

# **Laporan Praktikum Internet of Things (IoT) Simulasi Pemantauan Suhu dan Kelembaban Berbasis IoT Menggunakan ESP32 dan MQTT**

**ADITYA FERDIAN RAMDANI**  
**2331407007111018**

***Teknologi Informasi, Fakultas Vokasi, Universitas Brawijaya***  
[BRINGMETHEHORIZON389@GMAIL.COM](mailto:BRINGMETHEHORIZON389@GMAIL.COM)

**Version:0.9 StartHTML:0000000105 EndHTML:0000016867  
StartFragment:0000000141 EndFragment:0000016827**

## **Abstrak**

Praktikum ini bertujuan untuk merancang dan mengimplementasikan sistem pemantauan suhu dan kelembaban berbasis Internet of Things (IoT) dengan menggunakan mikrokontroler ESP32, sensor DHT22, dan protokol komunikasi MQTT. Sistem ini mampu membaca data suhu dan kelembaban secara berkala dari sensor

DHT22, kemudian mengirimkan data tersebut ke broker MQTT publik (test.mosquitto.org) melalui koneksi WiFi.

Selain itu, sistem juga mampu menerima perintah dari broker untuk mengendalikan LED sebagai indikator,

berdasarkan data yang dikirim ke topik tertentu. Komunikasi dua arah ini menjadikan sistem mampu melakukan

pemantauan dan kontrol secara real-time dari jarak jauh. Implementasi dilakukan melalui simulasi di platform

Wokwi yang memungkinkan pengujian tanpa perangkat keras fisik. Hasil pengujian menunjukkan bahwa sistem

berhasil membaca dan mengirimkan data sensor setiap dua detik serta merespon perintah kendali LED sesuai

input yang diterima melalui topik MQTT.

***Kata Kunci—Internet of Things, Wokwi, MQTT, DHT22, ESP32***

## **Abstract**

This practicum aims to design and implement an Internet of Things (IoT)-based temperature and humidity monitoring system using ESP32 microcontroller, DHT22 sensor, and MQTT communication protocol.

The system is able to periodically read temperature and humidity data from the DHT22 sensor, then send the data

to a public MQTT broker (test.mosquitto.org) via a WiFi connection. In addition, the system is also able to receive

commands from the broker to control LEDs as indicators, based on data sent to specific topics. This two-way

communication makes the system capable of real-time monitoring and control remotely. Implementation is done

through simulation on the Wokwi platform which allows testing without physical hardware. The test results show

that the system successfully reads and sends sensor data every two seconds and responds to LED control commands according to the input received through the MQTT topic.

**Keywords—** *Internet of Things, Wokwi, MQTT, DHT22, ESP32*. Introduction (Pendahuluan)

Perkembangan teknologi Internet of Things (IoT) telah membawa kemajuan dalam berbagai bidang salah satunya pada sistem monitoring dan kendali jarak jauh. IoT memungkinkan berbagai perangkat fisik untuk saling terhubung dan bertukar data melalui jaringan internet. Salah satu implementasi adalah

sistem pemantauan suhu dan kelembaban berbasis mikrokontroler. Dengan bantuan sensor lingkungan, perangkat IoT dapat secara real-time membaca dan mengirimkan informasi ke server atau pengguna secara langsung. Dalam praktikum ini, digunakan mikrokontroler ESP32 sebagai pusat pengendali, sensor DHT22 untuk membaca suhu dan kelembaban, serta protokol MQTT untuk komunikasi data secara efisien.

### 1.1 Latar Belakang

Pemantauan suhu dan kelembaban merupakan aspek penting dalam berbagai aspek baik dari pertanian maupun hal lain. Pemantauan seringkali dilakukan secara manual, yang tidak efisien dan kurang mampu memberikan data secara real-time, terutama untuk pemantauan jarak jauh. Dengan adanya teknologi Internet of Things (IoT), proses ini dapat diotomatisasi dengan memanfaatkan sensor dan perangkat mikrokontroler yang saling terhubung melalui jaringan internet.

Praktikum ini menggunakan mikrokontroler ESP32 sebagai pusat kendali, sensor DHT22 untuk membaca data suhu dan kelembaban, serta protokol komunikasi MQTT untuk mengirim dan menerima data. MQTT dipilih karena mendukung komunikasi ringan berbasis sistem publish subscribe, sehingga cocok digunakan dalam perangkat IoT dengan keterbatasan sumber daya.

Seluruh sistem dirancang untuk mampu melakukan pemantauan lingkungan dan juga mengendalikan perangkat seperti LED secara jarak jauh melalui internet. Praktikum ini dilaksanakan secara virtual tanpa menggunakan komponen fisik sehingga memungkinkan untuk memahami konsep dan alur kerja sistem IoT secara menyeluruh, mulai dari pemrograman mikrokontroler, pengambilan data sensor, hingga komunikasi dengan broker MQTT, dengan cara yang lebih fleksibel dan mudah diakses.

