**Pemantauan Suhu dan Kelembapan Kota Bandung Menggunakan ESP32 dan Layanan API Cuaca di Wokwi**

*Ibnu Jaisyurrahman Faiz*

*Program Studi Teknologi Informasi, Fakultas Vokasi, Universitas Brawijaya*

*Email: (*[*ibnufaiz72ub@student.ub.ac.id*](mailto:ibnufaiz72ub@student.ub.ac.id)*)*

**Abstrak:** Proyek ini bertujuan untuk merancang sistem pemantauan cuaca berbasis Internet of Things (IoT) menggunakan mikrokontroler ESP32 yang disimulasikan pada platform Wokwi. Data cuaca seperti suhu, kelembapan, dan deskripsi kondisi cuaca diambil secara real-time dari OpenWeatherMap API. Informasi yang diperoleh kemudian ditampilkan melalui LCD I2C agar dapat diakses secara langsung oleh pengguna. Sistem ini dapat menjadi solusi sederhana dan efisien dalam menampilkan informasi cuaca tanpa harus menggunakan perangkat berdaya komputasi tinggi. Eksperimen ini menunjukkan bagaimana ESP32 dapat terhubung ke internet, melakukan permintaan HTTP, serta menampilkan data pada perangkat output sederhana.

Kata Kunci : *ESP32, IoT, OpenWeatherMap, LCD I2C, Wokwi, Pemantauan Cuaca, HTTP Request*

**Abstract:** This project aims to design a weather monitoring system based on the Internet of Things (IoT) using an ESP32 microcontroller simulated on the Wokwi platform. Weather data such as temperature, humidity, and weather description are retrieved in real-time from the OpenWeatherMap API. The retrieved information is displayed on an I2C LCD to provide a simple and accessible way for users to monitor current weather conditions. This system offers a low-cost and efficient solution for displaying weather data without the need for high-performance computing devices. The experiment demonstrates how the ESP32 can connect to the internet, make HTTP requests, and display the data using a basic output device.

Key Word: *ESP32, IoT, OpenWeatherMap, LCD I2C, Wokwi, Weather Monitoring, HTTP Request*

1. **Pendahuluan**
2. **Latar belakang**

Perkembangan teknologi Internet of Things (IoT) memungkinkan berbagai perangkat untuk saling terhubung dan bertukar data melalui jaringan internet. Salah satu aplikasi umum dari IoT adalah sistem pemantauan cuaca yang dapat diakses secara real-time. Biasanya, sistem seperti ini memerlukan perangkat mahal dan kompleks, namun dengan hadirnya mikrokontroler seperti ESP32, sistem pemantauan cuaca dapat dibuat dengan biaya rendah dan efisien.

OpenWeatherMap merupakan salah satu layanan penyedia data cuaca gratis yang memungkinkan pengguna untuk mengambil data cuaca melalui API. Dengan menggabungkan ESP32, layanan OpenWeatherMap, dan tampilan LCD I2C, pengguna dapat memantau kondisi cuaca seperti suhu dan kelembapan dengan mudah. Proyek ini dikembangkan dan diuji secara virtual menggunakan Wokwi, sebuah simulator online yang mendukung berbagai komponen mikrokontroler.

1. **Tujuan eksperimen**

* Mempelajari cara menghubungkan ESP32 dengan jaringan Wi-Fi untuk mengambil data secara online.
* Mengambil dan mengolah data cuaca dari layanan OpenWeatherMap menggunakan HTTP request.
* Menampilkan data suhu, kelembapan, dan deskripsi cuaca secara real-time pada LCD I2C.
* Mensimulasikan dan menguji sistem secara virtual menggunakan platform Wokwi.

1. **Metodologi**
2. **Alat dan Bahan**

Pada eksperimen ini, simulasi dilakukan menggunakan Wokwi. Berikut adalah daftar alat dan bahan yang digunakan:

* Wokwi Simulator – Untuk simulasi rangkaian
* LCD 16x2 dengan I2C Module Menampilkan data cuaca (suhu, kelembapan, deskripsi)
* Koneksi Wi-Fi Virtual Untuk mengakses API cuaca dari internet
* ESP32 Mikrokontroler utama untuk koneksi Wi-Fi dan pemrosesan data
* OpenWeatherMap API Layanan penyedia data cuaca

1. **Langkah Implementasi**

Langkah implementasi untuk membuat Pemantauan Suhu dan Kelembapan Kota Bandung Menggunakan ESP32 dan Layanan API Cuaca di Wokwi sebagai berikut:

* Buka Wokwi Simulator dan buat proyek baru
* Tambahkan komponen ESP32 dan LCD 16x2 dengan I2C
* Hubungkan LCD I2C ke pin I2C default ESP32:

SDA → GPIO 21

SCL → GPIO 22

VCC dan GND sesuai daya ESP32

* Masukkan kode program Arduino C++ yang mengimpor:

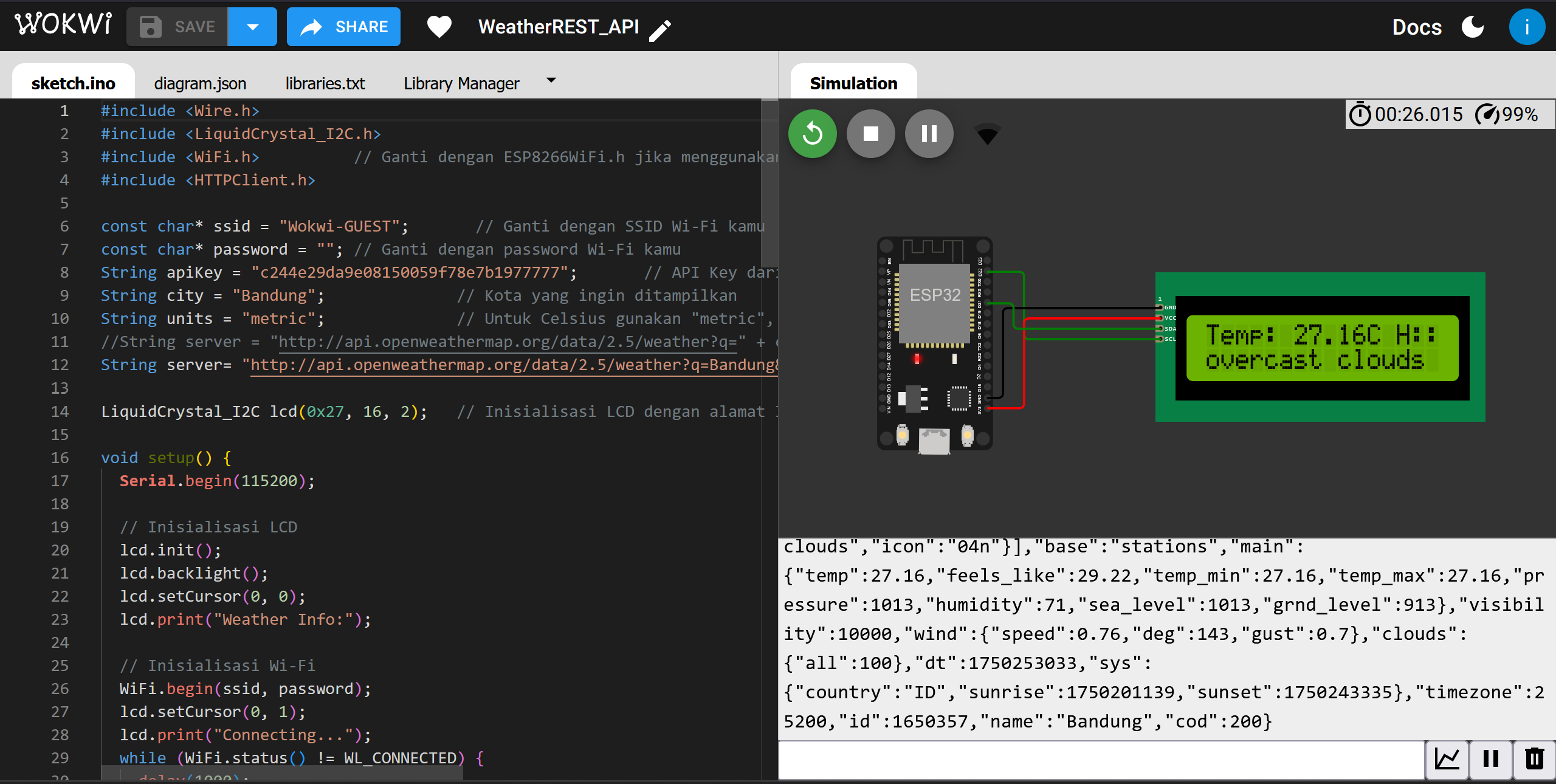
Library WiFi dan HTTPClient (untuk koneksi dan pengambilan data), LiquidCrystal\_I2C (untuk tampilan LCD), dan URL API OpenWeatherMap yang mencakup nama kota, API key, dan satuan suhu

* Kode akan menghubungkan ESP32 ke Wi-Fi, mengambil data dari server, dan menampilkan hasilnya di LCD
* Jalankan simulasi di Wokwi
* Cek Serial Monitor untuk melihat data mentah dari OpenWeatherMap
* Lihat tampilan suhu, kelembapan, dan deskripsi cuaca pada LCD

