Laporan

Praktikum Internet of Things (IoT) Membuat Rangkaian Pendeteksi Temperatur dan Humidity Menggunakan Wokwi yang Terhubung Ke Mqtt

Alwulida Nur Aini Umma

Fakultas Vokasi, Universitas Brawijaya

Email : alwulidaumma@student.ub.ac.id

Abstrak

Internet of Things (IoT) merupakan konsep teknologi yang memungkinkan perangkat fisik untuk saling terhubung dan bertukar data melalui jaringan internet. Dalam praktikum ini, dilakukan perancangan dan simulasi rangkaian pendeteksi temperatur dan kelembapan berbasis IoT dengan menggunakan simulator Wokwi dan protokol komunikasi MQTT sebagai media pengiriman data. Mikrokontroler ESP32 digunakan sebagai pusat kendali sistem dan diprogram menggunakan bahasa pemrograman C++ melalui editor Visual Studio Code (VSCode).

Sensor DHT22 dimanfaatkan untuk mendeteksi data suhu dan kelembapan secara real-time, yang kemudian dikirimkan ke broker MQTT menggunakan koneksi internet yang disimulasikan pada platform Wokwi. Data yang dikirim dapat dimonitor melalui MQTT client atau dikembangkan lebih lanjut ke dalam dashboard IoT untuk visualisasi secara langsung. Tujuan utama dari simulasi ini adalah untuk memahami proses integrasi sensor, mikrokontroler, dan protokol MQTT dalam sistem monitoring lingkungan secara efisien dan real-time.

Hasil simulasi menunjukkan bahwa sistem berhasil membaca dan mengirimkan data temperatur serta kelembapan ke broker MQTT dengan stabil. Praktikum ini memberikan pemahaman mendalam kepada mahasiswa mengenai cara kerja komunikasi MQTT dalam sistem IoT, pemrograman mikrokontroler, serta pemanfaatan cloud dalam pengiriman dan pemantauan data sensor lingkungan.

Kata kunci : *Internet of Things, Temperatur, Humidity, ESP32, MQTT, Wokwi, Visual Studio Code, DHT22, Pemantauan Lingkungan.*

Pendahuluan

Perkembangan teknologi Internet of Things (IoT) telah memberikan dampak signifikan dalam berbagai aspek kehidupan, terutama dalam hal pengumpulan, pengiriman, dan pemantauan data secara otomatis dan real-time. IoT memungkinkan perangkat fisik seperti sensor, aktuator, dan mikrokontroler untuk saling terhubung melalui jaringan internet guna bertukar data tanpa intervensi langsung dari manusia. Salah satu penerapan umum dari teknologi ini adalah sistem pemantauan suhu dan kelembapan yang banyak digunakan dalam sektor pertanian, kesehatan, industri, dan rumah pintar. Dalam konteks komunikasi data IoT, Message Queuing Telemetry Transport (MQTT) menjadi salah satu protokol yang banyak digunakan karena sifatnya yang ringan, efisien, dan cocok untuk perangkat dengan sumber daya terbatas. MQTT memungkinkan pengiriman data sensor ke cloud atau aplikasi pemantauan secara cepat dan andal melalui pendekatan publish-subscribe.

Pada praktikum ini, dilakukan perancangan dan simulasi sistem pendeteksi suhu dan kelembapan menggunakan sensor DHT22 yang dikendalikan oleh mikrokontroler ESP32. Sistem ini disimulasikan menggunakan platform Wokwi, yang memungkinkan pengujian perangkat keras dan perangkat lunak secara virtual. Data hasil pembacaan sensor dikirimkan ke broker MQTT melalui koneksi internet yang juga disimulasikan di Wokwi. Visual Studio Code digunakan sebagai lingkungan pengembangan kode (IDE), dengan bahasa pemrograman C++ sebagai basis pemrogramannya. Tujuan dari praktikum ini adalah untuk memahami konsep dasar integrasi perangkat IoT

dengan protokol MQTT, serta mengembangkan kemampuan dalam merancang sistem pemantauan lingkungan yang dapat dikembangkan lebih lanjut untuk kebutuhan nyata. Melalui simulasi ini, mahasiswa diharapkan mampu mengaplikasikan teori IoT ke dalam sistem kerja yang fungsional dan efisien.

