

# Praktik Menampilkan Informasi Cuaca pada LCD Menggunakan API

oleh

Nuha Rona Zahra

Fakultas Vokasi, Universitas Brawijaya

Email: [nuharonazz@gmail.com](mailto:nuharonazz@gmail.com)

## Abstrak

*Pada era modern ini, banyak sekali perangkat keras dibuat untuk mempermudah hidup. Pada praktik ini dilakukannya perancangan dan implementasi sistem monitoring cuaca menggunakan ESP32 yang terhubung dengan WiFi dimana data cuaca didapat dari OpenWeather API secara real-time lalu diproses dan akan ditampilkan menggunakan LCD 16x2 berbasis I2C. Sistem dari project ini akan mengakses informasi seperti suhu dan kondisi cuaca suatu kota. ESP32 berfungsi untuk pengirim permintaan HTTP dan penerima JSON dari OpenWeather. Proyek ini bertujuan untuk menunjukkan bagaimana sistem ini dapat digunakan untuk mengambil dan menunjukkan data, memperkenalkan HTTP, parsing data JSON, dan pengendalian antarmuka LCD.*

*Kata kunci: IoT, ESP32, OpenWeather API, LCD.*

## Abstract (Bahasa Inggris)

*In this modern era, a lot of hardware is made to make life easier. In this practice, the design and implementation of a weather monitoring system using ESP32 connected to WiFi where weather data is obtained from OpenWeather API in real-time and then processed and will be displayed using an I2C-based 16x2 LCD. The system of this project will access information such as temperature and weather conditions of a city. ESP32 works for both the sender of HTTP requests and the JSON receivers of OpenWeather. The project aims to demonstrate how this system can be used to retrieve and display data, introduce HTTP, JSON data parsing, and control of LCD interfaces.*

*Keyword: IoT, ESP32, DHT22, OpenWeather API, LCD*

## Pendahuluan

Internet of Things (IoT) telah menjadi teknologi yang berkembang pesat dan diterapkan dalam berbagai bidang, seperti monitoring data lingkungan. Salah satu contoh penerapan IoT dalam bidang lingkungan adalah pengambilan data cuaca dan menampilkannya menggunakan perangkat fisik.

Dalam praktik ini, dirancang sebuah sistem berbasis **ESP32** yang dapat terhubung ke jaringan WiFi dan menggunakan data yang di ambil dari OpenWeather API. Data yang diambil berisi suhu dan kondisi umum cuaca yang akan ditampilkan di LCD 16x2 berbasis I2C dan akan muncul informasi berupa “clear sky”, “broken clouds”, dan “rain”.

Dengan penggabungan pemrograman HTTP Client untuk pengambilan data serta pengolahan string JSON sederhana sebagai parsing informasi, proyek sistem ini menjadi contoh dari teknologi IoT yang mudah di akses. Praktik ini bertujuan untuk mengenalkan konsep komunikasi jaringan pada ESP32, data API, dan LCD.

## Metodologi

Metode yang digunakan dalam praktik ini mencakup beberapa tahapan utama, yaitu perancangan sistem, koneksi WiFi, pengambilan data cuaca, pengolahan data, dan penampilan data.

### 1. Perancangan Sistem

Perancangan dilakukan di perangkat keras seperti Mikrokontroler ESP32 Devkit V1 dan LCD 16x2. Sedangkan perangkat lunak yang digunakan adalah Arduino IDE dengan menggunakan library seperti WiFi.h, HTTPClient.h, Wire.h, dan LiquidCrystal\_IC2.h. Data didapat dari layanan OpenWeather API yang menyediakan data dengan format JSON.

2. **Koneksi WiFi**

ESP32 digunakan untuk terhubung ke jaringan WiFi dengan SSID dan password lalu digunakan agar ESP32 mendapatkan akses internet dan mengambil data dari API.

