MQTT Protocol dengan ESP32

Pada percobaan kali ini kita akan memulai membuat sistem IoT sederhana, yang akan kita gunakan untuk mengirimkan data sensor ke server yang kita kembangkan sendiri. Ada beberapa tahapan yang harus kita lakukan yaitu (1). Membuat platform yang akan mengirim data ke server, kita akan membuatnya dengn ESP32. (2). Membuat infrastruktur untuk menangkap data kita dari sisi server, kita akan menggunakan Node-red, dan (3). Kita akan membuat infrastruktur MQTT, dengan Mosquitto. dan yang terakhir kita akan menampilkan visualisasi datanya dalam dashboard.

> EL5057 | Sistem Penginderaan

Instalasi Node-red di *Virtual Machine*

Memastikan virtual machine kita up to date dengan menggunakan perintah berikut:

```
sudo apt update
sudo apt upgrade
```

Instalasi node-red versi terakhir, membutuhkan nodejs versi 18, untuk itu kita harus menginstall didalam VM kita dengan menggunakan perintah sebagai berikut.

```
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
```

Setelah installer terdownload dengan `curl`, kemudian kita jalankan proses instalasi dengan perintah sebagai berikut.

```
sudo apt-get install -y nodejs build-essential
```

Setelah instalasi nodejs selesai, kita cek versi dari nodejs yang telah kita install dengan perintah berikut.

```
nodejs --version
```

Install node-red, dengan command `npm` seperti di bawah ini.

```
sudo npm install -g --unsafe-perm node-red
```

Node-Red dijalankan dengan perintah `node-red`. Namun ketika terminal kita close, maka node-red akan close, karena kita jalankan di *foreground*. Untuk mengelola proses jalannya node-red background dan menjalankan node-red kembali jika mengalami crach, biasanya kita menggunakan PM2. PM2 adalah sebuah perangkat lunak **open-source** yang berfungsi sebagai **process manager** untuk aplikasi Node.js. PM2 memungkinkan kita untuk menjalankan aplikasi Node.js di background, bahkan ketika ketika kita menutup terminal, PM2 juga dilengkapi beberapa fitur untuk memonitor penggunaan CPU dan memory. Berikut cara instalasi PM2 di VM kita.

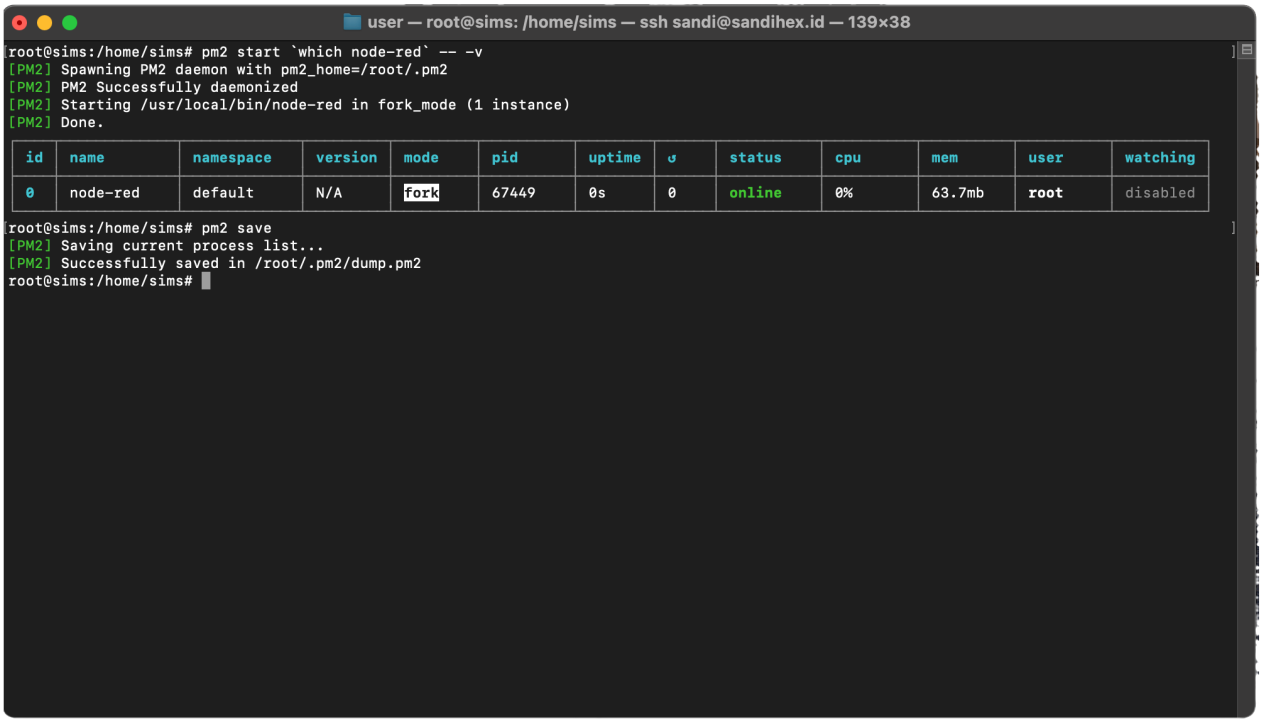
```
sudo npm install -g --unsafe-perm pm2
```

