LAPORAN PRAKTIKUM MENYALAKAN

TRAFIC LIGHT MENGGUNAKAN ESP32 DAN PUSHBUTTON



Dosen Pengampu :

Ir. Subairi, ST., MT., IPM

Disusun Oleh :

M.Naufal DZ H(233140707111062)

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS VOKASI UNIVERSITAS BRAWIJAYA

MALANG 2025

Abstract

This practicum aims to design and implement a traffic light system based on the ESP32 microcontroller. In this experiment, the ESP32 is used as the main control unit to regulate the automatic switching of red, yellow, and green lights according to a pre-programmed schedule. The experiment successfully validated the implementation of a timer on the ESP32, demonstrating its effectiveness in automating the operation of traffic lights.

Keywords—Internet of Things, Traffic Light, ESP32, Timing Control.

1. Introduction (Pendahuluan)

Internet of Things (IoT) adalah konsep di mana objek fisik (perangkat, sensor, atau alat) diintegrasikan dengan kemampuan komunikasi dan pemrosesan data sehingga dapat saling berinteraksi melalui jaringan internet. Dengan memanfaatkan mikrokontroler ESP32, sistem lampu lalu lintas dapat dikontrol secara efisien tanpa memerlukan intervensi manual, Push button ditambahkan sebagai input eksternal yang memungkinkan pengguna menginterupsi siklus normal lampu lalu lintas, misalnya untuk memberikan prioritas tertentu

1.1 Latar Belakang

Kemajuan teknologi dalam bidang otomasi dan Internet of Things (IoT) telah membawa berbagai inovasi dalam sistem transportasi, termasuk sistem lampu lalu lintas. Salah satu teknologi yang banyak digunakan dalam pengembangan sistem otomatis adalah mikrokontroler ESP32.

1.2 Tujuan Eksperimen

Eksperimen ini bertujuan untuk membuat sistem lampu sederhana yang dikontrol menggunakan ESP32 dan push button. SP32 berperan sebagai mikrokontroler utama yang mengatur nyala dan mati lampu LED berwarna merah, kuning, dan hijau sesuai ketentuan yang ditentukan. Push button digunakan sebagai input eksternal yang memungkinkan pengguna menginterupsi siklus normal untuk memberikan prioritas tertentu

2. Methodology (Metodologi)

2.1 Tools & Materials

• ESP32

• LED Merah, Kuning, Hijau

• Resistor

• Pushbutton

2.2 Implementation Steps

• Menyusun rangkaian LED Merah, Kuning, Hijau dengan menghubungkannya ke ESP32.

• Siapkan pushbutton, lalu hubungkan 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.

3. Results and Discussion (Hasil dan pembahasan)

3.1 Experimental Results

Hasil ekperimen menunjukkan bahwa sistem lalu lintas atau Trafic Light berhasil bekerja sesuai dengan yang sudah di rancamg. Berikut adalah tabel

Lampu Hasil

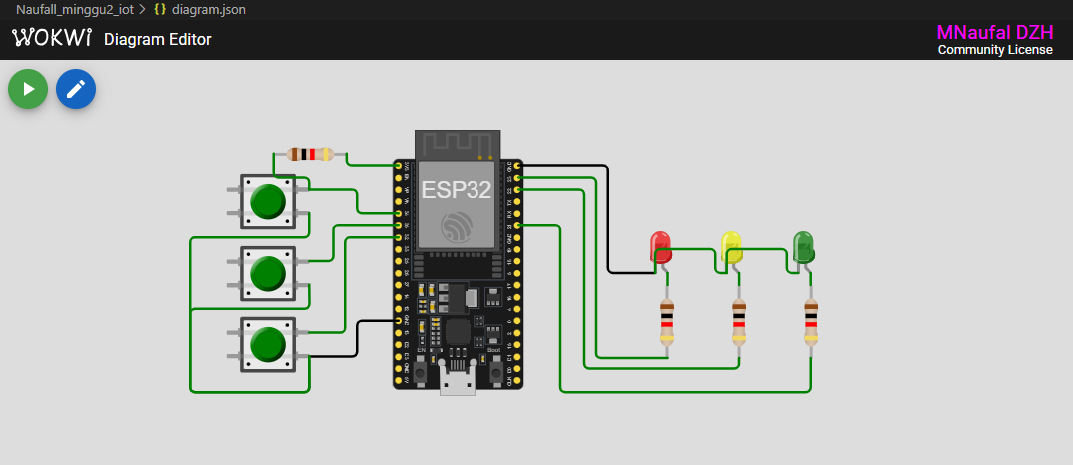
Lampu merah kedip 5 kali Berhasil

Lampu merah dan hijau kedip bergantian Berhasil

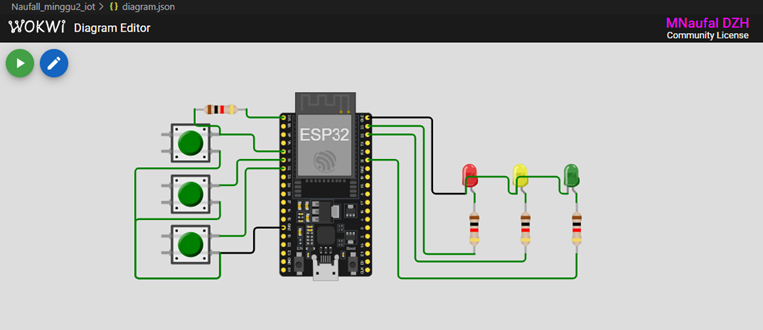
Lampu merah, kuninf dan hijau kedip bergantian Berhasil

