# LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Praktik Real Hardware ESP32 (DHT22)**

*Eva Latifah*

*Fakultas Vokasi, Universitas Brawijaya*

*Email: ev.tifa@gmail.com*

**Abstrak**

Internet of Things (IoT) merupakan konsep penting dalam pengembangan teknologi cerdas yang menghubungkan perangkat fisik ke jaringan internet untuk saling bertukar data. Praktikum ini bertujuan untuk menguji konektivitas modul ESP32 terhadap jaringan WiFi serta mengimplementasikan pengiriman data sensor suhu dan kelembaban ke server API berbasis Laravel. Tahapan dimulai dengan melakukan pemindaian jaringan WiFi di sekitar menggunakan fungsi WiFi.scanNetworks() pada ESP32, kemudian menghubungkannya ke hotspot smartphone. Selanjutnya, data dari sensor DHT22 dibaca secara berkala dan dikirim ke endpoint HTTP menggunakan protokol POST melalui URL yang disediakan oleh Ngrok. Hasil pengujian menunjukkan bahwa ESP32 berhasil mendeteksi jaringan, terhubung ke WiFi, membaca data sensor, serta mengirimkannya ke server dengan respons sukses. Praktikum ini menunjukkan bahwa ESP32 dapat digunakan sebagai perangkat inti dalam sistem IoT sederhana dengan komunikasi data real-time ke server berbasis web.

**Kata Kunci**—*Internet of Things, ESP32, WiFi, Sensor DHT22, Laravel API, Ngrok*

***Abstract***

*The Internet of Things (IoT) is a key concept in the development of smart technologies, enabling physical devices to connect to the internet for data exchange. This experiment aims to test the WiFi connectivity of the ESP32 module and implement the transmission of temperature and humidity sensor data to a Laravel-based API server. The process begins by scanning nearby WiFi networks using the WiFi.scanNetworks() function on the ESP32 and connecting it to a smartphone hotspot. Sensor data from the DHT22 is then periodically read and sent via HTTP POST requests to a server endpoint provided through Ngrok. The test results show that the ESP32 successfully detects available networks, connects to WiFi, reads sensor data, and sends it to the server with a successful HTTP response. This experiment demonstrates the capability of ESP32 to serve as the core component in a basic IoT system with real-time web-based data communication.*

***Keywords****—Internet of Things, ESP32, WiFi, DHT22 Sensor, Laravel API, Ngrok*

**1. Introduction**

**1.1 Latar belakang**

Perkembangan teknologi Internet of Things (IoT) yang pesat telah membuka peluang luas dalam menciptakan sistem yang memungkinkan perangkat fisik saling terhubung dan berkomunikasi melalui jaringan internet. Salah satu komponen utama yang sering digunakan dalam proyek IoT adalah ESP32, yaitu mikrokontroler yang memiliki konektivitas WiFi dan Bluetooth secara terintegrasi dengan harga yang terjangkau.

Dalam penerapan IoT, koneksi jaringan yang stabil dan andal sangatlah penting. Sebelum mengembangkan sistem pemantauan atau otomatisasi yang lebih kompleks, langkah awal yang harus dilakukan adalah memastikan bahwa perangkat ESP32 mampu mendeteksi dan terhubung dengan jaringan WiFi di sekitarnya. Hal ini menjadi dasar penting dalam proses selanjutnya, yaitu mengirimkan data sensor ke server berbasis web.

Eksperimen ini bertujuan untuk menguji kemampuan ESP32 dalam melakukan pemindaian jaringan WiFi serta menghubungkannya ke Access Point yang tersedia. Setelah itu, dilakukan implementasi pengiriman data dari sensor suhu dan kelembaban (DHT22) ke API berbasis Laravel menggunakan metode HTTP POST. Untuk menjadikan API dapat diakses publik, digunakan layanan tunneling Ngrok. Melalui proses ini, pengguna dapat memahami alur dasar komunikasi sistem IoT, mulai dari perangkat tepi (edge device) hingga sistem backend berbasis web.

