LAPORAN PRAKTIKUM INTERNET OF THINGS (IOT) PEMANTAUAN SUHU DAN KELEMBAPAN BERBASIS ESP32 MENGGUNAKAN NODE-RED DAN INFLUXDB

FAKULTAS VOKASI UNIVERSITAS BRAWIJAYA

Rifcha Sya'bani Fatullah

Fakultas Vokasi Universitas Brawijaya

Email: rifchasyabani30@gmail.com

ABSTRAK

Dalam praktik ini, ESP32 digunakan untuk memantau suhu dan kelembapan melalui simulasi sensor DHT22 di Wokwi. Data dari sensor dikirim ke Node-RED melalui MQTT untuk diproses dan disimpan di InfluxDB sebagai basis data baris waktu. Selain itu, Node-RED digunakan untuk memberikan visualisasi data secara real-time melalui dashboard. Simulasi ini memungkinkan pengujian sistem Internet of Things tanpa perangkat fisik. Hasil menunjukkan bahwa integrasi ESP32, Node-RED, dan InfluxDB berfungsi dengan baik dalam pengawasan lingkungan. Praktikum ini menunjukkan ide dasar tentang Internet of Things (IoT) yang dapat digunakan untuk aplikasi seperti smart home, pertanian, dan pengendalian kualitas udara dengan sistem yang efisien yang bekerja secara real-time.

Kata Kunci : ESP32, Wokwi, MQTT, Node-Red, InfluxDB

ABTRACT

In this practice, ESP32 is used to monitor temperature and humidity through DHT22 sensor simulation in Wokwi. Data from the sensor is sent to Node-RED via MQTT to be processed and stored in InfluxDB as a time-series database. In addition, Node-RED is used to provide real-time data visualization through a dashboard. This simulation allows testing of the Internet of Things system without physical devices. The results show that the integration of ESP32, Node-RED, and InfluxDB works well in environmental monitoring. This practicum shows the basic idea of the Internet of Things (IoT) that can be used for applications such as smart homes, agriculture, and air quality control with an efficient system that works in real-time.

Keywords: ESP32, Wokwi, MQTT, Node-Red, InfluxDB

1. PENDAHULUAN

1.1 Latar Belakang

Teknologi Internet of Things (IoT) telah mengubah cara kita melihat dan mengendalikan lingkungan kita. Dengan Internet of Things (IoT), perangkat fisik dapat secara otomatis terhubung satu sama lain dan bertukar data melalui jaringan internet, yang memudahkan pengambilan keputusan berbasis data real-time. Pememantauan parameter lingkungan seperti suhu dan kelembapan adalah salah satu aplikasi penting dari Internet of Things, yang berguna dalam berbagai industri seperti pertanian, kesehatan, pengelolaan gedung pintar, dan pengendalian kualitas udara. Dengan IoT, data sensor dapat dikumpulkan dan dianalisis secara otomatis, memberikan informasi yang akurat dan cepat. Ini meningkatkan kualitas pengelolaan lingkungan karena pengukuran manual yang tidak efisien dan kurang akurat untuk kebutuhan yang memerlukan data secara konstan dan real-time. ESP32 adalah modul mikrokontroler yang ideal untuk aplikasi IoT karena memiliki konektivitas Wi-Fi murah dan kuat.

Node-RED adalah platform pengembangan visual berbasis flow yang memungkinkan pengguna mengelola data, membuat logika alur kerja, dan menghubungkan berbagai perangkat dan layanan tanpa perlu menulis kode program yang rumit. Selain itu, InfluxDB adalah database seri waktu yang dibuat untuk menyimpan dan mengelola data berurutan waktu yang besar, seperti data sensor suhu dan kelembapan. Database ini memungkinkan akses dan analisis data masa lalu dengan cepat dan efektif.

Praktikum ini menggunakan Wokwi untuk mempermudah pembelajaran dan pengujian sistem IoT tanpa menggunakan perangkat keras fisik. Simulasi tersebut mensimulasikan pengambilan data dari sensor DHT22 yang terhubung ke ESP32. Selanjutnya, data dikirim ke Node-RED untuk diolah dan disimpan di InfluxDB. Praktikum ini memberikan gambaran menyeluruh tentang implementasi sistem IoT, mulai dari pengambilan data dari sensor hingga penyimpanan dan visualisasi data yang diambil dari sensor.

