|  |
| --- |
| **Laporan Proyek**  **Teknologi Internet of Things**  **( *Pendeteksi Hama )*** |
| Kelompok 01   |  |  |  |  | | --- | --- | --- | --- | | 13321002  13321008  13321043 | Thalia Aniceta Saragih  Dion Saputra Manurung  Louis Cristiano Panggabean | | | |  |  | | |  |  | | |  |  | | |  |  | | |
| **INSTITUT TEKNOLOGI DEL**  **FAKULTAS VOKASI** |

**DAFTAR GAMBAR**

[Gambar 1. Program yang Dijalankan 3](#_Toc153808458)

[Gambar 2. Rangkaian Alat 5](#_Toc153808459)

[Gambar 3. Node Red berjalan di CMD 6](#_Toc153808460)

[Gambar 4. EMQX berjalan di CMD 7](#_Toc153808461)

[Gambar 5. InfluxDB dijalankan di CMD 8](#_Toc153808462)

[Gambar 6. Mendownload EMQX Repository 9](#_Toc153808463)

[Gambar 7. Menginstall EMQX dengan menggunakan perintah 10](#_Toc153808464)

[Gambar 8. Menghubungkan File Konfigurasi dengan Symlink 10](#_Toc153808465)

[Gambar 9. Menjalankan EMQX dari Raspberry Pi 10](#_Toc153808466)

[Gambar 10. Memantau atau Mengelola Broker MQTT 11](#_Toc153808467)

[Gambar 11. Mengatur Mechanism pada Authentication 12](#_Toc153808468)

[Gambar 12. Mengatur Backend pada Authentication 13](#_Toc153808469)

[Gambar 13. Mengatur Configuration pada Authetication 13](#_Toc153808470)

[Gambar 14. Menambahkan User 1 pada Dashboard EMQX 14](#_Toc153808471)

[Gambar 15. Menambahkan User 2 pada Dashboard EMQX 15](#_Toc153808472)

[Gambar 16. Menambahkan Topik Metrics 16](#_Toc153808473)

[Gambar 17. Menambahkan User 1 pada MQTTX Dashboard 17](#_Toc153808474)

[Gambar 18. Menambahkan User 2 pada MQTTX Dashboard 18](#_Toc153808475)

[Gambar 19. Menambahkan Topic pada MQTTX Dashboard 18](#_Toc153808476)

[Gambar 20. Tampilan Hasil Instalasi Node Red 19](#_Toc153808477)

[Gambar 21. Node Red Berjalan 20](#_Toc153808478)

[Gambar 22. Tampilan Membuka Workspace Node Red 21](#_Toc153808479)

[Gambar 23. Menambahkan Konfigurasi Node MQTT In Broker 22](#_Toc153808480)

[Gambar 24. Mengedit MQTT In 23](#_Toc153808481)

[Gambar 25. Menambahkan Konfigurasi Node MQTT Out Broker 24](#_Toc153808482)

[Gambar 26. Menginstall InfluxDB 25](#_Toc153808483)

[Gambar 27. Menginstall Client InfluxDB 26](#_Toc153808484)

[Gambar 28. Mengkonfigurasi InfluxDB 27](#_Toc153808485)

[Gambar 29. Menginstalasi Grafana 28](#_Toc153808486)

[Gambar 30. Mengatur Connection pada MQTT Broker 29](#_Toc153808487)

[Gambar 31. Mengatur Security pada MQTT Broker 29](#_Toc153808488)

[Gambar 32. Mengatur Messages pada MQTT Broker 30](#_Toc153808489)

[Gambar 33. Mengatur Aliran Data di Workspace Node Red 30](#_Toc153808490)

[Gambar 34. Penyusunan di Dashboard Grafana 31](#_Toc153808491)

[Gambar 35. Melihat Data Kueri 32](#_Toc153808492)

[Gambar 36. Menjalankan Script Python Kelompok 1 33](#_Toc153808493)

[Gambar 37. Tampilan CMD saat InfluxDB dibuka 33](#_Toc153808494)

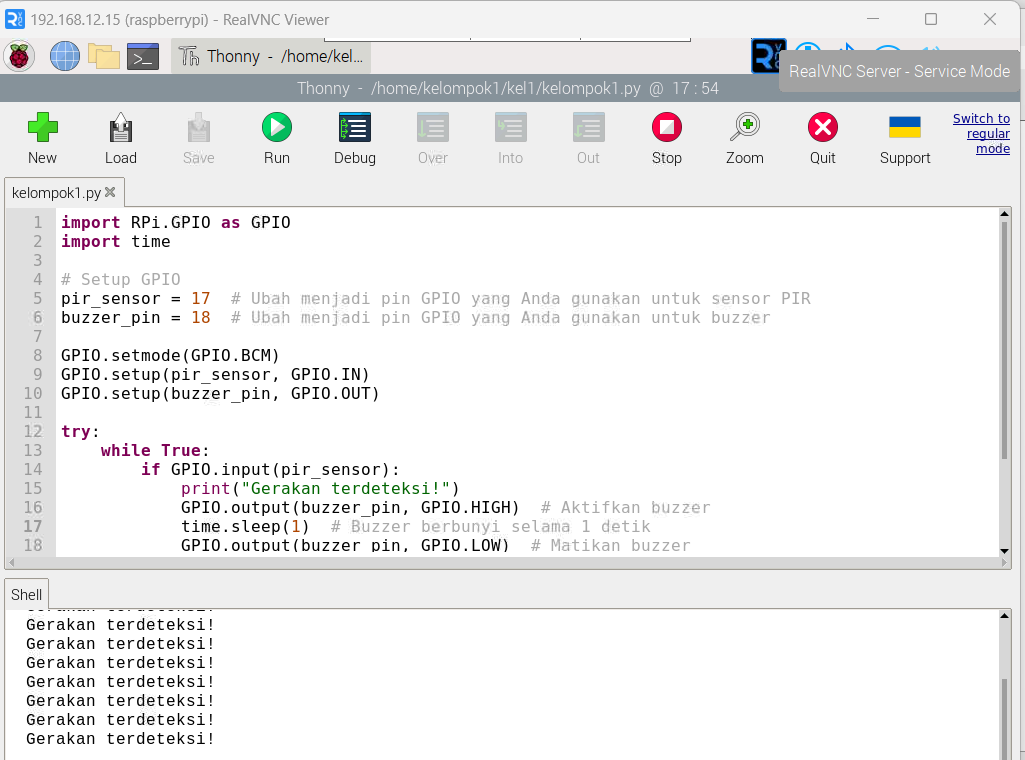
[Gambar 38. Tampilan CMD saat InfluxDB dijalankan 34](#_Toc153808495)

[Gambar 39. Mengedit Koneksi pada WorkSocket Client 35](#_Toc153808496)

[Gambar 40. Mengedit Subscription Pengiriman dan Penerimaan Pesan 35](#_Toc153808497)

**Sistem Pendeteksi Hama dengan Menggunakan Raspberry Pi 3 , Sensor dan Buzzer**

**Program dijalankan :**



Gambar . Program yang Dijalankan

**Kode Program**

import RPi.GPIO as GPIO

import time

# Setup GPIO

pir\_sensor = 17 # Ubah menjadi pin GPIO yang Anda gunakan untuk sensor PIR

buzzer\_pin = 18 # Ubah menjadi pin GPIO yang Anda gunakan untuk buzzer

GPIO.setmode(GPIO.BCM)

GPIO.setup(pir\_sensor, GPIO.IN)

GPIO.setup(buzzer\_pin, GPIO.OUT)

try:

while True:

if GPIO.input(pir\_sensor):

print("Gerakan terdeteksi!")

GPIO.output(buzzer\_pin, GPIO.HIGH) # Aktifkan buzzer

time.sleep(1) # Buzzer berbunyi selama 1 detik

GPIO.output(buzzer\_pin, GPIO.LOW) # Matikan buzzer

else:

GPIO.output(buzzer\_pin, GPIO.LOW) # Pastikan buzzer mati jika tidak ada gerakan

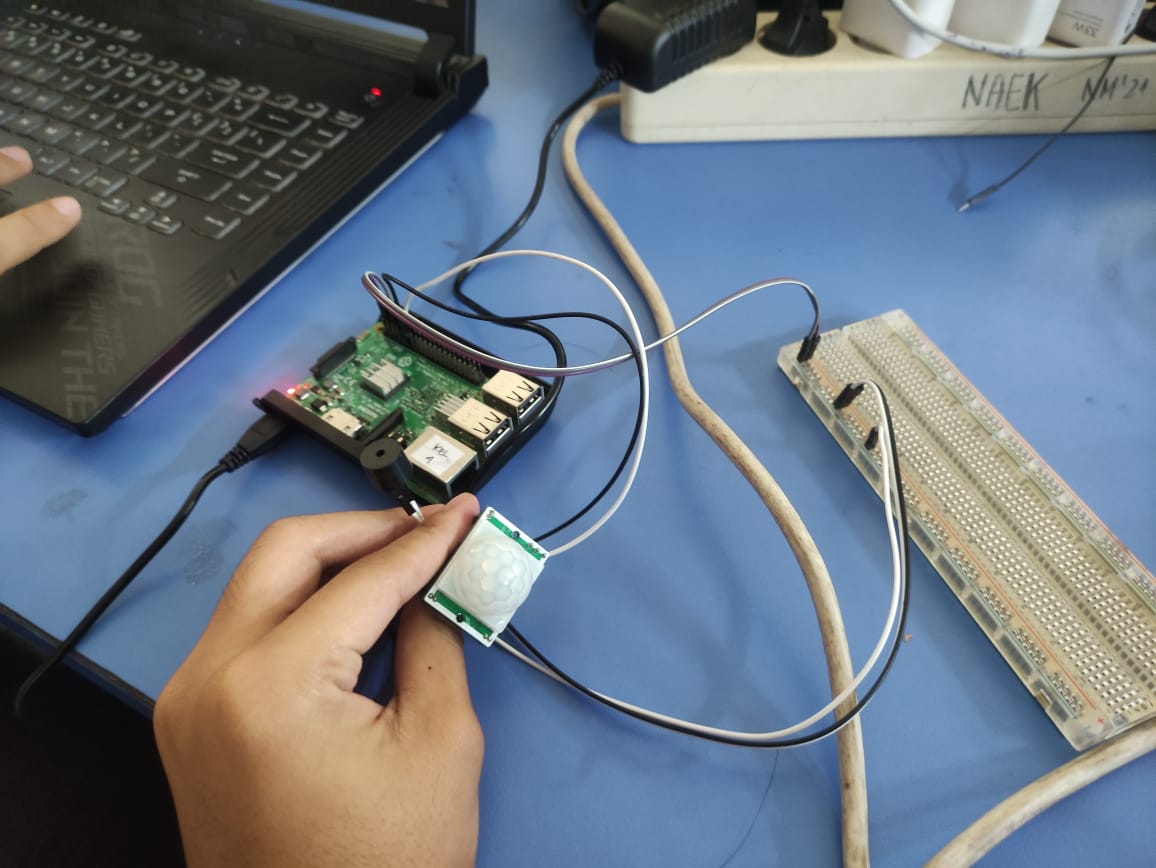
time.sleep(0.1) # Berhenti sebentar sebelum memeriksa lagi

except KeyboardInterrupt:

GPIO.cleanup()

* **Rangkaian**

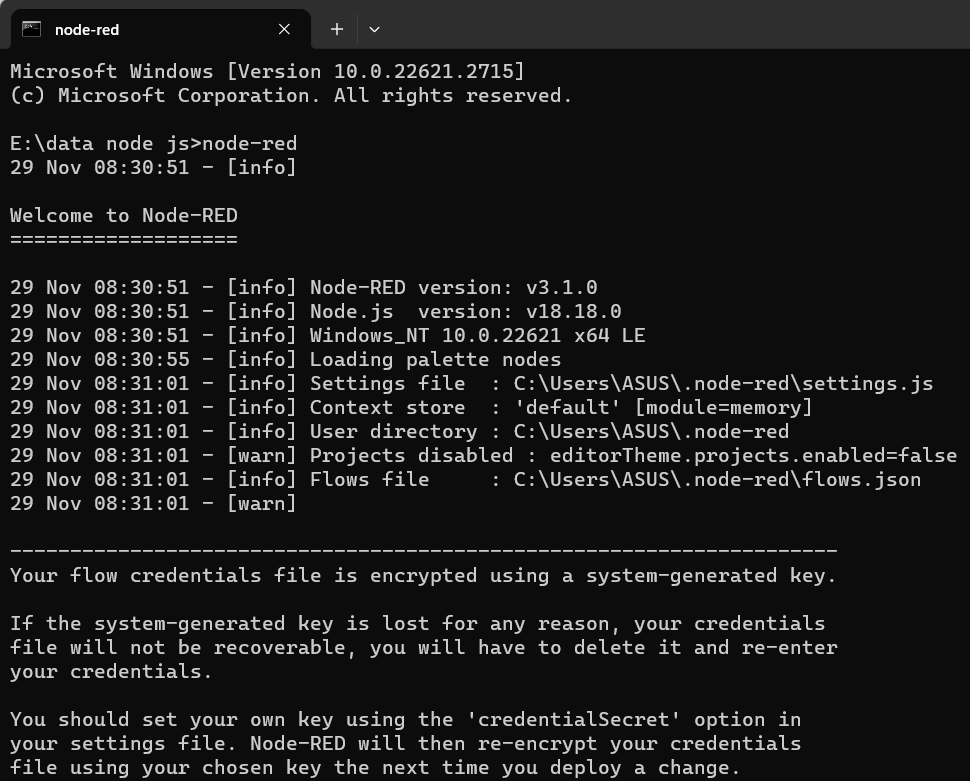
Berikut adalah rangkaian yang sudah di lakukan atau diterapkan



Gambar . Rangkaian Alat

* **Menjalankan node red dengan printah : node-red**

Node-RED adalah alat pengembangan visual yang memungkinkan Anda membuat alur aplikasi IoT dengan mudah tanpa menulis banyak kode. Setiap node mewakili fungsi atau tugas tertentu, dan dapat mengatur aliran data antara mereka. Node-RED menyediakan berbagai node yang dapat digunakan untuk mengolah data, mengirim dan menerima pesan, berkomunikasi dengan perangkat keras, dan banyak lagi. Anda dapat menggabungkan node-node ini dalam alur untuk membuat aplikasi IoT yang sesuai dengan kebutuhan proyek ini.

