**LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)**

Akses API Melalui Simulasi WOKWI

*Abdur Rohim Syah Sjadja‘ah*

*Fakultas Vokasi, Universitas Brawijaya*

*Email:* [abdurrohimsyah05@gmail.com](mailto:abdurrohimsyah05@gmail.com)

**Abstrak**

Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem API yang terintegrasi dengan simulasi IoT menggunakan platform Wokwi untuk mengelola data sensor suhu dan kelembaban. RESTful API dikembangkan dengan framework Laravel 11 sebagai backend untuk menerima, memproses, dan menyimpan data ke basis data MySQL. Simulasi IoT pada Wokwi melibatkan pembuatan prototipe virtual dengan sensor DHT22 yang menghasilkan data suhu dan kelembaban secara real-time. Data tersebut dikirim ke API melalui permintaan HTTP (POST/GET) menggunakan protokol MQTT atau HTTP langsung dari mikrokontroler virtual yaitu ESP32 di Wokwi.

Hasil pengujian menunjukkan bahwa API berhasil menerima data sensor dari simulasi Wokwi dengan struktur JSON dan menyimpannya ke tabel transaksi\_sensor di MySQL. Endpoint API terproteksi dengan middleware Laravel untuk keamanan, sementara validasi data memastikan integritas sebelum penyimpanan. Simulasi Wokwi juga memungkinkan pengujian skenario lingkungan berbeda, seperti perubahan suhu ekstrem atau kelembaban tinggi, untuk memverifikasi respons sistem. Integrasi ini membuktikan bahwa kombinasi Laravel dan Wokwi efektif untuk pengembangan sistem IoT berbasis API tanpa perlu perangkat fisik, sekaligus menyediakan fondasi untuk pengembangan aplikasi pemantauan lingkungan yang skalabel.

Kata Kunci: RESTful API, Laravel, IoT, Wokwi, MySQL, Sensor Suhu dan Kelembaban.

1. **Introduction (Pendahuluan)**

**1.1 Latar Belakang**

Pada perkembangan teknologi saat ini, Internet of Things (IoT) telah menjadi salah satu inovasi terpenting yang mengubah cara kita berinteraksi dengan lingkungan sekitar. Salah satu aplikasi utama IoT adalah pengukuran suhu dan kelembaban, yang memiliki relevansi signifikan dalam berbagai bidang, seperti pertanian, industri, dan pemantauan lingkungan. Sensor suhu dan kelembaban, seperti DHT22, memungkinkan pengumpulan data yang akurat dan real-time, memberikan informasi penting untuk pengelolaan sumber daya dan pengambilan keputusan yang lebih baik.

Pengembangan sistem yang efisien untuk mengakses dan mengelola data sensor menjadi sangat penting. Dalam konteks ini, penggunaan RESTful API sebagai jembatan komunikasi antara perangkat IoT dan aplikasi backend menawarkan solusi yang efektif. Laravel 11, sebagai framework PHP yang kuat, menyediakan alat dan fitur untuk membangun API yang handal dan scalable. Dengan memanfaatkan Laravel, pengembang dapat dengan mudah membuat endpoint untuk mengelola data dari sensor DHT22 yang terhubung dengan mikrokontroler ESP32.

Simulasi perangkat IoT menggunakan platform Wokwi memberikan kemudahan dalam merancang dan menguji prototipe tanpa perlu perangkat keras fisik. Wokwi memungkinkan pengguna untuk membuat rangkaian elektronik virtual, menghubungkan sensor dengan mikrokontroler, serta memprogramnya untuk mengumpulkan dan mengirimkan data. Dengan menggunakan Wokwi, pengembang dapat melakukan simulasi berbagai skenario lingkungan untuk memverifikasi respons sistem terhadap perubahan suhu dan kelembaban.

Melalui penelitian ini, diharapkan dapat dihasilkan sebuah sistem API yang terintegrasi dengan simulasi Wokwi untuk melakukan pengiriman dan penyimpanan data sensor suhu dan kelembaban ke dalam basis data MySQL. Pendekatan ini tidak hanya meningkatkan efisiensi dalam pengumpulan data tetapi juga memberikan fondasi bagi pengembangan aplikasi pemantauan lingkungan yang lebih canggih. Dengan demikian, penelitian ini bertujuan untuk menghadirkan solusi yang sistematis dalam pengelolaan data sensor melalui teknologi IoT dan API, serta memberikan panduan praktis bagi para pengembang dalam menciptakan sistem yang efektif dan terintegrasi.