**1.2 Tujuan eksperimen**

Praktikum ini bertujuan untuk:

1. Mengetes kemampuan ESP32 dalam mendeteksi dan memindai jaringan WiFi di sekitarnya.
2. Memastikan konektivitas ESP32 ke jaringan WiFi lokal menggunakan kredensial yang sesuai.
3. Menghubungkan perangkat ESP32 ke API Laravel melalui URL publik yang disediakan oleh Ngrok.
4. Mengintegrasikan sensor DHT22 dengan ESP32 untuk membaca data suhu dan kelembaban.
5. Mengirimkan data sensor ke server melalui metode HTTP POST dan menyimpannya ke database MySQL.
6. Mengevaluasi keandalan komunikasi antara ESP32 dengan server Laravel secara real-time.
7. Memberikan pemahaman praktis tentang dasar penerapan sistem IoT dari sisi perangkat keras hingga backend.

**2. Methodology**

**2.1 Tools & Materials**

1. Komputer/Laptop dengan sistem operasi Windows/Linux/MacOS
2. Mikrokontroler ESP32
3. Kabel USB untuk koneksi ESP32 ke komputer
4. Breadboard
5. LED (2 buah)
6. Resistor 220 ohm (2 buah)
7. Kabel jumper
8. PlatformIO (terintegrasi dalam Visual Studio Code)
9. Driver Silicon Labs CP210x
10. Laravel 12
11. MySQL
12. Postman

**2.2 Implementation Steps**

#### **Unduh driver dari situs resmi:**

https://www.silabs.com/developer-tools/usb-to-uart-bridge-vcp-drivers

#### **Instalansi Driver**

* 1. Klik kanan perangkat yang tidak dikenali → *Update Driver*
  2. Pilih *Browse my computer for drivers*
  3. Pilih *Let me pick from a list...* lalu arahkan ke folder hasil ekstraksi driver
  4. Klik *Next* dan tunggu proses instalasi hingga selesai
  5. Pastikan tidak ada tanda peringatan di Device Manager

#### **Uji Coba Koneksi Wifi**

1. Buat file main.cpp dengan skrip untuk memindai jaringan WiFi menggunakan library WiFi.h.
2. Atur konfigurasi file platformio.ini

Gunakan board esp32doit-devkit-v1.  
Tambahkan monitor\_speed = 115200

1. Upload program ke ESP32 dan buka Serial Monitor.
2. Amati hasil pemindaian jaringan WiFi yang terdeteksi di sekitar ESP32.

#### **Rakit Perangkat Keras dan Implementasi**

1. Hubungkan sensor DHT22 ke pin 27 ESP32.
2. Pastikan koneksi kabel sesuai dengan diagram
3. Ganti ssid dan password dengan jaringan WiFi (misal tethering HP).
4. Ganti URL Ngrok di skrip dengan yang aktif.
5. Ubah platformio.ini:
6. Tambahkan dependensi DHT sensor dan Adafruit Unified Sensor.
7. Upload program ke ESP32.
8. Buka Serial Monitor untuk memantau output dan respons HTTP.
9. Periksa data di database MySQL untuk memastikan data terkirim

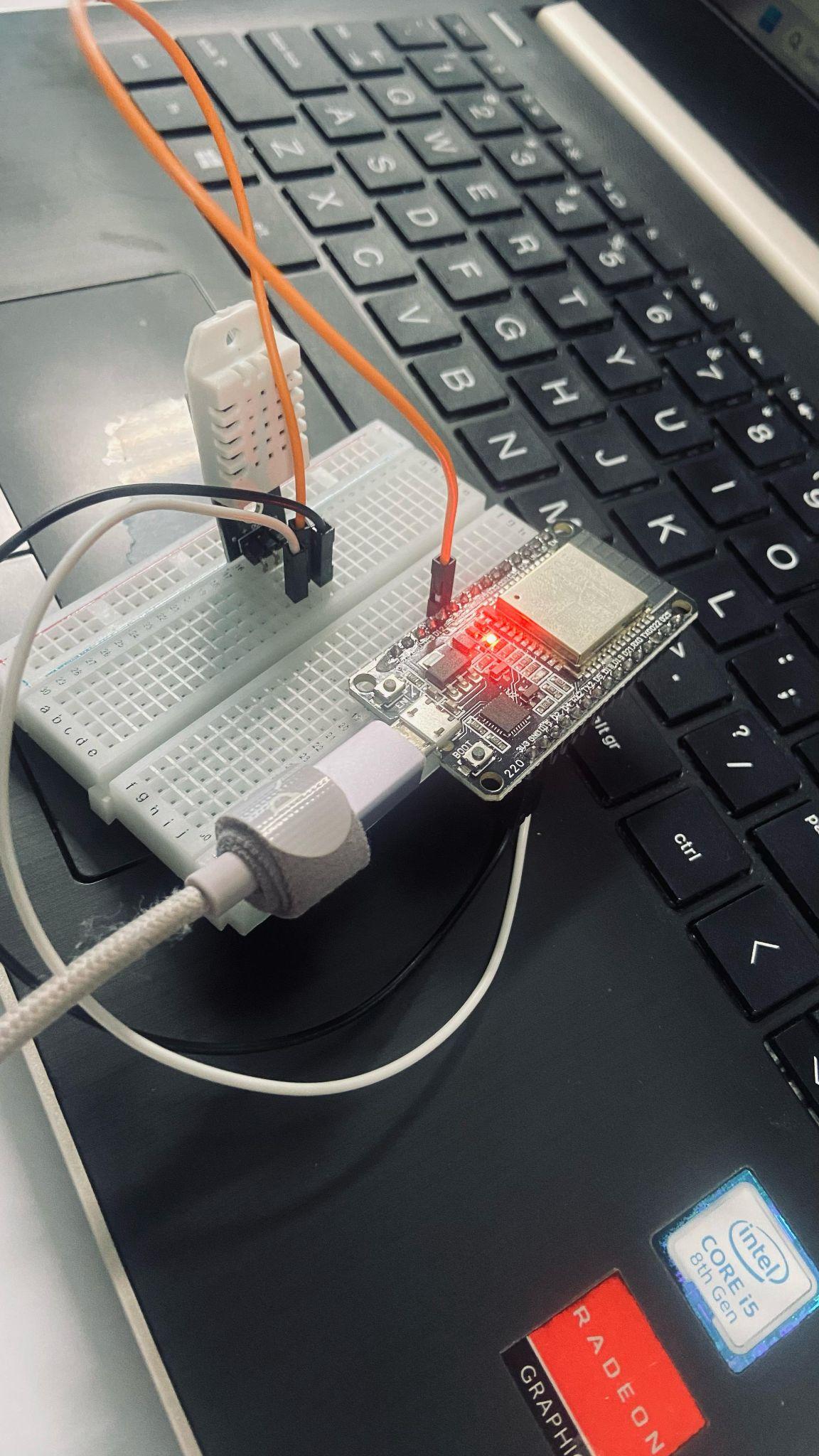
**3. Results and Discussion**

**3.1 Experimental Results**

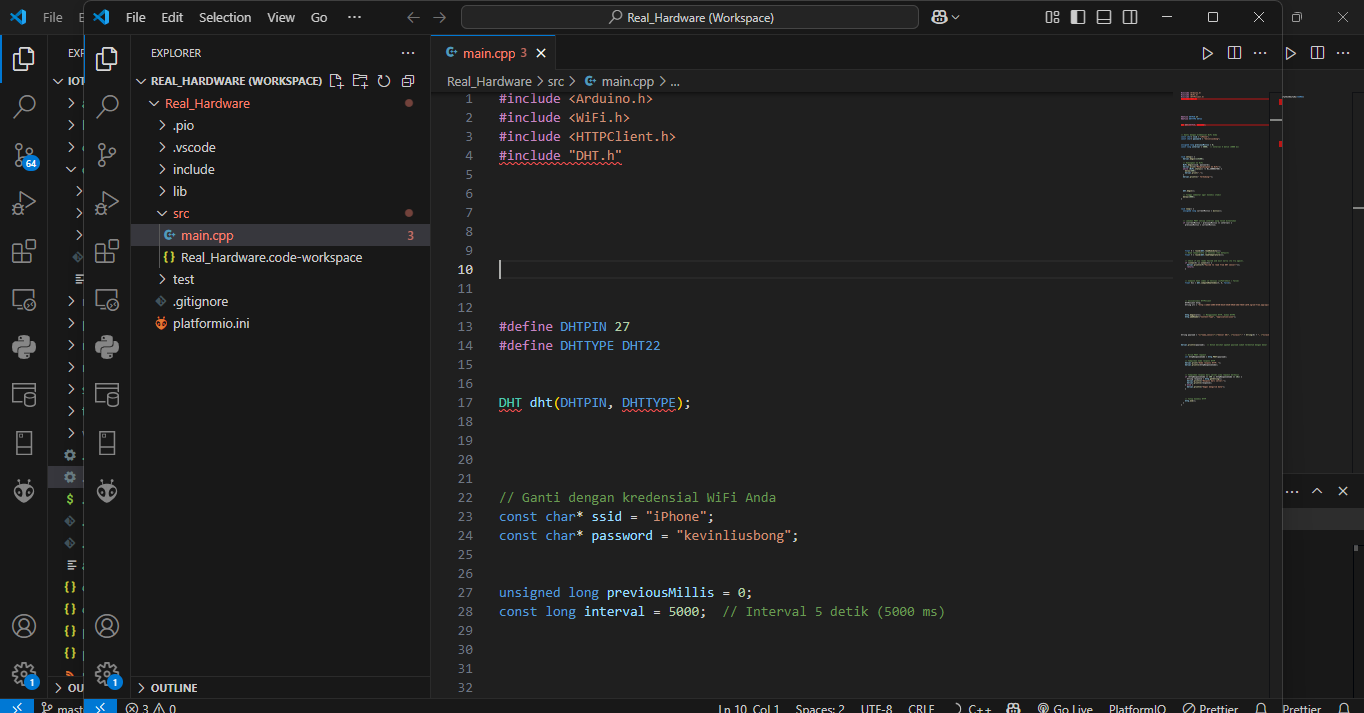
Pada eksperimen ini, proses pemindaian jaringan Wi-Fi dengan menggunakan ESP32 berhasil dilakukan dengan baik. ESP32 melakukan pemindaian jaringan Wi-Fi yang tersedia dan menampilkan informasi terkait kekuatan sinyal dari setiap jaringan yang terdeteksi. Setelah itu, perangkat terhubung dengan jaringan Wi-Fi yang dipilih, yaitu Wokwi-GUEST, yang memungkinkan perangkat untuk melanjutkan proses pengambilan data dari sensor DHT22. Data suhu dan kelembapan yang diperoleh dari sensor ini kemudian dikirimkan ke server Laravel melalui API menggunakan metode HTTP POST. URL API tersebut disambungkan melalui Ngrok, yang memungkinkan akses API Laravel secara publik. Setelah data dikirim, server merespons dengan kode HTTP yang mengonfirmasi apakah data berhasil diterima atau tidak. Eksperimen ini membuktikan bahwa ESP32 mampu terhubung ke jaringan Wi-Fi, mengambil data dari sensor, dan mengirimkan data tersebut secara otomatis ke server. Namun, masih terdapat beberapa tantangan terkait kestabilan jaringan dan penggunaan protokol HTTP yang tidak terenkripsi. Sebagai langkah perbaikan di masa depan, pengimplementasian HTTPS untuk komunikasi yang lebih aman dapat dipertimbangkan. Selain itu, dengan memastikan koneksi yang stabil dan penggunaan autentikasi API, sistem ini bisa lebih efektif dan aman untuk aplikasi IoT yang lebih besar.

#### 

#### **Hasil Real Hardware**



#### **Hasil VS Code**



**4. Appendix**

1. 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 = "iPhone";

const char\* password = "kevinliusbong";

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-7835-cd79.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();

}

}