

LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

**Praktik Akses API Melalui
Simulasi WOKWI**



Femas Alfaridzi

Fakultas Vokasi , Universitas Brawijaya

Email : femasalfaridzi17@gmail.com

**Fakultas Vokasi
Universitas Brawijaya**

Abstrak

Praktikum ini bertujuan untuk memahami dan mengimplementasikan akses Application Programming Interface (API) menggunakan simulasi Wokwi di Visual Studio Code (VSCode). Dengan menggunakan API, perangkat dapat berkomunikasi dan bertukar data dengan sistem eksternal.

Dalam simulasi ini, Wokwi digunakan untuk mensimulasikan perangkat mikrokontroler yang mengirim dan menerima data dari API melalui protokol HTTP. Pengujian yang dilakukan dengan mengintegrasikan kode pemrograman berbasis Arduino dan library yang mendukung koneksi ke API. Hasil praktik menunjukkan bahwa simulasi API pada Wokwi dapat berjalan dengan baik.

Keywords: Iot, HTTP, API

1. Pendahuluan

1.1 Latar Belakang

Dalam pengembangan sistem berbasis Internet of Things (IoT), penggunaan Application Programming Interface (API) sangat penting untuk memungkinkan perangkat berkomunikasi dengan server dan menyimpan data ke dalam database. Salah satu aspek utama dalam IoT adalah kemampuan perangkat untuk mendeteksi kondisi lingkungan dan mengirimkan data secara otomatis ke sistem pemrosesan.

Pada praktikum ini, dilakukan simulasi WiFi scanning dan pengukuran kelembapan menggunakan sensor, di mana data yang diperoleh akan dikirim ke server melalui API yang terhubung ke database. Wokwi digunakan sebagai simulator untuk menjalankan kode mikrokontroler secara virtual di Visual Studio Code (VSCode), sehingga memungkinkan pengujian tanpa memerlukan perangkat keras secara langsung.

Melalui praktikum ini, mahasiswa dapat memahami bagaimana perangkat IoT melakukan pemindaian jaringan WiFi untuk konektivitas, membaca data dari sensor kelembapan, serta mengirim dan menyimpan data ke database melalui API.

1.2 Tujuan Eksperimen

- Memahami konsep API dan perannya dalam komunikasi antara perangkat IoT dan server
- Mengintegrasikan API dengan database, sehingga data dari sensor dapat dikirim dan disimpan secara real-time

2. Metodologi

2.1 Alat & Bahan

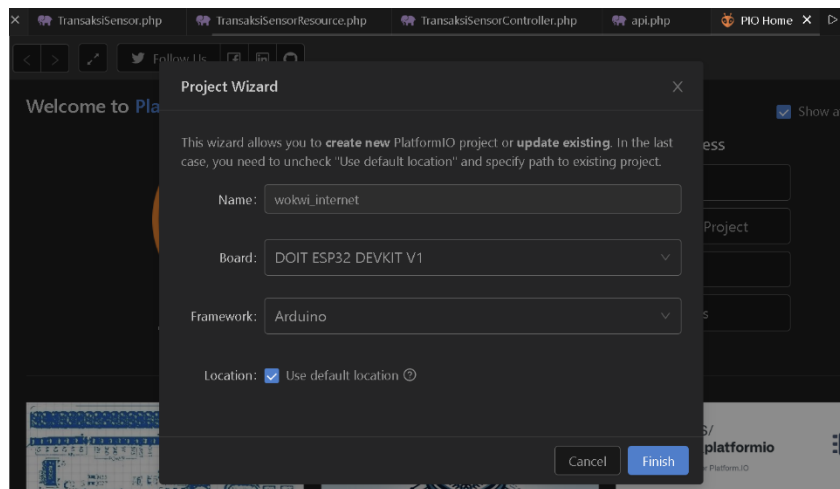
Laptop, Visual Studio Code, XAMPP, phpMyAdmin, Herd, Laravel 11, Postman, Ngrok, dan koneksi internet

