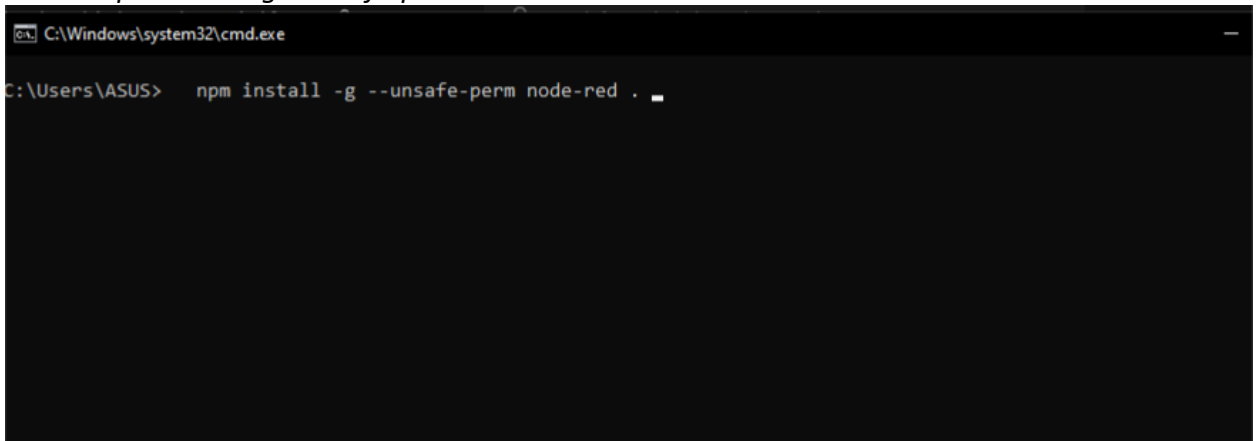


Tips Mudah Menggunakan Node-red

A. Project Banjir

1. Lakukan Instalasi node-red

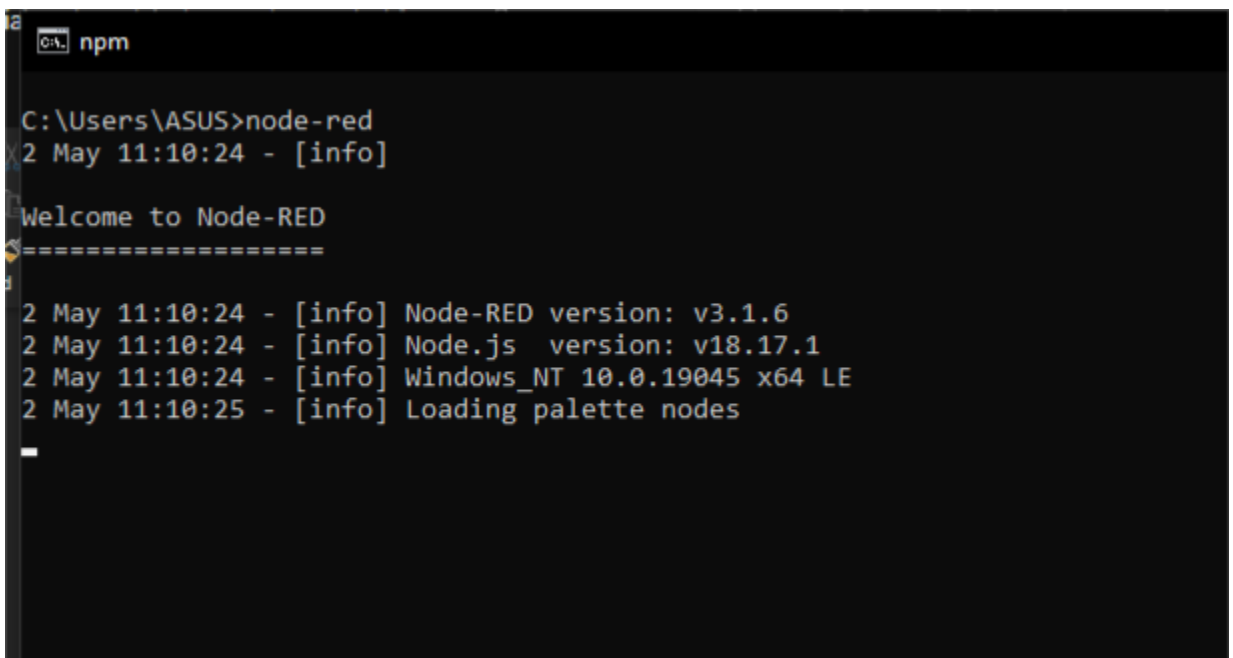
- Buka terminal atau command prompt di komputer Anda.
- Jalankan perintah berikut untuk menginstal Node-RED secara global:
npm install -g --unsafe-perm node-red .



