

# **Laporan Proyek**

## **Teknologi Internet of Things**

### **( *Pendeteksi Hama* )**



Kelompok 01

13321002 Thalia Aniceta Saragih  
13321008 Dion Saputra Manurung  
13321043 Louis Cristiano Panggabean

**INSTITUT TEKNOLOGI DEL**

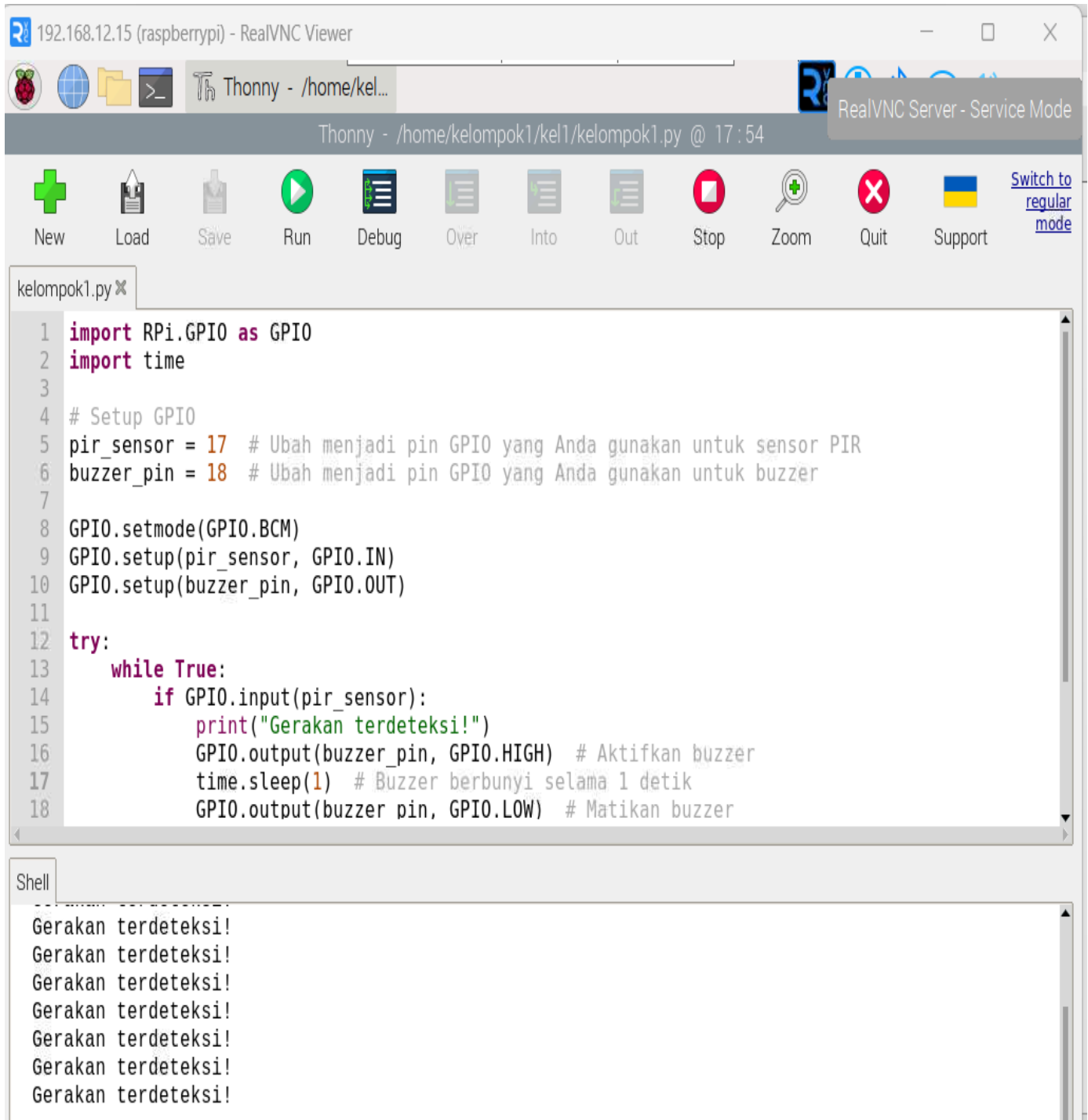
**FAKULTAS VOKASI**

## DAFTAR GAMBAR

Gambar 1. Program yang Dijalankan.....	3
Gambar 2. Rangkaian Alat.....	5
Gambar 3. Node Red berjalan di CMD .....	6
Gambar 4. EMQX berjalan di CMD.....	7
Gambar 5. InfluxDB dijalankan di CMD .....	8
Gambar 6. Mendownload EMQX Repository .....	9
Gambar 7. Menginstall EMQX dengan menggunakan perintah.....	10
Gambar 8. Menghubungkan File Konfigurasi dengan Symlink .....	10
Gambar 9. Menjalankan EMQX dari Raspberry Pi.....	10
Gambar 10. Memantau atau Mengelola Broker MQTT .....	11
Gambar 11. Mengatur Mechanism pada Authentication .....	12
Gambar 12. Mengatur Backend pada Authentication .....	13
Gambar 13. Mengatur Configuration pada Authetication.....	13
Gambar 14. Menambahkan User 1 pada Dashboard EMQX.....	14
Gambar 15. Menambahkan User 2 pada Dashboard EMQX.....	15
Gambar 16. Menambahkan Topik Metrics .....	16
Gambar 17. Menambahkan User 1 pada MQTTX Dashboard .....	17
Gambar 18. Menambahkan User 2 pada MQTTX Dashboard .....	18
Gambar 19. Menambahkan Topic pada MQTTX Dashboard.....	18
Gambar 20. Tampilan Hasil Instalasi Node Red.....	19
Gambar 21. Node Red Berjalan .....	20
Gambar 22. Tampilan Membuka Workspace Node Red.....	21
Gambar 23. Menambahkan Konfigurasi Node MQTT In Broker .....	22
Gambar 24. Mengedit MQTT In.....	23
Gambar 25. Menambahkan Konfigurasi Node MQTT Out Broker.....	24
Gambar 26. Menginstall InfluxDB .....	25
Gambar 27. Menginstall Client InfluxDB .....	26
Gambar 28. Mengkonfigurasi InfluxDB.....	27
Gambar 29. Menginstalasi Grafana .....	28
Gambar 30. Mengatur Connection pada MQTT Broker.....	29
Gambar 31. Mengatur Security pada MQTT Broker.....	29
Gambar 32. Mengatur Messages pada MQTT Broker.....	30
Gambar 33. Mengatur Aliran Data di Workspace Node Red.....	30
Gambar 34. Penyusunan di Dashboard Grafana .....	31
Gambar 35. Melihat Data Kueri.....	32
Gambar 36. Menjalankan Script Python Kelompok 1 .....	33
Gambar 37. Tampilan CMD saat InfluxDB dibuka .....	33
Gambar 38. Tampilan CMD saat InfluxDB dijalankan .....	34
Gambar 39. Mengedit Koneksi pada WorkSocket Client .....	35
Gambar 40. Mengedit Subscription Pengiriman dan Penerimaan Pesan.....	35

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

Program dijalankan :



```
1 import RPi.GPIO as GPIO
2 import time
3
4 # Setup GPIO
5 pir_sensor = 17 # Ubah menjadi pin GPIO yang Anda gunakan untuk sensor PIR
6 buzzer_pin = 18 # Ubah menjadi pin GPIO yang Anda gunakan untuk buzzer
7
8 GPIO.setmode(GPIO.BCM)
9 GPIO.setup(pir_sensor, GPIO.IN)
10 GPIO.setup(buzzer_pin, GPIO.OUT)
11
12 try:
13     while True:
14         if GPIO.input(pir_sensor):
15             print("Gerakan terdeteksi!")
16             GPIO.output(buzzer_pin, GPIO.HIGH) # Aktifkan buzzer
17             time.sleep(1) # Buzzer berbunyi selama 1 detik
18             GPIO.output(buzzer pin, GPIO.LOW) # Matikan buzzer
```

Gerakan terdeteksi!  
Gerakan terdeteksi!  
Gerakan terdeteksi!  
Gerakan terdeteksi!  
Gerakan terdeteksi!  
Gerakan terdeteksi!  
Gerakan terdeteksi!

Gambar 1. 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**

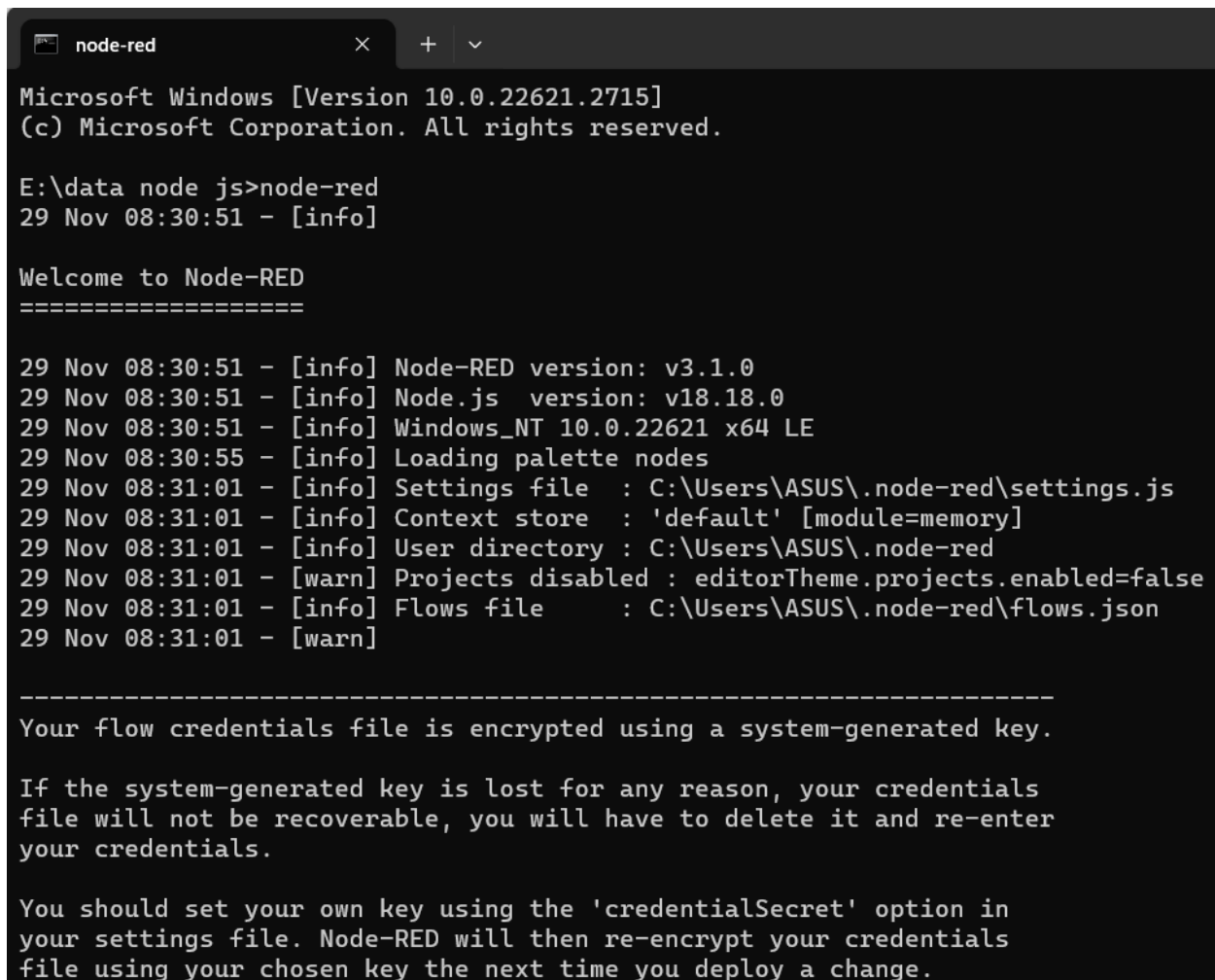
Berikut adalah rangkaian yang sudah di lakukan atau diterapkan



