Hands On Materi THINGSBOARD – DATA PREPARATION

Pada modul ini akan dibahas mengenai Thingsboard – Data Preparation. Dimulai dari input data menggunakan CURL, Rule Engine, visualisasi sederhana, dan agregasi data. Serta akan diberikan contoh penerapan pada kasus “Pengumpulan dan Visualisasi Data Termometer di Dalam Gedung dan Menaikkan Alarm Apabila Suhu Melebihi Ambang Batas Tertentu dan Memberitahu Pengguna Melalui Email”

# Deskripsi

Thingsboard merupakan sebuah IoT dashboard dan semua IoT Dashboard dibuat menggunakan widget Thingsboard yang terdapat pada Widget Library. Setiap widget menyediakan end-user function seperti Data Visualization, Kontrol Perangkat Jarak Jauh, Manajemen Alarm, dan Menampilkan Konten HTML yang Statis.

Hands on ini terdiri dari bagian-bagian berikut :

* Bagian 1 : Input Data Menggunakan CURL
* Bagian 2 : Rule Engine
* Bagian 3 : Pembacaan cuaca dengan pemanggilan REST API
* Bagian 4 : Contoh penerapan pada kasus “Pengumpulan dan Visualisasi Data Termometer di Dalam Gedung dan Menaikkan Alarm Apabila Suhu Melebihi Ambang Batas Tertentu dan Memberitahu Pengguna Melalui Email”

# Langkah Pengerjaan

**Bagian 1 – Input Data Menggunakan CURL**

**CURL** adalah sebuah program dan library untuk mengirim dan mengambil data melalui URL. Untuk menyederhanakan panduan ini, kami akan memasukkan data menggunakan protokol HTTP dari PC lokal Anda.

**Instalasi Library Sisi Klien**

Instal klien HTTP (cURL) menggunakan perintah berikut :

Untuk ubuntu : sudo apt-get install curl

**Contoh Perintah CURL**

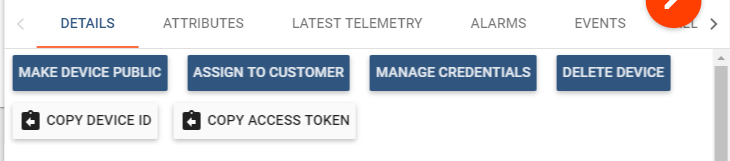
Misalkan kita akan mempush data temperatur sebesar 25 pada Thingsboard, maka perintah yang akan dijalankan sebagai berikut :

curl -v -X POST -d “{\”temperature\”:25}” [<Host\_name>/api/v1/](https://demo.thingsboard.io/api/v1/)<Access\_Token> /telemetry –header “Content-Type:application/json”

Keterangan :

Host name pada kasus ini : [https://demo.thingsboard.io](https://demo.thingsboard.io/)

Untuk akses token bisa kita lihat pada detail perangkat (devices) dan klik copy access token



**Bagian 2 – Rule Engine**

Rule Engine adalah kerangka kerja yang mudah digunakan untuk membangun alur kerja berbasis event. Terdapat 3 komponen utama pada Rule Engine, yaitu :

* Message

Event apa pun yang masuk. Itu bisa berupa data yang masuk dari perangkat, event siklus hidup perangkat, event REST API, permintaan RPC, dll.

* Rule Node

Fungsi yang dijalankan pada pesan yang masuk. Ada banyak jenis Node yang dapat memfilter, mengubah atau menjalankan beberapa tindakan pada Pesan yang masuk.

* Rule Chain

Node terhubung satu sama lain dengan relasi, sehingga pesan keluar dari node aturan dikirim ke node aturan terhubung berikutnya.

**Kasus Penggunaan Umum**

ThingsBoard Rule Engine adalah kerangka kerja yang sangat dapat disesuaikan untuk pemrosesan event yang kompleks. Berikut adalah beberapa kasus penggunaan umum yang dapat dikonfigurasikan melalui ThingsBoard Rule Chain :

* Validasi dan modifikasi data untuk telemetri atau atribut yang masuk sebelum disimpan ke database.
* Salin telemetri atau atribut dari perangkat ke aset terkait sehingga Anda dapat mengagregasi telemetri. Misalnya data dari beberapa perangkat dapat dikumpulkan dalam Aset terkait.
* Buat / Perbarui / Hapus alarm berdasarkan kondisi yang ditentukan.
* Tindakan pemicu berdasarkan peristiwa siklus hidup perangkat. Misalnya, buat lansiran jika Perangkat Online / Offline.
* Muat data tambahan yang diperlukan untuk diproses. Misalnya, muat nilai ambang suhu untuk perangkat yang didefinisikan dalam atribut Pelanggan atau Penyewa Perangkat.
* Trigger panggilan REST API ke sistem eksternal.
* Kirim email saat event kompleks terjadi dan gunakan atribut entitas lain di dalam Template Email.
* Memperhatikan preferensi Pengguna selama pemrosesan event.
* Lakukan panggilan RPC berdasarkan kondisi yang ditentukan.
* Integrasikan dengan pipelines eksternal seperti Kafka, Spark, layanan AWS, dll.

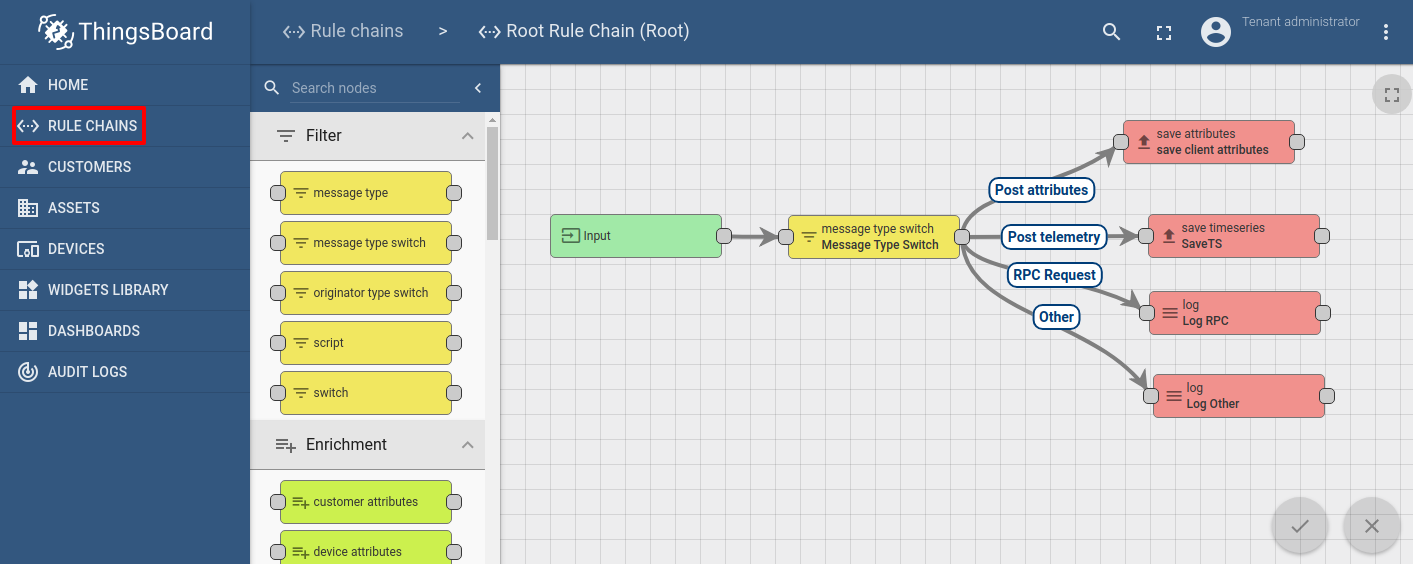
**Contoh kasus**

Anggap perangkat kalian menggunakan sensor DHT22 untuk mengumpulkan dan mempush suhu ke ThingsBoard. Sensor DHT22 dapat mengukur suhu dari -40 ° C hingga + 80 ° C.

Dalam tutorial ini kita akan mengkonfigurasi ThingsBoard Rule Engine untuk menyimpan semua suhu dalam rentang -40 hingga 80 ° C dan mencatat semua bacaan lainnya ke log sistem.

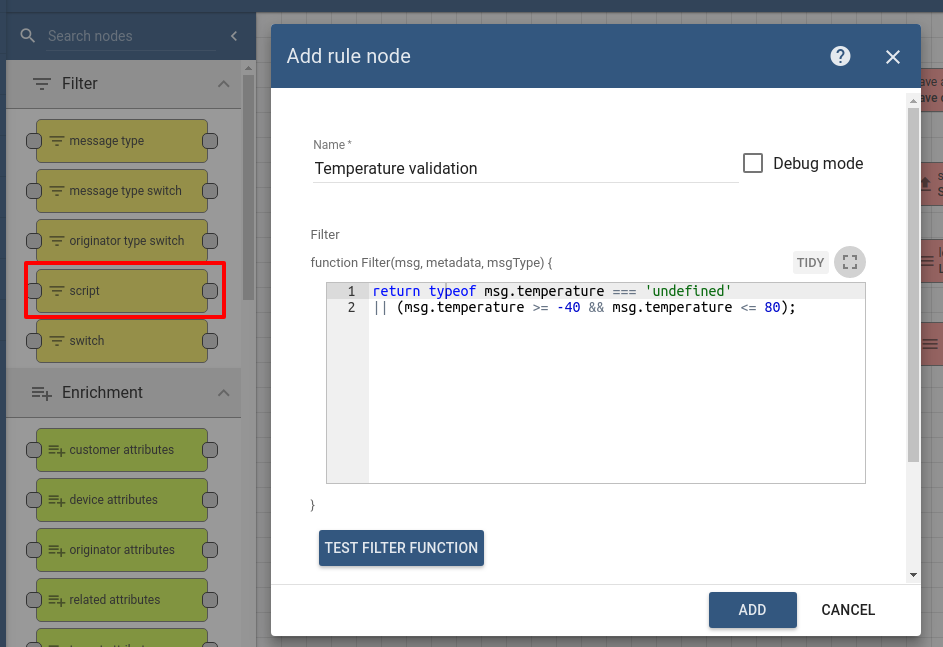
**Menambahkan Simpul Validasi Suhu**

Dalam Thingsboard UI klik bagian Rule Chains dan buka Root Rule Chain.



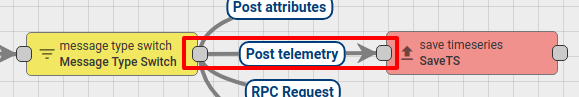
Drag dan letakkan Script Filter rule node ke rantai. Jendela konfigurasi simpul akan dibuka. Gunakan skrip ini untuk validasi data :

return typeof msg.temperature === 'undefined' || (msg.temperature >= -40 && msg.temperature <= 80);

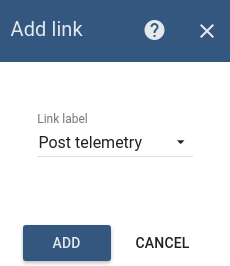


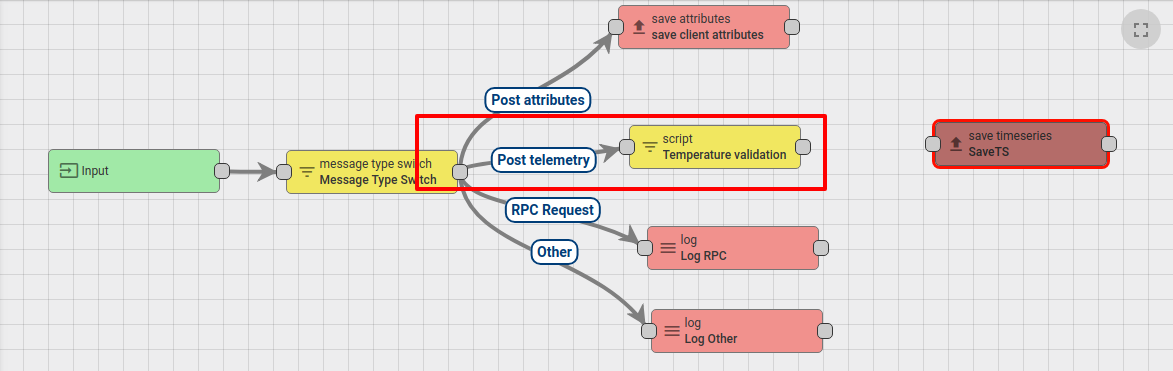
Jika properti suhu tidak ditentukan atau suhu valid - skrip akan mengembalikan True, jika tidak maka akan mengembalikan False. Jika script mengembalikan True incoming message akan dialihkan ke node berikutnya yang terhubung dengan True relation.