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

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

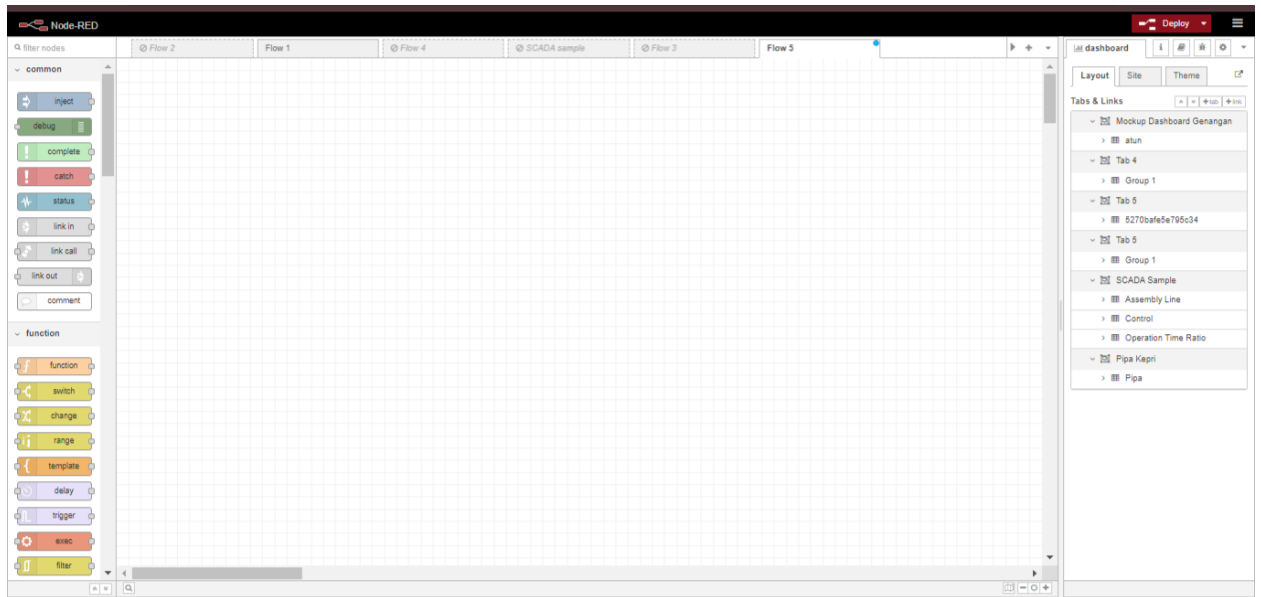
- Menjalankan Node-RED:
Jalankan Node-RED dengan menjalankan perintah: *node-red* pada command prompt



```
npm

C:\Users\ASUS>node-red
2 May 11:10:24 - [info]
Welcome to Node-RED
=====
2 May 11:10:24 - [info] Node-RED version: v3.1.6
2 May 11:10:24 - [info] Node.js version: v18.17.1
2 May 11:10:24 - [info] Windows_NT 10.0.19045 x64 LE
2 May 11:10:25 - [info] Loading palette nodes
```

