**LAPORAN PRAKTIKUM MENYALAKAN**

**TRAFIC LIGHT DENGAN PUSH BUTTON MENGGUNAKAN ESP32**

**Dosen Pengampu :**

Ir. Subairi, ST., MT., IPM

**Disusun Oleh :**

Muhammad Baihaqi (233140700111098)

**PROGRAM STUDI TEKNOLOGI INFORMASI**

**FAKULTAS VOKASI UNIVERSITAS BRAWIJAYA**

**MALANG 2025**

**Abstract**

This experiment aims to analyze the implementation of the **MQTT protocol in an IoT-based traffic light monitoring system**. The system utilizes **ESP32** as the main microcontroller to control and monitor the status of **traffic lights (red, yellow, and green)** in real-time. The communication between the microcontroller and the monitoring platform is established using **the Blynk application**. The results indicate that **MQTT-based communication is more efficient** in terms of **latency and power consumption** compared to **HTTP-based communication**, making it a suitable choice for real-time traffic monitoring applications.

traffic lights.

*Keywords—Internet of Things, Traffic Light, ESP32,* ***MQTT-based***

**1.Introduction (Pendahuluan)**

Perkembangan teknologi Internet of Things (IoT) semakin pesat dan telah diterapkan dalam berbagai bidang, termasuk sistem transportasi. Salah satu implementasi IoT dalam transportasi adalah **sistem pemantauan lampu lalu lintas berbasis IoT** yang bertujuan untuk meningkatkan efisiensi dan keselamatan lalu lintas.

* 1. **Latar Belakang**

Lampu lalu lintas memiliki peran penting dalam mengatur arus kendaraan di persimpangan jalan. Namun, dalam operasionalnya, sering terjadi berbagai permasalahan seperti **lampu padam, kerusakan sistem, atau kegagalan sensor,** yang dapat menyebabkan kemacetan bahkan kecelakaan. Oleh karena itu, diperlukan sistem pemantauan yang dapat **memeriksa kondisi lampu lalu lintas secara real-time** agar teknisi dapat segera mengetahui dan menangani jika terjadi gangguan.

* 1. **Tujuan Eksperimen**

Tujuan dari praktikum ini adalah merancang dan mengembangkan sistem lampu lalu lintas berbasis mikrokontroler **ESP32 s**ebagai pengendali utama. Eksperimen ini dilakukan untuk menguji implementasi **pengaturan waktu pada lampu lalu lintas** dengan push button yaitu lampu merah Di tekan push button kedip 5 kali, Di tekan push button lampu merah dan hijau kedip bergantian, dan lampu kuning Di tekan push button lampu merah,kuning, dan hijau hidup bergantian.

1. **Methodology (Metodologi)**
   1. **Tools & Materials**

* ESP32
* LED Merah, Kuning, Hijau
* Resistor
* Push Button
* Software Arduino IDE
  1. **Implementation Steps**
* Menyusun rangkaian LED Merah, Kuning, Hijau dengan menghubungkannya ke ESP32.
* Menulis kode program untuk mengatur durasi penyalaan masing-masing LED.
* Memindahkan kode ESP32 ke Visual Studio Code, dan mengamati hasil penyalaan ketiga LED.

