**LAPORAN PRAKTIKUM**

**IOT MINGGU KE-8**

**TEMPERATURE AND HUMIDITY MONITORING SIMULATION BASED ON IOT USING ESP32 AND NODERED**

****

**Dosen Pengampu :**

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

**Disusun Oleh:**

**Qaila Salsabila**

**(233140707111089)**

**PROGRAM STUDI TEKNOLOGI INFORMASI**

**FAKULTAS VOKASI**

**UNIVERSITAS BRAWIJAYA**

**2025**

**Abstrak**

Salah satu komponen penting dalam manajemen lingkungan, kesehatan, dan pertanian adalah pemantauan suhu dan kelembapan. Mikrokontroler ESP32 dan sensor DHT22 digunakan dalam sistem untuk mengukur suhu dan kelembapan secara real-time. Dengan menggunakan protokol MQTT, data dikirim ke dashboard Node-RED. Di sana, data ditampilkan dalam bentuk grafik yang mudah dipahami. Eksperimen dapat dilakukan tanpa perangkat fisik dengan platform simulasi Wokwi, yang memudahkan pengembangan sistem berbasis IoT. Hasil eksperimen menunjukkan bahwa sistem berfungsi dengan baik dengan koneksi data yang stabil dan pengolahan yang efisien pada Node-RED. Kombinasi ESP32, DHT22, dan Node-RED menawarkan solusi murah dan efisien untuk aplikasi IoT seperti rumah pintar dan pertanian cerdas.

*Keywords - ESP32, DHT22, Node-RED, IoT, Pemantauan Lingkungan, MQTT, Wokwi.*

**BAB I**

**PENDAHULUAN**

**1.1 Latar Belakang**

Aplikasi yang lebih terintegrasi untuk memantau lingkungan sekarang dapat diakses berkat kemajuan teknologi Internet of Things (IoT). Pememantauan suhu dan kelembapan adalah aplikasi penting, yang relevan untuk pengelolaan bangunan, kesehatan, dan pertanian. Mikrokontroler ESP32 dan kemampuan konektivitas Wi-Fi memungkinkan perangkat untuk menghubungkan data sensor ke platform visualisasi seperti Node-RED. Node-RED menawarkan lingkungan pemrograman berbasis alur yang mudah digunakan untuk mengelola data IoT. Dengan mengintegrasikan sensor suhu dan kelembapan dalam Node-RED, sistem dapat mengambil data secara real-time yang dapat divisualisasikan dan dianalisis untuk meningkatkan pengambilan keputusan. Sistem ini tidak hanya menawarkan solusi ekonomis tetapi juga memungkinkan pengembangan lebih lanjut.

**1.2 Tujuan Eksperimen**

1. Menggunakan ESP32 dan Node-RED untuk mengumpulkan data secara real-time untuk membuat sistem pemantauan suhu dan kelembapan berbasis IoT.
2. Mengevaluasi keandalan koneksi antara sensor, mikrokontroler, dan platform Node-RED untuk visualisasi data lingkungan.
3. Membuat model pemantauan lingkungan yang murah dan mudah diakses untuk mendukung otomatisasi dan analisis berbasis Internet of Things.

**BAB II**

**METODOLOGI**

**2.1 Alat dan Bahan**

* SP32 DevKit V1
* Sensor DHT22
* Breadboard
* Kabel Jumper (Virtual)
* Wokwi
* Koneksi Internet
* Software Arduino IDE atau editor yang kompatibel
* Pustaka DHTesp untuk pengolahan data sensor
* Platform Node-RED untuk visualisasi data.

**2.3 Implementasi Sistem**

1. Hubungkan ESP32 dengan sensor DHT22 di breadboard virtual Wokwi menggunakan pin VCC, GND, dan GPIO untuk data.
2. Program ESP32 menggunakan Arduino IDE untuk membaca data suhu dan kelembapan dari DHT22.
3. Tambahkan koneksi Wi-Fi dalam kode untuk mengirim data ke dashboard Node-RED.
4. Integrasikan Node-RED untuk menerima data melalui protokol MQTT atau HTTP dan menampilkan data pada grafik atau indikator.
5. Jalankan simulasi di Wokwi untuk memvalidasi data yang diterima di dashboard Node-RED.

