

BAB 6

NODERED

6.1. APA ITU NODE RED?

Nod-RED dikembangkan oleh IBM pada tahun 2013. Ini menjadi open source pada tahun 2013 dan oleh karena itu tersedia untuk masyarakat umum. Pada tahun 2016, Node-RED menjadi bagian dari JS Foundation.

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.

6.2. CARA KERJA NODERED

Fokus utamanya adalah menghubungkan perangkat keras, antarmuka, dan layanan seperti yang kita kenal di dunia baru Internet of Things (IoT). Node-RED menggunakan pemrograman grafis. Beberapa orang mengetahui pemrograman visual Blockly. Tapi Node-RED benar-benar berbeda. Ia menggunakan apa yang disebut node, yang membentuk aliran terhubung. Dengan menghubungkan node-node dalam antarmuka secara logis, rangkaian program yang sederhana hingga yang kompleks dapat diimplementasikan. Namun, Node-RED jelas lebih kuat daripada Blockly karena Anda juga dapat memasang ekstensi di sini. Misalnya, instalasi KNX dapat dikontrol dengan Node-RED atau openHAB 2 dapat diintegrasikan. Jika Anda ingin menghubungkan sesuatu yang spesifik, ada baiknya selalu melihat LIBRARY Node-RED. Di sini Anda dapat menemukan node dan aliran dari pengguna lain.


6.3. INSTALASI


Cara Install NODE-RED di Windows


Download the Node.js source code or a pre-built installer for your platform, and start developing today.

LTS
Recommended For Most Users

Current
Latest Features


Windows Installer
node-v16.17.1-win64.msi


macOS Installer
node-v16.17.1.pkg


Source Code
node-v16.17.1.tar.gz

Windows Installer (.msi)
Windows Binary (.zip)
macOS Installer (.pkg)
macOS Binary (.tar.gz)
Linux Binaries (x64)
Linux Binaries (ARM)
Source Code

32-bit	64-bit
32-bit	64-bit
64-bit / ARM64	
64-bit	ARM64
64-bit	
ARMv7	ARMv8
node-v16.17.1.tar.gz	

Additional Platforms

Docker Image
Linux on Power LE Systems
Linux on System z
AIX on Power Systems

Official Node.js Docker Image
64-bit
64-bit
64-bit

- Download NODEJS dari <https://nodejs.org/en/download> kemudian pilih operating system Windows dan “.msi” installer
- install nodejs di windows
- Buka command prompt (CMD) admin right
- kemudian ketik **node -v** untuk cek versi nodejs terpasang. jika muncul seperti pada gambar artinya node js sudah terinstall.

```
C:\WINDOWS\system32\cmd.exe

C:\>node -v
v10.15.3

C:\>
```

- ketik di CMD: **npm install -g --unsafe-perm node-red**

```
C:\WINDOWS\system32\cmd.exe

C:\>node -v
v10.15.3

C:\>npm install -g --unsafe-perm node-red
```

- ketik **node-red** di command prompt

```
node-red

C:\>node-red
15 May 20:28:38 - [info]

Welcome to Node-RED
=====

15 May 20:28:38 - [info] Node-RED version: v0.20.5
15 May 20:28:38 - [info] Node.js version: v10.15.3
15 May 20:28:38 - [info] Windows_NT 10.0.15063 x64 LE
15 May 20:28:38 - [info] Loading palette nodes
15 May 20:28:39 - [warn] rpi-gpio : Raspberry Pi specific node set inactive
15 May 20:28:40 - [info] Settings file : \Users\user\.node-red\settings.js
15 May 20:28:40 - [info] Context store : 'default' [module=memory]
15 May 20:28:40 - [info] User directory : \Users\user\.node-red
15 May 20:28:40 - [warn] Projects disabled : editorTheme.projects.enabled=false
15 May 20:28:40 - [info] Flows file : \Users\user\.node-red\flows_1\1234567891.json
15 May 20:28:40 - [warn]

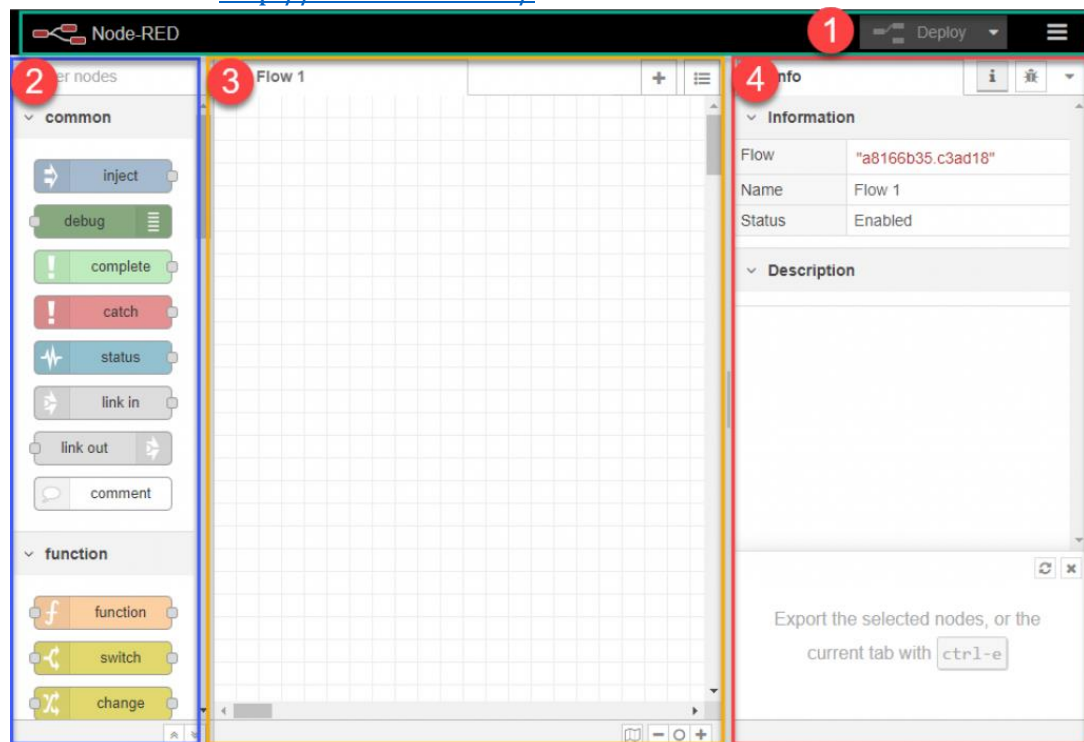
-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----

15 May 20:28:40 - [info] Starting flows
15 May 20:28:40 - [info] Started flows
15 May 20:28:40 - [info] Server now running at http://127.0.0.1:1880/
```