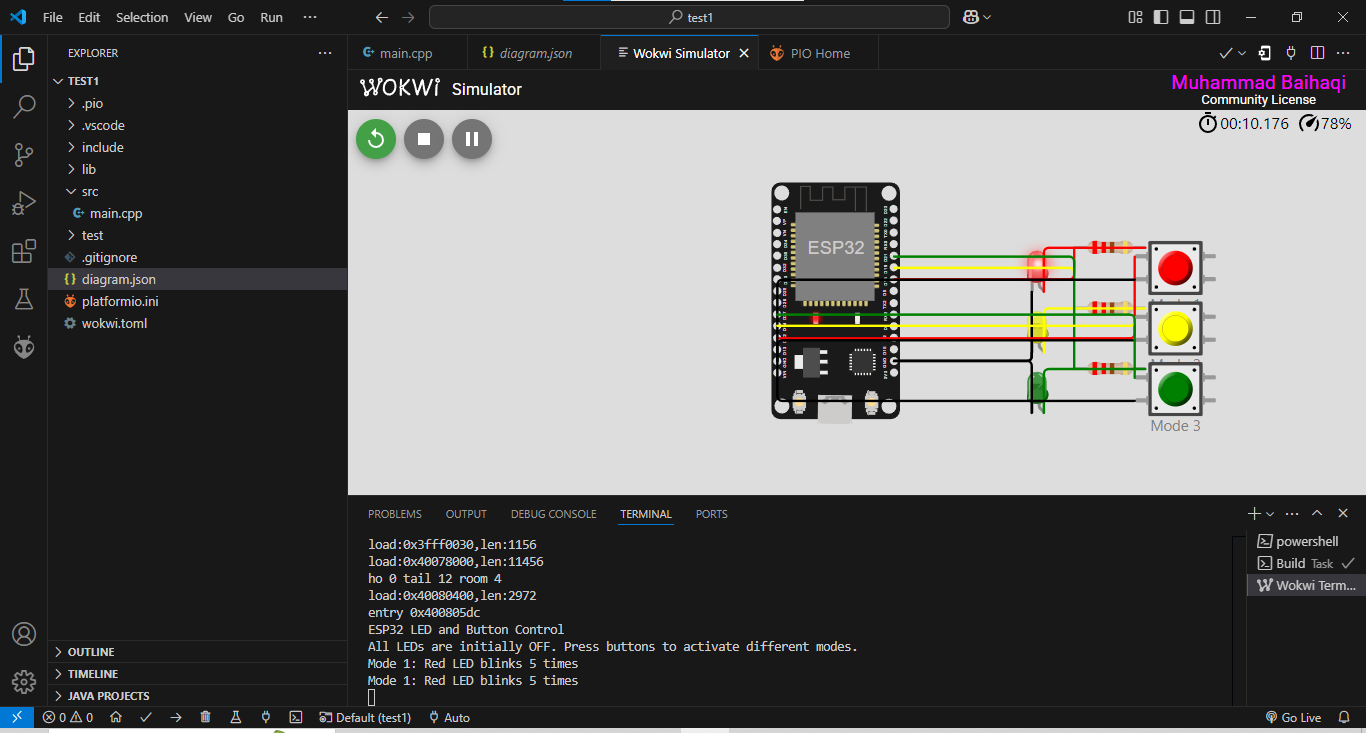
1. **Results and Discussion (Hasil dan pembahasan)**
   1. **Experiment Result**

Berikut ini merupakan hasil eksperimen dari code yang sudah dirancang

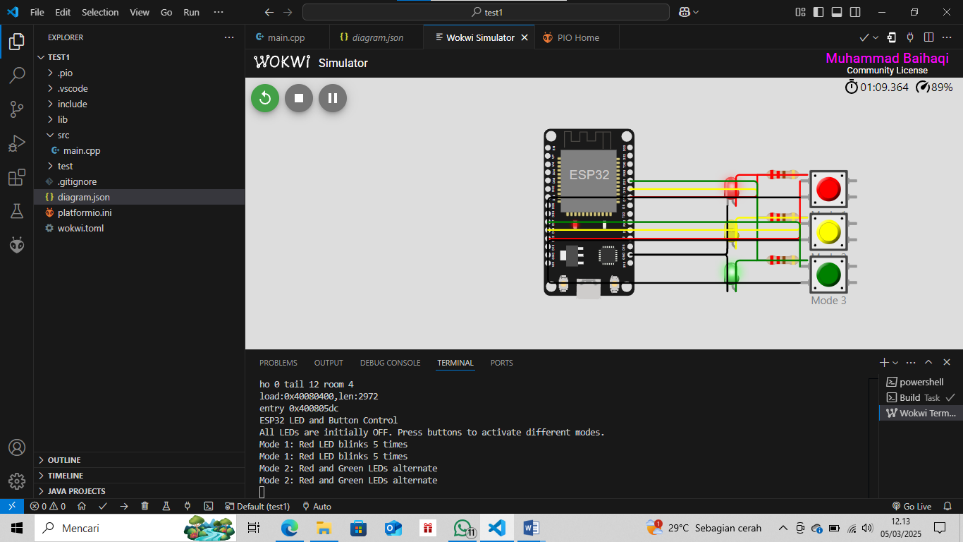
|  |  |
| --- | --- |
| **Lampu** | **Durasi** |
| Merah | Di tekan push button kedip 5 kali |
| Hijau | Di tekan push button lampu merah dan hijau kedip bergantian |
| Kuning | Di tekan push button lampu merah,kuning, dan hijau hidup bergantian |