1.2 Tujuan

Adapun tujuan dari praktikum ini ialah :

1. Memahami bagaimana pemantauan suhu dan kelembapan berbasis ESP32
2. Mengintegrasikan ESP32 dengan Node-Red dan InfluxDB
3. Memahami penyimpanan basis data di InfluxDB

2. METODOLOGI

2.1 Alat dan Bahan

Alat :

a. Laptop
b. Platform Wokwi
c. Platform Visual Studio Code (VSCode)
d. Platform MQTT
e. Platform Node-Red

f. Platform InfluxDB

Bahan :

a. ESP32 (Virtual dalam platform wokwi)
b. Sensor DHT22
c. LED
d. Bahasa Pemrograman C++ Pustaka Arduino
e. Library
    - PubSubClient.h
    - Wifi.h
    - DHTesp.h

2.2 Langkah Implementasi

1. Buka situs wokwi
2. Buat Proyek baru dengan board ESP32
3. Tambahkan sensor DHT22
4. Hubungkan data pin sensor ke GPIO 15
5. Tambahkan LED merah dan hubungkan dengan GPIO 2
6. Buka platform Visual Studio Code
7. Unduh dan install ekstensi PlatformIO dan Wokwi pada VSCode
8. Buat proyek baru di Platform untuk ESP32
9. Buat file baru di root dengan nama “diagram.json”
10. Salin kode diagram.json di wokwi ke VSCode
11. Buat kode program main.cpp
12. Gunakan library Wifi.h, PubSubClient.h dan DHTesp.h
13. Install Node-Red di CMD dengan perintah : npm install -g -unsafe-porm node-red
14. Jalankan Node-Red dengan perintah : node-red
15. Install modul Node-Red dan InfluxDB
16. Buat flow sesuai dengan kebutuhan (node MQTT, node function, node influxdb_out, node ui_gauge atau ui_chart)
17. Buka situs InfluxDB
18. Buat database untuk menyimpan data
19. Dapatkan API Key untuk menintegrasikannya dengan Node-Red
20. Buat Bucket
21. Kembali ke Node-Red, tambahkan konfigurasi InfluxDB dengan menambahkan
    a. URL Server : https://us-east-1-1.aws.cloud2.influxdata.com/
    b. Name: InfluxDB
    c. Server: [v2.0] InfluxDB
    d. Organiation: brawijaya university
    e. Bucket: NodeRed30

