# LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**PRAKTIK PEMBUATAN SIMULASI LAMPU LALU LINTAS**

*BAHYDZAKY FARREL JOCELLIN*

*Fakultas Vokasi, Universitas Brawijaya*

*Email: [fjocellin9@gmail.com](mailto:fjocellin9@gmail.com)*

**Abstrak**

Pada praktikum ini, dilakukan simulasi sistem lampu lalu lintas menggunakan mikrokontroler ESP32. Tujuan dari praktikum ini adalah memahami cara kerja mikrokontroler dalam mengendalikan lampu lalu lintas secara otomatis berdasarkan waktu yang telah ditentukan. ESP32 dipilih karena kemampuannya dalam pemrosesan yang cepat serta dukungan konektivitas yang luas. Dalam implementasinya, digunakan LED sebagai representasi lampu merah, kuning, dan hijau, serta kode pemrograman berbasis Arduino IDE untuk mengatur pola nyala lampu sesuai dengan aturan lalu lintas standar. Hasil dari praktikum menunjukkan bahwa sistem dapat beroperasi sesuai dengan logika yang diharapkan, yaitu mengatur pergantian lampu dalam interval waktu tertentu. Dengan demikian, simulasi ini dapat menjadi dasar dalam pengembangan sistem lalu lintas cerdas yang lebih kompleks di masa depan.

**Kata kunci**: ESP32, lampu lalu lintas, simulasi, mikrokontroler, Arduino IDE.

**Abstraction**

In this practicum, a traffic light system simulation was conducted using the ESP32 microcontroller. The objective of this practicum was to understand how a microcontroller controls traffic lights automatically based on predefined time intervals. The ESP32 was chosen due to its fast processing capabilities and extensive connectivity support. In the implementation, LEDs were used to represent red, yellow, and green lights, while the programming was done using Arduino IDE to regulate the light switching patterns according to standard traffic rules. The results of the practicum showed that the system operated as expected, managing the light transitions within specified time intervals. Thus, this simulation can serve as a foundation for developing more advanced intelligent traffic systems in the future.

**Keywords**: ESP32, traffic light, simulation, microcontroller, Arduino IDE.

**1. Pendahuluan**

* 1. **Latar belakang**

Lampu lalu lintas merupakan sistem yang berperan penting dalam mengatur arus kendaraan di jalan raya untuk mencegah kemacetan dan kecelakaan. Seiring dengan perkembangan teknologi, penggunaan mikrokontroler dalam sistem otomatisasi lalu lintas semakin meningkat. ESP32 merupakan salah satu mikrokontroler yang banyak digunakan dalam proyek elektronika dan IoT karena memiliki prosesor yang cepat serta dukungan konektivitas yang luas. Oleh karena itu, dalam praktikum ini dilakukan simulasi lampu lalu lintas menggunakan ESP32 untuk memahami cara kerja dan pengendalian sistem lampu lalu lintas berbasis mikrokontroler.

**1.2 Tujuan eksperimen**

Praktikum ini juga bertujuan untuk mempelajari penggunaan ESP32 sebagai perangkat utama dalam mengendalikan lampu lalu lintas. ESP32 dipilih karena memiliki kemampuan pemrosesan yang cepat serta dukungan berbagai fitur, termasuk konektivitas nirkabel yang memungkinkan pengembangan lebih lanjut ke sistem yang lebih kompleks.

Praktikum ini mengimplementasikan simulasi lampu lalu lintas yang berfungsi sesuai dengan aturan standar. Simulasi ini menggunakan LED sebagai representasi lampu merah, kuning, dan hijau, serta memanfaatkan kode pemrograman berbasis Arduino IDE untuk mengontrol pola nyala lampu. Dengan demikian, peserta dapat mengamati secara langsung bagaimana sistem ini bekerja dan bagaimana konsep yang diterapkan dapat dikembangkan lebih lanjut dalam sistem lalu lintas cerdas di masa depan.

