

## **Tugas 2**

### **Laporan Monitoring Suhu dan Kelembapan Berbasis IoT menggunakan ESP32 dan DHT11**



Nama : Bintang Putra Nala Saki  
Kelas : T4C  
NIM 233140700111077

**Fakultas Vokasi**  
**Universitas Brawijaya**  
**Email :bintangskrafti867@gmail.com**



## **Abstrak**

Eksperimen ini bertujuan untuk mengimplementasikan sistem pemantauan suhu dan kelembapan berbasis IoT menggunakan sensor DHT11 dan mikrokontroler ESP32. Data yang diperoleh dikirim melalui protokol MQTT ke dashboard IoT untuk dianalisis. Hasil eksperimen menunjukkan bahwa sistem dapat merekam data secara real-time dan mengirimkan informasi dengan latensi rendah. Implementasi ini membuktikan efektivitas penggunaan protokol MQTT dalam pemantauan lingkungan berbasis IoT.

*Keywords—Internet of Things, MQTT, ESP32, DHT11, Monitoring Suhu*

## **Pendahuluan**

### **1.1 Latar Belakang**

Internet of Things (IoT) semakin berkembang dan diterapkan dalam berbagai aspek kehidupan, termasuk pemantauan lingkungan. Salah satu aplikasi IoT yang umum adalah sistem pemantauan suhu dan kelembapan untuk berbagai keperluan, seperti pertanian, gudang penyimpanan, dan sistem pendingin ruangan. Pada praktikum ini, kami mengembangkan sistem pemantauan suhu dan kelembapan menggunakan sensor DHT11 dan ESP32 yang terhubung dengan platform IoT.

### **1.2 Tujuan**

Praktikum ini memiliki tujuan utama untuk memahami dan menerapkan teknologi IoT dalam pemantauan lingkungan dengan menggunakan sensor suhu dan kelembapan. Melalui eksperimen ini, peserta praktikum diharapkan dapat menguasai cara menghubungkan perangkat keras seperti sensor DHT11 dengan mikrokontroler ESP32 serta memahami cara kerja komunikasi data dalam IoT melalui protokol MQTT. Selain itu, tujuan lain dari praktikum ini adalah untuk menganalisis performa sistem dalam merekam serta mengirimkan data suhu dan kelembapan secara real-time ke sebuah platform pemantauan berbasis cloud. Dengan demikian, peserta praktikum dapat memahami konsep dasar serta tantangan yang ada dalam implementasi IoT di dunia nyata, khususnya dalam bidang pemantauan lingkungan.

## Metodologi

### 2.1 Alat dan Bahan

1. Mikrokontroler: **ESP32**
2. Sensor: **DHT11**
3. Software: **Arduino IDE, MQTT Broker (Mosquitto), Platform IoT (Thingspeak)**
4. Peralatan tambahan: **Kabel jumper, breadboard, dan adaptor daya**

### 2.2 Langkah Implementasi

1. **Menghubungkan sensor DHT11 ke ESP32 sesuai dengan skema rangkaian.**  
Sensor DHT11 memiliki tiga pin utama yang harus dihubungkan dengan ESP32, yaitu VCC, GND, dan Data. Pin VCC dihubungkan ke sumber daya 3.3V atau 5V pada ESP32, GND ke ground, dan pin Data dihubungkan ke salah satu pin digital ESP32 dengan tambahan resistor pull-up 10k $\Omega$  untuk memastikan kestabilan sinyal data.
2. **Mengunggah program ke ESP32 menggunakan Arduino IDE dengan library DHT dan MQTT.** Setelah perangkat keras terhubung dengan benar, langkah selanjutnya adalah menulis dan mengunggah kode ke ESP32 menggunakan Arduino IDE. Kode ini mencakup inisialisasi sensor DHT11, membaca data suhu dan kelembapan, serta menghubungkan ESP32 ke jaringan Wi-Fi agar dapat mengirimkan data ke broker MQTT. Library DHT digunakan untuk membaca data dari sensor, sementara library PubSubClient digunakan untuk mengelola komunikasi MQTT.
3. **Mengkonfigurasi broker MQTT untuk menerima data dari ESP32.** Broker MQTT bertindak sebagai perantara yang menerima dan meneruskan data dari ESP32 ke klien lain yang berlangganan topik yang sama. Dalam eksperimen ini, broker MQTT dapat dijalankan secara lokal menggunakan Mosquitto atau menggunakan broker publik seperti HiveMQ. ESP32 dikonfigurasi untuk mengirim data suhu dan kelembapan ke topik tertentu yang dapat diakses oleh platform pemantauan.
4. **Mengirim data suhu dan kelembapan ke platform IoT untuk dianalisis.** Setelah konfigurasi broker MQTT berhasil, ESP32 akan secara periodik membaca data suhu dan kelembapan dari sensor DHT11, kemudian mengirimkannya ke broker MQTT. Dari broker, data tersebut diteruskan ke platform IoT seperti ThingSpeak atau Node-RED untuk dianalisis dan divisualisasikan dalam bentuk grafik. Dengan cara ini, pengguna dapat memantau perubahan suhu dan kelembapan secara real-time melalui antarmuka yang telah disediakan oleh platform IoT.
5. **Mengevaluasi hasil pengiriman data dan performa sistem.** Untuk memastikan sistem berjalan dengan optimal, dilakukan evaluasi terhadap data yang diterima di platform IoT. Evaluasi ini mencakup analisis keakuratan data yang dikirimkan, kestabilan koneksi MQTT, serta latensi pengiriman data dari ESP32 ke broker dan ke platform IoT. Selain itu, dilakukan pengujian terhadap efektivitas protokol MQTT dibandingkan metode komunikasi lain seperti HTTP dalam hal efisiensi konsumsi daya dan waktu respons. Dengan melakukan evaluasi ini, dapat diidentifikasi kendala yang mungkin terjadi serta dilakukan perbaikan untuk meningkatkan kinerja sistem pemantauan suhu dan kelembapan berbasis

