

## 12.3 Praktikum Menampilkan Data Sensor dan Kendali Aktuator dengan Thingsboard



Indobot Academy 26 November 2022

### 1. Alat/Instrumen/Aparatus/Media

- Laptop/PC.
- Wemos D1 R1.
- Project Board.
- Kabel USB.
- Kabel Jumper.
- Sensor DHT11.
- LED.
- Thingsboard.
- Arduino IDE.

### 2. Keselamatan Kerja

#### 2.1. Pemasangan Komponen

Perhatikan bagian pin yang digunakan. Terutama bagian komponen yang memiliki polaritas, jangan sampai terbalik antara kaki positif dan negatif.

## 2.2. Penggunaan Ukuran Resistor

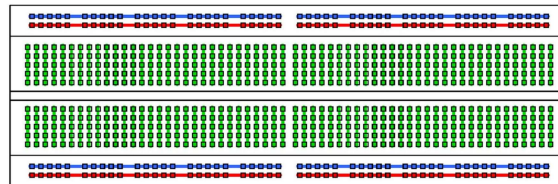
Hal yang perlu diperhatikan lainnya adalah mengenai ukuran resistor. Ukuran resistor dapat menyesuaikan dengan gambar rangkaiannya.

## 2.3. Perhatikan pin

Selanjutnya kita juga perlu memperhatikan pin yang ada dalam wemos D1 maupun sensor.

## 2.4. Pemahaman Jalur Project Board

Agar kita mengetahui tentang jalur yang ada pada project board, kita bisa melihat gambar skema dalam project board seri MB-102 berikut ini.



Gambar 1. Jalur Project Board

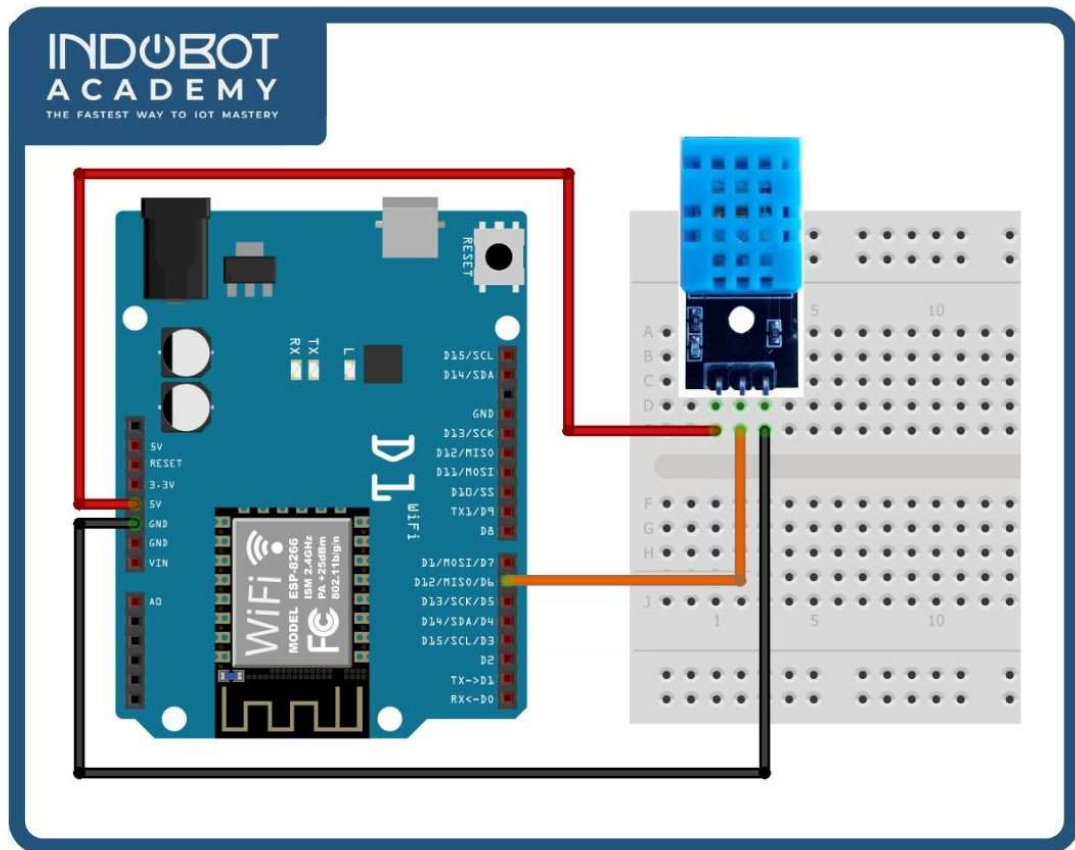
- Bagian tengah project board akan saling terhubung secara vertikal setiap 5 pin. Kemudian akan ada celah, nah celah ini bisa anda gunakan untuk meletakkan push button atau komponen lainnya.
- Untuk bagian atas dan bawah ini terhubung secara horizontal, dengan celah juga di bagian tengah dari project board.

## 3. Langkah Praktikum 1 – Menerima dan Menampilkan Data Sensor

### 3.1. Penjelasan Praktikum

Pada praktikum ini akan dilakukan percobaan mengirim dan menampilkan data pengukuran suhu dan kelembapan pada web dashboard Thingsboard. Adapun sensor yang digunakan untuk pengukuran adalah sensor DHT11. Dengan dashboard ini, anda dapat melakukan pengukuran suhu dan kelembapan secara realtime.

### 3.2. Skema Rangkaian



Keterangan :

- Pin Positif (+) DHT11 -> Pin 5V Wemos D1.
- Pin OUT DHT11 -> Pin D6 Wemos D1.
- Pin Negatif (-) DHT11 -> Pin GND Wemos D1.