2.2 Langkah Implementasi

1. Jalankan API laravel pada folder yang sudah dibuat di bab 12 dengan perintah

php artisan serve --host=0.0.0.0 --port=8080

2. Buat file baru wokwi simulator di platform.io



3. Jalankan Ngrok dengan perintah

ngrok http --scheme=http 8080

```
C:\Users\Jimmy Indriantc > ngrok

ngrok
♦ Goodbye tunnels, hello Agent Endpoints: https://ngrok.com/r/aep

Session Status      online
Account             Femas Alfaridzi (Plan: Free)
Update              update available (version 3.22.0, Ctrl-U to update)
Version             3.20.0
Region              Asia Pacific (ap)
Latency              35ms
Web Interface        http://127.0.0.1:4040
Forwarding           http://4c66-114-10-46-88.ngrok-free.app -> http://localhost:8080

Connections          ttl    opn    rt1    rt5    p50    p90
0                   0      0.00   0.00   0.00   0.00
```

4. Ubah kode script pada file main.cpp pada folder src menjadi seperti ini

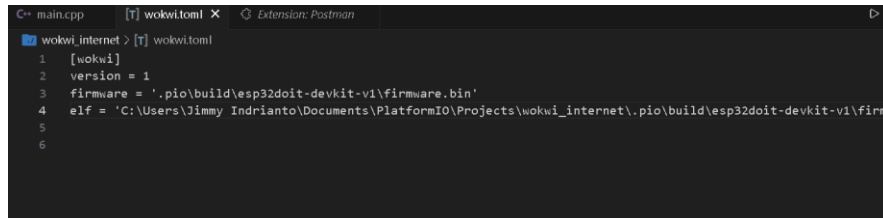
```
1 #include <WiFi.h>
2 #include <HTTPClient.h>
3
4
5 // Ganti dengan kredensial jaringan Wi-Fi Anda
6 // const char* ssid = "Lab IT";
7 // const char* password = "labit2024";
8
9
10
11
12 const char* ssid = "Kontrakan Gay";
13 const char* password = "11223344";
14
15
16
17
18 // URL lengkap server yang akan diakses
19 const char* serverUrl = "http://4c66-114-10-46-88.ngrok-free.app/api/posts";
20
21
22 // Interval waktu antara setiap permintaan (dalam milidetik)
23 const unsigned long interval = 5000;
24 unsigned long previousMillis = 0;
25
26
27 void setup() {
28   Serial.begin(115200);
29   WiFi.begin(ssid, password);
30
31
32   Serial.print("Menghubungkan ke WiFi...");
33   while (WiFi.status() != WL_CONNECTED) {
34     delay(500);
35     Serial.print(".");
36   }
37   Serial.println(" Terhubung!");
38 }
39
40
41
42
43
44
45 void loop() {
46
47
48
49
50   unsigned long currentMillis = millis();
51
52
53   // Periksa apakah interval waktu telah berlalu
54   if (currentMillis - previousMillis >= interval) {
55     previousMillis = currentMillis;
56
57
58     if (WiFi.status() == WL_CONNECTED) {
59       HTTPClient http;
60
61
62       // Inisialisasi HTTPClient dengan URL server
63       http.begin(serverUrl);
64
65
66       // Mengirim permintaan HTTP GET
67       int httpResponseCode = http.GET();
68
69
70       // Menampilkan kode status HTTP
71       Serial.print("Kode status HTTP: ");
72       Serial.println(httpResponseCode);
73
74
75       // Menutup koneksi
76       http.end();
77     } else {
78       Serial.println("WiFi tidak terhubung.");
79     }
80   }
81 }
82
```

