LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

# Real Hardware WiFi Scanning & ESP32 Sensor DHT22 With Laravel API



*Fawwaz Mufid Wardaya*

*Fakultas Vokasi, Universitas Brawijaya*

*Email : mahesfawwaz79@gmail.com*

# **Abstract**

Tujuan eksperimen ini adalah untuk mempelajari cara mengintegrasikan perangkat keras IoT dengan aplikasi berbasis web menggunakan API RESTful. Proyek pertama membahas pemindaian jaringan WiFi menggunakan ESP32, dan proyek kedua menggunakan sensor suhu dan kelembaban DHT22 yang terhubung ke ESP32 untuk mengirim data ke server Laravel API melalui protokol HTTP POST. Data dikirim secara berkala dan disimpan dalam database server.

# 1. Introduction

## **1.1 Latar belakang**

Internet of Things (IoT) memungkinkan perangkat fisik terhubung ke sistem digital melalui internet. Sensor seperti DHT22 digunakan untuk mengumpulkan data lingkungan seperti suhu dan kelembaban, yang kemudian dapat dikirim ke server untuk diproses. ESP32 adalah mikrokontroler populer yang mendukung koneksi WiFi, menjadikannya ideal untuk proyek IoT.

Dalam proyek pertama, kami akan menggunakan ESP32 untuk memindai jaringan WiFi di sekitar dan menampilkan hasilnya di Serial Monitor. Dalam proyek kedua, kami akan menggunakan ESP32 untuk membaca data dari sensor DHT22 dan mengirimkannya ke server Laravel API melalui HTTP POST. Server Laravel akan menyimpan data tersebut dalam database.

## 1.2 Tujuan eksperimen

Proyek 1:

* Memahami cara menggunakan ESP32 untuk memindai jaringan WiFi.
* Menampilkan daftar jaringan WiFi beserta kekuatan sinyal dan status enkripsi.

Proyek 2:

* Mengintegrasikan sensor DHT22 dengan ESP32 untuk membaca suhu dan kelembaban.
* Mengirim data dari ESP32 ke server Laravel API menggunakan HTTP POST.
* Menyimpan data sensor dalam database server.

# 2. Methodology (Metodologi)

## 2.1 Tools & Materials (Alat dan Bahan)

Proyek 1:

* Software:
* Arduino IDE atau VSCode dengan ekstensi PlatformIO
* Hardware:
* ESP32
* Kabel USB Micro

Proyek 2:

* Software:
* Arduino IDE atau VSCode dengan ekstensi PlatformIO
* Laravel Framework
* Ngrok (untuk membuat URL publik)
* Hardware:
* ESP32
* Sensor DHT22
* Breadboard
* Kabel Jumper

## 2.2 Implementation Steps (Langkah Implementasi)

Proyek 1: WiFi Scanning dengan ESP32

1. Persiapan Perangkat Keras:

* Hubungkan ESP32 ke komputer menggunakan kabel USB Micro.

1. Pemrograman ESP32:

* Buka Arduino IDE atau VSCode dengan PlatformIO.
* Salin kode berikut ke file main.cpp:

#include <Arduino.h>

#include <WiFi.h>

void setup() {

Serial.begin(115200);

WiFi.mode(WIFI\_STA);

WiFi.disconnect();

delay(100);

Serial.println("Pemindaian Jaringan Wi-Fi Dimulai...");

}

void loop() {

int n = WiFi.scanNetworks();

Serial.println("Pemindaian Selesai");

if (n == 0) {

Serial.println("Tidak ada jaringan Wi-Fi yang ditemukan.");

} else {

Serial.print(n);

Serial.println(" jaringan Wi-Fi ditemukan:");

for (int i = 0; i < n; ++i) {

Serial.print(i + 1);

Serial.print(": ");

Serial.print(WiFi.SSID(i));

Serial.print(" (");

Serial.print(WiFi.RSSI(i));

Serial.print("dBm)");

Serial.println((WiFi.encryptionType(i) == WIFI\_AUTH\_OPEN) ? " " : "\*");

delay(10);

}

}

Serial.println("");

delay(5000); // Lakukan pemindaian setiap 5 detik

}

1. Upload dan Testing

* Upload kode ke ESP32.
* Buka Serial Monitor untuk melihat hasil pemindaian WiFi.

Proyek 2: WiFi Scanning dengan ESP32

1. Persiapan Perangkat Keras:

* Hubungkan sensor DHT22 ke ESP32 menggunakan breadboard dan kabel jumper.
* Pin DHT22:
* VCC → 3.3V
* GND → GND
* Data → GPIO27

1. Pemrograman ESP32:

* Buka Arduino IDE atau VSCode dengan PlatformIO.
* Salin kode berikut ke file 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 = "Sing butuh ae";

const char\* password = "tukangbakso";

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://60aa-175-45-191-13.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. Setup Laravel API:

* Pastikan Laravel API sudah dikonfigurasi untuk menerima data POST.
* Gunakan Ngrok untuk membuat URL publik untuk API.

1. Upload dan Testing:

* Upload kode ke ESP32.
* Amati data yang dikirim ke server Laravel melalui Serial Monitor.

# 3. Results and Discussion (Hasil dan Pembahasan)

## 3.1 Experimental Results (Hasil Eksperimen)

Proyek 1:

* ESP32 berhasil memindai jaringan WiFi di sekitar dan menampilkan daftar SSID, kekuatan sinyal (RSSI), dan status enkripsi.

Proyek 2:

* ESP32 berhasil membaca data suhu dan kelembapan dari sensor DHT22.
* Data dikirim ke server Laravel API melalui HTTP POST dan disimpan dalam database.
* Respons server menunjukkan bahwa data berhasil diterima.

## 3.2 Discussion (Pembahasan)

Proyek 1:

* Pemindaian WiFi menggunakan ESP32 sangat berguna untuk memantau jaringan di sekitar.
* Kode sederhana dan mudah dimodifikasi untuk kebutuhan lain.

Proyek 2:

* Integrasi ESP32 dengan sensor DHT22 dan Laravel API menunjukkan potensi IoT dalam aplikasi nyata.
* Penggunaan Ngrok mempermudah pengujian tanpa perlu hosting server fisik.

# 4. Appendix (Lampiran, jika diperlukan)











