LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya



**Praktik Akses API Melalui Simulasi WOKWI**

*Andrian Alfini*

Fakultas Vokasi, Universitas Brawijaya

Email : andrianalfian2003@gmail.com

**Abstrak**

Wokwii adalah platform simulasi berbasis web yang memungkinkan pengujian mikrokontroler seperti ESP32. Sementara itu, Ngrok adalah alat terowongan yang memungkinkan akses ke server lokal diakses melalui internet. Metode yang digunakan termasuk Fire menggunakan kerangka kerja Laravel, konfigurasi NGROK untuk membuka akses publik ke server lokal, dan mengimplementasikan kode Wokwi untuk mengakses API. Hasil eksperimen menunjukkan bahwa Ngrok dapat dikombinasikan. Studi ini memberi pengembang wawasan tentang pengujian sistem IoT dengan cara yang lebih fleksibel dan lebih murah.

*Kata Kunci****:*** *ESP32,NGROK, Wokwi, PlatformIO, vscode.*

**1.Pendahuluan**

* 1. **Latar belakang**

**simulator IoT** seperti **WOKWI** menjadi solusi yang efektif untuk menguji sistem secara virtual sebelum diimplementasikan pada perangkat sebenarnya. Selain simulasi perangkat, sistem IoT juga membutuhkan layanan **Application Programming Interface (API)** untuk mengelola komunikasi data antara perangkat dan server. API memungkinkan perangkat IoT mengirim dan menerima data ke backend, seperti database atau aplikasi berbasis web. Dalam lingkungan pengembangan lokal, API biasanya berjalan di **localhost**, sehingga tidak dapat diakses oleh perangkat lain di luar jaringan lokal. Untuk mengatasi keterbatasan ini, **NGROK** digunakan sebagai alat **tunneling** yang memungkinkan akses ke server lokal melalui internet dengan menyediakan URL publik yang dapat digunakan oleh WOKWI atau perangkat IoT lainnya.

**1.2 Tujuan eksperimen**

1. Menganalisis bagaimana WOKWI dapat digunakan untuk mensimulasikan perangkat IoT dalam mengakses API secara virtual.Menggunakan PlatformIO di VS Code untuk pemrograman ESP32.
2. Menguji konektivitas antara API lokal yang dikembangkan menggunakan Laravel dengan simulasi perangkat di WOKWI melalui NGROK sebagai tunneling service.
3. Mengevaluasi kinerja NGROK dalam menyediakan akses jarak jauh ke API lokal, termasuk aspek kecepatan, kestabilan koneksi, dan keamanannya.

**2. Methodology (Metodologi)**

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

* ESP32 (simulasi)
* Wokwi Simulator
* PlatformIO di VS Code dan codingan
* NGROK
* PHP my admin

**2.2 Implementation Steps (Langkah Implementasi)**

1. Membuat database di phpMyAdmin dengan nama iot\_25.
2. Menginstal dan mengonfigurasi Laravel 11 menggunakan VS Code.
3. Membuat database di phpMyAdmin dengan nama iot\_25.
4. Membuat model TransaksiSensor dan melakukan migrasi database di Laravel.
5. Membuat API controller TransaksiSensorController untuk mengelola transaksi data sensor.
6. Mengonfigurasi route API di Laravel.
7. Menjalankan server Laravel dengan php artisan serve --host=0.0.0.0 --port=8000 di VS Code Terminal.
8. Mengonline-kan API menggunakan Ngrok dengan perintah ngrok http --scheme=http 8080.
9. Mengakses API melalui Wokwi dengan mengirimkan dan menerima data sensor.