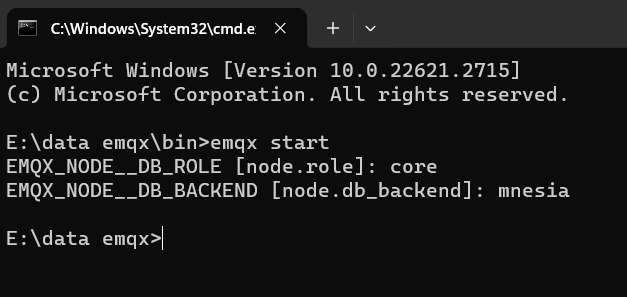


Gambar . Node Red berjalan di CMD

Node-RED menampilkan data dan hasil yang berkaitan dengan alur aplikasi IoT yang sedang berjalan dalam cmd. Ini dapat membantu memantau dan menganalisis data yang dikirim dan diterima oleh node-node serta memperbaiki masalah atau kesalahan yang mungkin terjadi dalam alur aplikasi IoT.

* **Menjalankan emqx dengan perintah : emqx start**
* **Lalu untuk menampilkan dashboard masukkan link 192.168.100.15:18083**

Dalam memantau dan mengelola broker MQTT EMQX, dashboard EMQX adalah antarmuka web. Dalam dashboard, nda dapat melihat informasi tentang koneksi, topik, pesan yang dikirim, dan banyak lagi. Dengan dashboard EMQX, Anda dapat memantau kinerja broker MQTT, menganalisis lalu lintas pesan, dan dengan mudah mengelola pengaturan dan izin. Ini adalah alat yang berguna untuk mengelola dan memantau lingkungan MQTT. Untuk mengakses dashboard EMQX melalui link yang diberikan, pastikan server EMQX berjalan dengan sukses dan memiliki koneksi jaringan yang benar.



Gambar . EMQX berjalan di CMD

Setelah mengetik perintah "emqx start" dan server EMQX berhasil dihidupkan, Anda dapat mengakses dashboard EMQX melalui browser web Anda dengan mengetik link "192.168.11.189:18083".

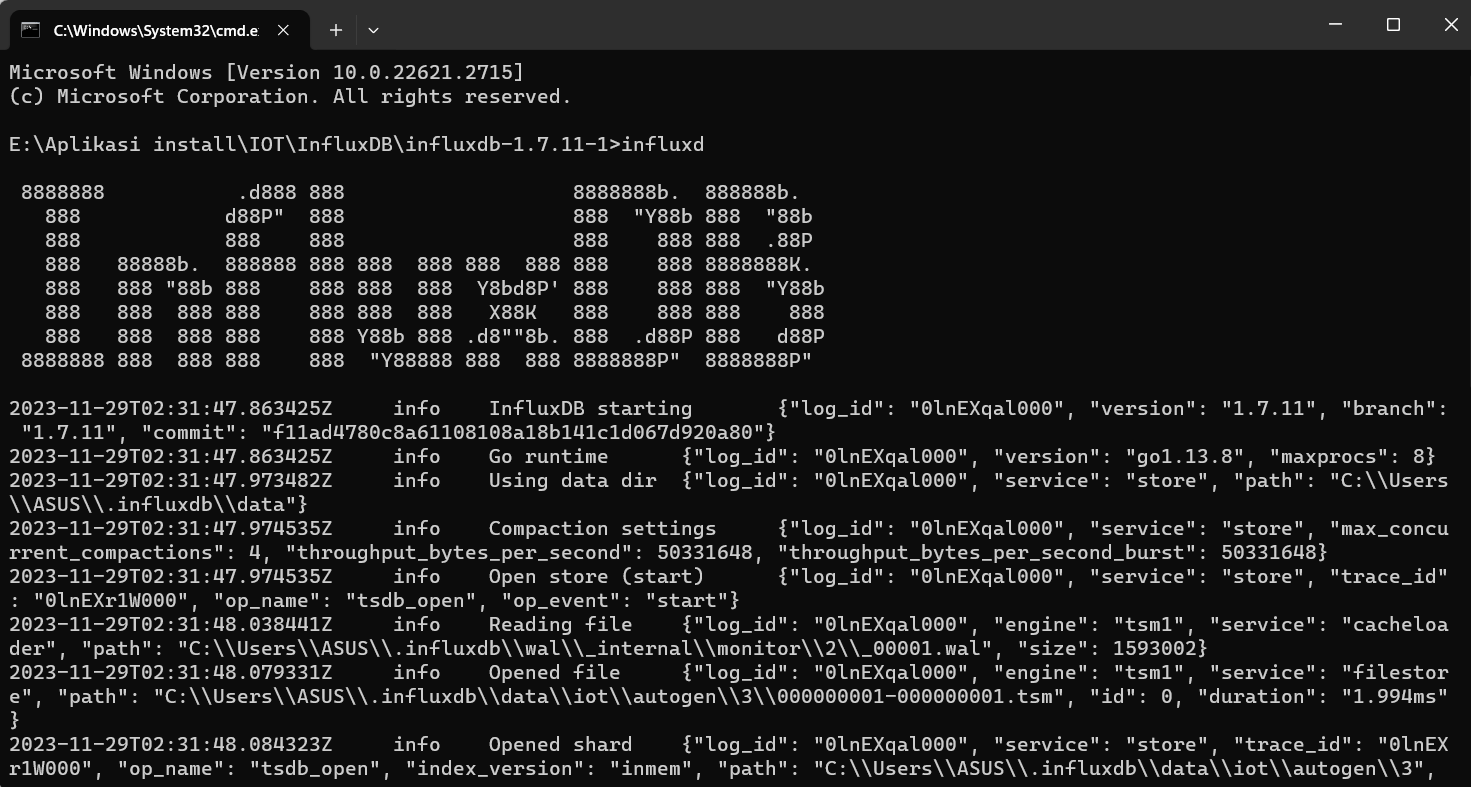
* **Menggunakan Username dan Password**

- Username yang digunakan adalah "admin".

- Password yang digunakan adalah "kelompok3".

* **Menjalankan influx dengan perintah : influxd**

InfluxDB adalah sebuah basis data seri waktu yang dapat digunakan untuk menyimpan dan mengelola data menggunakan timestamp.



Gambar . InfluxDB dijalankan di CMD

Untuk menjalankan InfluxDB, kita perlu menggunakan perintah "influxd" di terminal atau prompt komando. Pastikan bahwa InfluxDB telah diinstal sebelum menjalankannya. Setelah menjalankan perintah "influxd", InfluxDB akan mulai berjalan dan siap digunakan.

* **Menggunakan IP windows :** [**http://192.168.100.15:18083**](http://192.168.100.15:18083)

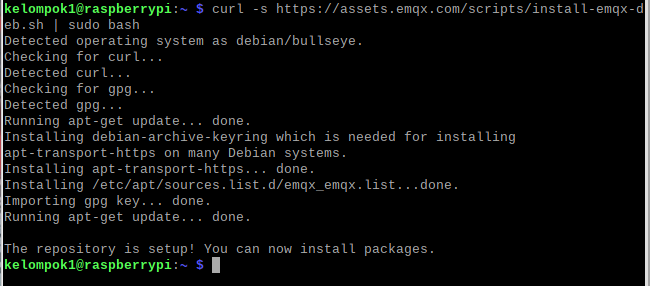
Alamat IP adalah alamat jaringan yang menunjukkan perangkat atau layanan yang berjalan di dalam jaringan. Dalam kasus ini, Anda dapat menemukan alamat IP ini di bilah alamat browser web. Kami akan terhubung menggunakan protokol HTTP dan port 18083. Pastikan firewall tidak menghentikan Anda untuk terhubung ke port ini.

**Lalu setting Raspberry pi**

Komputer mini yang dapat diprogram untuk proyek ini adalah Raspberry Pi.

* **Download the emqx repository**

EMQX adalah platform open-source yang digunakan untuk mengelola perangkat IoT melalui protokol MQTT. Kita dapat menggunakan perintah di terminal Raspberry Pi untuk mengunduh repositori EMQX, menggunakan perintah *curl -s https://assets.emqx.com/scripts/install-emqx-deb.sh | sudo bash*

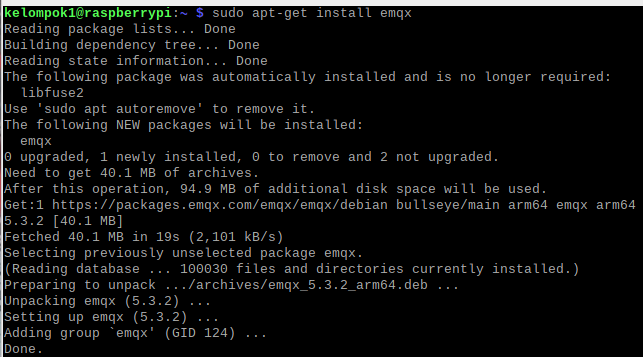
****

Gambar . Mendownload EMQX Repository

Perintah ini akan mengunduh dan menjalankan skrip instalasi EMQX secara otomatis

* **Install EMQX dengan memasukkan perintah : sudo apt-get install eqmx**

Untuk menginstal EMQX pada Raspberry Pi, kita dapat menggunakan perintah sudo apt-get install emqx. Namun, perlu diingat bahwa perintah yang kita sebutkan, sudo apt-get install eqmx, mungkin tidak valid karena mungkin terdapat kesalahan penulisan. Pastikan untuk menggunakan perintah yang benar, sudo apt-get install emqx. Setelah membuka terminal Raspberry Pi, gunakan perintah berikut untuk menjalankan perintah instalasi EMQX :

****

Gambar . Menginstall EMQX dengan menggunakan perintah

****

Gambar . Menghubungkan File Konfigurasi dengan Symlink

Setelah proses instalasi selesai, sistem akan mengunduh dan menginstal paket yang diperlukan. Tunggu hingga selesai.

* **Run emqx dari raspberry pi**

Untuk memulai EMQX, buka terminal Raspberry Pi dengan perintah berikut :

****

Gambar . Menjalankan EMQX dari Raspberry Pi

Layanan EMQX pada Raspberry Pi akan dimulai dengan perintah ini. Tunggu beberapa saat hingga layanan EMQX sepenuhnya dimulai. EMQX akan berjalan dan siap digunakan setelah menjalankan perintah ini. Dalam praktikum atau proyek ini, kita dapat melanjutkan ke langkah berikutnya.

