

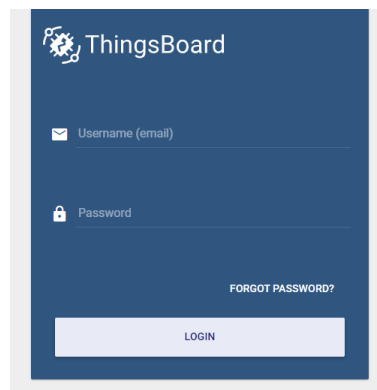
MODUL I

Pengenalan Open IoT Platform Thingsboard dan Protokol MQTT (Subscriber)

Percobaan 1

Membuat *device* pada Thingsboard

1. Buka dan buat akun di [Alamat IP yang ada di papan tulis](#)
2. Maka akan tertampil halaman login seperti berikut :



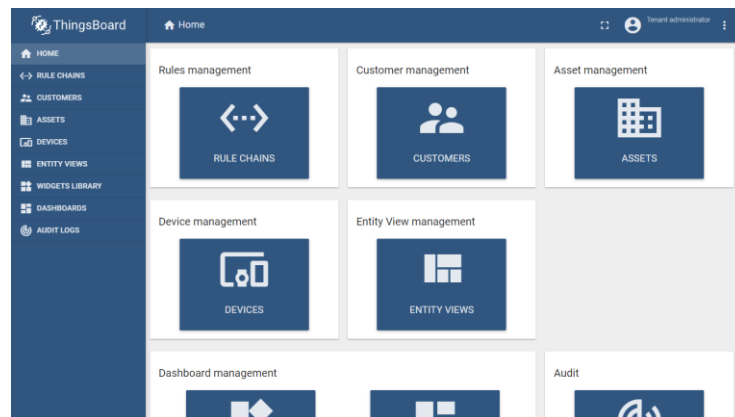
3. Masuk dengan user yang sudah tersedia sebagai berikut:

Email : praktikum_iot_1@thingsboard.com

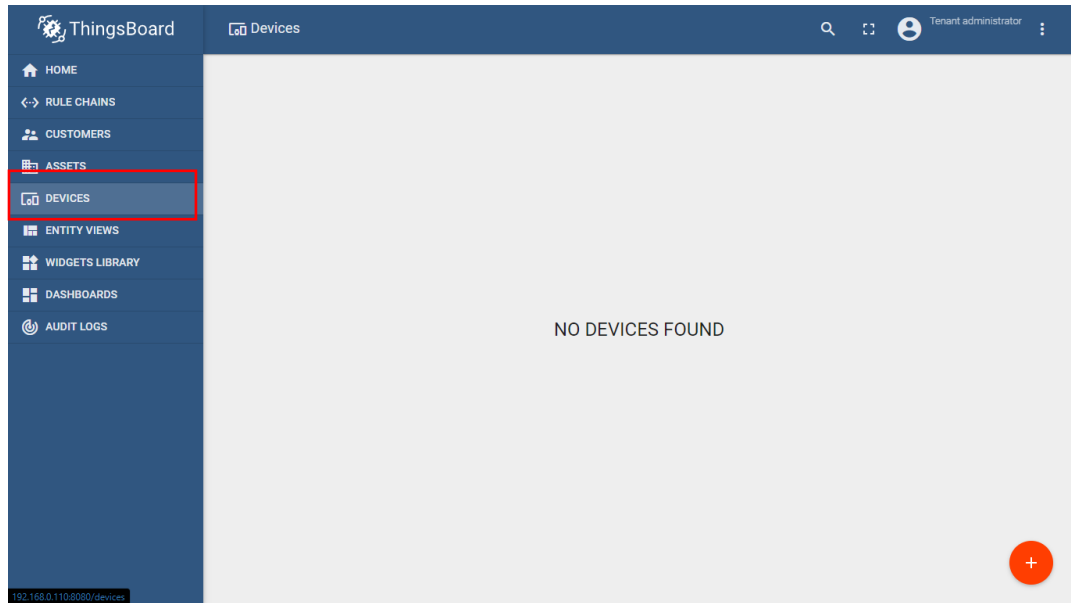
Pass : praktikumiot

*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.

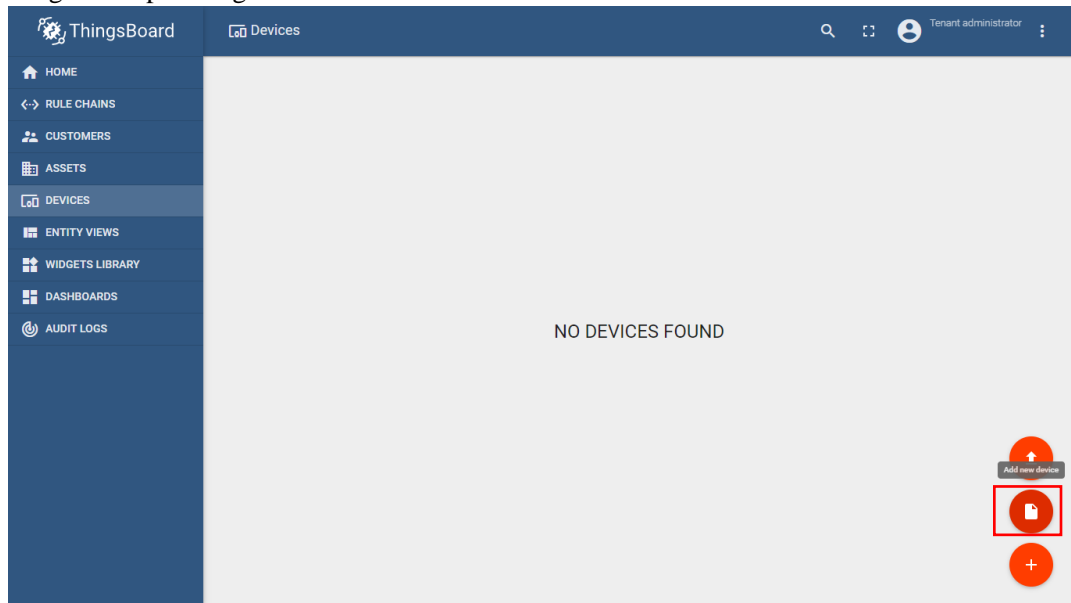
4. Maka akan muncul jendela seperti berikut ini



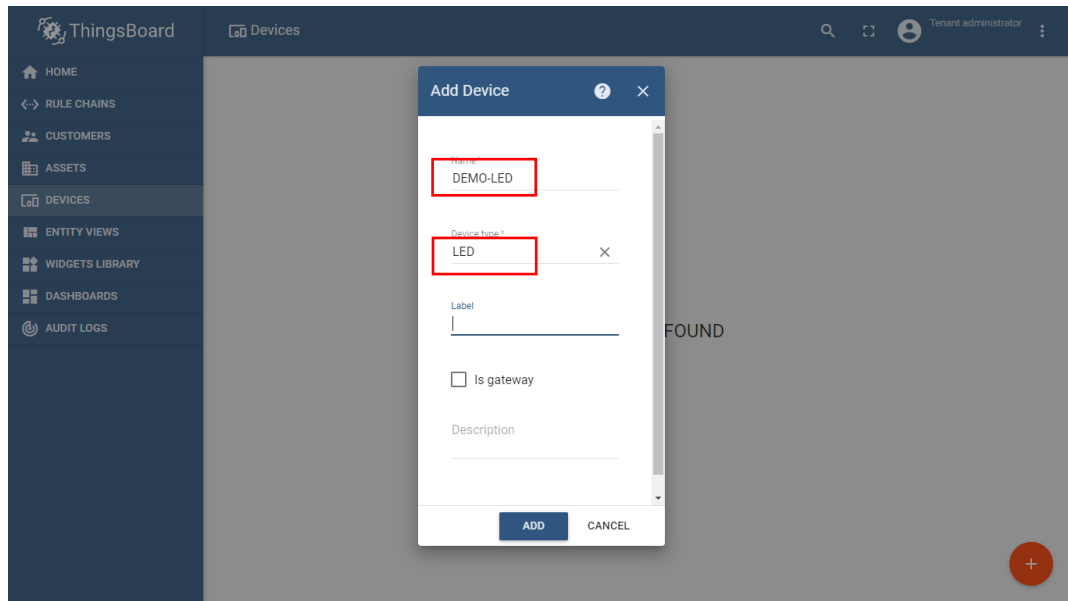
5. Kemudian pilih pada menu device.



