**LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)**

Simulasi dan Implementasi Relay, Button & LED dengan ESP32 di Wokwi & VSCode

*Author : Nelly Fananda Melani*

*Fakultas Vokasi, Universitas Brawijaya*

*Email:* [*nellyfananda15@student.ub.ac.id*](mailto:nellyfananda15@student.ub.ac.id)

**Abstract**

This experiment aims to simulate and implement the use of a relay, button, and LED with the ESP32 microcontroller in the development environment of PlatformIO on Visual Studio Code (VSCode) and the Wokwi Simulator. This implementation allows testing without requiring additional hardware. The system reads input from the button and then controls the LED and relay based on the button's status. If the button is pressed, the LED and relay will turn on; otherwise, both will remain off. The development process includes creating a new project in PlatformIO, writing code in C++ in the main.cpp file, configuring the platformio.ini file, and designing the connection diagram in the diagram.json file. After successful compilation, firmware.bin and firmware.elf files are generated for simulation in Wokwi. The simulation results show that the system can read the button status and control the LED and relay in real-time. This experiment provides fundamental insights into microcontroller programming, electronic component integration, and IoT-based system implementation. Beyond serving as an academic exercise, this project has the potential for further development in various applications, such as smart home automation, security systems, and remote electrical device control.

**Keywords**— *ESP32, Relay, Button, LED, Wokwi*

**1. Introduction (Pendahuluan)**

Perkembangan pesat dalam teknologi Internet of Things (IoT) telah memungkinkan otomatisasi dan kendali perangkat elektronik secara real-time di berbagai sektor, termasuk otomasi rumah pintar dan sistem kendali jarak jauh. Salah satu penerapan dasar IoT adalah pengendalian perangkat listrik menggunakan mikrokontroler, yang berperan dalam berbagai aplikasi seperti sistem keamanan, pengelolaan energi, dan otomasi industri. Untuk mewujudkan sistem kendali ini, diperlukan mikrokontroler yang efisien dalam membaca input dan mengontrol aktuator dengan respons yang cepat dan akurat.

Dalam praktikum ini, digunakan mikrokontroler ESP32 yang dikombinasikan dengan relay, tombol (button), dan LED untuk membangun sistem kendali sederhana. Sistem ini dikembangkan dan disimulasikan menggunakan PlatformIO di Visual Studio Code (VSCode) dengan bantuan Wokwi Simulator, sehingga memungkinkan pengujian tanpa memerlukan perangkat keras fisik. Dengan memanfaatkan PlatformIO dan Wokwi, pengembang dapat mensimulasikan perilaku mikrokontroler, menguji kode, serta melakukan debugging sebelum menerapkannya pada perangkat nyata.

Tujuan dari proyek ini adalah untuk mengembangkan, mensimulasikan, dan menganalisis implementasi sistem kendali berbasis ESP32 menggunakan relay, tombol, dan LED. Praktikum ini bertujuan untuk memberikan wawasan mengenai pemrograman mikrokontroler, integrasi sensor dan aktuator, serta implementasi sistem kendali berbasis IoT secara real-time. Selain itu, proyek ini dapat menjadi dasar untuk pengembangan lebih lanjut dalam sistem otomasi dan kendali perangkat listrik berbasis IoT.

**1.1 Background of the IoT Practicum (Latar Belakang Praktikum IoT)**

Perkembangan Internet of Things (IoT) telah membawa perubahan besar dalam berbagai sektor, seperti otomasi industri, sistem keamanan, serta pengendalian perangkat elektronik secara jarak jauh. IoT memungkinkan berbagai perangkat untuk saling terhubung dan berkomunikasi melalui jaringan internet, sehingga menciptakan sistem yang lebih efisien, otomatis, dan terintegrasi. Salah satu aplikasi dasar dalam IoT adalah sistem kendali dan monitoring, yang sering digunakan dalam berbagai bidang seperti rumah pintar (smart home), industri, dan sistem keamanan.

Dalam implementasi sistem IoT, mikrokontroler berperan penting sebagai penghubung antara sensor dan aktuator. ESP32 menjadi salah satu pilihan mikrokontroler yang populer karena mendukung konektivitas Wi-Fi dan Bluetooth, serta memiliki kemampuan pemrosesan yang tinggi. Selain itu, relay, tombol (button), dan LED merupakan komponen dasar yang sering digunakan dalam sistem kendali otomatis, baik untuk menyalakan perangkat listrik, memberikan indikasi visual, maupun menerima input dari pengguna.

Untuk memahami konsep dan implementasi sistem kendali berbasis IoT, dilakukan praktikum dengan mensimulasikan penggunaan relay, tombol, dan LED dengan ESP32. Simulasi ini dikembangkan menggunakan PlatformIO di Visual Studio Code (VSCode) dengan bantuan Wokwi Simulator, sehingga memungkinkan pengujian tanpa memerlukan perangkat keras fisik. Dengan adanya simulasi ini, peserta praktikum dapat mempelajari cara kerja mikrokontroler dalam membaca input dari tombol serta mengontrol relay dan LED secara real-time.

