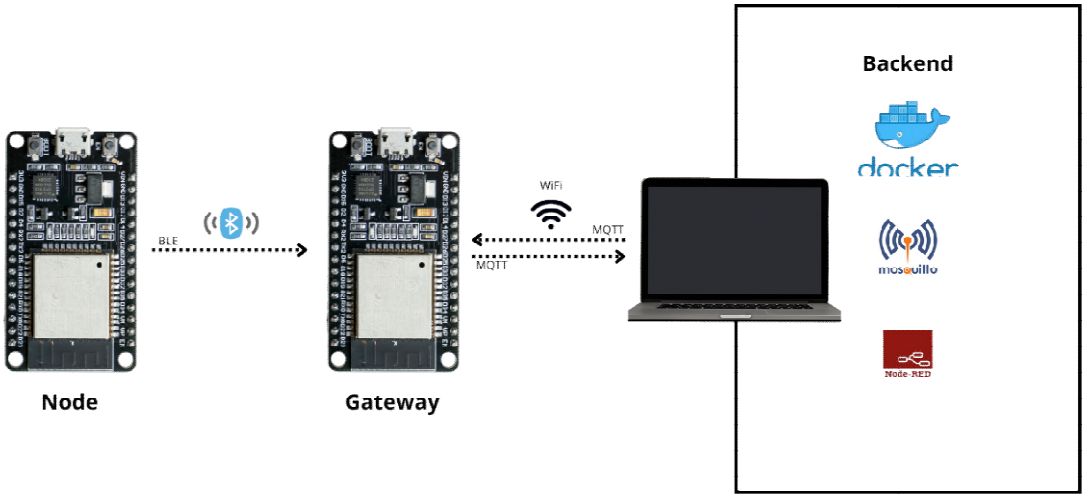
Tutorial *Gateway BLE-WiFi with MQTT (Message Queuing Telemetry Transport) to Backend*



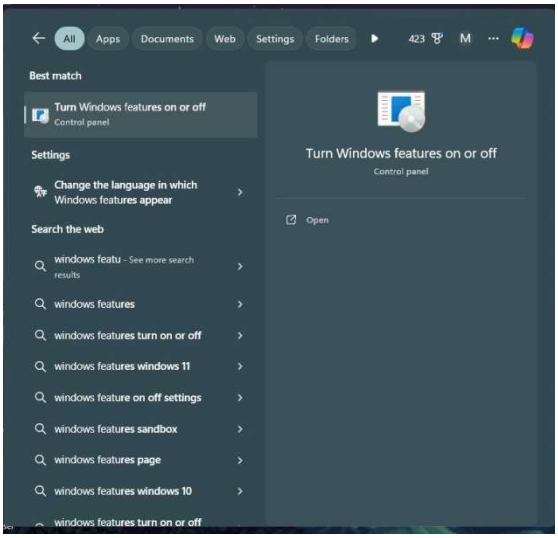
MQTT adalah protokol komunikasi ringan yang digunakan untuk berbagi pesan antara perangkat dalam jaringan IoT (*Internet of Things*). Broker adalah server yang bertanggung jawab untuk menerima, menyimpan, dan meneruskan pesan antara klien yang terhubung. Salah satu dari Server MQTT Broker adalah Mosquitto

Node-RED adalah platform berbasis *flow-based programming* yang menggunakan antarmuka *drag-and-drop* untuk membuat *flow* serangkaian blok atau node yang saling terhubung guna menjalankan berbagai fungsi secara otomatis.

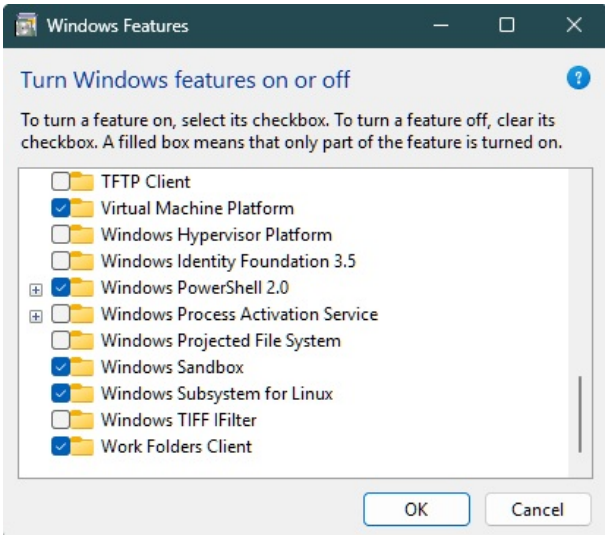
Docker Dekstop digunakan untuk membuat, mengelola, dan menjalankan *container* dengan lebih mudah tanpa harus mengonfigurasi Docker secara manual di sistem operasi.

