

# MQTT dan Node-Red

D3 Teknologi Telekomunikasi  
Fakultas Ilmu Terapan  
Telkom University

# Definisi

- Node-RED adalah alat yang digunakan untuk merangkai device hardware, API dan layanan online dengan cara yang baru dan menarik.

# Fitur-fitur Node-RED

## 1. Browser-based flow editing

- Node-RED menyediakan flow editor berbasis browser yang dapat digunakan untuk merangkai device hardware, API dan layanan online menggunakan representasi nodes dalam satu palette. Flow dapat dibangun ke dalam runtime hanya dengan satu klik saja.

## 2. Built on Node.js

- Light-weight runtime merupakan bawaan Node.js yang memilii banyak keuntungan dengan event-driven dan model non-blocking. Hal ini sangat ideal ketika dijalankan pada sisi jaringan hardware dengan harga murah seperti Raspberry Pi sebgus teknologi Cloud.

## 3. Social Development

- Flow yang dibangun dengan Node-RED akan disimpan dalam berkas JSON sehingga dapat diimpor dengan mudah dan diekspor untuk keperluan berbagi dengan Developer lainnya.

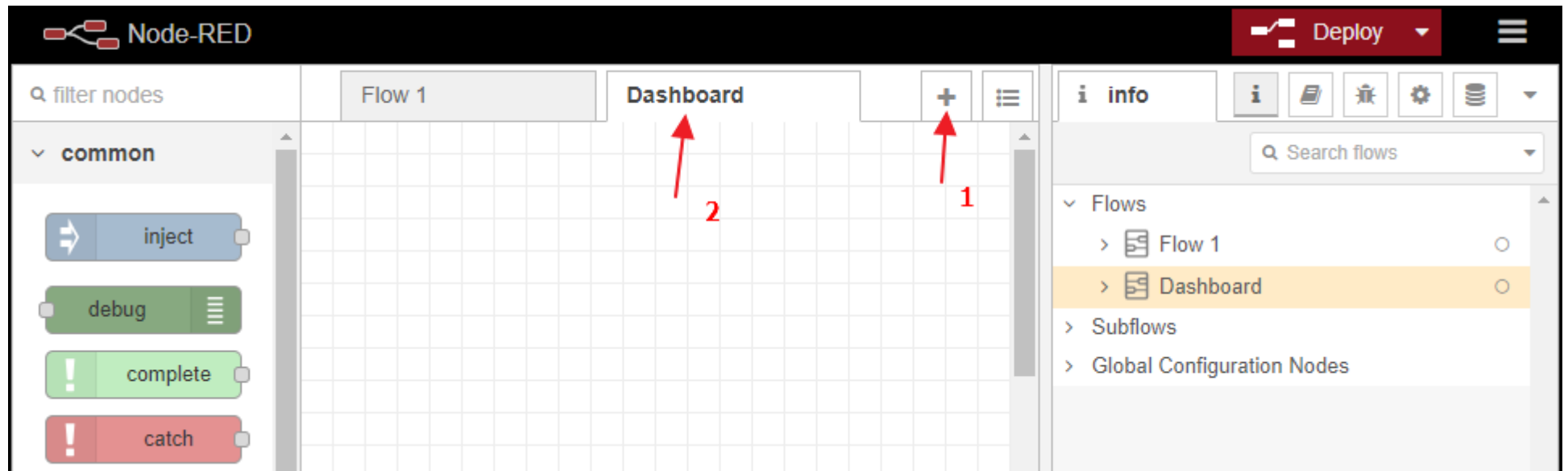
# Platform

- Run Locally
- On a device
- In the cloud

