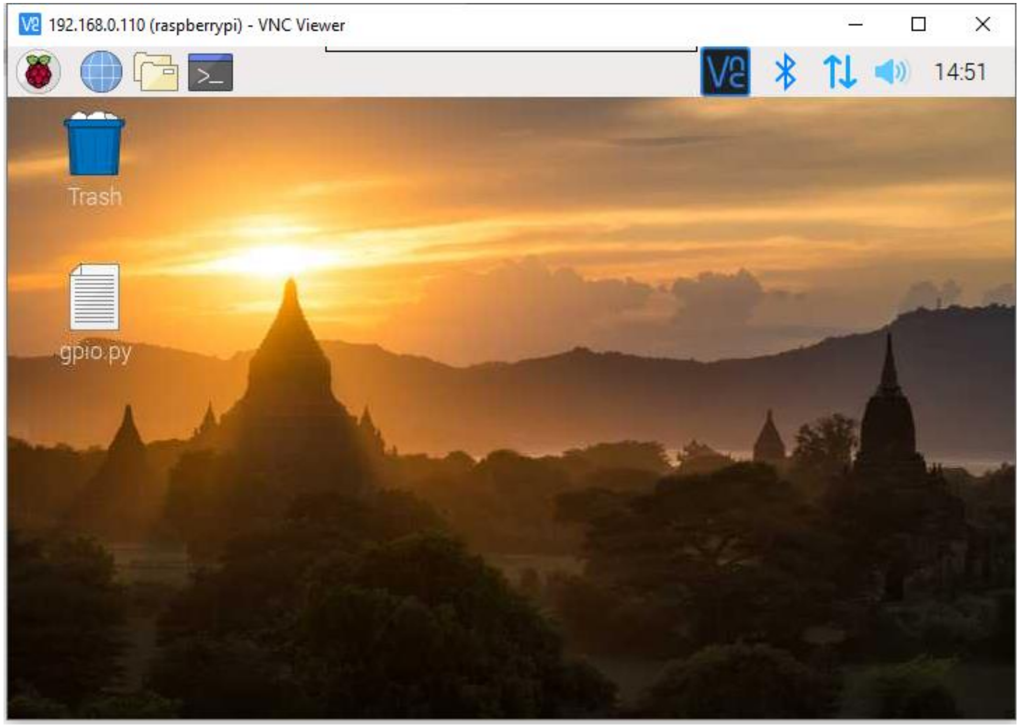
MODUL 4

Perancangan Node Device menggunakan Raspberry Pi 3 (Subscriber)

Percobaan 1

Programming Raspberry Pi

1. Pada jendela raspberry pi akan tampil seperti berikut ini



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

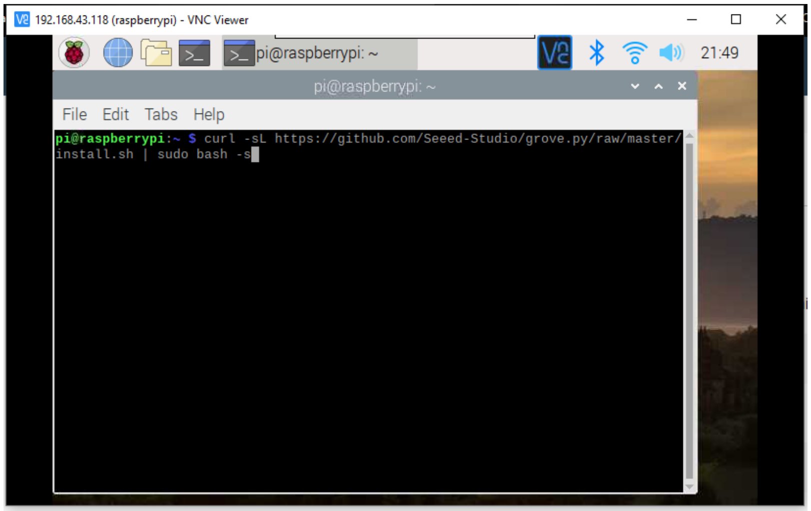
   curl -sL https://github.com/Seeed-Studio/grove.py/raw/master/install.sh | sudo bash -s

