

Nama : Fahrul Rozi
Nim : 09011181320022

Node-RED adalah sebuah tool berbasis browser untuk membuat aplikasi *Internet of Things* (IoT) yang mana lingkungan pemrograman visualnya mempermudah penggunaannya untuk membuat aplikasi sebagai “*flow*”. *Flow* ini terbentuk dari *node-node* yang saling berhubungan di mana tiap *node* melakukan tugas tertentu. Walaupun Node-RED didesain untuk Internet of Things (IoT), ia juga dapat digunakan untuk keperluan umum dan untuk berbagai macam jenis aplikasi.

Alasan mengapa dinamakan ‘Node’ adalah karena alat ini diimplementasikan sebagai aplikasi node tetapi dari sudut pandang konsumen yang benar hanya detail dari implementasi internal.

- Install Node-RED

Cara termudah untuk menginstal Node-RED adalah dengan menggunakan manajer paket node (NPM).

```
sudo npm install -g --unsafe-perm node-red
```

Catatan: Anda tidak harus menggunakan NPM 1.x untuk menginstal Node-RED. Anda dapat meng-upgrade ke NPM versi 2.x terbaru dengan perintah: `sudo NPM install -g npm@2.x`

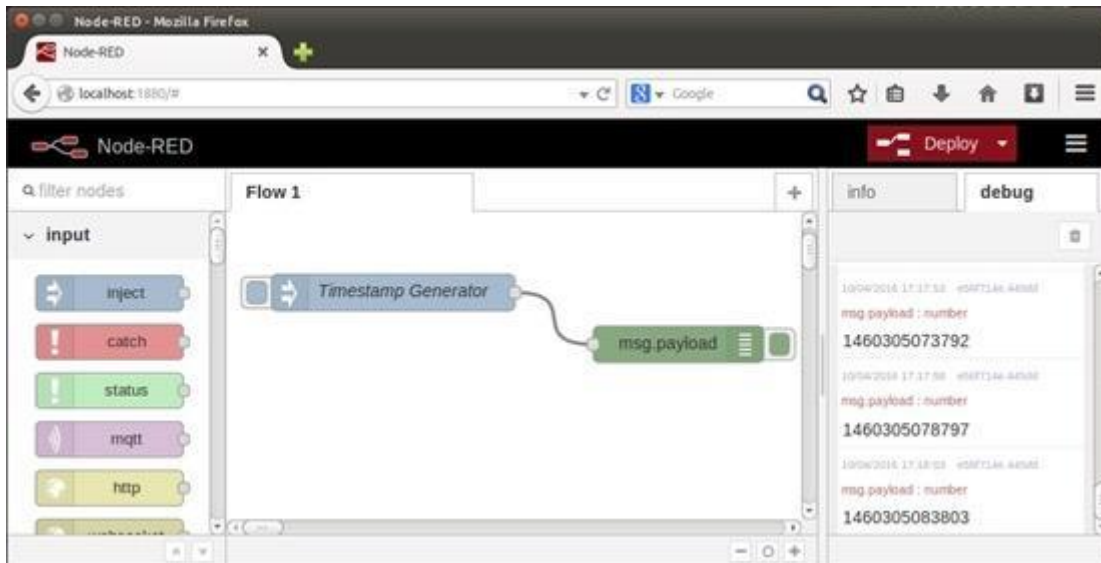
Catatan: sudo diperlukan jika berjalan sebagai pengguna non-root pada Linux / OS X. Jika berjalan pada Windows, Anda akan perlu untuk berjalan dalam shell perintah sebagai Administrator, tanpa perintah sudo.

- Berikutnya

Setelah terinstal, Anda siap untuk menjalankan Node-RED.

Nama : Fahrul Rozi
Nim : 09011181320022

- Membuat Flow Sederhana di Node-RED



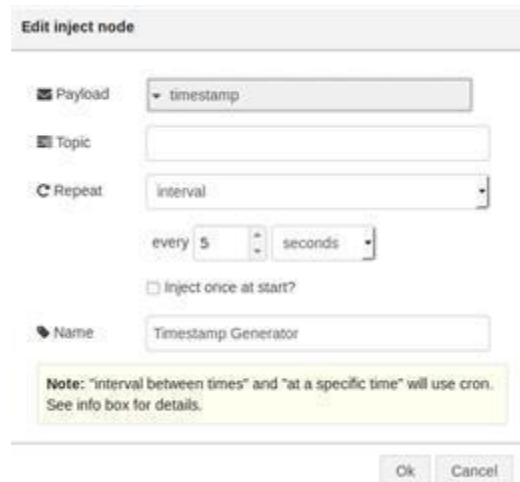
Gambar 1. Contoh flow di dalam lingkungan pengembangan Node-RED

Lihat pada Gambar 1 untuk memahami proses pengembangan aplikasi di Node-RED. Di bagian kanan adalah panel *node* yang mana mendaftarkan *node-node* yang tersedia untuk aplikasi kamu. Di panel tengah adalah di mana kamu membuat *flow*-mu.