* **Buka dashboard dengan memasukkan perintah di browser**

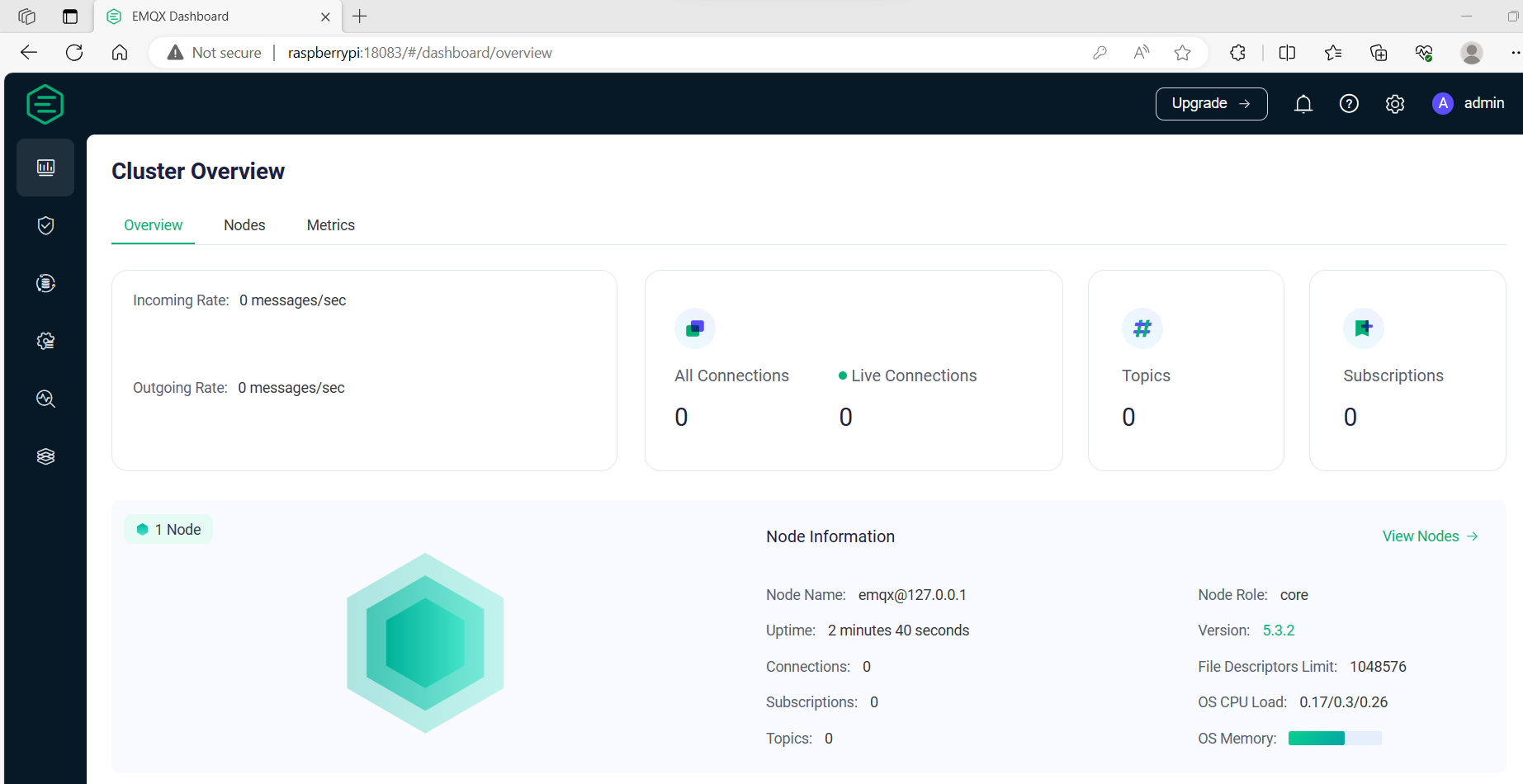
Masukkan alamat ini di bilah alamat browser Anda:

*raspberrypi:18083*

* **Dengan memasukkan username : admin dengan passwd : kelompok1**

Ini akan membuka dashboard EMQX, yang akan memungkinkan kita untuk mengelola dan memantau broker MQTT. Login dengan username dan password berikut:

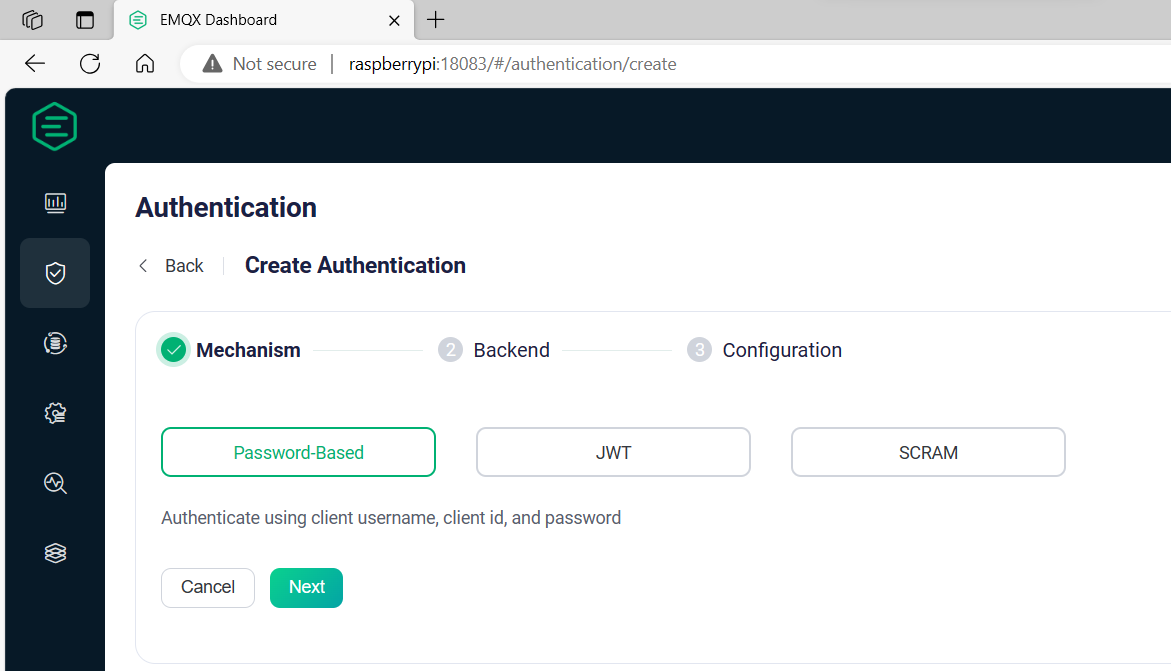
* Username : admin
* Password : kelompok1

****

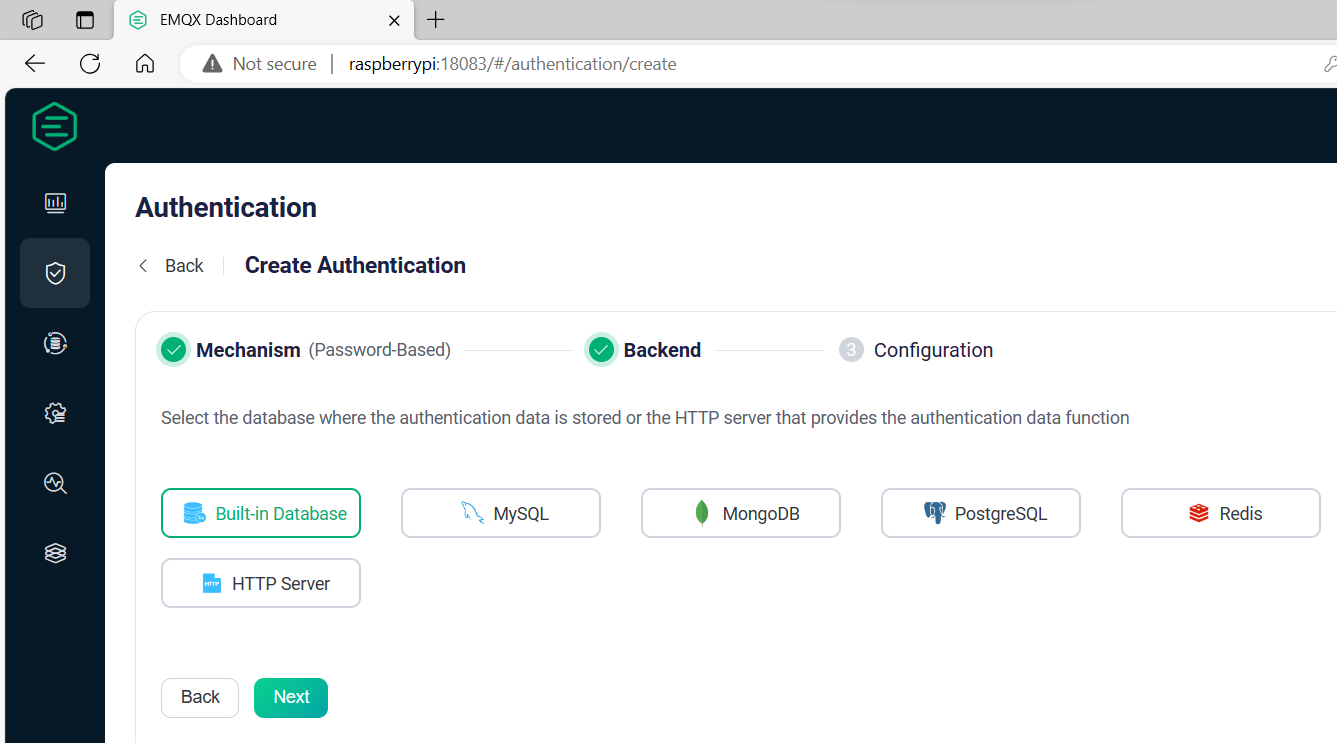
Gambar . Memantau atau Mengelola Broker MQTT

Setelah memasuki, kita seharusnya dapat mengakses dashboard EMQX, sehingga kita dapat melanjutkan untuk mengkonfigurasi dan mengelola broker sesuai kebutuhan praktikum atau proyek ini.

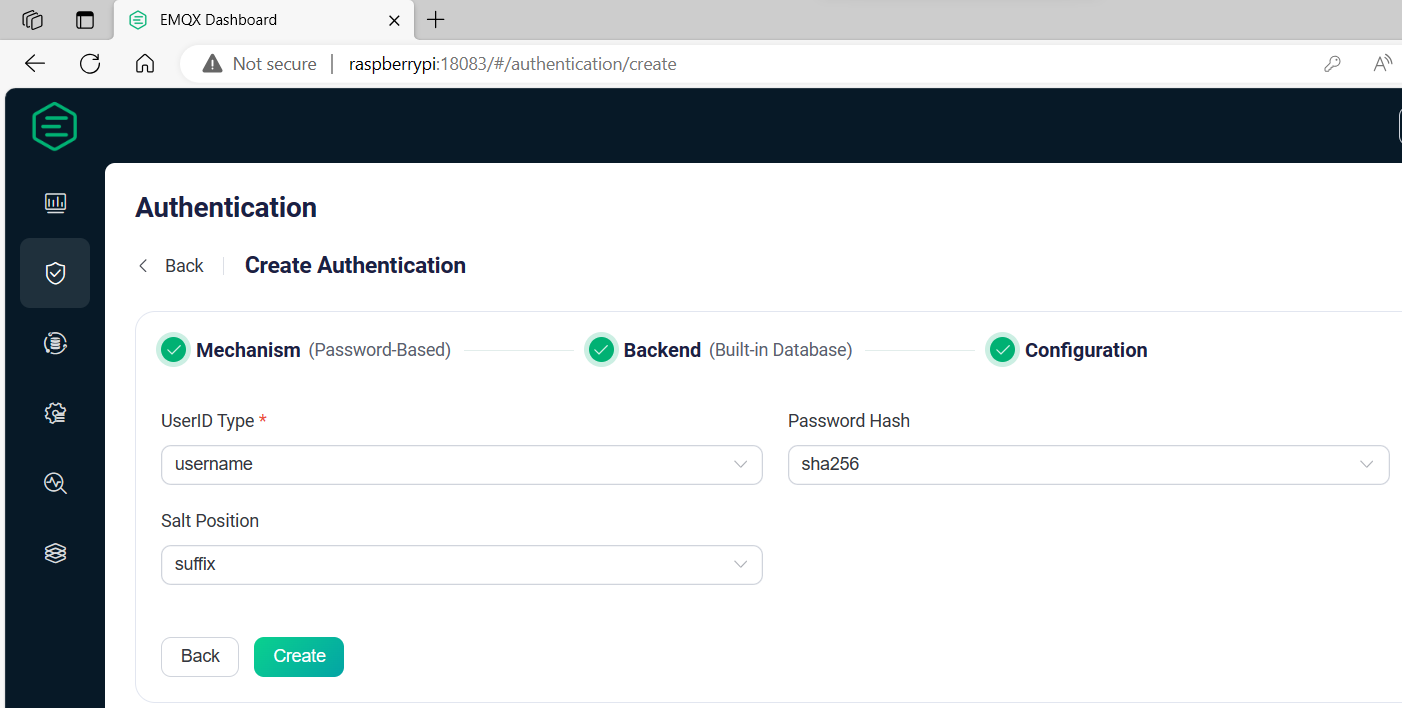
* **Create authentication**
* Klik pada tab "Authentication & Authorization" di menu sebelah kiri.
* Pilih "Authentication" dari submenu.
* Klik tombol "Add" untuk menambahkan autentikasi baru.
* Isi informasi yang diperlukan, seperti username dan password untuk autentikasi.
* Klik tombol "Add" untuk menyimpan autentikasi baru.
* Setelah autentikasi berhasil dibuat, kita dapat menggunakan username dan password tersebut untuk masuk ke dashboard EMQX di masa mendatang.