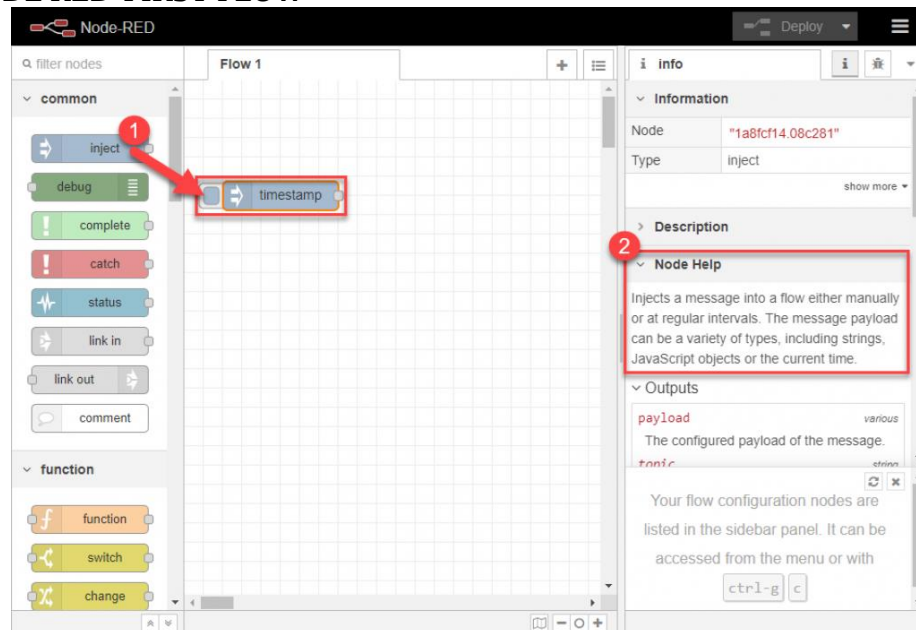
- Buka Browser <http://localhost:1880/>



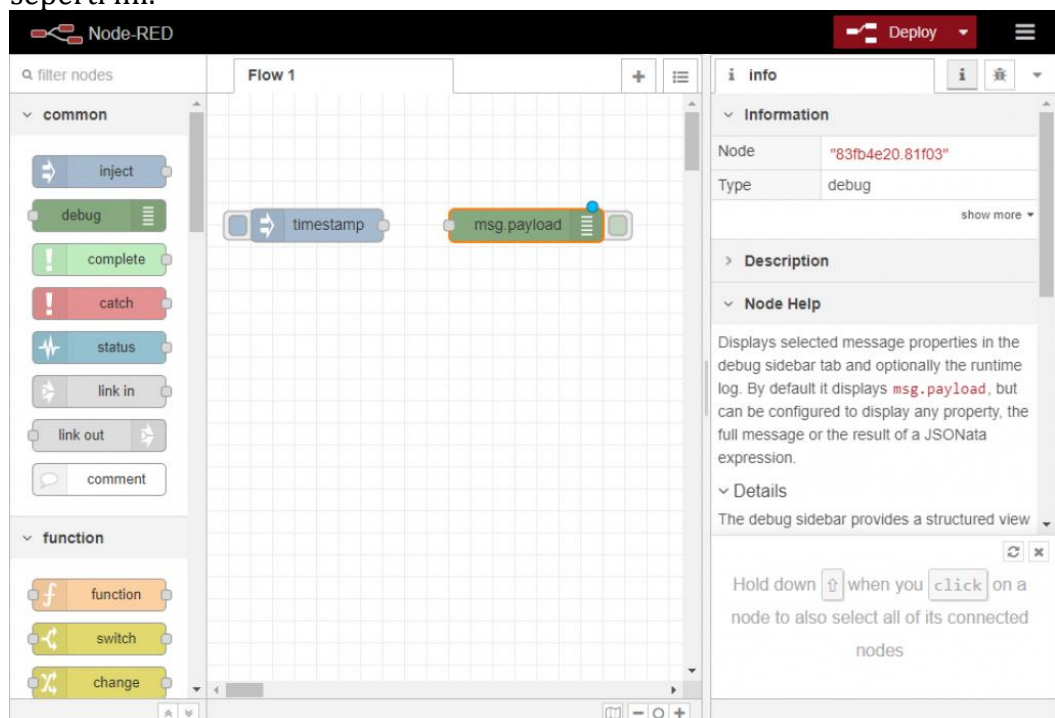
Halaman utama terbagi menjadi 4 area.

