

# LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

## Laporan Praktikum Suhu dan Kelembaban Cuaca Menggunakan Website Open Weather App

Salma Salsabila

Fakultas Vokasi Universitas Brawijaya

Email: salmasalsabila@student.ub.ac.id

### Abstrak

Di era digital ini, informasi cuaca menjadi kebutuhan penting dalam kehidupan sehari-hari. Praktikum ini bertujuan merancang sistem sederhana untuk memantau suhu, kelembapan, dan kondisi cuaca secara real-time menggunakan mikrokontroler ESP32. Sistem ini menggabungkan pembacaan suhu lokal melalui sensor DHT11 dengan informasi cuaca global dari layanan OpenWeatherMap. Hasilnya ditampilkan langsung pada layar LCD I2C 16x2. Melalui koneksi internet, sistem mampu memperbarui data setiap satu menit secara otomatis. Dengan bantuan library ArduinoJson, data dari API diproses dan ditampilkan dengan efisien. Proyek ini berhasil menunjukkan bahwa kombinasi sensor lokal dan data daring dapat menghasilkan sistem pemantauan cuaca yang praktis dan mudah digunakan, cocok untuk lingkungan rumah maupun pendidikan.

**Kata Kunci:** ESP32, DHT11, OpenWeatherMap, LCD, IoT.

### Pendahuluan

#### Latar Belakang

Cuaca sering kali memengaruhi aktivitas kita sehari-hari—mulai dari memilih pakaian, merencanakan perjalanan, hingga menentukan waktu bercocok tanam. Sayangnya, tidak semua orang memiliki akses mudah terhadap informasi cuaca yang akurat. Melalui praktikum ini, kami mencoba menghadirkan solusi sederhana namun bermanfaat: sebuah sistem pemantauan cuaca mini yang memanfaatkan teknologi Internet of Things (IoT).

Dengan menggunakan mikrokontroler ESP32 dan sensor DHT11, sistem ini dapat membaca suhu dan kelembapan secara langsung dari lingkungan sekitar. Ditambah lagi, ESP32 juga dapat mengambil data cuaca dari situs OpenWeatherMap melalui jaringan Wi-Fi. Semua informasi tersebut lalu ditampilkan dalam layar LCD kecil berukuran 16x2, sehingga pengguna bisa melihat kondisi cuaca tanpa perlu membuka aplikasi di ponsel atau komputer.

Harapannya, proyek ini bisa menjadi contoh awal penerapan IoT di bidang pemantauan lingkungan dan memberi inspirasi bagi pengembangan sistem yang lebih canggih.

## Metodologi

### Alat dan Bahan

Untuk menyusun sistem ini, digunakan beberapa komponen elektronik dan perangkat lunak, yaitu:

- ESP32
- Sensor suhu dan kelembapan DHT11
- Layar LCD 16x2 dengan modul I2C
- Kabel jumper dan breadboard
- Platform simulasi Wokwi
- Visual Studio Code dengan PlatformIO
- Koneksi internet dan API key dari OpenWeatherMap

### Langkah-Langkah Pengerjaan

#### 1. Merancang Rangkaian di Wokwi

Kami mulai dengan membuat simulasi di situs [Wokwi](#). Komponen ESP32, sensor DHT11, dan LCD 16x2 disusun dan dihubungkan sesuai skema.

#### 2. Menghubungkan ke OpenWeatherMap

Kami mendaftar di situs OpenWeatherMap untuk mendapatkan API key. API ini akan digunakan ESP32 untuk mengambil data cuaca berdasarkan lokasi.

#### 3. Pemrograman

Program ditulis menggunakan Arduino dan PlatformIO di Visual Studio Code. Kami menggunakan library ArduinoJson untuk membaca dan mengolah data dari API.

#### 4. Simulasi dan Uji Coba

Setelah semua komponen terpasang dan program selesai, kami menjalankan simulasi di Wokwi. Jika berhasil, suhu, kelembapan, dan cuaca akan tampil di layar LCD.

#### 5. Kode Program

Kode lengkap digunakan seperti berikut:

```
6. #include <Wire.h>
7. #include <LiquidCrystal_I2C.h>
8. #include <WiFi.h>
9. #include <HTTPClient.h>
10. #include "DHT.h"
11.
12. // Wi-Fi credentials
```

```

13.const char* ssid = "Wokwi-GUEST";
14.const char* password = "";
15.
16.// OpenWeatherMap API
17.String apiKey = "ede398fcf4f2b898ba4303f4425537c0";
18.String city = "Malang";
19.String units = "metric";
20.String server = "http://api.openweathermap.org/data/2.5/weather?q=" + city
    + "&units=" + units + "&appid=" + apiKey;
21.
22.// LCD setup
23.LiquidCrystal_I2C lcd(0x27, 16, 2);
24.
25.// DHT11 setup
26.#define DHTPIN 4
27.#define DHTTYPE DHT11
28.DHT dht(DHTPIN, DHTTYPE);
29.
30.void setup() {
31.    Serial.begin(115200);
32.
33.    // LCD init
34.    lcd.init();
35.    lcd.backlight();
36.    lcd.setCursor(0, 0);
37.    lcd.print("Weather Info:");
38.    delay(1000);
39.
40.    // Wi-Fi connection
41.    WiFi.begin(ssid, password);
42.    lcd.setCursor(0, 1);
43.    lcd.print("Connecting...");
44.    while (WiFi.status() != WL_CONNECTED) {
45.        delay(1000);
46.        Serial.println("Connecting to WiFi...");
47.    }
48.
49.    lcd.clear();
50.    lcd.setCursor(0, 0);
51.    lcd.print("Connected!");
52.    delay(2000);
53.    lcd.clear();
54.
55.    // DHT11 start
56.    dht.begin();

