

“Smart GreenHouse”



**FAKULTAS VOKASI
INSTITUT TEKNOLOGI DEL
2022**

Daftar Pembuat Proyek

Kelompok 4:		Kelompok 5:	
Fedrick Samuel Pasaribu	13320001	Putra Toba Tampubolon	13320015
Nurcahaya Kerentryna S	13320017	Jesika Laprina Manurung	13320022
Kelompok 6:		Kelompok 7:	
Febri Sinaga	13320021	Glorian Johan Einstein Purba	13320002
Mersi Suryani Siagian	13320013	Natasha Gabriela Sinaga	13320014
Kelompok 16:		Kelompok 21:	
Rio Putrawan Zalukhu	13320038	Felix Simanjuntak	13320031
Lidia Kesvina Pasaribu	13320051	Putri Kezia Nababan	13320044
Kelompok 22:		Kelompok 23:	
Riki Yoga Situmorang	13320030	Nicholas Canakya Pardosi	13320043
Harli Juita Sinabutar	13320054	Yemima Sri Rezeki Damanik	13320049

Daftar Isi

A.	Pendahuluan.....	3
B.	Hasil dan Implementasi.....	4
•	Kelompok 4 :	4
•	Kelompok 5 :	7
•	Kelompok 6 :	10
•	Kelompok 7 :	12
•	Kelompok 16 :	14
•	Kelompok 21 :	16
•	Kelompok 22 :	18
•	Kelompok 23 :	21
C.	Penutup	23
D.	Lampiran	24

A. Pendahuluan

Pada kelompok besar, judul yang diambil adalah “Smart Green House”, dan untuk setiap kelompok kecilnya memiliki beberapa part yang harus dikerjakan oleh masing-masing kelompok, yaitu :

Kelompok 4 : Deteksi hama menggunakan sensor PIR

Kelompok 5 : Deteksi Intensitas cahaya menggunakan sensor LDR

Kelompok 6 : Deteksi Intensitas cahaya menggunakan sensor LDR

Kelompok 7 : Kontrol Ketinggian air menggunakan sensor Ultrasonik

Kelompok 16 : Deteksi hama menggunakan sensor PIR

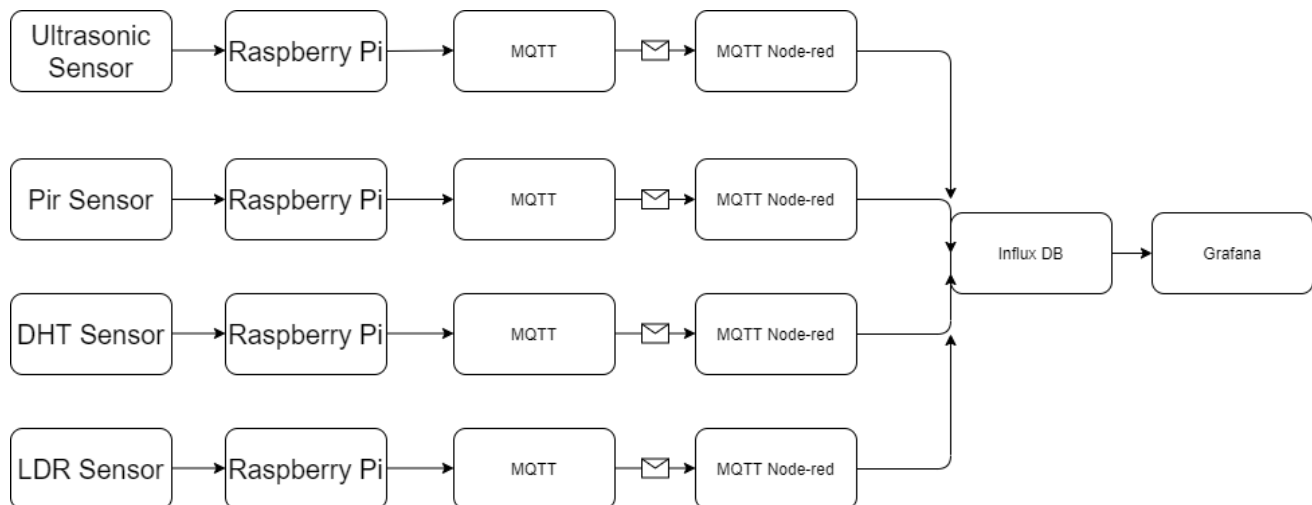
Kelompok 21 : Kontrol suhu menggunakan sensor DHT 11

Kelompok 22 : Kontrol suhu menggunakan sensor DHT 11

Kelompok 23 : Kontrol Ketinggian air menggunakan sensor Ultrasonik

B. Hasil dan Implementasi

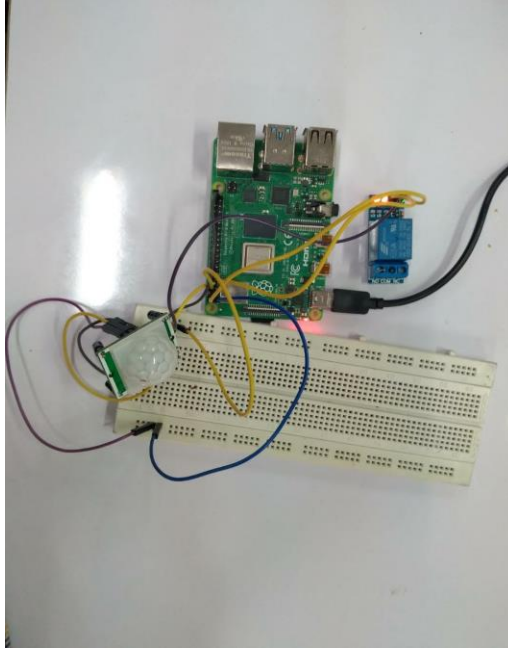
Pada bab ini akan melampirkan terkait hasil dan implementasi yang sudah dibuat oleh masing-masing kelompok sehingga membangun satu sistem yang besar yaitu “Smart Green House”



- Kelompok 4 :

1. Infrastruktur sistem

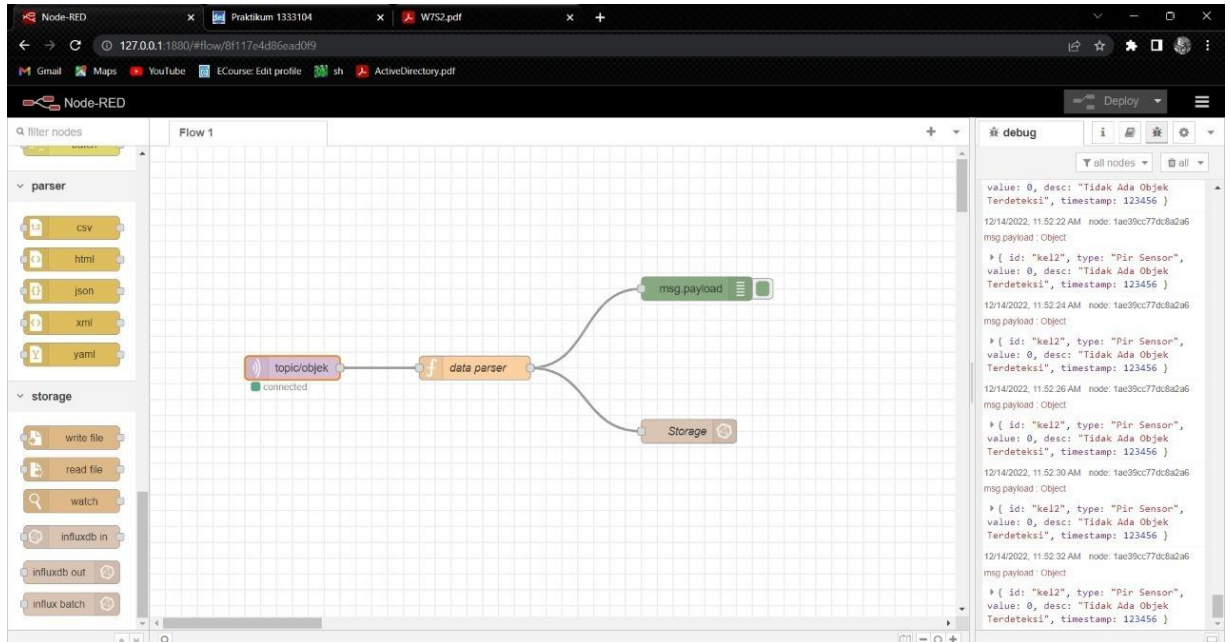
Pada proyek ini kelompok kami membuat deteksi hama menggunakan relay dan pir. Dimana jika ada objek yang terdeteksi maka nilai nya adalah 1 dan jika tida ada objek yang terdeteksi maka nilai 0.



Gambar 1 Rangkaian Menggunakan Sensor PIR

Pada gambar 1 dapat kita lihat rangkaian tersebut menggunakan mikrokontroler yaitu Raspberry dan juga sensor pir sebagai mendeteksi hama dan relay sebagai aktuator yaitu relay. Sebelumnya kita membuat kode program untuk menjalankan rangkaian diatas, saat kode program tersebut dijalankan akan mendapatkan data. Aktuator yaitu relay bertindak sebagai pendeteksi ada atau tidaknya hama sedangkan sensor pir berfungsi sebagai penyemprot hama.

Data sensor akan dikirimkan menggunakan raspberry pi ke MQTT node-red kemudian untuk menyimpan data yang telah didapat menggunakan database Influxdb.



Gambar 2 Rangkaian Node Red

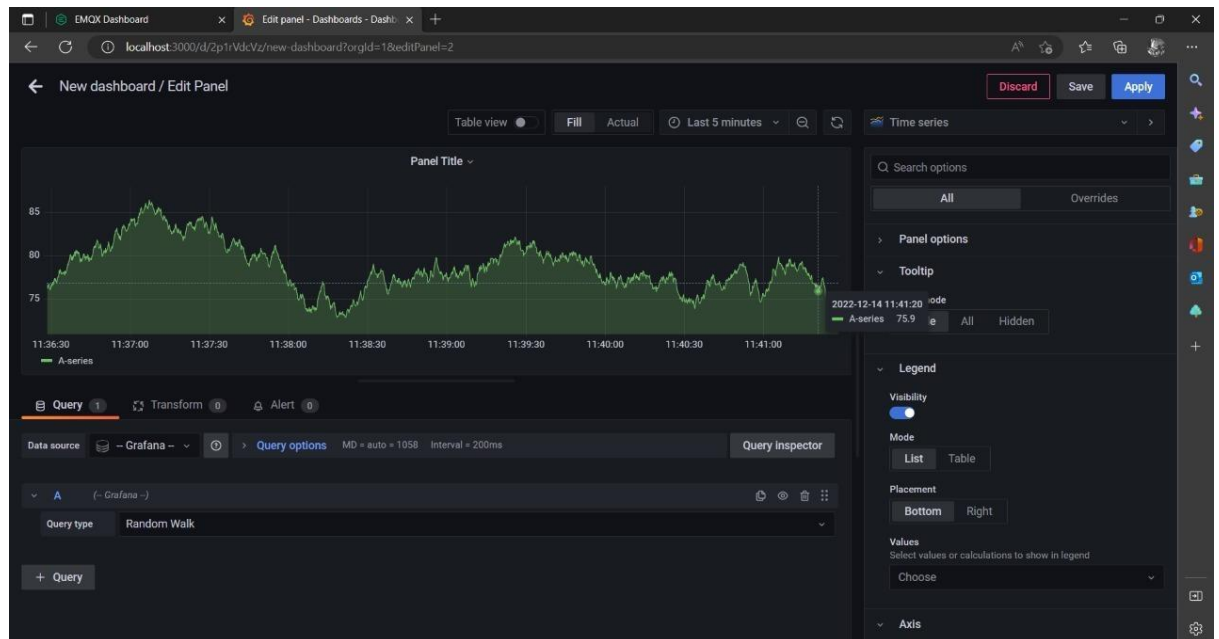
Pada gambar dibawah ini, kita bisa melihat data yang telah kita simpan pada database influxdb.