Langkah-langkah:

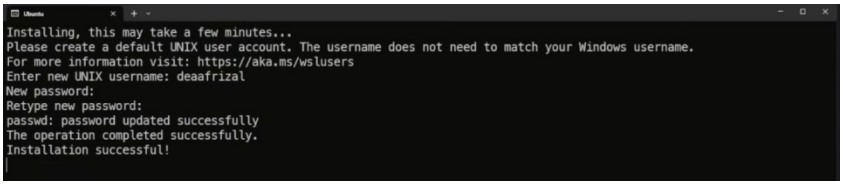
1. Sebelum instal Docker Dekstop, cari ***Windows Feature*** dengan cara pencet **Win + S**



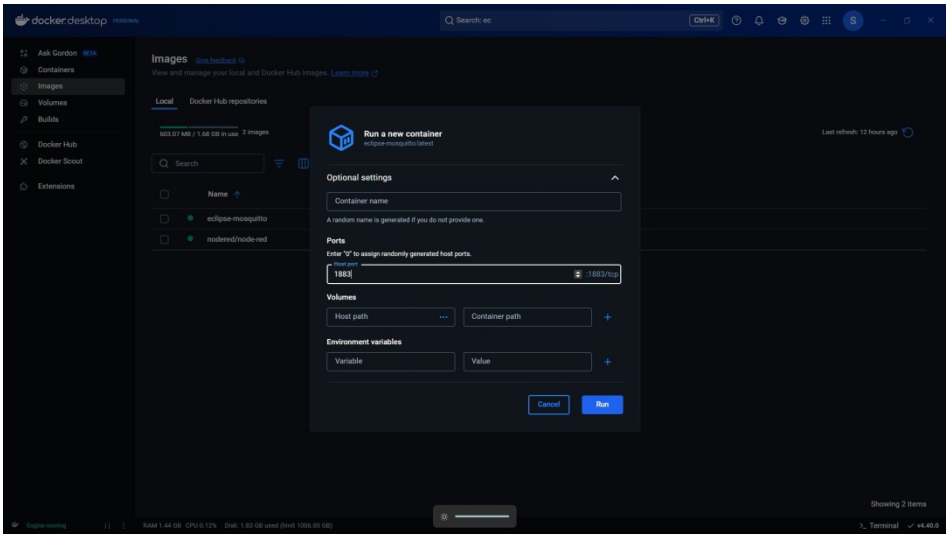
2. Ceklis ***Windows Subsytem for Linux.*** Lalu OK



3. Jalankan **CMD** sebagai ***Run as Administrator***
4. Ketikan *command* ***wsl –install*** tunggu hingga selesai instalasinya
5. Di akhir instalasinya buatlah *Username* dan *Password*

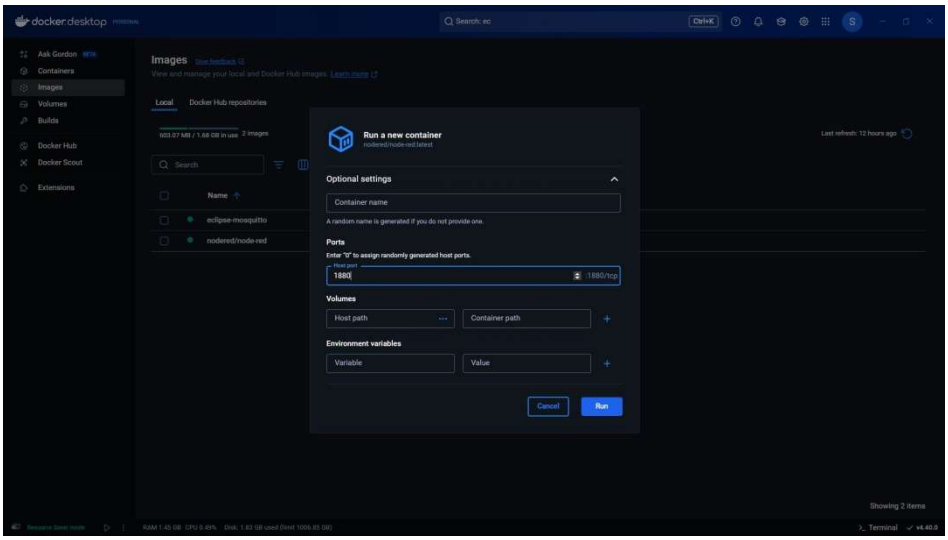


6. *Download* **Docker Dekstop *for Windows - AMD64*** dengan link [Docker: Accelerated Container Application Development]
7. Instal seperti biasa lalu *restart* laptopnya, lalu buka kembali Docker Dekstop-nya.
8. Carilah *Images*
    a. ***eclipse-mosquitto*** lalu *Pull*
    b. ***nodered/node-red*** lalu *Pull*
9. Buka menu *Images* untuk membuat *Container* dengan cara *Run* ***eclipse-mosquitto***
10. Pada ***Optional Settings*** isi ***Container Name*** dengan ***mosquitto*** dan kolom ***Host Port*** dengan **1883** lalu *Run*



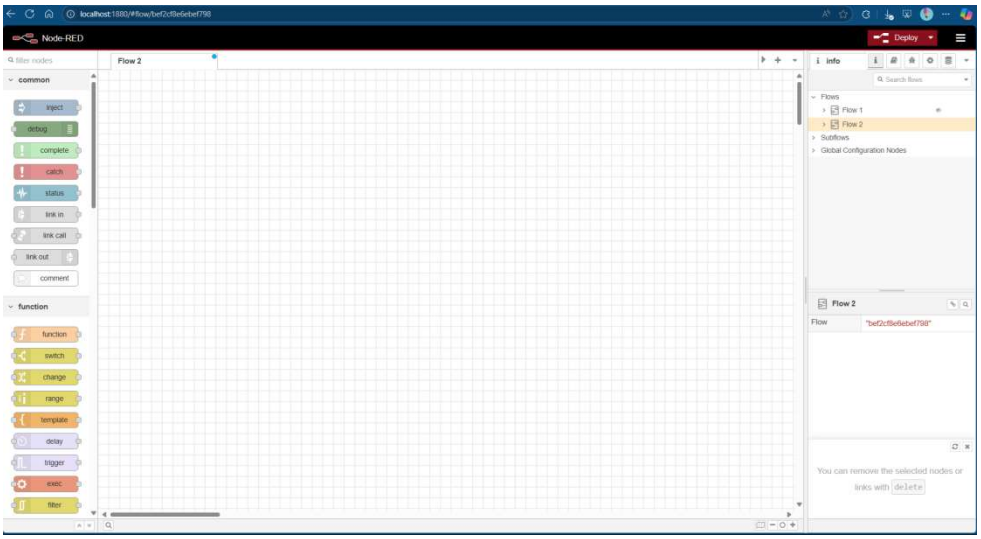
11. Selanjutnya membuat *Container* ***nodered/node-red***

