

LAPORAN PRAKTIKUM INTERNET OF THINGS (IOT)
PEMANTAUAN SUHU DAN KELEMBAPAN BERBASIS ESP32
MENGGUNAKAN MQTT

FAKULTAS VOKASI UNIVERSITAS BRAWIJAYA

Rifcha Sya'bani Fatullah

Fakultas Vokasi Universitas Brawijaya

Email: rifchasyabani30@gmail.com

ABSTRAK

Tujuan dari praktikum ini adalah untuk membangun sistem pemantauan suhu dan kelembapan berbasis ESP32 yang terhubung dengan broker MQTT. Ini akan dicapai dengan menggunakan simulator Wokwi dan editor kode Visual Studio Code. Sensor DHT22 membaca suhu dan kelembapan secara rutin. Sementara indikator LED digunakan untuk memberi tahu jika suhu atau kelembapan melebihi ambang batas tertentu, data yang diperoleh dikirim ke broker MQTT. Sistem ini dirancang, diuji, dan berfungsi dengan baik secara virtual tanpa perangkat fisik. Hasil menunjukkan bahwa komunikasi MQTT berjalan dengan baik dan LED beradaptasi dengan kondisi lingkungan.

Kata Kunci : ESP32, Wokwi, MQTT

ABTRACT

The purpose of this lab is to build an ESP32-based temperature and humidity monitoring system connected to an MQTT broker. This will be achieved using the Wokwi simulator and the Visual Studio Code code editor. The DHT22 sensor reads temperature and humidity routinely. While the LED indicator is used to notify if the temperature or humidity exceeds a certain threshold, the data obtained is sent to the MQTT broker. The system is designed, tested, and works well virtually without any physical devices. The results show that MQTT communication is running well and the LED adapts to environmental conditions.

Keywords: ESP32, Wokwi, MQTT

1. PENDAHULUAN

1.1 Latar Belakang

Dengan berkembangnya teknologi Internet of Things (IoT), ada banyak peluang untuk membangun sistem yang memungkinkan monitoring dan otomatisasi jarak jauh. Pemantauan suhu dan kelembapan lingkungan secara real-time adalah salah satu implementasinya. Karena keterbatasan alat, biaya, dan risiko kesalahan perakitan, penggunaan perangkat fisik seringkali terbatas dalam konteks pembelajaran. Akibatnya, praktikum ini membangun dan menguji sistem pemantauan suhu dan kelembapan berbasis ESP32 secara virtual dengan memanfaatkan Wokwi Simulator dan VSCode. MQTT sebagai protokol komunikasi memungkinkan pertukaran data yang efektif antara perangkat dan broker. Selain itu, menunjukkan cara sistem Internet of Things bekerja dalam praktik.

1.2 Tujuan

Adapun tujuan dari praktikum ini ialah :

1. Menggunakan ESP32 dan sensor DHT22 untuk membuat sistem yang memantau suhu dan kelembapan.
2. Mengirimkan data ke broker MQTT melalui koneksi WiFi.
3. Menyalakan LED sebagai indikator saat suhu atau kelembapan melebihi batas.

2. METODOLOGI

2.1 Alat dan Bahan

Alat :

- a. Laptop
- b. Platform Wokwi
- c. Platform Visual Studio Code (VSCode)
- d. Platform MQTT Explorer

Bahan :

- a. ESP32 (Virtual dalam platform wokwi)
- b. Sensor DHT22
- c. LED
- d. Bahasa Pemrograman C++ Pustaka Arduino
- e. Library
 - PubSubClient.h
 - Wifi.h
 - DHTesp.h

