

LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

Latihan 3: Praktik Simulasi Relay, Button, LED

Author(s) (Najwa Firdaus Azkiyah)
Fakultas Vokasi, Universitas Brawijaya
Email: najwazkiyah@student.ub.ac.id

Abstract (Abstrak)

Praktik simulasi ini memanfaatkan platform Wokwi untuk mengemulasi dan memprogram mikrokontroler ESP32 dengan komponen dasar seperti relay, tombol (button), dan LED. Wokwi, sebuah simulator berbasis web, memungkinkan pengguna untuk merancang dan menguji rangkaian elektronik secara virtual tanpa memerlukan perangkat keras fisik. Dalam simulasi ini, relay digunakan sebagai saklar elektromekanis untuk mengendalikan rangkaian dengan arus yang lebih tinggi, sedangkan tombol berfungsi sebagai input untuk memicu perubahan status pada relay. LED digunakan sebagai indikator visual untuk menunjukkan status aktif atau tidak aktif dari relay. Melalui simulasi ini, pengguna dapat mempelajari prinsip kerja komponen elektronik tersebut, menulis kode untuk ESP32, dan menguji logika rangkaian sebelum implementasi di dunia nyata. Praktik ini bertujuan untuk meningkatkan pemahaman pengguna tentang integrasi perangkat keras dan pemrograman mikrokontroler, serta memberikan pengalaman hands-on dalam pengembangan proyek IoT.

Keywords—Internet of Things, ESP32, relay, button, LED

1. Introduction (Pendahuluan)

1.1 Latar belakang

Simulasi komponen elektronik seperti relay, tombol (button), dan LED menggunakan mikrokontroler ESP32 adalah langkah penting dalam proses pembelajaran dan pengembangan proyek Internet of Things (IoT). ESP32 adalah mikrokontroler yang populer karena kemampuannya yang kuat, seperti konektivitas Wi-Fi dan Bluetooth, serta berbagai pin input/output. Wokwi menyediakan platform simulator berbasis web yang memungkinkan pengguna untuk merancang dan menguji rangkaian elektronik secara virtual tanpa membutuhkan perangkat keras fisik. Dengan memanfaatkan Wokwi, pengguna dapat memperoleh pemahaman yang mendalam tentang fungsi dan integrasi komponen elektronik dalam sistem mikrokontroler sebelum diimplementasikan dalam proyek nyata.

1.2 Tujuan eksperimen

- **Memahami Prinsip Kerja Relay:** Memperoleh pemahaman tentang bagaimana relay bekerja sebagai saklar elektromekanis untuk mengendalikan rangkaian berarus tinggi dengan menggunakan ESP32.
- **Mengembangkan Keterampilan Pemrograman:** Meningkatkan keterampilan pemrograman mikrokontroler ESP32 dengan menulis kode yang mengintegrasikan relay, tombol, dan LED.
- **Menguji Logika Rangkaian:** Melakukan pengujian logika rangkaian dan memastikan bahwa relay, tombol, dan LED berfungsi sebagaimana mestinya dalam simulasi.
- **Menyediakan Pengalaman Hands-On:** Memberikan pengalaman praktis dalam merancang, menyimulasikan, dan memecahkan masalah rangkaian elektronik melalui platform Wokwi.
- **Meningkatkan Efisiensi Pengembangan Proyek:** Menggunakan simulasi sebagai alat bantu untuk mengoptimalkan desain rangkaian dan meminimalkan kesalahan sebelum implementasi pada perangkat keras fisik.

2. Methodology (Metodologi)

2.1 Tools & Materials (Alat dan Bahan)

> Mikrokontroler (ESP8266, Arduino, Raspberry Pi, dll.), sensor (DHT22, PIR, dsb.), software (Arduino IDE, MQTT Broker, dsb.), LED, relay, button

2.2 Implementation Steps (Langkah Implementasi)

Berikut adalah langkah-langkah implementasi praktis untuk simulasi relay, tombol (button), dan LED menggunakan ESP32 di platform Wokwi:

1. ****Persiapan Lingkungan Simulasi:****

- Buka [Wokwi](<https://wokwi.com/>) di peramban web Anda.