Dibawah ini merupakan hasil dari eksperimen Trafic light ESP32 di VSCode

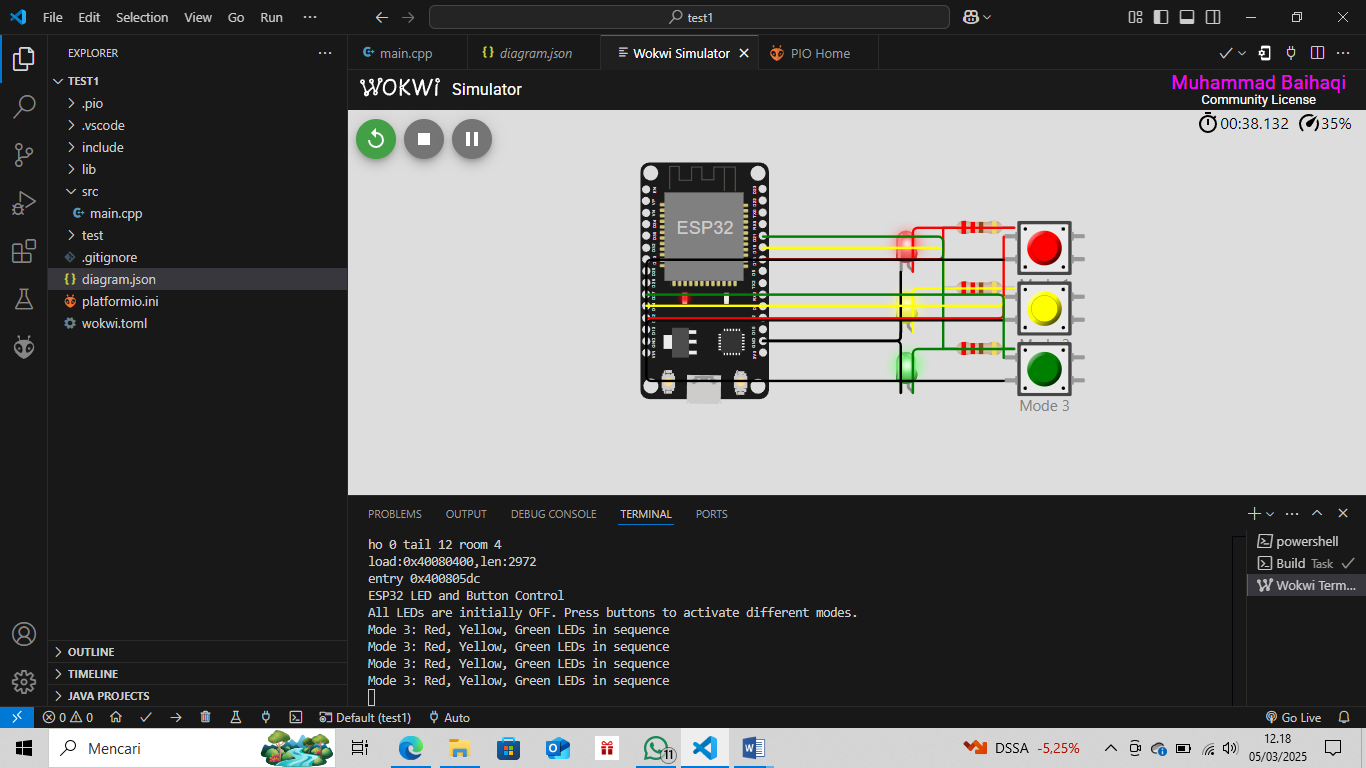
* Merah kedip 5 kali



* lampu merah dan hijau kedip bergantian



* lampu merah,kuning, dan hijau hidup bergantian



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

Dari File Main.cpp

#include <Arduino.h>

#define LED\_MERAH 18

#define LED\_KUNING 19

#define LED\_HIJAU 21

#define BUTTON\_MERAH 12

#define BUTTON\_KUNING 14

#define BUTTON\_HIJAU 27

// Constants

#define BLINK\_DELAY 200 // Delay between blinks in milliseconds

// For ESP32, LOW turns the LED on and HIGH turns it off when connected to GND

// Let's define macros to make the code more readable

#define LED\_ON LOW

#define LED\_OFF HIGH

void setup() {

  // Initialize serial communication

  Serial.begin(115200);

  Serial.println("ESP32 LED and Button Control");

  // Configure LED pins as outputs

  pinMode(LED\_MERAH, OUTPUT);

  pinMode(LED\_KUNING, OUTPUT);

  pinMode(LED\_HIJAU, OUTPUT);

  // Configure button pins as inputs with internal pull-up resistors

  pinMode(BUTTON\_MERAH, INPUT\_PULLUP);

  pinMode(BUTTON\_KUNING, INPUT\_PULLUP);

  pinMode(BUTTON\_HIJAU, INPUT\_PULLUP);

  Serial.println("All LEDs are initially OFF. Press buttons to activate different modes.");

}

// Function for Mode 1: Red LED blinks 5 times

