**LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)** Fakultas Vokasi, Universitas Brawijaya

**Praktik Akses API Melalui Simulasi WOKWI**



*Cantika Kelana*  
 Fakultas Vokasi, Universitas Brawijaya  
 Email: rahmadntr\_@student.ub.ac.id

## **Abstract (Abstrak)**

Praktikum ini bertujuan untuk menguji komunikasi antara ESP32 dan API Laravel menggunakan simulasi Wokwi. Dengan memanfaatkan Ngrok, ESP32 dapat terhubung ke server Laravel dan mengirimkan data sensor ke database. Sensor DHT22 digunakan untuk mengukur suhu dan kelembaban, kemudian hasil pengukuran dikirim melalui API ke sistem penyimpanan. Hasil pengujian menunjukkan bahwa data berhasil dikirim dan disimpan dengan baik dalam database, membuktikan bahwa sistem ini dapat digunakan sebagai solusi IoT berbasis API.

*Keywords: Laravel, API, Wokwi, ESP32, Ngrok, IoT*

## **1. Introduction (Pendahuluan)**

### **1.1 Latar Belakang**

Dalam sistem IoT, API memainkan peran penting dalam menghubungkan perangkat dengan server untuk bertukar data. Laravel 11 digunakan sebagai backend untuk mengelola komunikasi antara ESP32 dan server melalui internet. Wokwi berfungsi sebagai platform simulasi ESP32, sementara Ngrok digunakan untuk menyediakan URL publik agar API dapat diakses di luar jaringan lokal.

### **1.2 Tujuan**

1. Menghubungkan ESP32 ke API Laravel melalui simulasi Wokwi.
2. Mengirimkan data suhu dan kelembaban ke database melalui API.
3. Menggunakan Ngrok untuk menyediakan akses API secara online.

## **2. Methodology (Metodologi)**

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

* **Hardware :** ESP32 (simulasi di Wokwi)
* **Sensor :** DHT22
* **Software :** Laravel 11, Ngrok, Postman, Wokwi, PlatformIO