**1.2 Tujuan Eksperimen**

Eksperimen ini bertujuan untuk menguji integrasi antara simulasi IoT berbasis Wokwi dengan RESTful API yang dikembangkan menggunakan Laravel 11, dengan skenario utama pengiriman data suhu dan kelembaban dari sensor DHT22 virtual yang terhubung ke mikrokontroler ESP32 di Wokwi ke basis data MySQL. Data yang dihasilkan oleh sensor virtual akan dikirimkan ke API melalui permintaan HTTP POST/GET, kemudian disimpan ke tabel transaksi\_sensor yang telah dirancang sebelumnya. Eksperimen ini juga bertujuan untuk memvalidasi mekanisme penerimaan data oleh API, termasuk struktur JSON yang digunakan, validasi input (seperti rentang nilai suhu -40°C hingga 80°C dan kelembaban 0-100%), serta keamanan endpoint menggunakan autentikasi API key. Selain itu, eksperimen ini akan mengevaluasi respons sistem dalam berbagai skenario simulasi, seperti perubahan suhu ekstrem atau kelembaban tinggi, untuk memastikan keandalan penyimpanan data ke MySQL. Hasil akhir yang diharapkan adalah terbentuknya alur komunikasi yang stabil antara Wokwi dan API, sehingga membuktikan bahwa kombinasi simulasi IoT dan pengembangan API dapat menjadi alternatif efisien untuk pengujian sistem IoT tanpa perangkat fisik, sekaligus menyediakan dasar bagi pengembangan aplikasi pemantauan lingkungan berbasis data real-time.

1. **Methodology (Metodologi)**
   1. **Tools & Materials**
2. Laptop
3. Internet
4. Web Browser (Chrome)
5. Postman
6. Visual Studio Code
7. CMD/Windows PowerShell
8. XAMPP
9. Laravel 11 & Ngrok
   1. **Implementation Steps (Langkah Implementasi)**

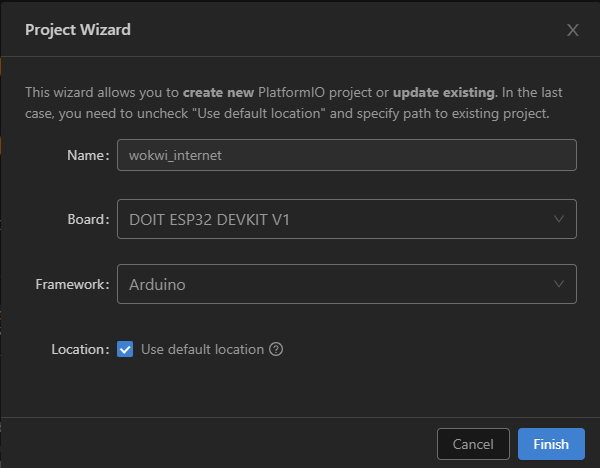
* Buka XAMPP lalu nyalakan Apache dan MySQL
* Buka VSCode dan jalankan server di terminal dengan ketik:

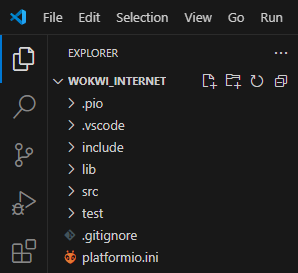
php artisan serve

* Jalankan API Laravel dengan perintah:

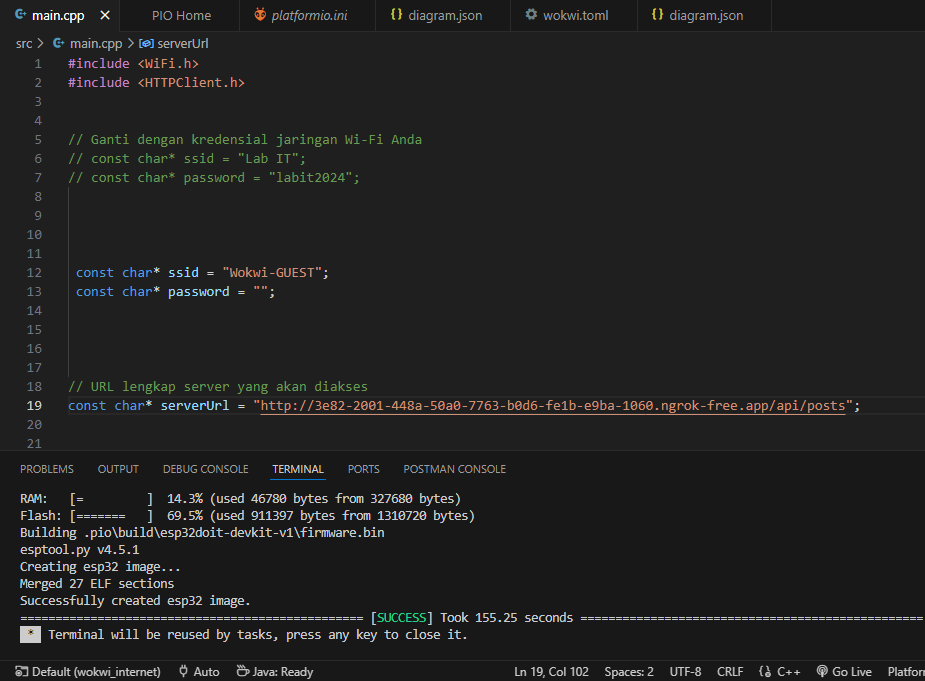
**php artisan serve --host=0.0.0.0 --port=8080**