```

```

57. delay(2000); // Tambahkan delay agar sensor siap
58.}
59.
60.void loop() {
61.  if (WiFi.status() == WL_CONNECTED) {
62.    HTTPClient http;
63.    http.begin(server);
64.    int httpCode = http.GET();
65.
66.    if (httpCode > 0) {
67.      String payload = http.getString();
68.      Serial.println(payload);
69.
70.      // Ambil suhu dari API
71.      int tempIndex = payload.indexOf("temp");
72.      String temp = payload.substring(tempIndex + 6, payload.indexOf(",",
tempIndex));
73.
74.      // Ambil deskripsi cuaca
75.      int descIndex = payload.indexOf("description");
76.      String desc = payload.substring(descIndex + 14,
payload.indexOf("\\"", descIndex + 14));
77.
78.      // Baca kelembapan dari DHT11
79.      float humidity = dht.readHumidity();
80.      if (isnan(humidity)) {
81.        delay(1000); // tunggu 1 detik
82.        humidity = dht.readHumidity(); // coba baca lagi
83.      }
84.
85.      // Tampilkan data di LCD
86.      lcd.clear();
87.      lcd.setCursor(0, 0);
88.      lcd.print("T:" + temp + "C H:");
89.
90.      lcd.setCursor(0, 1);
91.      if (desc.length() > 16) {
92.        lcd.print(desc.substring(0, 16));
93.      } else {
94.        lcd.print(desc);
95.      }
96.
97.    } else {
98.      Serial.println("Error on HTTP request");
99.      lcd.clear();

```

```

100.         lcd.setCursor(0, 0);
101.         lcd.print("HTTP Error");
102.     }
103.
104.         http.end();
105.     } else {
106.         Serial.println("WiFi disconnected");
107.         lcd.clear();
108.         lcd.setCursor(0, 0);
109.         lcd.print("WiFi lost...");
110.     }
111.
112.     delay(60000); // Update every 60 detik
113. }

```

## Hasil dan Pembahasan

Setelah semua dirancang dan dijalankan, sistem ini berhasil menampilkan suhu dan kelembapan dari sensor DHT11, serta informasi cuaca dari OpenWeatherMap secara bersamaan. LCD akan terus memperbarui informasi setiap 60 detik, sehingga pengguna dapat melihat kondisi terbaru secara real-time.

Sebagai contoh, saat suhu di Malang tercatat 29°C dan cuaca “clear sky”, informasi tersebut langsung muncul di layar LCD bersama dengan kelembapan dari sensor. Dengan kombinasi sensor lokal dan data daring, sistem ini mampu memberi gambaran yang cukup lengkap tentang cuaca saat ini.

Namun, ada beberapa hal yang masih bisa dikembangkan. LCD 16x2 memiliki keterbatasan dalam menampilkan teks panjang seperti “scattered clouds” yang kadang harus dipotong. Selain itu, jika koneksi internet terputus, data dari API tidak dapat diambil. Untuk ke depan, sistem bisa ditambahkan fitur cache data atau koneksi Bluetooth sebagai alternatif.

Meski sederhana, sistem ini sudah bisa menjadi alat bantu edukatif untuk memahami bagaimana data sensor dan API bisa bekerja bersama dalam satu proyek IoT.

## Penutup

Melalui praktikum ini, kami berhasil membangun sistem monitoring cuaca mini yang menggabungkan pembacaan suhu lokal dan data daring dalam satu alat. Dengan biaya rendah dan komponen yang mudah diperoleh, alat ini sangat cocok sebagai media pembelajaran atau prototipe awal dalam pengembangan aplikasi IoT di bidang lingkungan.

## Lampiran

- **Link Simulasi Wokwi:**  
<https://wokwi.com/projects/433584416419737601>
- **Kode Json**

```
WOKWI SAVE SHARE dht
sketch.ino diagram.json libraries.txt Library Manager
1 {
2   "version": 1,
3   "author": "salme",
4   "editor": "wokwi",
5   "parts": [
6     { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 0, "left": 0, "attrs": {} },
7     {
8       "type": "wokwi-lcd1602",
9       "id": "lcd1",
10      "top": 35.2,
11      "left": 255.2,
12      "attrs": { "pins": "i2c" }
13    }
14  ],
15  "connections": [
16    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
17    [ "esp:RX0", "$serialMonitor:TX", "", [] ],
18    [ "lcd1:5CL", "esp:D22", "green", [ "h-124.8", "v-66.9" ] ],
19    [ "lcd1:VCC", "esp:3V3", "red", [ "h-124.8", "v86.5" ] ],
20    [ "lcd1:GND", "esp:GND.1", "black", [ "h-144", "v57.6" ] ],
21    [ "lcd1:SDA", "esp:D21", "green", [ "h-134.4", "v-28.6" ] ]
22  ],
23  "dependencies": {}
24 }
```

- **Diagram Rangkaian**

Simulation

↺

⏏

⏸

🔍

⌚ 01:10.829 🔄 101%

```
{
  "temp": 25.58,
  "feels_like": 26.41,
  "temp_min": 25.58,
  "temp_max": 25.58,
  "pressure": 1013,
  "humidity": 85,
  "sea_level": 1013,
  "grnd_level": 929,
  "visibility": 10000,
  "wind": {
    "speed": 0.76,
    "deg": 71,
    "gust": 1.22
  },
  "clouds": {
    "all": 85
  },
  "dt": 1749757260,
  "sys": {
    "country": "ID",
    "sunrise": 1749767982,
    "sunset": 1749809967,
    "timezone": 25200,
    "id": 1636722,
    "name": "Malang",
    "cod": 200
  }
}
```