MODUL 4

Perancangan Node Device menggunakan Raspberry Pi 3 (Subscriber)

# Percobaan 1

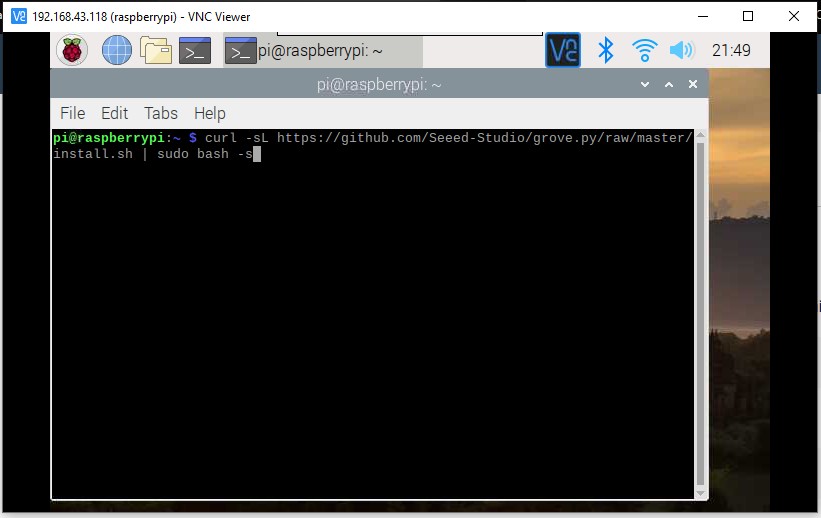
**Programming Raspberry Pi**

## Pada jendela raspberry pi akan tampil seperti berikut ini

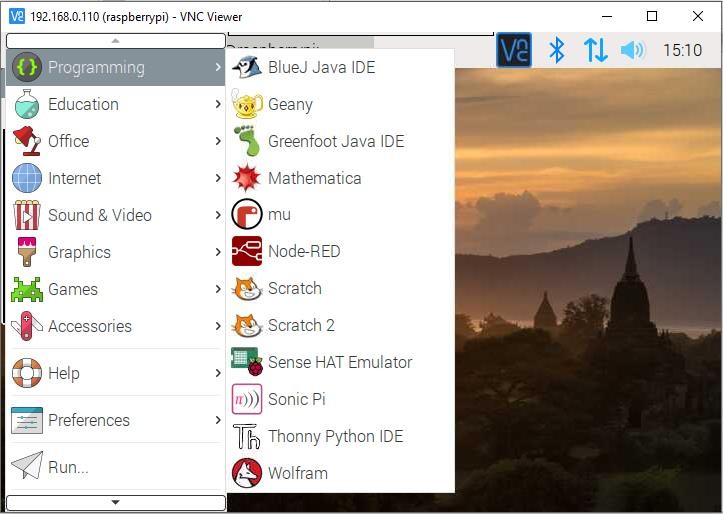
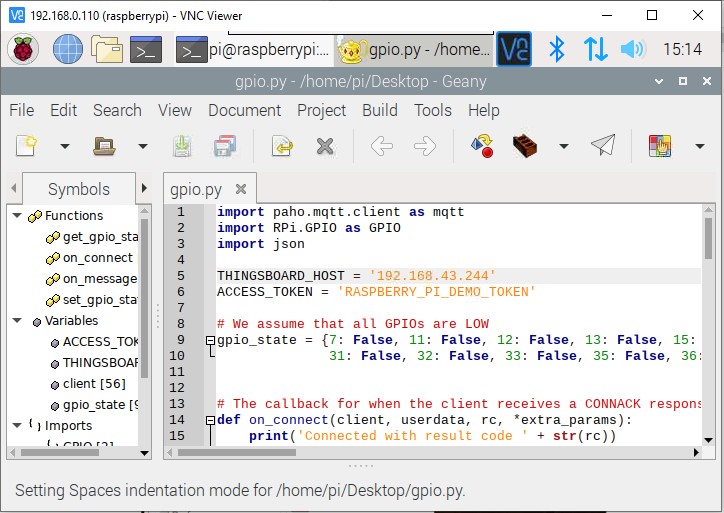
1. Kemudian buka terminal. Kemudian ketik dan install beberapa list berikut