Sekarang kami ingin agar semua permintaan telemetri melewati skrip validasi ini. Kita perlu menghapus relasi Post Telemetry yang ada antara node Message Type Switch dan node Save Telemetry :

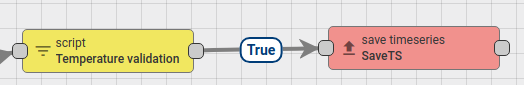


Dan hubungkan node **Message Type Switch** dengan node **Script Filter** menggunakan relasi **Post Telemetry** :

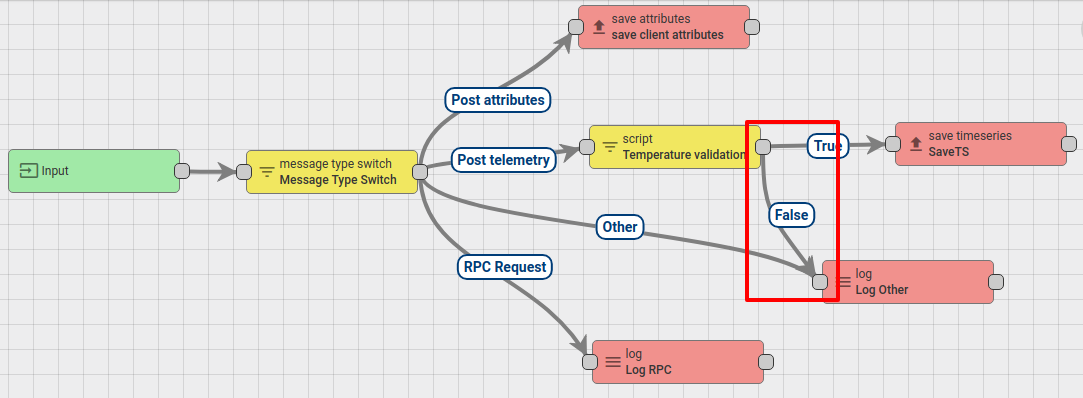




Selanjutnya, kita perlu menghubungkan node Script Filter dengan node Save Telemetry menggunakan relasi True. Jadi, semua telemetry valid akan disimpan :



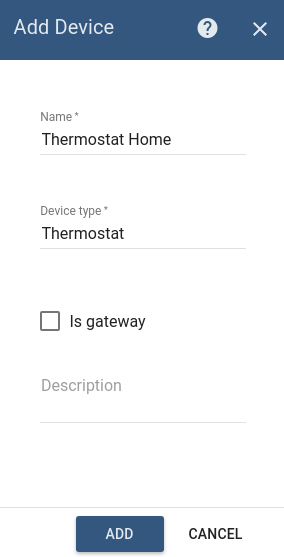
Dan juga, kita akan menghubungkan node Script Filter dengan node Log Other menggunakan relasi False. Sehingga semua telemetri yang tidak valid akan dicatat dalam log sistem :



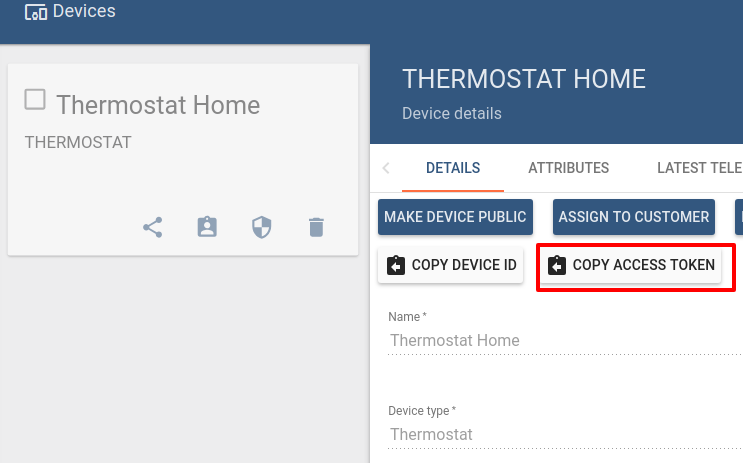
Tekan tombol simpan untuk mengaplikasikan perubahan.

**Validasi Hasil**

Untuk memvalidasi hasil, kita perlu membuat Perangkat dan mengirimkan telemetri ke Thingsboard. Jadi buka Devices dan buat Perangkat baru :



Untuk memposting perangkat telemetri, kami akan menggunakan REST API. Untuk melakukan ini, kita perlu menyalin token akses perangkat dari perangkat Thermostat Home.

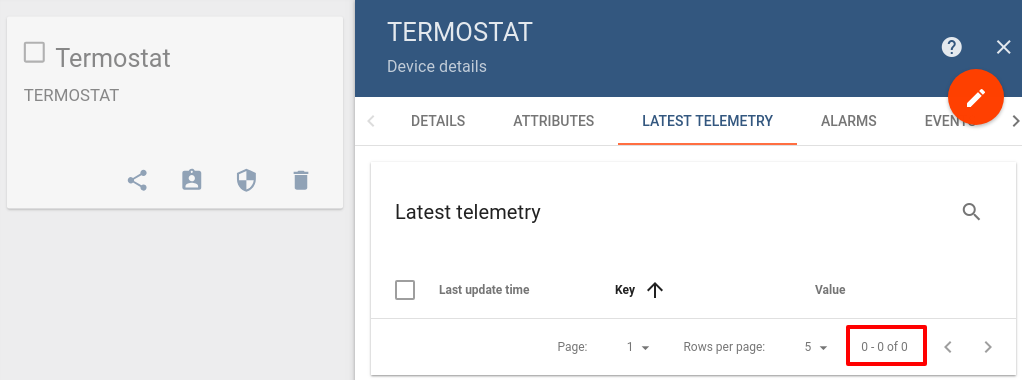


Kita buat suhu sebesar 99. Kita akan melihat bahwa telemetri tidak ditambahkan di bagian Telemetri Perangkat Terbaru:

curl -v -X POST -d '{"temperature": 99}' http: // localhost: 8080 / api / v1 / $ ACCESS\_TOKEN / telemetry - header "Content-Type: application / json"

Keterangan :

Kalian perlu mengganti $ ACCESS\_TOKEN dengan token perangkat yang sebenarnya



Kita buat suhu sebesar 24. Kita akan melihat bahwa telemetri berhasil disimpan.

curl -v -X POST -d '{"temperature": 24}' http: // localhost: 8080 / api / v1 / $ ACCESS\_TOKEN / telemetry - header "Content-Type: application / json"

Keterangan :

Kalian perlu mengganti $ ACCESS\_TOKEN dengan token perangkat yang sebenarnya

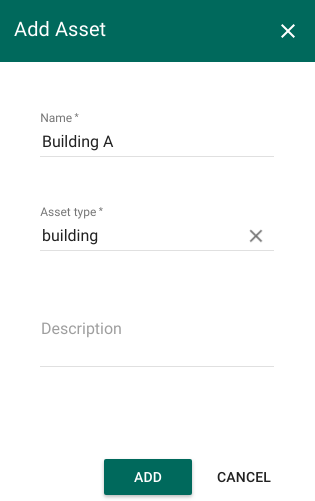
**Bagian 3 - Pembacaan cuaca dengan pemanggilan REST API**

Mari kita asumsikan kebutuhan kalian untuk mengetahui cuaca saat ini di lokasi aset. Kalian dapat menggunakan info cuaca untuk logika pemrosesan data tertentu atau hanya untuk melacak riwayat dan mengaktifkan visualisasi info ini di dashboard.

Dalam tutorial ini kita akan mengkonfigurasi ThingsBoard Rule Engine untuk secara otomatis mendapatkan informasi cuaca menggunakan REST API. Anda dapat menggunakan tutorial ini sebagai dasar untuk tugas yang lebih kompleks.

**Menambahkan Aset**

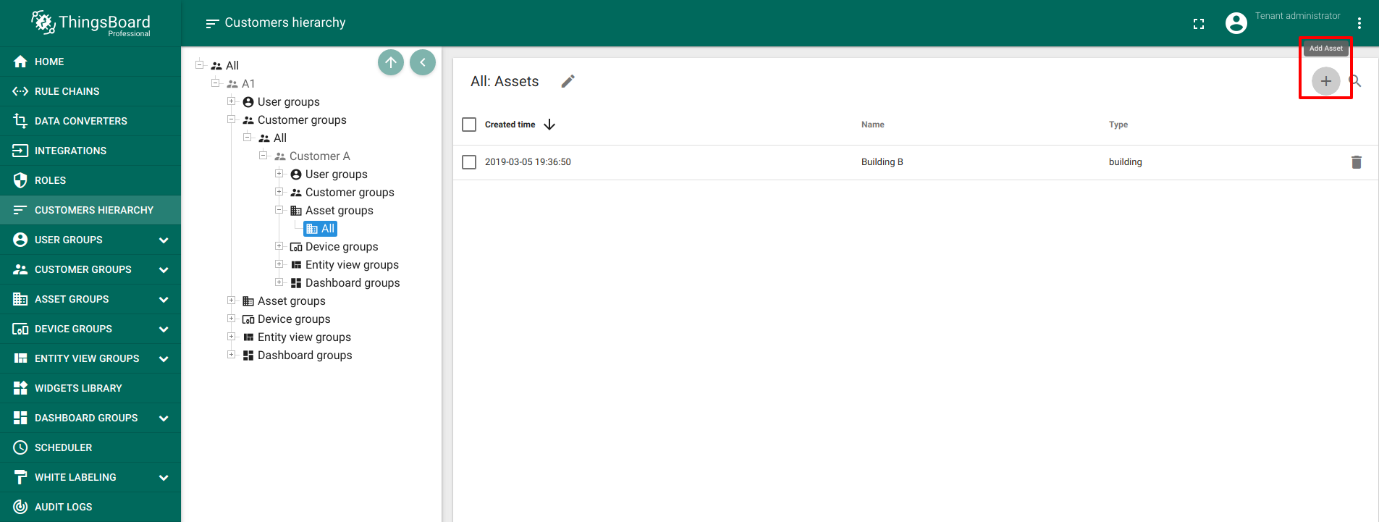
Tambahkan entitas Aset di ThingsBoard. Dengan nama “Building A” dan tipenya “Building”.



Catatan :

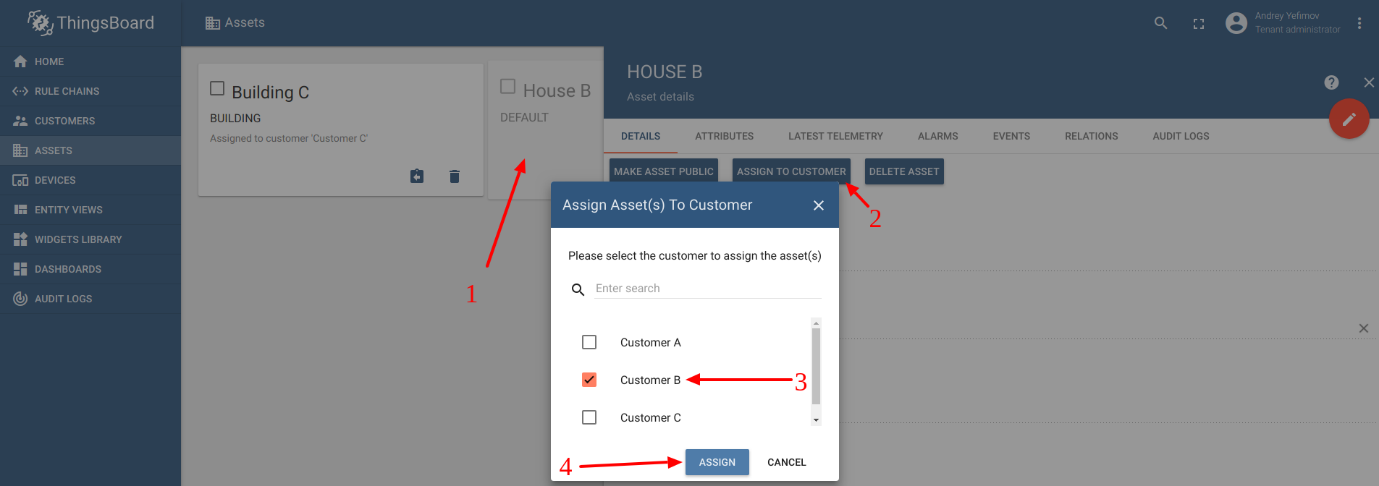
Jika Anda memiliki Edisi Profesional, Anda perlu menambahkan aset ke pelanggan menggunakan Hierarki Pelanggan dengan cara berikut:

Buka Customers Hierarchy -> All -> (Current tenant) -> Customer groups -> (Your customer group) -> (Your customer) -> Asset groups -> (Your asset group) -> Add asset



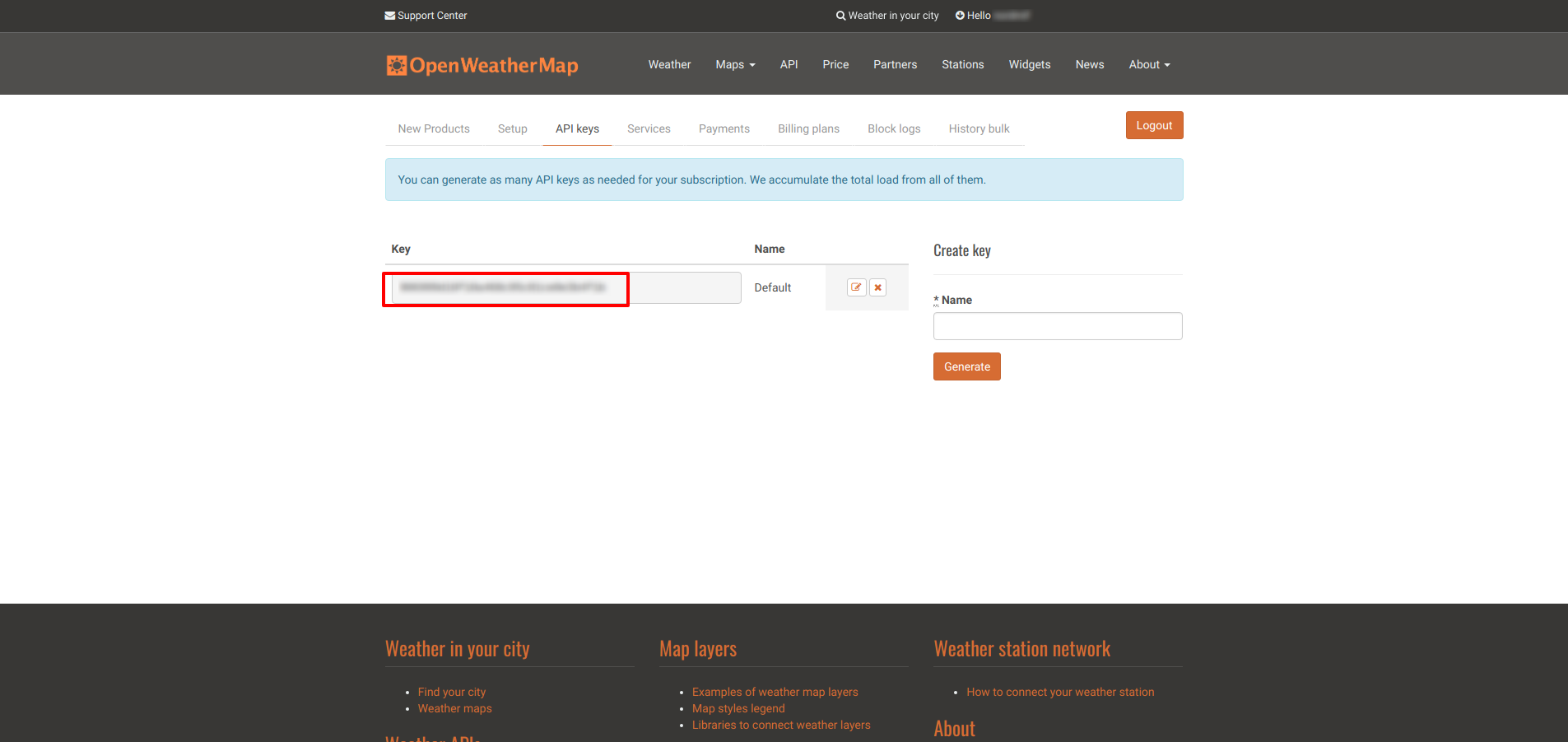
**Menetapkan Aset Kepada Pelanggan Dalam Edisi Komunitas**

Buka Assets -> Assign to customer -> (Your Customer) -> Assign



**Mendaftar di Situs Web Penyedia Data**

Untuk mendapatkan data cuaca, kalian harus mendaftar di situs web yang menyediakannya. Dalam hal ini OpenWeatherMap akan digunakan. Setelah mendaftar di sana, pergi ke halaman ini untuk mendapatkan kunci api Anda.

