

# LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

## **Praktik Simulasi Relay, Button & LED Serta Praktik Simulasi Sensor Jarak (Ultrasonic)**

*Salsabilla Permata Dewi - 233140701111016*

*Fakultas Vokasi, Universitas Brawijaya*

*Email: salsabillapermata02@gamil.com*

### **Abstract (Abstrak)**

Tujuan dari praktik ini adalah untuk mempelajari penggunaan relay, button, LED, dan sensor jarak (ultrasonik) dengan ESP32 melalui simulasi di platform Wokwi. Dalam simulasi, button digunakan sebagai input untuk mengontrol kondisi relay, yang kemudian mengaktifkan atau mematikan LED sebagai indikator. Selain itu, sensor ultrasonik HC-SR04 digunakan untuk mengukur jarak dan menampilkan hasilnya di serial monitor. ESP32 memproses sinyal dari button dan sensor, lalu mengendalikan relay serta LED sesuai dengan logika yang telah diprogram. Hasil simulasi menunjukkan bahwa ESP32 dapat membaca input dari button dan sensor, serta memproses sinyal relay dan LED dengan akurat. Wokwi memungkinkan pengujian tanpa perangkat fisik dan memberikan pemahaman dasar tentang integrasi komponen elektronik dengan mikrokontroler.

*Kata kunci— ESP32, Relay, Button, LED, Ultrasonik, Simulasi Wokwi, Mikrokontroler.*

## **1. Pendahuluan**

### **1.1. Latar Belakang**

Dalam dunia elektronika dan otomasi, relay digunakan sebagai saklar elektronik yang memungkinkan pengendalian perangkat dengan arus tinggi menggunakan sinyal berarus rendah dari mikrokontroler seperti ESP32. Button berfungsi sebagai input untuk memberikan perintah kepada sistem, sementara LED digunakan sebagai indikator perubahan status. Sensor ultrasonik HC-SR04 digunakan untuk mengukur jarak dengan memanfaatkan gelombang ultrasonik, memberikan kemampuan pengukuran jarak secara real-time.

Pemahaman tentang integrasi komponen ini sangat penting dalam pengembangan sistem otomatisasi, seperti sistem kendali rumah pintar dan industri. Namun, pengujian perangkat keras secara langsung dapat memerlukan biaya dan sumber daya tambahan. Oleh karena itu, simulasi berbasis platform seperti Wokwi menjadi solusi praktis untuk memahami konsep dasar dan menguji fungsionalitas sistem sebelum implementasi pada perangkat fisik. Dengan simulasi ini, pengguna dapat mempelajari bagaimana button digunakan sebagai pemicu, relay sebagai aktuator, LED sebagai indikator, dan sensor ultrasonik sebagai pengukur jarak dalam suatu rangkaian elektronik yang dikendalikan oleh ESP32.

### **1.2. Tujuan eksperimen**

Beberapa tujuan dari eksperimen ini adalah sebagai berikut:

- Memahami cara kerja relay, button, LED, dan sensor ultrasonik dalam sistem berbasis ESP32.
- Menguji integrasi keempat komponen tersebut melalui simulasi di Wokwi.

- Memastikan ESP32 dapat membaca input dari button dan sensor ultrasonik serta mengontrol relay dan LED dengan logika yang sesuai.
- Memberikan pengalaman dalam pemrograman mikrokontroler untuk mengendalikan aktuator secara efisien.
- Memvalidasi hasil simulasi sebagai dasar sebelum implementasi pada perangkat fisik.

## 2. Metodologi

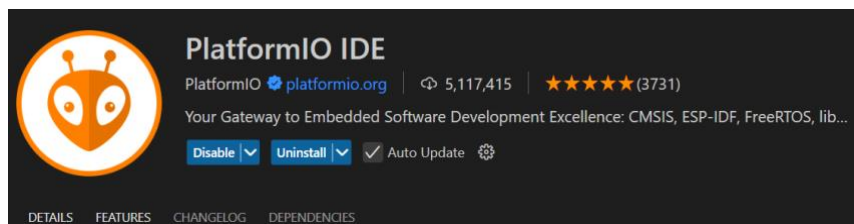
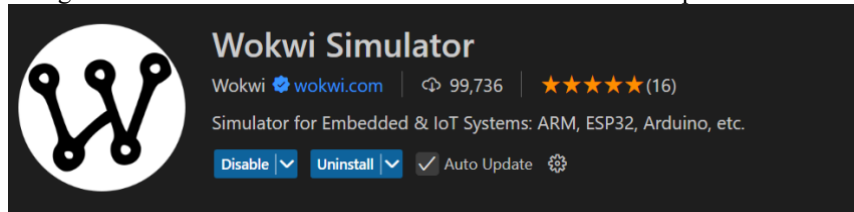
### 2.1. Alat dan Bahan

