

LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Praktik Real Hardware ESP32



FEMAS ALFARIDZI

Fakultas vokasi, Universitas Brawijaya

femasalfaridzi17@gmail.com

**Fakultas Vokasi
Universitas Brawijaya**

Abstract

Bab ini membahas praktik langsung penggunaan perangkat ESP32 dalam pengembangan sistem Internet of Things (IoT) menggunakan Visual Studio Code dan PlatformIO. Tiga percobaan utama dilakukan secara bertahap. Pertama, pengendalian lampu LED sebagai dasar pemahaman terhadap penggunaan output digital pada mikrokontroler. Kedua, pengecekan konektivitas WiFi pada ESP32 untuk memastikan perangkat mampu terhubung ke jaringan internet. Ketiga, implementasi IoT dengan menghubungkan sensor suhu dan kelembaban ke sistem API dan database menggunakan PHP Laravel Artisan dan Ngrok. Data sensor berhasil dikirim dan tersimpan ke dalam database melalui endpoint API yang dibuat. Hasil dari ketiga percobaan menunjukkan bahwa ESP32 mampu menjalankan fungsi-fungsi dasar hingga kompleks dalam sistem IoT secara stabil, serta dapat berkomunikasi secara real-time dengan server berbasis web.

Keywords: Sensor DHT22, API, Laravel, LED, ESP32, Ngrok, IoT.

1. Introduction (Pendahuluan)

1.1. Latar Belakang

Pada era digital saat ini, teknologi *Internet of Things*(IoT) semakin berkembang dan diterapkan di berbagai sektor, mulai dari industri, pertanian, hingga rumah tangga. Salah satu perangkat populer yang mendukung pengembangan IoT adalah ESP32, sebuah mikrokontroler dengan konektivitas WiFi dan Bluetooth yang terintegrasi, serta kemampuan pemrosesan yang mumpuni. Dalam bab ini, dilakukan serangkaian praktik nyata menggunakan perangkat keras ESP32 dengan tujuan memperkuat pemahaman dan kemampuan implementasi sistem IoT berbasis mikrokontroler.

Percobaan pertama yang dilakukan adalah pengendalian LED menggunakan ESP32. Praktik ini bertujuan untuk memahami proses dasar pengendalian output digital melalui mikrokontroler dengan pemrograman menggunakan Visual Studio Code dan PlatformIO. Penggunaan dua tools tersebut memberikan fleksibilitas dalam pengelolaan proyek dan efisiensi dalam penulisan kode.

Percobaan kedua fokus pada pengujian konektivitas WiFi dari ESP32. Koneksi internet merupakan aspek krusial dalam sistem IoT karena memungkinkan perangkat untuk saling terhubung dan bertukar data secara real-time. Dengan memastikan ESP32 dapat terhubung ke jaringan WiFi, maka perangkat siap digunakan untuk mengakses atau mengirim data ke server.

Percobaan ketiga merupakan implementasi nyata dari sistem IoT, yaitu dengan menghubungkan sensor suhu dan kelembaban (seperti DHT22) ke sistem backend melalui API dan database. Pada tahap ini, ESP32 dikonfigurasi untuk membaca data sensor lalu mengirimkannya ke server berbasis PHP yang dijalankan menggunakan Artisan (Laravel) dan Ngrok sebagai tunneling untuk membuat server lokal dapat diakses secara publik. Proses ini mencerminkan implementasi nyata pengumpulan data lingkungan dari perangkat fisik dan integrasinya ke sistem berbasis web.

Secara keseluruhan, tiga percobaan ini membentuk rangkaian pembelajaran IoT yang berkesinambungan: dari pengendalian perangkat sederhana, konektivitas, hingga integrasi sistem cloud, yang merupakan pondasi utama dalam membangun solusi berbasis IoT di masa kini.

1.2. Tujuan Eksperimen

- Menerapkan dasar pengendalian output digital menggunakan ESP32, melalui praktik menyalakan dan mematikan lampu LED menggunakan kode program yang ditulis di Visual Studio Code dengan bantuan PlatformIO.