****

Gambar . Mengatur Mechanism pada Authentication

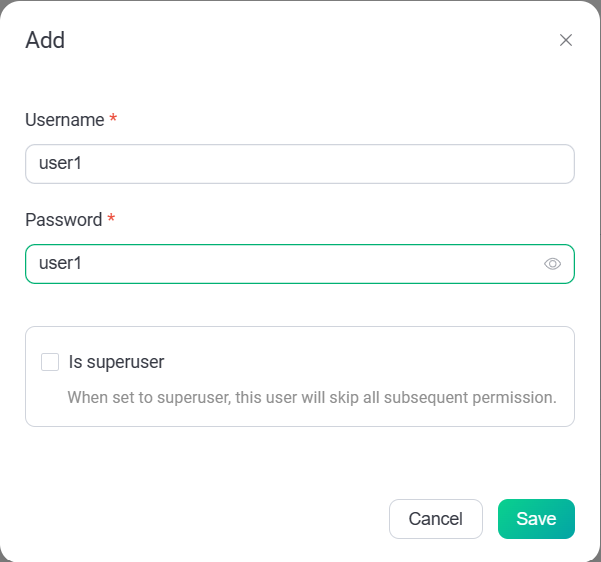
****

Gambar . Mengatur Backend pada Authentication

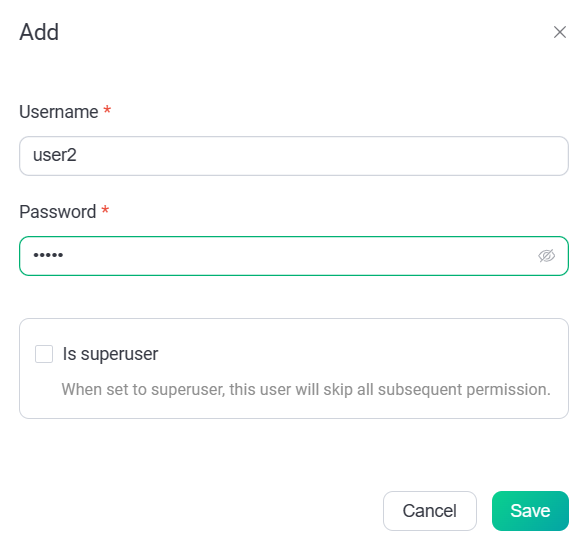
****

Gambar . Mengatur Configuration pada Authetication

* **Menambah user pada dashboard emqx pada Authentication**
* Dalam submenu, pilih "Autentikasi". Klik tombol "Add" atau "Tambah" untuk menambahkan pengguna baru.
* Masukkan informasi yang diperlukan, seperti nama pengguna dan kata sandi untuk pengguna baru.
* Pastikan untuk memilih jenis autentikasi yang sesuai dengan kebutuhan proyek ini (misalnya, username-password, certificate , atau lainnya, tergantung pada konfigurasi yang diinginkan).
* Setelah mengisi detail pengguna, klik tombol "Add" atau "Simpan” .

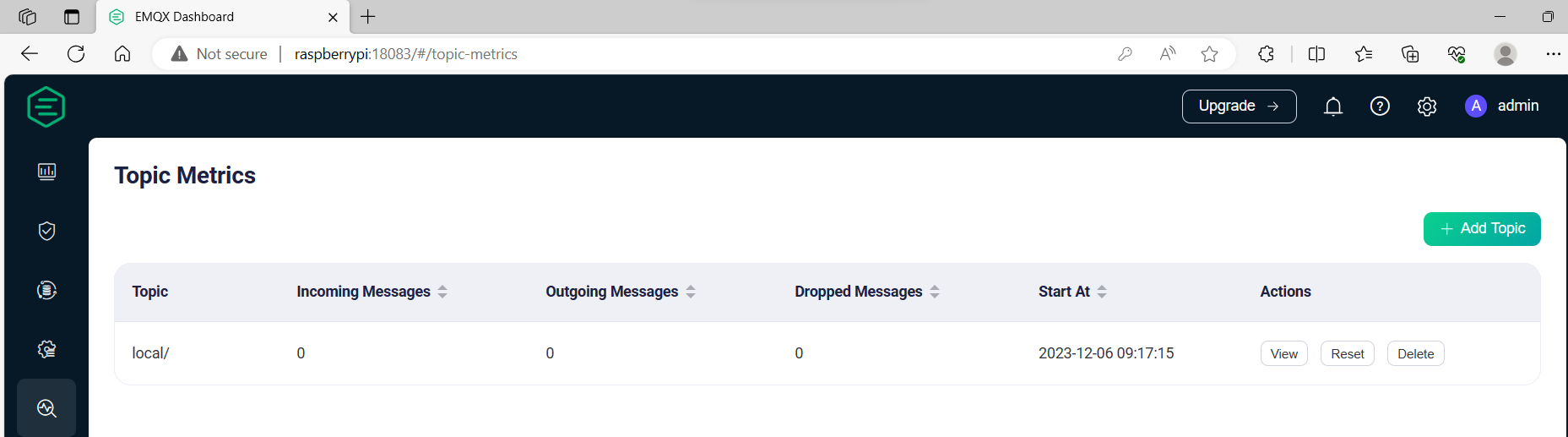
****

Gambar . Menambahkan User 1 pada Dashboard EMQX

****

Gambar . Menambahkan User 2 pada Dashboard EMQX

* **Add topic metrics**
* Masukkan nama topik yang ingin Anda tambahkan, misalnya "metrics".
* Kita juga dapat mengatur opsi lain seperti QoS (Quality of Service) dan Retain (Retained Message) sesuai kebutuhan.
* Setelah mengisi detail topik, klik tombol "Add" atau "Simpan" untuk menyimpan topik baru.
* Setelah topik "metrics" ditambahkan, pastikan untuk memverifikasi bahwa topik tersebut telah berhasil ditambahkan dan dapat digunakan untuk publikasi dan/atau langganan pesan.

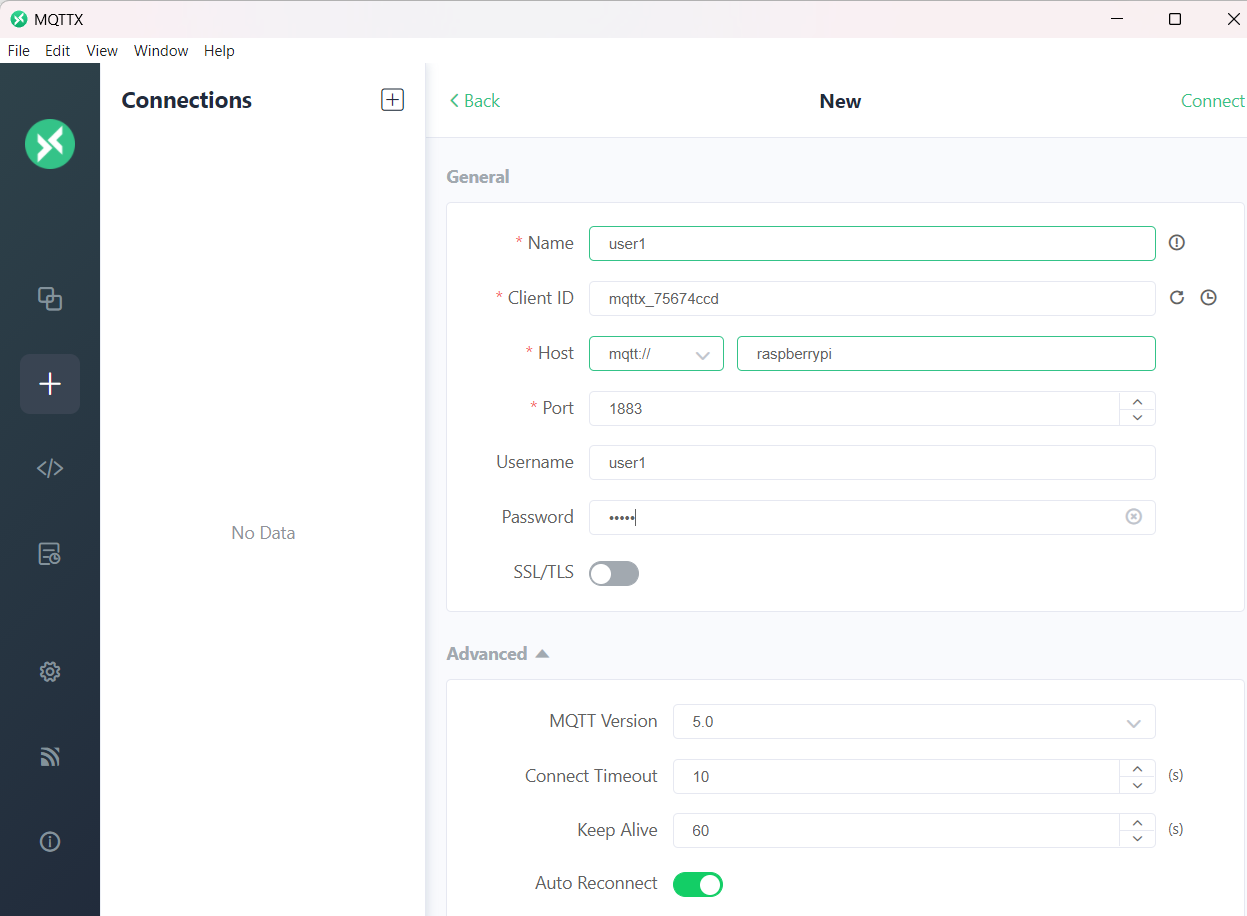
****

Gambar . Menambahkan Topik Metrics

* **Setting MQTTX**

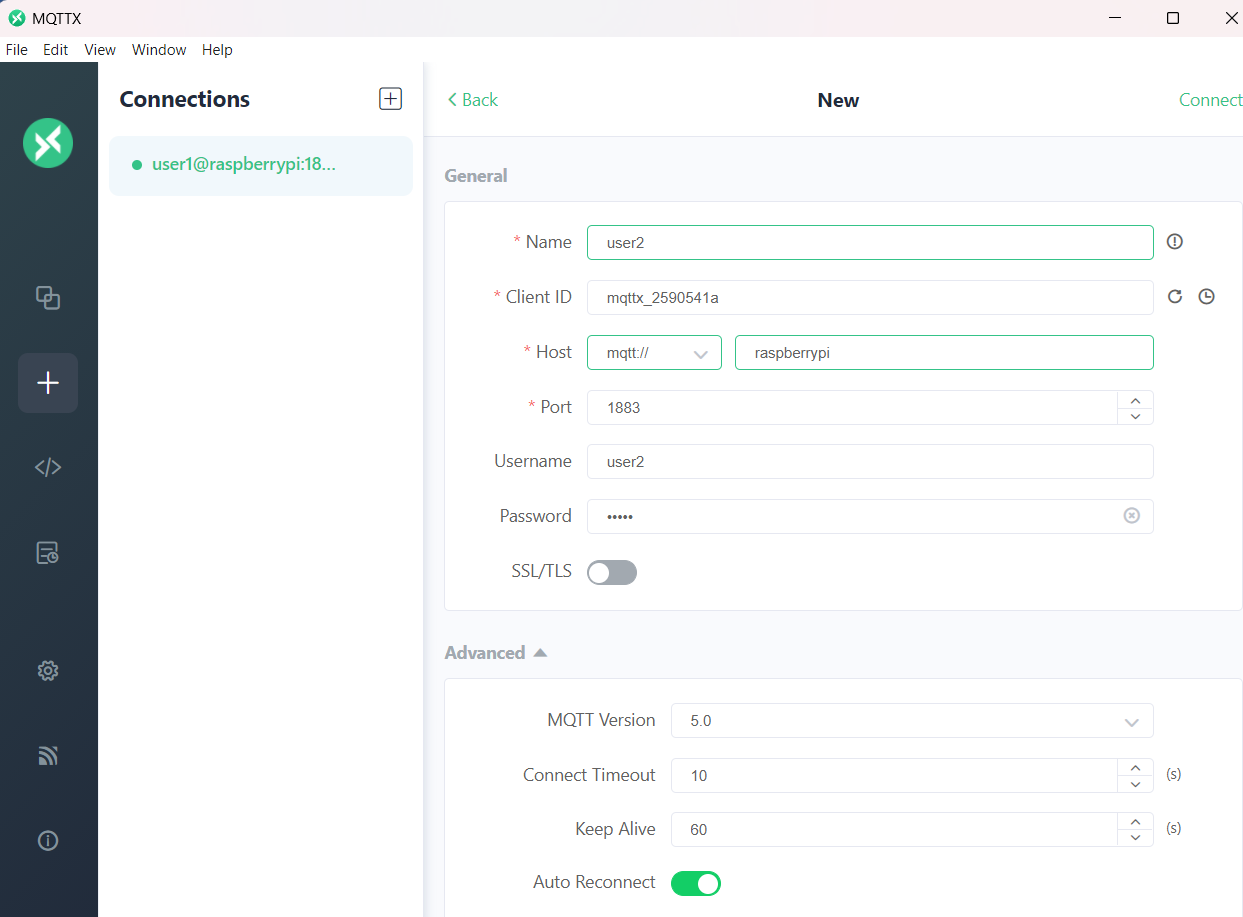
MQTT (Message Queuing Telemetry Transport) adalah protokol komunikasi sederhana yang digunakan untuk pertukaran pesan antara perangkat dalam jaringan Internet of Things (IoT). Modelnya berdasarkan publikasi-subskripsi, di mana perangkat yang bertindak sebagai penerbit mengirim pesan ke topik, dan perangkat yang bertindak sebagai pelanggan menerima pesan dari topik yang mereka langganan. Berikut adalah langkah-langkah yang dilakukan dalam mengatur MQTTX

1. Buka MQTTX dan buat profil koneksi baru. Masukkan informasi tentang koneksi seperti alamat broker MQTT, port, dan protokol yang digunakan, biasanya TCP. Selain itu, masukkan informasi autentikasi seperti nama pengguna dan kata sandi jika diperlukan.