Berikut adalah dokumentasi eksperimen meliputi screenshoot simulasi ESP32 di Visual Studio Code:

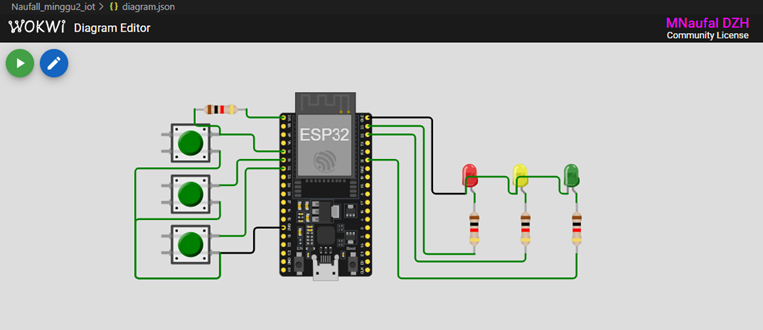
• LED Merah kedip 5 kali



• LED Merah dan hijau kedip bergantian



• LED Merah, kuning, dan hijau kedip bergantian



4. Appendix (Lampiran, jika diperlukan)

#include <Arduino.h>

// Definisikan pin untuk tombol dan LED

const int button1Pin = 34; // Tombol 1

const int button2Pin = 35; // Tombol 2

const int button3Pin = 32; // Tombol 3

const int led1Pin = 23; // LED merah

const int led2Pin = 22; // LED kuning

const int led3Pin = 21; // LED hijau

// Variabel untuk debouncing

unsigned long lastDebounceTime = 0;

const unsigned long debounceDelay = 50;

void setup() {

// Inisialisasi pin tombol sebagai input

pinMode(button1Pin, INPUT\_PULLUP);

pinMode(button2Pin, INPUT\_PULLUP);

pinMode(button3Pin, INPUT\_PULLUP);

// Inisialisasi pin LED sebagai output

pinMode(led1Pin, OUTPUT);

pinMode(led2Pin, OUTPUT);

pinMode(led3Pin, OUTPUT);

// Matikan semua LED pada awal

digitalWrite(led1Pin, LOW);

digitalWrite(led2Pin, LOW);

digitalWrite(led3Pin, LOW);

}

void loop() {

// Membaca status dari tombol dengan debounce

int button1State = digitalRead(button1Pin);

int button2State = digitalRead(button2Pin);

int button3State = digitalRead(button3Pin);

// Pastikan tombol ditekan dalam waktu yang lebih lama dari debounceDelay

if (millis() - lastDebounceTime > debounceDelay) {

// Logika Tombol 1 (Tombol ditekan, LED merah berkedip 5 kali)

if (button1State == LOW) {

lastDebounceTime = millis();

for (int i = 0; i < 5; i++) {

digitalWrite(led1Pin, HIGH); // Nyalakan LED merah

delay(500); // Tunggu 500ms

digitalWrite(led1Pin, LOW); // Matikan LED merah

delay(500); // Tunggu 500ms

}

}

// Logika Tombol 2 (Tombol ditekan, LED merah dan hijau berkedip bergantian)

if (button2State == LOW) {

lastDebounceTime = millis();

for (int i = 0; i < 5; i++) {

digitalWrite(led1Pin, HIGH); // Nyalakan LED merah

digitalWrite(led3Pin, LOW); // Matikan LED hijau

delay(500); // Tunggu 500ms

digitalWrite(led1Pin, LOW); // Matikan LED merah

digitalWrite(led3Pin, HIGH); // Nyalakan LED hijau

delay(500); // Tunggu 500ms

}

// Matikan kedua LED setelah selesai

digitalWrite(led1Pin, LOW);

digitalWrite(led3Pin, LOW);

}

// Logika Tombol 3 (Tombol ditekan, LED merah, kuning, dan hijau berkedip bergantian)

if (button3State == LOW) {

lastDebounceTime = millis();

for (int i = 0; i < 5; i++) {

digitalWrite(led1Pin, HIGH); // Nyalakan LED merah

digitalWrite(led2Pin, LOW); // Matikan LED kuning

digitalWrite(led3Pin, LOW); // Matikan LED hijau

delay(500); // Tunggu 500ms

digitalWrite(led1Pin, LOW); // Matikan LED merah

digitalWrite(led2Pin, HIGH); // Nyalakan LED kuning

delay(500); // Tunggu 500ms

digitalWrite(led2Pin, LOW); // Matikan LED kuning

digitalWrite(led3Pin, HIGH); // Nyalakan LED hijau

delay(500); // Tunggu 500ms

digitalWrite(led3Pin, LOW); // Matikan LED hijau

}

}

}

}