

# Praktik Implementasi Monitoring Suhu dan Kelembapan dengan Node-red dan Influxdb

oleh

*Nadia Alya Paramitha Erwanto<sup>1</sup>*

*Fakultas Vokasi, Universitas Brawijaya*

*Email: [nadiaalya2729@student.ub.ac.id](mailto:nadiaalya2729@student.ub.ac.id)*

## Abstrak

Perkembangan teknologi Internet of Things (IoT) semakin mendorong inovasi dalam pengumpulan dan analisis data sensor secara real-time. Salah satu pendekatan yang semakin banyak digunakan adalah integrasi dengan Node-RED dan InfluxDB, yang dapat membantu dalam mengelola aliran data sensor, melakukan visualisasi, serta menyimpan data secara efisien.

Dalam praktik ini, digunakan mikrokontroler ESP32 untuk membaca data sensor yang kemudian dikirimkan ke sistem berbasis Node-RED untuk dianalisis dan diteruskan ke InfluxDB sebagai basis penyimpanan data time-series. Proses dilakukan melalui simulasi menggunakan platform Wokwi, dengan alur pengiriman data dirancang agar mencerminkan kondisi nyata dalam sistem IoT.

Studi kasus yang diangkat adalah pengiriman data suhu dan kelembapan, yang kemudian dipantau melalui dasbor visualisasi guna mendeteksi pola atau tren kondisi lingkungan. Hasil praktik menunjukkan bahwa integrasi antara ESP32, Node-RED, dan InfluxDB dapat menjadi solusi cerdas dan fleksibel dalam pengembangan sistem IoT berbasis data.

**Kata kunci:** *IoT, ESP32, Node-RED, InfluxDB, Data Sensor, Wokwi*

## Abstract

Perkembangan teknologi Internet of Things (IoT) semakin mendorong inovasi dalam pengumpulan dan analisis data sensor secara real-time. Salah satu pendekatan yang semakin banyak digunakan adalah integrasi dengan Node-RED dan InfluxDB, yang dapat membantu dalam mengelola aliran data sensor, melakukan visualisasi, serta menyimpan data secara efisien.

Dalam praktik ini, digunakan mikrokontroler ESP32 untuk membaca data sensor yang kemudian dikirimkan ke sistem berbasis Node-RED untuk dianalisis dan diteruskan ke InfluxDB sebagai basis penyimpanan data time-series. Proses dilakukan melalui simulasi menggunakan platform Wokwi, dengan alur pengiriman data dirancang agar mencerminkan kondisi nyata dalam sistem IoT.

Studi kasus yang diangkat adalah pengiriman data suhu dan kelembapan, yang kemudian dipantau melalui dasbor visualisasi guna mendeteksi pola atau tren kondisi lingkungan. Hasil praktik menunjukkan bahwa integrasi antara ESP32, Node-RED, dan InfluxDB dapat menjadi solusi cerdas dan fleksibel dalam pengembangan sistem IoT berbasis data.

**Kata kunci:** *IoT, ESP32, Node-RED, InfluxDB, Data Sensor, Wokwi*

## Pendahuluan

Internet of Things (IoT) adalah sebuah konsep yang memungkinkan perangkat elektronik untuk terhubung dan bertukar informasi melalui jaringan internet. Salah satu penerapan utama dari IoT adalah pemantauan kondisi lingkungan secara langsung, seperti suhu dan kelembapan. Dalam sektor industri, pengawasan terhadap suhu dan kelembapan sangat krusial untuk memastikan kelancaran proses produksi serta penyimpanan.

Node-RED merupakan alat berbasis pemrograman alur yang mempermudah integrasi serta pengolahan data dari sensor dalam sistem IoT. Sementara itu, InfluxDB adalah jenis basis data time-series yang ideal untuk menyimpan data lingkungan yang bersifat terus-menerus. Melalui kombinasi kedua teknologi ini, pemantauan suhu dan kelembapan dapat disimulasikan dengan cara yang efisien.

