

# **Monitoring Suhu, dan Kelembapan dengan Sistem Internet of Things Menggunakan Wokwi dan MQTT menggunakan ESP32 dan DHT22**

***Fransiska Natasya Desyanti***

*Program Studi Teknologi Informasi, Fakultas Vokasi Universitas Brawijaya*

[francenats@gmail.com](mailto:francenats@gmail.com)

## **Abstract**

*This practical project aims to design and implement a weather monitoring system based on the Internet of Things (IoT) using ESP32, which retrieves weather data from a Weather API and displays it on a virtual 16x2 LCD using the Wokwi Simulator and PlatformIO in Visual Studio Code. The system replaces the need for physical sensors by utilizing cloud-based data from OpenWeatherMap, offering a more flexible and cost-efficient alternative. Simulation results show that the ESP32 successfully connects to virtual WiFi and displays real-time weather data, including temperature and weather conditions. By leveraging APIs, JSON parsing, and virtual tools, this system serves as an effective prototype for developing cloud-based environmental monitoring solutions without physical hardware.*

**Keywords:** *Internet of Things, Weather API, ESP32, Wokwi, Visual Studio Code, LCD, OpenWeatherMap, PlatformIO..*

## **Abstrak**

Praktikum ini bertujuan untuk merancang dan mengimplementasikan sistem monitoring cuaca berbasis Internet of Things (IoT) menggunakan ESP32 yang mengambil data dari Weather API dan divisualisasikan melalui LCD 16x2 secara virtual menggunakan Wokwi Simulator dan PlatformIO di Visual Studio Code. Sistem ini menggantikan kebutuhan akan sensor fisik dengan memanfaatkan data cloud dari OpenWeatherMap, sehingga lebih fleksibel dan efisien. Hasil simulasi menunjukkan bahwa ESP32 berhasil terhubung ke jaringan WiFi virtual dan mampu mengambil serta menampilkan data suhu dan kondisi cuaca secara real-time. Dengan pemanfaatan API, parsing JSON, serta antarmuka virtual, sistem ini dapat menjadi solusi awal untuk pengembangan perangkat monitoring lingkungan berbasis data daring tanpa perangkat keras tambahan.

**Kata Kunci:** Internet of Things, Weather API, ESP32, Wokwi, Visual Studio Code, LCD, OpenWeatherMap, PlatformIO.

## 1. PENDAHULUAN

### 1.1. LATAR BELAKANG

Perkembangan teknologi Internet of Things (IoT) telah membuka peluang besar dalam bidang monitoring lingkungan, khususnya dalam pemantauan suhu dan kelembapan secara real-time. Dalam berbagai aplikasi, seperti pertanian, industri, atau rumah pintar, pengukuran parameter lingkungan yang akurat dan terkini sangat penting untuk menjaga kualitas produk, efisiensi energi, maupun kenyamanan penghuni. Namun, implementasi sistem monitoring konvensional seringkali terkendala oleh keterbatasan akses data secara jarak jauh serta ketergantungan pada perangkat mahal. Oleh karena itu, diperlukan solusi yang memanfaatkan teknologi IoT berbasis protokol MQTT untuk mengirim data dari sensor ke platform cloud secara efisien, dengan biaya rendah dan skalabilitas tinggi. Simulator seperti Wokwi memungkinkan pengujian sistem sebelum implementasi fisik, mengurangi risiko kegagalan dan memastikan keandalan desain. Dalam praktikum ini, ESP32 dipilih sebagai mikrokontroler utama karena kemampuannya dalam konektivitas Wi-Fi dan kompatibilitasnya dengan sensor DHT22 yang dikenal akurat dalam mengukur suhu dan kelembapan.