Gambar 2. Rangkaian Alat

- **Menjalankan node red dengan perintah : 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.



```
node-red
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. All rights reserved.

E:\data node js>node-red
29 Nov 08:30:51 - [info]

Welcome to Node-RED
=====

29 Nov 08:30:51 - [info] Node-RED version: v3.1.0
29 Nov 08:30:51 - [info] Node.js version: v18.18.0
29 Nov 08:30:51 - [info] Windows_NT 10.0.22621 x64 LE
29 Nov 08:30:55 - [info] Loading palette nodes
29 Nov 08:31:01 - [info] Settings file : C:\Users\ASUS\.node-red\settings.js
29 Nov 08:31:01 - [info] Context store : 'default' [module=memory]
29 Nov 08:31:01 - [info] User directory : C:\Users\ASUS\.node-red
29 Nov 08:31:01 - [warn] Projects disabled : editorTheme.projects.enabled=false
29 Nov 08:31:01 - [info] Flows file : C:\Users\ASUS\.node-red\flows.json
29 Nov 08:31:01 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
```

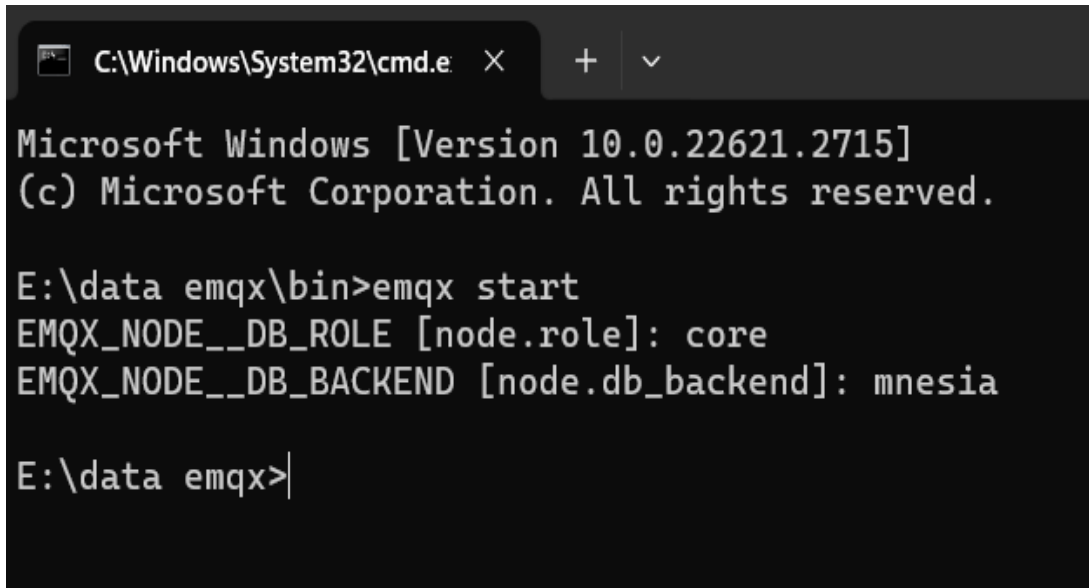
Gambar 3. Node Red berjalan di CMD

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, anda 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.



```
C:\Windows\System32\cmd.e X + v
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. All rights reserved.

E:\data emqx\bin>emqx start
EMQX_NODE__DB_ROLE [node.role]: core
EMQX_NODE__DB_BACKEND [node.db_backend]: mnesia

E:\data emqx>
```

Gambar 4. 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.

```
C:\Windows\System32\cmd.e X + v
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. All rights reserved.

E:\Aplikasi install\IOT\InfluxDB\influxdb-1.7.11-1>influxd

88888888      .d888 888              88888888b. 8888888b.
888          d88P" 888              888  "Y88b 888  "88b
888          888  888              888  888 888  .88P
888 888888b. 888888 888 888 888 888 888 888 88888888K.
888 888 "88b 888 888 888 888 Y8bd8P' 888 888 888 "Y88b
888 888 888 888 888 888 888 X88K 888 888 888 888
888 888 888 888 888 Y88b 888 .d8""8b. 888 .d88P 888 d88P
88888888 888 888 888 888 "Y88888 888 888 8888888P" 8888888P"

2023-11-29T02:31:47.863425Z info InfluxDB starting {"log_id": "0lnEXqal000", "version": "1.7.11", "branch":
"1.7.11", "commit": "f11ad4780c8a61108108a18b141c1d067d920a80"}
2023-11-29T02:31:47.863425Z info Go runtime {"log_id": "0lnEXqal000", "version": "go1.13.8", "maxprocs": 8}
2023-11-29T02:31:47.973482Z info Using data dir {"log_id": "0lnEXqal000", "service": "store", "path": "C:\\Users
\\ASUS\\.influxdb\\data"}
2023-11-29T02:31:47.974535Z info Compaction settings {"log_id": "0lnEXqal000", "service": "store", "max_concu
rrent_compactions": 4, "throughput_bytes_per_second": 50331648, "throughput_bytes_per_second_burst": 50331648}
2023-11-29T02:31:47.974535Z info Open store (start) {"log_id": "0lnEXqal000", "service": "store", "trace_id"
: "0lnEXr1W000", "op_name": "tsdb_open", "op_event": "start"}
2023-11-29T02:31:48.038441Z info Reading file {"log_id": "0lnEXqal000", "engine": "tsm1", "service": "cacheloa
der", "path": "C:\\Users\\ASUS\\.influxdb\\wal\\_internal\\monitor\\2\\_00001.wal", "size": 1593002}
2023-11-29T02:31:48.079331Z info Opened file {"log_id": "0lnEXqal000", "engine": "tsm1", "service": "filestor
e", "path": "C:\\Users\\ASUS\\.influxdb\\data\\iot\\autogen\\3\\000000001-000000001.tsm", "id": 0, "duration": "1.994ms"
}
2023-11-29T02:31:48.084323Z info Opened shard {"log_id": "0lnEXqal000", "service": "store", "trace_id": "0lnEX
r1W000", "op_name": "tsdb_open", "index_version": "inmem", "path": "C:\\Users\\ASUS\\.influxdb\\data\\iot\\autogen\\3",
```

Gambar 5. 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>

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.

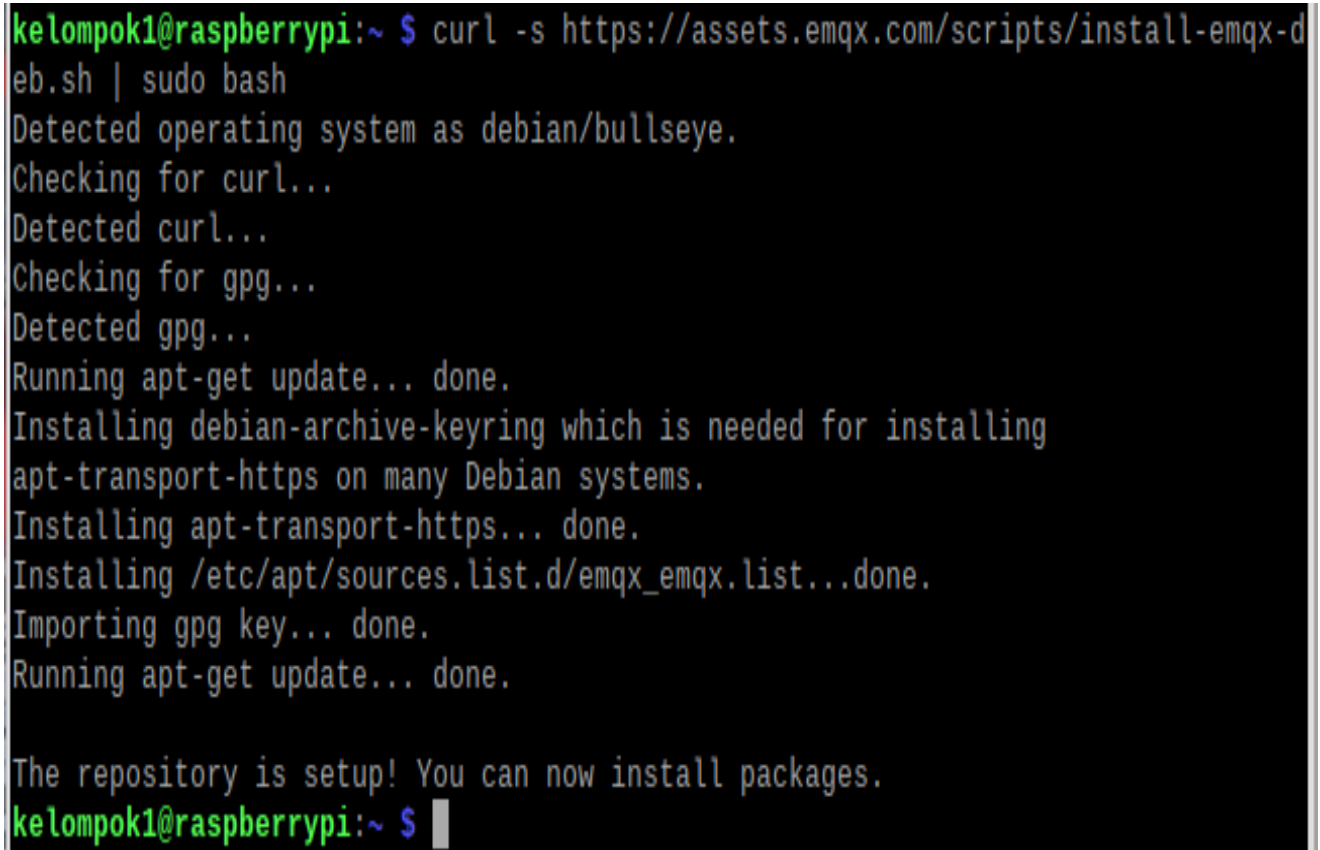
## Lalu setting Raspberry pi

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`



```
kelompok1@raspberrypi:~ $ curl -s https://assets.emqx.com/scripts/install-emqx-deb.sh | sudo bash
Detected operating system as debian/bullseye.
Checking for curl...
Detected curl...
Checking for gpg...
Detected gpg...
Running apt-get update... done.
Installing debian-archive-keyring which is needed for installing
apt-transport-https on many Debian systems.
Installing apt-transport-https... done.
Installing /etc/apt/sources.list.d/emqx_emqx.list...done.
Importing gpg key... done.
Running apt-get update... done.

The repository is setup! You can now install packages.
kelompok1@raspberrypi:~ $
```

