



SOFTWARE DEFINED NETWORKING (SDN)

CONTROLLER NETWORK
MANAGEMENT

I PUTU HARIYADI

putu.hariyadi@universitasbumigora.ac.id

www.iputuhariyadi.net



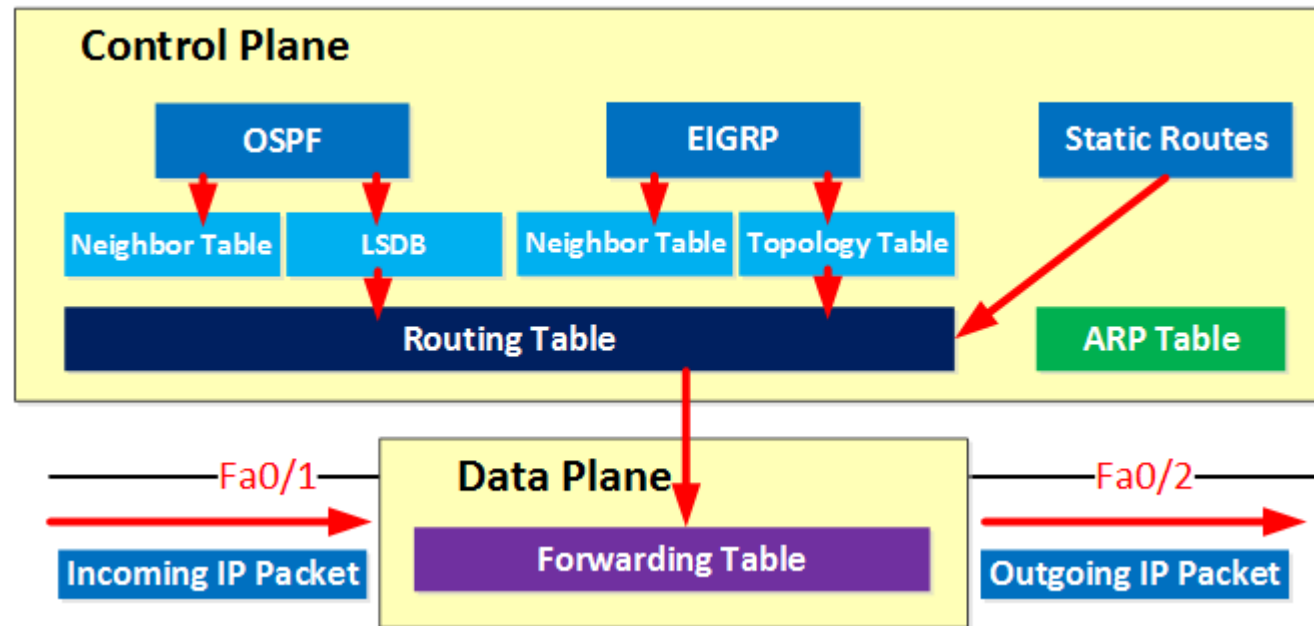
MATERI

- **Pengenalan Software Defined Networking (SDN).**
- **Konfigurasi SDN Controller pada Cisco Packet Tracer.**
- **Topology Discovery menggunakan SDN Controller.**
- **Pengumpulan Informasi menggunakan SDN Controller.**
- **Konfigurasi Jaringan menggunakan SDN Controller.**
- **Implementasi REST API dengan SDN Controller.**



PENGENALAN SOFTWARE DEFINED NETWORKING (SDN)

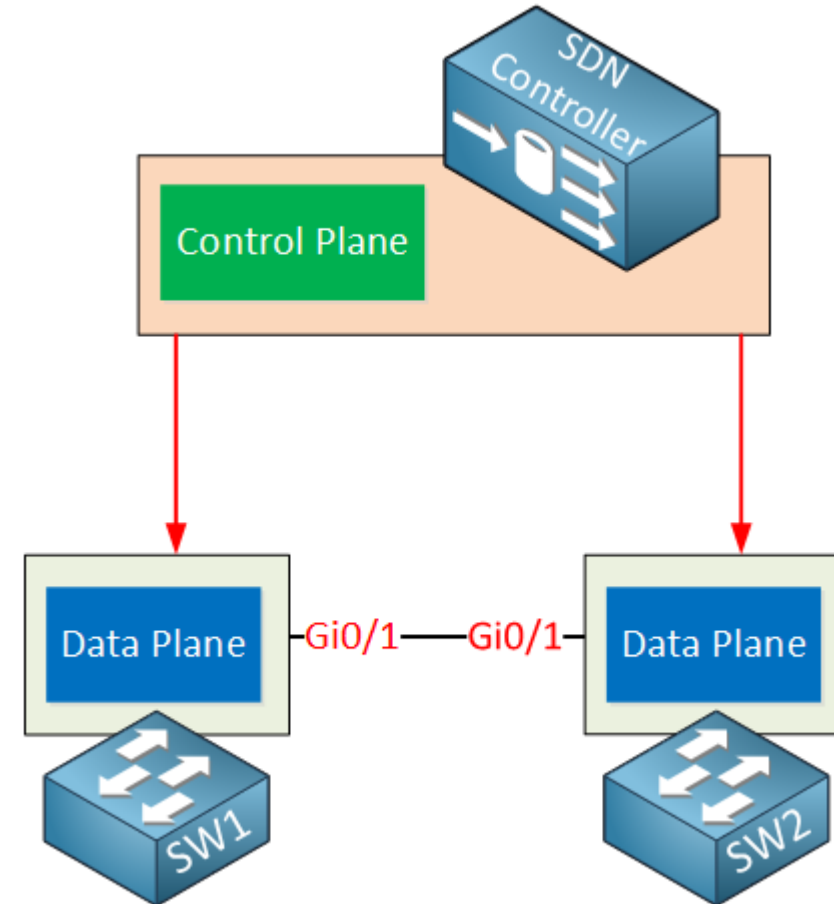
- Merupakan pendekatan inovatif untuk merancang, mengimplementasikan dan mengelola jaringan yang memisahkan pengontrolan jaringan (**control plane**) dan proses **forwarding (data plane)** sehingga menawarkan banyak manfaat dalam hal fleksibilitas jaringan dan kemampuan pengontrolan (Benzekki, El Fergougui, & Elbelrhiti Elalaoui, 2017).



