`

LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Simulasi ESP32 & Sensor Suhu Kelembaban**

*Abdillah Jibran - 233140700111046*

*Fakultas Vokasi, Universitas Brawijaya*

*Email: abdillahjibran12@gmail.com*

**Abstract** (Abstrak)

Kegiatan ini bertujuan untuk mensimulasikan sistem berbasis ESP32 dengan modul relay dan push button di Wokwi menggunakan pemrograman C++ di Visual Studio Code. Simulasi ini memungkinkan pengujian pengendalian relay secara real-time, di mana tombol digunakan sebagai pemicu untuk mengaktifkan dan menonaktifkan relay, serta LED sebagai indikator. Hasil simulasi menunjukkan bahwa ESP32 dapat mengontrol relay dengan respons yang baik terhadap input dari push button. Selain itu, Wokwi terbukti efektif sebagai alat pengujian sebelum implementasi dengan perangkat keras nyata. Kesimpulannya, ESP32 dapat digunakan sebagai pengontrol sistem relay, dan simulasi di Wokwi memungkinkan validasi kode sebelum pengujian fisik. Pengembangan lebih lanjut dapat mencakup integrasi dengan sistem otomatisasi berbasis IoT untuk kendali jarak jauh.

**Kata kunci**: ESP32, relay, Wokwi, Visual Studio Code, simulasi, IoT, push button, LED

**1. Pendahuluan**

**1.1 Latar belakang**

Dalam era digital, simulasi berbasis perangkat lunak menjadi solusi efektif untuk mengembangkan dan menguji sistem sebelum diimplementasikan dalam bentuk fisik. Wokwi merupakan salah satu simulator berbasis web yang memungkinkan pengguna untuk merancang, memprogram, dan menguji sistem berbasis mikrokontroler tanpa perlu perangkat keras nyata.

Relay merupakan komponen penting dalam sistem otomatisasi, memungkinkan pengendalian perangkat listrik menggunakan sinyal elektronik. Pada eksperimen ini, ESP32 digunakan untuk mengontrol relay, yang dipicu oleh push button dan diindikasikan dengan LED. Penggunaan simulasi di Wokwi memungkinkan validasi logika program sebelum diimplementasikan ke perangkat keras nyata.

**1.2 Tujuan eksperimen**

1. Mendesain dan mensimulasikan sistem sensosr suhu dan kelembapan menggunakan mikrokontroler ESP32 di Wokwi.
2. Mengimplementasikan program berbasis C++ untuk mengontrol relay dan LED berdasarkan input dari push button..

**2. Methodology (Metodologi)**

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

Alat dan Perangkat Lunak:

1. Wokwi Simulator – Platform berbasis web untuk simulasi mikrokontroler dan rangkaian elektronik.
2. Visual Studio Code – Editor kode untuk menulis dan mengunggah program ke ESP32.
3. Arduino Framework – Digunakan sebagai dasar pemrograman mikrokontroler ESP32 dalam bahasa C++.

Bahan (Simulasi Komponen Elektronik):

Bahan (Simulasi Komponen Elektronik):

1. ESP32 – Mikrokontroler yang digunakan sebagai pengendali utama dalam simulasi.
2. Relay Module – Digunakan untuk mengontrol perangkat eksternal berdasarkan input dari ESP32.
3. Push Button – Digunakan untuk mengaktifkan dan menonaktifkan relay.
4. LED – Sebagai indikator saat relay aktif atau tidak.
   1. **Implementation Steps (Langkah Implementasi)**

1. Perancangan Rangkaian di Wokwi

1. Membuka platform Wokwi dan memilih ESP32 sebagai mikrokontroler.
2. Menambahkan Relay Module, Push Button, dan LED ke dalam desain rangkaian.
3. Menghubungkan komponen sesuai skema berikut:

* Relay Module
  + VCC → 5V (ESP32)
  + GND → GND (ESP32)
  + IN → GPIO 23 (ESP32)
* Push Button
  + Satu kaki ke GND (ESP32)
  + Satu kaki ke GPIO 21 (ESP32)
* LED
  + Anoda (+) ke GPIO 22 (ESP32)
  + Katoda (-) ke GND (ESP32)

2. Pemrograman Sistem rlay

1. Membuka Visual Studio Code dan menulis kode dalam bahasa C++ menggunakan Arduino framework.
2. Mendefinisikan pin untuk relay, LED, dan push button, serta mengatur pin tersebut sebagai output dan input pada fungsi setup().
3. Menulis logika pengendalian relay dan LED dalam fungsi loop(), sehingga ketika tombol ditekan, relay aktif dan LED menyala, serta sebaliknya..

3. Pengujian Simulasi

1. Menjalankan program di Wokwi untuk melihat apakah program relay bekerja dnegan baik
2. Jika ditemukan kesalahan maka melakukan debugging dan perbaikan pada kode program.

**3. Hasil dan Pembahasan**

**3.1 Hasil Eksperimen**

1. Rangkaian Berhasil Dijalankan

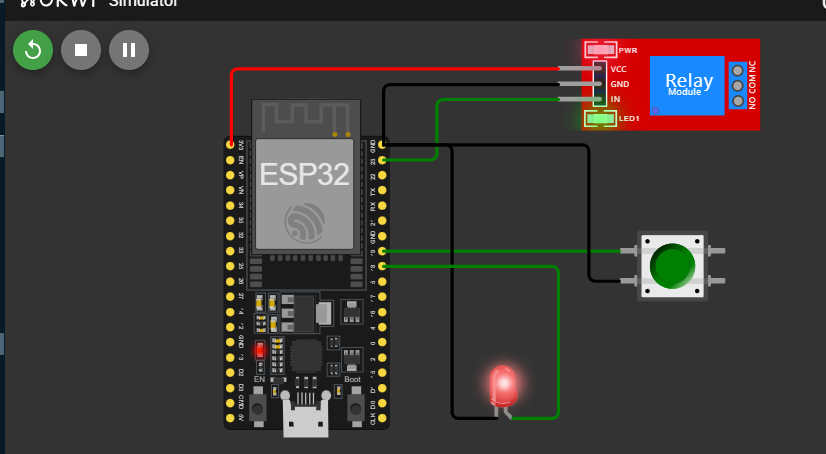
1. Rangkaian yang telah dirancang di Wokwi bekerja sesuai dengan desain, di mana tombol dapat mengontrol relay dan LED dengan baik.
2. ESP32 mampu mengontrol relay dan LED menggunakan fungsi digitalWrite() sesuai dengan logika yang telah diprogram.

2. Urutan Operasi Sesuai dengan Konsep Kontrol Relay

1. Saat tombol ditekan, relay aktif dan LED menyala sebagai indikator.
2. Saat tombol dilepas, relay mati dan LED padam.
3. Proses ini berjalan terus-menerus dalam loop yang telah diprogram.

3. Hasil Simulasi di Wokwi

1. Tidak ditemukan error atau kesalahan dalam eksekusi kode.
2. Serial monitor menampilkan pesan "Hello, ESP32!" sebagai indikasi bahwa sistem berjalan dengan baik.



**4. Lampiran**

Kode Program

*#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*

  }

}