**BAB III**

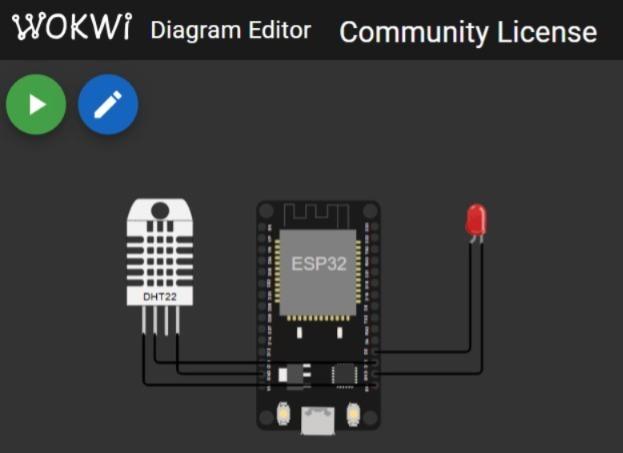
**HASIL DAN PEMBAHASAN**

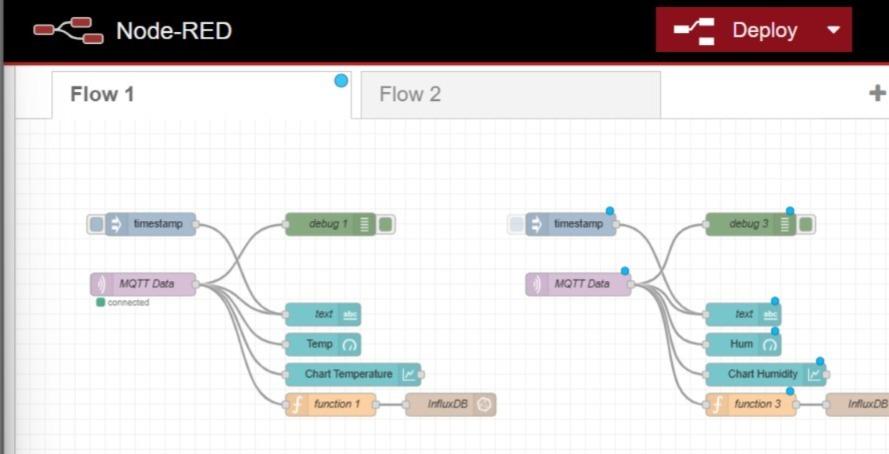
**3.1 Hasil Eksperimen**

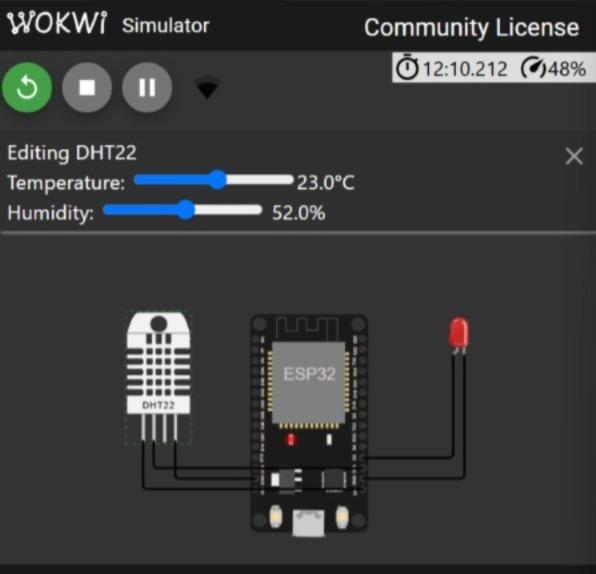
Eksperimen menunjukkan bahwa sistem pemantauan suhu dan kelembapan berbasis ESP32 dan Node-RED berfungsi dengan baik. Data sensor DHT22 dapat dibaca oleh ESP32 dan dikirim ke dashboard Node-RED melalui protokol MQTT. Data ini ditampilkan dalam grafik yang menunjukkan perubahan suhu dan kelembapan secara dinamis. Selama pengujian, sistem berfungsi dengan baik. Ini termasuk pemulihan koneksi Wi-Fi setelah gangguan sementara.

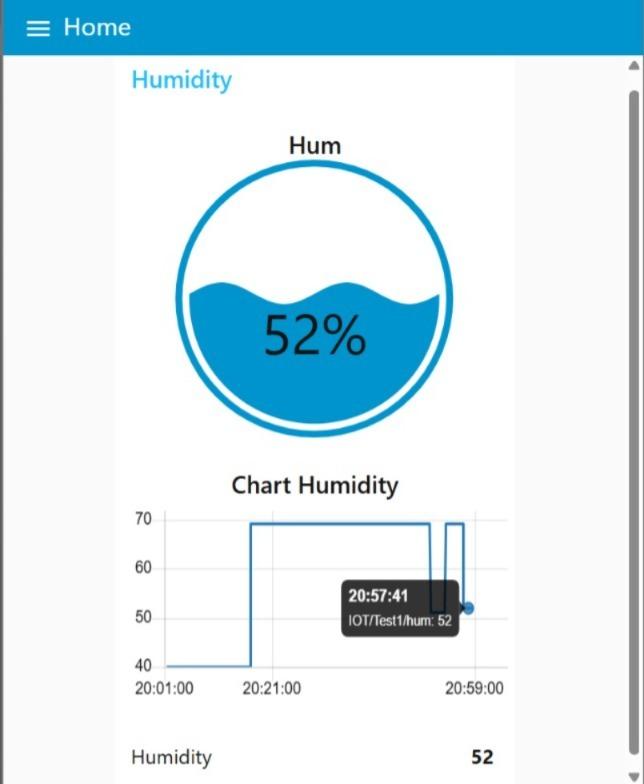
Pengujian yang dilakukan tanpa perangkat fisik dengan platform Wokwi memungkinkan integrasi perangkat lunak dan perangkat keras yang lancar. Node-RED mudah diatur dan memungkinkan penyesuaian tampilan data sesuai kebutuhan. Hasil eksperimen menunjukkan bahwa kombinasi ESP32, DHT22, dan Node-RED berfungsi dengan baik untuk aplikasi IoT yang membutuhkan pemantauan data lingkungan secara real-time.

**3.2 Dokumentasi eksperimen meliputi screenshoot simulasi :**

****

****

****

****

**Lampiran**

**Kode Program 1 :**

"v134.4",

"h-9.6",

"v0",

"h0",

"v-115.2",

"h-28.8",

"v134.4",

"h0",

"v9.6"

]

]

],

"dependencies": {}

}

**Kode Program 2 :**

#include <Arduino.h>

#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 = "broker.emqx.io"; //server

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()); //Topic Temperature

String hum = String(data.humidity, 1);

client.publish("IOT/Test1/hum", hum.c\_str()); //Topic Humidity

Serial.print("Temperature: ");

Serial.println(temp);

Serial.print("Humidity: ");

Serial.println(hum);

}

}