LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Praktik Real Hardware ESP32**

***Imran Darajati***

***Fakultas Vokasi, Universitas Brawijaya***

***Email :*** [***imran.darajati@gmail.com***](mailto:imran.darajati@gmail.com)

**Abstrac**

Praktik ini bertujuan untuk menerapkan pemrograman ESP32 secara langsung menggunakan perangkat keras (real hardware). Terdapat dua percobaan yang dilakukan. Percobaan pertama menggunakan ESP32 untuk mengendalikan dua buah lampu LED secara bersamaan maupun bergantian. Percobaan kedua menggunakan ESP32 dengan sensor suhu, yang kemudian mengirimkan data ke database MySQL melalui API Laravel menggunakan Ngrok. Semua praktik dijalankan melalui VS Code dan PlatformIO tanpa menggunakan simulasi.

*Kata kunci: ESP32, LED, Sensor Suhu, Ngrok, Real Hardware, API, IoT, PlatformIO*

1. **Pendahuluan**

**1.1 Latar Belakang**

Setelah melakukan serangkaian simulasi menggunakan Wokwi, penting untuk menguji hasil pemrograman pada perangkat nyata agar dapat memahami cara kerja dan permasalahan yang muncul di dunia nyata. ESP32 merupakan mikrokontroler yang mendukung koneksi WiFi dan cocok untuk keperluan Internet of Things (IoT).

**1.2 Tujuan Eksperimen**

Tujuan dari eksperimen ini adalah :

1. Mengendalikan dua buah LED menggunakan ESP32.
2. Membaca data suhu dari sensor dan mengirimkannya ke database melalui API.
3. Mempelajari alur koneksi ESP32 secara langsung dengan komputer.
4. **Metodologi**

**2.1 Alat dan Bahan**

* Software yang Digunakan:

1. Visual Studio Code (VS Code)
2. PlatformIO (untuk pemrograman ESP32)
3. Ngrok (untuk tunneling API)
4. Database MySQL (iot\_25)
5. API Laravel (sudah dibuat sebelumnya)

* Hardware yang Digunakan:

1. 1 Board ESP32 DevKit v1
2. 2 Lampu LED
3. Breadboard & kabel jumper
4. Sensor Suhu (DHT11/DHT22)
5. Kabel USB micro
6. Laptop (sebagai perangkat penghubung)

**2.2 Langkah Implementasi**

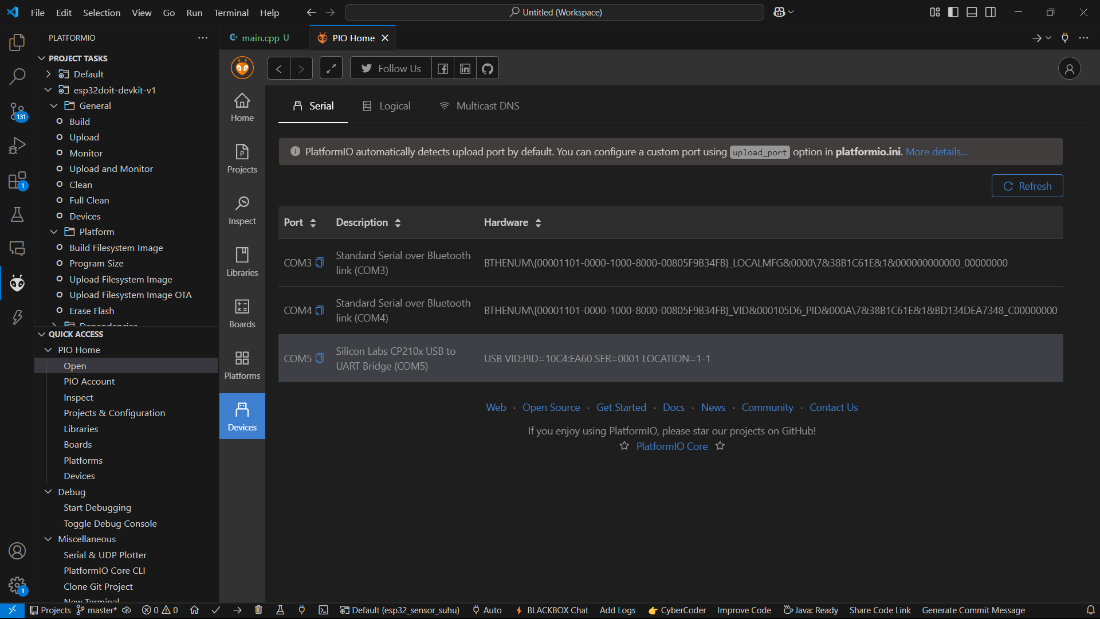
1. Langkah Persiapan Awal:

* Unduh dan instal driver Silicon Labs dari:

<https://www.silabs.com/developer-tools/usb-to-uart-bridge0-vcp-drivers?tab=downloads>

* Sambungkan ESP32 ke laptop menggunakan kabel USB.
* Buka VS Code dan pastikan perangkat terdeteksi oleh PlatformIO.

Seperti ini contohnya :



* Buat proyek baru di PlatformIO.

1. Percobaan 1: Kendali LED

* Sambungkan dua buah LED ke pin digital ESP32 menggunakan kabel jumper.
* Tulis kode untuk menyalakan dan mematikan LED secara bersamaan dan bergantian.
* Kode untuk menyalakan dan mematikan LED secara bersamaan :

#include <Arduino.h> // Wajib untuk PlatformIO + ESP32

// Deklarasi pin LED

int lampu1 = 25;

int lampu2 = 33;

void setup() {

Serial.begin(115200);

Serial.println("ESP32 Blinking 2 LEDs Together");

// Atur semua pin sebagai OUTPUT

pinMode(lampu1, OUTPUT);

pinMode(lampu2, OUTPUT);

}

void loop() {

// Nyalakan kedua lampu secara bersamaan

digitalWrite(lampu1, HIGH);

digitalWrite(lampu2, HIGH);

Serial.println("Kedua lampu ON");

delay(500);

// Matikan kedua lampu secara bersamaan

digitalWrite(lampu1, LOW);

digitalWrite(lampu2, LOW);

Serial.println("Kedua lampu OFF");

delay(500);

}

* Kode untuk menyalakan dan mematikan LED secara bergantian :

#include <Arduino.h> // Wajib untuk PlatformIO + ESP32

// Deklarasi pin LED

int lampu1 = 25;

int lampu2 = 33;

void setup() {

Serial.begin(115200);

Serial.println("ESP32 Blinking 2 LEDs Alternately");

// Atur semua pin sebagai OUTPUT

pinMode(lampu1, OUTPUT);

pinMode(lampu2, OUTPUT);

}

void loop() {

// Nyalakan lampu 1, matikan lampu 2

digitalWrite(lampu1, HIGH);

digitalWrite(lampu2, LOW);

Serial.println("Lampu 1 ON, Lampu 2 OFF");

delay(500);

// Nyalakan lampu 2, matikan lampu 1

digitalWrite(lampu1, LOW);

digitalWrite(lampu2, HIGH);

Serial.println("Lampu 1 OFF, Lampu 2 ON");

delay(500);

}

* Ubah code pada platform.ini menggunakan code ini

[env:esp32doit-devkit-v1]

platform = espressif32

board = esp32doit-devkit-v1

framework = arduino

upload\_port = COM5

monitor\_port = COM5

* Build terlebih dahulu kode yang sudah diberikan melalui VS Code.
* Upload program ke ESP32 melalui VS Code.
* Amati perubahan pada LED secara langsung.

1. Percobaan 2: Sensor Suhu & Pengiriman Data

* Hubungkan sensor suhu ke ESP32.
* Hubungkan ESP32 ke WiFi yang tersedia.
* Aktifkan layanan Ngrok untuk membuka API Laravel.
* Jalankan API laravel kembali dengan perintah

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

* Kemudian jalankan NGROK

ngrok http --scheme=http 8080

* Buat kode untuk membaca suhu dari sensor dan mengirimkan data melalui HTTP POST ke API Laravel.
* Tulis kode berikut :

#include <Arduino.h>

#include <WiFi.h>

#include <HTTPClient.h>

#include "DHT.h"