C:\Windows\System32\cmd.exe - influx

1670994072812961100	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994073244412300	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994074959811900	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994081315925000	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994083564133900	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994085409038900	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994086846040300	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994087457125600	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994088888165800	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994089349594000	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994090410235000	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994095289403100	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994097493851000	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994098930440400	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994099436207600	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994100974725800	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994103020807200	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994104864107100	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994111417905000	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994113362674800	Ada Objek Terdeteksi	kel2	123456	Pir Sensor	1
1670994114911785700	Ada Objek Terdeteksi	kel2	123456	Pir Sensor	1
1670994115411315000	Ada Objek Terdeteksi	kel2	123456	Pir Sensor	1
1670994116946818000	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994117303165100	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994119304918300	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994121442625500	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994123501930800	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994124937585700	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994125344724300	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994126984095600	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994127391779100	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994129030473300	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994130075767500	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994135379121200	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994137528777400	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994139373501800	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994140908233600	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994141420516900	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994142894462300	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994143366223600	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994144923194000	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994145415527900	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994147049679300	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994147243819500	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994148870934200	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0
1670994150946549000	Ada Objek Terdeteksi	kel2	123456	Pir Sensor	1
1670994152873782900	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor	0

Gambar 3 Data masuk pada Influxdb

Untuk menampilkan datanya, kita dapat menggunakan Grafana untuk memvisualisasikan data.

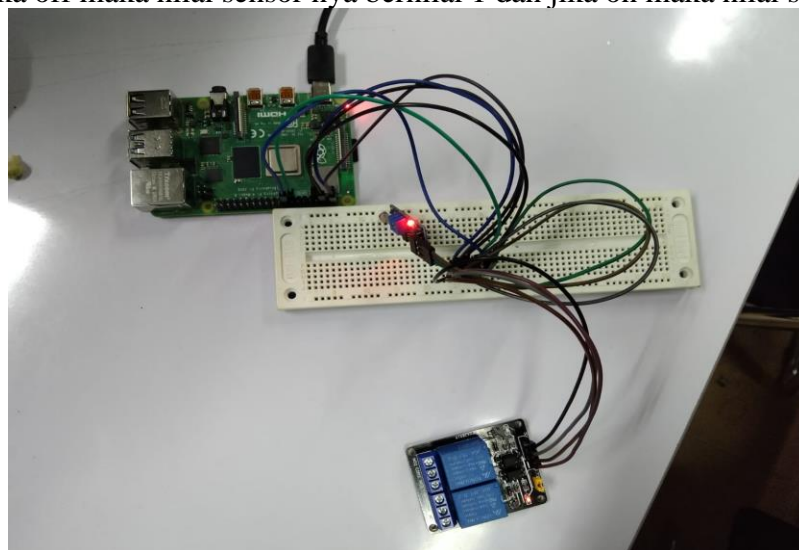


Gambar 4 Dashboard grafana sebagai visualisasi data

- Kelompok 5 :

1. Infrastruktur sistem

Pada proyek ini kelompok kami membuat deteksi intensitas cahaya menggunakan sensor LDR. dan relay. Dimana nilai LDR berubah seiring intensitas cahaya yang diterima. Jika off maka nilai sensor nya bernilai 1 dan jika on maka nilai sensor nya 0.



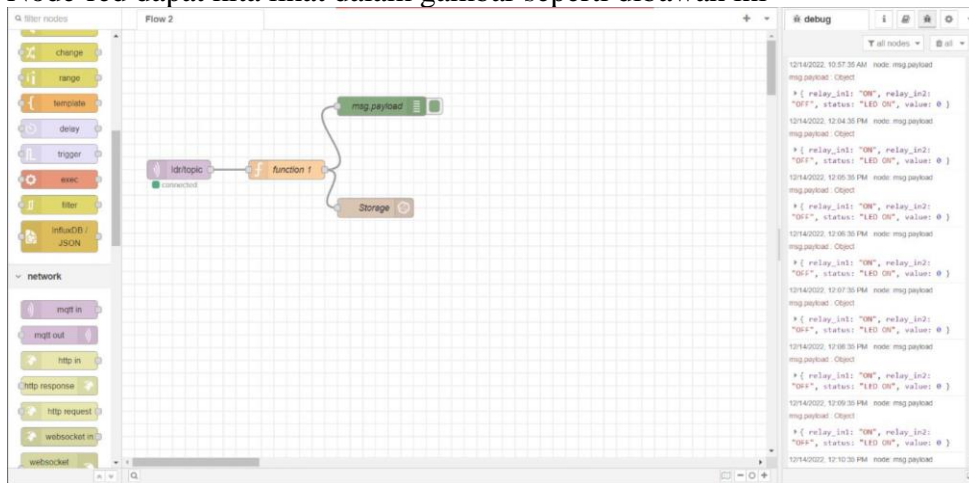
di rangkaian dapat kita tahu bahwa mikrokontroler yang kami gunakan yaitu Raspberry dan juga sensor ldr sebagai pendeteksi cahaya dan juga relay sebagai aktuator. Dimana

kita akan buat code untuk menajlankan si sensor di dalam Raspberry agar si sensor bisa mendapatkan datannya,dan akuator bisa bertindak apakah akan menyalakan lampu atau tidak(disini lampu kami buat sebagai penghangat ruangan saat tidak adanya cahaya matahari menyinari lahan kami.Disini saat kondisi dari si ldr menerima cahaya maka akan bernilai satu dan relay akan memadamkan lampu,namun sebaliknya jika ldr bernilai 0 maka relay akan menyalakan lampu.

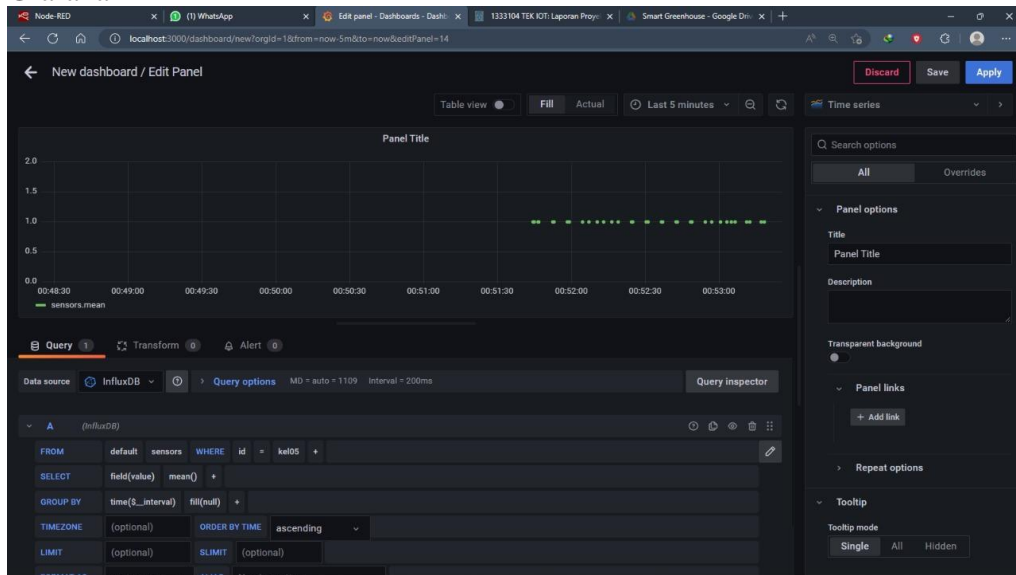
Data base yang kami bentuk di dalam influxdb,dan dapat kita lihat bahwa data dapat disampaikan dengan baik ke dalam data base yang telah kita buat.

```
> select * from ldr
name: ldr
time                relay_in1 relay_in1_1 relay_in2 relay_in2_1 status  status_1 value
-----
1670980944804529600 OFF      ON      LED_OFF 1
1670983590799404900 ON      OFF      LED_ON 0
1670983592937547300 ON      OFF      LED_ON 0
1670983594787427300 ON      OFF      LED_ON 0
1670983596866038100 OFF      ON      LED_OFF 1
1670983598913034700 OFF      ON      LED_OFF 1
1670983601065550900 OFF      ON      LED_OFF 1
1670983602910109700 OFF      ON      LED_OFF 1
1670983604856077300 OFF      ON      LED_OFF 1
1670983606912599600 OFF      ON      LED_OFF 1
1670983608949799100 ON      OFF      LED_ON 0
1670983610898483100 ON      OFF      LED_ON 0
1670983612843053200 OFF      ON      LED_OFF 1
1670983614888984200 ON      OFF      LED_ON 0
1670983616834766000 OFF      ON      LED_OFF 1
1670983618986952200 ON      OFF      LED_ON 0
1670983620933016600 ON      OFF      LED_ON 0
1670983622878348100 ON      OFF      LED_ON 0
1670983624929005700 ON      OFF      LED_ON 0
1670983626970795900 OFF      ON      LED_OFF 1
1670983629021644000 OFF      ON      LED_OFF 1
1670984284603233100 ON      OFF      LED_ON 0
1670984286627872600 ON      OFF      LED_ON 0
1670984288602838900 OFF      ON      LED_OFF 1
1670984290621918800 OFF      ON      LED_OFF 1
1670984292669019600 OFF      ON      LED_OFF 1
1670984294817929700 OFF      ON      LED_OFF 1
1670984296659150100 OFF      ON      LED_OFF 1
1670984298708995500 OFF      ON      LED_OFF 1
1670984300655877400 OFF      ON      LED_OFF 1
1670984302805175100 ON      OFF      LED_ON 0
1670984305074603500 OFF      ON      LED_OFF 1
1670984306902581200 ON      OFF      LED_ON 0
1670984308645995000 OFF      ON      LED_OFF 1
1670984310690539200 OFF      ON      LED_OFF 1
1670984312948009700 OFF      ON      LED_OFF 1
```