6. Buatlah device baru dengan menekan tombol berwarna jingga di pojok kanan bawah, dengan klik pada bagian add new device.



7. Maka akan muncul tampilan seperti berikut ini :

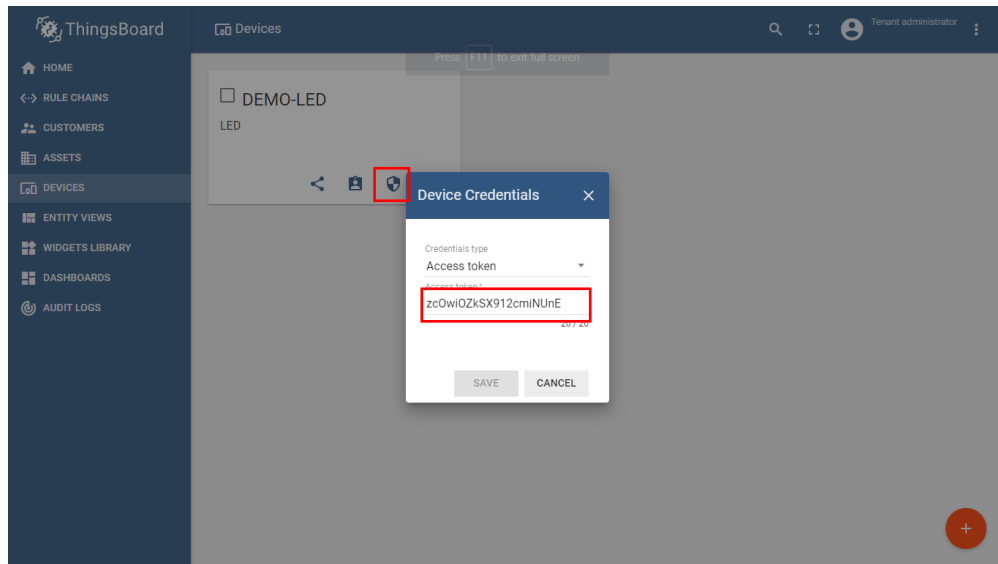


Kemudian isikan **Name** dengan nama **DEMO-LED**, dan **Device type** dengan **LED**. Jika sudah diisi kemudian klik add untuk menambahkan device.

8. Kemudian akan muncul sebuah sub menu dengan nama DEMO-LED



9. Kemudian pilih pada bagian Device Credential, simpan atau gunakan **access token** untuk kepentingan nantinya.

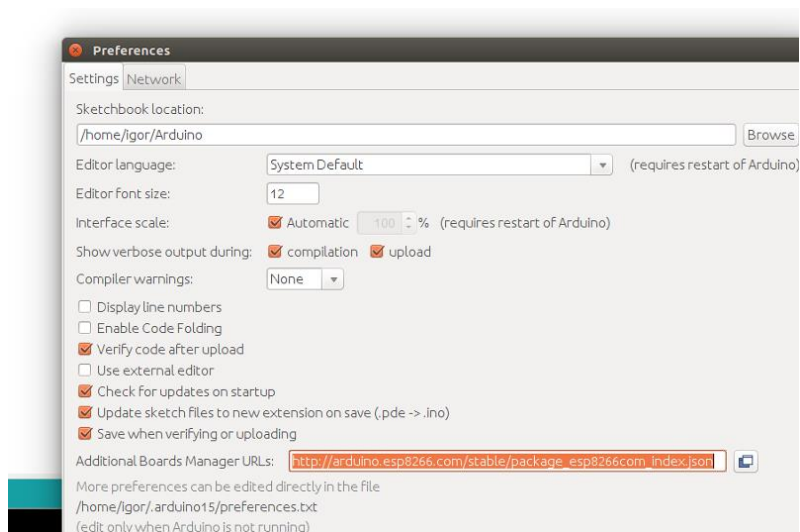


Percobaan 2

Pemrograman NodeMCU

NodeMCU dan Arduino IDE *setup*.

1. Buka Arduino IDE, kemudian buka bagian *file* dan buka bagian *preferences*.

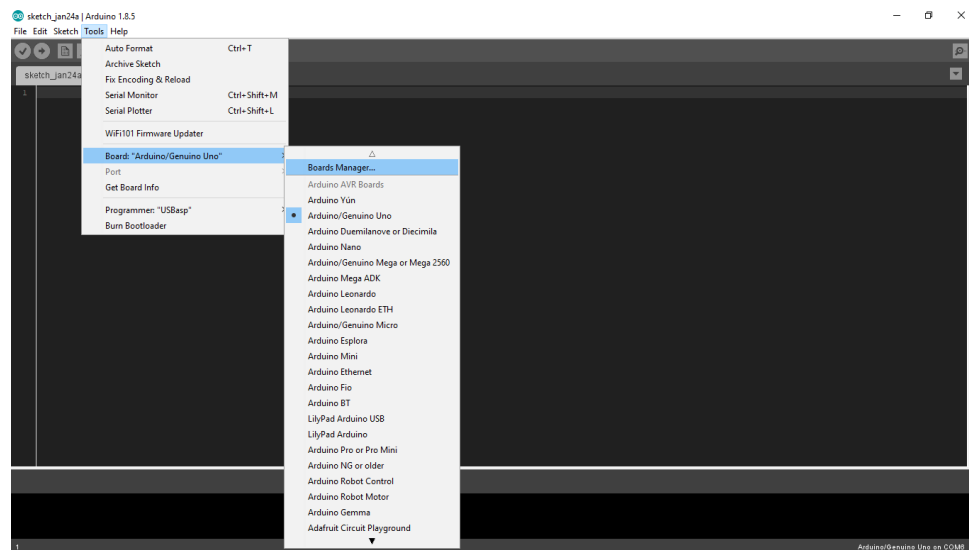


Paste URL berikut pada bagian **“Additional board managers URL”**:

http://arduino.esp8266.com/stable/package_esp8266com_index.json

Kemudian tekan tombol **OK**.