   pip3 install tb-mqtt-client

   git clone https://github.com/Seeed-Studio/Seeed_Python_DHT.git

   sudo python3 ./Seeed_Python_DHT/setup.py install

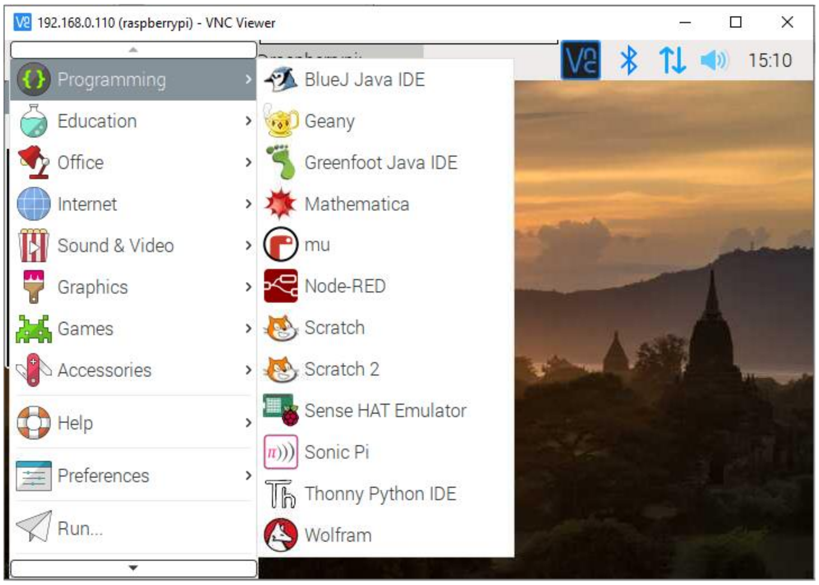
   *Proses install membutuhkan waktu 15 menit tergantung koneksi internet



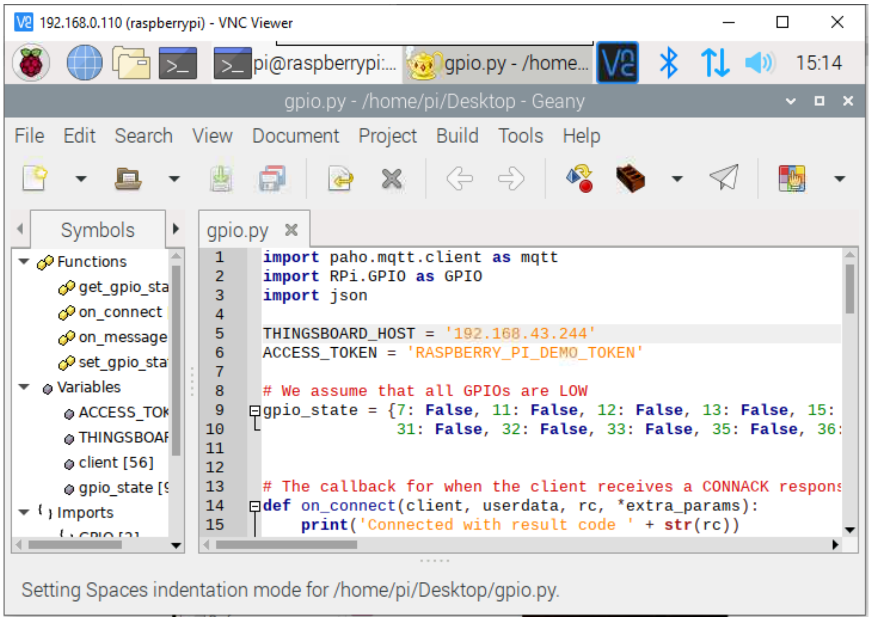
```
pi@raspberrypi:~ $ curl -sL https://github.com/Seeed-Studio/grove.py/raw/master/
install.sh | sudo bash -s
```

   Apabila ditanyakan password ketikkan : **raspberry**

3. Buka programing IDE geany untuk memasukkan program



4. Maka akan muncul halaman pemrograman berikut ini



```
import paho.mqtt.client as mqtt
import RPi.GPIO as GPIO
import json

THINGSBOARD_HOST = '192.168.43.244'
ACCESS_TOKEN = 'RASPBERRY_PI_DEMO_TOKEN'

# We assume that all GPIOs are LOW
gpio_state = {7: False, 11: False, 12: False, 13: False, 15:
              31: False, 32: False, 33: False, 35: False, 36:



# The callback for when the client receives a CONNACK respons
def on_connect(client, userdata, rc, *extra_params):
    print('Connected with result code ' + str(rc))
```