### 3.3. Coding

```
// Pemanggilan Library
#include <DHTesp.h>
#include <ThingsBoard.h>
#include <PubSubClient.h>
#include <ESP8266WiFi.h>

// Konfigurasi WiFi & IoT Platform
#define ssid "One Peace"
#define password "bismillah88"
#define TOKEN "9niwGBj0xT8wM7K8hWGX"
#define THINGSBOARD_SERVER "thingsboard.cloud"
```

```

// Inisialisasi Variabel & Objek Baru
WiFiClient espclient;
DHTesp dhtSensor;
ThingsBoard tb(espclient);
const int dhtPin = D6;
int status = WL_IDLE_STATUS;

// Method yang mengatur koneksi
void initWifi(){
    Serial.println("Connecting to AP...");
    WiFi.begin(ssid, password);
    while(WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.println(".");
    }
    Serial.println("Connecting to AP");
}

// Method yang digunakan untuk mengulang koneksi
jaringan ketika bermasalah
void reconnect(){
    status = WiFi.status();
    if(status != WL_CONNECTED) {
        WiFi.begin(ssid, password);
        while(WiFi.status() != WL_CONNECTED) {
            delay(500);
            Serial.println(".");
        }
        Serial.println("connecting again");
    }
}

// Method yang hanya dijalankan sekali setiap device
dinyalakan
void setup(){
    Serial.begin(115200);
    Serial.println();
    initWifi();
}

```

```

    dhtSensor.setup(dhtPin, DHTesp::DHT11);
}

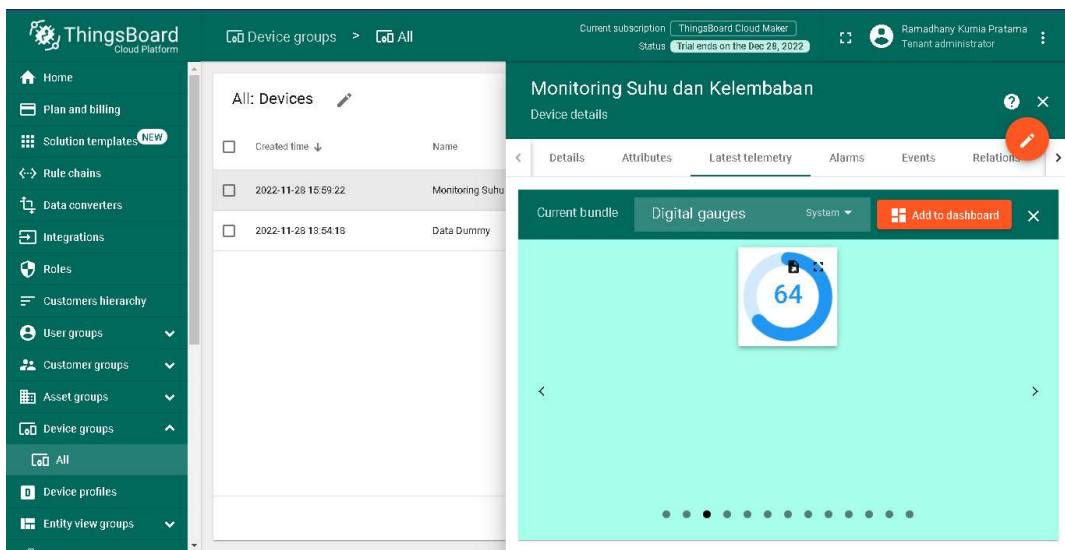
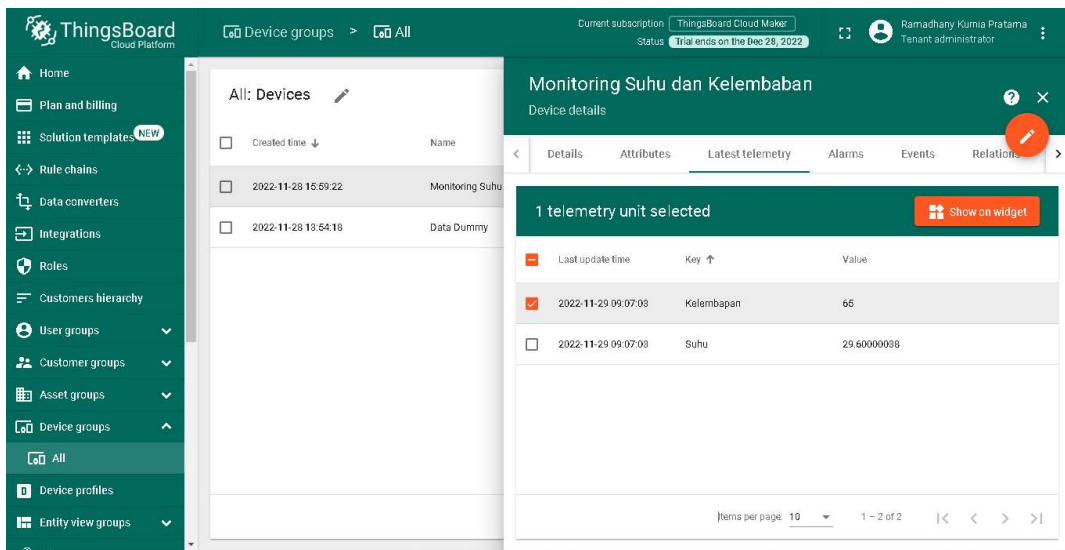
// Method yang dijalankan secara terus menerus
void loop(){
    if (!tb.connected()){
        Serial.print("Menghububgkan ke: ");
        Serial.print(THINGSBOARD_SERVER);
        Serial.print("Dengan Token: ");
        Serial.print(TOKEN);
        if (!tb.connect(THINGSBOARD_SERVER, TOKEN)){
            Serial.println("Gagal menghubungkan");
            return;
        }
    } Serial.println("Sending data....");
    TempAndHumidity data =
dhtSensor.getTempAndHumidity();
    tb.sendTelemetryFloat("Suhu", data.temperature);
    tb.sendTelemetryFloat("Kelembapan", data.humidity);
    Serial.print("Humidity = ");
    Serial.print(data.humidity);
    Serial.print("Temperature = ");
    Serial.print(data.temperature);
}

