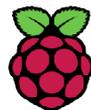


Panduan Penggunaan HMI Berbasis Raspberry Pi dengan Modbus dan Node-RED

1. Sekilas Tentang Raspberry Pi



Raspberry Pi



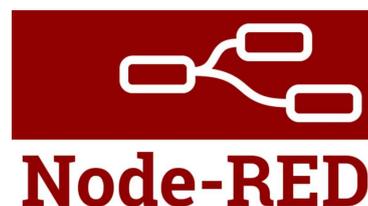
Raspberry Pi merupakan sebuah *Single Board Computer* (SBC) berukuran kecil yang ringkas.

Sama dengan Komputer pada umumnya, Raspberry Pi juga menggunakan Antarmuka pengguna Monitor, Keyboard, Mouse, dan dapat digunakan sebagaimana sebuah Komputer, antara lain: Menulis dokumen, Browsing Internet, Memutar Video, dan untuk Pemrograman.

Tidak seperti komputer yang berjalan pada sistem operasi **Windows**, Raspberry Pi berjalan pada **Raspbian OS**.



Dengan ukurannya yang ringkas, sebuah Raspberry Pi yang dipadukan dengan Monitor Touchscreen dapat difungsikan sebagai sebuah *Human Machine Interface* (HMI). Dalam kasus ini menggunakan **Modbus** sebagai protokol komunikasinya dan **Node-RED** sebagai *software*-nya.



2. Sekilas Tentang Modbus



Modbus adalah protokol komunikasi serial yang dipublikasikan pada tahun 1979 oleh **Modicon** (Sekarang bernama **Schneider Electric**). Nama Modbus sendiri merupakan akronim dari **Modicon Bus**.

Modbus pada awalnya diciptakan untuk diaplikasikan ke dalam *Programmable Logic Controller* (PLC), Namun saat ini Modbus sudah menjadi standar protokol yang umum dan banyak digunakan untuk menghubungkan peralatan elektronik industri.

Kelebihan menggunakan protokol komunikasi Modbus pada peralatan industri :

- Dipublikasikan secara terbuka dan bebas royalti.
- Mudah digunakan dan diimplementasikan.
- Kompatibel dengan berbagai vendor alat elektronik industri.
- Dalam proses perpindahan datanya dilengkapi *Error checking*

3. Jenis Modbus

- **Modbus RTU**

Merupakan varian modbus yang digunakan pada Port Serial **RS-485**. Modbus RTU merupakan implementasi protokol Modbus yang paling umum digunakan dikarenakan kemudahan dalam penggunaannya.

- **Modbus TCP/IP**

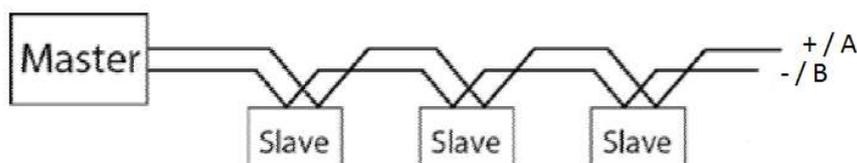
Merupakan varian Modbus yang digunakan pada jaringan TCP/IP. menggunakan Port **Ethernet**.

- **Modbus ASCII**

Digunakan pada komunikasi serial dengan memanfaatkan karakter ASCII.

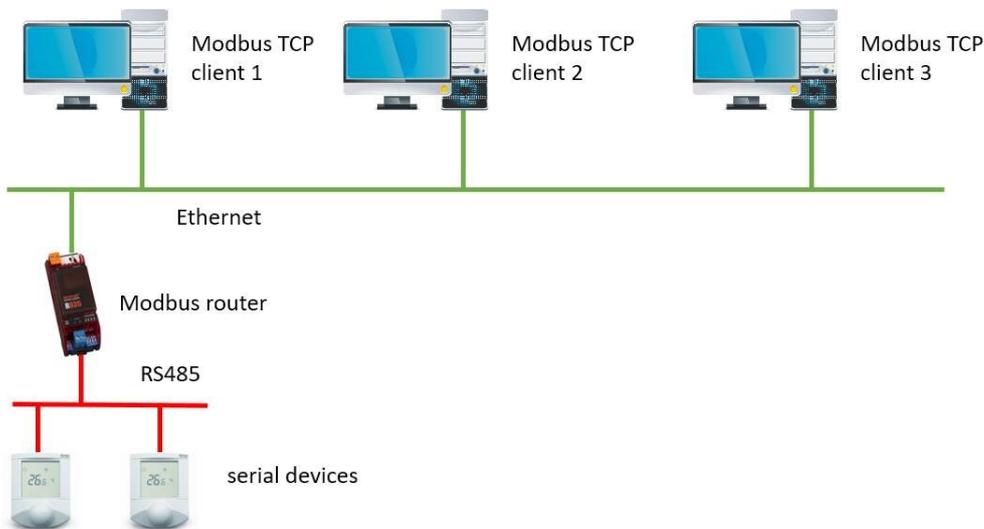
4. Topologi Modbus

1. Modbus RTU



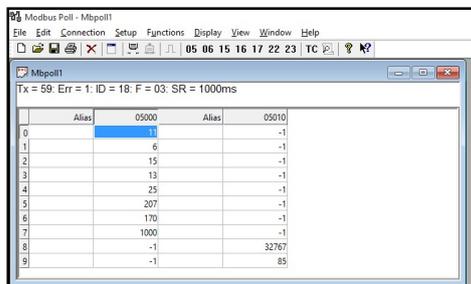
Modbus RTU berjalan diatas port RS-485 2 kabel (+/A atau -/B), **Master** merupakan pihak yang mengambil data/mengontrol, master umumnya berupa *Human Machine Interface*(HMI) atau PLC, sedangkan **Slave** merupakan pihak yang akan diambil datanya/dikontrol, umumnya berupa Sensor, Meter, atau Inverter/VFD. Satu master dapat terhubung dengan banyak slave sekaligus yang terhubung berurutan.

2. Modbus RTU dan TCP/IP



Modbus RTU dan Modbus TCP terhubung dalam satu jaringan dengan Modbus converter/router, untuk mengkonversi data dari Modbus RTU dengan Port RS-485 menjadi Modbus TCP/IP dengan Port Ethernet.

5. Modbus Polling



Modbus Polling bertujuan untuk memeriksa komunikasi modbus antara master dan slave device. Pemeriksaan yang dilakukan adalah pembacaan dan pengiriman data. Dengan melakukan polling, kita dapat memastikan sebuah device/alat dapat terbaca datanya sebelum berlanjut ke proses berikutnya.

Untuk melakukan modbus polling, diperlukan beberapa alat:

- Laptop/PC
- Konverter USB to RS-485
- Driver CH340
- *Software* Modbus Poll.

Link download Driver CH340 : <https://shorturl.at/otzV6>

Link Download *software* Modbus Poll : <https://shorturl.at/beuw3>

6. Konfigurasi Komunikasi Serial pada Master dan Slave

Ada beberapa parameter yang perlu dikonfigurasi antara Master dan Slave supaya bisa saling berkomunikasi, antara lain:

- **Baud Rate**

merupakan nilai frekuensi sinyal transmisi data pada komunikasi serial, antara master dan slave harus berada pada frekuensi yang sama untuk bisa berkomunikasi.

Contoh : 4800, **9600**, 19200, 115200

- **Parity**

digunakan pada pemeriksaan dalam komunikasi serial untuk menentukan apakah karakter data yang dikirimkan diterima dengan benar oleh perangkat lain. Sama seperti Baud Rate, antara master dan slave harus memiliki Parity yang sama.

Contoh : **8N1**, 8N2, 8E1, 8E2, 8O1, 8O2

- **Slave ID**

merupakan nomor untuk identifikasi slave device, setiap slave device yang terhubung tidak boleh memiliki slave id yang sama, nomor slave ID ada pada rentang 1-247.

7. Modbus Function

Function pada modbus merupakan sebuah perintah yang digunakan untuk berkomunikasi antara master dan slave, ada 4 function utama pada modbus:

Object type	Access	Size
Coil	Read-write	1 bit
Discrete input	Read-only	1 bit
Input register	Read-only	16 bits
Holding register	Read-write	16 bits

- **Coil**

function yang memiliki akses **membaca dan merubah data** yang ada pada slave, besar data yang dibaca 1 bit.

- **Discrete Input**

function yang memiliki akses **hanya membaca data** yang ada pada slave, besar data yang dibaca 1 bit.

- **Input Register**

function yang memiliki akses **hanya membaca data** yang ada pada slave, besar data yang dibaca 16 bit.

- **Holding Register**

function yang memiliki akses **membaca dan merubah data** yang ada pada slave, besar data yang dibaca 16 bit.

8. Modbus Address

Merupakan alamat yang berisikan parameter data pada sebuah device/alat. Setiap alat memiliki alamat yang berbeda-beda tergantung dari pembuat alat tersebut. Informasi mengenai address modbus umumnya terdapat pada manual book alat tersebut.

