**LAPORAN PRAKTIKUM**

**IOT MINGGU KE-6**

**MQTT-Based Environment Monitoring With Esp32 dan DHT22**

****

**Dosen Pengampu :**

**Ir. Subairi, ST., MT., IPM**

**Disusun Oleh:**

**Qaila Salsabila**

**(233140707111089)**

**PROGRAM STUDI TEKNOLOGI INFORMASI**

**FAKULTAS VOKASI**

**UNIVERSITAS BRAWIJAYA**

**2025**

**Abstrak**

Memanfaatkan protokol MQTT untuk pengiriman data ringan dan efisien, pemantauan lingkungan berbasis Internet of Things (IoT) telah berkembang menjadi pendekatan kontemporer. ESP32 berfungsi sebagai mikrokontroler utama dalam eksperimen ini, dan dia terhubung dengan sensor DHT22 untuk mengukur suhu dan kelembapan. Aplikasi monitoring memungkinkan akses ke data sensor, yang dikirim ke broker MQTT seperti Mosquitto secara real-time melalui koneksi Wi-Fi. Sistem ini memungkinkan pemantauan lingkungan yang mudah diakses dan hemat energi. Hasil eksperimen menunjukkan bahwa sistem dapat beroperasi dengan stabil jika dihubungkan ke jaringan. Integrasi ke sistem otomatisasi rumah pintar, pertanian cerdas, dan pemantauan lingkungan skala besar adalah beberapa potensi aplikasi.

*Keywords - IoT, MQTT, ESP32, DHT22, pemantauan lingkungan, broker MQTT, protokol komunikasi.*

**BAB I**

**PENDAHULUAN**

**1.1 Latar Belakang**

Peluang baru untuk memantau kondisi lingkungan secara real-time telah muncul berkat kemajuan teknologi Internet of Things (IoT). Protokol Message Queuing Telemetry Transport (MQTT), yang memungkinkan pengiriman data ringan dan efisien, adalah salah satu pendekatan komunikasi IoT yang paling populer. Sistem ini menggunakan ESP32 sebagai mikrokontroler dan sensor DHT22 untuk mengukur suhu dan kelembapan dengan akurat. Data yang diperoleh dikirim ke platform monitoring melalui broker MQTT, dan pengguna dapat mengaksesnya. Kombinasi ESP32 dan DHT22 memungkinkan pencatatan data lingkungan yang skalabilitas tinggi, hemat biaya, dan menggunakan sedikit daya. Sistem ini tidak hanya menawarkan solusi untuk monitoring, tetapi juga menawarkan peluang untuk pengembangan lebih lanjut, seperti integrasi ke dalam sistem pertanian cerdas atau otomatisasi rumah pintar.

**1.2 Tujuan Eksperimen**

1. Mengembangkan sistem pemantauan lingkungan berbasis MQTT dengan menggunakan sensor suhu dan kelembapan real-time ESP32 dan DHT22.
2. Menggabungkan protokol MQTT sebagai metode pengiriman data ke platform monitoring atau server yang dapat diakses pengguna.
3. Menguji kemanjuran komunikasi data MQTT dalam skenario Internet of Things untuk memantau lingkungan secara konsisten.

**BAB II**

**METODOLOGI**

**2.1 Alat dan Bahan**

* ESP32 DevKit V1
* Sensor Suhu dan Kelembapan (DHT22)
* Resistor (1kΩ dan 10kΩ)
* Kabel Penghubung Virtual
* Breadboard Virtual
* Platform Simulasi Wokwi

**2.3 Implementasi Sistem**

1. Untuk menghubungkan ESP32 dengan sensor DHT22, gunakan resistor pull-up untuk menyambungkan pin data sensor DHT22 ke salah satu pin GPIO ESP32. Ini akan memastikan pembacaan data yang stabil.
2. Dengan menggunakan Arduino IDE, program ESP32 dapat dikonfigurasi untuk menyambungkan perangkat ke jaringan Wi-Fi. Ini melibatkan pengaturan SSID dan password yang digunakan.
3. Mengatur Komunikasi MQTT: Gunakan pustaka MQTT untuk menghubungkan ESP32 dengan broker MQTT seperti Mosquitto dan mendaftarkan topik pengiriman data suhu dan kelembapan.
4. Mengunggah dan menjalankan program unggah kode yang telah disiapkan ke ESP32 dengan Arduino IDE. Untuk menguji pengiriman data, jalankan simulasi di Wokwi.
5. Memantau Data melalui MQTT: Gunakan aplikasi seperti MQTT Explorer untuk memantau data yang dipublikasikan oleh ESP32 secara real-time melalui broker MQTT.

