



Panduan Integrasi Node-RED ke Home Assistant via MQTT



1. Check Palette Manager & Instal Node Tambahan

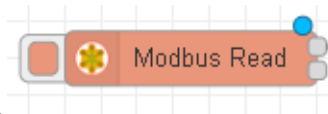
Sebelum memulai pastikan Node-RED sudah terinstal add-on penting:

1. Buka Flow Editor → menu ☰ → **Manage palette**
2. Pada tab Install, cari dan instal:
 - a. **node-red-node-mqtt**
 - b. **Node-red-contrib-modbus**
 - c. (Opsional) node-red-dashboard untuk UI
3. Tunggu proses instalasi, lalu Restart Node-RED.

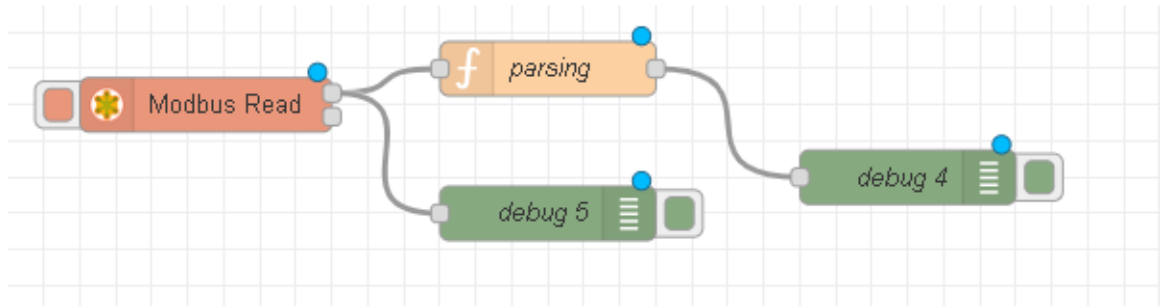
2. Import Flow JSON ke Canvas Node-RED

1. Unduh file .json flow dari GitHub Anda (contoh: ModBus_MQTT_Flow.json).
2. Di Flow Editor, klik ☰ → Import → tab Upload.
3. Pilih file .json → klik Import.
4. Drag flow yang muncul ke tab baru, lalu Deploy.

3. Membaca Data melalui ModBus (Modbus Read)



1. Tambahkan node modbus-read ke canvas.
2. Konfigurasi:
 - Server: koneksi ModBus TCP/RTU (cek ch serial yang digunakan oleh usb/modbus)
 - Data Type: HoldingRegister
 - Address: register awal (misal 1)
 - Quantity: jumlah register (misal 1)
 - Rate: interval polling (ms)
3. Sambungkan output dari modbus ke node debug (untuk melihat keluaran) dan function untuk parsing agar data yang didapat bisa diolah/digunakan::

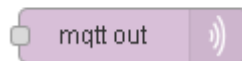


4. Karena output modbus read dalam bentuk json array, maka parsing cukup simpel dengan kode:

```
let data = msg.payload[0];
```

Data yang sudah didapat dapat divisualisasikan oleh ui gauge atau chart maupun diteruskan ke mqtt untuk home assistant.

4. Kirim Perintah/Data ke HA via MQTT



1. Siapkan node mqtt out:

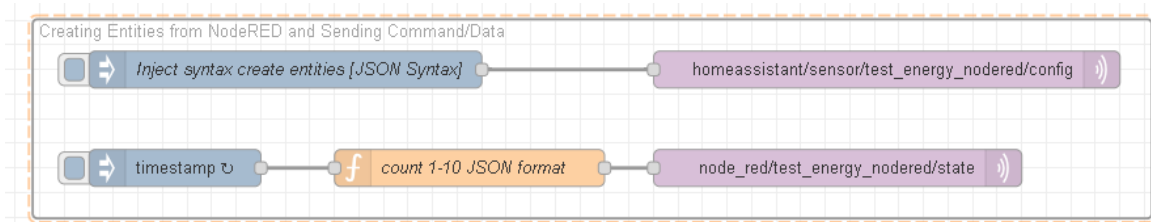
- Broker: koneksi MQTT Anda, contoh server **192.168.0.194** dan port **1883**. Untuk mqtt ferbos/hijau
- Topic: homeassistant/sensor/pressure_sensor/config (topic bisa disesuaikan dengan directory yang ingin dituju)
- Inject mqtt out dengan data dari sensor maupun node inject. Contoh pada flow ini adalah dengan node inject, masukan tipe data json dengan syntax:

```
{
  "name": "<your entity's friendly name>",
  "state_topic": "<your MQTT state topic>",
  "unit_of_measurement": "<unit, e.g. °C, kWh, %, etc.>",
  "value_template": "{{ value_json.<your_json_key> }}",
  "unique_id": "<a unique identifier for this entity>",
  "device": {
    "identifiers": [
      "<unique device identifier>"
    ],
    "name": "<device's friendly name>",
    "model": "<device model>",
    "manufacturer": "<manufacturer name>"
  }
}
```