### **2.2 Implementation Steps (Langkah Implementasi)**

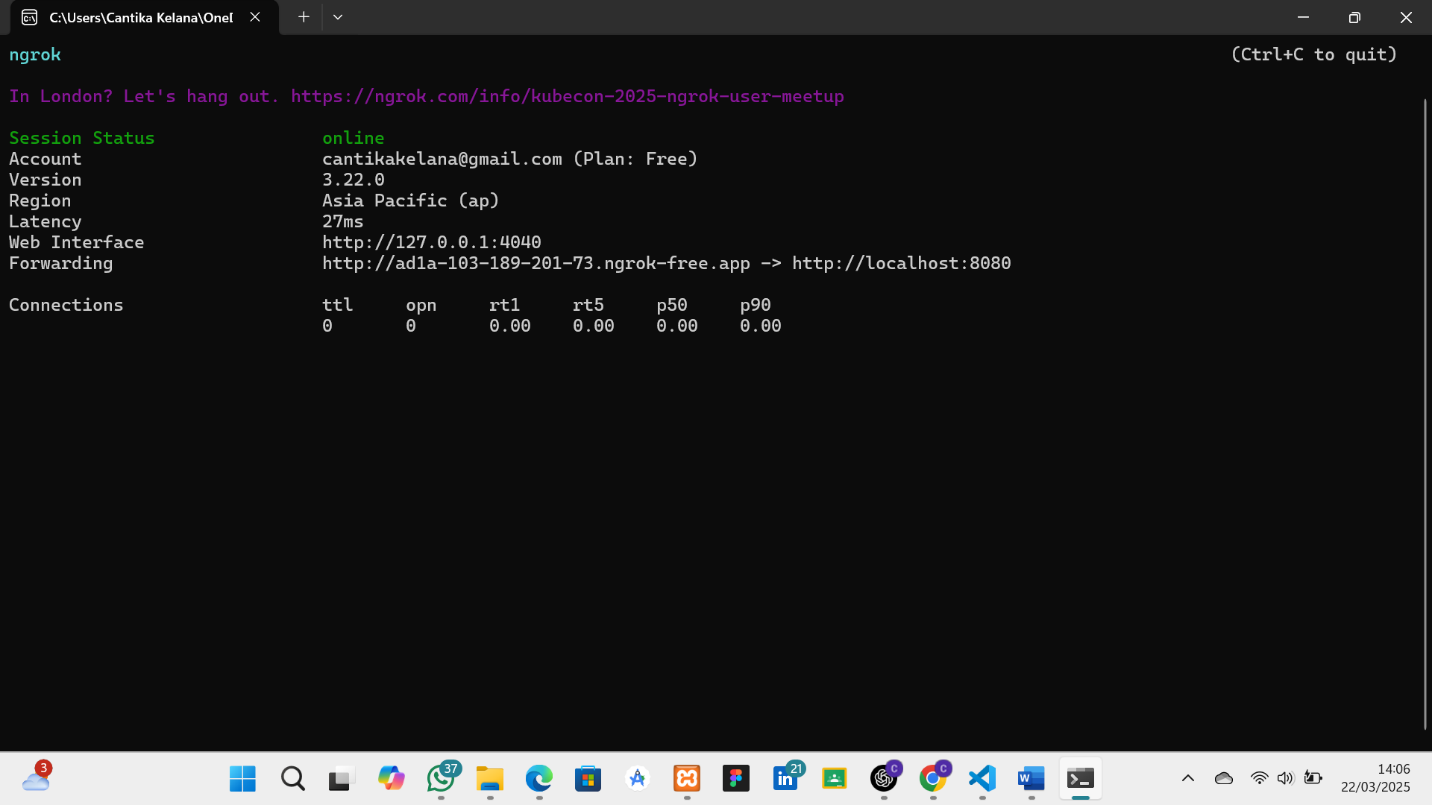
* **Menjalankan API Laravel**
  + Jalankan server Laravel dengan perintah:
  + php artisan serve --host=0.0.0.0 --port=8080
  + Perintah ini memastikan API dapat diakses dari IP mana pun.
* **Menghubungkan API Laravel ke Internet dengan Ngrok**
  + Jalankan perintah berikut untuk mendapatkan URL publik dari Ngrok:
  + ngrok http --scheme=http 8080
  + Catat URL hasil dari Ngrok dan gunakan dalam kode ESP32.
* **Membuat File Konfigurasi Wokwi**
  + Buat file wokwi.toml dan tambahkan:
  + [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'
  + Buat file diagram.json dengan konfigurasi ESP32 dan sensor DHT22.
* **Menulis Kode Program untuk ESP32**
  + Buat file main.cpp dan tambahkan kode berikut:
  + #include <Arduino.h>
  + #include <WiFi.h>
  + #include <HTTPClient.h>
  + #include "DHT.h"
  + #define DHTPIN 27
  + #define DHTTYPE DHT22
  + DHT dht(DHTPIN, DHTTYPE);
  + const char\* ssid = "Wokwi-GUEST";
  + const char\* password = "";
  + const char\* serverUrl = "http://YOUR\_NGROK\_URL/api/posts";
  + void setup() {
    - Serial.begin(115200);
    - WiFi.begin(ssid, password);
    - while (WiFi.status() != WL\_CONNECTED) {
    - delay(500);
    - Serial.print(".");
    - }
    - Serial.println(" Terhubung!");
    - dht.begin();
  + }
  + void loop() {
    - float h = dht.readHumidity();
    - float t = dht.readTemperature();
    - if (isnan(h) || isnan(t)) {
    - Serial.println("Sensor gagal membaca data!");
    - return;
    - }
    - HTTPClient http;
    - http.begin(serverUrl);
    - http.addHeader("Content-Type", "application/json");
    - String payload = "{\"nama\_sensor\":\"Sensor GD\", \"nilai1\":" + String(h) + ", \"nilai2\":" + String(t) + "}";
    - int httpResponseCode = http.POST(payload);
    - Serial.print("Kode respons HTTP: ");
    - Serial.println(httpResponseCode);
    - http.end();
    - delay(5000);
  + }
  + Sesuaikan YOUR\_NGROK\_URL dengan URL yang diperoleh dari Ngrok.
* **Menjalankan Simulasi di Wokwi**
  + Jalankan simulasi dengan perintah:
  + Wokwi Start Simulator
  + Periksa serial monitor untuk memastikan data berhasil dikirim.