Catat bahwa di contoh *flow* ini, dua *node* didefinisikan; pertama adalah *inject node* yang membuat pesan pada interval yang telah ditentukan pengguna (pada kasus ini akan dibuat pesan *timestamp* setiap 5 detik seperti yang ditampilkan di Gambar 2).

Node tersebut dikoneksikan dengan *debug node* yang mana digunakan untuk mengirim data ke sebuah catatan. *Node*-nya diberi nama *msg.payload* yang mengindikasikan apa yang *node* tersebut akan hasilkan. Terakhir, pada panel kanan adalah keluaran hasil *debug*. Tiap tiga segmen di Gambar 1 adalah keluaran individu dari *debug node* yang jika diperhatikan mereka terpisah dalam 5 detik.

Nama : Fahrul Rozi
Nim : 09011181320022



Gambar 2. Konfigurasi dari inject node

Gambar di atas adalah *flow* sederhana untuk mengilustrasikan ide pengembangan aplikasi di Node-RED. Sebuah *inject node* menghasilkan pesan yang berisi *timestamp* dan sebuah *debug node* menerima pesannya dan mengirim ke catatan *debug*.

- Node dan Jenisnya

Node-RED menyediakan berbagai jenis *node* yang dapat membuat kamu langsung menjadi produktif. Gambar 3 menampilkan *input node* dan *output node* yang mana mengizinkan subskripsi dan tanda terima dari topik MQ Telemetry Transport (MQTT) dan keluaran dari topik MQTT ke sebuah broker; pengembangan layanan web melalui permintaan HTTP (beserta pembuatan balasan HTTP); dan TCP level rendah dan layanan User Datagram Protocol yang dapat membuat server, menerima input, dan menghasilkan output.

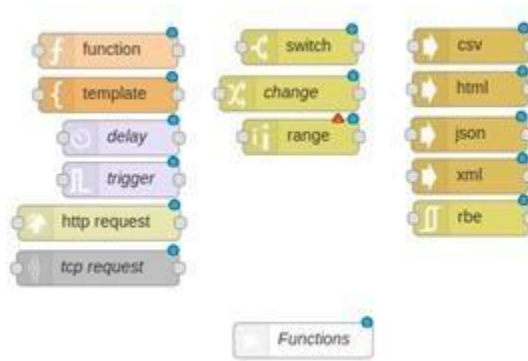


Nama : Fahrul Rozi
Nim : 09011181320022

Gambar 3: Input node dan output node di Node-RED

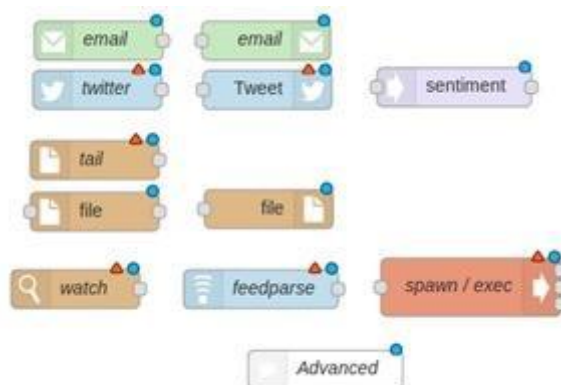
Processing node ditampilkan di Gambar 4. Dengan *node-node* tersebut, kamu dapat membuat *node* dengan fungsi tersendiri (dalam JavaScript), menghasilkan pesan dengan pemicunya berbasis waktu, dan menunda pesan untuk menilai batas *flow*.

Kamu juga dapat mengubah jalur dari pesan-pesan menggunakan saklar dan sebuah set ekpresi kondisional, mengubah jalur pesan (atau menghapus pesan) berdasarkan konten pesannya, dan mengubah jarak jarak dari pesan numerik berdasarkan tindakan dari pengguna. Terakhir, kamu dapat menerjemahkan berbagai forma data input dan mengkonversikan mereka.



Gambar 4. Function node di Node-RED

Set *node* standar terakhir adalah *advanced node* (Gambar 5). *Node-node* ini meliputi berbagai jenis kasus penggunaan seperti menerima dan mengirim email dan melakukan analisa sentimen pada *tweet*. Kamu dapat mengirim atau menerima pesan dari penyimpanan dan memantau perubahan dalam sistem *file*. Terakhir, kamu dapat memonitor *feed* RSS/Atom untuk entri baru dan bahkan menghasilkan atau mengeksekusi perintah untuk sistem operasi.