2. Setelah discovery, gunakan mqtt out lain untuk state:

- Topic: node_red/pressure_sensor/state

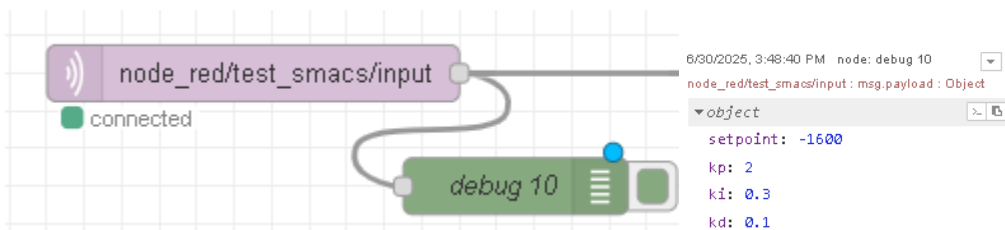
3. Setelah Deploy, Home Assistant secara otomatis akan membuat entity baru sesuai format inject dan entities pada Home Assistant tersebut akan otomatis ter-update jika ada data yang lebih baru .



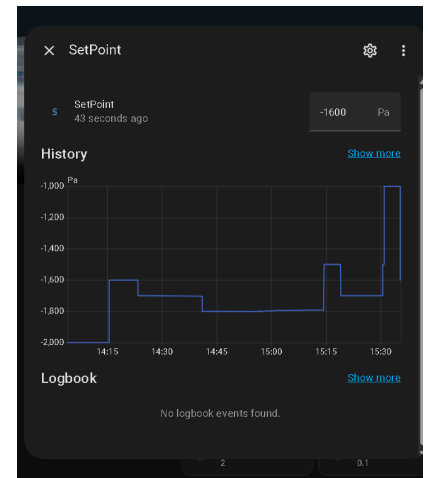
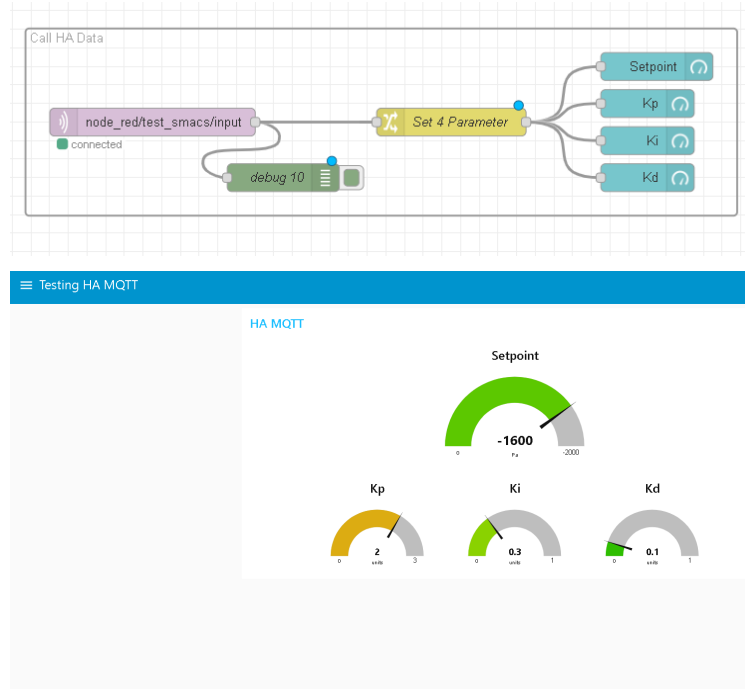
5. Terima Perintah/Data dari HA ke Node-RED via MQTT

1. Tambahkan node mqtt in:

- Topic: homeassistant/+ /set atau custom topic lainnya dari mqtt HA
- QoS: 0 atau 1
- Hubungkan ke node debug untuk melihat luaran mqtt.