* Perintah diatas memastikan API laravel dapat diakses dari IP Address manapun dan memastikan bekerja pada port 8080.
* Buat file baru wokwi simulator di platform.io





* Berikut adalah 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:// http://3b0f-2001-448a-50a0-7763-b0d6-fe1b-e9ba-1060.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.");

    }

  }

}

Perhatikan pada bagian

// URL lengkap server yang akan diakses

const char\* serverUrl = "<http://3e82-2001-448a-50a0-7763-b0d6-fe1b-e9ba-1060.ngrok-free.app>/api/posts";

URL diatas adalah URL hasil dari generate perintah NGROK di komputer Anda. Sesuaikan dengan alamat URL yang diberikan oleh NGROK. **Cara menjalankan perintah NGROK berbeda dengan Bab sebelumnya, perintah berikut memastikan NGROK memberikan alamat URL dalam bentuk http bukan https. Pada saat ini ESP32 yang digunakan hanya support http sehingga pastikan NGROK memberikan URL dalam bentuk http bukan https**.

**Steps** tunneling menggunakan Ngrok**:**

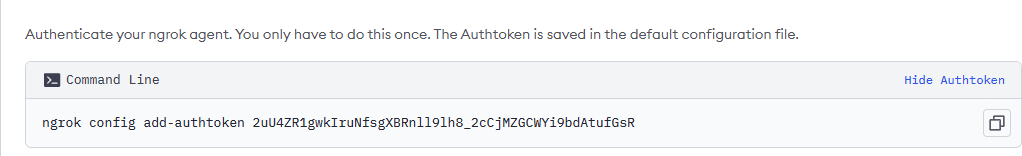
Tunneling dengan Ngrok berarti membuat alamat internet sementara untuk mengakses server di komputer

**Jalankan PowerShell Administrator**

**Masuk folder tempat kita menyimpan atau mengekstrak** **file ngrok.exe**.



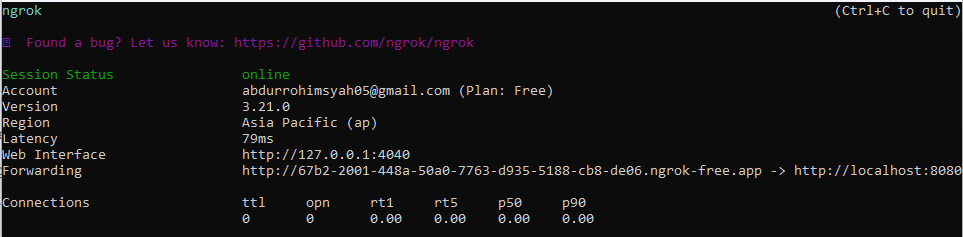
**Akses dan Copy Authtoken kita melalui** <https://dashboard.ngrok.com/get-started/your-authtoken>



**Tambahkan Authtoken ke konfigurasi ngrok di PowerShell:**

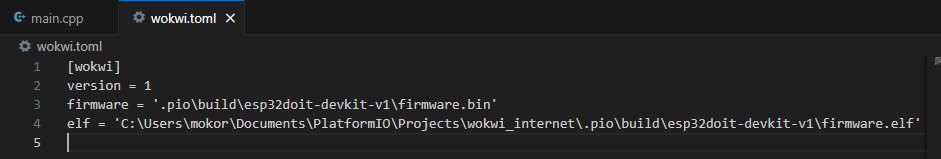


**Jalankan perintah untuk membuka akses internet ke server lokal Laravel di port 8000 dengan ketik:** .\ngrok http --scheme=http 8080



<http://67b2-2001-448a-50a0-7763-d935-5188-cb8-de06.ngrok-free.app>

Tambahkan file **wokwi.toml**



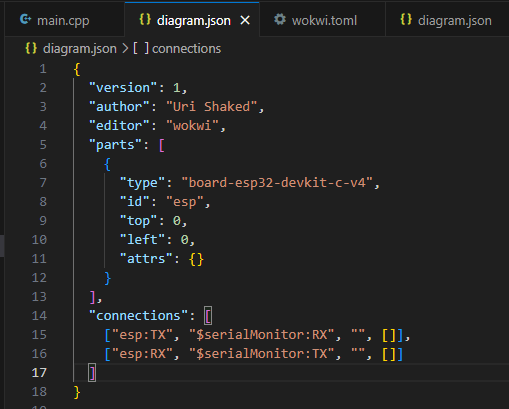
[wokwi]

version = 1

firmware = '.pio\build\esp32doit-devkit-v1\firmware.bin'

elf = 'C:\Users\mokor\Documents\PlatformIO\Projects\wokwi\_internet\.pio\build\esp32doit-devkit-v1\firmware.elf'