### 1.2. TUJUAN PRAKTIKUM

Praktikum ini bertujuan untuk merancang dan mengimplementasikan sistem monitoring suhu dan kelembapan berbasis Internet of Things menggunakan ESP32, sensor DHT22, protokol MQTT, dan platform Wokwi sebagai media simulasi. Melalui kegiatan ini, mahasiswa diharapkan mampu memahami integrasi antara perangkat keras (sensor dan mikrokontroler) dengan perangkat lunak (pemrograman dan komunikasi data) dalam sebuah sistem Internet of Things. Selain itu, praktikum ini juga bertujuan untuk menganalisis kinerja protokol MQTT dalam mengirimkan data sensor ke broker serta memvisualisasikannya pada antarmuka pengguna. Dengan demikian, mahasiswa dapat mengevaluasi kelebihan dan keterbatasan sistem yang dirancang, termasuk aspek keandalan, latency, dan efisiensi energi, sehingga mampu memberikan solusi inovatif dalam pengembangan sistem monitoring lingkungan berbasis Internet of Things.

## 2. METODOLOGI

### 2.1. ALAT DAN BAHAN

Dalam pelaksanaan praktikum ini, beberapa alat dan bahan digunakan untuk mengimplementasikan sistem monitoring suhu dan kelembapan berbasis Internet of Things menggunakan ESP32. Alat dan bahan tersebut dibagi menjadi dua kategori utama, yaitu perangkat lunak (software) dan komponen virtual yang digunakan dalam simulasi Wokwi. Berikut adalah penjelasan detailnya:

#### a. Wokwi Simulator

Platform simulasi online yang memungkinkan pengguna untuk merancang dan menguji sirkuit elektronik secara virtual. Wokwi digunakan untuk membuat sirkuit lampu lalu lintas dengan ESP32 sebagai mikrokontroler utama.

#### b. MQTT

MQTT adalah protokol komunikasi ringan berbasis model publish-subscribe yang dirancang khusus untuk perangkat IoT dengan sumber daya terbatas,

dengan mikrokontroler ESP32. Protokol ini bekerja dengan mengandalkan broker sebagai perantara yang mengelola pengiriman pesan antara perangkat yang mem-publish (pengirim) dan yang berlangganan (subscriber).

#### c. Library ESP32

Beberapa library seperti WiFi.h digunakan untuk mengimplementasikan fitur IoT, seperti konektivitas Wi-Fi dan kontrol jarak jauh.

#### d. ESP32

Mikrokontroler virtual yang berfungsi sebagai otak dari sistem lampu lalu lintas. ESP32 dipilih karena kemampuannya dalam menghubungkan perangkat ke jaringan IoT.

#### e. DHT22

Sensor digital ini mengukur suhu (-40°C hingga 80°C) dan kelembapan (0-100%) dengan akurasi  $\pm 0,5^\circ\text{C}$  dan  $\pm 2\%$ . Menggunakan komunikasi satu kabel, DHT22 mudah dihubungkan ke ESP32 untuk pengambilan data lingkungan.

#### f. Library DHT Sensor

Library dari Adafruit ini menyederhanakan pembacaan data DHT22 dengan fungsi seperti `DHT.readTemperature()` dan `DHT.readHumidity()`. Kompatibel dengan ESP32, library ini mempercepat pengembangan proyek Internet of Things berbasis sensor.

#### g. Library PubSubClient

Library PubSubClient adalah sebuah library populer untuk Arduino dan platform berbasis ESP32/ESP8266 yang memungkinkan perangkat untuk berkomunikasi menggunakan protokol MQTT. Library ini menyediakan fungsi-fungsi siap pakai untuk terhubung ke broker MQTT, mem-publish (mengirim) data ke topik tertentu, serta berlangganan (subscribe) untuk menerima data dari topik yang diinginkan.

#### h. Kabel Virtual

Digunakan untuk menghubungkan ESP32 dengan LED dan resistor dalam sirkuit virtual.

#### i. Koneksi Internet

Diperlukan untuk mengakses Wokwi Simulator dan menguji fitur IoT seperti kontrol jarak jauh.