Node-red dapat kita lihat dalam gambar seperti dibawah ini



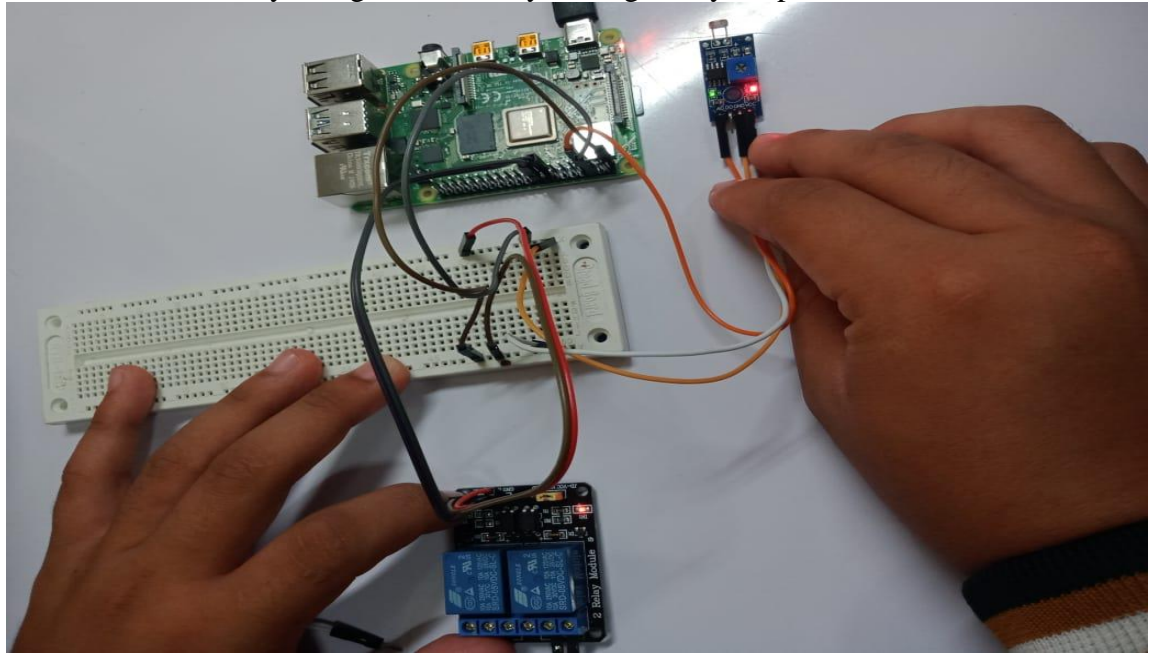
Grafana



- Kelompok 6 :

1. Infrastruktur sistem

pada proyek greenhouse ini kelompok kami membuat sistem dengan menggunakan sensor LDR dan relay sebagai aktuator nya. Rangkaiannya dapat kita lihat di bawah ini:



Pada rangkaian di atas dapat kita lihat ada raspberry pi sebagai mikrokontroler pada sistem yang kami buat. Dimana sensor LDR sebagai pendeteksi intensitas cahaya dan relay sebagai penggerak untuk membuka dan menutup tirai jika cahaya terdeteksi oleh sensor. Apabila nilai sensor bernilai 1 maka relay akan hidup dan jika nilai nya 0 maka relay akan mati. setelah itu raspberry pi akan mempublish hasil dari sensor yang terdeteksi ke broker MQTT yaitu MQTXX. MQTXX akan melanjutkan hasil dari sensor ke Influxdb melalui Node-Red sebagai penghubung dimana influxdb sebagai database untuk menyimpan hasil sensor.

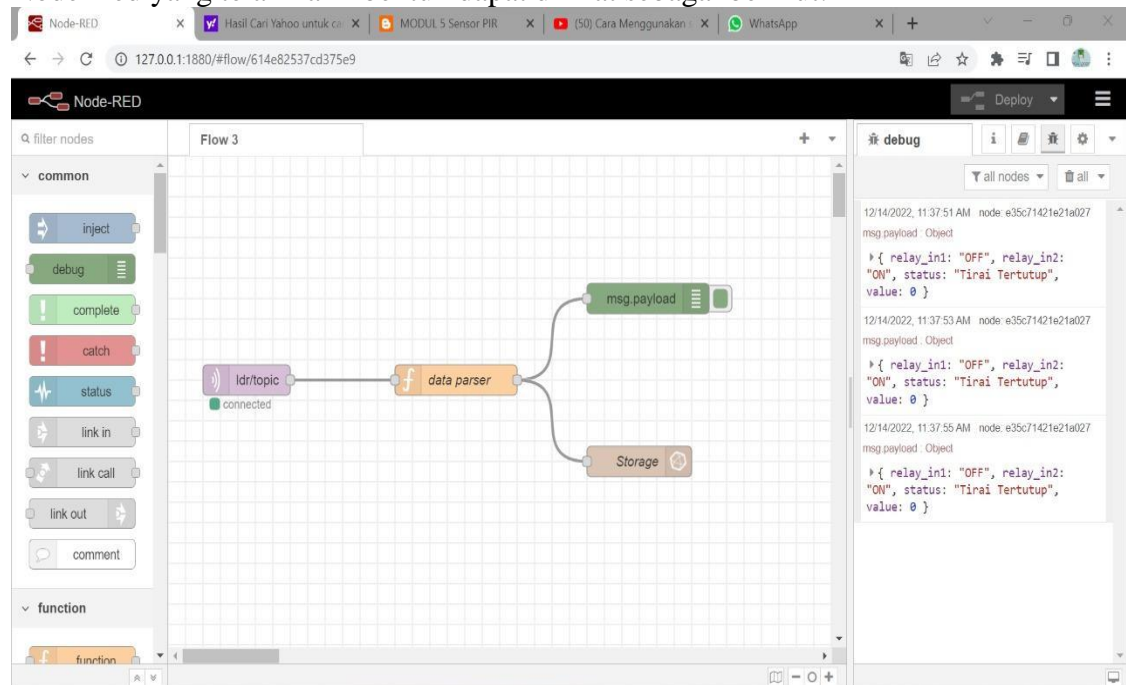
Database yang terbentuk di dalam InfluxDB dapat kita lihat pada gambar berikut:

```
C:\Windows\System32\cmd.exe - influx
Microsoft Windows [Version 10.0.22000.1219]
(c) Microsoft Corporation. All rights reserved.

E:\febri sinaga\perkuliahan\Semester 1\influxdb-1.7.11-1>influx
Connected to http://localhost:8086 version 1.7.11
InfluxDB shell version: 1.7.11
> use ldr
Using database ldr
> select * from ldr
name: ldr
time relay_in1 relay_in2 status value
-----
1670992743090979800 OFF ON Tirai Tertutup 0
1670992745096764600 OFF ON Tirai Tertutup 0
1670992747073421500 OFF ON Tirai Tertutup 0
1670992749080351800 OFF ON Tirai Tertutup 0
1670992751079370900 OFF ON Tirai Tertutup 0
1670992753078712100 OFF ON Tirai Tertutup 0
1670992755098400700 OFF ON Tirai Tertutup 0
1670992757113576300 OFF ON Tirai Tertutup 0
1670992759084277100 OFF ON Tirai Tertutup 0
1670992761086786400 OFF ON Tirai Tertutup 0
1670992763095490400 OFF ON Tirai Tertutup 0
1670992765124782300 OFF ON Tirai Tertutup 0
1670992767103905500 OFF ON Tirai Tertutup 0
1670992769264558900 OFF ON Tirai Tertutup 0
1670992771125111300 OFF ON Tirai Tertutup 0
1670992773109890900 OFF ON Tirai Tertutup 0
>
```

Dapat kita lihat dari database diatas bahwa data sensor yang dihasilkan telah berhasil disimpan ke dalam database yang telah dibentuk.

Node-Red yang telah kami bentuk dapat dilihat sebagai berikut:

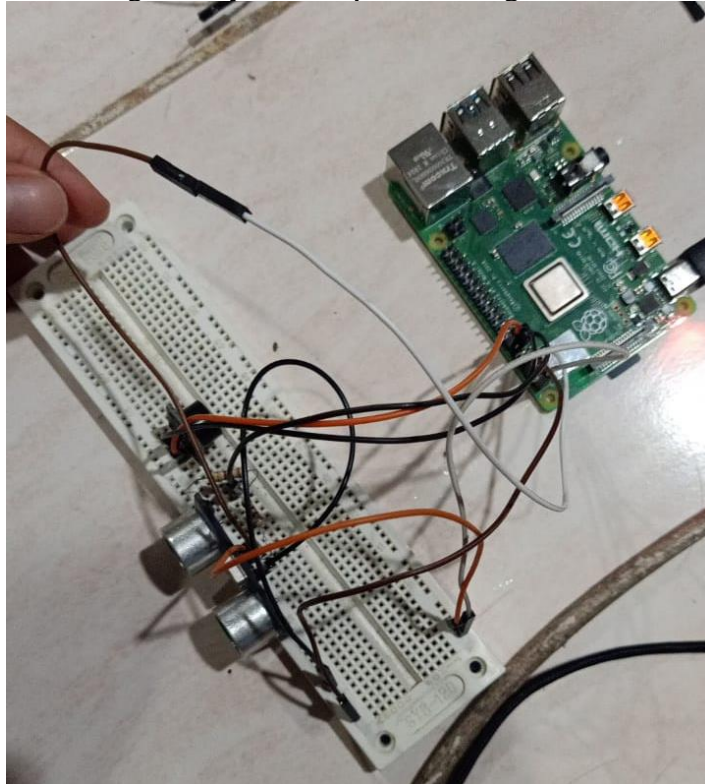


Dari rangkaian Node-Red diatas dapat kita simpulkan bahwa node-red berhasil konek menghubungkan antara MQTTX dan InfluxDB.

- Kelompok 7 :

1. Infrastruktur Sistem

Pada bagian ini kami membuat kontrol ketinggian air menggunakan sensor ultrasonik. untuk rangkaiannya kita dapat melihat gambar dibawah ini.

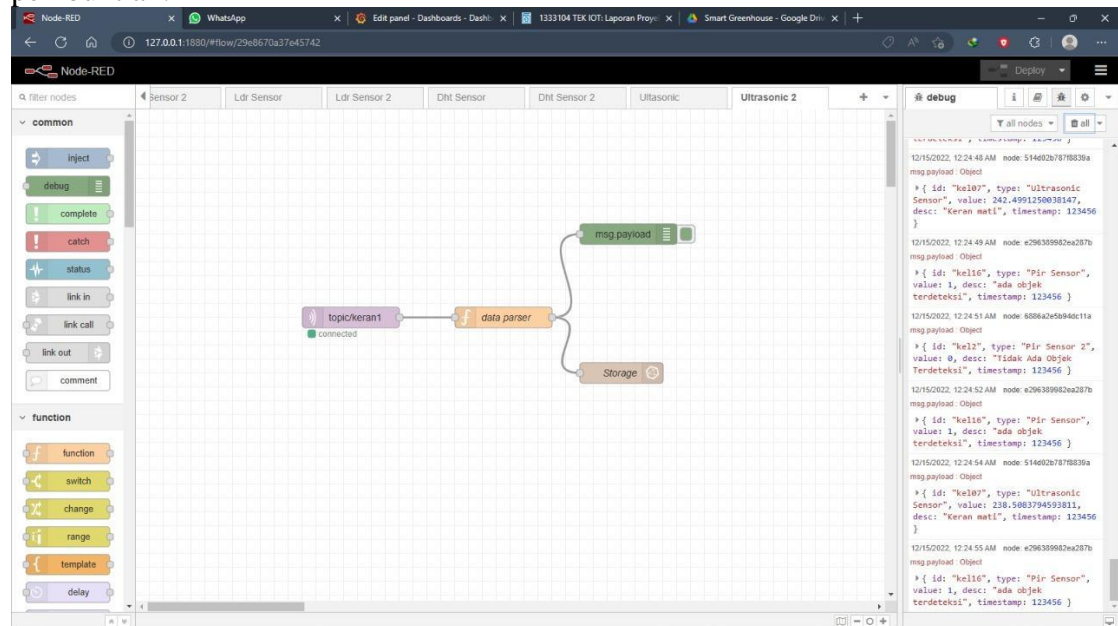


Pada rangkaian di atas dapat kita lihat ada raspberry pi sebagai mikrokontroler pada sistem yang kami buat. Dimana sensor ultrasonic sebagai pengontrol ketinggian air. Apabila nilai sensor bernilai lebih kecil dari 10 maka keran akan hidup untuk mengisi air. setelah itu raspberry pi akan mempublish hasil dari sensor yang terdeteksi ke broker MQTT yaitu MQTTX. MQTTX akan melanjutkan hasil dari sensor ke Influxdb melalui Node-Red sebagai penghubung dimana influxdb sebagai database untuk menyimpan hasil sensor.

