

# **PRAKTIKUM SISTEM MONITORING CUACA DENGAN ESP32 BERBASIS API DAN SENSOR DHT22**

*Fredlina Devhania Kholishah<sup>1</sup>*  
*Fakultas Vokasi, Universitas Brawijaya*  
*[devhania88@gmail.com](mailto:devhania88@gmail.com)*

## **ABSTRAK**

Pada praktikum ini dikembangkan sebuah sistem pemantauan cuaca berbasis ESP32 yang mampu mengambil data cuaca real-time melalui API OpenWeatherMap serta membaca suhu dan kelembapan menggunakan sensor DHT22. Informasi cuaca yang diperoleh ditampilkan pada OLED display dan indikator LED menyala berdasarkan kondisi suhu. Sistem ini mampu memberikan informasi cuaca terkini seperti suhu, kelembapan, tekanan udara, deskripsi cuaca, dan waktu matahari terbit/terbenam. Proyek ini bermanfaat sebagai media pembelajaran integrasi mikrokontroler dengan layanan API berbasis internet (IoT).

***Kata kunci:*** *ESP32, IoT, DHT22, OLED, OpenWeatherMap, Pemantauan Cuaca*

## **ABSTRACT**

This practicum developed a weather monitoring system based on ESP32, capable of retrieving real-time weather data using the OpenWeatherMap API and reading temperature and humidity with the DHT22 sensor. The retrieved weather data is displayed on an OLED screen, and LEDs are used as indicators based on temperature conditions. The system provides weather information such as temperature, humidity, air pressure, weather description, and sunrise/sunset time. This project serves as a learning medium for integrating microcontrollers with internet-based API services in IoT applications.

***Keywords:*** *ESP32, IoT, DHT22, OLED, OpenWeatherMap, Weather Monitoring*

# **1. PENDAHULUAN**

## **1.1 Latar Belakang**

Internet of Things (IoT) adalah konsep yang memungkinkan perangkat fisik saling terhubung dan bertukar data melalui jaringan internet. Dalam dunia modern, IoT memainkan peran penting dalam berbagai bidang seperti rumah pintar, pertanian, transportasi, hingga sistem monitoring lingkungan. Salah satu implementasi yang umum dan berguna adalah sistem pemantauan cuaca.

Pemantauan cuaca sangat penting untuk aktivitas sehari-hari maupun industri seperti pertanian, perikanan, dan transportasi. Dengan memanfaatkan teknologi mikrokontroler seperti ESP32 dan sensor DHT22, serta integrasi dengan layanan data cuaca global seperti OpenWeatherMap, kita dapat menciptakan sistem cerdas yang mampu menampilkan informasi cuaca real-time baik dari lingkungan sekitar maupun dari data internet.

## **1.2 Rumusan Masalah**

- Bagaimana merancang sistem pemantauan cuaca menggunakan ESP32 yang mampu mengambil data dari sensor DHT22 dan API OpenWeatherMap?
- Bagaimana menampilkan data cuaca secara real-time melalui OLED display dan indikator LED?

## **1.3 Tujuan Praktikum**

- Menerapkan integrasi antara ESP32, sensor DHT22, OLED, dan API cuaca berbasis HTTP.
- Menampilkan informasi cuaca seperti suhu, kelembapan, tekanan udara, dan kondisi langit.
- Memberikan visualisasi kondisi suhu melalui LED indikator.

## **1.4 Manfaat Praktikum**

- Mahasiswa memahami konsep dan penerapan Internet of Things dalam bidang monitoring lingkungan.
- Memberikan pengalaman langsung dalam mengakses dan memproses data dari API.
- Melatih kemampuan merangkai sistem perangkat keras dan pemrograman mikrokontroler secara simultan.

## 2. METODOLOGI

### 2.1 Perangkat Keras yang Digunakan

- ESP32 sebagai mikrokontroler utama
- Sensor DHT22 untuk pengukuran suhu dan kelembapan
- OLED display 0.96" untuk menampilkan informasi cuaca
- 3 LED (merah, kuning, hijau) sebagai indikator suhu
- Koneksi internet (melalui Wokwi)

### 2.2 Rangkaian

Komponen-komponen disusun pada simulator Wokwi dengan skema sebagai berikut:

- DHT22 (salah satu pin digital ESP32)
- OLED (SDA ke GPIO21, SCL ke GPIO22)
- LED Merah, Kuning, Hijau ke pin digital yang berbeda
- ESP32 terkoneksi internet untuk akses API