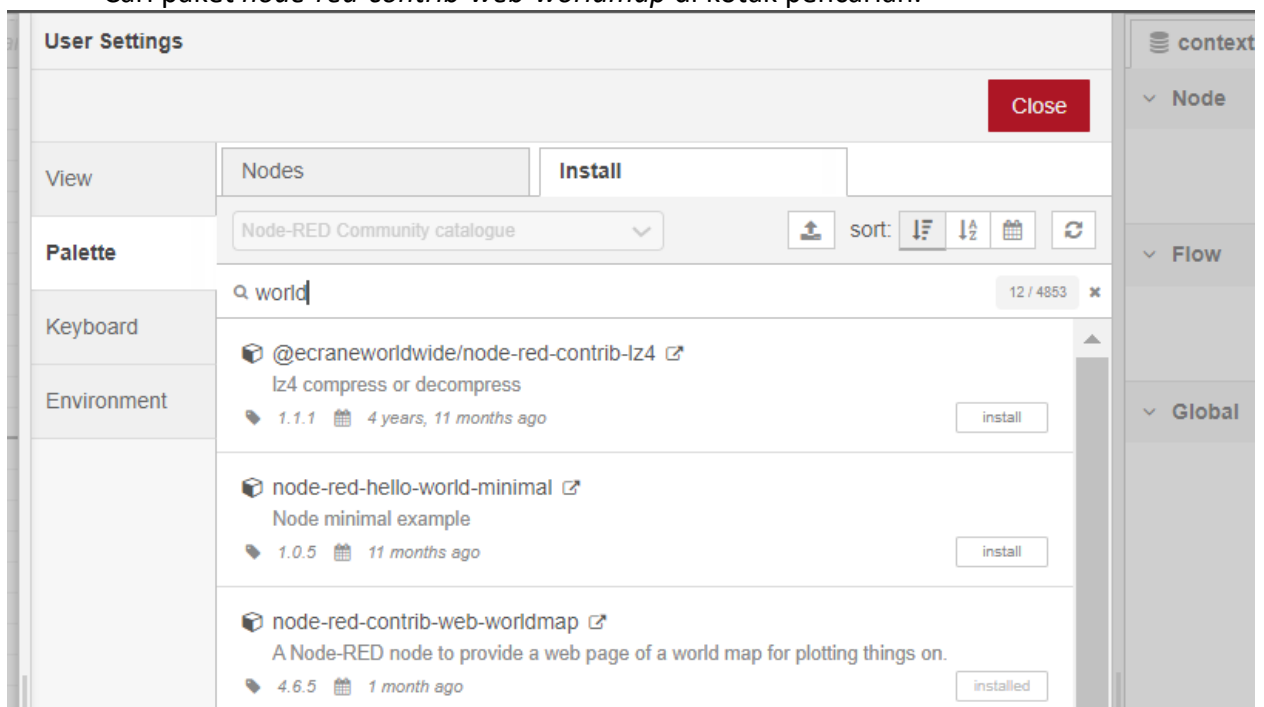
- Akses url : *http://localhost:1880* untuk antarmuka Node-RED.



- Dan akses url : <http://localhost:1880/ui> untuk melihat hasilnya.

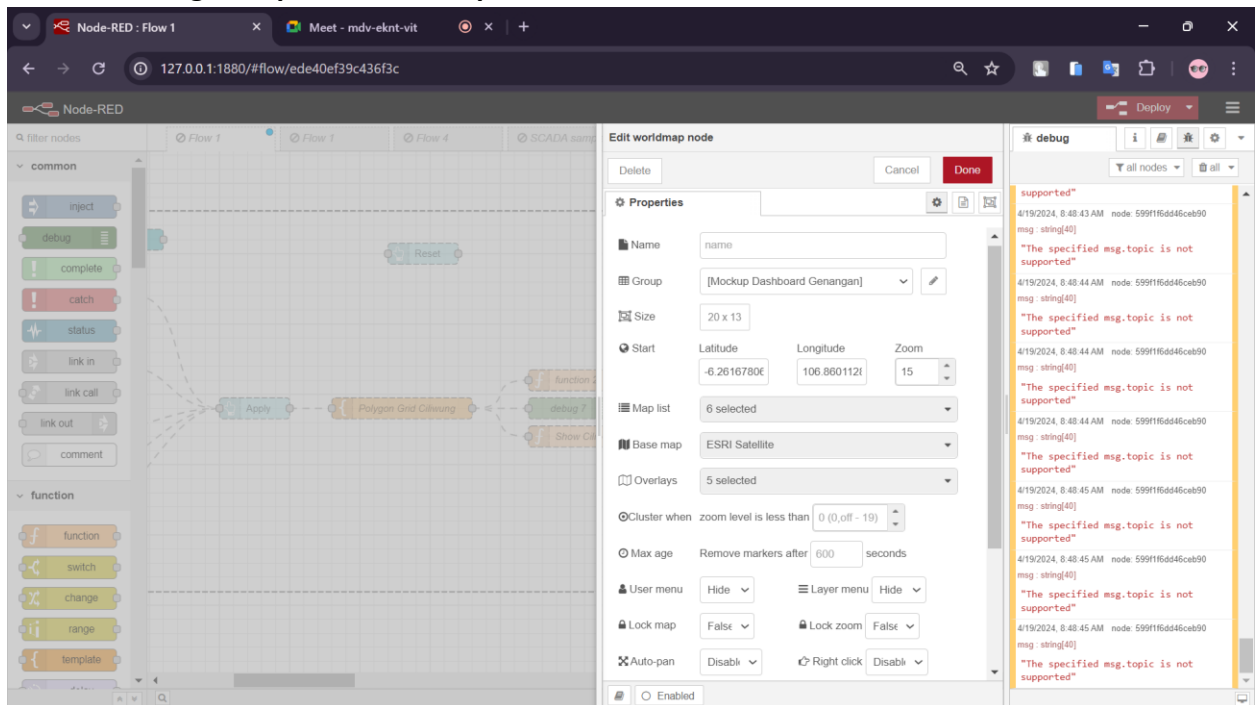
2. Lakukan instalasi package (ui_worldmap) di dalam node-red

- Klik garis tiga di sudut kanan atas.
- Pilih opsi "Manage palette".
- Di jendela "Palette", pilih tab "Install".
- Cari paket *node-red-contrib-web-worldmap* di kotak pencarian.



