

13.1 Memanfaatkan Fitur Alarm pada Sistem Monitoring Suhu dan Kelembapan Udara



Indobot Academy 30 November 2022

1. Alat/Instrumen/Aparatus/Media

- Laptop/PC.
- Wemos D1 R1.
- Sensor DHT11.
- Project Board.
- Kabel Jumper.
- Thingsboard.

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

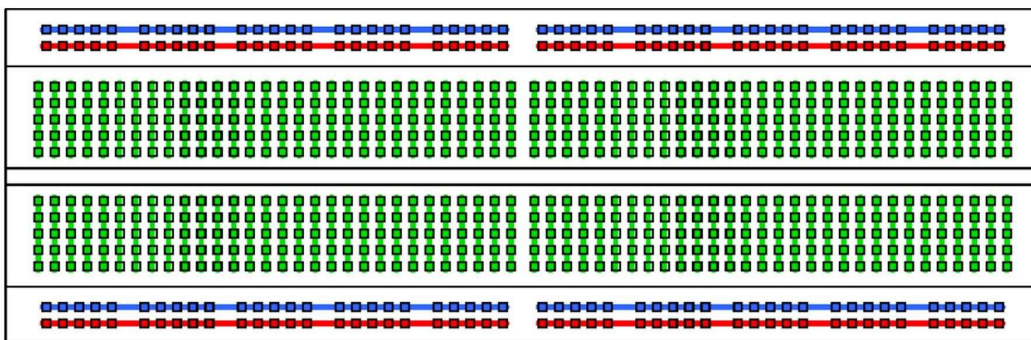
Hal yang perlu diperhatikan lainnya adalah mengenai ukuran resistor. Ukuran resistor dapat menyesuaikan dengan gambar rangkaiannya.

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

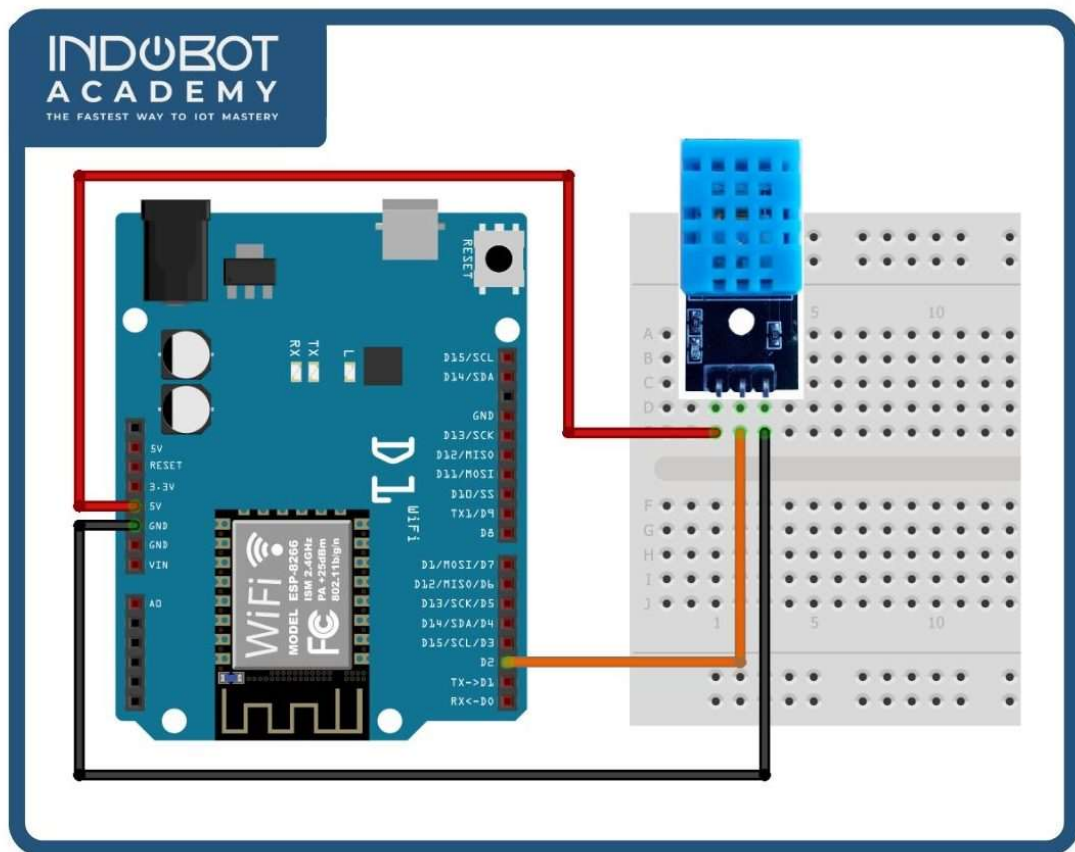
- Bagian tengah project board akan saling terhubung secara vertikal setiap 5 pin. Kemudian akan ada celah, nah celah ini bisa anda gunakan untuk meletakkan push button atau komponen lainnya.
- Untuk bagian atas dan bawah ini terhubung secara horizontal, dengan celah juga di bagian tengah dari project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