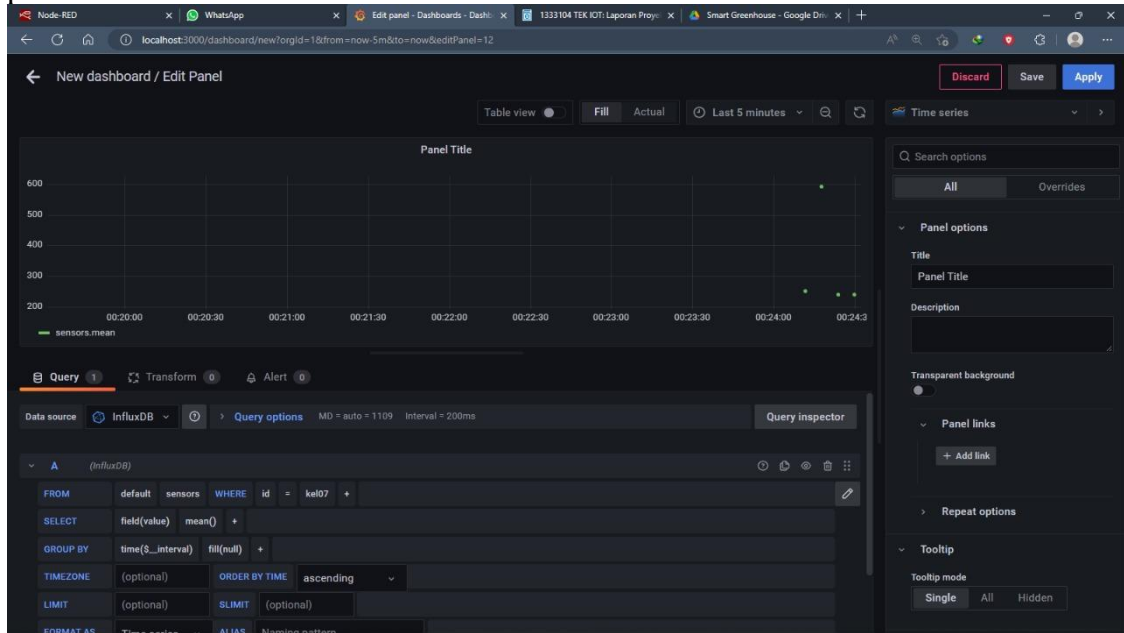
Database yang terbentuk di dalam InfluxDB dapat kita lihat pada gambar berikut:

1671839087738170800	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671839089740121500	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671839091744416400	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671839093745941500	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671839095753767600	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671839097755358700	ada objek terdeteksi	kel16	123456	Pir Sensor	1
167183909759745500	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671839101788591000	Keran mati	kel07	123456	Ultrasonic Sensor	251.85447931289673
1671839102765390800	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671839103802479300	Keran mati	kel07	123456	Ultrasonic Sensor	247.50800132751465
1671839104769215000	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671839105819446400	Keran mati	kel07	123456	Ultrasonic Sensor	247.34444618225098
1671839106774162400	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671839107838232300	Keran mati	kel07	123456	Ultrasonic Sensor	247.07049131393433
1671839108778599700	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671839109852531000	Keran mati	kel07	123456	Ultrasonic Sensor	251.83812379837036
1671839110780512600	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671839111891108900	Keran mati	kel07	123456	Ultrasonic Sensor	594.9236631393433
1671839112784408800	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671839113907843900	Keran mati	kel07	123456	Ultrasonic Sensor	250.90177059173584
16718391147911472200	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671839116000791100	Keran mati	kel07	123456	Ultrasonic Sensor	250.93857049942017
1671839116797988800	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671839118079052800	Keran mati	kel07	123456	Ultrasonic Sensor	57.00305700302124
1671839118794883300	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671839119962770000	Keran mati	kel07	123456	Ultrasonic Sensor	250.9671926498413
1671839120800310900	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671839121980995100	Keran mati	kel07	123456	Ultrasonic Sensor	252.28790044784546
1671839122803314300	ada objek terdeteksi	kel16	123456	Pir Sensor	1

data sensor akan dikirimkan menggunakan raspberry pi ke MQTT node-red dan kemudian data akan dimasukkan ke dalam influx db
pembuktian:



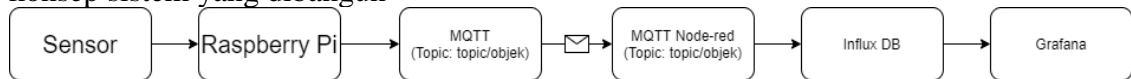
kemudian data yang masuk pada influx db akan divisualisasikan menggunakan grafana pembuktian:



- Kelompok 16 :

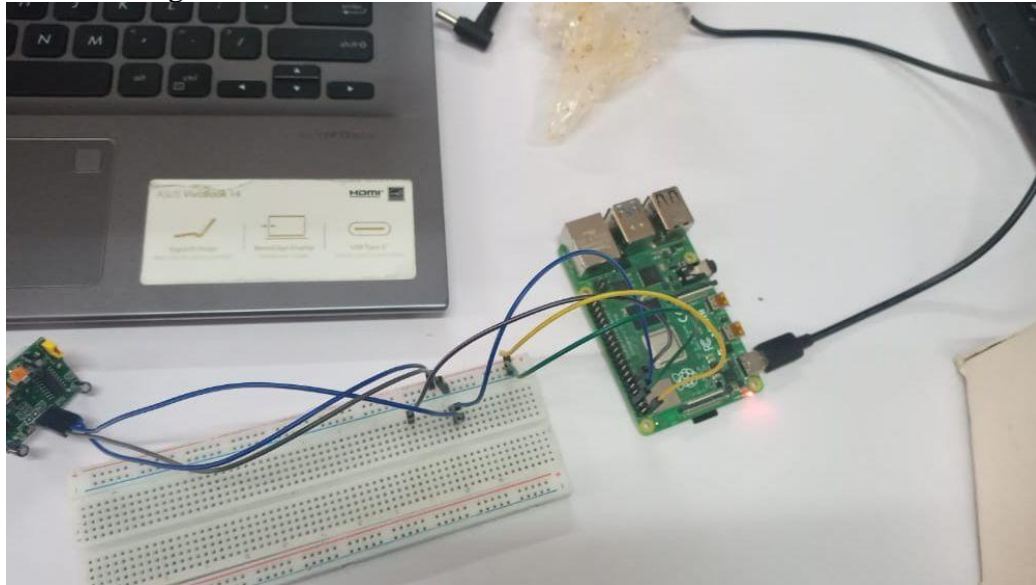
1. infrastruktur sistem

konsep sistem yang dibangun



sensor yang kami gunakan adalah sensor pir dan aktuator kami adalah buzzer yang dimana sensor pir akan mendeteksi objek dan jika ada objek yang terdeteksi maka buzzer akan menyala dan sesuai dengan konsep sistem di atas, bahwa sensor dan aktuator akan dihubungkan dengan raspberry pi kemudian data sensor akan dikirimkan melalui MQTT dan kemudian MQTT node-red akan menerimanya kemudian data dari MQTT node-red akan dimasukkan ke dalam influxdb dan kemudian akan divisualisasikan menggunakan grafana.

Gambar Rangkaian:

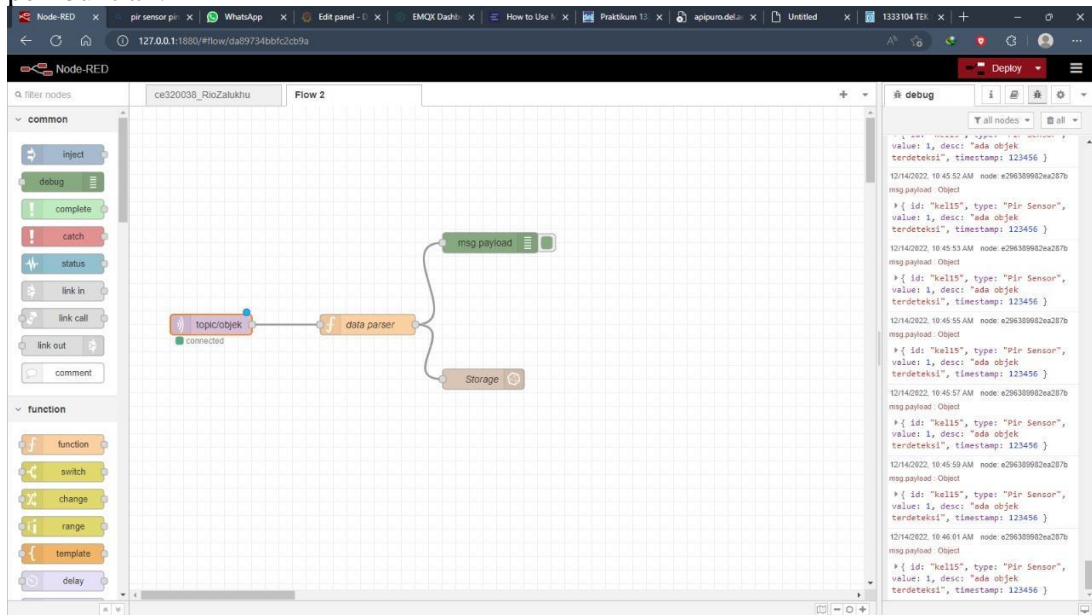


penjelasan: PIR sensor akan dihubungkan dengan raspberry pi yaitu pin vcc,ground,dan output kemudian buzzer juga dihubungkan dengan raspberry pi kemudian sensor akan mendeteksi objek dengan input data jika data sensor bernilai 1 maka buzzer akan hidup dan akan menampilkan pesan “ada objek terdeteksi” dan jika data sensor bernilai 0 maka buzzer akan tetap mati ada ada pesan yang muncul yaitu “tidak ada objek yang terdeteksi”

