

LAPORAN PRAKTIKUM MENAMPILKAN DATA CUACA REALTIME DARI OPENWEATHERMAP API KE LCD

MENGGUNAKAN ESP32

Sugeng Aldi Widodo

Fakultas Vokasi, Universitas Brawijaya

sugengaldi330@student.ub.ac.id

ABSTRAK

Praktikum ini dilakukan untuk mempelajari bagaimana sistem Internet of Things (IoT) dapat diimplementasikan untuk mengambil dan menampilkan data cuaca dari internet secara real-time menggunakan ESP32. Data cuaca diperoleh dari API OpenWeatherMap dalam format JSON dan ditampilkan pada LCD 16x2 menggunakan komunikasi I2C. Sistem ini dilengkapi tombol navigasi sehingga pengguna dapat menelusuri 10 parameter cuaca secara bergantian, di antaranya: suhu, teras, kelembaban, curah hujan, kecepatan angin, arah angin, tekanan udara, awan, serta waktu terbit dan terbenam matahari. ESP32 secara berkala mengambil data baru setiap 60 detik. Sistem diuji menggunakan simulator online Wokwi, dan hasilnya menunjukkan bahwa ESP32 mampu terhubung ke WiFi, mengambil data melalui protokol HTTP, mem-parsing JSON, dan menampilkan informasi cuaca dengan benar.

Kata kunci: Internet of Things, ESP32, LCD I2C, OpenWeather API, HTTP Client

ABSTRACT

This practicum is conducted to learn how an Internet of Things (IoT) system can be implemented to retrieve and display weather data from the internet in real-time using ESP32. Weather data is obtained from the OpenWeatherMap API in JSON format and displayed on a 16x2 LCD using I2C communication. This system is equipped with navigation buttons so that users can browse 10 weather parameters alternately, including: temperature, feel, humidity, rainfall, wind speed, wind direction, air pressure, clouds, and sunrise and sunset times. ESP32 periodically retrieves new data every 60 seconds. The system was tested using the Wokwi online simulator, and the results showed that the ESP32 was able to connect to WiFi, retrieve data via the HTTP protocol, parse JSON, and display weather information correctly.

Keywords—Internet of Things, ESP32, LCD I2C, OpenWeather API, HTTP Client

Pendahuluan

Internet of Things (IoT) adalah konsep di mana berbagai perangkat fisik saling terhubung dan mampu bertukar data melalui internet. Dalam bidang meteorologi dan lingkungan, IoT dapat dimanfaatkan untuk membuat sistem monitoring cuaca yang terhubung secara langsung ke sumber data global, seperti API cuaca online. Hal ini memungkinkan pengguna untuk memantau kondisi cuaca secara real-time, tanpa memerlukan sensor fisik lokal seperti DHT atau BMP.

Salah satu mikrokontroler yang sangat cocok digunakan dalam implementasi IoT adalah ESP32. Dengan fitur konektivitas WiFi bawaan, serta performa prosesor yang tinggi, ESP32 dapat menjalankan berbagai operasi jaringan termasuk permintaan HTTP ke server API. Dengan menggabungkan ESP32, layanan OpenWeatherMap API, dan LCD sebagai display, maka dapat dibuat sebuah perangkat kecil namun informatif yang mampu menampilkan kondisi cuaca dari kota manapun di dunia.

Tujuan Praktikum

Praktikum ini memiliki tujuan:

1. Menerapkan komunikasi ESP32 dengan layanan internet melalui HTTP.
2. Mengambil dan membaca data JSON dari OpenWeatherMap API.
3. Menampilkan parameter cuaca secara real-time pada LCD 16x2.
4. Menyediakan fitur navigasi antar parameter menggunakan tombol.
5. Melakukan simulasi sistem secara virtual melalui platform Wokwi.
6. Memberikan pemahaman praktis mengenai integrasi perangkat keras dan layanan cloud API dalam sistem IoT.