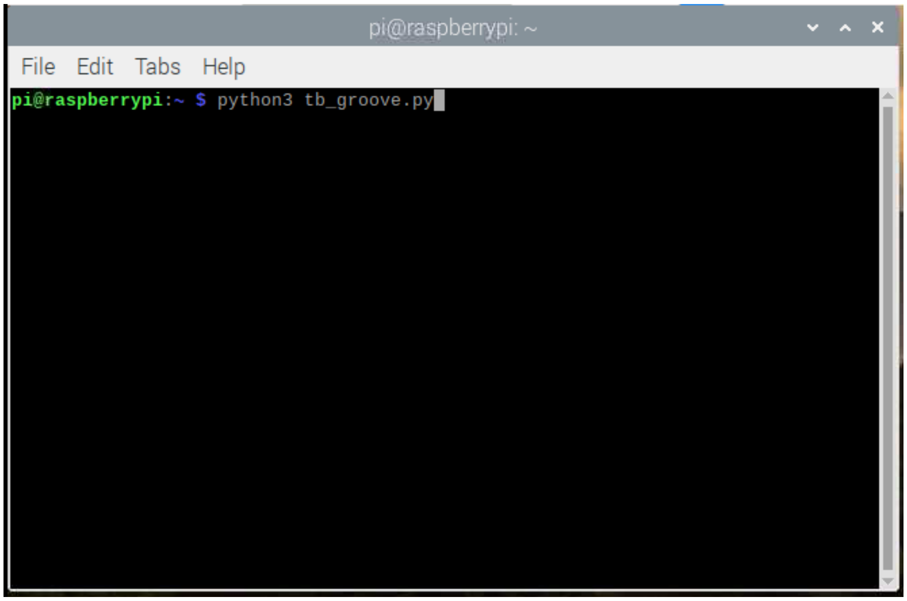
Setting Spaces indentation mode for /home/pi/Desktop/gpio.py.

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()
```

5. Karene menggunakan python 3, untuk menjalankan program harus melewati terminal. Syntaxnya untuk menjalankan program adalah **python3 nama_file.py.** Kemudian **Enter**



