**LAPORAN PRAKTIKUM**

**INTERNET OF THINGS**



**Disusun oleh :**

Atika Fitria Arifiana (233140700111064)

**PRODI D-III TEKNOLOGI INFORMASI**

**FAKULTAS VOKASI**

**UNIVERSITAS BRAWIJAYA**

**2025**

LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Praktik Akses API Melalui Simulasi WOKWI**

*Atika Fitria Arifiana*

*Fakultas Vokasi, Universitas Brawijaya*

*Email :* [*atikafit.arifiana@gmail.com*](mailto:atikafit.arifiana@gmail.com)

**ABSTRACT**

Dalam era Internet of Things (IoT), konektivitas antara perangkat keras dan perangkat lunak menjadi faktor utama dalam pengembangan sistem yang efisien. Praktikum ini bertujuan untuk mensimulasikan akses API melalui platform WOKWI dengan menggunakan ESP32. API yang diakses berbasis Laravel dan dijalankan melalui server lokal menggunakan NGROK agar dapat diakses dari internet. Dalam simulasi ini, ESP32 akan mengirimkan data sensor suhu dan kelembaban (DHT22) ke API Laravel dan menyimpannya ke dalam database. Hasil simulasi menunjukkan bahwa perangkat ESP32 berhasil menghubungkan diri ke jaringan Wi-Fi, mengakses API, serta mengirimkan data sensor dengan kode status HTTP 200. Keberhasilan ini menunjukkan bahwa ESP32 dapat digunakan sebagai perangkat IoT untuk sistem monitoring berbasis cloud.

Kata kunci: *ESP32, WOKWI, API, Laravel, NGROK, IoT*

**PENDAHULUAN**

* 1. **Latar belakang**

Perkembangan teknologi IoT memungkinkan perangkat untuk saling terhubung dan berbagi data secara real-time melalui internet. Salah satu tantangan utama dalam pengembangan sistem IoT adalah bagaimana perangkat dapat berkomunikasi dengan server menggunakan API. API memungkinkan pertukaran data antara perangkat keras seperti **ESP32** dan server berbasis **Laravel**.

Pada praktikum ini, digunakan simulator **WOKWI**, sebuah platform berbasis web yang memungkinkan pengujian dan simulasi perangkat keras tanpa memerlukan perangkat fisik. Untuk mengakses API yang berjalan di localhost, digunakan **NGROK** agar API tersebut dapat diakses dari internet. Dengan demikian, ESP32 dapat mengirimkan data dari sensor **DHT22** ke server dan menyimpannya ke dalam database.

* 1. **Tujuan eksperimen**

1. Mempelajari cara mengakses API menggunakan ESP32 melalui simulasi di WOKWI.
2. Menggunakan NGROK untuk membuat API Laravel dapat diakses secara global.
3. Mengirimkan data suhu dan kelembaban dari sensor DHT22 ke API dan menyimpannya ke database.
4. Menganalisis hasil koneksi dan pengiriman data melalui serial monitor.