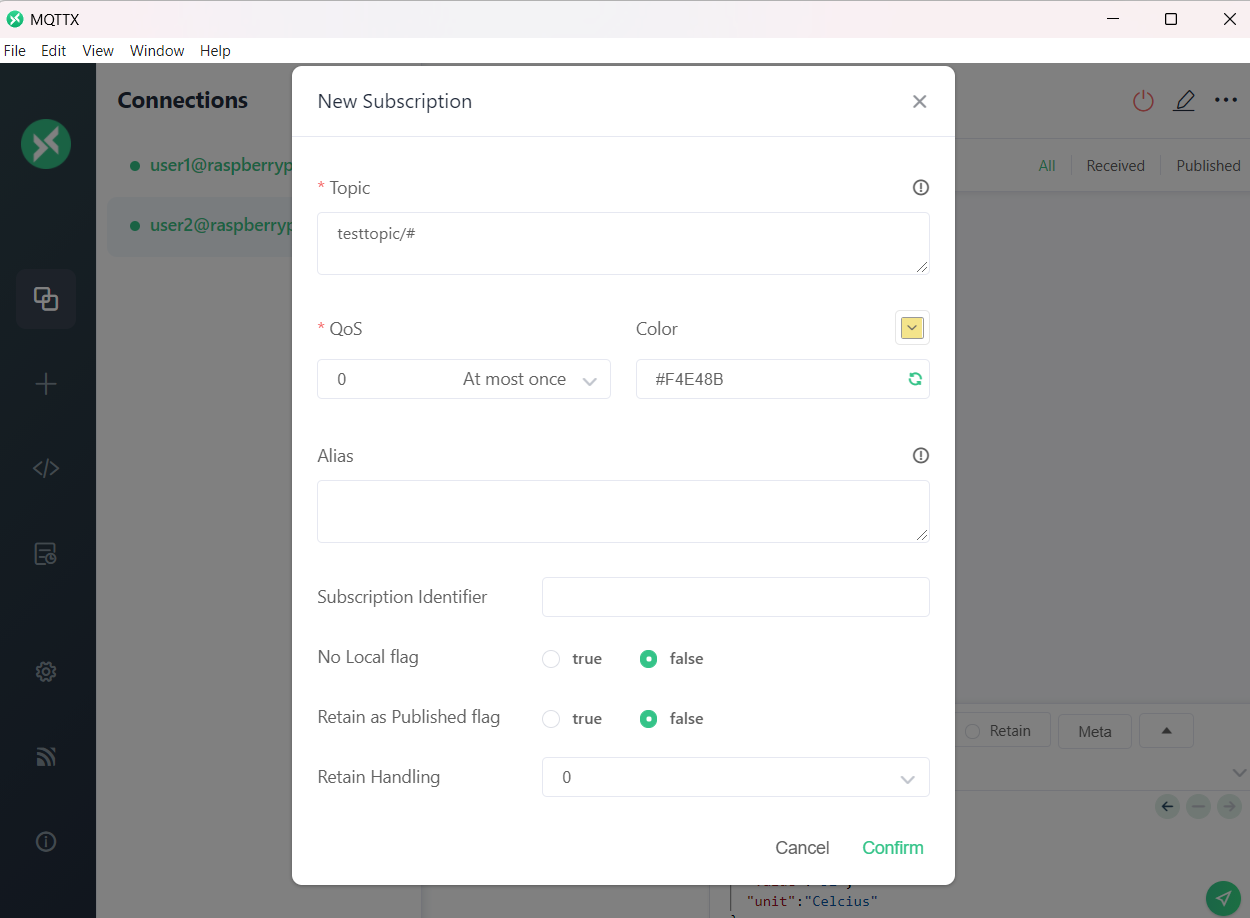
****

Gambar . Menambahkan User 1 pada MQTTX Dashboard

1. Setelah terhubung ke broker MQTT, buat topik baru di MQTTX. Topik ini adalah alamat yang digunakan untuk mengirim dan menerima pesan.

****

Gambar . Menambahkan User 2 pada MQTTX Dashboard



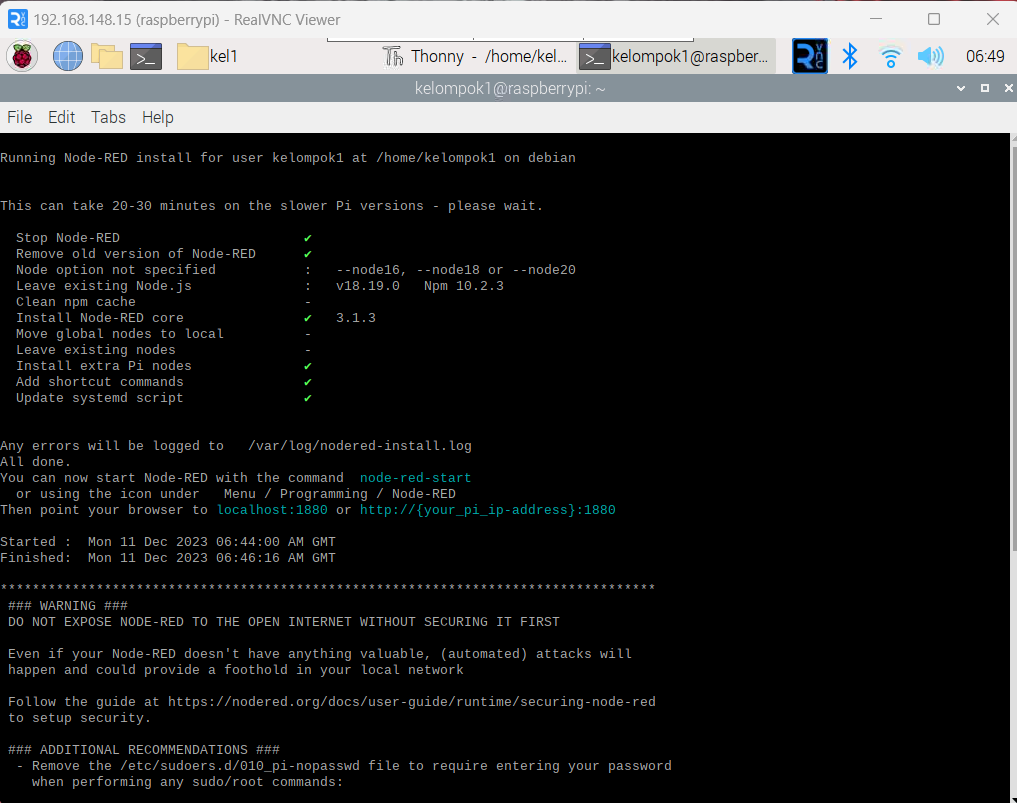
Gambar . Menambahkan Topic pada MQTTX Dashboard

1. **Setting node – red**
   1. **Install node red dengan perintah**

Sebelum menginstal Node-RED, pastikan Node.js terinstal di sistem kita. Jika tidak, kita dapat menginstalnya dengan perintah berikut:

*bash <(curl -sL* [*https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered*](https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)*)*

Skrip instalasi Node.js dan Node-RED akan diunduh dan dijalankan pada sistem dengan menggunakan perintah di atas.

****

Gambar . Tampilan Hasil Instalasi Node Red

* 1. **Jalankan node red**

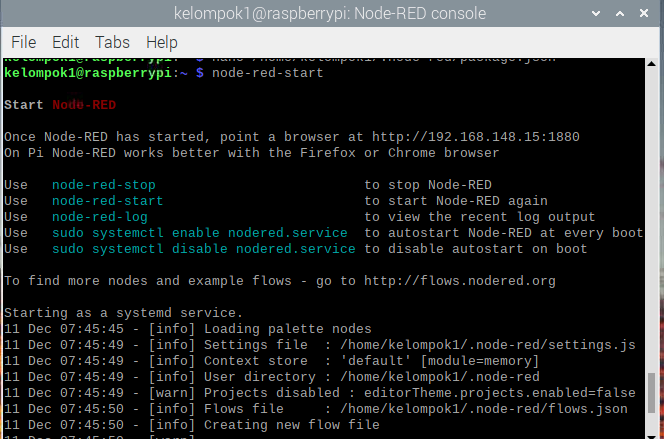
Setelah Node-RED diinstal, perintah berikut dapat digunakan untuk menjalankannya:

*node-red-start*

Setelah server Node-RED berhasil berjalan, Anda dapat mengakses antarmuka pengguna melalui browser web. Buka browser dan akses URL berikut:

*http://raspberrypi:1880*

Kita akan melihat data proses dengan perintah ini, yang akan memulai server Node-RED.

****

Gambar . Node Red Berjalan

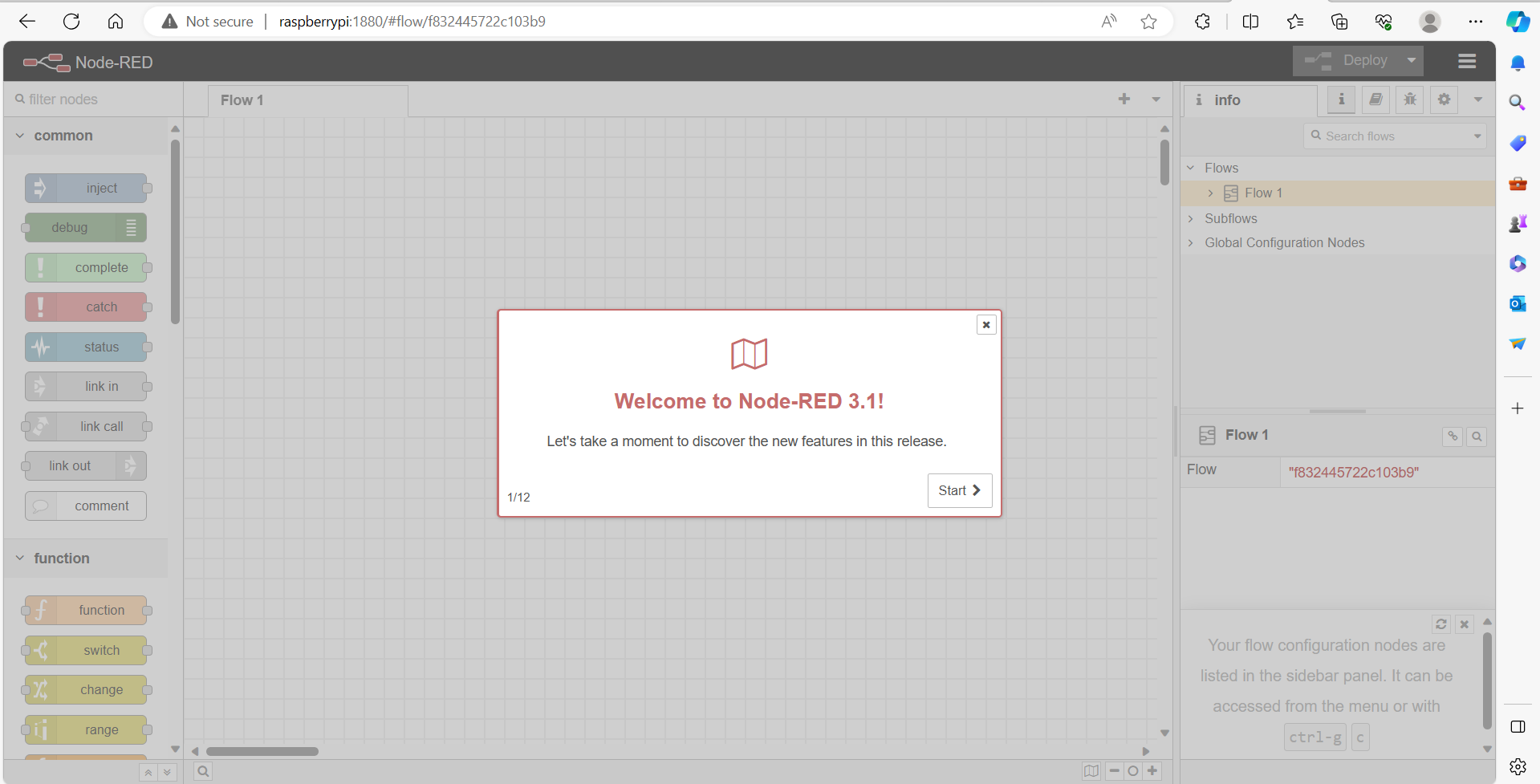
* 1. **Buka node red pada browser**

Sebelum kita dapat mengakses antarmuka pengguna Node-RED, pastikan bahwa Node-RED sudah berjalan di perangkat Raspberry Pi. Kita dapat memastikan ini dengan menjalankan perintah "node-red-start" di terminal.

Buka browser web seperti Google Chrome, Mozilla Firefox, atau Safari di perangkat Anda. Di bilah alamat browser, masukkan alamat URL berikut:

*raspberrypi:1880*

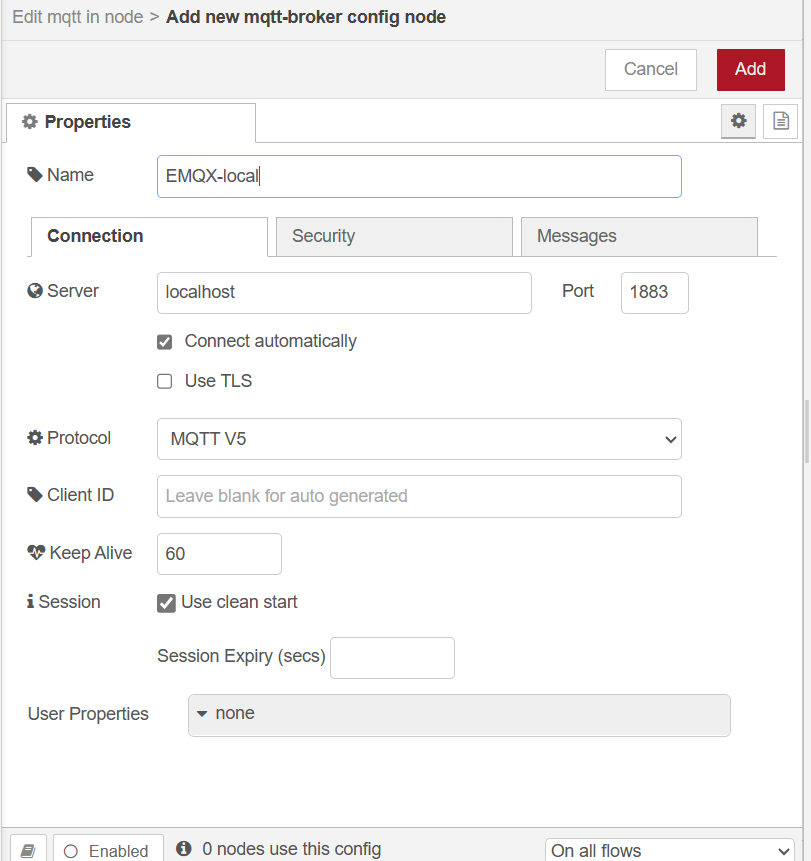
Pastikan untuk mengganti "raspberrypi" dengan alamat IP atau nama host yang sesuai dengan perangkat Raspberry Pi yang kita gunakan.

