LAPORAN PRAKTIKUM: SIMULASI MONITORING SUHU DAN KELEMBAPAN DENGAN NODE-RED DAN INFLUXDB

TB Rangga Gilang Yanuari
Fakultas Vokasi, Universitas Brawijaya
gilangyanuarirangga@gmail.com

ABSTRAK

Sistem monitoring suhu dan kelembapan berbasis Internet of Things (IoT) memungkinkan pemantauan kondisi lingkungan secara real-time dan historis. Dalam praktikum ini digunakan platform Node-RED untuk alur pemrosesan data, InfluxDB sebagai penyimpanan basis data time-series, dan Dashboard UI untuk visualisasi. Data disimulasikan menggunakan node inject atau dapat diterima dari sensor DHT. Node-RED mengatur alur data mulai dari input, pemrosesan, penyimpanan ke InfluxDB, dan visualisasi melalui chart serta gauge. Hasil praktik menunjukkan sistem bekerja optimal untuk pengumpulan, penyimpanan, dan penyajian data suhu dan kelembapan secara realtime dan historis.

Kata Kunci: IoT, Node-RED, InfluxDB, suhu, kelembapan, dashboard, time-series

ABSTRACT

Temperature and humidity monitoring systems based on Internet of Things (IoT) enable real-time and historical environmental tracking. In this practicum, we use Node-RED for data flow control, InfluxDB as a time-series database, and a UI Dashboard for visualization. Data is simulated via inject nodes or real sensors and processed through Node-RED to be stored in InfluxDB and displayed in real time. The result confirms that the system successfully collects, stores, and displays temperature and humidity data effectively.

Keywords: IoT, Node-RED, InfluxDB, temperature, humidity, dashboard, time-series

1. PENDAHULUAN

1.1 Latar Belakang

Dalam era IoT, data lingkungan seperti suhu dan kelembapan sangat krusial. Penyimpanan data historis tidak hanya memungkinkan pemantauan saat ini, tapi juga analisis tren cuaca. Node-RED menyediakan alur pemrograman visual untuk membaca dan memproses data. Sedangkan InfluxDB, sebagai database time-series, dirancang untuk menyimpan data berbasis waktu seperti suhu dan kelembapan. Kombinasi keduanya mendukung sistem monitoring yang handal dan efisien.

1.2 Tujuan Praktikum

Praktikum ini bertujuan untuk:

1) Menyimulasikan pengumpulan data suhu dan kelembapan
2) Menyimpan data tersebut ke dalam InfluxDB

3) Menampilkan data secara real-time dan historis melalui dashboard
4) Menggunakan Node-RED sebagai orchestrator antar komponen

2. METODOLOGI

2.1 Alat dan Bahan

- **Alat:**
  a) Laptop/computer
  b) Wokwi Simulator
  c) Node-RED (local)
  d) InfluxDB
- **Bahan:**
  a) ESP32 Devkit V1.
  b) Sensor DHT22
  c) Koneksi Wi-Fi (Wokwi Guest)
  d) Node-RED nodes tambahan seperti:
    - node-red-dashboard (untuk UI)
    - node-red-contrib-influxdb (untuk koneksi ke InfluxDB)
    - function, inject, gauge, chart, dll

2.2 Langkah Implementasi

Berikut adalah langkah-langkah implementasi secara rinci dalam melakukan simulasi sistem monitoring suhu dan kelembapan berbasis ESP32 dan DHT22 menggunakan Wokwi, Node-RED, dan InfluxDB:

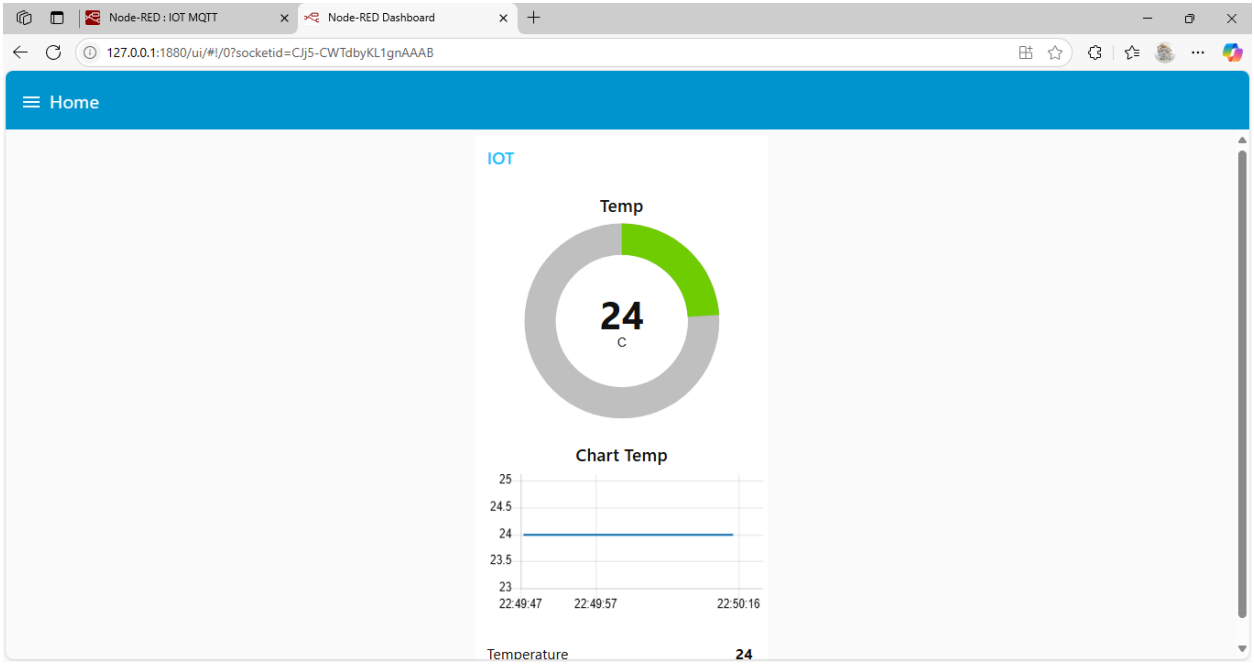
1. **Simulasi di Wokwi Web**
   1) Buka https://wokwi.com.
   2) Klik **"New Project"**, pilih **ESP32** sebagai board utama.
   3) Tambahkan komponen:
      a) ESP32 Devkit V1
        - ESP32 DevKit V1
        - Sensor DHT22
   4) Tulis dan upload kode program di Wokwi Web.
        - VCC DHT22 → 3.3V ESP32
        - GND DHT22 → GND ESP32
        - DATA DHT22 → GPIO15 ESP32
   5) Tambahkan dan tulis kode program untuk membaca data suhu dan kelembapan, lalu kirimkan via WiFi ke Node-RED (gunakan protokol MQTT atau HTTP).
   6) Klik **"Start Simulation"** untuk memulai simulasi.

7) Pastikan data suhu dan kelembapan berhasil terbaca dan terkirim dari ESP32.

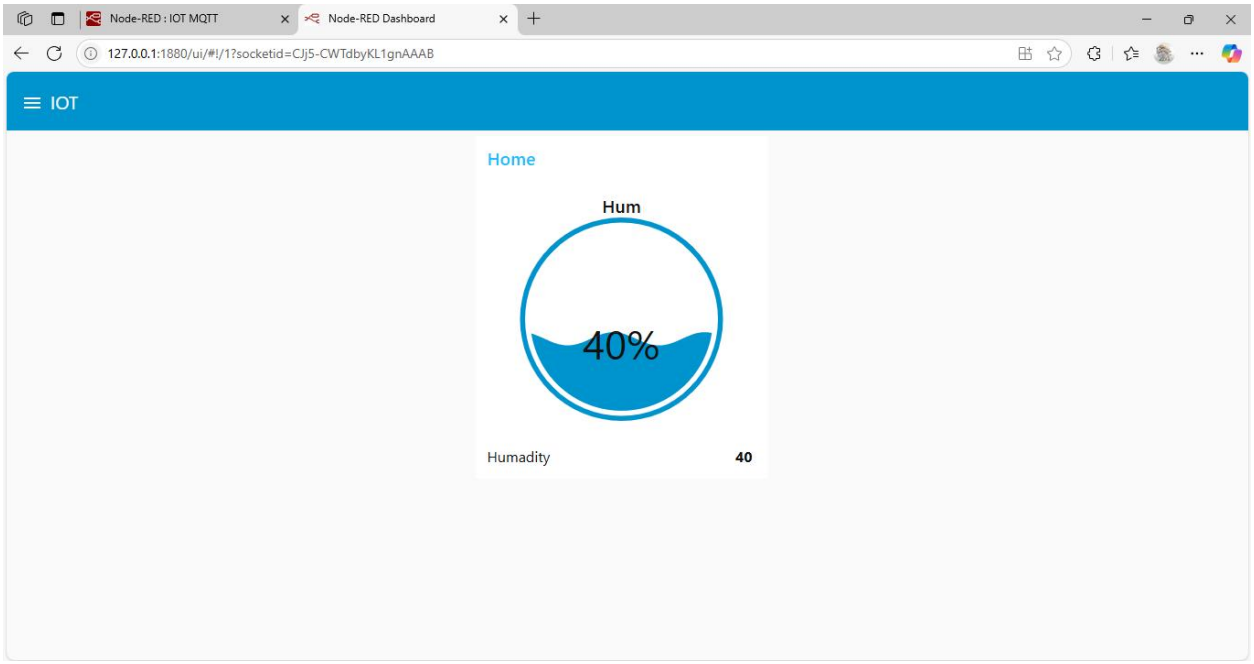
2. **Konfigurasi Node-RED**
   1) Jalankan Node-RED di laptop (via localhost:1880).
   2) Buat flow baru:
      - Gunakan node **MQTT in**
      - Gunakan node **function** untuk parsing data suhu dan kelembapan.
      - Gunakan node **influxdb out** untuk menyimpan data ke InfluxDB.
      - Gunakan node **ui_gauge** atau **ui_chart** untuk menampilkan data suhu & kelembapan secara real-time.
3. **Setup InfluxDB**
   1) Install dan jalankan InfluxDB (local atau Docker).
   2) Buat database baru (misalnya: sensor_data).
   3) Konfigurasikan node InfluxDB di Node-RED:
      - Name: InfluxDB
      - Server: [v2.0] InfluxDB
      - Organiation: Organisation
      - Bucket: NodeRed
      - Measuremnt: Temp
   4) Pastikan data dari Node-RED masuk ke InfluxDB.