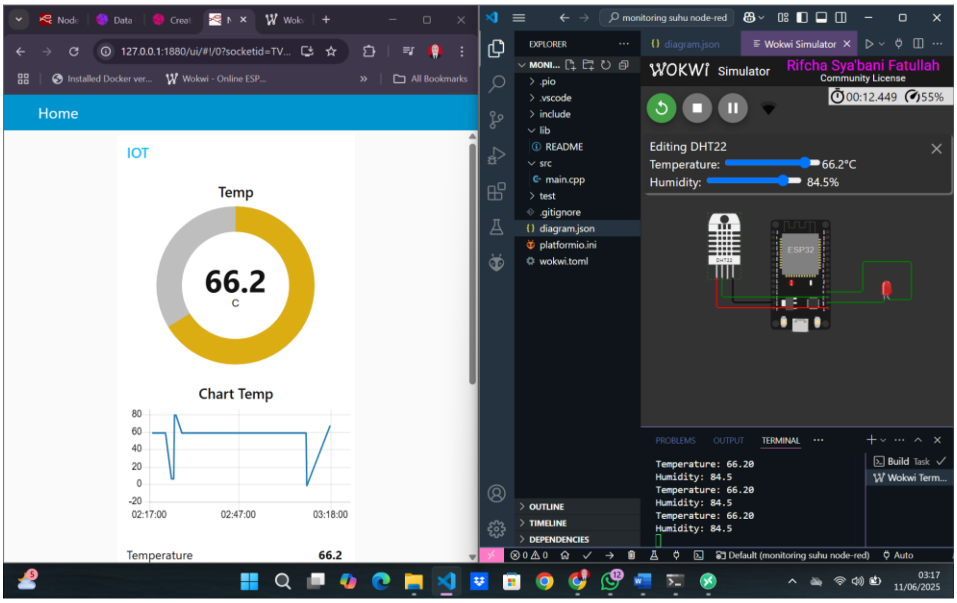
f. Measuremnt: Temp

22. Hubungkan node function yang memproses data ke node InfluxDB out
23. Deploy jika sudah selesai, pastikan terhubung
24. Jalankan program

3. Hasil dan Pembahasan

3.1 Hasil Eksperimen

a. Sistem berhasil terhubung dengan Node-Red (Temp)

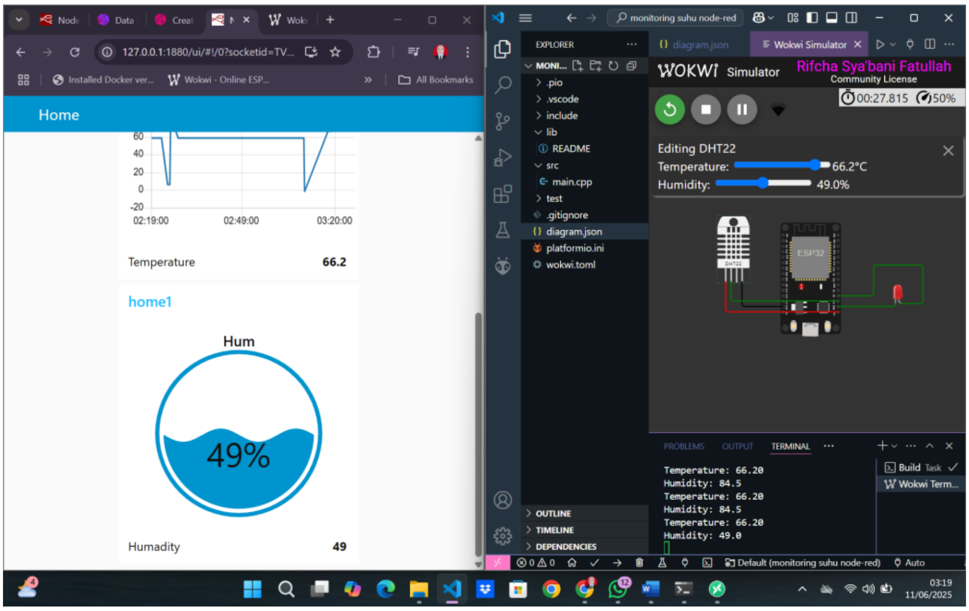


Temperature 66.2

Editing DHT22
Temperature: 66.2°C
Humidity: 84.5%

```
Temperature: 66.20
Humidity: 84.5
Temperature: 66.20
Humidity: 84.5
Temperature: 66.20
Humidity: 84.5
```

b. Sistem berhasil terhubung dengan Node-Red (Hum)