12. Pada ***Optional Settings*** isi ***Container Name*** dengan ***nodered*** dan kolom ***Host Port*** dengan **1880** lalu *Run*



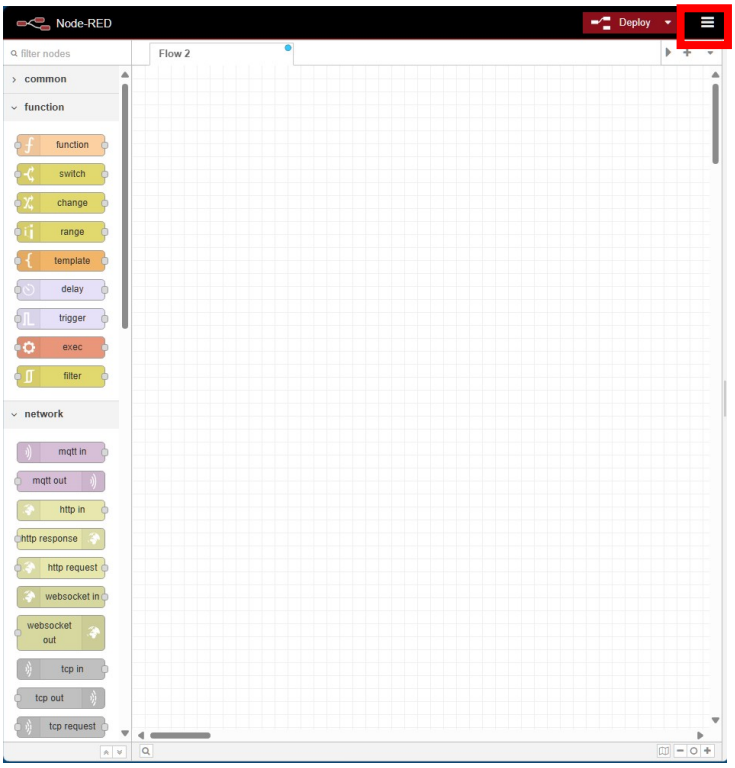
13. Pilih menu *Container,* lalu *Run* kedua *Container*-nya

14. Klik ***Ports nodered*** untuk melihat apa sudah berjalan. Nanti akan tertampil seperti ini



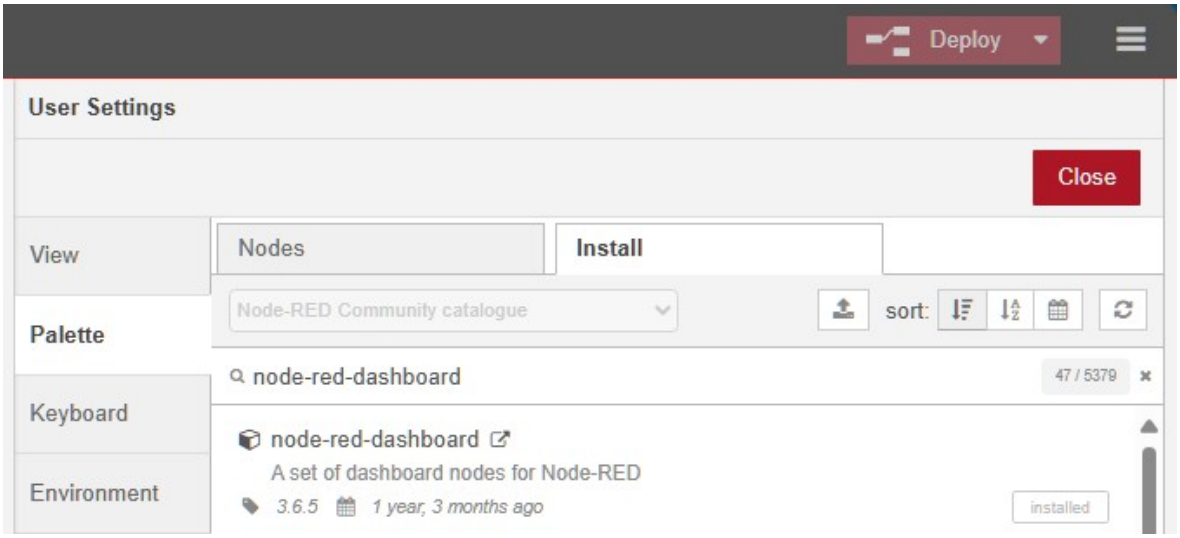
15. Lalu instal *pallete* ***node-red-dashboard,*** dengan cara