Tambahkan file **diagram.json**



{

  "version": 1,

  "author": "Uri Shaked",

  "editor": "wokwi",

  "parts": [

    {

      "type": "board-esp32-devkit-c-v4",

      "id": "esp",

      "top": 0,

      "left": 0,

      "attrs": {}

    }

  ],

  "connections": [

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

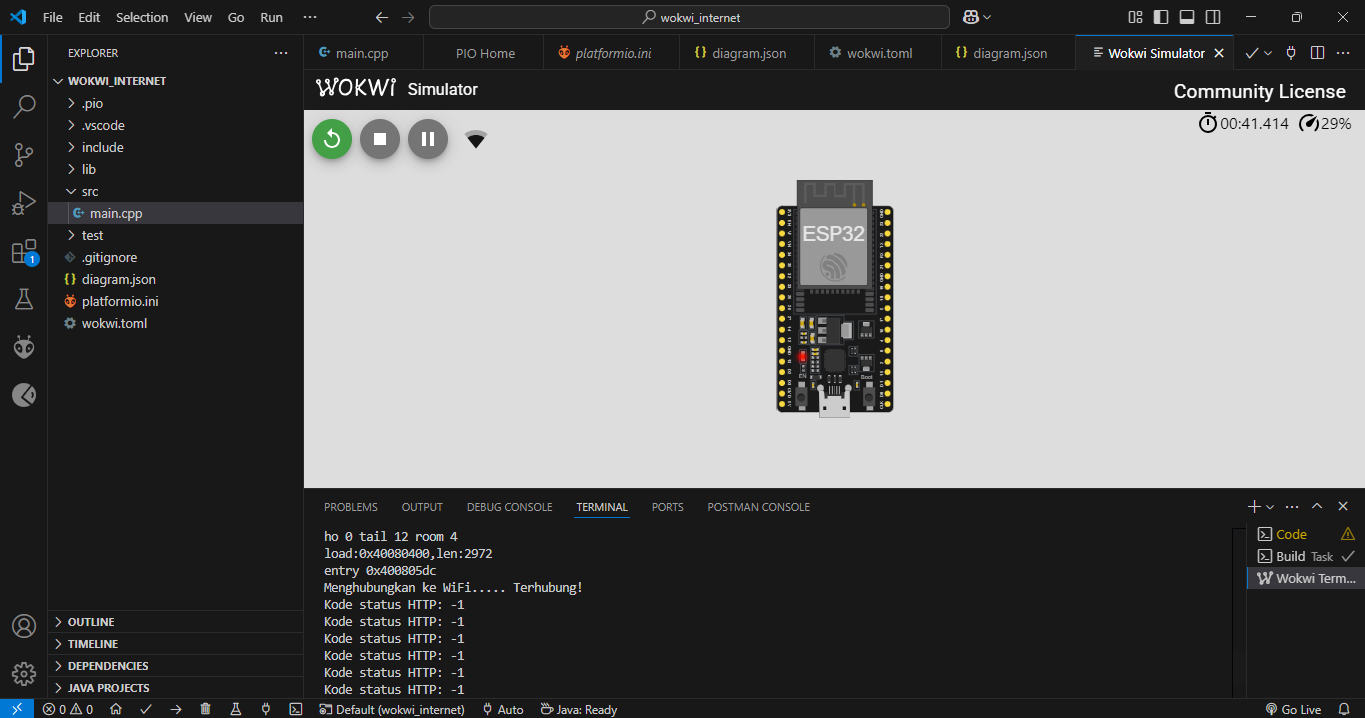
    ["esp:RX", "$serialMonitor:TX", "", []]

  ]

}

Langkah berikutnya adalah melakukan simulasi. Build file main.cpp dan jalankan simulasi dengan perintah