2.2 Langkah Implementasi

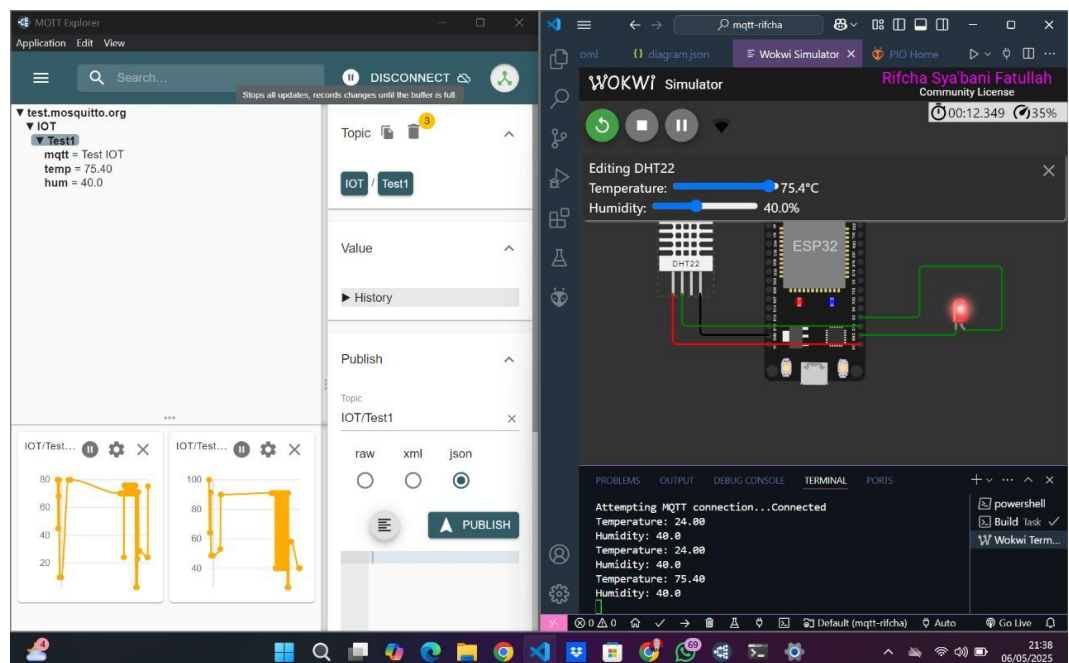
1. Buka Visual Studio Code (VSCode)
2. Unduh dan install ekstensi PlatformIO dan Wokwi pada VSCode

3. Buat proyek baru di Platform untuk ESP32
4. Buka situs Wokwi
5. Buat proyek baru
6. Buat design dengan menambahkan sensor DHT22 dan LED
7. Sambungkan pin VCC dan GND DHT22 ke pin 3V3 dan GND ESP32
8. Sambungkan pin data DHT22 ke pin D15 ESP32
9. Sambungkan pin anoda LED merah ke pin D2 ESP32 dan katoda ke GND
10. Jika sudah selesai membuat rancangan design, Kembali ke VSCode untuk membuat program utama
11. Gunakan library Wifi.h, PubSubClient.h dan DHTesp.h
12. Buka platform MQTT Explorer
13. Atur MQTT connection dengan nama test.mosquitto.org
14. Gunakan protokol mqtt://
15. Gunakan Host test.mosquitto.org
16. Klik advanced
17. Tambahkan topik IOT/Test1/#
18. Jika sudah simpan dan klik connect
19. Jalankan program utama

