

LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

FAKULTAS VOKASI, UNIVERSITAS BRAWIJAYA

Restu Nur Kharisma

Fakultas Vokasi, Universitas Brawijaya

Email: restukharisma@student.ub.ac.id

ABSTRACT

This practical report presents the implementation of an Internet of Things (IoT) project using the ESP32 microcontroller and DHT22 temperature and humidity sensor. The system is designed to collect real-time environmental data and transmit it to a remote database for monitoring and analysis purposes. The ESP32 was programmed to read data from the DHT22 sensor at regular intervals and send the readings via Wi-Fi to a web-based server using HTTP POST requests. The received data was stored in a structured database for future retrieval and visualization. This project demonstrates the integration of hardware and software in an IoT ecosystem, highlighting the ESP32's capabilities in wireless data transmission and sensor interfacing.

Keywords — ESP32, DHT22, Internet of Things, Temperature Sensor, Data Logging

1. Pendahuluan

1.1. Latar Belakang

Perkembangan teknologi Internet of Things (IoT) semakin pesat dan telah banyak diterapkan dalam berbagai bidang, termasuk pemantauan lingkungan. Salah satu penerapan yang umum adalah sistem monitoring suhu dan kelembaban secara real-time. Dengan memanfaatkan mikrokontroler seperti ESP32 yang memiliki konektivitas Wi-Fi, serta sensor suhu dan kelembaban seperti DHT22, data lingkungan dapat dikumpulkan dan dikirimkan ke server atau database untuk dianalisis dan disimpan.

Penggunaan sensor DHT22 memungkinkan akuisisi data suhu dan kelembaban dengan tingkat akurasi yang cukup baik, sedangkan ESP32 berfungsi sebagai pengendali utama yang mengatur pembacaan sensor dan pengiriman data ke database. Dengan mengintegrasikan kedua perangkat ini, maka dapat dibangun sistem monitoring sederhana namun efektif, yang dapat dimanfaatkan untuk berbagai kebutuhan seperti sistem kontrol suhu ruangan, pertanian cerdas, maupun pemantauan lingkungan di area tertentu.

1.2. Tujuan Eksperimen

- 1) Merancang dan mengimplementasikan sistem monitoring suhu dan kelembaban menggunakan mikrokontroler ESP32 dan sensor DHT22.
- 2) Menghubungkan sistem ke jaringan WiFi untuk mengirimkan data ke database secara real-time.
- 3) Menyimpan data suhu dan kelembaban ke dalam database untuk keperluan dokumentasi dan analisis.

2. Metodologi

2.1. Alat dan Bahan

- 1) Alat: ESP32, kabel jumper, kabel USB, laptop

- 2) Bahan: Sensor DHT22, database, Visual Studio Code, XAMPP, jaringan internet

2.2. Langkah Implementasi

- 1) Persiapan Perangkat Keras
 - Sambungkan sensor DHT22 ke ESP32
 - Hubungkan ESP32 ke laptop menggunakan kabel USB
- 2) Persiapan program untuk API antara ESP32 ke database
 - Jalankan API Laravel dengan perintah
php artisan serve --host=0.0.0.0 --port=8080
 - Jalankan NGROK dengan command
ngrok http --scheme=http 8080
pastikan alamat URL ngrok berupa HTTP bukan HTTPS
- 3) Persiapan program untuk ESP32 dan sensor DHT22
 - Buat new project di PlatformIO dan ubah isi main.cpp seperti berikut

```
• #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 = "Restu";
• const char* password = "restu12345";
•
• 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://01a7-175-45-191-11.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();
• }
• }

```

- Kemudian lakukan modifikasi kembali pada file platformio.ini seperti berikut.

```

• [env:esp32doit-devkit-v1]
• platform = espressif32
• board = esp32doit-devkit-v1
• framework = arduino
• upload_port = /dev/tty.usbserial-0001
• monitor_port = /dev/tty.usbserial-0001
• monitor_speed = 115200
• lib_deps =
•   adafruit/DHT sensor library@^1.4.4
•   adafruit/Adafruit Unified Sensor@^1.1.14

```

- Lakukan proses build dan upload, lalu jalankan simulasi
- Testing data yang dikirim dari hardware ESP32 dapat masuk ke database

3. Hasil dan Pembahasan

3.1. Hasil Eksperimen

Data berhasil masuk ke dalam tabel database

The screenshot shows the phpMyAdmin interface in a web browser. The 'transaksi_sensor' table is selected, and the data is displayed in a table format. The table has 6 columns: id, nama_sensor, nilai1, nilai2, created_at, and updated_at. The data shows 24 rows of sensor readings, including Sensor A, Sensor B, Sensor C, Sensor D, and Sensor GD, with values for nilai1 and nilai2, and timestamps for created_at and updated_at.

id	nama_sensor	nilai1	nilai2	created_at	updated_at
1	Sensor A	100	200	2025-02-25 10:54:46	2025-02-25 10:54:46
2	Sensor B	87	176	2025-02-25 10:55:02	2025-02-25 10:55:02
3	Sensor C	111	211	2025-02-25 03:39:39	2025-02-25 03:39:39
5	Sensor D	200	300	2025-02-25 05:35:41	2025-02-25 05:35:41
6	Sensor GD	40	24	2025-03-17 09:50:10	2025-03-17 09:50:10
7	Sensor GD	40	24	2025-03-17 09:50:24	2025-03-17 09:50:24
8	Sensor GD	65	27	2025-04-15 04:23:36	2025-04-15 04:23:36
9	Sensor GD	65	27	2025-04-15 04:23:41	2025-04-15 04:23:41
10	Sensor GD	65	27	2025-04-15 04:23:45	2025-04-15 04:23:45
11	Sensor GD	65	27	2025-04-15 04:23:50	2025-04-15 04:23:50
12	Sensor GD	65	27	2025-04-15 04:23:56	2025-04-15 04:23:56
13	Sensor GD	65	27	2025-04-15 04:24:00	2025-04-15 04:24:00
14	Sensor GD	65	27	2025-04-15 04:24:06	2025-04-15 04:24:06
15	Sensor GD	65	27	2025-04-15 04:24:11	2025-04-15 04:24:11
16	Sensor GD	65	27	2025-04-15 04:24:15	2025-04-15 04:24:15
17	Sensor GD	65	27	2025-04-15 04:24:20	2025-04-15 04:24:20