

LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

Bab 10

Praktik Simulasi Relay, Button & LED

Adi Prasetyo

233140707111011

Fakultas Vokasi, Universitas Brawijaya

student@tomi.engineer

Contents

1	Introduction (Pendahuluan)	3
1.1	Latar belakang praktikum IoT yang dilakukan	3
1.2	Tujuan eksperimen	3
2	Methodology (Metodologi)	3
2.1	Tools & Materials (Alat dan Bahan)	3
2.2	Implementation Steps (Langkah Implementasi)	4
3	Results and Discussion (Hasil dan Pembahasan)	4
3.1	Experimental Results (Hasil Eksperimen)	4
4	Appendix (Lampiran)	5
4.1	Kode Program	5
4.2	Konfigurasi Wokwi	6

Abstract

Proyek ini merupakan implementasi sistem kontrol sederhana menggunakan tombol (pushbutton), LED, dan modul relay yang dihubungkan ke mikrokontroler ESP32. Sistem dirancang untuk mendemonstrasikan prinsip input-output dasar dalam IoT, di mana penekanan tombol (input) mengaktifkan LED dan relay (output) secara bersamaan. Sistem menggunakan resistor pull-up internal ESP32 untuk tombol dan mengatur pin GPIO sebagai output untuk LED dan relay. Proyek ini mendemonstrasikan konsep dasar Internet of Things (IoT) dalam mengontrol perangkat elektronik sederhana melalui antarmuka fisik.

Keywords—Internet of Things, Button Interface, LED Indicator, Relay Control, ESP32, Digital I/O

1. Introduction (Pendahuluan)

1.1 Latar belakang praktikum IoT yang dilakukan

Kontrol perangkat elektronik melalui antarmuka fisik merupakan aspek fundamental dalam aplikasi IoT seperti rumah pintar, otomasi industri, dan sistem keamanan. Kemampuan untuk membaca input dari pengguna (seperti melalui tombol) dan mengendalikan aktuator (seperti LED dan relay) adalah keterampilan dasar yang diperlukan dalam pengembangan sistem IoT. Relay khususnya memiliki peran penting karena memungkinkan kontrol perangkat dengan daya tinggi menggunakan sinyal kontrol berdaya rendah dari mikrokontroler.

Proyek ini bertujuan untuk memahami prinsip dasar pembacaan input digital dan pengendalian output pada mikrokontroler. Melalui simulasi sederhana ini, mahasiswa dapat mempelajari bagaimana teknologi IoT dapat diimplementasikan untuk sistem kontrol manual yang dapat dikembangkan menjadi sistem otomasi yang lebih kompleks.

1.2 Tujuan eksperimen

1. Memahami cara kerja tombol (pushbutton) sebagai perangkat input digital.
2. Mempelajari teknik interfacing LED dan modul relay dengan mikrokontroler ESP32.
3. Mengimplementasikan resistor pull-up internal untuk pembacaan input.
4. Memahami konsep dasar pengontrolan aktuator melalui input manual.
5. Menguji simulasi rangkaian elektronik secara virtual melalui platform Wokwi.

2. Methodology (Metodologi)

2.1 Tools & Materials (Alat dan Bahan)

1. Mikrokontroler: ESP32 DevKit C V4
2. Input: Tombol pushbutton

3. Output: LED dan modul relay
4. Software: VS Code, Wokwi Simulator, PlatformIO

2.2 Implementation Steps (Langkah Implementasi)

1. Tombol pushbutton terhubung ke pin GPIO 19 ESP32 dengan resistor pull-up internal.
2. LED terhubung ke pin GPIO 18 ESP32, dengan anoda ke pin GPIO dan katoda ke ground.
3. Modul relay terhubung ke pin GPIO 23 ESP32, dengan VCC ke 3.3V dan GND ke ground.
4. Program dikembangkan untuk membaca status tombol dan mengontrol LED serta relay berdasarkan kondisi tombol.
5. Ketika tombol ditekan (keadaan LOW), LED dan relay diaktifkan. Ketika tombol dilepas, keduanya dinonaktifkan.
6. Pengujian dilakukan untuk memastikan sistem bekerja dengan benar, di mana tombol dapat mengontrol LED dan relay secara bersamaan.

