Laporan Praktikum Pembuatan Rangkaian Pengukuran Suhu, Kelembapan, Dan Cuaca Menggunakan Website Open Weather App

Khayru Rifaa Permana
Fakultas Vokasi , Universitas Brawijaya
Email : Khayrurifaa@student.ub.ac.id

Abstrak :
Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem monitoring informasi cuaca berbasis ESP32, yang menampilkan data suhu dan kondisi cuaca secara real-time melalui integrasi antara sensor fisik dan API daring. Perangkat keras yang digunakan meliputi mikrokontroler ESP32, sensor suhu dan kelembapan DHT11, serta layar LCD I2C 16x2. Sistem ini menggabungkan data suhu lokal dari DHT11 dan data cuaca global dari layanan OpenWeatherMap API melalui koneksi Wi-Fi. Pengambilan data dilakukan secara periodik setiap 60 detik. Data suhu dan deskripsi cuaca ditampilkan langsung pada layar LCD, memungkinkan pengguna untuk memantau kondisi secara cepat tanpa perlu akses ke perangkat lain. Parsing data dari API dilakukan menggunakan library ArduinoJson, guna memastikan pembacaan informasi dari format JSON lebih akurat dan efisien.Hasil implementasi menunjukkan bahwa sistem mampu berfungsi dengan baik dalam menampilkan informasi cuaca dan kelembapan secara real-time, serta dapat dijadikan prototipe awal untuk pengembangan sistem IoT monitoring cuaca yang lebih kompleks.

Kata Kunci : *Arduino, ESP32, Wokwi, open weather app , DHT22, Internet Of Things.*

Abstract:
This study aims to design and implement a weather monitoring system based on the ESP32 microcontroller, capable of displaying real-time temperature and weather condition data by integrating both physical sensors and online APIs. The hardware components used include the ESP32 board, a DHT11 temperature and humidity sensor, and a 16x2 I2C LCD display. The system combines local temperature data from the DHT11 sensor with global weather information obtained from the OpenWeatherMap API via a Wi-Fi connection. Data is collected periodically every 60 seconds. The temperature and weather description are displayed on the LCD, allowing users to easily monitor environmental conditions without needing additional devices. The ArduinoJson library is utilized for parsing JSON-formatted data from the API, ensuring accurate and efficient data extraction.The implementation results show that the system performs effectively in displaying real-time weather and humidity information. This prototype demonstrates potential for further development into a more advanced IoT-based weather monitoring solution.
Key Word : *Arduino, ESP32, Wokwi, Open Weather App , Internet Of Things.*

PENDAHULUAN

Perkembangan teknologi Internet of Things (IoT) semakin pesat dan telah memberikan dampak signifikan dalam berbagai bidang, termasuk sistem monitoring cuaca. Informasi cuaca yang akurat dan real-time sangat penting untuk berbagai aktivitas, baik di bidang pertanian, perikanan, transportasi, hingga kehidupan sehari-hari. Namun, tidak semua masyarakat memiliki akses terhadap perangkat atau aplikasi canggih untuk mendapatkan informasi tersebut secara cepat dan mudah.

Dalam hal ini, mikrokontroler ESP32 menjadi solusi alternatif yang efisien dan terjangkau untuk membangun sistem monitoring berbasis IoT. Dengan dukungan koneksi Wi-Fi dan kemampuan pemrosesan yang mumpuni, ESP32 dapat digunakan untuk mengakses data cuaca secara daring melalui API publik seperti OpenWeatherMap, serta menggabungkannya dengan data dari sensor fisik seperti DHT11.

Pada proyek ini, dirancang sebuah prototipe sistem monitoring cuaca sederhana yang mampu menampilkan suhu lokal, kelembapan, dan deskripsi cuaca secara real-time menggunakan ESP32, sensor DHT11, dan LCD I2C 16x2. Sistem ini diharapkan dapat menjadi solusi praktis yang dapat digunakan dalam skala rumah tangga maupun skala kecil lainnya, sekaligus sebagai dasar pengembangan sistem monitoring cuaca yang lebih kompleks di masa depan.