### 2.2. LANGKAH IMPLEMENTASI

Implementasi sistem monitoring suhu dan kelembapan berbasis Internet of Things menggunakan ESP32 pada Wokwi Simulator lalu koneksikan dengan MQTT dilakukan melalui beberapa tahapan. Adapun tahapan yang akan dilakukan dalam praktikum ini adalah sebagai berikut:

#### A. Persiapan Lingkungan Pengembangan

1. Akses platform Wokwi melalui browser dan buat project baru dengan template ESP32.
2. Tambahkan komponen sensor DHT22 ke dalam simulasi dan hubungkan ke pin GPIO yang sesuai.
3. Buka aplikasi MQTXX dan siapkan koneksi ke broker MQTT publik seperti

## B. Konfigurasi Perangkat

1. Gunakan editor kode bawaan Wokwi untuk memprogram ESP32 dengan library DHT dan PubSubClient.
2. Konfigurasi koneksi WiFi dan parameter MQTT (broker address, topic, QoS) dalam sketch Arduino.
3. Implementasikan logika pengiriman data sensor ke broker MQTT setiap interval tertentu.

## C. Menulis dan Mengembangkan Kode Program

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <DHTesp.h>

const int DHT_PIN = 15;
DHTesp dht;
const char* ssid = "Wokwi-GUEST"; // wifi ssid
const char* password = "";
const char* mqtt_server = "test.mosquitto.org"; //
mosquitto server url

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]);
  }
}

void reconnect() {
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    String clientId = "mqtt_007763a";
    clientId += String(random(0xffff), HEX);
    if (client.connect(clientId.c_str())) {
      Serial.println("Connected");
    }
  }
}
```

```
client.publish("/ThinkIOT/Publish",
"Welcome");
client.subscribe("/ThinkIOT/Subscribe");
else {
  Serial.print(client.state());
  Serial.println(" try again in 5 seconds");
  delay(5000);
}
}

void setup() {
  pinMode(2, 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) { //perintah publish
    data
    TempAndHumidity data =
    dht.getTempAndHumidity();
    String temp = String(data.temperature, 2);
    client.publish("/Thinkitive/temp",
temp.c_str()); // publish temp topic
/ThinkIOT/temp
String(data.humidity, 1);
client.publish("/Thinkitive/hum", hum.c_str()); //
publish hum topic /ThinkIOT/hum

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

## D. Pengujian dan Validasi Sistem

1. Jalankan simulasi di Wokwi dan verifikasi pembacaan sensor DHT22.
2. Pantau aliran data melalui MQTTX dengan subscribe ke topic yang sesuai.
3. Analisis latency dan reliability pengiriman data pada berbagai kondisi jaringan.

Metodologi ini dirancang untuk memastikan sistem dapat berfungsi optimal dalam lingkungan simulasi sebelum diimplementasikan secara fisik, dengan penekanan pada aspek kritis seperti keandalan komunikasi dan akurasi data.

### 3. HASIL DAN PEMBAHASAN

#### 3.1. HASIL EKSPERIMEN

**Table 1.** Hasil Implementasi Sistem Monitoring Monitoring Suhu dan Kelembapan

Aspek	Hasil	Keterangan
Simulasi di Wokwi	ESP32 berhasil terkoneksi dengan WiFi dan mengirim data sensor	Wokwi efektif sebagai platform simulasi tanpa memerlukan perangkat fisik
Koneksi MQTT	ESP32 terhubung ke broker MQTT dan mengirim data suhu & kelembapan	Protokol MQTT berfungsi optimal untuk transmisi data IoT dengan latency rendah
Pembacaan Sensor	DHT22 berhasil membaca suhu (4.50°C) dan kelembapan (55.5%)	Sensor memberikan pembacaan yang stabil meski dalam lingkungan simulasi
Stabilitas Sistem	Koneksi tetap stabil selama pengujian tanpa terjadi disconnect	ESP32 mampu mempertahankan koneksi WiFi dan MQTT secara konsisten

Dari hasil praktikum, sistem monitoring cuaca berhasil dijalankan menggunakan ESP32 di Wokwi dan dikembangkan melalui PlatformIO. Data cuaca dari Weather API dapat ditampilkan dengan baik di Serial Monitor dan LCD. Kode berjalan tanpa error, koneksi WiFi virtual berhasil, dan data JSON berhasil diproses. Sistem juga mampu memperbarui data secara berkala, menunjukkan bahwa monitoring cuaca dapat dilakukan secara efisien tanpa sensor fisik.