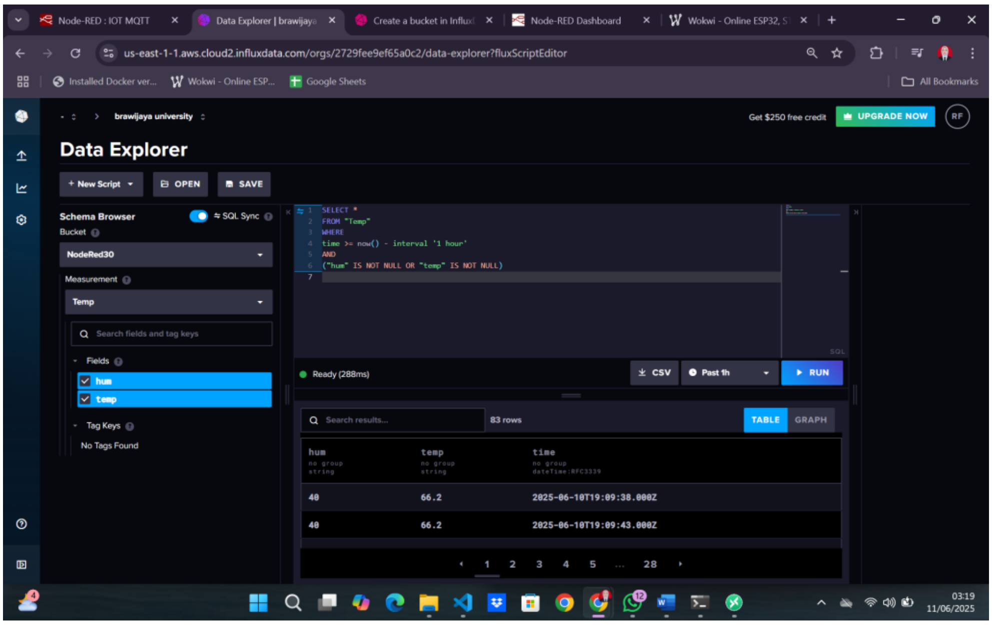
Temperature 66.2

Humadity 49

Editing DHT22
Temperature: 66.2°C
Humidity: 49.0%

```
Temperature: 66.20
Humidity: 84.5
Temperature: 66.20
Humidity: 84.5
Temperature: 66.20
Humidity: 49.0
```

c. Database di InfluxDB



```
SELECT *
FROM "Temp"
WHERE
time >= now() - interval '1 hour'
AND
("hum" IS NOT NULL OR "temp" IS NOT NULL)
```

Ready (288ms)

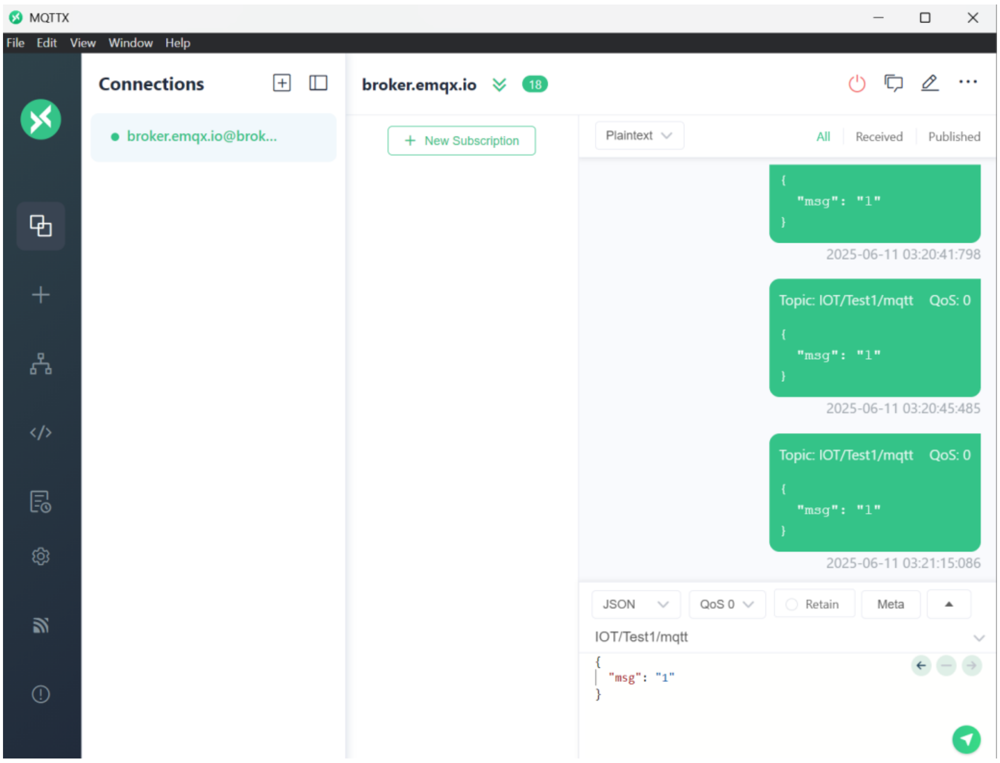
83 rows

hum no group string	temp no group string	time no group dateTime:RFC3339
40	66.2	2025-06-10T19:09:38.000Z
40	66.2	2025-06-10T19:09:43.000Z