Gambar 6. Mendownload EMQX Repository

Perintah ini akan mengunduh dan menjalankan skrip instalasi EMQX secara otomatis

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

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 emqx`, 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 :

```
kelompok1@raspberrypi:~ $ sudo apt-get install emqx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfuse2
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  emqx
0 upgraded, 1 newly installed, 0 to remove and 2 not upgraded.
Need to get 40.1 MB of archives.
After this operation, 94.9 MB of additional disk space will be used.
Get:1 https://packages.emqx.com/emqx/emqx/debian bullseye/main arm64 emqx arm64
5.3.2 [40.1 MB]
Fetched 40.1 MB in 19s (2,101 kB/s)
Selecting previously unselected package emqx.
(Reading database ... 100030 files and directories currently installed.)
Preparing to unpack .../archives/emqx_5.3.2_arm64.deb ...
Unpacking emqx (5.3.2) ...
Setting up emqx (5.3.2) ...
Adding group `emqx' (GID 124) ...
Done.
```

Gambar 7. Menginstall EMQX dengan menggunakan perintah

```
Created symlink /etc/systemd/system/multi-user.target.wants/emqx.service → /lib/
systemd/system/emqx.service.
kelompok1@raspberrypi:~ $
```

Gambar 8. 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 :

```
kelompok1@raspberrypi:~ $ sudo systemctl start emqx
kelompok1@raspberrypi:~ $
```

Gambar 9. 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**

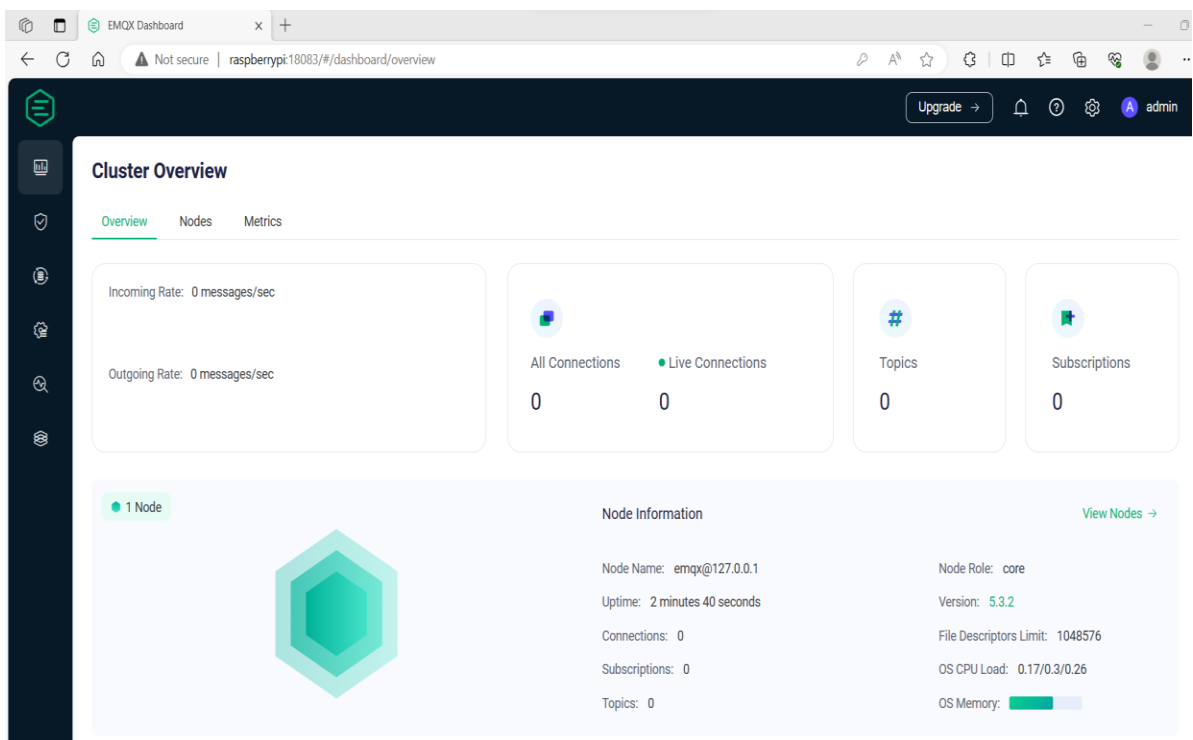
Masukkan alamat ini di bilah alamat browser Anda:

*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

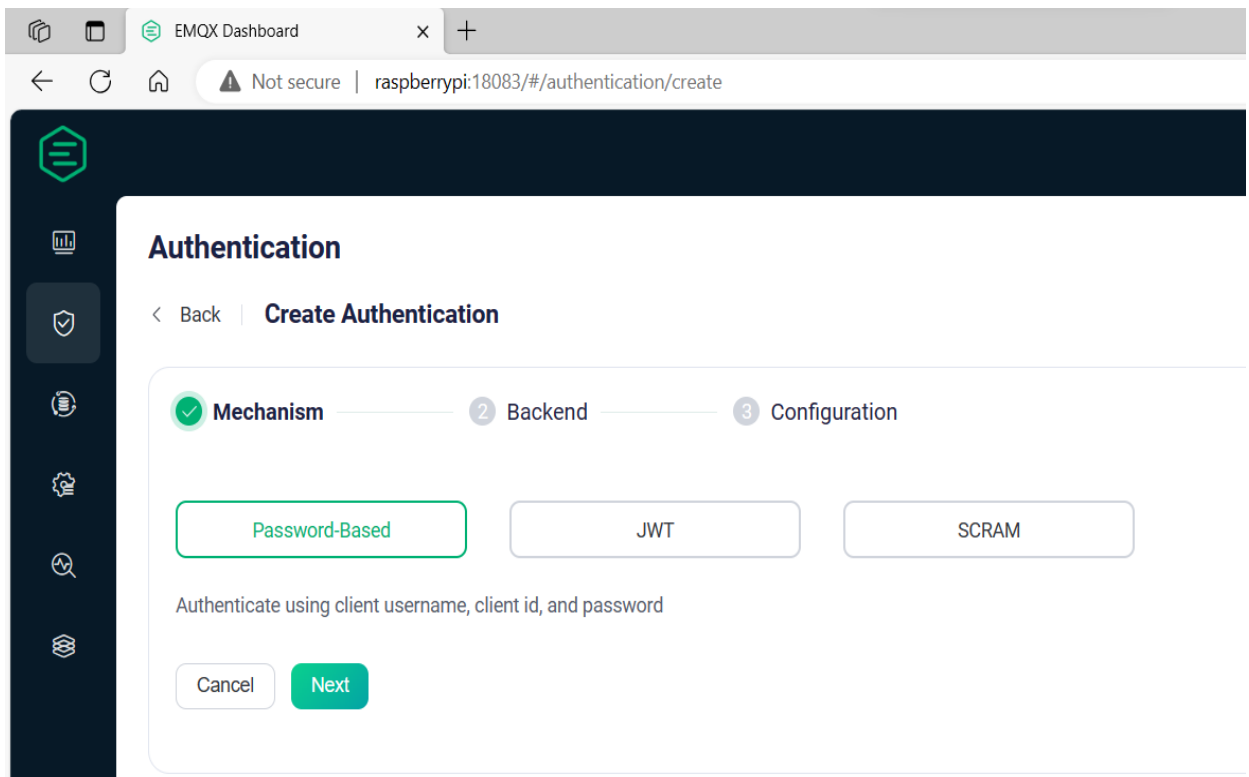


Gambar 10. Memantau atau Mengelola Broker MQTT

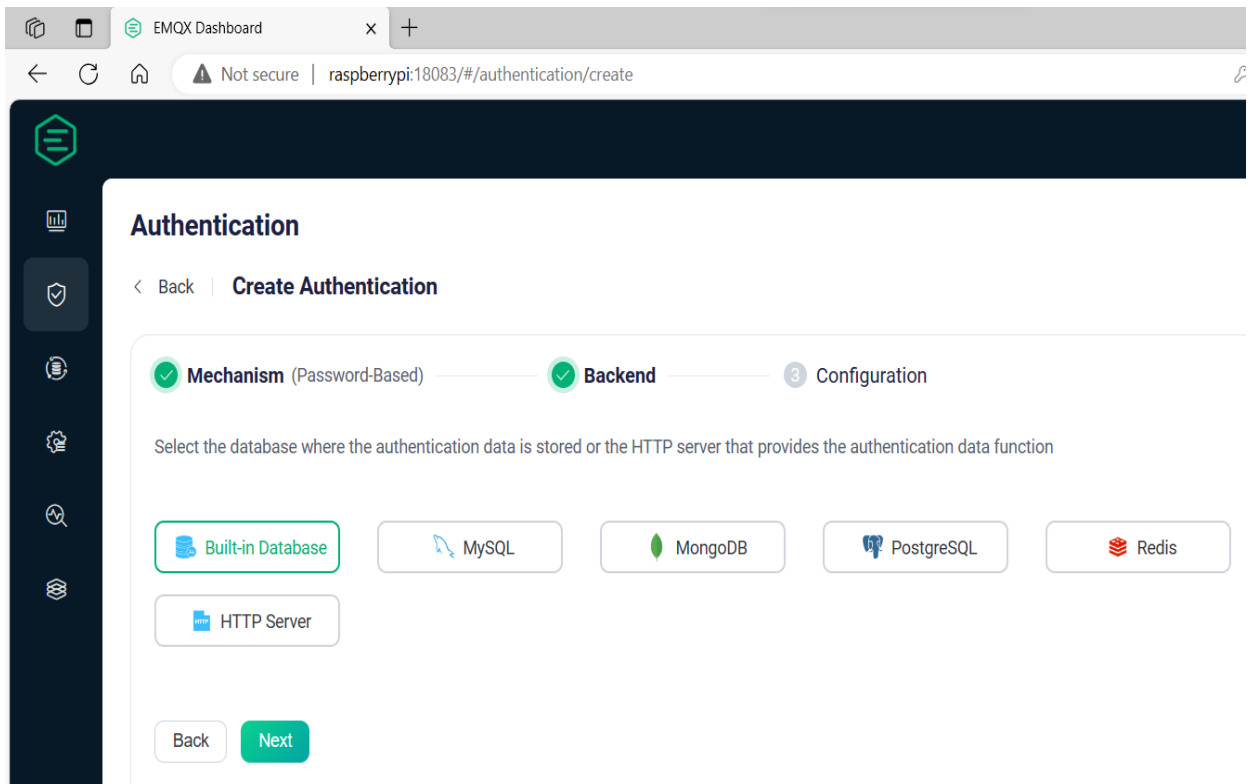
Setelah memasuki, kita seharusnya dapat mengakses dashboard EMQX, sehingga kita dapat melanjutkan untuk mengkonfigurasi dan mengelola broker sesuai kebutuhan praktikum atau proyek ini.