- Setelah menemukannya, klik tombol "Install" di sebelah paket tersebut.

3. Lakukan konfigurasi pada world map



World map sendiri bisa di konfigurasi sesuai yang kita perlukan

- Group > berguna untuk penempatan
- Size > mengatur ukuran world map
- Start > mengatur peta awal yang akan ditampilkan
- Dll

4. Data yang perlu disiapkan

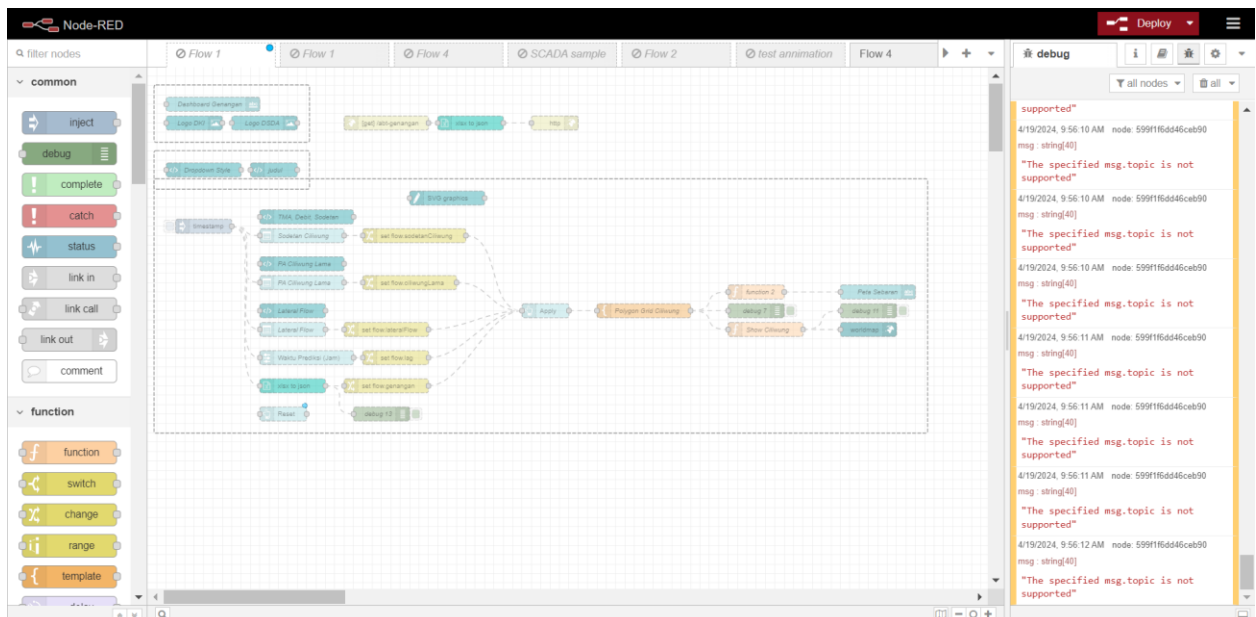
- Untuk konfigurasi data world map bisa bermacam macam dan dapat dilihat pada link berikut: <https://flows.nodered.org/node/node-red-contrib-web-worldmap>

```
Node-RED
Edit template node > Text editor

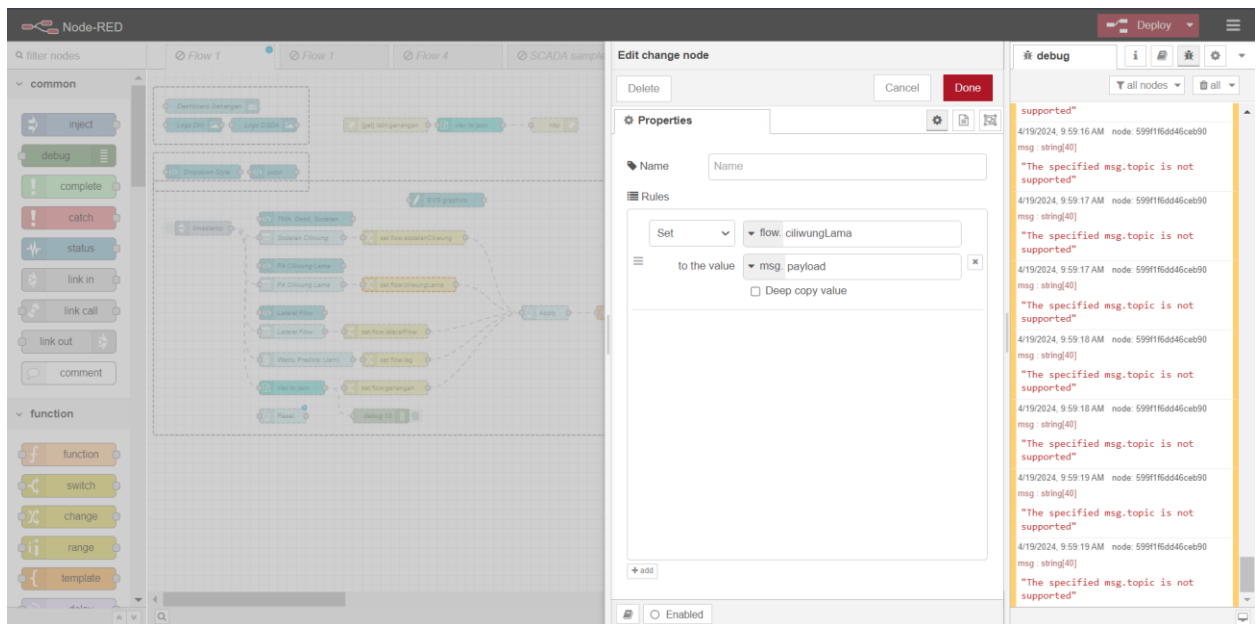
1 [
2   "type": "FeatureCollection",
3   "name": "grid_untuk_genangan",
4   "crs": { "type": "name", "properties": { "name": "urn:ogc:def:crs:OGC:1.3:CRS84" } },
5   "features": [
6     { "type": "Feature", "properties": { "GID": "G181464", "id_kel": "31.74.02.1006", "id_kec": "317402", "id_kota": "3174", "kel_name": "KELURAHAN"
7     { "type": "Feature", "properties": { "GID": "G181465", "id_kel": "31.74.02.1006", "id_kec": "317402", "id_kota": "3174", "kel_name": "KELURAHAN"
8     { "type": "Feature", "properties": { "GID": "G181466", "id_kel": "31.74.02.1006", "id_kec": "317402", "id_kota": "3174", "kel_name": "KELURAHAN"
9     { "type": "Feature", "properties": { "GID": "G181467", "id_kel": "31.74.02.1006", "id_kec": "317402", "id_kota": "3174", "kel_name": "KELURAHAN"
10    { "type": "Feature", "properties": { "GID": "G181468", "id_kel": "31.74.02.1006", "id_kec": "317402", "id_kota": "3174", "kel_name": "KELURAHAN"
11    { "type": "Feature", "properties": { "GID": "G181469", "id_kel": "31.74.02.1006", "id_kec": "317402", "id_kota": "3174", "kel_name": "KELURAHAN"
12    { "type": "Feature", "properties": { "GID": "G181470", "id_kel": "31.74.02.1006", "id_kec": "317402", "id_kota": "3174", "kel_name": "KELURAHAN"
13    { "type": "Feature", "properties": { "GID": "G181471", "id_kel": "31.74.02.1006", "id_kec": "317402", "id_kota": "3174", "kel_name": "KELURAHAN"
14    { "type": "Feature", "properties": { "GID": "G181472", "id_kel": "31.71.06.1001", "id_kec": "317106", "id_kota": "3171", "kel_name": "KELURAHAN"
15    { "type": "Feature", "properties": { "GID": "G181473", "id_kel": "31.71.06.1001", "id_kec": "317106", "id_kota": "3171", "kel_name": "KELURAHAN"
16    { "type": "Feature", "properties": { "GID": "G181474", "id_kel": "31.71.06.1001", "id_kec": "317106", "id_kota": "3171", "kel_name": "KELURAHAN"