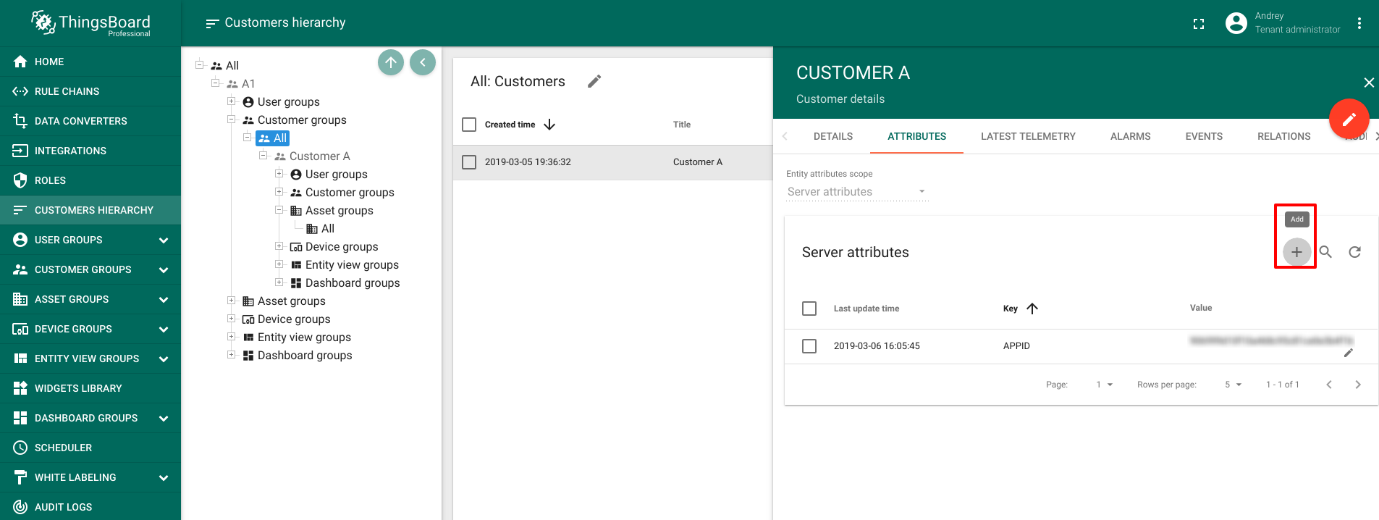


**Membuat Atribut**

Untuk melakukan panggilan REST API, kami memerlukan parameter URL berikut : Kunci API, bujur, lintang, dan satuan pengukuran. Disarankan untuk menambahkan parameter kunci API ke atribut sisi server pelanggan dan parameter lain ke atribut sisi server aset.

Atribut pelanggan akan terlihat seperti ini :

(Assigned customer) -> Attributes -> Add



Tambahkan atribut seperti berikut :

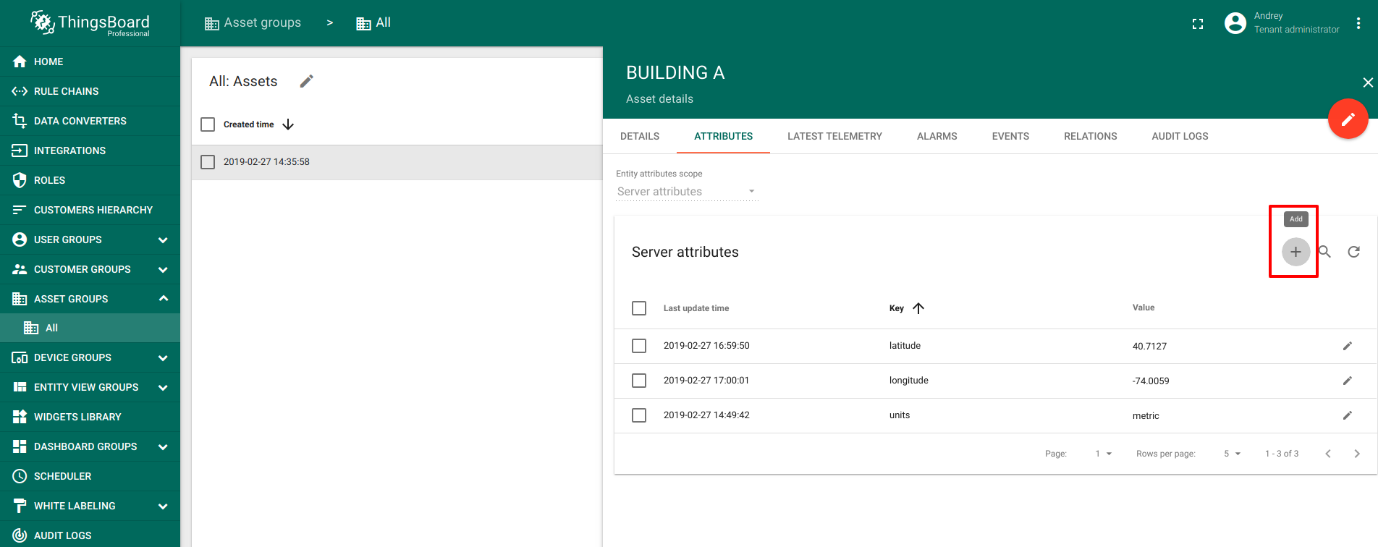
Field : APPID

Data Type : String

Input Data : (an API key you got from OpenWeatherMap)

Atribut aset akan terlihat seperti ini :

Buka Building A -> Attributes -> Add

****

Isi atribut dengan input data yang ditunjukkan pada tabel berikut :

| **Field** | **Data Type** | **Input Data** |
| --- | --- | --- |
| latitude | Double | latitude of an asset |
| longitude | Double | longitude of an asset |
| units | String | "metric" for meters per second wind speed and Celsius temperature, "imperial" for miles per hour wind speed and Fahrenheit temperature, empty for meters per second wind speed and Kelvin temperature |

Dalam contoh ini, koordinat New York City dan satuan metrik akan digunakan.

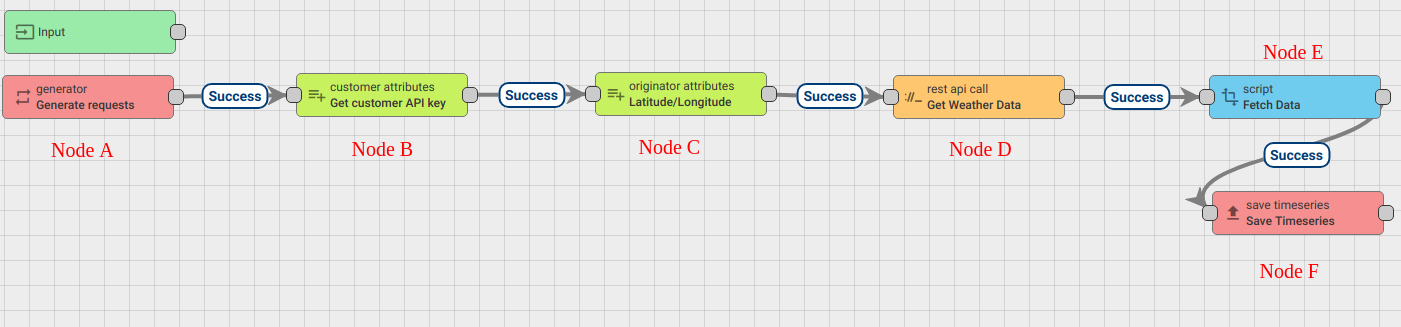
**Aliran Pesan**

Di bagian ini, kami menjelaskan tujuan setiap node dalam tutorial ini. Akan ada satu Rule Chain yang terlibat:

* Suhu Luar / Kelembaban

Rule Chain mengirim panggilan API ke OpenWeatherMap setiap 15 detik dan mengirimkan data tentang kelembaban dan suhu ke aset yang dipilih.

Tangkapan layar berikut menunjukkan bagaimana seharusnya Rule Chain di atas:

****

**Buat Rule Chain Baru (Suhu Luar / Kelembaban)**

Buka Rule Chains -> Add new Rule Chain

Konfigurasi:

* Nama : Outside Temperature / Humidity

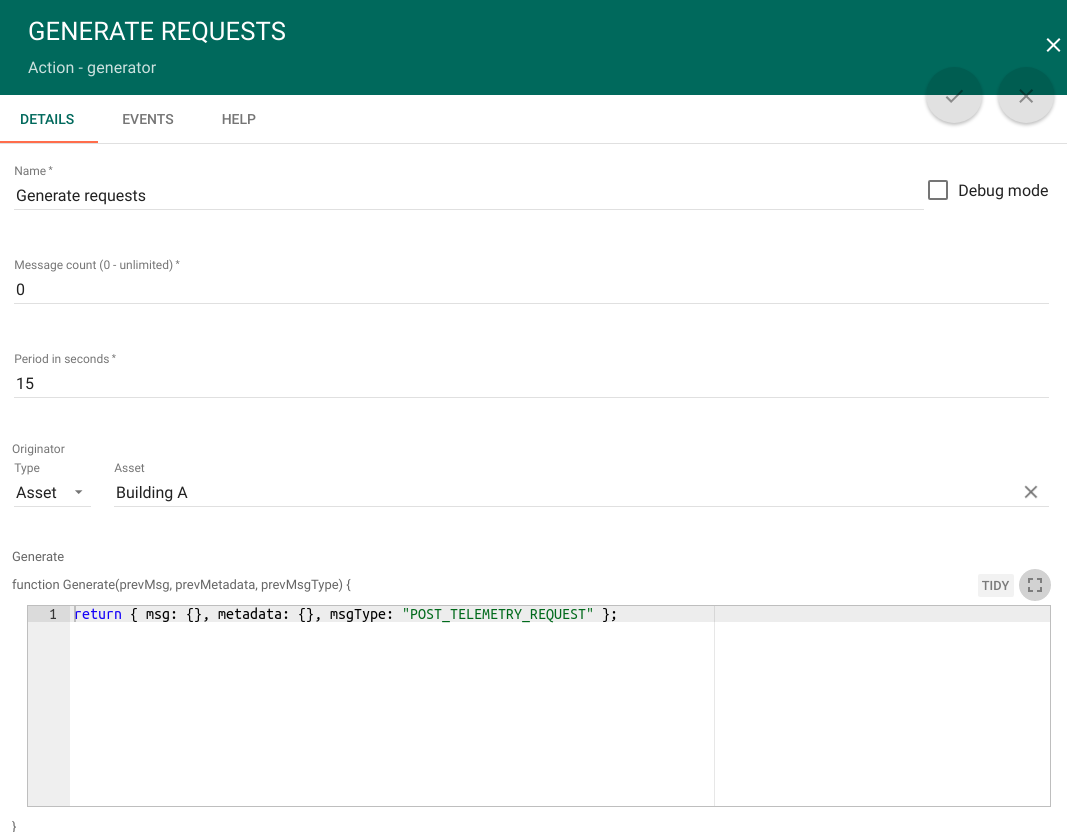
Rule Chain Baru dibuat. Tekan tombol Edit dan konfigurasikan Chain.