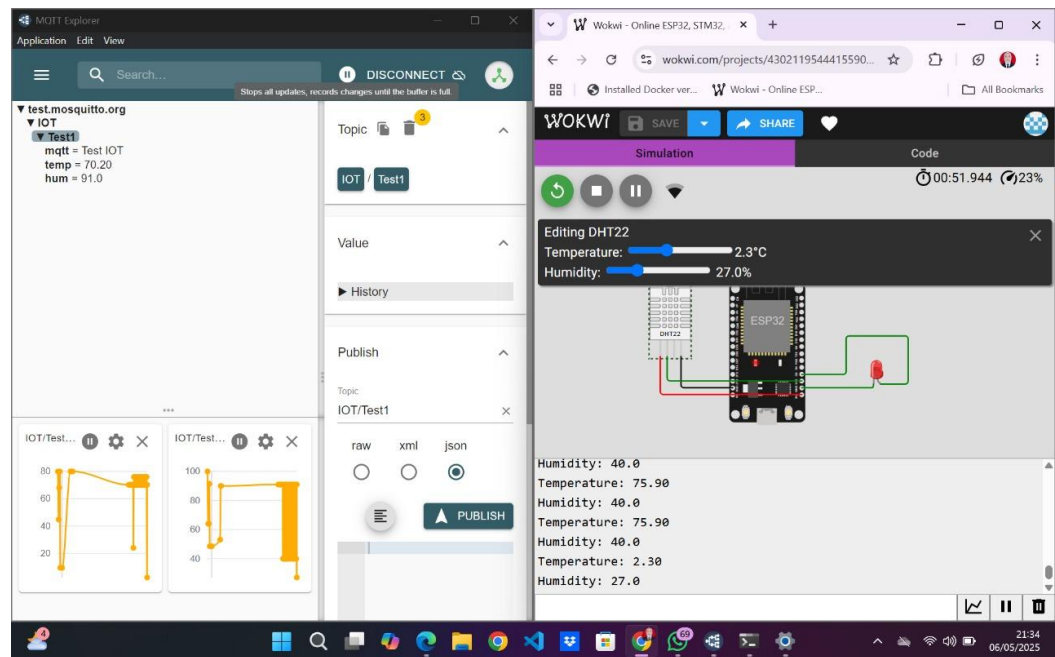
3. Hasil dan Pembahasan

3.1 Hasil Eksperimen

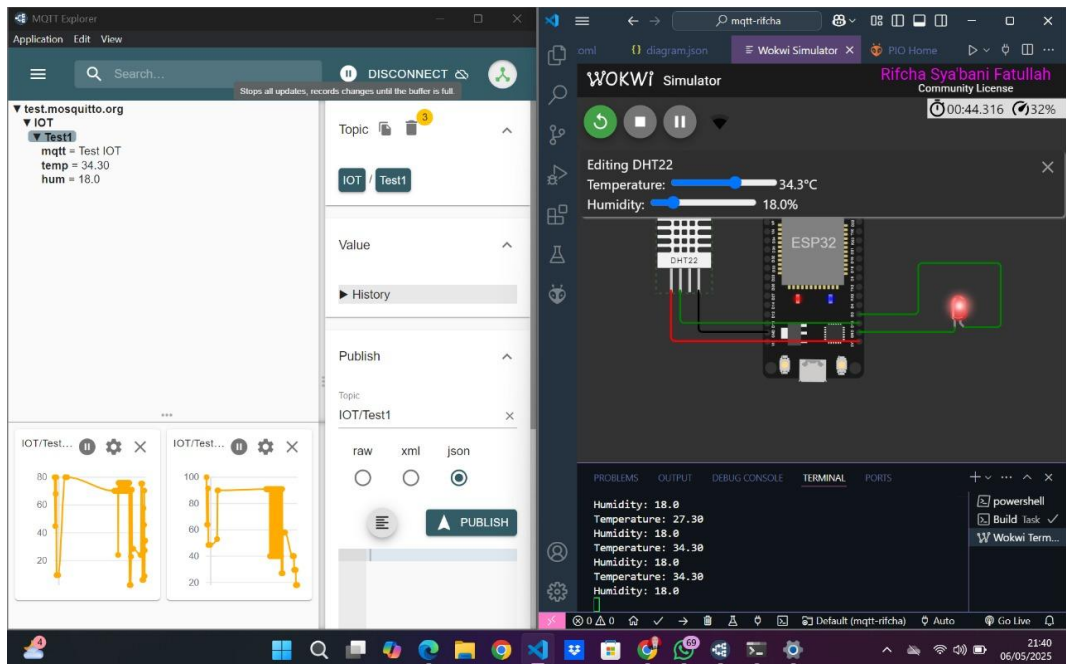
a. Sistem berhasil terhubung dengan MQTT