2. Kemudian buka pada bagian board manager

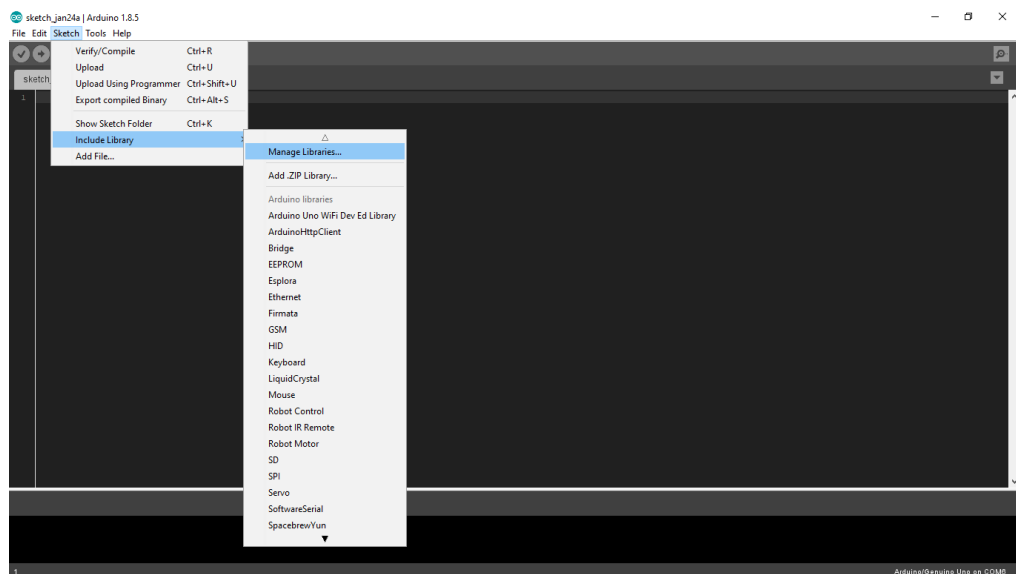


3. Kemudian install board ESP8266 seperti pada gambar berikut :

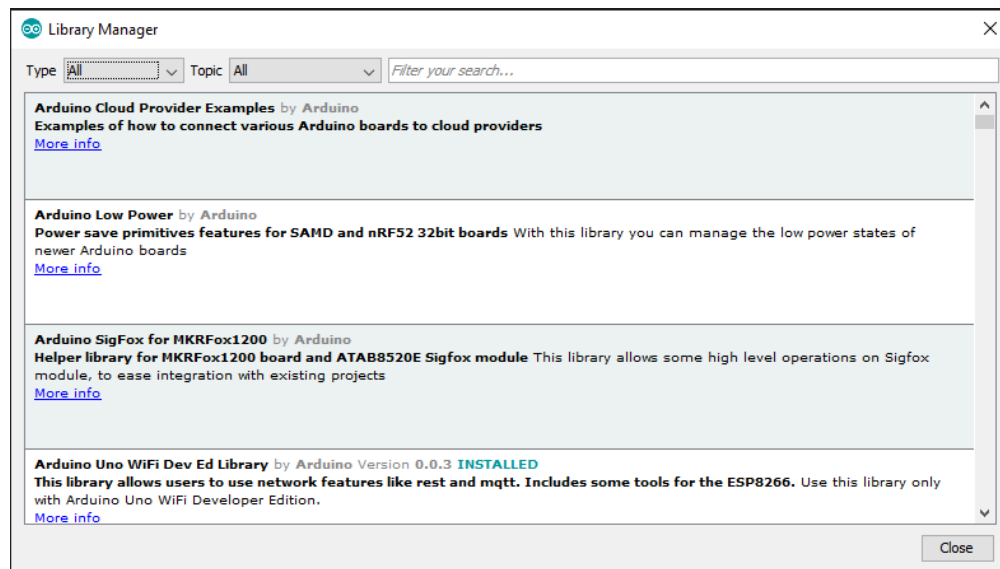


Proses install membutuhkan waktu sesuai dengan kecepatan koneksi internet.

4. Selanjutnya adalah menginstall library yang ada pada Arduino IDE
Open Arduino IDE and go to Sketch -> Include Library -> Manage Libraries.



5. Akan muncul halaman seperti berikut ini:



Install Library Berikut

- [PubSubClient by Nick O'Leary](#) Version 2.6
- [ArduinoJson by Benoit Blanchon](#). Version 5.8.0

6. Apabila sudah terinstall kembali ke halaman utama Arduino

```
#include <ArduinoJson.h>
#include <PubSubClient.h>
#include <ESP8266WiFi.h>

#define WIFI_AP "YOUR_WIFI_AP"
#define WIFI_PASSWORD "YOUR_WIFI_PASSWORD"

#define TOKEN "ESP8266_DEMO_TOKEN" // ganti bagian ini dengan token device anda

#define GPIO0 1
#define GPIO2 2

#define GPIO0_PIN 3
#define GPIO2_PIN 5

char thingsboardServer[] = "demo.thingsboard.io";

WiFiClient wifiClient;

PubSubClient client(wifiClient);

int status = WL_IDLE_STATUS;

// We assume that all GPIOs are LOW
boolean gpioState[] = {false, false};

void setup() {
  Serial.begin(115200);
  // Set output mode for all GPIO pins
  pinMode(GPIO0, OUTPUT);
  pinMode(GPIO2, OUTPUT);
```

```

    delay(10);
    InitWiFi();
    client.setServer( thingsboardServer, 1883 );
    client.setCallback(on_message);
}

void loop() {
    if ( !client.connected() ) {
        reconnect();
    }

    client.loop();
}

// The callback for when a PUBLISH message is received from the server.
void on_message(const char* topic, byte* payload, unsigned int length) {

    Serial.println("On message");

    char json[length + 1];
    strncpy (json, (char*)payload, length);
    json[length] = '\0';

    Serial.print("Topic: ");
    Serial.println(topic);
    Serial.print("Message: ");
    Serial.println(json);

    // Decode JSON request
    StaticJsonBuffer<200> jsonBuffer;
    JsonObject& data = jsonBuffer.parseObject((char*)json);

    if (!data.success())
    {
        Serial.println("parseObject() failed");
        return;
    }

    // Check request method
    String methodName = String((const char*)data["method"]);

    if (methodName.equals("getGpioStatus")) {
        // Reply with GPIO status
        String responseTopic = String(topic);
        responseTopic.replace("request", "response");
        client.publish(responseTopic.c_str(), get_gpio_status().c_str());
    } else if (methodName.equals("setGpioStatus")) {
        // Update GPIO status and reply
        set_gpio_status(data["params"]["pin"], data["params"]["enabled"]);
        String responseTopic = String(topic);
        responseTopic.replace("request", "response");
        client.publish(responseTopic.c_str(), get_gpio_status().c_str());
        client.publish("v1/devices/me/attributes", get_gpio_status().c_str());
    }
}

String get_gpio_status() {
    // Prepare gpios JSON payload string
    StaticJsonBuffer<200> jsonBuffer;
    JsonObject& data = jsonBuffer.createObject();
    data[String(GPIO0_PIN)] = gpioState[0] ? true : false;
    data[String(GPIO2_PIN)] = gpioState[1] ? true : false;
    char payload[256];
    data.printTo(payload, sizeof(payload));
    String strPayload = String(payload);
    Serial.print("Get gpio status: ");

```



```

    Serial.println(strPayload);
    return strPayload;
}

void set_gpio_status(int pin, boolean enabled) {
    if (pin == GPIO0_PIN) {
        // Output GPIOs state
        digitalWrite(GPIO0, enabled ? HIGH : LOW);
        // Update GPIOs state
        gpioState[0] = enabled;
    } else if (pin == GPIO2_PIN) {
        // Output GPIOs state
        digitalWrite(GPIO2, enabled ? HIGH : LOW);
        // Update GPIOs state
        gpioState[1] = enabled;
    }
}

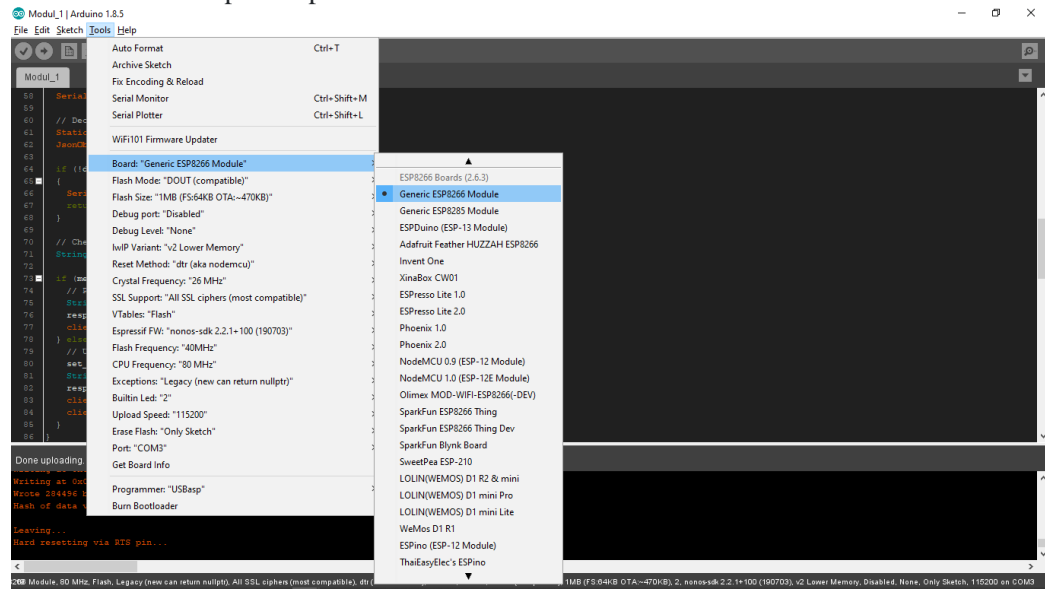
void InitWiFi() {
    Serial.println("Connecting to AP ...");
    // attempt to connect to WiFi network

    WiFi.begin(WIFI_AP, WIFI_PASSWORD);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("Connected to AP");
}

void reconnect() {
    // Loop until we're reconnected
    while (!client.connected()) {
        status = WiFi.status();
        if ( status != WL_CONNECTED) {
            WiFi.begin(WIFI_AP, WIFI_PASSWORD);
            while (WiFi.status() != WL_CONNECTED) {
                delay(500);
                Serial.print(".");
            }
            Serial.println("Connected to AP");
        }
        Serial.print("Connecting to ThingsBoard node ...");
        // Attempt to connect (clientId, username, password)
        if ( client.connect("ESP8266 Device", TOKEN, NULL) ) {
            Serial.println( "[DONE]" );
            // Subscribing to receive RPC requests
            client.subscribe("v1/devices/me/rpc/request/+");
            // Sending current GPIO status
            Serial.println("Sending current GPIO status ...");
            client.publish("v1/devices/me/attributes", get_gpio_status().c_str());
        } else {
            Serial.print( "[FAILED] [ rc = " );
            Serial.print( client.state() );
            Serial.println( " : retrying in 5 seconds]" );
            // Wait 5 seconds before retrying
            delay( 5000 );
        }
    }
}

