**LAPORAN PRAKTIKUM INTERNET OF THINGS**

**PRAKTIK Akses API Melalui Simulasi WOKWI**

Fakultas Vokasi, Universitas Brawijaya

**Autors**

Fadhilah Azkia

Fakultas Vokasi, Universitas Brawijaya

Email : [fadhilahazkia95@gmail.com](mailto:fadhilahazkia95@gmail.com)

**Abstract**

Dalam praktik ini, dilakukan simulasi penggunaan ESP32 untuk mengakses dan mengirim data ke server Laravel melalui API, menggunakan platform simulator WOKWI. Proyek ini mencakup koneksi WiFi, pengambilan data dari sensor DHT22 (suhu dan kelembaban), serta pengiriman data ke server melalui metode HTTP POST. Hasil simulasi menunjukkan keberhasilan komunikasi antara ESP32 dan server Laravel, dengan HTTP status code 200 sebagai indikator suksesnya pertukaran data. Kegiatan ini memberikan pemahaman praktis mengenai integrasi sistem IoT dengan backend API.

1. **Introduction**  
   1. **Latar Belakang**

Praktikum ini bertujuan untuk mensimulasikan kondisi nyata di mana perangkat IoT mengirimkan data ke backend server. Dengan menggunakan WOKWI sebagai platform simulasi dan Laravel sebagai backend API, praktik ini menunjukkan bagaimana sistem IoT dapat berfungsi tanpa perlu perangkat fisik secara langsung, namun tetap memberikan hasil yang sama akurat.

* 1. **Tujuan Eksperimen**

- Mempelajari cara kerja API (Application Programming Interface) dalam sistem IoT.

- Mengimplementasikan komunikasi antara ESP32 dengan server Laravel melalui HTTP request.

- Melakukan simulasi penggunaan sensor suhu dan kelembaban (DHT22) untuk mengirim data ke API server.

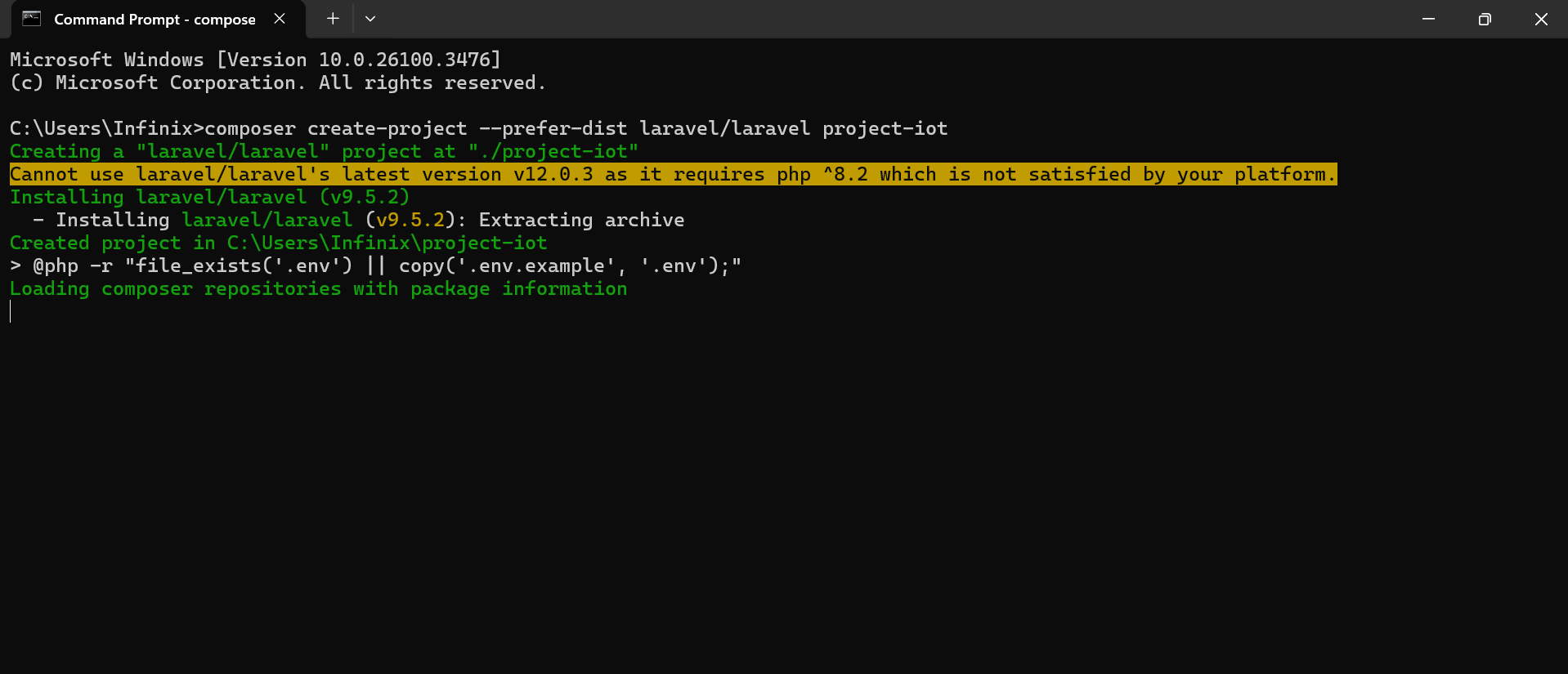
- Menggunakan WOKWI sebagai media simulasi ESP32 dan sensor.