- Menguji koneksi jaringan pada perangkat ESP32 untuk memastikan perangkat mampu terhubung ke jaringan WiFi sebagai syarat utama dalam komunikasi data pada sistem IoT.
- Mengimplementasikan sistem IoT yang terhubung dengan sensor suhu dan kelembaban, serta mengirimkan data sensor tersebut ke server melalui API yang dibangun dengan PHP Laravel Artisan dan diakses menggunakan Ngrok.
- Memahami alur integrasi antara perangkat keras dan server, mulai dari pembacaan data sensor, pengiriman data melalui protokol HTTP, hingga penyimpanan ke dalam database secara real-time.

-

2. Methodology (Metodologi)

2.1. Tools & Materials (Alat dan Bahan)

- Laptop
- Visual Studio Code
- phpMyAdmin
- XAMPP
- Herd
- Ngrok
- Laravel 11
- Koneksi internet
- ESP32
- Sensor DHT22

2.2. Implementation Steps (Langkah Implementasi)

1. Download Driver Silicon Labs CP210x melalui link berikut :
<https://www.silabs.com/developer-tools/usb-to-uart-bridge-vcp-drivers?tab=downloads>
2. Set up pada Device manager, untuk tutorialnya bisa dilihat di link berikut :
https://www.youtube.com/watch?v=r_eMEXvt0v0
3. Buat project baru di PlatformIo

4. Ubah file **platformio.ini** menjadi sebagai berikut:

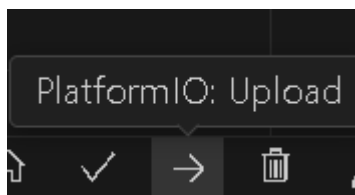
```
[env:esp32doit-devkit-v1]
platform = espressif32
board = esp32doit-devkit-v1
framework = arduino
upload_port = COM3
monitor_port = COM3
```

5. Kemudian pada file **main.cpp** masukkan koding lampu LED yang telah dibuat di bab 8

A screenshot of a code editor window with a dark background. The code is written in C++ for an ESP32. It includes an Arduino header, declares two LED pins (25 and 26), and defines setup and loop functions. The setup function initializes the serial port and sets the pins as outputs. The loop function turns both LEDs on, waits for 1000ms, turns them off, and waits for another 1000ms before repeating.

```
1 #include <Arduino.h> // Wajib untuk PlatformIO + ESP32
2
3 // Deklarasi pin LED
4 int lampu = 25;
5 int lampu2 = 26;
6
7
8 void setup() {
9     Serial.begin(115200); // Inisialisasi komunikasi Serial
10    Serial.println("ESP32 Blinking LED");
11
12    // Atur pin sebagai OUTPUT
13    pinMode(lampu, OUTPUT);
14    pinMode(lampu2, OUTPUT);
15 }
16
17
18 void loop() {
19     // Nyalakan kedua LED
20     digitalWrite(lampu, HIGH);
21     digitalWrite(lampu2, HIGH);
22     Serial.println("LED ON");
23
24     delay(1000); // Tunggu 1 detik
25
26
27
28     // Matikan kedua LED
29     digitalWrite(lampu, LOW);
30     digitalWrite(lampu2, LOW);
31     Serial.println("LED OFF");
32
33     delay(1000); // Tunggu 1 detik sebelum mengulang
34 }
35
36
```