1. Menu Bar
2. Palette with nodes
3. Flow area
4. Sidebar with, misalnya informasi, pesan debug

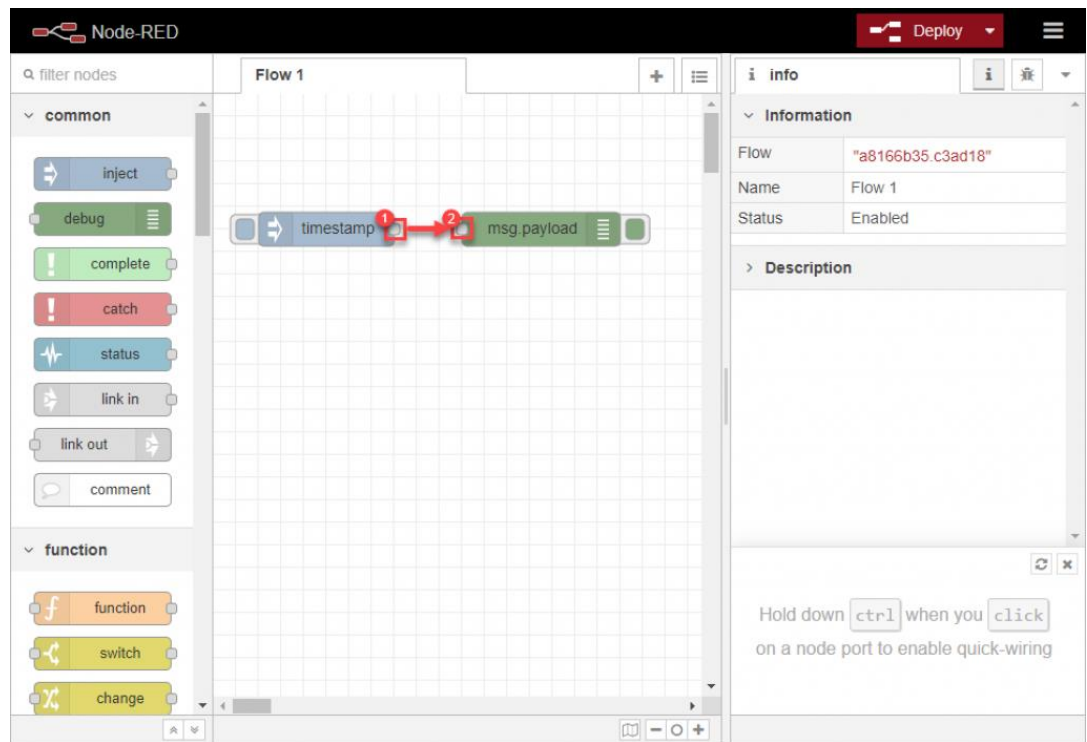
6.4. NODE RED-FIRST FLOW



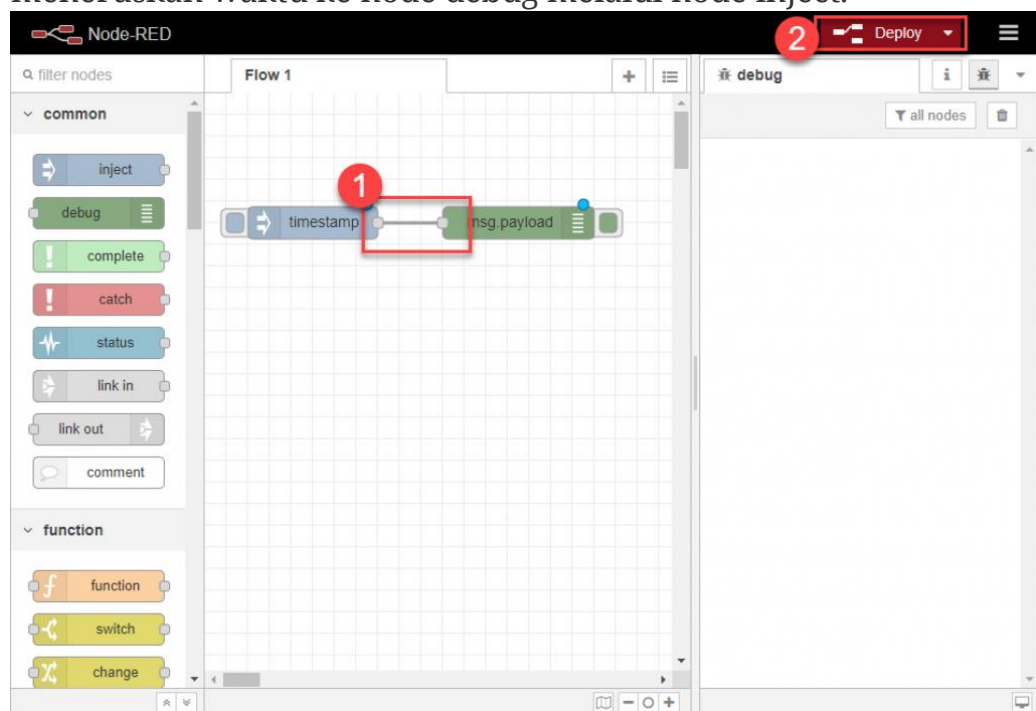
- Untuk memberi Anda pemahaman yang lebih baik tentang Node-RED, kami akan membuat aliran pertama kami. Pertama, cukup klik pada node Inject (area 2) dengan tombol kiri mouse (1) dan seret ke dalam aliran (area 3). Sidebar (area 4) dengan informasi akan beradaptasi secara langsung. Jika node benar-benar mendapat bantuan (2), Anda akan melihatnya langsung di sidebar.
- Kedua, node debug berpindah ke aliran. Aliran Anda sekarang akan terlihat seperti ini.



- Ketiga, hubungkan node satu sama lain. Anda membuat koneksi dengan mengklik output (1) dengan tombol kiri mouse dan menyeretnya ke input (2).

