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

**PRAKTIK AKSES API MELALUI SIMULASI WOKWI**

****

*Amelya Eka Wulandari*

233140700111005

[amelyaaeka@gmail.com](mailto:amelyaaeka@gmail.com)

**PROGRAM STUDI TEKNOLOGI INFORMASI**

**DEPARTEMEN INDUSTRI KREATIF DAN INOVASI**

**FAKULTAS VOKASI**

**UNIVERSITAS BRAWIJAYA**

**2025**

**ABSTRACTS**

Pada Praktikum ini, mahasiswa akan mempelajari bagaimana cara mengakses dan menghubungkan API Laravel menggunkana ESP32 melalui simulasi Wokwi. Langkah pertama yang dilakukan adalah menjalankan server Laravel agar bisa diakses secara public menggunakan NGROK, kemudian ESP32 dihubungkan ke jaringan WiFi Wokwi-Guest. Setelah itu, dilakukan simulasi pengiriman request HTTP GET dari ESP32 ke endpoint API untuk memastikan koneksi berhasil, yang ditandai dengan munculnya kode status HTTP 200.

Selanjutnya, praktikum dilanjutkan dengan menambahkan sensor suhu dan kelembaban (DHT22) pada simulasi. Data dari sensor ini dikirim ke API menggunakan metode HTTP POST, lalu disimpan ke dalam database MySQL. Semua proses diuji melalui tampilan serial monitor dan pengecekan data di database. Hasil dari simulasi ini menunjukkan bahwa ESP32 bisa digunakan untuk mengirim data sensor ke server Laravel secara real-time meskipun hanya melalui simulasi Wokwi. Praktikum ini memberikan gambaran bagaimana IoT bisa dimanfaatkan untuk integrasi perangkat keras dengan aplikasi web secara efisien.

**Kata Kunci:** IoT, ESP32, Laravel, Wokwi, HTTP

**BAB I**

**INTRODUCTION**

* 1. Latar Belakang

Internet of Things (IoT) merupakan konsep teknologi yang memungkinkan perangkat-perangkat fisik dapat saling terhubung dan bertukar data melalui internet. Dalam konteks praktikum ini, pembelajaran difokuskan pada bagaimana mikrokontroler ESP32 dapat digunakan untuk mengakses dan mengirimkan data ke sebuah API berbasis Laravel. Dengan bantuan simulator Wokwi, mahasiswa dapat memahami alur komunikasi data antara perangkat IoT dan server web tanpa perlu menggunakan perangkat keras secara langsung. Praktikum ini juga menunjukkan bagaimana integrasi antara sensor fisik (seperti DHT22) dengan sistem backend berbasis Laravel bisa dilakukan secara real-time, data yang dikirim disimpan langsung ke dalam databases MySQL.

* 1. Tujuan Eksperimen

1. Memahami proses integrasi ESP32 dengan API berbasis Laravel melalui simulasi
2. Melakukan simulasi pengiriman data dari sensor DHT22 ke API Laravel menggunakan metode HTTP POST.
3. Mengetahui bagaimana cara mengakses API Laravel secara publik menggunakan NGROK
4. Menyimpan dan menampilkan data sensor ke dalam database MySQL melalui API

**BAB II**

**METHODOLOGY**

2.1 Tools & Materials

Alat dan bahan yang digunakan untuk praktikum ini:

* Software : PlatformIO, Visual Studio Code, Ngrok, Mysql, Wokwi Simuator
* Environment : Xampp, Laravel11

2.2 Implementation Steps

1. Jalankan server Laravel dengan perintah php artisan serve --host=0.0.0.0 --port=8080
2. Membuka akses publik server Laravel menggunakan perintah ngrok http --scheme=http 8080
3. Membuat file main.cpp di PlatformIO untuk ESP32, dimulai dari simulasi HTTP GET ke API Laravel.
4. Menguji apakah ESP32 berhasil terhubung ke WiFi dan API dapat diakses (dengan mengecek status HTTP 200).
5. Menambahkan sensor DHT22 dalam simulasi dan menghubungkannya ke ESP32 melalui pin GPIO.
6. Memodifikasi kode main.cpp agar dapat membaca data suhu dan kelembaban, lalu mengirimkannya ke API Laravel menggunakan HTTP POST.
7. Mengecek hasil data yang dikirim melalui Serial Monitor dan memverifikasi bahwa data tersimpan ke database MySQL.