## Metodologi

Metodologi yang digunakan dalam praktik ini terdiri dari beberapa tahapan mulai dari perancangan sistem, pemrograman mikrokontroler, pengumpulan data sensor, integrasi sistem monitoring, hingga penyimpanan dan visualisasi data. Praktik dilakukan secara simulatif menggunakan Wokwi sebagai platform simulasi perangkat keras, Node-RED sebagai alat pemrosesan dan visualisasi data, serta InfluxDB sebagai basis data time-series untuk menyimpan data lingkungan. Berikut tahapan secara rinci:

### 1. Perancangan Sistem

Pada tahap awal dilakukan perancangan sistem konseptual, yang melibatkan komponen-komponen berikut:

- ESP32 Devkit V1: Sebagai mikrokontroler yang membaca data dari sensor.
- Sensor DHT22: Digunakan untuk mengukur suhu dan kelembapan secara real-time.
- Wokwi Simulator: Platform untuk menyimulasikan ESP32 dan sensor secara virtual tanpa perangkat fisik.
- Node-RED (local): Digunakan untuk memproses data dari ESP32 dan menampilkan data dalam dashboard UI.
- InfluxDB: Digunakan sebagai penyimpanan data suhu dan kelembapan secara kontinu dalam format time-series.

### 2. Pemrograman ESP32

ESP32 diprogram langsung di Wokwi menggunakan bahasa C++ (Arduino-style), dengan fungsi utama:

- Inisialisasi sensor DHT22.
- Membaca data suhu dan kelembapan pada interval waktu tertentu.
- Mengirimkan data ke Node-RED melalui koneksi Wi-Fi menggunakan protokol HTTP atau MQTT.

### 3. Integrasi dengan Node-RED

Di sisi lokal, Node-RED digunakan sebagai pusat pemrosesan dan pengelolaan data. Beberapa langkah yang dilakukan:

- Menerima data dari ESP32 melalui node http in atau mqtt in.
- Mengolah data menggunakan node function untuk memformat payload.
- Menampilkan data di UI dengan node gauge dan chart dari node-red-dashboard.
- Menghubungkan ke InfluxDB menggunakan node node-red-contrib-influxdb untuk menyimpan data.

### 4. Penyimpanan Data dengan InfluxDB

Data suhu dan kelembapan yang diterima Node-RED disimpan ke dalam InfluxDB dalam format time-series. Proses ini memungkinkan pengguna untuk:

- Menganalisis tren data dari waktu ke waktu.
- Melakukan query terhadap data historis dengan akurasi waktu tinggi.
- Menyediakan data yang siap digunakan untuk sistem analitik lebih lanjut.