3. **Pengambilan Data Cuaca**

Setelah berhasil tersambung dengan WiFi, mikrokontroler akan mengirim permintaan ke HTTP GET ke server OpenWeather API menggunakan library HTTPClient dan server API akan memberikan data cuaca terkini dalam format JSON.

4. **Pengolahan Data**

Data JSON yang sudah diterima akan di proses untuk mendapatkan suhu dan deskripsi cuaca.

5. **Penampilan Data**

Data yang berupa suhu dan deskripsi cuaca yang sudah didapat akan diproses untuk ditampilkan di LCD 16x2.

### **Hasil dan Pembahasan**

Praktik ini melakukan beberapa tahap yang berupa perancangan, pemrograman, hingga pengujian sisten sehingga didapat hasil sebagai berikut:

1. Sistem dapat menghubungkan ESP32 ke WiFi menggunakan jaringan Wokwi-GUEST
2. Berhasil mendapatkan data cuaca yang didapat dari OpenWeather API
3. Informasi suhu dan deskripsi cuaca didapat dari data JSON respons server.
4. Data suhu ditampilkan pada baris pertama LCD, sedangkan deskripsi cuaca di tampilkan setelah informasi suhu

### **Pembahasan**

Berdasarkan hasil yang didapat, ESP32 mampu menjalankan sebagai perangkat yang dapat menghubungkan perangkat fisik dengan OpenWeather API. Parsing data JSON dilakukan dengan metode `indexOf()` dan `substring()` yang berguna untuk mencari posisi data yang akan diambil nilainya.

Digunakannya LCD 16x2 karena LCD 16x2 hanya membutuhkan dua pin komunikasi SCL dan SDA sehingga tidak terlalu membutuhkan pin ESP32.

### **Lampiran**

```
#include <Wire.h>
```

```
#include <LiquidCrystal_I2C.h>
```

```
#include <WiFi.h>
```

```
#include <HTTPClient.h>
```

```
// WiFi Settings
```

```
const char* ssid = "Wokwi-GUEST";
```

```
const char* password = "";
```

*// Weather API Settings*

String apiKey = "20ca0ff523294dcdeb424dfc5802e21b";

String city = "Malang";

String units = "metric";

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

*// LCD Settings*

LiquidCrystal\_I2C lcd(0x27, 16, 2);

void setup() {

Serial.begin(115200);

lcd.init();

lcd.backlight();

lcd.setCursor(0, 0);

lcd.print("Weather Info:");

WiFi.begin(ssid, password);

lcd.setCursor(0, 1);

lcd.print("Connecting...");

while (WiFi.status() != WL\_CONNECTED) {

delay(1000);

Serial.println("Connecting to WiFi...");

}

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Connected!");

delay(2000);

lcd.clear();

}

void loop() {

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 manual
    int tempIndex = payload.indexOf("temp");
    String temp = payload.substring(tempIndex + 6, payload.indexOf(",", tempIndex));

    int humIndex = payload.indexOf("humidity");
    String humidity = payload.substring(humIndex + 9, payload.indexOf(",", humIndex));

    int descIndex = payload.indexOf("description");
    String desc = payload.substring(descIndex + 14, payload.indexOf("\\"", descIndex + 14));

    // Display suhu
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Temp: " + temp + " C");
    lcd.setCursor(0, 1);
    lcd.print(desc);
    delay(5000);

    // Display kelembaban dengan scroll kalau panjang
    lcd.clear();
    String humMsg = "Humidity: " + humidity + "%";
    for (int i = 0; i <= humMsg.length() - 16; i++) {
        lcd.setCursor(0, 0);
        lcd.print(humMsg.substring(i, i + 16));
        delay(300);
    }
}

```

```

    lcd.setCursor(0, 1);

    lcd.print(desc);

    delay(5000);

} else {

    Serial.println("Error on HTTP request");

}

http.end();

}

delay(60000); // Update tiap 1 menit

}

```