#### ▪ Create authentication

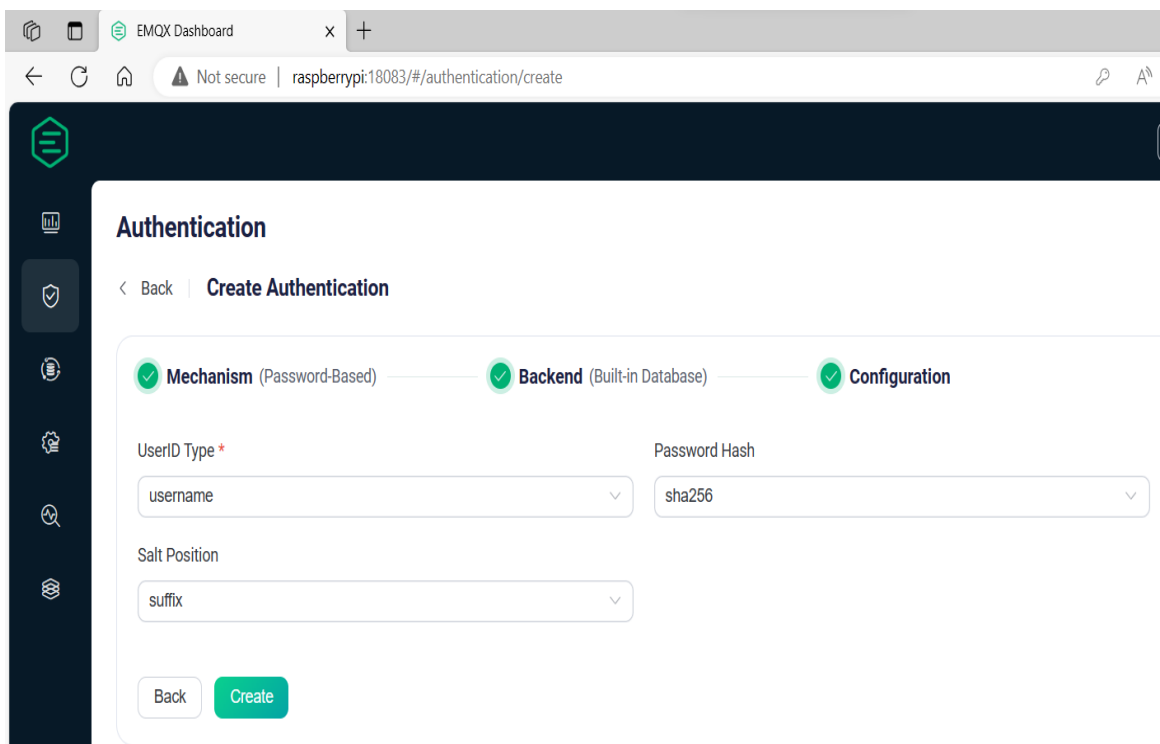
- Klik pada tab "Authentication & Authorization" di menu sebelah kiri.
- Pilih "Authentication" dari submenu.
- Klik tombol "Add" untuk menambahkan autentikasi baru.
- Isi informasi yang diperlukan, seperti username dan password untuk autentikasi.
- Klik tombol "Add" untuk menyimpan autentikasi baru.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



Gambar 11. Mengatur Mechanism pada Authentication



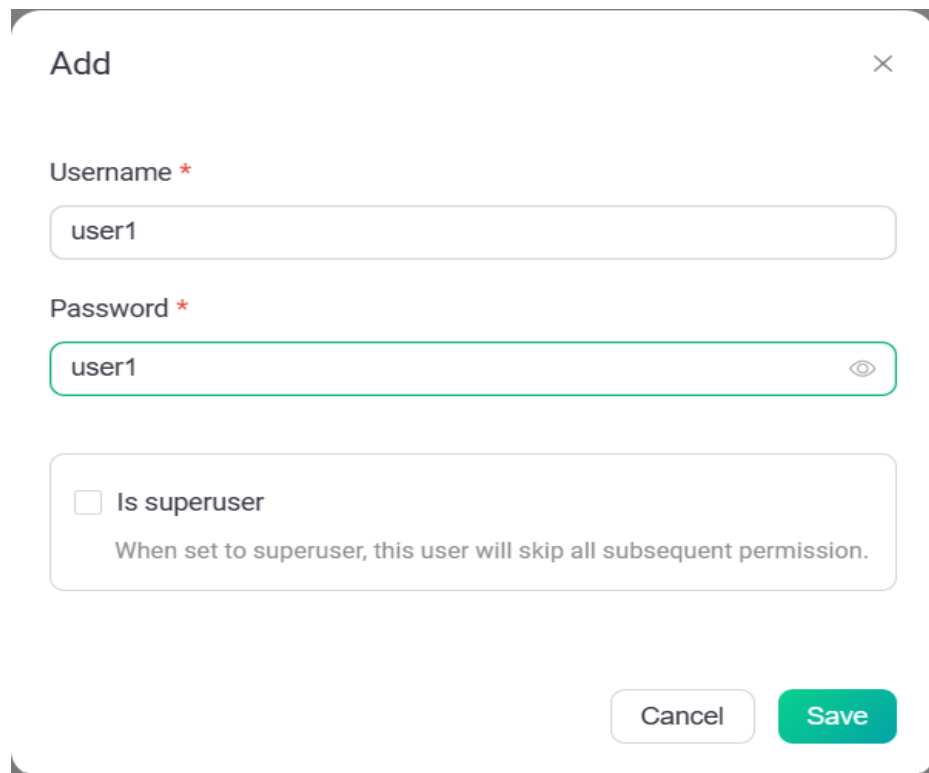
Gambar 12. Mengatur Backend pada Authentication



Gambar 13. 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” .



The image shows a modal window titled "Add" for creating a new user. It includes the following elements:

- Username \***: A text input field containing "user1".
- Password \***: A password input field containing "user1" with a toggle icon on the right.
- Is superuser**: A checkbox that is currently unchecked, with a descriptive text below it: "When set to superuser, this user will skip all subsequent permission."
- Buttons**: "Cancel" and "Save" buttons at the bottom right.

Gambar 14. Menambahkan User 1 pada Dashboard EMQX

Add X

Username \*

user2

Password \*

.....

☐ Is superuser

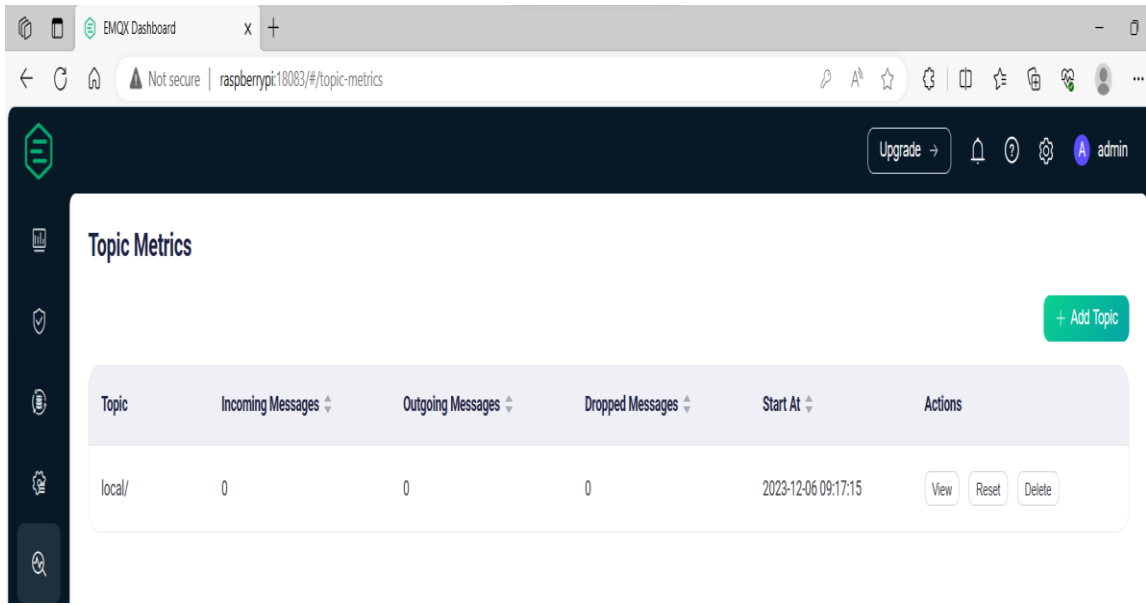
When set to superuser, this user will skip all subsequent permission.

Cancel Save

Gambar 15. 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.



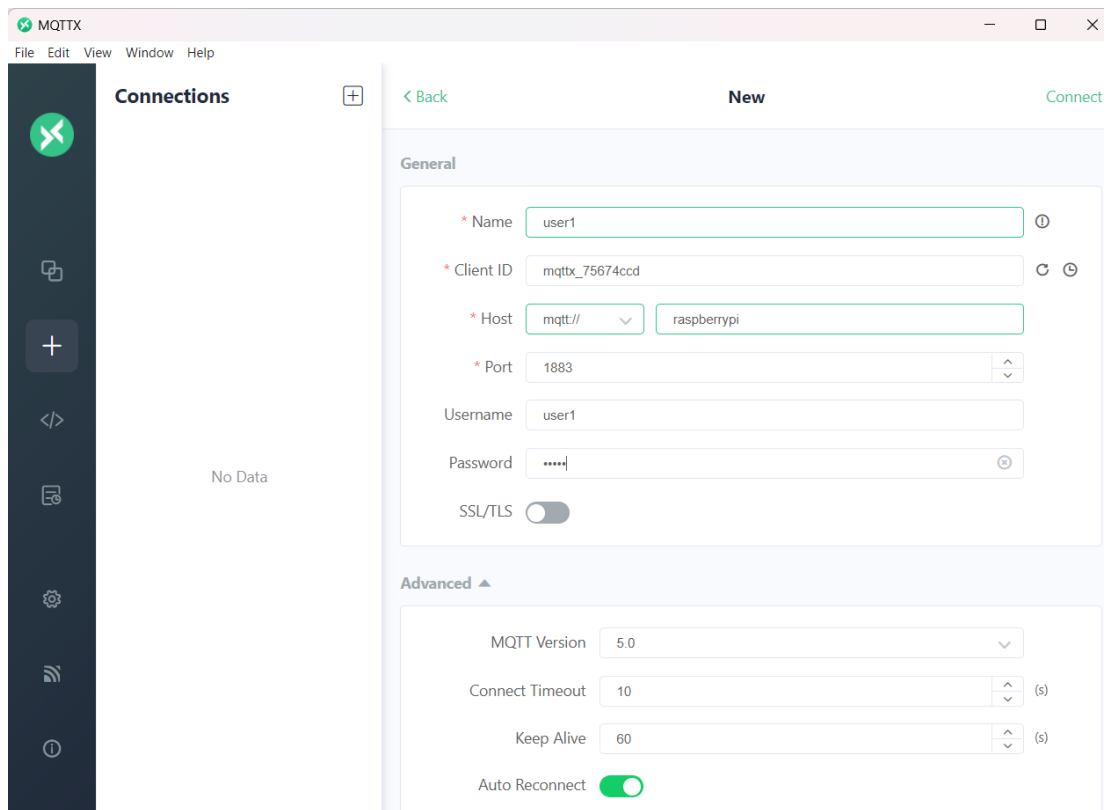
Gambar 16. Menambahkan Topic Metrics

#### ▪ **Setting MQTXX**

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 langgan. Berikut adalah langkah-langkah yang dilakukan dalam mengatur MQTXX