```

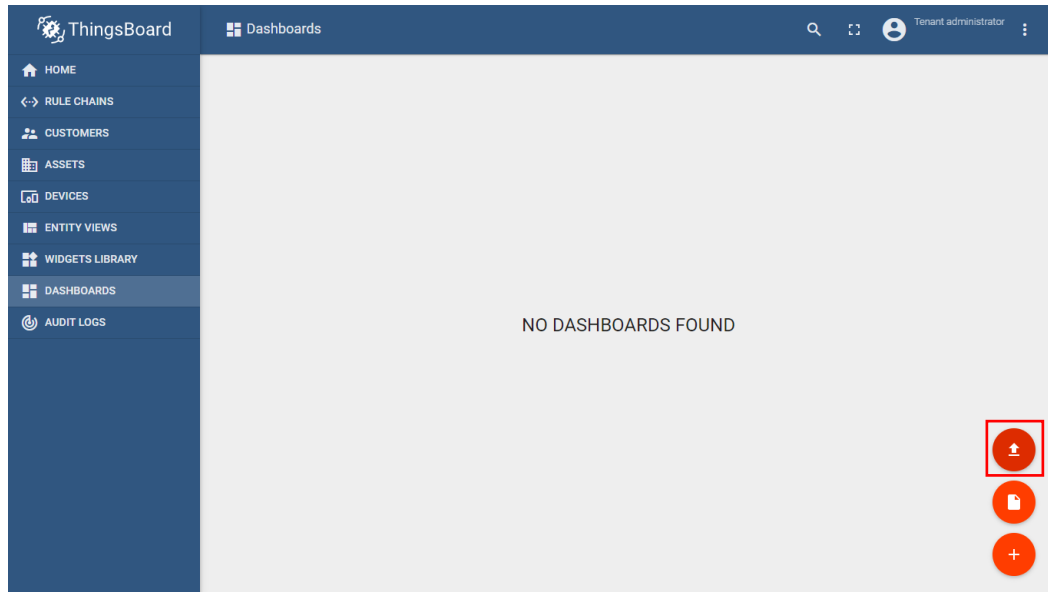
7. Gunakan Compiler seperti berikut ini



Percobaan 3

Membuat Dashboard

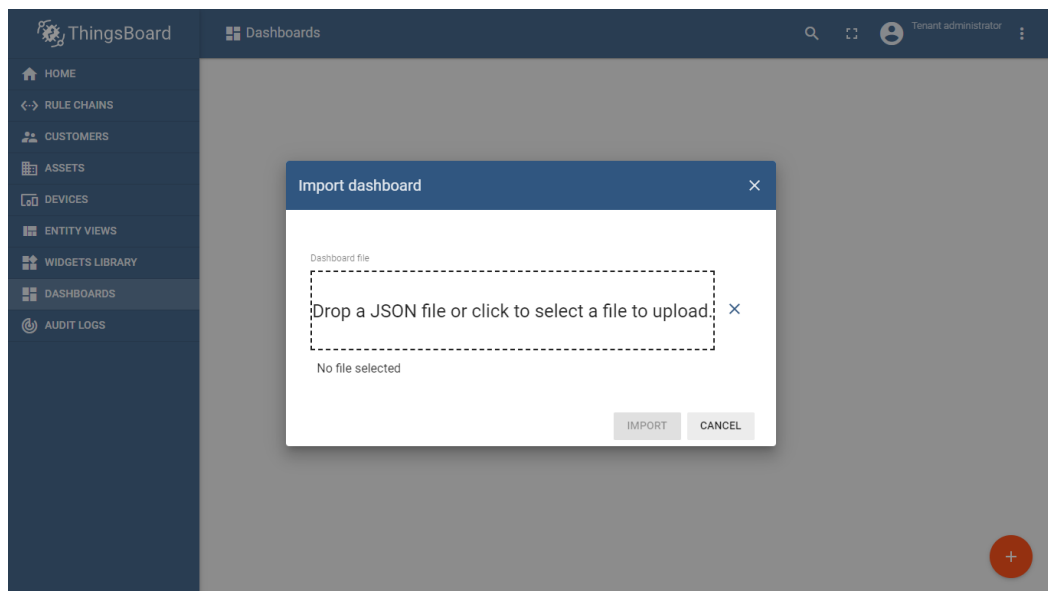
1. Buka bagian Dashboard, kemudian pilih bagian import dashboard



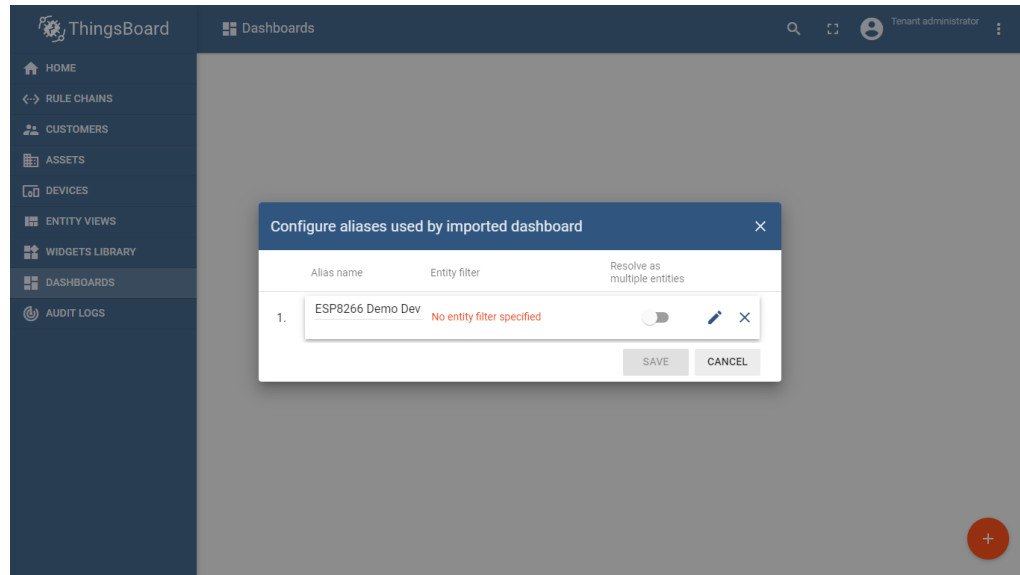
2. Upload atau drag extensi JSON dari link program berikut :

https://thingsboard.io/docs/samples/esp8266/resources/esp8266_gpio_dashboard_v2.json