Hasil:

data sensor akan dikirimkan menggunakan raspberry pi ke MQTT node-red dan kemudian data akan dimasukkan ke dalam influx db

pembuktian:

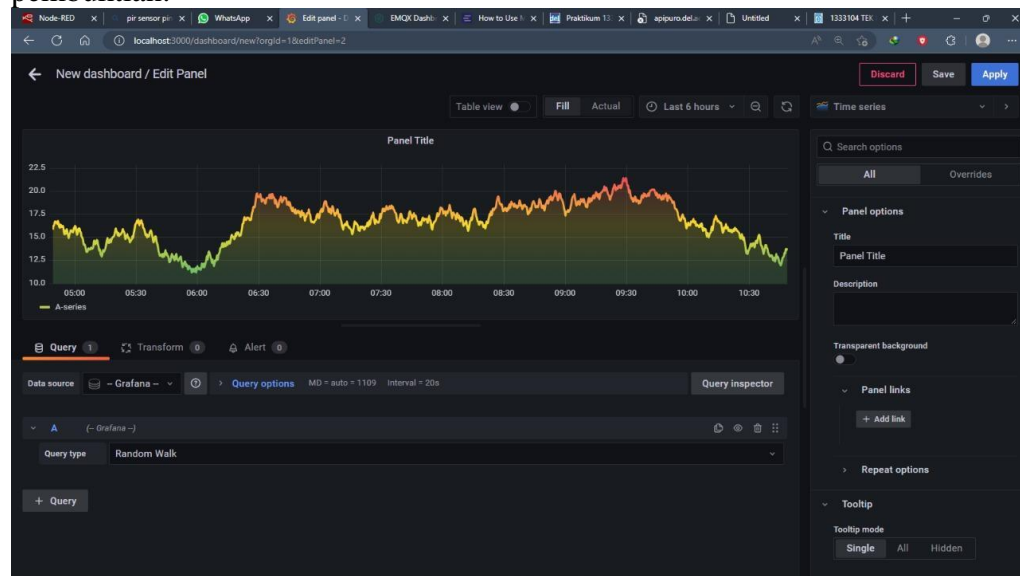


kemudian kita bisa melihat data yang masuk tadi dengan melihat data yang masuk pada influx db

pembuktian:

1679988076642197880	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988078642853580	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988078643854280	ada objek terdeteksi	kel15	123456	Pir Sensor	1
167998808064613280	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988080645649980	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988082647527980	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988084651553680	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988086656896880	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988088661928880	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988090666666980	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988092668601880	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988092696922880	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988094668815280	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988096669876980	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988098672902480	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988098673902580	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988108674748880	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988102884272680	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988102885271280	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988104788475780	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988106684798880	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988106685881780	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988108689486180	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988108688567280	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988112684837880	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988114687452580	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988116782876880	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988118785147180	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988120786620280	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988122718414680	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988124713471980	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988126718279880	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988128728334880	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988130726293380	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988132738668580	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988134738378880	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988136735688880	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988136736611580	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988138737146380	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988140738908680	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988142728299980	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988142739391880	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988144704713780	ada objek terdeteksi	kel15	123456	Pir Sensor	1
1679988144705713680	ada objek terdeteksi	kel15	123456	Pir Sensor	1

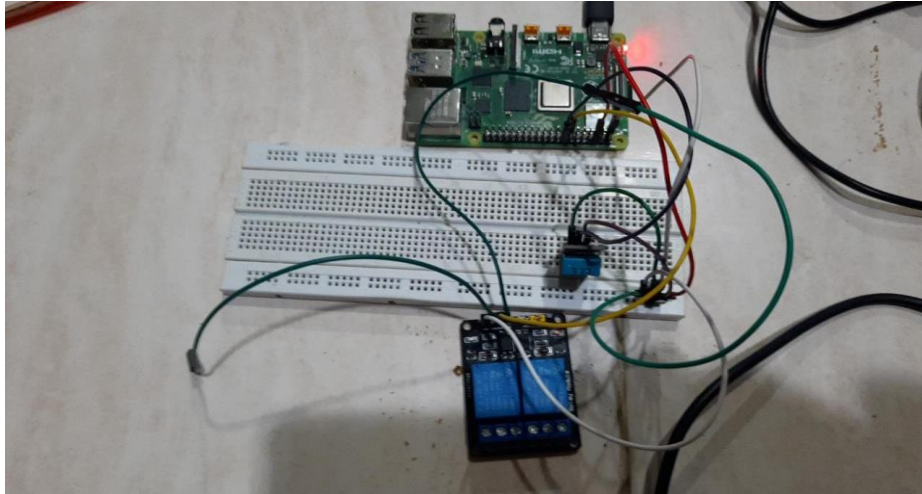
kemudian data yang masuk pada influx db akan divisualisasikan menggunakan grafana pembuktian:



- Kelompok 21 :

1. infrastruktur sistem

Pada proyek ini, sensor yang kami gunakan adalah sensor DHT-11 dan aktuator yang kami gunakan adalah relay untuk menjalankan kipas. Namun, kipas pada saat ini belum tersedia, sehingga kami menggunakan aktuator relay sebagai asumsi bahwa itu adalah kipas. DHT-11 akan mengukur kondisi suhu pada ruangan dan relay sebagai output. Jika DHT-11 mendeteksi suhu dengan ketentuan tertentu maka relay akan menerima output. Sensor dan aktuator akan dihubungkan dengan raspberry pi kemudian data sensor akan dikirimkan melalui MQTT dan kemudian MQTT node-red akan menerimanya kemudian data dari MQTT node-red akan dimasukkan ke dalam influxdb dan kemudian akan divisualisasikan menggunakan grafana.

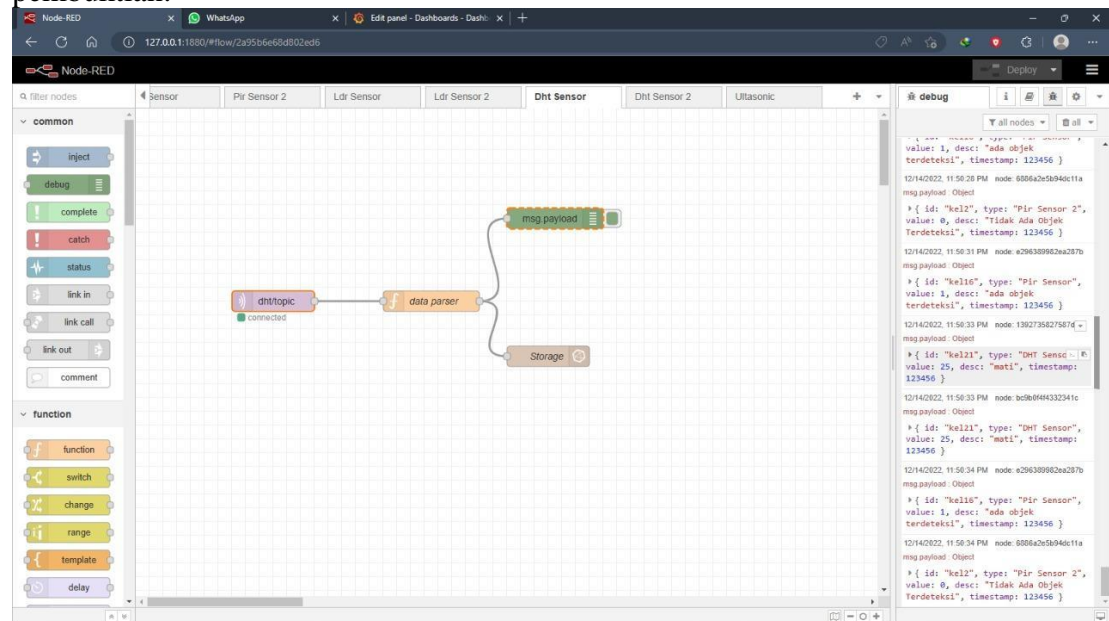


penjelasan: Sensor DHT-11 akan dihubungkan dengan raspberry pi yaitu pin vcc,ground,dan output kemudian relay juga dihubungkan dengan raspberry pi kemudian sensor akan mendeteksi suhu dengan input data jika data sensor bernilai diatas 28°C maka Relay akan hidup dan akan menampilkan pesan “Hidup” dan jika data sensor bernilai dibawah 28°C maka Relay akan tetap mati ada ada pesan yang muncul yaitu “Mati”

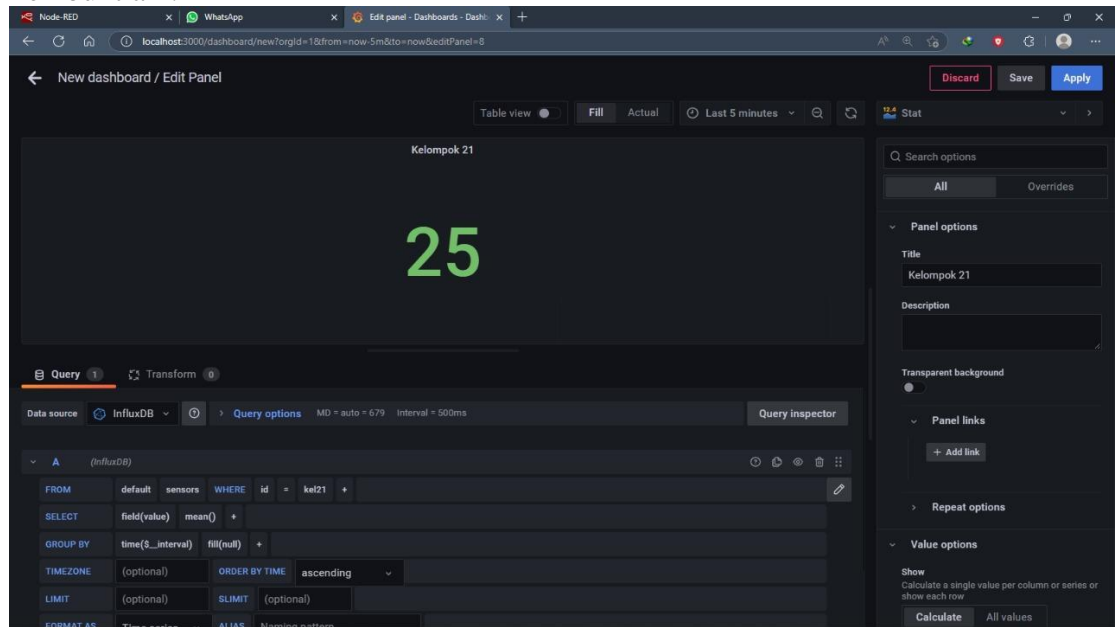
Hasil:

Data sensor akan dikirimkan menggunakan raspberry pi ke MQTT node-red dan kemudian data akan dimasukkan ke dalam influx db

pembuktian:



Kemudian data yang masuk pada influx db akan divisualisasikan menggunakan grafana
Pembuktian :



- **Kelompok 22 :**

- 1. **Infrastruktur Sistem**

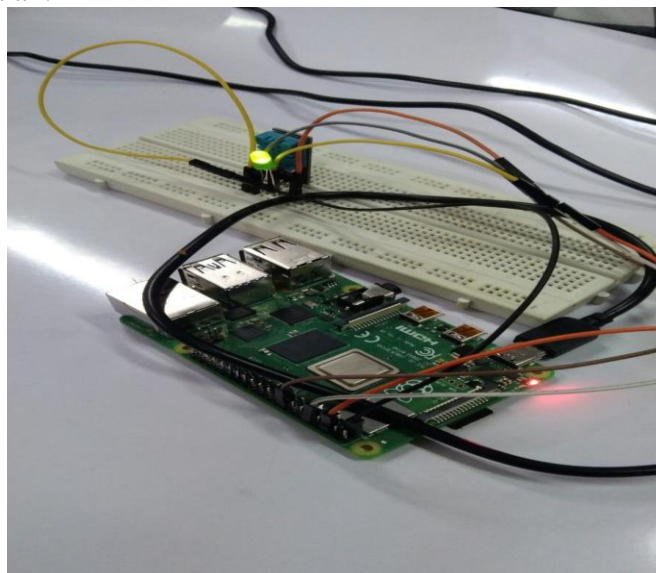
- Pada proyek ini, sensor yang kami gunakan adalah sensor DHT-11 dan aktuator kami yang gunakan adalah Relay.

