LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Praktik Akses API Melalui   
Simulasi WOKWI**

*Assyfaul Khoiriyah*

Fakultas Vokasi, Universitas Brawijaya

[assyfaassyfa0@gmail.com](mailto:assyfaassyfa0@gmail.com)

**Abstract (Abstrak)**

Internet of Things (IoT) adalah konsep yang menghubungkan perangkat elektronik ke internet untuk bertukar data secara otomatis. Dalam eksperimen ini, dilakukan simulasi pengiriman data sensor menggunkan ESP32 melalui API yang dibuat dengan laravel dan diuji menggunakan wokwi simulator. Tujuan dari eksperimen ini adalah untuk memahami cara menghubungkan ESP32 ke jaringan WiFi, menggunakan API laravel untuk mngolah data, serta mnguji komunikasi IoT menggunkan simulasi. Hasil yang diperoleh menunjukkan bahwa data sensor dapat dikirim dengan sukses ke server menggunakan metode HTTP POST, dan sistem berjalan dengan baim pada lingkungan simulasi wokwi.

**Kata kunci:** IoT, ESP32, Wokwi, API, Laravel

1. **Introduction (Pendahuluan)**
   1. **Latar Belakang**

Perkembamham teknologi informasi dan komunikasi telah membawa dampak signifikan pada berbagai aspek kehidupan, salah satunya adalah penerapan Internet of Things (IoT) . IoT merupakan konsep yang memungkinkan perangkat elektronik untuk terhubung dan saling berkomunikasi melalui jaringan internet, sehingga memungkinkan pengambilan keputusan secara otomatis berdasarkan data yang dikumpulkan. Dalam industri dan kehidupan sehari hari, IoT digunakan untuk berbagai keperluan seperti pemantauan lingkungan, sistem otomatisasi rumah,dan pengelolaan data sensor di bidang kesehatan, pertanian, serta industri manufaktur. Salah tau teknologi yang sering digunakan dalam sistem IoT adalah mikrokontroler ESP32, yang memiliki kemampuan komunikasi nirkabel menggunkan wifi serta mendukung berbagai sensor untuk pengambilan data.

Simulasi IoT menjadi bagian penting dalm pengujian sistem sebelum diterapkan dalam lingkungan nyata. Wokwi simulator adalah salah satu platform yang memungkinkan pengujian perangkat Keras secara virtual tanpa memerlukan perangkat fisik. Dengan bantuan wokwi, kita dapat menguji kode program dan komunikasi antar perangkat IoT dengan efisien.Dalam eksperimen ini, dilakukan simulasi akses API menggunakan ESP32 dan laravel untuk memahami bagaimana data sensor dapat dikirim dan diproses oleh server. Data sensor suhu dan kelembapan akan dikirim ke API yang dibuat dengan laravel, dan hasilya akan dianalisis untuk mengevaluasi keberhasilan komunikasi IoT dalam lingkungan simulasi.

**1.2 Tujuan Eksperimen**

1. Mempelajari cara menjalankan API menggunakan laravel
2. Menghubungkan ESP32 ke jaringan WiFi
3. Menggunakan wokwi simulator untuk menguji koneksi IoT dengan API
4. Mengirim data sensor suhu dan kelembapan ke server menggunakan HTTP.
5. Mengevaluasi respons server terhadap data yang dikirim dari ESP32
6. **Methodology (Metodologi)**

**2.1 Tools & Materials (Alat dan Bahan)**

* Laravel 11
* Laragon (apache dan MySQL)
* Visual studio code
* Ngrok
* Wokwi simulator
* PlatformIO
  1. **Implementation Steps (Langkah Implementasi)**