Berikut ini Modbus Adress dari unit VFD kit trainer:

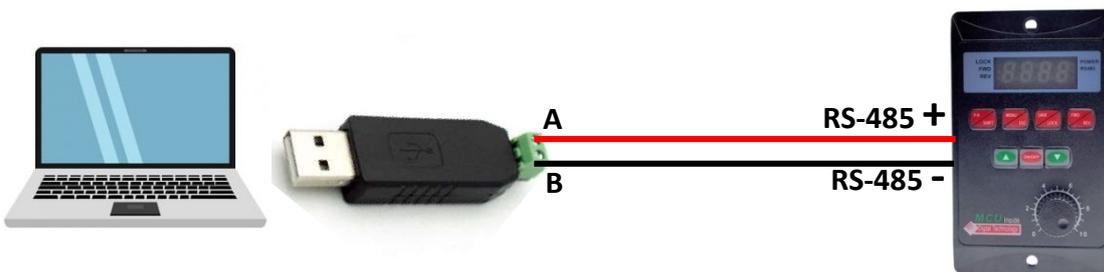
MODBUS-RTU Protocol Specification V1.3						
The data in the protocol include: hexadecimal number, integer, BCD code, floating point number						
Register address the attributes in the following table refer to the read-write attributes of data:						
		R-read	W-write	R/ W-read and write		
attribute	address(hex/w ord)	Register length(word)	data type	describe	explain	Remarks (for example)
Collect data information						
R	0	2	UINT	--	Display panel firmware version number	--
R	1	2	UINT	--	Power driver board firmware version number	--
R/W	2	2	UINT	Hz	Inverter speed setting and speed feedback	500/10=50.0Hz
R/W	3	2	UINT	--	Inverter start stop and running state feedback	1: forward 2: stop 5: reverse
R	4	2	--	--	--	spare
R	5	2	--	--	--	spare
R	6	2	--	--	--	spare
R	7	2	--	--	--	--
R	8	2	UINT	V	Bus voltage value	3100/10=310.0 V
R	9	2	UINT	A	Bus current value	132/100=1.32A
R	10	2	UINT	°C	Radiator temperature	43=43°C

Keterangan:

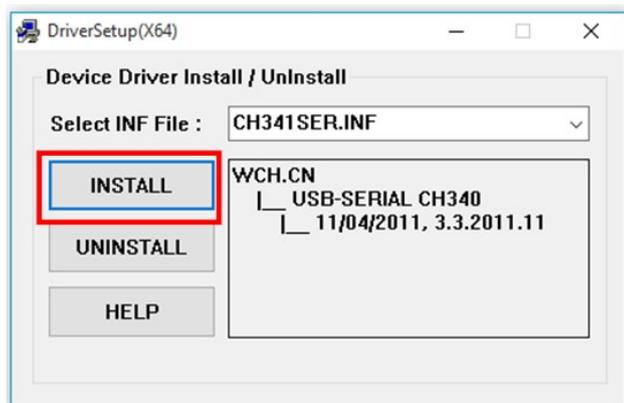
- **Attribute** : “R” menandakan bahwa alamat tersebut hanya **Read-only** (hanya bisa membaca data saja) , sedangkan “R/W” menandakan **Read-Write** (bisa membaca data dan merubah data).
- **Address** : merupakan nomor alamat data tersebut.
- **Register Length** : merupakan panjang data dalam satuan Bit.
- **Data Type** : merupakan type/jenis data.
- **Describe**: merupakan deskripsi alamat data (satuan ukur).
- **Explain** : merupakan keterangan dari alamat tersebut.
- **Remark** : merupakan keterangan tambahan, berisi informasi skala pengukuran.

9. Modbus Pooling dengan Software Modbus Poll

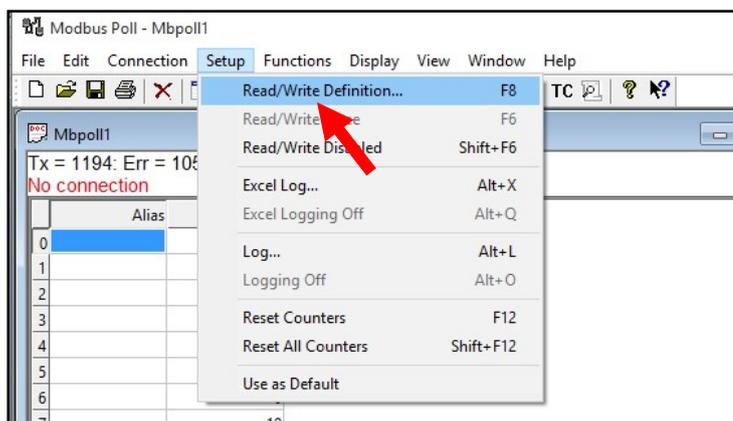
- a. Silahkan konfigurasi komunikasi serial pada alat yang akan ditarik datanya, parameter yang perlu disetting antara lain:
 - Baud rate : 9600
 - Parity : 8 None 1 (8N1)
 - Slave ID : 1
- b. Hubungkan alat dengan konverter RS-485 to USB, dengan skema rangkaian seperti gambar berikut:



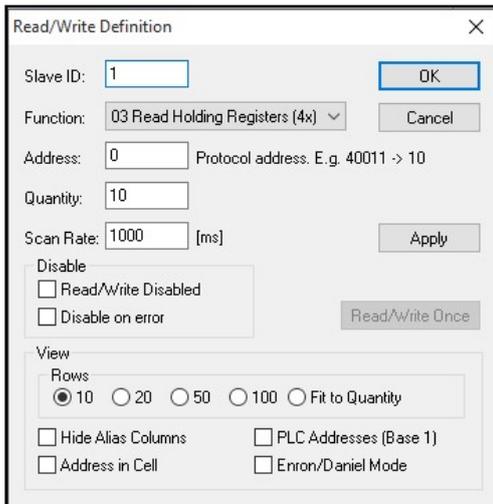
- c. Silahkan instal **driver CH340** pada laptop/PC anda, buka aplikasi installer kemudian klik install



- d. Jika sudah instal driver, silahkan Buka *Software* Modbus Poll kemudian klik **Setup > Read/Write Definition** pada atas layar

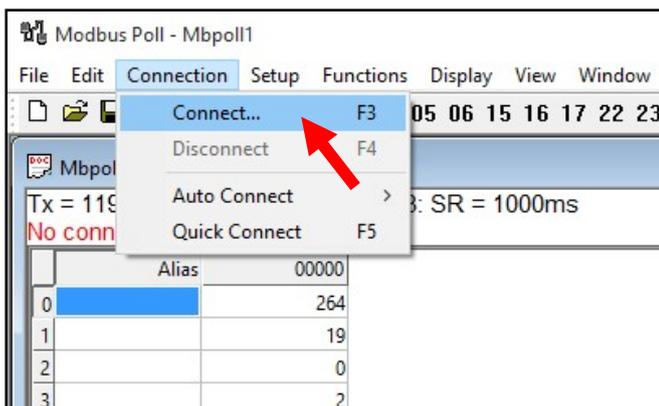


e. Akan terbuka halaman setup **Read/Write Definition**, silahkan setup sesuai berikut:

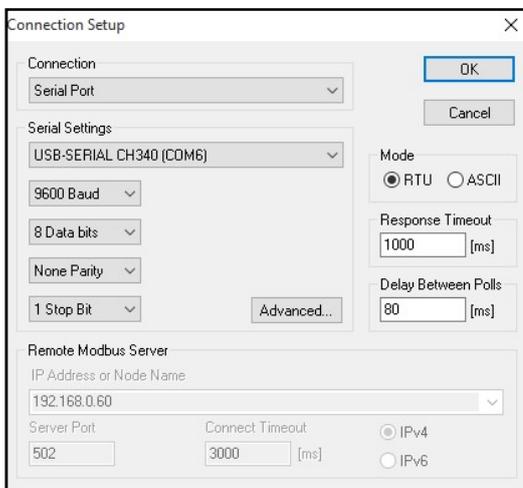


- **Slave ID** : 1
- **Function** : Read Holding Register (4x)
- **Address** : 0
- **Quantity** : 10

f. Silahkan klik **Connection > Connect** pada atas layar



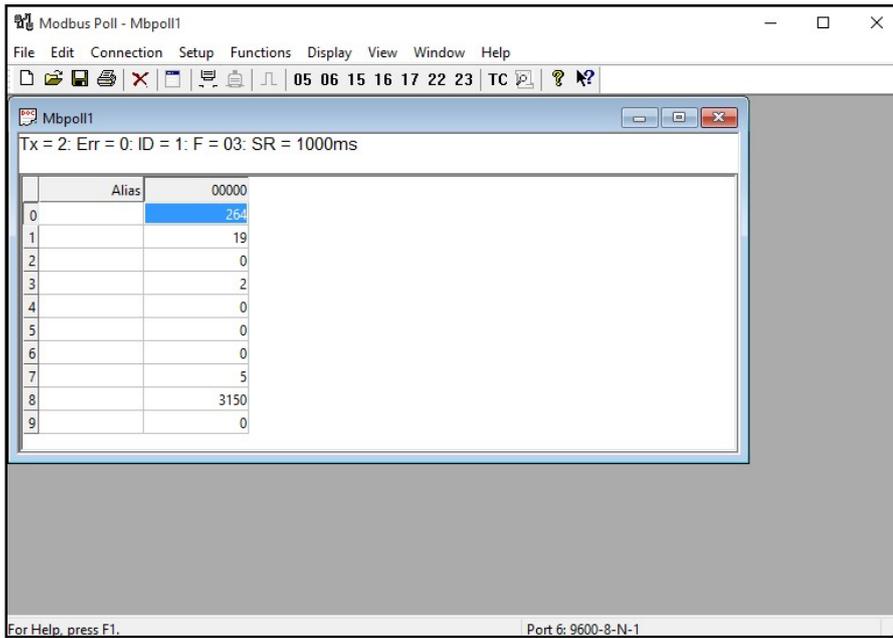
g. Akan terbuka halaman **Connection Setup**, silahkan setup sesuai berikut:



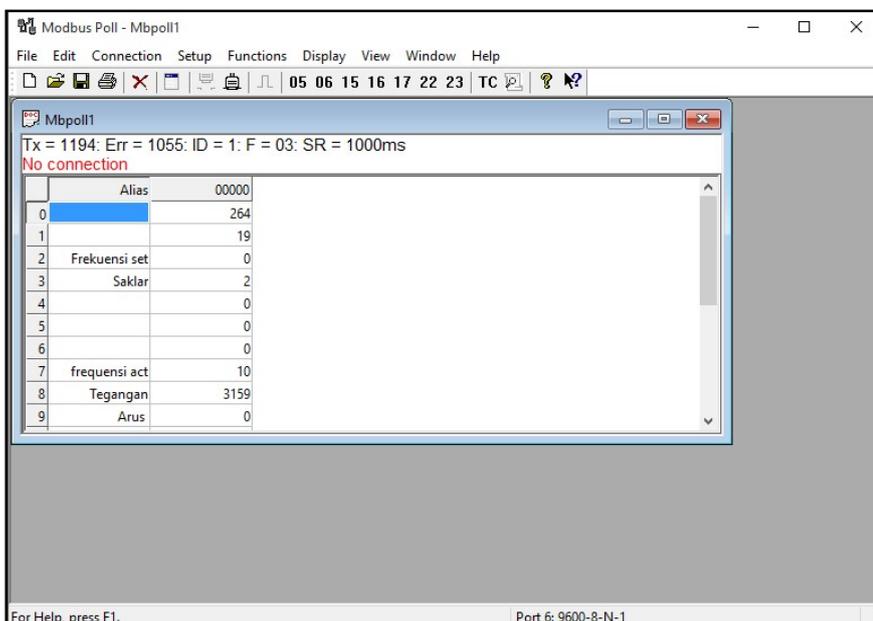
- **Mode** : RTU
- **Connection** : Serial Port
- **Serial Settings**: USB Serial Port (COM6)
- **Baud** : 9600
- **Data bit** : 8
- **Parity** : None
- **Stop Bit** : 1

Catatan: Nilai Baud, Data Bit, Parity, dan Stop bit disamakan dengan Alat.

h. Data berhasil terbaca oleh modbus Poll

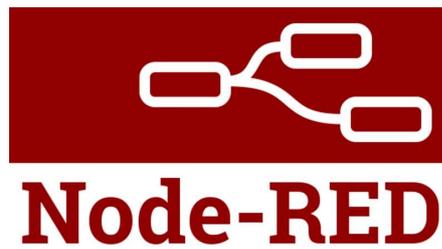


i. Menambahkan nama pada kolom Alias agar data mudah terbaca



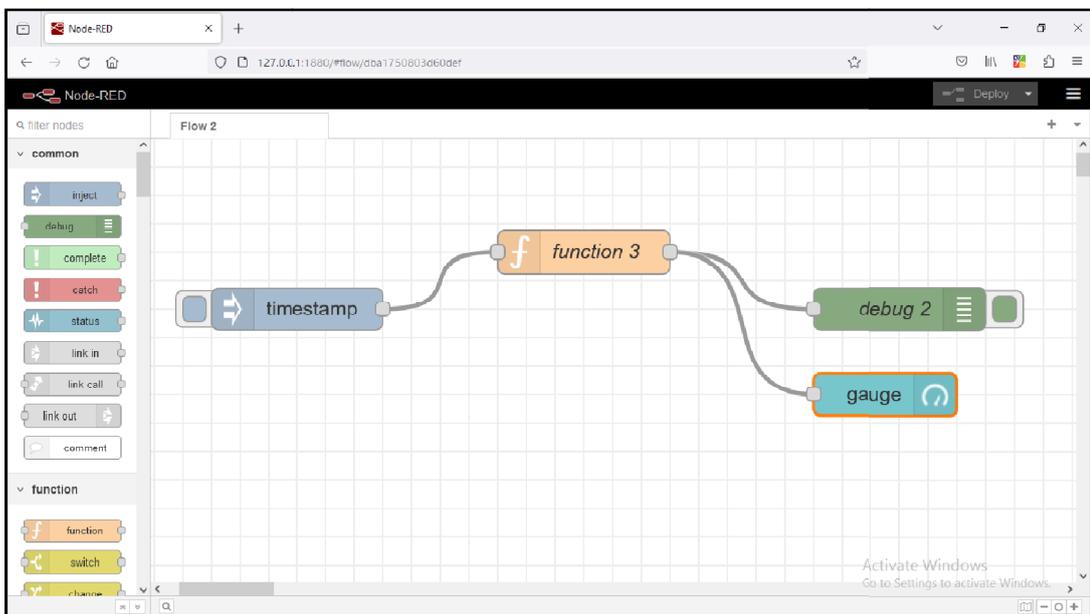
Link download program modbus poll: <https://rb.gy/he2jil>

10. Sekilas Tentang Node-RED



Node-RED adalah sebuah tool berbasis **browser** untuk membuat aplikasi Internet of Things (IoT) yang mana lingkungan pemrograman visualnya berupa **flow**.

Flow ini terbentuk dari beberapa **node** yang saling berhubungan di mana tiap node melakukan tugas tertentu.



Halaman Node-RED Editor

11. Fitur yang dimiliki node-RED

- Node-RED dapat digunakan sepenuhnya secara gratis.
- Pembuatan program yang intuitif dan relatif lebih mudah.
- Dapat berjalan secara *local* (Laptop/PC) maupun secara *Cloud* (internet)
- Dukungan dari komunitas yang sudah banyak beredar di internet.

12. Instalasi Node-RED Pada Laptop/PC Windows

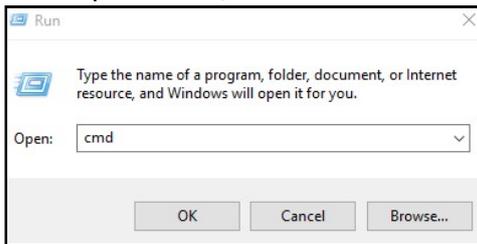
1. Untuk bisa menjalankan Node-RED diperlukan **Node.js**, silahkan download dan lakukan instalasi Node.js pada Laptop/PC.

Untuk Mendownload silahkan klik link berikut:

System Windows 32-bit : <https://shorturl.at/bioG1>

Sytem windows 64-bit : <https://shorturl.at/hoIU8>

2. Buka **Command Prompt** dengan menekan tombol **Windows + R**, kemudian ketikkan **cmd** pada kolom pencarian, kemudian tekan **OK**.



3. Setelah command Prompt terbuka, ketikkan **node -v** kemudian klik **Enter** untuk memeriksa apakah node.js telah terinstall pada laptop. jika instalasi sukses maka akan muncul nomor versi dari Node.js pada command prompt.

```
C:\WINDOWS\system32\cmd.exe
C:\>node -v
v10.15.3
C:\>
```

4. Ketikkan **npm install -g --unsafe-perm node-red** untuk menginstall node-red, tunggu hingga instalasi selesai.

```
C:\WINDOWS\system32\cmd.exe
C:\>node -v
v10.15.3
C:\>npm install -g --unsafe-perm node-red
```

13. Menjalankan Node-RED pada Laptop/PC Windows

1. Silahkan Buka **Command Prompt** kemudian Ketikkan **node-red** kemudian tunggu sampai proses selesai, jika proses sudah selesai, mohon jangan menutup command prompt, cukup di **minimize** saja.