- Konsep:**

- Saat kelembapan udara $>70\%$,maka kipas akan menyala

- Saat kelembapan udara $<30\%$ maka pompa air akan menyala

Gambar rangkaian:

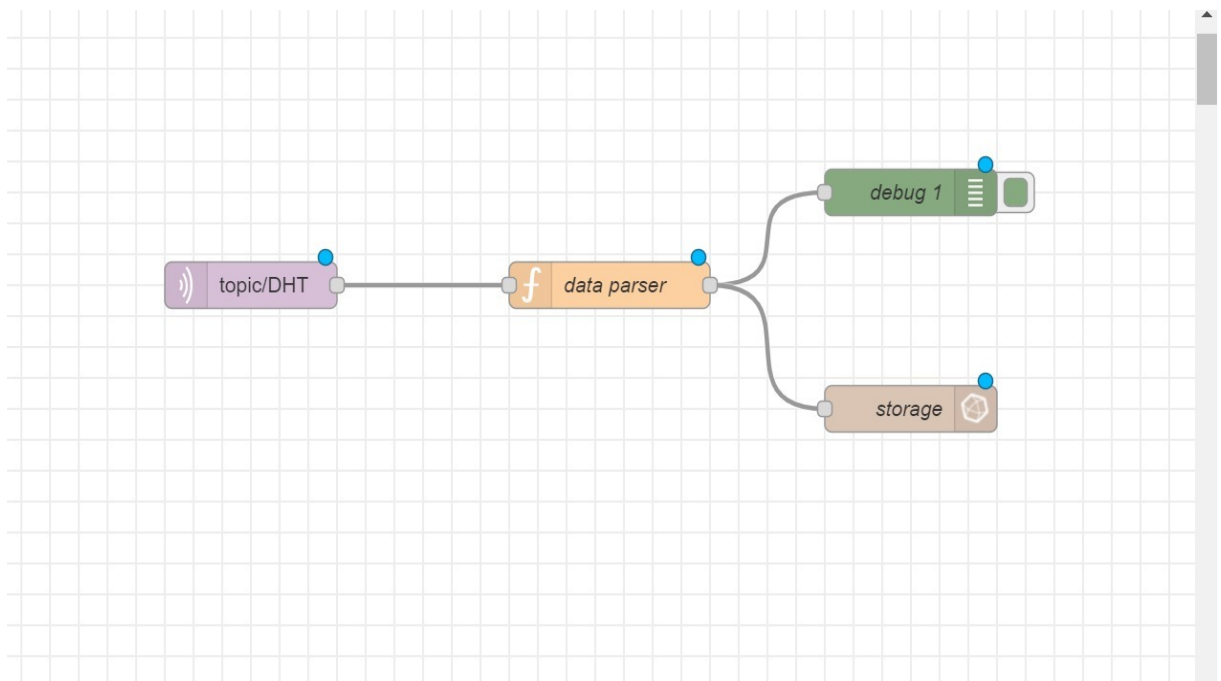


Berikut komponen yang digunakan dalam proyek ini:

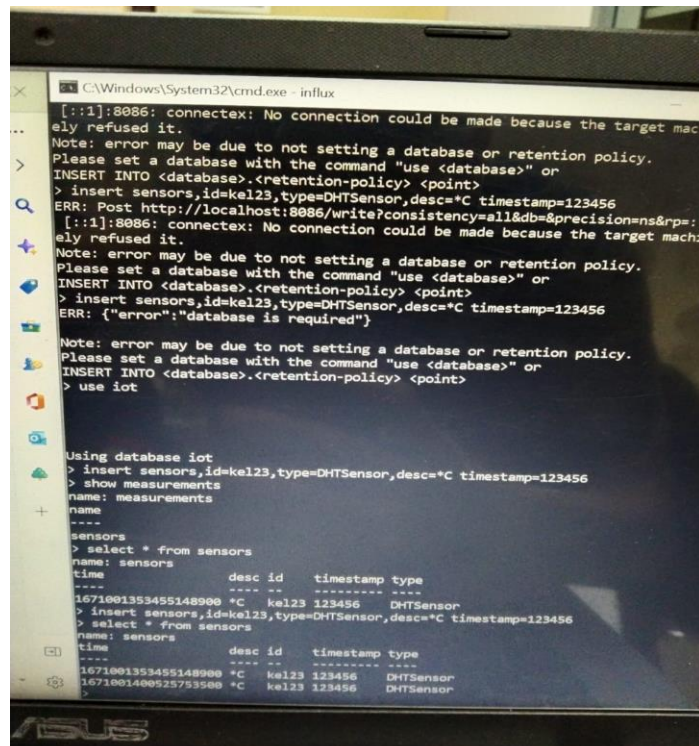
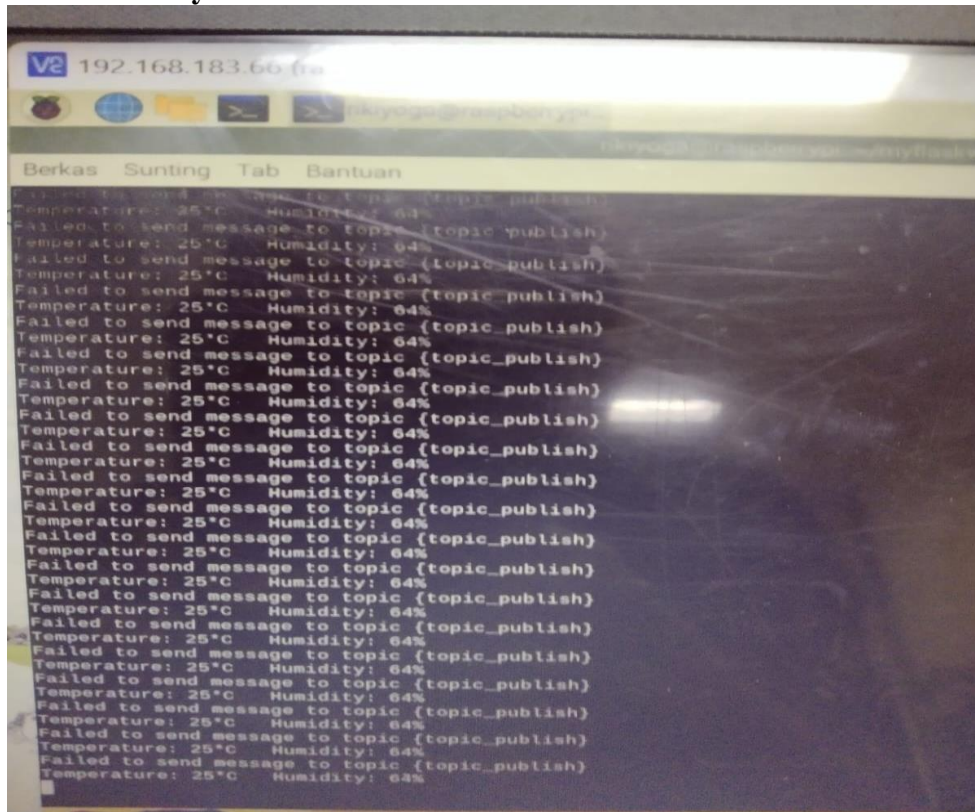
- Relay 2 buah
- Sensor DHT11
- Raspberry
- Kabel Jumper
- Pin
- Bread Board
- Resistor

Hasil:

Lalu, data sensor akan dikirimkan menggunakan raspberry pi ke MQTT node-red dan kemudian data akan dimasukkan ke dalam influx db



Kemudian,kita dapat melihat data yang masuk tadi dengan melihat data yang masuk pada influx db
Pembuktiannya :



- Kelompok 23 :

1. Infrastruktur Sistem

Pada infrastruktur sistem ini menjelaskan peralatan atau komponen yang dibutuhkan untuk membangun sistem kontrol ketinggian air. Ada beberapa komponen yang diperlukan untuk membangun sistem, sebagai berikut:

- Sensor Ultrasonik
- Raspberry
- Kabel jumper
- Breadboard
- Resistor
- led

Pada proyek ini, sensor yang kami gunakan adalah sensor ultrasonik dan aktuator kami yang gunakan adalah led.

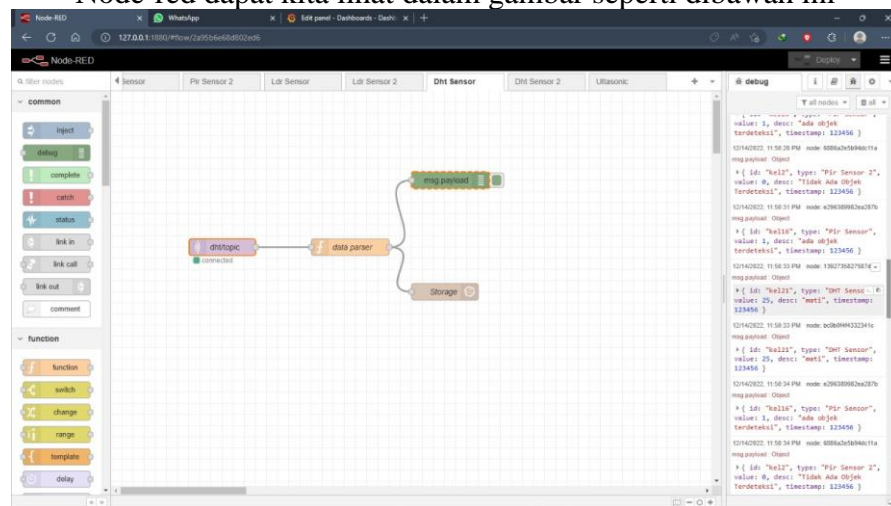
Konsep :

Pada proyek ini kelompok kami membuat deteksi ketinggian air menggunakan sensor ultrasonik dan led. Dimana nilai ultrasonik berubah seiring ketinggian air yang diterima. Jika nilai sensor dibawah 50% maka nilai led akan menyala, jika nilai sensor diatas 50 maka led mati.