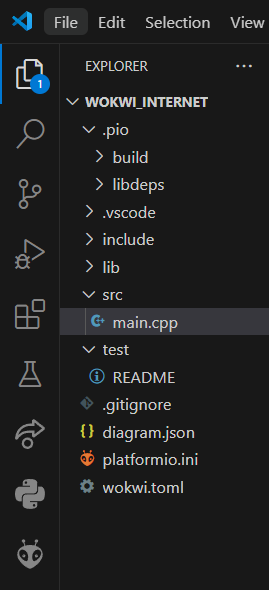
1. Nyalakan Laragon (Apache dan MySQL)
2. Search Folder LARAVEL-11
3. Buka Menggunakan Vscode
4. Buka Terminal ketik:

php artisan serve

1. Jalankan API laravel dengan perintah

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

1. Buat file baru wokwi simulator di platform.io

****

1. Script main.cpp

#include <WiFi.h>

#include <HTTPClient.h>

// Ganti dengan kredensial jaringan Wi-Fi Anda

// const char\* ssid = "Lab IT";

// const char\* password = "labit2024";

 const char\* ssid = "Wokwi-GUEST";

 const char\* password = "";

// URL lengkap server yang akan diakses

const char\* serverUrl = "http://e6d3-2405-8740-6315-3520-5928-26b-7835-cd79.ngrok- free.app/api/posts";

// Interval waktu antara setiap permintaan (dalam milidetik)

const unsigned long interval = 5000;

unsigned long previousMillis = 0;

void setup() {

  Serial.begin(115200);

  WiFi.begin(ssid, password);

  Serial.print("Menghubungkan ke WiFi...");

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

    delay(500);

    Serial.print(".");

  }

  Serial.println(" Terhubung!");

}

void loop() {

  unsigned long currentMillis = millis();

  // Periksa apakah interval waktu telah berlalu

  if (currentMillis - previousMillis >= interval) {

    previousMillis = currentMillis;

    if (WiFi.status() == WL\_CONNECTED) {

      HTTPClient http;

      // Inisialisasi HTTPClient dengan URL server

      http.begin(serverUrl);

      // Mengirim permintaan HTTP GET

      int httpResponseCode = http.GET();

      // Menampilkan kode status HTTP

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

      Serial.println(httpResponseCode);

      // Menutup koneksi

      http.end();

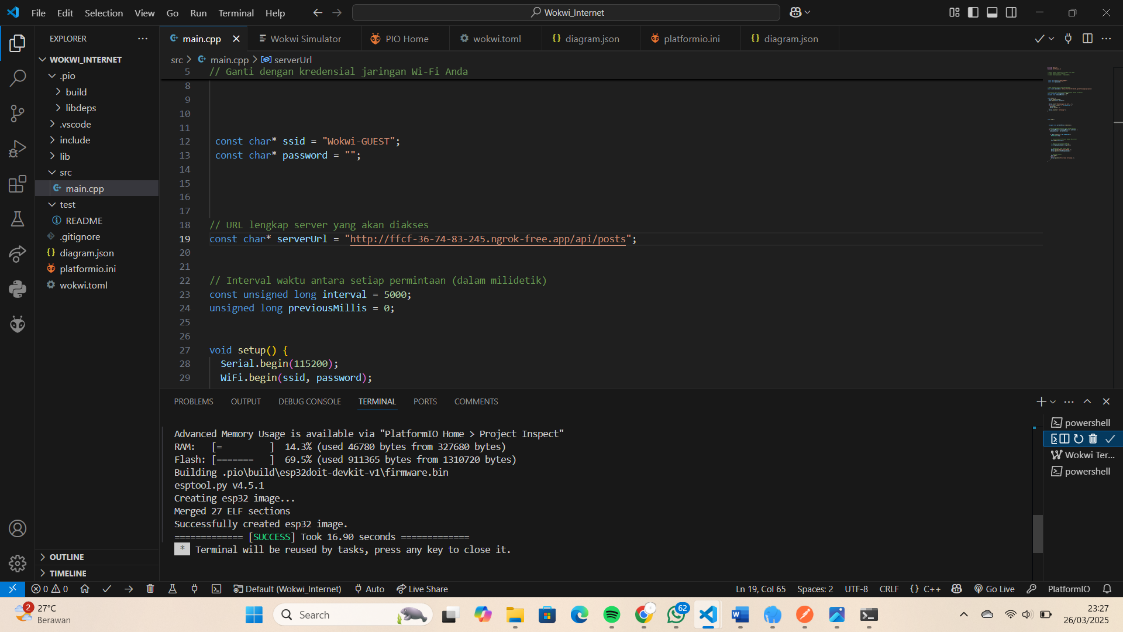
    } else {

      Serial.println("WiFi tidak terhubung.");