3. Langkah Praktikum 1 – Monitoring Suhu dan Kelembapan Udara dengan Alarm Peringatan

3.1. Penjelasan Praktikum

Pada praktikum ini, kita akan membuat sebuah sistem alarm suhu udara menggunakan Thingsboard. Alarm merupakan fitur yang disediakan dalam Thingsboard. Dengan alarm ini, kita dapat membuat sebuah sistem pengingat atau peringatan berdasarkan nilai dari sensor.

3.2. Skema Rangkaian



Keterangan :

- Hubungkan PIN VCC DHT11 -> PIN 5V Wemos.
- Hubungkan PIN GND DHT11 -> PIN GND Wemos.
- Hubungkan PIN Data DHT11 -> PIN D2 Wemos.

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 "JONZP5apRwiRwefLNt6H"
#define THINGSBOARD_SERVER "thingsboard.cloud"
```

```

// Inisialisasi Variabel & Objek Baru
WiFiClient espclient;
DHTesp dhtSensor;
ThingsBoard tb(espclient);
const int dhtPin = D2;
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();
}

```

```

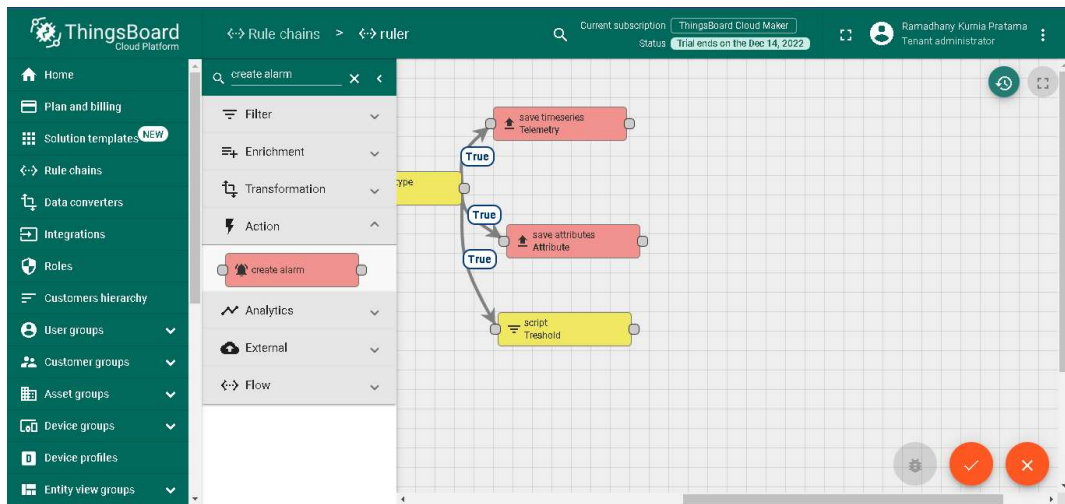
    dhtSensor.setup(dhtPin, DHTesp::DHT11);
}

// 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

- Buatlah rule chain baru dengan nama "ruler", untuk isi dari rule chain tersebut sama seperti sebelumnya. Untuk mempersingkat pembuatan bisa dengan blok semua node dari rule chain "My Rule" sebelumnya (kecuali node Input), lalu pastekan di rule chain "ruler".
- Pada rule chain ini, node "send email" diganti dengan node "create alarm".