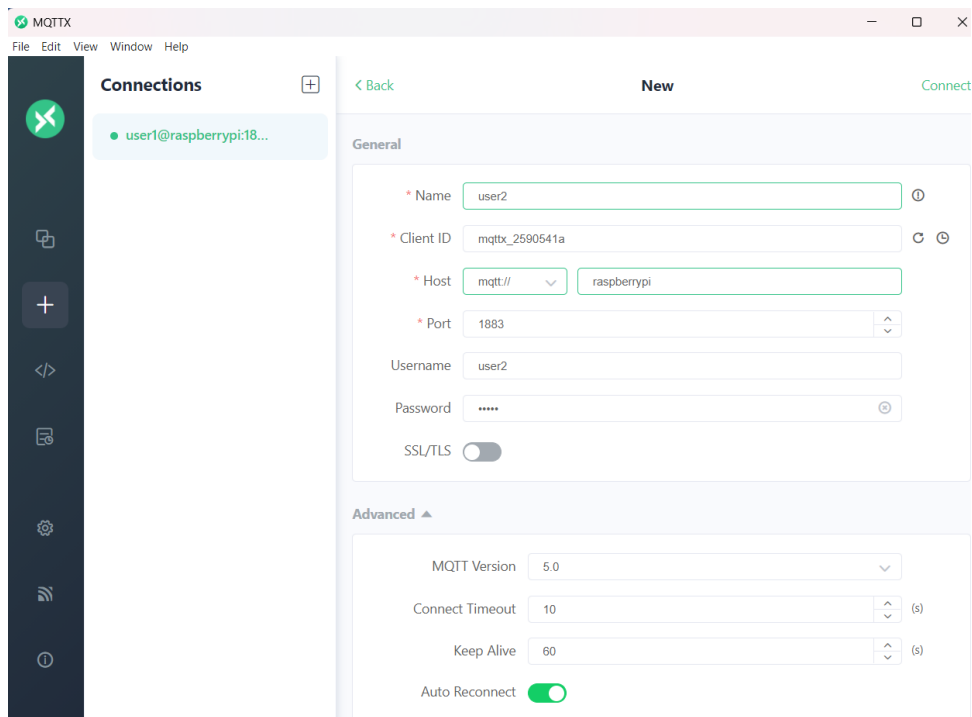
1. Buka MQTXX dan buat profil koneksi baru. Masukkan informasi tentang koneksi seperti alamat broker MQTT, port, dan protokol yang digunakan, biasanya TCP. Selain itu, masukkan informasi autentikasi seperti nama pengguna dan kata sandi jika diperlukan.



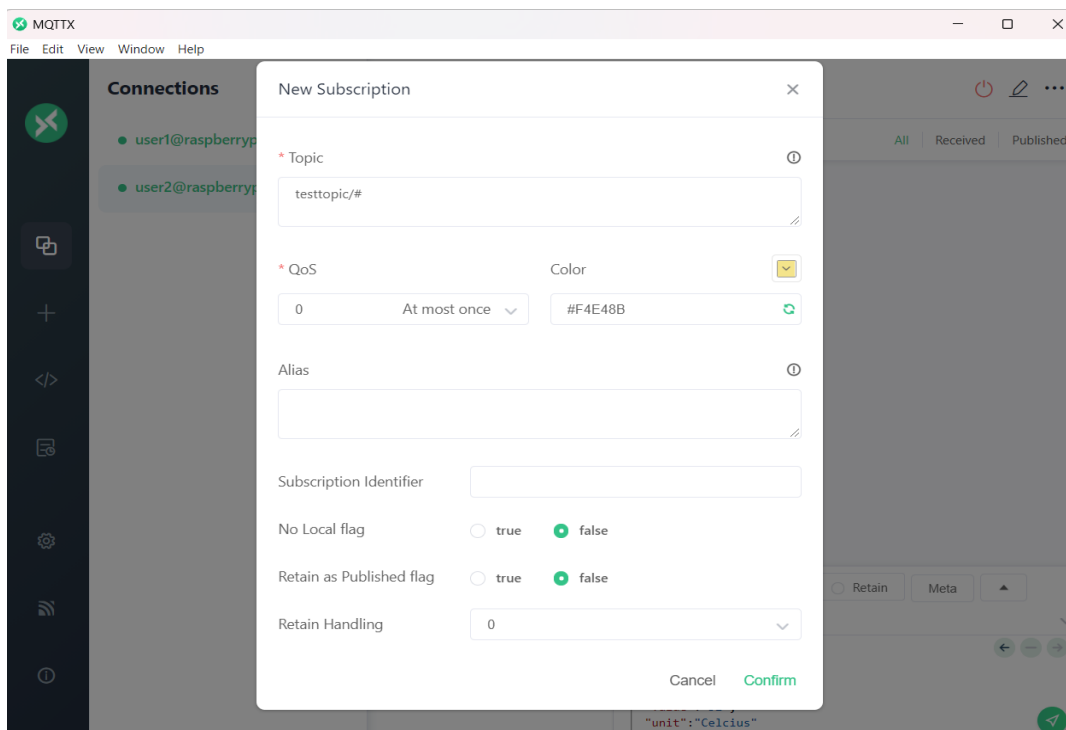


Gambar 17. Menambahkan User 1 pada MQTTX Dashboard

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



Gambar 18. Menambahkan User 2 pada MQTTX Dashboard



Gambar 19. Menambahkan Topic pada MQTTX Dashboard

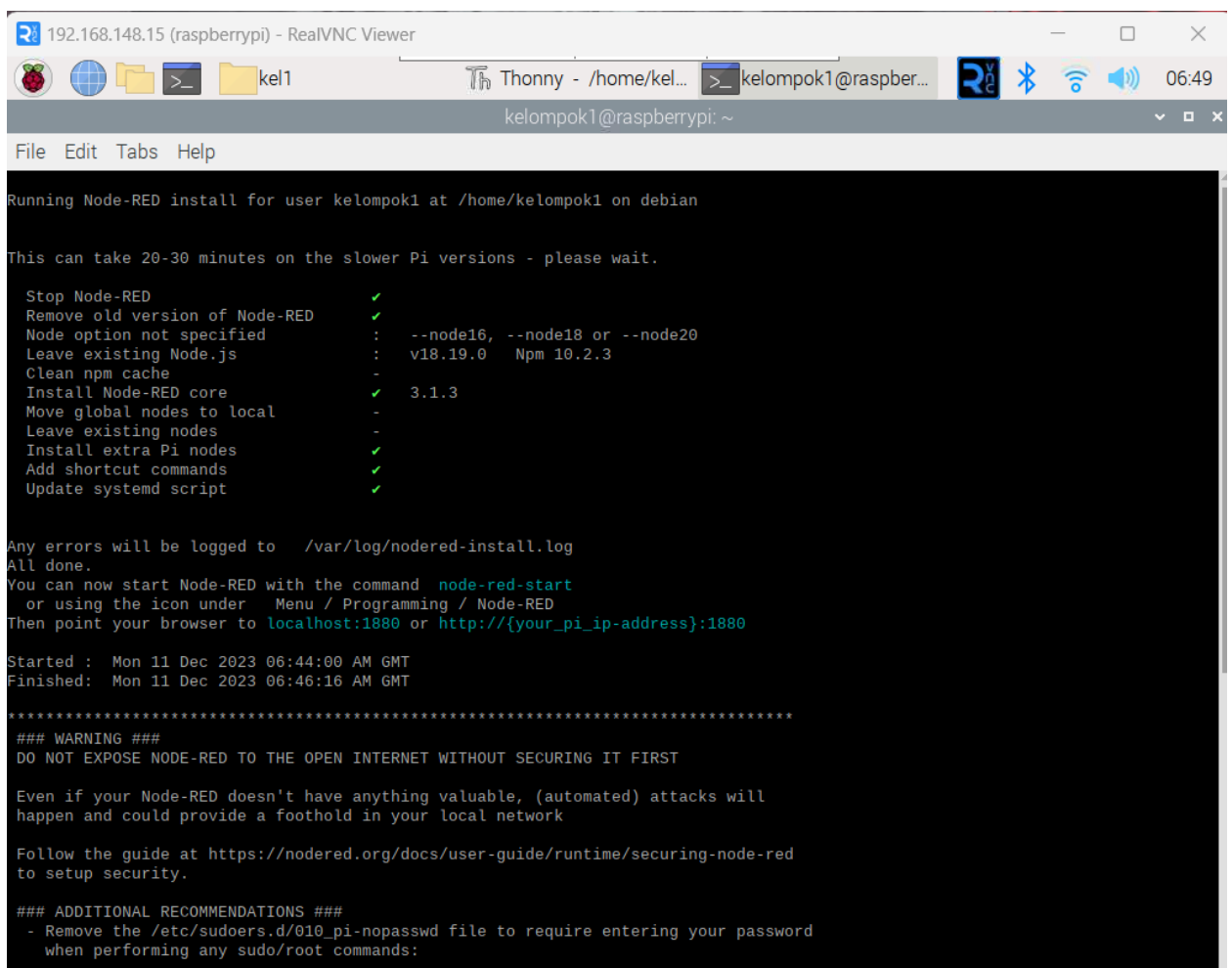
## 1. Setting node – red

### a. 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)
```

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



```
192.168.148.15 (raspberrypi) - RealVNC Viewer
Thonny - /home/kel...
kelompok1@raspberrypi: ~
File Edit Tabs Help

Running Node-RED install for user kelompok1 at /home/kelompok1 on debian

This can take 20-30 minutes on the slower Pi versions - please wait.

Stop Node-RED                                ✓
Remove old version of Node-RED                ✓
Node option not specified                    : --node16, --node18 or --node20
Leave existing Node.js                        : v18.19.0   Npm 10.2.3
Clean npm cache                              -
Install Node-RED core                        -   3.1.3
Move global nodes to local                   -
Leave existing nodes                          -
Install extra Pi nodes                       ✓
Add shortcut commands                       ✓
Update systemd script                        ✓

Any errors will be logged to /var/log/nodered-install.log
All done.
You can now start Node-RED with the command node-red-start
or using the icon under Menu / Programming / Node-RED
Then point your browser to localhost:1880 or http://{your_pi_ip-address}:1880

Started : Mon 11 Dec 2023 06:44:00 AM GMT
Finished: Mon 11 Dec 2023 06:46:16 AM GMT

#####
### WARNING ###
DO NOT EXPOSE NODE-RED TO THE OPEN INTERNET WITHOUT SECURING IT FIRST

Even if your Node-RED doesn't have anything valuable, (automated) attacks will
happen and could provide a foothold in your local network

Follow the guide at https://nodered.org/docs/user-guide/runtime/securing-node-red
to setup security.