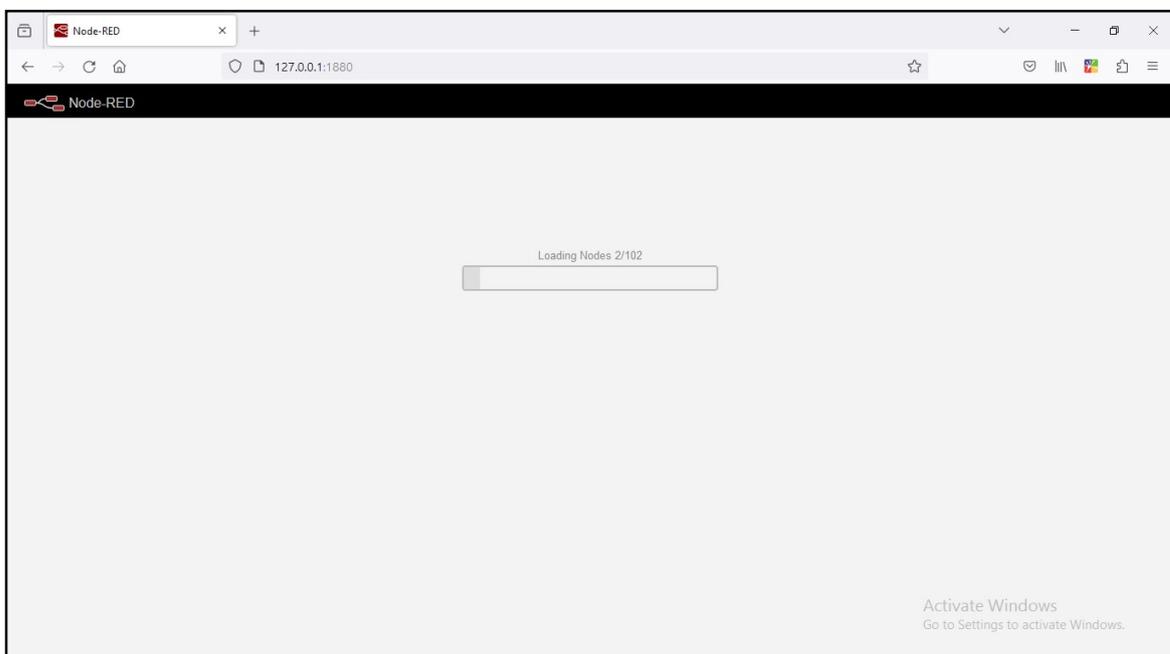
```

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\...\.node-red\settings.js
15 May 20:28:40 - [info] Context store : 'default' [module=memory]
15 May 20:28:40 - [info] User directory : \Users\...\.node-red
15 May 20:28:40 - [warn] Projects disabled : editorTheme.projects.enabled=false
15 May 20:28:40 - [info] Flows file : \Users\...\.node-red\flows_1...\.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.

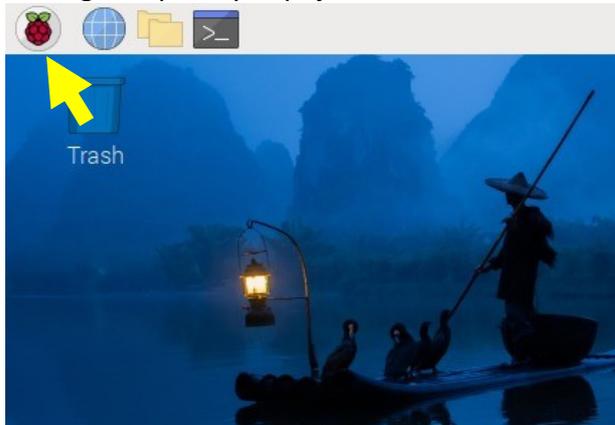
You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----
15 May 20:28:40 - [info] Starting flows
15 May 20:28:40 - [info] Started flows
15 May 20:28:40 - [info] Server now running at http://127.0.0.1:1880/
    
```

2. Silahkan Buka Browser (Google Chrome atau Mozilla Firefox) kemudian Ketikkan **http://127.0.0.1:1880/** pada kolom pencarian.

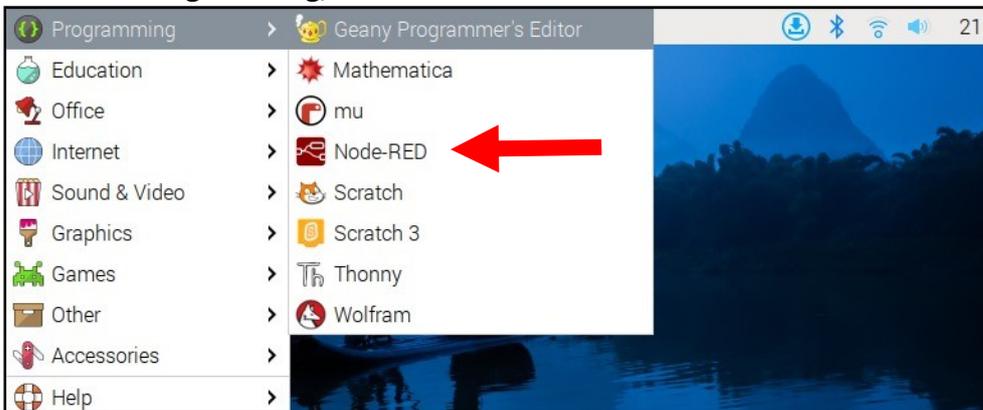


14. Menjalankan Node-RED pada HMI Raspberry Pi

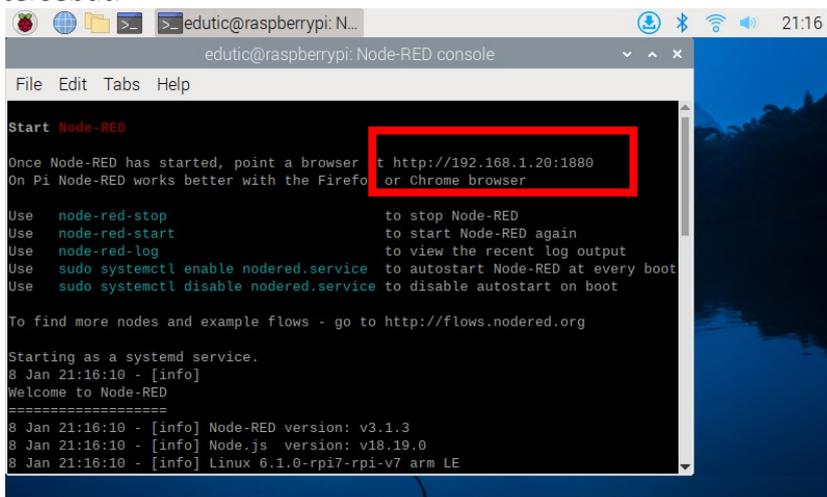
1. Klik logo raspberry di pojok kiri atas



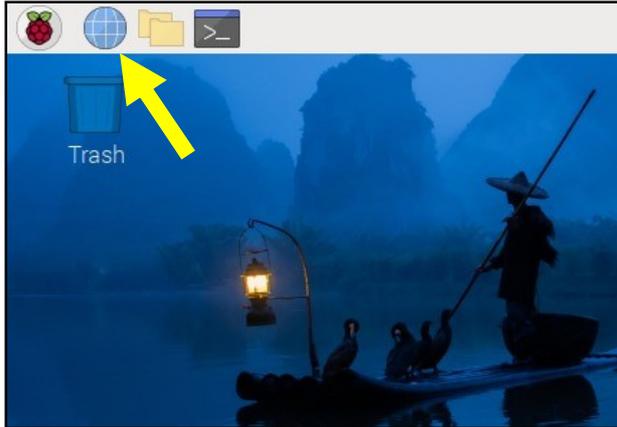
2. Klik Menu **Programming**, kemudian Pilih **Node-RED**



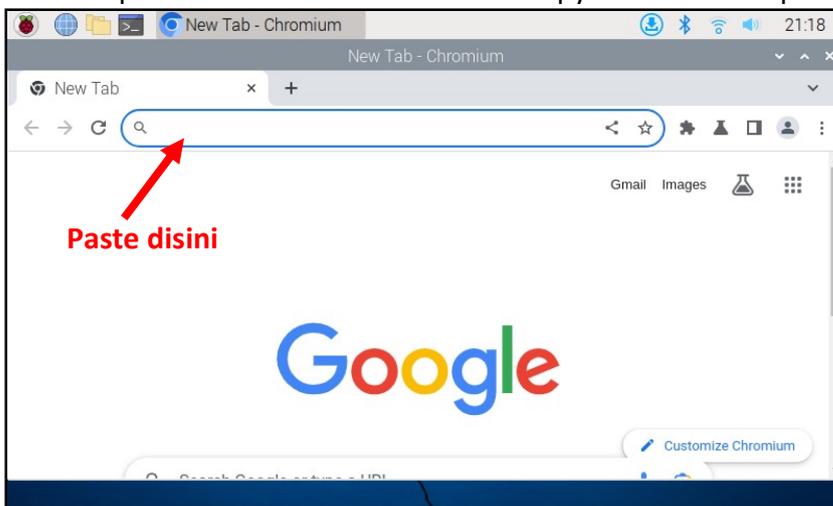
3. Akan terbuka sebuah halaman **Command Prompt**, pada halaman tersebut terdapat alamat IP (<http://192.168.X.X:1880>) yang menjadi alamat Node-RED, silahkan copy alamat tersebut.