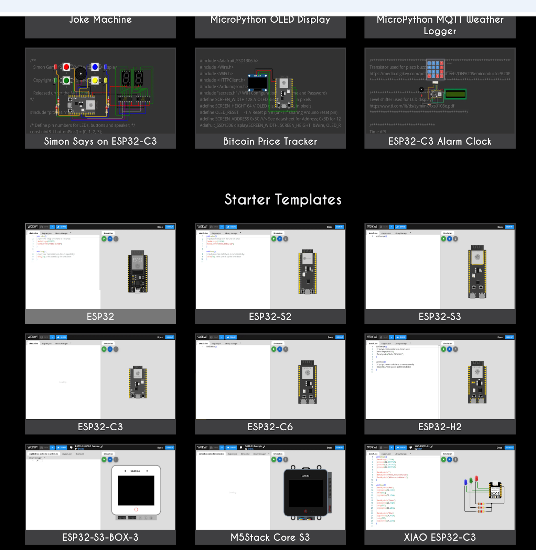
**2. Metodologi**

**2.1 Alat dan Bahan**

1. ESP32
2. Visual Studio Code
3. Laptop
4. Wokwi
   1. **Implementation Steps (Langkah Implementasi)**
5. Buka website <https://wokwi.com/>, lalu login atau daftar jika belum mempunyai akun.

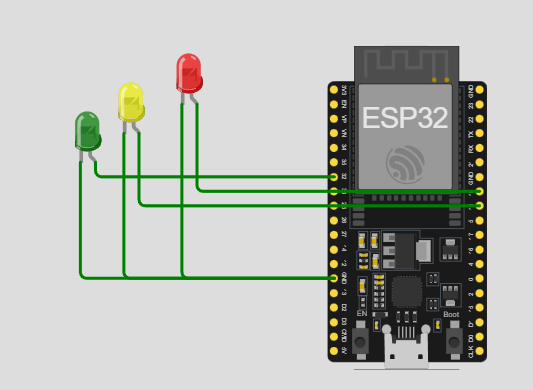
|  |  |
| --- | --- |
| Gambar 2.1 | Gambar 2.2 |