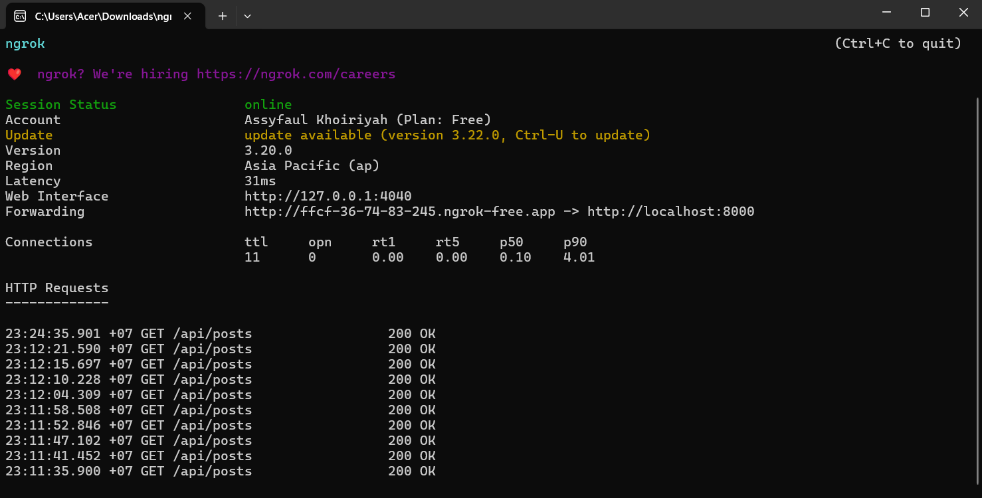
    }

  }

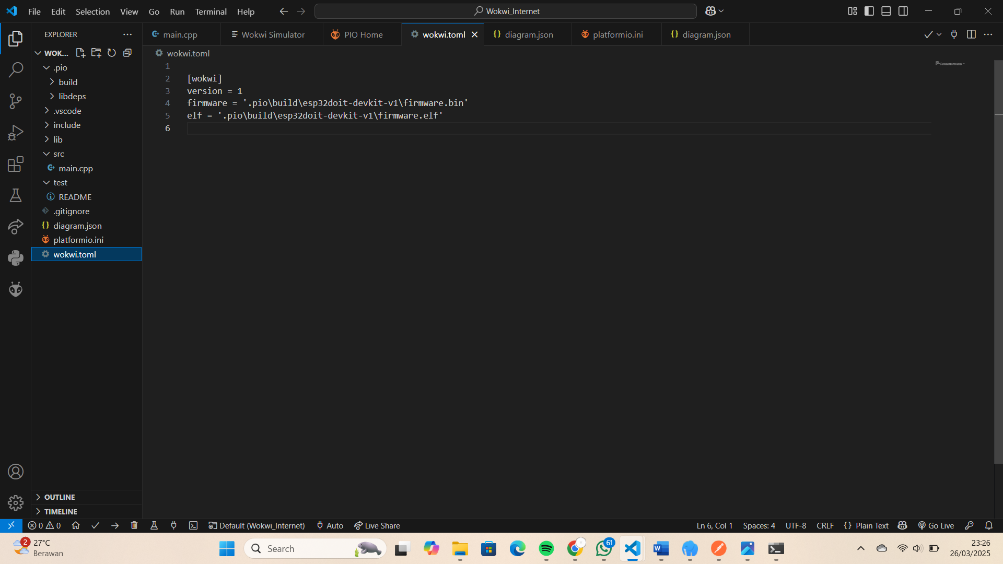
}



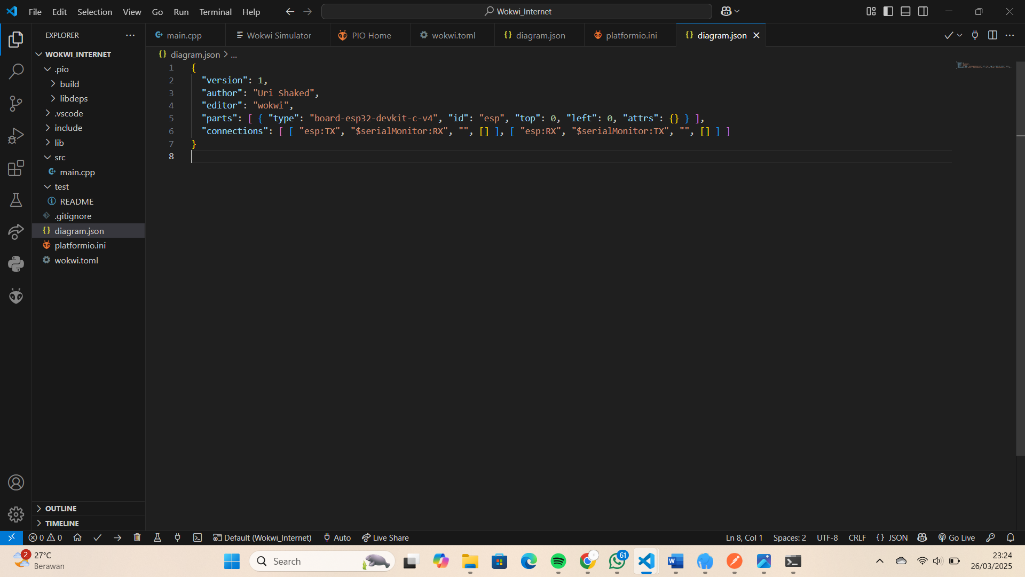
1. Menyesuaikan alamat port 8000



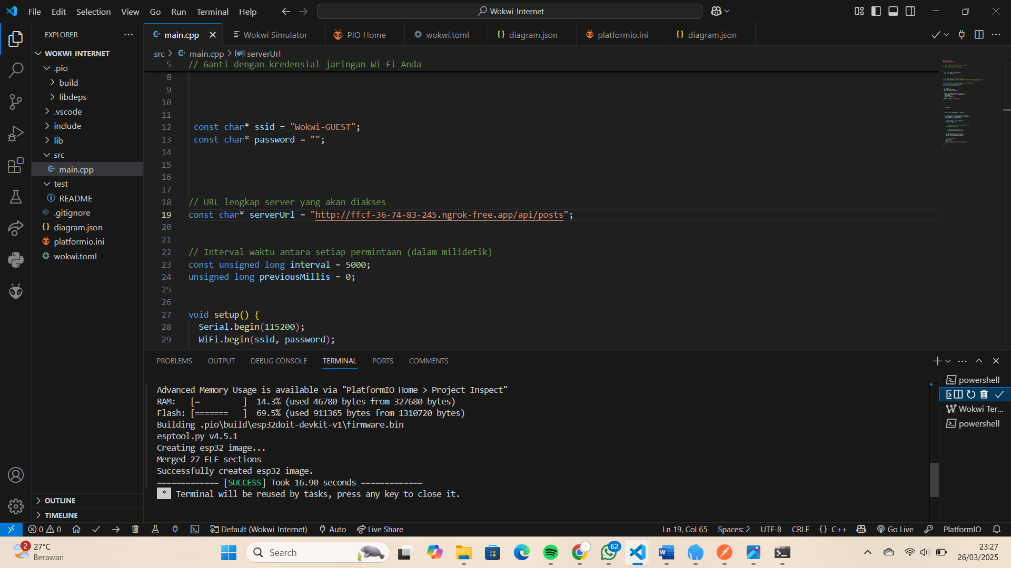
1. Tambahkan file wokwi.toml

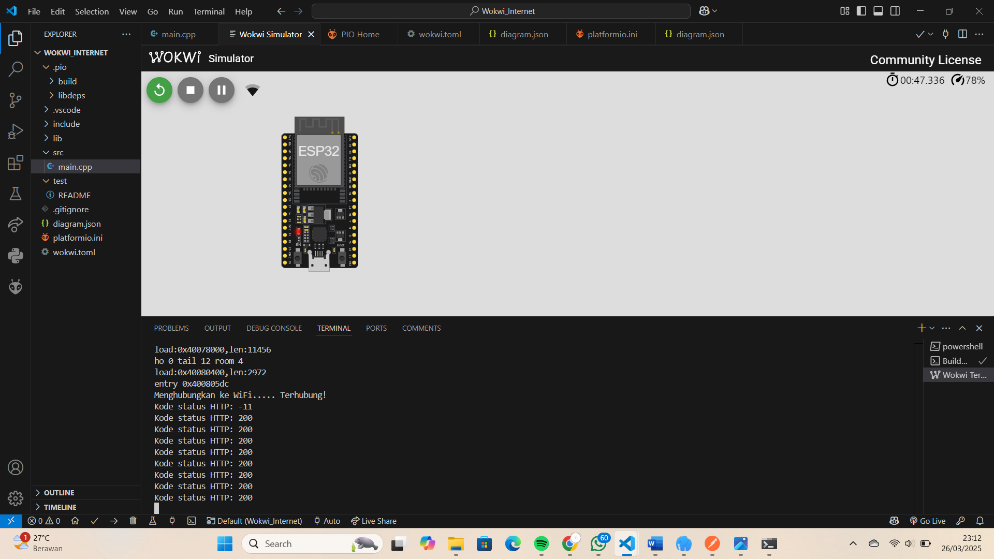


1. Tambahkan file diagram.json