- Didapatkan luaran berupa json dengan 4 parameter. Setiap 4 parameter ini dapat kita set dan visualisasikan
2. Sambungkan output ke node function / change untuk atur flow.context atau kirim ke PID. Maupun ke node ui seperti gauge untuk divisualisasikan

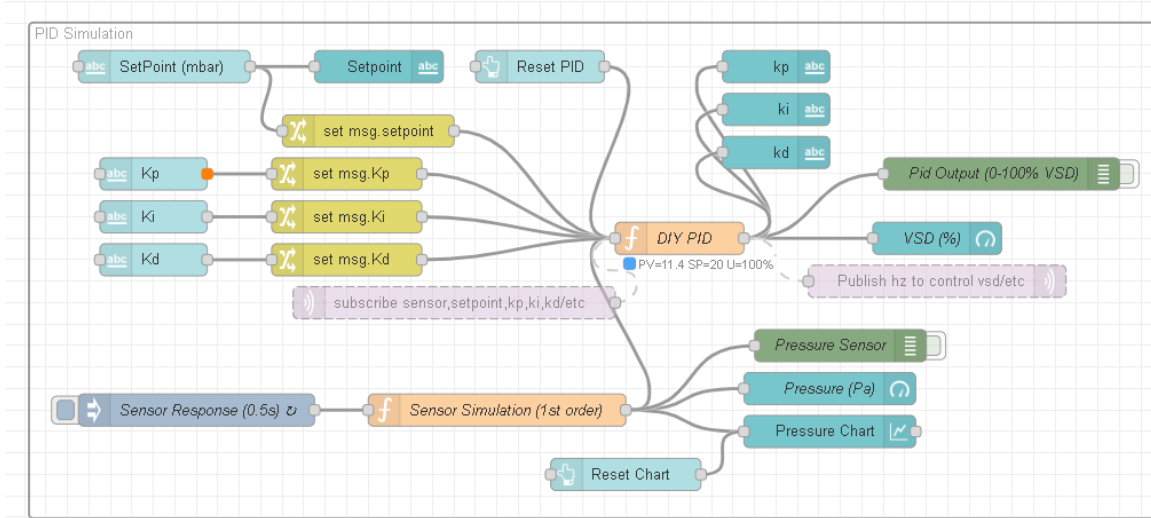


3. Jika parameter dari entity pada HA ini berubah, akan responsive diterima oleh nodered melalui mqtt dan terlihat perubahannya pada ui nodered

6. Extras: Simulasi PID Controller dengan Node-RED

A PID controller (Proportional–Integral–Derivative) adalah algoritma kontrol umpan balik yang sangat umum digunakan untuk menjaga variabel proses (seperti suhu, tekanan, kecepatan) agar tetap mengikuti **nilai setpoint** yang diinginkan.

Pada section ini kita akan mencoba mensimulasikan PID dengan node-red untuk mendapatkan solusi controller yang dapat dituning, edit, dan sesuaikan dengan berbagai sensor dan vsd



1. Import flow PID Simulation (PID_Controller_Simulation.json) ke Node-RED
2. Struktur alur:
 - Inject (tick interval) → Sensor Simulation → DIY PID → UI Gauges
 - Text Input / Button → Change → update msg.setpoint, msg.Kp, msg.Ki, msg.Kd
3. Atur parameter lewat Dashboard, lihat output VSD (%) dan Pressure (Pa).
4. Gunakan node debug untuk memantau error, P/I/D term, dan state flow.
5. Node yang penting disini adalah DIY PID berisi algoritma untuk mengatur VSD. Simulasi dilakukan dengan menginput nilai setpoint (tekanan yang diinginkan sistem), kp,ki,kd (koefisien PID).