### 1.2 Tujuan Eksperimen

Eksperimen ini bertujuan untuk:

-

Mengirimkan data sensor secara berkala ke broker MQTT menggunakan protokol komunikasi MQTT dengan koneksi WiFi.

-

Memahami alur kerja sistem IoT, mulai dari pengambilan data sensor, pengiriman data ke broker, hingga penerimaan perintah untuk aktuator melalui jaringan internet.

## 2. Methodology (Metodologi)

### 2.1 Tools & Materials (Alat dan Bahan)

Untuk menjalankan eksperimen ini, beberapa alat dan bahan yang digunakan adalah:

-

Platform simulasi Wokwi: Digunakan untuk melakukan simulasi sistem secara digital.

-

Visual Studio Code dengan ekstensi PlatformIO

-

Kode program berbasis C/C++: Digunakan untuk mengontrol urutan nyala-mati LED virtual dalam simulasi.

-

Mikrokontroler ESP32 (virtual pada Wokwi dan PlatformIO): Sebagai unit pemrosesan utama dan sebagai pengendali yang dilengkapi dengan Wi-Fi dan Bluetooth.

-

Sensor DHT22 : untuk membaca suhu dan kelembaban

-

**LED (merah) :** Sebagai indikator output, dikontrol via perintah dari MQTT

-

**MQTT Broker (test.mosquitto.org)** Layanan broker MQTT publik untuk komunikasi data

-

**Library :** PubSubClient.h, WiFi.h, DHTesp.h

## **2.2 Implementation Steps (Langkah Implementasi)**

-

**Buat project baru di Visual Studio Code dengan project baru melewati PlatformIO.**

-

**Buat project di wokwi simulator.**

-

**Tambahkan perangkat lalu sambungkan dan tambahkan library yang dibutuhkan.**

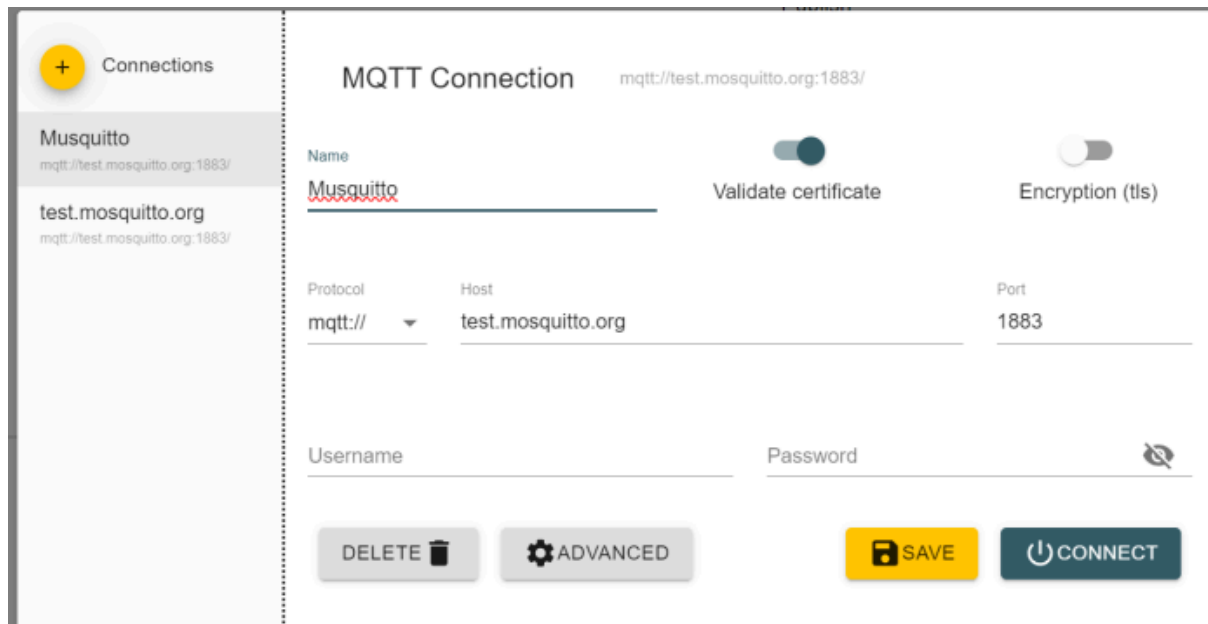
-

**Tambahkan kode program lalu uji.-**

**Buka MQTT Explorer yang sudah terinstall di laptop**

-

**Atur MQTT Connection, pada bagian advance masukkan topik IOT/Test1/# lalu simpan**



*Jalankan program di wokwi lalu amati*

-

*Buka VSC dengan project baru*

-

*Tambahkan semua yang dibutuhkan seperti kode program di main.cpp, kode untuk diagram.json, dan library yang dibutuhkan lalu build*