- Pilih **New Project** dan pilih **ESP32** sebagai mikrokontroler yang akan digunakan.

2. **Penambahan Komponen:**

- Tambahkan komponen **relay**, **button**, dan **LED** dari pustaka komponen yang tersedia di Wokwi.
- Atur posisi komponen di canvas dan hubungkan pin-pin komponen sesuai dengan skema rangkaian.

3. **Koneksi Komponen:**

- Sambungkan salah satu ujung button ke pin GPIO ESP32 dan ujung lainnya ke ground.
- Sambungkan salah satu ujung relay ke pin GPIO ESP32 dan ujung lainnya ke ground.
- Sambungkan kaki negatif LED ke ground dan kaki positif ke relay.

4. **Pemrograman ESP32:**

- Klik **Code** di Wokwi untuk membuka editor kode.
- Tulis kode untuk mengatur pin GPIO sebagai input untuk button dan output untuk relay dan LED. Contoh kode selengkapnya:

```
#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. ****Simulasi:****

- Klik ****Start Simulation**** untuk menjalankan simulasi dan mengamati cara kerja rangkaian.
- Tekan button di simulasi untuk melihat bagaimana relay mengaktifkan dan mematikan LED.

6. ****Troubleshooting dan Optimasi:****

- Uji dan pantau rangkaian untuk memastikan semuanya berfungsi dengan baik.
- Lakukan penyesuaian pada rangkaian atau kode jika diperlukan untuk mencapai hasil yang diinginkan.

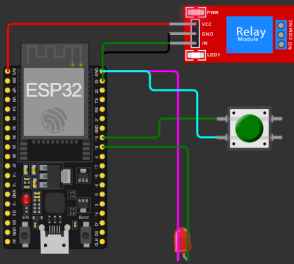
Langkah-langkah ini memberikan panduan singkat dan efektif untuk memahami dan mengimplementasikan simulasi relay, tombol, dan LED menggunakan Wokwi dengan ESP32.

3. Results and Discussion (Hasil dan Pembahasan)

3.1 Experimental Results (Hasil Eksperimen)

WOKWI Simulator

00:05.300 31%



main.cpp relay_project · src wokwi.toml relay_project diagram.json relay_project Wokwi Simulator PIO Home pl ✓ 31%

EXPLORE UNTITLED (WORKSPACE) relay_project .pio\build esp32doit-devkit-v1 > src .sconsign311.dblite bootloader.bin firmware.bin firmware.elf firmware.map idedata.json libFrameworkArd... partitions.bin project.checksum .vscode > include > lib > src main.cpp > test .gitignore diagram.json platformio.ini wokwi.toml OUTLINE TIMELINE

