

LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

(Simulasi Praktik Simulasi Relay, Button & LED

**Berbasis IoT dengan ESP32 Menggunakan Platform Wokwi yang Diintegrasikan
dengan VS Code)**



Muhammad Kadavi
Fakultas Vokasi, Universitas Brawijaya
Email: kadavi2945@student.ub.ac.id

Abstrak

Pada praktikum ini, mikrokontroler ESP32 digunakan untuk sistem kontrol simulasi yang didasarkan pada Internet of Things dengan menggunakan komponen Relay, Button, dan LED. Simulasi tersebut ditujukan untuk memberikan contoh cara komponen elektronik berfungsi dalam IoT dan memeriksa kemampuan alat pengujian sebagai pengontrol dan pengawas digital. Wokwi Platform berfungsi sebagai lingkungan simulasi, beberapa kali diintegrasikan dengan Visual Studio Code menggunakan PlatformIO extension untuk memperlancar pemrograman dan debugging. Relay berfungsi sebagai saklar elektronik yang dikendalikan oleh ESP32, Button adalah ancak trigger, dan LED digunakan sebagai indikator visual. Sintaks pemrograman mengatur jalannya sistem, dari kontrol Relay atas input tombolannya, hingga perubahan status LED sengaja ditetapkan. Berdasarkan hasil eksperimen yang diperoleh, dapat diketahui bahwa simulasi berjalan stabil, Relay dapat sukses mengaktifkan/menonaktifkan rangkaian listrik secara virtual, Button berperan sebagai pemicu perlakuan status, dan LED mampu memvisualisasikan jalannya simulasi sistem secara real-time. Selain itu, hal yang terpenting yaitu melalui implementasi integrasi IoT, memungkinkan hamon kampanye monitoring dan kontrol komponen melalui antarmuka digital. Eksperimen ini membuktikan bahwa ESP32, logika pemrograman, dan platform simulasi seperti Wokwi mampu menciptakan replikasi sistem kontrol IoT nyata, dengan kata lain sebagai batu pijakan untuk pengembangan sistem otomasi lebih canggih di masa depan.

Pendahuluan

Perkembangan teknologi Internet of Things (IoT) dalam tiga dasawarsa terakhir, membuka celah peluang unik terhadap sistem kontrol elektronik untuk dipelajari secara virtual. Simulasi komponen seperti Relay, Button, dan LED memberikan solusi efisien terhadap pembelajaran bertema IoT yang hemat biaya, serta menghemat waktu dalam pengujian kemampuan kontrol digital komponen Aktor dan Sensor sulit didapatkan. Relay dalam sistem ini adalah saklar elektronik yang dikendalikan secara otomatis, Button sebagai input yang mewakili input interaksi pengguna, dan LED sebagai output yang memiliki fungsi sebagai indikator visual sinyal output. Uniknya, simulasi Relay, Button, dan LED pada mikrokontroler ESP32 memiliki ciri khas sebagai alat programmer yang memungkinkan pengaturan loop dan selang waktu sebagai pencegahan output yang tidak jelas. Integrasi Platform Wokwi dengan Visual Studio Code dan PlatformIO memungkinkan pengaturan resource penugasan kontrol, membuat logika kontrol, juga lead time yang sangat efisien serta menghemat waktu praktikum. Simulasi ini memungkinkan pengguna mendekatkan dan mengeksplorasi beberapa komponen penunjang IoT lain seperti antara lain komunikasi antara input-output dan manajemen pengaturan kontrol logika pemrograman. Simulasi sistem ini memungkinkan pengguna untuk memvalidasi algoritma kontrol, menguji kestabilan sistem, dan memahami bagaimana komponen saling terkait satu sama lain dalam batasan dan kondisi pengujian yang beragam. Oleh karena itu, solusi ini menciptakan landasan tindakan yang kuat bagi pembelajaran konsep Internet of Things dan pengembangan sistem otomasi yang luas dan semakin kompleks masa depan tanpa terbatas ketersediaan komponen fisik.