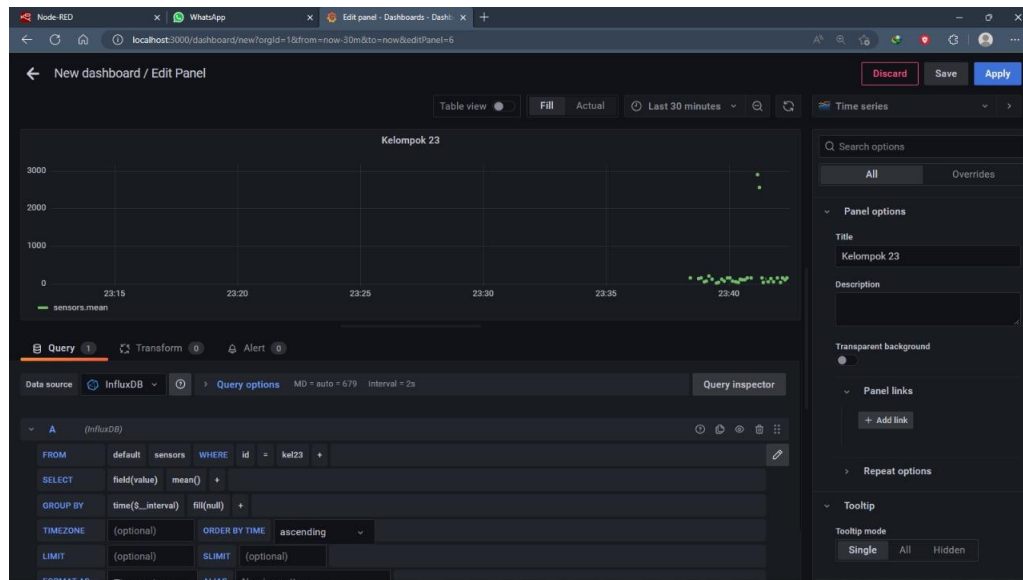
Node-red dapat kita lihat dalam gambar seperti dibawah ini



Influxdb :

1671036330649587900	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671036333652227800	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671036336495306900	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor 2	0
1671036336657811800	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671036337065343200	Keran Nyala	kel23	123456	ultrasonic	124.78439807891846
1671036338658974200	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671036341667687700	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671036342503709700	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor 2	0
1671036343485507500	Keran Nyala	kel23	123456	ultrasonic	3934.9364399909973
1671036344668986900	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671036347675662500	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671036348511737600	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor 2	0
1671036349920468400	Keran Nyala	kel23	123456	ultrasonic	92.2123908996582
1671036350680529600	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671036353687616800	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671036354185095300	Keran Nyala	kel23	123456	ultrasonic	166.37647151947021
1671036354520480600	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor 2	0
1671036356692017600	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671036358693607400	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671036360734893700	Keran Nyala	kel23	123456	ultrasonic	59.108829498291016
1671036361697525500	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671036362530688200	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor 2	0
16710363640812609600	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671036367706143000	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671036368546123500	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor 2	0
1671036369843778000	Keran Nyala	kel23	123456	ultrasonic	168.54357719421387
1671036370712364800	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671036373719336500	ada objek terdeteksi	kel16	123456	Pir Sensor	1
1671036375978118100	Keran Nyala	kel23	123456	ultrasonic	121.08805179595947
1671036376553474500	Tidak Ada Objek Terdeteksi	kel2	123456	Pir Sensor 2	0

kemudian data yang masuk pada influx db akan divisualisasikan menggunakan grafana pembuktian:



C. Penutup

Berdasarkan hasil uji dan pembahasan implementasi konsep Internet Of Things pada sistem Smart Green House maka diperoleh kesimpulan :

1. Dari hasil uji masing-masing kelompok dapat beroperasi dengan baik.
2. Hasil uji pengiriman data ke grafana dapat diterima dengan baik.

D. Lampiran

1. Lampiran kode sistem pendeteksi hama dengan aktuator buzzer

```
import random
import json
import time
from paho.mqtt import client as mqtt_client

GPIO.setmode(GPIO.BOARD)

GPIO_PIR = 7
GPIO_BUZZER = 11

GPIO.setup(GPIO_BUZZER, GPIO.OUT)
GPIO.setup(GPIO_PIR, GPIO.IN)

broker = '192.168.181.19'
port = 1883
topic_publiimport RPi.GPIO as GPIOsh = "topic/objek"
topic_subscribe = "topic/windSpeed"
client_id = 'python-mqtt'
username
password = ""
id_dev = 'kel15'
tipe = 'Pir Sensor'
timestamp = 123456
def connect_mqtt():
    def on_connect(client, userdata, flags, rc):
        if rc == 0:
            print("connected to mqtt broker!!")
        else:
            print("Failed to connect, return code %d\n", rc)

    client = mqtt_client.Client(client_id)
    client.username_pw_set(username, password)
    client.on_connect = on_connect
    client.connect(broker, port)
    return client

def publishSubscribe(client):
    msg_count = 0
    while True:
```



```

value = GPIO.input(GPIO_PIR)
if value == 1 :
    GPIO.output(GPIO_BUZZER, 1)
    status = "ada objek terdeteksi"
else:
    GPIO.output(GPIO_BUZZER, 0)
    status = "tidak ada objek terdeteksi"

objek = value
msg = json.dumps({ "id": id_dev, "type": tipe, "value": objek, "desc": status,
"timestamp": timestamp })
result = client.publish(topic_publish, msg)
status = result [0]
if status == 0:
    print("send" +msg+"to topic"+topic_publish)
else:
    print("Failed to send message to topic {topic_publish}")
def on_message(client, userdata, msg):
    print("received" +str(msg.payload)+"from topic")
    client.subscribe(topic_subscribe)
    client.on_message = on_message
    time.sleep(2)
def run():
    client = connect_mqtt()
    client.loop_start()
    publishSubscribe(client)

if __name__ == '__main__':
    run()

```

2. Lampiran kode Kontrol suhu menggunakan sensor DHT 11

```

import RPi.GPIO as GPIO
import time
import board
import adafruit_dht
import psutil
import json
import random
from paho.mqtt import client as mqtt_client

for proc in psutil.process_iter():
    if proc.name() == 'libgpiod_pulsein' or proc.name() == 'libgpiod_pulsei':

```

```

    proc.kill()
sensor = adafruit_dht.DHT11(board.D23)
led = 10
buzzer = 15
temp = sensor.temperature
humidity = sensor.humidity

GPIO.setup(led, GPIO.OUT)
GPIO.setup(buzzer, GPIO.OUT)

broker = '192.168.183.187'
port = 8083
topic_publish = "topic/objek"
client_id = 'python-mqtt'
username = ""
password = ""
id_dev = 'kel23'
tipe = 'DHT Sensor'
timestamp = 123456

def connect_mqtt():
    def on_connect(client, userdata, flags, rc):
        if rc == 0:
            print("connected to mqtt broker!!")
        else:
            print("Failed to connect, return code %d\n", rc)

    client = mqtt_client.Client(client_id)
    client.username_pw_set(username, password)
    client.on_connect = on_connect
    client.connect(broker, port)
    return client

def publishSubscribe(client):
    msg_count = 0
    while True:
        try:
            print("Temperature: {}*C Humidity: {}% ".format(temp, humidity))

            if humidity < 80 :
                GPIO.output(buzzer,1)
                GPIO.output(led,0)
                status = "hidup"
            else:
                GPIO.output(buzzer,0)
                GPIO.output(led,1)

```

```

        status = "mati"

    except RuntimeError as error:
        print(error.args[0])
        time.sleep(2.0)
        continue
    except Exception as error:
        dhtDevice.exit()
        raise error
        time.sleep(2.0)
    objek = humidity
    msg = json.dumps({ "id": id_dev, "type": tipe, "value": objek, "desc": status,
"timestamp": timestamp })
    result = client.publish(topic_publish, msg)
    status = result [0]

    if status == 0:
        print("send" +msg+"to topic"+topic_publish)
    else:
        print("Failed to send message to topic {topic_publish}")
def on_message(client, userdata, msg):
    print("received" +str(msg.payload)+"from topic")
    client.subscribe(topic_subscribe)
    client.on_message = on_message
    time.sleep(2)
def run():
    client = connect_mqtt()
    client.loop_start()
    publishSubscribe(client)

if __name__ == '__main__':
    run()

```

3. Lampiran kode sistem pendeteksi hama dengan aktuator relay

```
import RPi.GPIO as GPIO
import random
import json
import time
from paho.mqtt import client as mqtt_client

GPIO.setmode(GPIO.BOARD)

GPIO_PIR = 13
GPIO_RELAY = 15

GPIO.setup(GPIO_RELAY, GPIO.OUT)
GPIO.setup(GPIO_PIR, GPIO.IN)

broker = '192.168.209.19'
port = 1883
topic_publish = "topic/objek2"
topic_subscribe = "topic/windSpeed"
client_id = 'python-mqtt'
username = ""
password = ""
id_dev = 'kel2'
tipe = 'Pir Sensor 2'
timestamp = 123456
def connect_mqtt():
    def on_connect(client, userdate, flags, rc):
        if rc == 0:
            print("connected to mqtt broker!!")
        else:
            print("Failed to connect, return code %d\n", rc)

    client = mqtt_client.Client(client_id)
    client.username_pw_set(username, password)
    client.on_connect = on_connect
    client.connect(broker, port)
    return client

def publishSubscribe(client):
    msg_count = 0
    while True:

        value = GPIO.input(GPIO_PIR)
        if value == 1 :
```

```

        GPIO.output(GPIO_RELAY, 1)
        status = "Ada Objek Terdeteksi"
    else:
        GPIO.output(GPIO_RELAY, 0)
        status = "Tidak Ada Objek Terdeteksi"

    objek = value
    msg = json.dumps({ "id": id_dev, "type": tipe, "value": objek, "desc": >
    result = client.publish(topic_publish, msg)
    status = result [0]
    if status == 0:
        print("send" +msg+"to topic"+topic_publish)
    else:
        print("Failed to send message to topic {topic_publish}")
    def on_message(client, userdata, msg):
        print("received" +str(msg.payload)+"from topic")
    client.subscribe(topic_subscribe)
    client.on_message = on_message
    time.sleep(2)
def run():
    client = connect_mqtt()
    client.loop_start()
    publishSubscribe(client)

if __name__ == '__main__':
    run()