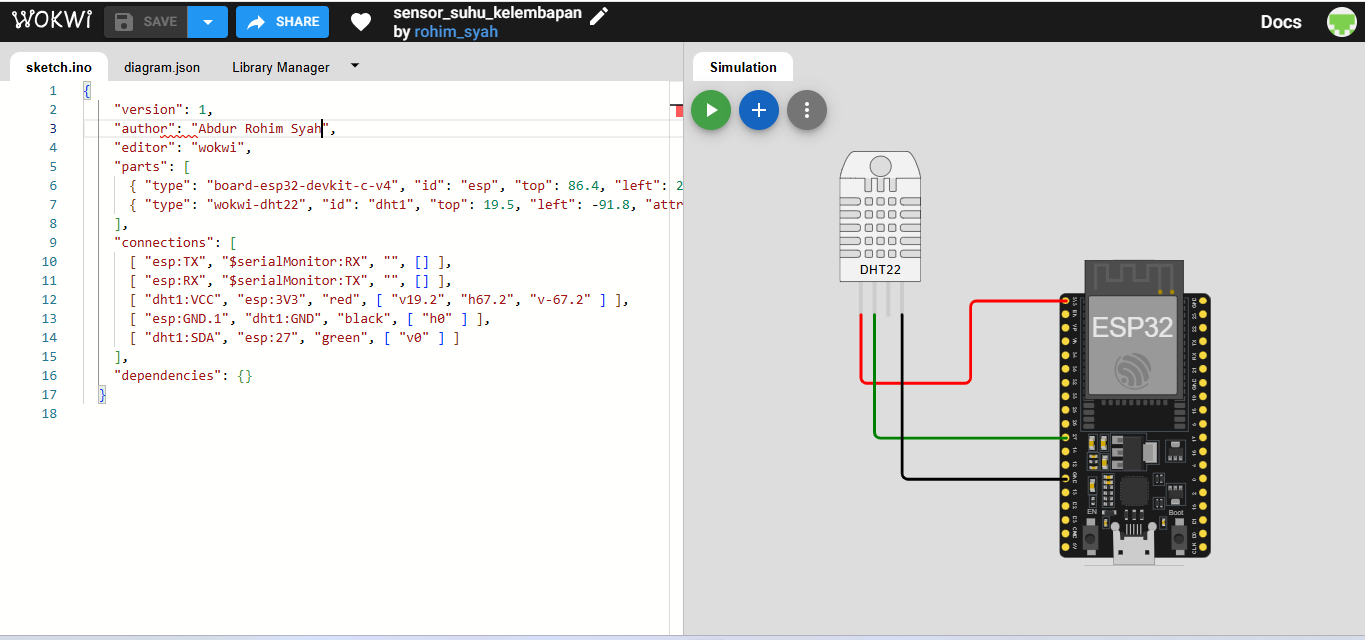
**> Wokwi Start Simulator**



Simulasi diatas menunjukkan, ESP32 berhasil terhubung ke WIFI Wokwi-GUEST dan berhasil mengakses API laravel yang sudah dibuat pada bab sebelumnya.

**Kode Status HTTP:200**

Berikutnya adalah melakukan modifikasi simulasi dengan menambahkan sensor suhu dan kelembaban. Skenarionya adalah, wokwi simulator akan mengirimkan data suhu dan kelembaban ke API dan menyimpannya ke database mysql seperti yang telah dibuat pada bab sebelumnya.\



Rangkai sensor DHT22 dengan ESP32 seperti contoh diatas. Kemudian salin kode **diagram.json** ke file diagram.json yang ada di vscode.

{

    "version": 1,

    "author": "Abdur Rohim Syah",

    "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" ] ],

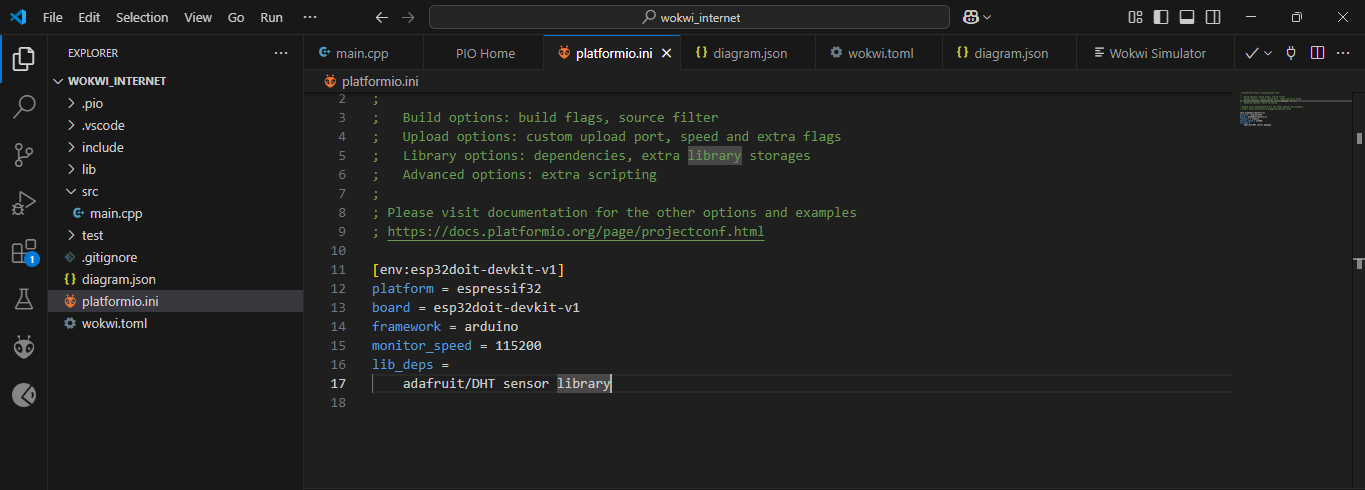
      [ "dht1:SDA", "esp:27", "green", [ "v0" ] ]

    ],

    "dependencies": {}

  }

Kemudian ubah setting file **platformio.ini** sebagai berikut :



Pada perubahan diatas, ada tambahan 2 setting yaitu monitor speed dan lib\_deps

Monitor speed digunakan untuk memonitor status pengiriman data dari wokwi simulator ke server api laravel yang telah dibuat. Sedangkan lib\_deps adalah library yang digunakan sensor DHT (sensor suhu dan kelembaban).