Metodologi

Dalam praktikum simulasi pembuatan rangkaian pendeteksi temperatur dan humidity yang 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. Perangkat Lunak dan Kebutuhan Sistem

a) Laptop atau Komputer
Digunakan sebagai media untuk mengakses platform Wokwi, menulis kode program, serta memantau proses simulasi dan data yang dikirimkan ke broker MQTT.

b) Koneksi Internet
Diperlukan untuk menjalankan simulasi di Wokwi serta menghubungkan ESP32 dengan broker MQTT secara online.

c) Wokwi Simulator
Platform simulasi online yang digunakan untuk merancang dan menjalankan sistem secara virtual tanpa perangkat keras fisik. Dalam simulasi ini, komponen-komponen yang digunakan meliputi:
- ESP32 DevKit V1
Sebagai mikrokontroler utama yang mengelola pembacaan data dari sensor serta koneksi ke broker MQTT melalui jaringan WiFi.
- Sensor DHT22
Digunakan untuk mendeteksi data suhu dan kelembapan lingkungan. Sensor ini terhubung ke pin digital D15 pada ESP32.
- LED Merah
Digunakan sebagai indikator sederhana, misalnya untuk menunjukkan status koneksi atau peringatan suhu/kelembapan tinggi. LED ini dikontrol melalui pin D2.
- Serial Monitor
Digunakan untuk menampilkan hasil pembacaan data sensor dan status koneksi selama proses simulasi berlangsung.

2. Konfigurasi Koneksi pada Wokwi
Berdasarkan simulasi, berikut konfigurasi hubungan antar komponen:

a) DHT22 terhubung ke ESP32:
   - VCC → 3V3
   - GND → GND
   - SDA (data) → D15

b) LED Merah:
   - Anoda (A) → D2
   - Katoda (C) → GND

c) ESP32 juga terhubung ke Serial Monitor melalui pin TX0 dan RX0 untuk keperluan debugging.

3. Pemrograman Sistem

a) Visual Studio Code (VSCode) digunakan sebagai editor kode untuk menulis program menggunakan bahasa C++ dengan library pendukung seperti DHT.h dan PubSubClient.h.

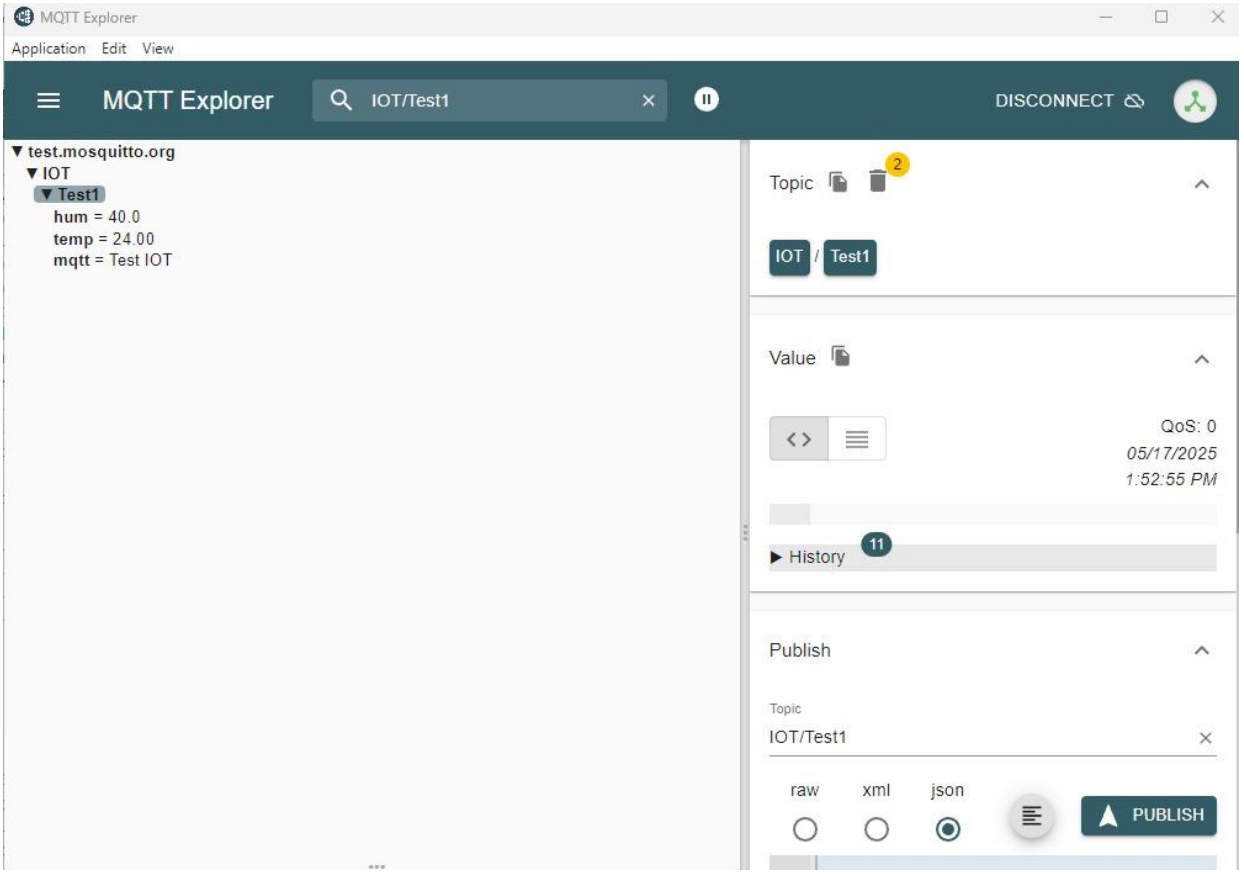
b) Program ditulis untuk:
   - Membaca data dari sensor DHT22
   - Menghubungkan ESP32 ke jaringan WiFi simulasi
   - Mengirim data suhu dan kelembapan ke broker MQTT (seperti broker gratis broker.hivemq.com atau broker lokal)