### ADDITIONAL RECOMMENDATIONS ###
- Remove the /etc/sudoers.d/010_pi-nopasswd file to require entering your password
  when performing any sudo/root commands:
```

Gambar 20. Tampilan Hasil Instalasi Node Red

**b. Jalankan node red**

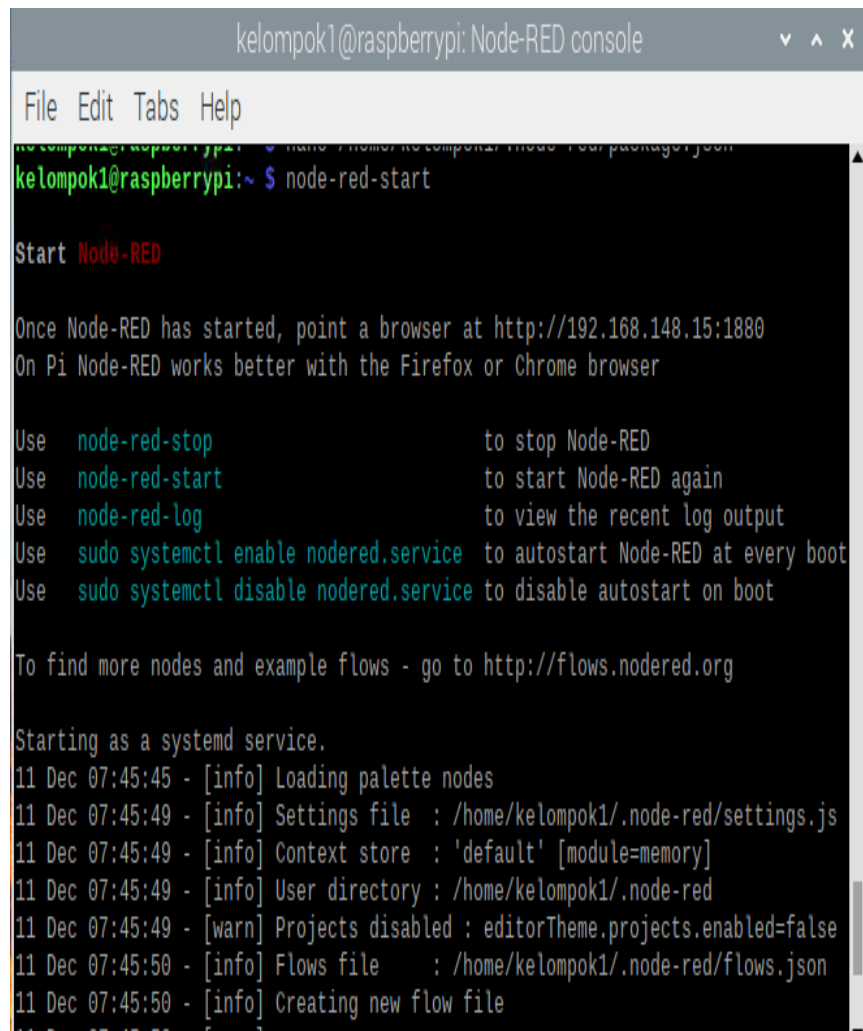
Setelah Node-RED diinstal, perintah berikut dapat digunakan untuk menjalankannya:

*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.



```
kelompok1@raspberrypi:~ $ node-red-start

Start Node-RED

Once Node-RED has started, point a browser at http://192.168.148.15:1880
On Pi Node-RED works better with the Firefox or Chrome browser

Use node-red-stop to stop Node-RED
Use node-red-start to start Node-RED again
Use node-red-log to view the recent log output
Use sudo systemctl enable nodered.service to autostart Node-RED at every boot
Use sudo systemctl disable nodered.service to disable autostart on boot

To find more nodes and example flows - go to http://flows.nodered.org

Starting as a systemd service.
11 Dec 07:45:45 - [info] Loading palette nodes
11 Dec 07:45:49 - [info] Settings file : /home/kelompok1/.node-red/settings.js
11 Dec 07:45:49 - [info] Context store : 'default' [module=memory]
11 Dec 07:45:49 - [info] User directory : /home/kelompok1/.node-red
11 Dec 07:45:49 - [warn] Projects disabled : editorTheme.projects.enabled=false
11 Dec 07:45:50 - [info] Flows file : /home/kelompok1/.node-red/flows.json
11 Dec 07:45:50 - [info] Creating new flow file
```

Gambar 21. Node Red Berjalan

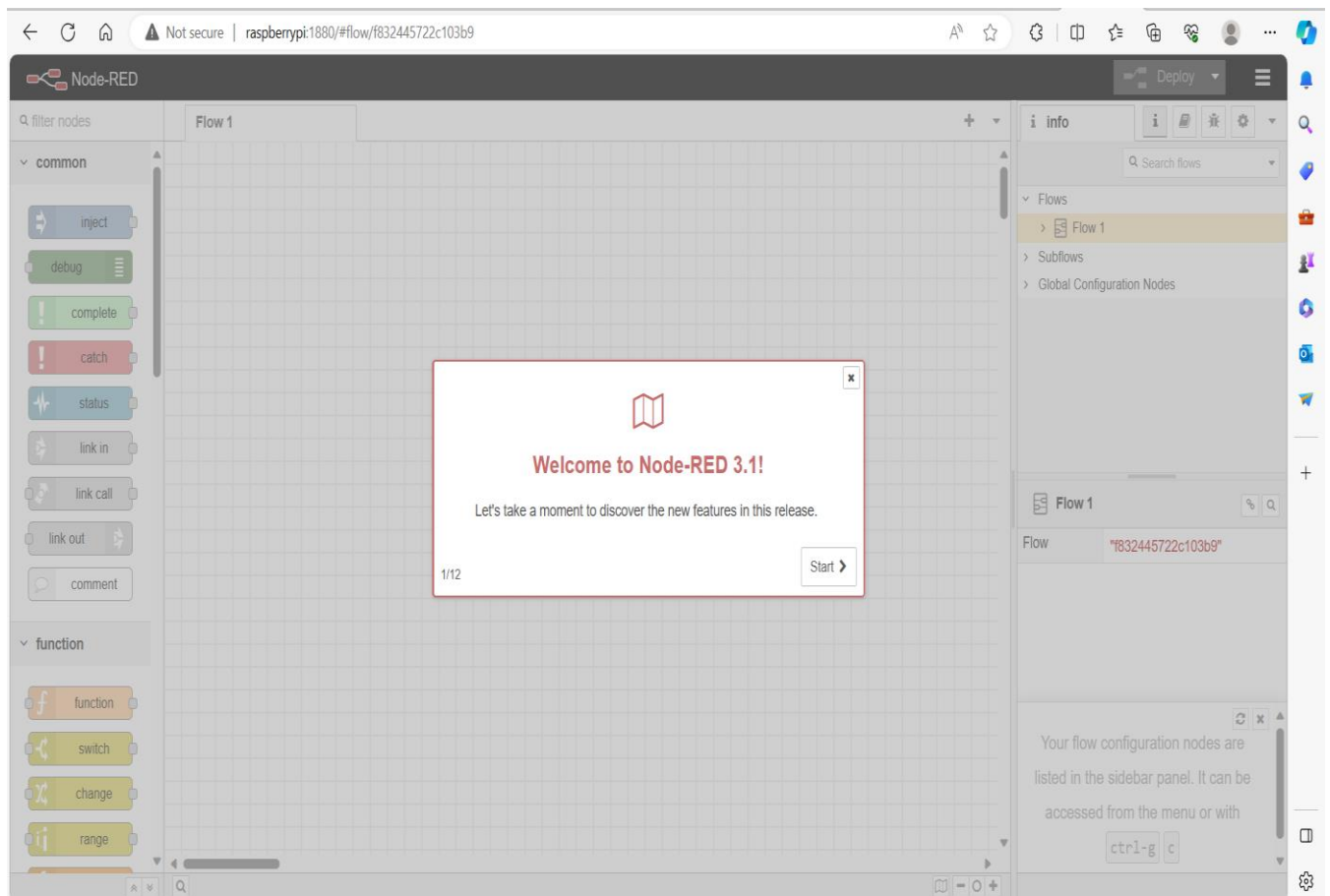
**c. Buka node red pada browser**

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 22. Tampilan Membuka Workspace Node Red

#### d. 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.

The screenshot shows the 'Add new mqtt-broker config node' dialog in Node-RED. The 'Properties' tab is active, showing the following configuration:

- Name:** EMQX-local
- Connection Tab:**
  - Server:** localhost
  - Port:** 1883
  - ☒ Connect automatically
  - ☐ Use TLS
- Protocol:** MQTT V5
- Client ID:** Leave blank for auto generated
- Keep Alive:** 60
- Session:**
  - ☒ Use clean start
  - Session Expiry (secs):
- User Properties:** none

At the bottom, there is a status bar showing '0 nodes use this config' and a dropdown menu set to 'On all flows'.

Gambar 23. 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 pasanginya 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.

The image shows the 'Edit mqtt in node' configuration window. At the top, there are three buttons: 'Delete', 'Cancel', and 'Done'. Below these is a 'Properties' section with a gear icon and three sub-panels: 'Server', 'Topic', and 'Flags'. The 'Server' panel has a dropdown menu set to 'EMQX-local' and a pencil icon. The 'Topic' panel has a text input field containing 'test1'. The 'Flags' panel has a 'QoS' dropdown set to '2' and two checkboxes: 'Do not receive messages published by this client' (unchecked) and 'Keep retain flag of original publish' (checked). Below the 'Flags' panel is a 'Retained message handling' section with a dropdown menu set to 'Send retained messages'. At the bottom, there is an 'Output' section with a dropdown menu set to 'auto-detect (parsed JSON object, string or bufl)' and a 'Name' section with a text input field containing 'Name'.

Gambar 24. Mengedit MQTT In

#### e. 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 25. Menambahkan Konfigurasi Node MQTT Out Broker

#### f. Influxdb

##### - Instalasi dan konfigurasi influxdb

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



*bash sudo apt install influxdb*

```
kelompok1@raspberrypi:~$ sudo apt install influxdb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  influxdb
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.
Need to get 4,272 kB of archives.
After this operation, 17.5 MB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bullseye/main arm64 influxdb arm64 1.6.7-rc0-1+b5 [4,272 kB]
Fetched 4,272 kB in 8s (564 kB/s)
Selecting previously unselected package influxdb.
(Reading database ... 108200 files and directories currently installed.)
Preparing to unpack .../influxdb_1.6.7-rc0-1+b5_arm64.deb ...
Unpacking influxdb (1.6.7-rc0-1+b5) ...
Setting up influxdb (1.6.7-rc0-1+b5) ...
Adding system user `influxdb' (UID 117) ...
Adding new user `influxdb' (UID 117) with group `nogroup' ...
Not creating home directory `/var/lib/influxdb'.
Adding group `influxdb' (GID 125) ...
Done.
Adding user `influxdb' to group `influxdb' ...
Adding user influxdb to group influxdb
Done.
Created symlink /etc/systemd/system/influxd.service → /lib/systemd/system/influxdb.service.
Created symlink /etc/systemd/system/multi-user.target.wants/influxdb.service → /lib/systemd/system/influxdb.service.
Processing triggers for man-db (2.9.4-2) ...
kelompok1@raspberrypi:~$ systemctl unmask influxdb
kelompok1@raspberrypi:~$ systemctl enable influxdb
Synchronizing state of influxdb.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable influxdb
kelompok1@raspberrypi:~$ systemctl start influxdb
kelompok1@raspberrypi:~$
```

Gambar 26. Menginstall InfluxDB

### - Instalasi Client InfluxDB

Kami dapat menginstall InfluxDB server dan juga menginstall 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.

