

LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Praktik Real Hardware ESP32

*Lucky Ardiansyah 233140701111015
Fakultas Vokasi, Universitas Brawijaya
Email: luckyardiansyah685@gmail.com*

Abstract (Abstrak)

Praktikum ini bertujuan untuk memahami dan mengimplementasikan penggunaan mikrokontroler ESP32 dalam pengendalian perangkat keras secara langsung (*real hardware*). ESP32 merupakan mikrokontroler yang memiliki kemampuan konektivitas nirkabel seperti WiFi dan Bluetooth, serta mendukung berbagai protokol komunikasi dan antarmuka input/output. Pada praktikum ini, dilakukan pemrograman ESP32 menggunakan bahasa pemrograman C/C++ melalui platform Arduino IDE untuk mengontrol perangkat seperti LED, sensor suhu, dan modul lainnya. Selain itu, praktikan juga mempelajari proses upload program ke board ESP32 serta proses debugging secara langsung. Hasil praktikum menunjukkan bahwa ESP32 mampu menjalankan fungsi-fungsi dasar kendali perangkat keras dengan baik dan responsif. Praktikum ini memberikan pemahaman mendalam mengenai integrasi antara perangkat lunak dan perangkat keras dalam sistem embedded, serta memperkuat keterampilan praktis dalam pengembangan sistem IoT skala kecil.

Kata Kunci: ESP32, mikrokontroler, real hardware, Arduino IDE, IoT, embedded system

1. Introduction (Pendahuluan)

1.1 Latar Belakang

API (Application Programming Interface) adalah serangkaian protokol yang memungkinkan satu aplikasi saling berkomunikasi dengan aplikasi lain misalnya klien dengan server. Pembuatan API dapat menggunakan berbagai macam framework, salah satunya yang paling populer adalah laravel 11. Laravel 11 merupakan *framework* PHP yang menyediakan berbagai *tools* dan *library* yang memudahkan pengembang dalam pembuatan API. Laravel 11 memungkinkan kita mengelola berbagai *tools* seperti *routing*, autentikasi, *middleware*, dan pengelolaan database. Ngrok adalah *proxy* server untuk membuat jaringan *private* melalui NAT atau *firewall* untuk menghubungkan server lokal ke internet dengan aman. Ngrok membuat URL publik yang dapat digunakan untuk mengakses API secara *online* kapanpun dan dimanapun. Ngrok sangat berguna untuk mengelola perangkat IoT dari jarak jauh tanpa perlu pengaturan IP publik atau NAT traversal. ESP32 adalah mikrokontroler SoC (System on Chip) yang memiliki Wi-Fi dan Bluetooth terintegrasi, sangat cocok untuk aplikasi Internet of Things (IoT). Sedangkan sensor DHT22 adalah alat yang digunakan untuk mengukur tingkat kelembapan udara dan suhu di lingkungan atau ruangan.

1.2 Tujuan Eksperimen

Tujuan dari praktik ini adalah mengakses API yang diintegrasikan dengan perangkat IoT (ESP32 dan sensor DHT22) menggunakan laravel 11, perangkat IoT dihubungkan dengan url dari Ngrok supaya dapat diakses secara online, dan perangkat IoT akan mengirimkan data ke database.

2. Methodology (Metodologi)

2.1 Tools & Materials (Alat & Bahan)

Laptop, Visual Studio Code, ESP32, sensor HC-SR04, sensor DHT22, lampu LED, XAMPP, phpMyAdmin, dan koneksi internet.