- Drag dan Drop node "create alarm" di sebelah node "threshold", persisnya posisikan ke atas. Kemudian, lakukan konfigurasi seperti gambar di bawah.

The screenshot shows the 'Add rule node: create alarm' dialog box. The title bar is green with a question mark and a close button. The main area has a white background. At the top, there is a 'Name*' field with the value 'Alarm Suhu' and a 'Debug mode' checkbox. Below this is a 'Use message alarm data' checkbox. The 'Alarm details builder' section contains a JavaScript function definition:

```
function Details(msg, metadata, msgType) {  
  1 var details = {};  
  2   details = "Suhu Terlalu Tinggi";  
  3 return details;  
}
```

 To the right of the code editor are buttons for 'Tidy', a question mark, and a refresh icon. At the bottom right are 'Cancel' and 'Add' buttons.

Add rule node: create alarm
?
X

}

Test details function

Alarm type *
Alarm Peringatan Suhu

Hint: use \${metadata.aKey} for value from metadata, \${messageKey} for value from message body

☐ Use alarm severity pattern

Alarm severity *
Critical

☐ Propagate alarm to related entities

☐ Propagate alarm to entity owner (Customer or Tenant)

☐ Propagate alarm to entity owners hierarchy

☐ Propagate alarm to Tenant

Cancel
Add

- Sambungkan node "threshold" ke node "create alarm", lalu pilih Add Link "True".

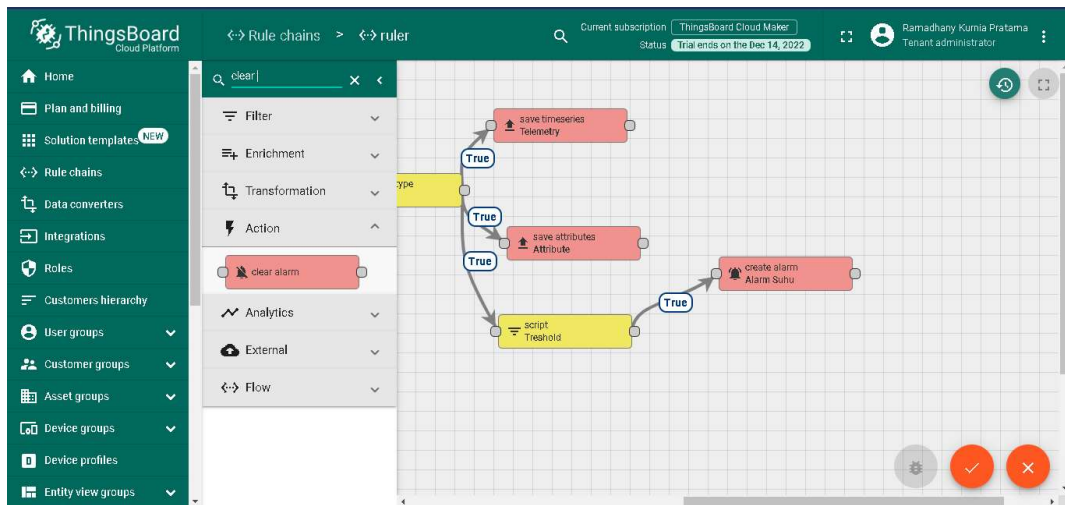
Add link
?
X

Link labels:

True X Link label

Cancel
Add

- Setelah itu tambahkan node "clear alarm", lalu letakkan di sebelah node "threshold", persisnya posisikan ke bawah.



- Lakukan konfigurasi seperti gambar di bawah.

Add rule node: clear alarm

Name * ☐ Debug mode

Alarm details builder

function Details(msg, metadata, msgType) {

```
1 var details = {};  
2 if (metadata.prevAlarmDetails) {  
3     details = JSON.parse(metadata.prevAlarmDetails  
4     );  
5     //remove prevAlarmDetails from metadata  
6     delete metadata.prevAlarmDetails;  
7     //now metadata is the same as it comes IN this  
8     rule node  
9 }  
}
```


Add rule node: clear alarm ? ✕

```

6    //now metadata is the same as it comes IN this
7    rule node
8  }
9  }
10  return details;

```

Test details function

Alarm type*
Clear Alarm Suhu

Hint: use `${metadat aKey}` for value from metadata, `${messageKey}` for value from message body

Description

Cancel Add

- Jika sudah, klik "Add" untuk menyimpan.
- Sambungkan node "threshold" ke node "clear alarm", kemudian pilih Add Link "False".