#define DHTPIN 27

#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

// Ganti dengan kredensial WiFi Anda

const char\* ssid = "id\_kyekti";

const char\* password = "lkjhgfdsa";

unsigned long previousMillis = 0;

const long interval = 5000; // Interval 5 detik (5000 ms)

void setup() {

Serial.begin(115200);

// Hubungkan ke WiFi

WiFi.begin(ssid, password);

Serial.print("Menghubungkan ke WiFi");

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println(" Terhubung!");

dht.begin();

// Tunggu sebentar agar koneksi stabil

delay(1000);

}

void loop() {

unsigned long currentMillis = millis();

// Lakukan POST setiap interval yang telah ditentukan

if (currentMillis - previousMillis >= interval) {

previousMillis = currentMillis;

float h = round(dht.readHumidity());

// Read temperature as Celsius (the default)

float t = round(dht.readTemperature());

// Check if any reads failed and exit early (to try again).

if (isnan(h) || isnan(t)) {

Serial.println(F("Failed to read from DHT sensor!"));

return;

}

// Compute heat index in Celsius (isFahreheit = false)

float hic = dht.computeHeatIndex(t, h, false);

// Inisialisasi HTTPClient

HTTPClient http;

String url = "http://6cd9-203-78-118-221.ngrok-free.app/api/posts"; // Ganti dengan URL ngrok yang benar

http.begin(url); // Menggunakan HTTP, bukan HTTPS

http.addHeader("Content-Type", "application/json");

String payload = "{\"nama\_sensor\":\"Sensor GD\", \"nilai1\":" + String(h) + ", \"nilai2\":" + String(t) + "}";

Serial.println(payload); // Untuk melihat apakah payload sudah terbentuk dengan benar

// Kirim POST request

int httpResponseCode = http.POST(payload);

// Tampilkan kode respons HTTP

Serial.print("Kode respons HTTP: ");

Serial.println(httpResponseCode);

// Tampilkan respons dari server jika request berhasil

if (httpResponseCode == 200 || httpResponseCode == 201) {

String response = http.getString();

Serial.println("Respons dari server:");

Serial.println(response);

} else {

Serial.println("Gagal mengirim data");

}

// Tutup koneksi HTTP

http.end();

}

}

* Ubah code pada platform.ini menggunakan code ini

[env:esp32doit-devkit-v1]

platform = espressif32

board = esp32doit-devkit-v1

framework = arduino

upload\_port = COM5

monitor\_port = COM5

monitor\_speed = 115200

lib\_deps =

adafruit/DHT sensor library@^1.4.4

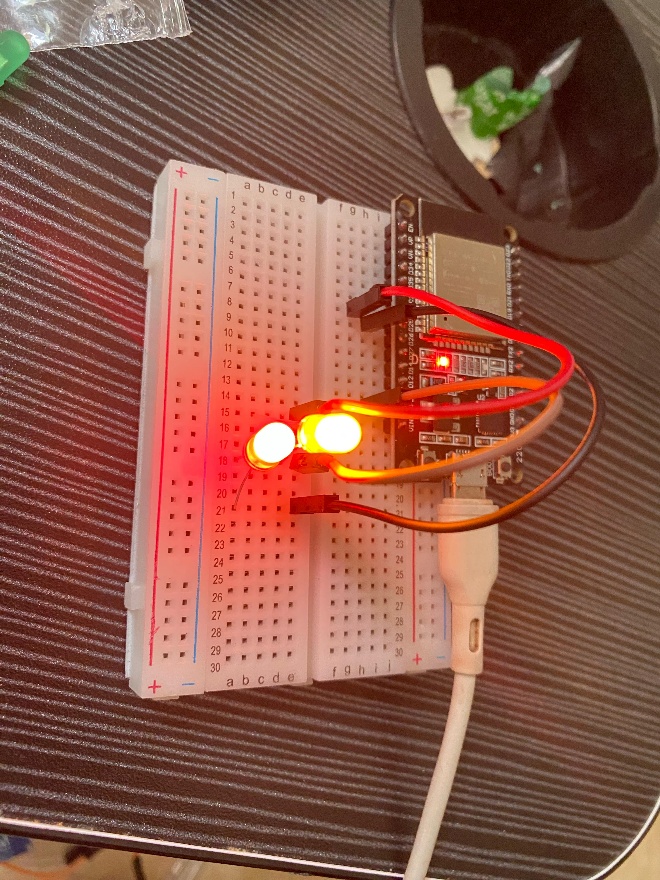
adafruit/Adafruit Unified [Sensor@^1.1.14](mailto:Sensor@%5e1.1.14)

* Build terlebih dahulu kode yang sudah diberikan melalui VS Code.
* Upload program ke ESP32 melalui VS Code.
* Periksa hasil pengiriman data melalui database iot\_25.