**curl -sL https://github.com/Seeed-Studio/grove.py/raw/master/install.sh | sudo bash -s pip3 install tb-mqtt-client**

**git clone https://github.com/Seeed-Studio/Seeed\_Python\_DHT.git sudo python3 ./Seeed\_Python\_DHT/setup.py install**

\*Proses install membutuhkan waktu 15 menit tergantung koneksi internet

Apabila ditanyakan password ketikkan : **raspberry**

1. Buka programing IDE geany untuk memasukkan program
2. Maka akan muncul halaman pemrograman berikut ini

Masukkan program berikut ini, dan save dengan nama file **\*\*\*.py**

import logging import time

from tb\_device\_mqtt import TBDeviceMqttClient, TBPublishInfo

from grove.grove\_mini\_pir\_motion\_sensor import GroveMiniPIRMotionSensor from grove.grove\_ultrasonic\_ranger import GroveUltrasonicRanger

from Seeed\_Python\_DHT.seeed\_dht import DHT

from grove.grove\_moisture\_sensor import GroveMoistureSensor from grove.button import Button

from grove.grove\_ryb\_led\_button import GroveLedButton

from grove.grove\_light\_sensor\_v1\_2 import GroveLightSensor from grove.grove\_servo import GroveServo

# Configuration of logger, in this case it will send messages to console logging.basicConfig(level=logging.INFO,

format='%(asctime)s - %(levelname)s - %(module)s -

%(lineno)d - %(message)s',

datefmt='%Y-%m-%d %H:%M:%S') log = logging.getLogger( name )

thingsboard\_server = 'THINGSBOARD\_HOST' access\_token = 'ACCESS\_TOKEN'

def main():

# Grove - Servo connected to PWM port servo = GroveServo(12)

servo\_angle = 90

# Grove - mini PIR motion pir\_sensor connected to port D5 pir\_sensor = GroveMiniPIRMotionSensor(5)

# Grove - Ultrasonic Ranger connected to port D16 ultrasonic\_sensor = GroveUltrasonicRanger(16)

# Grove - LED Button connected to port D18 button = GroveLedButton(18)

# Grove - Moisture Sensor connected to port A0 moisture\_sensor = GroveMoistureSensor(0)

# Grove - Light Sensor connected to port A2 light\_sensor = GroveLightSensor(2) light\_state = False

# Grove - Temperature&Humidity Sensor connected to port D22 dht\_sensor = DHT('11', 22)

# Callback for server RPC requests (Used for control servo and led blink) def on\_server\_side\_rpc\_request(request\_id, request\_body):

log.info('received rpc: {}, {}'.format(request\_id, request\_body)) if request\_body['method'] == 'getLedState':

client.send\_rpc\_reply(request\_id, light\_state) elif request\_body['method'] == 'setLedState':

light\_state = request\_body['params'] button.led.light(light\_state)

elif request\_body['method'] == 'setServoAngle': servo\_angle = float(request\_body['params']) servo.setAngle(servo\_angle)

elif request\_body['method'] == 'getServoAngle': client.send\_rpc\_reply(request\_id, servo\_angle)

# Connecting to ThingsBoard

client = TBDeviceMqttClient(thingsboard\_server, access\_token) client.set\_server\_side\_rpc\_request\_handler(on\_server\_side\_rpc\_request) client.connect()

# Callback on detect the motion from motion sensor def on\_detect():

log.info('motion detected') telemetry = {"motion": True} client.send\_telemetry(telemetry) time.sleep(5)

# Deactivating the motion in Dashboard client.send\_telemetry({"motion": False}) log.info("Motion alert deactivated")

# Callback from button if it was pressed or unpressed def on\_event(index, event, tm):

if button.\_GroveLedButton btn.is\_pressed(): log.debug('button: single click') telemetry = {"button\_press": True} client.send\_telemetry(telemetry) log.info("Pressed")

else:

log.debug('button: single click') telemetry = {"button\_press": False} client.send\_telemetry(telemetry) log.info("Unpressed")

if event & Button.EV\_SINGLE\_CLICK: button.led.light(True)

elif event & Button.EV\_DOUBLE\_CLICK: button.led.blink()

elif event & Button.EV\_LONG\_PRESS: button.led.light(False)

# Adding the callback to the motion sensor pir\_sensor.on\_detect = on\_detect

# Adding the callback to the button button.on\_event = on\_event

try:

while True:

distance = ultrasonic\_sensor.get\_distance() log.debug('distance: {} cm'.format(distance))

humidity, temperature = dht\_sensor.read() log.debug('temperature: {}C, humidity: {}%'.format(temperature,

humidity))

moisture = moisture\_sensor.moisture log.debug('moisture: {}'.format(moisture))

log.debug('light: {}'.format(light\_sensor.light))

