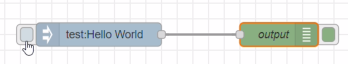
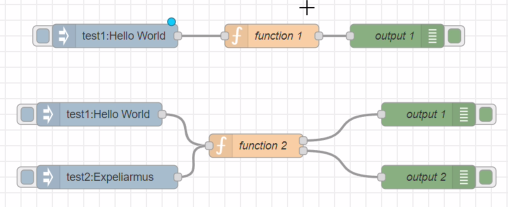
1. Abstrak

Praktikum ini bertujuan untuk memberikan pemahaman dasar pemrograman menggunakan Node-Red serta kemampuan untuk membuat aplikasi server sederhana. Node-RED adalah sebuah alat pemrograman berbasis grafis, memungkinkan pengguna menyatukan perangkat keras, API, dan layanan online dengan cara yang inovatif. Dengan editor berbasis browser, pengguna dapat dengan mudah menyusun alur program menggunakan berbagai node yang tersedia dalam palet, yang dapat dijalankan secara instan. Node-Red juga memiliki kemampuan untuk berfungsi sebagai backend server IoT dengan menggunakan bahasa pemrograman JavaScript. Percobaan ini mencakup beberapa sub-topik, seperti Basic Flow, Function Node, Switch Node, dan JSON Parsing, yang bertujuan untuk memperdalam pemahaman praktis terkait penggunaan Node-Red. Selanjutnya, pada praktikum melibatkan tugas untuk membuat arsitektur sistem IoT yang menggunakan semua fungsi Hands-On, dengan format data multi-input termasuk String, Boolean, Number, dan JSON. Setiap input hasilnya ditampilkan dalam Debug Node yang berbeda, memperlihatkan aplikasi praktis dari pemahaman dasar yang diperoleh melalui praktikum ini. Diharapkan bahwa hasil praktikum ini akan memberikan pemahaman mengenai pemrograman berbasis Node-Red dan penggunaan berbagai node dalam konteks pemrograman aplikasi server sederhana.

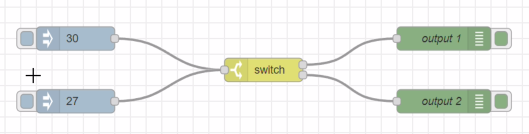
1. Rangkaian
2. Sub A Basic Flow



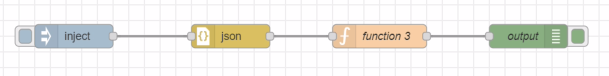
1. Sub B Function Node



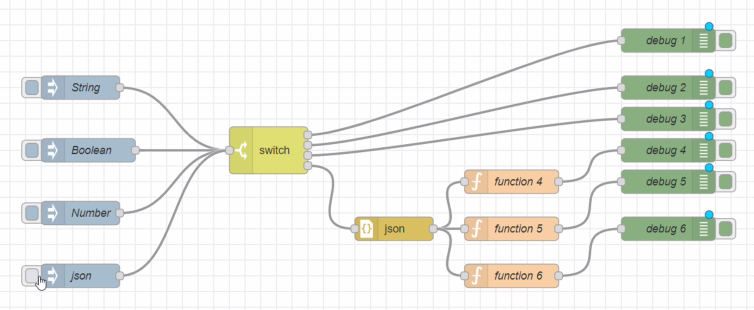
1. Sub C Switch Node



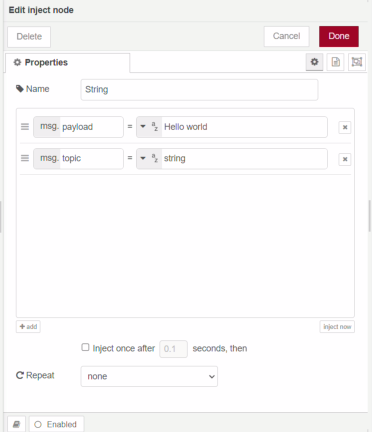
1. Sub D JSON Parsing

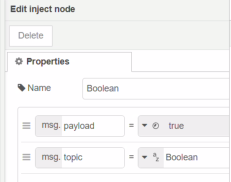


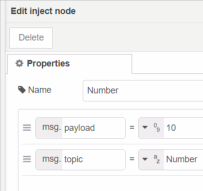
1. Tugas

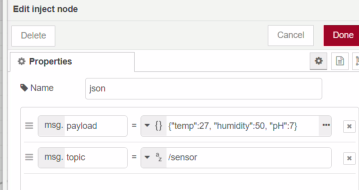


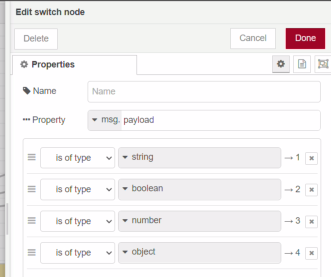
1. Source Code/flow Chard
2. Sub A Basic Flow
3. Sub B Function Node
4. Sub C Switch Node
5. Sub D JSON Parsing
6. Tugas