Metodologi

1. Alat dan Bahan Virtual

No.	Nama Alat / Bahan	Fungsi
1	ESP32	Mikrokontroler utama, mengakses API & menampilkan data
2	LCD 16x2 I2C	Menampilkan informasi parameter cuaca
3	Push Button (2 buah)	Untuk navigasi antar parameter
4	Breadboard & Jumper Wire	Simulasi rangkaian
5	Wokwi.com	Platform simulasi perangkat IoT secara online
6	Arduino IDE	Menulis dan meng-upload kode
7	OpenWeatherMap API	Sumber data cuaca global real-time

Library yang digunakan:

- WiFi.h – untuk koneksi internet
- HTTPClient.h – untuk HTTP GET
- ArduinoJson.h – parsing data JSON
- LiquidCrystal_I2C.h – komunikasi dengan LCD 16x2

2.2 Implementation Steps

Berikut langkah-langkah pembuatan sistem:

Langkah 1: Mendesain Sistem di Wokwi

- Buka situs <https://wokwi.com>
- Klik “New Project”, pilih ESP32 Project
- Tambahkan komponen: ESP32, LCD 16x2 (I2C), dua tombol
- Sambungkan:
 - a) LCD SDA → GPIO 21
 - b) LCD SCL → GPIO 22
 - c) Tombol Next → GPIO 12
 - d) Tombol Prev → GPIO 14
- Gunakan diagram.json untuk menyusun komponen
- Tambahkan library di libraries.txt

Langkah 2: Menulis Program di Wokwi

- Buka tab sketch.ino
- Tulis program yang melakukan:
 - a) Inisialisasi WiFi dan LCD
 - b) Request data ke API
 - c) Parsing JSON dan menyimpan nilai
 - d) Menampilkan data cuaca
 - e) Navigasi menggunakan tombol

Langkah 3: Koneksi dan API

- Sambungkan ESP32 ke WiFi: Wokwi-GUEST
- Gunakan API berikut:

http://api.openweathermap.org/data/2.5/weather?q=Nganjuk&units=metric&appid=API_KEY

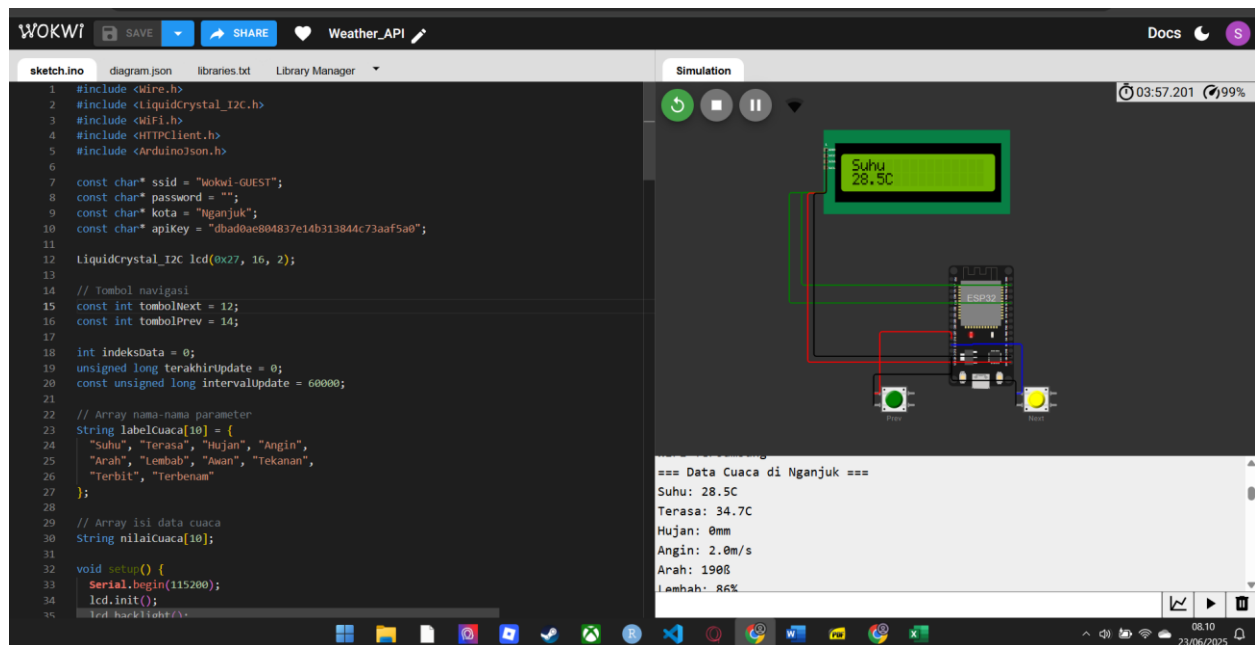
- Ambil parameter seperti:
 - a) temp, feels_like, humidity, pressure
 - b) wind.speed, wind.deg, rain.1h
 - c) sys.sunrise, sys.sunset, clouds.all