a. Klik **garis tiga** di pojok kanan,



b. Pilih ***manage pallete***

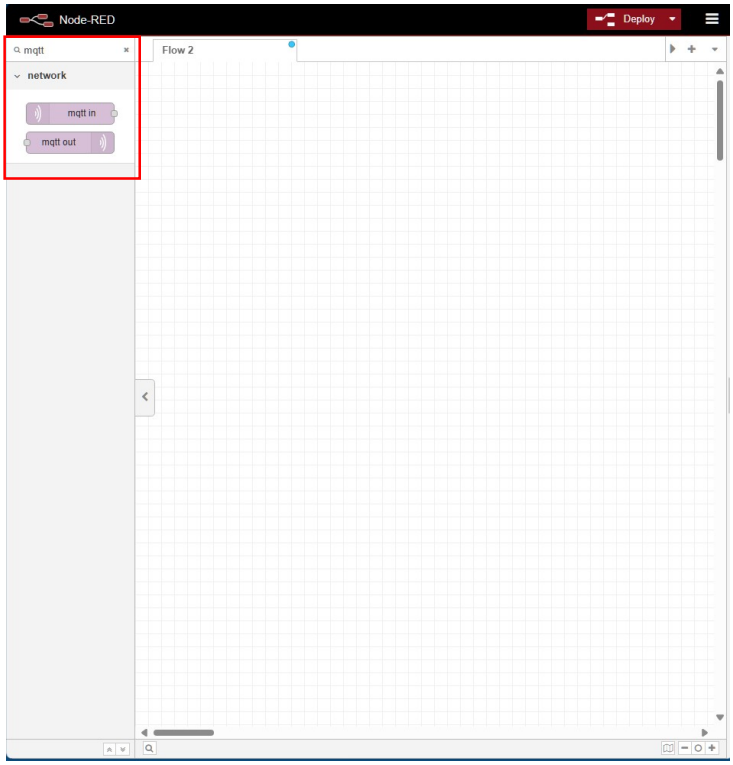
c. Pilih tab menu instal, cari dan instal ***node-red-dashboard.***



d. *Lalu close*

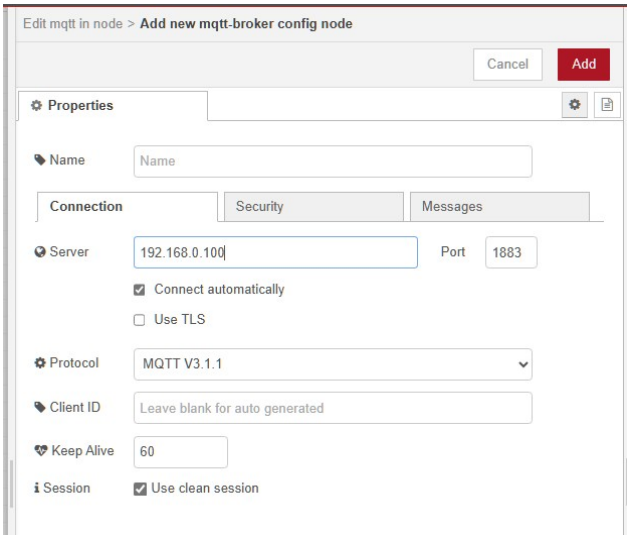
16. Membuat *Dashboard Backend* menggunakan *nodered* dengan cara

a. Cari ***mqtt in*** lalu *drag*

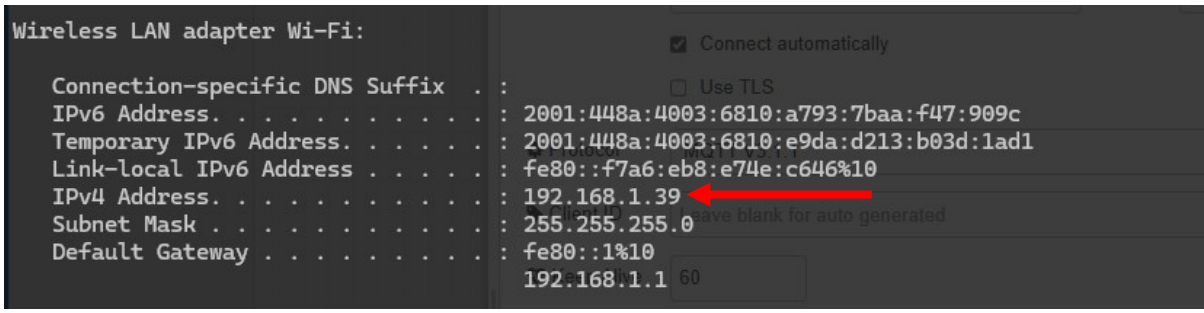


b. *Double click* pada ***mqtt in,***

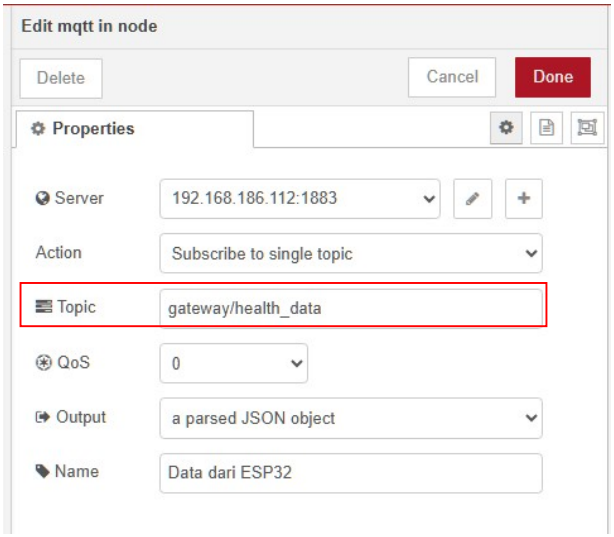
i. Klik ikon **+** lalu isi parameternya seperti ini. Lalu add



