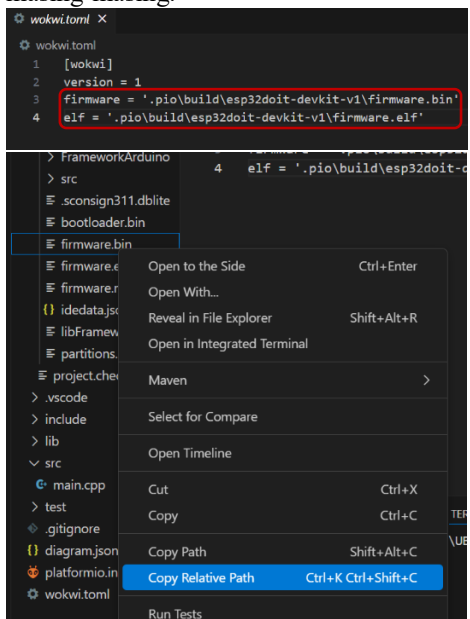
- Buat file baru pada Visual Studio Code, kemudian buka wokwi.toml pada google, kemudian copy ke file baru yang diberi nama wokwi.toml di VScode.

wokwi.toml

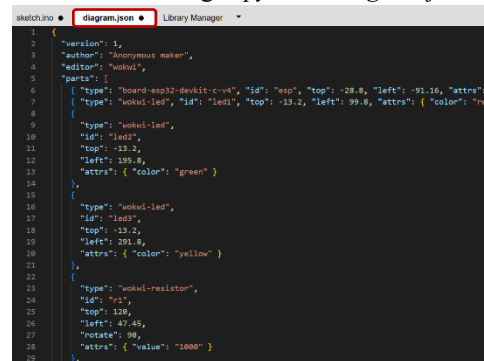
A basic wokwi.toml file looks like this:

```
[wokwi]
version = 1
firmware = 'path-to-your-firmware.hex'
elf = 'path-to-your-firmware.elf'
```

- Setelah itu ganti menjadi firmware bin, dan esp32 firmware elf pada Visual Studio Code masing-masing.



- Buat file baru bernama diagram.json pada Visual Studio Code. Kemudian pergi ke wokwi untuk mengcopy code diagram.json.



```
{
  "version": 1,
  "author": "Anonymous maker",
  "editor": "wokwi",
  "parts": [
    {
      "type": "board-esp32-devkit-c-v4", "id": "esp", "top": -28.8, "left": -91.16, "attrs": {
        "type": "wokwi-led", "id": "led1", "top": -13.2, "left": 99.8, "attrs": { "color": "red" }
      }
    },
    {
      "type": "wokwi-led",
      "id": "led2",
      "top": -13.2,
      "left": 155.8,
      "attrs": { "color": "green" }
    },
    {
      "type": "wokwi-led",
      "id": "led3",
      "top": 120,
      "left": 191.8,
      "attrs": { "color": "yellow" }
    },
    {
      "type": "wokwi-resistor",
      "id": "r1",
      "top": 120,
      "left": 47.45,
      "rotate": 90,
      "attrs": { "value": "1000" }
    },
    {
      "type": "wokwi-led",
      "id": "led1",
      "top": 28,
      "left": 150.6,
      "rotate": 90,
      "attrs": { "color": "red", "flip": "1" }
    },
    {
      "type": "wokwi-led",
      "id": "led2",
      "top": 85.6,
      "left": 150.6,
      "rotate": 90,
      "attrs": { "color": "green", "flip": "" }
    },
    {
      "type": "wokwi-led",
      "id": "led3",
      "top": 56.8,
      "left": 150.6,
      "rotate": 90,
      "attrs": { "color": "yellow", "flip": "1" }
    },
    {
      "type": "wokwi-resistor",
      "id": "r1",
      "top": 71.15,
      "left": 48,
      "attrs": { "value": "1000" }
    },
    {
      "type": "wokwi-resistor",
      "id": "r2",

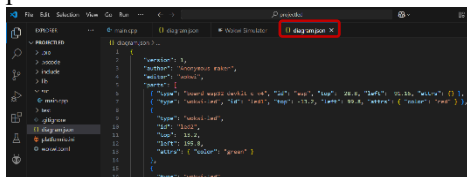
```

```

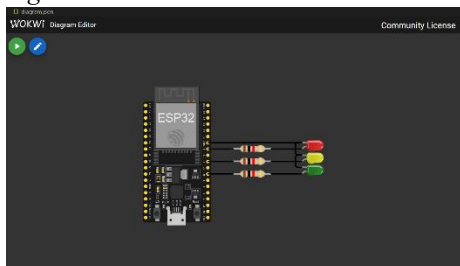
"top": 80.75,
"left": 48,
"attrs": { "value": "1000" }
},
{
"type": "wokwi-resistor",
"id": "r3",
"top": 61.55,
"left": 48,
"attrs": { "value": "1000" }
}
],
"connections": [
[ "esp:TX", "$serialMonitor:RX", "", [] ],
[ "esp:RX", "$serialMonitor:TX", "", [] ],
[ "esp:18", "r1:1", "gold", [ "h0" ] ],
[ "esp:19", "r3:1", "red", [ "h0" ] ],
[ "esp:5", "r2:1", "green", [ "h0" ] ],
[ "r3:2", "led1:A", "red", [ "v0" ] ],
[ "r1:2", "led3:A", "gold", [ "v9.6",
"h46.8" ] ],
[ "r2:2", "led2:A", "green", [ "v0" ] ],
[ "esp:GND.3", "led1:C", "black", [ "h0"
] ],
[ "led1:C", "led3:C", "black", [ "h-38.4",
"v28.4" ] ],
[ "led3:C", "led2:C", "black", [ "h-38.4",
"v28.4" ] ]
],
"dependencies": {}
}