```

kelompok1@raspberrypi:~ $ sudo apt install influxdb-client
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  influxdb-client
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.
Need to get 1,987 kB of archives.
After this operation, 7,108 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bullseye/main arm64 influxdb-client arm64 1.6.7-rc0-1+b5 [1,987 kB]
Fetched 1,987 kB in 5s (366 kB/s)
Selecting previously unselected package influxdb-client.
(Reading database ... 108215 files and directories currently installed.)
Preparing to unpack ../influxdb-client_1.6.7-rc0-1+b5_arm64.deb ...
Unpacking influxdb-client (1.6.7-rc0-1+b5) ...
Setting up influxdb-client (1.6.7-rc0-1+b5) ...
Processing triggers for man-db (2.9.4-2) ...
kelompok1@raspberrypi:~ $ influx
Connected to http://localhost:8086 version 1.6.7-rc0
InfluxDB shell version: 1.6.7-rc0
> exit
kelompok1@raspberrypi:~ $

```

Gambar 27. 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
```

```
GNU nano 5.4 /etc/influxdb/influxdb.conf *
###
### Controls how the HTTP endpoints are configured. These are the primary
### mechanism for getting data into and out of InfluxDB.
###

[http]
    auth-enabled = true
    pprof-enabled = true
    pprof-auth-enabled = true
    ping-auth-enabled = true

# Determines whether HTTP endpoint is enabled.
# enabled = true
```

Gambar 28. 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.

#### g. Grafana

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

- **Install Grafana**

Menginstal grafana biasanya dilakukan dengan cara berikut:

```

kelompok1@raspberrypi:~ $ wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
kelompok1@raspberrypi:~ $ echo "deb https://packages.grafana.com/oss/debian stable main" | sudo tee -a /etc/apt/sources.list.d
.grafana.list
deb https://packages.grafana.com/oss/debian stable main
kelompok1@raspberrypi:~ $ sudo apt update
Hit:1 http://security.debian.org/debian-security bullseye-security InRelease
Hit:2 http://deb.debian.org/debian bullseye InRelease
Hit:3 http://deb.debian.org/debian bullseye-updates InRelease
Hit:4 https://deb.nodesource.com/node_18.x nodistro InRelease
Hit:5 http://archive.raspberrypi.org/debian bullseye InRelease
Hit:6 https://packages.emqx.com/emqx/debian bullseye InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
3 packages can be upgraded. Run 'apt list --upgradable' to see them.
kelompok1@raspberrypi:~ $ sudo apt-get install grafana -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package grafana
kelompok1@raspberrypi:~ $

```

Gambar 29. Menginstalasi Grafana

**- Kemudian edit mqtt broker nya**

Dilakukan dengan cara berikut :

Edit mqtt in node > **Edit mqtt-broker node**

Delete Cancel Update

**Properties**

Name motion\_detection

**Connection** Security Messages

Server localhost Port 1883

☒ Connect automatically

☐ Use TLS

Protocol MQTT V3.1.1

Client ID Leave blank for auto generated

Keep Alive 60

Session ☒ Use clean session

Gambar 30. Mengatur Connection pada MQTT Broker

**Edit mqtt in node**

Delete Cancel Done

**Properties**

Server motion\_detection

Action Subscribe to single topic

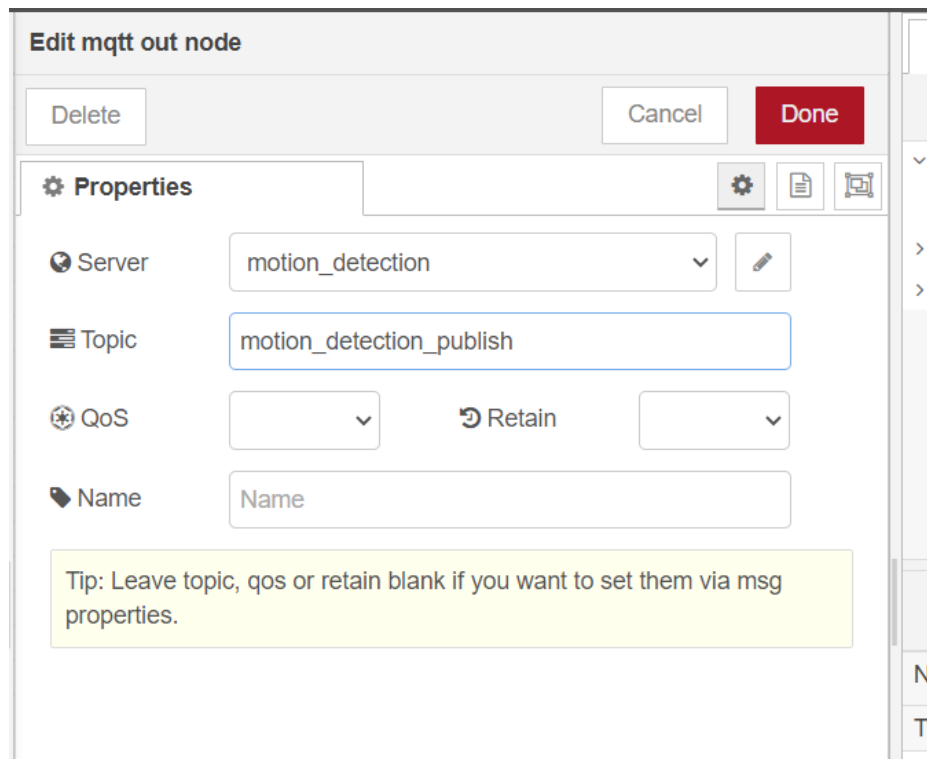
Topic motion\_detection\_publish

QoS 2

Output auto-detect (parsed JSON object, string or buf)

Name Name

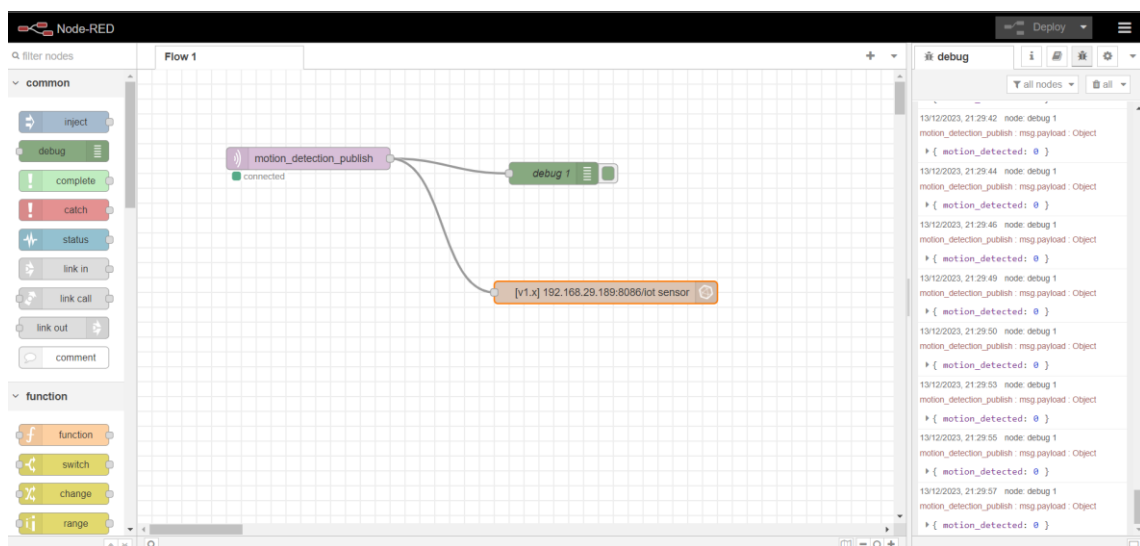
Gambar 31. Mengatur Security pada MQTT Broker



Gambar 32. Mengatur Messages pada MQTT Broker

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

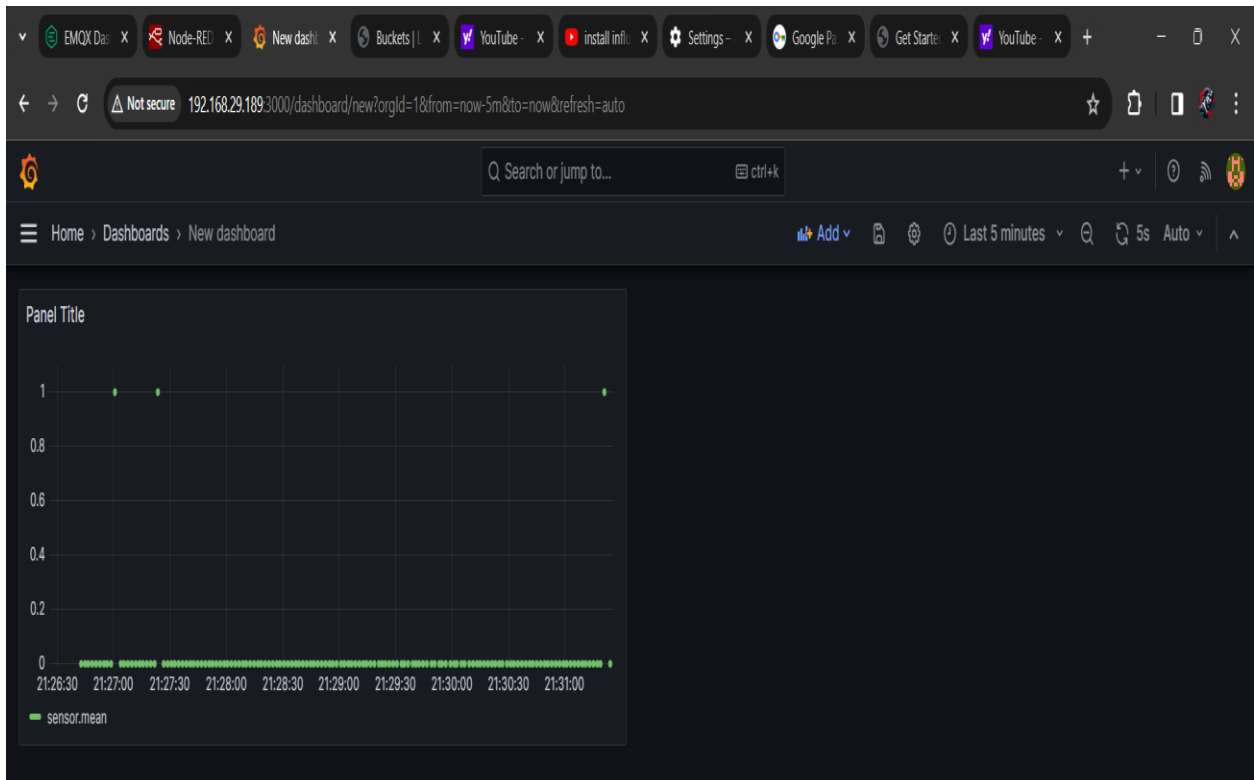
Area kerja ini disebut dengan workspace yang berbentuk grid atau canvas kosong yang diisi dengan node dan dihubungkan dengan kabel. Berikut adalah gambaran aliran data yang dibuat di Node-red :