CATATAN: kolom *server (localhost)* isi menggunakan IP Server MQTT dengan cara cek command **ipconfig** pada cmd laptop



ii. isi parameternya seperti gambar di bawah. Lalu **done**

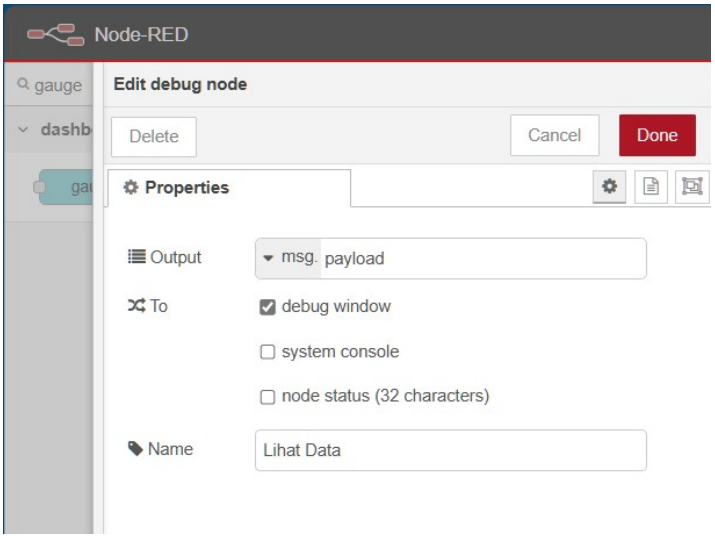


CATATAN: pada kolom ***Topic*** cocokan dengan penamaan Topic pada kode gateway

```
// --- Konfigurasi MQTT ---
const char* mqtt_server = "192.168.186.112"; // Ganti dengan IP address server MQTT
Anda
const int mqtt_port = 1883;                  // Port default MQTT
const char* mqtt_topic = "gateway/health_data";    // Topik MQTT
```

c. Cari ***debug*** lalu drag

d. *Double click* lalu ubah pada kolom *name* menjadi **lihat data,** lalu ***done***