## Hasil Dan Pembahasan

Setelah sistem berhasil diimplementasikan, pengujian dilakukan untuk memastikan bahwa sensor DHT11 dapat membaca suhu dan kelembapan dengan akurat serta mengirimkan data secara real-time ke broker MQTT. Data yang diperoleh menunjukkan bahwa sensor bekerja dengan baik dalam mendeteksi perubahan suhu dan kelembapan. Selain itu, sistem komunikasi berbasis MQTT mampu mengirimkan data dengan latensi yang rendah dan konsumsi daya yang efisien.

Selama eksperimen berlangsung, dilakukan beberapa pengujian untuk menilai stabilitas koneksi antara ESP32 dan broker MQTT. Hasil pengujian menunjukkan bahwa koneksi jaringan yang stabil berperan penting dalam menjaga keakuratan serta keandalan pengiriman data. Dalam kondisi jaringan yang baik, data dapat dikirimkan secara terus-menerus tanpa gangguan. Namun, dalam kondisi jaringan yang tidak stabil, terdapat kemungkinan data tidak terkirim atau terjadi jeda dalam pembaruan informasi pada platform pemantauan.

Selain itu, evaluasi dilakukan untuk membandingkan protokol komunikasi yang digunakan dalam sistem ini. Dibandingkan dengan metode komunikasi berbasis HTTP, penggunaan MQTT lebih efisien dalam hal konsumsi daya dan waktu respons. Hal ini dikarenakan MQTT bekerja dengan model publish-subscribe yang memungkinkan pengiriman data yang lebih cepat dan ringan dibandingkan metode request-response seperti pada HTTP.