4. Pengujian dan Simulasi
- Setelah semua koneksi dan kode selesai, simulasi dijalankan di Wokwi.
- Serial monitor digunakan untuk memastikan bahwa data sensor terbaca dengan benar dan status koneksi MQTT berhasil.
- LED dapat difungsikan sebagai indikator visual (misalnya menyala saat kelembapan melebihi ambang batas).

Hasil dan Pembahasan

- Hasil Eksperimen

Berdasarkan hasil simulasi dari kode yang dibuat di Wokwi yang terhubung ke Aplikasi Mqtt, hasil dari run kode di wokwi akan muncul juga pada aplikasi Mqtt.



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 Kode yang terhubung ke Mqtt :

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

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

// Update these with values suitable for your network.

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

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

```
float hum = 0;

void setup_wifi() { //perintah koneksi wifi
  delay(10);
  // We start by connecting to a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.mode(WIFI_STA); //setting wifi chip sebagai station/client
  WiFi.begin(ssid, password); //koneksi ke jaringan wifi

  while (WiFi.status() != WL_CONNECTED) { //perintah tunggu esp32 sampi terkoneksi ke wifi
    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) { //perintah untuk menampilkan data ketika
esp32 di setting sebagai subscriber
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++) { //mengecek jumlah data yang ada di topik mqtt
    Serial.print((char)payload[i]);
  }
  Serial.println();

  // Switch on the LED if an 1 was received as first character
  if ((char)payload[0] == '1') {
    digitalWrite(LED_RED, HIGH);   // Turn the LED on
  } else {
    digitalWrite(LED_RED, LOW);  // Turn the LED off
  }
}

void reconnect() { //perintah koneksi esp32 ke mqtt broker baik itu sebagai publusher atau subscriber
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // perintah membuat client id agar mqtt broker mengenali board yang kita gunakan
    String clientId = "ESP32Client-";
    clientId += String(random(0xffff), HEX);
    // Attempt to connect
    if (client.connect(clientId.c_str())) {
      Serial.println("Connected");
      // Once connected, publish an announcement...
      client.publish("IOT/Test1/mqtt", "Test IOT"); //perintah publish data ke alamat topik yang di setting
      // ... and resubscribe
      client.subscribe("IOT/Test1/mqtt"); //perintah subscribe data ke mqtt broker
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
```