METODOLOGI

A. Alat dan Bahan

Mikrokontroler ESP32, Library Blynk, Sensor DHT, Kabel jumper, Breadboard, software Visual Studio Code, Website Blynk, PlatformIo, dan platform Wokwi .

B. Langkah Perancangan

1. Pilih ESP32 sebagai mikrokontroler pada situs Wokwi.
2. Tambahkan komponen pendukung untuk menyusun rangkaian yang lengkap. Pastikan seluruh komponen tersambung dengan benar ke ESP32.
3. Buka website wokwi pada browser

4. Copy script diagram.json di website wokwi.

```
{
  "version": 1,
  "author": "khayru",
  "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": {}
}
```

5. Buka website openweather pada browser dan get API keys.
6. Copy script diagram.json di website wokwi.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include "DHT.h"

// Wi-Fi credentials
const char* ssid = "Wokwi-GUEST";
const char* password = "";

// OpenWeatherMap API
String apiKey = "7139e220a9966e69475dd62ef063b3ff";
String city = "Malang";
String units = "metric";
String server = "http://api.openweathermap.org/data/2.5/weather?q=" + city
+ "&units=" + units + "&appid=" + apiKey;

// LCD setup
LiquidCrystal_I2C lcd(0x27, 16, 2);

// DHT11 setup
```

```
#define DHTPIN 4
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

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

  // LCD init
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Weather Info:");
  delay(1000);

  // Wi-Fi connection
  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();

  // DHT11 start
  dht.begin();
  delay(2000); // Tambahkan delay agar sensor siap
}

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);

      // Ambil suhu dari API
      int tempIndex = payload.indexOf("temp");
      String temp = payload.substring(tempIndex + 6, payload.indexOf(",",
tempIndex));
```

```
      // Ambil deskripsi cuaca
      int descIndex = payload.indexOf("description");
      String desc = payload.substring(descIndex + 14,
payload.indexOf("\"", descIndex + 14));

      // Baca kelembapan dari DHT11
      float humidity = dht.readHumidity();
      if (isnan(humidity)) {
        delay(1000);  // tunggu 1 detik
        humidity = dht.readHumidity(); // coba baca lagi
      }

      // Tampilkan data di LCD
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("T:" + temp + "C H:");

      lcd.setCursor(0, 1);
      if (desc.length() > 16) {
        lcd.print(desc.substring(0, 16));
      } else {
        lcd.print(desc);
      }

    } else {
      Serial.println("Error on HTTP request");
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("HTTP Error");
    }

    http.end();
  } else {
    Serial.println("WiFi disconnected");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("WiFi lost...");
  }