Langkah 4: Navigasi dan Auto Update

- Navigasi antar data menggunakan tombol
- Setiap 60 detik, data akan diperbarui otomatis menggunakan millis()
- Data yang ditampilkan disimpan dalam array nilaiCuaca[] dengan label labelCuaca[]

Hasil dan Pembahasan

Data Cuaca Kota Nganjuk (Output Serial Monitor):



Implementasi Kode:

sketch.ino

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>

```

```

const char* ssid = "Wokwi-GUEST";
const char* password = "";
const char* kota = "Nganjuk";
const char* apiKey = "dbad0ae804837e14b313844c73aaf5a0";

```

```

LiquidCrystal_I2C lcd(0x27, 16, 2);

```

```

// Tombol navigasi
const int tombolNext = 12;
const int tombolPrev = 14;

```

```

int indeksData = 0;
unsigned long terakhirUpdate = 0;
const unsigned long intervalUpdate = 60000;

```

```

// Array nama-nama parameter
String labelCuaca[10] = {
    "Suhu", "Terasa", "Hujan", "Angin",
    "Arah", "Lembab", "Awan", "Tekanan",
    "Terbit", "Terbenam"
};

// Array isi data cuaca
String nilaiCuaca[10];

void setup() {
    Serial.begin(115200);
    lcd.init();
    lcd.backlight();

    pinMode(tombolNext, INPUT_PULLUP);
    pinMode(tombolPrev, INPUT_PULLUP);

    lcd.setCursor(0, 0);
    lcd.print("Menghubungkan...");
    Serial.println("Mencoba menyambung ke WiFi...");

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

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Tersambung WiFi");
    Serial.println("\nWiFi Tersambung");
    delay(1000);
    lcd.clear();

    ambilCuaca();
    tampilkanData();
}

void loop() {

```

```

// Pembaruan otomatis data setiap 1 menit
if (millis() - terakhirUpdate > intervalUpdate) {
    terakhirUpdate = millis();
    ambilCuaca();
    tampilkanData(); // tampilkan setelah update
}

// Navigasi tombol
if (digitalRead(tombolNext) == LOW) {
    Serial.println("Tombol Next ditekan");
    indeksData = (indeksData + 1) % 10;
    tampilkanData();
    delay(300);
}
if (digitalRead(tombolPrev) == LOW) {
    Serial.println("Tombol Prev ditekan");
    indeksData = (indeksData - 1 + 10) % 10;
    tampilkanData();
    delay(300);
}
}

void ambilCuaca() {
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        String url = "http://api.openweathermap.org/data/2.5/weather?q=" + String(kota) +
"&units=metric&appid=" + apiKey;
        http.begin(url);
        int kode = http.GET();

        if (kode > 0) {
            String respon = http.getString();
            StaticJsonDocument<1024> data;
            DeserializationError err = deserializeJson(data, respon);

            if (!err) {
                JsonObject main = data["main"];
                JsonObject wind = data["wind"];
                JsonObject sys = data["sys"];
                JsonObject clouds = data["clouds"];
                JsonObject rain = data["rain"];
            }
        }
    }
}

```

```

    nilaiCuaca[0] = String(main["temp"].as<float>(), 1) + "C";
    nilaiCuaca[1] = String(main["feels_like"].as<float>(), 1) + "C";
    nilaiCuaca[2] = rain.containsKey("1h") ? String(rain["1h"].as<float>(), 1) + "mm" :
"0mm";
    nilaiCuaca[3] = String(wind["speed"].as<float>(), 1) + "m/s";
    nilaiCuaca[4] = String(wind["deg"].as<int>()) + (char)223;
    nilaiCuaca[5] = String(main["humidity"].as<int>()) + "%";
    nilaiCuaca[6] = String(clouds["all"].as<int>()) + "%";
    nilaiCuaca[7] = String(main["pressure"].as<int>()) + "hPa";
    nilaiCuaca[8] = formatWaktu(sys["sunrise"]);
    nilaiCuaca[9] = formatWaktu(sys["sunset"]);

```