# Formatting the data for sending to ThingsBoard telemetry = {'distance': distance,

'temperature': temperature, 'humidity': humidity, 'moisture': moisture, 'light': light\_sensor.light}

# Sending the data client.send\_telemetry(telemetry).get()

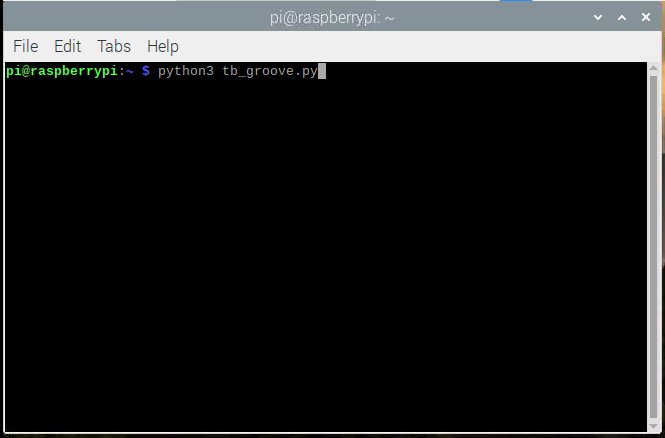
time.sleep(.1) except Exception as e:

raise e finally:

client.disconnect()

if name == ' main ': main()

## Karene menggunakan python 3, untuk menjalankan program harus melewati terminal.

Syntaxnya untuk menjalankan program adalah **python3 nama\_file.py.** Kemudian **Enter**

# Percobaan 2

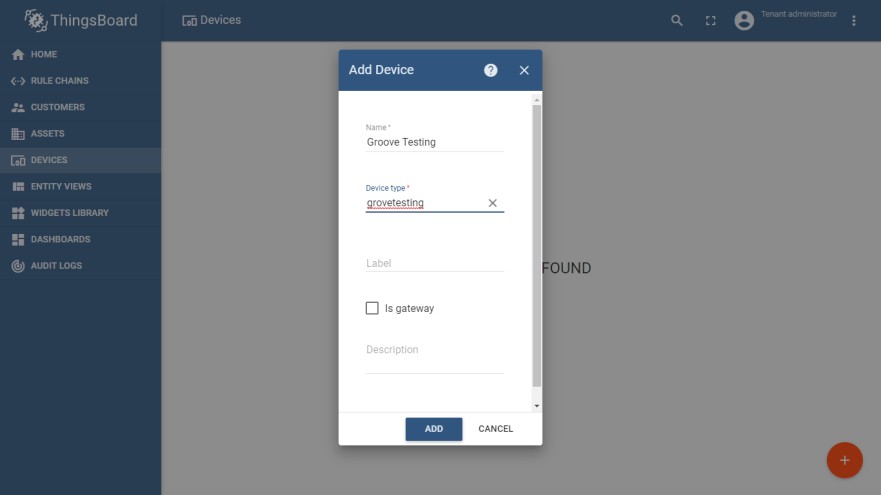
**Halaman Dashboard**

## Buka dan buat akun di Alamat IP yang ada di papan tulis

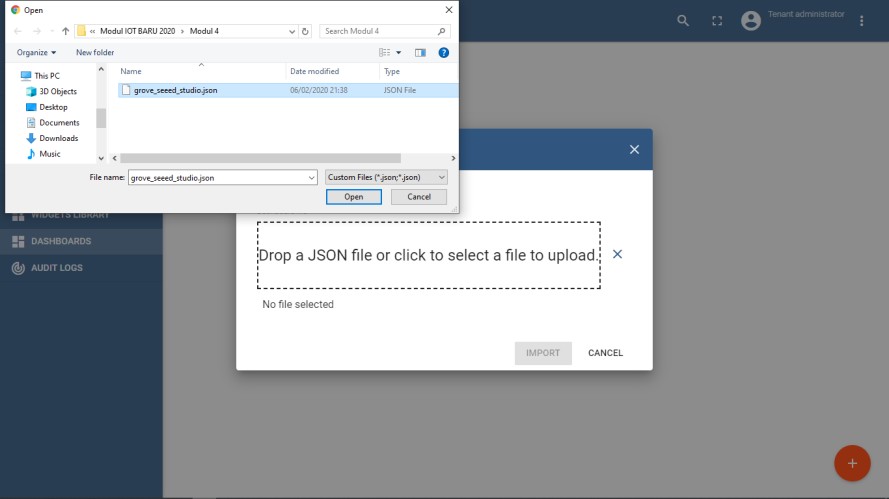
1. Masuk dengan user yang sudah tersedia sebagai berikut: Email : [praktikum\_iot\_1@thingsboard.com](mailto:praktikum_iot_1@thingsboard.com)