```

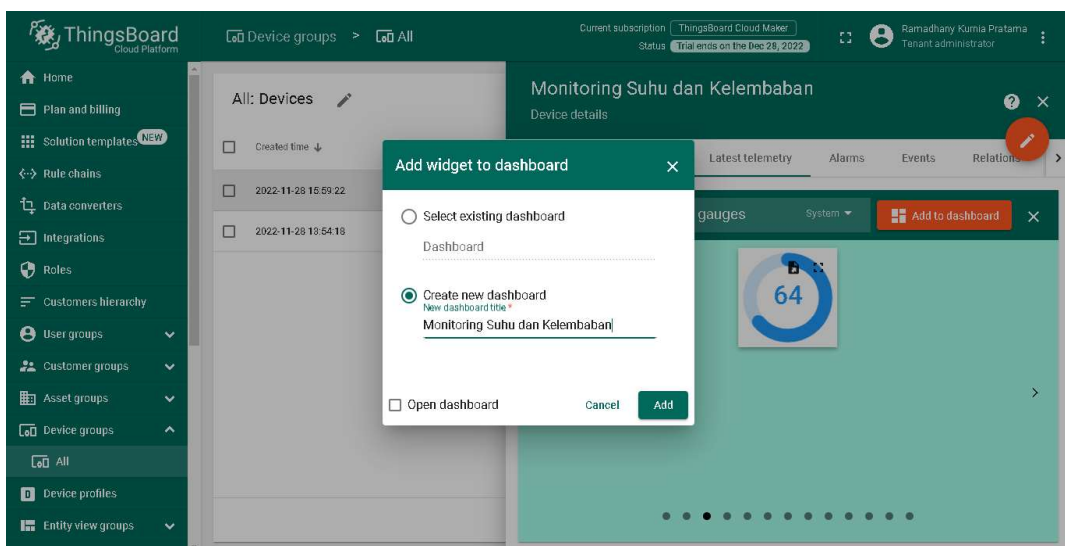
### 3.4. Konfigurasi Thingsboard

- Pertama, buatlah device dengan nama Monitoring Suhu dan Kelembapan. Cara pembuatan device sama seperti praktikum sebelumnya.
- Setelah itu, salin acces token dari device, tempelkan ke kode program yang ada buat di Arduino IDE dengan memakai referensi dari poin: 3.3. Coding.
- Jika sudah, jalankan program.
- Lalu, buka serial monitor, maka akan muncul tampilan seperti pada gambar di bawah ini.

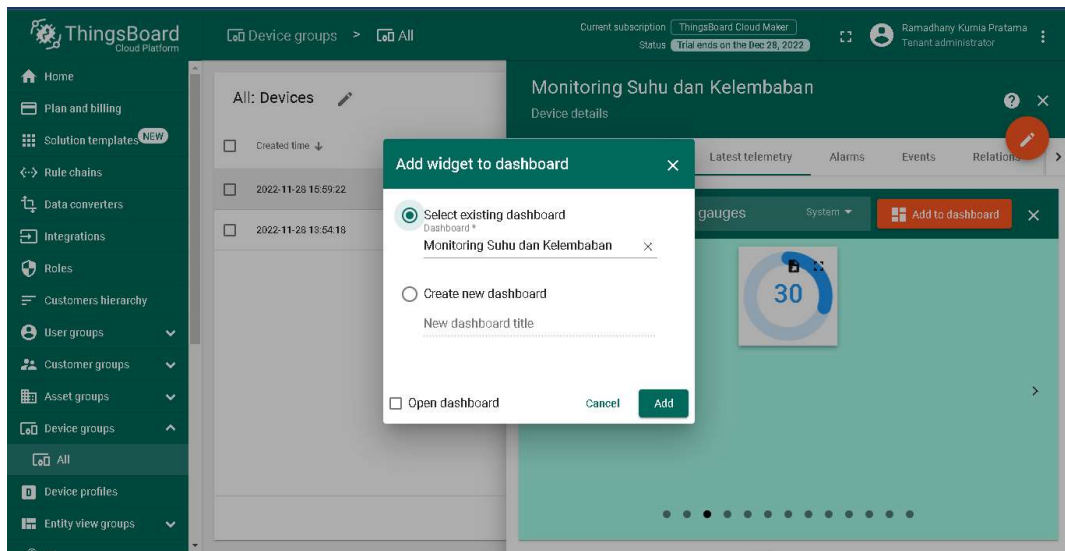




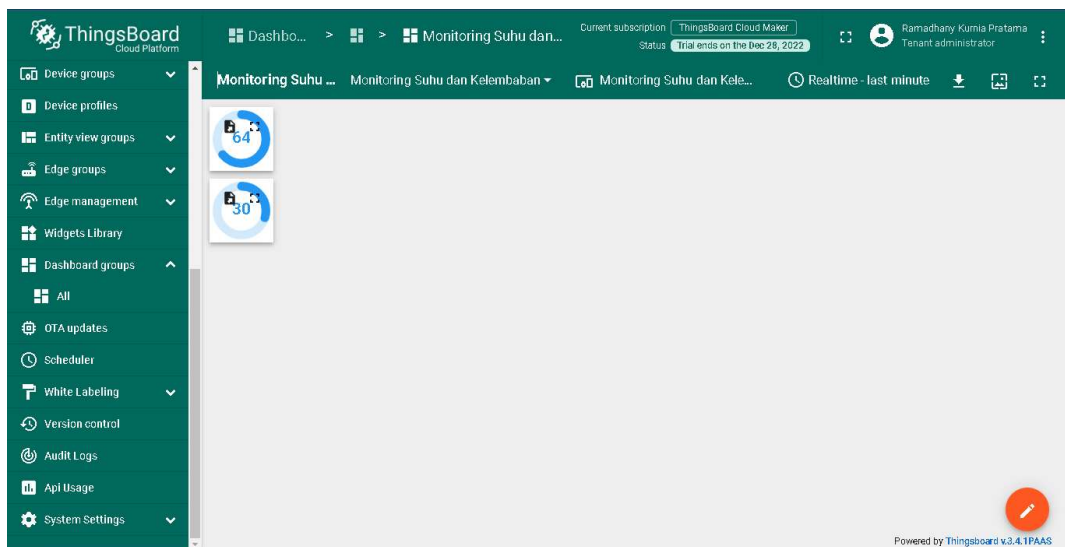
Setelah itu tambahkan ke dashboard dengan klik Add to dashboard -> Create New Dashboard, lalu anda dapat menuliskan: Monitoring Suhu dan Kelembaban -> Klik Add.