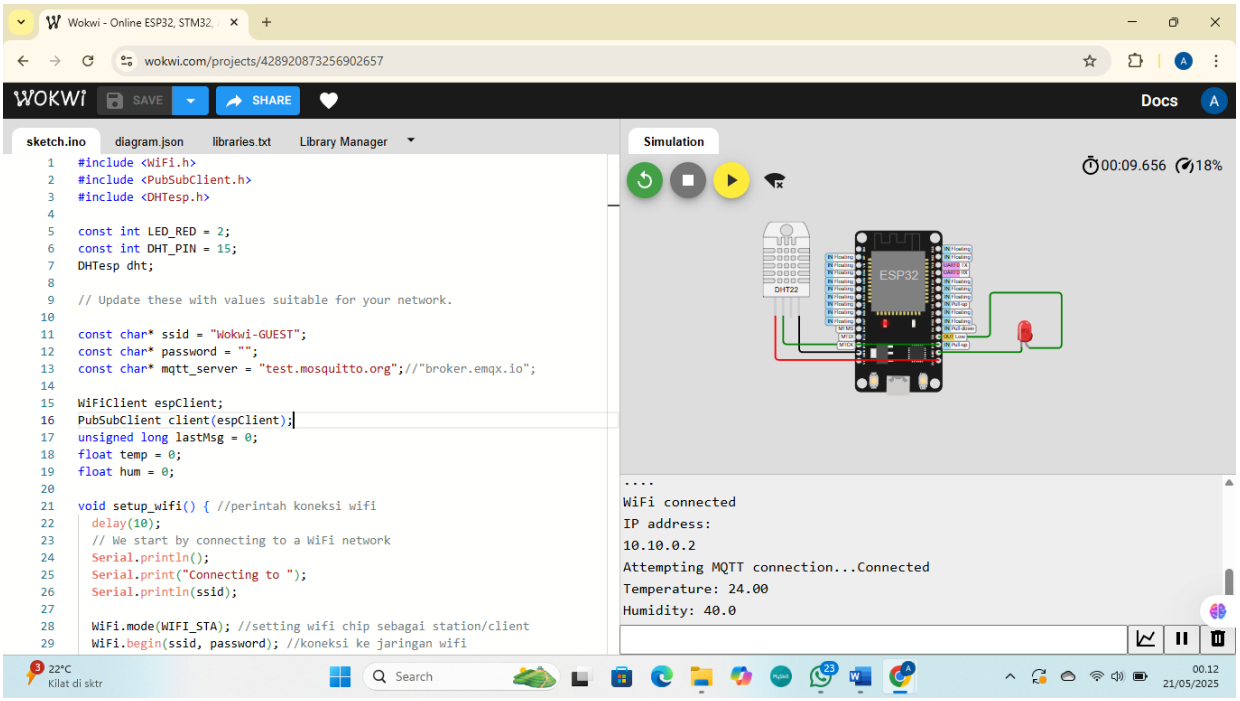
```
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}

void setup() {
  pinMode(LED_RED, OUTPUT);     // inisialisasi pin 2 / ledbuiltin sebagai output
  Serial.begin(115200);
  setup_wifi(); //memanggil void setup_wifi untuk dieksekusi
  client.setServer(mqtt_server, 1883); //perintah connecting / koneksi awal ke broker
  client.setCallback(callback); //perintah menghubungkan ke mqtt broker untuk subscribe data
  dht.setup(DHT_PIN, DHTesp::DHT22);//inisialiasi komunikasi dengan sensor dht22
}

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

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

    String temp = String(data.temperature, 2); //membuat variabel temp untuk di publish ke broker mqtt
    client.publish("IOT/Test1/temp", temp.c_str()); //publish data dari varibel temp ke broker mqtt

    String hum = String(data.humidity, 1); //membuat variabel hum untuk di publish ke broker mqtt