```
pi@raspberrypi:~ $ python3 tb_groove.py
```

Percobaan 2

Halaman Dashboard

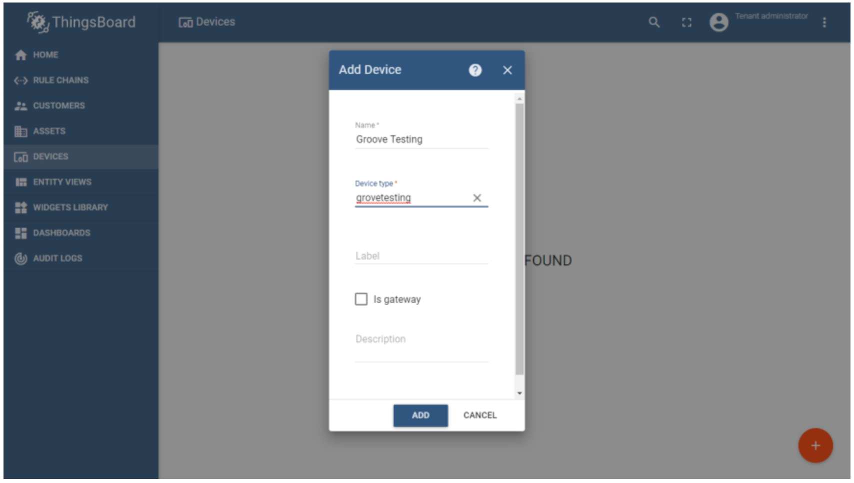
1. Buka dan buat akun di Alamat IP yang ada di papan tulis
2. Masuk dengan user yang sudah tersedia sebagai berikut:

   Email : 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, dan seterusnya. Untuk password adalah sama.

3. Buatlah device dengan nama berikut:



Add Device

Name * : Groove Testing

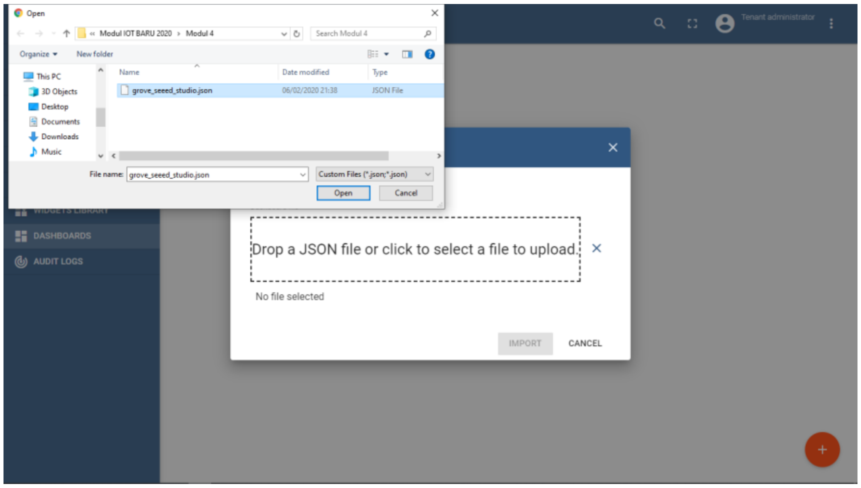
Device type * : grovetesting

Label

□ Is gateway

Description

4. Kemudian import dashboard yang tersedia di url berikut: https://thingsboard.io/docs/samples/raspberry/resources/grove_seeed_studio.json



Name	Date modified	Type
grove_seeed_studio.json	06/02/2020 21:38	JSON File

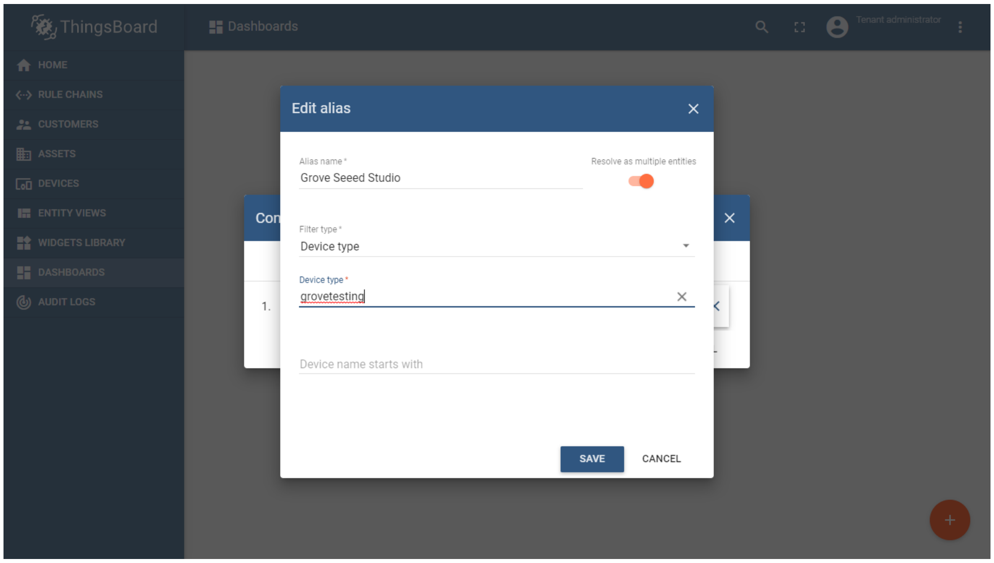
File name: grove_seeed_studio.json

Custom Files (*.json;*.json)

Drop a JSON file or click to select a file to upload.

No file selected

5. Buatlah alias terlebih dahulu



Edit alias

Alias name * Grove Seeed Studio

Resolve as multiple entities

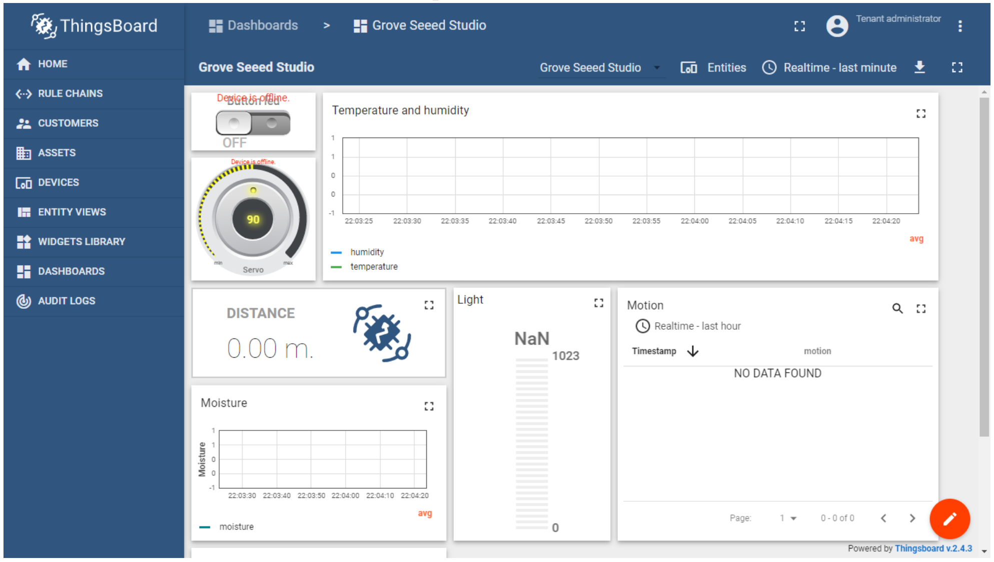
Filter type * Device type

Device type * grovetesting

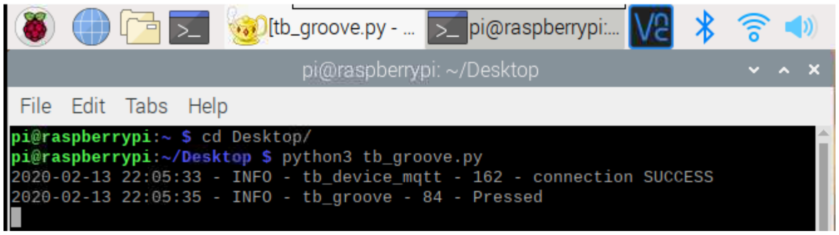
Device name starts with

SAVE CANCEL

6. Maka nanti akan muncul dashboard seperti berikut



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