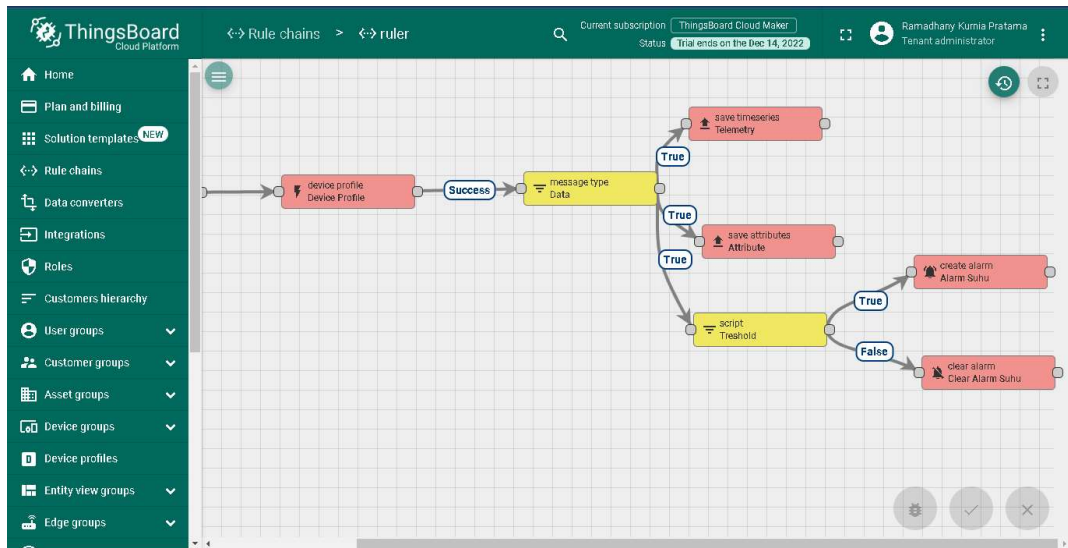
Add link ? ✕

Link labels

False ✕ Link label

Cancel Add

- Jika sudah, maka rule chain secara keseluruhan akan terlihat seperti gambar di bawah.



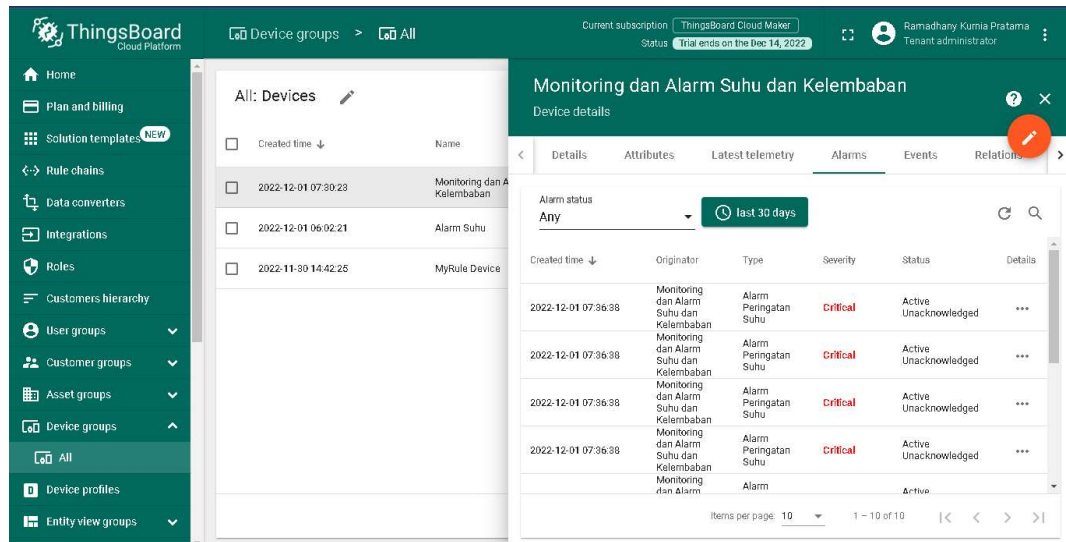
- Selanjutnya, buatlah "Device profile" dengan cara Klik "Device Profiles" -> "Create New Device Profile". Lalu, lakukan konfigurasi seperti gambar di bawah. Kemudian, klik "Add" untuk menyimpan.

