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

**Fakultas Vokasi, Universitas Brawijaya**

**Praktik Akses API Melalui   
Simulasi WOKWI**

*Avrilla Agnesya Meifilistiara*

*Fakultas Vokasi, Universitas Brawijaya*

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

**Abstract**

Penelitian ini mengimplementasikan komunikasi antara perangkat IoT dan server melalui API menggunakan simulasi Wokwi. Eksperimen dilakukan dengan mengintegrasikan ESP32, sensor DHT22, VSCode, dan server Laravel yang diekspos menggunakan Ngrok. Simulasi berhasil mengirimkan data suhu dan kelembaban dari sensor DHT22 ke server API dengan interval waktu tertentu. Hasil menunjukkan bahwa komunikasi berjalan dengan baik, dibuktikan dengan kode respons HTTP 201 yang mengindikasikan data berhasil diterima dan disimpan di database. Pengujian juga memverifikasi bahwa sistem pengelolaan data di server berfungsi dengan baik, dengan setiap pengiriman data menerima ID unik. Kesimpulannya, simulasi menggunakan Wokwi menyediakan metode efektif untuk menguji komunikasi IoT berbasis API tanpa memerlukan perangkat fisik atau infrastruktur server yang kompleks.

*Keywords—Internet of Things, API, ESP32, Wokwi, DHT22, Simulasi, Ngrok*

**1. Introduction**

* 1. **Latar belakang**

Dalam pengembangan sistem IoT, komunikasi antara perangkat dengan server menjadi aspek penting. Salah satu metode yang digunakan adalah dengan memanfaatkan API untuk menghubungkan perangkat IoT dengan server. Dengan integrasi tersebut dapat diketahui bahwa API bisa menghubungkan simulasi melalui VSCode. Oleh karena itu, simulasi menggunakan Wokwi, Postman, dan Ngrok menjadi solusi efektif untuk menguji API tanpa harus memiliki server langsung di internet.

**1.2 Tujuan eksperimen**

1. Mengimplementasikan komunikasi antara ESP32 dengan server API menggunakan simulasi Wokwi
2. Menguji dan memverifikasi komunikasi data sensor dari ESP32 ke server API
3. Menganalisis hasil pengujian melalui terminal dan database

**2. Methodology (Metodologi)**

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

1. Mikrokontroler : ESP32 dan Sensor DHT22
2. Visual Studio Code, Wokwi
3. XAMPP, phpmyAdmin
4. Server API
5. Postman/Ngrok untuk menguji server
   1. **Implementation Steps (Langkah Implementasi)**
6. Menjalankan API dengan mengaktifkan server Laravel di folder iot\_25
7. Salin url dari halaman Ngrok, untuk dijalankan lewat terminal
8. Membuat project baru pada wokwi di platform.io dengan nama wokwi\_internet
9. Kemudian ganti kode program pada main.cpp dengan lampiran dibawah
10. Buka wokwi dan tambahkan ESP32 dan sensor DHT22 untuk diintegrasi
11. Jalankan perintah di terminal untuk mengakses Ngrok dengan url **ngrok http --scheme=http 8080**
12. Buat file baru diagram.json dan wokwi.toml
13. Setelah itu, maka dapat mulai untuk build kode program main.cpp
14. Lanjutkan dengan > request new license dan >start wokwi simulator
15. Berikutnya adalah melakukan modifikasi simulasi dengan menambahkan sensor suhu dan kelembaban
16. Simulator disini akan mengirim data ke database dengan mengganti kode diagram.json
17. Salin kode diagram.json yang ada di simulator wokwi ke file vscode yang sudah dibuat
18. Kemudian ubah setting file **platformio.ini**  dengan menambahkan **monitor\_speed** = 115200 **lib\_deps** = adafruit/DHT sensor library
19. Setelah itu, modifikasi file main.cpp dengan mengganti url ngrok yang sudah dijalankan di terminal
20. Untuk menjalankan, dapat memilai >wokwi simulator Tunggu terminal menginisialisasi dan menunjukkan server berada kode status http:200 agar pengiriman data berhasil dijalankan
21. Jika berhasil dijalankan, dapat dicek pada database, apakah data tersebut masuk. Jika data masuk, maka data tersebut berhasil dikirimkan.

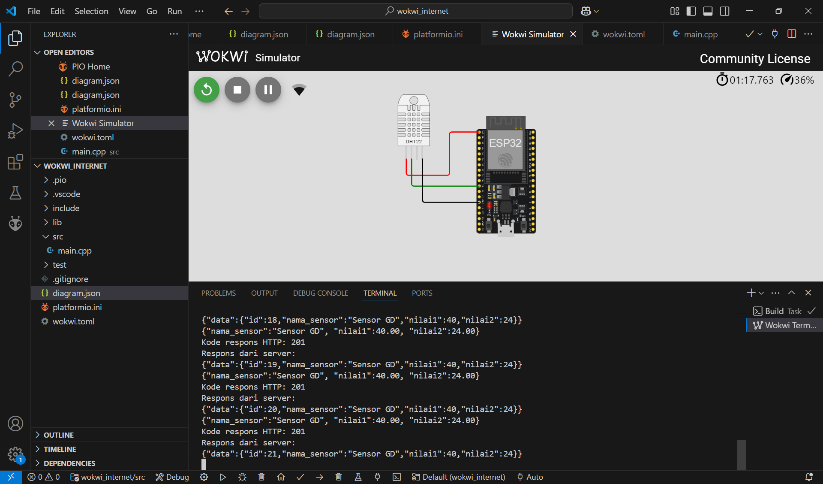
**3. Results and Discussion (Hasil dan Pembahasan)**

**3.1 Experimental Results (Hasil Eksperimen)**

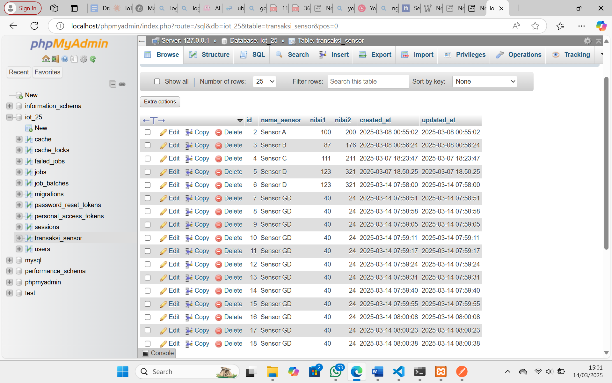
Sistem berhasil mengirimkan data sensor dari ESP32 ke server API melalui simulasi Wokwi. Kode HTTP 201 menunjukkan bahwa server berhasil menerima dan menyimpan data. Penggunaan Ngrok untuk mengekspos server lokal ke internet juga berhasil, terbukti dengan kemampuan ESP32 untuk mengakses endpoint API yang diekspos.

Respons dari server menunjukkan bahwa data yang dikirimkan sesuai dengan format yang diharapkan. Server juga memberikan ID unik untuk setiap pengiriman data, yang menunjukkan bahwa sistem pengelolaan data di server berfungsi dengan baik. Praktikum ini menunjukkan bahwa simulasi Wokwi dapat digunakan untuk mengimplementasikan dan menguji komunikasi antara perangkat IoT dengan server API.

Berikut hasil dari pengiriman data:



Hasil data yang masuk pada database:



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

Kode program pada main.cpp setelah dimodifikasi:

#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://060b-175-45-191-14.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();

  }

}

Kode Diagram.json:

{

    "version": 1,

    "author": "Avrilla",

    "editor": "wokwi",

    "parts": [

      { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 0, "left": 0, "attrs": {} },

      { "type": "wokwi-dht22", "id": "dht1", "top": -38.1, "left": -139.8, "attrs": {} }

    ],

    "connections": [

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

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

      [ "dht1:VCC", "esp:3V3", "red", [ "v28.8", "h76.8", "v-76.8" ] ],

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

      [ "dht1:GND", "esp:GND.1", "black", [ "v0" ] ]

    ],

    "dependencies": {}

  }

Kode 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'

**Kesimpulan:**

Praktikum akses API melalui simulasi Wokwi telah berhasil dilaksanakan dan memberikan pemahaman mendalam tentang komunikasi IoT berbasis API. Berdasarkan hasil eksperimen yang dilakukan Dengan mengintegrasikan VSCode, Wokwi, Postman, dan Ngrok, mahasiswa dapat memahami konsep komunikasi IoT berbasis API. ESP32 yang disimulasikan melalui Wokwi dapat berhasil mengirimkan data sensor ke server API dan menerima respons sesuai dengan yang diharapkan. Kode respons HTTP 201 menunjukkan bahwa server berhasil menerima dan menyimpan data yang dikirimkan.