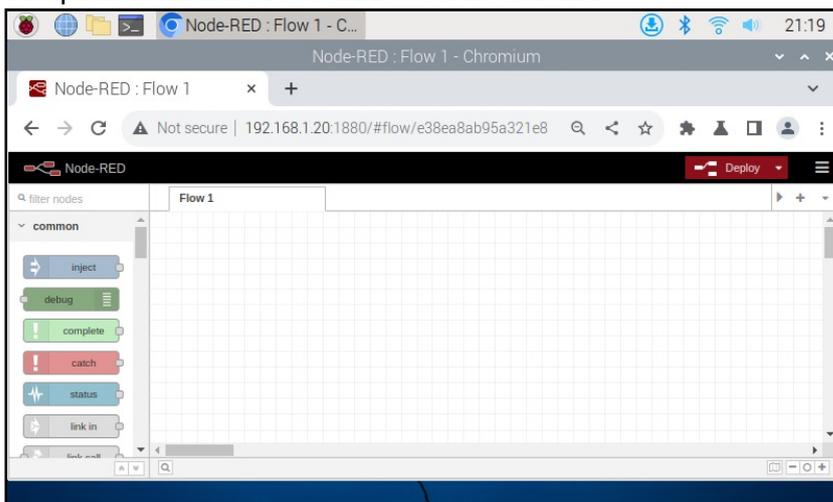
4. Silahkan kembali ke halaman utama kemudian buka **Browser**



5. Silahkan paste alamat node red telah di copy tadi ke kolom pencarian kemudian klik **enter**.



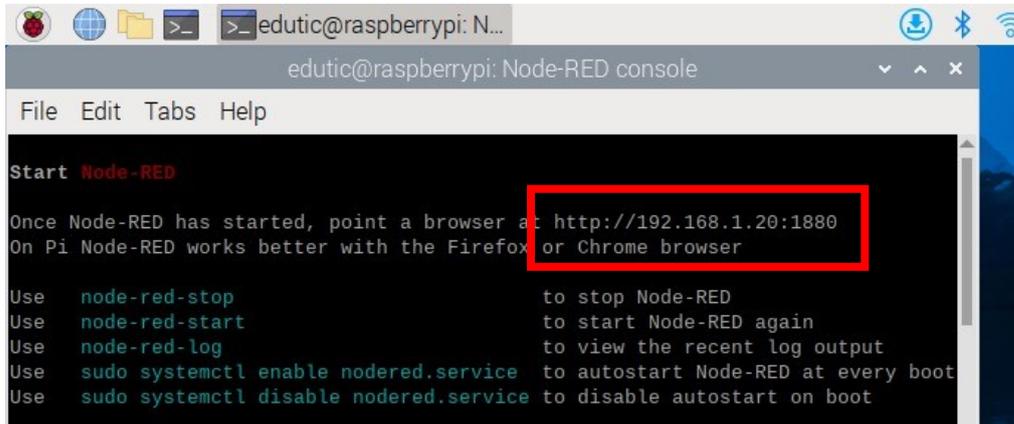
6. Tampilan Halaman Node-RED setelah terbuka



15. Mengakses Node-RED HMI dari Laptop/PC

Catatan: Raspberry Pi dan Laptop harus terhubung pada jaringan WiFi yang sama.

1. Silahkan Buka Node-RED pada HMI, pada halaman node-RED terdapat alamat IP (<http://192.168.X.X:1880>) yang menjadi alamat Node-RED, silahkan catat alamat IP tersebut.



```

Start Node-RED

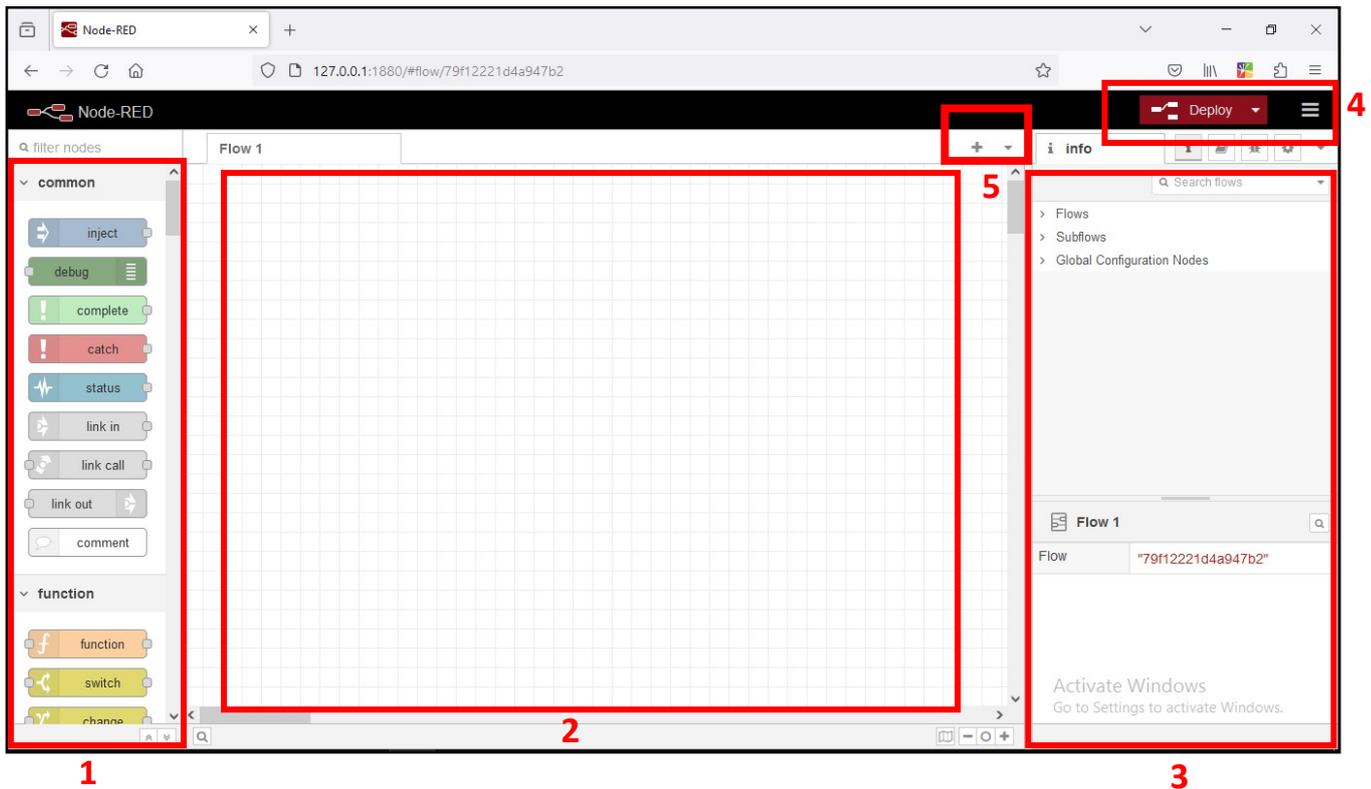
Once Node-RED has started, point a browser at http://192.168.1.20:1880
On Pi Node-RED works better with the Firefox or Chrome browser

Use node-red-stop           to stop Node-RED
Use node-red-start          to start Node-RED again
Use node-red-log            to view the recent log output
Use sudo systemctl enable nodered.service to autostart Node-RED at every boot
Use sudo systemctl disable nodered.service to disable autostart on boot
    
```

2. Silahkan buka **Browser** pada PC/Laptop anda, kemudian tuliskan alamat IP Node-RED tadi ke kolom pencarian browser, kemudian Klik **Enter**



16. Bagian Node-RED Editor

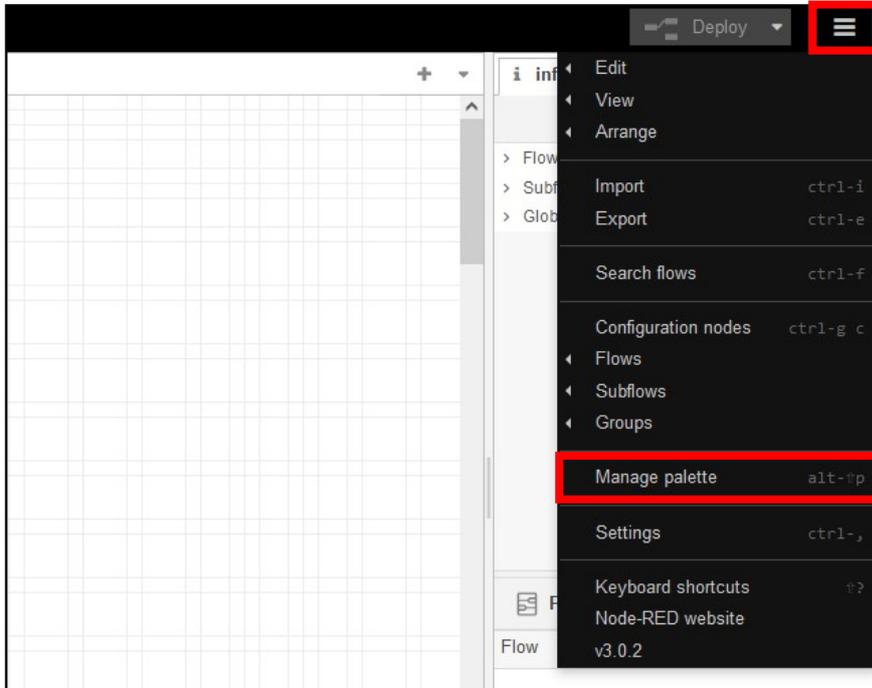


Keterangan:

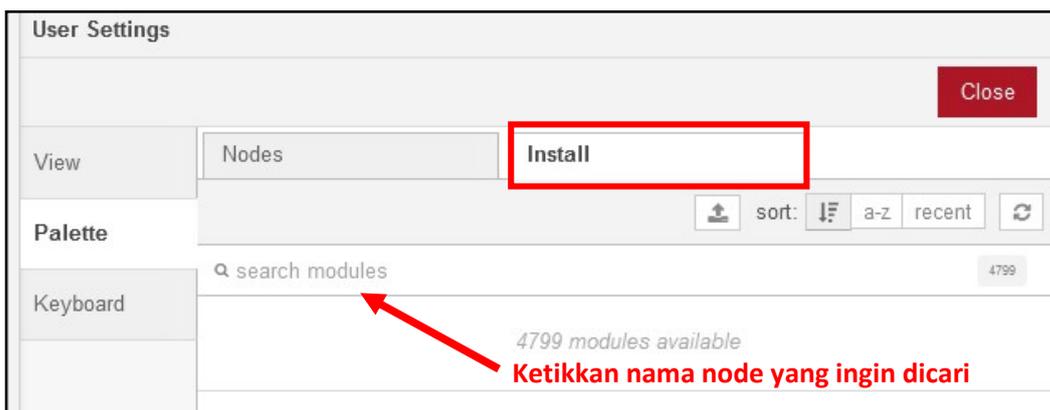
1. **Node Palette** : berisikan node-node yang dapat digunakan untuk membuat program tertentu.
2. **Halaman Utama** : tempat untuk meletakkan node dan menghubungkan node-node tersebut sehingga tercipta alur kerja program yang diinginkan.
3. **Halaman Sekunder** : untuk pengaturan halaman Dashboard, Flow, dan Debug.
4. **Deploy dan Setting** : untuk mendeploy/menyimpan ubahan pada flow program node-RED.
5. **Add (+)** : Untuk menambah/membuat Flow Program baru.

17. Mendownload/Menambahkan Node Palette Baru

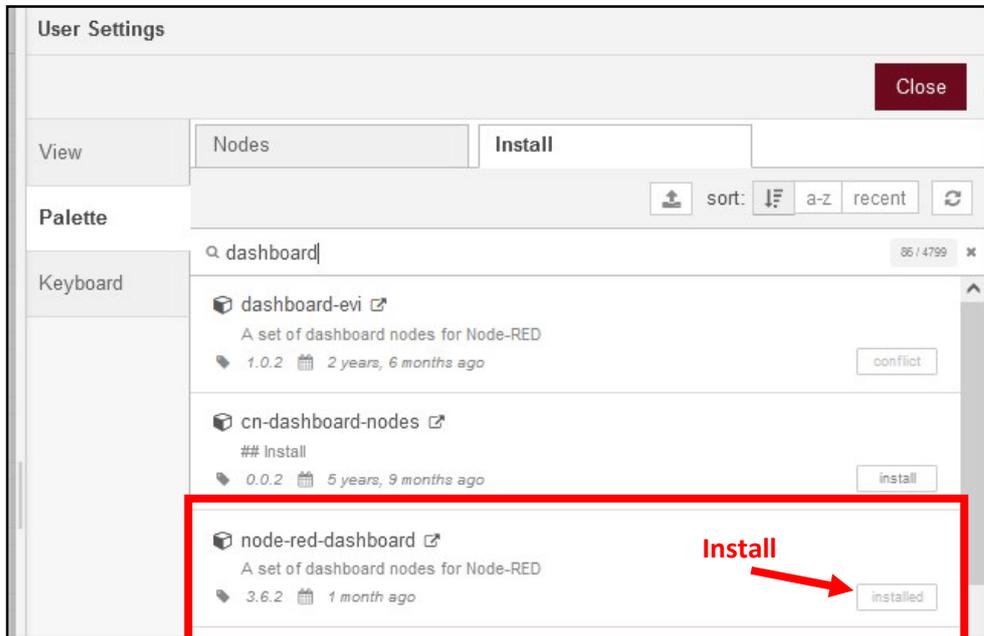
1. Silahkan Klik **gambar 3 garis** pada pojok kanan atas, kemudian pilih **Manage Palette**, atau dengan jalan pintas menekan (**Alt + Shift + P**)



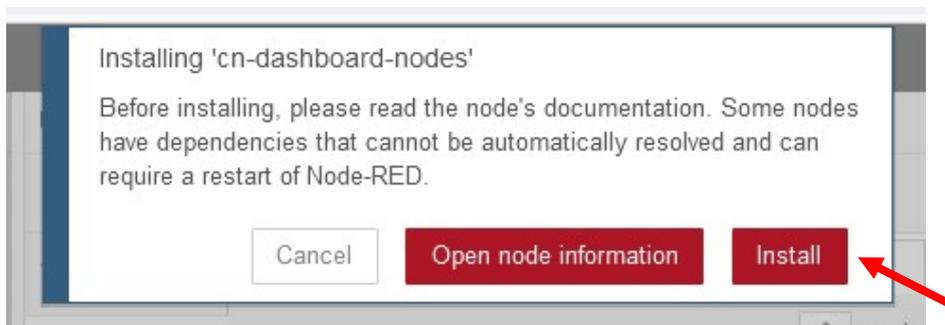
2. Klik halaman **Install**, kemudian ketikkan nama node yang ingin anda tambahkan pada kolom pencarian.



- Setelah menuliskan nama node, maka akan muncul daftar Node, silahkan klik **instal** pada node yang ingin di install, dalam contoh ini adalah menginstall **node-red-dashboard**.