Alat dan bahan yang digunakan dalam eksperimen ini adalah:

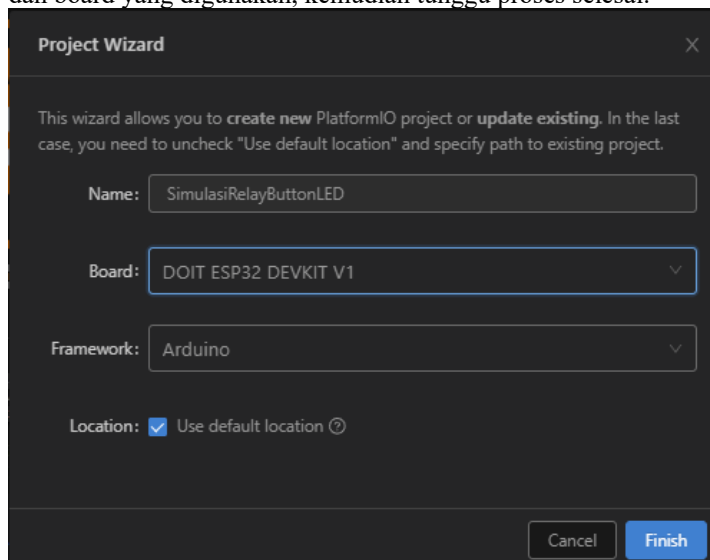
- Laptop
- Koneksi internet
- Visual Studio Code
- Mikrokontroler ESP32
- Relay
- Button
- LED
- Sensor Ultrasonik HC-SR04 (dalam simulator)
- Extension di Visual Studio Code berupa Wokwi Simulator dan PlatformIO IDE

### 2.2. Langkah Implementasi

1. Meng-install extension Wokwi Simulator dan PlatformIO IDE pada Visual Studio Code.



2. Membuat project baru untuk simulasi relay pada PlatformIO IDE dengan mengisi nama project dan board yang digunakan, kemudian tunggu proses selesai.



3. Menuliskan kode program untuk relay, button, dan LED pada file **src/main.cpp** (jika sudah membuat kode pada wokwi.com bisa langsung dicopy-paste).

```
#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
  }
}
```

4. Build kode program yang sudah selesai dan tunggu sampai success.

```
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM:  [= ] 6.4% (used 21112 bytes from 327680 bytes)
Flash: [= ] 18.3% (used 239949 bytes from 1310720 bytes)
Building .pio\build\esp32doit-devkit-v1\firmware.bin
esptool.py v4.5.1
Creating esp32 image...
Merged 2 ELF sections
Successfully created esp32 image.
===== [SUCCESS] Took 103.55 seconds =====
Terminal will be reused by tasks, press any key to close it.
```

5. Buat file baru bernama wokwi.toml dan kemudian ketik seperti berikut (bisa copy-paste dari file wokwi.toml di project sebelumnya).

```
wokwi.toml
1 [wokwi]
2 version = 1
3 firmware = '.pio\build\esp32doit-devkit-v1\firmware.bin'
4 elf = '.pio\build\esp32doit-devkit-v1\firmware.elf'
5
```