Jalankan Node-Red dengan PM2 di VM kita melalui perintah sebagai berikut.

```
pm2 start `which node-red` -- -v
```

Simpan session dengan perintah berikut.

```
pm2 save
```



Untuk melihat seluruh proses yang di handle oleh PM2, dapat dengan menggunakan perintah berikut.

```
pm2 list
```

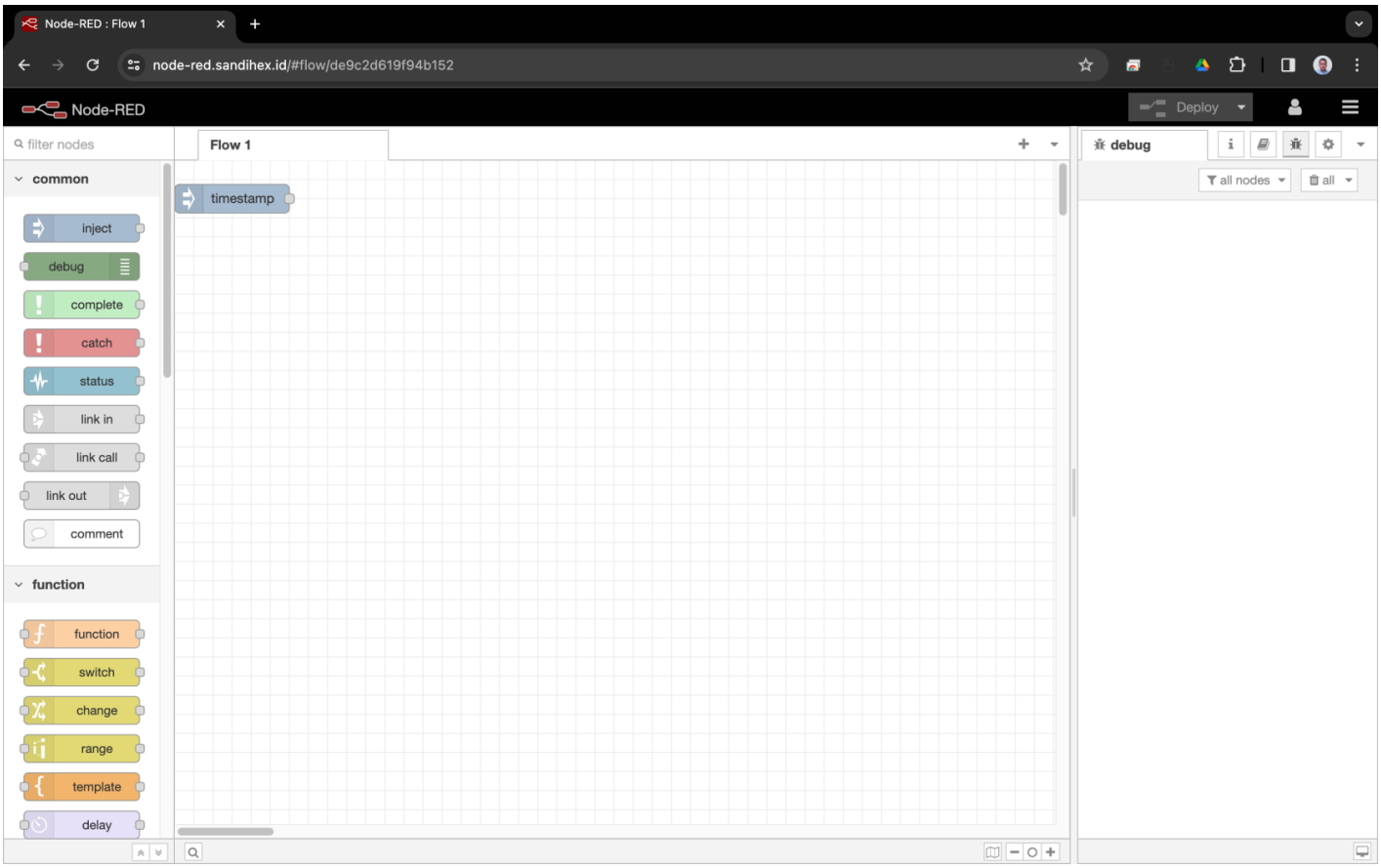
Untuk merestart proses, misalnya ketika kita selesai melakukan konfigurasi node-red, dengan menggunakan perintah berikut.

```
pm2 restart node-red
```