Pada bagian ini sesuaikan dengan Forwarding Ngrok masing-masing

// URL lengkap server yang akan diakses

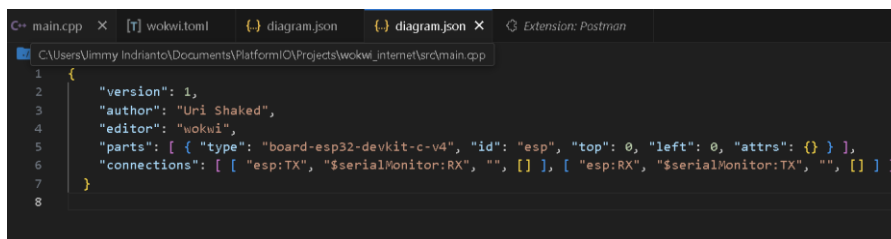
```
const char* serverUrl = "http://4c66-114-10-46-88.ngrok-free.app/api/posts";
```

5. Tambahkan file wokwi.toml



```
wokwi_internet > [T] wokwi.toml
1 [wokwi]
2 version = 1
3 firmware = '.pio\build\esp32doit-devkit-v1\firmware.bin'
4 elf = 'C:\Users\Jimmy Indrianto\Documents\PlatformIO\Projects\wokwi_internet\.pio\build\esp32doit-devkit-v1\firmware.bin'
5
6
```

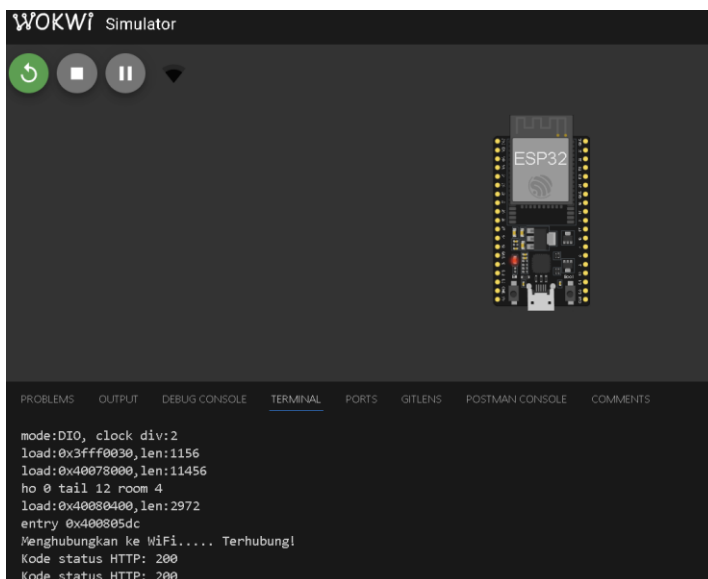
6. Tambahkan file diagram.json



```
C:\Users\Jimmy Indrianto\Documents\PlatformIO\Projects\wokwi_internet\src\main.cpp
1 {
2   "version": 1,
3   "author": "Uri Shaked",
4   "editor": "wokwi",
5   "parts": [ { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 0, "left": 0, "attrs": { } } ],
6   "connections": [ [ "esp:TX", "$serialMonitor:RX", "", [ ] ], [ "esp:RX", "$serialMonitor:TX", "", [ ] ] ]
7 }
8
```

7. Langkah berikutnya adalah melakukan simulasi. Build file main.cpp dan jalankan simulasi dengan perintah

> Wokwi Start Simulator



Simulasi diatas menunjukkan, ESP32 berhasil terhubung ke WIFI Wokwi-GUEST dan berhasil mengakses API laravel yang sudah dibuat pada bab sebelumnya.

Kode Status HTTP:200

HTTP status code 200 artinya adalah "OK". Ini berarti bahwa permintaan (request) yang dikirim oleh klien (misalnya browser web atau aplikasi IoT) telah berhasil diproses oleh server. Dengan kata lain, halaman web atau data yang diminta telah berhasil dikirim kembali oleh server dan ditampilkan dengan benar kepada pengguna.