**BAB III**

**RESULTS AND DISCUSSION**

3.1 Experimental Results

1. Koneksi ESP32 ke API Laravel

Saat ESP32 berhasil terkoneksi dengan WiFi Wokwi-GUEST dan API Laravel yang sudah dibuka melalui NGROK, sistem mengembalikan HTTP status code **200 (OK)** yang menunjukkan bahwa server merespons dengan benar.

1. Pengiriman Data Sensor ke API

Setelah penambahan sensor DHT22, ESP32 dapat membaca data suhu dan kelembaban, lalu mengirimkannya ke endpoint API Laravel dalam format JSON

3.2 Kode Program

* 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://e6d3-2405-8740-6315-3520-5928-26b-7835-cd79.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 {

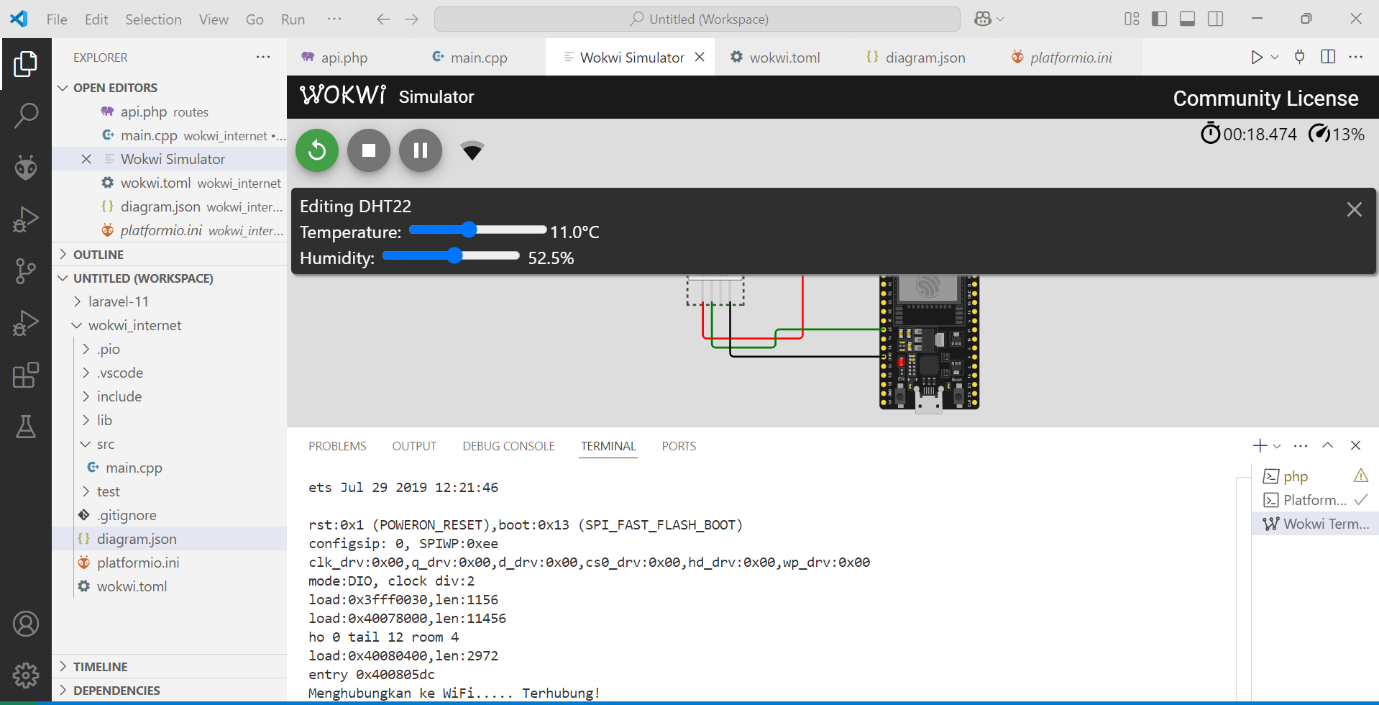
      Serial.println("WiFi tidak terhubung.");

    }

  }

}

**APPENDIX**

****