Latar Belakang

Perkembangan Internet of Things dan sistem otomasi telah menjadi bagian utama dari generasi masa kini, di mana integrasi antara periferal elektronik, sensor dan fungsi logika pemrograman menjadi lebih lazim untuk menciptakan solusi di bidang efisiensi. Seiring dengan itu, pembelajaran IoT salah satunya menuntut beberapa konsep dasar mengenai interaksi antara komponen elektronik dasar seperti Relay, Button dan LED, yang berperan penting dalam sistem kontrol otomatis. Beberapa kendala seperti keterbatasan pembelian komponen, kerusakan hardware, dan risiko lain yang berkembang dari eksperimen seringkali menjadi hambatan, terutama dalam edukasi atau proses pengelaborasi proyek. Simulasi virtual telah menjadi solusi inovatif untuk mengatasi tantangan terkait. Dengan memanfaatkan platform yang disediakan oleh Wokwi, pengguna dapat mendesain, memprogram, dan mengubah beberapa komponen elektronik digital tanpa harus bergantung pada hardware. ESP32, mikrokontroler yang dikenal karena kemampuannya dalam menghubungkan periferal dengan jaringan IoT, digunakan dalam simulasi ini sebagai komponen utama. Platform ini diintegrasikan dengan Visual Studio Code melalui ekstensi bernama PlatformIO, yang memungkinkan untuk menulis kode, melakukan debugging dan beberapa simulasi terdapat dalam satu environment terintegrasi. Praktikum ini melakukan simulasi sistem kontrol dasar berbasis IoT, dengan beberapa fungsi konvensional yang perlu dikonfigurasi untuk memahami algoritma. Relay dikendalikan menjadi saklar yang diaktifkan atau non-aktifkan oleh kondisi Button, dan LED akan merespon status sistem. Dengan simulasi virtual, pengguna dapat memahami konsep komunikasi digital, manajemen fungsi logika kontrol, dan respon dari sistem perihwal input pengguna tanpa batasan waktu dan risiko teknis. Selain itu, ini juga memecahkan masalah dengan sumber daya alam, dan praktikum ini memperkuat pengalaman dengan konsep IoT dan membuat beberapa skenario tes lebih optimal dari segi waktu dan sumber daya. Ini menjadikan langkah strategis dalam persiapan generasi berikutnya tentang perkembangan otomasi dan industri 4.0.

Tujuan

tujuan praktik ini adalah memberikan pemahaman yang lebih baik tentang logika algoritma kontrol relay, button, dan LED IoT berbasis IoT melalui aktivitas simulasi virtual. Oleh karena itu, melibatkan penggunaan platform Wokwi dan ESP32 dengan perangkat simulasi tanpa perangkat fisik untuk menguji fungsi komponen elektronik. Selain itu, memverifikasi respons sistem dengan dasar program yang telah dirancang oleh pemrograman. Dari sini, harapannya, simulasi ini menjadi landasan untuk pengembangan sistem IoT dan otomasi most lebih kompleks dan dapat diukur.

Metodelogi

Simulasi sistem Relay, Button, dan LED dengan menggunakan ESP32 dilakukan dengan menulis program di Visual Studio Code melalui PlatformIO untuk mengontrol logika sistem: input trigger yang digunakan pada button, digital saklar utama yang diatur oleh Relay, dan indicator dari kondisi sistem yaitu LED. Simulasi dijalankan di platform Wokwi untuk menguji responsif sistem seperti aktivasi Relay ketika Button ditekan dan perubahan status LED saat sistem beroperasi. Sistem dilakukan validasi secara visual dan logika terhadap algoritma berdasarkan konsistensinya.

Software & Hardware

Software	Hardware
Wokwi	Esp32 (virtual)
Extension PlatformIO	Lampu led 1 buah (virtual)
VSCode	Relay Module(virtual)
	PressButoon 6mm (Virtual)
	Laptop

2.2 Implementation Steps (Langkah Implementasi)

1. Persiapan

1. Buka Wokwi (<https://wokwi.com>)
2. Buat proyek baru dengan ESP32, lalu tambahkan komponen:
 - 3 LED (Merah, Kuning, Hijau)
 - ILI9341 2.8" TFT-LCD Display
3. Hubungkan komponen virtual ke ESP32 menggunakan diagram JSON.

2. Instalasi VS Code dan PlatformIO

1. Download dan Install VS Code dari <https://code.visualstudio.com/>
2. Buka VS Code, lalu buka **Extensions Marketplace** (Ctrl+Shift+X).
3. Cari "PlatformIO", lalu **install extension PlatformIO**.
4. **Restart VS Code** setelah instalasi selesai.

3. Membuat Proyek di PlatformIO

1. Buka VS Code dan **klik ikon PlatformIO** (di sidebar kiri).
2. Buat proyek baru dengan:
 - **Board:** DOIT ESP32 DEVKIT V1
 - **Framework:** Arduino
 - **Nama Proyek:** TrafficLightESP32

4. Buat File wokwi.toml,dan salin syntax di bawah ini

```
[wokwi]
version = 1
firmware = '.pio/build/esp32doit-devkit-v1/firmware.bin'
```

```
elf = '.pio/build/esp32doit-devkit-  
v1/firmware.elf'
```

4. Menulis Kode ESP32 di VS Code

1. Buka file `src/main.cpp` dan masukkan kode berikut:

```
#include <Arduino.h>  
  
// Define pin numbers  
const int ButtonPin = 19; // GPIO19 connected to the pushbutton  
const int LedPin = 18; // GPIO18 connected to the LED  
const int RelayPin = 23; // GPIO23 connected to the relay module  
  
void setup() {  
    // Set pin modes  
    pinMode(ButtonPin, INPUT_PULLUP); // Set the button pin as an input with an internal pull-up  
    resistor  
    pinMode(LedPin, OUTPUT); // Set the LED pin as an output  
    pinMode(RelayPin, OUTPUT); // Set the relay pin as an output  
  
    // Initialize the outputs to be OFF  
    digitalWrite(LedPin, LOW);  
    digitalWrite(RelayPin, LOW);  
}  
  
void loop() {  
    // Read the state of the button  
    int buttonState = digitalRead(ButtonPin);  
  
    // Check if the button is pressed  
    // Since the button is wired to pull the pin LOW when pressed, we check for LOW  
    if (buttonState == LOW) {  
        digitalWrite(LedPin, HIGH); // Turn on the LED  
        digitalWrite(RelayPin, HIGH); // Turn on the relay  
    } else {  
        digitalWrite(LedPin, LOW); // Turn off the LED  
        digitalWrite(RelayPin, LOW); // Turn off the relay  
    }  
}
```

