LAPORAN PRAKTIKUM INTERNET OF THINGS (IOT)

Fakultas Vokasi, Universitas Brawijaya **Praktik Akses API Melalui Simulasi WOKWI**

*Raka Sanjaya – 233140701111018*

*Fakultas Vokasi, Universitas Brawijaya*

*Email: sanjayaraka722@gmail.com*

**Abstract** (Abstrak)

Laporan ini membahas implementasi dan pengujian akses API menggunakan simulator Wokwi dengan mikrokontroler ESP32 yang terhubung ke backend Laravel. Percobaan dimulai dengan konfigurasi API Laravel agar dapat diakses melalui NGROK dan menghubungkan ESP32 ke jaringan Wi-Fi. Pengujian awal dilakukan dengan mengirim permintaan HTTP GET untuk memastikan koneksi ke API berjalan dengan baik. Selanjutnya, sensor suhu dan kelembaban DHT22 diintegrasikan dengan ESP32 untuk mengirimkan data sensor secara real-time melalui permintaan HTTP POST ke API, yang kemudian disimpan dalam database MySQL. Simulasi ini berhasil menunjukkan transmisi dan penyimpanan data secara real-time, membuktikan bahwa ESP32 dapat berkomunikasi dengan API Laravel menggunakan Wokwi simulator. Hasil pengujian mengonfirmasi bahwa kombinasi ESP32 dan Wokwi dapat digunakan secara efektif untuk pengumpulan dan pemrosesan data berbasis IoT..

**Kata kunci**: ESP32, Wokwi Simulator, API Laravel, NGROK, HTTP Request, IoT, Sensor DHT22, Transmisi Data, Database MySQL

1. **Pendahuluan**
2. **Latar Belakang**

Internet of Things (IoT) telah berkembang pesat dalam berbagai bidang, termasuk industri, kesehatan, dan sistem pemantauan lingkungan. Salah satu tantangan utama dalam pengembangan IoT adalah memastikan perangkat dapat berkomunikasi dengan server untuk mengirim dan menerima data secara real-time. Untuk mengatasi tantangan ini, diperlukan metode yang memungkinkan simulasi dan pengujian tanpa harus menggunakan perangkat keras secara langsung.

Wokwi Simulator adalah salah satu solusi yang memungkinkan simulasi perangkat keras seperti ESP32 tanpa harus memiliki perangkat fisik. Dengan simulator ini, pengembang dapat menguji koneksi jaringan, komunikasi dengan API, serta pengolahan data sensor secara virtual. Dalam laporan ini, ESP32 digunakan untuk mengakses API Laravel yang dihosting menggunakan NGROK, sehingga dapat diakses dari mana saja. Selain itu, sensor suhu dan kelembaban DHT22 diintegrasikan untuk mengirimkan data ke API dan menyimpannya dalam database MySQL.

Dengan adanya simulasi ini, pengujian IoT dapat dilakukan dengan lebih mudah, cepat, dan efisien tanpa perlu perangkat keras. Hal ini menjadi langkah penting dalam pengembangan sistem berbasis IoT, terutama bagi pengembang yang ingin melakukan validasi konsep sebelum menerapkannya pada perangkat fisik.

1. **Tujuan eksperimen**

* Mempelajari cara mengakses API menggunakan ESP32 melalui simulasi di Wokwi.
* Mengonfigurasi Laravel API agar dapat diakses dari ESP32 menggunakan NGROK.
* Menguji komunikasi antara ESP32 dan server dengan metode HTTP GET dan HTTP POST.
* Mengintegrasikan sensor suhu dan kelembaban DHT22 dengan ESP32 untuk mengirimkan data ke API.
* Menyimpan data sensor yang dikirim dari ESP32 ke dalam database MySQL melalui API Laravel.
* Mengevaluasi efektivitas Wokwi Simulator dalam mensimulasikan komunikasi IoT tanpa perangkat fisik.

1. **Metodologi**
2. **Alat dan Bahan**

* Alat dan bahan yang digunakan dalam eksperimen ini meliputi:
* Wokwi Simulator (untuk simulasi ESP32)
* PlatformIO (untuk pengembangan kode ESP32)
* Laravel 11 (sebagai backend API)
* NGROK (untuk menghubungkan API Laravel ke internet)
* MySQL Database (untuk menyimpan data sensor)
* Postman (opsional, untuk pengujian API)
* ESP32 DevKit V1
* Sensor DHT22 (suhu dan kelembaban)
* Wi-Fi (Wokwi-GUEST)
* HTTP Request (GET dan POST)

1. **Langkah Implementasi**
2. Menjalankan API Laravel.

A screenshot of a computer

AI-generated content may be incorrect.

1. Menjalankan NGROK untuk Menghubungkan API Laravel ke Internet

A screenshot of a computer program

AI-generated content may be incorrect.

1. Membuat File Proyek di Wokwi Simulator
2. Main.cpp

#include <Arduino.h>

#include <WiFi.h>

#include <WiFiClientSecure.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 = "";

WiFiClientSecure client;

HTTPClient http;

unsigned long previousMillis = 0;

const long interval = 5000;

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!");

  delay(2000); // Tambahan delay agar koneksi lebih stabil

  dht.begin();

  delay(1000);

  client.setInsecure();  // Menonaktifkan SSL verification (untuk ngrok)

}

void loop() {

  unsigned long currentMillis = millis();

  if (currentMillis - previousMillis >= interval) {

    previousMillis = currentMillis;

    float h = round(dht.readHumidity());

    float t = round(dht.readTemperature());

    if (isnan(h) || isnan(t)) {

      Serial.println(F("Failed to read from DHT sensor!"));

      return;

    }

    float hic = dht.computeHeatIndex(t, h, false);

    String url = "https://d674-175-45-191-254.ngrok-free.app:443/api/posts/";

    Serial.print("Menghubungkan ke: ");

    Serial.println(url);

    http.begin(client, url);

    http.useHTTP10(true);  // Gunakan HTTP 1.0 agar lebih stabil

    http.addHeader("Content-Type", "application/json");

    String payload = "{\"nama\_sensor\":\"Sensor GD\", \"nilai1\":" + String(h) + ", \"nilai2\":" + String(t) + "}";

    Serial.println("Mengirim data:");

    Serial.println(payload);

    int httpResponseCode = http.POST(payload);

    Serial.print("Kode respons HTTP: ");

    Serial.println(httpResponseCode);

    if (httpResponseCode > 0) {

      String response = http.getString();

      Serial.println("Respons dari server:");

      Serial.println(response);

    } else {

      Serial.println("Gagal mengirim data");

    }

    http.end();

  }

}

1. Menambahkan Sensor DHT22 dalam Simulasi

A screenshot of a computer

AI-generated content may be incorrect.

1. Mengonfigurasi Kode ESP32 untuk Mengirim Data Sensor ke API (HTTP POST)

A screenshot of a computer

AI-generated content may be incorrect.