17    { "type": "Feature", "properties": { "GID": "G181475", "id_kel": "31.71.06.1001", "id_kec": "317106", "id_kota": "3171", "kel_name": "KELURAHAN"
18    { "type": "Feature", "properties": { "GID": "G181476", "id_kel": "31.71.06.1001", "id_kec": "317106", "id_kota": "3171", "kel_name": "KELURAHAN"
19    { "type": "Feature", "properties": { "GID": "G181477", "id_kel": "31.71.06.1001", "id_kec": "317106", "id_kota": "3171", "kel_name": "KELURAHAN"
20    { "type": "Feature", "properties": { "GID": "G181478", "id_kel": "31.71.06.1001", "id_kec": "317106", "id_kota": "3171", "kel_name": "KELURAHAN"
21    { "type": "Feature", "properties": { "GID": "G181479", "id_kel": "31.71.06.1001", "id_kec": "317106", "id_kota": "3171", "kel_name": "KELURAHAN"
22    { "type": "Feature", "properties": { "GID": "G181480", "id_kel": "31.71.06.1001", "id_kec": "317106", "id_kota": "3171", "kel_name": "KELURAHAN"
23    { "type": "Feature", "properties": { "GID": "G181481", "id_kel": "31.71.06.1001", "id_kec": "317106", "id_kota": "3171", "kel_name": "KELURAHAN"
24    { "type": "Feature", "properties": { "GID": "G181482", "id_kel": "31.74.01.1007", "id_kec": "317401", "id_kota": "3174", "kel_name": "KELURAHAN"
25    { "type": "Feature", "properties": { "GID": "G181483", "id_kel": "31.75.01.1005", "id_kec": "317501", "id_kota": "3175", "kel_name": "KELURAHAN"
26    { "type": "Feature", "properties": { "GID": "G181484", "id_kel": "31.75.01.1005", "id_kec": "317501", "id_kota": "3175", "kel_name": "KELURAHAN"
27    { "type": "Feature", "properties": { "GID": "G181485", "id_kel": "31.75.01.1005", "id_kec": "317501", "id_kota": "3175", "kel_name": "KELURAHAN"
28    { "type": "Feature", "properties": { "GID": "G181486", "id_kel": "31.75.01.1005", "id_kec": "317501", "id_kota": "3175", "kel_name": "KELURAHAN"
29    { "type": "Feature", "properties": { "GID": "G181487", "id_kel": "31.75.01.1005", "id_kec": "317501", "id_kota": "3175", "kel_name": "KELURAHAN"
30    { "type": "Feature", "properties": { "GID": "G181488", "id_kel": "31.75.01.1005", "id_kec": "317501", "id_kota": "3175", "kel_name": "KELURAHAN"
31    { "type": "Feature", "properties": { "GID": "G181489", "id_kel": "31.75.01.1005", "id_kec": "317501", "id_kota": "3175", "kel_name": "KELURAHAN"
32    { "type": "Feature", "properties": { "GID": "G181490", "id_kel": "31.75.01.1005", "id_kec": "317501", "id_kota": "3175", "kel_name": "KELURAHAN"
  ]
}
```