The screenshot shows the 'Add device profile' dialog in ThingsBoard. The dialog has four tabs: '1 Device profile details', '2 Transport configuration Optional', '3 Alarm rules (0) Optional', and '4 Device provisioning Optional'. The 'Device profile details' tab is active. It contains the following fields: 'Name*' (Alarm Suhu), 'Rule chain' (ruler), 'Mobile dashboard' (Used by mobile application as a device details dashboard), 'Queue', and 'Device profile image' (No image selected). The 'Next: Transport configuration' button is visible at the bottom right, along with 'Cancel' and 'Add' buttons.

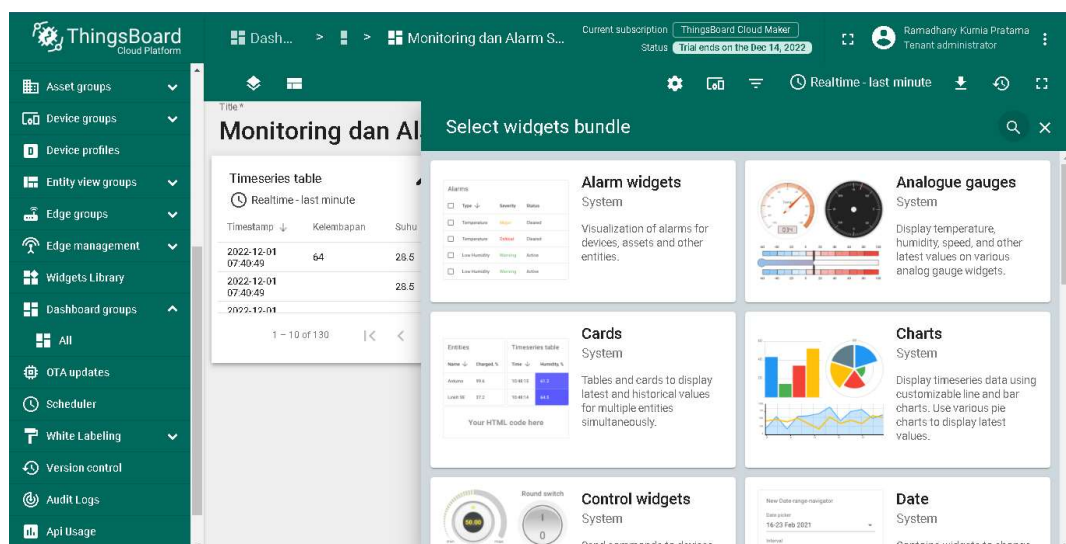
- Selanjutnya adalah membuat device, buatlah device dengan konfigurasi seperti gambar di bawah.

The screenshot shows the 'Add new device' dialog in ThingsBoard. The dialog has two tabs: '1 Device details' and '2 Credentials Optional'. The 'Device details' tab is active. It contains the following fields: 'Name*' (Monitoring dan Alarm Suhu dan Kelembaban), 'Label', 'Device profile*' (Select existing device profile: Alarm Suhu), 'Is gateway' (checkbox), and 'Description'. The 'Next: Credentials' button is visible at the bottom right, along with 'Cancel' and 'Add' buttons.