```

// Tampilkan ke Serial Monitor semua data
Serial.println("=== Data Cuaca di " + String(kota) + " ===");
for (int i = 0; i < 10; i++) {
    Serial.print(labelCuaca[i]);
    Serial.print(": ");
    Serial.println(nilaiCuaca[i]);
}
Serial.println("=====");
} else {
    Serial.println("Gagal parsing JSON");
}
} else {
    Serial.println("Gagal HTTP, kode: " + String(kode));
}
http.end();
} else {
    Serial.println("WiFi tidak terhubung");
}
}

```

```

void tampilkanData() {
    lcd.clear();
    lcd.setCursor(0, 0); lcd.print(labelCuaca[indeksData]);
    lcd.setCursor(0, 1); lcd.print(nilaiCuaca[indeksData]);

    // Serial Monitor Output
    Serial.print("[LCD] ");
    Serial.print(labelCuaca[indeksData]);
}

```

```

Serial.print(": ");
Serial.println(nilaiCuaca[indeksData]);
}

String formatWaktu(unsigned long unixTime) {
    unixTime += 7 * 3600; // UTC+7
    int jam = (unixTime % 86400L) / 3600;
    int menit = (unixTime % 3600) / 60;
    char waktu[6];
    sprintf(waktu, "%02d:%02d", jam, menit);
    return String(waktu);
}

```

Diagram.json

```

{
  "version": 1,
  "author": "Aldi",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 50, "left": 150, "attrs": {} },
    {
      "type": "wokwi-lcd1602",
      "id": "lcd1",
      "top": -166.4,
      "left": -52,
      "attrs": { "pins": "i2c" }
    },
    {
      "type": "wokwi-pushbutton",
      "id": "btn1",
      "top": 246.2,
      "left": 28.8,
      "attrs": { "color": "green", "xray": "1", "label": "Prev" }
    },
    {
      "type": "wokwi-pushbutton",
      "id": "btn2",
      "top": 246.2,
      "left": 259.2,

```