‹ 1 2 3 4 5 … 28 ›

03:19
11/06/2025

d. Kirim pesan melalui MQTT



```
{
  "msg": "1"
}
```

2025-06-11 03:20:41:798

Topic: IOT/Test1/mqtt QoS: 0

```
{
  "msg": "1"
}
```

2025-06-11 03:20:45:485

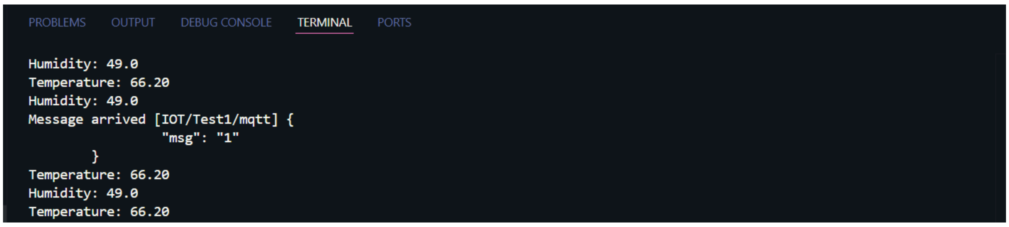
Topic: IOT/Test1/mqtt QoS: 0

```
{
  "msg": "1"
}
```

2025-06-11 03:21:15:086

IOT/Test1/mqtt

```
{
  "msg": "1"
}
```



```
Humidity: 49.0
Temperature: 66.20
Humidity: 49.0
Message arrived [IOT/Test1/mqtt] {
                  "msg": "1"
        }
Temperature: 66.20
Humidity: 49.0
Temperature: 66.20
```

3.2 Pembahasan

Data dari topik MQTT dikumpulkan melalui platform Node-RED, yang kemudian diproses dan ditampilkan secara visual melalui dashboard. Node-RED berfungsi sebagai visualisasi dan juga berfungsi sebagai penghubung ke database InfluxDB, yang dapat menyimpan data secara historis dalam format baris waktu. Nilai suhu dan kelembapan ditunjukkan secara real-time dalam bentuk grafik. Hal ini memungkinkan pengguna melihat kondisi secara langsung dan melihat perubahan suhu dan kelembapan dari waktu ke waktu. InfluxDB cocok untuk proyek ini karena mampu menangani data berurutan dan kontinu. Struktur data yang dikirim ke InfluxDB diformat dalam Node-RED untuk mematuhi ketentuan, yang termasuk pengukuran (metrik), nilai (kolom), dan waktu. Arsitektur sistem ini memungkinkan seluruh komponen Internet of Things, mulai dari perangkat sensor, komunikasi, pengolahan, penyimpanan, dan visualisasi, untuk berintegrasi dengan baik.

Simulasi ini menunjukkan bahwa sistem dapat beroperasi secara real-time, stabil, dan berfungsi tanpa perangkat keras fisik. Data yang diambil dari sensor dapat dikirim, disimpan, dan dikomunikasikan dengan baik. Praktikum ini telah memberikan gambaran lengkap tentang bagaimana teknologi Internet of Things dapat digunakan untuk memantau lingkungan, meskipun ada keterbatasan karena lingkungan simulasi tidak sepenuhnya mencerminkan keadaan nyata.

4. Lampiran

a. Kode Program Main.cpp

```
#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 = "broker.emqx.io";//"test.mosquitto.org";//

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 sampi 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++) {
    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);
  } else {
    digitalWrite(LED_RED, LOW);
  }
}

void reconnect() {
  // 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");
      // ... 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);     // inisialisasi 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);//inisialiasi 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);
  }
}
```

b. Kode Program diagram.json

```
{
  "version": 1,
  "author": "Rifcha Sya'bani Fatullah",
  "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": {}
}
```