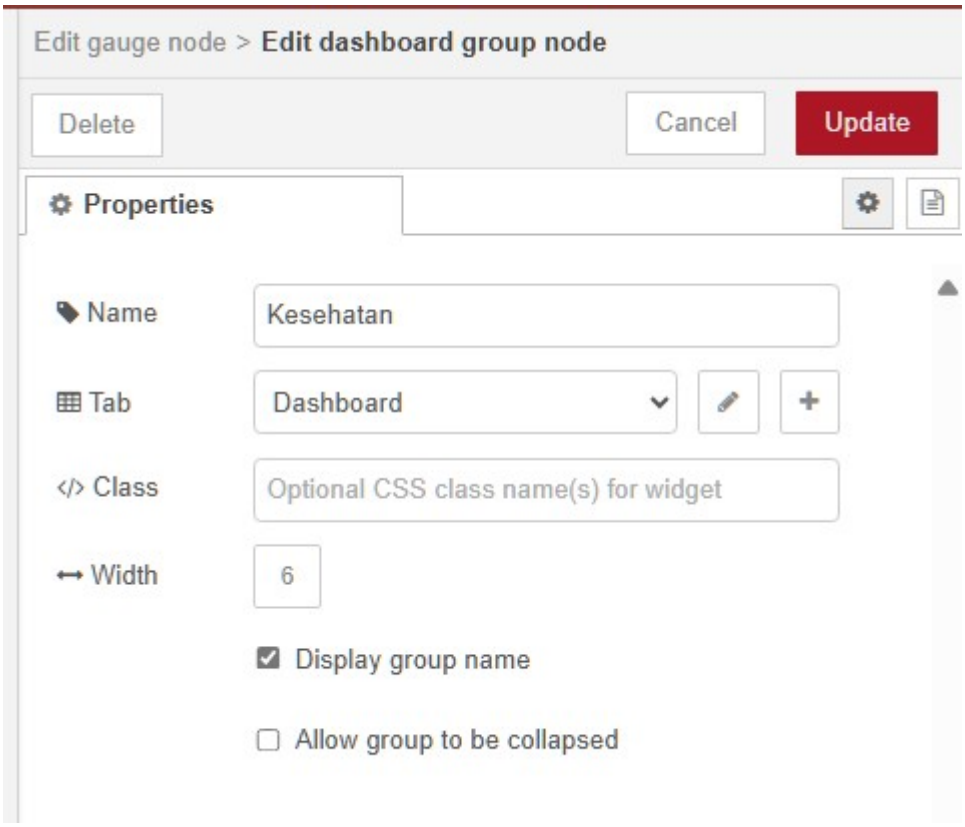


e. Cari ***gauge*** lalu drag

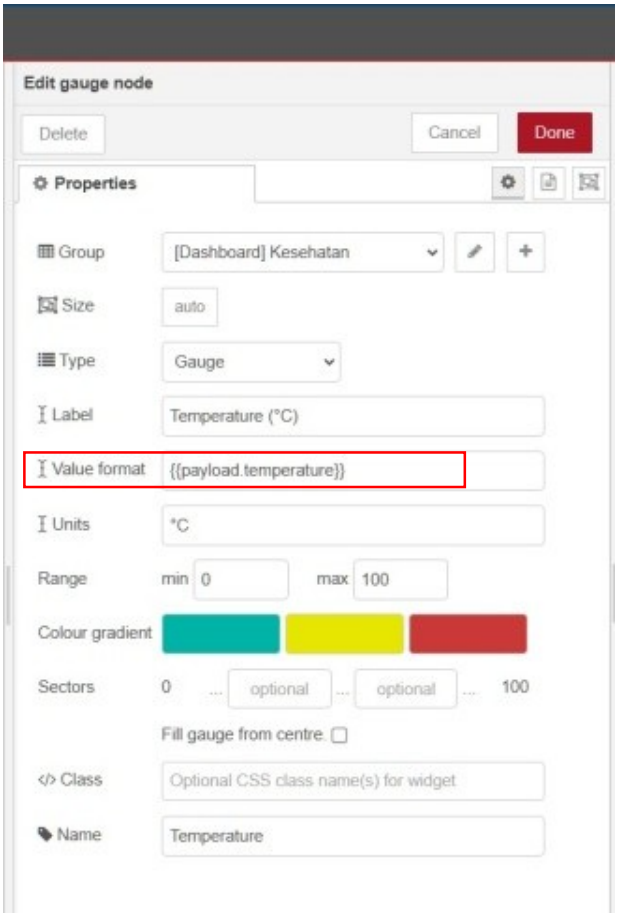
f. *Double click* **gauge**.

i. Klik +

ii. Lalu isi kolom *Name* dengan **Kesehatan** dan kolom *Tab* dengan **Dashboard.** Lalu Add



iii. isi parameter lainnya seperti ini



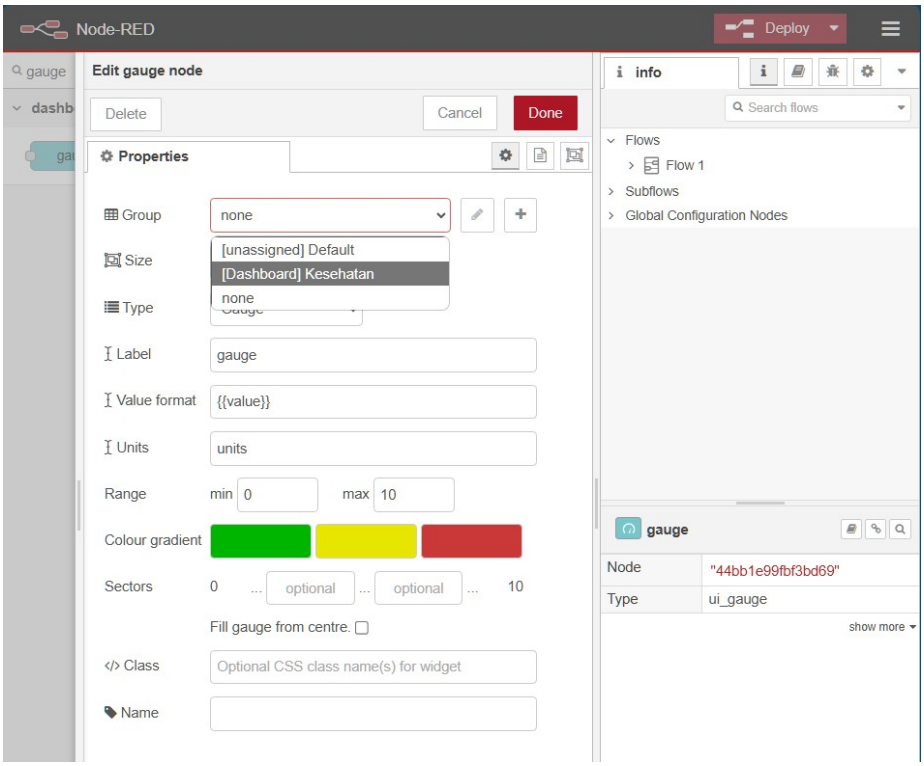
CATATAN: *untuk bagian* ***value format*** *cocokan dengan penamaan pada* ***string data*** *kode gateway*

```
void publishHealthData() {
  if (client.connected() && has_ble_health_data) { // Check for new data
    StaticJsonDocument<200> doc;
    doc["temperature"] = ble_temp;
    doc["bpm"] = ble_bpm;
    doc["spo2"] = ble_spo2;
    String jsonPayload;
    serializeJson(doc, jsonPayload);
    client.publish(mqtt_topic, jsonPayload.c_str());
    Serial.printf("Data Kesehatan dipublikasikan ke MQTT: %s\n",
jsonPayload.c_str());
    has_ble_health_data = false; // Reset the flag after publishing
  }
}
```