6. Buat file diagram.json, lalu copy kode diagram.json dari wokwi.com dan paste pada file diagram.json pada Visual Studio Code, setelah itu tutup dan buka lagi file diagram.json untuk melihat simulasi.

```

{
  "version": 1,
  "author": "Anonymous maker",
  "editor": "wokwi",
  "parts": [
    { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 0, "left": 0,
      "attrs": {} },
    { "type": "wokwi-relay-module", "id": "relay1", "top": -124.6, "left":
240, "attrs": {} },
    {
      "type": "wokwi-led",
      "id": "led1",
      "top": 159.6,
      "left": 157.4,
      "attrs": { "color": "red" }
    },
    {
      "type": "wokwi-pushbutton",
      "id": "btn1",
      "top": 83,
      "left": 230.4,
      "attrs": { "color": "green", "xray": "1" }
    }
  ],
  "connections": [
    [ "esp:TX", "$serialMonitor:RX", "", [] ],
    [ "esp:RX", "$serialMonitor:TX", "", [] ],
    [ "esp:GND.2", "relay1:GND", "black", [ "v0" ] ],
    [ "esp:23", "relay1:IN", "green", [ "h24.04", "v-124.6" ] ],
    [ "esp:GND.2", "led1:C", "green", [ "h43.24", "v172.8" ] ],
    [ "esp:18", "led1:A", "green", [ "h100.84", "v96" ] ],
    [ "btn1:2.L", "esp:GND.2", "green", [ "h-19.2", "v-67" ] ],
    [ "esp:19", "btn1:1.L", "green", [ "v-19.2", "h100.84" ] ],
    [ "esp:3V3", "relay1:VCC", "red", [ "h-23.81", "v-134.4" ] ]
  ],
  "dependencies": {}
}

```

7. Buat project untuk simulasi sensor jarak (Ultrasonic) pada PlatformIO IDE

Project Wizard

This wizard allows you to **create new** PlatformIO project or **update existing**. In the last case, you need to uncheck "Use default location" and specify path to existing project.

Name:

Board:

Framework:

Location: ☒ Use default location ?

Cancel

Finish

8. Menuliskan kode program untuk sensor jarak pada file src/main.cpp

```
#include <Arduino.h>

const int trigPin = 5;
const int echoPin = 18;

//define sound speed in cm/uS
#define SOUND_SPEED 0.034
#define CM_TO_INCH 0.393701

long duration;
float distanceCm;
float distanceInch;

void setup() {
  Serial.begin(115200); // Starts the serial communication
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
}

void loop() {
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculate the distance
  distanceCm = duration * SOUND_SPEED/2;
  // Convert to inches
  distanceInch = distanceCm * CM_TO_INCH;
  // Prints the distance in the Serial Monitor
  Serial.print("Distance (cm): ");
  Serial.println(distanceCm);
  // Serial.print("Distance (inch): ");
  // Serial.println(distanceInch);
  delay(1000);
}
```

9. Build kode program yang sudah selesai dan tunggu sampai success.

```
Building in release mode
Retrieving maximum program size .pio\build\esp32doit-devkit-v1\firmware.elf
Checking size .pio\build\esp32doit-devkit-v1\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [ = ] 6.6% (used 21472 bytes from 327680 bytes)
Flash: [ == ] 20.6% (used 270137 bytes from 1310720 bytes)
===== [SUCCESS] Took 21.71 seconds =====
* Terminal will be reused by tasks, press any key to close it.
```

10. Buat file baru bernama wokwi.toml dan kemudian copy-paste dari file sebelumnya

```
wokwi.toml
1 [wokwi]
2 version = 1
3 firmware = '.pio\build\esp32doit-devkit-v1\firmware.bin'
4 elf = '.pio\build\esp32doit-devkit-v1\firmware.elf'
5
```