# Langkah Praktikum

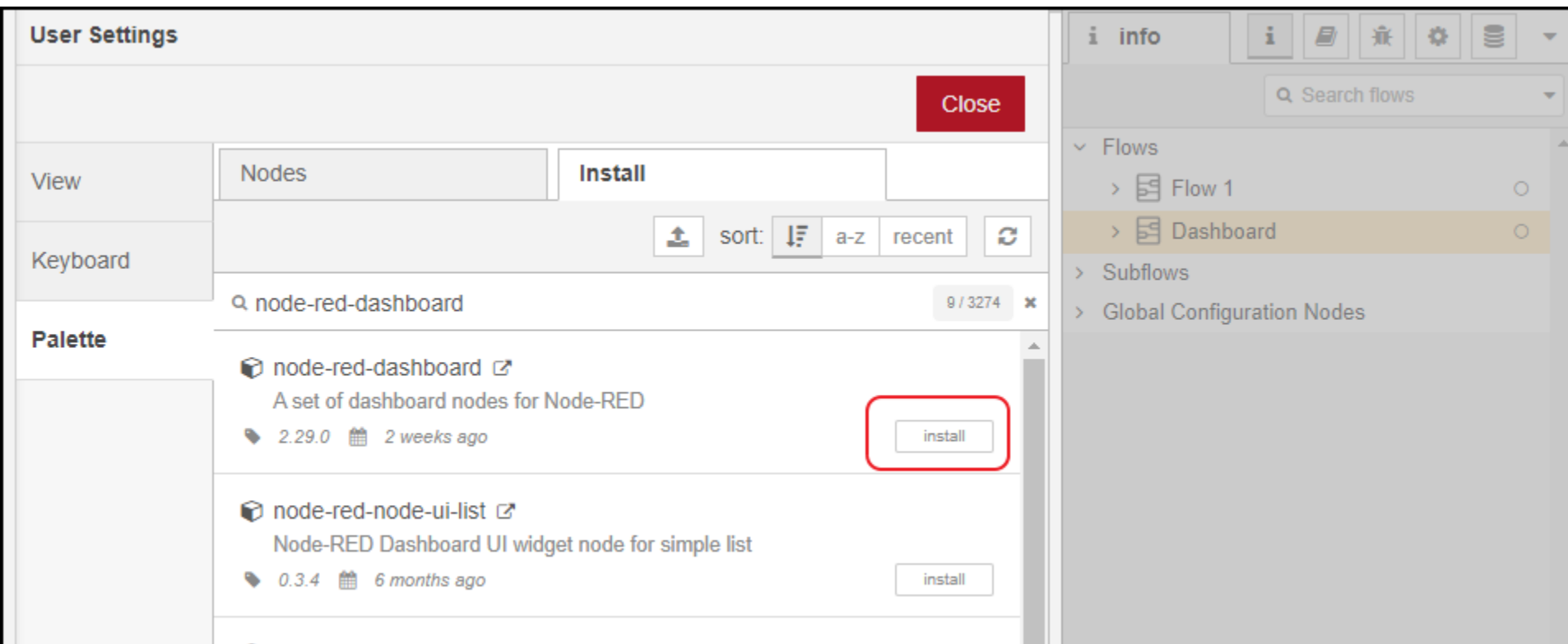
# Flow

- Buat flow baru
- Rename Flow dengan Dashboard



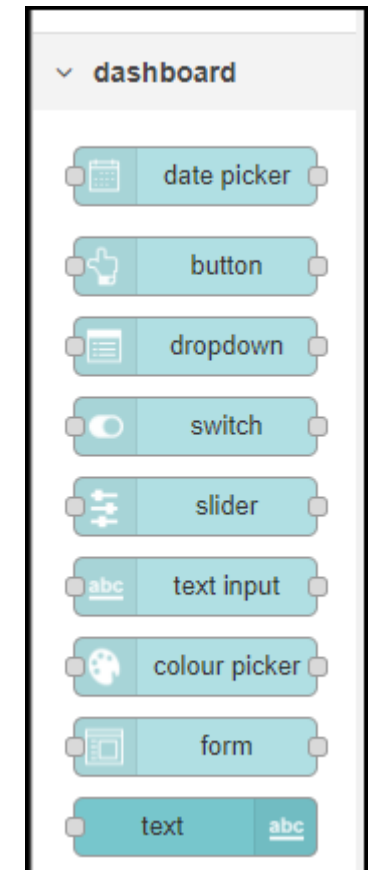
# Install Dashboard

- Pilih Manage Palette
- Ketikkan “node-red-dashboard”
- Click Install



The screenshot shows the Node-RED Manage Palette interface. The search bar contains the text "node-red-dashboard" and shows 9 results. The first result, "node-red-dashboard", is highlighted with a red box around its "install" button. The interface includes a "User Settings" panel, a "View" dropdown set to "Nodes", and an "Install" button. The search results list includes the following items:

Name	Description	Version	Time	Action
node-red-dashboard	A set of dashboard nodes for Node-RED	2.29.0	2 weeks ago	install
node-red-node-ui-list	Node-RED Dashboard UI widget node for simple list	0.3.4	6 months ago	install

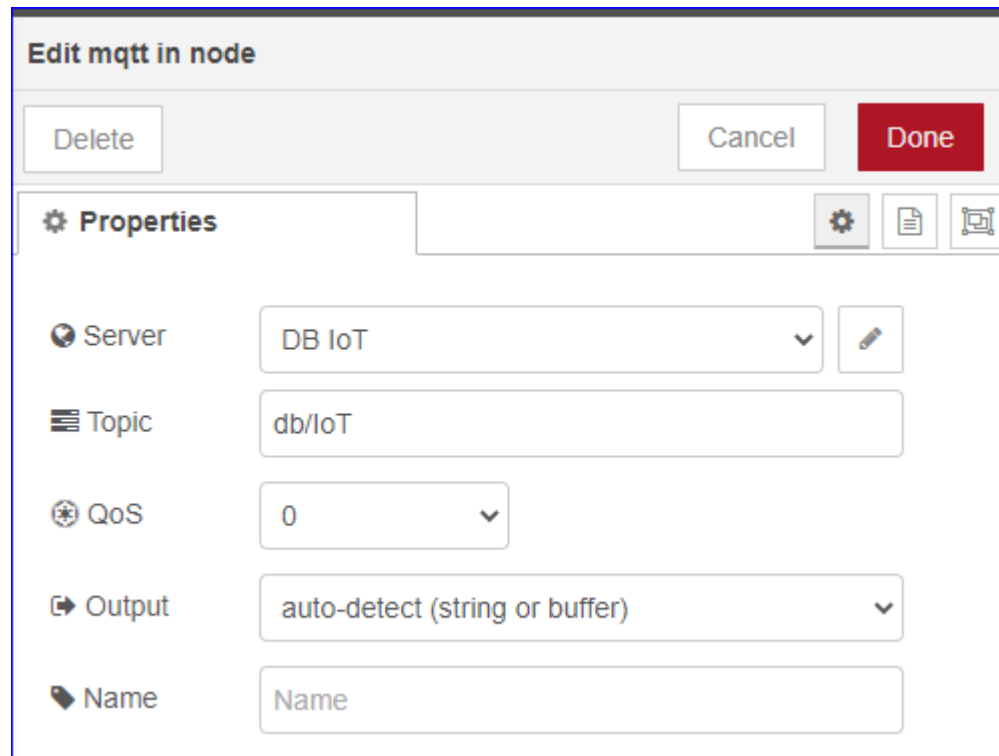


The screenshot shows the Node-RED Dashboard node palette. The palette is titled "dashboard" and contains the following nodes:

- date picker
- button
- dropdown
- switch
- slider
- text input
- colour picker
- form
- text