Praktikum ini diharapkan dapat memberikan wawasan mendalam mengenai pemrograman mikrokontroler, integrasi sensor dan aktuator, serta implementasi sistem kendali berbasis IoT. Selain sebagai latihan akademik, proyek ini dapat menjadi dasar bagi pengembangan lebih lanjut dalam otomasi rumah, sistem keamanan, serta kendali perangkat listrik berbasis IoT.

**1.2 Experimental Objectives (Tujuan Eksperimen)**

Eksperimen ini bertujuan untuk:

1. Mempelajari Implementasi IoT

* Memahami konsep dasar Internet of Things (IoT) serta bagaimana mikrokontroler dapat digunakan untuk mengendalikan perangkat elektronik secara real-time.

1. Menggunakan ESP32 sebagai Mikrokontroler

* Menguji kemampuan ESP32 dalam membaca input dari tombol serta mengontrol relay dan LED sesuai dengan kondisi yang diinginkan.

1. Mengembangkan dan Mensimulasikan Sistem Kendali

* Menggunakan Wokwi Simulator di PlatformIO (Visual Studio Code) untuk mensimulasikan sistem tanpa memerlukan perangkat keras fisik.

1. Mengimplementasikan Pemrograman Mikrokontroler

* Mempraktikkan pemrograman Arduino (C++) untuk membaca input tombol dan mengendalikan relay serta LED berdasarkan logika yang telah dirancang.

1. Menganalisis Respons Sistem

* Menginterpretasikan hasil pembacaan tombol serta memahami faktor-faktor yang dapat memengaruhi performa sistem kendali berbasis ESP32.

1. Menjadi Dasar Pengembangan Sistem IoT Lanjutan

* Memberikan wawasan bagi pengembangan proyek IoT yang lebih kompleks, seperti otomasi rumah, sistem keamanan, dan kendali perangkat listrik berbasis IoT.

**2. Methodology (Metodologi)**

Eksperimen ini dilakukan dengan mensimulasikan sistem berbasis **ESP32, relay, tombol (button), dan LED** menggunakan **Wokwi Simulator**. Langkah-langkah dalam metodologi ini meliputi:

1. Membuat Diagram Simulasi di Wokwi Simulator untuk menghubungkan ESP32, relay, tombol, dan LED.
2. Menulis Kode Program dalam Arduino (C++) menggunakan PlatformIO (VSCode).
3. Konfigurasi Lingkungan dengan mengedit platformio.ini dan wokwi.toml.
4. Kompilasi & Simulasi di Wokwi Simulator tanpa perangkat keras fisik.
5. Pengamatan Hasil Simulasi melalui serial monitor dan memeriksa respons sistem.
6. Analisis dan Evaluasi kinerja sistem dalam membaca input dan mengontrol relay serta LED..

**2.1 Tools & Materials (Alat dan Bahan)**

Eksperimen ini dilakukan secara virtual menggunakan **Wokwi Simulator**, sehingga tidak memerlukan perangkat keras fisik. Alat dan bahan yang digunakan dalam proyek ini meliputi::

* ESP32 Devkit V1 (simulasi pada Wokwi)
* Relay module (simulasi pada Wokwi)
* Tombol (Push Button) (simulasi pada Wokwi)
* LED (simulasi pada Wokwi)
* Laptop/PC dengan Visual Studio Code dan PlatformIO sebagai lingkungan pengembangan
* Wokwi Simulator untuk menjalankan dan menguji kode tanpa perangkat fisik
* Library Arduino untuk pemrograman mikrokontroler
* Koneksi internet untuk mengakses dan menjalankan simulasi di Wokwi

**2.2 Implementation Steps (Langkah Implementasi)**

Eksperimen ini dilakukan melalui beberapa langkah utama:

1. **Menyiapkan Lingkungan Pengembangan**

* Instal Visual Studio Code (VSCode) dan tambahkan ekstensi PlatformIO IDE.

Sebuah gambar berisi teks, Font, cuplikan layar, logo

Konten yang dihasilkan AI mungkin salah.

* Buat proyek ESP32 baru menggunakan PlatformIO.

PLATFORMIO -> QUICK ACCESS -> Open -> New Project -> Project Wizard -> Finish

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

* Akses Wokwi Simulator untuk merancang Simulasi Relay, Button & LED

**Sebuah gambar berisi teks, cuplikan layar, Software multimedia, software

Konten yang dihasilkan AI mungkin salah.**

**Sebuah gambar berisi cuplikan layar, teks, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.**

(Pilih Starter Templates -> **ESP32**)

1. **Merancang Diagram Simulasi di Wokwi**

* Buat diagram baru di wokwi.com.

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

* **Hubungkan ESP32 dengan relay, tombol, dan LED secara virtual sesuai skema rangkaian.**

Sebuah gambar berisi teks, cuplikan layar, Font

Konten yang dihasilkan AI mungkin salah.

