

Monitoring dan Pengendalian Suhu serta Kelembapan Berbasis ESP32 dengan Protokol MQTT

oleh

Ifadah Aulia Muhti Sinaga
233140707111034

ifadahauliasinaga@gmail.com

Abstrak

Proyek ini bertujuan untuk merancang dan mengimplementasikan sistem pemantauan suhu dan kelembapan berbasis Internet of Things (IoT). Sistem memanfaatkan mikrokontroler ESP32, sensor DHT22, dan protokol komunikasi MQTT untuk mengirimkan data suhu dan kelembapan secara berkala ke broker MQTT melalui koneksi Wi-Fi. Data tersebut dikirim menggunakan topik tertentu sehingga dapat diakses dari berbagai perangkat yang terhubung. Sistem juga dilengkapi dengan LED sebagai indikator visual yang akan menyala secara otomatis apabila suhu melebihi 30°C atau kelembapan melampaui 70%, menandakan kondisi lingkungan yang tidak normal. Simulasi dilakukan menggunakan platform Wokwi untuk menguji fungsi sistem secara virtual.

Kata kunci: IoT, ESP32, DHT22, MQTT, LED

Abstract

This project aims to design and implement an Internet of Things (IoT)-based temperature and humidity monitoring system. The system utilizes an ESP32 microcontroller, a DHT22 sensor, and the MQTT communication protocol to periodically transmit temperature and humidity data to an MQTT broker over a Wi-Fi connection. The data is published under predefined topics, enabling access from multiple connected devices. Additionally, the system features an LED as a visual indicator, which automatically lights up when the temperature exceeds 30°C or the humidity surpasses 70%, signaling abnormal environmental conditions. The simulation is conducted on the Wokwi platform, allowing virtual testing of system functionality.

Keywords: IoT, ESP32, DHT22, MQTT, LED

Pendahuluan

Internet of Things (IoT) merupakan salah satu teknologi yang berkembang pesat dan telah diterapkan secara luas dalam berbagai sektor, seperti industri, kesehatan, transportasi, dan sistem pemantauan. IoT memungkinkan perangkat elektronik untuk saling terhubung dan bertukar data secara real-time melalui jaringan internet. Salah satu penerapan praktisnya adalah sistem pemantauan suhu dan kelembapan yang sangat berguna dalam bidang pertanian, manajemen ruangan, dan kontrol lingkungan industri.

Dalam tugas ini, dirancang sebuah sistem pemantauan suhu dan kelembapan berbasis IoT dengan menggunakan ESP32 sebagai mikrokontroler utama, sensor DHT22 untuk akuisisi data, serta protokol MQTT sebagai media komunikasi data. Informasi yang diperoleh dikirim ke broker MQTT agar dapat diakses oleh perangkat lain yang terhubung ke topik yang sama. Sistem juga dilengkapi dengan indikator LED yang menyala secara otomatis ketika suhu atau kelembapan mencapai ambang batas tertentu. Seluruh sistem diuji secara virtual menggunakan platform Wokwi.

Metodologi

Metode pelaksanaan proyek ini mencakup empat tahapan utama, yaitu:

1. Perancangan Sistem

Sistem dibangun menggunakan mikrokontroler ESP32 sebagai unit pemrosesan utama, sensor DHT22 untuk membaca data suhu dan kelembapan, serta LED sebagai indikator visual.

2. Pembuatan Rangkaian Simulasi

Perakitan simulasi dilakukan di Wokwi, dengan konfigurasi sebagai berikut:

- Pin data DHT22 dihubungkan ke GPIO 15 pada ESP32.
- Anoda LED merah terhubung ke GPIO 2.
- Sensor DHT22 mendapatkan suplai dari pin 3V3 dan GND ESP32.

3. Pemrograman ESP32

Program pada ESP32 mencakup fungsi-fungsi berikut:

- Membaca data dari sensor DHT22.
- Mengirimkan data tersebut ke broker MQTT melalui topik tertentu.
- Mengontrol status LED berdasarkan data suhu dan kelembapan.
- Menerima pesan dari topik MQTT untuk pengendalian LED secara manual.

4. Pengujian Sistem

Pengujian dilakukan dengan menjalankan simulasi melalui platform Wokwi. Validasi dilakukan dengan mengamati data yang dikirim ke MQTT dan perilaku LED sesuai kondisi lingkungan yang disimulasikan.