**Menambahkan Node yang Diperlukan**

Dalam Rule Chain ini, kalian akan membuat 5 node dan akan dijelaskan sebagai berikut:

* Node A : Node generator

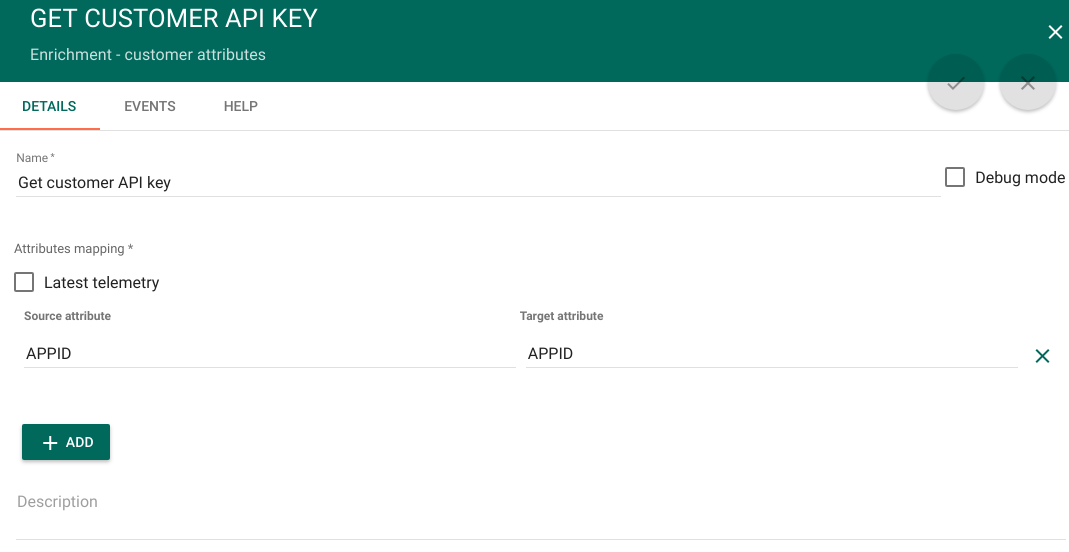
Tambahkan node Generator. Node aturan ini Menghasilkan pesan kosong untuk memicu panggilan REST API. Isi sebagai berikut:



* Node B : Node pengayaan atribut pelanggan

Tambahkan node atribut Pelanggan dan hubungkan ke simpul Generator dengan tipe relasi Sukses. Node ini akan menempatkan atribut pelanggan APPID ke dalam metadata pesan.

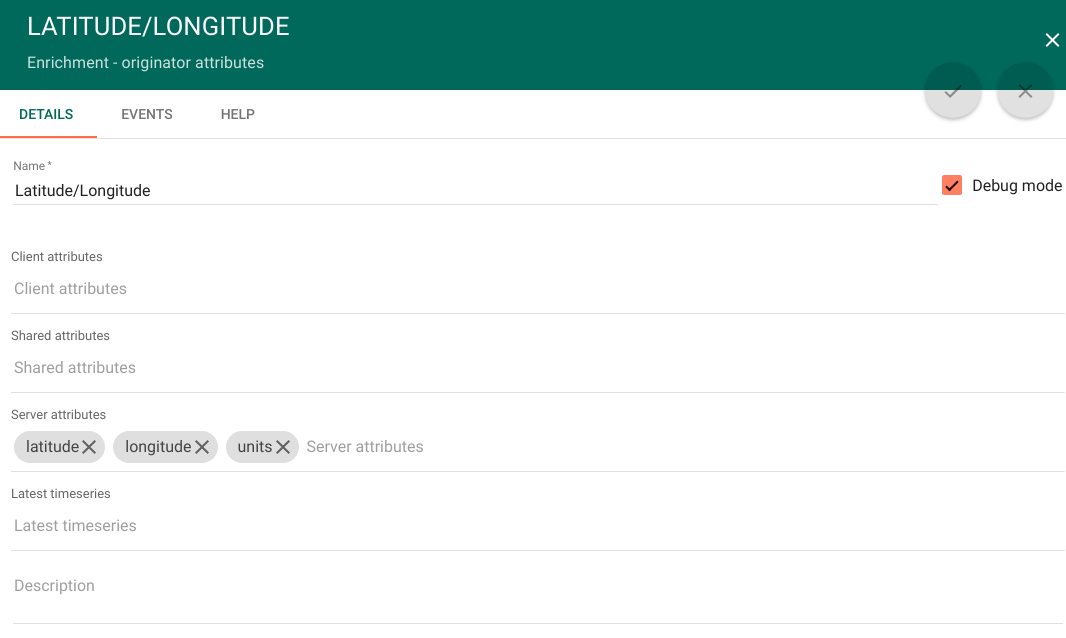
Isi sebagai berikut :



* Node C : Node pengayaan atribut originator

Tambahkan node pengayaan atribut Originator dan hubungkan ke node atribut Pelanggan dengan tipe relasi Success. Node ini akan mengambil atribut server lintang, bujur dan unit pencetus yang diatur dalam node Generator ke dalam metadata

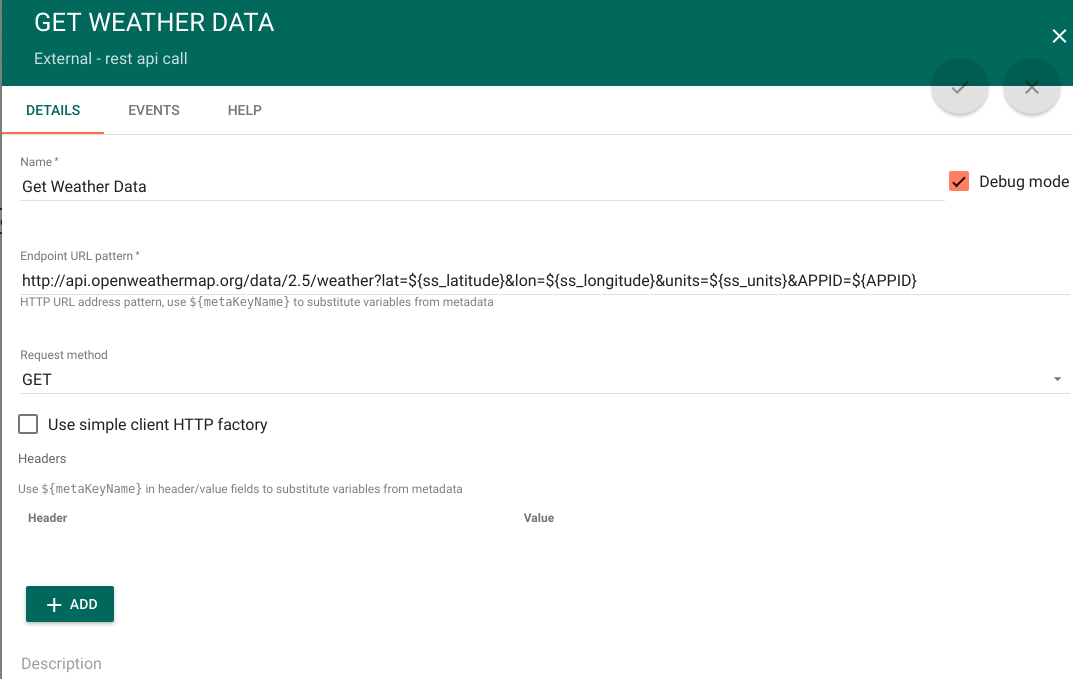
Isi sebagai berikut:



* Node D : Node panggilan REST API eksternal

Tambahkan node panggilan REST API Eksternal dan hubungkan ke node pengayaan atribut Originator dengan tipe relasi Success. Node ini akan melakukan panggilan REST API ke OpenWeatherMap.

Isi sebagai berikut:



ss\_latitude, ss\_longitude, ss\_units, ss\_APPID adalah atribut server yang diambil dari metadata yang diletakkan di sana oleh node pengayaan atribut originator

* Node E : Node transformasi skrip

Tambahkan node transformasi Script dan hubungkan ke node panggilan REST API Eksternal dengan tipe relasi Success. Node ini akan menempatkan suhu luar, suhu maksimal, suhu minimal dan kelembaban ke dalam pesan.

Isi fungsi Transform dengan cara berikut:

var newMsg = {

          "outsideTemp": msg.main.temp,

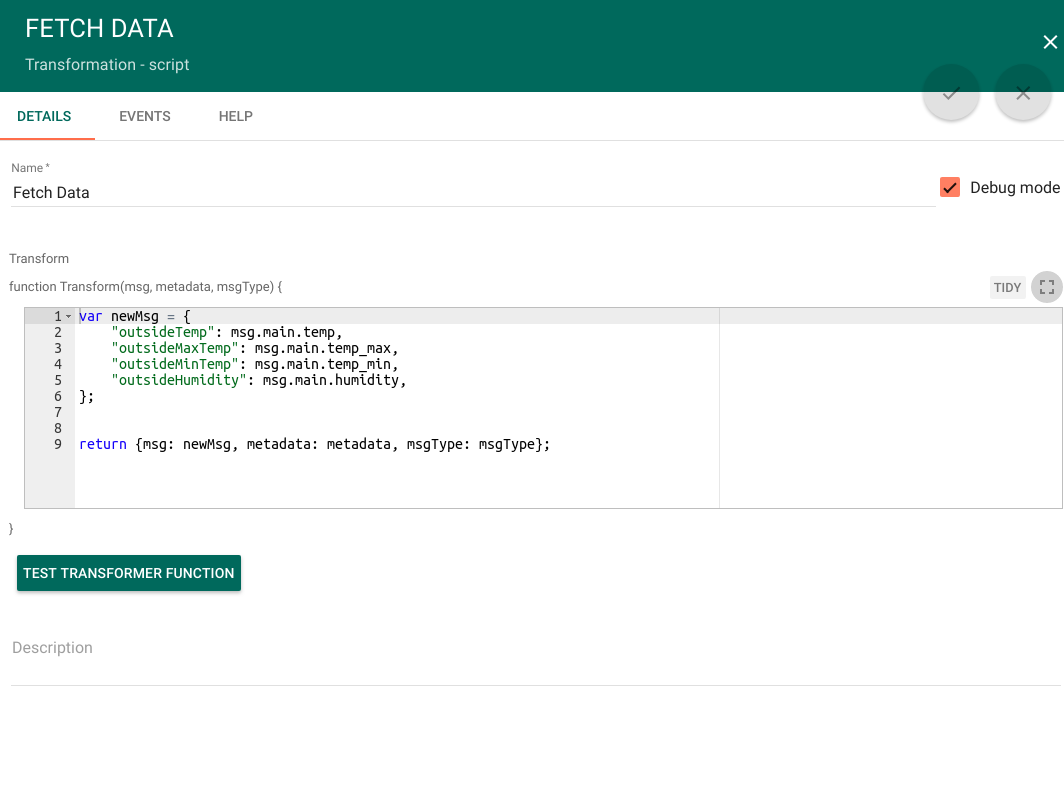
          "outsideMaxTemp": msg.main.temp\_max,

          "outsideMinTemp": msg.main.temp\_min,

          "outsideHumidity": msg.main.humidity,

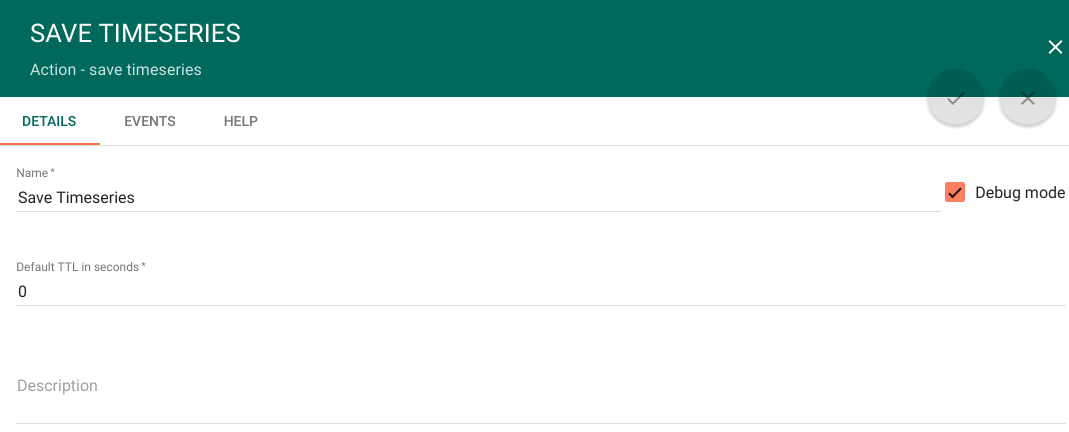
      };

return {msg: newMsg, metadata: metadata, msgType: msgType};



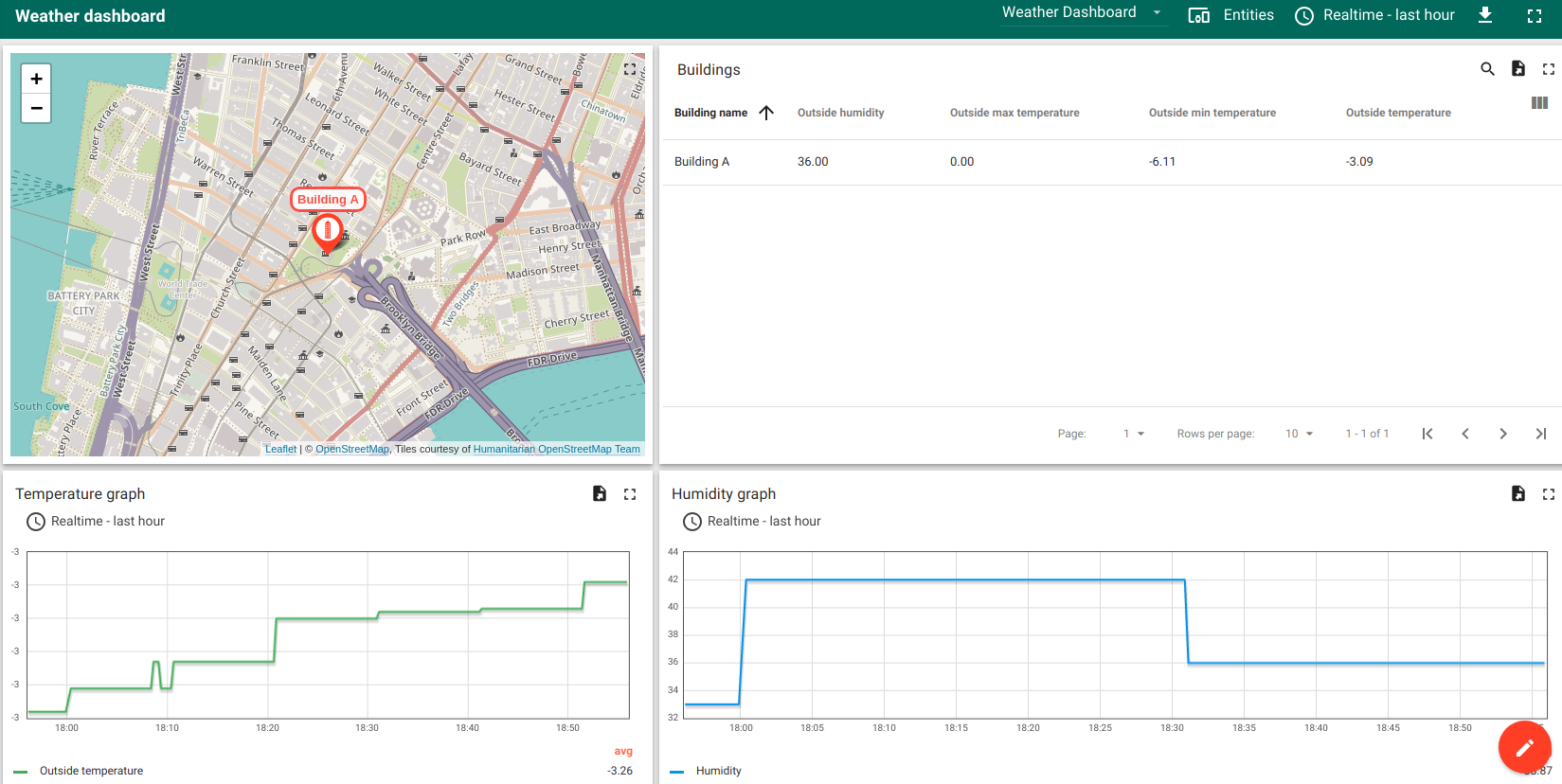
* Node F : Node Penyimpan Rentang Waktu

Tambahkan node penyimpan rentang waktu dan hubungkan ke node panggilan REST API Eksternal dengan tipe relasi Success. Node ini akan memasukkan pesan ke telemetri.