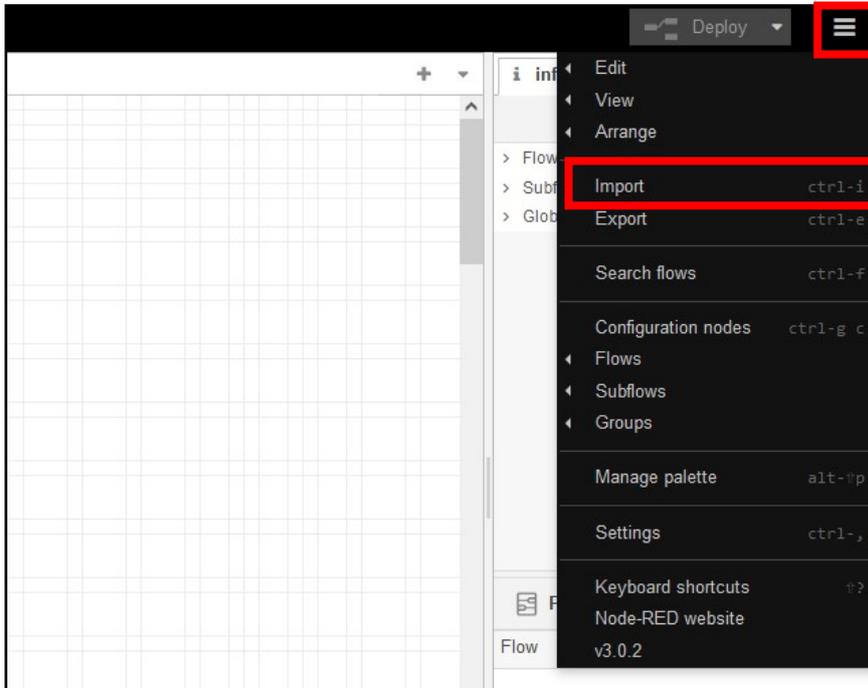


- Akan muncul panel Notifikasi, Silahkan Klik **Install**

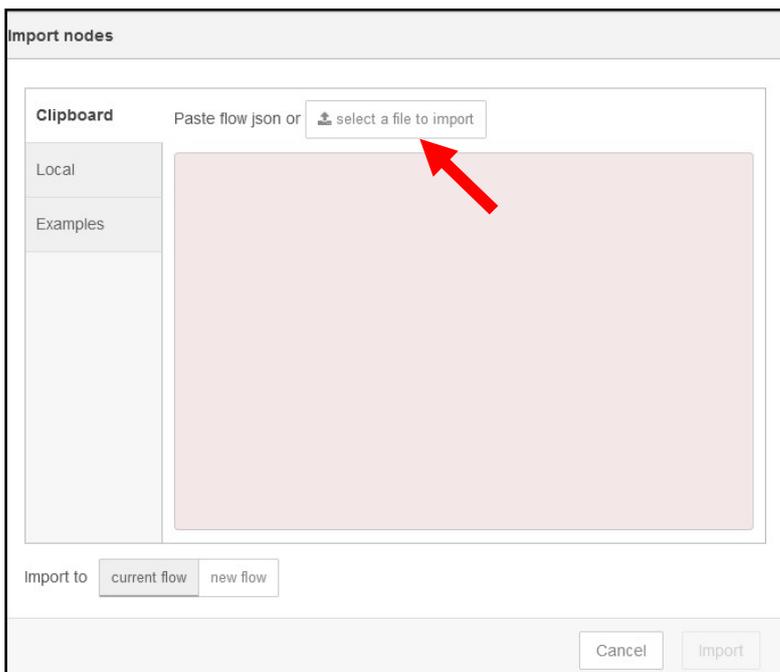


18. Import Flow Program pada Node-RED

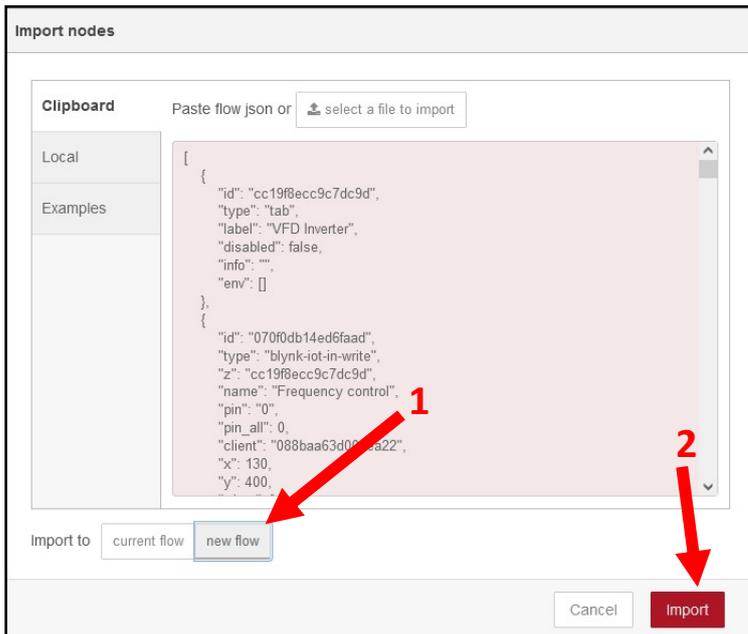
1. Klik **gambar 3 garis** pada pojok kanan atas, kemudian pilih **Import**, atau dengan jalan pintas menekan (**Ctrl + i**).



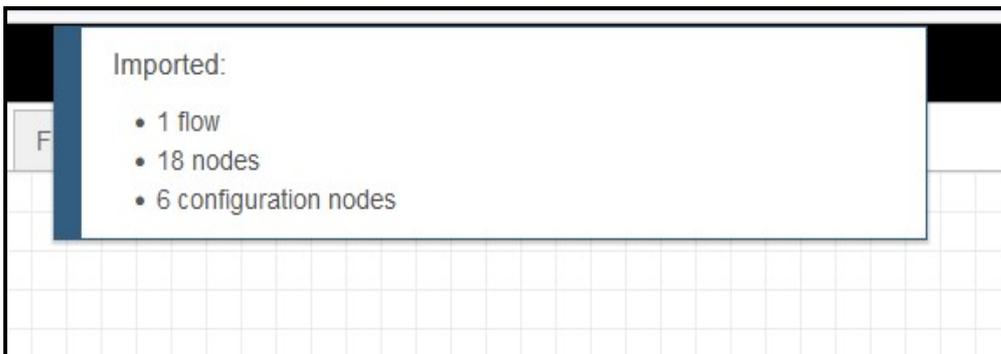
2. Akan terbuka Halaman Import Node, silahkan klik **select file to import**, untuk memasukkan file yang ingin di import. Kemudian pilih file program berformat . JSON yang akan diupload.



3. Untuk opsi inport to, pilih **new Flow (panah 1)**, kemudian klik **Inport (panah 2)**

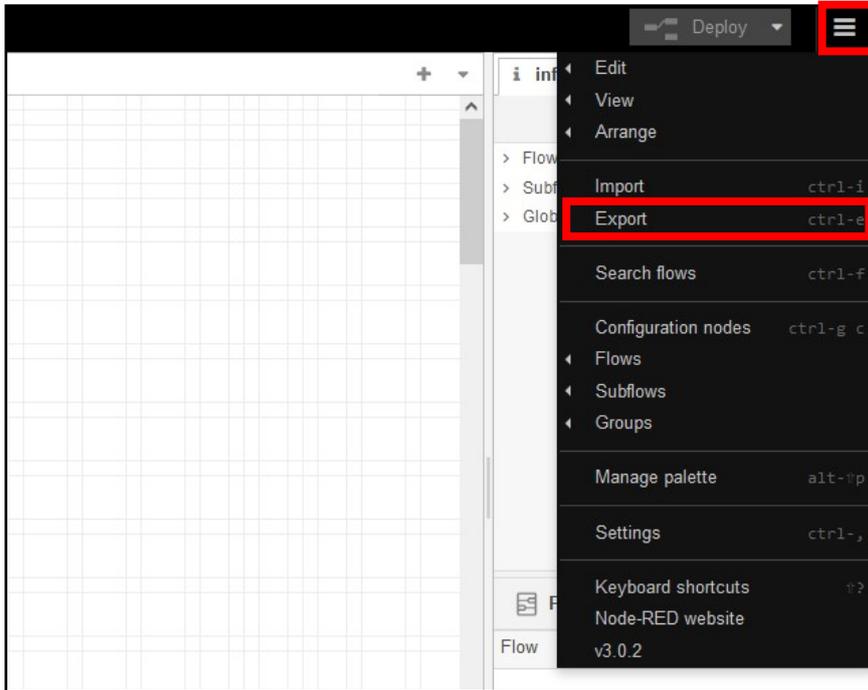


4. Akan muncul notifikasi yang menandakan Inport Flow Program Sukses

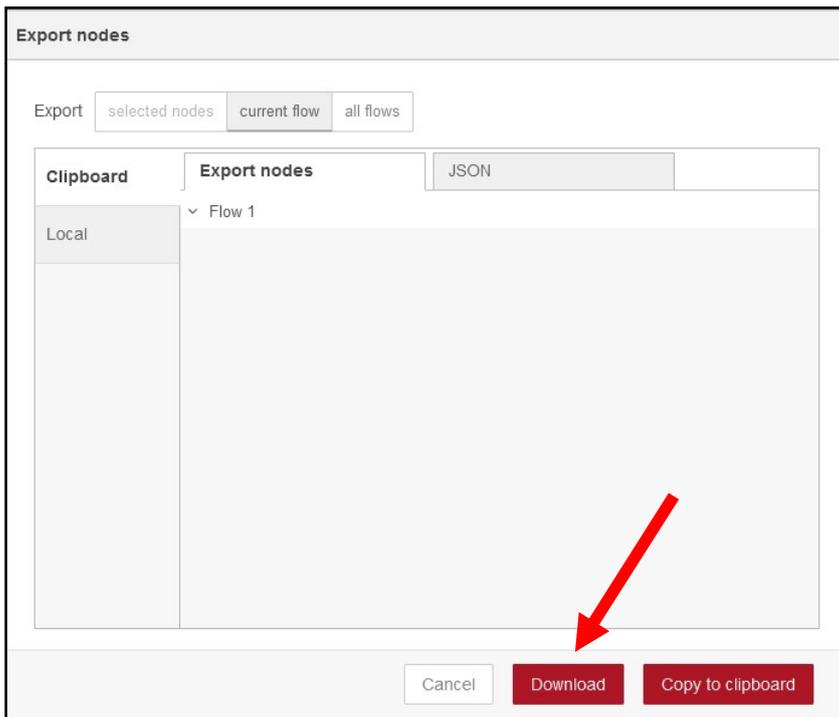


19. Export Flow Program pada Node-RED

1. Klik **gambar 3 garis** pada pojok kanan atas, kemudian pilih **Export**, atau dengan jalan pintas menekan (**Ctrl + E**).