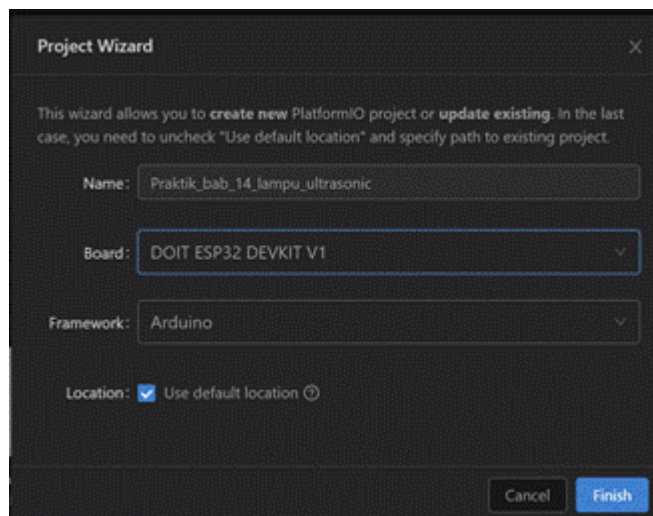
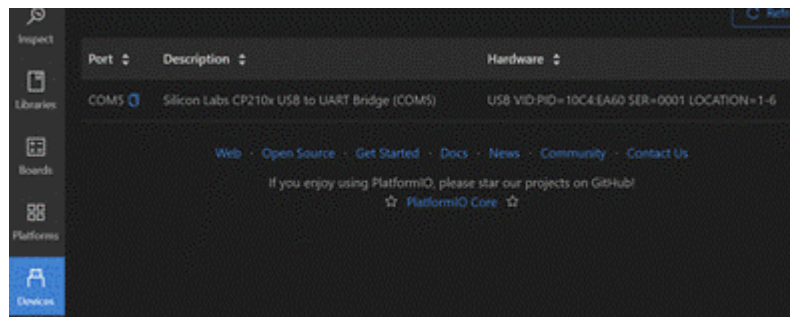
2.2 Implementation Steps (Langkah Implementasi)

IMPLEMENTASI LAMPU LED

1. Menghubungkan perangkat ESP32 dengan laptop menggunakan kabel USB, pastikan ESP32 terdeteksi di laptop dengan membuka device manager dan cek pada menu ports harus terdapat port Silicon Labs



2. Buka vs code dan extensi PlatformIo, cek apakah ESP32 sudah terdeteksi atau belum, jika sudah buat project baru di PlatformIo.



3. Ubah kode pada file platformio.ini menjadi seperti dibawah, nama port disesuaikan

```
10
11 [env:esp32doit-devkit-v1]
12 platform = espressif32
13 board = esp32doit-devkit-v1
14 framework = arduino
15 upload_port = COM5
16 monitor_port = COM5
```

4. Rakit ESP32 dengan sensor HC-SR04 dan juga lampu LED berdasarkan tabel dibawah

Komponen	ESP32
HC-SR04 VCC	3.3V
HC-SR04 GND	GND

HC-SR04 Trig	GPIO 18 (D18)
HC-SR04 Echo	GPIO 19 (D19)
LED Kuning +	GPIO 23 (D23)
LED Merah +	GPIO 22 (D22)
Kedua LED -	GND