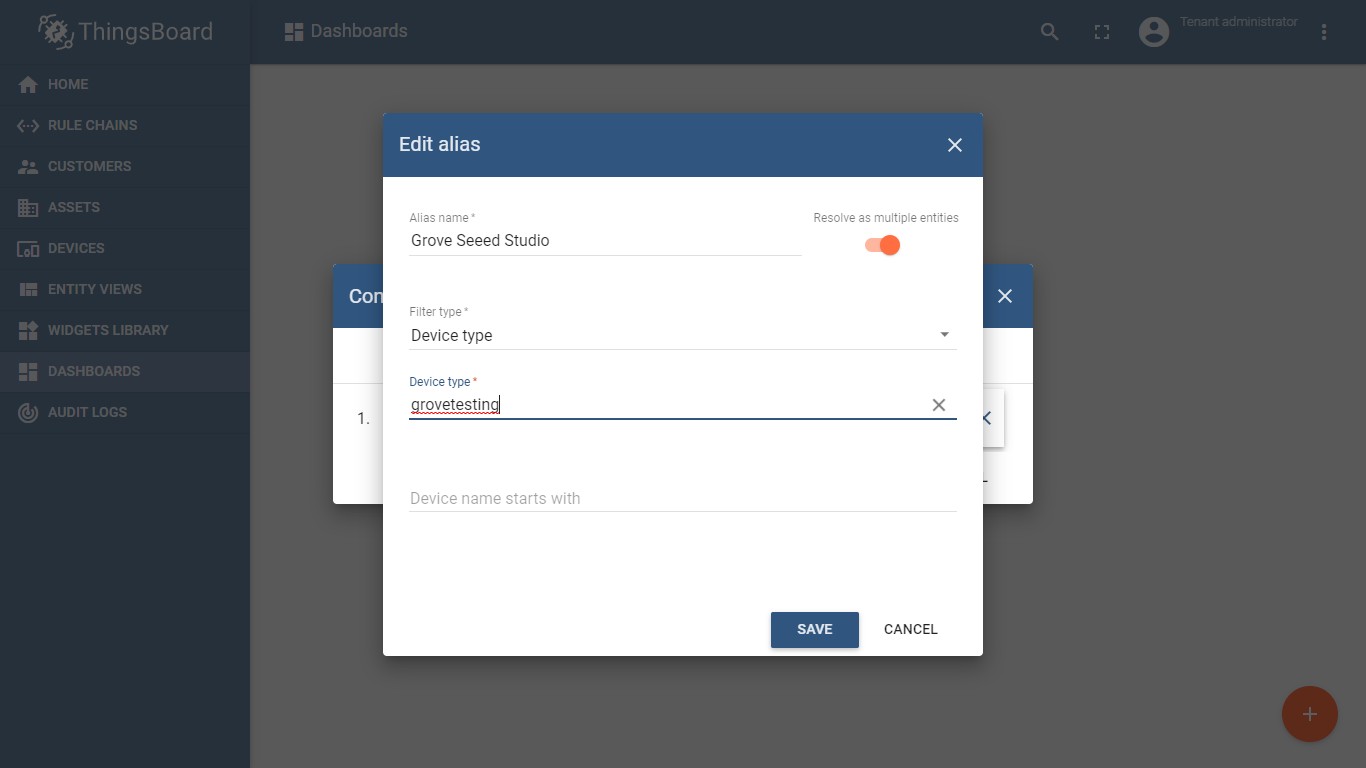
Pass : praktikumiot

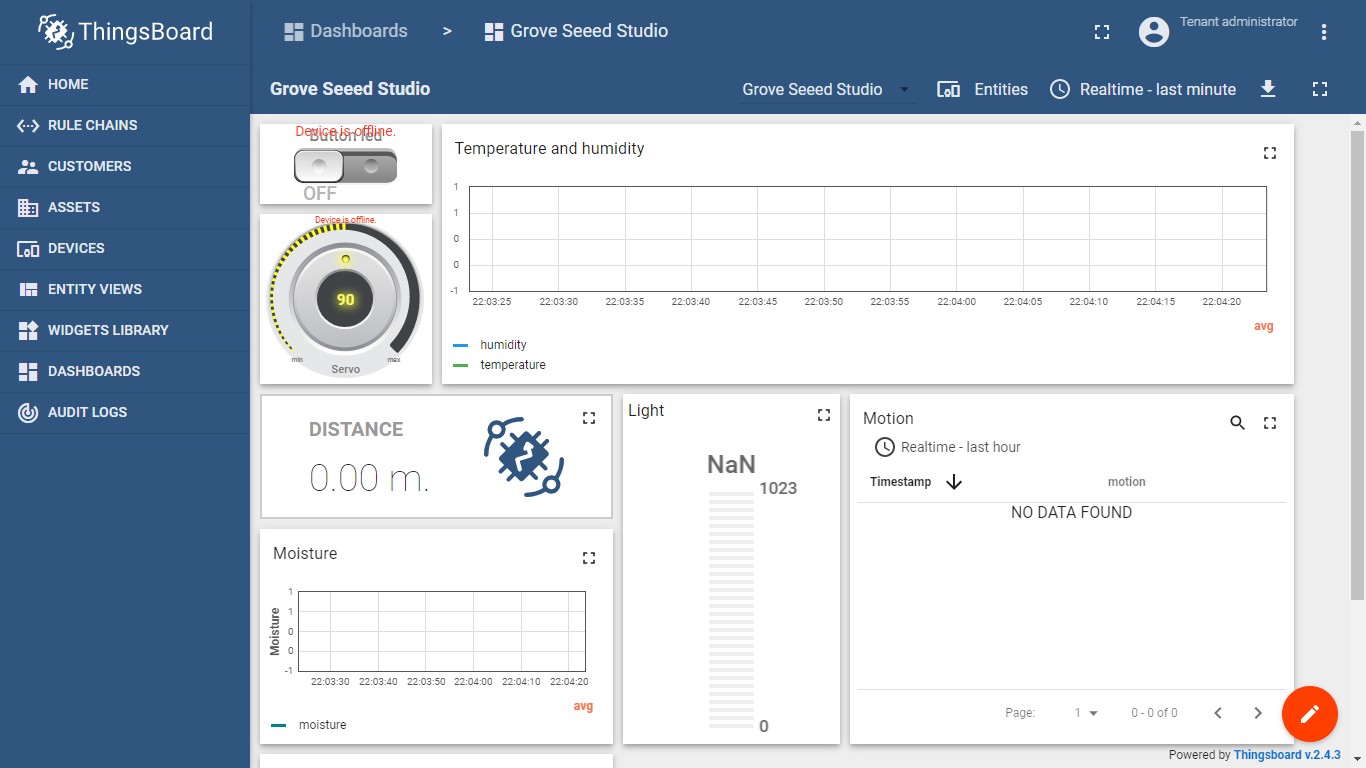
\*untuk email yang digunakan sesuai dengan kelompok praktikum, jika kelompok 2 maka emailnya adalah [praktikum\_iot\_2@thingsboard.com,](mailto:praktikum_iot_2@thingsboard.com) dan seterusnya. Untuk password adalah sama.

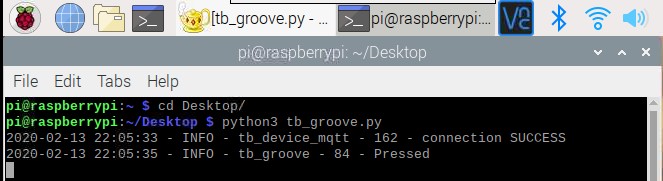
1. Buatlah device dengan nama berikut:
2. Kemudian import dashboard yang tersedia di url berikut:

<https://thingsboard.io/docs/samples/raspberry/resources/grove_seeed_studio.json>



1. Buatlah alias terlebih dahulu
2. Maka nanti akan muncul dashboard seperti berikut



1. Kemudian buka kembali program python pada terminal. Dan jika berhasil akan muncul tampilan seperti berikut ini

Tugas

1. Masih dengan halaman dashboard yang sama, buatlah rangkaian dengan breadboard untuk menjalankan servo dan menyalakan LED.

|  |  |
| --- | --- |
| Servo | Raspberry Pi |
| Data | GPIO 12 / PIN32 |
| VCC | 5V |
| GND | GND |

|  |  |
| --- | --- |
| LED | Raspberry Pi |
| Anode | GPIO 18 / PIN 12 |
| Katode | GND |

1. Pastikan pada Dashboard panel dapat ditekan dan dapat dikontrol.