Membuat password untuk credentials node-red, dengan perintah berikut.

```
node-red admin hash-pw
```

Ok, sekarang kita sudah menginstall Node-Red. Standarnya Node-red diakses dari localhost atau 127.0.0.1 di port 1880 untuk kepentingan akses jarak jauh, kita menggunakan reverse-proxy untuk mengakses node-red yang telah kita install. Kita menggunakan web-server Nginx untuk membuat reverse-proxy, kemudian untuk memudahkan mengakses Node-red tersebut, kita binding dengan domain developer kita yaitu sandihex.id, sehingga kita bisa mengaksesnya dengan alamat https://node-red.sandihex.id .



Instalasi MQTT - Server

Fuih, panjang juga jalannya ya... baiklah kita lanjutkan, kali ini kita menginstall MQTT broker sebagai salah satu primadona gateway IoT. Sedikit pengalaman tentang MQTT ini, pertama kali saya berkenalan dengan sistem ini adalah ketika membantu **Prof. Masaaki Wada** dari Future University of Hakodate dalam membangun Smart Sistem untuk Kelautan dan Perikanan. Beliau menyebutkan bahw MQTT yang saya buat harus di jaga keamanannya dengan menggunakan CA certificate. Terima kasih Sensei...

Jadi apa ini MQTT, gampangannya MQTT server ini adalah sebuah kaca yang memantulkan informasi dari alat sensor dilapangan ke siapa saja yang membutuhkan informasi tersebut. Namun syaratnya kita harus memonitor informasi tersebut di topik yang sama. Nah kali ini kita akan membuat MQTT server dengan Mosquitto, kemudian karena untuk kepentingan percobaan, kita tidak akan menggunakan fitur keamanan.

Install Mosquitto server dengan perintah sebagai berikut.

```
sudo apt-get install mosquitto mosquitto-clients
```

Setelah terinstall, kita membuat file konfigurasi di path `/etc/mosquitto/mosquitto.conf` dengan konfigurasi sebagai berikut.

```
allow_anonymous true
listener 0.0.0.0 1883
```

Restart service mosquitto dengan perintah sebagai berikut.

```
systemctl restart mosquitto
```

Untuk mengirim pesan (publish) gunakan perintah berikut.

```
mosquitto_pub -h mqtt2.sandihex.id -p 1883 -t "test" -m "Hello world"
```

Untuk menerima pesan (subscribe) gunakan perintah sebagai berikut.

```
mosquitto_sub -h mqtt2.sandihex.id -p 1883 -t "test"
```

Selanjutnya kita tes, hasil konfigurasi sistem IoT kita seperti video berikut ini.