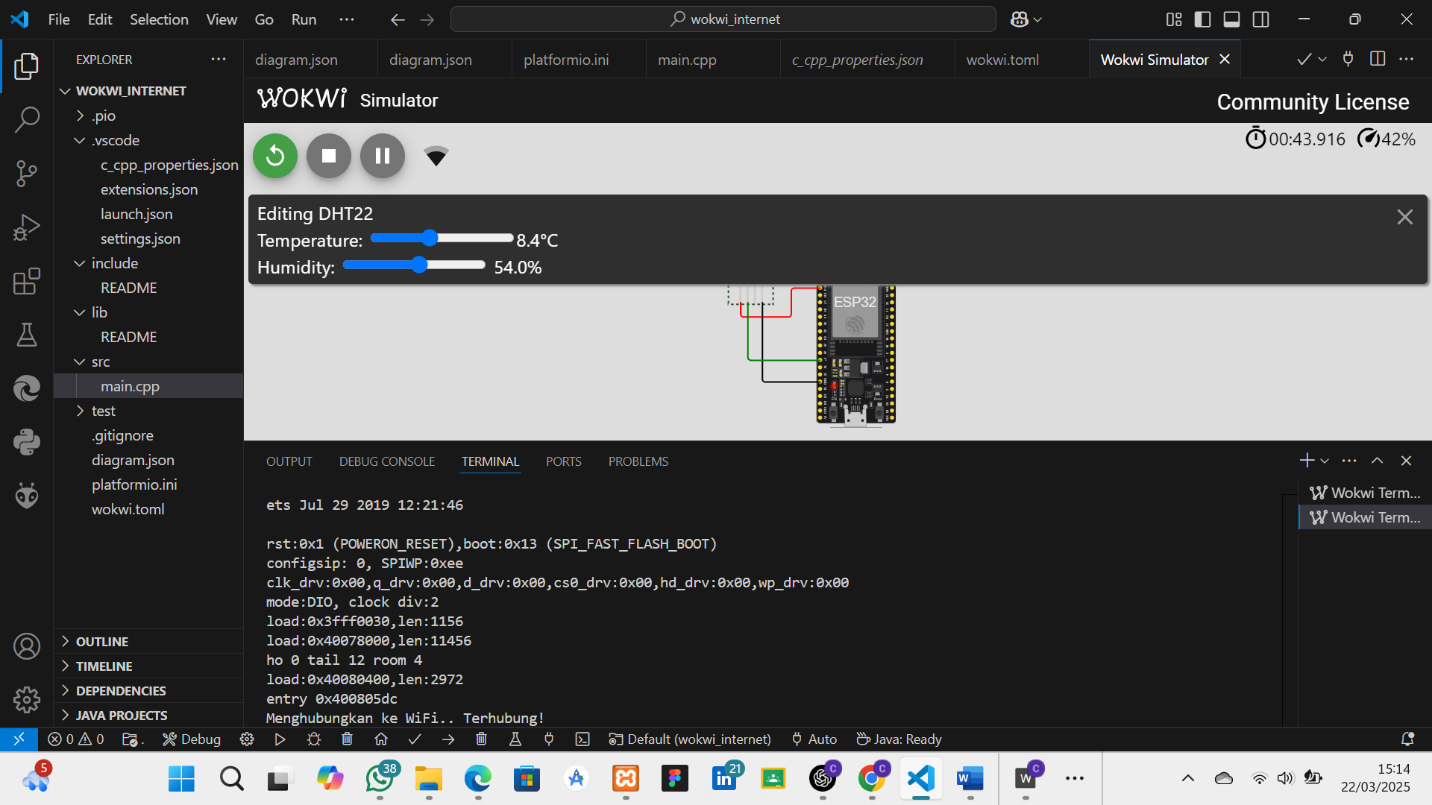
## **3. Results and Discussion (Hasil dan Pembahasan)**

### **3.1 Experimental Results (Hasil Eksperimen)**

* ESP32 berhasil terhubung ke WiFi Wokwi-GUEST.
* Data suhu dan kelembaban berhasil dikirim ke API Laravel.
* Data tersimpan di database dengan format JSON.

**Screenshot hasil simulasi:**





## **4. Appendix (Lampiran)**

### **4.1 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://8f96-125-163-128-149.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();

  }

}