8. Berikutnya adalah melakukan modifikasi simulasi dengan menambahkan sensor suhu dan kelembaban. Skenarionya adalah, wokwi simulator akan mengirimkan data suhu dan kelembaban ke API dan menyimpannya ke database mysql seperti yang telah dibuat pada bab sebelumnya.

```
{ } diagram.json > ...
1 {
2   "version": 1,
3   "author": "KAVITH BUDVIN",
4   "editor": "wokwi",
5   "parts": [
6     { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 86.4, "left": 24.04, "attrs": {} },
7     { "type": "wokwi-dht22", "id": "dht1", "top": 19.5, "left": -91.8, "attrs": {} }
8   ],
9   "connections": [
10    [ "esp:TX", "$serialMonitor:RX", "", [ ] ],
11    [ "esp:RX", "$serialMonitor:TX", "", [ ] ],
12    [ "dht1:VCC", "esp:3V3", "red", [ "v19.2", "h67.2", "v-67.2" ] ],
13    [ "esp:GND.1", "dht1:GND", "black", [ "h0" ] ],
14    [ "dht1:SDA", "esp:27", "green", [ "v0" ] ]
15  ],
16  "dependencies": {}
17 }
18
```

Rangkai sensor DHT22 dengan ESP32 seperti contoh diatas. Kemudian salin kode diagram.json ke file diagram.json yang ada di vscode

9. Kemudian ubah setting file platformio.ini sebagai berikut :

```
platformio.ini
C:\Users\Uimmy Indrianto\Documents\PlatformIO\Projects\wokwi_internet\platformio.ini
2 ;
3 ; Build options: build flags, source filter
4 ; Upload options: custom upload port, speed and extra flags
5 ; Library options: dependencies, extra library storages
6 ; Advanced options: extra scripting
7 ;
8 ; Please visit documentation for the other options and examples
9 ; https://docs.platformio.org/page/projectconf.html
10
11 [env:esp32doit-devkit-v1]
12 platform = espressif32
13 board = esp32doit-devkit-v1
14 framework = arduino
15 monitor_speed = 115200
16 lib_deps =
17   adafruit/DHT sensor library
```

Pada perubahan diatas, ada tambahan 2 setting yaitu monitor speed dan lib_deps

Monitor speed digunakan untuk memonitor status pengiriman data dari wokwi simulator ke server api laravel yang telah dibuat. Sedangkan lib_deps adalah library yang digunakan sensor DHT (sensor suhu dan kelembaban).

10. Modifikasi file main.cpp

```
1 #include <Arduino.h>
2 #include <WiFi.h>
3 #include <HTTPClient.h>
4
5
6 #include "DHT.h"
7
8
9
10
11 #define DHTPIN 27
12 #define DHTTYPE DHT22
13
14
15
16
17 DHT dht(DHTPIN, DHTTYPE);
18
19
20
21
22
23
24 // Ganti dengan kredensial WiFi Anda
25 const char* ssid = "Wokwi-UE51";
26 const char* password = "";
27
28
29 unsigned long previousMillis = 0;
30 const long interval = 5000; // Interval 5 detik (5000 ms)
31
32
33 void setup() {
34   Serial.begin(115200);
35
36   // Hubungkan ke WiFi
37   WiFi.begin(ssid, password);
38   Serial.print("Menghubungkan ke WiFi");
39   while (WiFi.status() != WL_CONNECTED) {
40     delay(500);
41     Serial.print(".");
42   }
43   Serial.println(" Terhubung!");
44
45
46
47
48
49
50
51
52 dht.begin();
53
54 // Tunggu sebentar agar koneksi stabil
55 delay(1000);
56 }
57
58
59 void loop() {
60   unsigned long currentMillis = millis();
61
62
63   // Lakukan aksi setiap interval yang telah ditentukan
64   if (currentMillis - previousMillis >= interval) {
65     previousMillis = currentMillis;
66
67
68
69
70
71     float h = round(dht.readHumidity());
72     // Read temperature as Celsius (the default)
73     float t = round(dht.readTemperature());
74
75
76     // Check if any reads failed and exit early (to try again).
77     if (isnan(h) || isnan(t)) {
78       Serial.println("Failed to read from DHT sensor!");
79       return;
80     }
81
82
83
84
85     // Compute heat index in Celsius (isFahrenheit = false)
86     float hix = dht.computeHeatIndex(t, h, false);
87
88
89
90
91 // Inisialisasi HTTPClient
92 HTTPClient http;
93 String url = "http://d5fb-114-10-46-88.ngrok-free.app/api/posts"; // Ganti dengan URL ngrok yang benar
94
95
96 http.begin(url); // Menggunakan HTTP, bukan HTTPS
97 http.addHeader("Content-type", "application/json");
98
99
100
101 String payload = "{\"nama_sensor\":\"Sensor GH\", \"nilai1\":\" " + String(h) + ", \"nilai2\":\" " + String(t) + "\"}";
102
103
104 Serial.println(payload); // Untuk melihat apakah payload sudah terbentuk dengan benar
105
106
107
108 // Kirim POST request
109 int httpResponseCode = http.POST(payload);
110
111
112 // Tampilkan kode respons HTTP
113 Serial.print("Kode respons HTTP: ");
114 Serial.println(httpResponseCode);
115
116
117 // Tampilkan respons dari server jika request berhasil
118 if (httpResponseCode == 200 || httpResponseCode == 201) {
119   String response = http.getString();
120   Serial.println("respons dari server:");
121   Serial.println(response);
122 } else {
123   Serial.println("Gagal mengirim data");
124 }
125
126
127 // Tutup koneksi HTTP
128 http.end();
129 }
130 }
```