c. **Kode flows_temp.json**

```
[
{
    "id": "8b88f65241fd2f4e",
    "type": "tab",
    "label": "IOT MQTT",
    "disabled": false,
    "info": "",
    "env": []
  },
  {
    "id": "b7be3fd5d19d83a0",
    "type": "mqtt in",
    "z": "8b88f65241fd2f4e",
    "name": "MQTT data",
    "topic": "IOT/Test1/temp",
    "qos": "0",
    "datatype": "auto-detect",
    "broker": "fd4cbcbcd29913ab",
    "nl": false,
    "rap": true,
    "rh": 0,
    "inputs": 0,
    "x": 390,
    "y": 340,
    "wires": [
      [
        "02e211b7bb351f90",
        "5b6b7f5c5d77d1a6",
                "f933b932defe4689",
                "1de42b0ad1ab856b",
                "bd388ae6b35881f9"
            ]
        ]
    },
    {
        "id": "6dc2c7101b73f23e",
        "type": "inject",
        "z": "8b88f65241fd2f4e",
        "name": "",
        "props": [
            {
                "p": "payload"
            },
            {
                "p": "topic",
                "vt": "str"
            }
        ],
        "repeat": "",
        "crontab": "",
        "once": false,
        "onceDelay": 0.1,
        "topic": "",
        "payload": "",
        "payloadType": "date",
        "x": 380,
        "y": 200,
        "wires": [
            [
                "02e211b7bb351f90"
            ]
        ]
    },
    {
        "id": "02e211b7bb351f90",
        "type": "ui_text",
        "z": "8b88f65241fd2f4e",
        "group": "6cb91646811ccc32",
        "order": 0,
        "width": 0,
        "height": 0,
        "name": "",
        "label": "Temperature",
        "format": "{{msg.payload}}",
        "layout": "row-spread",
        "className": "",
        "style": false,
        "font": "",
        "fontSize": 16,
        "color": "#000000",
        "x": 650,
        "y": 320,
        "wires": []
    },
    {
        "id": "5b6b7f5c5d77d1a6",
        "type": "ui_gauge",
        "z": "8b88f65241fd2f4e",
        "name": "",
        "group": "6cb91646811ccc32",
        "order": 1,
        "width": 0,
        "height": 0,
        "gtype": "donut",
        "title": "Temp",
        "label": "C",
        "format": "{{value}}",
        "min": 0,
        "max": "100",
        "colors": [
        "#00b500",
        "#e6e600",
        "#ca3838"
      ],
      "seg1": "",
      "seg2": "",
      "diff": false,
      "className": "",
      "x": 630,
      "y": 360,
      "wires": []
    },
    {
      "id": "f933b932defe4689",
      "type": "ui_chart",
      "z": "8b88f65241fd2f4e",
      "name": "",
      "group": "6cb91646811ccc32",
      "order": 2,
      "width": 0,
      "height": 0,
      "label": "Chart Temp",
      "chartType": "line",
      "legend": "false",
      "xformat": "HH:mm:ss",
      "interpolate": "linear",
      "nodata": "",
      "dot": false,
      "ymin": "",
      "ymax": "",
      "removeOlder": 1,
      "removeOlderPoints": "",
      "removeOlderUnit": "3600",
      "cutout": 0,
      "useOneColor": false,
      "useUTC": false,
        "colors": [
            "#1f77b4",
            "#aec7e8",
            "#ff7f0e",
            "#2ca02c",
            "#98df8a",
            "#d62728",