```
pi@raspberrypi:~ $ cd Desktop/
pi@raspberrypi:~/Desktop $ python3 tb_groove.py
2020-02-13 22:05:33 - INFO - tb_device_mqtt - 162 - connection SUCCESS
2020-02-13 22:05:35 - INFO - tb_groove - 84 - Pressed
```

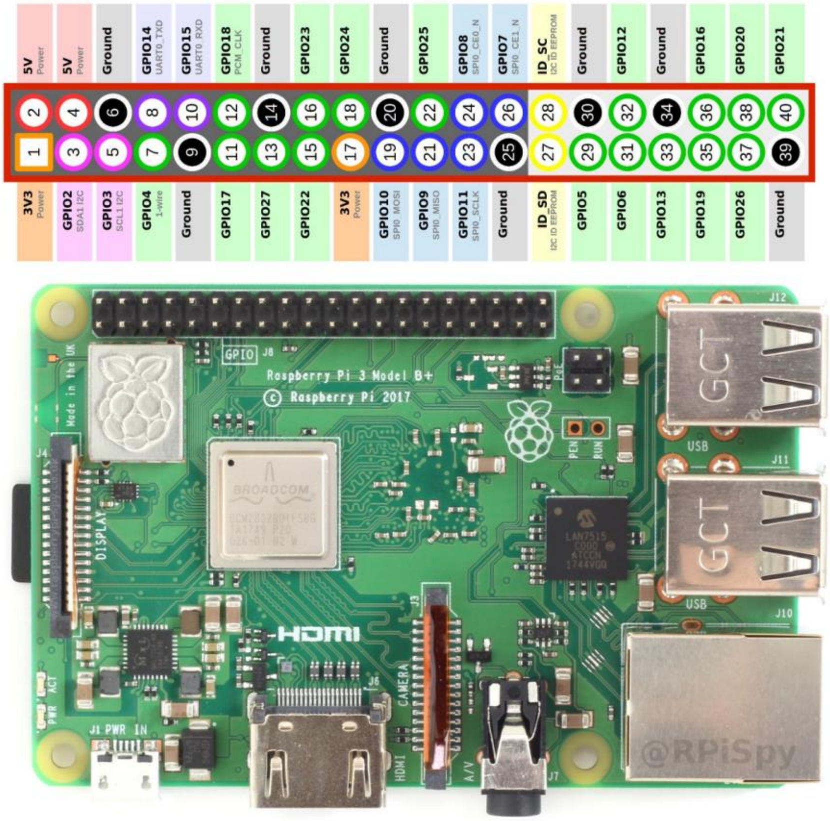
Tugas

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

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


5V Power
5V Power
Ground
GPIO14 UART0_TXD
GPIO15 UART0_RXD
GPIO18 PCM_CLK
Ground
GPIO23
GPIO24
Ground
GPIO25
GPIO8 SPI0_CE0_N
GPIO7 SPI0_CE1_N
ID_SC I2C ID EEPROM
Ground
GPIO12
Ground
GPIO16
GPIO20
GPIO21
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39
3V3 Power
GPIO2 SDA1 I2C
GPIO3 SCL1 I2C
GPIO4 1-wire
Ground
GPIO17
GPIO27
GPIO22
3V3 Power
GPIO10 SPI0_MOSI
GPIO9 SPI0_MISO
GPIO11 SPI0_SCLK
Ground
ID_SD I2C ID EEPROM
GPIO5
GPIO6
GPIO13
GPIO19
GPIO26
Ground
GPIO J8
Raspberry Pi 3 Model B+
© Raspberry Pi 2017
DISPLAY
HDMI
CAMERA
J1 PWR IN
USB
J12
J11
J10
@RPiSpy