Untuk konfigurasi data Suhu, caranya sama seperti sebelumnya. Hanya saja untuk menambahkannya ke dashboard yang sudah kita buat, anda dapat pilih opsi: Select existing dashboard, lalu silahkan pilih nama dashboard yang sudah kita buat sebelumnya.



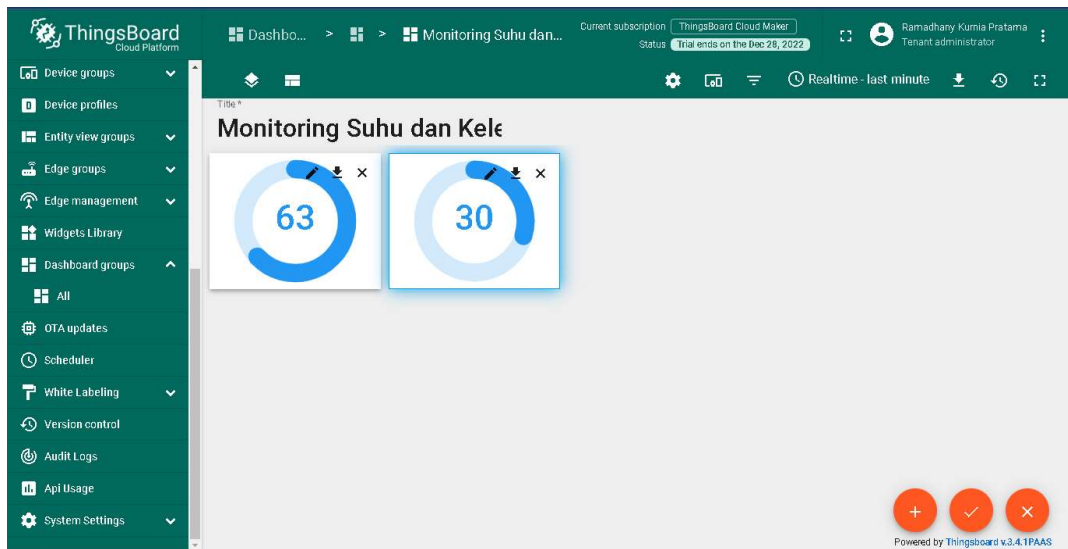
- Setelah selesai menambahkan widget, sekarang kita bisa membuka dashboard yang dibuat.
- Untuk membuka dashboard, pada menu, pilih Dashboard Groups -> Pilih All -> Pilih Dashboard yang baru dibuat -> Open Dashboard.
- Tampilan dashboard yang baru dibuat seperti gambar di bawah ini.



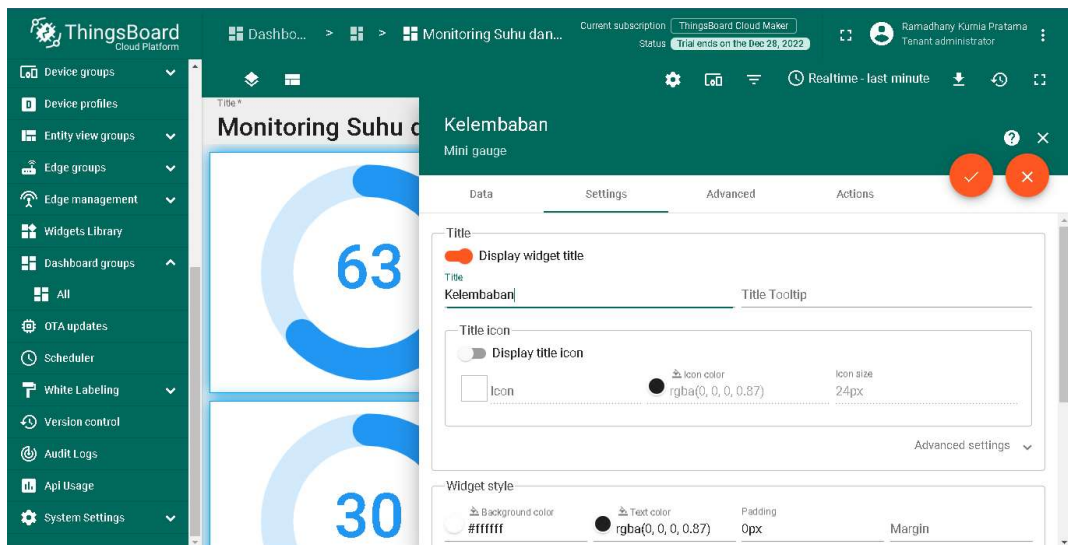
- Tampilan yang diberikan masih kurang terlihat bagus karena terlalu kecil dan sulit membedakan data.
- Untuk mengedit dashboard, klik icon pencil di pojok kanan bawah, setelah itu anda dapat mengedit tampilan dashboard, seperti mengubah ukuran



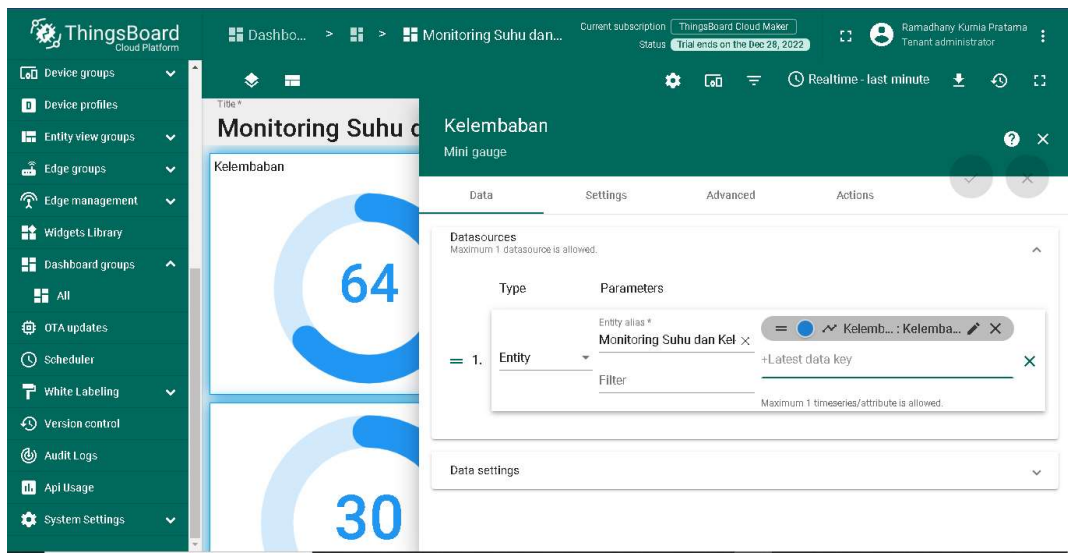
dengan cara tekan ujung kotak atau tepi widget lalu ditarik. Untuk memindahkan posisi widget tekan bagian tubuh widget (bukan tepi) lalu lakukan drag.