**Menyiapkan Dashboard**

Dasbor akan terlihat seperti ini :

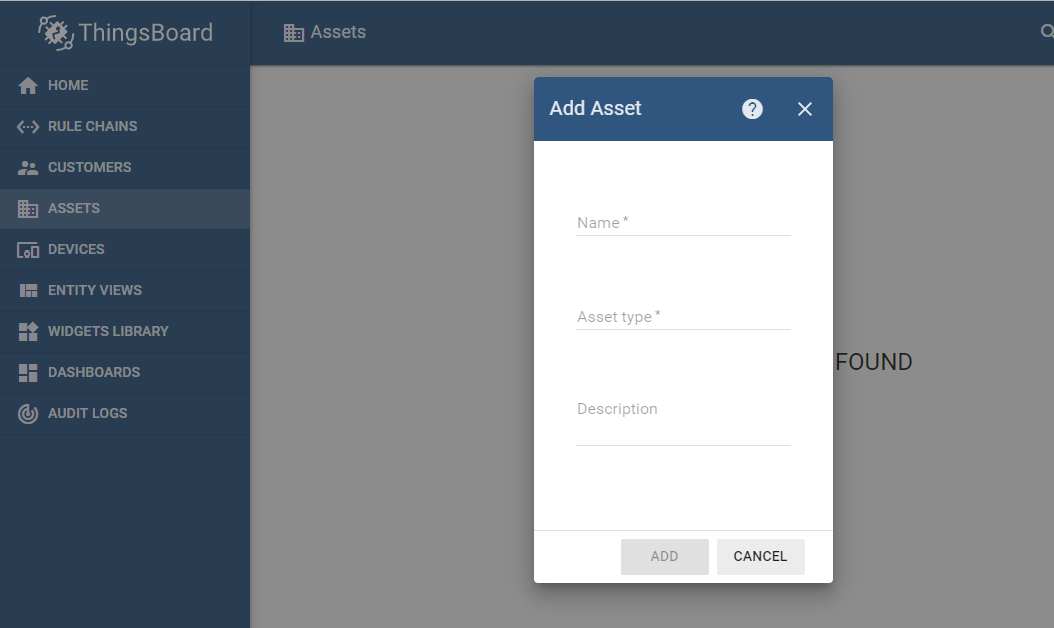


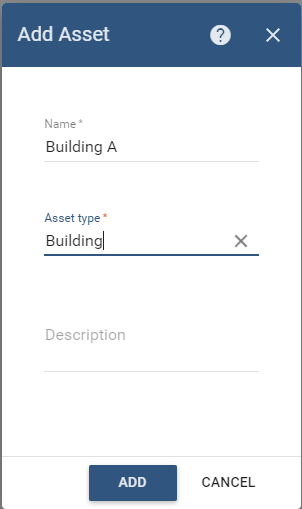
**Bagian 4 - Contoh penerapan pada kasus “Pengumpulan dan Visualisasi Data Termometer di Dalam Gedung dan Menaikkan Alarm Apabila Suhu Melebihi Ambang Batas Tertentu dan Memberitahu Pengguna Melalui Email”**

Jika tidak memiliki akses ke instance ThingsBoard yang sedang berjalan, kita dapat menggunakan fitur Live Demo.

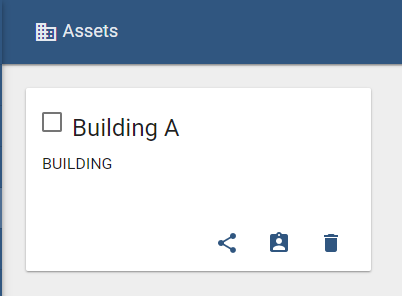
1. **Penyediaan Aset dan Perangkat**

Pertama-tama, kita ke menu assets dan tambahkan aset dengan cara mengklik tombol “+” pada pojok kanan bawah, lalu akan muncul tampilan sebagai berikut

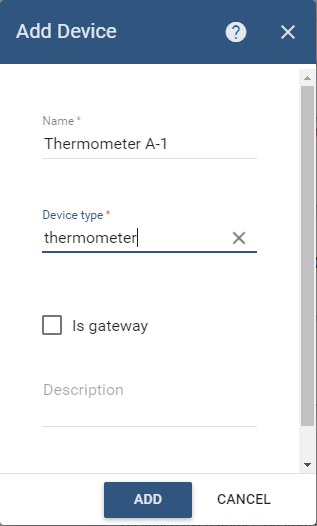
Lalu tambahkan aset “Building A” dengan tipe aset “Building”, sebagai berikut



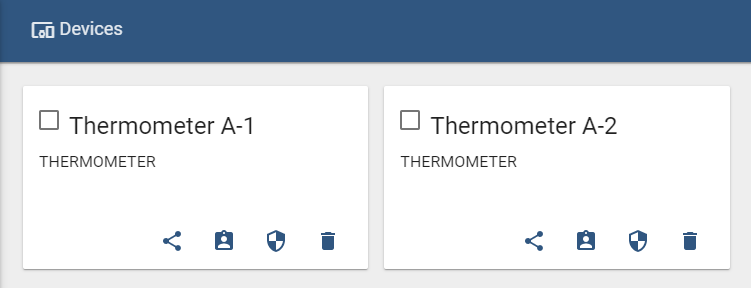
Setelah ditambahkan, “Building A” akan masuk ke dalam assets



Selanjutnya kita ke menu devices dan tambahkan perangkat “Thermometer A-1” dan “Thermometer A-2” deng tipe perangkat “Thermometer”, sebagai berikut

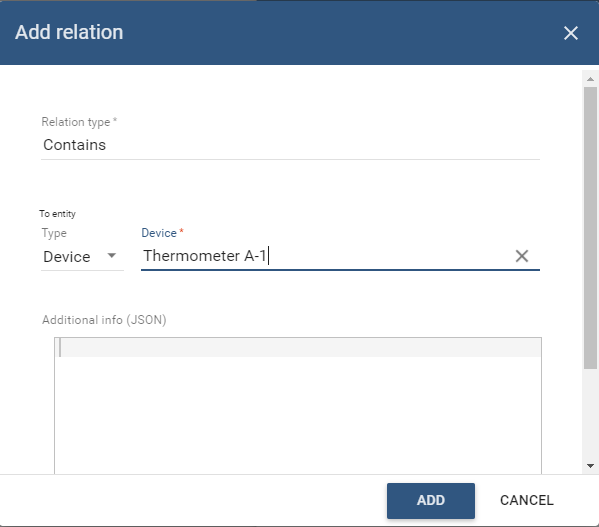


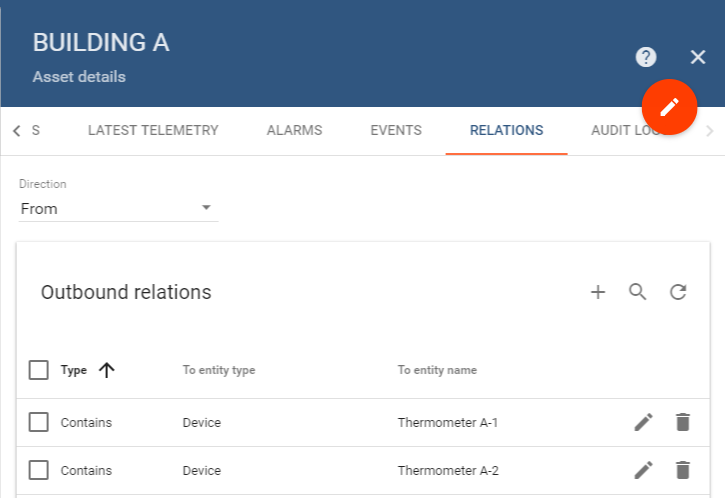
Lalu perangkat yang telah di tambahkan akan muncul di devices



1. **Mendefinisikan Relasi Antara Aset dan Perangkat**

Untuk mendefinisikan relasi antara aset dan perangkat. Pertama kita ke menu assets, kemudian klik “Building A” lalu klik relation, lalu tambahkan relasi dengan perangkat “Thermometer A-1” dan “Thermometer A-2”





1. **Push Data dari Perangkat ke Thingsboard Menggunakan CURL**

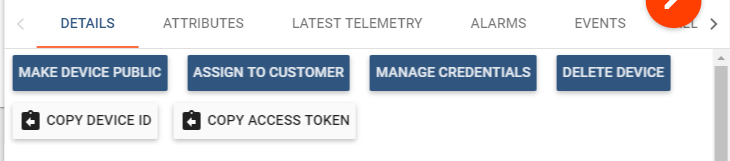
Pada contoh ini, kita akan mempush data temperatur sebesar 25. Pertama-tama kita ke menu devices, kemudian klik “Thermometer A-1”, lalu klik latest telemetry, kemudian buka terminal dan jalankan perintah berikut ini

curl -v -X POST -d “{\”temperature\”:25}” [<Host\_name>/api/v1/](https://demo.thingsboard.io/api/v1/)<Access\_Token> /telemetry –header “Content-Type:application/json”

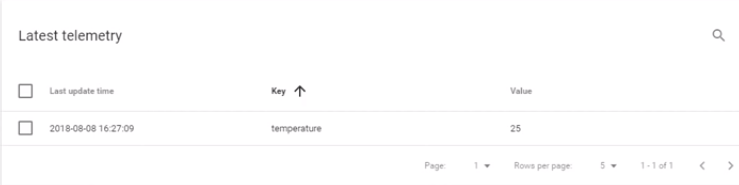
Keterangan :

Host name pada kasus ini : [https://demo.thingsboard.io](https://demo.thingsboard.io/)

Untuk akses token bisa kita lihat pada detail dan klik copy access token



Lalu datanya akan masuk di Latest Telemetry



1. **Tambahkan Rule Chain dan Rule Node**

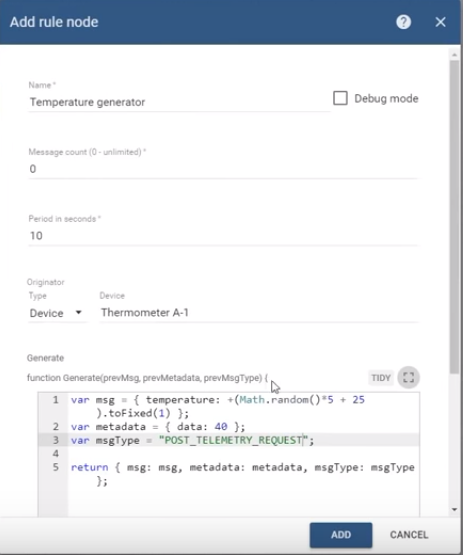
Pertama-tama ke menu Rule Chain kemudian tambahkan Rule Chain “Temperature Emulation”. Lalu pada “Temperature Emulation” tambahkan Rule Node-Generator dengan nama “Temperatur Generator”, atur periode waktunya 10 detik dan ubah script generatornya sebagai berikut

var msg = { temperature: +(Math.random()\*5 + 25).toFixed(1)};

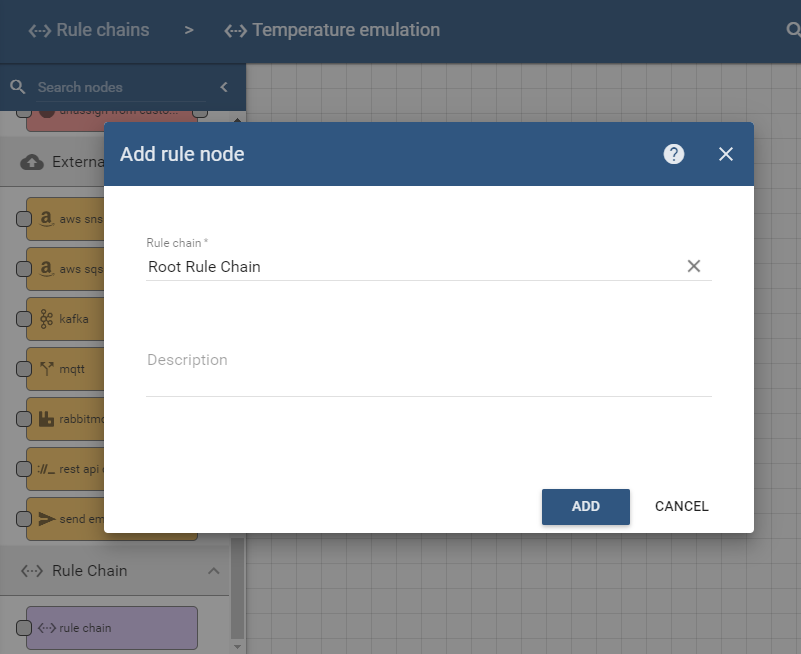
var metadata = {data:40};

var msgType = "POST\_TELEMETRY\_REQUEST";