Modifikasi file **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();

  }

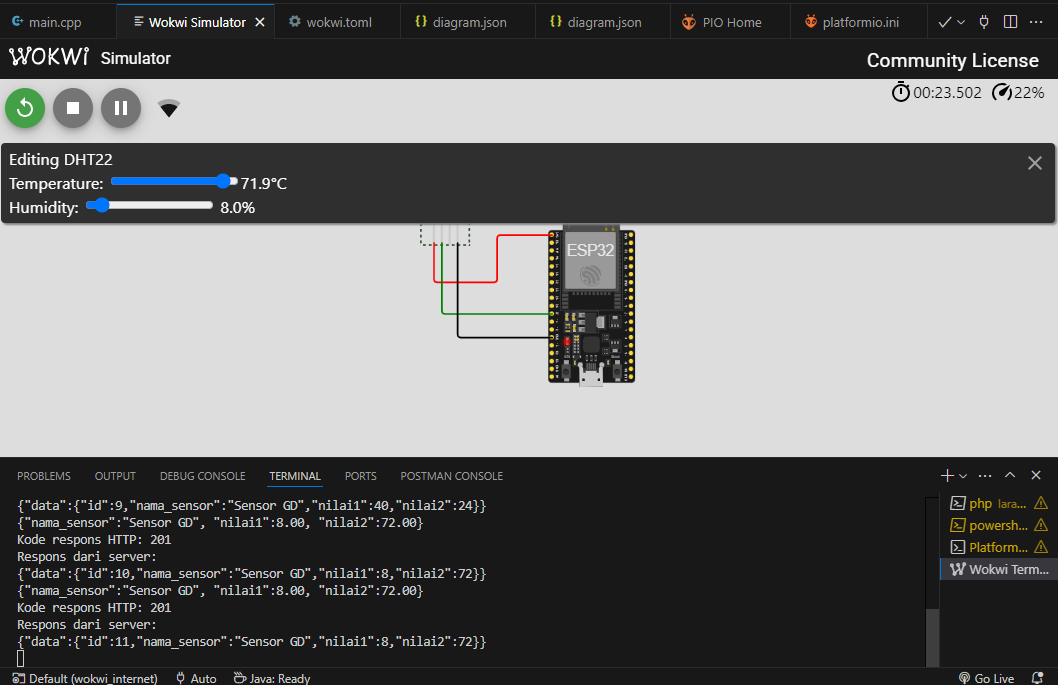
}

Pada bagian berikut sesuaikan dengan URL NGROK anda

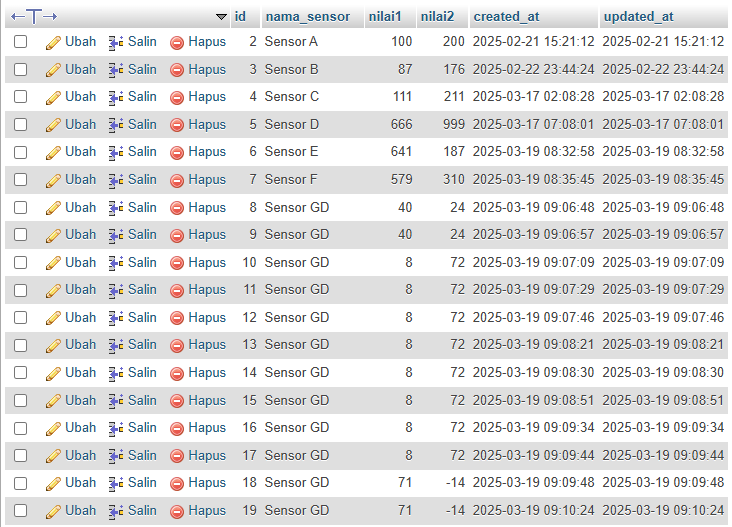
"**http://3b0f-2001-448a-50a0-7763-b0d6-fe1b-e9ba-1060.ngrok-free.app**/api/posts"; // Ganti dengan URL ngrok yang benar

Jalankan simulasi

**> Wokwi Start Simulator**



Hasil masuk ke database phpMyAdmin



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

Eksperimen akses API melalui simulasi Wokwi dengan sensor DHT22 dan mikrokontroler ESP32 menghasilkan beberapa temuan sebagai berikut:

1. **Integrasi Wokwi dengan API Laravel:**
   * Simulasi berhasil dilakukan dengan menggunakan Wokwi untuk menghubungkan sensor DHT22 ke ESP32 virtual.
   * Mikrokontroler ESP32 berhasil mengirimkan data suhu dan kelembaban ke RESTful API Laravel melalui permintaan HTTP POST.
   * Payload JSON yang dikirimkan memiliki struktur yang sesuai, yaitu:

json

{

"nama\_sensor": "Sensor GD",

"nilai1": 65,

"nilai2": 27

}

Data ini berisi nama sensor, nilai kelembaban (nilai1), dan nilai suhu (nilai2).