Untuk data yang digunakan pada project banjir merupakan geojson yang berisikan seperti diatas. Pentingnya data ini perlu diperhatikan!

5. Alur node-red



Pada gambar tersebut terdapat opsi dropdown, dan Ketika dropdown dipilih akan disimpan pada variable (flow) berikut konfigurasi yang dilakukan:



Lalu pada sentuhan terakhir ialah pemberian warna pada area yang terkena banjir

```

1 // Inisialisasi objek polygonCiliwung
2 var polygonCiliwung = {
3   name: "dki_grid_ciliwung",
4   layer: "dki_grid_ciliwung",
5   geojson: {
6     type: "FeatureCollection",
7     features: []
8   }
9 };
10 |

```

Kita menyiapkan sebuah objek geojson yang memiliki type:"FeatureCollection"

```

// Mendapatkan data geojson dari payload
var geojsonData = msg.payload.features;

```

```

// Mendapatkan data genangan dari variabel flow
const abt_genangan = flow.get("genangan") || null
const sodetanCiliwung = flow.get("sodetanCiliwung") || 0
const ciliwungLama = flow.get("ciliwungLama") || 0
const lateralFlow = flow.get("lateralFlow") || 0
const lag = flow.get("lag") || null

```

```

// Mendefinisikan skenario yang dipilih
const result = `${sodetanCiliwung}${ciliwungLama}${lateralFlow}`;

```

Kita mengambil semua flow dengan cara flow.get("nama_flow") Terdapat || 0 yang mengartikan jika tidak ada flow yang bernama lateralFlow maka variable lateralFlow akan bernilai 0

```

// Mendefinisikan skenario yang dipilih
const result = `${sodetanCiliwung}${ciliwungLama}${lateralFlow}`;

// Iterasi melalui data geojson
for (let i = 0; i < geojsonData.length; i++) {
  var feature = geojsonData[i];
  var properties = feature.properties;
  const geometry = feature.geometry;

  // Iterasi melalui data genangan
  for (let x = 0; x < abt_genangan.length; x++) {
    var abtData = abt_genangan[x];
    // Memeriksa apakah ID cocok
    if (properties.GID === abtData.gid) { // Membandingkan GID
      var genangan = null;

      // Memeriksa apakah lag > 0
      if (lag > 0) {
        switch (result) {
          case "000":
            genangan = abtData["genangan_sk1_lag" + lag]
            break;
          case "010":
            genangan = abtData["genangan_sk2_lag" + lag]
            break;
          case "001":
            genangan = abtData["genangan_sk3_lag" + lag]
            break;
          case "011":
            genangan = abtData["genangan_sk4_lag" + lag]
            break;
          case "111":
            genangan = abtData["genangan_sk5_lag" + lag]
            break;
          case "110":
            genangan = abtData["genangan_sk6_lag" + lag]
            break;
          case "101":
            genangan = abtData["genangan_sk7_lag" + lag]
            break;
        }
      }
    }
  }
}