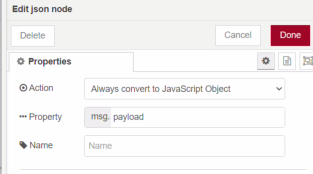


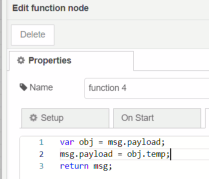
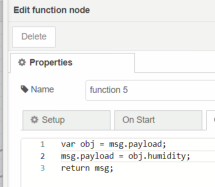
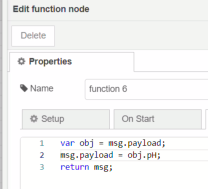


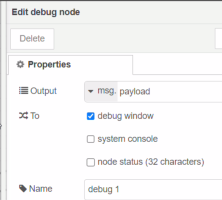










1. Hasil dan Pembahasan
2. Sub A Basic Flow

Penjelasan:

Pada percobaan Basic Flow ini ada beberapa hal penting:

* 1. Persiapan Perangkat dan Node-RED:

Sebelum melakukan percobaan, pastikan perangkat komputer sudah terpasang Node-RED. Langkah ini menjadi prasyarat utama untuk menjalankan percobaan dengan sukses.

* 1. Pembuatan Basic Flow:

Pembuatan Basic Flow dilakukan dengan mengintegrasikan node input dan output. Proses ini bertujuan untuk memahami dasar dari penyusunan alur program menggunakan Node-RED.

* 1. Konfigurasi Node Input:

Dengan double klik pada node input, dilakukan konfigurasi pada inject/input node. Pesan ("msg.payload=Hello World" dan "msg.topic=test") yang dihasilkan akan menjadi data input untuk proses selanjutnya. Ini menunjukkan fleksibilitas dalam mengatur parameter input pada Node-RED.

* 1. Konfigurasi Node Output/Debug:

Langkah selanjutnya adalah melakukan konfigurasi pada node output/debug dengan mengatur output "msg.payload" ke jendela debug. Hal ini berarti pesan yang dihasilkan dari node input akan ditampilkan pada debug window. Konfigurasi ini penting untuk memonitor dan memahami alur program serta hasilnya.

* 1. Tahap Akhir:

Setelah konfigurasi selesai, langkah terakhir adalah mengeksekusi program dengan menekan tombol "Deploy". Hasil yang diharapkan adalah munculnya pesan "successfully injected: test: Hello World". Keberhasilan ini menunjukkan bahwa flow program berjalan dengan baik, dan data input yang telah dikonfigurasi berhasil diinjeksikan dan ditampilkan dengan benar pada debug window.

Melalui analisis langkah-langkah percobaan Basic Flow, dapat disimpulkan bahwa Node-RED memberikan pengguna kemudahan dalam mengembangkan flow program secara visual dan efisien. Pengaturan input dan output pada node-node tersebut memberikan kontrol yang baik terhadap alur program yang dibuat.

1. Sub B Function Node

Penjelasan:

Pada praktikum ini melibatkan fungsi dengan terdiri dari 2 flow, yakni flow tunggal dan berganda. Pada langkah 1 melibatkan pembuatan flow fungsi output tunggal, yang mencakup node input, fungsi, dan output tunggal. Node input digunakan untuk menyuntikkan pesan ke dalam flow, fungsi node berfungsi untuk memproses pesan tersebut, dan output node digunakan untuk menampilkan hasilnya. Pada langkah 2 melibatkan konfigurasi pada node Input1, di mana payload diatur sebagai "Hello World" dan topik sebagai "test1". Pada langkah 3 adalah deploy program dan dokumentasikan hasilnya. Hasilnya akan menunjukkan bahwa pesan "Hello World" dengan topik "test1" akan diteruskan oleh fungsi node dan ditampilkan pada output tunggal. Pada langkah 4 melibatkan pembuatan flow fungsi output berganda, yang kali ini mencakup node input, fungsi, dan dua output nodes. Pada langkah 5 adalah konfigurasi pada node Input2, dengan payload "Expeliarmus" dan topik "test2".Pada langkah 6 melibatkan konfigurasi pada node fungsi, di mana pesan akan dipisahkan berdasarkan topik. Jika topik adalah "test1" atau "test2", pesan akan diteruskan ke output masing-masing; jika tidak, pesan akan diabaikan. Dan pada langkah 7 adalah melakukan deploy program, dan hasilnya adalah pemisahan pesan "Hello World" dan "Expeliarmus" sesuai dengan topiknya. Pesan dengan topik "test1" akan muncul pada output 1, sementara pesan dengan topik "test2" akan muncul pada output 2.

Secara keseluruhan, percobaan ini menunjukkan kemampuan Node-RED dalam mengolah pesan dan memisahkannya berdasarkan kondisi tertentu menggunakan fungsi node. Ini memberikan fleksibilitas dalam pengaturan alur program untuk kebutuhan khusus dan meningkatkan fungsionalitas sistem secara modular.