### 2.3 Alur Sistem

1. ESP32 tersambung ke WiFi.
2. Data cuaca diambil dari OpenWeatherMap menggunakan HTTP request.
3. Informasi seperti suhu, kelembapan, tekanan, dan cuaca ditampilkan pada OLED.
4. Sensor DHT22 membaca suhu dan kelembapan lokal sebagai perbandingan.
5. LED menyala sesuai rentang suhu:
  - Merah:  $> 30^{\circ}\text{C}$
  - Kuning:  $25\text{--}30^{\circ}\text{C}$
  - Hijau:  $< 25^{\circ}\text{C}$

## 3. HASIL DAN PEMBAHASAN

### 3.1 Hasil Terminal

Data cuaca berhasil ditampilkan di terminal:

```
}, "clouds": {"all": 92}, "dt": 1750088173, "sys": {"country": "ID", "sunrise": 1750027223, "sunset": 1750069197}, "timezone": 25200, "id": 1636722, "name": "Malang", "cod": 200}
===== Weather Info =====
City      : Malang
Country   : ID
Weather   : Clouds
Description : overcast clouds
Temperature : 24.01 °C
Feels Like  : 25.05 °C
Min Temp   : 24.01 °C
Max Temp   : 24.01 °C
Pressure   : 1013 hPa
Humidity    : 99 %
Wind Speed  : 0.24 m/s
Wind Degree : 123 °
Sunrise     : 22:40:23
Sunset      : 10:19:57
=====
```

## 3.2 Tampilan OLED

Layar OLED menampilkan:

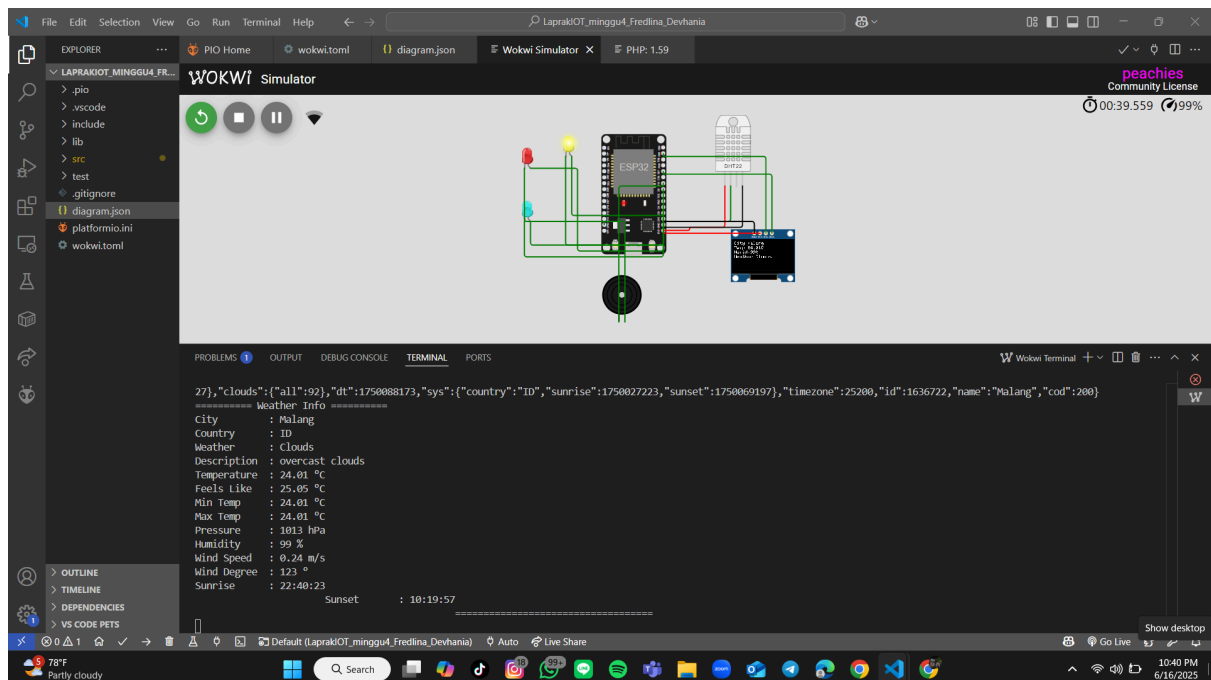


## 3.3 Pembahasan

Data dari API menunjukkan suhu dan kelembapan sesuai dengan yang dibaca sensor. LED hijau menyala karena suhu di bawah 25°C. Proyek berhasil menunjukkan pemrosesan data dari API dan sensor lokal secara simultan serta menampilkannya secara visual.