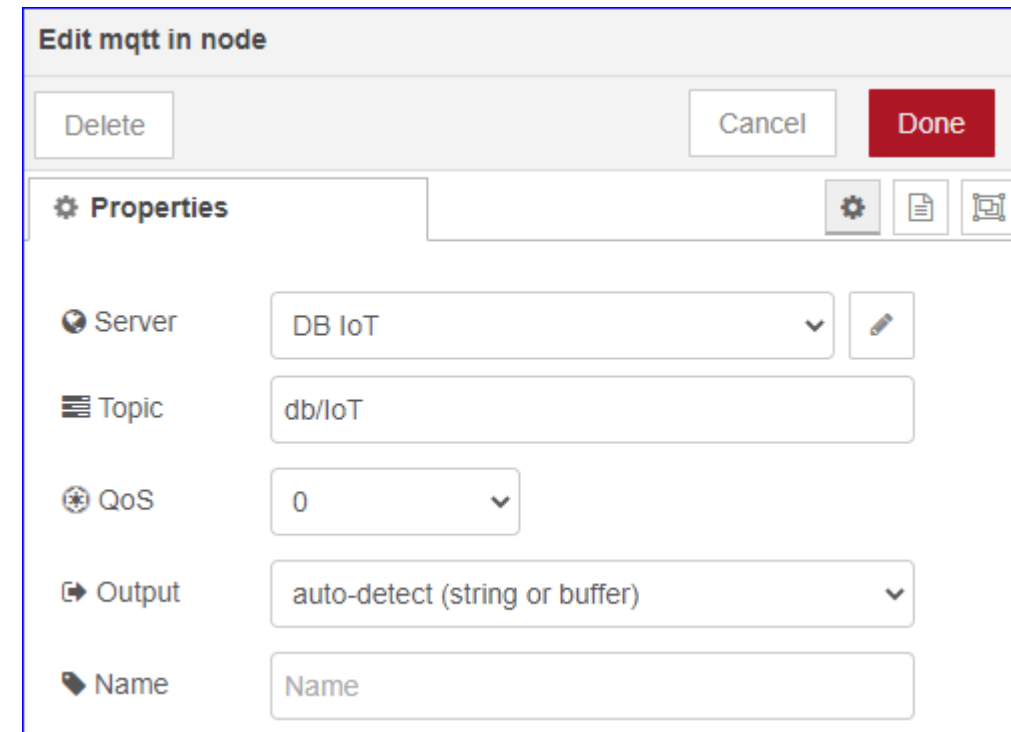
# mqtt in

- Server : DB IoT (bebas)
- Topic : db/IoT (bebas)
- QoS : 0



The screenshot shows a dialog box titled "Edit mqtt in node". At the top, there are three buttons: "Delete", "Cancel", and "Done". Below the buttons is a "Properties" section with a gear icon and three utility icons (gear, document, refresh). The properties are:

Property	Value
Server	DB IoT
Topic	db/IoT
QoS	0
Output	auto-detect (string or buffer)
Name	Name



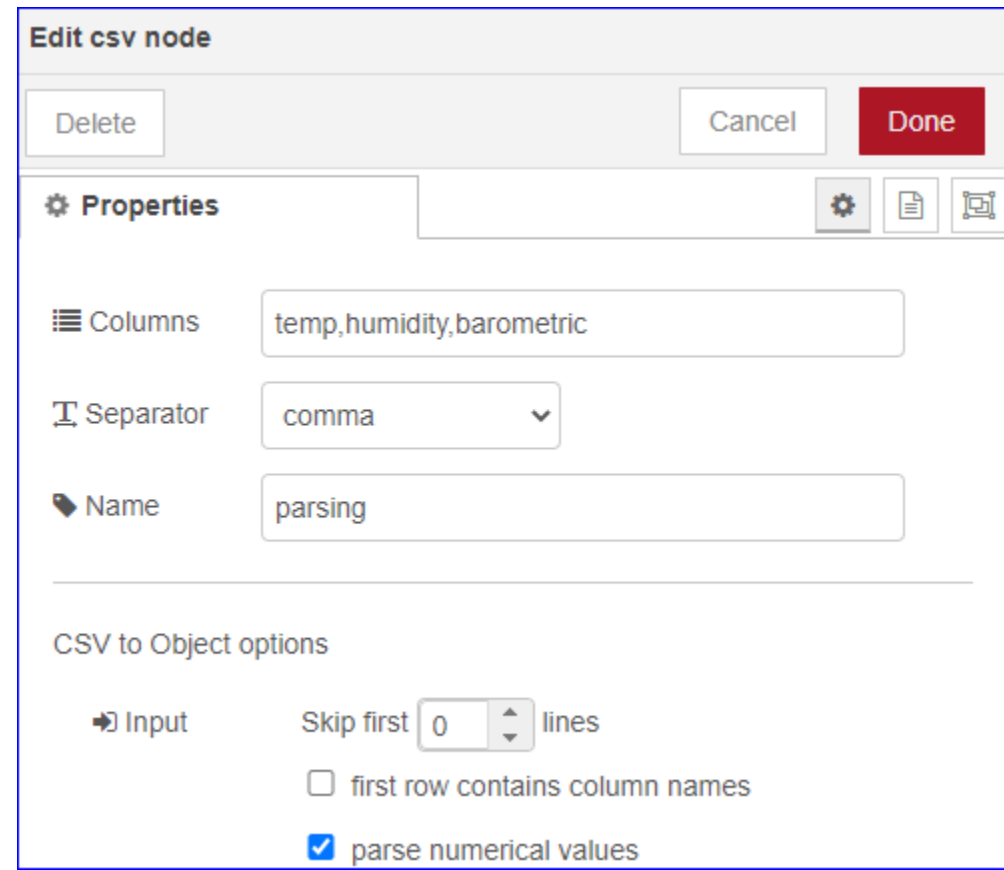
The screenshot shows a dialog box titled "Edit mqtt in node". At the top, there are three buttons: "Delete", "Cancel", and "Done". Below the buttons is a "Properties" section with a gear icon and three utility icons (gear, document, refresh). The properties are:

Property	Value
Server	DB IoT
Topic	db/IoT
QoS	0
Output	auto-detect (string or buffer)
Name	Name



# parser

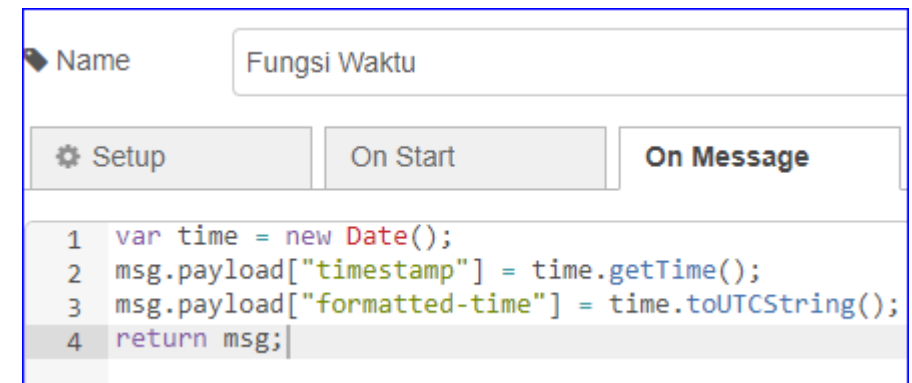
- Untuk membagi data berdasarkan koma
- Pilih : csv
- Edit csv node
  - Columns : temp,humidity,barometric
  - Separator : comma
  - Name : parsing



The screenshot shows the 'Edit csv node' configuration window. It features a top bar with 'Delete', 'Cancel', and 'Done' buttons. Below this is a 'Properties' section with three main fields: 'Columns' (text input with 'temp,humidity,barometric'), 'Separator' (dropdown menu with 'comma' selected), and 'Name' (text input with 'parsing'). A horizontal line separates this from the 'CSV to Object options' section, which includes an 'Input' field, a 'Skip first' spinner set to '0' lines, and two checkboxes: 'first row contains column names' (unchecked) and 'parse numerical values' (checked).

# function

- Name : Fungsi Waktu
- Function :
  - `var time = new Date();`
  - `msg.payload["timestamp"] = time.getTime();`
  - `msg.payload["formatted-time"] = time.toUTCString();`
  - `return msg;`

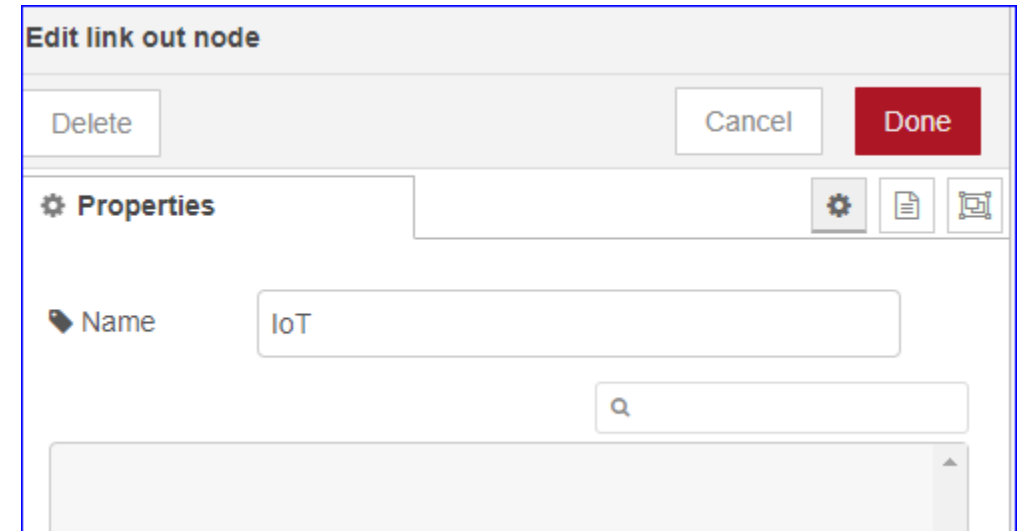
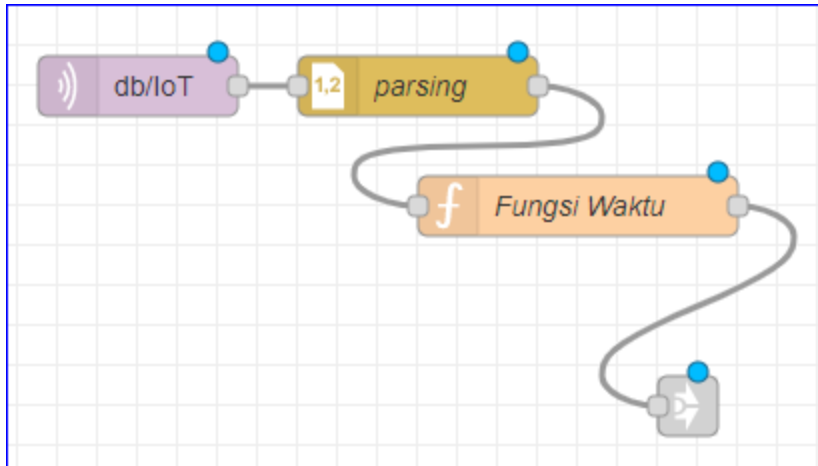


The screenshot shows a Node-RED function node configuration. The 'Name' field is set to 'Fungsi Waktu'. Below the name field are three tabs: 'Setup', 'On Start', and 'On Message'. The 'On Message' tab is selected, and the following JavaScript code is entered in the code editor:

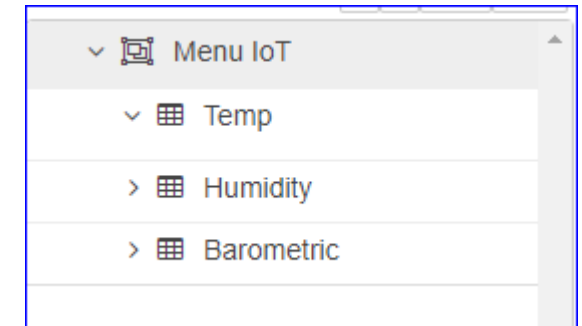
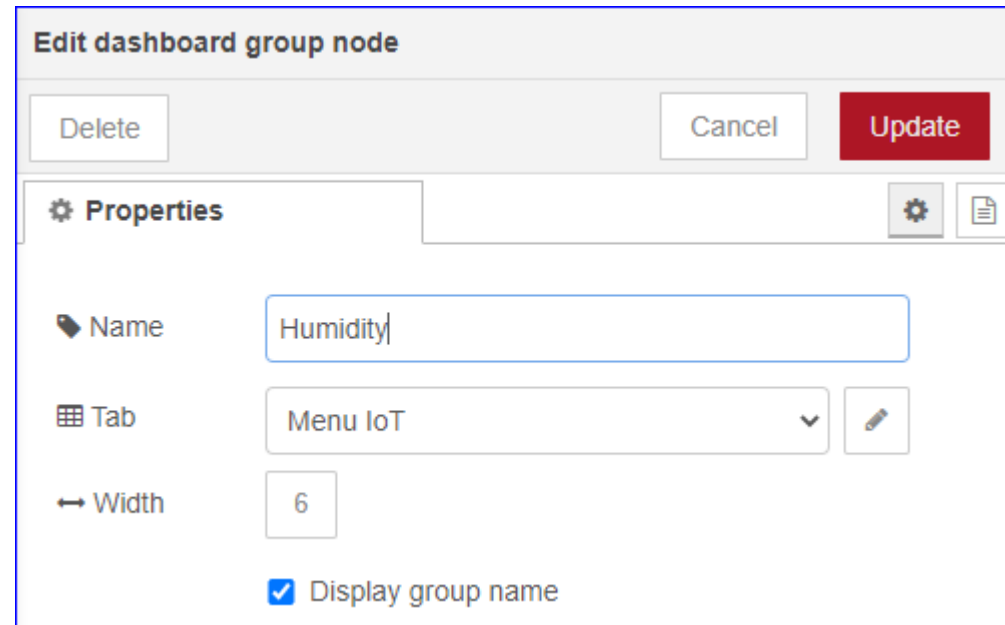
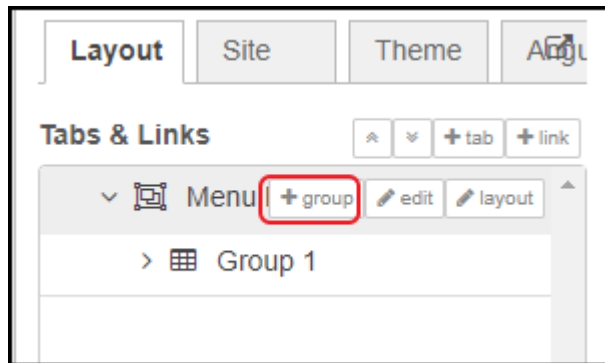
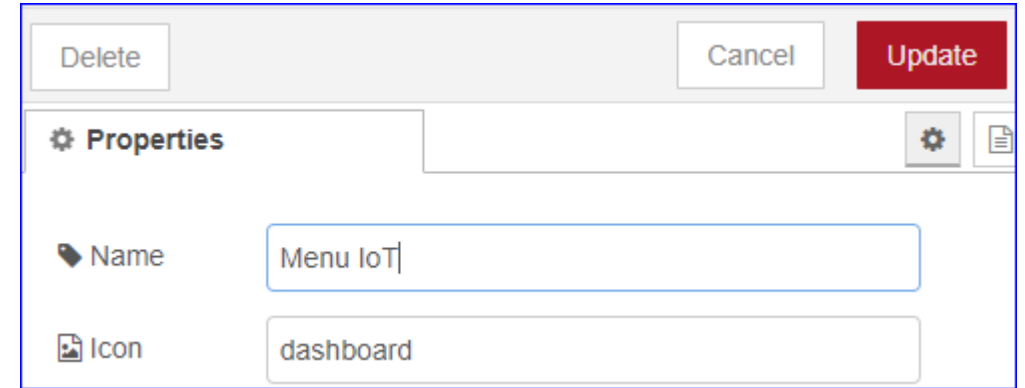
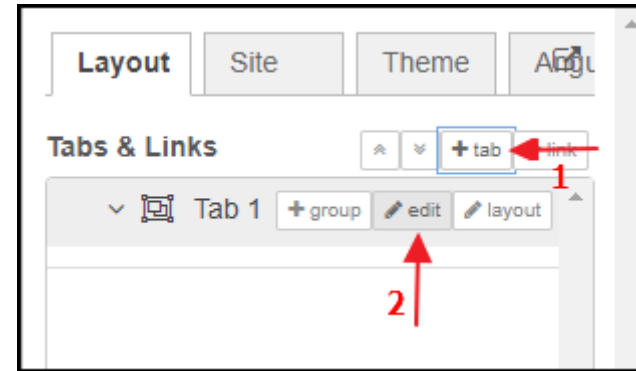
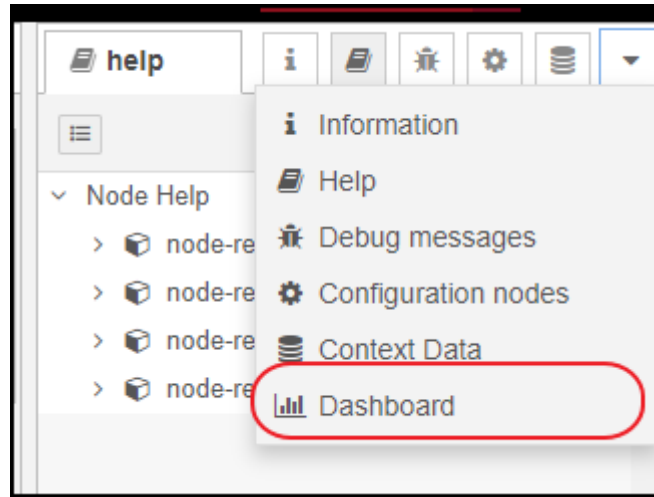
```
1 var time = new Date();
2 msg.payload["timestamp"] = time.getTime();
3 msg.payload["formatted-time"] = time.toUTCString();
4 return msg;
```