**METODOLOGI**

**2.1 Alat dan Bahan**

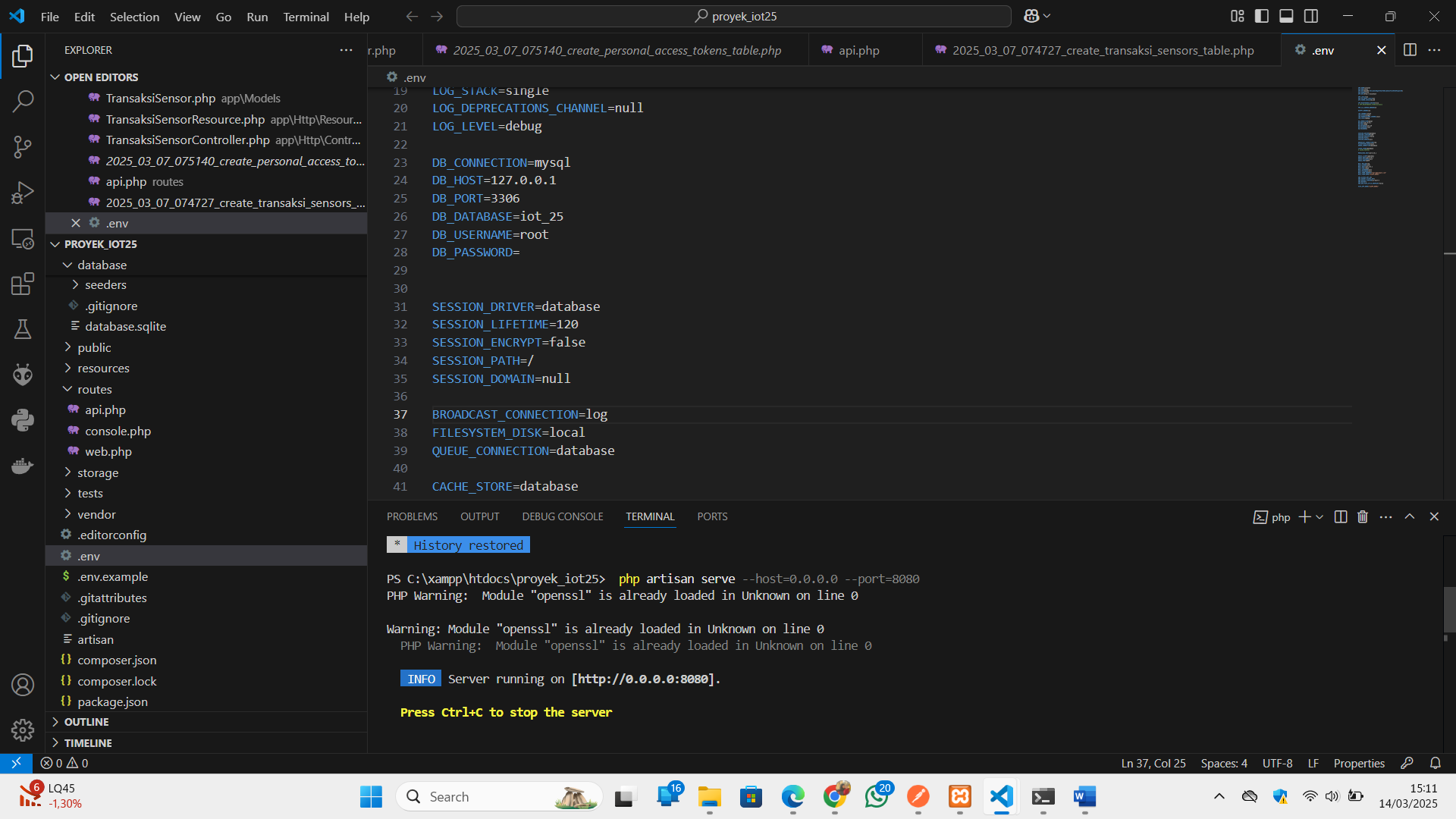
* ESP32 (disimulasikan melalui WOKWI)
* Sensor DHT22 (disimulasikan di WOKWI)
* PlatformIO (untuk pengembangan kode ESP32)
* Laravel (sebagai API server)
* NGROK (untuk menghubungkan API lokal ke internet)
* Database MySQL

