

# **LAPORAN PRAKTIKUM INTERNET OF THINGS (IOT) SISTEM INFORMASI CUACA *REALTIME* MENGGUNAKAN API**

***OPENWEATHERMAP***

**FAKULTAS VOKASI**

**UNIVERSITAS BRAWIJAYA**

*Nasywa Anindya Q.E*

*Fakultas Vokasi Universitas Brawijaya*

*Email: [nasywaanindya@student.ub.ac.id](mailto:nasywaanindya@student.ub.ac.id)*

## **ABSTRAK**

Praktik ini bertujuan untuk merancang sistem informasi cuaca berbasis ESP32 yang terhubung ke jaringan WiFi dan menampilkan data cuaca secara real-time pada layar LCD I2C. Sistem ini mengambil data cuaca dari API OpenWeatherMap, meliputi suhu, kelembapan, tekanan udara, kecepatan angin, serta waktu matahari terbit dan terbenam. Dua buah tombol disediakan untuk memudahkan pengguna berpindah antar tampilan informasi cuaca. Data cuaca diperbarui setiap satu menit agar informasi yang ditampilkan selalu terbaru. Proyek ini dikembangkan menggunakan simulator Wokwi dan teks editor VSCode. Hasil pengujian menunjukkan bahwa sistem dapat menampilkan dan memperbarui data cuaca secara akurat dan berkala.

Kata Kunci: ESP32, Wokwi, *OpenWeatherMap*, *Internet of Things*, API

## **ABSTRACT**

*This project aims to design a weather information system based on ESP32, connected to WiFi, and displaying real-time weather data on an I2C LCD screen. The system retrieves weather data from the OpenWeatherMap API, including temperature, humidity, air pressure, wind speed, and sunrise and sunset times. Two buttons are provided to allow users to switch between different weather displays. The weather data is refreshed every minute to ensure up-to-date information. The project was developed using the Wokwi simulator and VSCode as the text editor. The test results show that the system is able to display and update weather data accurately and periodically.*

*Keywords: ESP32, Wokwi, OpenWeatherMap, Internet of Things, API*

# 1. PENDAHULUAN

## 1.1. Latar Belakang

Cuaca berperan penting dalam mendukung aktivitas manusia sehari-hari, sehingga informasi cuaca yang tepat dan terbaru sangat dibutuhkan. Pada proyek ini digunakan ESP32, sebuah board berbasis Arduino open-source, untuk membuat sistem informasi cuaca yang mampu menampilkan data suhu, kelembapan, tekanan udara, kecepatan angin, serta waktu matahari terbit dan terbenam. Dengan memanfaatkan koneksi WiFi pada ESP32, sistem ini dapat memperoleh data langsung dari internet menggunakan API OpenWeatherMap. Data cuaca ditampilkan secara real-time melalui LCD I2C, sehingga pengguna dapat memantau perubahan cuaca secara langsung. Proyek ini dikembangkan menggunakan simulator Wokwi dan teks editor Visual Studio Code (VSCode).

## 1.2. Tujuan

Tujuan dari pelaksanaan praktikum ini adalah:

1. Membangun sistem informasi cuaca berbasis ESP32 yang terhubung ke internet melalui WiFi.
2. Menampilkan informasi cuaca secara real-time pada LCD I2C.
3. Mempelajari proses pengambilan dan pengolahan data cuaca melalui API menggunakan Arduino.

