

LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

Praktikum Implementasi Simulasi Relay menggunakan ESP32 melalui IoT



Dafa Putra Ady Pratama
Fakultas Vokasi, Universitas Brawijaya
Email: dafaputra@student.ub.ac.id

Abstract

Relay merupakan komponen elektromekanis yang berfungsi sebagai saklar otomatis dalam rangkaian listrik. Praktikum ini bertujuan untuk memahami prinsip kerja relay serta mengimplementasikan simulasi relay dalam berbagai aplikasi kelistrikan. Dalam simulasi ini, relay dikendalikan oleh sinyal listrik untuk mengaktifkan atau menonaktifkan beban, seperti lampu atau motor, sesuai dengan kondisi yang diberikan. Metode yang digunakan dalam praktikum meliputi perancangan rangkaian relay, penyusunan skema kelistrikan, serta pengujian menggunakan simulator atau perangkat keras secara langsung. Hasil praktikum menunjukkan bahwa relay dapat digunakan sebagai pengendali dalam sistem otomatisasi dengan respon yang cepat dan efisien. Dengan memahami prinsip kerja relay melalui simulasi, mahasiswa dapat menerapkan konsep ini dalam berbagai bidang, termasuk otomasi industri dan sistem keamanan.

Keywords—*Internet of Things, ESP32, LED, Relay, Saklar*

1. Introduction

1.1 Latar Belakang Praktikum IoT yang Dilakukan

Relay merupakan komponen penting dalam sistem kelistrikan yang berfungsi sebagai saklar otomatis untuk mengendalikan arus listrik. Penggunaannya luas dalam otomasi, proteksi listrik, dan perangkat elektronik. Praktikum ini bertujuan untuk memahami prinsip kerja relay melalui simulasi, sehingga mahasiswa dapat mengaplikasikannya dalam sistem otomatisasi dan pengendalian listrik.

1.2 Tujuan Eksperimen

Praktikum ini bertujuan untuk memahami prinsip kerja relay serta cara mengimplementasikannya dalam sistem kelistrikan melalui simulasi. Dengan praktikum ini, mahasiswa dapat menganalisis kinerja relay dalam berbagai kondisi, mengembangkan keterampilan dalam perancangan rangkaian berbasis relay, serta memahami penerapannya dalam otomasi dan pengendalian listrik.

2. Methodology

2.1 Tools & Materials

- **Mikrokontroler:** ESP32
- **LED:** 1 buah LED
- **Relay:** 1 buah Relay
- **Software:** Visual Code, Web Browser
- **Koneksi Jaringan:** Wi-Fi

2.2 Implementation Steps

1. **Persiapan Hardware:**
 - Hubungkan 1 Led dengan ESP32
 - Hubungkan Relay dengan ESP32
 - Hubungkan Saklar dengan ESP32
2. **Pengaturan Software:**
 - Program ESP32 menggunakan Arduino IDE.
 - Konfigurasi Wi-Fi untuk menghubungkan ESP32 ke jaringan internet.
 - Gunakan server web lokal pada ESP32 untuk mengontrol LED.

3. Pengkodean dan Implementasi Web:

- Tulis kode untuk mengendalikan ESP32, yang akan merespon perintah dari halaman web.
- Program untuk mengubah status ESP32