6. Setelah wiring komponen , lakukan **upload** di vscode

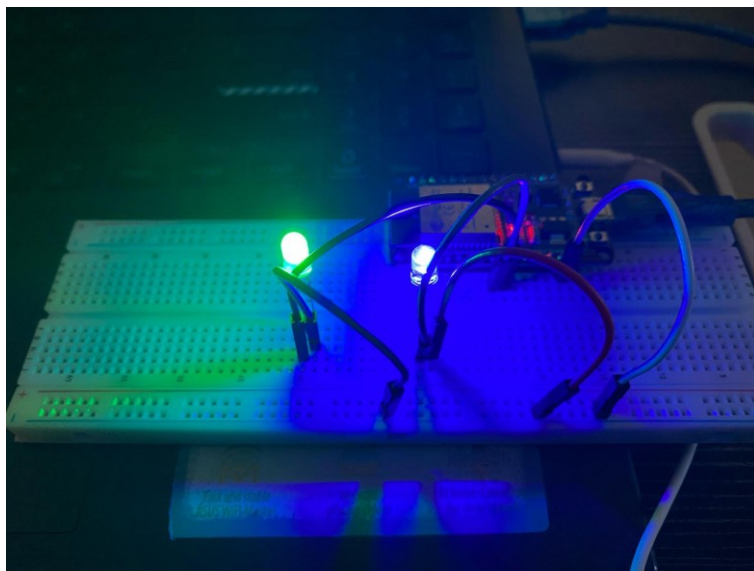


7. Proses compiling dan upload akan berjalan dan pastikan berhasil seperti tampilan berikut

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS  POSTMAN CONSOLE

Writing at 0x00010000... (10 %)
Writing at 0x0001c3bd... (20 %)
Writing at 0x00024a72... (30 %)
Writing at 0x00029cef... (40 %)
Writing at 0x0002f24c... (50 %)
Writing at 0x000347cb... (60 %)
Writing at 0x0003d296... (70 %)
Writing at 0x000462a9... (80 %)
Writing at 0x0004b847... (90 %)
```

Setelah langkah ini, jika wiring kabel dilakukan dengan benar seharusnya lampu LED menyala sesuai logika program.

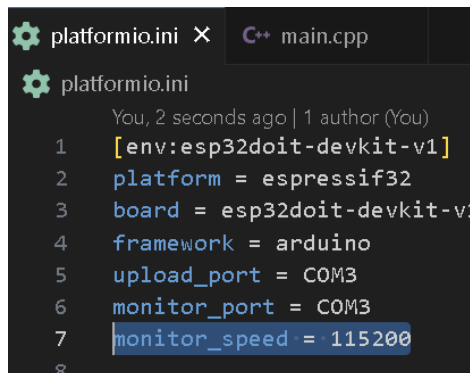


Mengecek Koneksi Wi-Fi pada Hardware ESP32

Pada latihan berikutnya perlu dilakukan upload program untuk mengecek apakah hardware ESP32 dapat terhubung ke Access Point WIFI disekitar. Untuk melakukannya perlu melakukan perubahan koding pada file main.cpp seperti berikut ini :

```
1  #include <WiFi.h>
2
3
4  void setup() {
5      Serial.begin(115200);
6
7
8      WiFi.mode(WIFI_STA);
9      WiFi.disconnect();
10     delay(100);
11
12
13     Serial.println("Pemindaian Jaringan Wi-Fi Dimulai...");
14 }
15
16
17 void loop() {
18     int n = WiFi.scanNetworks();
19     Serial.println("Pemindaian Selesai");
20     if (n == 0) {
21         Serial.println("Tidak ada jaringan Wi-Fi yang ditemukan.");
22     } else {
23         Serial.print(n);
24         Serial.println(" jaringan Wi-Fi ditemukan:");
25         for (int i = 0; i < n; ++i) {
26             Serial.print(i + 1);
27             Serial.print(": ");
28             Serial.print(WiFi.SSID(i));
29             Serial.print(" ");
30             Serial.print(WiFi.RSSI(i));
31             Serial.print(" dBm");
32             Serial.println((WiFi.encryptionType(i) == WIFI_AUTH_OPEN) ? " " : "****");
33             delay(10);
34         }
35     }
36     Serial.println("");
37
38
39     delay(5000); // Lakukan pemindaian setiap 5 detik
40 }
41
```

Lalu pada file **platformio.ini** tambahkan code seperti ini

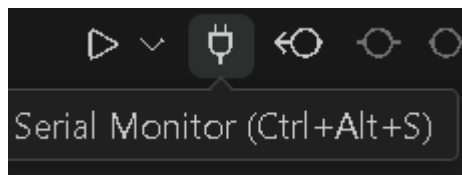