**2.2 Langkah Implementasi**

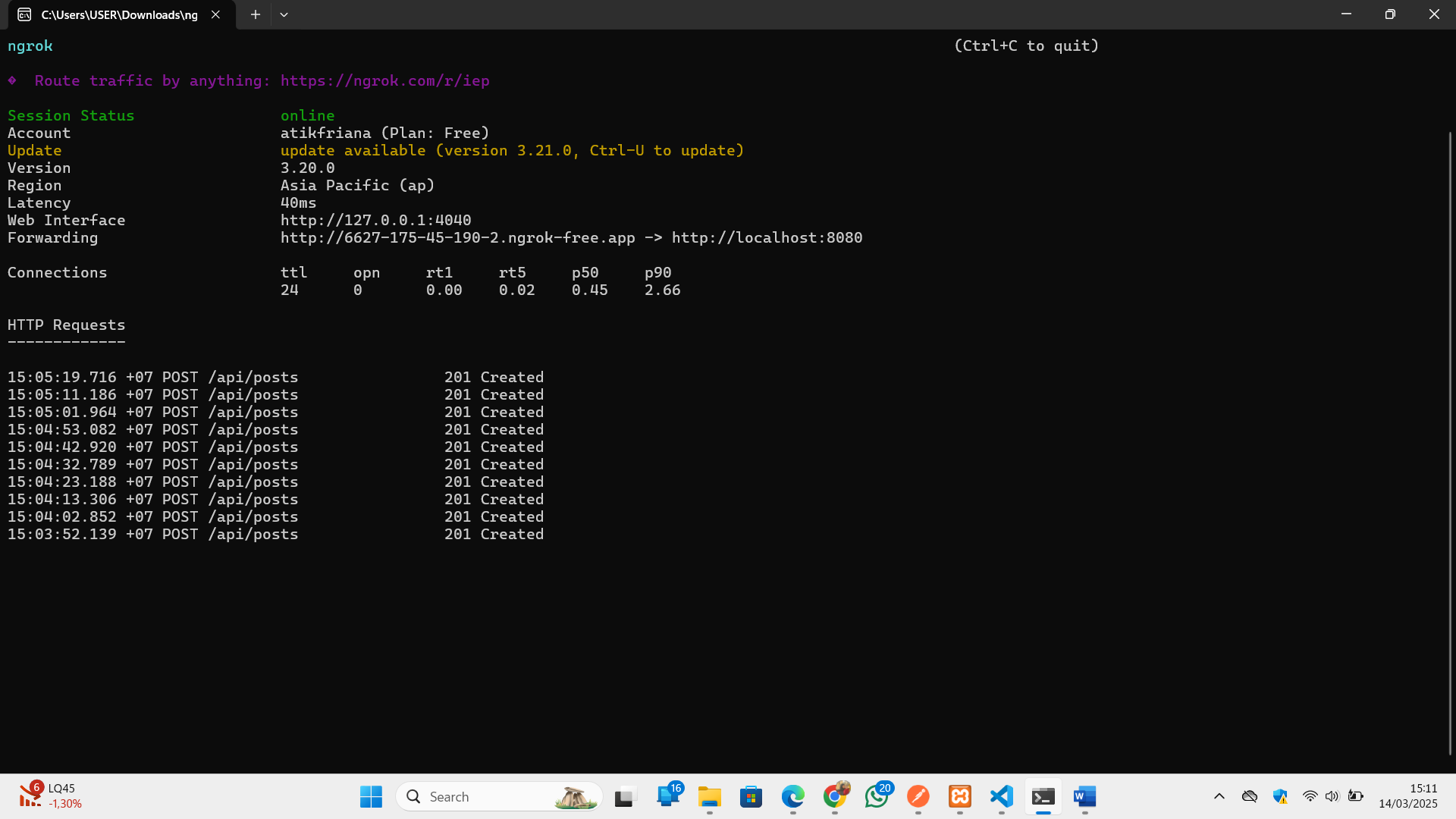
Proses implementasi eksperimen ini dilakukan dalam beberapa tahapan sebagai berikut:

1. **Menjalankan API Laravel**

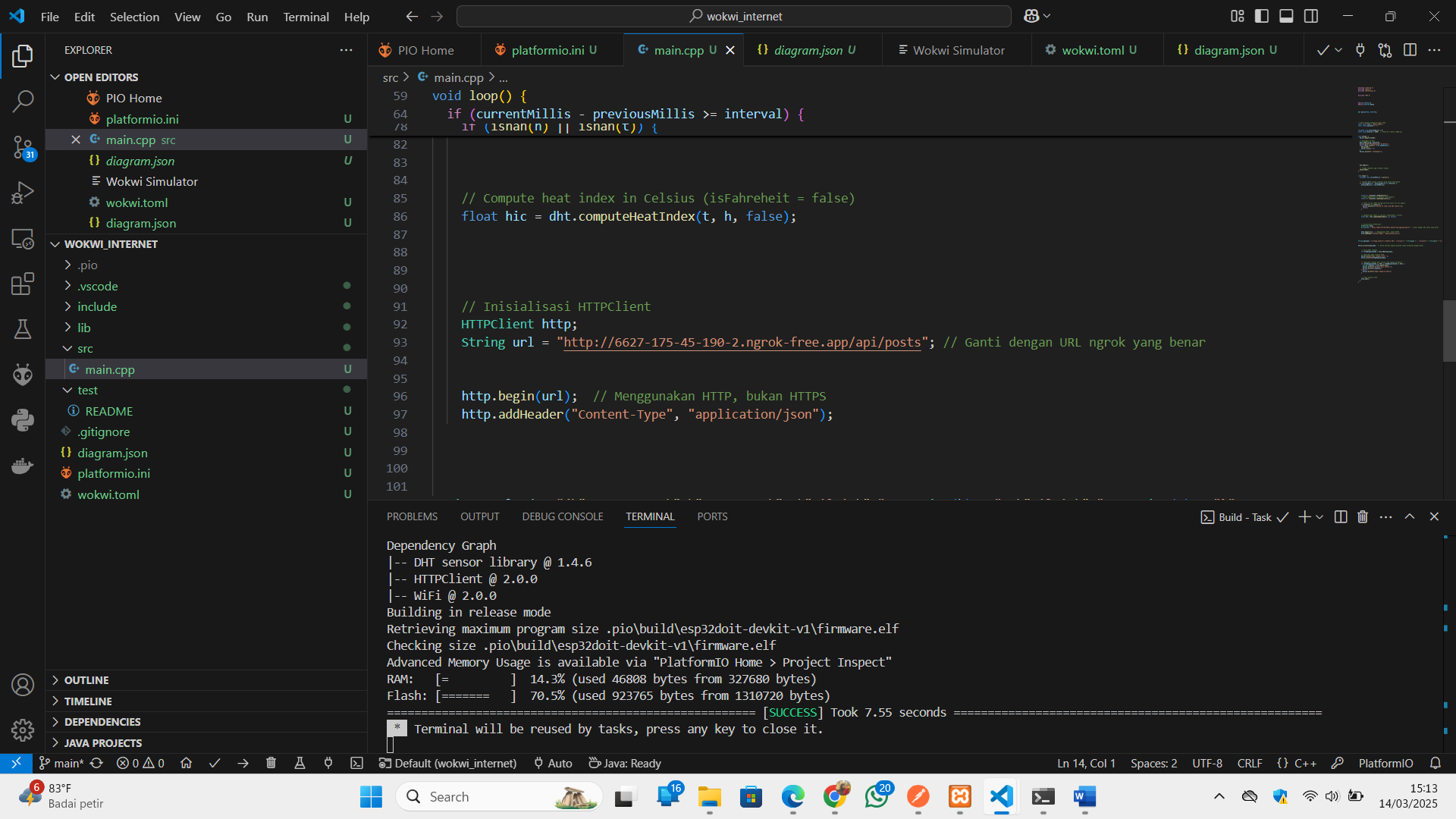
Jalankan perintah berikut di terminal untuk menjalankan API di localhost:



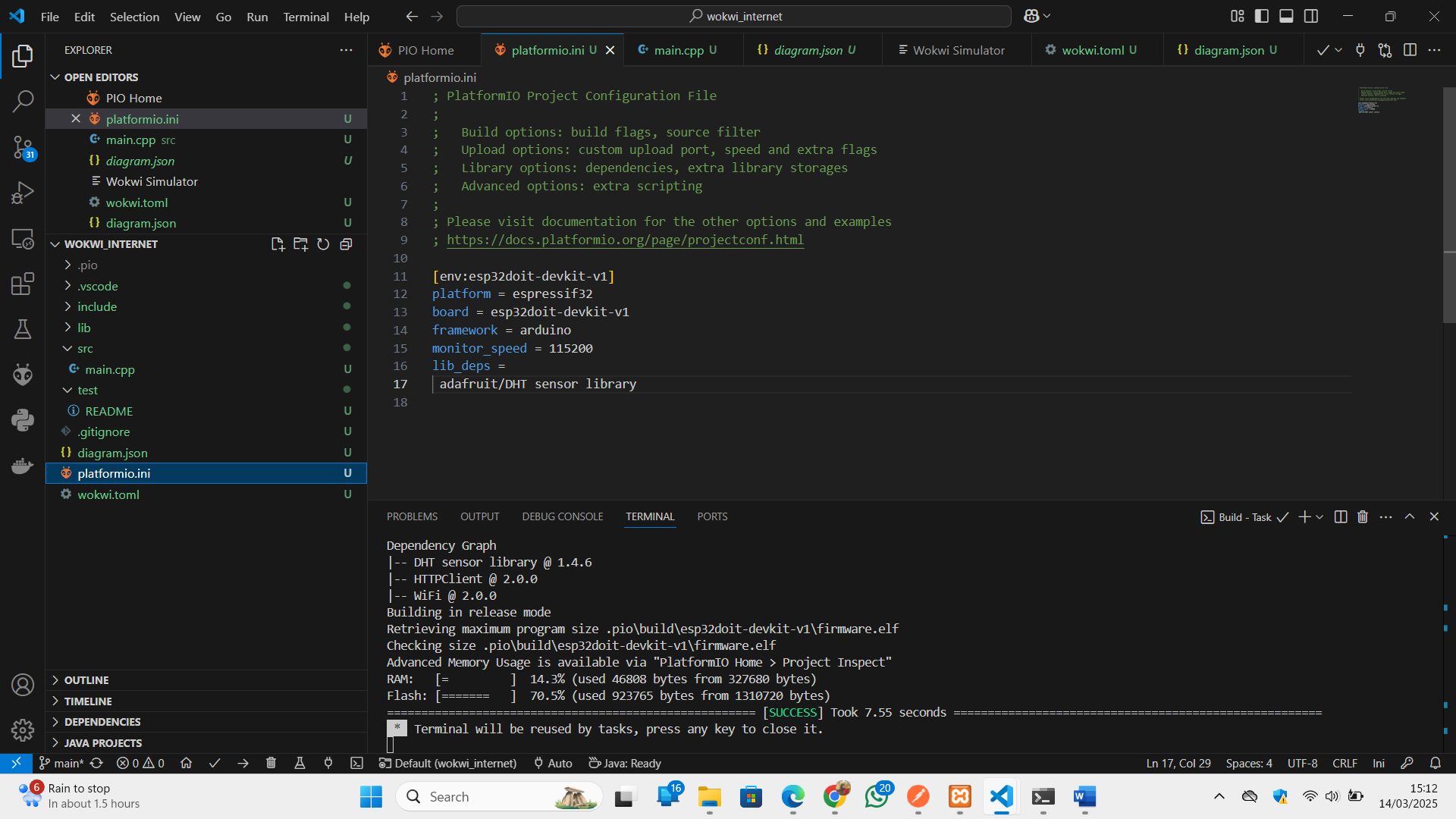
1. **Menjalankan NGROK**  
   Agar API dapat diakses dari internet, jalankan perintah berikut:



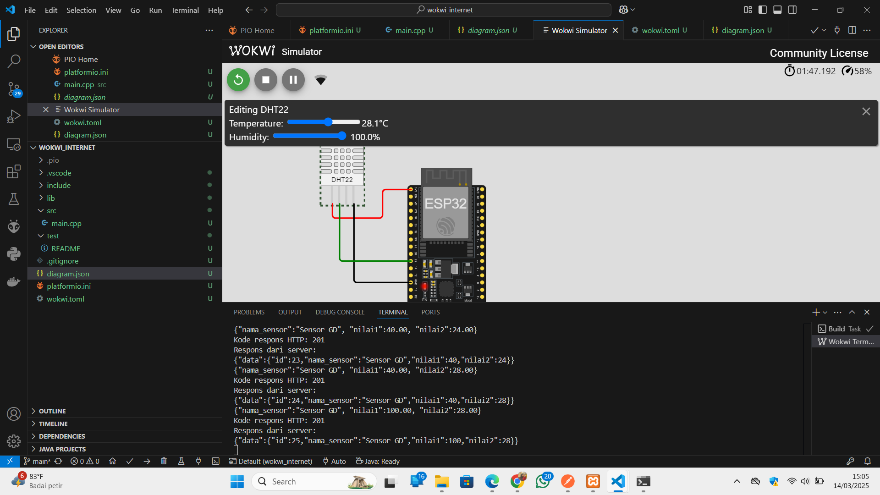
1. **Menambahkan Kode untuk ESP32**  
   Buat file main.cpp di **PlatformIO** dan tambahkan kode berikut:



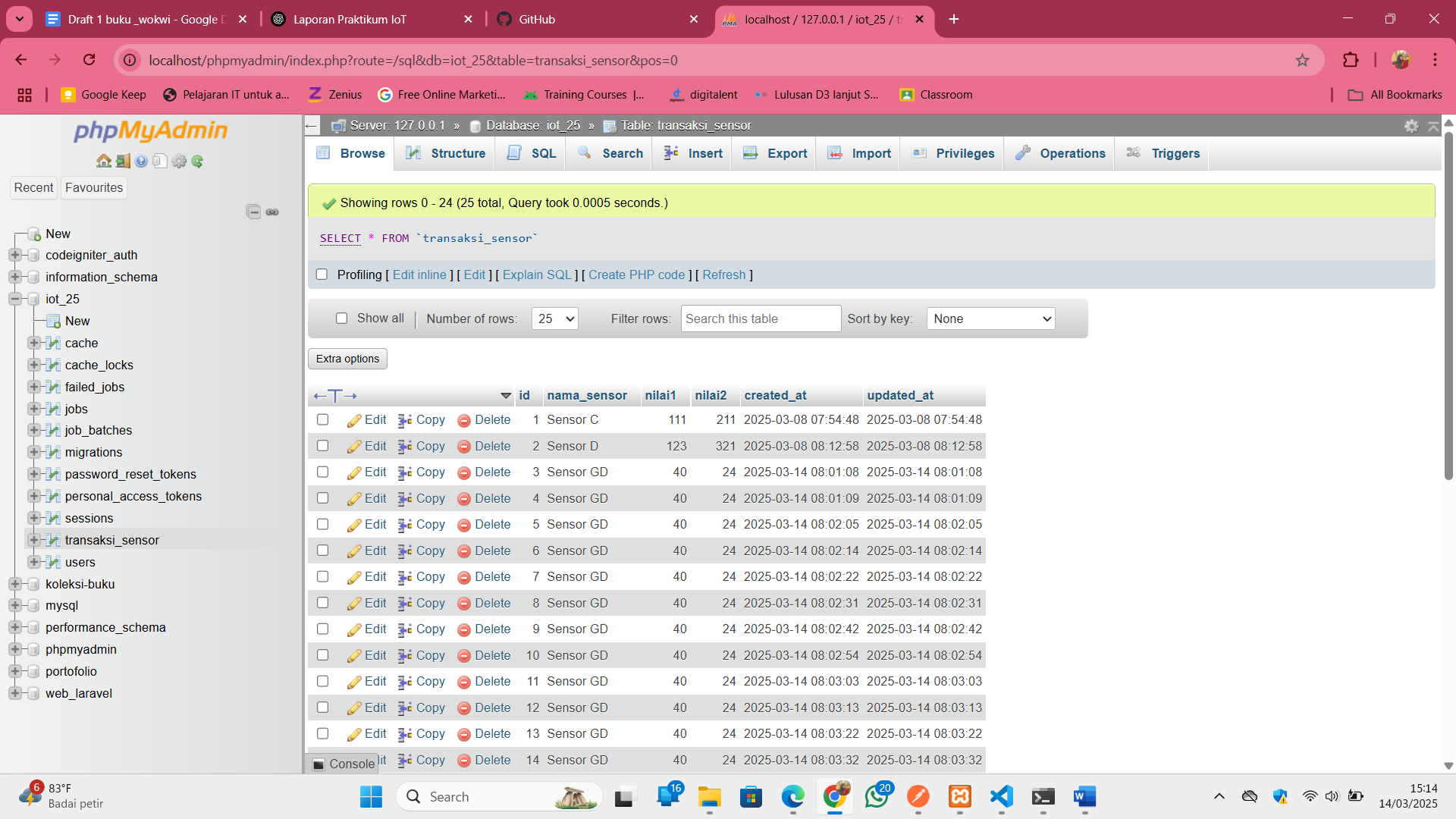
1. **Menambahkan Kode untuk ESP32**  
   Menambahkan kode di platfrom.ini :