4. Pengujian:

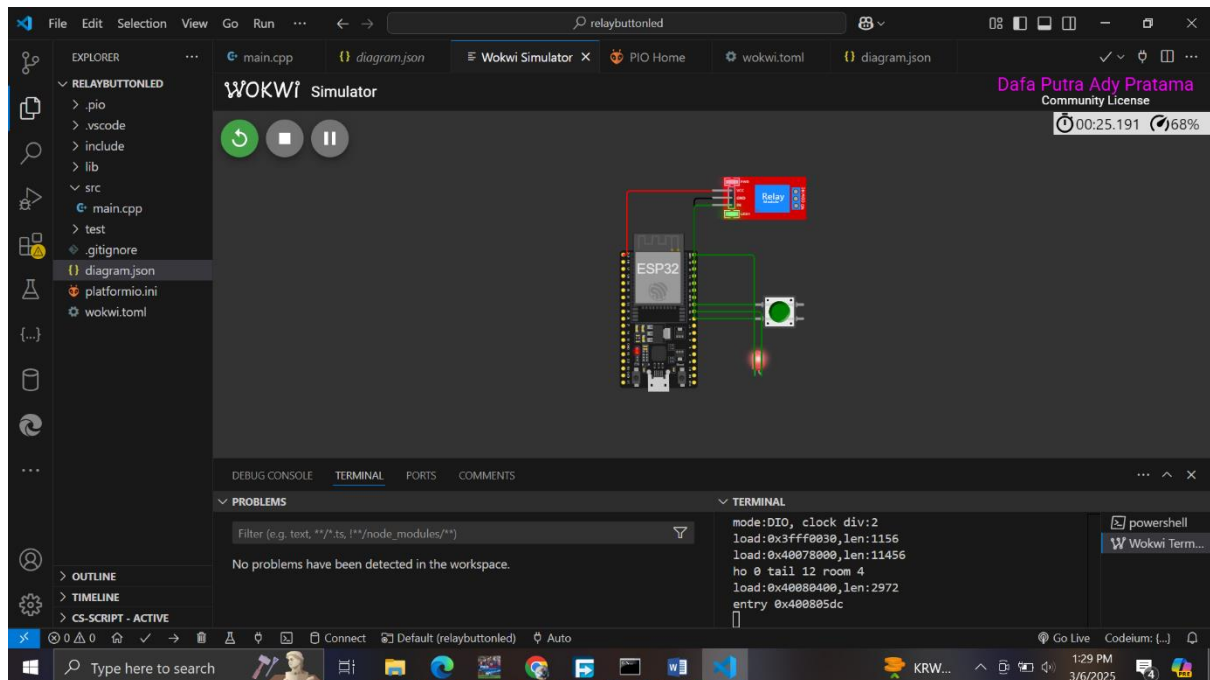
- Uji sistem dengan mengakses halaman web dan mengontrol saklar untuk LED. Pastikan bahwa status LED tercermin dengan benar di perangkat keras.

3. Results and Discussion

3.1 Experimental Results

Pada praktikum ini, relay berhasil diimplementasikan dalam simulasi untuk mengontrol beban listrik sesuai dengan sinyal pemicu yang diberikan. Pengujian menunjukkan bahwa relay dapat bekerja dengan baik dalam menghubungkan dan memutus arus listrik sesuai dengan kondisi yang telah ditentukan. Saat tegangan aktif diberikan ke kumparan relay, kontak saklar berubah posisi dan mengaktifkan beban, sedangkan ketika tegangan dilepas, relay kembali ke kondisi awal. Selain itu, analisis menunjukkan bahwa waktu respons relay bergantung pada jenis dan spesifikasi yang digunakan. Hasil ini membuktikan bahwa relay dapat digunakan secara efektif dalam sistem otomatisasi dan pengendalian listrik.

Berikut adalah hasilnya:



4. Appendix

```
1  #include <Arduino.h>
2
3  // Define pin numbers
4  const int ButtonPin = 19; // GPIO19 connected to the pushbutton
5  const int LedPin = 18;    // GPIO18 connected to the LED
6  const int RelayPin = 23;  // GPIO23 connected to the relay module
7
8  void setup() {
9      // Set pin modes
10     pinMode(ButtonPin, INPUT_PULLUP); // Set the button pin as an input with an internal pull-up resistor
11     pinMode(LedPin, OUTPUT);          // Set the LED pin as an output
12     pinMode(RelayPin, OUTPUT);        // Set the relay pin as an output
13
14     // Initialize the outputs to be OFF
15     digitalWrite(LedPin, LOW);
16     digitalWrite(RelayPin, LOW);
17 }
18
19 void loop() {
20     // Read the state of the button
21     int buttonState = digitalRead(ButtonPin);
22
23     // Check if the button is pressed
24     // Since the button is wired to pull the pin LOW when pressed, we check for LOW
25     if (buttonState == LOW) {
26         digitalWrite(LedPin, HIGH); // Turn on the LED
27         digitalWrite(RelayPin, HIGH); // Turn on the relay
28     } else {
29         digitalWrite(LedPin, LOW); // Turn off the LED
30         digitalWrite(RelayPin, LOW); // Turn off the relay
31     }
32 }
```