# Link out

- Name : IoT

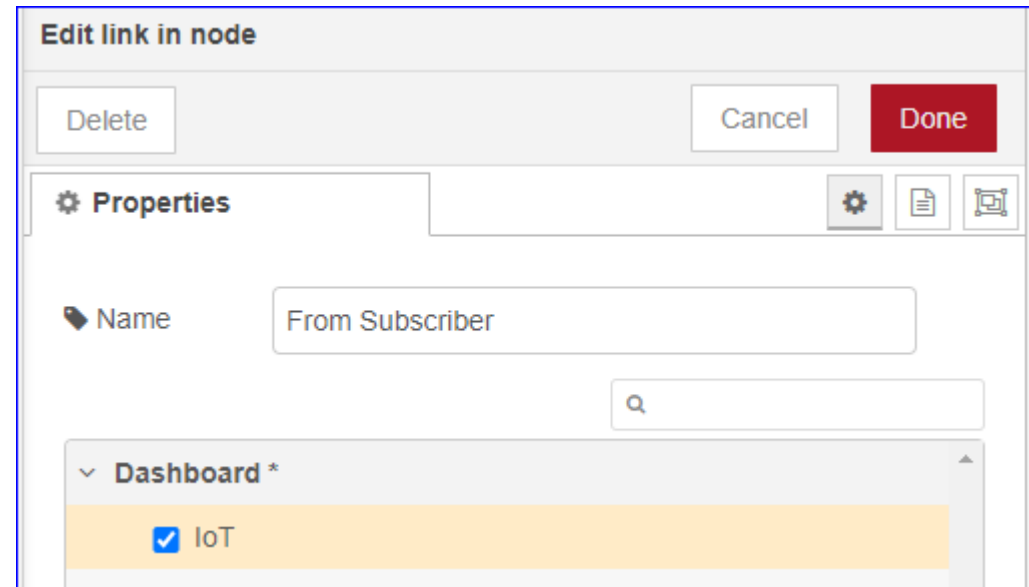


# Menu Dashboard



# Link in

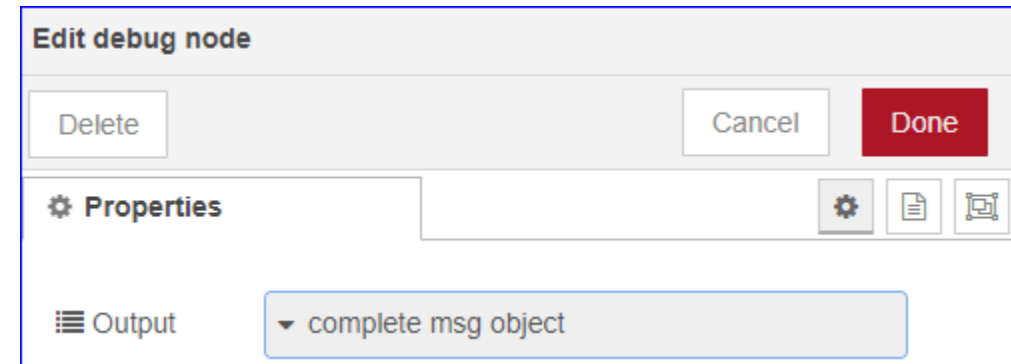
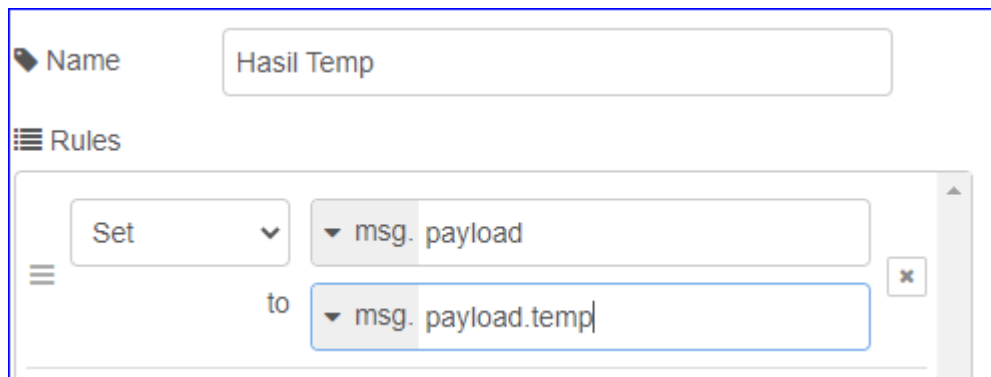
- Name : From Subscriber
- Pilih : IoT



The screenshot shows a dialog box titled "Edit link in node". At the top, there are three buttons: "Delete", "Cancel", and "Done". Below the buttons is a "Properties" section with a gear icon and three smaller icons (gear, document, and camera). The "Name" field is set to "From Subscriber". Below the name field is a search bar with a magnifying glass icon. Underneath the search bar is a dropdown menu labeled "Dashboard \*". The dropdown menu is open, showing a single option "IoT" which is selected with a blue checkmark.

