

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

Nasywa Anindya Q.E

Fakultas Vokasi Universitas Brawijaya

Email: nasywaanindya@student.ub.ac.id

ABSTRAK

Praktikum ini bertujuan untuk merancang dan mengimplementasikan sistem monitoring suhu dan kelembapan menggunakan ESP32 yang terintegrasi dengan broker MQTT. Sistem dikembangkan melalui simulator Wokwi serta Visual Studio Code sebagai editor pemrograman. Sensor DHT22 secara berkala mendeteksi nilai suhu dan kelembapan, sedangkan LED difungsikan sebagai penanda apabila nilai melebihi batas yang telah ditetapkan. Data hasil pembacaan dikirimkan ke broker MQTT untuk dipantau. Sistem berhasil dibuat, diuji, dan berjalan optimal pada lingkungan virtual tanpa perangkat keras nyata. Hasil pengujian menunjukkan komunikasi MQTT berjalan stabil dan LED merespons sesuai perubahan kondisi

Kata Kunci: ESP32, Wokwi, MQTT

ABSTRACT

This laboratory work aims to design and implement a temperature and humidity monitoring system using ESP32 integrated with an MQTT broker. The system is developed through the Wokwi simulator and Visual Studio Code as the code editor. The DHT22 sensor periodically measures temperature and humidity values, while the LED serves as an indicator when readings exceed the predefined threshold. The data is transmitted to the MQTT broker for monitoring purposes. The system was successfully built, tested, and operated effectively in a virtual environment without physical hardware. The results indicate stable MQTT communication and appropriate LED responses to environmental changes.

Keywords: ESP32, Wokwi, MQTT

1. PENDAHULUAN

1.1. Latar Belakang

Kemajuan teknologi Internet of Things (IoT) membuka peluang luas untuk merancang sistem pemantauan dan otomatisasi jarak jauh. Salah satu implementasinya adalah monitoring suhu dan kelembapan secara real-time. Namun, dalam konteks pembelajaran, penggunaan perangkat keras nyata sering terkendala biaya, risiko kesalahan perakitan, dan keterbatasan alat. Oleh karena itu, praktikum ini memanfaatkan simulator Wokwi dan Visual Studio Code untuk merancang serta menguji sistem pemantauan suhu dan kelembapan berbasis ESP32 secara virtual. Sistem ini dilengkapi protokol MQTT yang mendukung pertukaran data antara perangkat dengan broker, sekaligus memperlihatkan bagaimana sistem IoT diimplementasikan secara praktis.

1.2. Tujuan

Praktikum ini bertujuan untuk:

1. Merancang sistem pemantauan suhu dan kelembapan menggunakan ESP32 dan sensor DHT22.
2. Mengirimkan data hasil pembacaan ke broker MQTT melalui jaringan WiFi.
3. Mengaktifkan LED sebagai indikator saat suhu atau kelembapan melampaui batas yang ditentukan.