1. **Hasil dan Pembahasan**
2. **Hasil Eksperimen**

Setelah sistem dirancang dan disimulasikan menggunakan platform Wokwi, ESP32 berhasil melakukan koneksi ke jaringan Wi-Fi virtual "Wokwi-GUEST" dan mengambil data cuaca secara real-time dari OpenWeatherMap API untuk kota Bandung. Data yang berhasil ditampilkan pada LCD 16x2 meliputi:

* Suhu (temperature) dalam satuan Celcius
* Kelembapan udara (humidity) dalam persen (%)
* Deskripsi cuaca (weather description), seperti "clear sky", "few clouds", dll.



1. **Lampiran**

* Kode program main.cpp
* #include <Wire.h>
* #include <LiquidCrystal\_I2C.h>
* #include <WiFi.h>          // Ganti dengan ESP8266WiFi.h jika menggunakan ESP8266
* #include <HTTPClient.h>
* const char\* ssid = "Wokwi-GUEST";       // Ganti dengan SSID Wi-Fi kamu
* const char\* password = ""; // Ganti dengan password Wi-Fi kamu
* String apiKey = "c244e29da9e08150059f78e7b1977777";       // API Key dari OpenWeatherMap
* String city = "Bandung";              // Kota yang ingin ditampilkan
* String units = "metric";              // Untuk Celsius gunakan "metric", untuk Fahrenheit "imperial"
* //String server = "http://api.openweathermap.org/data/2.5/weather?q=" + city + "&units=" + units + "&appid=" + apiKey;
* String server= "http://api.openweathermap.org/data/2.5/weather?q=Bandung&appid=c244e29da9e08150059f78e7b1977777&units=metric";
* LiquidCrystal\_I2C lcd(0x27, 16, 2);   // Inisialisasi LCD dengan alamat I2C 0x27
* void setup() {
* **Serial**.begin(115200);
* // Inisialisasi LCD
* lcd.init();
* lcd.backlight();
* lcd.setCursor(0, 0);
* lcd.print("Weather Info:");
* // Inisialisasi Wi-Fi
* 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)) {  // Check WiFi connection status
* HTTPClient http;
* http.begin(server);     // Specify the URL
* int httpCode = http.GET();  // Make the request
* if (httpCode > 0) { // Check for the returning code
* String payload = http.getString();
* **Serial**.println(payload);   // Print the response payload
* // Parse data (extract temperature)
* int tempIndex = payload.indexOf("temp");
* String temp = payload.substring(tempIndex + 6, payload.indexOf(",", tempIndex));
* // Ambil data humidity
* int humIndex = payload.indexOf("humidity");
* String humidity = payload.substring(humIndex + 9, payload.indexOf(",", humIndex));

* // Display temperature on LCD
* lcd.setCursor(0, 0);
* lcd.print("Temp:");
* lcd.setCursor(6, 0);
* lcd.print(temp);
* // lcd.print(" C");
* lcd.print("C H:");
* lcd.print(humidity);
* lcd.print("%");
* // Extract weather description
* int descIndex = payload.indexOf("description");
* String desc = payload.substring(descIndex + 14, payload.indexOf("\"", descIndex + 14));
* // Display description on LCD
* lcd.setCursor(0, 1);
* lcd.print(desc);
* } else {
* **Serial**.println("Error on HTTP request");
* }
* http.end();  // Free the resources
* }
* delay(60000);  // Update every minute
* }
* Kode program diagram.json
* {
* "version": 1,
* "author": "VIOLA ROSA EMAWARDANI 2020",
* "editor": "wokwi",
* "parts": [
* { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 0, "left": 0, "attrs": {} },
* {
* "type": "wokwi-lcd1602",
* "id": "lcd1",
* "top": 35.2,
* "left": 255.2,
* "attrs": { "pins": "i2c" }
* }
* ],
* "connections": [
* [ "esp:TX0", "$serialMonitor:RX", "", [] ],
* [ "esp:RX0", "$serialMonitor:TX", "", [] ],
* [ "lcd1:SCL", "esp:D22", "green", [ "h-124.8", "v-66.9" ] ],
* [ "lcd1:VCC", "esp:3V3", "red", [ "h-124.8", "v86.5" ] ],
* [ "lcd1:GND", "esp:GND.1", "black", [ "h-144", "v57.6" ] ],
* [ "lcd1:SDA", "esp:D21", "green", [ "h-134.4", "v-28.6" ] ]
* ],
* "dependencies": {}
* }