Pada bagian berikut sesuaikan dengan URL NGROK masing-masing

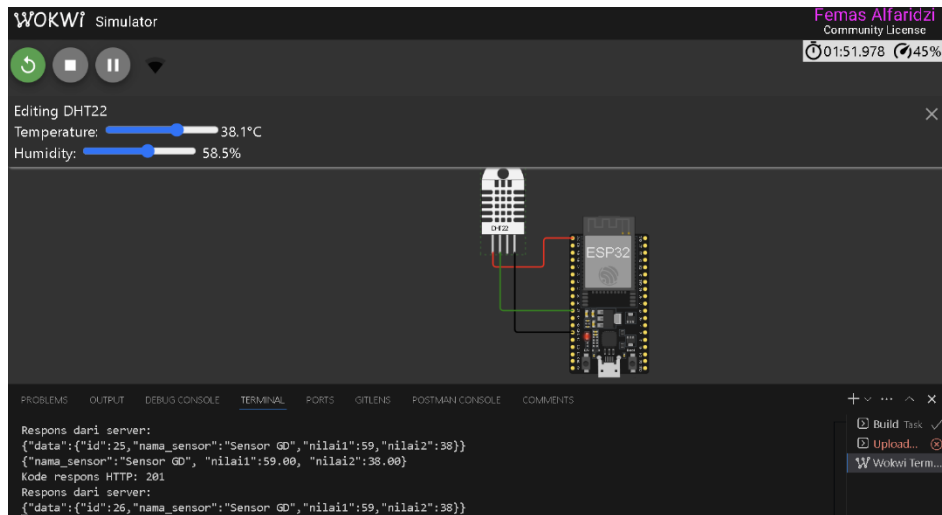
// Inisialisasi HTTPClient

HTTPClient http;

String url = "http://d5fb-114-10-46-88.ngrok-free.app/api/posts"; // Ganti dengan URL ngrok yang benar