```

4. Kontrol suhu menggunakan sensor DHT11 untuk Menjalankan Kipas

```
import RPi.GPIO as GPIO
import time
import board
import adafruit_dht
import psutil
import json
import random
from paho.mqtt import client as mqtt_client

for proc in psutil.process_iter():
    if proc.name() == 'libgpiod_pulsein' or proc.name() == 'libgpiod_pulsei':
        proc.kill()
sensor = adafruit_dht.DHT11(board.D23)
in1 = 15

temp = sensor.temperature
humidity = sensor.humidity

GPIO.setup(in1, GPIO.OUT)

broker = '192.168.205.84'
port = 1883
topic_publish = "dht/topic"
topic_subscribe = "dht/topic2"
client_id = 'python-mqtt'
username = ""
password = ""
id_dev = 'kel21'
tipe = 'DHT Sensor'
timestamp = 123456

def connect_mqtt() :
    def on_connect(client, userdata, flags, rc):
        if rc == 0:
            print("Connected to MQTT Broker!")
        else:
            print("Failed to connect,return code %d\n", rc)
    client = mqtt_client.Client(client_id)
    client.username_pw_set(username,password)
    client.on_connect = on_connect
    client.connect(broker, port)
    return client

def publishSubscribe(client):
```

```

msg_count = 0
while True:
    try:
        print("Temperature: {}*C Humidity: {}% ".format(temp, humidity))

        if humidity < 20 :
            GPIO.output(in1,1)
            status = "hidup"
        else:
            GPIO.output(in1,0)
            status = "mati"
    except RuntimeError as error:
        print(error.args[0])
        time.sleep(1.0)
        continue
    except Exception as error:
        sensor.exit()
        raise error
    time.sleep(2.0)
    objek = humidity
    msg = json.dumps({ "id": id_dev, "type": tipe, "value": objek, "desc": status, "timestamp":
timestamp })
    result = client.publish(topic_publish, msg)
    status = result [0]

    if status == 0:
        print("send" +msg+"to topic"+topic_publish)
    else:
        print("Failed to send message to topic {topic_publish}")
    def on_message(client, userdata, msg):
        print("received" +str(msg.payload)+"from topic")
    client.subscribe(topic_subscribe)
    client.on_message = on_message
    time.sleep(2)
def run():
    client = connect_mqtt()
    client.loop_start()
    publishSubscribe(client)

if __name__ == '__main__':
    run()

```

5. Deteksi Intensitas cahaya menggunakan sensor LDR

```
import random
import time
import json
import RPi.GPIO as GPIO
from paho.mqtt import client as mqtt_client

broker = '192.168.209.19'
port = 1883
topic_publish = "ldr/topic"
client_id = 'python-mqtt'
username = ''
password = ''
id_dev = 'kel05'
tipe = 'LDR Sensor'
timestamp = 123456

ldr = 7
in1 = 15
in2 = 18

GPIO.setmode(GPIO.BOARD)
GPIO.setup(ldr, GPIO.IN)
GPIO.setup(in1, GPIO.OUT)
GPIO.setup(in2, GPIO.OUT)

GPIO.output(in1, False)
GPIO.output(in2, False)

def connect_mqtt() :
    def on_connect(client, userdata, flags, rc):
        if rc == 0:
            print("Connected to MQTT Broker!")
        else:
            print("Failed to connect,return code %d\n", rc)
    client = mqtt_client.Client(client_id)
    client.username_pw_set(username,password)
    client.on_connect = on_connect
    client.connect(broker, port)
```



```

        return client

def publishSubscribe(client):
    msg_count = 0
    while True:
        print(GPIO.input(ldr))
        if GPIO.input(ldr)==1:
            print("LED OFF")
            GPIO.output(in1, True)
            GPIO.output(in2, False)
            msg = json.dumps({"id":id_dev , "type" : tipe, "value" : 1,
"desc":"OFF","timestamp": timestamp})
            result = client.publish(topic_publish, msg)
            time.sleep(2)
        else:
            print("LED ON")
            GPIO.output(in1, False)
            GPIO.output(in2, True)
            msg = json.dumps({"id":id_dev , "type" : tipe, "value" : 1,
"desc":"ON","timestamp": timestamp})
            result = client.publish(topic_publish, msg)
            time.sleep(2)

    status = result[0]
    if status == 0:
        print("Send "+msg+" to topic {topic_publish}")
    else:
        print("Failed to send message to topic {topic_publish}")

def on_message(client, userdata, msg):
    print("Received "+str(msg.payload)+" from topic")
    client.on_message = on_message
    time.sleep(1)

def run():
    client = connect_mqtt()
    client.loop_start()
    publishSubscribe(client)

if __name__ == '__main__':
    run()

```

6. Deteksi Intensitas cahaya menggunakan sensor LDR

```
import random
import time
import json
import RPi.GPIO as GPIO
from paho.mqtt import client as mqtt_client

broker = '192.168.155.19'
port = 1883
topic_publish = "ldr1/topic"
client_id = 'python-mqtt'
username = ''
password = ''
id_dev = 'kel6'
tipe = 'LDR Sensor'
timestamp = 123456

ldr = 7
in1 = 15
in2 = 18

GPIO.setmode(GPIO.BOARD)
GPIO.setup(ldr, GPIO.IN)
GPIO.setup(in1, GPIO.OUT)
GPIO.setup(in2, GPIO.OUT)

GPIO.output(in1, False)
GPIO.output(in2, False)

def connect_mqtt() :
    def on_connect(client, userdata, flags, rc):
        if rc == 0:
            print("Connected to MQTT Broker!")
        else:
            print("Failed to connect,return code %d\n", rc)
    client = mqtt_client.Client(client_id)
    client.username_pw_set(username,password)
    client.on_connect = on_connect
    client.connect(broker, port)
    return client
```

```

def publishSubscribe(client):
    msg_count = 0
    while True:
        print(GPIO.input(ldr))
        if GPIO.input(ldr)==1:
            print("Tirai Terbuka")
            GPIO.output(in1, True)
            GPIO.output(in2, False)
            msg = json.dumps({
                "id":id_dev,"type":tipe,"value":1,"desc":"ON","timestamp":timestamp})
            result = client.publish(topic_publish, msg)
            time.sleep(2)
        else:
            print("Tirai Tertutup")
            GPIO.output(in1, False)
            GPIO.output(in2, True)
            msg = json.dumps({
                "id":id_dev,"type":tipe,"value":0,"desc":"OFF","timestamp":timestamp})
            result = client.publish(topic_publish, msg)
            time.sleep(2)

    status = result[0]
    if status == 0:
        print("Send "+msg+" to topic {topic_publish}")
    else:
        print("Failed to send message to topic {topic_publish}")

    def on_message(client, userdata, msg):
        print("Received "+str(msg.payload)+" from topic")
    client.on_message = on_message
    time.sleep(1)

def run():
    client = connect_mqtt()
    client.loop_start()
    publishSubscribe(client)

if __name__ == '__main__':
    run()

```

7. Kontrol Ketinggian air menggunakan sensor Ultrasonik

```
import random
import time
import json
import RPi.GPIO as GPIO
from paho.mqtt import client as mqtt_client

broker = '192.168.155.84'
port = 1883
topic_publish = "topic/keran"
client_id = 'python-mqtt'
username = ''
password = ''
GPIO.setwarnings(False)

#GPIO Mode (BOARD / BCM)
GPIO.setmode(GPIO.BOARD)

#set GPIO Pins
GPIO_TRIGGER = 7
GPIO_ECHO = 11
BUZZER = 13

#set GPIO direction (IN / OUT)
GPIO.setup(BUZZER, GPIO.OUT)
GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
GPIO.setup(GPIO_ECHO, GPIO.IN)
def connect_mqtt():
    def on_connect(client, userdata, flags, rc):
        if rc == 0:
            print("Connected to MQTT Broker!")
        else:
            print("Failed to connect, return code %d\n", rc)
    client = mqtt_client.Client(client_id)
    client.username_pw_set(username, password)
    client.on_connect = on_connect
    client.connect(broker, port)
    return client

def publishSubscribe(client):
```

```

msg_count = 0
while True:
    GPIO.output(GPIO_TRIGGER, True)
    time.sleep(2)
    GPIO.output(GPIO_TRIGGER, False)
    StartTime = time.time()
    StopTime = time.time()
    while GPIO.input(GPIO_ECHO) == 0:
        StartTime = time.time()
    while GPIO.input(GPIO_ECHO) == 1:
        StopTime = time.time()
    TimeElapsed = StopTime - StartTime
    distance = (TimeElapsed * 34300) / 2
    if distance < 10 :
        GPIO.output(BUZZER, True)
        status = "Keran menyala"
    else:
        GPIO.output(BUZZER, False)
        status = "Keran mati"
    jarak = str(distance)
    msg = json.dumps({"jarak": jarak, "status_Keran": status})
    result = client.publish(topic_publish, msg)
    status = result[0]
    if status == 0:
        print("send "+msg+" to topic "+topic_publish)
    else:
        print("Failed to send message to topic {topic_publish}")
def on_message(client, userdata, msg):
    print("Received"+str(msg.payload)+"from topic")
client.subscribe(topic_subscribe)
client.on_message = on_message
time.sleep(1)

def run():
    client = connect_mqtt()
    client.loop_start()
    publishSubscribe(client)

if __name__ == '__main__':
    run()

```

8. Kontrol Ketinggian air menggunakan sensor Ultrasonik

```
import random
import time
import json
import RPi.GPIO as GPIO
from paho.mqtt import client as mqtt_client

broker = '192.168.155.19'
port = 1883
topic_publish = "topic/keran"
topic_subscribe = "topic/air"
id_dev = "kel23"
tipe = "ultrasonic"
timestamp = 123456
client_id = 'python-mqtt'
username = ''
password = ''

GPIO.setwarnings(False)
#GPIO Mode (BOARD / BCM)
GPIO.setmode(GPIO.BOARD)
#set GPIO Pins
GPIO_TRIGGER = 7
GPIO_ECHO = 11
led = 23

#set GPIO direction (IN / OUT)
GPIO.setup(led, GPIO.OUT)
GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
GPIO.setup(GPIO_ECHO, GPIO.IN)

def connect_mqtt():
    def on_connect(client, userdata, flags, rc):
        if rc == 0:
            print("Connected to MQTT Broker!")
        else:
            print("Failed to connect, return code %d\n", rc)
    client = mqtt_client.Client(client_id)
```

```

client.username_pw_set(username, password)
client.on_connect = on_connect
client.connect(broker, port)
return client

def publishSubscribe(client):
    msg_count = 0
    while True:
        GPIO.output(GPIO_TRIGGER, True)
        time.sleep(2)
        GPIO.output(GPIO_TRIGGER, False)
        StartTime = time.time()
        StopTime = time.time()
        while GPIO.input(GPIO_ECHO) == 0:
            StartTime = time.time()
        while GPIO.input(GPIO_ECHO) == 1:
            StopTime = time.time()
        TimeElapsed = StopTime - StartTime
        distance = (TimeElapsed * 34300) / 2
        if distance > 50 :
            GPIO.output(led, True)
            status = "Keran Nyala"
        else :
            GPIO.output(led, False)
            status = "Keran Mati"

        jarak = distance
        msg = json.dumps({"id":id_dev,"type":tipe,"value":jarak,"desc":status,"timestamp":timestamp})
        result = client.publish(topic_publish, msg)
        status = result[0]
        if status == 0:
            print("send "+msg+" to topic "+topic_publish)
        else:
            print("Failed to send message to topic {topic_publish}")
    def on_message(client, userdata, msg):
        print("Received"+str(msg.payload)+"from topic")
    client_subscribe(topic_subscribe)
    client.on_message = on_message

```

```
time.sleep(0)

def run():
    client = connect_mqtt()
    client.loop_start()
    publishSubscribe(client)

if __name__ == '__main__':
    run()
```