# Change + debug

- Pilih change : 3 buah (to msg.payload.tmp, humidity, baro)
- Pilih debug : 1 buah (complete msg object)



# Chart

**Group** [Menu IoT] Temp    
**Size** auto   
**Label** chart   
**Type**   enlarge points  
**X-axis** last   OR  points  
**X-axis Label**   as UTC  
**Y-axis** min  max   
**Legend**   Interpolate   
**Series Colours**     
    
    
**Blank label**   
**Name**

**Group** [Menu IoT] Humidity    
**Size** auto   
**Type**   
**Label** gauge   
**Value format**   
**Units**   
**Range** min  max   
**Colour gradient**     
**Vectors** 0 ...  ...  ... 100  
**Name**

**Group** [Menu IoT] Barometric    
**Size** auto   
**Label** chart   
**Type**   enlarge points  
**X-axis** last   OR  points  
**X-axis Label**   as UTC  
**Y-axis** min  max   
**Legend**   Interpolate   
**Series Colours**     
    
    
**Blank label**   
**Name**


# Text

Group: [Menu IoT] Temp

Size: auto

Label: Temperature :

Value format: `{{msg.payload}} &deg;C`

Layout: 


Name: Temperature Text

Group: [Menu IoT] Humidity

Size: auto

Label: Humidity :

Value format: `{{msg.payload}} %`

Layout: 


Name: Humidity Text

Group: [Menu IoT] Barometric

Size: auto

Label: Barometric :

Value format: `{{msg.payload}} psi`

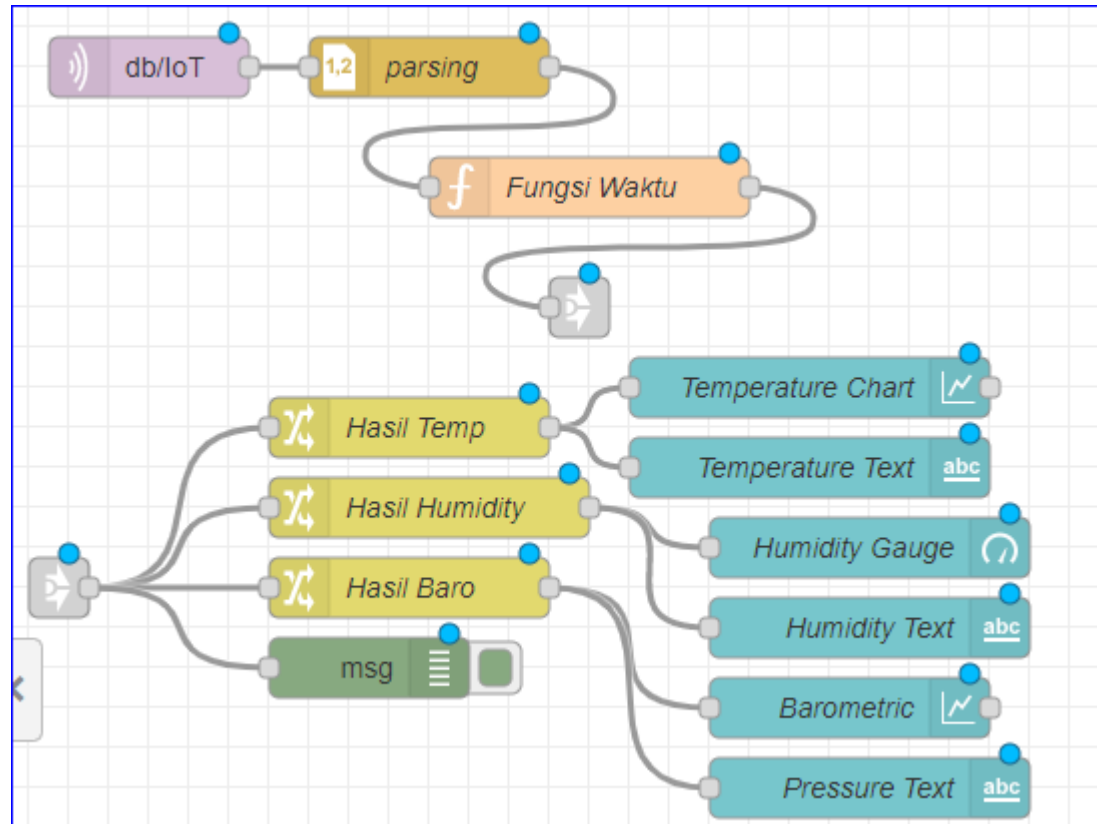
Layout: 