****

Gambar . Tampilan Membuka Workspace Node Red

1. **Konfigurasi untuk mqtt in**

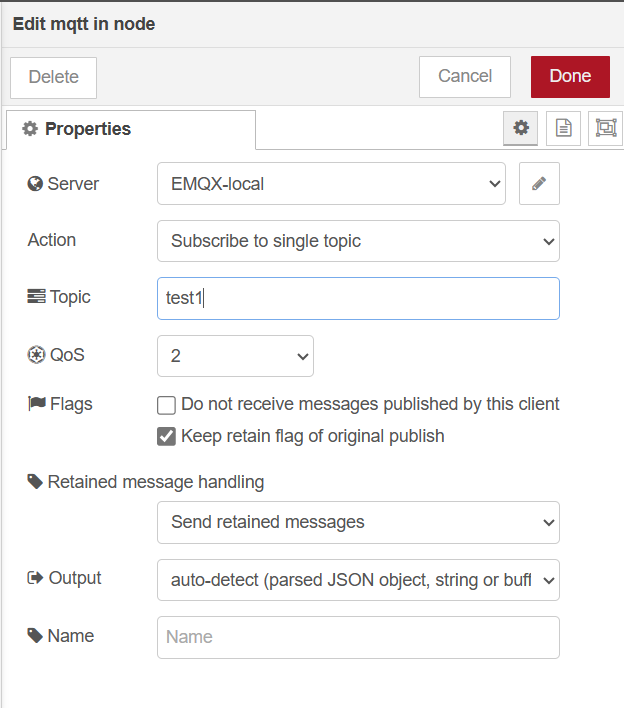
Untuk membuka jendela konfigurasi, seret node MQTT In dari panel alat ke area kerja Node-RED. Di sini, Anda harus mengatur broker MQTT yang akan digunakan. Masukkan konfigurasi yang diperlukan, seperti nama host broker MQTT, alamat IP, dan port. Selesaikan dengan menyimpan konfigurasi.

****

Gambar . Menambahkan Konfigurasi Node MQTT In Broker

Di sudut kanan atas antarmuka Node-RED, klik ikon menu. Kemudian, pilih opsi "Manage palette". Jika node "node-red-contrib-mqtt-broker" belum terpasang, cari dan pasangnya di tab "Install". Setelah itu, kita akan dapat menemukan node MQTT In di panel alat Node-RED.

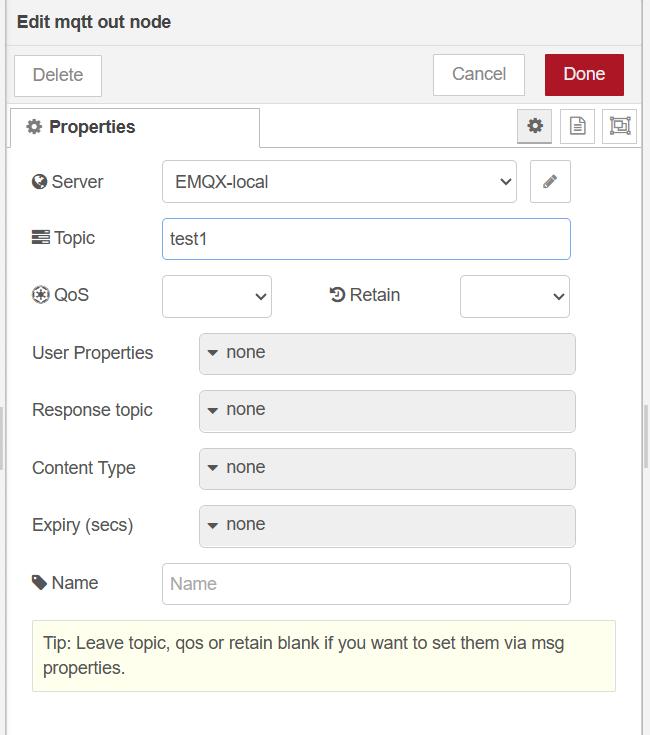
Setelah itu, Kita harus mengatur broker MQTT yang akan digunakan. Masukkan nama host broker MQTT, alamat IP, port, dan konfigurasi lainnya yang diperlukan. Setelah selesai, pastikan untuk menyimpan konfigurasi. Setelah menyelesaikan konfigurasi dan menghubungkan node MQTT In ke aliran data, kita dapat menambahkan node tambahan, seperti node debug, untuk melihat pesan yang dikirim oleh broker MQTT. Untuk menyimpan dan menjalankan aliran data, klik tombol "Deploy" di sudut kanan atas antarmuka node-RED. Node MQTT In akan terhubung ke broker MQTT yang dikonfigurasi dan mulai menerima pesan yang dikirimkan ke topik tertentu. Kami dapat melihat pesan yang diterima di node debug atau melakukan tindakan lain sesuai dengan kebutuhan proyek ini.

****

Gambar . Mengedit MQTT In

1. **Konfigurasi untuk mqtt out**

Untuk membuka jendela konfigurasi, seret node MQTT Out dari panel alat ke area kerja Node-RED. Klik dua kali pada node tersebut. Di sini, kita perlu mengatur broker MQTT, topik MQTT, dan pesan yang akan dikirim. Setelah selesai, pastikan untuk menyimpan konfigurasi. Untuk memicu pengiriman pesan MQTT, Anda dapat menghubungkan node MQTT Out dengan node lain di aliran data. Kita juga dapat menambahkan node tambahan, seperti node injeksi. Node MQTT Out akan terhubung ke broker MQTT yang dikonfigurasi dan mulai mengirim pesan ke topik tertentu. Ini memastikan bahwa broker MQTT dapat menerima pesan yang dikirim.

****

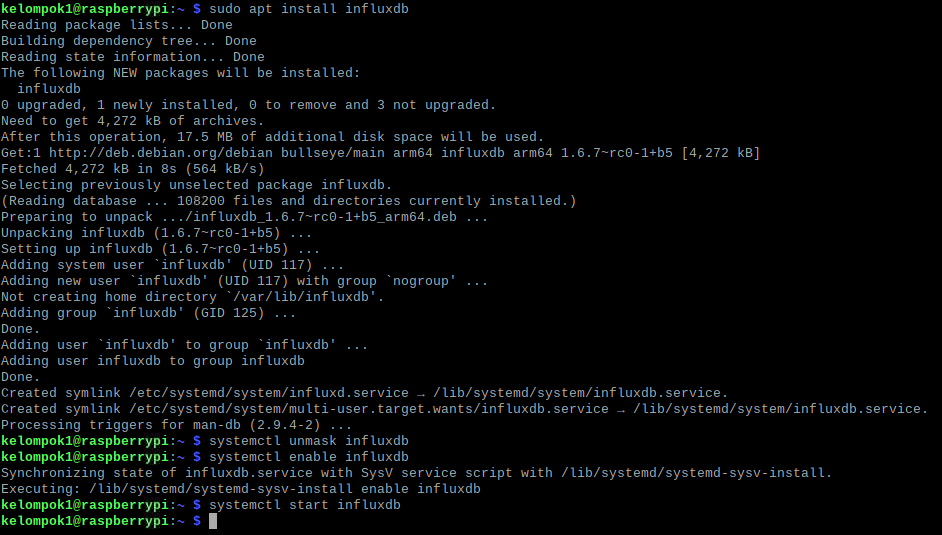
Gambar . Menambahkan Konfigurasi Node MQTT Out Broker

1. **Influxdb**

* **Instalasi dan konfigurasi influxdb**

Untuk menginstal InfluxDB pada sistem operasi Anda, buka terminal dengan perintah berikut :

*bash sudo apt install influxdb*

****

Gambar . Menginstall InfluxDB

* **Instalasi Client InfluxDB**

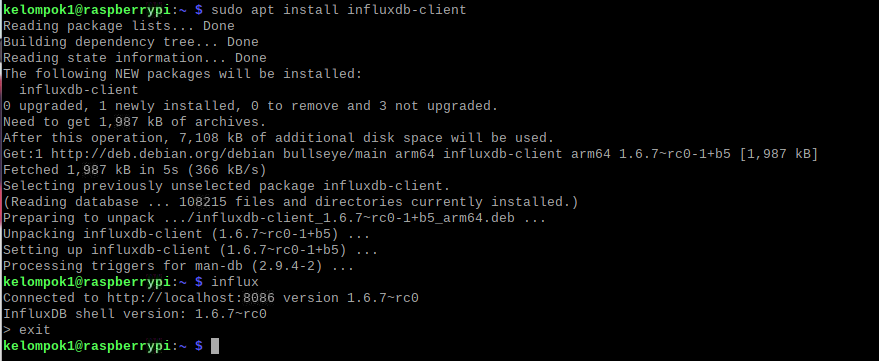
Kami dapat menginstal InfluxDB server dan juga menginstal InfluxDB client untuk berinteraksi dengan server InfluxDB. Untuk melakukannya, gunakan perintah berikut:

*sudo apt install influxdb-client*

Setelah instalasi dan konfigurasi selesai, kita dapat mengakses InfluxDB menggunakan InfluxDB client. Untuk masuk ke shell InfluxDB, gunakan perintah berikut:

*influx*

Ini akan membuka antarmuka shell InfluxDB. Di sana, kita dapat memulai pembuatan database, mengatur pengguna, dan melakukan tugas lainnya.

****

Gambar . Menginstall Client InfluxDB

Setelah instalasi selesai, kita perlu mengonfigurasi InfluxDB. File konfigurasi InfluxDB terletak di /etc/influxdb/influxdb.conf. Kita dapat mengedit file ini dengan editor teks seperti Nano atau Vim dengan perintah berikut :

*sudo nano /etc/influxdb/influxdb.conf.*

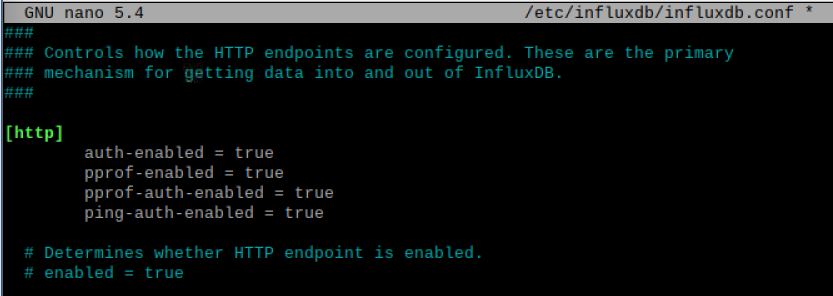
File direktori berisikan perintah berikut :

*auth-enabled = true*

*pprof-enabled = true*

*pprof-auth-enabled = true*

*ping-auth-enabled = true*

****

Gambar . Mengkonfigurasi InfluxDB

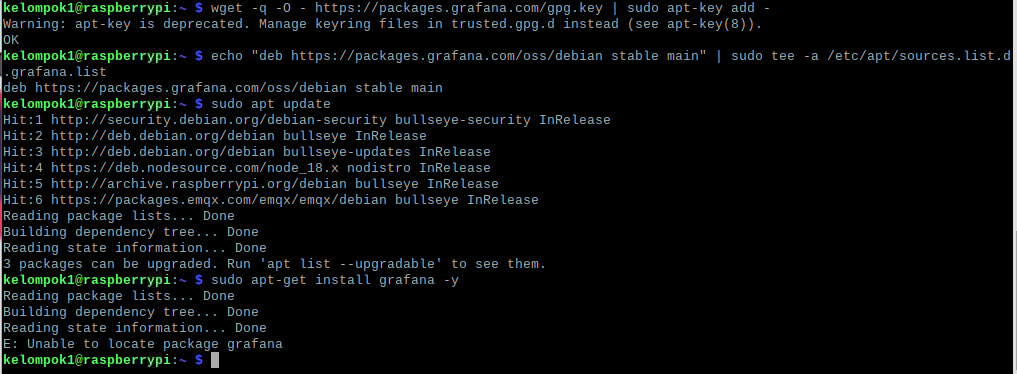
Kita memiliki kemampuan untuk mengatur parameter seperti port yang digunakan, lokasi penyimpanan data, dan parameter lainnya di dalam file konfigurasi sesuai dengan kebutuhan proyek ini. Setelah kita membuat perubahan, simpan perubahan dan keluar dari editor.