void mode1() {

  Serial.println("Mode 1: Red LED blinks 5 times");

  for (int i = 0; i < 5; i++) {

    digitalWrite(LED\_MERAH, LED\_ON);  // Turn on Red LED

    delay(BLINK\_DELAY);

    digitalWrite(LED\_MERAH, LED\_OFF); // Turn off Red LED

    delay(BLINK\_DELAY);

    digitalWrite(LED\_MERAH, LOW);

  }

}

// Function for Mode 2: Red and Green LEDs alternate

void mode2() {

  Serial.println("Mode 2: Red and Green LEDs alternate");

  for (int i = 0; i < 5; i++) {

    // Red ON, Green OFF

    digitalWrite(LED\_MERAH, LED\_ON);

    digitalWrite(LED\_HIJAU, LED\_OFF);

    delay(BLINK\_DELAY);

    // Red OFF, Green ON

    digitalWrite(LED\_MERAH, LED\_OFF);

    digitalWrite(LED\_HIJAU, LED\_ON);

    delay(BLINK\_DELAY);

    digitalWrite(LED\_MERAH, LOW);

  }

  // Turn all LEDs off at the end

}

// Function for Mode 3: Red, Yellow, Green LEDs in sequence

void mode3() {

  Serial.println("Mode 3: Red, Yellow, Green LEDs in sequence");

  for (int i = 0; i < 2; i++) {

    // Red ON, others OFF

    digitalWrite(LED\_MERAH, LED\_ON);

    digitalWrite(LED\_KUNING, LED\_OFF);

    digitalWrite(LED\_HIJAU, LED\_OFF);

    delay(BLINK\_DELAY);

    // Yellow ON, others OFF

    digitalWrite(LED\_MERAH, LED\_OFF);

    digitalWrite(LED\_KUNING, LED\_ON);

    digitalWrite(LED\_HIJAU, LED\_OFF);

    delay(BLINK\_DELAY);