Nama : Fahrul Rozi
Nim : 09011181320022

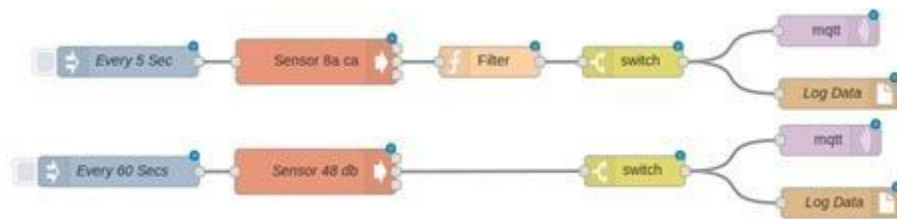
Kamu juga dapat mengunduh *node* baru jika kamu tidak melihat apa yang kamu perlukan (seperti pengenalan wajah dan antarmuka untuk berbagai macam perangkat fisik). Sekarang, mari lihat pengembangan dari *flow* yang lebih kompleks.

- Membuat Flow yang Lebih Kompleks di Node-RED

Kekuatan utama dari Node-RED berasal dari pengaplikasiannya di IoT. Pada contoh ini, kamu membaca sensor dari *embedded device* dan lalu mendistribusikan mereka untuk digunakan. Pada *flow* berikut seperti yang diperlihatkan di gambar 6, hal pertama yang akan kamu lihat adalah alih-alih *flow* tunggal, kamu memiliki dua untuk dua sensor yang ingin dimonitor.

Data dari sensor pertama diambil secara periodik dan kemudian difilter untuk dilaporkan hanya jika datanya melampaui batas (seperti alarm) menggunakan sebuah fungsi. Ketika batasnya dilampaui, kamu mengirimkan pesan yang dihasilkan ke baik MQTT sebagai topik alarm dan ke *file* catatan.

Data dari sensor kedua juga diambil secara periodik namun datanya akan selalu dikirimkan ke MQTT sebagai topik lainnya dan *file* catatan. Node-RED menyajikan paralelisme ini untuk kamu sebagai model programman mereka



Gambar 6. Flow yang lebih kompleks

Node-RED mengizinkan pembuata banyak *flow* yang terdiri dari banyak *node* sehingga kamu dapat menggunakan untuk melakukan pengkoleksian dan manipulasi data lokal pada *embedded device* atau dalam sebuah server untuk menerima data dan melakukan pemrosesan yang lebih kompleks.

- Menggunakan Model

Node-RED bergantung pada Node.js, sebuah *open source runtime environment* untuk mengembangkan aplikasi web di sisi server. Node.js mendukung model berbasis *event-driven* yang mana cocok untuk lingkungan berbasis *flow* asinkron dari Node-RED.

Nama : Fahrul Rozi
Nim : 09011181320022

Kamu dapat membuat *flow* dengan Node-RED yang berjadalan *embedded edge device* seperti modul komputasi Intel Edison dan papan pengembangan Intel Galileo bersertifikasi Arduino. Perangkat-perangkat ini berada di ujung ekosistem IoT dan berinteraksi langsung dengan antarmuka *hardware* baik yang *legacy* ataupun baru seperti sensor temperatur dan penggerak motor.

Kamu juga dapat menjalankan Node-RED pada perangkat *gateway* seperti Intel IoT Gateway. Pada model ini kamu bisa menggunakan Node-RED untuk berkomunikasi dengan banyak perangkat komputisai untuk mengkoleksi dan memproses data mereka untuk kemudian didistribusikan ke broker MQTT untuk kemudian disebar lebih lanjut.

Di kedua medua tersebut, Node-RED mengeksekusi pada perangkat: Kamu sederhananya dapat mengaitkan *browser* di atas jaringan ke perangkat untuk membuat dan mengetes *flow*. Pada cara ini, lingkungan pengembangan dan pengetesan dan juga *runtime*-nya berada pada perangkatnya sehingga hanya memerlukan *browser* untuk melakukan koneksi.

Kesimpulan

Node-RED adalah *tool* pemrograman yang kuat untuk mengkoneksikan secara virtual *flow-flow* yang mengumpulkan, memproses, dan menyebarkan data. Node-RED menyajikan berbagai macam *node* untuk memperkenalkan kamu ke model pemrogramannya dan juga koleksi *node* dan *flow* yang besar yang dapat diunduh dan digunakan untuk aplikasi kamu.

Yang terbaik dari semuanya adalah kamu dapat membawa papan pengembangan Intel favorit kamu atau *gateway* IoT, instal Node-RED, dan mengeksekusi *flow* dengan waktu dan usaha yang minim. Ini membuat Node-RED menjadi bagian integral dari ekosistem IoT apapun.