1. **Grafana**

Grafana adalah platform open-source yang memungkinkan untuk melihat metrik dan memantau data.

* **Install Grafana**

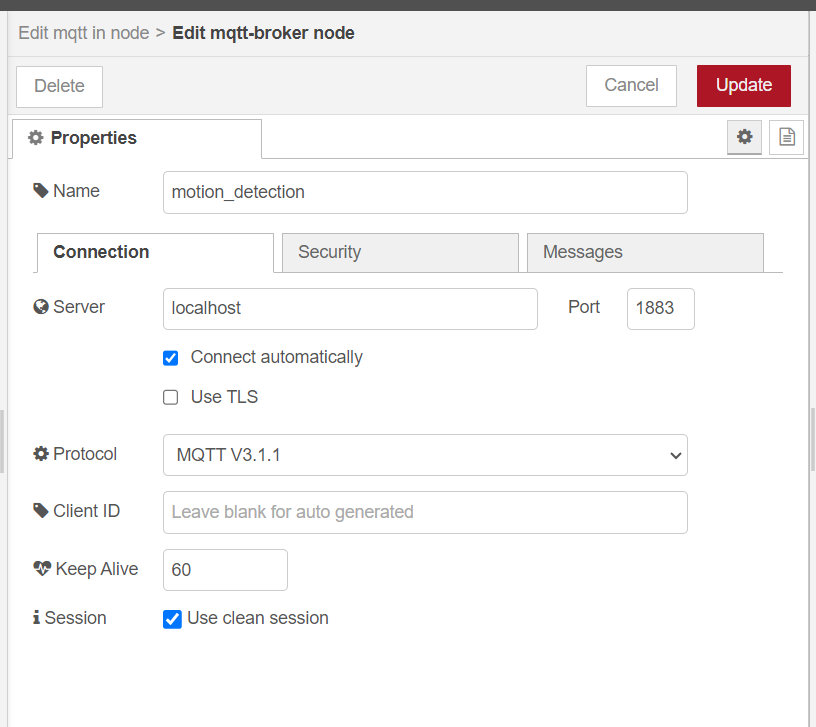
Menginstal grafana biasanya dilakukan dengan cara berikut:

****

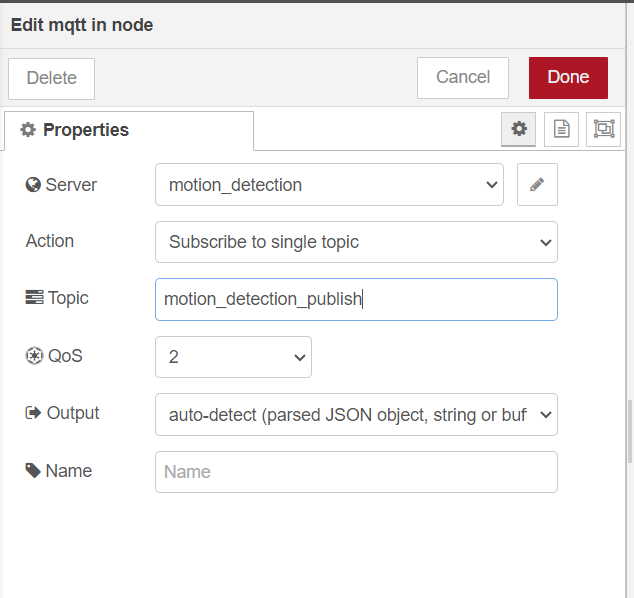
Gambar . Menginstalasi Grafana

* **Kemudian edit mqtt broker nya**

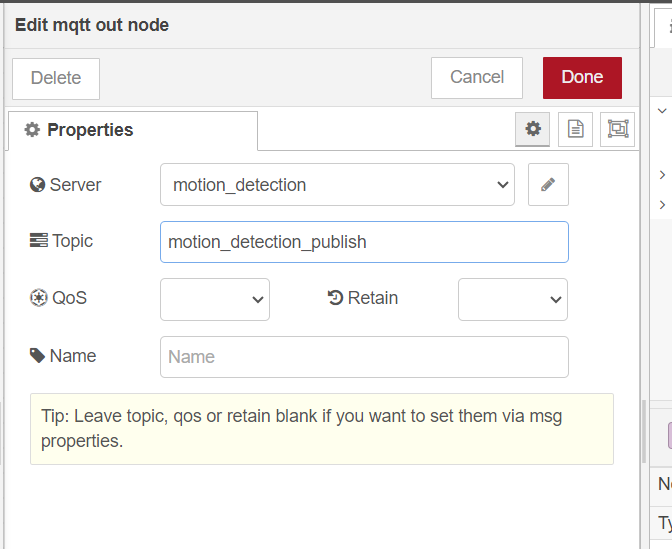
Dilakukan dengan cara berikut :

****

Gambar . Mengatur Connection pada MQTT Broker

****

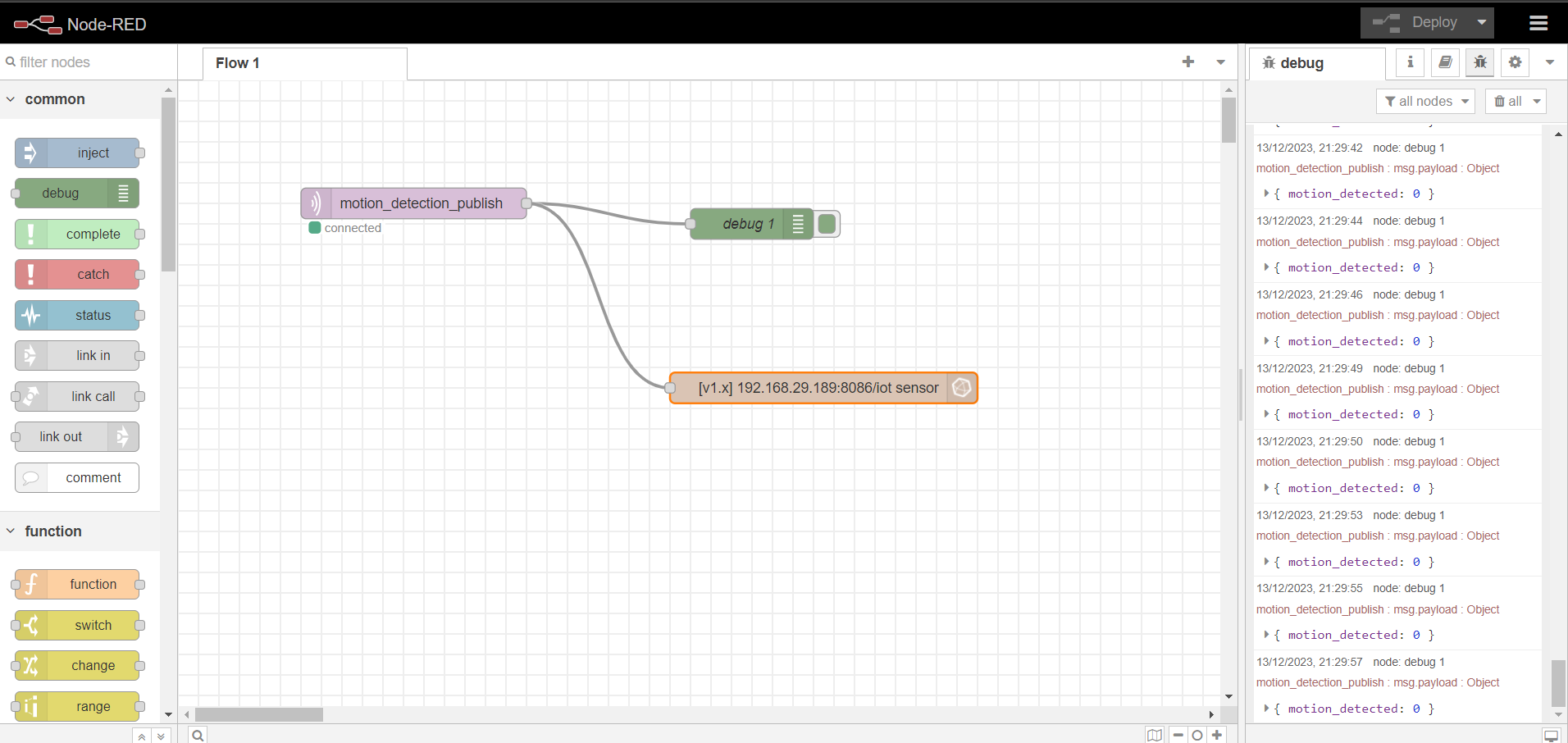
Gambar . Mengatur Security pada MQTT Broker

****

Gambar . Mengatur Messages pada MQTT Broker

* **Mengatur aliran data di area kerja utama Node-Red**

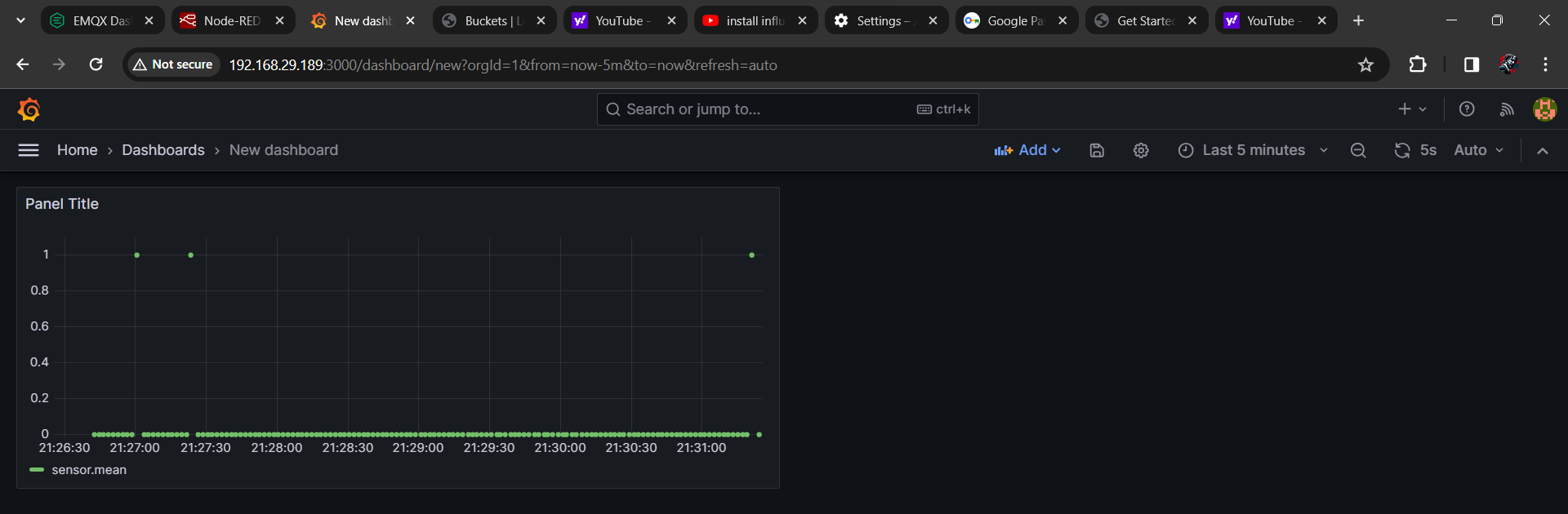
Area kerja ini disebut dengan workspace yang berbentuk grid atau canvas kosong yang diisi dengan node dan dihubungkan dengan kabel. Berikut adalah gambaran aliran data yang dibuat di Node-red :

****

Gambar . Mengatur Aliran Data di Workspace Node Red

* **Penyusunan di Dashboard**

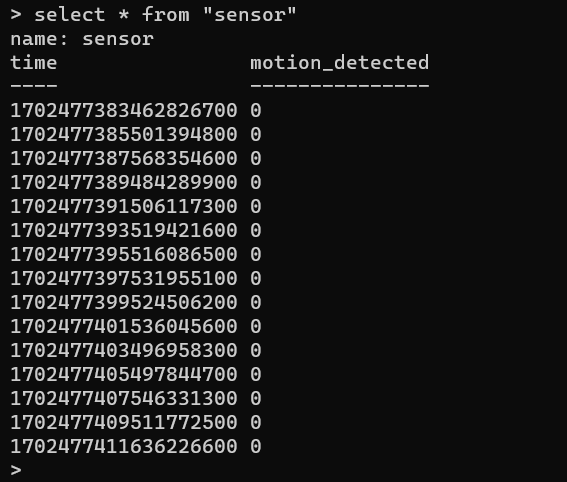
Setelah membuat panel, kemudian dibuat penyusunan dashboard dengan menambahkan panel ke halaman dashboard. Kita dapat menambahkan panel baru dengan menyeret dan melepaskan panel dari menu samping halaman dashboard.

****

Gambar . Penyusunan di Dashboard Grafana

Setelah menyelesaikan dashboard, kita dapat mempublikasikannya untuk digunakan oleh orang lain. kita dapat membagikan tautan dashboard atau menambahkan dashboard.