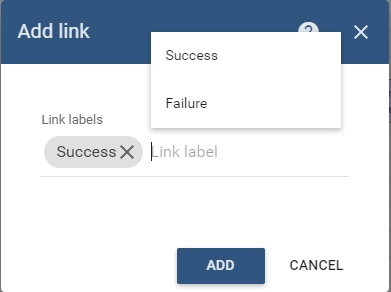
return { msg: msg, metadata: metadata, msgType: msgType };

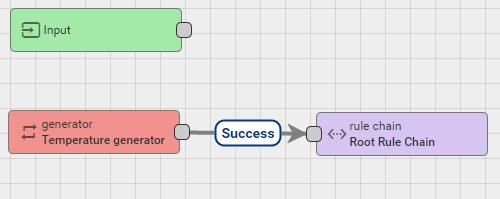


Kemudian tambahkan Root Rule Chain pada Rule Node – Rule Chain

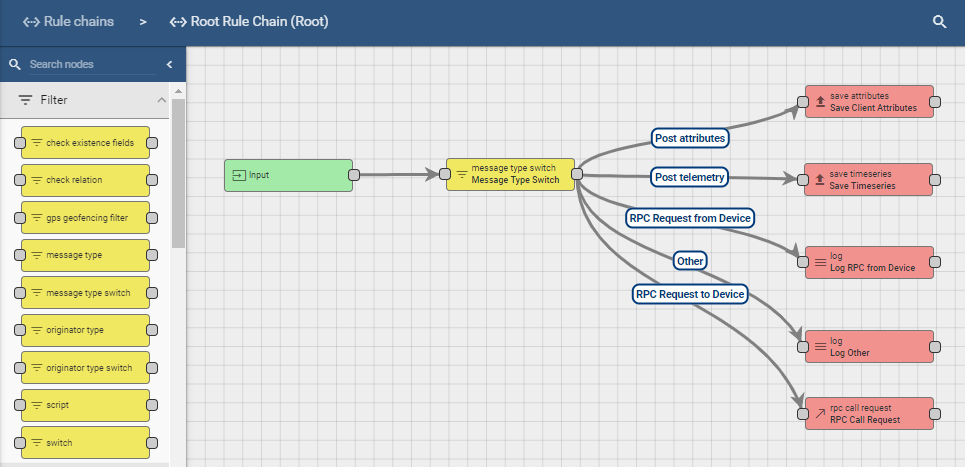


Kemudian hubungkan generator ke Rule Chain dengan hubungan relasi Success



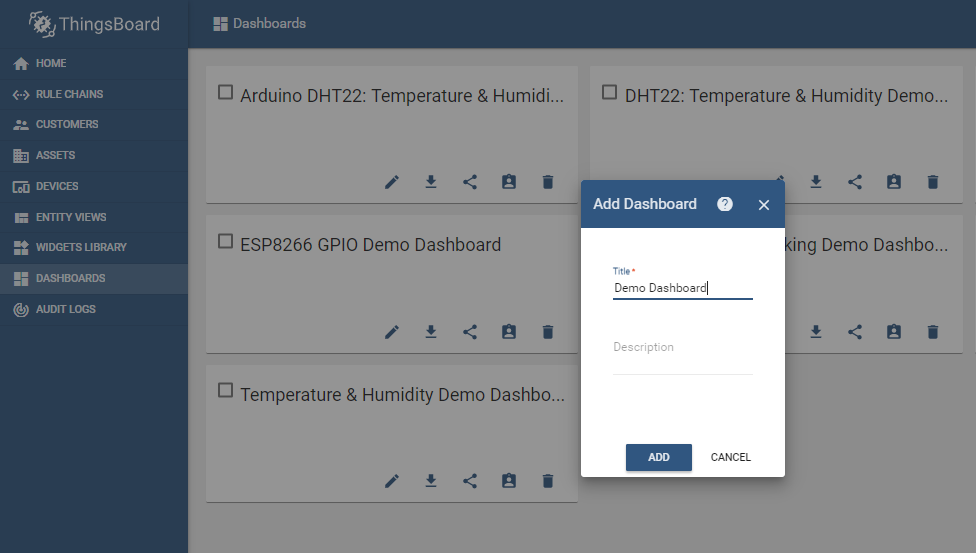


Buka Root Rule Chain pada menu Rule Chain, maka akan muncul tampilan sebagai berikut

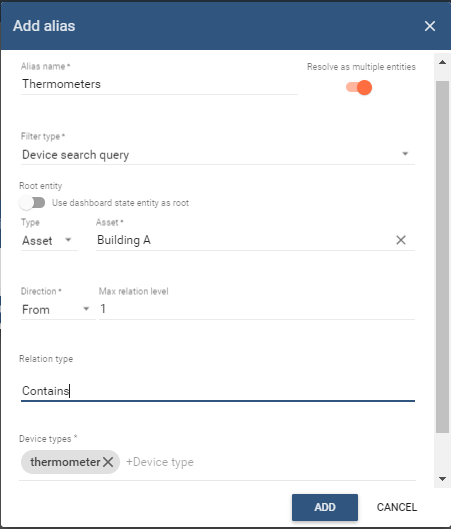


1. **Pengaturan Dashboard**

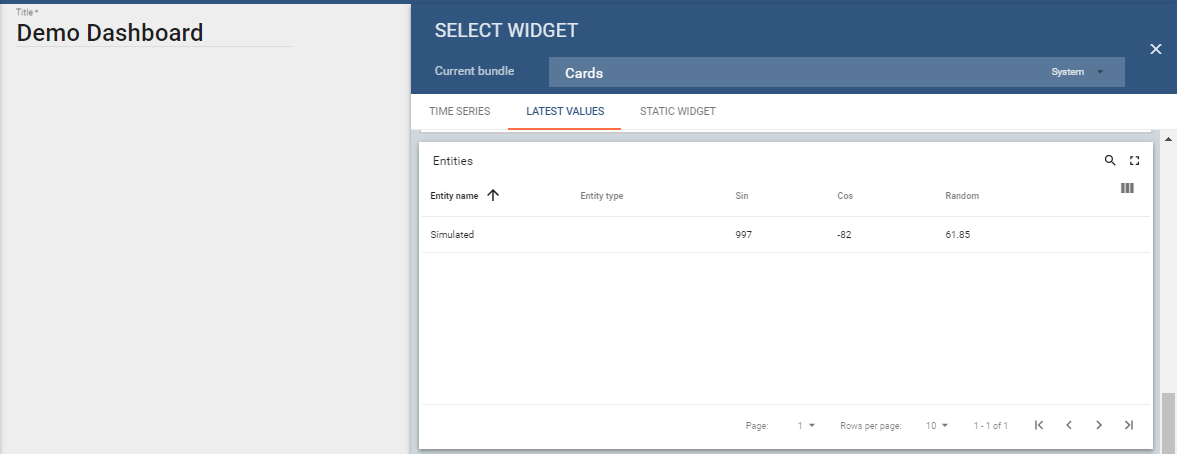
Kemudian ke menu dashboard dan tambahkan dashboard “Demo Dashboard”



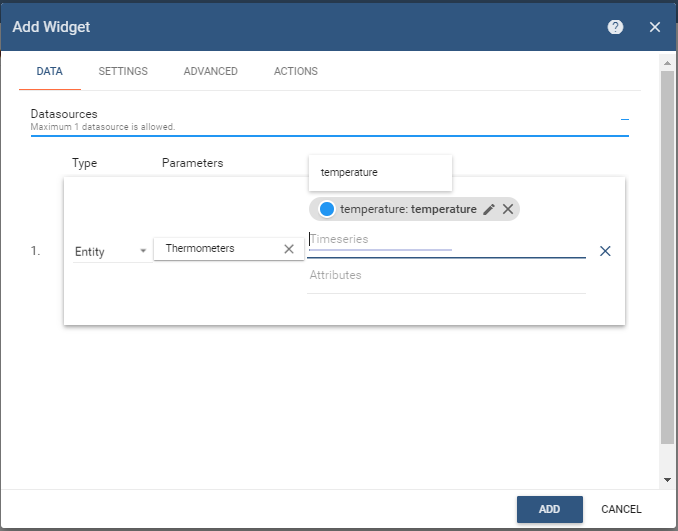
Lalu, klik “Demo Dashboard” dan klik Entity Aliases, lalu tambahkan alias dengan nama “Thermometers”, sebagai berikut

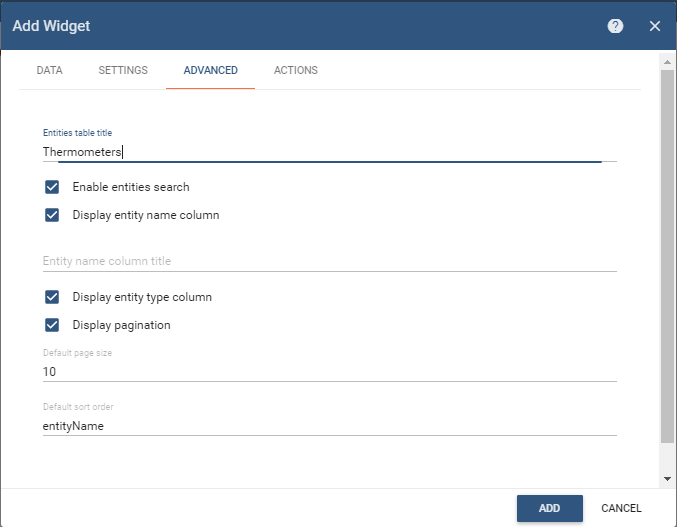


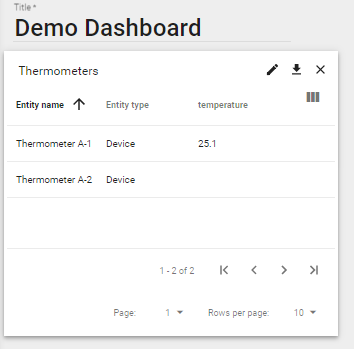
Kemudian klik Add New Widget, dan pilih widget bundle cards, lalu ke tab latest value dan klik entities



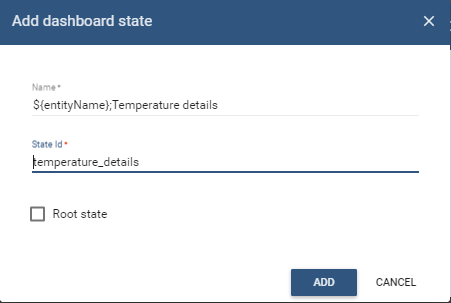
Dan tambahkan entitas sebagai berikut



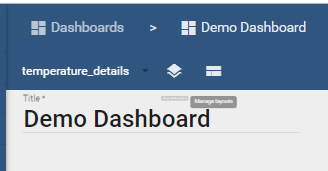




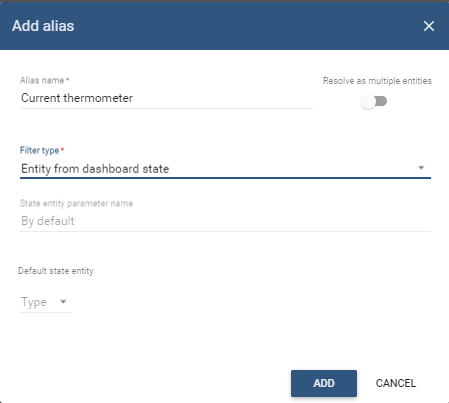
Lalu tambahkan status dashboard sebagai berikut



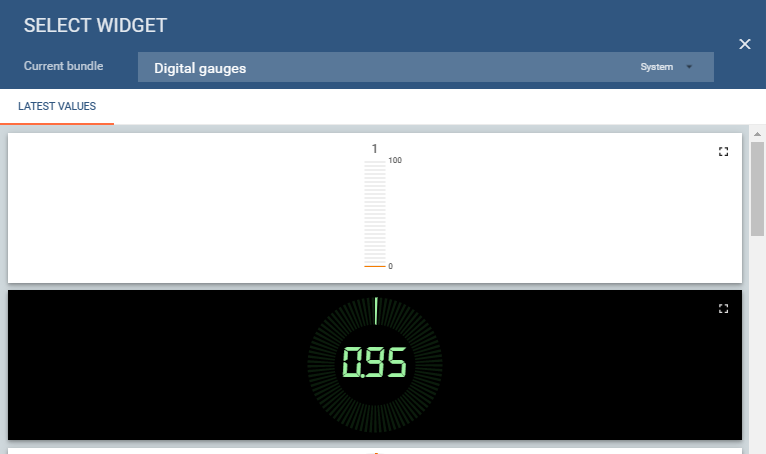
Dan ubah defaultnya menjadi temperature\_details