            "#ff9896",
            "#9467bd",
            "#c5b0d5"
        ],
        "outputs": 1,
        "useDifferentColor": false,
        "className": "",
        "x": 650,
        "y": 400,
        "wires": [
            []
        ]
    },
    {
        "id": "1de42b0ad1ab856b",
        "type": "debug",
        "z": "8b88f65241fd2f4e",
        "name": "debug 1",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "payload",
        "targetType": "msg",
        "statusVal": "",
        "statusType": "auto",
        "x": 640,
        "y": 240,
        "wires": []
    },
    {
        "id": "bd388ae6b35881f9",
        "type": "function",
        "z": "8b88f65241fd2f4e",
        "name": "function 1",
        "func": "var xx = msg.payload;\nvar Newobject = {};\nNewobject = {\n    \"temp\": msg.payload.toString()\n}\nmsg.payload = Newobject;\nreturn msg;\n",
        "outputs": 1,
        "timeout": 0,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 640,
        "y": 480,
        "wires": [
            [
                "efc79a152f28c53b"
            ]
        ]
    },
    {
        "id": "efc79a152f28c53b",
        "type": "influxdb out",
        "z": "8b88f65241fd2f4e",
        "influxdb": "2fe9efe09dfa8dfd",
        "name": "InfluxDB",
        "measurement": "Temp",
        "precision": "",
        "retentionPolicy": "",
        "database": "database",
        "precisionV18FluxV20": "s",
        "retentionPolicyV18Flux": "",
        "org": "organisation",
        "bucket": "NodeRed",
        "x": 840,
        "y": 480,
        "wires": []
    },
    {
        "id": "fd4cbcbcd29913ab",
        "type": "mqtt-broker",
        "name": "",
        "broker": "broker.emqx.io",
        "port": 1883,
        "clientid": "",
        "autoConnect": true,
        "usetls": false,
        "protocolVersion": 4,
        "keepalive": 60,
        "cleansession": true,
        "autoUnsubscribe": true,
        "birthTopic": "",
        "birthQos": "0",
        "birthRetain": "false",
        "birthPayload": "",
        "birthMsg": {},
        "closeTopic": "",
        "closeQos": "0",
        "closeRetain": "false",
        "closePayload": "",
        "closeMsg": {},
        "willTopic": "",
        "willQos": "0",
        "willRetain": "false",
        "willPayload": "",
        "willMsg": {},
        "userProps": "",
        "sessionExpiry": ""
    },
    {
```

```
    "id": "6cb91646811ccc32",
    "type": "ui_group",
    "name": "IOT",
    "tab": "acc17c2264463766",
    "order": 1,
    "disp": true,
    "width": 6,
    "collapse": false,
    "className": ""
  },
  {
    "id": "2fe9efe09dfa8dfd",
    "type": "influxdb",
    "hostname": "127.0.0.1",
    "port": 8086,
    "protocol": "http",
    "database": "database",
    "name": "InfluxDB",
    "usetls": false,
    "tls": "",
    "influxdbVersion": "2.0",
    "url": "https://us-east-1-1.aws.cloud2.influxdata.com/",
    "timeout": 10,
    "rejectUnauthorized": true
  },
  {
    "id": "acc17c2264463766",
    "type": "ui_tab",
    "name": "Home",
    "icon": "dashboard",
    "disabled": false,
    "hidden": false
  }
]
```