```
platformio.ini
You, 2 seconds ago | 1 author (You)
1 [env:esp32doit-devkit-v1]
2 platform = espressif32
3 board = esp32doit-devkit-v1
4 framework = arduino
5 upload_port = COM3
6 monitor_port = COM3
7 monitor_speed = 115200
8
```

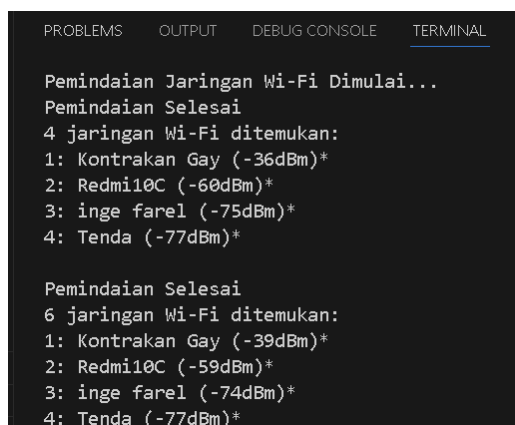
Lakukan proses upload

Perhatikan bagian `monitor_speed` , baris tersebut berfungsi untuk melakukan preview serial monitor.

Kemudian klik tombol serial monitor



Pastikan tampilan serial monitor menunjukkan nama Access Point WIFI disekitar berikut juga dengan keterangan kekuatan sinyal seperti tampilan berikut ini



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Pemindaian Jaringan Wi-Fi Dimulai...
Pemindaian Selesai
4 jaringan Wi-Fi ditemukan:
1: Kontrakan Gay (-36dBm)*
2: Redmi10C (-60dBm)*
3: inge farel (-75dBm)*
4: Tenda (-77dBm)*

Pemindaian Selesai
6 jaringan Wi-Fi ditemukan:
1: Kontrakan Gay (-39dBm)*
2: Redmi10C (-59dBm)*
3: inge farel (-74dBm)*
4: Tenda (-77dBm)*
```

Pada langkah ini, hardware ESP32 telah berhasil melakukan scanning WIFI disekitar.

Langkah berikutnya adalah implementasi Internet of Things dengan menghubungkan sensor suhu dan kelembaban ke sistem API dan database yang telah dibuat. Proses ini sudah pernah dilakukan

pada bab 13, namun diperlukan beberapa penyesuaian dan implementasi koding ke hardware ESP32 nyata (bukan simulator WOKWI)

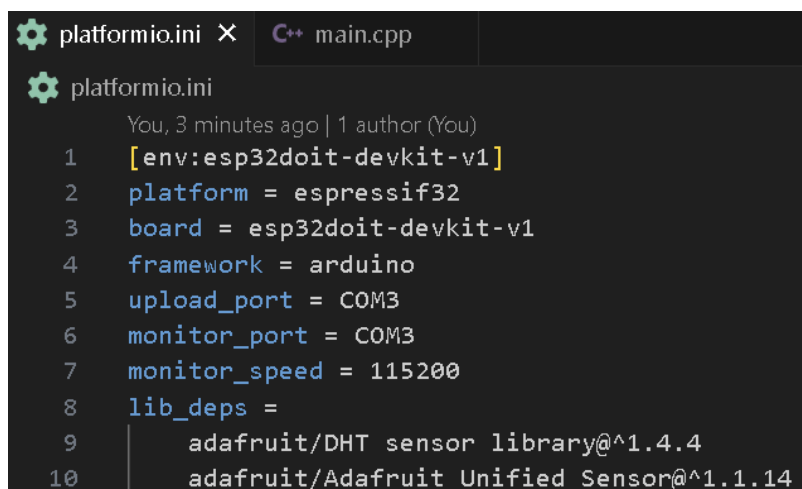
Jalankan API laravel kembali dengan perintah

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

Kemudian jalankan NGROK

ngrok http --scheme=http 8080

Tambahkan library pada file platformio.ini



```
platformio.ini × C++ main.cpp
platformio.ini
You, 3 minutes ago | 1 author (You)
1  [env:esp32doit-devkit-v1]
2  platform = espressif32
3  board = esp32doit-devkit-v1
4  framework = arduino
5  upload_port = COM3
6  monitor_port = COM3
7  monitor_speed = 115200
8  lib_deps =
9      adafruit/DHT sensor library@^1.4.4
10     adafruit/Adafruit Unified Sensor@^1.1.14
```