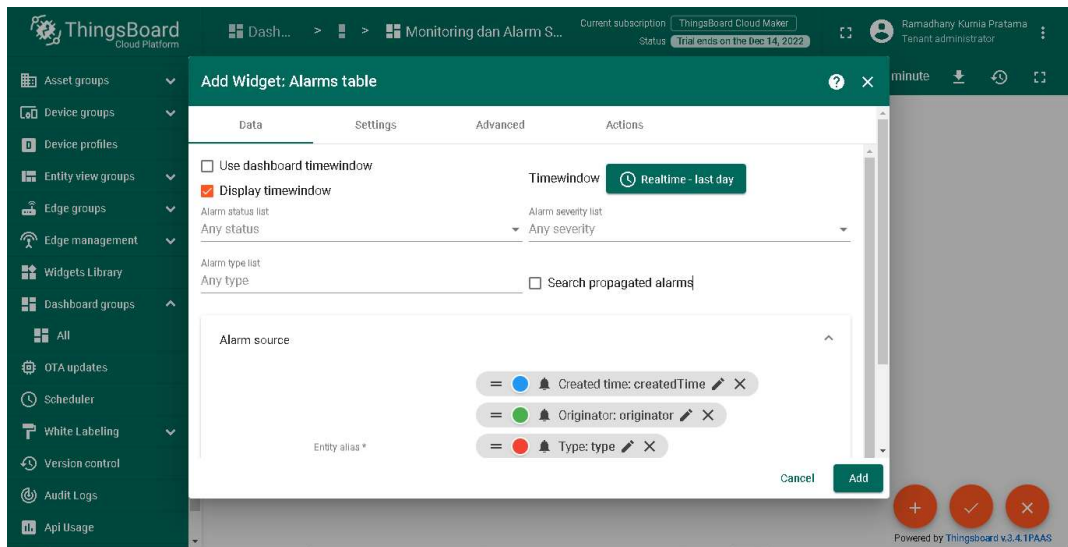
- Salin Access Token, lalu masukkan ke kode program, kemudian jalankan.
- Jika sudah, perhatikan data suhu pada Latest Telemetry, dan cobalah untuk menaikkan suhunya lebih dari 30 derajat celcius.
- Buka Alarm, maka akan muncul seperti gambar berikut ini.



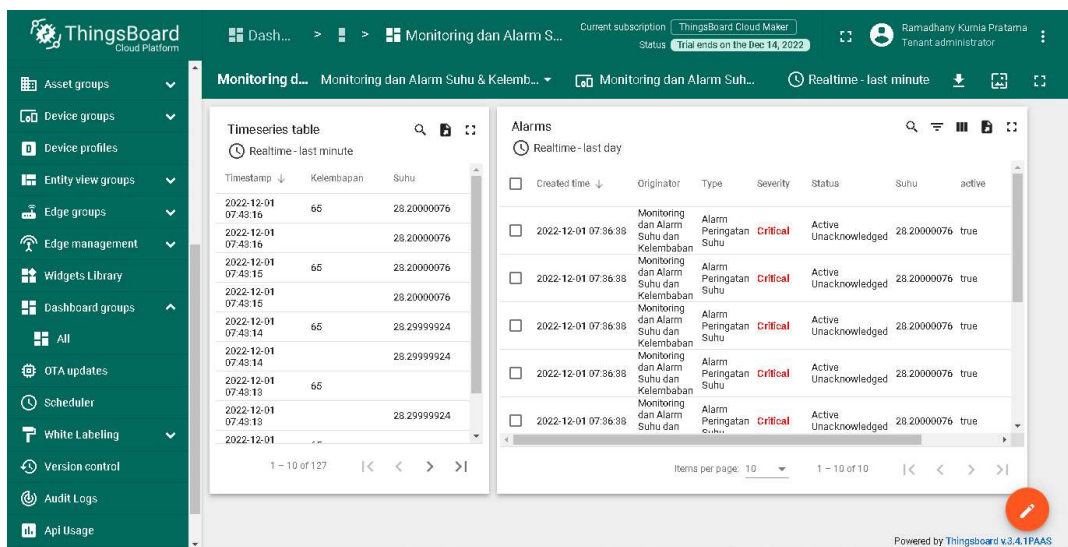
- Jika alarm muncul ketika suhu lebih dari 30 derajat celcius, maka rule chain yang dibuat tadi telah berhasil. Untuk membuat alarm di dalam dashboard, caranya adalah dengan membuat dashboard baru, lalu masukkan widget alarm.



- Lakukan konfigurasi seperti gambar di bawah ini, untuk widget alarm.



- Pada gambar di bawah ini, untuk menambah tampilan data Suhu dan Kelembapan, maka anda dapat memilih widget cards.



- Widget alarm berhasil dibuat, dan siap digunakan.