## 4. LAMPIRAN

### 4.1 Gambar Rangkaian Wokwi



## 4.2 Kode Program

### Kode Program Main.cpp

```
#include <Wire.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_GFX.h>
#include <DHT.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>

// WiFi Setup
const char* ssid = "Wokwi-GUEST";
const char* password = "";

// OpenWeatherMap
String apiKey = "6375be75703e403d828c83238072ae30";
String city = "Malang";
String units = "metric";
String server = "http://api.openweathermap.org/data/2.5/weather?q=" +
city + "&units=" + units + "&appid=" + apiKey;

// OLED
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

// DHT Sensor
#define DHTPIN 15
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

// LED Pins
#define LED_HOT 2
#define LED_COLD 4
#define LED_NORMAL 5

// Buzzer Pin
#define BUZZER_PIN 18

void setup() {
    Serial.begin(115200);
```

```

// Start OLED
if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
}
display.display();
delay(1000);
display.clearDisplay();

// Start DHT
dht.begin();

// LED and Buzzer setup
pinMode(LED_HOT, OUTPUT);
pinMode(LED_COLD, OUTPUT);
pinMode(LED_NORMAL, OUTPUT);
pinMode(BUZZER_PIN, OUTPUT);

// Connect WiFi
WiFi.begin(ssid, password);
display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.setCursor(0, 10);
display.print("Connecting WiFi...");
display.display();

while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}

display.clearDisplay();
display.setCursor(0, 10);
display.print("WiFi Connected!");
display.display();
delay(1000);
}

void loop() {
    display.clearDisplay();

    // DHT Local Sensor Read
    float h = dht.readHumidity();

```

```

float t = dht.readTemperature();

// Check WiFi
if ((WiFi.status() == WL_CONNECTED)) {
    HTTPClient http;
    http.begin(server);
    int httpCode = http.GET();

    if (httpCode > 0) {
        String payload = http.getString();
        Serial.println(payload);

        // Parsing JSON
        StaticJsonDocument<1024> doc;
        DeserializationError error = deserializeJson(doc, payload);

        if (!error) {
            String cityName = doc["name"];
            String country = doc["sys"]["country"];
            String weatherMain = doc["weather"][0]["main"];
            String weatherDesc = doc["weather"][0]["description"];
            float temp = doc["main"]["temp"];
            float feelsLike = doc["main"]["feels_like"];
            float tempMin = doc["main"]["temp_min"];
            float tempMax = doc["main"]["temp_max"];
            int pressure = doc["main"]["pressure"];
            int humidity = doc["main"]["humidity"];
            float windSpeed = doc["wind"]["speed"];
            int windDeg = doc["wind"]["deg"];
            long sunrise = doc["sys"]["sunrise"];
            long sunset = doc["sys"]["sunset"];

            // Print ke Serial Monitor
            Serial.println("===== Weather Info =====");
            Serial.println("City          : " + cityName);
            Serial.println("Country       : " + country);
            Serial.println("Weather       : " + weatherMain);
            Serial.println("Description   : " + weatherDesc);
            Serial.print("Temperature   : "); Serial.print(temp);
Serial.println(" °C");
            Serial.print("Feels Like    : "); Serial.print(feelsLike);
Serial.println(" °C");

```

```

        Serial.print("Min Temp      : "); Serial.print(tempMin);
Serial.println(" °C");
        Serial.print("Max Temp      : "); Serial.print(tempMax);
Serial.println(" °C");
        Serial.print("Pressure     : "); Serial.print(pressure);
Serial.println(" hPa");
        Serial.print("Humidity     : "); Serial.print(humidity);
Serial.println(" %");
        Serial.print("Wind Speed    : "); Serial.print(windSpeed);
Serial.println(" m/s");
        Serial.print("Wind Degree   : "); Serial.print(windDeg);
Serial.println(" °");

// Konversi sunrise & sunset dari epoch ke jam:menit:detik
time_t rawSunrise = sunrise;
struct tm * sunriseTime = gmtime(&rawSunrise);
Serial.print("Sunrise      : ");
Serial.printf("%02d:%02d:%02d\n", sunriseTime->tm_hour,
sunriseTime->tm_min, sunriseTime->tm_sec);

time_t rawSunset = sunset;
struct tm * sunsetTime = gmtime(&rawSunset);
Serial.print("Sunset        : ");
Serial.printf("%02d:%02d:%02d\n", sunsetTime->tm_hour,
sunsetTime->tm_min, sunsetTime->tm_sec);

Serial.println("=====");

// OLED Tampilan sederhana
display.setTextSize(1);
display.setCursor(0,0);
display.print("City:");
display.print(cityName);
display.setCursor(0,10);
display.print("Temp:");
display.print(temp);
display.print("C");
display.setCursor(0,20);
display.print("Humid:");
display.print(humidity);
display.print("%");
display.setCursor(0,30);
display.print("Weather:");