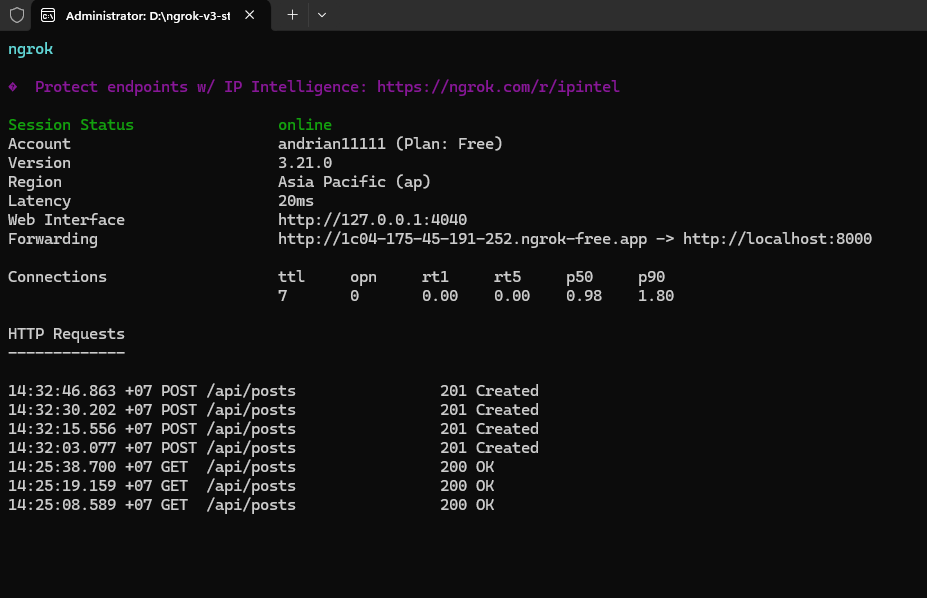
**3. Results and Discussion (Hasil dan Pembahasan)**