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

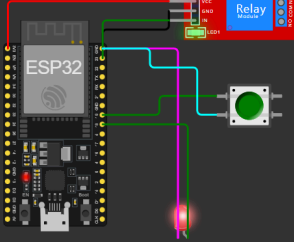
```
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:1156
load:0x40078000,len:11456
ho 0 tail 12 room 4
load:0x40080400,len:2972
entry 0x400805dc
```

powershell Wokwi Term...

Type here to search 1:08 PM 3/3/2025

WOKWI Simulator

00:02.400 6%



main.cpp relay_project · src wokwi.toml relay_project diagram.json relay_project Wokwi Simulator PIO Home pl ✓ 6%

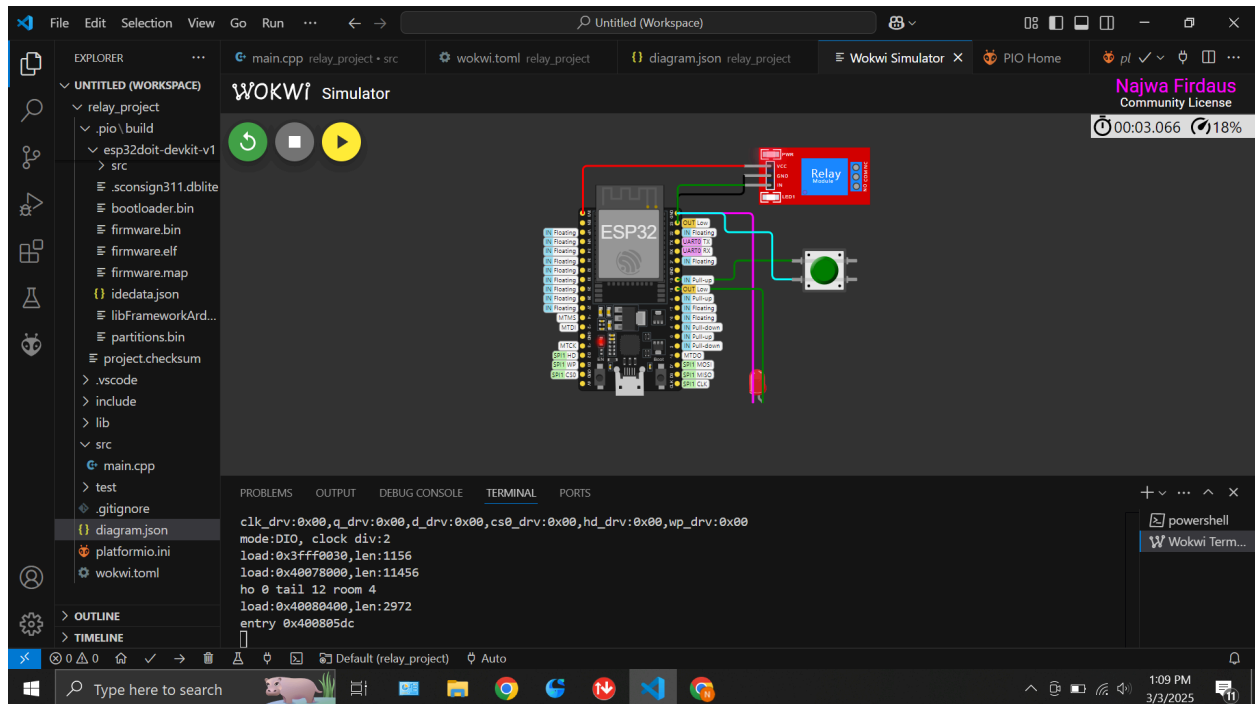
EXPLORE UNTITLED (WORKSPACE) relay_project .pio\build esp32doit-devkit-v1 > src .sconsign311.dblite bootloader.bin firmware.bin firmware.elf firmware.map idedata.json libFrameworkArd... partitions.bin project.checksum .vscode > include > lib > src main.cpp > test .gitignore diagram.json platformio.ini wokwi.toml OUTLINE TIMELINE

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:1156
load:0x40078000,len:11456
ho 0 tail 12 room 4
load:0x40080400,len:2972
entry 0x400805dc
```

powershell Wokwi Term...

Type here to search 1:09 PM 3/3/2025



4. Appendix (Lampiran, jika diperlukan)

> sketch.ino / main.cpp

```
#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
    }
}

```

>diagram.json

```

{
  "version": 1,
  "author": "Najwa Firdaus",
  "editor": "wokwi",
  "parts": [
    {
      "type": "board-esp32-devkit-c-v4",
      "id": "esp",
      "top": -115.2,
      "left": -129.56,
      "attrs": {}
    },
    { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -200.1,
"left": 82.3, "attrs": {} }
  ],
  "connections": [

```

```
[ "esp:TX", "$serialMonitor:RX", "", [ ] ],  
[ "esp:RX", "$serialMonitor:TX", "", [ ] ],  
[ "ultrasonic1:GND", "esp:GND.2", "black", [ "v0" ] ],  
[ "ultrasonic1:TRIG", "esp:19", "yellow", [ "v0" ] ],  
[ "ultrasonic1:ECHO", "esp:18", "green", [ "v0" ] ],  
[ "ultrasonic1:VCC", "esp:3V3", "red", [ "v48", "h-278.55" ] ]  
],  
"dependencies": {}  
}
```