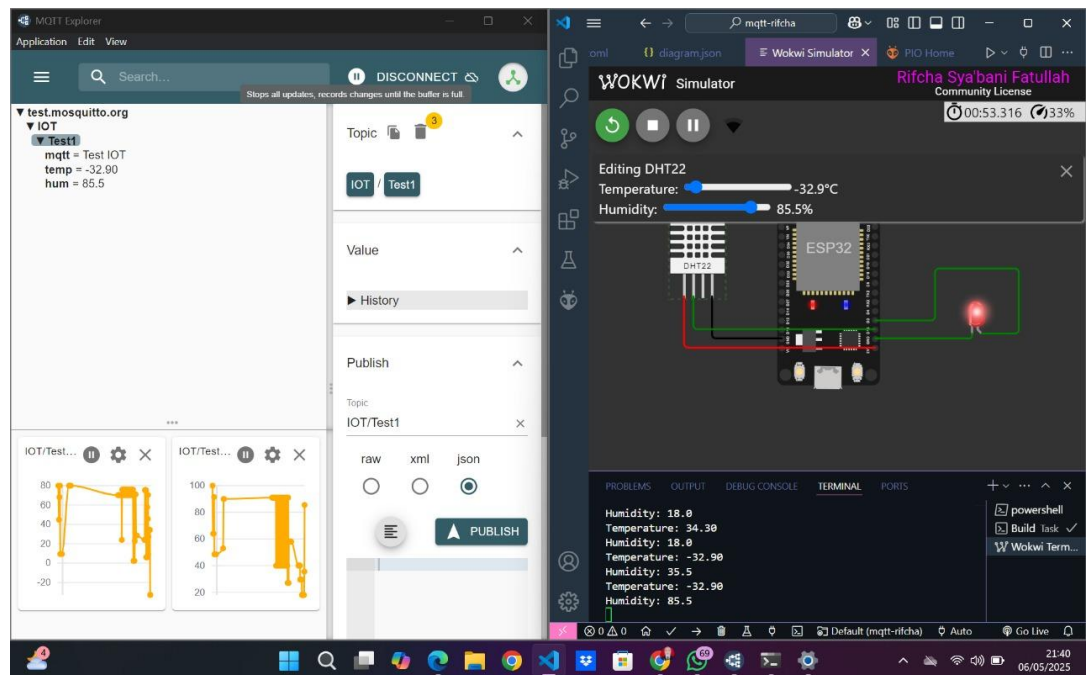
- b. Kondisi lampu indikator akan padam jika temperature kurang dari 30 dan humidity kurang dari 70



- c. Kondisi lampu indikator akan menyala jika temperature lebih dari 30



d. Kondisi lampu indikator akan menyala jika humidity lebih dari 70



3.2 Pembahasan

Hasil simulasi menunjukkan bahwa data suhu dan kelembapan dapat dikirimkan dengan sukses ke broker MQTT setiap dua detik. Saat suhu atau kelembapan melampaui batas, LED indikator menyala secara otomatis. Data sensor dan status koneksi ditampilkan dengan jelas di Serial Monitor. Sistem ini memiliki ESP32 sebagai mikrokontroler, sensor DHT22 untuk mengukur kelembapan dan suhu, dan LED merah sebagai indikator visual. Ketika ESP32 dimulai, ia pertama kali terhubung ke jaringan WiFi, lalu ke broker MQTT. Dengan menggunakan protokol MQTT, data dari DHT22 secara berkala diambil dan dikirim ke broker. Logika pemantauan merancang LED untuk menyala jika kondisi lingkungan melebihi ambang batas.

4. Lampiran

a. Kode Program Main.cpp

```
#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";/"broker.emqx.io";
```

```
WiFiClient espClient;  
PubSubClient client(espClient);  
unsigned long lastMsg = 0;  
float temp = 0;  
float hum = 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);
} else {
    digitalWrite(LED_RED, LOW);
}
}

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);
    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();

  String temp = String(data.temperature, 2);
  client.publish("IOT/Test1/temp", temp.c_str());

  String hum = String(data.humidity, 1);
  client.publish("IOT/Test1/hum", hum.c_str());

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

  if (data.temperature > 30.0 || data.humidity > 70.0) {
    digitalWrite(LED_RED, HIGH);
  } else {
    digitalWrite(LED_RED, LOW);
  }
}
}

```

b. Kode Program diagram.json

```

{
  "version": 1,
  "author": "Rifcha Syabani Fatullah",
  "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": {}  
}
```