  delay(60000);  // Update every 60 detik
}
```

7. Jalankan simulasi dan jika tidak dapat dirunning maka copy semua kode yang telah ditulis dan masukkan ke dalam VSCode Arduino.

HASIL DAN PEMBAHASAN

Setelah melalui proses perancangan perangkat keras dan pemrograman perangkat lunak, sistem monitoring cuaca berbasis ESP32 berhasil diimplementasikan dengan baik. Sistem ini terdiri dari beberapa komponen utama, yaitu mikrokontroler ESP32, sensor suhu dan kelembapan DHT11, serta LCD I2C 16x2 sebagai media tampilan informasi. Ketika sistem pertama kali dinyalakan, ESP32 akan langsung mencoba terhubung ke jaringan Wi-Fi yang telah ditentukan. Setelah koneksi berhasil, sistem akan melakukan permintaan data ke API OpenWeatherMap untuk memperoleh data cuaca berdasarkan lokasi yang telah ditentukan, dalam hal ini adalah kota Malang. Secara bersamaan, sensor DHT11 juga akan membaca suhu dan kelembapan udara dari lingkungan sekitar perangkat.

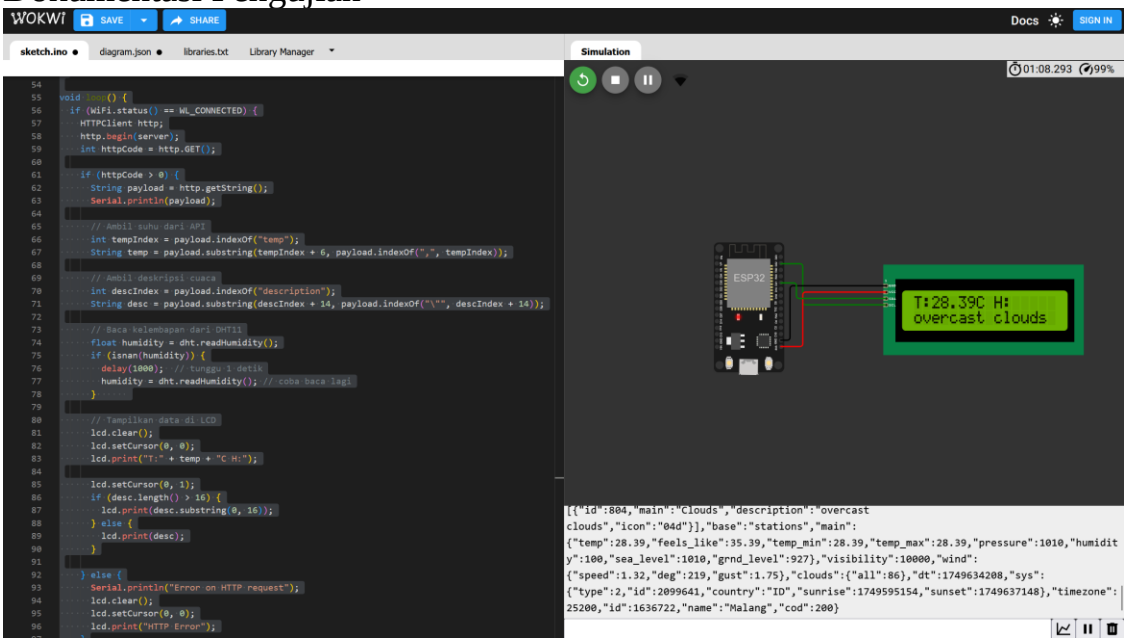
Informasi yang ditampilkan pada layar LCD terbagi menjadi dua baris. Baris pertama menampilkan suhu (yang diambil dari API) dan kelembapan (yang dibaca dari DHT11), sedangkan baris kedua menampilkan deskripsi cuaca, seperti "light rain" atau "clear sky". Proses pembaruan data dilakukan secara otomatis setiap 60 detik, sehingga informasi yang ditampilkan tetap up-to-date. Berdasarkan hasil pengujian di lingkungan nyata, tampilan pada LCD menunjukkan informasi yang dapat dibaca dengan jelas, serta sistem mampu mempertahankan koneksi Wi-Fi dengan stabil selama pengoperasian.

Secara umum, integrasi antara data lokal dan data daring dapat berjalan dengan baik. ESP32 mampu mengambil data cuaca secara real-time dari OpenWeatherMap menggunakan metode HTTP GET, kemudian memproses data JSON untuk mengekstrak nilai suhu dan deskripsi cuaca. Library ArduinoJson digunakan untuk mempermudah parsing data. Di sisi lain, sensor DHT11 memberikan data kelembapan dengan cukup baik meskipun dengan tingkat akurasi yang masih terbatas. Untuk kebutuhan monitoring dasar, sensor ini sudah memadai.

Namun demikian, terdapat beberapa keterbatasan dalam sistem ini. Salah satunya adalah keterbatasan jumlah karakter pada layar LCD, sehingga deskripsi cuaca yang panjang harus dipotong agar dapat ditampilkan dalam satu baris. Selain itu, sistem belum dilengkapi dengan mekanisme pemulihan otomatis jika koneksi Wi-Fi terputus di tengah pengambilan data. Di sisi sensor, DHT11 tidak cocok untuk aplikasi yang memerlukan presisi tinggi karena keterbatasan resolusi dan rentang pembacaannya. Meski begitu, secara keseluruhan sistem ini menunjukkan kinerja yang stabil dan dapat dijadikan sebagai prototipe awal untuk pengembangan sistem IoT monitoring cuaca yang lebih kompleks di masa mendatang.

LAMPIRAN

1. Dokumentasi Pengujian



2. Link Wokwi : https://wokwi.com/projects/433457215213463553