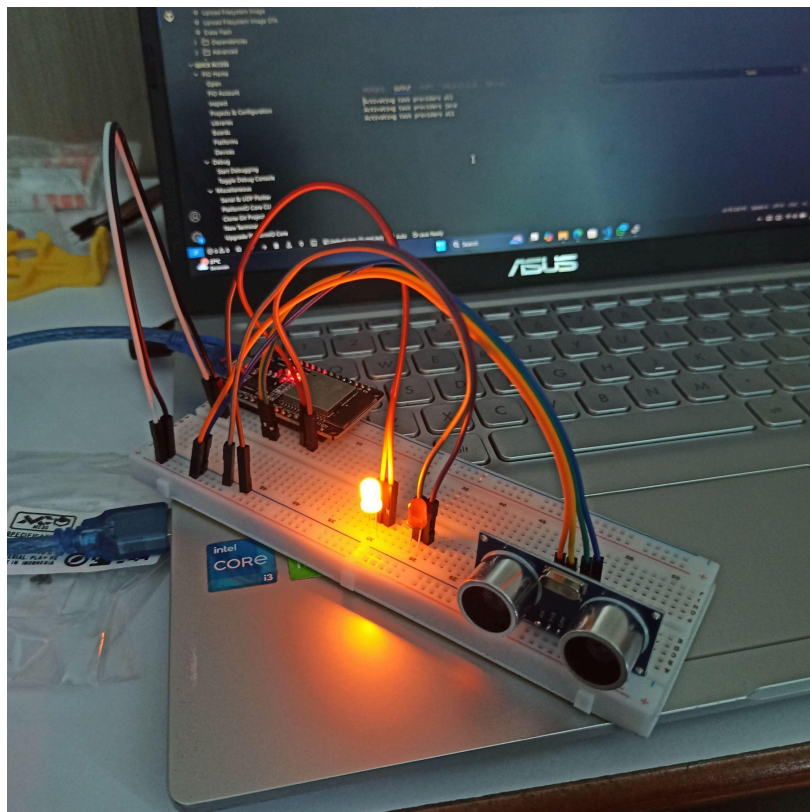
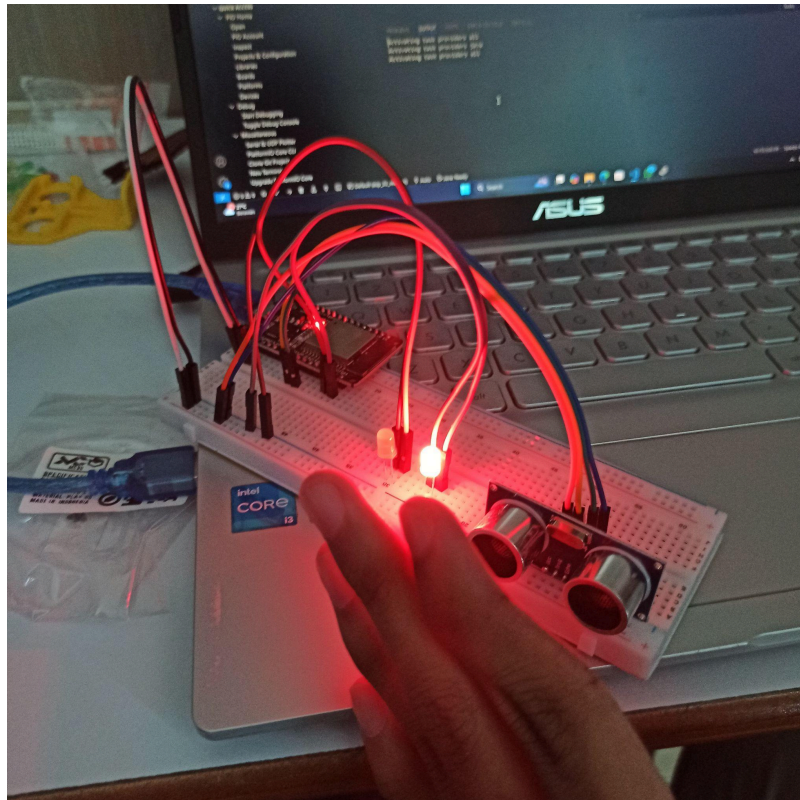
5. Kode dibawah ini bertujuan untuk mendeteksi jarak suatu objek dengan sensor HC-SR04 yang akan menyalakan lampu merah ketika tidak ada objek dekat, dan menyalakan lampu kuning jika terdapat objek dekat.

```

1  #include <Arduino.h>
2
3  // Pin konfigurasi
4  const int trigPin = 18;
5  const int echoPin = 19;
6
7  const int ledKuning = 23; // LED kuning menyala jika objek dekat
8  const int ledMerah = 22; // LED merah menyala jika tidak ada objek
9
10 long duration;
11 int distance;
12
13 void setup() {
14   pinMode(trigPin, OUTPUT);
15   pinMode(echoPin, INPUT);
16   pinMode(ledKuning, OUTPUT);
17   pinMode(ledMerah, OUTPUT);
18
19   Serial.begin(115200);
20 }
21
22 void loop() {
23   // Trigger sensor
24   digitalWrite(trigPin, LOW);
25   delayMicroseconds(2);
26   digitalWrite(trigPin, HIGH);
27   delayMicroseconds(10);
28   digitalWrite(trigPin, LOW);
29
30   // Baca pantulan echo
31   duration = pulseIn(echoPin, HIGH);
32   distance = duration * 0.034 / 2;
33
34   Serial.print("Jarak: ");
35   Serial.print(distance);
36   Serial.println(" cm");
37
38   if (distance > 0 && distance < 20) {
39     digitalWrite(ledKuning, HIGH);
40     digitalWrite(ledMerah, LOW);
41   } else {
42     digitalWrite(ledKuning, LOW);
43     digitalWrite(ledMerah, HIGH);
44   }
45
46   delay(300);
47 }
48