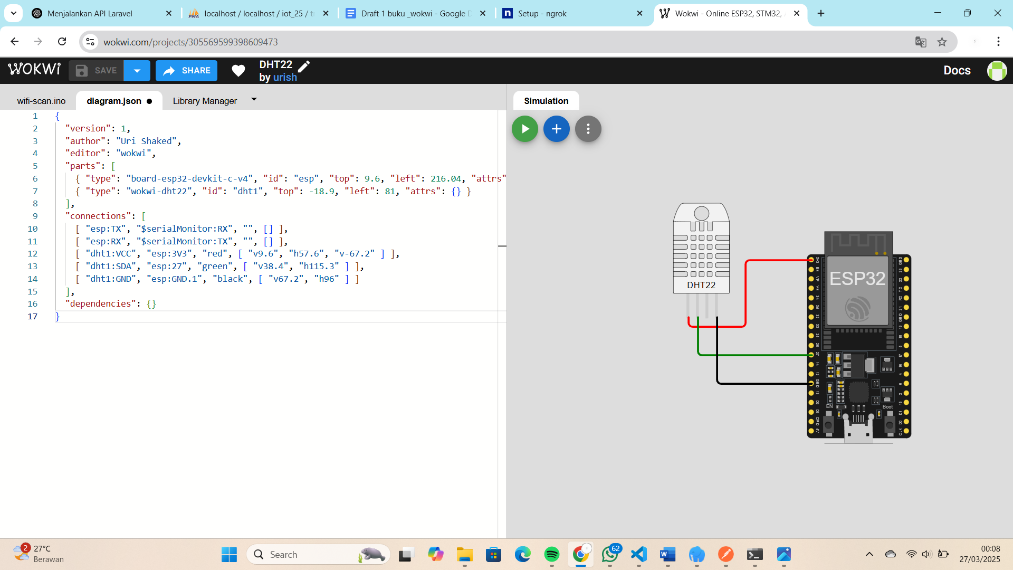


1. Lakukan build file main.cpp dan jalankan **> Wokwi Start Simulator**

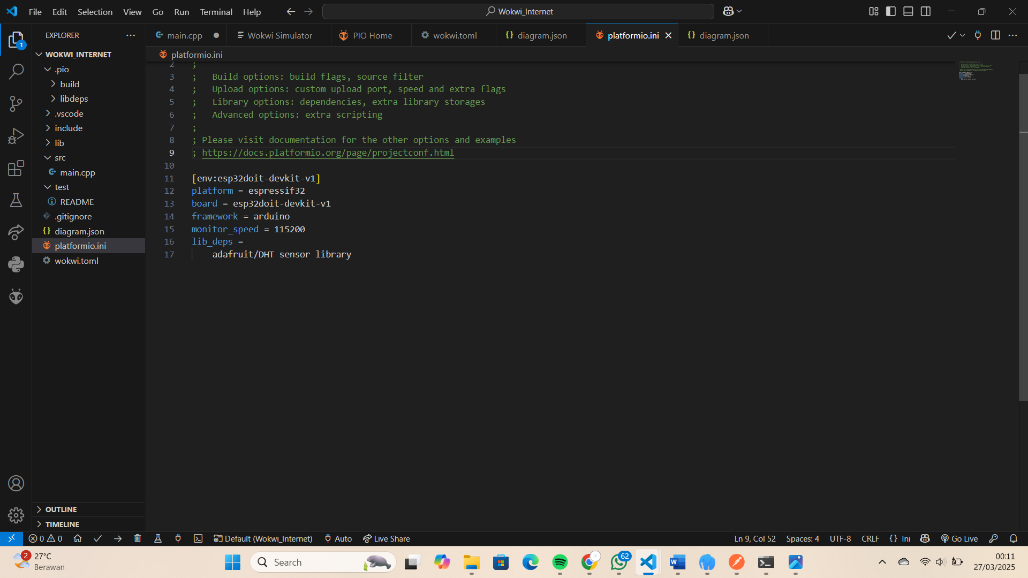


****

Setelah berhasil disambungkan maka selanjutnya melakukan modifikasi simulasi dengan menambahkan sensor suhu dan kelembapan

****

1. Kemudian ubah setting file **platformio.ini**



1. Modifikasi main.cpp

#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 = "Wokwi-GUEST";

const char\* password = "";

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://e6d3-2405-8740-6315-3520-5928-26b-7835-cd79.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");