# 2. METODOLOGI

## 2.1. Alat dan Bahan

### 2.1.1. Alat

- Laptop
- Simulator Wokwi
- Visual Studio Code (VSCode)
- OpenWeatherMap

### 2.1.2. Bahan

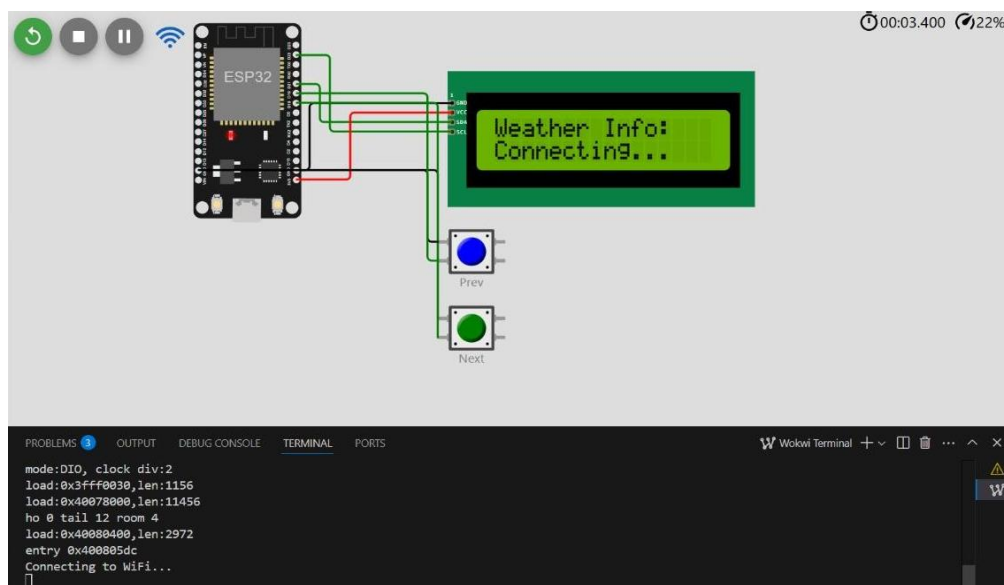
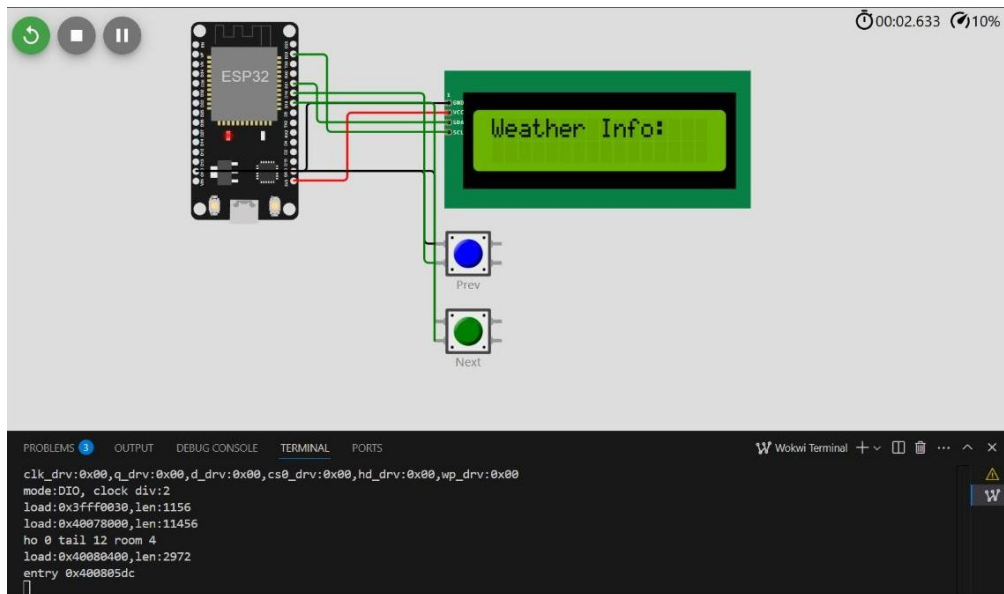
- ESP32 (dalam bentuk virtual pada Wokwi)
- LCD 16x2 I2C
- Push button
- WiFi
- Bahasa pemrograman C++ dengan pustaka Arduino
- Library:
  - a. WiFi.h
  - b. Wire.h
  - c. HTTPClient.h
  - d. LiquidCrystal\_I2C.h
  - e. ArduinoJson.h