1. **Menulis Kode Program di VSCode**

* Menulis kode dalam C++ menggunakan PlatformIO untuk membaca input tombol dan mengontrol relay serta LED.
* #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
* }
* }

1. **Konfigurasi PlatformIO**

* Edit file platformio.ini dengan isi berikut:
* ; PlatformIO Project Configuration File
* ;
* ;   Build options: build flags, source filter
* ;   Upload options: custom upload port, speed and extra flags
* ;   Library options: dependencies, extra library storages
* ;   Advanced options: extra scripting
* ;
* ; Please visit documentation for the other options and examples
* ; https://docs.platformio.org/page/projectconf.html
* [env:esp32doit-devkit-v1]
* platform = espressif32
* board = esp32doit-devkit-v1
* framework = arduino

1. **Menyusun File Konfigurasi Wokwi**

* Buat file wokwi.toml dengan isi berikut:

**[wokwi]**

**version = 1**

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

**elf = '.pio\build\esp32doit-devkit-v1\firmware.elf'**

1. **Kompilasi dan Pembuatan Firmware**

* Lakukan kompilasi (build) pada main.cpp menggunakan PlatformIO.

Sebuah gambar berisi teks, cuplikan layar, Font, nomor

Konten yang dihasilkan AI mungkin salah.

* Setelah berhasil, akan dihasilkan **firmware.bin** dan **firmware.elf.**

1. **Simulasi dan Pengujian**

* Request license di wokwi.com.

Masuk ke di command palette -> Request a New License

Sebuah gambar berisi teks, cuplikan layar, Software multimedia, software

Konten yang dihasilkan AI mungkin salah.

-> open -> Get Your License

Sebuah gambar berisi teks, cuplikan layar, Font, Software multimedia

Konten yang dihasilkan AI mungkin salah.

* Jalankan simulasi dengan mengetik perintah **wokwi simulator** di command palette.

Sebuah gambar berisi teks, cuplikan layar, Software multimedia, software

Konten yang dihasilkan AI mungkin salah.

Verifikasi bahwa **relay dan LED** menyala saat tombol ditekan dan mati saat tombol dilepas.**Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.**

1. **Analisis dan Peningkatan**

* Debug dan perbaiki kode jika hasil simulasi tidak sesuai dengan yang diharapkan.
* Pertimbangkan peningkatan, seperti penambahan indikator status pada serial monitor atau pengendalian jarak jauh menggunakan IoT Cloud.

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

**3.1 Experimental Results (Hasil Eksperimen)**

Simulasi sistem Relay, Button, dan LED menggunakan ESP32 di Wokwi Simulator telah berhasil diimplementasikan. Proses kompilasi dan build menggunakan PlatformIO di Visual Studio Code (VSCode) menunjukkan hasil berhasil (SUCCESS) tanpa error.

**Hasil pengujian menunjukkan bahwa:**

* Tombol (Button) dapat mengontrol Relay dan LED sesuai dengan logika yang telah diprogram.
* Relay dapat beralih antara keadaan ON dan OFF ketika tombol ditekan.
* LED menyala ketika Relay dalam keadaan aktif dan mati ketika Relay tidak aktif.
* ESP32 mampu membaca input dari tombol dan memberikan output ke Relay dan LED dengan respons yang cepat dan akurat.

**Berikut adalah rata-rata hasil simulasi:**

* Respon tombol: Deteksi 100% sukses dalam simulasi.
* Relay switching: Berfungsi sesuai dengan kondisi yang diprogram.
* LED indikator: Menyala dan mati mengikuti status Relay.

**3.2 Discussion (Pembahasan)**

Keberhasilan eksperimen ini menunjukkan bahwa ESP32 dapat digunakan sebagai mikrokontroler untuk mengendalikan perangkat eksternal seperti Relay dan LED melalui tombol input. Simulasi di Wokwi memungkinkan pengujian kode tanpa perangkat fisik, sehingga mempermudah proses debugging dan pengembangan.

**Faktor yang Mempengaruhi Hasil:**

* Logika pemrograman: Jika kode tidak benar, Relay mungkin tidak merespons dengan baik.
* Konfigurasi Pin: Pin yang digunakan untuk tombol, Relay, dan LED harus sesuai dengan yang diprogram.
* Delay dan debounce tombol: Jika tombol ditekan terlalu cepat tanpa debounce, mungkin terjadi kesalahan pembacaan input.

**Peningkatan yang Dapat Dilakukan:**

* Menambahkan debounce pada tombol untuk menghindari kesalahan input.
* Mengintegrasikan sistem dengan konektivitas IoT agar Relay dan LED dapat dikontrol melalui jaringan Wi-Fi menggunakan ESP32.
* Menggunakan indikator tambahan seperti buzzer untuk memberikan umpan balik suara ketika tombol ditekan.

Hasil eksperimen ini dapat menjadi dasar bagi pengembangan sistem otomatisasi yang lebih kompleks, seperti kendali perangkat listrik jarak jauh atau otomasi rumah pintar (smart home automation).

**4. Appendix (Lampiran)**

**Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.**