

Laporan Praktikum Internet of Things (IoT)

Membuat Rangkaian Pendeteksi Suhu

Menggunakan Wokwi dan Openweathermap

Alwulida Nur Aini Umma

Fakultas Vokasi, Universitas Brawijaya

Email : alwulidaumma@student.ub.ac.id

Abstrak

Internet of Things (IoT) adalah konsep teknologi yang memungkinkan perangkat fisik terhubung ke internet untuk saling berkomunikasi, bertukar data, dan berinteraksi secara otomatis tanpa campur tangan manusia secara langsung. Dalam praktikum ini, dilakukan perancangan dan simulasi rangkaian pendeteksi suhu berbasis IoT dengan menggunakan platform Wokwi sebagai simulator dan OpenWeatherMap sebagai sumber data suhu eksternal. Mikrokontroler ESP32 digunakan sebagai inti pengendali sistem yang diprogram menggunakan bahasa pemrograman C++ melalui editor Visual Studio Code (VSCode).

Sistem dirancang untuk mengambil data suhu secara real-time dari layanan API OpenWeatherMap dengan memanfaatkan koneksi internet yang disimulasikan melalui Wokwi. Data suhu yang diperoleh kemudian ditampilkan melalui antarmuka serial monitor atau dapat dikembangkan lebih lanjut ke media visual seperti OLED display atau dashboard berbasis IoT. Tujuan dari simulasi ini adalah untuk memahami cara kerja integrasi antara perangkat IoT dan data cloud-based serta mengevaluasi kinerja sistem dalam menampilkan data lingkungan secara akurat dan tepat waktu.

Hasil simulasi menunjukkan bahwa ESP32 mampu mengambil dan memproses data suhu dari lokasi yang telah ditentukan dengan baik. Selain itu, sistem menunjukkan stabilitas dalam pembacaan data serta fleksibilitas untuk dikembangkan ke berbagai aplikasi monitoring berbasis IoT. Praktikum ini memberikan pengalaman langsung kepada mahasiswa dalam merancang sistem IoT yang terintegrasi dengan sumber data eksternal, memperkuat pemahaman terhadap pemrograman mikrokontroler, komunikasi data, serta penerapan teknologi cloud dalam kehidupan nyata.

Kata kunci : *Internet of Things, Sensor Suhu, Wokwi, Visual Studio Code, ESP32, OpenWeatherMap, API, Pemantauan Lingkungan.*

Pendahuluan

Perkembangan teknologi Internet of Things (IoT) semakin pesat dan telah membawa dampak signifikan dalam berbagai aspek kehidupan, termasuk dalam bidang pemantauan lingkungan. IoT memungkinkan berbagai perangkat elektronik untuk saling terhubung melalui jaringan internet dan bertukar data secara otomatis tanpa campur tangan manusia secara langsung. Salah satu implementasi teknologi IoT yang umum dijumpai adalah sistem pemantauan suhu berbasis sensor dan layanan data cuaca digital.

Pemantauan suhu memiliki peranan penting dalam berbagai sektor, mulai dari pertanian, kesehatan, transportasi, hingga sistem peringatan dini bencana. Dalam konteks ini, kemampuan untuk mendapatkan data suhu secara real-time dan akurat menjadi hal yang sangat penting. Oleh karena itu, penggunaan layanan data cuaca seperti OpenWeatherMap menjadi solusi yang efisien dalam mengakses informasi suhu berdasarkan lokasi tertentu tanpa memerlukan sensor fisik.

Praktikum ini bertujuan untuk merancang dan mensimulasikan sistem pendeteksi suhu berbasis IoT menggunakan mikrokontroler ESP32 yang diprogram melalui Visual Studio Code (VSCode) dan disimulasikan menggunakan platform Wokwi. Sistem ini mengakses data suhu dari OpenWeather.