* **Melihat Data Kueri**

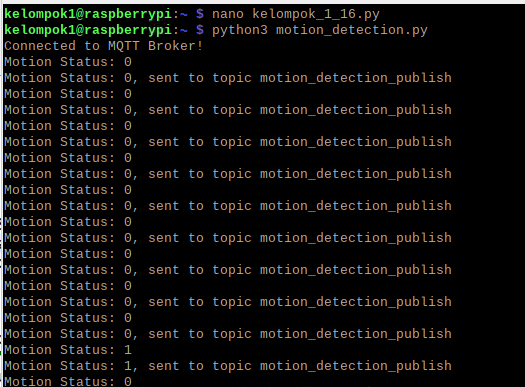
****

Gambar . Melihat Data Kueri

Kueri ini menggunakan pola ekspresi reguler untuk mencocokkan karakter non-numerik apa pun di kolom -motion detected- dan menghapus baris yang cocok dengan pola tersbeut. Kueri ini juga memperbarui nilai non-numerik di kolom motion-detected menjaddi 0 agar valid .

* **Menjalankan skrip Python**

Digunakan untuk mendeteksi gerakan.

****

Gambar . Menjalankan Script Python Kelompok 1

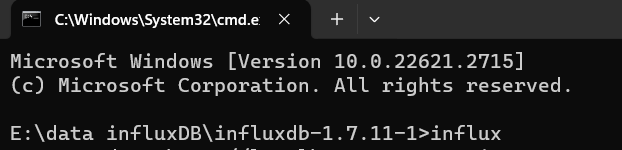
Berikut adalah penjelaan untuk setiap perintah :

*nano kelompok1\_16.py*

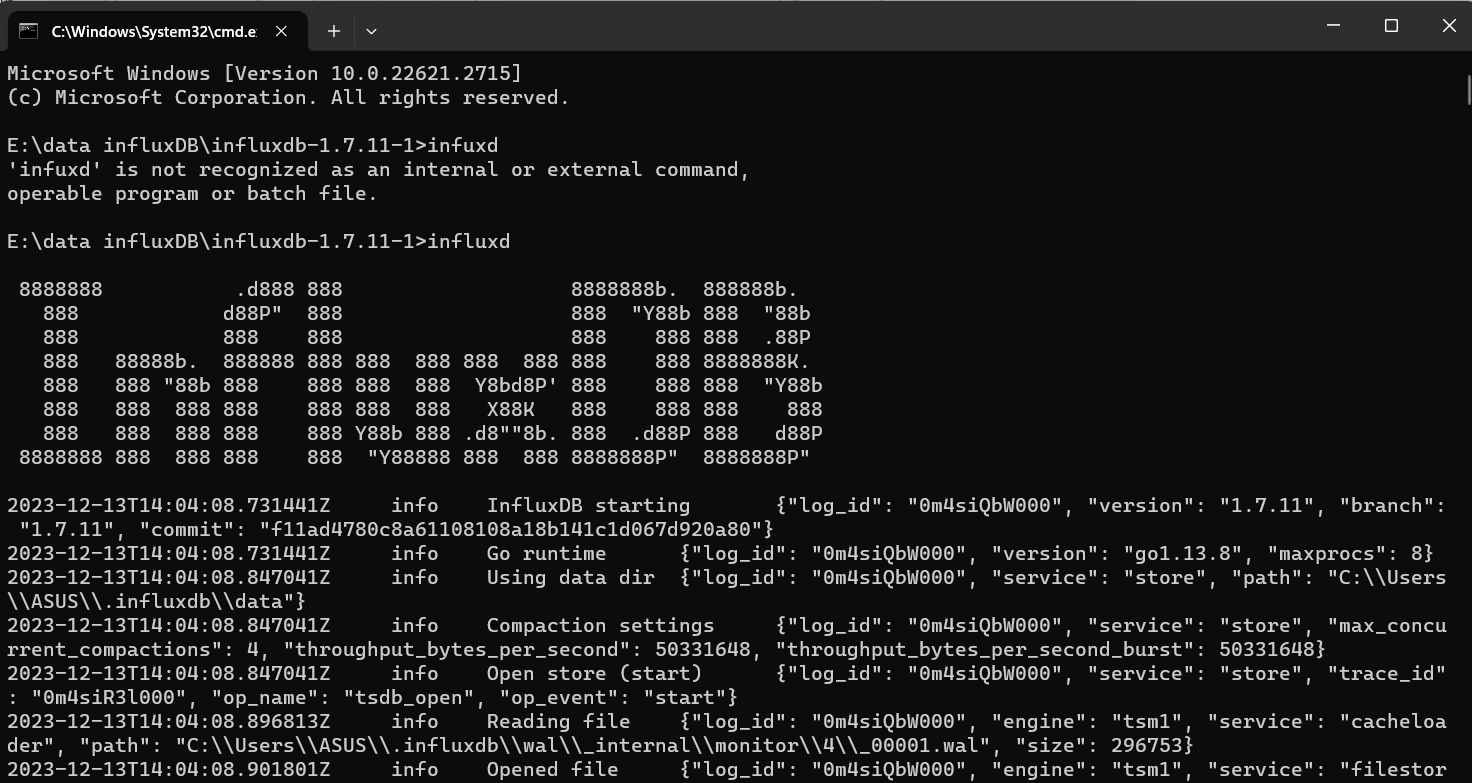
Perintah ini digunakan untuk membuka atau membuat file dengan nama “*kelompok1\_16.py”* dengan menggunakan editor teks nano untuk menulis atau mengedit skrip Python yang akan digunakan untuk mendeteksi gerakan . Setelah menjalankan perintah ini, editor nano akan terbuka, dan dapat mulai menulis atau mengedit kode Python di dalamnya. Setelah selesai, dapat menyimpan dan keluar dari editor nano.

* **Menjalankan InfluxDB dengan menggunakan CMD**

Berikut adalah tampilan CMD saat InfluxDB dijalankan

****

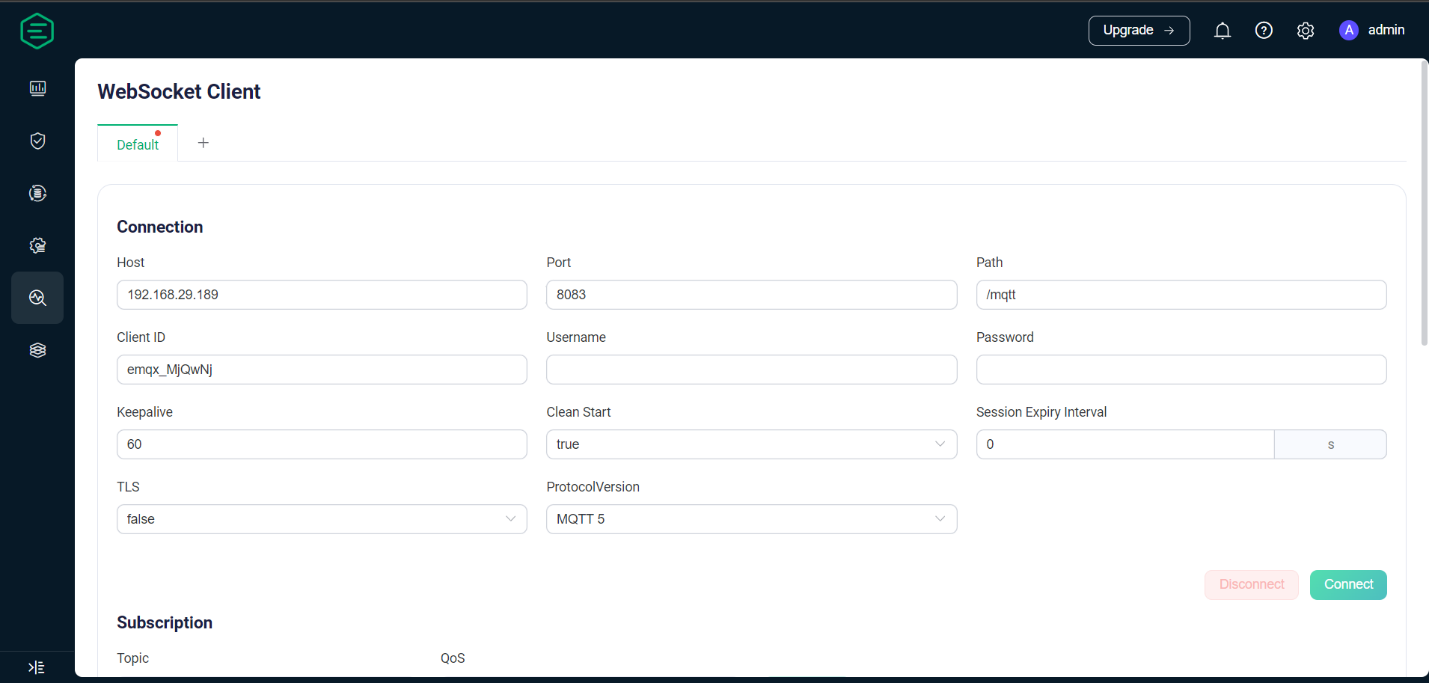
Gambar . Tampilan CMD saat InfluxDB dibuka

****

Gambar . Tampilan CMD saat InfluxDB dijalankan

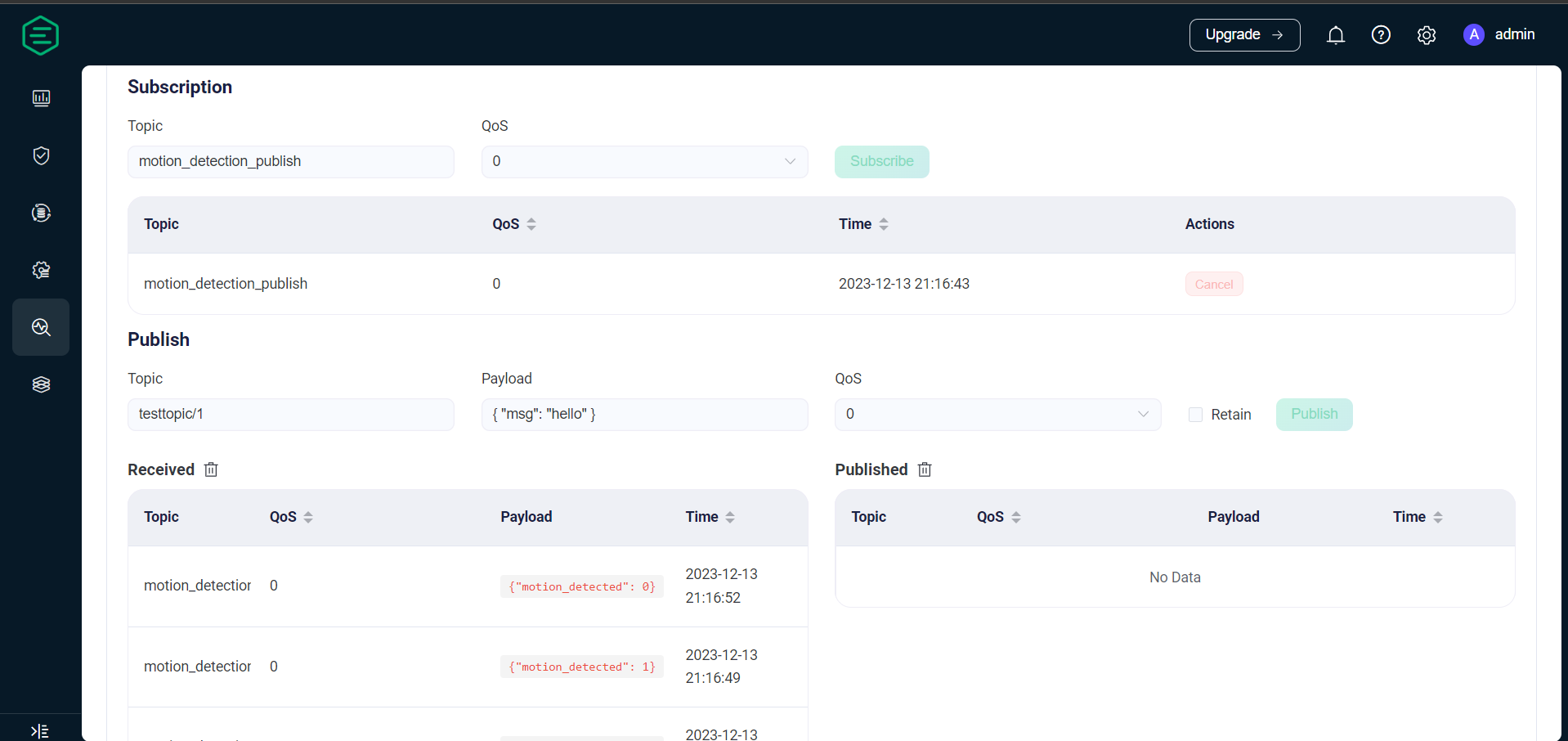
* **Membuat Koneksi WebSocket**

Setelah lingkungan pengembangan selesai, langkah berikutnya adalah menghubungkan WebSocket ke server. Untuk melakukan ini, kita perlu menggunakan API WebSocket yang disediakan oleh bahasa pemrograman yang kita gunakan. Berikut adalah tampilan dari WebSocket Client pada proyek ini :

Gambar . Mengedit Koneksi pada WorkSocket Client

* **Mengirim dan Menerima Pesan**

Setelah membuat koneksi WebSocket, kita dapat mengirim dan menerima pesan secara real-time antara client dan server. Kita dapat mengirim pesan ke server dan menangani pesan yang diterima dari server dengan menggunakan metode yang disediakan oleh API WebSocket.

****

Gambar . Mengedit Subscription Pengiriman dan Penerimaan Pesan