https://www.youtube.com/embed/8UiZvGoxW6Q

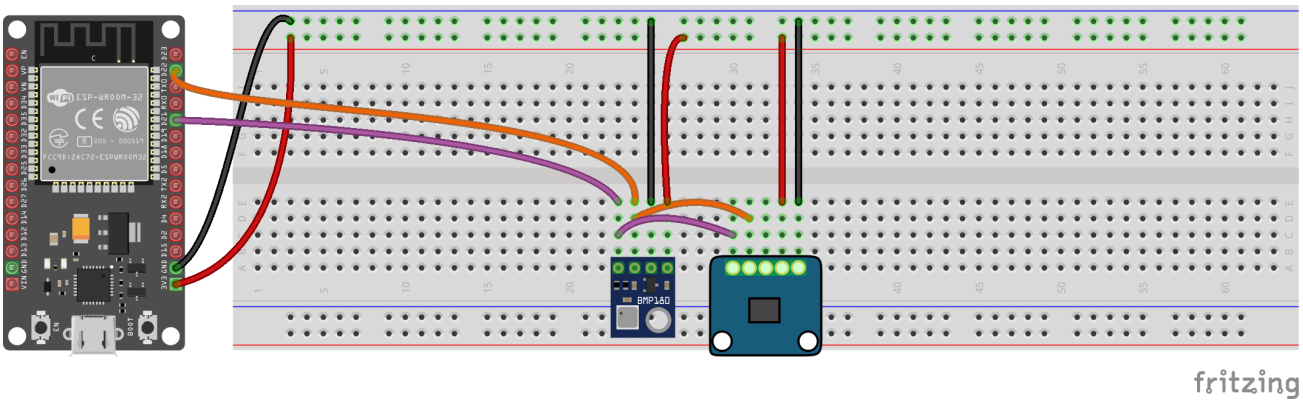
ESP 32 - MQTT

Let's practice with ESP32....

Komponen yang kita butuhkan untuk percobaan kali ini adalah sebagai berikut.

No.	Nama Komponen
1	Dev Board ESP-32
2	Jumper wires
3	Breadboard
4	Sensor HTU21DF
5	Sensor BMP 180

Skema rangkaian yang kita gunakan seperti ditampilkan di bawah ini, semua sensor dihubungkan melalui protokol I2C dalam satu bus karena memiliki address yang berbeda.



Sketch Arduino IDE seperti dibawah ini. Kita akan membaca dua sensor yaitu BMP 180 dan HTU21DF. Data hasil pengukuran akan di wrapping dalam format Json sebagai payload yang akan dikirim ke Node-Red. Topik MQTT yang kita gunakan kita adalah **smartx**.

Parameter MQTT	**Value**
hosts	mqqt2.sandihex.id
Topic	smartx

```
#include <ArduinoJson.h>
#include <WiFi.h>
#include <PubSubClient.h>
#include <Wire.h>
#include "Adafruit_HTU21DF.h"
#include <Adafruit_BMP085.h>

const char* ssid = "iot";
const char* password = "";
const char* mqtt_server = "mqtt2.sandihex.id";

Adafruit_HTU21DF htu = Adafruit_HTU21DF();
WiFiClient espClient;
PubSubClient client(espClient);
Adafruit_BMP085 bmp;

long lastMsg = 0;

void setup() {
  delay(100);
  Serial.begin(115200);
  while (!Serial);
```

```
  // Cek inisialisasi Sensor HTU
  if (!htu.begin()) {
    Serial.println("Sensor HTU, tidak terkoneksi");
    while (1);
  }
  // Cek inisialisasi BMP
  if (!bmp.begin()) {
    Serial.println("Sensor BMP, tidak terkoneksi");
    while (1);
  }
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());

  client.setServer(mqtt_server, 1883);
}

void loop() {

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

  StaticJsonDocument<100> doc;
  char output[100];

  long now = millis();
```