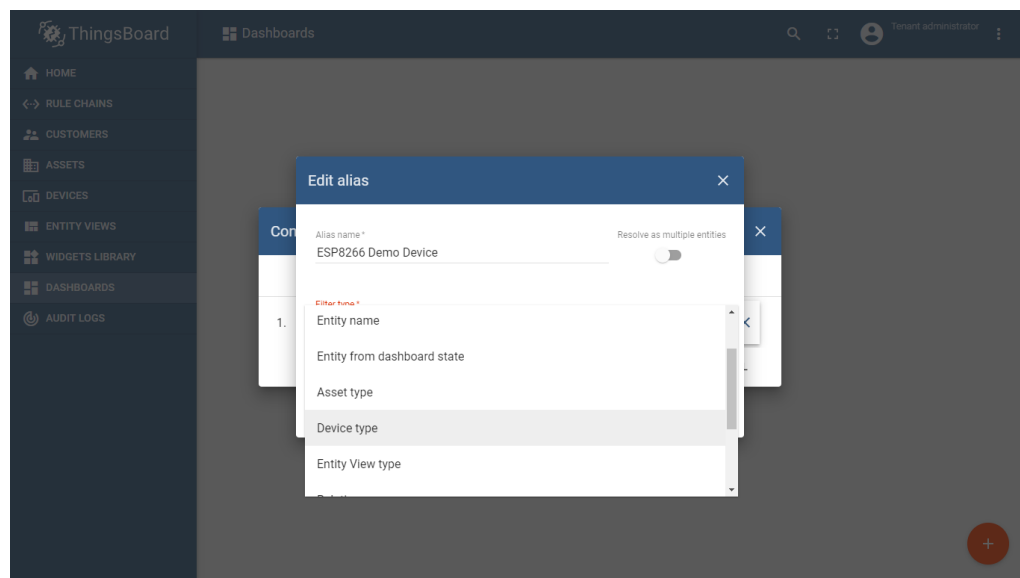
kemudian tekan **import**.



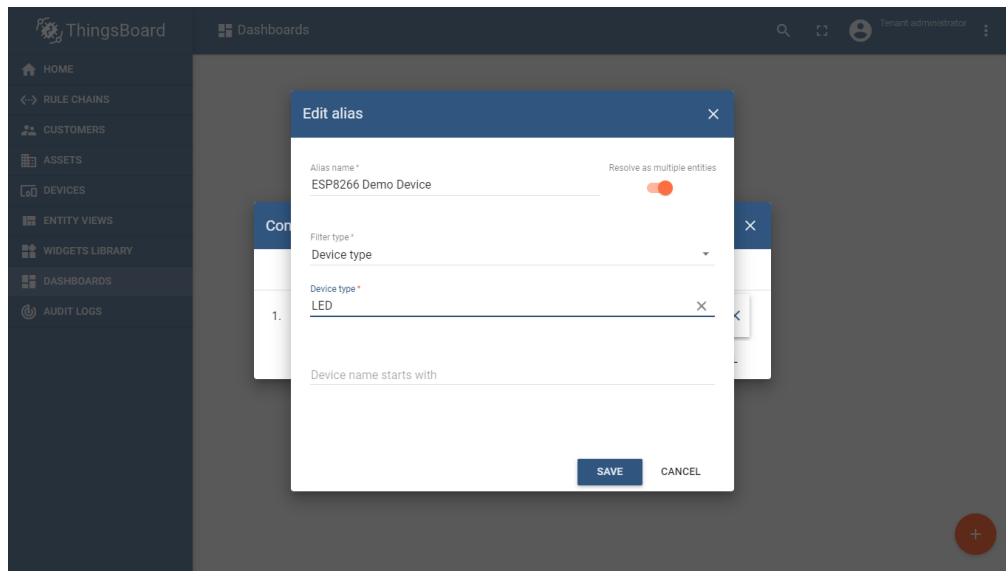
3. Apabila sudah, akan muncul jendela seperti berikut:



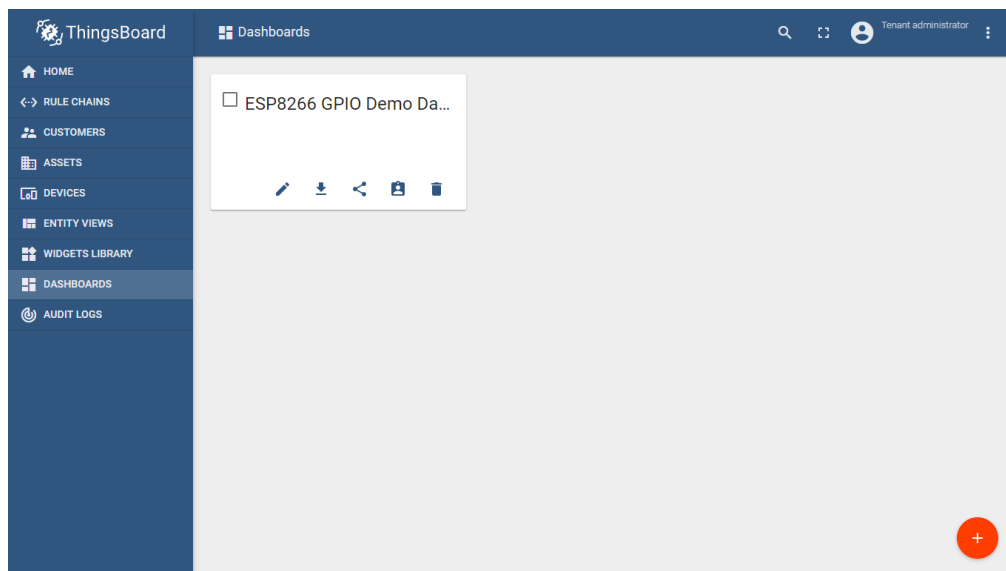
Klik bagian **edit alias**, kemudian isikan **filter type** dengan jenis **device type**

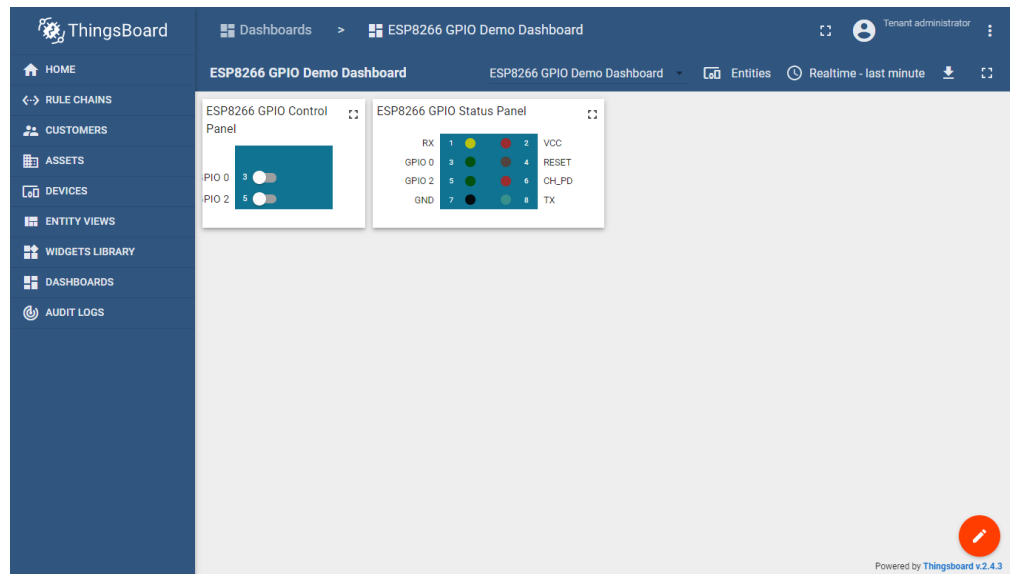


Kemudian pilih bagian **Device Type** dengan **LED** kemudian klik **save**.