Metodologi

Dalam praktikum simulasi pembuatan rangkaian pendeteksi suhu berbasis Internet of Things (IoT) ini, digunakan beberapa alat dan bahan utama yang mendukung proses perancangan, pemrograman, serta pengujian sistem. Adapun alat dan bahan yang digunakan meliputi:

1. Laptop atau Komputer
Digunakan sebagai perangkat utama untuk mengakses platform simulasi, menulis dan mengunggah kode program, serta memantau output yang dihasilkan dari simulasi.
2. Koneksi Internet
Berperan penting dalam mengakses platform simulasi Wokwi serta menghubungkan mikrokontroler ESP32 ke layanan API OpenWeatherMap guna memperoleh data suhu secara real-time dari internet.
3. Visual Studio Code (VSCode)
Digunakan sebagai lingkungan pengembangan perangkat lunak (IDE) untuk menulis dan mengedit kode program berbasis C++ yang akan dijalankan pada mikrokontroler ESP32.
4. Wokwi Simulator
Platform simulasi online yang digunakan untuk merancang dan menguji rangkaian secara virtual tanpa memerlukan perangkat keras fisik. Dalam simulasi ini, komponen yang digunakan antara lain:
 - o ESP32 DevKit V1 sebagai mikrokontroler utama yang mendukung koneksi WiFi dan mampu mengambil data suhu dari internet.
 - o LCD 1602 dengan modul I2C sebagai media tampilan yang digunakan untuk menampilkan data suhu yang diperoleh dari OpenWeatherMap API.
 - o Koneksi virtual serial monitor juga digunakan untuk memantau proses pengambilan data dari API dan debugging selama proses simulasi.
5. OpenWeatherMap API
Merupakan layanan API berbasis web yang menyediakan data cuaca, termasuk suhu, dari berbagai lokasi di seluruh dunia. Dalam praktikum ini, ESP32 terhubung ke layanan tersebut untuk mengambil data suhu berdasarkan lokasi yang telah ditentukan.

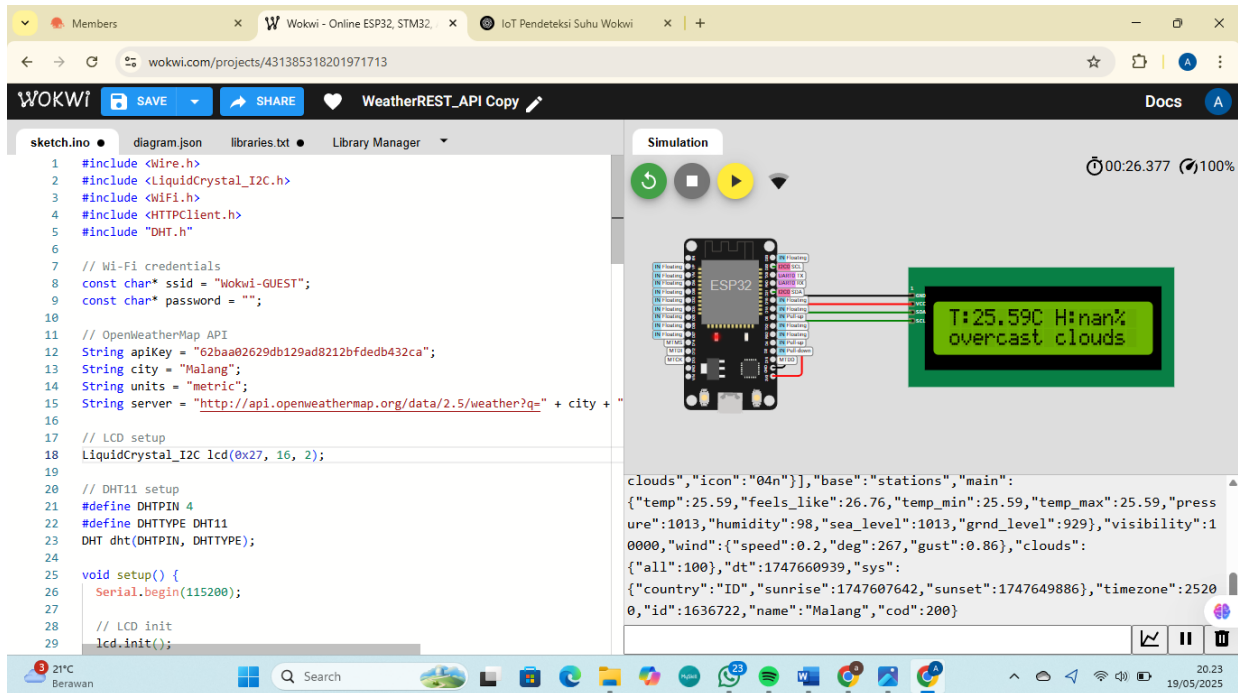
Langkah-langkah Praktikum Pembuatan dan Simulasi Rangkaian Pendeteksi Suhu menggunakan Wokwi dan Visual Studio Code (VSCode):