```

Melakukan iterasi dan penyocokan pada id yang terdapat pada file excel yang disimpan pada flow abt_genangan

```

// Pemberian styling warna pada grid berdasarkan genangan
if (genangan === null) {
  properties['color'] = "transparent";
} else if (genangan >= 4 && genangan < 5) {
  properties['color'] = "orange";
} else if (genangan >= 5) {
  properties['color'] = "red";
} else if (genangan > 0) {
  properties['color'] = "yellow";
} else {
  properties['color'] = "transparent";
}

// Mengatur geometri fitur
if (geometry) {
  feature.geometry = geometry;
  polygonCiliwung.geojson.features.push(feature);
}

// Keluar dari perulangan setelah menemukan kecocokan
break;
}
}

```

Pemberian styling warna pada data yang memenuhi kondisi

```

// Konfigurasi map
var m = {
  overlay: "grid warna",
  geojson: polygonCiliwung.geojson,
  opt: {
    opacity: 0,
    fillOpacity: 0.6,
    gradient: true
  },
  fit: true
};

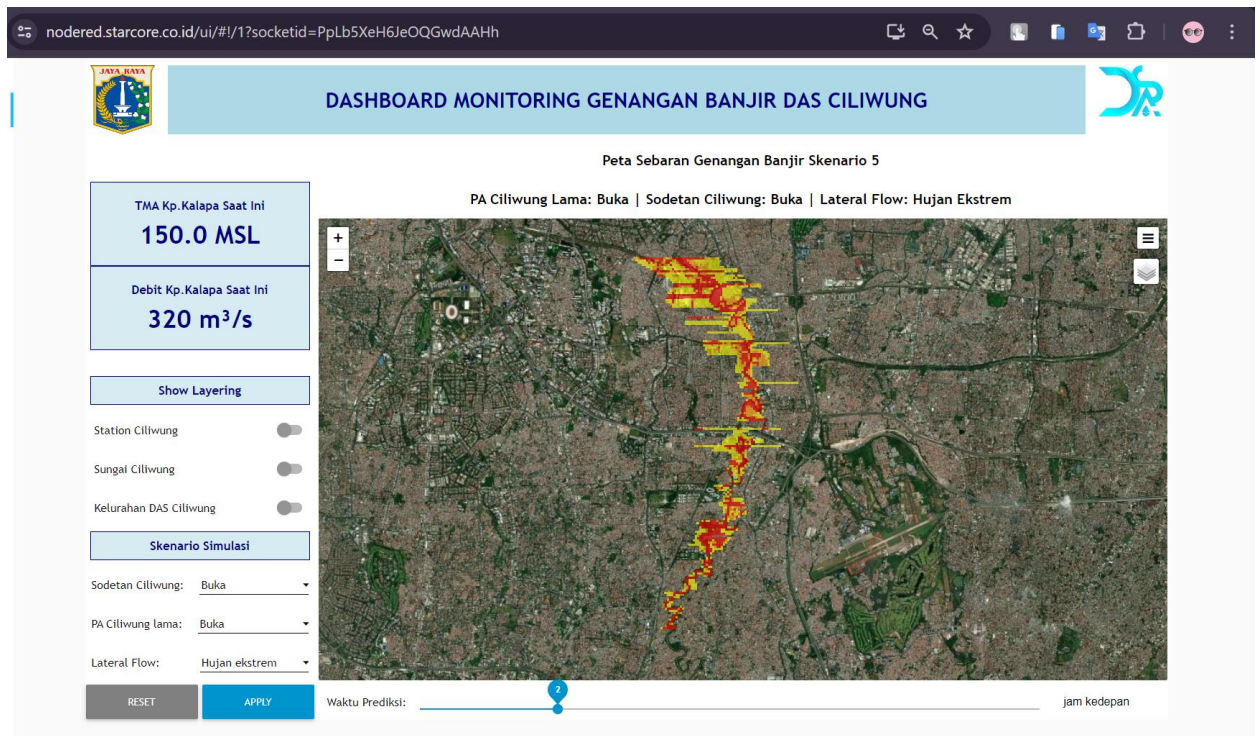
// Mengirimkan pesan dengan perintah map
msg.payload = {
  command: { map: m }
};

return msg;