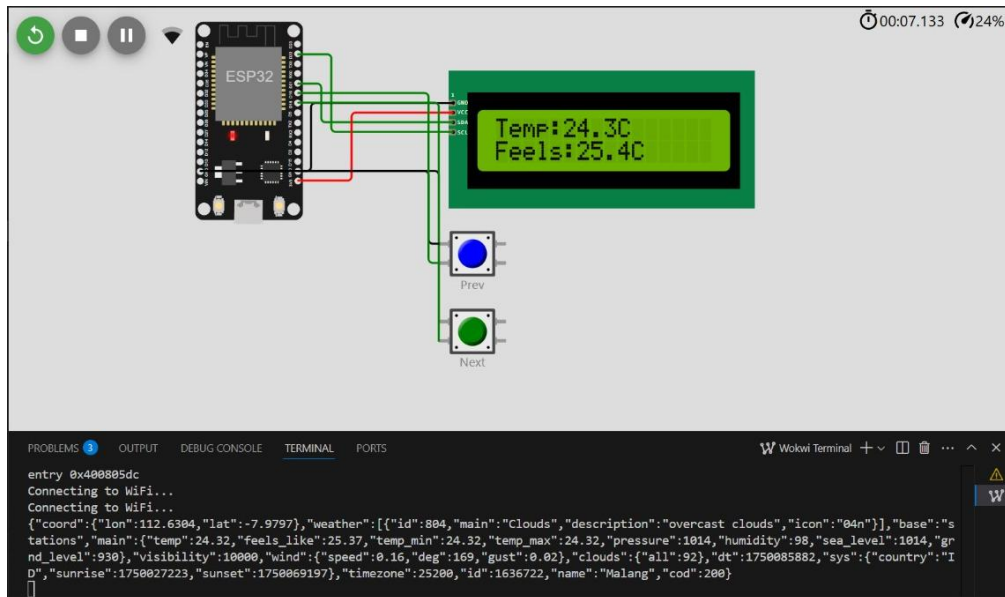
## 2.2. Langkah Implementasi

- a. Jalankan VSCode.
- b. Pasang ekstensi PlatformIO dan Wokwi di VSCode.
- c. Buat proyek baru dengan board ESP32 pada PlatformIO.
- d. Akses situs Wokwi untuk membuat simulasi baru.
- e. Tambahkan komponen LCD 16x2 I2C dan dua push button pada diagram.
- f. Gunakan library LiquidCrystal\_I2C untuk kendali LCD dan WiFi.h untuk koneksi WiFi.
- g. Manfaatkan HTTPClient untuk mengambil data dari OpenWeatherMap API.
- h. Tulis kode program untuk sistem.
- i. Daftar dan dapatkan API key dari OpenWeatherMap.
- j. Tentukan lokasi (kota atau negara) yang datanya ingin ditampilkan.
- k. Atur agar data diperbarui setiap satu menit.
- l. Salin kode diagram.json dari Wokwi ke VSCode setelah desain selesai.
- m. Jalankan dan uji program.

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Hasil





### 3.2. Pembahasan

Proyek ini membuktikan bahwa Arduino, khususnya ESP32, dapat dimanfaatkan untuk mengambil data cuaca secara real-time dan menampilkannya dengan jelas pada layar LCD. Sistem mampu terhubung ke API OpenWeatherMap serta mengolah data dalam format JSON dengan baik. Namun, ada beberapa kendala yang ditemui, terutama saat menangani data dalam jumlah besar dari API.

- a. Keterbatasan Tampilan LCD  
LCD 16x2 memiliki ruang tampilan yang terbatas sehingga tidak semua informasi cuaca dapat ditampilkan sekaligus. Untuk itu, tombol digunakan agar pengguna dapat berpindah antar halaman informasi.
- b. Stabilitas Koneksi WiFi  
Pada simulator Wokwi, koneksi WiFi terkadang memerlukan waktu lebih lama untuk stabil. Hal ini dapat memengaruhi kecepatan sistem dalam memperoleh data cuaca.
- c. Navigasi Informasi Cuaca  
Tombol disediakan agar pengguna dapat dengan mudah berganti halaman untuk melihat informasi cuaca yang berbeda.
- d. Pembaruan Data  
Sistem secara otomatis memperbarui data cuaca setiap satu menit guna menampilkan informasi yang selalu terkini.

## 4. LAMPIRAN

### 4.1. Kode Program Main.cpp

```
5. #include <Arduino.h>
6.
7. #define BLYNK_TEMPLATE_ID "TMPL69S1hvRW1"
8. #define BLYNK_TEMPLATE_NAME "Nasywa Anindya"
9. #define BLYNK_AUTH_TOKEN "GGx0zN43sAX4pglUa_TcOuILjabUWQS1"
10. #define BLYNK_DEVICE_NAME "suhu dan kelembapan"
11. #define BLYNK_PRINT Serial
12.
13. #include <WiFi.h>
14. #include <BlynkSimpleEsp32.h>
```

```
15. #include <DHTesp.h>
16.
17. char auth[] = BLYNK_AUTH_TOKEN;
18. char ssid[] = "Wokwi-GUEST";
19. char pass[] = "";
20.
21. const int DHT_PIN = 15;
22. DHTesp dht;
23. BlynkTimer timer;
24.
25. // Fungsi untuk kirim data sensor ke Blynk
26. void sendSensor()
27. {
28.   TempAndHumidity data = dht.getTempAndHumidity();
29.
30.   Serial.print("% Temperature: ");
31.   Serial.print(data.temperature);
32.   Serial.println(" C");
33.
34.   Serial.print("% Kelembaban: ");
35.   Serial.print(data.humidity);
36.   Serial.println(" %");
37.
38.   Blynk.virtualWrite(V0, data.temperature);
39.   Blynk.virtualWrite(V1, data.humidity);
40. }
41.
42. void setup()
43. {
44.   Serial.begin(115200);
45.   dht.setup(DHT_PIN, DHTesp::DHT22);
46.
47.   Blynk.begin(auth, ssid, pass);
48.   timer.setInterval(1000, sendSensor);
49. }
50.
51. void loop()
52. {
53.   Blynk.run();
54.   timer.run();
55. }
56.
```