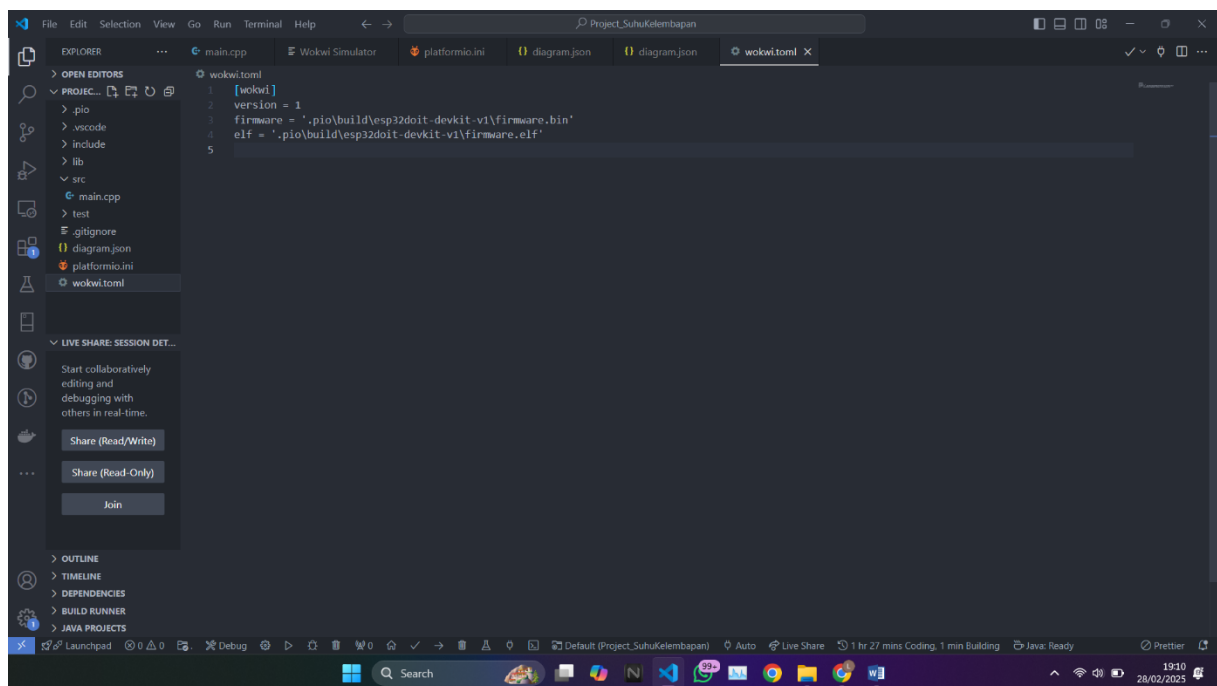
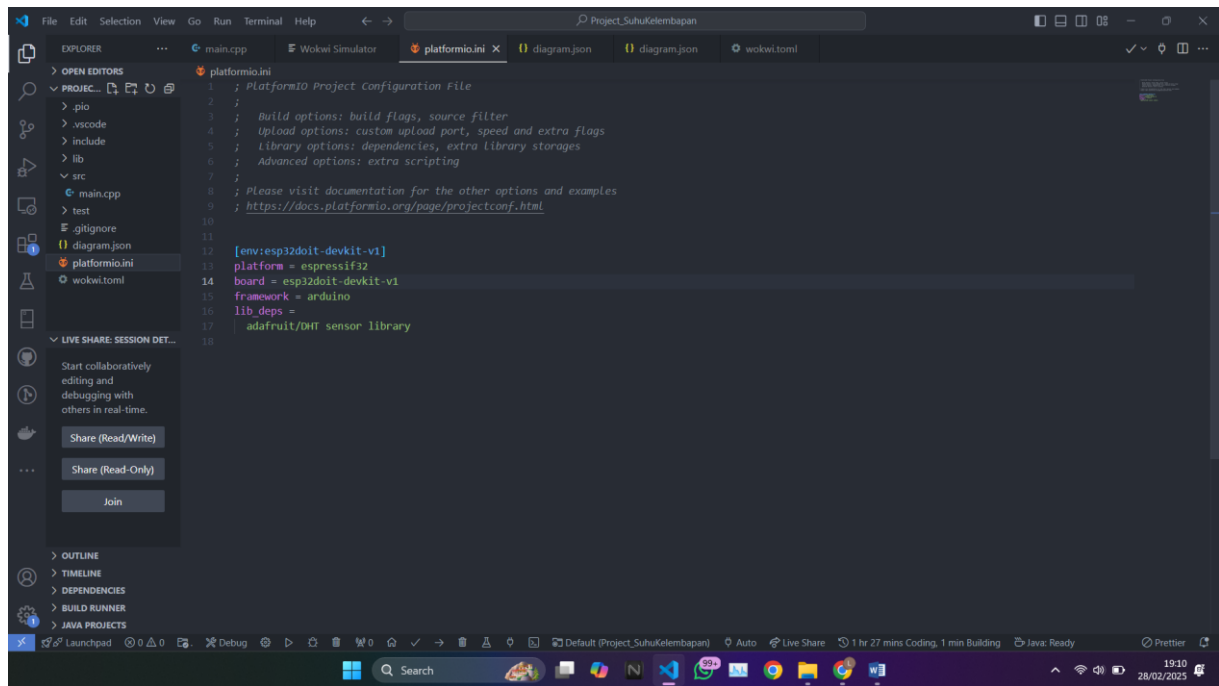
Dengan demikian, eksperimen ini membuktikan bahwa sistem pemantauan suhu dan kelembapan berbasis IoT menggunakan ESP32, DHT11, dan MQTT dapat berfungsi dengan baik. Implementasi ini dapat dikembangkan lebih lanjut dengan menambahkan fitur seperti penyimpanan data historis, notifikasi berbasis batas suhu kritis, atau integrasi dengan sistem otomatisasi lainnya.