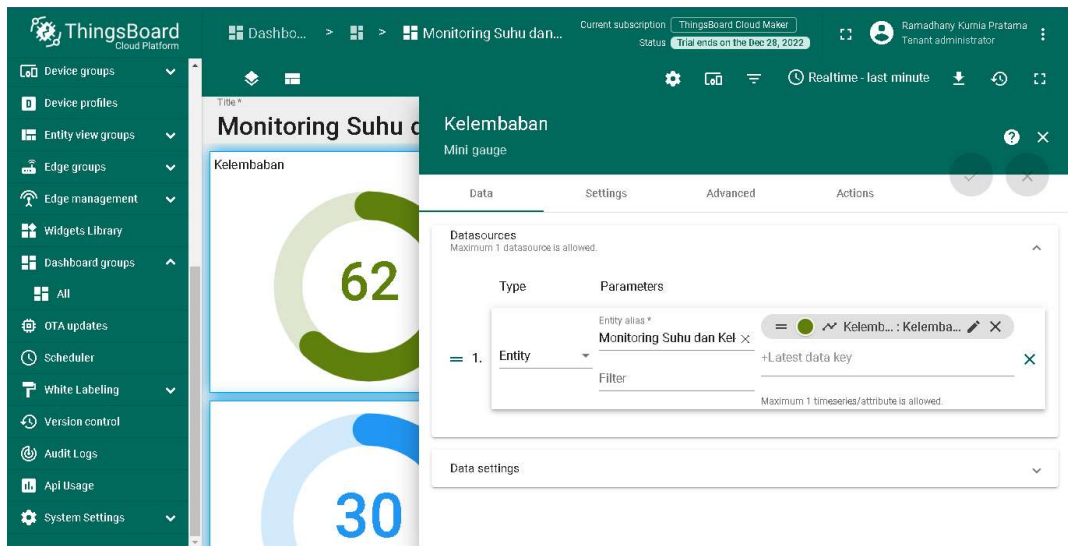
- Untuk memberi nama atau judul widget klik icon pensil pada kotak widget, lalu klik setting -> klik Display widget title -> Beri nama widget dengan mengisi title. Jika sudah klik checklist untuk menyimpan.



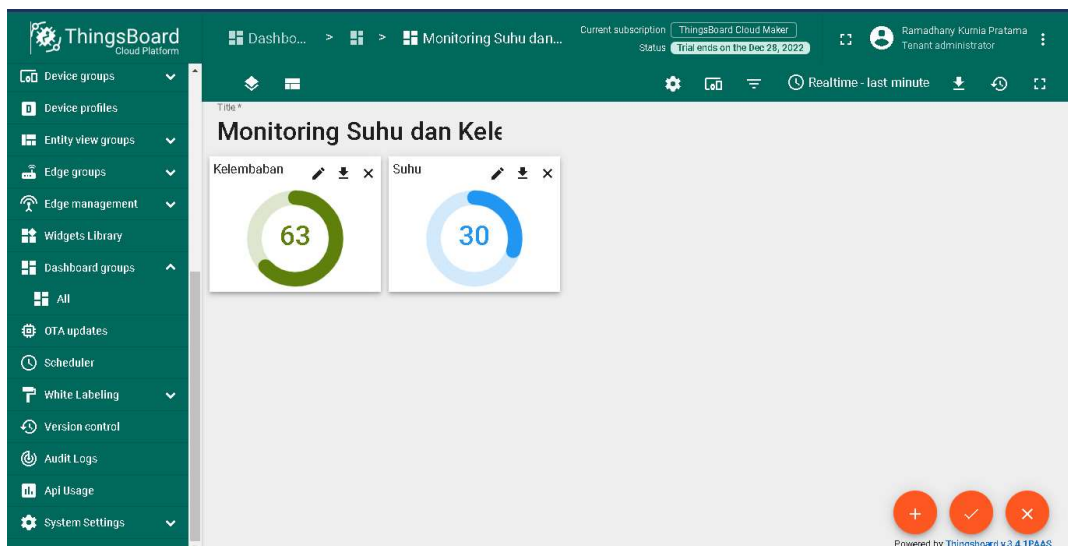
- Untuk mengganti warna widget, klik data lalu terdapat icon berwarna pada Datasources. Klik pada icon berwarna lalu pilih warna untuk mengganti warna. Jika sudah klik checklist untuk menyimpan.