11. Buat file diagram.json, lalu copy kode diagram.json dari wokwi.com dan paste pada file diagram.json pada Visual Studio Code, setelah itu tutup dan buka lagi file diagram.json untuk melihat simulasi.

```

{
  "version": 1,
  "author": "Uri Shaked",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -4.9, "left":
43, "attrs": {} },
    {
      "type": "wokwi-hc-sr04",
      "id": "ultrasonic1",
      "top": 33.9,
      "left": 202.5,
      "rotate": 90,
      "attrs": {}
    }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [] ],
    [ "ultrasonic1:VCC", "esp:VIN", "red", [ "h-48", "v163.2", "h-144" ] ],
    [ "ultrasonic1:TRIG", "esp:D5", "green", [ "h0" ] ],
    [ "ultrasonic1:ECHO", "esp:D18", "green", [ "h0" ] ],
    [ "esp:GND.1", "ultrasonic1:GND", "black", [ "h28.5", "v-56.5" ] ]
  ],
  "dependencies": {}
}

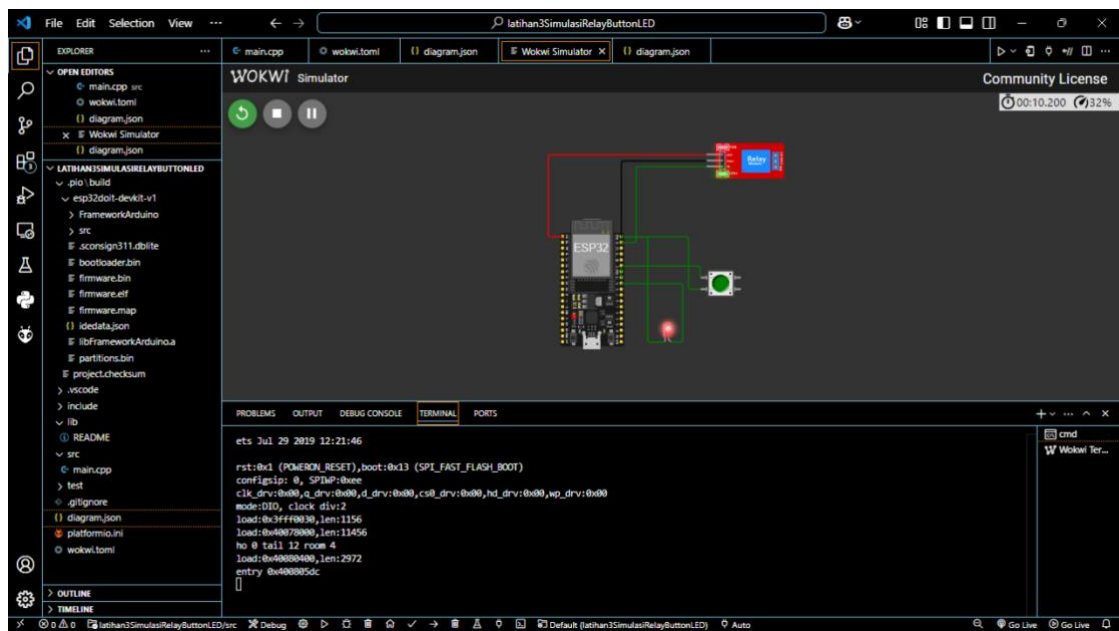
```

