

Perancangan Sistem Monitoring Suhu, Kelembapan, dan Status LED Berbasis IoT Menggunakan Wokwi, MQTT, Node-RED, dan InfluxDB

oleh

Nuha Rona Zahra

Fakultas Vokasi, Universitas Brawijaya

Email: nuharonazz@gmail.com

Abstrak

Penelitian ini bertujuan untuk merancang dan mengimplementasi sistem monitoring berbasis Internet of Things yang mampu mengukur suhu, kelembapan, dan status LED secara real-time. Sistem ini menggunakan simulasi ESP32 pada platform Wokwi yang terhubung dengan broker MQTT untuk pengiriman data. Data yang diterima oleh Node-RED diproses dan disimpan ke dalam basis data time-series InfluxDB, kemudian divisualisasikan dalam dashboard yang interaktif. Hasil dari proyek ini menunjukkan bahwa sistem berhasil menampilkan data sensor dan status LED secara real-time sehingga dapat digunakan sebagai dasar pengembangan sistem pemantauan lingkungan berbasis IoT.

Kata Kunci: Node-RED, MQTT, InfluxDB

Abstract (Bahasa Inggris)

This study aims to design and implement an Internet of Things-based monitoring system that is able to measure temperature, humidity, and LED status in real-time. This system uses ESP32 simulation on the Wokwi platform connected to the MQTT broker for data transmission. Data received by Node-RED is processed and stored in the InfluxDB time-series database, then visualized in an interactive dashboard. The results of this project show that the system successfully displays sensor data and LED status in real-time so that it can be used as a basis for developing an IoT-based environmental monitoring system.

Keywords: Node-RED, MQTT, InfluxDB

Pendahuluan

Kebutuhan sistem monitoring berbasis IoT meningkat di era modern. Salah satunya yaitu aplikasi untuk memantau kondisi lingkungan seperti suhu dan kelembapan serta untuk mengontrol perangkat seperti LED. Dalam laporan ini membuat sistem pemantauan sederhana menggunakan platform Wokwi dan simulator ESP32 untuk mengirimkan data sensor ke Node-RED melalui teknologi MQTT. data ini kemudian disimpan ke InfluxDB dan divisualisasikan ke dalam dashboard.

Metodologi

Metode yang digunakan dalam praktik ini mencakup beberapa tahapan utama yaitu:

1. Simulasi perangkat dengan menggunakan Wokwi untuk mensimulasikan ESP32 yang terhubung dengan sensor suhu/kelembapan.
2. Pengiriman Data dari ESP32 ke dalam format JSON ke broker MQTT publik.
3. Pemrosesan Node-RED yaitu
 - Menerima data dari MQTT
 - Menguraikan data JSON
 - Memformat dan mengirim data ke InfluxDB

- Memvisualisasikan data menggunakan Node-RED dashboard.
4. Penyimpanan data menggunakan InfluxDB sebagai penyimpanan data time-series
 5. Analisis data yang ditampilkan dalam bentuk teks dan grafik (gauge) untuk menunjukkan nilai suhu, kelembapan, dan status LED.

Hasil dan Pembahasan

Sistem berhasil menerima dan menampilkan data suhu dan kelembapan secara real-time dari simulasi ESP32. Data ditransmisikan secara stabil melalui broker MQTT dan dapat dilihat secara langsung ke dashboard Node-RED. LED yang dikontrol dari kode juga tercatat statusnya. Data yang diterima diformat menjadi measurement pada InfluxDB, dan berhasil disimpan serta di query melalui fitur data explorer.

The figure consists of three screenshots arranged vertically, showing the Node-RED interface, the Wokwi simulation environment, and the InfluxDB Data Explorer.

Node-RED Screenshot: Shows the Node-RED interface with three flows. Flow 1 starts with an "inject" node, followed by a timestamp node, and ends at an "influxDB" node. Flow 2 starts with an "IOT/Test1/data" node, followed by a "temperature" node, a "debug f" node, and ends at a "Chart Temp" node. Flow 3 starts with the same "IOT/Test1/data" node, followed by a "Temp" node, and ends at the "influxDB" node. The "influxDB" node is connected to a "msgq-broker" node. A sidebar shows various nodes and their connections.

Wokwi Simulation Screenshot: Shows the Wokwi IDE with an open sketch named "sketch.ino". The code initializes WiFi, connects to a network, and prints connection details and sensor values (Temperature: 24.00, Humidity: 40.0). To the right, the Node-RED "IOT monitoring" dashboard displays a gauge chart titled "Temp" with a value of 24.00.

InfluxDB Data Explorer Screenshot: Shows the InfluxDB Data Explorer interface with a query editor. The query is:

```

SELECT *
  WHERE
    time > now() - interval '1 hour'
  
```

The results table shows 0 rows. Below the table, a note says "Select data and run query to view results".

Kesimpulan

Sistem monitoring berbasis IoT menggunakan Wokwi, MQTT, Node-RED, dan InfluxDB telah berhasil dirancang dan diimplementasikan. Sistem mampu menangkap data dari sensor, mengirimkannya ke server, dan menampilkan secara real-time dalam bentuk dashboard. Sistem ini dapat dijadikan dasar untuk pengembangan pemantauan lingkungan atau smart system lainnya dengan skala lebih luas dan penggunaan perangkat fisik.

Lampiran

```
#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 = "test.mosquitto.org";

WiFiClient espClient;
PubSubClient client(espClient);
unsigned long lastMsg = 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);
        Serial.println("LED dinyalakan lewat MQTT");
    } else {
        digitalWrite(LED_RED, LOW);
        Serial.println("LED dimatikan lewat MQTT");
    }
}

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);
digitalWrite(LED_RED, LOW); // pastikan mati di awal
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();

float temp = data.temperature;
float hum = data.humidity;

```

```

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

String tempStr = String(temp, 2);
client.publish("IOT/Test1/temp", tempStr.c_str());

String humStr = String(hum, 1);
client.publish("IOT/Test1/hum", humStr.c_str());

if (isnan(temp) || isnan(hum)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
}

// 🔥 Ganti ambang jadi lebih rendah agar LED pasti nyala
if (temp > 20.0 || hum > 30.0) {
    digitalWrite(LED_RED, HIGH);
    Serial.println("LED ON karena suhu atau kelembaban tinggi (test)");
} else {
    digitalWrite(LED_RED, LOW);
    Serial.println("LED OFF karena suhu & kelembaban normal");
}
}

{
    "version": 1,
    "author": "Subairi",
    "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": {}  
}
```