1. Pilih menu adruino dan pilih starter template esp32.



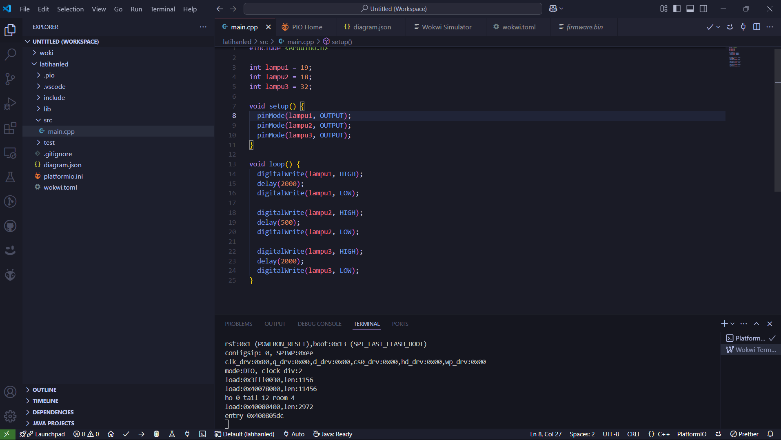
Gambar 2.3

1. Lalu tambahkan komponen seperti lampu dan yang lain



Gambar 2.4

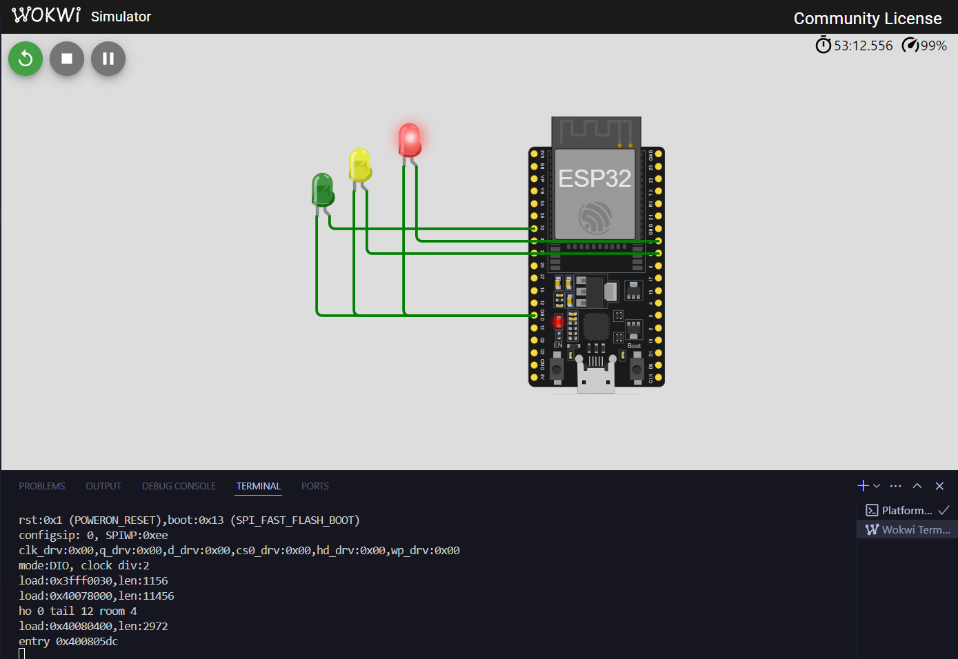
1. Lalu project dengan platform io kemudian buat file diagram.json dan wokwi.toml untuk konfigurasi wokmi dengan visual studio code.



Gambar 2.4

**3. Results and Discussion (Hasil dan Pembahasan)**

**3.1 Experimental Results (Hasil Eksperimen)**



**4. Appendix (Lampiran, jika diperlukan)**

Kode Program untuk diagram.json

{

  "version": 1,

  "author": "Anonymous maker",

  "editor": "wokwi",

  "parts": [

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

    {

      "type": "wokwi-led",

      "id": "led1",

      "top": -3.6,

      "left": -111.4,

      "attrs": { "color": "red" }

    },

    {

      "type": "wokwi-led",

      "id": "led2",

      "top": 15.6,

      "left": -149.8,

      "attrs": { "color": "yellow" }

    },

    {

      "type": "wokwi-led",

      "id": "led3",

      "top": 34.8,

      "left": -178.6,

      "attrs": { "color": "green" }

    }

  ],

  "connections": [

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

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

    [ "led1:C", "esp:GND.1", "green", [ "v0" ] ],

    [ "led2:C", "esp:GND.1", "green", [ "v0" ] ],

    [ "led3:A", "esp:32", "green", [ "v0" ] ],

    [ "led3:C", "esp:GND.1", "green", [ "v0" ] ],

    [ "led1:A", "esp:19", "green", [ "v0" ] ],

    [ "led2:A", "esp:18", "green", [ "v0" ] ]

  ],

  "dependencies": {}

}

Kode Program untuk main.cpp

#include <Arduino.h>

int lampu1 = 19;

int lampu2 = 18;

int lampu3 = 32;

void setup() {

  pinMode(lampu1, OUTPUT);

  pinMode(lampu2, OUTPUT);

  pinMode(lampu3, OUTPUT);

}

void loop() {

  digitalWrite(lampu1, HIGH);

  delay(2000);

  digitalWrite(lampu1, LOW);

  digitalWrite(lampu2, HIGH);

  delay(500);

  digitalWrite(lampu2, LOW);

  digitalWrite(lampu3, HIGH);

  delay(2000);

  digitalWrite(lampu3, LOW);

}

Kode Program untuk wokmi.toml

[wokwi]

version = 1

firmware = '.pio\build\esp32doit-devkit-v1\firmware.bin'

elf = '.pio\build\esp23doit-devkit-v1\firmware.elf'