1. Sub C Switch Node

Penjelasan:

Pada praktikum ini melibatkan Switch Node. Ada beberapa langkh dalam hal ini. Langkah 1 melibatkan pembuatan flow dengan dua input (30 dan 27) yang terhubung ke switch node, dan switch node terhubung ke dua output. Input nodes berisi angka 30 dan 27, sementara switch node memproses pesan sesuai dengan kondisi yang diatur, mengarahkan pesan ke salah satu output berdasarkan nilai payload. Pada langkah 2 melibatkan konfigurasi pada Inject Nodes. Inject Node 1 diatur dengan payload 28, sedangkan Inject Node 2 diatur dengan payload 27. Payload ini nantinya akan diinjeksikan ke dalam flow untuk diproses oleh switch node. Langkah 3 melibatkan konfigurasi pada switch node. Property switch diatur untuk memeriksa nilai payload (msg.payload) dan membaginya ke dua output berdasarkan kondisi: yang pertama jika payload > 28, yang kedua jika payload <= 28. "Stopping after first match" diaktifkan, sehingga setelah kondisi pertama terpenuhi, switch node tidak akan memproses kondisi lainnya. Langkah 4 adalah mendeploy flow. Hasilnya adalah ketika input 30 diinjeksikan, output pada debug node akan menampilkan nilai 30. Begitu pula ketika input 27 diinjeksikan, output pada debug node akan menampilkan nilai 27. Hal ini menunjukkan bahwa switch node secara efektif memisahkan dan mengarahkan pesan sesuai dengan kondisi yang telah diatur.

Denagn demikian secara keseluruhan, percobaan ini mengilustrasikan penggunaan switch node untuk mengarahkan alur program berdasarkan nilai payload, memberikan fleksibilitas dalam pengolahan data dan pengambilan keputusan dalam alur program Node-RED.

1. Sub D JSON Parsing

Penjelasan:

Pada praktikum ini melibatkan JSON Parsing. Ada beberapa langkh dalam hal ini. Langkah 1 melibatkan pembuatan flow dengan input node (name=V1) yang terhubung ke node JSON, node JSON terhubung ke node fungsi, dan node fungsi terhubung ke output node (msg.payload). Ini menciptakan alur data yang melibatkan parsing JSON. Langkah 2 melibatkan konfigurasi pada Inject Node, JSON Parser Node, dan Function Node. Pada Inject Node, pesan dikonfigurasi dengan payload {temp:27, humidity:50} dan topik "sensor". Pada JSON Parser Node, pengaturan dilakukan dengan mengubah pesan menjadi objek JavaScript dan mengatur propertinya pada msg.payload. Pada Function Node, dilakukan pengaturan variabel obj untuk menyimpan nilai payload, dan kemudian nilai msg.payload diubah menjadi obj.temperature, lalu pesan dikembalikan. Dan pada langkah 3 adalah mendeploy flow. Hasilnya adalah ketika input diinjeksikan, nilai 27 dari properti "temperature" pada objek JSON akan diambil, diproses oleh fungsi, dan hasilnya akan ditampilkan pada debug node.

Dengan begitu secara keseluruhan, percobaan ini berhasil menunjukkan penggunaan JSON parsing dalam Node-RED. Input JSON berhasil diubah menjadi objek JavaScript dan nilai tertentu diambil dan dimanipulasi menggunakan fungsi node. Hal ini memperlihatkan kemampuan Node-RED untuk mengolah dan memanipulasi data dalam format JSON dengan mudah dan efektif.

1. Tugas

Penjelasan:

Arsitektur sistem IoT yang dibuat menggunakan Node-RED mencakup empat input (string, boolean, number, dan JSON) yang terhubung ke switch. Switch ini kemudian terhubung ke tiga output dan satu node JSON. JSON node tersebut menghubungkan tiga fungsi (fungtion 4, fungtion 5, dan fungtion 6), dan setiap fungsi terhubung ke satu output.

Pada konfigurasi input, setiap input memiliki pengaturan khusus sesuai dengan format data yang diinginkan (string, boolean, number, dan JSON). Setelah melewati switch, data kemudian diarahkan ke fungsi-fungsi yang spesifik sesuai dengan payload yang ada pada masing-masing input.

Fungsi-fungsi (fungtion 4, fungtion 5, dan fungtion 6) dirancang untuk mengambil nilai tertentu dari objek JSON yang diterima. Fungtion 4 mengambil nilai suhu (temp), fungtion 5 mengambil nilai kelembaban (humidity), dan fungtion 6 mengambil nilai pH dari objek JSON.

Hasil dari percobaan ini mengindikasikan bahwa sistem bekerja sesuai yang diinginkan:

* 1. Saat input string di-klik, debug node pertama menampilkan 'Hello world'.
  2. Saat input boolean di-klik, debug node kedua menampilkan 'true'.
  3. Saat input number di-klik, debug node ketiga menampilkan '10'.
  4. Saat input JSON di-klik, debug node keempat menampilkan 'debug 4=27, debug 5=50, dan debug 6=7'.

Dengan demikian, percobaan ini berhasil membangun arsitektur sistem IoT yang memanfaatkan berbagai jenis data dan menjalankan fungsi-fungsi khusus sesuai dengan input yang diterima, menunjukkan kemampuan fleksibel dan modularitas Node-RED dalam mengelola data IoT.