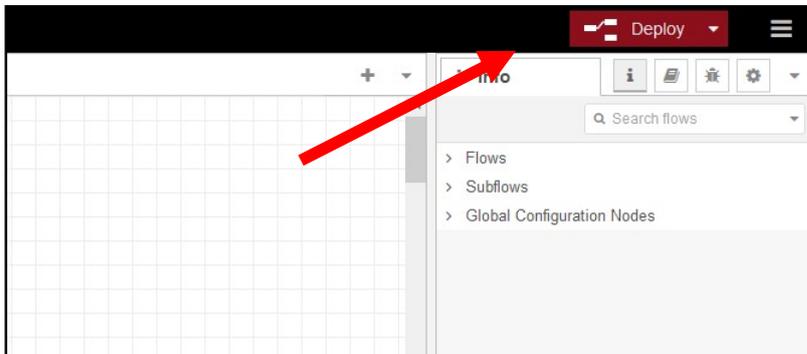


2. Akan Terbuka menu **Export Nodes**, silahkan Klik tombol download untuk export/download Flow Program.



20. Membuka Dashboard Node-RED

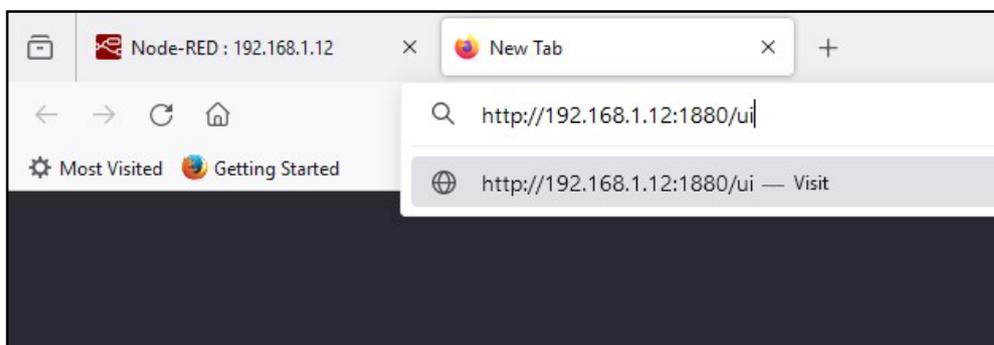
1. Silahkan klik **Deploy** pada bagian kanan atas layar untuk menyimpan perubahan program.



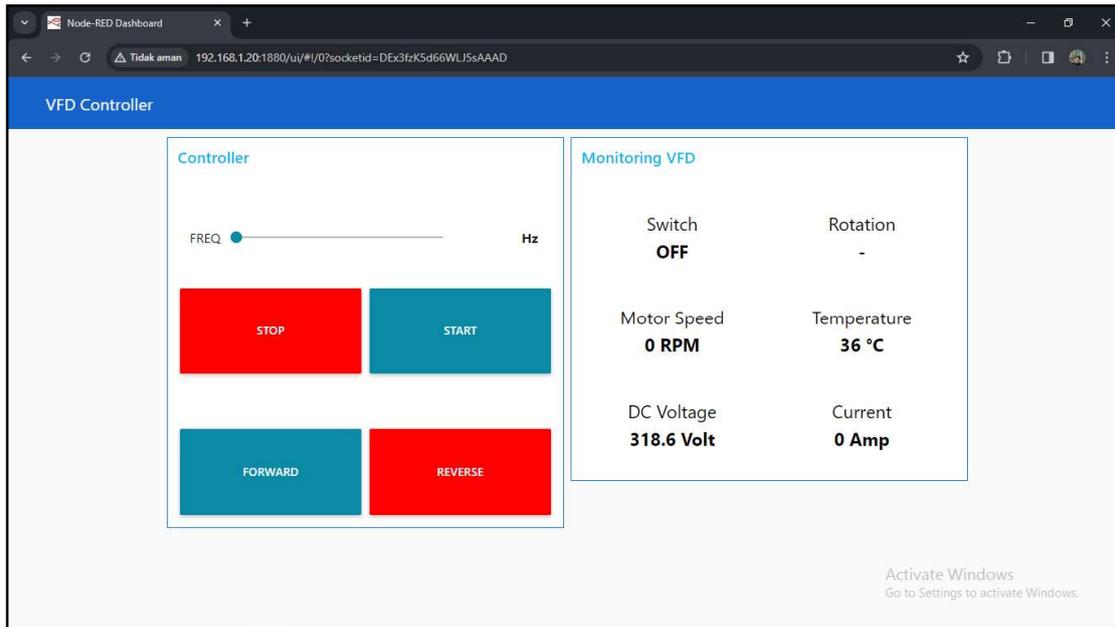
2. Pada halaman utama silahkan copy alamat IP address pada bagian atas (Kolom pencarian)



3. Buka halaman web baru, kemudian pastekan alamat IP tersebut disertai **/ui**.
Contoh penulisan : <http://192.168.1.12:1880/ui> .



4. Dashboard Node-RED setelah terbuka



21. Pengenalan HTTP dan MQTT



HTTP (*Hypertext Transfer-Transfer Protocol*) merupakan sebuah protokol jaringan lapisan aplikasi yang umumnya digunakan dalam membantu proses pertukaran data di internet antar komputer yang satu dengan yang lainnya. Data ini dapat berupa file, gambar, maupun video. Protokol ini juga menggunakan hipermedia yang kemudian dihubungkan dengan link atau disebut dokumen hypertext yang membentuk WWW atau World Wide Web.

MQTT (*Message Queuing Telemetry Transport*) adalah protokol pesan berbasis standar atau seperangkat aturan, yang digunakan untuk komunikasi *machine-to-machine* dan Sensor pintar. Perangkat *Internet of Things* (IoT) biasanya harus mengirim dan menerima data melalui jaringan dengan sumber daya dan bandwidth terbatas. Perangkat IoT ini menggunakan MQTT untuk transmisi data, karena mudah diterapkan dan dapat mengkomunikasikan data IoT secara efisien. MQTT mendukung pengiriman pesan antara perangkat ke *cloud* dan *cloud* ke perangkat.

22. Perbandingan HTTP dengan MQTT

Parameter	MQTT	HTTP
Penggunaan Daya	Ringan dalam penggunaan sumber daya	Lebih berat dalam penggunaan sumber daya
Model Komunikasi	Publish-subscribe	Request-response
Overheard Protokol	Rendah	Tinggi
Skalabilitas	Sangat luas untuk menangani perangkat dan koneksi berjumlah banyak	Memiliki jumlah perangkat dan koneksi yang terbatas
Kontrol Koneksi	Dikelola oleh server MQTT (broker)	Koneksi harus dibuat dan dijaga oleh klien
Efisiensi	Cocok untuk protokol IoT	Cocok untuk aplikasi web

23. Bagian penting pada MQTT

- **Publisher**, Merupakan pihak yang mempublish/mengirim data.
- **Subscriber**, Merupakan pihak yang mensubscribe/menerima data.
- **Broker**, Merupakan pihak yang menjadi **mediator** antara publisher dan subscriber, Broker akan Menerima pesan dari publisher, memfilter pesan tersebut, mencari tahu subscriber dengan topik tertentu, kemudian Mengirim pesan tersebut ke subscriber.

Terdapat broker publik yang dapat digunakan secara gratis dan broker privat yang berbayar namun menawarkan keamanan dalam transfer datanya.

- **Topic**, sebuah nama/penanda yang diberikan pada data yang dikirim oleh publisher, yang nanti digunakan subscriber untuk menerima data tersebut.

24. Daftar Broker Public yang dapat digunakan secara gratis

Name	Broker Address	TCP Port
 Eclipse	mqtt.eclipse.org	1883
 Mosquitto	test.mosquitto.org	1883
 HiveMQ	broker.hivemq.com	1883
 Flespi	mqtt.flespi.io	1883
 Dioty	mqtt.dioty.co	1883
 Flux	mqtt.fluux.io	1883
 EMQX	broker.emqx.io	1883

Catatan: ketika menggunakan broker public, usahakan menggunakan nama Topic yang unik supaya tidak tercampur dengan topic milik pengguna lain.

25. QoS (Quality of Service)

Merupakan kesepakatan antara pengirim dan penerima pesan yang menentukan tingkat jaminan pengiriman untuk pesan tertentu pada MQTT.

Ada 3 level QoS pada MQTT, antara lain:

- **QoS 0**, pesan terkirim **paling banyak** satu kali.

Pesan dikirim hanya sekali. Pesan yang terkirim tergantung dari reliabiliti stack TCP alias tergantung keberadaan network dan tidak ada usaha untuk mentransmisikan pesan kembali.

- **QoS 1**, pesan terkirim **setidaknya** sekali.

Pesan dikirimkan setidaknya satu kali. Jadi subscriber setidaknya akan menerima pesan sekali. Jika subscriber tidak mengakui (*acknowledge*) maka broker akan mengirimkan pesan sampai publisher menerima status pengakuan pesan dari klien.

- **QoS 2**, pesan terkirim **tepat** satu kali.

Pesan pasti diterima satu kali. Protokol dengan level ini memastikan bahwa pesan pasti tersampaikan dan tidak terjadi duplikasi pesan yang terkirim.

26. Topology HTTP dan MQTT