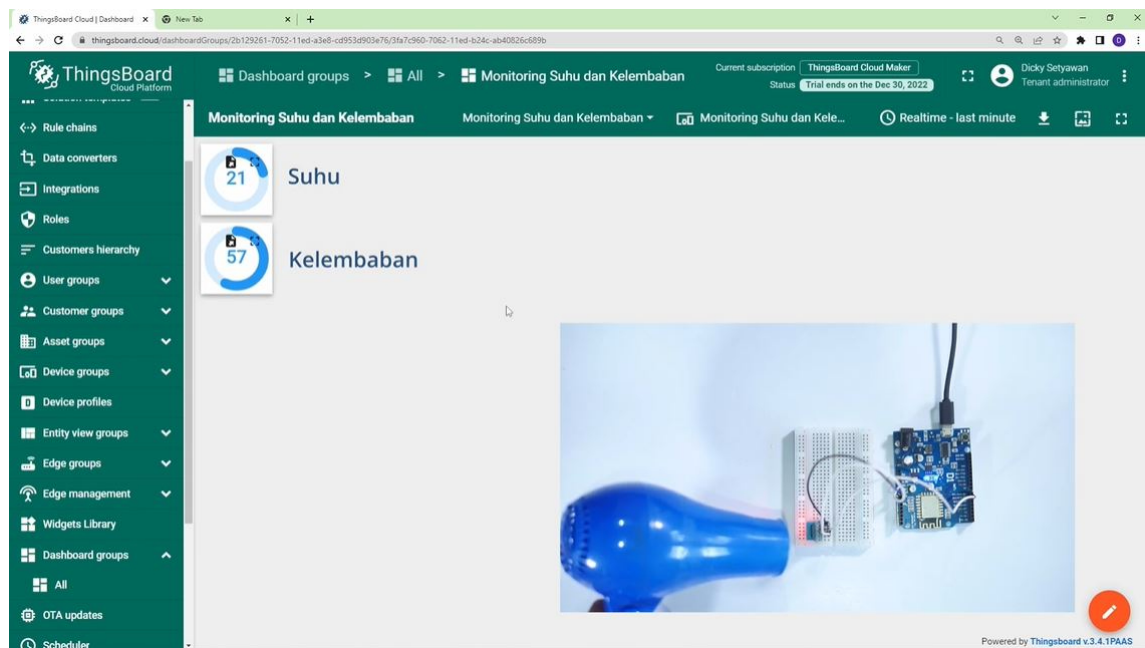


- Jika sudah tersimpan maka warna widget akan berubah sesuai dengan yang kita pilih.



- Lakukan konfigurasi juga pada widget lainnya.
- Jika sudah, klik checklist untuk menyimpan dan dashboard siap digunakan.



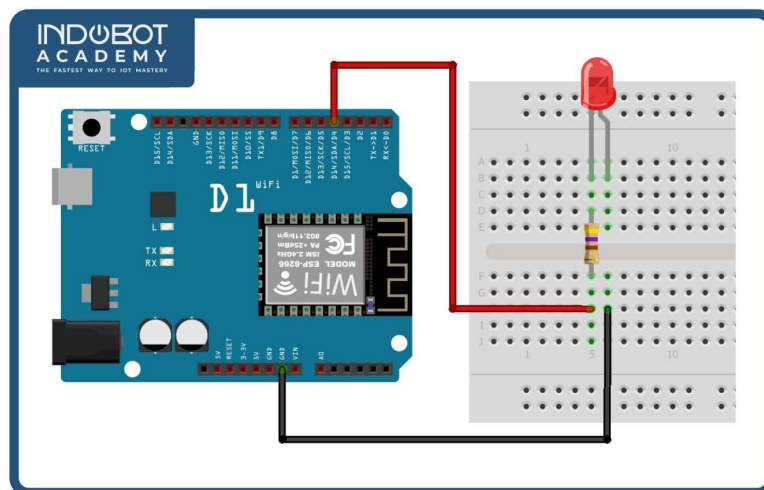


## 4. Langkah Praktikum 2 – Kontrol LED dengan Web Dashboard Thingsboard

### 4.1. Penjelasan Praktikum

Pada praktikum ini, anda akan melakukan percobaan untuk mengendalikan LED dengan menggunakan web dashboard Thingsboard.

### 4.2. Skema Rangkaian



Keterangan :

- Pin Positif (+) LED -> Kaki Resistor.
- Kaki Resistor yang terhubung ke LED -> Pin D4 Wemos D1.
- Pin Negatif (–) LED -> Pin GND Wemos D1.

### 4.3. Coding

```
// Pemanggilan Library
#include <ThingsBoard.h>
#include <PubSubClient.h>
#include <ESP8266WiFi.h>

// Konfigurasi WiFi & IoT Platform
WiFiClient espclient;
#define ssid "One Peace"
#define password "bismillah88"
#define TOKEN "mot12KNyUX9u5O5iLV8R"
#define THINGSBOARD_SERVER "thingsboard.cloud"

// Inisialisasi Variabel & Objek Baru
ThingsBoard tb(espclient);
const int led = D4;
bool pumpStatus = false;
bool previousState = false;
int status = WL_IDLE_STATUS;
bool subscribed = false;
const size_t callbacks_size = 2;
RPC_Callback callbacks[callbacks_size] = {{
    "example_set_switch", processSwitchChange }};

// Method yang mengatur koneksi
void initWifi(){
    Serial.println("Connecting to AP...");
    WiFi.begin(ssid, password);
    while(WiFi.status() != WL_CONNECTED){
        delay(500);
        Serial.println(".");
    } Serial.println("Connecting to AP");
}

// Method yang digunakan untuk mengulang koneksi
jaringan ketika bermasalah
void reconnect(){
    status = WiFi.status();
```

```

    if(status != WL_CONNECTED){
        WiFi.begin(ssid, password);
        while(WiFi.status() != WL_CONNECTED){
            delay(500);
            Serial.println(".");
        } Serial.println("connecting again");
    }
}

// Method yang hanya dijalankan sekali setiap device
dinyalakan
void setup(){
    Serial.begin(115200);
    Serial.println();
    initWifi();
    pinMode(led, OUTPUT);
}

// Method untuk mengatur RPC
RPC_Response processSwitchChange(const RPC_Data &data)
{
    Serial.println("Received the set switch method");
    char params[10];
    serializeJson(data, params);
    Serial.println(params);
    String _params = params;
    if(_params == "true"){
        Serial.println("True");
        pumpStatus = true;
        digitalWrite(led, HIGH);
        return RPC_Response("example_set_switch", "true");
    } else if (_params == "false"){
        Serial.println("False");
        pumpStatus = false;
        digitalWrite(led, LOW);
        return RPC_Response("example_set_switch",
"false");
    }

    // bool switch_state = data["switch"];
    // Serial.print("Example switch state: ");