Hasil dan Pembahasan

Hasil

Simulasi sistem berhasil dijalankan dengan baik. Mikrokontroler ESP32 mampu membaca data suhu dan kelembapan dari sensor DHT22 secara berkala, kemudian mengirimkannya ke broker MQTT sesuai dengan topik yang telah ditentukan. LED menyala secara otomatis ketika suhu melebihi 30°C atau kelembapan lebih dari 70%, sesuai dengan logika yang tertanam dalam program.

Pembahasan

Berdasarkan hasil simulasi, sistem telah berfungsi sesuai dengan tujuan perancangan. ESP32 secara konsisten dapat membaca dan mengirim data ke MQTT serta memberikan tanggapan visual melalui LED. Penggunaan indikator LED terbukti efektif sebagai bentuk peringatan terhadap kondisi lingkungan yang dianggap tidak normal. Selain pengiriman data, sistem juga mampu menerima perintah dari MQTT untuk mengontrol LED secara manual, menunjukkan keberhasilan implementasi komunikasi dua arah. Penggunaan platform Wokwi mempermudah proses simulasi dan verifikasi tanpa memerlukan perangkat keras fisik, sehingga mempercepat pengembangan sistem.

Lampiran

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

```
const int LED_RED = 2;
const int DHT_PIN = 15;
DHTesp dht;

const char* ssid = "Wokwi-GUEST";
const char* password = "";
const char* mqtt_server = "test.mosquitto.org";

WiFiClient espClient;
PubSubClient client(espClient);
unsigned long lastMsg = 0;

void setup_wifi() {
    delay(10);
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);

    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password);

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

    randomSeed(micros());

    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}
```

```

void callback(char* topic, byte* payload, unsigned int length) {
    Serial.print("Message arrived [");
    Serial.print(topic);
    Serial.print("] ");
    for (int i = 0; i < length; i++) {
        Serial.print((char)payload[i]);
    }
    Serial.println();
}

if ((char)payload[0] == '1') {
    digitalWrite(LED_RED, HIGH);
    Serial.println("LED dinyalakan lewat MQTT");
} else {
    digitalWrite(LED_RED, LOW);
    Serial.println("LED dimatikan lewat MQTT");
}
}

void reconnect() {
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        String clientId = "ESP32Client-";
        clientId += String(random(0xffff), HEX);
        if (client.connect(clientId.c_str())) {
            Serial.println("Connected");
            client.publish("IOT/Test1/mqtt", "Test IOT");
            client.subscribe("IOT/Test1/mqtt");
        } else {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 5 seconds");
            delay(5000);
        }
    }
}

```

```
}

}

void setup() {
    pinMode(LED_RED, OUTPUT);
    digitalWrite(LED_RED, LOW); // pastikan mati di awal
    Serial.begin(115200);
    setup_wifi();
    client.setServer(mqtt_server, 1883);
    client.setCallback(callback);
    dht.setup(DHT_PIN, DHTesp::DHT22);
}

void loop() {
    if (!client.connected()) {
        reconnect();
    }
    client.loop();

    unsigned long now = millis();
    if (now - lastMsg > 2000) {
        lastMsg = now;
        TempAndHumidity data = dht.getTempAndHumidity();

        float temp = data.temperature;
        float hum = data.humidity;

        Serial.print("Temperature: ");
        Serial.println(temp);
        Serial.print("Humidity: ");
        Serial.println(hum);

        String tempStr = String(temp, 2);
        client.publish("IOT/Test1/temp", tempStr.c_str());
    }
}
```

```

String humStr = String(hum, 1);

client.publish("IOT/Test1/hum", humStr.c_str());

if (isnan(temp) || isnan(hum)) {

    Serial.println("Failed to read from DHT sensor!");

    return;
}

// 🔥 Ganti ambang jadi lebih rendah agar LED pasti nyala

if (temp > 20.0 || hum > 30.0) {

    digitalWrite(LED_RED, HIGH);

    Serial.println("LED ON karena suhu atau kelembaban tinggi (test)");

} else {

    digitalWrite(LED_RED, LOW);

    Serial.println("LED OFF karena suhu & kelembaban normal");

}

}

```