Name: Pressure Text



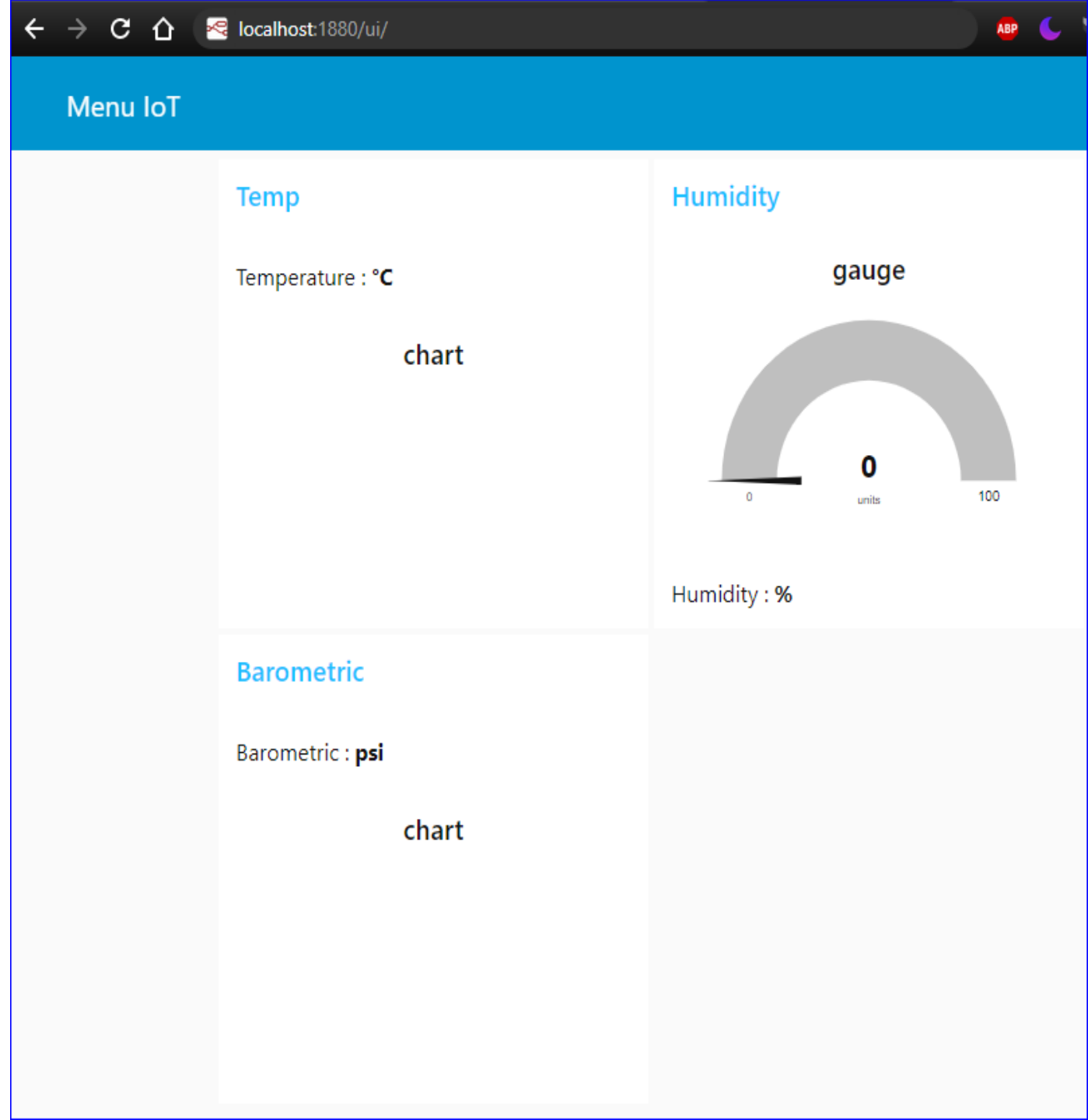
# Deploy

- Jalankan mqtt broker



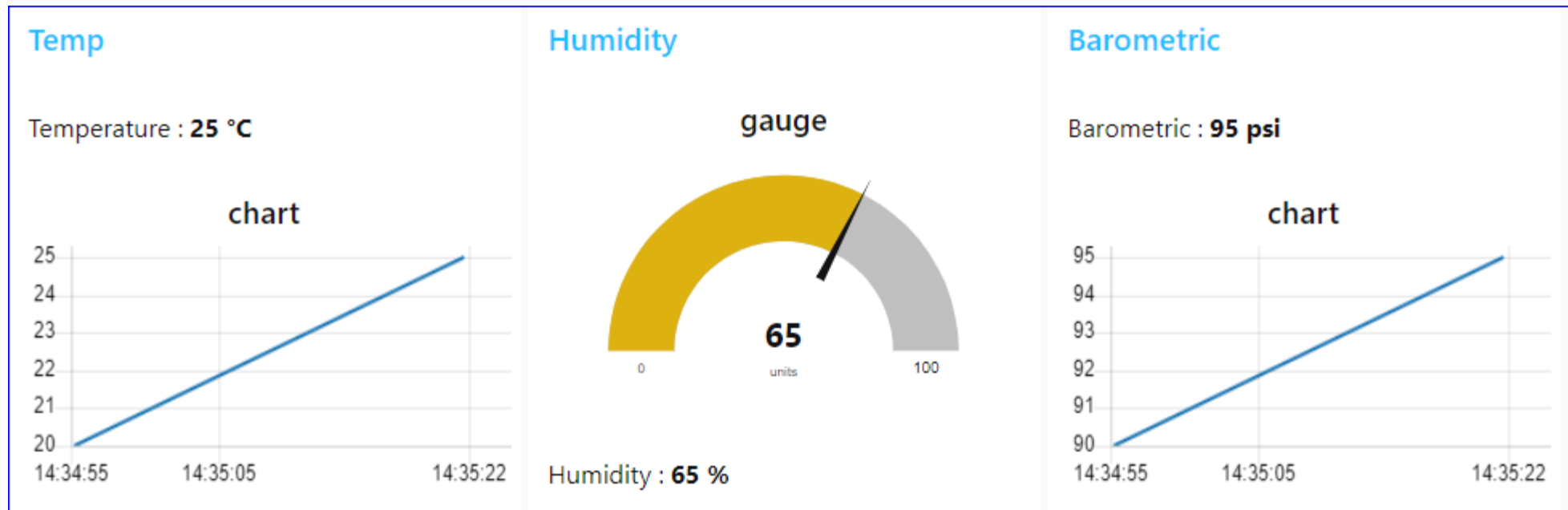
# Tampilan Dashboard

- <http://localhost:1880/ui>



# Insert Data

- Buka mqtt publisher
- `mosquitto_pub.exe -h 127.0.0.1 -t db/loT -m 25,65,95`



**Thank You**