Gambar 33. 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 34. 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

```

> select * from "sensor"
name: sensor
time          motion_detected
----          -
1702477383462826700 0
1702477385501394800 0
1702477387568354600 0
1702477389484289900 0
1702477391506117300 0
1702477393519421600 0
1702477395516086500 0
1702477397531955100 0
1702477399524506200 0
1702477401536045600 0
1702477403496958300 0
1702477405497844700 0
1702477407546331300 0
1702477409511772500 0
1702477411636226600 0
>

```

Gambar 35. 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 tersebut. Kueri ini juga memperbarui nilai non-numerik di kolom motion-detected menjadidi 0 agar valid .

- **Menjalankan skrip Python**

Digunakan untuk mendeteksi gerakan.



```
kelompok1@raspberrypi:~ $ nano kelompok_1_16.py
kelompok1@raspberrypi:~ $ python3 motion_detection.py
Connected to MQTT Broker!
Motion Status: 0
Motion Status: 0, sent to topic motion_detection_publish
Motion Status: 0
Motion Status: 0, sent to topic motion_detection_publish
Motion Status: 0
Motion Status: 0, sent to topic motion_detection_publish
Motion Status: 0
Motion Status: 0, sent to topic motion_detection_publish
Motion Status: 0
Motion Status: 0, sent to topic motion_detection_publish
Motion Status: 0
Motion Status: 0, sent to topic motion_detection_publish
Motion Status: 0
Motion Status: 0, sent to topic motion_detection_publish
Motion Status: 0
Motion Status: 0, sent to topic motion_detection_publish
Motion Status: 0
Motion Status: 0, sent to topic motion_detection_publish
Motion Status: 1
Motion Status: 1, sent to topic motion_detection_publish
Motion Status: 0
```

Gambar 36. Menjalankan Script Python Kelompok 1

Berikut adalah penjelasan 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

```
C:\Windows\System32\cmd.e  X  +  v
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. All rights reserved.

E:\data influxDB\influxdb-1.7.11-1>influx
```

Gambar 37. Tampilan CMD saat InfluxDB dibuka

```
C:\Windows\System32\cmd.e x + v
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. All rights reserved.

E:\data influxDB\influxdb-1.7.11-1>influxd
'influxd' is not recognized as an internal or external command,
operable program or batch file.

E:\data influxDB\influxdb-1.7.11-1>influxd

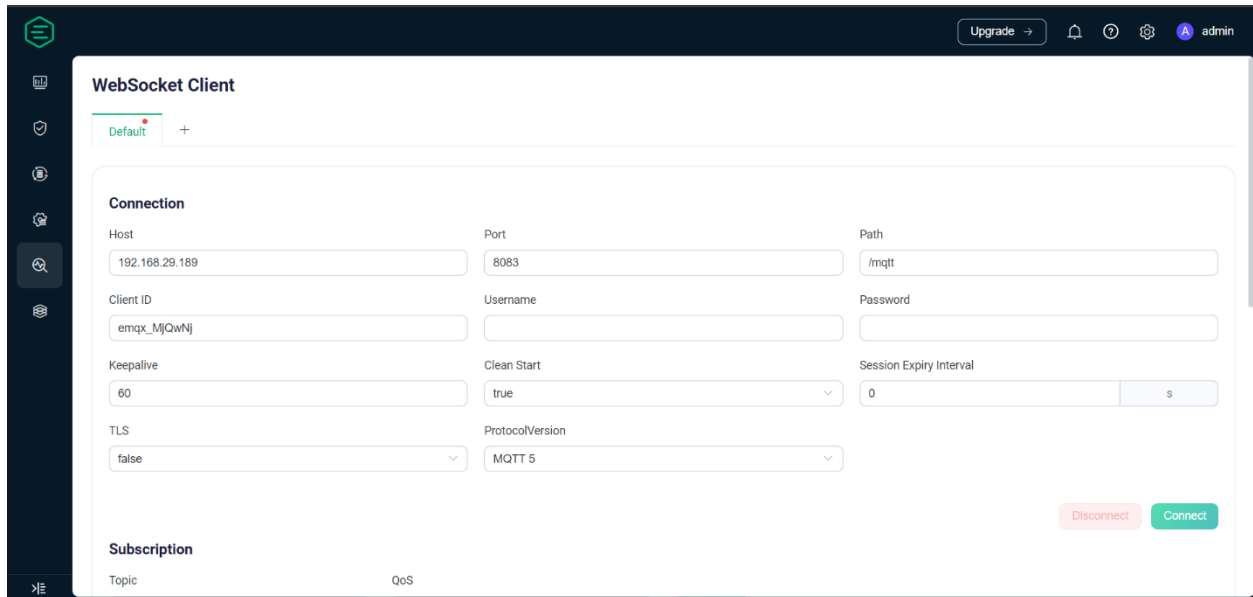
88888888      .d888 888      88888888b. 8888888b.
888      d88P" 888      888  "Y88b 888  "88b
888      888 888 888      888 888 888 88P
888 888888b. 8888888 888 888 888 888 888 88888888K.
888 888 "88b 888 888 888 888 Y8bd8P' 888 888 888 "Y88b
888 888 888 888 888 888 888 X88K 888 888 888 888
888 888 888 888 888 Y88b 888 .d8""8b. 888 .d88P 888 d88P
88888888 888 888 888 888 "Y88888 888 888 8888888P" 88888888P"

2023-12-13T14:04:08.731441Z info InfluxDB starting {"log_id": "0m4siQbW000", "version": "1.7.11", "branch":
"1.7.11", "commit": "f11ad4780c8a61108108a18b141c1d067d920a80"}
2023-12-13T14:04:08.731441Z info Go runtime {"log_id": "0m4siQbW000", "version": "go1.13.8", "maxprocs": 8}
2023-12-13T14:04:08.847041Z info Using data dir {"log_id": "0m4siQbW000", "service": "store", "path": "C:\\Users
\\ASUS\\.influxdb\\data"}
2023-12-13T14:04:08.847041Z info Compaction settings {"log_id": "0m4siQbW000", "service": "store", "max_concu
rrent_compactions": 4, "throughput_bytes_per_second": 50331648, "throughput_bytes_per_second_burst": 50331648}
2023-12-13T14:04:08.847041Z info Open store (start) {"log_id": "0m4siQbW000", "service": "store", "trace_id"
: "0m4siR3l000", "op_name": "tsdb_open", "op_event": "start"}
2023-12-13T14:04:08.896813Z info Reading file {"log_id": "0m4siQbW000", "engine": "tsml", "service": "cachelo
ader", "path": "C:\\Users\\ASUS\\.influxdb\\wal\\_internal\\monitor\\4\\_00001.wal", "size": 296753}
2023-12-13T14:04:08.901801Z info Opened file {"log_id": "0m4siQbW000", "engine": "tsml", "service": "filestor
```

Gambar 38. Tampilan CMD saat InfluxDB dijalankan

## - Membuat Koneksi WebSocket

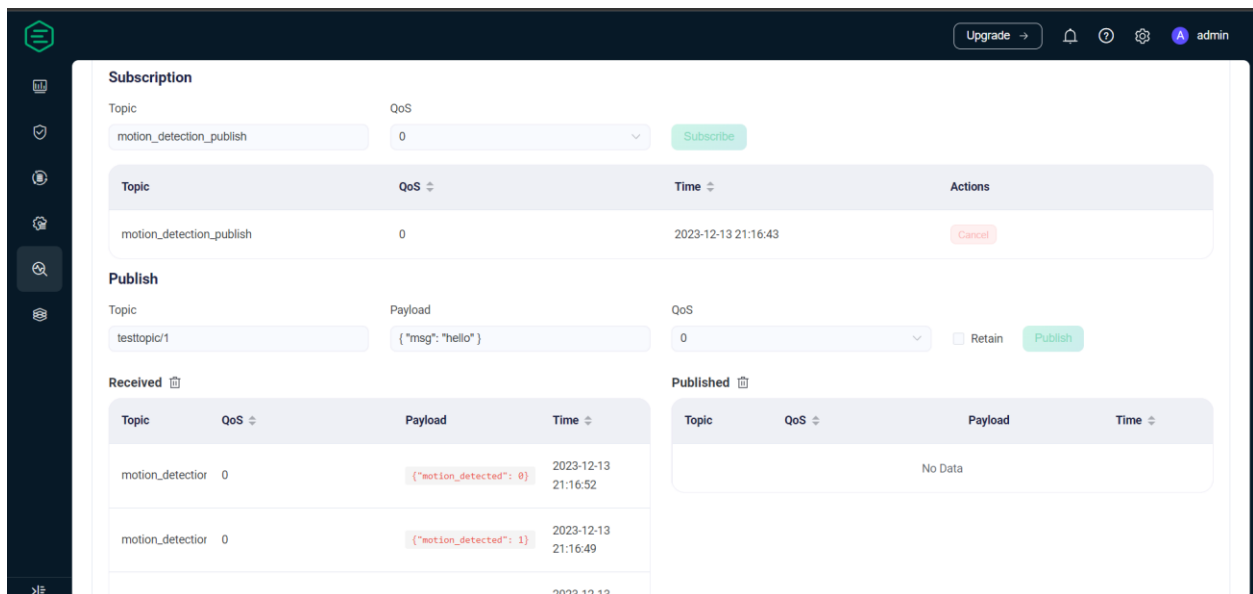
Setelah lingkungan pengembangan selesai, langkah berikutnya adalah menghubungkan WebSocket ke server. Untuk melakukan ini, kita perlu menggunakan API WebSocket yang disediakan oleh bahasa pemrograman yang kita gunakan. Berikut adalah tampilan dari WebSocket Client pada proyek ini :



Gambar 39. 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 40. Mengedit Subscription Pengiriman dan Penerimaan Pesan