11. Jalankan simulasi

Dengan perintah ctrl+shift+p pilih yang > wokwi simulator



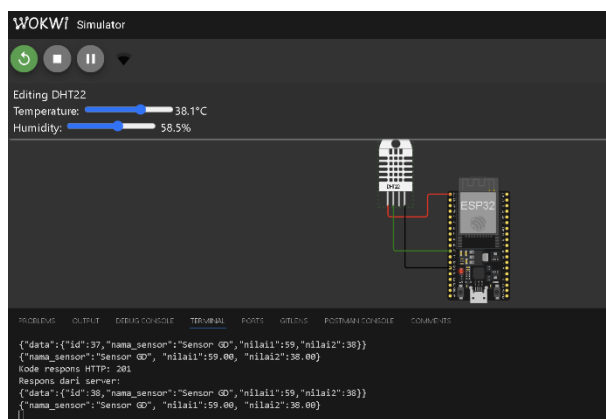
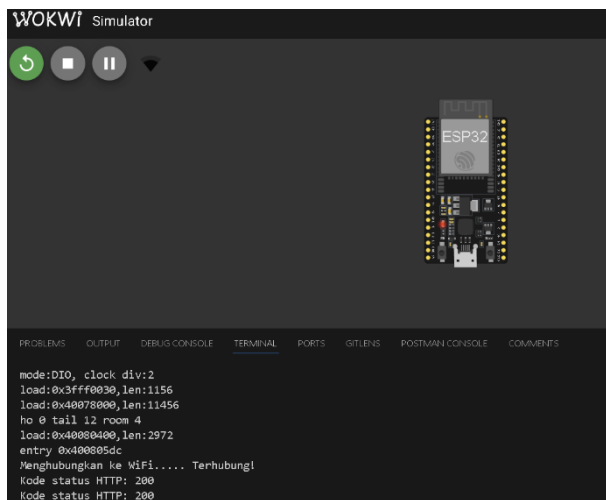
Cek di database, data telah muncul dan tersimpan

Server: 127.0.0.1 » Database: iot_25 » Tabel: transaksi_sensor

Jelajahi		Struktur	SQL	Cari	Tambahkan	Ekspor	Impor	Hak Akses
		id	nama_sensor	nilai1	nilai2	created_at	updated_at	
<input type="checkbox"/>	Ubah	Salin	Hapus	4	Sensor D	123	321	2025-03-13 14:45:17
<input type="checkbox"/>	Ubah	Salin	Hapus	5	Sensor GD	40	24	2025-03-21 07:45:31
<input type="checkbox"/>	Ubah	Salin	Hapus	6	Sensor GD	40	24	2025-03-21 07:45:42
<input type="checkbox"/>	Ubah	Salin	Hapus	7	Sensor GD	40	24	2025-03-21 07:45:55
<input type="checkbox"/>	Ubah	Salin	Hapus	8	Sensor GD	40	24	2025-03-21 07:46:05
<input type="checkbox"/>	Ubah	Salin	Hapus	9	Sensor GD	40	24	2025-03-21 07:46:15
<input type="checkbox"/>	Ubah	Salin	Hapus	10	Sensor GD	40	24	2025-03-21 07:46:28
<input type="checkbox"/>	Ubah	Salin	Hapus	11	Sensor GD	40	24	2025-03-21 07:46:40
<input type="checkbox"/>	Ubah	Salin	Hapus	12	Sensor GD	40	24	2025-03-21 07:46:50
<input type="checkbox"/>	Ubah	Salin	Hapus	13	Sensor GD	40	24	2025-03-21 07:47:00