12. Jalankan simulasi masing-masing project

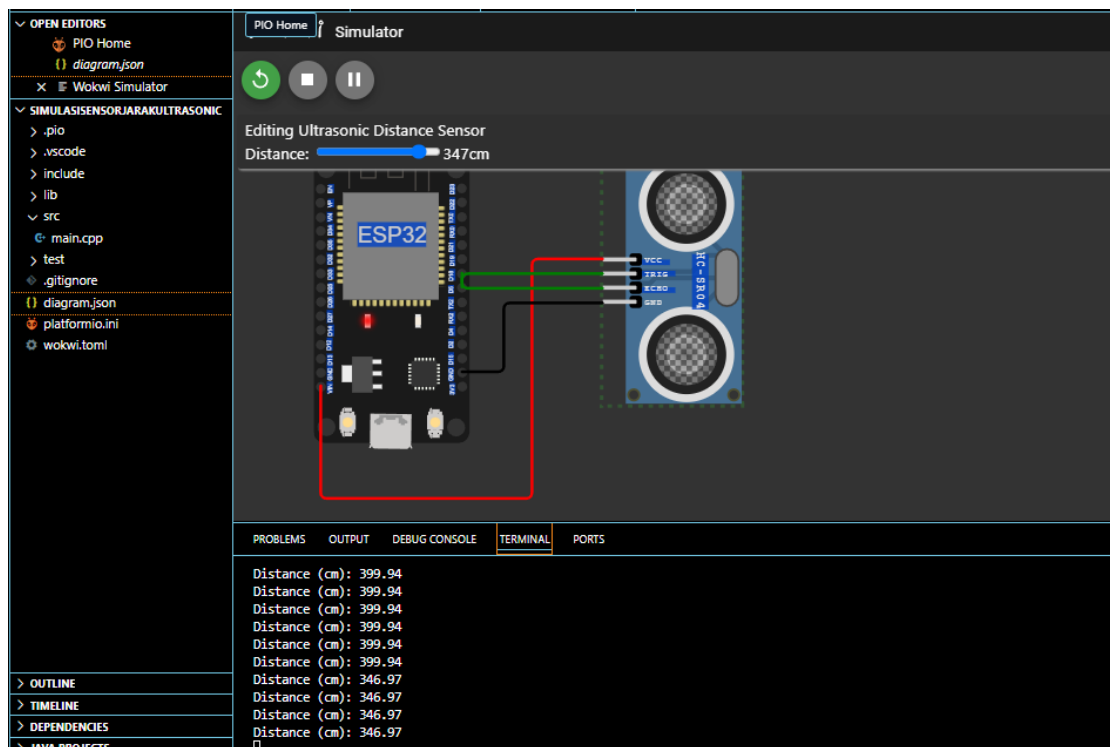
### 3. Hasil dan Pembahasan

#### 3.1 Hasil Eksperimen

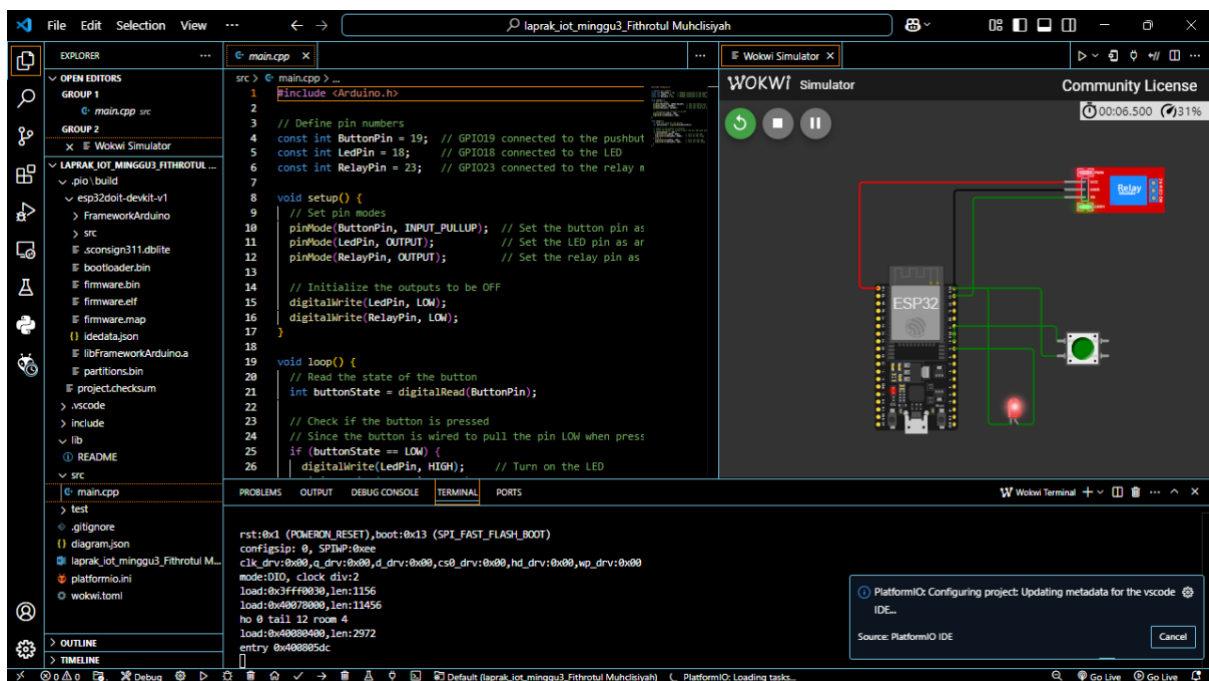
Tampilan simulasi relay, *button*, dan led pada file diagram.json:



Tampilan simulasi sensor jarak pada file diagram.json:



#### 4. Appendix



SimulasiSensorJarakUltrasonic

WOKWI Simulator Community License

Editing Ultrasonic Distance Sensor  
Distance: 347cm

```
src > main.cpp > ...
1 #include <Arduino.h>
2
3
4 const int trigPin = 5;
5 const int echoPin = 18;
6
7
8 //define sound speed in cm/uS
9 #define SOUND_SPEED 0.034
10 #define CM_TO_INCH 0.393701
11
12 long duration;
13 float distanceCm;
14 float distanceInch;
15
16
17
18 void setup() {
19   Serial.begin(115200); // Starts the serial communication
20   pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
21   pinMode(echoPin, INPUT); // Sets the echoPin as an Input
22 }
23
24
25 void loop() {
26   // Clears the trigPin
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Distance (cm): 346.97  
Distance (cm): 346.97  
Distance (cm): 346.97  
Distance (cm): 346.97  
Distance (cm): 346.94  
Distance (cm): 346.97  
Distance (cm): 346.97  
Distance (cm): 346.97  
Distance (cm): 346.97  
Distance (cm): 346.97

Build Task ✓  
Wokwi Ter...