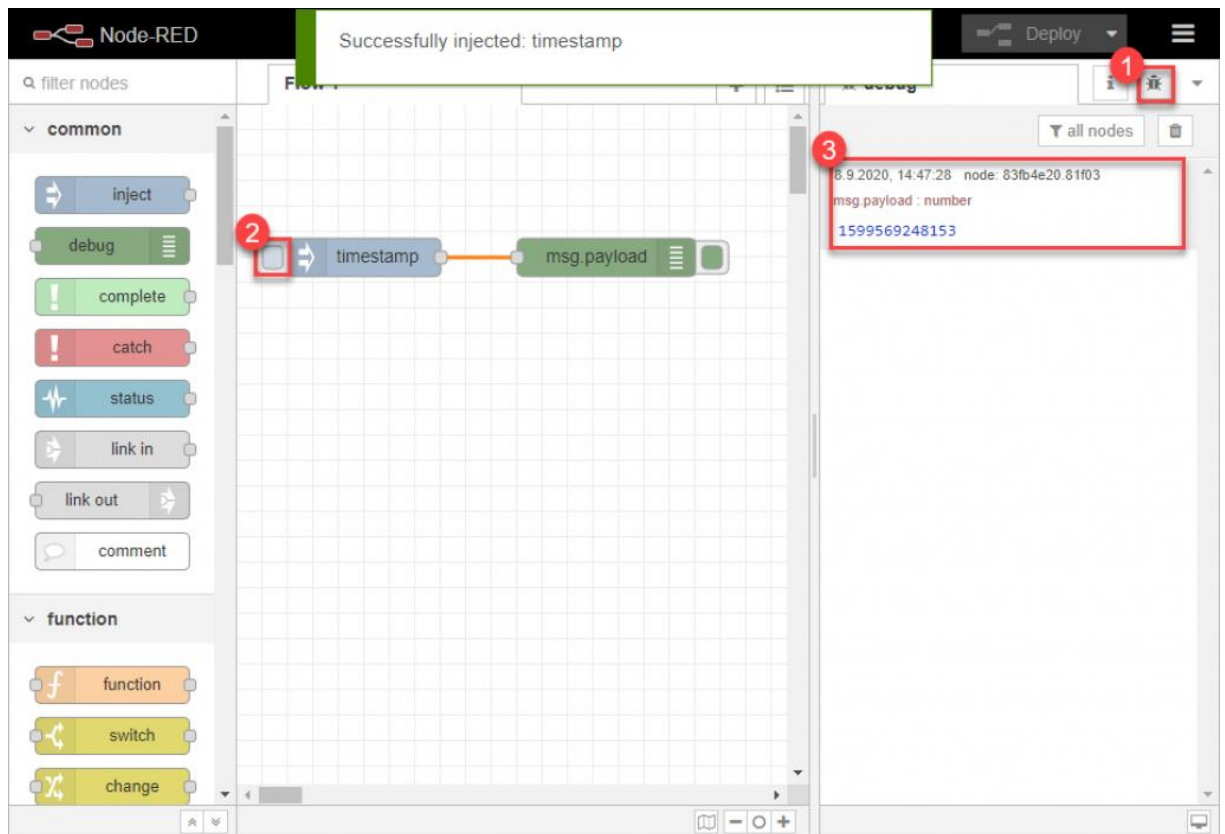


- Seharusnya ada garis (1) di antara node. Anda masih harus menerapkan perubahan. Untuk melakukan ini, tekan terapkan (2)
- Selamat, Anda sekarang telah membuat alur pertama yang meneruskan waktu ke node debug melalui node inject.

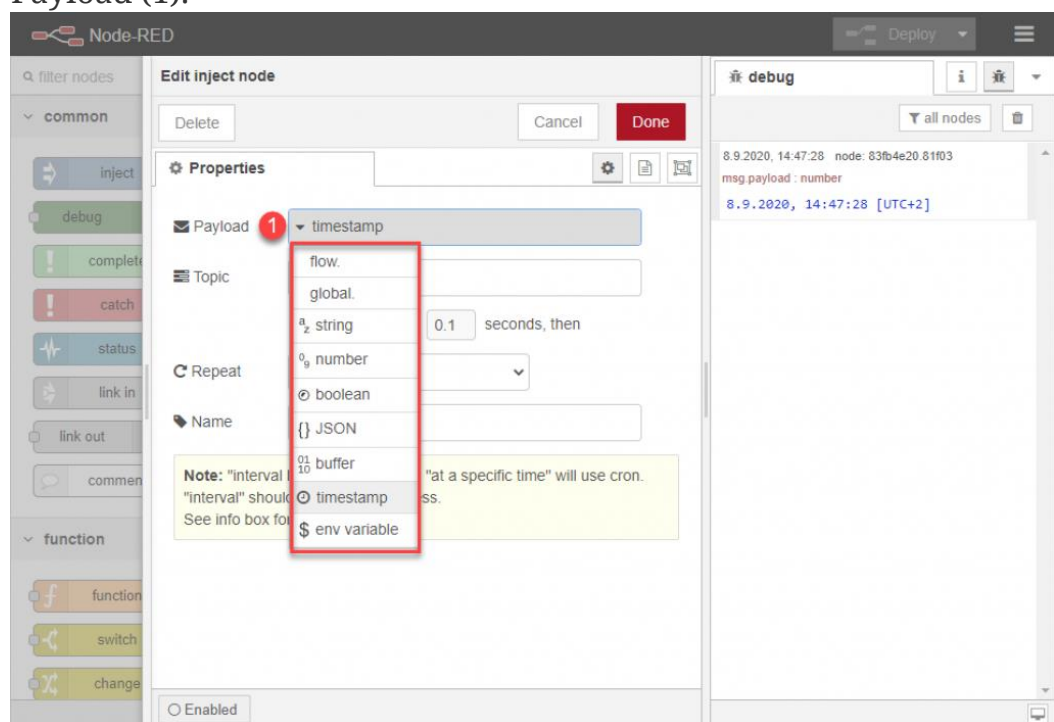


- Kemudian uji fungsinya dengan membuka jendela debug (1). Kemudian tekan tombol di sebelah kiri (2) node inject. Anda bisa melihat hasilnya langsung di sidebar (3). Jangan kaget, waktu diberikan dalam milidetik dan sesuai dengan tanggal dan waktu saat ini. Jika Anda mengklik font biru, Anda dapat mengubah format