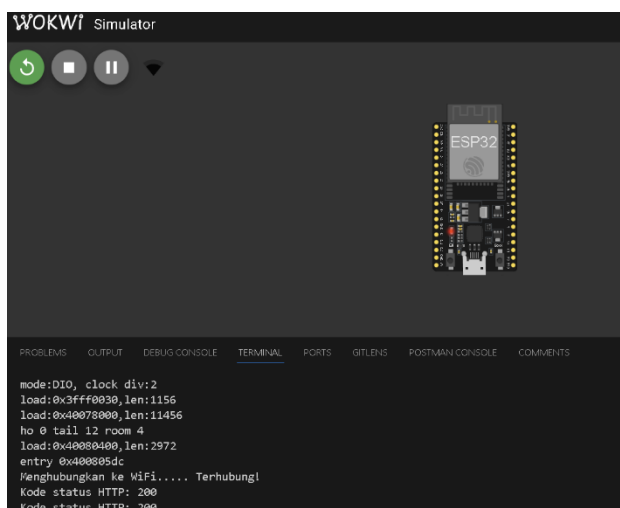
3. Hasil dan Pembahasan

3.1 Hasil Eksperimen

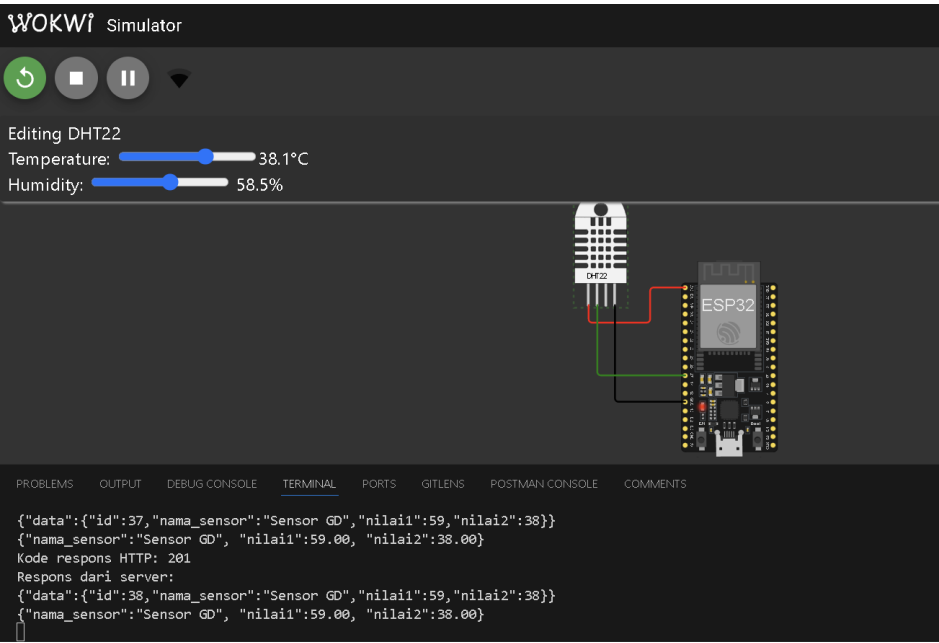


4. Lampiran

Gambar Percobaan pertama Scanning Wi-fi



Gambar percobaan sensor suhu



Gambar tabel database

Server: 127.0.0.1 » Database: iot_25 » Tabel: transaksi_sensor							
Jelajahi	Struktur	SQL	Cari	Tambahkan	Ekspor	Impor	Hak Akses
<div>← T →</div>							
		id	nama_sensor	nilai1	nilai2	created_at	updated_at
<input type="checkbox"/>	Ubah Salin Hapus	4	Sensor D	123	321	2025-03-13 14:45:17	2025-03-13 14:45:17
<input type="checkbox"/>	Ubah Salin Hapus	5	Sensor GD	40	24	2025-03-21 07:45:31	2025-03-21 07:45:31
<input type="checkbox"/>	Ubah Salin Hapus	6	Sensor GD	40	24	2025-03-21 07:45:42	2025-03-21 07:45:42
<input type="checkbox"/>	Ubah Salin Hapus	7	Sensor GD	40	24	2025-03-21 07:45:55	2025-03-21 07:45:55
<input type="checkbox"/>	Ubah Salin Hapus	8	Sensor GD	40	24	2025-03-21 07:46:05	2025-03-21 07:46:05
<input type="checkbox"/>	Ubah Salin Hapus	9	Sensor GD	40	24	2025-03-21 07:46:15	2025-03-21 07:46:15
<input type="checkbox"/>	Ubah Salin Hapus	10	Sensor GD	40	24	2025-03-21 07:46:28	2025-03-21 07:46:28
<input type="checkbox"/>	Ubah Salin Hapus	11	Sensor GD	40	24	2025-03-21 07:46:40	2025-03-21 07:46:40
<input type="checkbox"/>	Ubah Salin Hapus	12	Sensor GD	40	24	2025-03-21 07:46:50	2025-03-21 07:46:50
<input type="checkbox"/>	Ubah Salin Hapus	13	Sensor GD	40	24	2025-03-21 07:47:00	2025-03-21 07:47:00