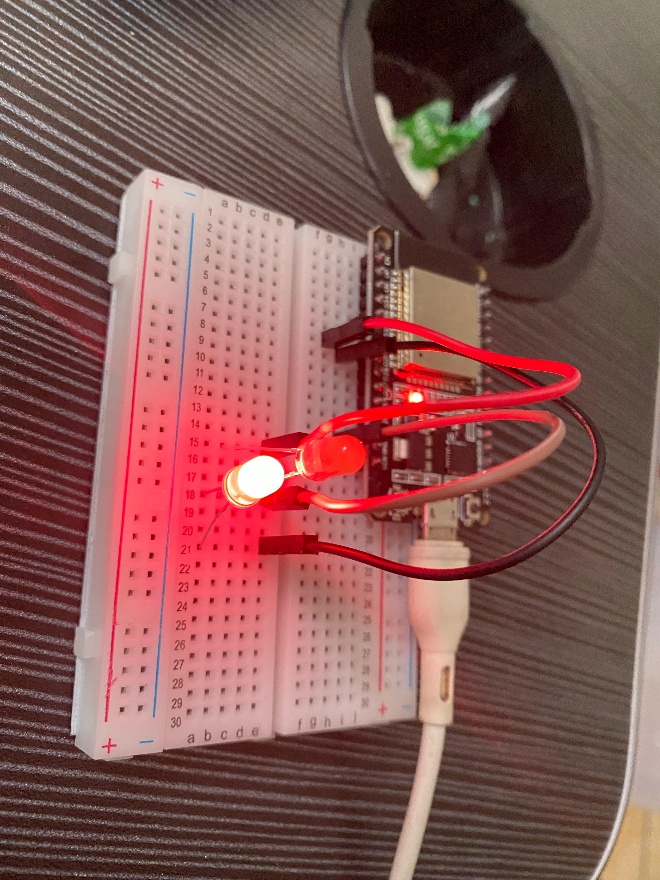
1. **Hasil dan Pembahasan**
2. Percobaan 1: Kendali LED

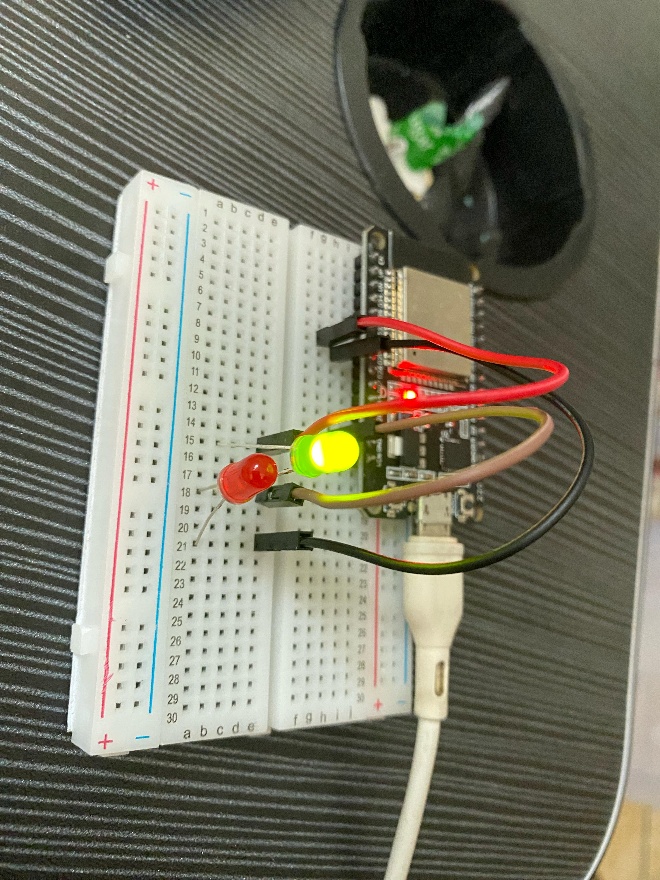
Berikut adalah tampilan hasil pengujian kendali LED menggunakan ESP32 secara langsung melalui perangkat keras:

* ESP32 yang terhubung ke dua buah LED pada breadboard, dengan konfigurasi kabel jumper tanpa resistor:
* LED menyala secara bersamaan :

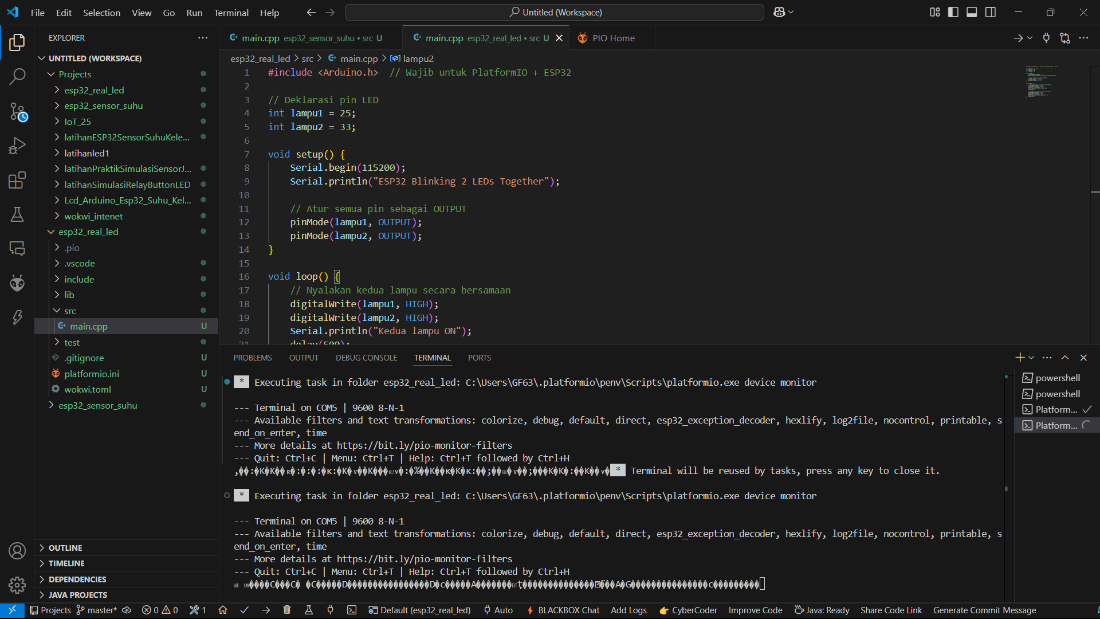


* LED menyala secara bergantian :

