**Implementasi Sistem Monitoring Suhu dan Kelembapan**

**Berbasis ESP32 dengan MQTT**

*Difa Aqilah*

*Fakultas Vokasi, Universitas Brawijaya*

*Email :* [*difaaqilah04@student.ub.ac.id*](mailto:difaaqilah04@student.ub.ac.id)

**Abstrak**

Menjaga suhu dan kelembapan tetap stabil adalah hal penting di berbagai sektor, mulai dari pertanian hingga ruang server. Di sinilah peran teknologi hadir untuk membantu. Dalam proyek ini, dirancang sebuah sistem berbasis Internet of Things (IoT) yang dapat memantau suhu dan kelembapan secara real-time menggunakan ESP32 dan sensor DHT22. Untuk menghubungkan data dari perangkat ke pengguna, digunakan protokol komunikasi MQTT yang ringan dan efisien. ESP32 bertugas mengirimkan data ke broker MQTT publik, dan data ini bisa dilihat dalam bentuk angka dan grafik lewat tampilan web. Selain memantau, pengguna juga bisa menyalakan atau mematikan LED dari jarak jauh. Proyek ini diuji lewat simulasi Wokwi, dan hasilnya membuktikan sistem berjalan lancar dan stabil.

Kata Kunci — *Internet of Things (IoT), MQTT, ESP32, DHT22, Monitoring Real-Time, Wokwi.*

**PENDAHULUAN**

Suhu dan kelembapan bukan sekadar angka dua hal ini bisa sangat berpengaruh terhadap kenyamanan, keamanan, bahkan kualitas suatu proses. Di Indonesia, pemantauan dua parameter ini sangat penting, terutama di bidang pertanian, industri, dan pengelolaan ruang server. Sayangnya, cara lama yang masih mengandalkan pencatatan manual sudah tidak lagi efektif. Solusinya adalah teknologi IoT yang memungkinkan kita memantau secara otomatis dan real-time, kapan pun dan di mana pun. Proyek ini ingin membuktikan bahwa dengan ESP32, sensor DHT22, dan protokol MQTT, kita bisa menciptakan sistem monitoring yang praktis dan terhubung dengan internet. Bahkan, data bisa langsung dikirim ke tampilan web dan pengguna bisa mengontrol LED dari jauh.

**TUJUAN**

 Membuat sistem monitoring suhu dan kelembapan berbasis ESP32 yang bekerja dengan baik.

 Membangun tampilan web sederhana yang bisa menampilkan data sensor secara langsung dalam bentuk angka dan grafik.

 Menerapkan protokol MQTT sebagai jembatan pengiriman data antara ESP32 dan web interface.

**METODELOGI**

**Alat dan Bahan**

Proyek ini dilakukan secara virtual menggunakan Wokwi. Komponen yang digunakan antara lain:

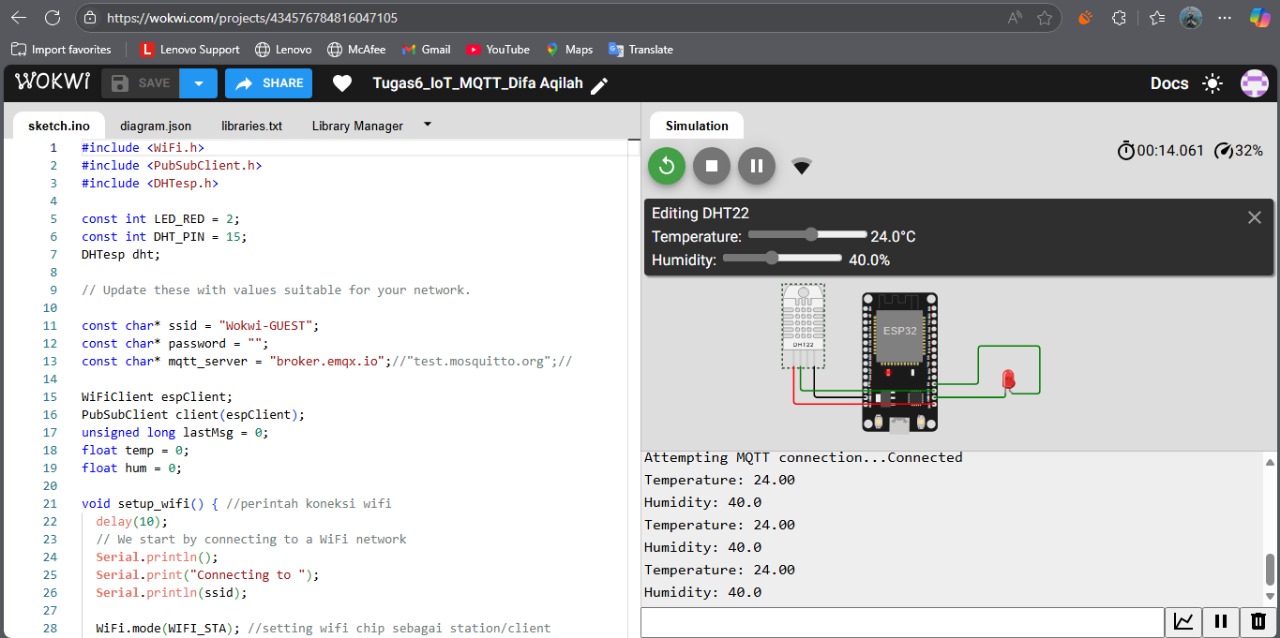
* **ESP32**: Otak dari sistem, mengatur koneksi internet dan pengiriman data.
* **Sensor DHT22**: Mengukur suhu dan kelembapan.
* **LED**: Sebagai indikator, bisa dikendalikan dari jarak jauh.
* **Wokwi**: Tempat kita menyimulasikan rangkaian tanpa perangkat fisik.
* **MQTT broker**: Menggunakan test.mosquitto.org sebagai perantara komunikasi data.
* **Library Arduino**: Untuk membantu koneksi WiFi dan MQTT serta membaca data sensor.

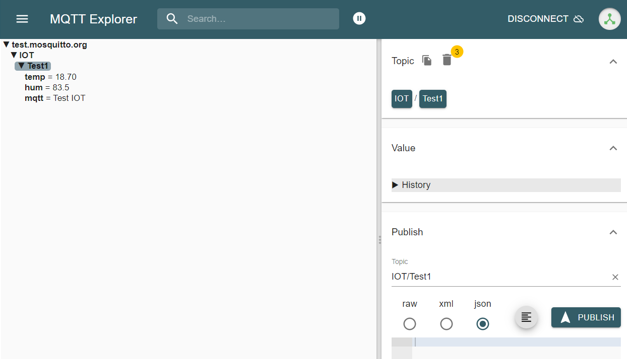
**Langkah Implementasi (Humanis)**

1. **Rancang rangkaian di Wokwi**  
   Sensor dan LED dihubungkan ke ESP32 menggunakan konfigurasi pin tertentu.
2. **Tulis kodenya**  
   ESP32 akan:
   * Terkoneksi ke WiFi Wokwi-GUEST.
   * Terhubung ke broker MQTT.
   * Membaca data suhu dan kelembapan.
   * Mengirimkan data tersebut ke topik MQTT "IOT/Test1".
3. **Uji sistem**  
   Pastikan ESP32 berhasil terhubung dan data muncul di aplikasi MQTT Client seperti MQTT Explorer.

**HASIL DAN PEMBAHASAN**

Secara keseluruhan, sistem monitoring suhu dan kelembapan berbasis ESP32 dan MQTT ini telah berhasil dibuat dan diuji. ESP32 mampu mengirim data sensor secara berkala ke broker MQTT, dan data tersebut bisa diterima oleh aplikasi lain dengan lancar. Format JSON yang digunakan juga menunjukkan sistem ini siap dikembangkan ke level berikutnya. Simulasi di Wokwi membuat semuanya lebih praktis tanpa butuh perangkat fisik. Proyek ini bisa jadi contoh nyata penerapan sistem monitoring IoT yang sederhana, efisien, dan fungsional.





**LAMPIRAN**

**Code sketch.io**

#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 = "broker.emqx.io";//"test.mosquitto.org";//

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);

  }

}

**Code diagram.json**

{

  "version": 1,

  "author": "Difa Aqilah",

  "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": {}

}

**IMPLEMENTASI WOKWI**

<https://wokwi.com/projects/434576784816047105>