```

Melakukan konfigurasi tambahan dan return msg

6. Deploy and running

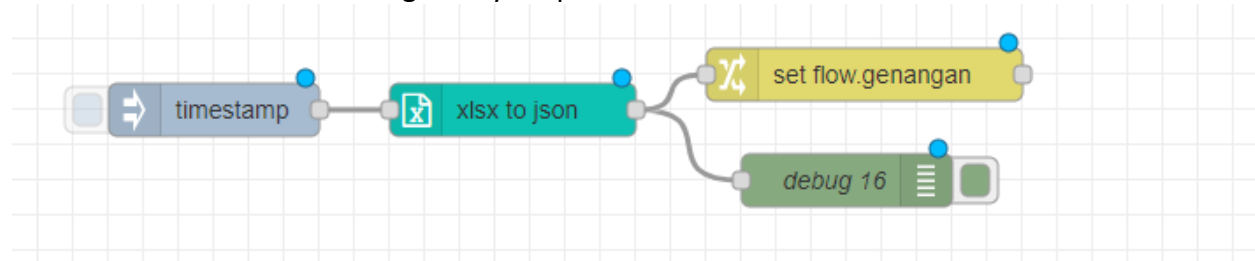


B. Source Code

1. Excel to Json

- Install package xlsx-to-json : <https://flows.nodered.org/node/node-red-contrib-xlsx-to-json>

Setelah install kalian bisa mengikutinya seperti ini:



Name	Name
Filepath	filename.xlsx
Rangecell	A1:C5
Columkey	{"A": "Title", "B": "Description"}
Sheet	Sheet 1

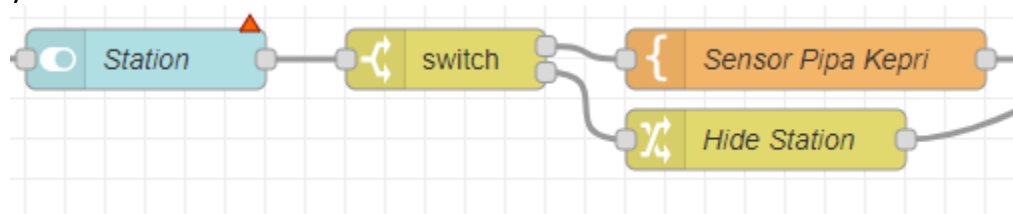
Seperti yang di tertulis terdapat kolom name,filepath,rangecell,columnkey,dan sheet yang akan konversi ke json. **Filepath** adalah tempat file itu disimpan di **local** bisa diisi sebagai berikut

Properties	
Name	xlsx to json
Filepath	C:\Users\moham\OneDrive\Dokumen\abt_genang
Rangecell	A2:ZN6908
Columkey	{"A": "Title", "B": "Description"}
Sheet	Sheet1

Pada kolom ColumKey tidak diisi yang artinya memilih semua kolom yang ada di excel di konversi ke json

2. Penggunaan switch

- Switch bisa berjalan jika memenuhi suatu kondisi, kita bisa memberinya 2 opsi yaitu true dan false



Flow akan berjalan sesuai kondisi pada station jika bernilai true maka flow atas akan berjalan jika bernilai false maka flow bawah akan berjalan

- Switch tidak hanya menerima 2 kondisi tetapi bisa menerima banyak kondisi

3. Pentingnya memahami konteks Subflow dan Subglobal

- Subflow atau Subglobal dapat membantu kita untuk menyimpan sebuah nilai mau itu string, integer, Boolean, function, ataupun objek
- Fungsi subflow sama seperti variabel yang mampu menampung sebuah nilai, serta cara pembuatan dan pemanggilan subflow pada function tergolong cukup mudah

```
70     }, 1000);  
71     flow.set("intervalID", intervalID);  
72 } else if (intervalID && statusPipa != "Semua") {  
73     clearInterval(intervalID);  
74     flow.set("intervalID", null);  
75 }  
76
```

Pada contoh di atas kita menggunakan `flow.set("nama_subflow", nilai)` pada baris atas saya memberi nilai menggunakan sebuah variabel dan pada baris bawah saya memberinya null

```
// Mendapatkan data genangan dari variabel flow  
const abt_genangan = flow.get("genangan") || null  
const sodetanCiliwung = flow.get("sodetanCiliwung") || 0  
const ciliwungLama = flow.get("ciliwungLama") || 0  
const lateralFlow = flow.get("lateralFlow") || 0  
const lag = flow.get("lag") || null
```

Serta pada contoh ini adalah cara kita mengambil subflow yang sudah di definisikan kita tamping pada variabel lalu bisa kita gunakan sesuai keperluan.