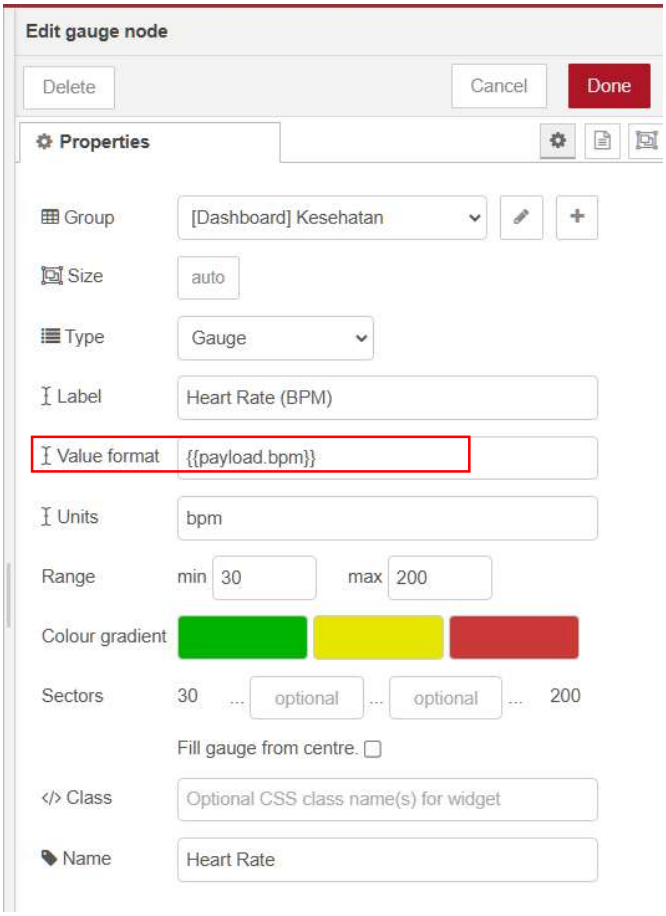
g. Cari ***gauge*** lagi, lalu drag

h. *Double click* ***gauge***,

i. Klik panah kecil pada Group pilih **Dashboard kesehatan**



ii. isi parameter lainnya seperti gambar di bawah



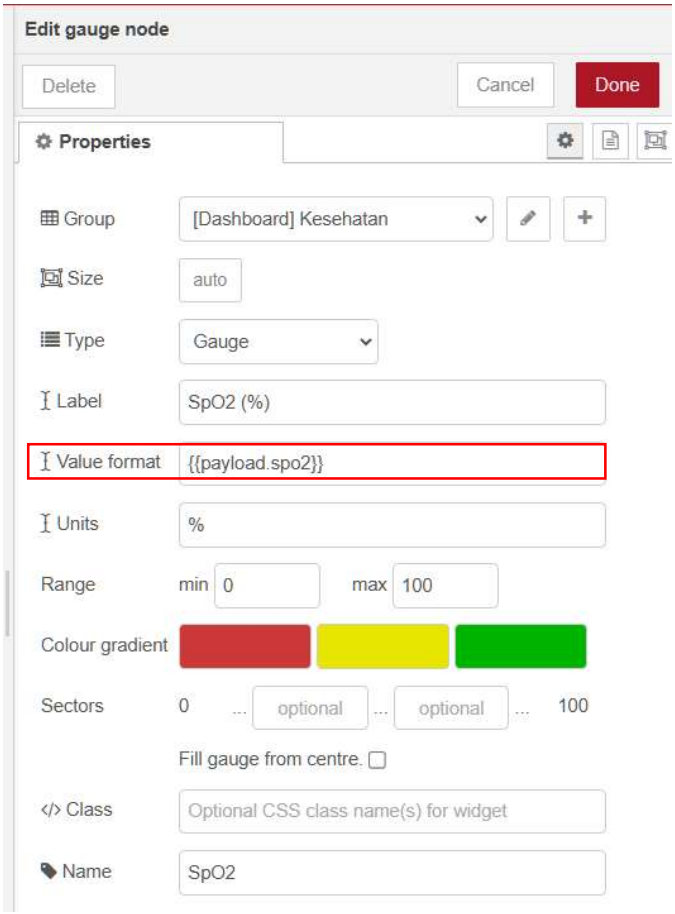
CATATAN: *untuk bagian* ***value format*** *cocokan dengan penamaan pada* ***string data*** *kode gateway*

```
void publishHealthData() {
  if (client.connected() && has_ble_health_data) { // Check for new data
    StaticJsonDocument<200> doc;
    doc["temperature"] = ble_temp;
    doc["bpm"] = ble_bpm;
    doc["spo2"] = ble_spo2;
    String jsonPayload;
    serializeJson(doc, jsonPayload);
    client.publish(mqtt_topic, jsonPayload.c_str());
    Serial.printf("Data Kesehatan dipublikasikan ke MQTT: %s\n",
jsonPayload.c_str());
    has_ble_health_data = false; // Reset the flag after publishing
  }
}
```

i. Cari ***gauge*** lagi, lalu drag

   i. Klik panah kecil pada Group pilih **Dashboard kesehatan**

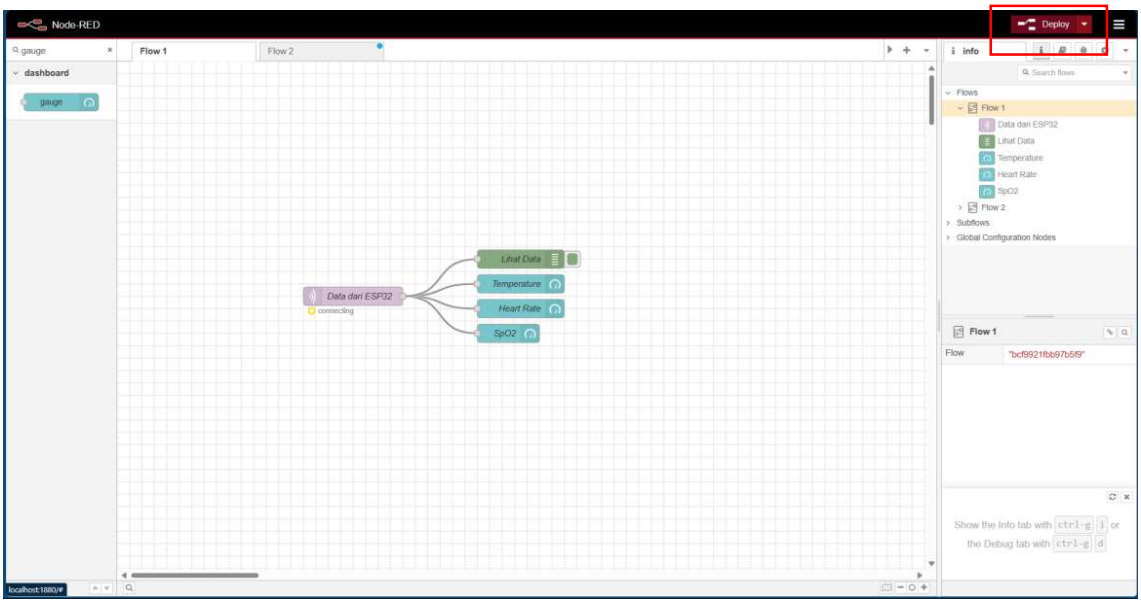
   ii. isi parameter seperti gambar di bawah