3. HASIL DAN PEMBAHASAN

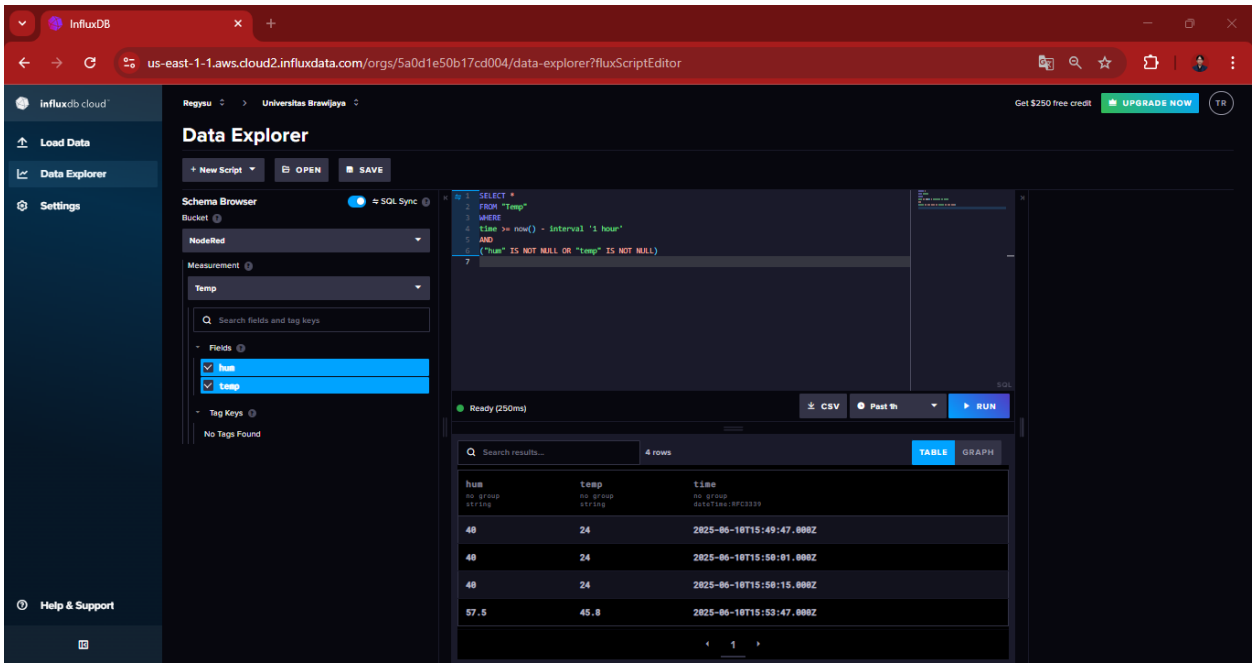
3.1 Hasil UI Temperature di NodeRed



3.2 Hasil UI Humadity di NodeRed



3.3 Hasil Database menggunakan InfluxDB



3.6 Pembahasan

Dalam proyek ini, digunakan Wokwi sebagai simulator mikrokontroler ESP32 yang terhubung dengan sensor DHT22 untuk membaca data suhu dan kelembapan secara real-time. Data yang diperoleh kemudian dikirimkan ke Node-RED, sebuah platform berbasis alur (flow-based) yang memudahkan integrasi antarperangkat IoT tanpa perlu banyak

penulisan kode. Di dalam Node-RED, data ditampilkan secara visual melalui dashboard serta disimpan ke InfluxDB, sebuah database time-series yang dirancang khusus untuk merekam data sensor dalam interval waktu tertentu. Kombinasi ketiganya memberikan solusi simulasi IoT yang efisien dan praktis tanpa memerlukan perangkat keras fisik.

4. LAMPIRAN

4.1 Kode program Wokwi Temp dan Hum

```
#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++) { //mengecek jumlah data yang ada di topik
mqtt
    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);   // Turn the LED on
  } else {
    digitalWrite(LED_RED, LOW);  // Turn the LED off
  }
}

void reconnect() { //perintah koneksi esp32 ke mqtt broker baik itu sebagai
publusher atau subscriber
  // 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"); //perintah publish data
ke alamat topik yang di setting
      // ... 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);
  }
}
```

4.2 Kode diagram.json

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

4.3 Kode Json NodeRed Temperature

[{"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,"classNam e":"","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}]

4.4 Kode Json NodeRed Humadity

[{"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","typ

e":"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","classNname":"","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,"classNname":"","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,"classNname":""},{"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}]