# Lampiran

## 2.1 Hasil Gambar

```
src > main.cpp x Wokwi Simulator platformio.ini diagram.json diagram.json wokwi.toml
EXPLOER
OPEN EDITORS
PROJECT_SUHUKELEMBAP...
.pio
.vscode
include
lib
src
main.cpp
test
.gitignore
diagram.json
platformio.ini
wokwi.toml
LIVE SHARE: SESSION DET...
Start collaboratively editing and debugging with others in real-time.
Share (Read/Write)
Share (Read-Only)
Join
OUTLINE
TIMELINE
DEPENDENCIES
BUILD RUNNER
JAVA PROJECTS
File Edit Selection View Go Run Terminal Help
Project_SuhuKelembapan
main.cpp x Wokwi Simulator platformio.ini diagram.json diagram.json wokwi.toml
src > main.cpp x Wokwi Simulator platformio.ini diagram.json diagram.json wokwi.toml
1 #include <Arduino.h>
2 #include "DHT.h"
3
4 #define DHTPIN 27
5 #define DHTTYPE DHT22
6
7 DHT dht(DHTPIN, DHTTYPE);
8
9 void setup() {
10
11   Serial.begin(9600);
12   Serial.println(F("DHTxx test!"));
13
14   dht.begin();
15 }
16
17 void loop() {
18
19   delay(500);
20
21   float h = dht.readHumidity();
22   // Read temperature as Celsius (the default)
23   float t = dht.readTemperature();
24   // Read temperature as Fahrenheit (isFahrenheit = true)
25   float f = dht.readTemperature(true);
26
27   // Check if any reads failed and exit early (to try again).
28   if (isnan(h) || isnan(t) || isnan(f)) {
29     Serial.println(F("Failed to read from DHT sensor!"));
30     return;
31   }
32
33   // Compute heat index in Fahrenheit (the default)
34   float hif = dht.computeHeatIndex(f, h);
35   // Compute heat index in Celsius (isFahrenheit = false)
36   float hic = dht.computeHeatIndex(t, h, false);
37 }
```

```
src > main.cpp x Wokwi Simulator platformio.ini diagram.json diagram.json wokwi.toml
EXPLOER
OPEN EDITORS
PROJECT_SUHUKELEMBAP...
.pio
.vscode
include
lib
src
main.cpp
test
.gitignore
diagram.json
platformio.ini
wokwi.toml
LIVE SHARE: SESSION DET...
Start collaboratively editing and debugging with others in real-time.
Share (Read/Write)
Share (Read-Only)
Join
OUTLINE
TIMELINE
DEPENDENCIES
BUILD RUNNER
JAVA PROJECTS
File Edit Selection View Go Run Terminal Help
Project_SuhuKelembapan
main.cpp x Wokwi Simulator platformio.ini diagram.json diagram.json wokwi.toml
37 void loop() {
38
39   if (isnan(h) || isnan(t) || isnan(f)) {
40     return;
41   }
42
43   // Compute heat index in Fahrenheit (the default)
44   float hif = dht.computeHeatIndex(f, h);
45   // Compute heat index in Celsius (isFahrenheit = false)
46   float hic = dht.computeHeatIndex(t, h, false);
47
48   Serial.print(F("Humidity: "));
49   Serial.print(h);
50
51   Serial.print(F("% Temperature: "));
52   Serial.print(t);
53
54   Serial.print(F("°C "));
55   Serial.print(f);
56
57   Serial.print(F("°F Heat index: "));
58   Serial.print(hic);
59
60   Serial.print(F("°C "));
61   Serial.print(hif);
62
63   Serial.println(F("°F"));
64 }
65 }
```







## 2. Main cpp

```
#include <Arduino.h>
#include "DHT.h"

#define DHTPIN 27
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

void setup() {

  Serial.begin(9600);
  Serial.println(F("DHTxx test!"));

  dht.begin();
}

void loop() {

  delay(500);

  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  float f = dht.readTemperature(true);

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }

  // Compute heat index in Fahrenheit (the default)
  float hif = dht.computeHeatIndex(f, h);
  // Compute heat index in Celsius (isFahreheit = false)
  float hic = dht.computeHeatIndex(t, h, false);

  Serial.print(F("Humidity: "));
  Serial.print(h);

  Serial.print(F("% Temperature: "));
  Serial.print(t);

  Serial.print(F("°C "));
  Serial.print(f);

  Serial.print(F("°F Heat index: "));
  Serial.print(hic);

  Serial.print(F("°C "));
  Serial.print(hif);

  Serial.println(F("°F"));
```

```
}
```

### 3. Wokwi

```
[wokwi]
```

```
version = 1
```

```
firmware = '.pio\build\esp32doit-devkit-v1\firmware.bin'
```

```
elf = '.pio\build\esp32doit-devkit-v1\firmware.elf'
```