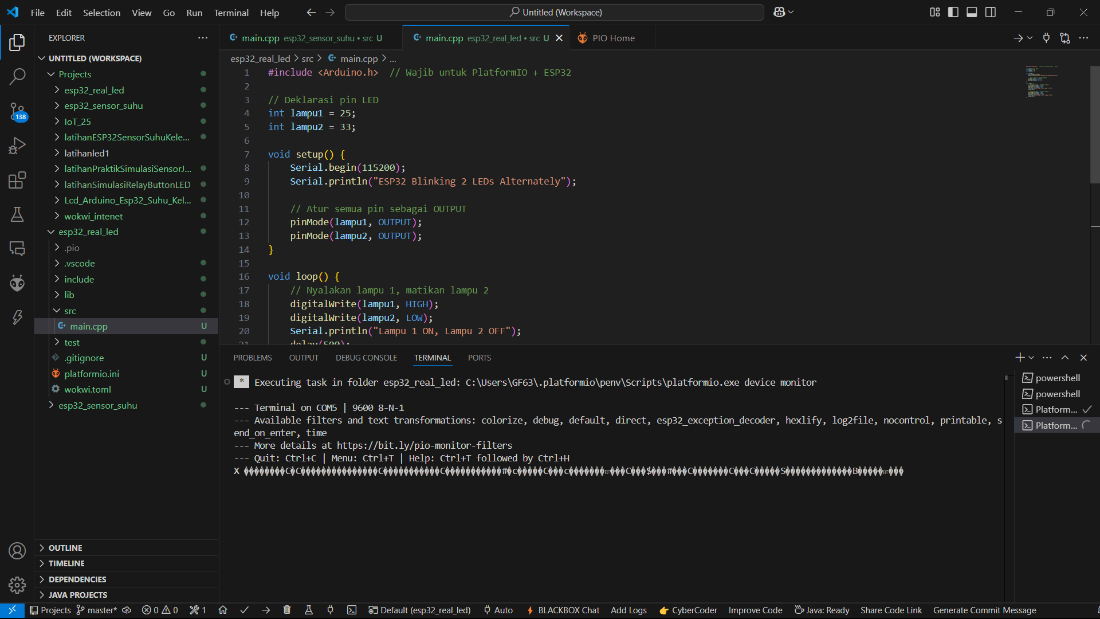




* Status LED yang menyala dan mati secara bersamaan dan bergantian ditampilkan melalui Serial Monitor:
* Status LED yang menyala dan mati secara bersamaan di Serial Monitor :



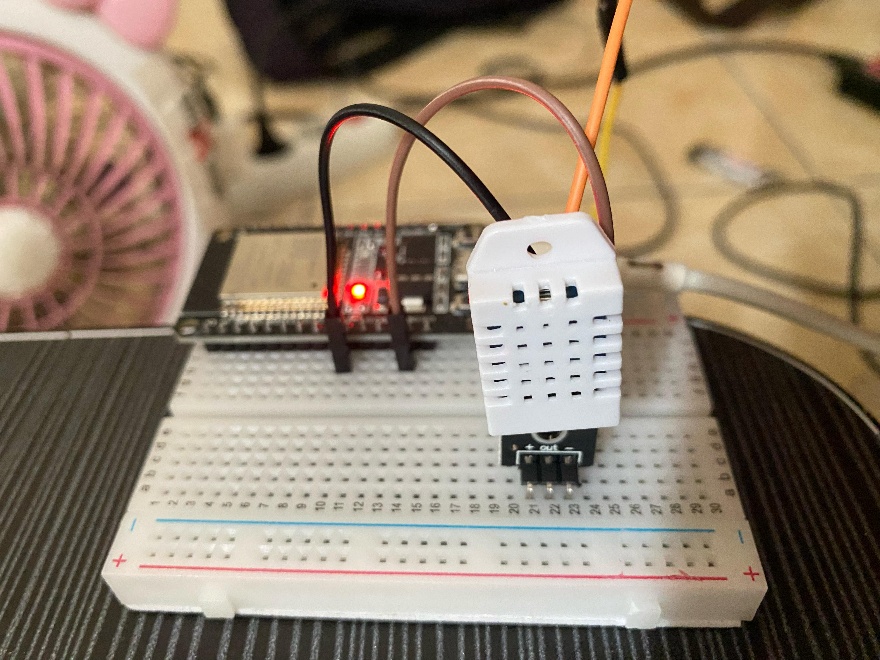
* Status LED yang menyala dan mati secara bergantian di Serial Monitor :