1. **Penyimpanan Data ke Basis Data MySQL:**
   * Data yang diterima oleh API berhasil disimpan ke tabel transaksi\_sensor di basis data MySQL. Setiap entri mencakup id, nama\_sensor, nilai1 (kelembaban), nilai2 (suhu), serta timestamp waktu penyimpanan.
   * Validasi input pada API memastikan bahwa hanya data dengan format yang benar yang dapat diterima dan disimpan.
2. **Penggunaan Ngrok untuk Tunneling:**
   * Ngrok digunakan untuk membuat URL publik sementara, memungkinkan API Laravel diakses oleh ESP32 di Wokwi. URL yang dihasilkan berbentuk HTTP sesuai kebutuhan perangkat ESP32.
   * URL Ngrok berhasil digunakan untuk mengirimkan data dari simulasi ke server Laravel.
3. **Respons API:**
   * Respons API menunjukkan kode status HTTP 200 atau 201 untuk permintaan POST yang berhasil, dengan pesan konfirmasi dari server.
   * Simulasi menunjukkan bahwa setiap data yang dikirimkan dalam interval 5 detik berhasil diterima oleh API dan disimpan ke basis data.
4. **Monitoring dan Debugging:**
   * Serial monitor pada Wokwi menampilkan log proses pengiriman data, termasuk payload JSON, kode respons HTTP, dan respons dari server.
   * Tidak ditemukan error kritis selama pengujian, kecuali ketika sensor gagal membaca nilai suhu atau kelembaban karena simulasi belum stabil.
   1. **Discussion (Pembahasan)**
5. **Keberhasilan Integrasi IoT dengan API:**

Eksperimen ini membuktikan bahwa integrasi antara simulasi IoT menggunakan Wokwi dan RESTful API berbasis Laravel dapat berjalan dengan baik. Sensor DHT22 virtual pada Wokwi mampu mengirimkan data real-time ke server Laravel melalui protokol HTTP POST. Hal ini menunjukkan bahwa kombinasi teknologi ini dapat menjadi solusi efektif untuk pengembangan sistem IoT tanpa memerlukan perangkat keras fisik.

1. **Keandalan Pengiriman Data:**

Penggunaan HTTPClient pada ESP32 memungkinkan pengiriman data secara efisien dalam interval waktu tertentu (5 detik). Payload JSON yang dikirimkan sesuai dengan format yang diharapkan oleh API, sehingga meminimalkan risiko error dalam proses parsing data di server.

1. **Validasi dan Keamanan:**

Middleware Laravel digunakan untuk melindungi endpoint API dari akses tidak sah. Selain itu, validasi input memastikan bahwa hanya data dengan format yang benar yang dapat diterima oleh server, mencegah potensi serangan injeksi atau kesalahan penyimpanan data.

1. **Manfaat Penggunaan Ngrok:**

Ngrok memberikan kemudahan dalam membuka akses publik ke server lokal Laravel tanpa perlu konfigurasi jaringan tambahan. Namun, URL Ngrok bersifat sementara sehingga perlu diperbarui setiap kali sesi baru dimulai, yang menjadi salah satu keterbatasan dalam eksperimen ini.

1. **Simulasi Lingkungan Berbeda:**

Dengan menggunakan Wokwi, eksperimen dapat mensimulasikan berbagai skenario lingkungan seperti perubahan suhu ekstrem atau kelembaban tinggi. Hal ini memungkinkan pengembang untuk menguji respons sistem terhadap kondisi lingkungan yang dinamis tanpa memerlukan perangkat fisik.

1. **Kendala Teknis:**

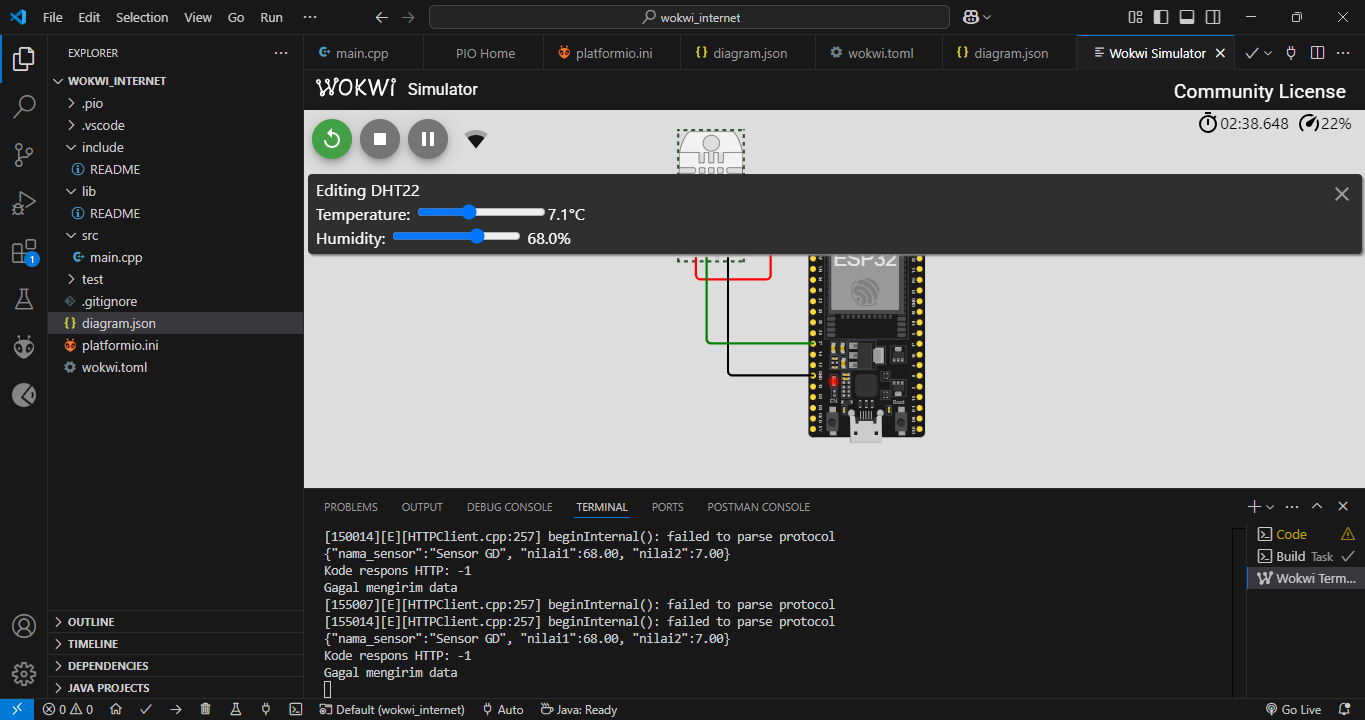
Beberapa kendala teknis ditemukan selama eksperimen:

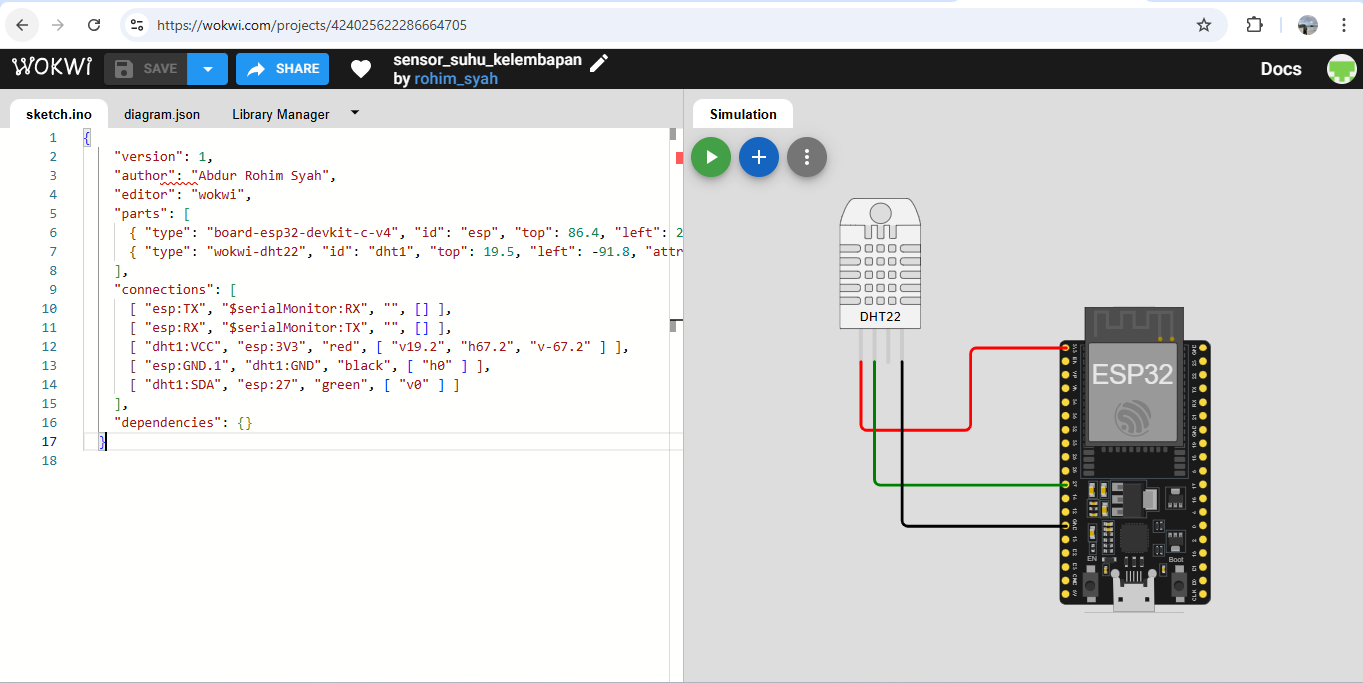
* Ketergantungan pada koneksi internet untuk menjalankan simulasi Wokwi dan Ngrok.
* Sensor DHT22 terkadang gagal membaca nilai suhu atau kelembaban jika simulasi belum stabil.

1. **Potensi Pengembangan Lebih Lanjut**

* Menambahkan autentikasi berbasis token untuk meningkatkan keamanan API.
* Mengintegrasikan sistem ini dengan aplikasi frontend untuk visualusasi data secara real-time.
* Menggunakan layanan hosting permanen seperti AWS atau Heroku agar API dapat diakses tanpa Batasan waktu seperti pada ngrok.

1. **Appendix**



****