-

*Tambahkan wokwi.toml*

-

*Jalankan program lalu amati antara yang ada di VSC dan di MQTT Explorer,*

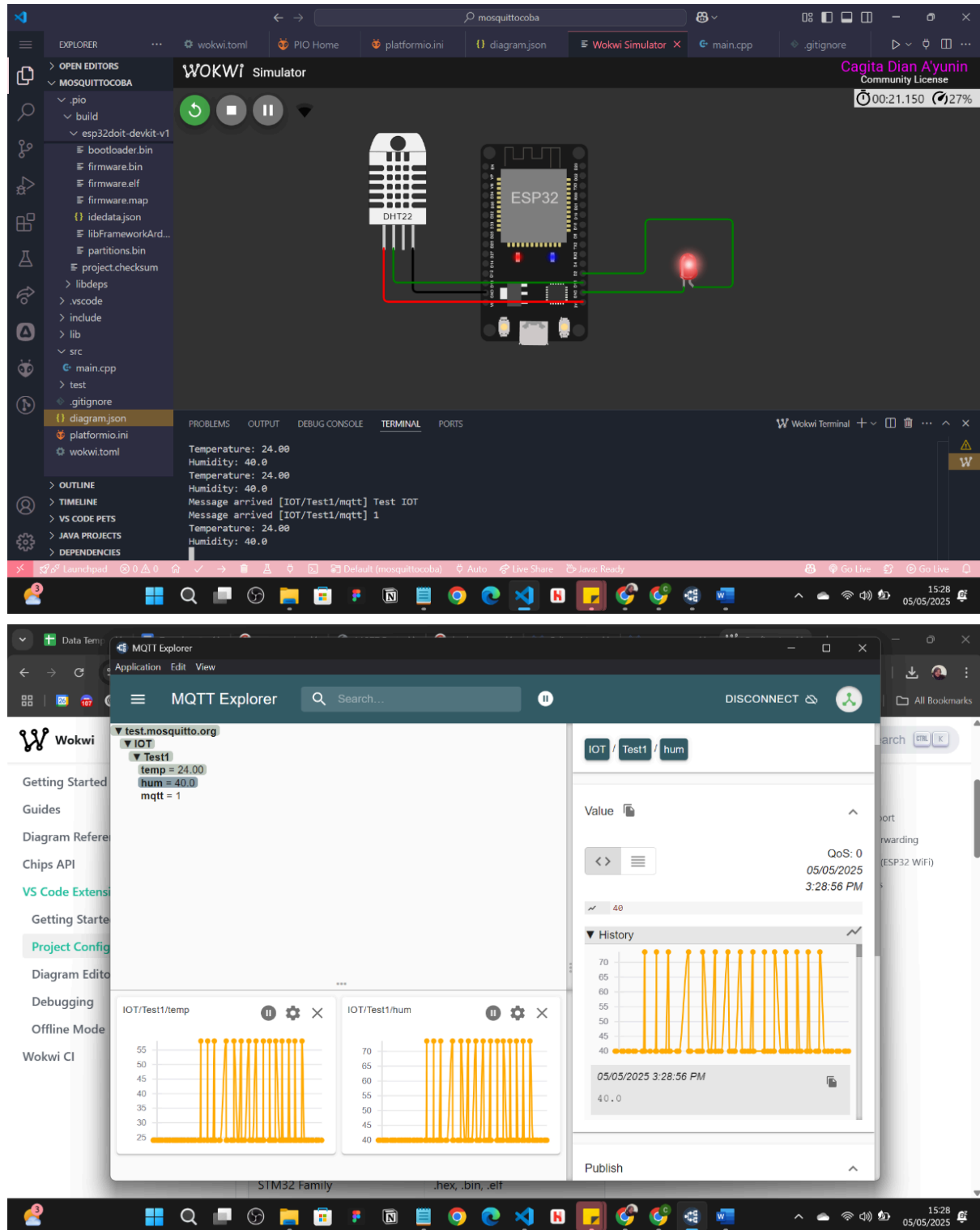
-

*LED menyala : 1, LED mati : 0*

## **3. Result dan Discussion (Hasil dan Pembahasan)**

### **3.1 Experimental Result (Hasil Eksperimen)**

*Berikut adalah screenshot hasil simulasi pada Visual Studio Code:*



Hasil praktikum menunjukkan bahwa MQTT terbukti dalam komunikasi anatar perangkat IoT. Pada percobaan ini berhasil tanpa adanya hambatan

## 4. Appendix (Lampiran)

### 4.1 Kode Program (Main.cpp)

```
#include <Arduino.h>
#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 sampe 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 publuser 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); // inialisasi 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); //inisialisasi 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);
}
}
}

```

#### 4.2 Kode Program (diagram.json)

```
{
"version": 1,
"author": "Aditya Ferdian Ramdani",
"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": {}
}
```