### 5. Visualisasi Data

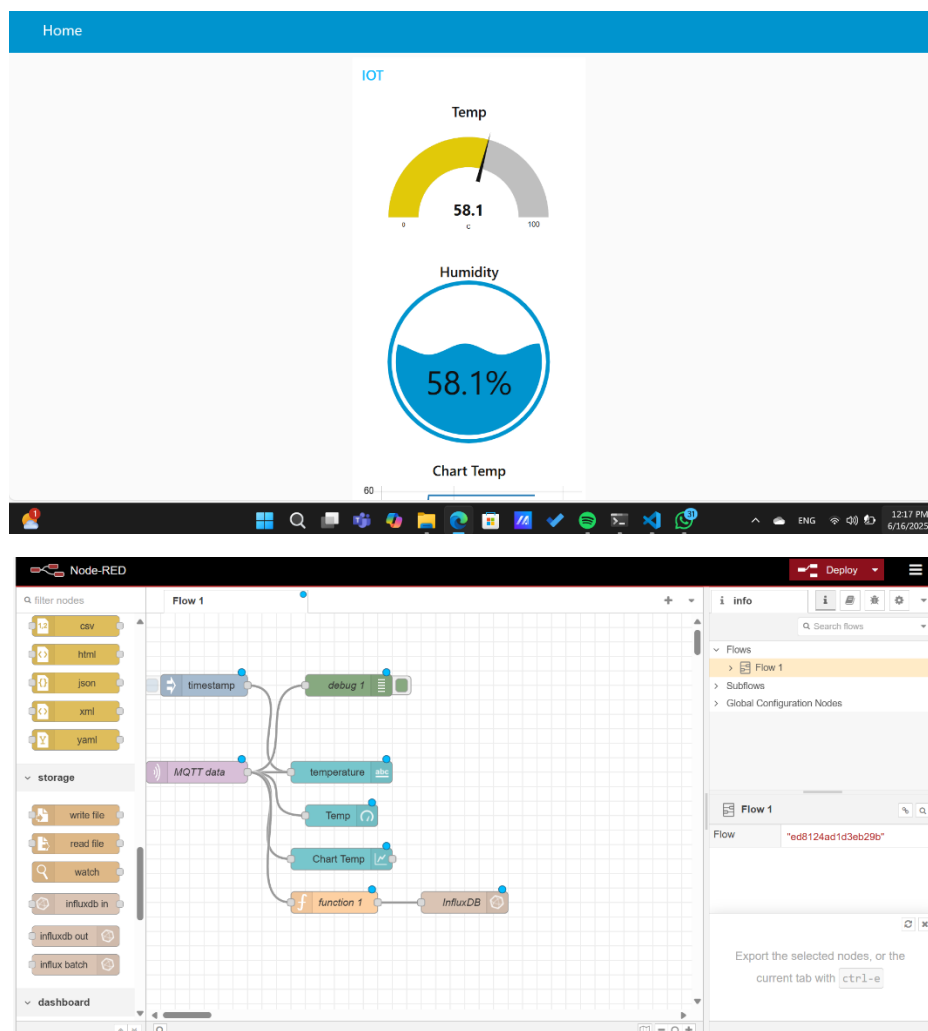
Visualisasi dilakukan melalui dashboard antarmuka pengguna dari node-red-dashboard. Elemen UI yang digunakan meliputi:

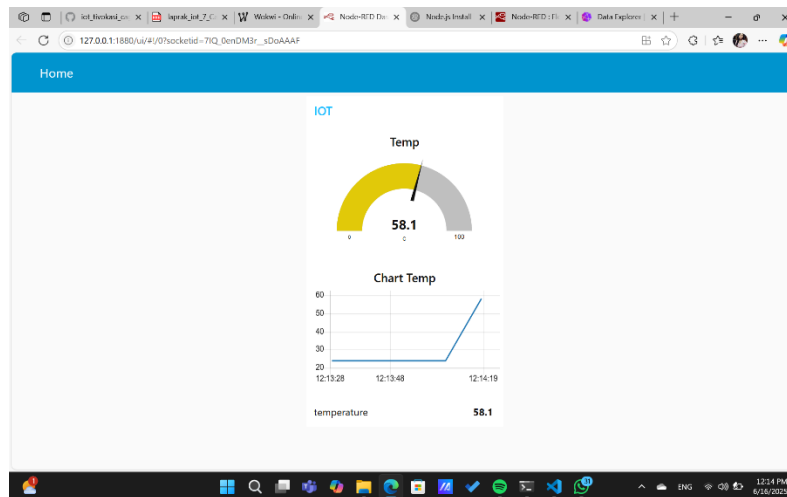
- Gauge: Menampilkan nilai suhu dan kelembapan secara real-time.

- Chart: Menampilkan grafik historis untuk memantau perubahan suhu dan kelembapan dalam jangka waktu tertentu.
6. Uji Coba Sistem
- Pengujian dilakukan secara menyeluruh melalui simulasi di Wokwi dan eksekusi flow di Node-RED.
- Aspek yang diuji meliputi:
- Keberhasilan pembacaan data sensor oleh ESP32.
  - Konektivitas Wi-Fi dan keberhasilan pengiriman data ke Node-RED.
  - Kemampuan Node-RED dalam memproses dan menyimpan data ke InfluxDB.
  - Kualitas visualisasi data secara real-time dan historis pada dashboard.

## Hasil dan Pembahasan

Setelah implementasi, sistem berhasil menerima dan menampilkan data suhu serta kelembapan dari simulator secara real-time dalam bentuk grafik garis pada dashboard. Data tersebut juga tersimpan dengan baik di InfluxDB dalam format time-series, memungkinkan pencatatan perubahan data berdasarkan waktu. Node-RED terbukti mampu memproses data dengan cepat dan efisien berkat latensi yang rendah, sementara InfluxDB mendukung penyimpanan data secara optimal dan mendukung kueri berdasarkan rentang waktu tertentu. Secara keseluruhan, penggunaan Node-RED memudahkan pembuatan alur data tanpa banyak kode, dan InfluxDB sangat sesuai untuk menangani data sensor karena performanya yang tinggi.