- Memastikan data berhasil diterima dan disimpan ke dalam database melalui Laravel API.

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

* Hardware : Laptop
* Laravel
* Visual Studio Code
* XAMPP
* Ngrok
  1. **Langkah Implementasi**

1. Membuka Visual Studio Code
2. Install Laravel menggunakan command prompt



**Gambar 2.2.1** Install Laravel menggunakan command prompt.

1. Buat proyek baru di **PlatformIO** dengan board **ESP32 DevKit v1**. Tambahkan file main.cpp dengan skrip berikut:

**main.cpp (HTTP GET Request ke API Laravel)**

#include <WiFi.h>

#include <HTTPClient.h>

const char\* ssid = "Wokwi-GUEST";

const char\* password = "";

// URL server dari NGROK (pastikan http, bukan https)

const char\* serverUrl = "http://e6d3-2405-8740-6315-3520-5928-26b-7835-cd79.ngrok-free.app/api/posts";

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();

  if (currentMillis - previousMillis >= interval) {

    previousMillis = currentMillis;

    if (WiFi.status() == WL\_CONNECTED) {

      HTTPClient http;

      http.begin(serverUrl);

      int httpResponseCode = http.GET();

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

      Serial.println(httpResponseCode);

      http.end();

    } else {

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

    }

  }

}

1. Menjalankan Ngrok

Untuk mendapatkan URL pubik HTTP ( bukan HTTPS )

*ngrok http --scheme=http 8080*

1. Tambahkan Konfigurasi wokwi.toml

[wokwi]

version = 1

firmware = '.pio\build\esp32doit-devkit-v1\firmware.bin'

elf = '.pio\build\esp32doit-devkit-v1\firmware.elf'

1. Tambahkan Konfigurasi File diagram.json

{

  "version": 1,

  "author": "Kii",

  "editor": "wokwi",

  "parts": [

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

  ],

  "connections": [

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

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

  ]

}

1. Mulain menjalankan Simulasi

*Wokwi Start Simulator*

1. Modifikasi Simulasi sensor

Tujuannya adalah mengirimkan data suhu dan kemebaban ke API

{

  "version": 1,

  "author": "KAVITH BUDVIN",

  "editor": "wokwi",

  "parts": [

    { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 86.4, "left": 24.04, "attrs": {} },

    { "type": "wokwi-dht22", "id": "dht1", "top": 19.5, "left": -91.8, "attrs": {} }

  ],

  "connections": [

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

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

    [ "dht1:VCC", "esp:3V3", "red", [ "v19.2", "h67.2", "v-67.2" ] ],

    [ "esp:GND.1", "dht1:GND", "black", [ "h0" ] ],

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

  ],

  "dependencies": {}

}

1. Perubahan platform.ini

[env:esp32doit-devkit-v1]

platform = espressif32

board = esp32doit-devkit-v1

framework = arduino

monitor\_speed = 115200

lib\_deps = adafruit/DHT sensor library@^1.4.4

1. Mengubah file mine.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://e6d3-2405-8740-6315-3520-5928-26b-7835cd79.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();

  }

}

1. **Hasil dan Pembahasan**
   1. **Hasil Eksperimen**