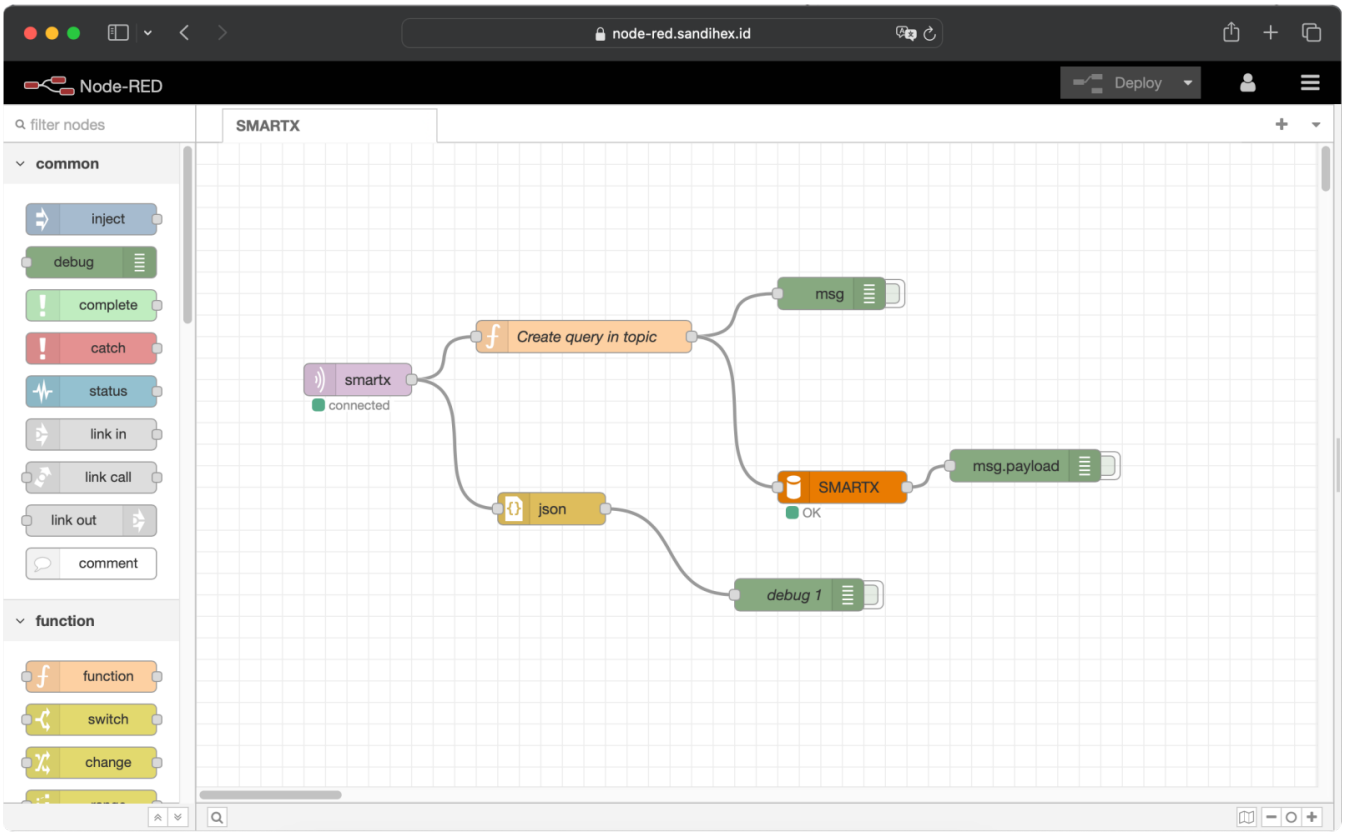
```
  if (now - lastMsg > 10000) {
    lastMsg = now;
    float temp = htu.readTemperature();
    float rel_hum = htu.readHumidity();
    float t2 = bmp.readTemperature();
    float pressure = bmp.readPressure();
    float altitude = bmp.readAltitude();
    doc["t1"] = temp;
    doc["t2"] = t2;
    doc["h"] = rel_hum;
    doc["alt"] = altitude;
    doc["p"] = pressure;
    serializeJson(doc, output, sizeof(output));
    Serial.println(output);
    client.publish("smartx", output);
  }

}

void reconnect() {
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    String clientId = "ESP32-SMARTX";
    if (client.connect(clientId.c_str())) {
      Serial.println("connected");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      delay(5000);
    }
  }
}
```

Data sensor yang dikirimkan oleh ESP32 melalui MQTT gateway akan kita simpan dalam database. Untuk itu kita akan buat database dengan perintah seperti di bawah ini.

```
CREATE TABLE iot_esp (
    id INT AUTO_INCREMENT PRIMARY KEY,
    timestamp VARCHAR(255),
    topic VARCHAR(200),
```

```
    data JSON,
    recorded DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

Konfigurasi di Node-Red, akan kita lakukan dengan membuat tiga blok palette yaitu MQTT untuk menerima data (subscribe) dari ESP, kemudian function palette untuk memasukan data kedalam database, kemudian palette mysql untuk mengkoneksikan data kedalam database. Secara lengkap ditampilkan sebagai berikut.



Hasil rekaman database akan kita tampilkan dengan dashboard Grrafana yang telah kita install sebelumnya. Detail instalasi Grafana dan visualisasi akan kita bahas di percobaan selanjutnya. Secara lengkap hasil dari percobaan kita ini dapat dilihat dari video berikut.

https://www.youtube.com/embed/yvjB-Nx2EWM

Revision #20
Created 3 March 2024 01:35:42 by Sandi Wibowo
Updated 31 March 2024 03:05:29 by Sandi Wibowo