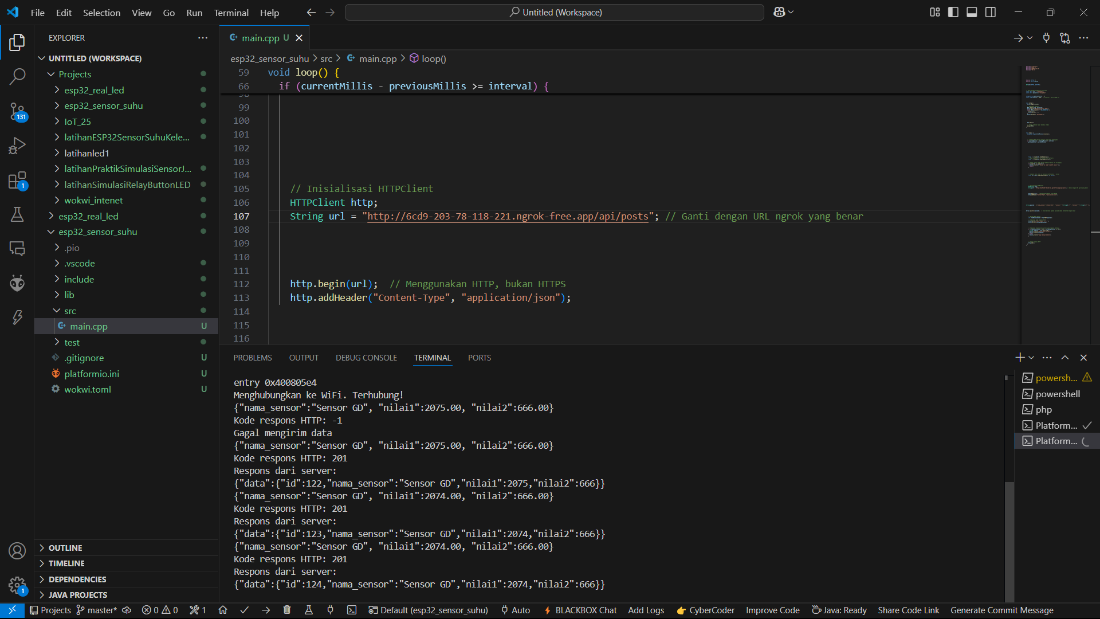
1. Percobaan 2: Sensor Suhu & Pengiriman ke Database

Berikut adalah tampilan hasil pengujian pengiriman data sensor suhu ke API Laravel melalui ESP32 secara real hardware:

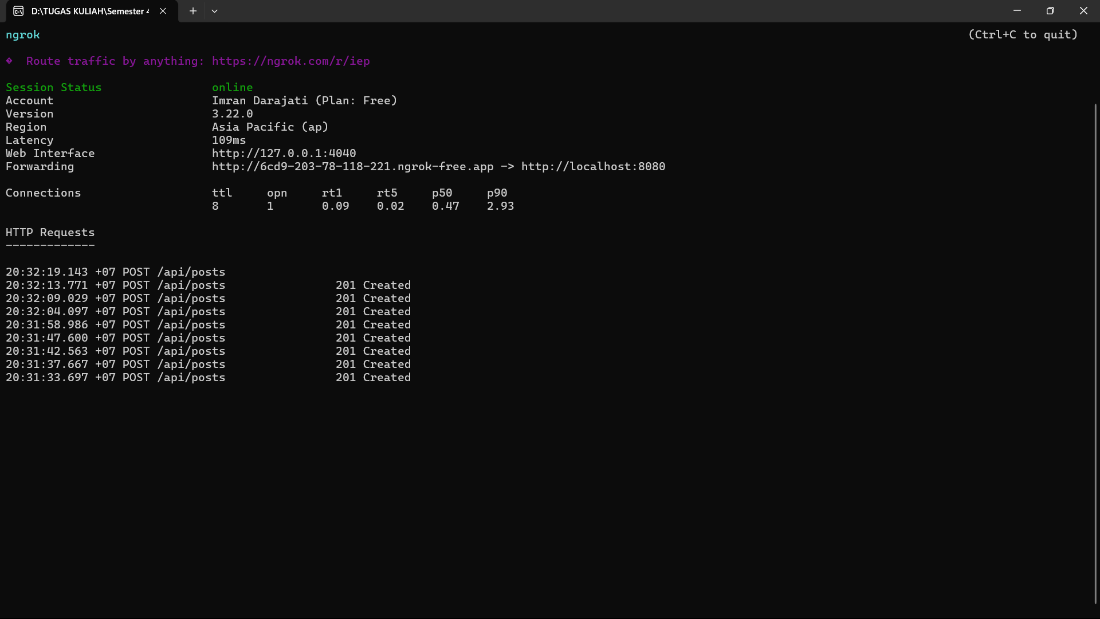
* ESP32 yang terhubung ke sensor suhu (DHT11/DHT22) dengan rangkaian di breadboard:



* Hasil pembacaan data suhu dari sensor yang ditampilkan di Serial Monitor:



* Tampilan URL Ngrok yang aktif dan digunakan oleh ESP32 untuk mengakses API Laravel secara online:



* Tampilan data suhu yang berhasil dikirim oleh ESP32 ke API Laravel dan tersimpan di database MySQL iot\_25:

