**PRAKTIKUM SIMULASI KOMUNIKASI MQTT MENGGUNAKAN ESP32 DAN PLATFORM WOKWI DALAM MATA KULIAH INTERNET OF THINGS**

**(IOT)**



Muhammad Fa’iz Ramadhan

Fakultas Vokasi, Universitas Brawijaya

[faizramadhan23@student.ub.ac.id](mailto:faizramadhan23@student.ub.ac.id)

**Abstrak**

Praktikum ini mencoba mensimulasikan komunikasi data antara ESP32 dan broker MQTT dengan protokol publish-subscribe. Dengan menggunakan platform simulasi Wokwi, ESP32 dihubungkan dengan sensor DHT22 dan LED. Data suhu dan kelembapan dikirim ke broker MQTT, dan kontrol LED dapat dilakukan melalui dashboard MQTT seperti MQTT Explorer. Hasil penelitian menunjukkan bahwa ESP32 dengan lancar mengirim dan menerima data melalui protokol MQTT, yang memungkinkan integrasi dengan berbagai aplikasi IoT secara real-time.   
Kata kunci: MQTT, ESP32, Internet of Things, Wokwi, DHT22 Sensor

1. **Pendahuluan**
   1. **Latar belakang**

Protokol MQTT (Message Queuing Telemetry Transport) yang ringan dan efektif adalah salah satu cara Internet of Things (IoT) berkembang dalam sistem otomasi. Arsitektur publish-subscribe digunakan oleh MQTT untuk pertukaran data antar perangkat. ESP32 dapat diintegrasikan dengan broker MQTT untuk mengirim (publish) dan menerima (subscribe) data sensor dengan dukungan koneksi WiFi. Dalam praktikum ini, ESP32 disimulasikan di Wokwi dengan DHT22 sebagai sensor untuk mengukur suhu dan kelembapan, dan LED digunakan sebagai aktuator yang dikontrol melalui pesan MQTT.

* 1. **Tujuan eksperimen**

Tujuan dilakukannya praktikum ini adalah :

1. Mengimplementasikan koneksi ESP32 ke broker MQTT.
2. Memahami konsep komunikasi data MQTT dalam IoT
3. Mengontrol LED melalui perintah MQTT dari dashboard
4. Mempublikasikan data sensor DHT22 ke broker**Metodologi**
   1. **Alat dan Bahan**
5. Mikrokontroler ESP32
6. Sensor DHT22 (temperatur dan kelembapan).
7. LED untuk indikator output.
8. Broker MQTT publik: test.mosquitto.org, port 1883.
   1. **Langkah Implempentasi**
9. Tambahkan ESP32, sensor DHT22, dan LED.
10. Menggunakan library WiFi.h, PubSubClient.h, dan DHTesp.h
11. Menghubungkan ESP32 ke WiFi Wokwi
12. Mengatur koneksi ke broker MQTT
13. Publish data suhu dan kelembapan ke topic MQTT setiap 2 detik
14. Jalankan proyek, buka Serial Monitor
15. Pastikan koneksi WiFi dan MQTT berhasil
16. Buka MQTT Explorer
17. Tambahkan koneksi ke test.mosquitto.org:1883
18. **Hasil dan Pembahasan**

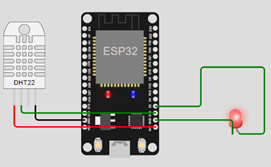
MQTT adalah protokol ringan dan real-time yang efektif untuk komunikasi perangkat Internet of Things (IoT). Dalam simulasi ini, data berhasil dikirim dan diterima melalui broker publik dengan cepat dan efisien. Keunggulan ini memungkinkan pengembangan lebih lanjut, seperti integrasi dengan dashboard visual untuk monitoring atau penyimpanan data ke dalam database cloud untuk analisis lebih lanjut. MQTT sangat cocok digunakan pada sistem IoT karena kemampuannya mengelola data secara cepat dan hemat bandwidth.

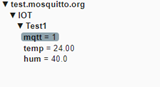
* 1. **Hasil Eksperimen**

Eksperimen simulasi menunjukkan bahwa sistem monitoring berbasis ESP32 dapat berhasil mengintegrasikan sensor DHT22 dan photoresistor untuk memantau parameter lingkungan secara real-time. Sistem mampu mengukur suhu dalam rentang tipikal 0-50°C dengan resolusi 0.1°C, kelembapan 20-90% dengan resolusi 0.1%, dan intensitas cahaya yang dipetakan ke skala 0-100%.

Data yang diperoleh ditampilkan dengan jelas pada layar OLED dengan pembaruan setiap 2 detik, memungkinkan pemantauan yang efektif. Simulasi juga memverifikasi logika program dan koneksi antar komponen, meminimalkan potensi kesalahan dalam implementasi fisik.

Hasil eksperimen ini menunjukkan bahwa kombinasi ESP32, DHT22, photoresistor, dan layar OLED menawarkan solusi yang layak untuk aplikasi monitoring lingkungan skala kecil hingga menengah, dengan potensi pengembangan lebih lanjut seperti integrasi konektivitas nirkabel untuk pemantauan jarak jauh dan penyimpanan data untuk analisis tren jangka panjang.





* 1. **Kode Program**

#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);

}

}