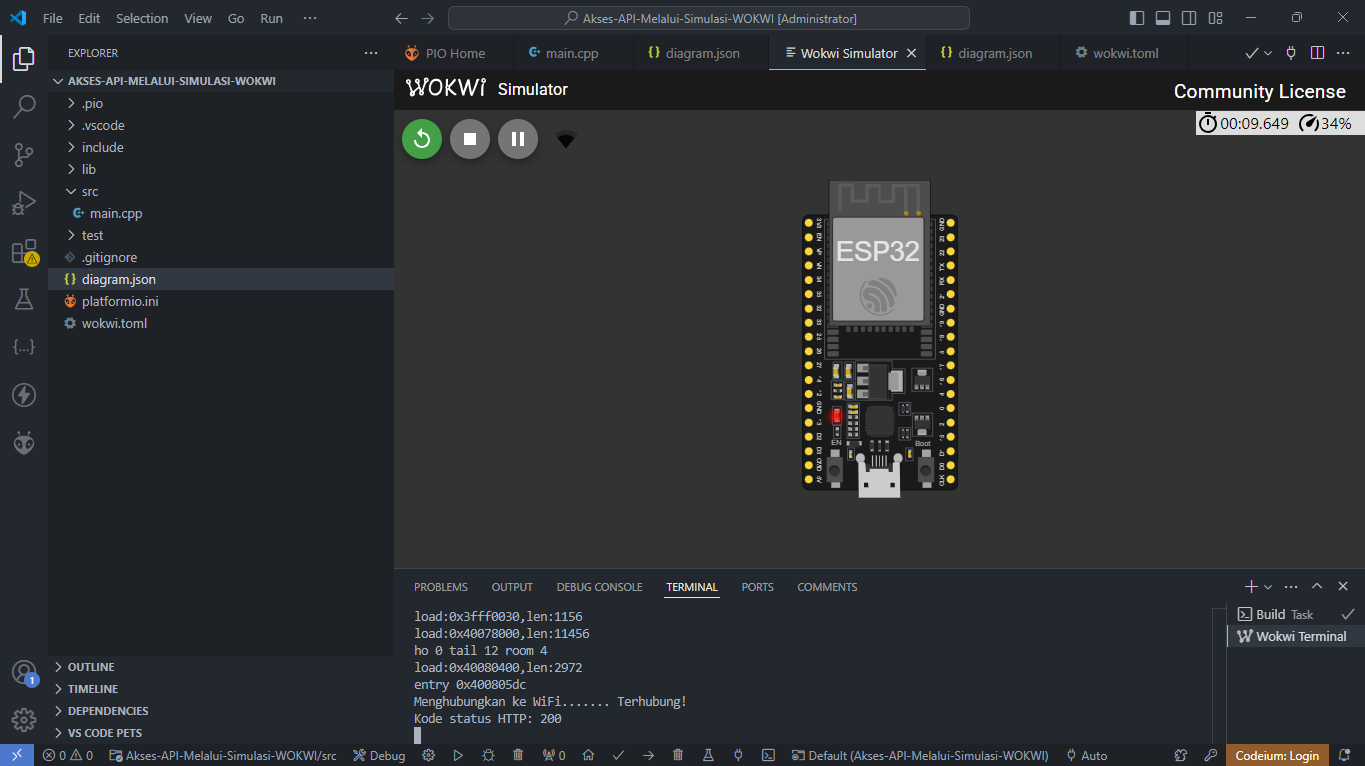
**3.1 Experimental Results**

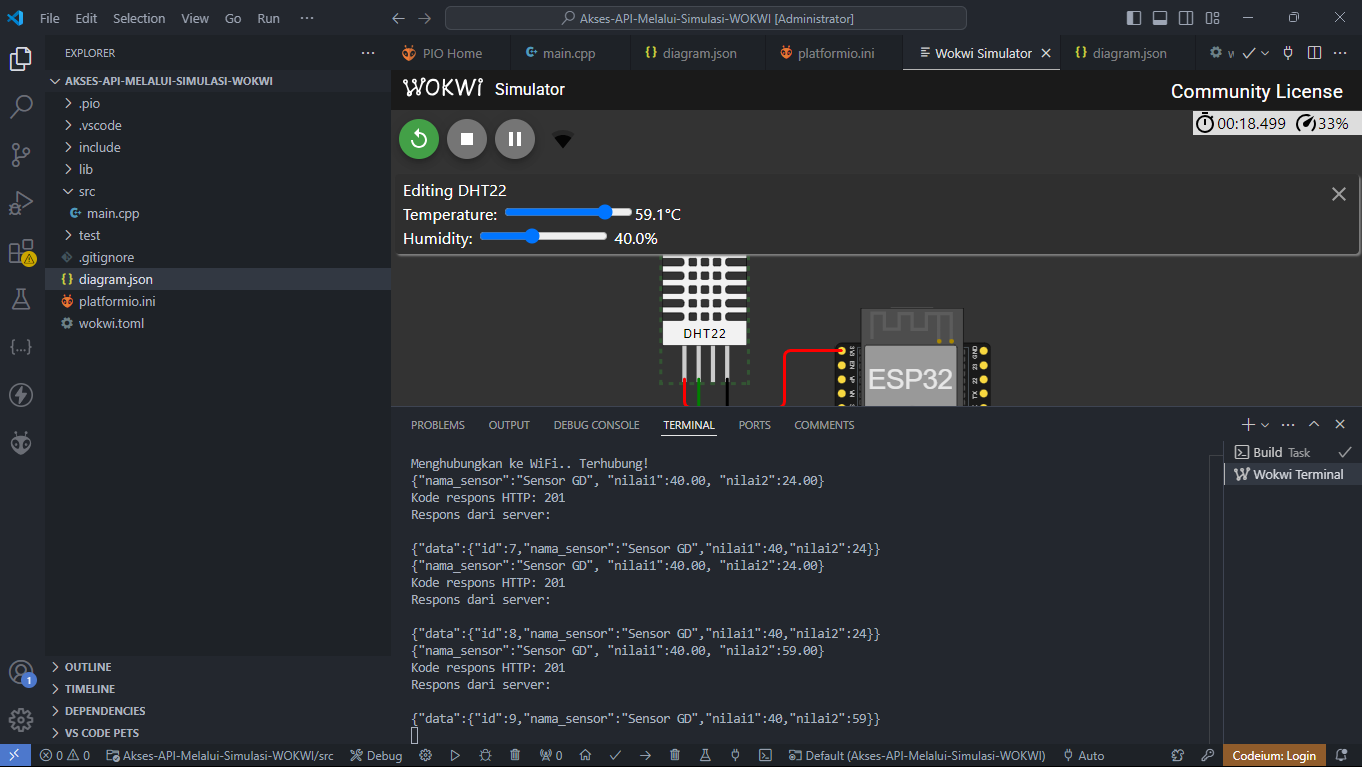
Dari hasil pengujian, API yang dibuat menggunakan Laravel dapat diakses dengan baik oleh simulator Wokwi:

* 1. Data sensor dapat dikirimkan ke API dan tersimpan di database.
  2. Data dapat diambil dari API dan ditampilkan pada Serial Monitor di Wokwi.
  3. API Laravel berjalan dengan baik saat diakses menggunakan Ngrok.

**4. Appendix**





  
  
CODE :

#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://1c04-175-45-191-252.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();

  }

}