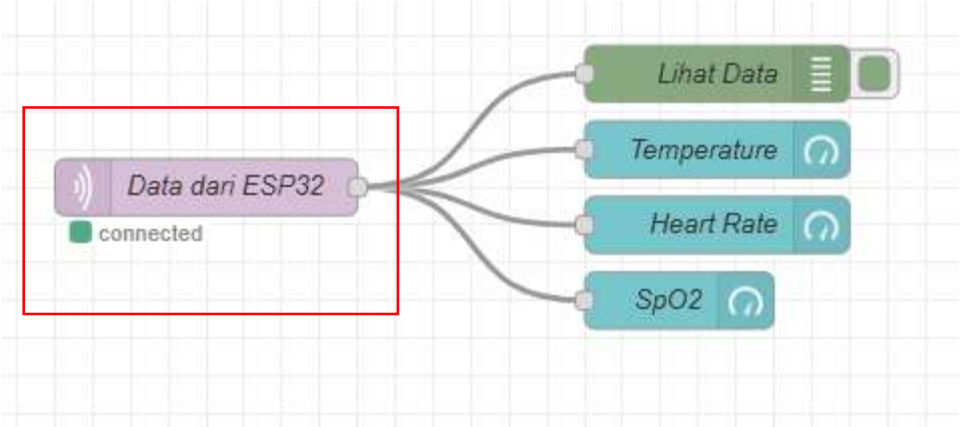
CATATAN: *untuk bagian* ***value format*** *cocokan dengan penamaan pada* ***string data*** *kode gateway*

```
void publishHealthData() {
  if (client.connected() && has_ble_health_data) { // Check for new data
    StaticJsonDocument<200> doc;
    doc["temperature"] = ble_temp;
    doc["bpm"] = ble_bpm;
    doc["spo2"] = ble_spo2;
    String jsonPayload;
    serializeJson(doc, jsonPayload);
    client.publish(mqtt_topic, jsonPayload.c_str());
    Serial.printf("Data Kesehatan dipublikasikan ke MQTT: %s\n",
jsonPayload.c_str());
    has_ble_health_data = false; // Reset the flag after publishing
  }
}
```

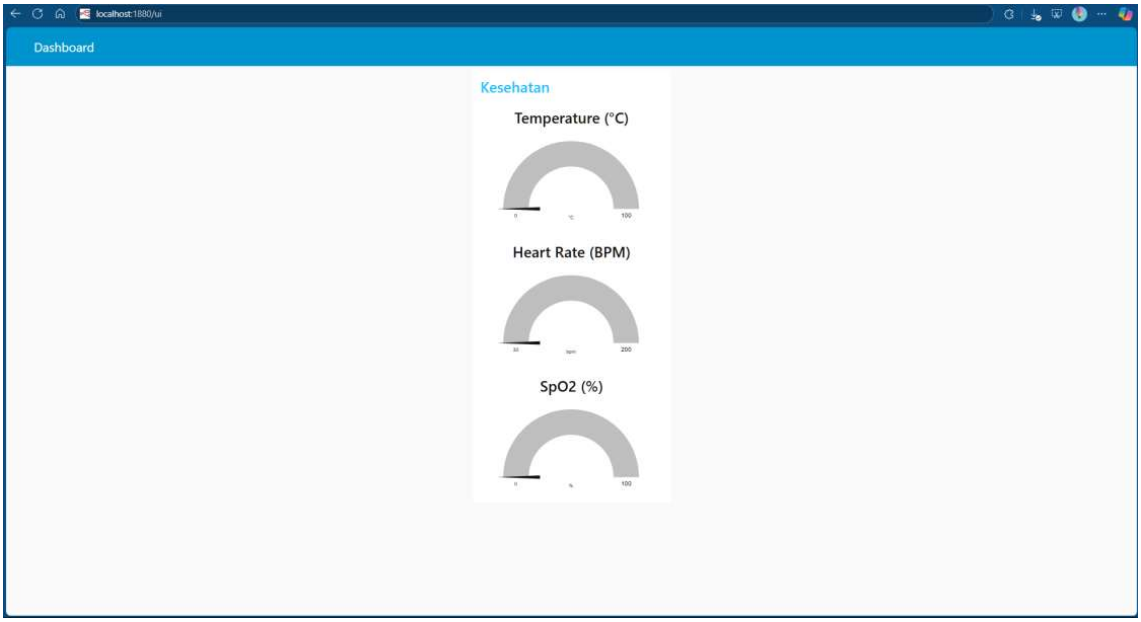
j. Hubungkan semua bloknya, lalu klik ***Deploy***



Nanti akan tertampil seperti ini, tandanya **Node-Red** berhasil konek ke **Mqtt Server**



k. Buat tab baru lalu ketikan **http://localhost:1880/ui**



CATATAN: *selalu pastikan IP pada kode esp32 gateway dan parameter" nodered itu sama dengan IP MQTT Server / berada di jaringan yg sama*