**BAB III**

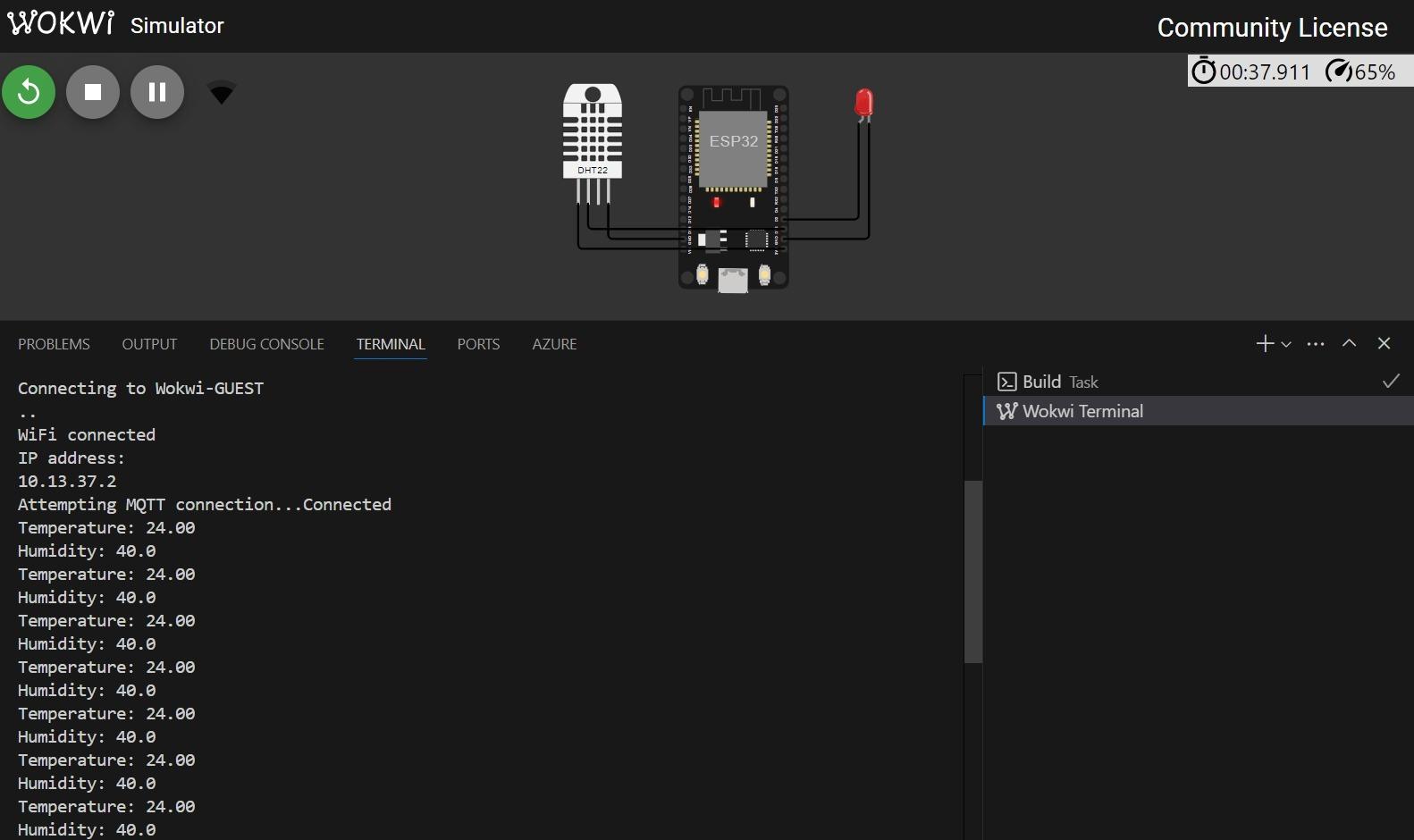
**HASIL DAN PEMBAHASAN**

**3.1 Hasil Eksperimen**

Eksperimen menunjukkan bahwa sistem yang berbasis ESP32 dan DHT22 untuk memantau lingkungan dapat bekerja dengan baik. Aplikasi MQTT Explorer dapat memantau data yang dipublikasikan secara real-time. ESP32 membaca data suhu dan kelembapan dari sensor DHT22 dan kemudian mengirimkannya ke broker MQTT melalui koneksi Wi-Fi.

Pengiriman data ke broker dan koneksi Wi-Fi stabil selama pengujian. Untuk meningkatkan interval pengiriman data dan mengurangi penggunaan daya, beberapa perubahan kecil dilakukan. Hasil ini menunjukkan bahwa sistem dapat digunakan untuk berbagai jenis aplikasi pemantauan lingkungan. Misalnya, rumah pintar dan pertanian cerdas adalah contohnya. Platform Wokwi memudahkan simulasi dan pengujian dan memungkinkan kesalahan diidentifikasi dan diperbaiki tanpa perangkat fisik.

**3.2 Dokumentasi eksperimen meliputi screenshoot simulasi :**

****

**Lampiran**

**Kode Program 1 :**

{

"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": 0.3, "left": -111, "attrs": {} },

{

"type": "wokwi-led",

"id": "led1",

"top": -3.6,

"left": 157.8,

"attrs": { "color": "red", "flip": "1" }

}

],

"connections": [

[ "esp:TX0", "$serialMonitor:RX", "", [] ],

[ "esp:RX0", "$serialMonitor:TX", "", [] ],

[ "dht1:GND", "esp:GND.2", "black", [ "v0" ] ],

[ "dht1:VCC", "esp:3V3", "black", [ "v0" ] ],

[ "dht1:SDA", "esp:D15", "black", [ "v0" ] ],

[ "led1:A", "esp:D2", "black", [ "v0" ] ],

[ "led1:C", "esp:GND.1", "black", [ "v0" ] ]

],

"dependencies": {}

}

**Kode Program 2 :**

#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);

}

}