```



```
display.print(weatherMain);
display.display();

// LED & Buzzer Logic
digitalWrite(LED_HOT, LOW);
digitalWrite(LED_COLD, LOW);
digitalWrite(LED_NORMAL, LOW);
digitalWrite(BUZZER_PIN, LOW);

if (temp >= 30.0) {
    digitalWrite(LED_HOT, HIGH);
} else if (temp <= 20.0) {
    digitalWrite(LED_COLD, HIGH);
} else {
    digitalWrite(LED_NORMAL, HIGH);
}

if (weatherDesc.indexOf("rain") >= 0 ||
weatherDesc.indexOf("storm") >= 0) {
    tone(BUZZER_PIN, 1000);
    delay(500);
    noTone(BUZZER_PIN);
}

} else {
    Serial.println("Failed to parse JSON");
}

} else {
    Serial.println("Error on HTTP request");
}

http.end();
}

delay(60000); // Update per 1 menit
}
```

## Code Program diagram.json

```
{  
  
  "version": 1,  
  
  "author": "Fredlina",  
  
  "editor": "wokwi",  
  
  "parts": [  
  
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 0, "left":  
0, "attrs": {} },  
  
    {  
  
      "type": "board-ssd1306",  
  
      "id": "oled1",  
  
      "top": 156.74,  
  
      "left": 211.43,  
  
      "attrs": { "i2cAddress": "0x3c" }  
  
    },  
  
    { "type": "wokwi-dht22", "id": "dht1", "top": -28.5, "left": 186.6,  
"attrs": {} },  
  
    {  
  
      "type": "wokwi-led",  
  
      "id": "led1",  
  
      "top": 102,  
  
      "left": -140.2,  
  
      "attrs": { "color": "cyan" }  
  
    },  
  
    {  
  
      "type": "wokwi-led",
```

```
    "id": "led2",

    "top": 15.6,

    "left": -140.2,

    "attrs": { "color": "red" }

  },

  {

    "type": "wokwi-led",

    "id": "led3",

    "top": -3.6,

    "left": -73,

    "attrs": { "color": "yellow" }

  },

  {

    "type": "wokwi-buzzer",

    "id": "bz1",

    "top": 223.2,

    "left": 1.8,

    "attrs": { "volume": "0.1" }

  }

],

"connections": [

  [ "esp:TX0", "$serialMonitor:RX", "", [] ],

  [ "esp:RX0", "$serialMonitor:TX", "", [] ],

  [ "dht1:VCC", "esp:3V3", "red", [ "v67.2", "h-57.6", "v4.9" ] ],

  [ "dht1:GND", "esp:GND.1", "black", [ "v67.2", "h-124.8" ] ],
```

```
[ "dht1:SDA", "esp:D15", "green", [ "v57.6", "h-105.5" ] ],  
  
[ "oled1:VCC", "esp:3V3", "red", [ "v0", "h-153.45" ] ],  
  
[ "oled1:GND", "esp:GND.1", "black", [ "v-19.2", "h-144" ] ],  
  
[ "oled1:SCL", "esp:D22", "green", [ "v-124.8", "h-162.9" ] ],  
  
[ "oled1:SDA", "esp:D21", "green", [ "v-96", "h-172.73" ] ],  
  
[ "led2:A", "esp:D2", "green", [ "h76.8", "v134.4", "h-349.9" ] ],  
  
[ "led1:A", "esp:D4", "green", [ "v38.4", "h-268.8" ] ],  
  
[ "led3:A", "esp:D5", "green", [ "v96", "h-211.2" ] ],  
  
[ "led1:C", "esp:GND.1", "green", [ "v57.6", "h-258.8" ] ],  
  
[ "led3:C", "esp:GND.1", "green", [ "v144", "h-201.2" ] ],  
  
[ "led2:C", "esp:GND.2", "green", [ "v86.4", "h125.2" ] ],  
  
[ "bz1:1", "esp:D18", "green", [ "v-220.8", "h67.2" ] ],  
  
[ "bz1:2", "esp:GND.2", "green", [ "v-163.2", "h-38.8" ] ]  
  
],  
  
"dependencies": {}  
  
}
```