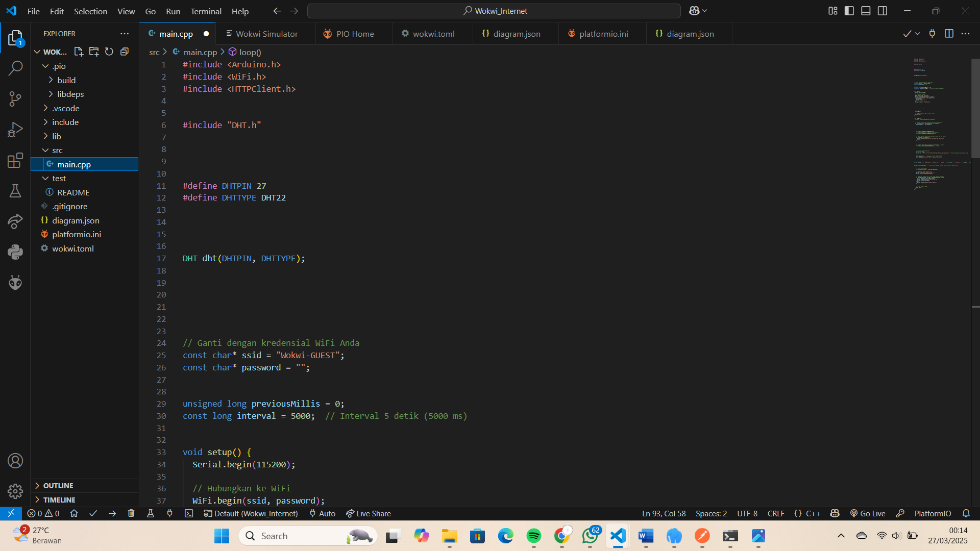
    }

    // Tutup koneksi HTTP

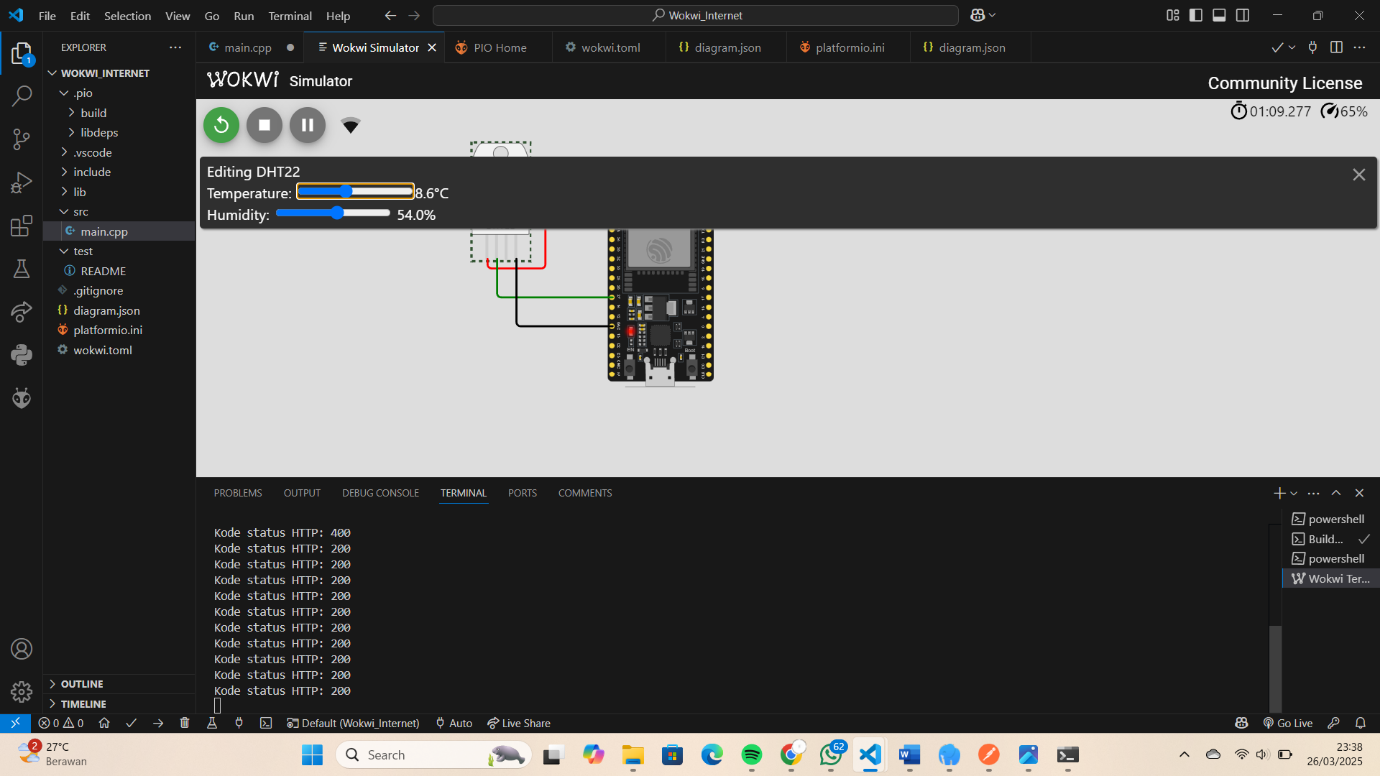
    http.end();

  }

**}**

****

Lalu jalankan simulasi **> Wokwi Start Simulator**

****

1. **Results and Discussion (Hasil dan Pembahasan)**

**3.1 Experimental Results (Hasil Eksperimen)**

ESP32 berhasil terhubung ke jaringan WiFi dan mampu mengakses API laravel melalui internet. Data suhu dan kelembapan berhasil dikirim ke server menggunakan metode HTTP POST, menunjukkan bahwa koneksi antara perangkat IoT dan server berjalan dengan baik. Penggunaan wokwi simulator dalam eksperimen ini terbukti efektif dalam mensimulasikan interaksi ESP32 dengan server. Dari hasil pengujian, server memberikan respons dengan formast JSON yang sesuai, menandakan bahwa sistem dapat berfungsi sebagaimana mestinya dalam lingkungan simulasi.

1. **Appendix (Lampiran, jika diperlukan)**

#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 = "Wokwi-GUEST";

const char\* password = "";

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://ffcf-36-74-83-245.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();

  }

}