```

6. Kemudian pada menu berikut pilih Upload untuk mengupload kode main.cpp ke perangkat ESP32, dan tunggu prosesnya sampai selesai. Dan berikut hasilnya



MENGECEK KONEKSI WiFi PADA HARDWARE ESP32

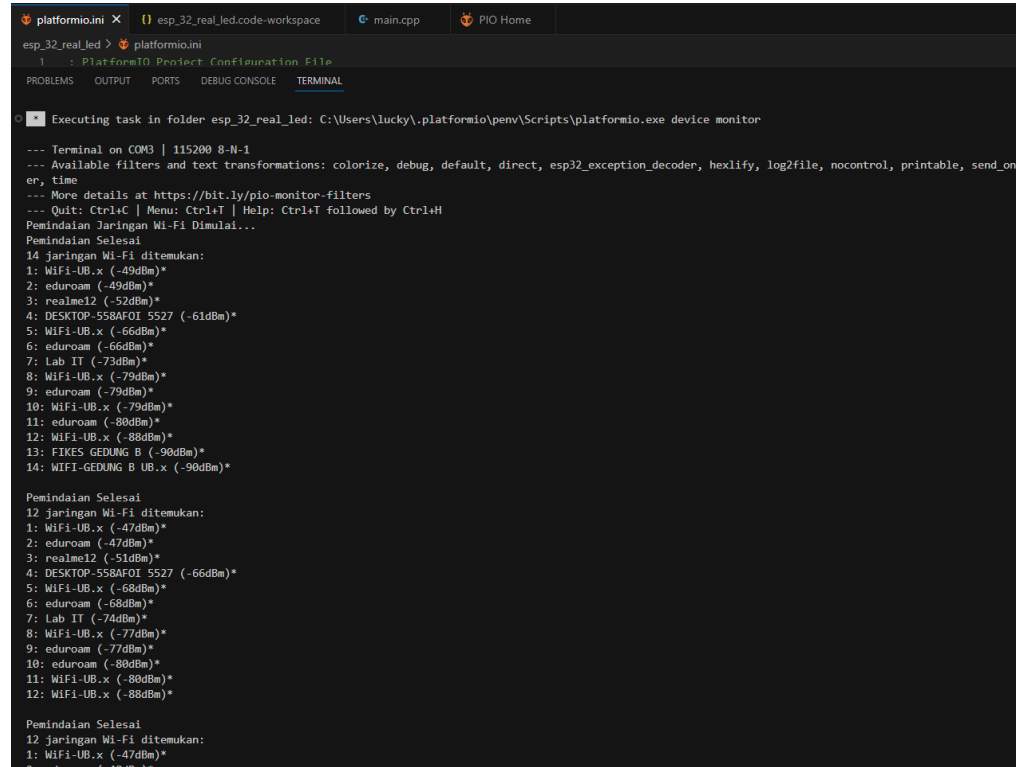
1. Coding main.cpp

```
platformio.ini  {} esp_32_real_led.code-workspace  main.cpp x  PIO Home
esp_32_real_led > src > main.cpp > ...
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
```

2. Coding file platformio.ini

```
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  upload_port = COM3
16  monitor_port = COM3
17  monitor_speed = 115200
```

3. Upload kemudian mengklik Serial Monitor dan muncul hasil pada Terminal



```
platformio.ini X | esp_32_real_led.code-workspace | main.cpp | PIO Home
esp_32_real_led > platformio.ini
1 : PlatformIO Project Configuration File
PROBLEMS OUTPUT PORTS DEBUG CONSOLE TERMINAL

[ ] Executing task in folder esp_32_real_led: C:\Users\lucky\.platformio\penv\Scripts\platformio.exe device monitor

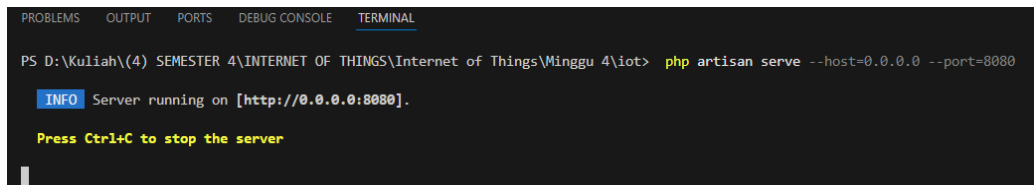
--- Terminal on COM3 | 115200 8-N-1
--- Available filters and text transformations: colorize, debug, default, direct, esp32_exception_decoder, hexlify, log2file, nocontrol, printable, send_on
er, time
--- More details at https://bit.ly/pio-monitor-filters
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H
Pemindaian Jaringan Wi-Fi Dimulai...
Pemindaian Selesai
14 jaringan Wi-Fi ditemukan:
1: WiFi-UB.x (-49dBm)*
2: eduroam (-49dBm)*
3: realme12 (-52dBm)*
4: DESKTOP-5S8AFOI 5527 (-61dBm)*
5: WiFi-UB.x (-66dBm)*
6: eduroam (-66dBm)*
7: Lab IT (-73dBm)*
8: WiFi-UB.x (-79dBm)*
9: eduroam (-79dBm)*
10: WiFi-UB.x (-79dBm)*
11: eduroam (-80dBm)*
12: WiFi-UB.x (-88dBm)*
13: FIKES GEDUNG B (-90dBm)*
14: WIFI-GEDUNG B UB.x (-90dBm)*

Pemindaian Selesai
12 jaringan Wi-Fi ditemukan:
1: WiFi-UB.x (-47dBm)*
2: eduroam (-47dBm)*
3: realme12 (-51dBm)*
4: DESKTOP-5S8AFOI 5527 (-66dBm)*
5: WiFi-UB.x (-68dBm)*
6: eduroam (-68dBm)*
7: Lab IT (-74dBm)*
8: WiFi-UB.x (-77dBm)*
9: eduroam (-77dBm)*
10: eduroam (-80dBm)*
11: WiFi-UB.x (-80dBm)*
12: WiFi-UB.x (-88dBm)*

Pemindaian Selesai
12 jaringan Wi-Fi ditemukan:
1: WiFi-UB.x (-47dBm)*
2: eduroam (-47dBm)*
```

Implementasi Internet of Things Sensor suhu dan kelembaban ke sistem API dan Database

1. Menjalankan kembali Laravel



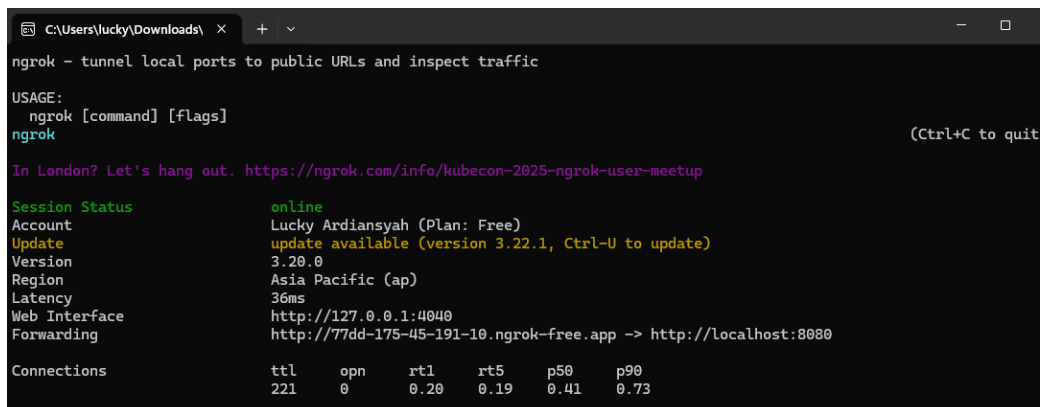
```
PROBLEMS OUTPUT PORTS DEBUG CONSOLE TERMINAL

PS D:\Kuliah\4) SEMESTER 4\INTERNET OF THINGS\Internet of Things\Minggu 4\iot> php artisan serve --host=0.0.0.0 --port=8080

[INFO] Server running on [http://0.0.0.0:8080].

Press Ctrl+C to stop the server
```

2. Menjalankan NGROK



```
C:\Users\lucky\Downloads\ x + v - □

ngrok - tunnel local ports to public URLs and inspect traffic

USAGE:
  ngrok [command] [flags]
ngrok                                     (Ctrl+C to quit)

In London? Let's hang out. https://ngrok.com/info/kubecon-2025-ngrok-user-meetup

Session Status      online
Account             Lucky Ardiansyah (Plan: Free)
Update              update available (version 3.22.1, Ctrl-U to update)
Version             3.20.0
Region              Asia Pacific (ap)
Latency             36ms
Web Interface       http://127.0.0.1:4040
Forwarding           http://77dd-175-45-191-10.ngrok-free.app -> http://localhost:8080

Connections          ttl    opn    rt1    rt5    p50    p90
221                0      0.20   0.19   0.41   0.73
```

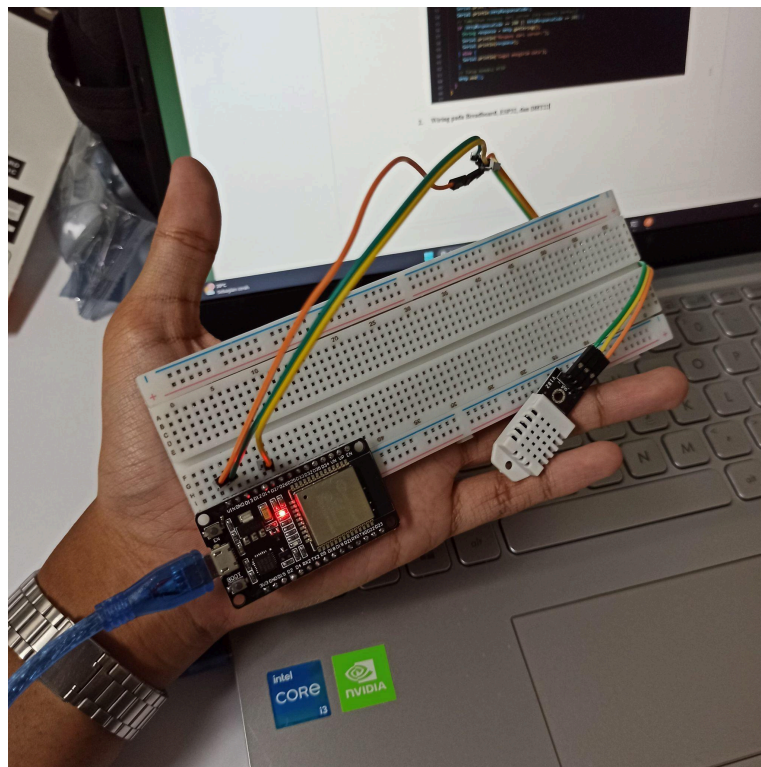
3. Coding pada main.cpp

```
PIO Home main.cpp platformio.ini .env
wokwi_internet > src > main.cpp > setup()
1  #include <Arduino.h>
2  #include <WiFi.h>
3  #include <HTTPClient.h>
4  #include "DHT.h"
5  #define DHTPIN 27
6  #define DHTTYPE DHT22
7  DHT dht(DHTPIN, DHTTYPE);
8  // Ganti dengan kredensial WiFi Anda
9  const char* ssid = "Lucky";
10 const char* password = "qwertyui";
11 unsigned long previousMillis = 0;
12 const long interval = 5000; // Interval 5 detik (5000 ms)
13 void setup() {
14     Serial.begin(115200);
15     // Hubungkan ke WiFi
16     WiFi.begin(ssid, password);
17     Serial.print("Menghubungkan ke WiFi");
18     while (WiFi.status() != WL_CONNECTED) {
19         delay(500);
20         Serial.print(".");
21     }
22     Serial.println(" Terhubung!");
23     dht.begin();
24
25     // Tunggu sebentar agar koneksi stabil
26     delay(1000);
27 }
28 void loop() {
29     unsigned long currentMillis = millis();
30     // Lakukan POST setiap interval yang telah ditentukan
31     if (currentMillis - previousMillis >= interval) {
32         previousMillis = currentMillis;
33         float h = round(dht.readHumidity());
34         // Read temperature as Celsius (the default)
35         float t = round(dht.readTemperature());
36         // Check if any reads failed and exit early (to try again).
37         if (isnan(h) || isnan(t)) {
38             Serial.println(F("Failed to read from DHT sensor!"));
39             return;
40         }
41         // Compute heat index in Celsius (isFahrenheit = false)
42         float hic = dht.computeHeatIndex(t, h, false);
43         // Inisialisasi HTTPClient
44         HTTPClient http;
45         String url = "http://77dd-175-45-191-10.ngrok-free.app/api/posts"; // Ganti dengan URL ngrok yang benar
46         http.begin(url); // Menggunakan HTTP, bukan HTTPS
47         http.addHeader("Content-Type", "application/json");
48         String payload = "{\"nama_sensor\":\"Sensor GD\", \"nilai1\":\"" + String(h) + ", \"nilai2\":\"" + String(t) + "\"}";
49         Serial.println(payload); // Untuk melihat apakah payload sudah terbentuk dengan benar
50         // Kirim POST request
51         int httpStatusCode = http.POST(payload);
52         // Tampilkan kode respons HTTP
53         Serial.print("Kode respons HTTP: ");
54         Serial.println(httpStatusCode);
55         // Tampilkan respons dari server jika request berhasil
56         if (httpStatusCode == 200 || httpStatusCode == 201) {
57             String response = http.getString();
58             Serial.println("Respons dari server:");
59             Serial.println(response);
60         } else {
61             Serial.println("Gagal mengirim data");
62         }
63         // Tutup koneksi HTTP
64         http.end();
65     }
66 }
67
```

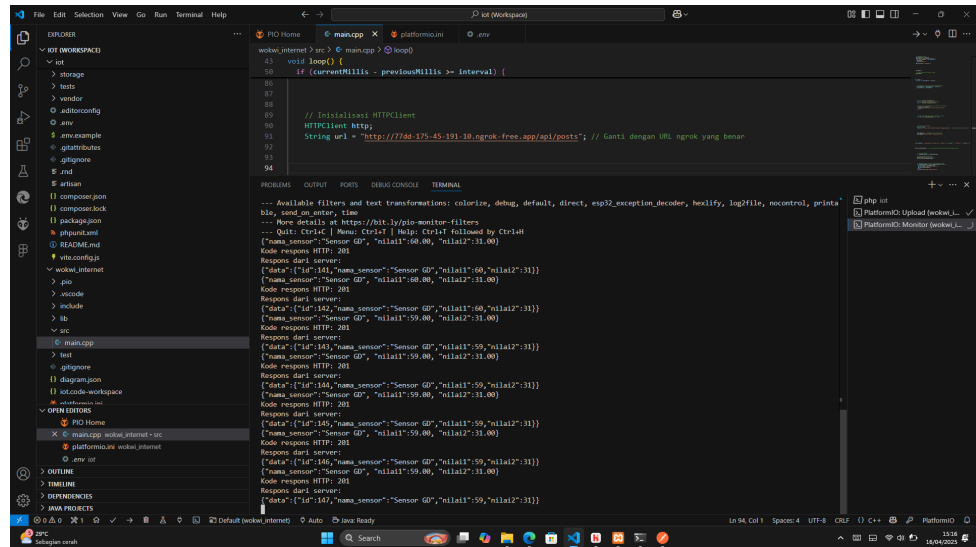

4. Modifikasi platformio.ini

```
PIO Home | main.cpp | platformio.ini | .env
wokwi_internet > platformio.ini
1 ; PlatformIO Project Configuration File
2 ;
3 ; Build options: build flags, source filter
4 ; Upload options: custom upload port, speed and e
5 ; Library options: dependencies, extra library st
6 ; Advanced options: extra scripting
7 ;
8 ; Please visit documentation for the other options
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 upload_port = COM3
16 monitor_port = COM3
17 monitor_speed = 115200
18 lib_deps =
19     adafruit/DHT sensor library@^1.4.4
20     adafruit/Adafruit Unified Sensor@^1.1.14
21
```

5. Wiring pada Breadboard, ESP32, dan DHT22



6. Hasil pada Terminal



7. Hasil pada phpMyAdmin