1. Membuat rangkaian dan simulasi pendeteksi suhu di Wokwi:
 - Buka situs Wokwi simulator dan pilih proyek baru.
 - Pilih mikrokontroler berbasis ESP32 DevKit V1 karena mendukung koneksi internet melalui WiFi.
 - Tambahkan LCD 1602 dengan modul I2C ke dalam simulasi.
 - Sambungkan pin SCL dari LCD ke pin D22 pada ESP32.
 - Sambungkan pin SDA dari LCD ke pin D21 pada ESP32.
 - Sambungkan pin VCC LCD ke pin 3V3 ESP32, dan GND LCD ke GND ESP32.
 - Pastikan juga ESP32 terhubung ke serial monitor untuk debugging.
 - Buat kode program di sketch.ino menggunakan bahasa C++ untuk:
 - a) Menghubungkan ESP32 ke jaringan WiFi.
 - b) Mengakses data suhu dari API OpenWeatherMap berdasarkan kota tertentu.
 - c) Menampilkan data suhu pada LCD 1602 serta mencetak data ke serial monitor.
 - Jalankan simulasi dengan klik "Run" untuk menguji keberhasilan koneksi dan tampilan data suhu.
2. Memindahkan proyek ke Visual Studio Code (VSCode):
 - Buka aplikasi Visual Studio Code (VSCode) dan pastikan telah menginstall ekstensi PlatformIO.
 - Buat proyek baru menggunakan PlatformIO dengan board ESP32 DevKit.
 - Pada file main.cpp yang berada di folder src, salin dan tempelkan kode program dari sketch.ino di Wokwi.
 - Buat file wokwi.toml di direktori utama proyek dan isi file tersebut dengan konfigurasi yang mengarah ke firmware.bin dan firmware.elf di folder pio/build/... menggunakan copy relative path.
 - Buat file baru bernama diagram.json, dan tempelkan isi dari file diagram Wokwi yang telah digunakan.
 - Rename file tersebut menjadi diagram.json agar VSCode dapat mengenali dan menampilkan simulasi rangkaian secara visual.
 - Build dan jalankan program menggunakan tombol Build dan Upload di PlatformIO, kemudian buka serial monitor untuk melihat output secara real-time.

Hasil dan Pembahasan

- Hasil Eksperimen

Berdasarkan hasil simulasi dari kode yang dibuat di Wokwi yang terhubung ke API Openweathermap, Esp32 di wokwi dihubungkan ke Oled yang menghasilkan temperature pada wilayah yang di tentukan.



Untuk mendapatkan hasil tersebut, dibutuhkan beberapa kode program yang kode tersebut memiliki fungsi yang berbeda-beda, seperti berikut ini :

1. Kode sketch.ino pada wokwi yang terdapat API Openweathermap :

```
#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 = "62baa02629db129ad8212bfddb432ca";
```

```
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();
}

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();

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

      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
}

```

2. Kode diagram.json pada wokwi :

```

{
  "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"
}

```

- Lampiran

1. Pada Website Wokwi yang Tidak Ada Deskripsi Cuaca

The screenshot shows a Wokwi project titled "Wokwi - Online ESP32, STM32". The project is a simulation of an ESP32 microcontroller connected to an LCD screen. The code in the sketch.ino file includes libraries for Wire, LiquidCrystal_I2C, WiFi, and HTTPClient. It sets up an ESP32 with SSID "Wokwi-GUEST" and password, and connects to an OpenWeatherMap API to fetch weather data for Malang. The simulation shows the ESP32 board connected to an LCD screen displaying "Temp: 25.58 C overcast clouds".

2. Pada Webite Wokwi yang Ada Deskripsi Cuaca

The screenshot shows a Wokwi project titled "WeatherREST_API Copy". The project is a simulation of an ESP32 microcontroller connected to an LCD screen. The code in the sketch.ino file includes libraries for Wire, LiquidCrystal_I2C, WiFi, and DHT. It sets up an ESP32 with SSID "Wokwi-GUEST" and password, and connects to an OpenWeatherMap API to fetch weather data for Malang. The simulation shows the ESP32 board connected to an LCD screen displaying "T:25.59C H:nan% overcast clouds".

Link Pengerjaan Wokwi :

<https://wokwi.com/projects/431385318201971713>