3. Results and Discussion (Hasil dan Pembahasan)

3.1 Experimental Results (Hasil Eksperimen)

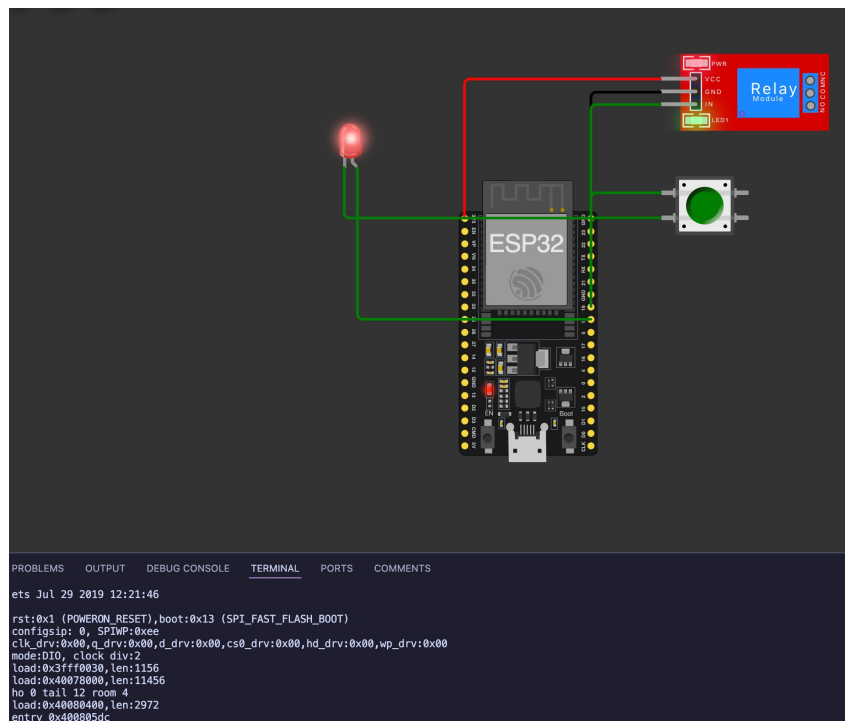


Figure 1: Hasil simulasi sistem kontrol Relay, Button & LED pada Wokwi Simulator.

Hasil simulasi menunjukkan beberapa temuan sebagai berikut:

- Sistem berhasil membaca input dari tombol pushbutton dan mengontrol LED serta relay sesuai dengan status tombol.
- Ketika tombol ditekan, LED menyala dan relay aktif yang ditandai dengan bunyi "klik" pada relay dan indikator pada modul relay menyala.
- Ketika tombol dilepas, LED mati dan relay kembali ke posisi nonaktif.
- Resistor pull-up internal pada ESP32 berfungsi dengan baik untuk memastikan pembacaan tombol yang konsisten tanpa memerlukan resistor eksternal.
- Respons sistem terhadap penekanan tombol bersifat instan, tanpa penundaan yang terlihat.
- Tidak terdapat efek bouncing yang signifikan pada tombol, sehingga tidak diperlukan implementasi debounce dalam program.
- Modul relay dapat diaktifkan dengan aman menggunakan sinyal 3.3V dari ESP32.

Meskipun menggunakan platform simulasi, sistem berjalan dengan baik dan memberikan respons yang cepat dan konsisten. Dalam implementasi nyata, sistem ini dapat digunakan untuk mengontrol berbagai perangkat elektronik seperti lampu, kipas, atau perangkat rumah tangga lainnya melalui relay.

4. Appendix (Lampiran)

Kode program dan diagram konfigurasi Wokwi untuk sistem kontrol Relay, Button & LED:

4.1 Kode Program

```

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 }

```

Listing 1: main.cpp

4.2 Konfigurasi Wokwi

```

1 {
2   "version": 1,
3   "author": "Tomi",
4   "editor": "wokwi",
5   "parts": [
6     {
7       "type": "board-esp32-devkit-c-v4",
8       "id": "esp",
9       "top": 0,
10      "left": 0,
11      "attrs": {}
12    },
13    {
14      "type": "wokwi-relay-module",
15      "id": "relay1",
16      "top": -95.8,
17      "left": 153.6,
18      "attrs": {}
19    },
20    {
21      "type": "wokwi-pushbutton",
22      "id": "btn1",
23      "top": -3.4,
24      "left": 153.6,
25      "attrs": {
26        "color": "green",
27        "xray": "1"
28      }
29    }
30  ]
31 }

```

```

29     },
30     {
31         "type": "wokwi-led",
32         "id": "led1",
33         "top": -51.6,
34         "left": -101.8,
35         "attrs": {
36             "color": "red"
37         }
38     }
39 ],
40 "connections": [
41     [
42         "esp:TX",
43         "$serialMonitor:RX",
44         "",
45         []
46     ],
47     [
48         "esp:RX",
49         "$serialMonitor:TX",
50         "",
51         []
52     ],
53     [
54         "relay1:VCC",
55         "esp:3V3",
56         "red",
57         [
58             "h0"
59         ]
60     ],
61     [
62         "relay1:GND",
63         "esp:GND.2",
64         "black",
65         [
66             "h0"
67         ]
68     ],
69     [
70         "relay1:IN",
71         "esp:23",
72         "green",
73         [
74             "h0"
75         ]

```

```

76     ],
77     [
78         "btn1:1.1",
79         "esp:19",
80         "green",
81         [
82             "h0"
83         ]
84     ],
85     [
86         "btn1:2.1",
87         "esp:GND.3",
88         "green",
89         [
90             "h0"
91         ]
92     ],
93     [
94         "led1:A",
95         "esp:18",
96         "green",
97         [
98             "v0"
99         ]
100    ],
101    [
102        "led1:C",
103        "esp:GND.2",
104        "green",
105        [
106            "v0"
107        ]
108    ]
109 ],
110 "dependencies": {}
111 }

```

Listing 2: diagram.json