```

```

    // Serial.println(switch_state);
    // digitalWrite(pump, switch_state);
    // return RPC_Response("example_set_switch",
switch_state);
}

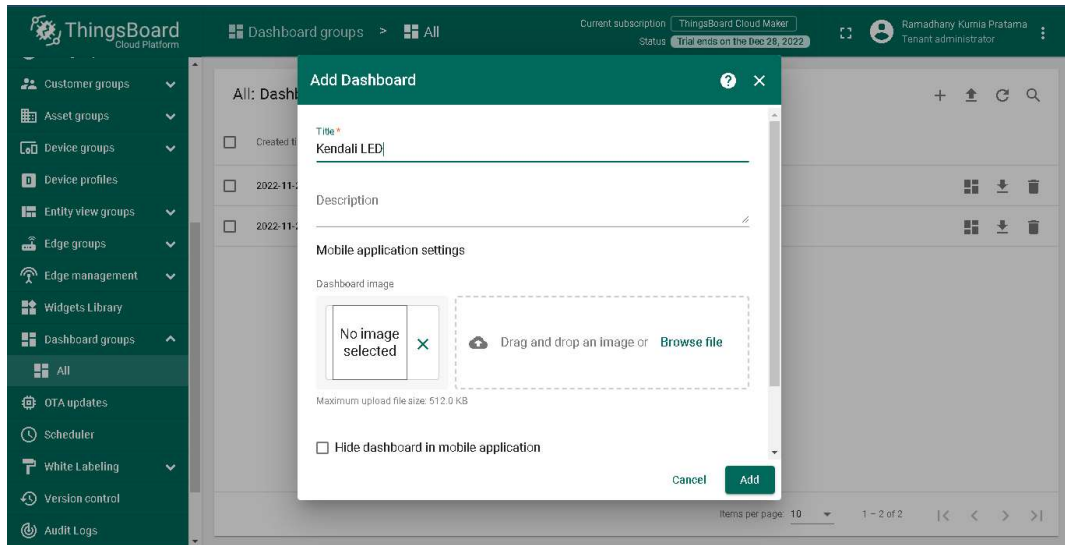
// Method yang dijalankan secara terus menerus
void loop(){
    if(!tb.connected()){
        Serial.print("Menghubungkan ke: ");
        Serial.print(THINGSBOARD_SERVER);
        Serial.print("Dengan Token: ");
        Serial.print(TOKEN);
        if (!tb.connect(THINGSBOARD_SERVER, TOKEN)) {
            Serial.println("Gagal menghubungkan");
            return;
        }
    }
    if (!subscribed) {
        Serial.println("Subscribing for RPC...");
        if(!tb.RPC_Subscribe(callbacks, callbacks_size)) {
            Serial.println("Failed to subscribe for RPC");
            return;
        }
        Serial.println("Subscribe done");
        subscribed = true;
    }
    // Serial.println("Sending data...");
    if (previousState != pumpStatus){
        digitalWrite(led, pumpStatus);
        previousState = pumpStatus;
        tb.sendAttributeBool("Status LED", pumpStatus);
    } tb.loop();
}

```

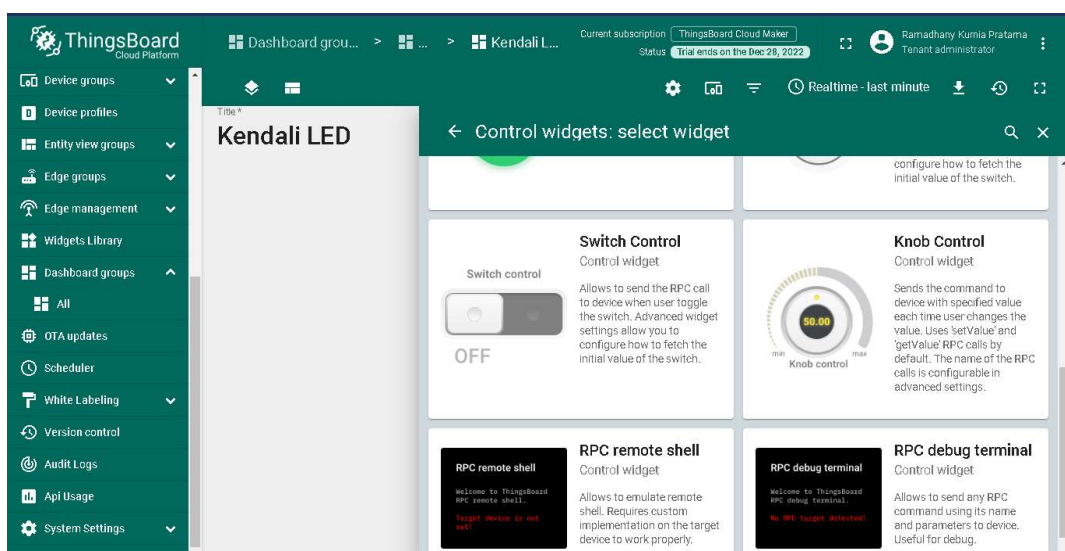
#### 4.4. Konfigurasi Thingsboard

- Pertama, buatlah device dengan nama Monitoring Suhu dan Kelembapan. Cara pembuatan device sama seperti praktikum sebelumnya.

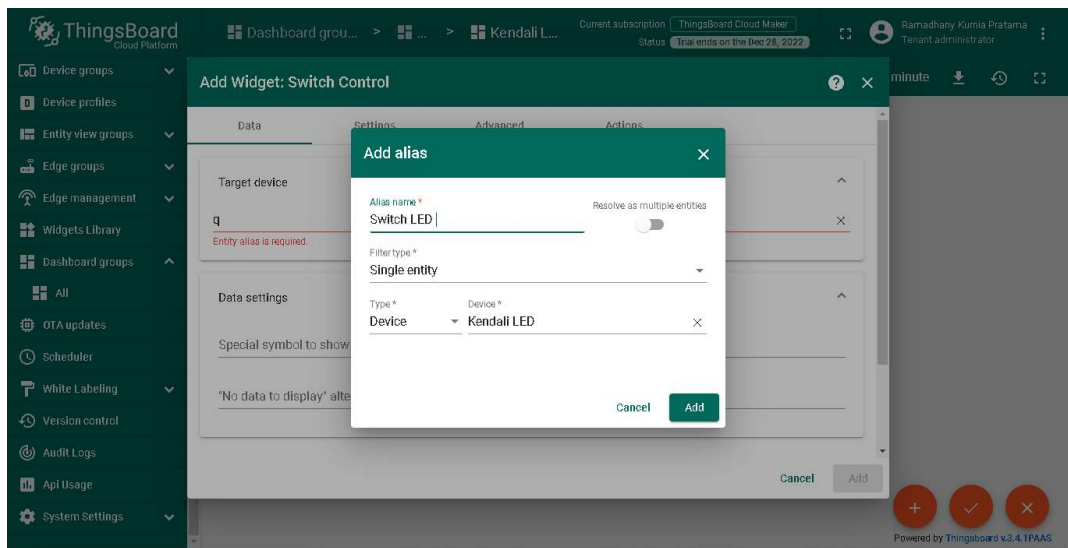
- Setelah itu, salin acces token dari device, tempelkan ke kode program yang ada buat di Arduino IDE dengan memakai referensi dari poin: 4.3. Coding.
- Pada menu, klik Dashboard Groups -> All -> Lalu buatlah dashboard dengan nama Kendali LED.