2. METODOLOGI

2.1. Alat dan Bahan

2.1.1. Alat

- Laptop
- Wokwi Simulator
- Visual Studio Code (VSCode)
- MQTT Explorer

2.1.2. Bahan

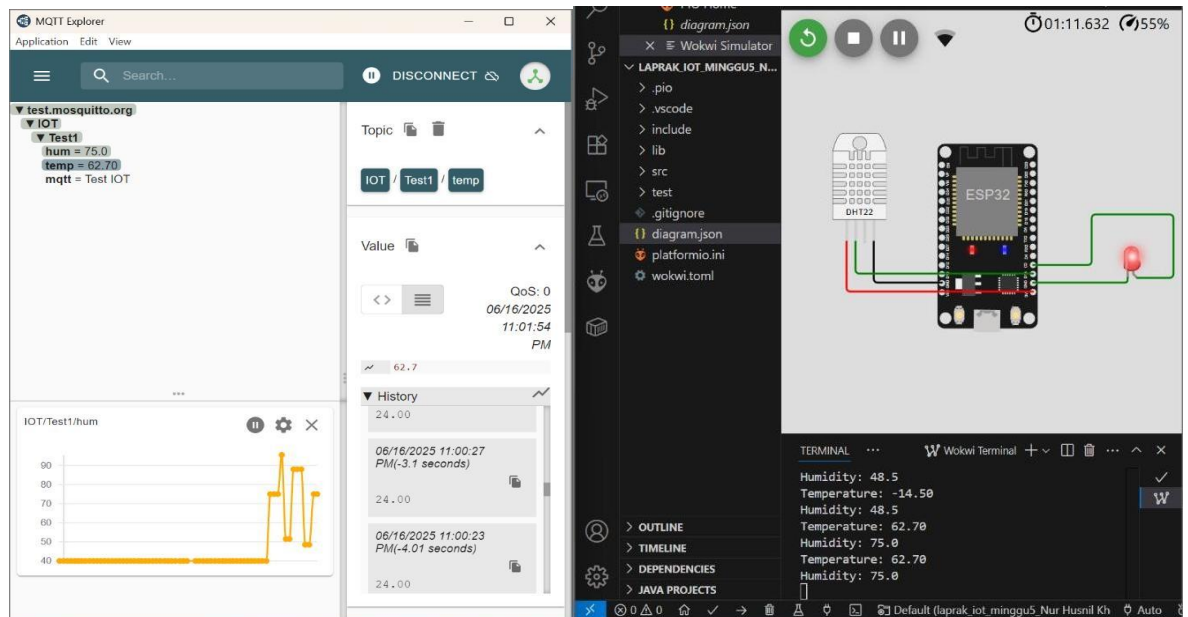
- ESP32 (simulasi di Wokwi)
- Sensor DHT22
- LED
- Bahasa pemrograman C++ dengan pustaka Arduino
- Library:
 - a. Wifi.h
 - b. PubSubClient.h
 - c. DHTesp.h

2.2. Langkah Implementasi

- a. Buka Visual Studio Code dan install ekstensi PlatformIO serta Wokwi.
- b. Buat proyek baru untuk ESP32 di PlatformIO.
- c. Akses Wokwi dan buat proyek baru.
- d. Rancang rangkaian dengan menambahkan sensor DHT22 dan LED.
- e. Sambungkan pin VCC dan GND DHT22 ke 3V3 dan GND ESP32.
- f. Hubungkan pin data DHT22 ke pin D15 ESP32.
- g. Hubungkan anoda LED ke pin D2 ESP32 dan katoda ke GND.
- h. Setelah desain selesai, tulis kode program di VSCode menggunakan library Wifi.h, PubSubClient.h, dan DHTesp.h.
- i. Buka MQTT Explorer, atur koneksi ke broker dengan alamat test.mosquitto.org menggunakan protokol mqtt://.
- j. Di pengaturan lanjutan, tambahkan topik IOT/Test1/#.
- k. Simpan pengaturan lalu klik *Connect*.
- l. Jalankan program utama dan amati hasilnya.

3. HASIL DAN PEMBAHASAN

3.1. Hasil Eksperimen



3.2. Pembahasan

Hasil simulasi menunjukkan bahwa sistem berhasil mengirim data suhu dan kelembapan ke broker MQTT secara rutin setiap dua detik. LED indikator menyala otomatis apabila nilai suhu atau kelembapan melebihi batas yang telah ditentukan. Informasi hasil pembacaan sensor beserta status koneksi tampil jelas pada Serial Monitor. Sistem ini menggunakan ESP32 sebagai mikrokontroler utama, sensor DHT22 untuk mendeteksi suhu dan kelembapan, serta LED merah sebagai indikator visual. Saat ESP32 dinyalakan, perangkat akan lebih dulu terkoneksi ke jaringan WiFi, lalu dilanjutkan dengan sambungan ke broker MQTT. Melalui protokol MQTT, data dari sensor DHT22 dikirimkan secara berkala, dan LED dirancang untuk aktif sebagai penanda jika lingkungan melampaui batas ambang yang sudah ditetapkan.

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";
```

```

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": "Nasywa Anindya",
  "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": {}  
}
```