1. **Menjalankan Simulasi di WOKWI**  
   Jalankan perintah berikut untuk memulai simulasi:



1. **Memeriksa Database**  
   Pastikan bahwa data suhu dan kelembaban yang dikirim dari ESP32 telah tersimpan di dalam database.



**HASIL DAN PEMBAHASAN**

**3.1 Hasil Eksperimen**

Berdasarkan hasil simulasi, ESP32 berhasil:

* Terhubung ke jaringan Wi-Fi WOKWI-GUEST
* Mengakses API Laravel melalui NGROK
* Mengirimkan data sensor suhu dan kelembaban ke server
* Menampilkan status HTTP 200 (OK) yang menunjukkan bahwa permintaan berhasil diproses

**3.2 Analisis**

Keberhasilan komunikasi antara ESP32 dan API Laravel menunjukkan bahwa perangkat IoT dapat dengan mudah dikembangkan menggunakan teknologi berbasis cloud. Dengan adanya simulasi WOKWI, pengujian dapat dilakukan tanpa memerlukan perangkat keras fisik, sehingga lebih efisien dalam pengembangan sistem IoT.

**3.3 Kesimpulan**

Berdasarkan praktikum yang telah dilakukan, dapat disimpulkan bahwa:

1. ESP32 dapat terhubung ke Wi-Fi dan mengakses API melalui NGROK.
2. Simulasi di WOKWI memungkinkan pengujian sistem IoT tanpa perangkat fisik.
3. Data sensor suhu dan kelembaban berhasil dikirim dan disimpan di database.
4. Implementasi API berbasis Laravel dan ESP32 dapat digunakan dalam proyek IoT berbasis cloud.

**LAMPIRAN**

**4.1 Kode Program ESP32 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://6627-175-45-190-2.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();

  }

}

**4.2 Diagram Skematik dalam Wokwi**

{

    "version": 1,

    "author": "KAVITH BUDVIN",

    "editor": "wokwi",

    "parts": [

      { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 86.4, "left": 24.04, "attrs": {} },

      { "type": "wokwi-dht22", "id": "dht1", "top": 19.5, "left": -91.8, "attrs": {} }

    ],

    "connections": [

      [ "esp:TX", "$serialMonitor:RX", "", [] ],

      [ "esp:RX", "$serialMonitor:TX", "", [] ],

      [ "dht1:VCC", "esp:3V3", "red", [ "v19.2", "h67.2", "v-67.2" ] ],

      [ "esp:GND.1", "dht1:GND", "black", [ "h0" ] ],

      [ "dht1:SDA", "esp:27", "green", [ "v0" ] ]

    ],

    "dependencies": {}

  }

**4.3 ESP32 berhasil terhubung ke WIFI Wokwi-GUEST dan berhasil mengakses API laravel yang sudah dibuat pada bab sebelumnya.**