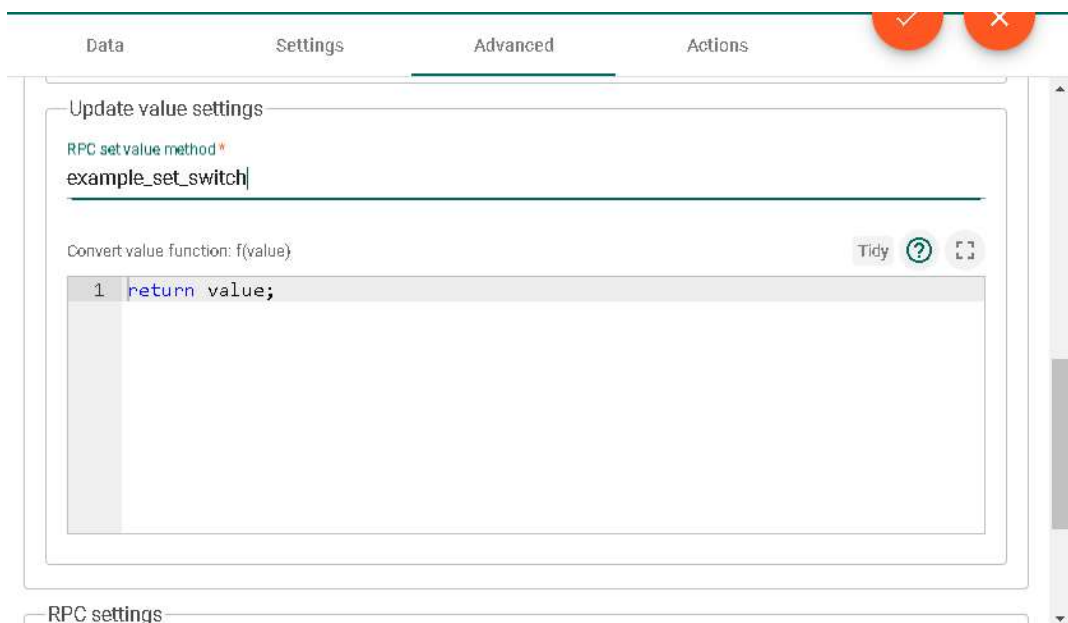
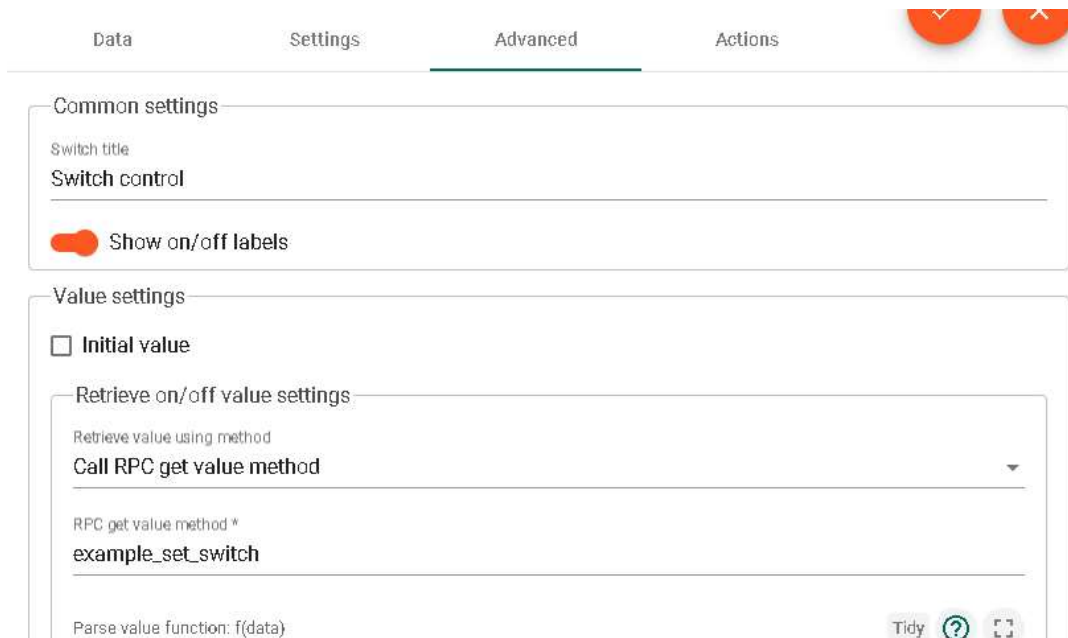
- Jika sudah, jalankan program.
- Buka dashboard, tambahkan control widget yaitu silahkan anda pilih switch.



- Lakukan konfigurasi pada switch, klik data lalu isi Target Device -> Pilih Create New One -> Lakukan konfigurasi dengan beri nama Switch LED, lalu Filter Type silahkan pilih Single Entity, kemudian Type silahkan pilih Device, selanjutnya untuk Device silahkan pilih nama device yang sudah kita buat sebelumnya.



- Selanjutnya klik Advanced, ganti isi RPC get Value method dan RPC set Value method menjadi example\_set\_switch.





- Jika sudah, klik Add.
- Simpan dashboard, lalu cobalah untuk klik switch pada dashboard dan perhatikan LED pada rangkaian. Jika program berhasil, LED akan menyala ketika Switch kondisi On begitu juga sebaliknya.