```

    "attrs": { "color": "yellow", "xray": "1", "label": "Next" }
  }
],
"connections": [
  [ "esp:TX0", "$serialMonitor:RX", "", [] ],
  [ "esp:RX0", "$serialMonitor:TX", "", [] ],
  [ "lcd1:SCL", "esp:D22", "green", [ "v40", "h-40", "v90" ] ],
  [ "lcd1:SDA", "esp:D21", "green", [ "v50", "h-60", "v80" ] ],
  [ "lcd1:VCC", "esp:3V3", "red", [ "v60", "h-30", "v70" ] ],
  [ "lcd1:GND", "esp:GND.1", "black", [ "v70", "h-20", "v60" ] ],
  [ "btn1:1.l", "esp:D14", "red", [ "h10", "v-100", "h120" ] ],
  [ "btn1:2.l", "esp:GND.2", "black", [ "h0", "v-50", "h150" ] ],
  [ "btn2:1.l", "esp:D12", "blue", [ "h10", "v-80", "h-80", "v-80" ] ],
  [ "btn2:2.l", "esp:GND.2", "black", [ "h0", "v-40", "h-110" ] ]
],
"dependencies": {}
}

```

libraries.txt

Wokwi Library List

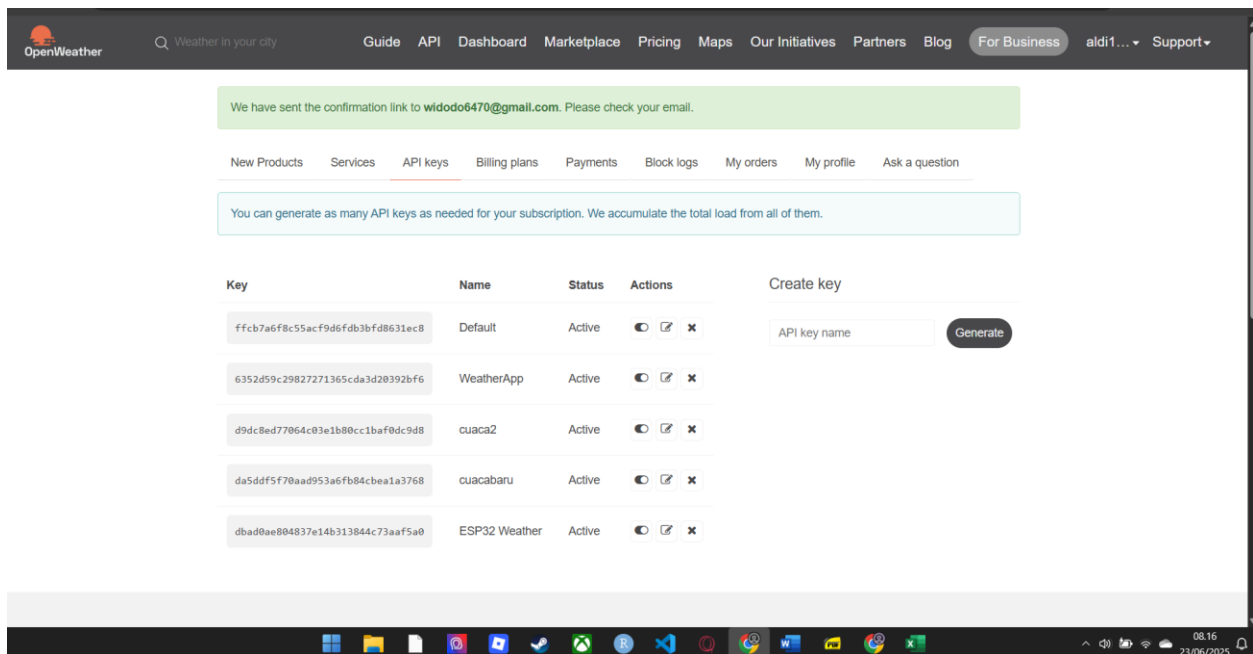
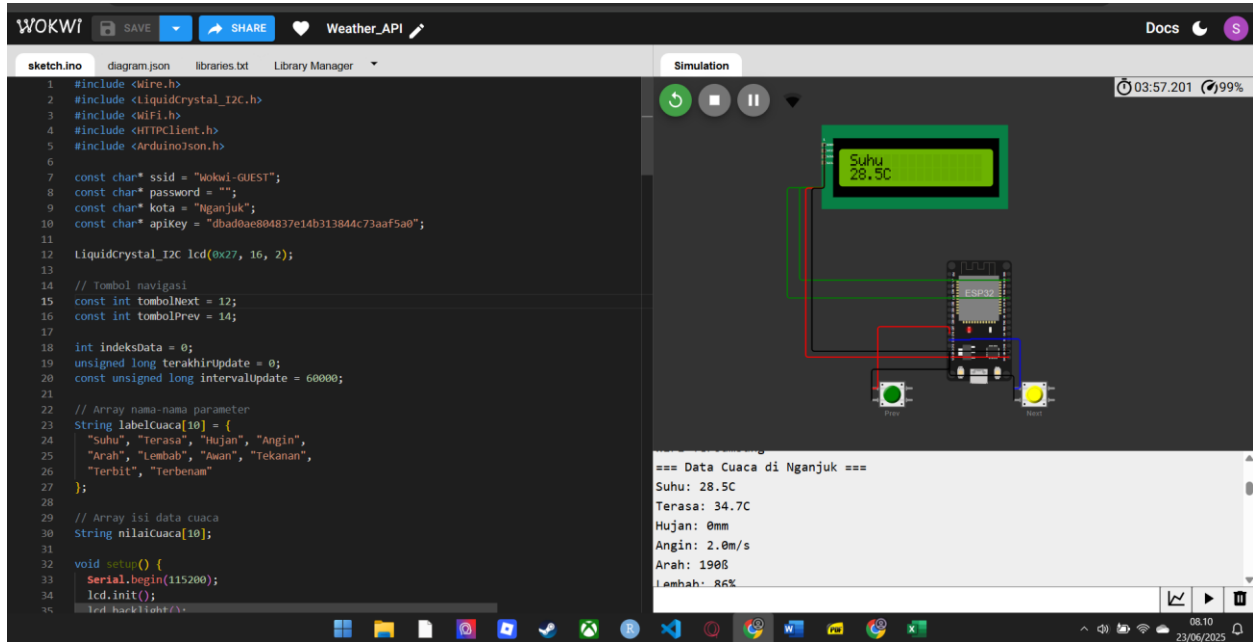
See <https://docs.wokwi.com/guides/libraries>

Automatically added based on includes:

LiquidCrystal I2C

ArduinoJson

Lampiran:



Link Wokwi :

<https://wokwi.com/projects/434500876822265857>