Lakukan proses wiring cable sesuai

Setelah memastikan wiring cable dengan benar, langkah berikutnya adalah implementasi kode main.cpp di hardware ESP32. Lakukan modifikasi file main.cpp sebagai berikut

```
1 #include <Arduino.h>
2 #include <WiFi.h>
3 #include <HTTPClient.h>
4 #include "DHT.h"
5
6
7 #define DHTPIN 27
8 #define DHTTYPE DHT22
9
10
11 DHT dht(DHTPIN, DHTTYPE);
12
13 // Ganti dengan kredensial WiFi Anda
14 const char* ssid = "Kontrakan Gay";
15 const char* password = "11223344";
16
17 unsigned long previousMillis = 0;
18 const long Interval = 5000; // Interval 5 detik (5000 ms)
19
20 void setup() {
21   Serial.begin(115200);
22
23   // Hubungkan ke WiFi
24   WiFi.begin(ssid, password);
25   Serial.print("Menghubungkan ke WiFi");
26   while (WiFi.status() != WL_CONNECTED) {
27     delay(500);
28     Serial.print(".");
29   }
30   Serial.println(" Terhubung!");
31
32   dht.begin();
33
34   // Tunggu sebentar agar koneksi stabil
35   delay(1000);
36 }
37
38 void loop() {
39   unsigned long currentMillis = millis();
40
41
42
43   // Lakukan POST setiap interval yang telah ditentukan
44   if (currentMillis - previousMillis >= Interval) {
45     previousMillis = currentMillis;
46
47
48     float h = round(dht.readHumidity());
49     // Read temperature as Celsius (the default)
50     float t = round(dht.readTemperature());
51
52
53     // Check if any reads failed and exit early (to try again).
54     if (isnan(h) || isnan(t)) {
55       Serial.println("Failed to read from DHT sensor!");
56       return;
57     }
58
59     // Compute heat index in Celsius (isFahrenheit = false)
60     float hic = dht.computeHeatIndex(t, h, false);
61
62
63     // Inisialisasi HTTPClient
64     HTTPClient http;
65     String url = "http://0ae3-114-10-46-236.ngrok-free.app/api/posts"; // Ganti dengan URL ngrok yang benar
66
67
68     http.begin(url); // Menggunakan HTTP, bukan HTTPS
69     http.addHeader("Content-Type", "application/json");
70
71
72     String payload = "{\"nama_sensor\":\"Sensor GD\", \"nilai1\":\"" + String(h) + ", \"nilai2\":\"" + String(t) + "\"}";
73
74
75
76
77
78     Serial.println(payload); // Untuk melihat apakah payload sudah terbentuk dengan benar
79
80
81
82
83     // Kirim POST request
84     int httpResponseCode = http.POST(payload);
85
86     // Tampilkan kode respons HTTP
87     Serial.print("Kode respons HTTP: ");
88     Serial.println(httpResponseCode);
89
90
91
92
93     // Tampilkan respons dari server jika request berhasil
94     if (httpResponseCode == 200 || httpResponseCode == 201) {
95       String response = http.getString();
96       Serial.println("Respons dari server:");
97       Serial.println(response);
98     } else {
99       Serial.println("Gagal mengirim data");
100     }
101
102
103
104
105     // Tutup koneksi HTTP
106     http.end();
107   }
108 }
109
```

Lakukan proses upload. Kemudian jalankan simulasi. Pastikan data yang dikirim dari hardware ESP32 dapat masuk ke database.

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    GITLENS    POSTMAN CONSOLE

Respons dari server:
{"data":{"id":62,"nama_sensor":"Sensor GD","nilai1":76,"nilai2":30}}
{"nama_sensor":"Sensor GD", "nilai1":77.00, "nilai2":30.00}
Kode respons HTTP: 201
Respons dari server:
{"data":{"id":63,"nama_sensor":"Sensor GD","nilai1":77,"nilai2":30}}
{"nama_sensor":"Sensor GD", "nilai1":77.00, "nilai2":30.00}
Kode respons HTTP: 201
Respons dari server:
{"data":{"id":64,"nama_sensor":"Sensor GD","nilai1":77,"nilai2":30}}
```

←

Server: 127.0.0.1 » Database: iot_25 » Tabel: transaksi_sensor

Jelajahi

Struktur

SQL

Cari

Tambahkan

Ekspor

Impor

Hak Akses

← T →

id

nama_sensor

nilai1

nilai2

created_at

updated_at

Ubah

Salin

Hapus

15

Sensor GD

59

38

2025-03-21 07:47:22

2025-03-21 07:47:22

Ubah

Salin

Hapus

16

Sensor GD

59

38

2025-03-21 07:47:33

2025-03-21 07:47:33

Ubah

Salin

Hapus

17

Sensor GD

59

38

2025-03-21 07:47:43

2025-03-21 07:47:43

Ubah

Salin

Hapus

18

Sensor GD

59

38

2025-03-21 07:47:53

2025-03-21 07:47:53

Ubah

Salin

Hapus

19

Sensor GD

59

38

2025-03-21 07:48:03

2025-03-21 07:48:03

Ubah

Salin

Hapus

20

Sensor GD

59

38

2025-03-21 07:48:13

2025-03-21 07:48:13

Ubah

Salin

Hapus

21

Sensor GD

59

38

2025-03-21 07:48:23

2025-03-21 07:48:23

Ubah

Salin

Hapus

22

Sensor GD

59

38

2025-03-21 07:48:33

2025-03-21 07:48:33

Ubah

Salin

Hapus

23

Sensor GD

59

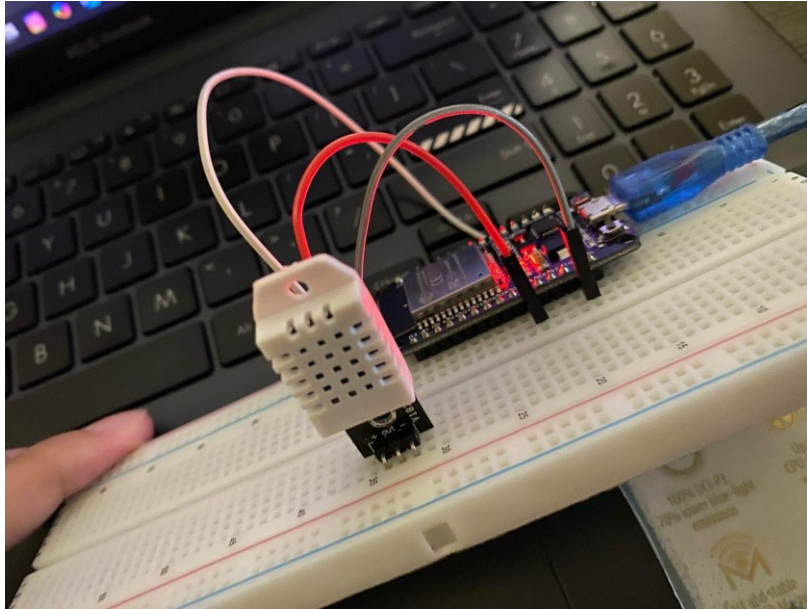
38

2025-03-21 07:48:43

2025-03-21 07:48:43

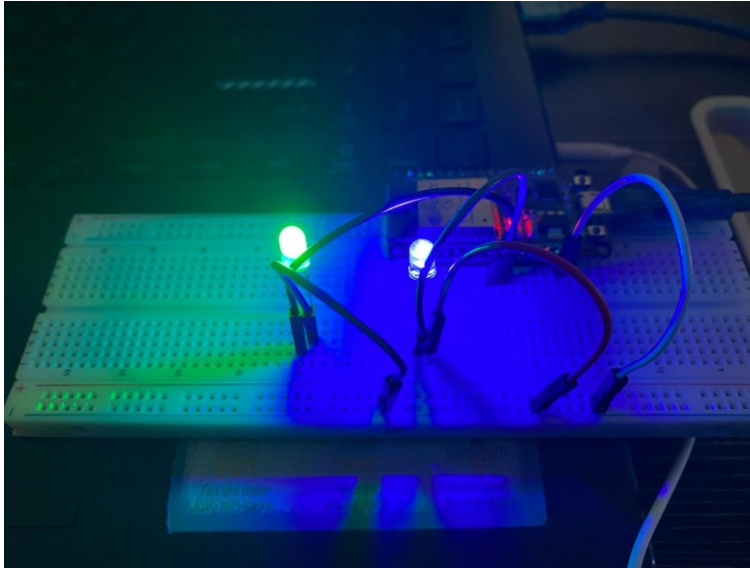
3. Results and Discussion (Hasil dan Pembahasan)

3.1. Experimenetal Results (Hasil Eksperimen)



4. Appendix (Lampiran)

Lampu LED



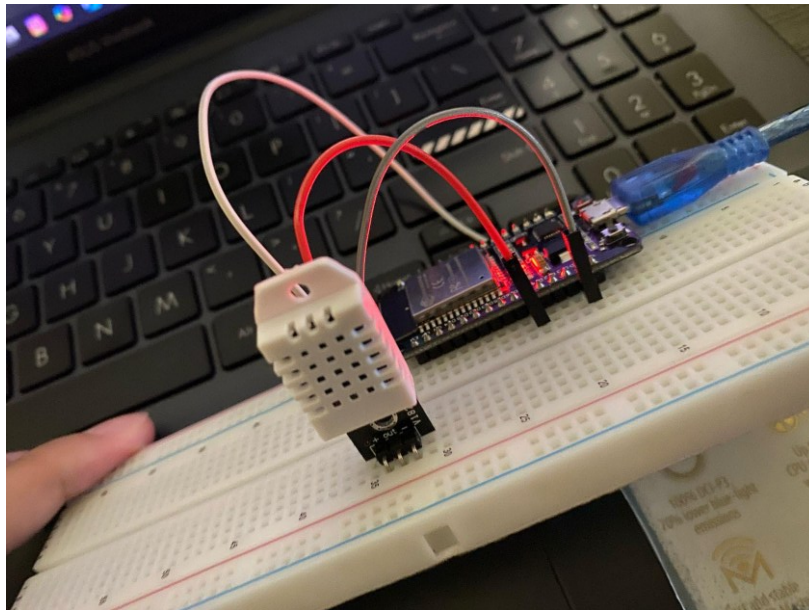
Scanning Wi-fi

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Pemindaian Jaringan Wi-Fi Dimulai...
Pemindaian Selesai
4 jaringan Wi-Fi ditemukan:
1: Kontrakan Gay (-36dBm)*
2: Redmi10C (-60dBm)*
3: inge farel (-75dBm)*
4: Tenda (-77dBm)*

Pemindaian Selesai
6 jaringan Wi-Fi ditemukan:
1: Kontrakan Gay (-39dBm)*
2: Redmi10C (-59dBm)*
3: inge farel (-74dBm)*
4: Tenda (-77dBm)*
```

Sensor DHT22



Tabel database

Server: 127.0.0.1 » Database: iot_25 » Tabel: transaksi_sensor									
Jelajahi Struktur SQL Cari Tambahkan Ekspor Impor Hak Akses									
← T →									
				id	nama_sensor	nilai1	nilai2	created_at	updated_at
<input type="checkbox"/>	Ubah	Salin	Hapus	15	Sensor GD	59	38	2025-03-21 07:47:22	2025-03-21 07:47:22
<input type="checkbox"/>	Ubah	Salin	Hapus	16	Sensor GD	59	38	2025-03-21 07:47:33	2025-03-21 07:47:33
<input type="checkbox"/>	Ubah	Salin	Hapus	17	Sensor GD	59	38	2025-03-21 07:47:43	2025-03-21 07:47:43
<input type="checkbox"/>	Ubah	Salin	Hapus	18	Sensor GD	59	38	2025-03-21 07:47:53	2025-03-21 07:47:53
<input type="checkbox"/>	Ubah	Salin	Hapus	19	Sensor GD	59	38	2025-03-21 07:48:03	2025-03-21 07:48:03
<input type="checkbox"/>	Ubah	Salin	Hapus	20	Sensor GD	59	38	2025-03-21 07:48:13	2025-03-21 07:48:13
<input type="checkbox"/>	Ubah	Salin	Hapus	21	Sensor GD	59	38	2025-03-21 07:48:23	2025-03-21 07:48:23
<input type="checkbox"/>	Ubah	Salin	Hapus	22	Sensor GD	59	38	2025-03-21 07:48:33	2025-03-21 07:48:33
<input type="checkbox"/>	Ubah	Salin	Hapus	23	Sensor GD	59	38	2025-03-21 07:48:43	2025-03-21 07:48:43