    // Green ON, others OFF

    digitalWrite(LED\_MERAH, LED\_OFF);

    digitalWrite(LED\_KUNING, LED\_OFF);

    digitalWrite(LED\_HIJAU, LED\_ON);

    delay(BLINK\_DELAY);

    digitalWrite(LED\_MERAH, LOW);

    digitalWrite(LED\_KUNING, LOW);

  }

}

void loop() {

  // Check if Button 1 (Red) is pressed

  if (digitalRead(BUTTON\_MERAH) == LOW) {

    mode1();

    // Wait for button release to prevent multiple activations

    while (digitalRead(BUTTON\_MERAH) == LOW) {

      delay(10);

    }

  }

  // Check if Button 2 (Yellow) is pressed

  if (digitalRead(BUTTON\_KUNING) == LOW) {

    mode2();

    // Wait for button release to prevent multiple activations

    while (digitalRead(BUTTON\_KUNING) == LOW) {

      delay(10);

    }

  }

  // Check if Button 3 (Green) is pressed

  if (digitalRead(BUTTON\_HIJAU) == LOW) {

    mode3();

    // Wait for button release to prevent multiple activations

    while (digitalRead(BUTTON\_HIJAU) == LOW) {

      delay(10);

    }

  }

  // Small delay for stability

  delay(10);

}

Code dari Diagram.json

{

  "version": 1,

  "author": "Muhammad Baihaqi",

  "editor": "wokwi",

  "parts": [

    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 0, "left": 0, "attrs": {} },

    { "type": "wokwi-led", "id": "led\_merah", "top": 50, "left": 200, "attrs": { "color": "red" } },

    { "type": "wokwi-led", "id": "led\_kuning", "top": 100, "left": 200, "attrs": { "color": "yellow" } },

    { "type": "wokwi-led", "id": "led\_hijau", "top": 150, "left": 200, "attrs": { "color": "green" } },

    { "type": "wokwi-pushbutton", "id": "btn\_merah", "top": 50, "left": 300, "attrs": { "color": "red", "label": "Mode 1" } },

    { "type": "wokwi-pushbutton", "id": "btn\_kuning", "top": 100, "left": 300, "attrs": { "color": "yellow", "label": "Mode 2" } },

    { "type": "wokwi-pushbutton", "id": "btn\_hijau", "top": 150, "left": 300, "attrs": { "color": "green", "label": "Mode 3" } },

    { "type": "wokwi-resistor", "id": "r1", "top": 50, "left": 250, "attrs": { "value": "220" } },

    { "type": "wokwi-resistor", "id": "r2", "top": 100, "left": 250, "attrs": { "value": "220" } },

    { "type": "wokwi-resistor", "id": "r3", "top": 150, "left": 250, "attrs": { "value": "220" } }

  ],

  "connections": [

    [ "esp:TX0", "$serialMonitor:RX", "", [] ],

    [ "esp:RX0", "$serialMonitor:TX", "", [] ],

    [ "esp:GND.1", "led\_merah:C", "black", [] ],

    [ "esp:GND.1", "led\_kuning:C", "black", [] ],

    [ "esp:GND.1", "led\_hijau:C", "black", [] ],

    [ "esp:D18", "r1:1", "red", [] ],

    [ "r1:2", "led\_merah:A", "red", [] ],

    [ "esp:D19", "r2:1", "yellow", [] ],

    [ "r2:2", "led\_kuning:A", "yellow", [] ],

    [ "esp:D21", "r3:1", "green", [] ],

    [ "r3:2", "led\_hijau:A", "green", [] ],

    [ "esp:D12", "btn\_merah:1.l", "red", [] ],

    [ "btn\_merah:2.l", "esp:GND.2", "black", [] ],

    [ "esp:D14", "btn\_kuning:1.l", "yellow", [] ],

    [ "btn\_kuning:2.l", "esp:GND.2", "black", [] ],

    [ "esp:D27", "btn\_hijau:1.l", "green", [] ],

    [ "btn\_hijau:2.l", "esp:GND.2", "black", [] ]

  ],

  "dependencies": {}

}