5. Menjalankan Simulasi di Wokwi

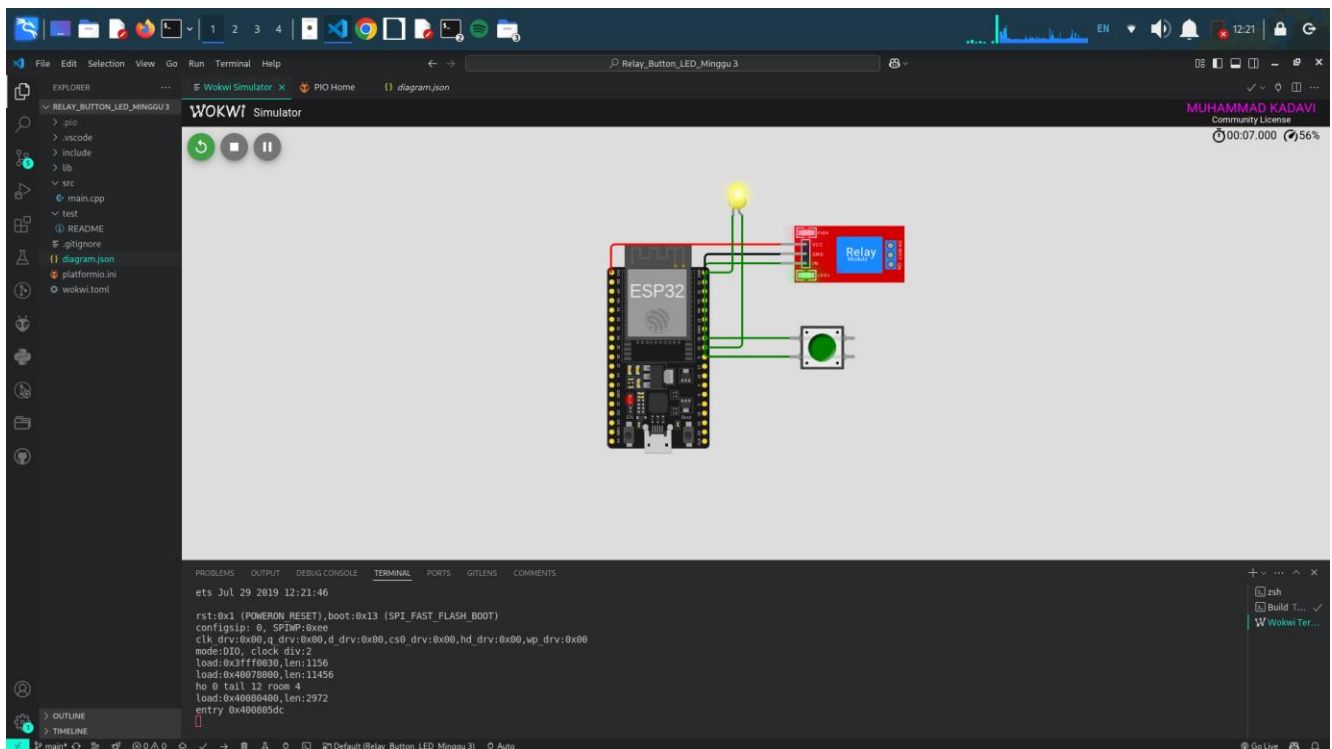
1. Buka proyek Wokwi yang sudah dibuat.
2. Tambahkan diagram JSON untuk menghubungkan ESP32 dengan LED dan ILI9341.
3. Klik tombol "RUN" untuk menjalankan simulasi dan melihat hasilnya.

3. Results and Discussion (Hasil dan Pembahasan)

Simulasi ini dilakukan dan hasilnya menunjukkan bahwa balikan sistem berlangsung real-time tanpa adanya penundaan

atau jeda fungsi. Fungsi-fungsi dari ketiga komponen sistem tersebut, yaitu Tombol dan LED, tertulis dan disimulasikan sesuai logika algoritma yang sudah diprogram. Ketika Tombol ditekan, status dari Relay bisa berubah dari ON ke OFF atau sebaliknya. LED tidak memiliki lampu yang nyata, namun tetap menyala atau tidak bergantung status dari Relay.

3.1 Experimental Results (Hasil Eksperimen)



4. Appendix (Lampiran, jika diperlukan)

```
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 }
33
```