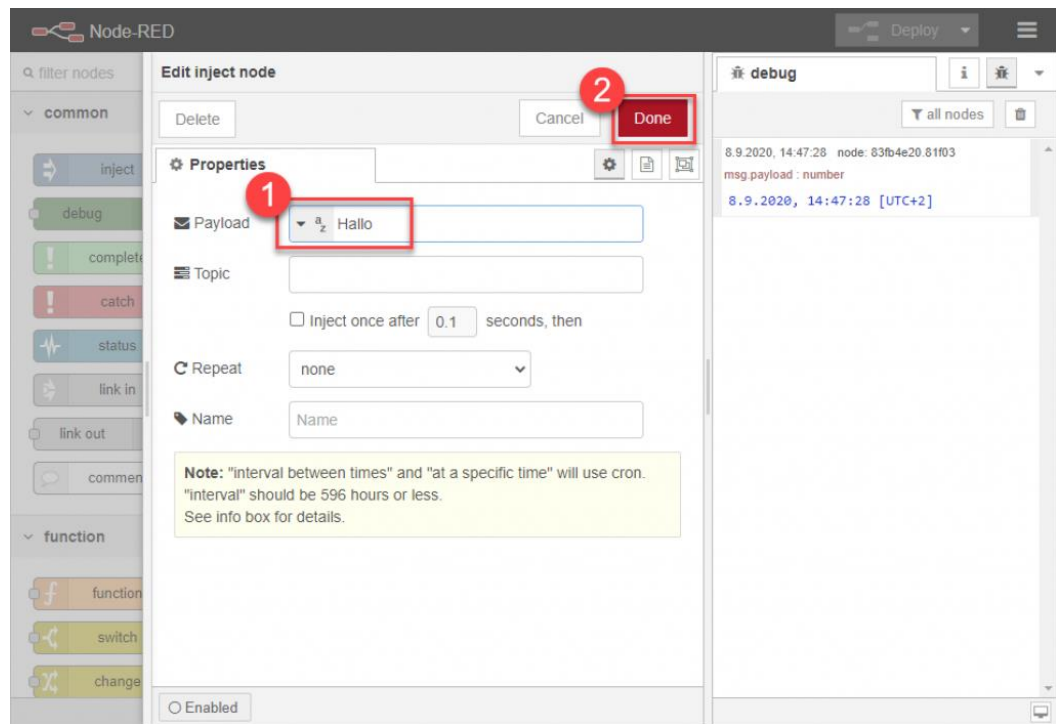
tampilan.



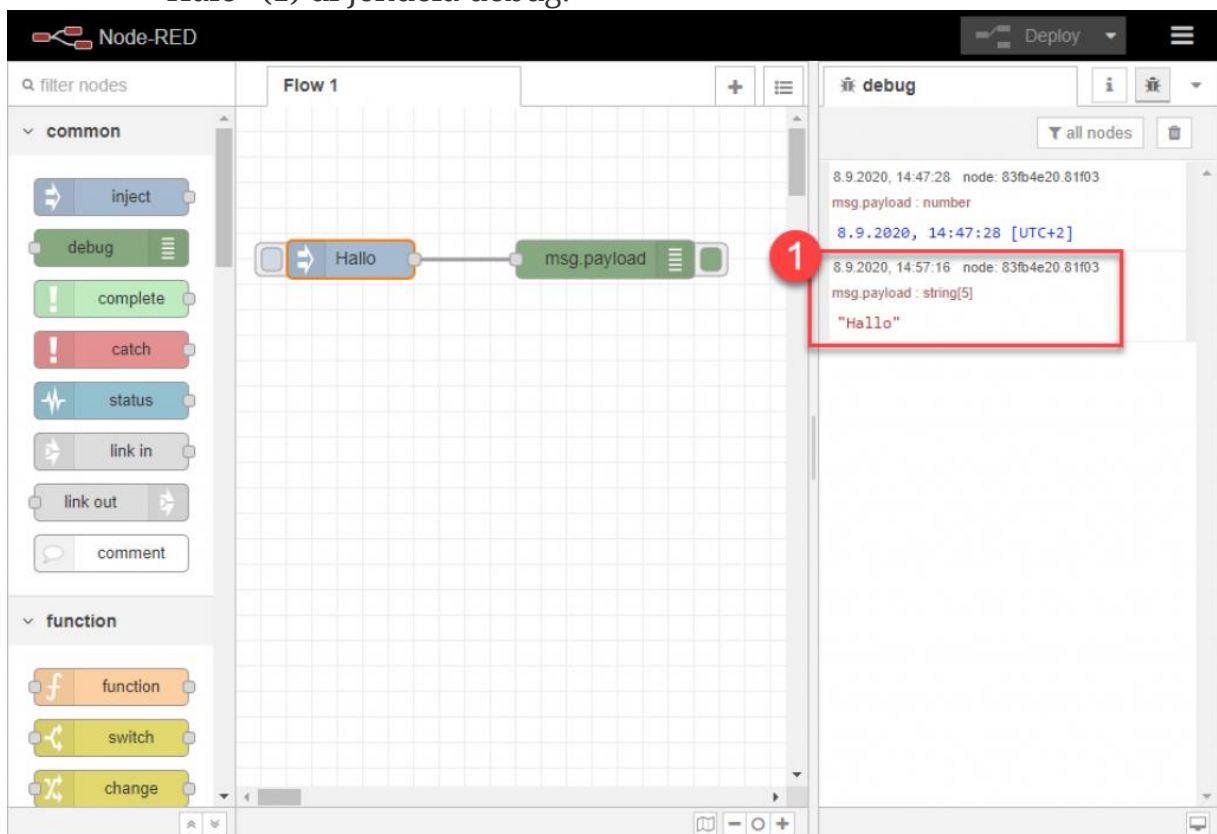
- Jika sekarang Anda ingin memasukkan nilai lain selain waktu, klik dua kali pada node inject.
- Di sini Anda dapat mengubah nilai yang akan ditransfer dalam Payload (1).



- Misalnya, pilih string di sini dan masukkan “Halo” (1) sebagai nilainya. Kemudian klik Selesai (2).



- Setelah setiap perubahan, Anda harus menekan Deploy, jika tidak, perubahan tidak akan berlaku. Anda dapat melihat apakah perubahan perlu diterapkan ketika tombol Deploy aktif (merah). Anda dapat mengenali perubahan individual dalam aliran dengan titik biru kecil di aliran. Setelah menjalankan node inject lagi kita melihat teks “Halo” (1) di jendela debug.



6.5. NODE MASUKAN

Ada 7 node input dasar yang diinstal secara default. Hal ini mencakup mekanisme komunikasi dasar yang mungkin digunakan oleh aplikasi IoT. Mulai dari protokol internet tingkat rendah seperti UDP dan TCP hingga HTTP tingkat tinggi dan MQTT terbitkan/berlangganan.

Nama Node	Keterangan
inject	Memasukkan stempel waktu atau teks yang dikonfigurasi pengguna ke dalam pesan. Dapat dikonfigurasi untuk menginjeksi secara manual, pada interval yang ditentukan, atau pada waktu tertentu (menggunakan Cron).
catch	Menangkap kesalahan yang terjadi oleh node pada tab yang sama. Jika sebuah node menimbulkan kesalahan saat menangani pesan, aliran biasanya akan terhenti. Node ini dapat digunakan untuk menangkap kesalahan yang mengembalikan pesan dengan properti error yang merinci kesalahan serta node sumber dan jenisnya.
mqtt	Berlangganan ke broker MQTT dan mendengarkan suatu topik, mengembalikan data apa pun yang dipublikasikan tentang topik tersebut sebagai pesan baru. Mendukung tingkat Kualitas Layanan dan retensi data terakhir.
http	Menerima permintaan HTTP, memungkinkan Node-RED bertindak sebagai server web dasar. Badan HTTP dikirimkan sebagai pesan keluaran bersama dengan respons apa pun. Pesan dapat berisi data berkode URL standar atau JSON.
websocket	Menyediakan titik akhir bagi browser untuk membuat koneksi websocket dengan Node-RED. Menawarkan koneksi dupleks untuk kombinasi browser/server.
tcp	Digunakan untuk menerima permintaan TCP yang masuk pada port tertentu atau untuk menyambung ke port TCP jarak jauh. Menghasilkan pesan yang berisi data TCP sebagai buffer tunggal – atau aliran –, string, atau base64 yang dikodekan.
udp	Digunakan untuk menerima paket UDP masuk (atau paket multicast) pada port tertentu. Menghasilkan pesan yang berisi data UDP sebagai BUFFER, string, atau string yang dikodekan base64.
serial in	Membaca dari port serial pada perangkat lokal. Dapat dikonfigurasi untuk membaca buffer, jangka waktu tertentu atau menunggu jeda baris.

6.6. NODE KELUARAN

Node keluaran pada dasarnya adalah gambaran cermin dari kumpulan node masukan dasar dan menyediakan cara untuk mengirim data pada kumpulan protokol yang sama, misalnya mqtt, http, udp, dll.

Nama Node	Keterangan
debug	Menyediakan cara sederhana untuk melihat pesan yang ditampilkan di panel debug. Dapat dikonfigurasi untuk menampilkan msg.payload saja atau seluruh objek msg.
mqtt	Berlangganan ke broker MQTT dan memposting data apa pun (msg.payload) yang diterimanya dalam pesan masuk ke suatu topik. Mendukung tingkat Kualitas Layanan dan retensi data terakhir.
http	Mengirim respons kembali ke permintaan HTTP yang diterima dari node Input HTTP. Isi respons ditentukan oleh msg.payload, dan dapat menentukan header dan kode status.
websocket	Mengirim msg.payload keluar pada websocket yang dikonfigurasi. Jika msg.session ditentukan, kirim ke klien asal, jika tidak, siarkan ke semua klien yang terhubung
tcp	Membalas ke port TCP yang dikonfigurasi. Dapat juga digunakan untuk mengirim ke port tertentu.
udp	Mengirim pesan UDP ke host (alamat ip) dan port yang dikonfigurasi. Mendukung siaran. Seperti kebanyakan node, dikonfigurasi melalui UI atau properti pesan.
serial out	Mengirim ke port serial yang ditentukan. Dapat dikonfigurasi untuk mengirim baris baru opsional setelah muatan pesan apa pun.

6.7. NODE FUNGSI

Kategori fungsi berisi berbagai node yang menjalankan fungsi pemrosesan tertentu. Mulai dari node tunda dan saklar sederhana hingga node fungsi yang dapat diprogram yang dapat disesuaikan dengan hampir semua kebutuhan pemrograman.

Nama Node	Keterangan
function	Node fungsi umum yang dapat diprogram. Dengan menggunakan JavaScript standar, node dapat disesuaikan untuk melakukan pemrosesan kompleks pada pesan masukannya yang menghasilkan satu atau lebih pesan keluaran.
template	Dikonfigurasi dengan templat (menggunakan format kumis) dengan kompleksitas yang berubah-ubah, simpul ini mengambil pesan masukan yang berisi pasangan nama:nilai dan menyisipkannya ke dalam templat. Berguna untuk membuat pesan, HTML, file konfigurasi, dll.
delay	Node umum yang menunda pesan pada waktu tertentu atau acak. Dapat juga dikonfigurasi untuk membatasi aliran pesan (misalnya 10 pesan per detik).