    client.publish("IOT/Test1/hum", hum.c_str()); //publish data dari varibel hum ke broker mqtt

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

2. Kode diagram.json pada wokwi :

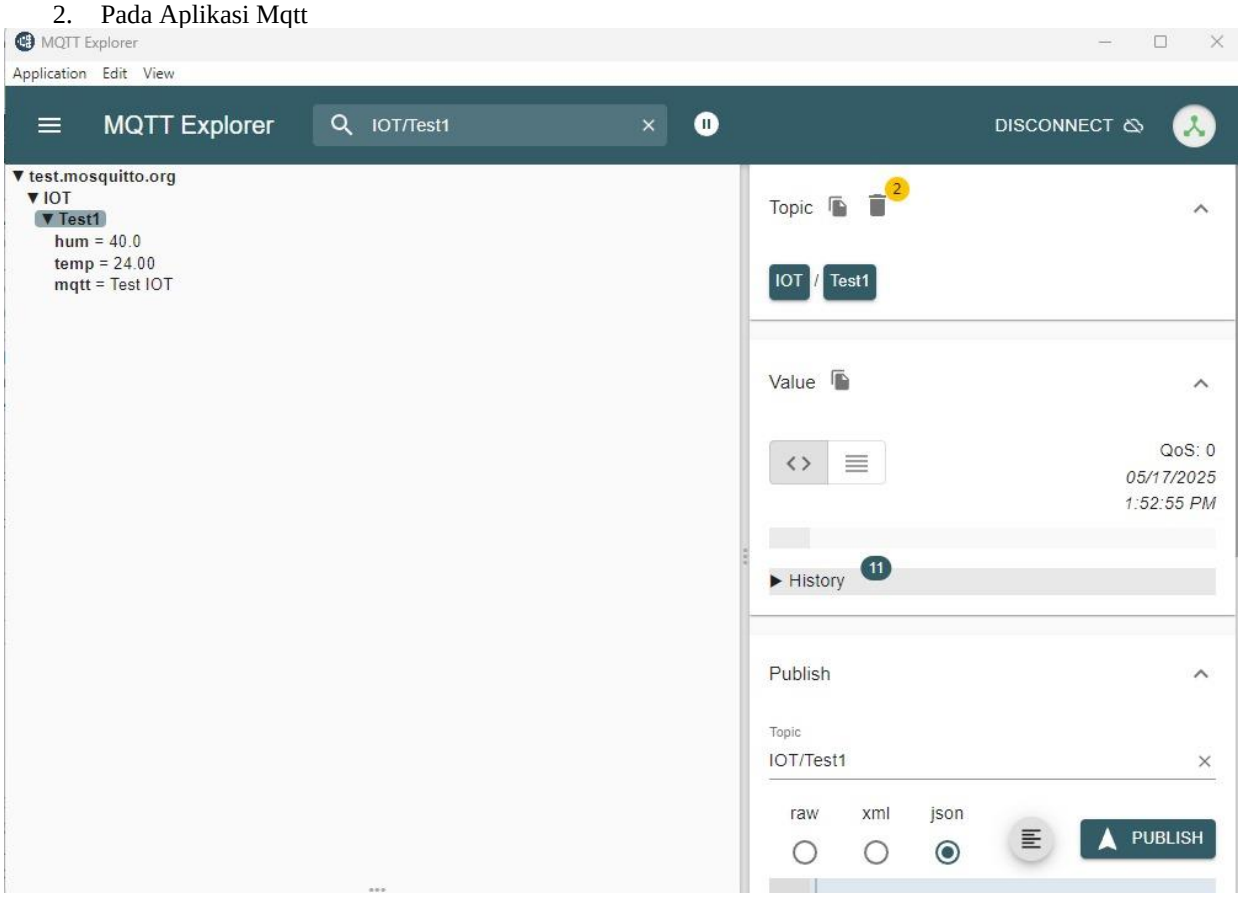
```
{
  "version": 1,
  "author": "Subairi",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 0, "left": 0, "attrs": {} },
    { "type": "wokwi-dht22", "id": "dht1", "top": -9.3, "left": -111, "attrs": {} },
    { "type": "wokwi-led", "id": "led1", "top": 102, "left": 186.2, "attrs": { "color": "red" } }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [] ],
    [ "dht1:GND", "esp:GND.2", "black", [ "v0" ] ],
    [ "dht1:VCC", "esp:3V3", "red", [ "v0" ] ],
    [ "dht1:SDA", "esp:D15", "green", [ "v0" ] ],
    [ "led1:C", "esp:GND.1", "green", [ "v0" ] ],
    [ "esp:D2", "led1:A", "green", [ "h61.9", "v-53.6", "h86.4", "v57.6" ] ]
  ],
  "dependencies": {}
}
```

- Lampiran

1. Pada Webite Wokwi



2. Pada Aplikasi Mqtt



Link Pengerjaan Wokwi
https://wokwi.com/projects/428920873256902657