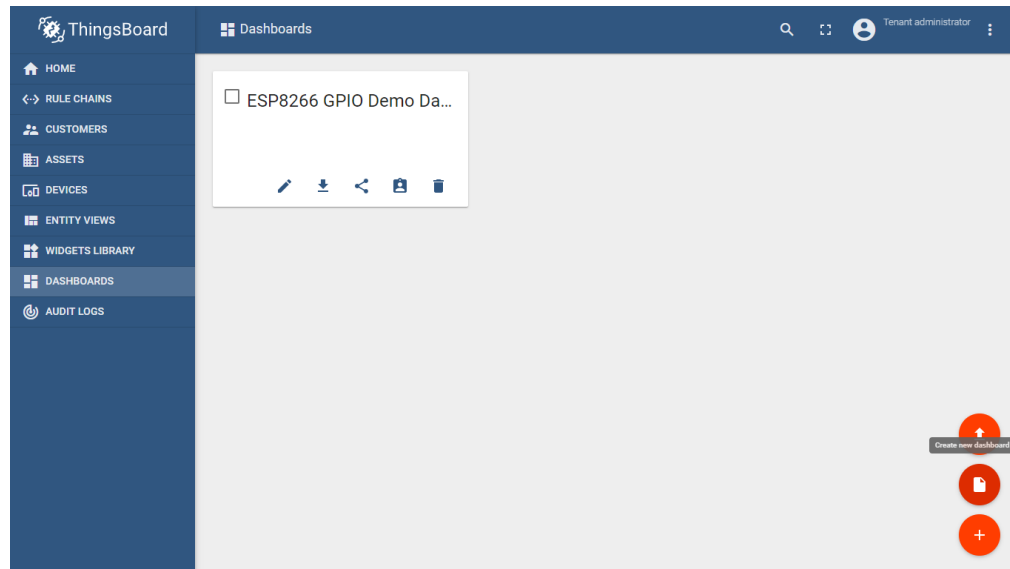
4. Maka akan muncul pilihan dashboard baru jika berhasil



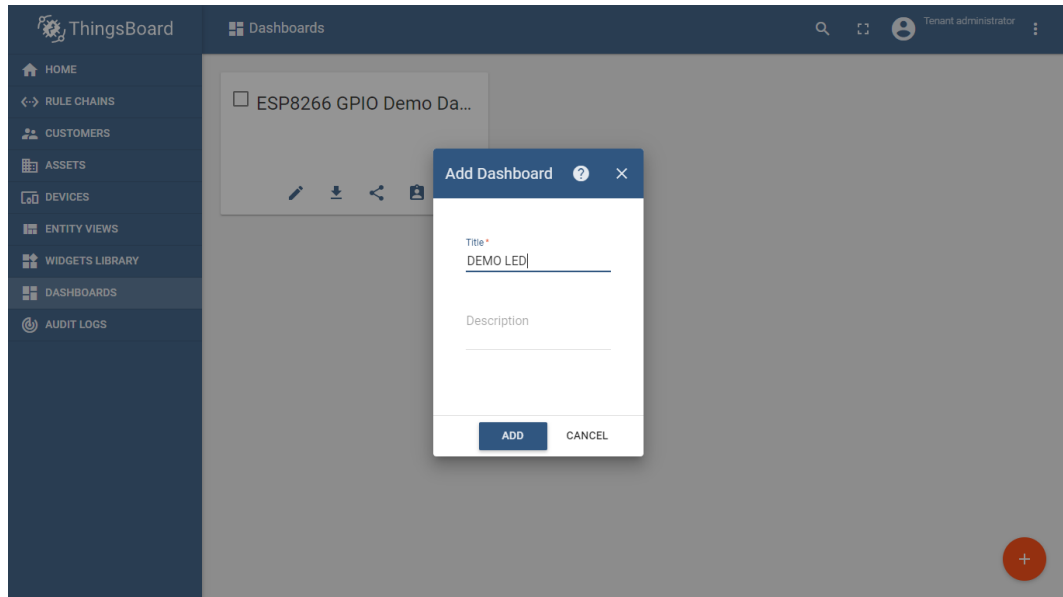


Anda sekarang sudah bisa mengatur LED built in pin 2 yang ada pada board ESP8266

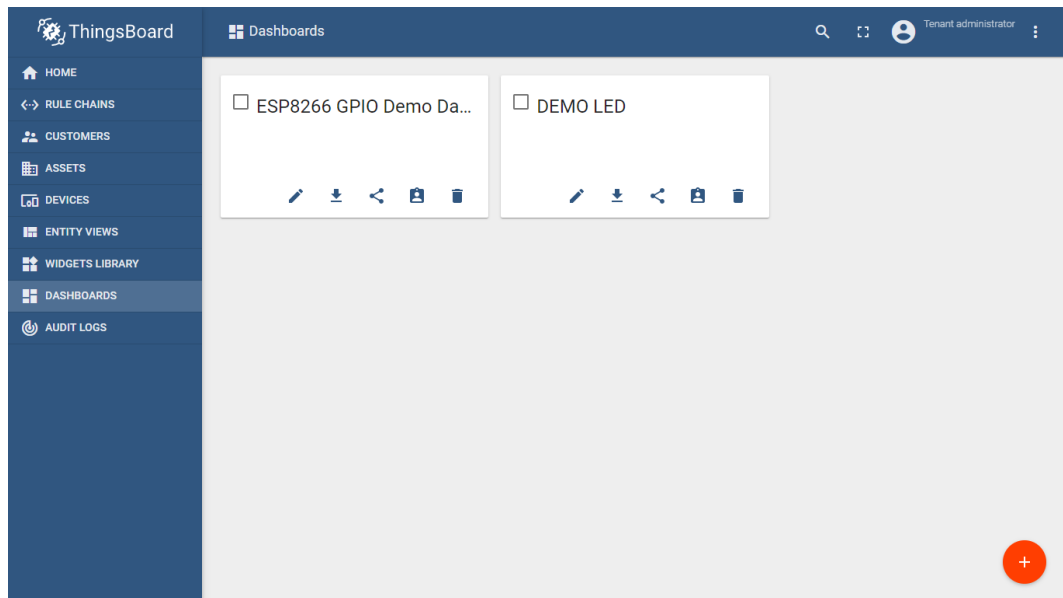
5. Membuat dashboard sendiri dengan menekan tombol add dashboard



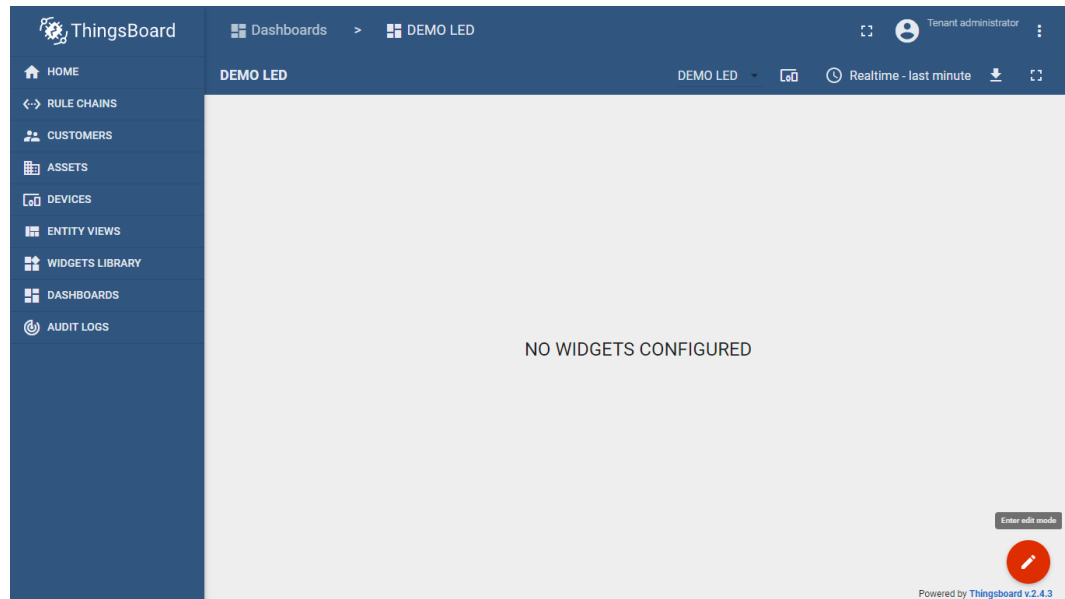
6. Kemudian beri nama DEMO LED, kemudian klik add