trigger	Membuat dua pesan keluaran yang dipisahkan oleh interval waktu yang dapat dikonfigurasi setiap kali pesan masukan diterima. Dapat juga digunakan sebagai pengatur waktu pengawas.
comment	Komentar visual sederhana yang dikonfigurasi dengan judul dan isi.
http request	Memungkinkan Anda membuat dan mengirim permintaan HTTP ke URL tertentu. Metode (PUT, GET, dll), header dan payload semuanya dapat dikonfigurasi melalui UI atau secara terprogram.
tcp request	Node permintaan TCP sederhana. Ia mengirimkan msg.payload ke port tcp server dan mengharapkan respons. Dapat dikonfigurasi untuk menunggu data, menunggu karakter tertentu, atau segera kembali.
switch	Node ini merutekan pesan berdasarkan propertinya. Properti dikonfigurasi menggunakan UI dan dapat berupa berbagai logika (>, <, >= dll) yang diterapkan ke properti pesan.
change	Node perubahan dapat digunakan untuk mengatur, mengubah atau menghapus properti pesan masuk. Berbagai aturan yang dapat dikonfigurasi memungkinkan perubahan kompleks termasuk pencarian dan penggantian di msg.payload
range	Node penskalaan sederhana yang akan memetakan masukan numerik ke keluaran baru. Berguna untuk mengonversi atau membatasi rentang nilai masukan, misalnya suhu. Tidak ditentukan untuk data non-numerik.
csv	Node ini mem-parsing msg.payload dan mencoba mengonversi ke/dari CSV. Jika menerima string, ia akan mengeluarkan objek JavaScript, dan jika menerima objek JavaScript, ia akan mengeluarkan string CSV.
html	Mengekstrak elemen dari dokumen html di msg.payload menggunakan pemilih yang dapat dikonfigurasi (sintaks pemilih CSS). Pada dasarnya memungkinkan Anda mengurai HTML dan mengembalikan serangkaian elemen yang cocok.
json	Node ini mengkonversi ke/dari objek JSON. Jika menerima objek JavaScript, ia akan mengeluarkan JSON, dan jika menerima JSON, ia akan mengeluarkan objek JavaScript.
xml	Node ini mengkonversi ke/dari format XML. Jika menerima objek JavaScript, ia akan mengeluarkan string XML, dan jika menerima string XML, ia akan mengeluarkan objek JavaScript.
rbe	Laporan Berdasarkan simpul Pengecualian. Menghasilkan pesan hanya ketika masukannya berbeda dari masukan sebelumnya (string atau angka) atau jika masukan telah diubah dengan jumlah yang dapat dikonfigurasi (mode pita mati) – hanya untuk angka.

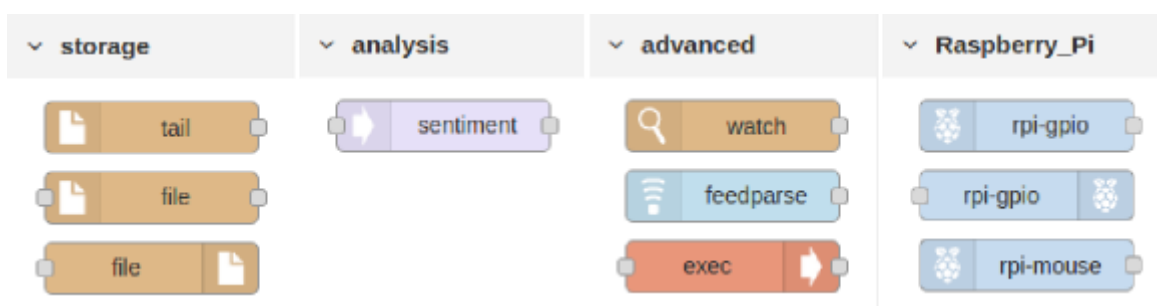
6.8. NODE SOSIAL

Node media sosial dasar mendukung interaksi dengan email dan Twitter. Mereka memungkinkan aliran untuk mengirim atau menerima email, atau untuk mengirim atau menerima tweet.

Nama Node	Keterangan
email in	Dapat dikonfigurasi untuk membaca berulang kali dari server IMAP yang mengembalikan email baru saat email tersebut diterima. Setel msg.topic ke subjek email dan msg.payload ke badan teks email atau msg.html jika emailnya HTML.
twitter in	Mengembalikan tweet sebagai pesan. Dapat digunakan untuk mencari tweet publik atau aliran pengguna yang berisi istilah pencarian yang dikonfigurasi atau semua tweet oleh pengguna tertentu atau pesan langsung yang diterima oleh pengguna yang diautentikasi.
email out	Mengirim pesan masuk sebagai email melalui server IMAP yang dikonfigurasi. Topik dan penerima semuanya dapat dikonfigurasi. Akan mengonversi data biner menjadi lampiran.
twitter out	Tweet msg.payload pada akun yang dikonfigurasi. Dapat mengirim pesan langsung dan akan mengirimkan data biner sebagai gambar.

6.9. NODE PENYIMPANAN (STORAGE NODE)

Node default ditetapkan untuk penyimpanan, karena ditargetkan pada perangkat seperti Raspberry Pi yang cukup terbatas dan fokus pada penyimpanan berbasis file.



Gambar Penyimpanan default, analisis, node lanjutan dan Raspberry Pi.

Anda harus mencatat bahwa FRED, karena ini adalah layanan cloud, tidak mendukung node file dasar. Sebaliknya, ia menggantikannya dengan berbagai node penyimpanan mulai dari Mongo hingga Dropbox. Namun, untuk kelengkapannya, node default dibahas di sini, sehingga Anda dapat menggunakannya, jika Anda menginstal Node-RED sendiri.