```

10. Paste ke Visual Studio Code, pada file *diagram.json*. Jika *diagram* tidak berfungsi, maka lakukan edit pada *diagram.json*, kemudian *paste code* yang sudah di *copy* pada *wokwi*.

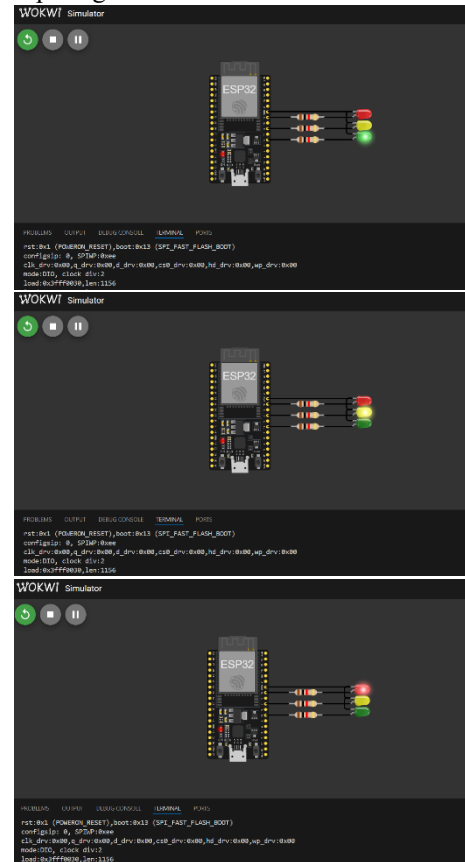


11. Kemudian *wokwi* akan menampilkan hasil dari *diagram.json* pada simulasi *Traffic Light* di *Visual Studio Code*.



III. HASIL DAN PEMBAHASAN

Traffic Light dengan 3 LED Hijau (30 detik), Kuning (5 detik) , dan Merah (20 detik) yang menyala secara bergantian, seperti gambar berikut.



IV. KESIMPULAN

Setelah melakukan perancangan dan pengujian terhadap alat secara keseluruhan. Maka dapat diambil kesimpulan :

1. Perangkat yang digunakan oleh penulis dapat bekerja dengan baik sesuai dengan yang diharapkan.
2. Mikrokontroler ESP32 yang digunakan sebagai pengendali utama dapat bekerja dalam menjalankan perintah yang diberikan

V. DAFTAR PUSTAKA

Muhammad Husni Rifqol, Heni Aprianti2. Januari (2020). Sistem Respon Lalu Lintas Terhadap Pelanggaran Pengendara Menggunakan Ultrasonik. *JSAT* . Volume 3 Nomor 1. 2614-3062; E-ISSN: 2614-3054.

<https://media.neliti.com/media/publication/s/439068-none-037851e2.pdf>