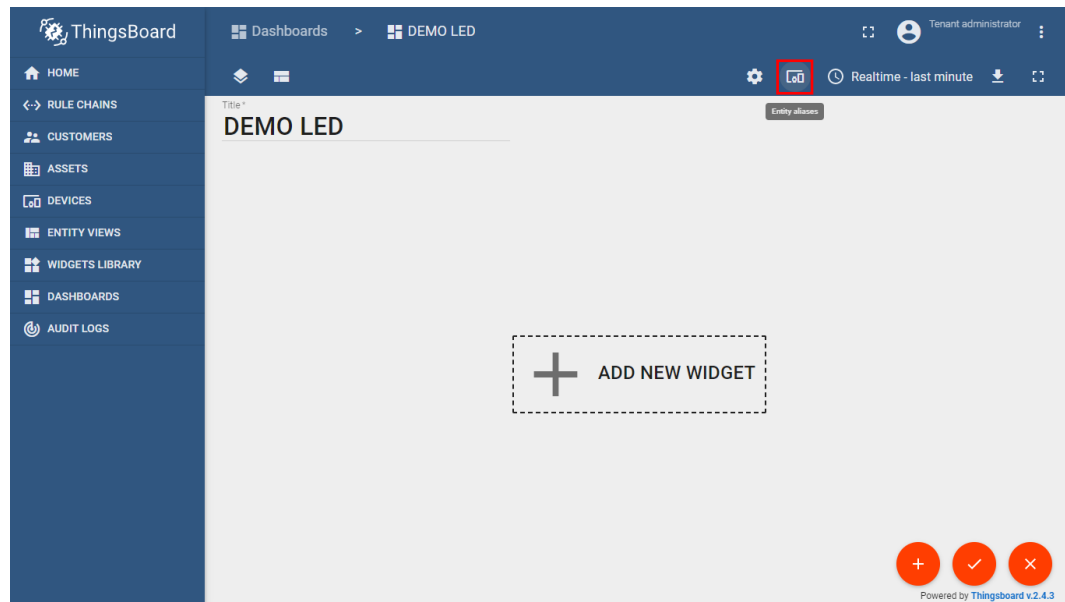
7. Buka Bagian dashboard DEMO LED yang dibuat



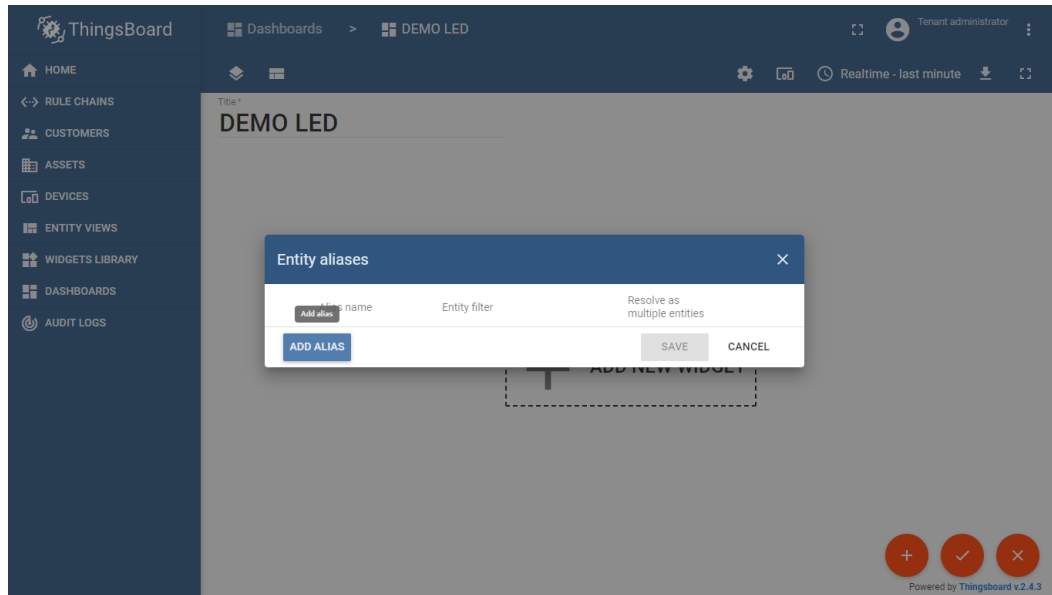
8. Masuk pada bagian **edit mode** pada bagian kanan bawah



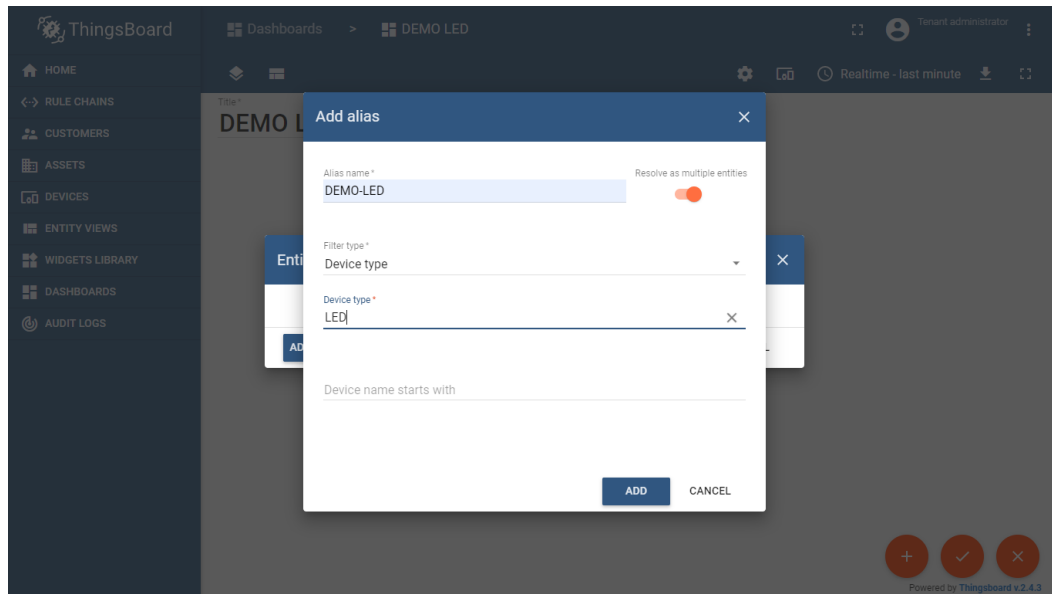
9. Apabila sudah masuk pada edit mode, pilih bagian edit alias pada kolom bagian atas



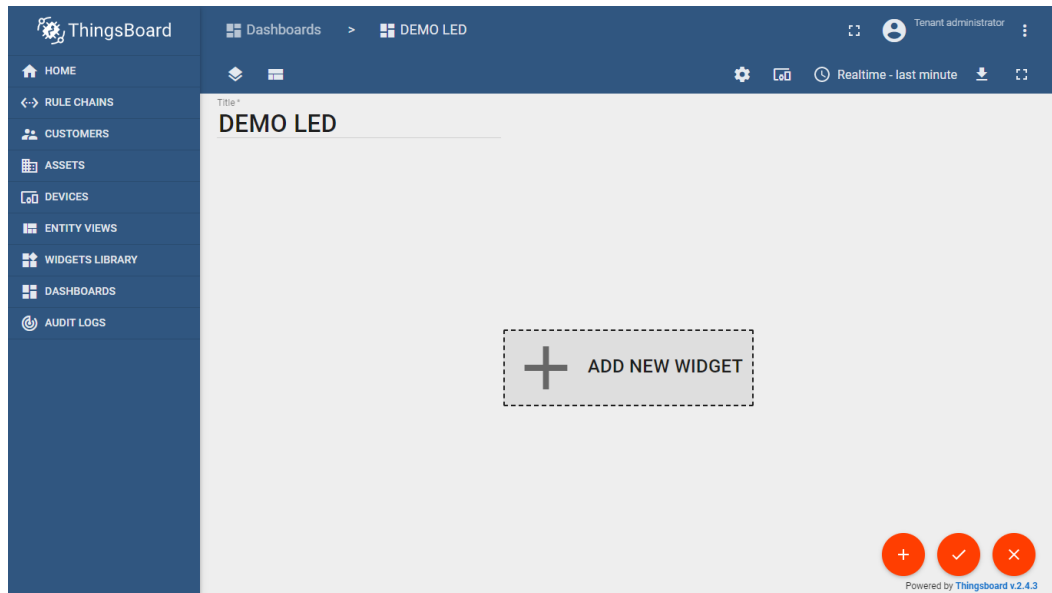
10. Kemudian masuk pada bagian add alias