## Lampiran

Source code **main.cpp** :

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <DHTesp.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

const int LED_RED = 2;
const int DHT_PIN = 15;
const int BUTTON_PIN = 4;

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;
```

```
unsigned long lastButtonDisplay = 0;
```

```
bool lastButtonState = HIGH;
```

```
bool showButtonMessage = false;
```

```
void setup_wifi() {
```

```
    delay(10);
```

```
    WiFi.mode(WIFI_STA);
```

```
    WiFi.begin(ssid, password);
```

```
    while (WiFi.status() != WL_CONNECTED) {
```

```
        delay(500);
```

```
    }
```

```
    Serial.println("WiFi connected");
```

```
}
```

```
void callback(char* topic, byte* payload, unsigned int length) {
```

```
    if ((char)payload[0] == '1') {
```

```
        digitalWrite(LED_RED, HIGH);
```

```
    } else {
```

```
        digitalWrite(LED_RED, LOW);
```

```
    }
```

```
}
```

```
void reconnect() {
```

```
    while (!client.connected()) {
```

```
        String clientId = "ESP32Client-";
```

```
        clientId += String(random(0xffff), HEX);
```

```
        if (client.connect(clientId.c_str())) {
```

```

        client.publish("IOT/Test1/mqtt", "Test IOT");
        client.subscribe("IOT/Test1/mqtt");
    } else {
        delay(5000);
    }
}
}

void setup() {
    Serial.begin(115200);
    pinMode(LED_RED, OUTPUT);
    pinMode(BUTTON_PIN, INPUT_PULLUP);

    setup_wifi();
    client.setServer(mqtt_server, 1883);
    client.setCallback(callback);
    dht.setup(DHT_PIN, DHTesp::DHT22);

    // OLED setup
    if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
        Serial.println(F("OLED gagal dimulai"));
        while (true);
    }
    display.clearDisplay();
    display.setTextSize(1);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(0, 0);
    display.println("Starting...");
    display.display();
    delay(1000);
}

```

```

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  unsigned long now = millis();

  // Baca tombol
  bool buttonState = digitalRead(BUTTON_PIN);
  if (buttonState == LOW && lastButtonState == HIGH) {
    Serial.println("Tombol ditekan!");
    client.publish("IOT/Test1/button", "Tombol ditekan!");
    digitalWrite(LED_RED, !digitalRead(LED_RED)); // Toggle LED

    showButtonMessage = true;
    lastButtonDisplay = now;
  }
  lastButtonState = buttonState;

  // Baca suhu dan kelembaban tiap 2 detik
  if (now - lastMsg > 2000) {
    lastMsg = now;
    TempAndHumidity data = dht.getTempAndHumidity();

    String tempStr = String(data.temperature, 2);
    String humStr = String(data.humidity, 1);

    client.publish("IOT/Test1/temp", tempStr.c_str());
    client.publish("IOT/Test1/hum", humStr.c_str());

    Serial.println("Temperature: " + tempStr);
  }
}

```

```
Serial.println("Humidity: " + humStr);
```

```
// Tampilkan ke OLED
```

```
display.clearDisplay();
```

```
display.setCursor(0, 0);
```

```
display.print("Temp: ");
```

```
display.print(tempStr);
```

```
display.println(" C");
```

```
display.print("Hum : ");
```

```
display.print(humStr);
```

```
display.println(" %");
```

```
if (showButtonMessage) {
```

```
    display.setCursor(0, 40);
```

```
    display.println("Button Pressed!");
```

```
}
```

```
display.display();
```

```
}
```

```
// Hilangkan tulisan tombol setelah 2 detik
```

```
if (showButtonMessage && (now - lastButtonDisplay > 2000)) {
```

```
    showButtonMessage = false;
```

```
}
```

```
}
```