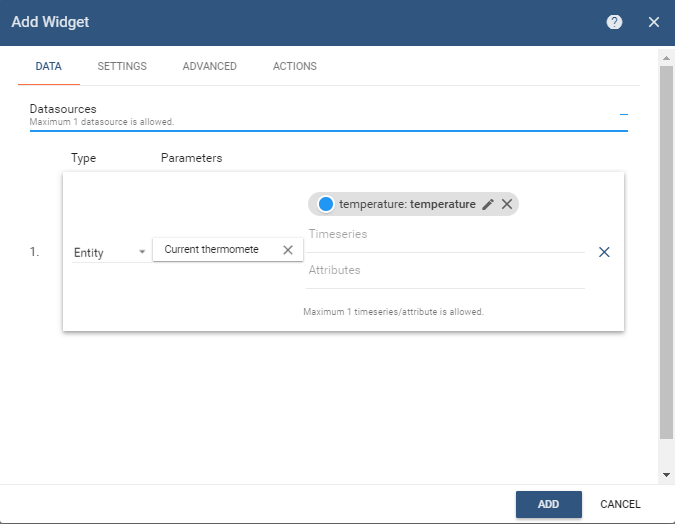


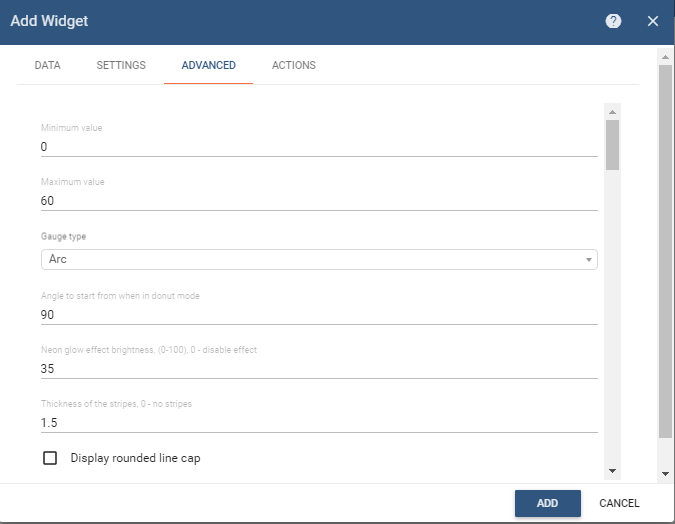
Lalu tambahkan alias dengan nama “Current thermometer” sebagai berikut

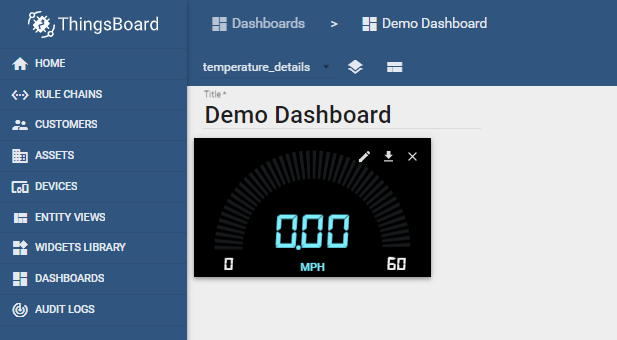


Lalu klik add new widget dan pilih widget bundle Digital Gauges, dan tambahkan widget sebagai berikut



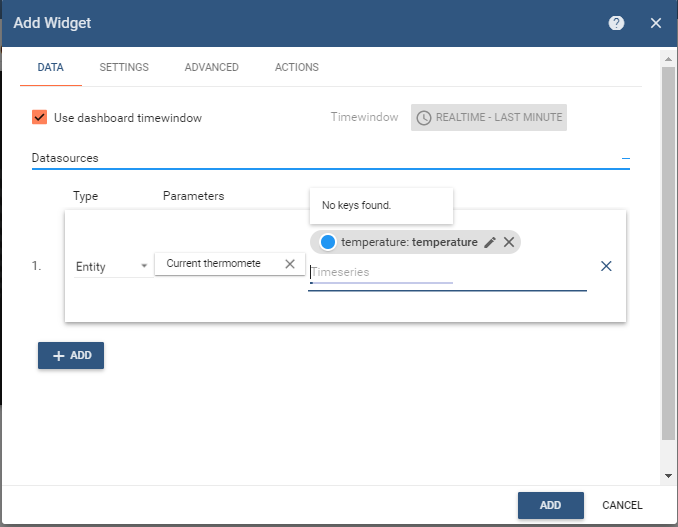


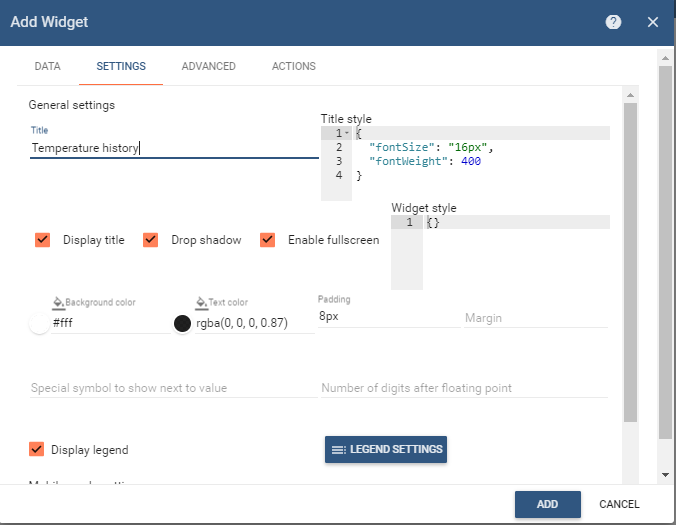


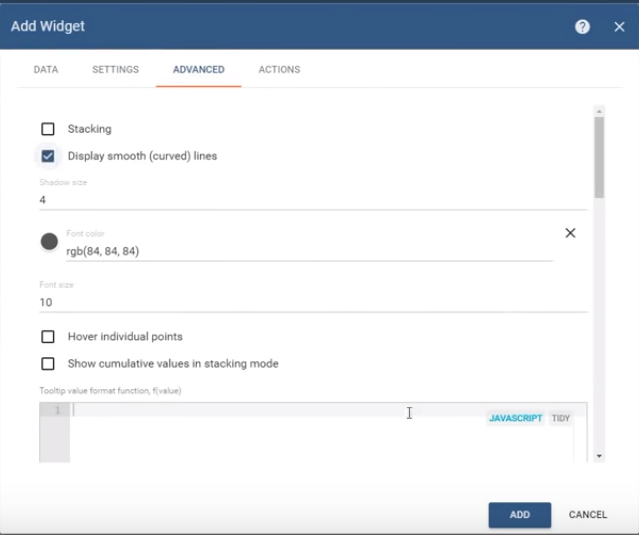


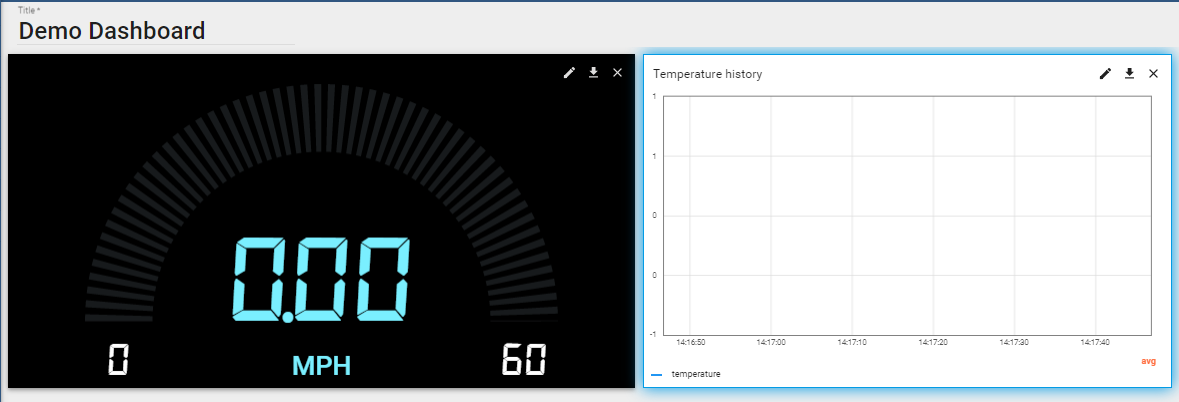
Lalu add new widget dan pilih widget bundle charts, dan tambahkan widget sebagai berikut



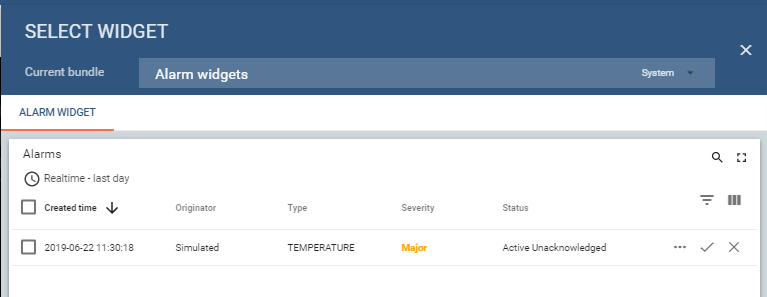


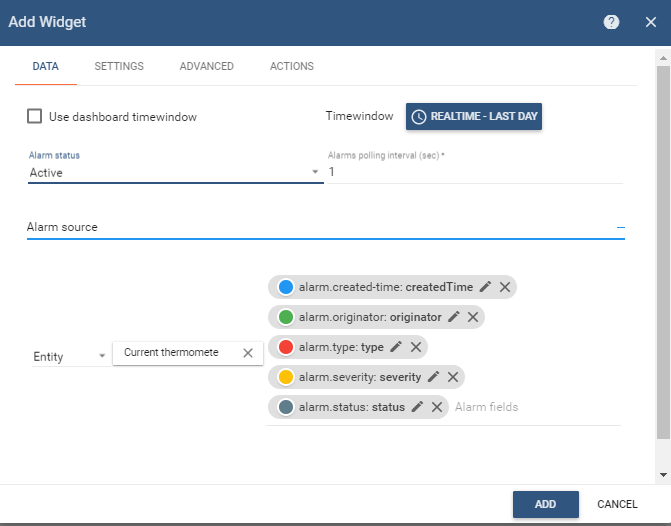


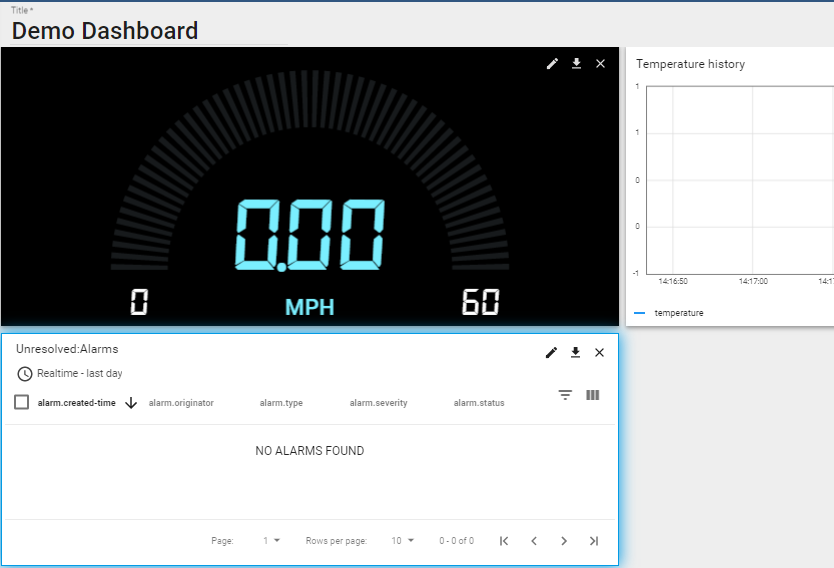




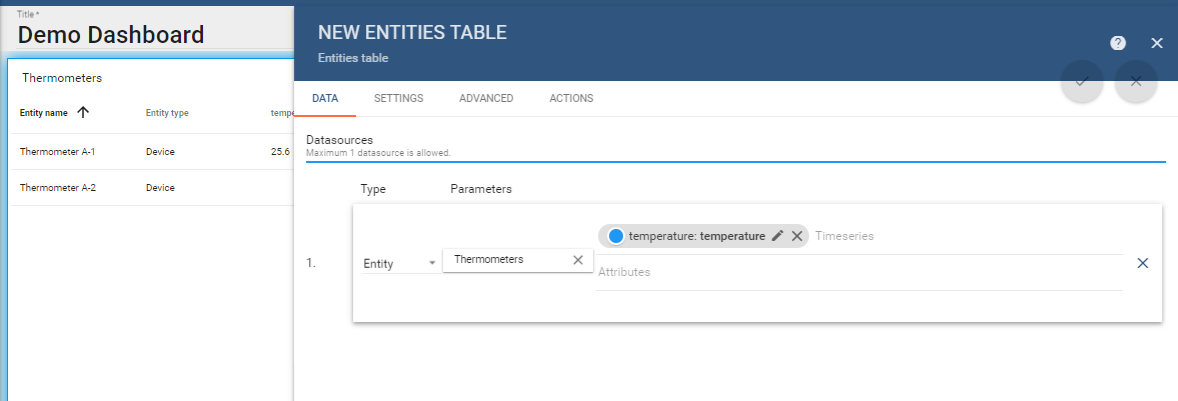
Lalu add new widget dan pilih widget bundle Alarm Widgets, dan tambahkan widget sebagai berikut