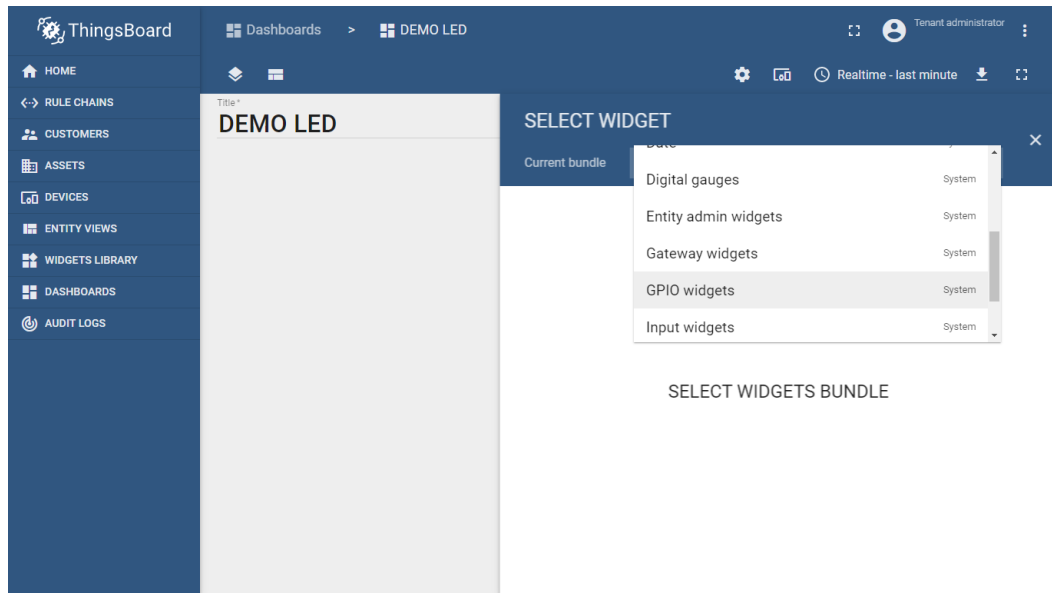
11. Isikan seperti berikut, kemudian klik **add** dan **save**

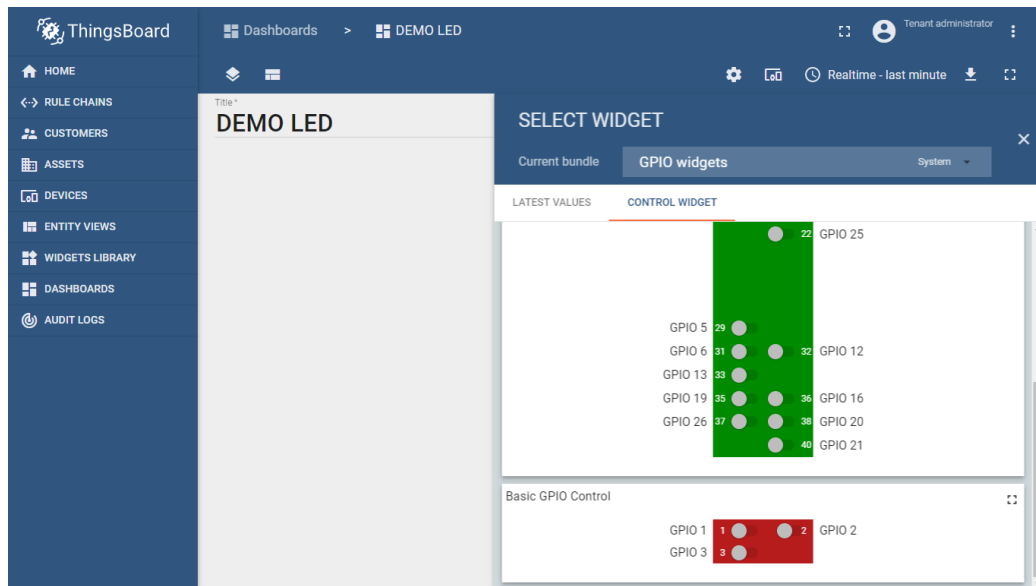


12. Kemudian menambahkan widget

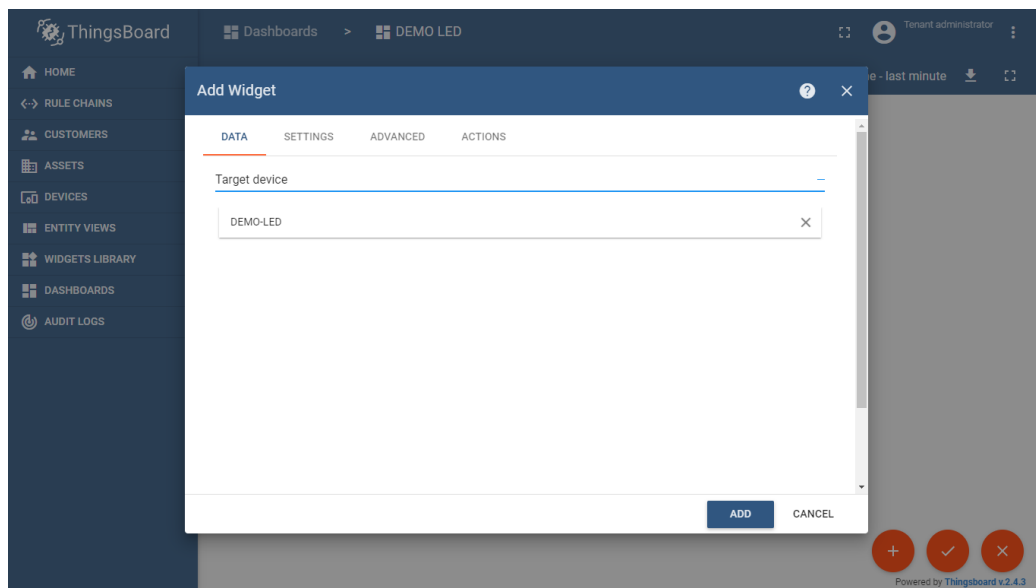


13. Pilih pada GPIO Widget dan pilih GPIO basic

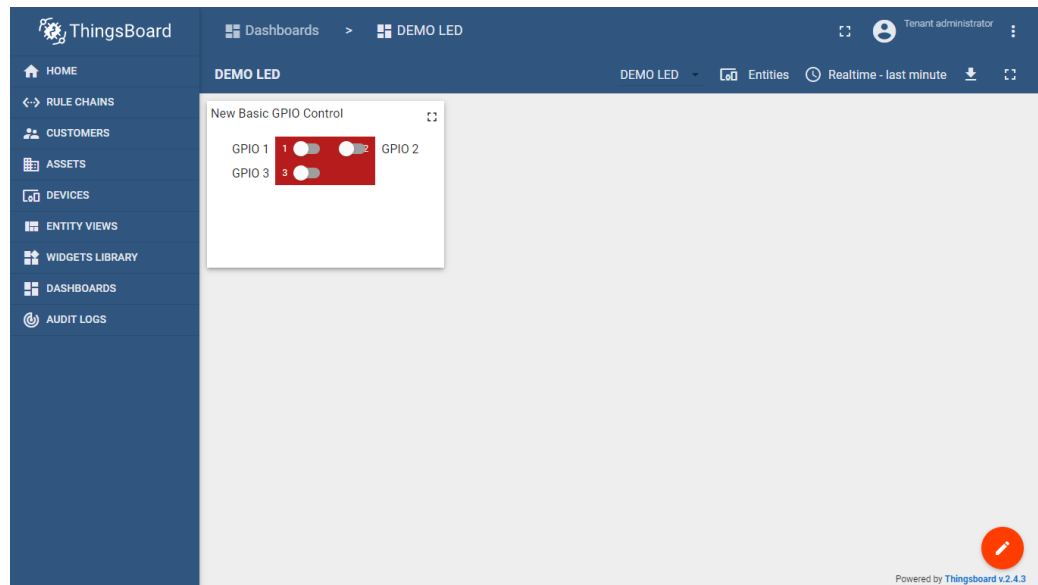




14. Isikan alias dengan DEMO-LED kemudian klik add.



15. GPIO dashboard sudah bisa digunakan sendiri. Hubungkan terlebih dahulu PIN GPIO 1 menggunakan LED.



Tugas

1. Masih dengan Dashboard buatan sendiri yang sama, buatlah rangkaian dengan menggunakan breadboard untuk menghidupkan 2 buah LED.