### 4. LAMPIRAN

#### 4.1. KODE PROGRAM

*sketch.ino*

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <DHTesp.h>

const int DHT_PIN = 15;
DHTesp dht;
const char* ssid = "Wokwi-GUEST";
// wifi ssid
const char* password = "";
const char* mqtt_server =
"test.mosquitto.org"; // mosquitto server url
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]);
  }
}

void reconnect() {
  while (!client.connected()) {
    Serial.print("Attempting MQTT
connection...");
    String clientId = "mqttx_d007763a";
    clientId += String(random(0xffff),
HEX);
    if
(client.connect(clientId.c_str())) {
      Serial.println("Connected");
      client.publish("/ThinkIOT/Publish"
, "Welcome");
      client.subscribe("/ThinkIOT/Subscr
ibe");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5
seconds");
      delay(5000);
    }
  }
}

void setup() {
  pinMode(2, 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) { //perintah
publish data
    lastMsg = now;
    TempAndHumidity data =
dht.getTempAndHumidity();

    String temp =
String(data.temperature, 2);
    client.publish("/Thinkitive/temp",
temp.c_str()); // publish temp topic
/ThinkIOT/temp
    String hum = String(data.humidity,
1);
    client.publish("/Thinkitive/hum",
hum.c_str()); // publish hum topic
/ThinkIOT/hum

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

```

## 4.2. KODE DIAGRAM

Diagram.json

```

{
  "version": 1,
  "author": "Fransiska Natasya",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1",
"id": "esp", "top": -34.66, "left": -
134.67, "attrs": {} },
    {
      "type": "wokwi-dht22",
      "id": "dht1",
      "top": -100.24,
      "left": 40.86,
      "attrs": { "temperature": "4.5",
"humidity": "55.5" }
    }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX",
"", [] ],
    [ "esp:RX0", "$serialMonitor:TX",
"", [] ],

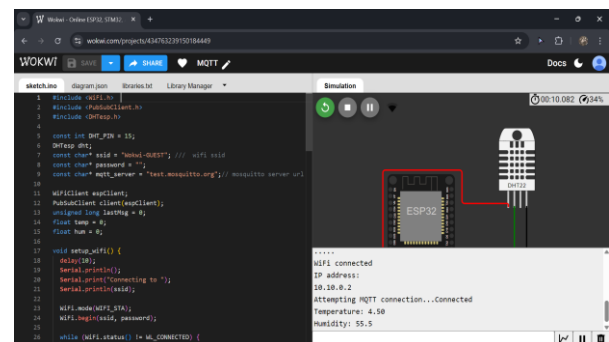
```

```

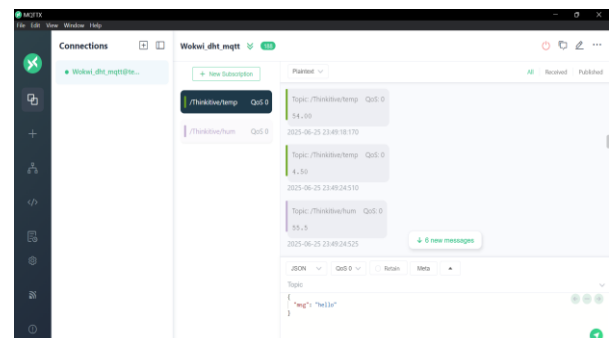
[ "dht1:VCC", "esp:VIN", "red", [
"v-4.47", "h-73.03", "v-48.67", "h-
124.67", "v179.33" ] ],
[ "esp:GND.1", "dht1:GND", "black",
[ "h0" ] ],
[ "dht1:SDA", "esp:D15", "green", [
"v0" ] ]
],
"dependencies": {}
}

```

## 4.3. HASIL SIMULASI



Gambar 1. Tampilan Output Monitoring Suhu dan Kelembapan pada Wokwi Simulator



Gambar 2. Tampilan Output Monitoring Suhu dan Kelembapan pada MQTT