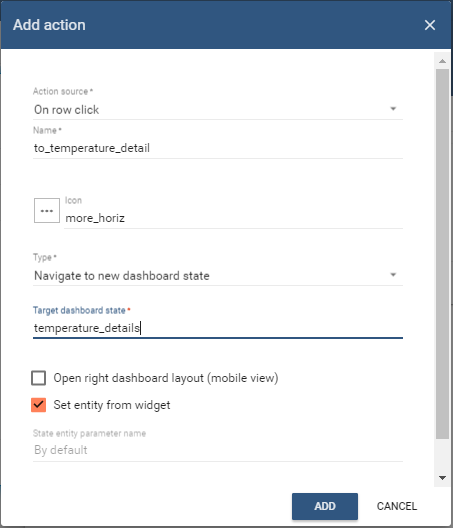




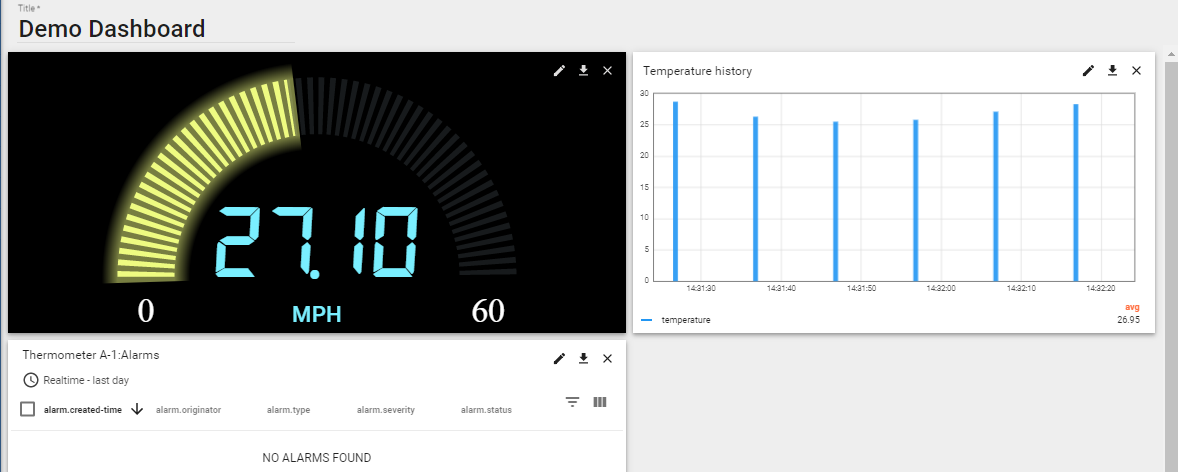


Setelah di simpan, lalu kita edit widget pertama dan tambahkan action sebagai berikut



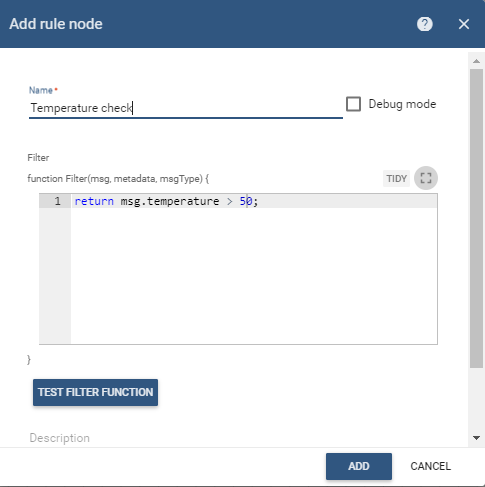


Lalu ketika kita klik Thermometer A-1 tampilannya sebagai berikut

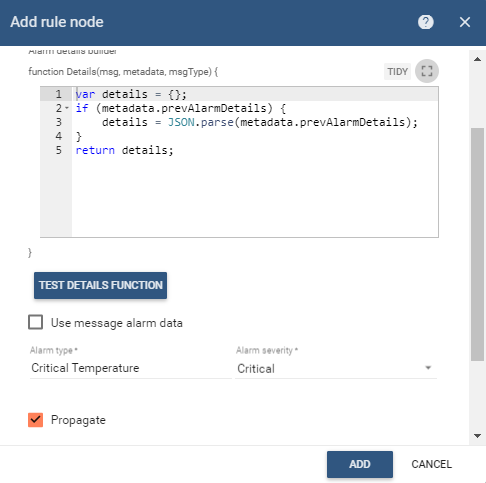


1. **Ambang Batas dan Alarm**

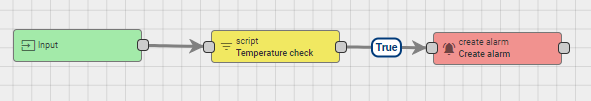
Pertama-tama kita ke menu Rule Chain, kemudian tambahkan Rule Chain dengan nama “Create Alarm”. Lalu tambahkan Rule Node Script sebagai berikut



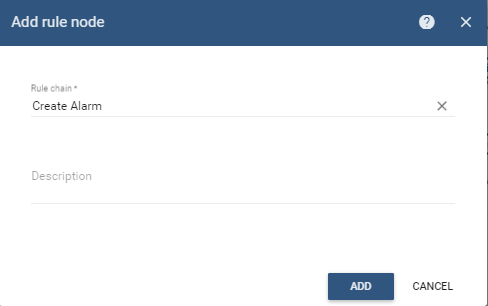
Sehingga apabila temperature lebih besar dari 50, maka alarm akan aktif. Lalu tambahkan Rule Node Creat Alarm sebagai berikut

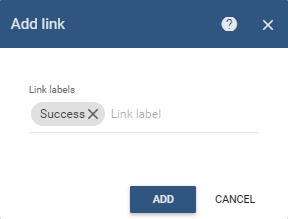


Hubungkan input ke script dan hubungkan node script dengan node creat alarm menggunakan relasi True.

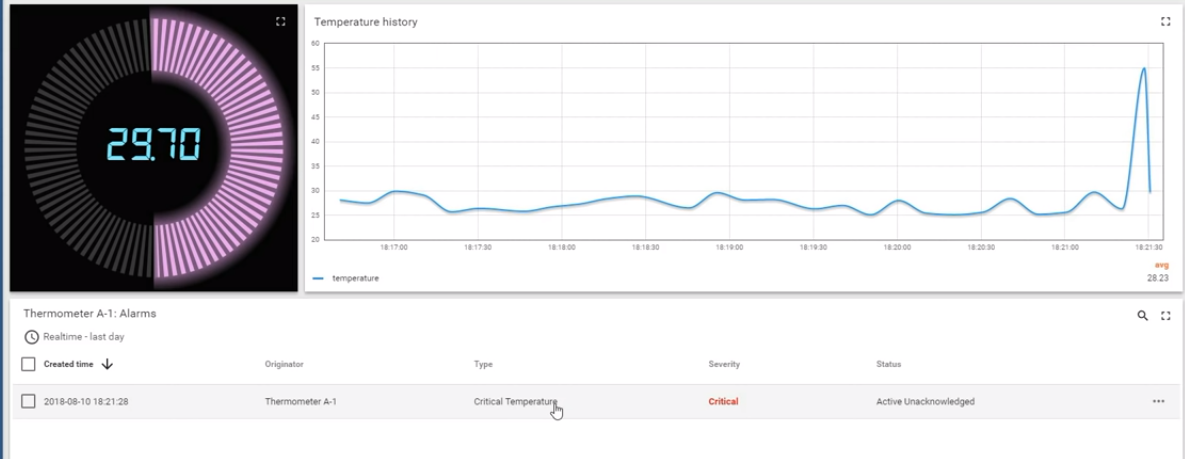


Lalu kita kembali ke Root Rule Chain, dan tambahkan node Create Alarm Rule Chain dan hubungkan node Save Timeseries dengan node Create Alarm menggunakan hubungan relasi Success sebagai berikut



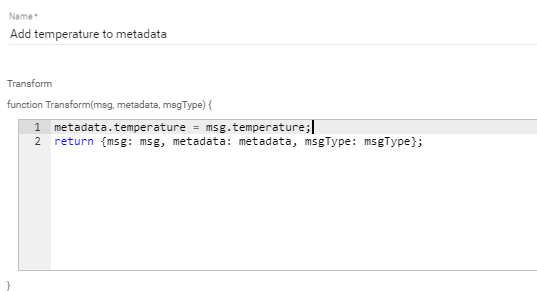


Sehingga, apabila kita memasukan data temperature di atas 50, alarm akan aktif

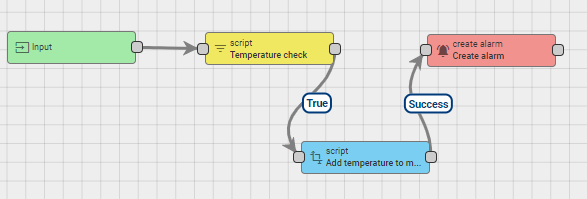


1. Notifications Over Email

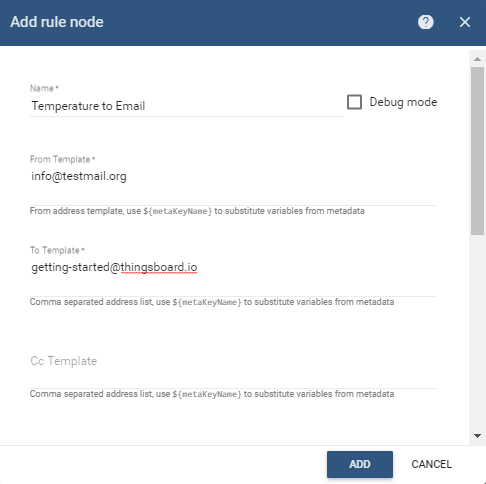
Kita modifikasi Create Alarm Rule Chain. Pertama kita hapus relasi antara node script dengan node Creat Alarm. Kemudian tambahkan node Transformation script sebagai berikut

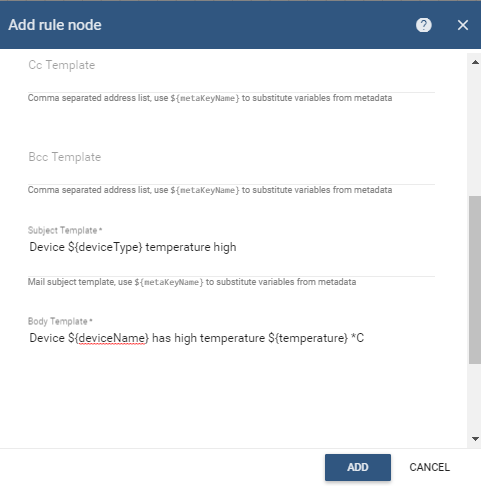


Kemudian hubungkan node Filter script dengan node Transformation script dengan hubungan relasi True. Dan hubungkan node Transformation script dengan Create alarm menggunakan hubungan relasi Success.

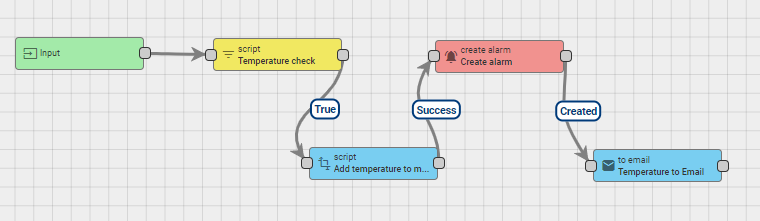


Lalu tambahkan node Transformation to email sebagai berikut

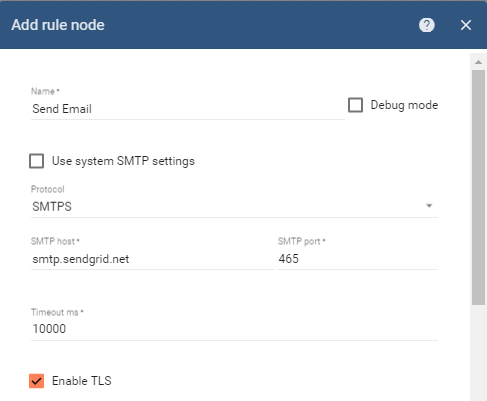


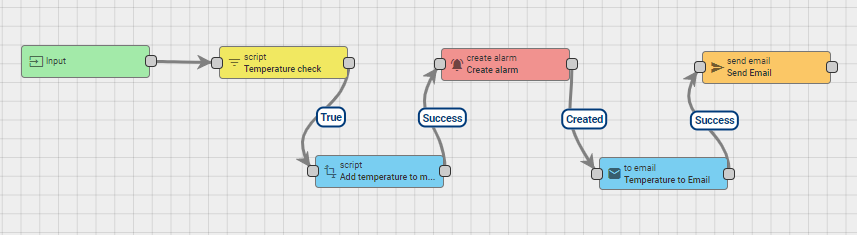


Hubungkan node Create Alarm dengan node Transformation to email dengan hubungan relasi Created.

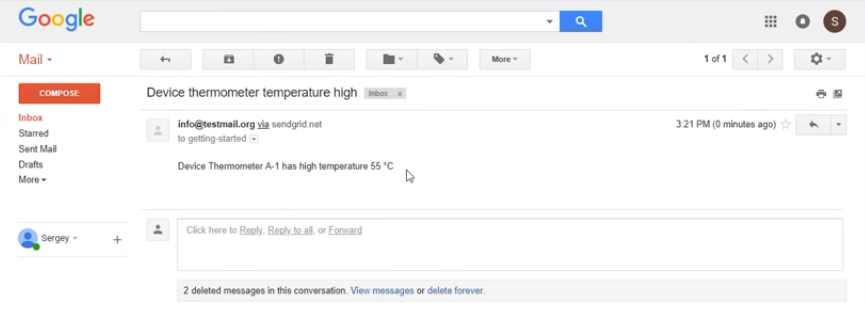


Tambahkan node send email dan hubungkan node Transformation to email dengan node Send email menggunakan hubungan relasi Success sebagai berikut





Sehingga apabila temperature lebih besar dari 50, alarm akan aktif dan akan diberikan pemberitahuan ke email [getting-started@thingsboard.io](mailto:getting-started@thingsboard.io) dari email [info@testmail.org](mailto:info@testmail.org)



# Soal latihan

# Referensi

1. <https://www.youtube.com/watch?v=TwJJbxwQX5k>. [Diakses: 22-Jun-2019].
2. <https://thingsboard.io/docs/getting-started-guides/helloworld/>. [Diakses: 22-Jun-2019].
3. <https://thingsboard.io/docs/user-guide/rule-engine-2-0/re-getting-started/>. [Diakses: 22-Jun-2019].
4. <https://thingsboard.io/docs/user-guide/rule-engine-2-0/tutorials/get-weather-using-rest-api-call/>. [Diakses: 22-Jun-2019].