d. Kode flows_hum.json

```
[
  {
    "id": "9ee437252d2b0c81",
    "type": "tab",
    "label": "test",
    "disabled": false,
    "info": "",
    "env": []
  },
  {
    "id": "f1172be69061b358",
    "type": "mqtt in",
    "z": "9ee437252d2b0c81",
    "name": "MQTT data",
    "topic": "IOT/Test1/hum",
    "qos": "0",
    "datatype": "auto-detect",
    "broker": "fd4cbcbcd29913ab",
    "nl": false,
    "rap": true,
    "rh": 0,
    "inputs": 0,
    "x": 230,
    "y": 260,
    "wires": [
      [
        "4db5b2a12e92f52f",
        "3fbf3dbcbd4b18e1",
        "9fc136140df8a5b4",
        "3923030809a0bd4a"
      ]
    ]
  },
  {
        "id": "0b9c798ef7faf830",
        "type": "inject",
        "z": "9ee437252d2b0c81",
        "name": "",
        "props": [
            {
                "p": "payload"
            },
            {
                "p": "topic",
                "vt": "str"
            }
        ],
        "repeat": "",
        "crontab": "",
        "once": false,
        "onceDelay": 0.1,
        "topic": "",
        "payload": "",
        "payloadType": "date",
        "x": 220,
        "y": 120,
        "wires": [
            [
                "4db5b2a12e92f52f"
            ]
        ]
    },
    {
        "id": "4db5b2a12e92f52f",
        "type": "ui_text",
        "z": "9ee437252d2b0c81",
        "group": "66596c22c53900d8",
        "order": 0,
        "width": 0,
        "height": 0,
      "name": "",
      "label": "Humadity",
      "format": "{{msg.payload}}",
      "layout": "row-spread",
      "className": "",
      "style": false,
      "font": "",
      "fontSize": 16,
      "color": "#000000",
      "x": 480,
      "y": 240,
      "wires": []
    },
    {
      "id": "3fbf3dbcbd4b18e1",
      "type": "ui_gauge",
      "z": "9ee437252d2b0c81",
      "name": "",
      "group": "66596c22c53900d8",
      "order": 1,
      "width": 0,
      "height": 0,
      "gtype": "wave",
      "title": "Hum",
      "label": "%",
      "format": "{{value}}",
      "min": 0,
      "max": "100",
      "colors": [
        "#00b500",
        "#e6e600",
        "#ca3838"
      ],
      "seg1": "",
      "seg2": "",
      "diff": false,
        "className": "",
        "x": 470,
        "y": 280,
        "wires": []
    },
    {
        "id": "9fc136140df8a5b4",
        "type": "debug",
        "z": "9ee437252d2b0c81",
        "name": "debug 2",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "payload",
        "targetType": "msg",
        "statusVal": "",
        "statusType": "auto",
        "x": 480,
        "y": 160,
        "wires": []
    },
    {
        "id": "3923030809a0bd4a",
        "type": "function",
        "z": "9ee437252d2b0c81",
        "name": "function 2",
        "func": "var xx = msg.payload;\nvar Newobject = {};\nNewobject = {\n    \"hum\": msg.payload.toString()\n}\nmsg.payload = Newobject;\nreturn msg;\n",
        "outputs": 1,
        "timeout": 0,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 480,
        "y": 400,
        "wires": [
            [
                "95ab26848d5e0e2e"
            ]
        ]
    },
    {
        "id": "95ab26848d5e0e2e",
        "type": "influxdb out",
        "z": "9ee437252d2b0c81",
        "influxdb": "2fe9efe09dfa8dfd",
        "name": "InfluxDB",
        "measurement": "Temp",
        "precision": "",
        "retentionPolicy": "",
        "database": "database",
        "precisionV18FluxV20": "s",
        "retentionPolicyV18Flux": "",
        "org": "organisation",
        "bucket": "NodeRed",
        "x": 680,
        "y": 400,
        "wires": []
    },
    {
        "id": "fd4cbcbcd29913ab",
        "type": "mqtt-broker",
        "name": "",
        "broker": "broker.emqx.io",
        "port": 1883,
        "clientid": "",
        "autoConnect": true,
        "usetls": false,
        "protocolVersion": 4,
        "keepalive": 60,
      "cleansession": true,
      "autoUnsubscribe": true,
      "birthTopic": "",
      "birthQos": "0",
      "birthRetain": "false",
      "birthPayload": "",
      "birthMsg": {},
      "closeTopic": "",
      "closeQos": "0",
      "closeRetain": "false",
      "closePayload": "",
      "closeMsg": {},
      "willTopic": "",
      "willQos": "0",
      "willRetain": "false",
      "willPayload": "",
      "willMsg": {},
      "userProps": "",
      "sessionExpiry": ""
    },
    {
      "id": "66596c22c53900d8",
      "type": "ui_group",
      "name": "Home",
      "tab": "33e1fa2b35d5f28e",
      "order": 1,
      "disp": true,
      "width": 6,
      "collapse": false,
      "className": ""
    },
    {
      "id": "2fe9efe09dfa8dfd",
      "type": "influxdb",
      "hostname": "127.0.0.1",
      "port": 8086,
        "protocol": "http",
        "database": "database",
        "name": "InfluxDB",
        "usetls": false,
        "tls": "",
        "influxdbVersion": "2.0",
        "url": "https://us-east-1-1.aws.cloud2.influxdata.com/",
        "timeout": 10,
        "rejectUnauthorized": true
    },
    {
        "id": "33e1fa2b35d5f28e",
        "type": "ui_tab",
        "name": "IOT",
        "icon": "dashboard",
        "disabled": false,
        "hidden": false
    }
]
```