Nama Node	Keterangan
Tail	Tails (yaitu mengawasi hal-hal yang akan ditambahkan) ke file yang dikonfigurasi. (KHUSUS Linux/Mac)

	Ini tidak akan berfungsi pada sistem file Windows, karena bergantung pada perintah tail -F.
file in	Membaca file yang ditentukan dan mengirimkan konten sebagai msg.payload, dan nama file sebagai msg.nama file. Nama file dapat dikonfigurasi di node. Jika dibiarkan kosong, itu harus diatur pada msg.nama file dalam pesan masuk.
File	Menulis msg.payload ke file yang ditentukan, misalnya untuk membuat log. Nama file dapat dikonfigurasi di node. Jika dibiarkan kosong, itu harus diatur pada msg.nama file dalam pesan masuk. Perilaku defaultnya adalah menambahkan ke file. Ini dapat diubah untuk menimpa file setiap saat; misalnya, jika Anda ingin menampilkan halaman web atau laporan “statis”.

6.10. NODE RASPBERRY PI

Nama Node	Keterangan
rpi_gpio in	Node masukan Raspberry Pi. Menghasilkan msg.payload dengan 0 atau 1, bergantung pada status pin input. Anda juga dapat mengaktifkan resistor pullup input atau resistor pulldown. Topik pesan disetel ke pi/{nomor pin} Membutuhkan pustaka python RPi.GPIO versi 0.5.8 (atau lebih baik) agar dapat berfungsi. Catatan: kami menggunakan nomor pin fisik sebenarnya pada konektor P1 karena lebih mudah ditemukan.
rpi_gpio out	Node keluaran Raspberry Pi. Mengharapkan msg.payload dengan 0 atau 1 (atau benar atau salah). Akan mengatur pin fisik yang dipilih tinggi atau rendah, tergantung pada nilai yang dimasukkan. Nilai awal pin pada waktu penerapan juga dapat diatur ke 0 atau 1. Saat menggunakan mode PWM, mengharapkan nilai input berupa angka 0 – 100 . Membutuhkan pustaka RPi.GPIO Python versi 0.5.8 (atau lebih baik) agar dapat berfungsi.
rpi_mouse	Node tombol mouse Raspberry Pi. Menghasilkan pesan.payload dengan 1 atau 0 ketika tombol mouse yang dipilih ditekan dan dilepaskan. Juga menyetel msg.button ke nilai kode, 1 = kiri, 2 = kanan, 4 = tengah, sehingga Anda dapat mengetahui tombol atau kombinasi mana yang ditekan.

PRAKTIKUM

Praktekkan tutorial install Nodered dan contoh firstflow.

STUDI KASUS PRAKTIKUM (Boleh dilanjutkan Take Home dan di Demokan di

minggu selanjutnya jika belum selesai dikerjakan di Lab/Kelas)

Sistem Peringatan Suhu Berlebih dengan Output Pesan di Node-RED

Deskripsi:

Anda diminta untuk membuat sistem peringatan sederhana menggunakan **Node-RED** yang akan mengeluarkan pesan jika suhu melebihi batas tertentu. Suhu disimulasikan berubah-ubah secara otomatis setiap beberapa detik, dan sistem harus menampilkan pesan peringatan di konsol jika suhu melebihi ambang batas yang sudah ditentukan (misalnya 35°C).

Langkah-langkah:

1. **Node Input (Simulasi Data Suhu):**
 - Buat node yang mensimulasikan perubahan suhu setiap 3 detik. Suhu disimulasikan secara acak berada dalam rentang 20°C hingga 50°C.
 - Gunakan node `inject` atau `function` untuk mensimulasikan data suhu yang terus menerus diperbarui.
2. **Logika Perbandingan Suhu:**
 - Buat sebuah node `function` yang akan memeriksa apakah suhu yang diterima melebihi ambang batas (default 35°C).
 - Jika suhu melebihi ambang batas, kirimkan pesan peringatan seperti:
 - **"Peringatan: Suhu Berlebih! Suhu saat ini: X°C".**
 - Jika suhu di bawah ambang batas, kirim pesan normal seperti:
 - **"Suhu Normal. Suhu saat ini: X°C".**
3. **Output (Menampilkan Pesan):**
 - Tampilkan hasil peringatan tersebut menggunakan node `debug` (atau output console lainnya), yang akan menampilkan pesan secara real-time saat suhu berubah.
 - Gunakan node `debug` untuk mengamati setiap kali pesan "Peringatan" atau "Suhu Normal" dikirim berdasarkan suhu yang disimulasikan.

Instruksi Penggunaan Node:

- Gunakan node `inject` atau `function` untuk membuat simulasi suhu secara acak.
- Gunakan node `function` untuk membuat logika perbandingan suhu dengan ambang batas.
- Gunakan node `debug` untuk menampilkan output pesan yang sesuai berdasarkan kondisi suhu yang diterima.

Kriteria Penilaian:

1. Sistem dapat mensimulasikan perubahan suhu secara otomatis setiap 3 detik.
2. Jika suhu melebihi 35°C, pesan peringatan ditampilkan di `debug` node: **"Peringatan: Suhu Berlebih! Suhu saat ini: X°C".**
3. Jika suhu di bawah 35°C, pesan normal ditampilkan di `debug` node: **"Suhu Normal. Suhu saat ini: X°C".**
4. Pesan yang ditampilkan harus akurat dan sesuai dengan data suhu yang diterima.

Contoh Output yang Diharapkan:

- **Suhu Normal. Suhu saat ini: 28°C**
- **Suhu Normal. Suhu saat ini: 33°C**

- **Peringatan: Suhu Berlebih! Suhu saat ini: 37°C**
- **Peringatan: Suhu Berlebih! Suhu saat ini: 40°C**

Fitur Tambahan (Optional):

- Tambahkan opsi untuk mengubah ambang batas suhu melalui `function node` (misalnya dari 35°C ke nilai lain).
- Tambahkan logika untuk menghitung rata-rata suhu dalam 5 pembacaan terakhir dan tampilkan dalam pesan.

Tips:

- Gunakan `node random` atau logika acak dalam `function node` untuk mensimulasikan data suhu dengan rentang yang sesuai (misalnya antara 20°C hingga 50°C).
- Gunakan `switch node` atau logika `if-else` dalam `function` untuk membedakan antara kondisi "normal" dan "peringatan".