(Sumber gambar: NetworkLessons.com)

PENGENALAN SOFTWARE DEFINED NETWORKING (SDN)

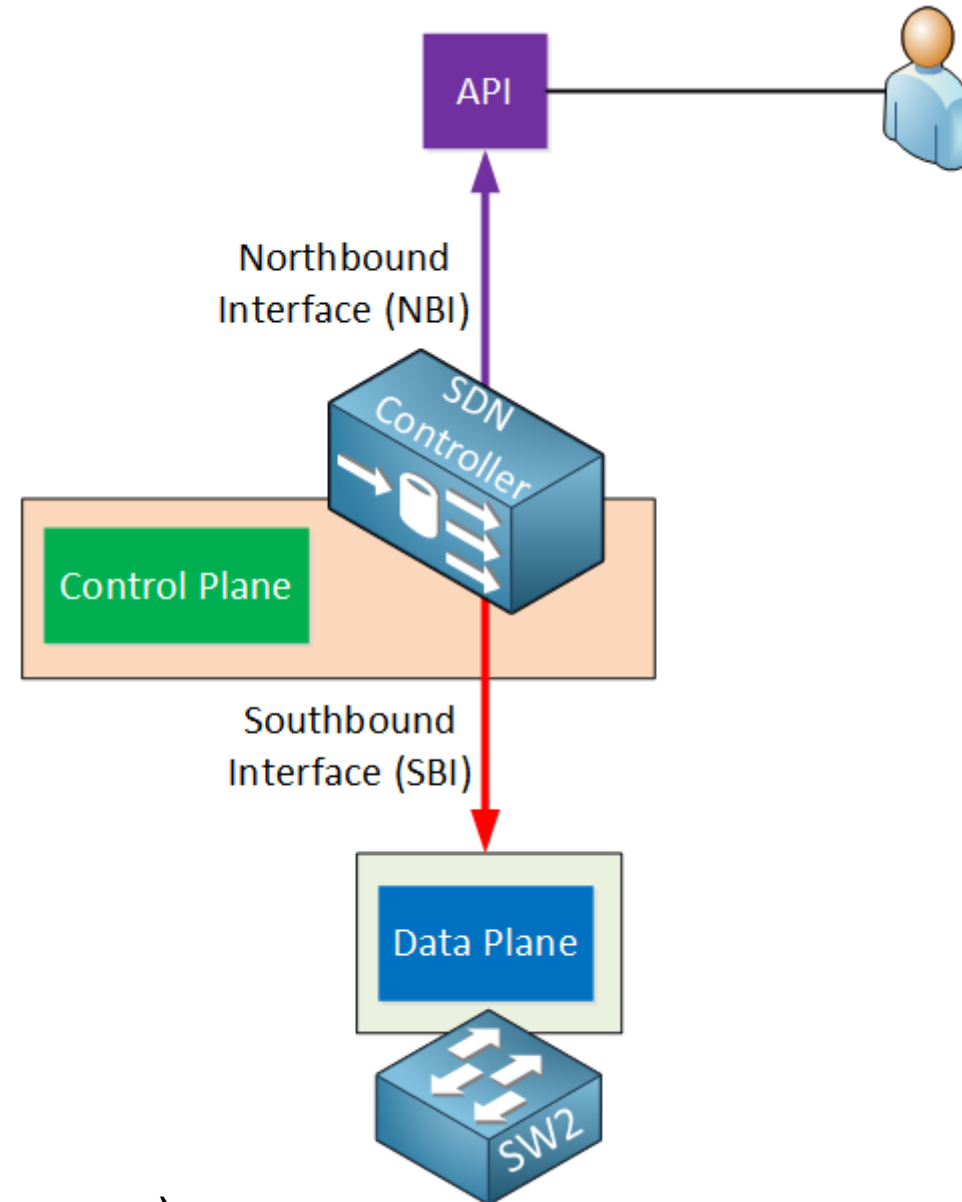
- Perangkat jaringan seperti **router**, **switch** dan **firewall** memiliki fungsi berbeda yang harus dilakukan.
- Fungsi berbeda tersebut dibagi menjadi **plane** berbeda. Terdapat 3 (tiga) **plane** yaitu:
 - a) **Control Plane**, bertanggungjawab dalam mempertukarkan informasi **routing**.
 - b) **Data Plane**, bertanggung jawab dalam meneruskan trafik (**forwarding**).
 - c) **Management Plane**, digunakan untuk mengakses dan mengelola (**management**) perangkat jaringan.
- **SDN menggunakan controller terpusat untuk control plane.**



(Sumber: NetworkLessons.com)

ANTARMUKA (INTERFACE) SDN

- **SDN** menggunakan 2 (dua) antarmuka atau interface khusus yaitu **Northbound interface (NBI)** dan **Southbound interface (SBI)**.
- **SDN controller** berkomunikasi dengan perangkat jaringan untuk memprogram data plane yang dilakukan melalui **Southbound interface**.
- Umumnya menggunakan antarmuka (**interface**) perangkat lunak yang dikenal dengan nama **Application Programming Interface (API)**.



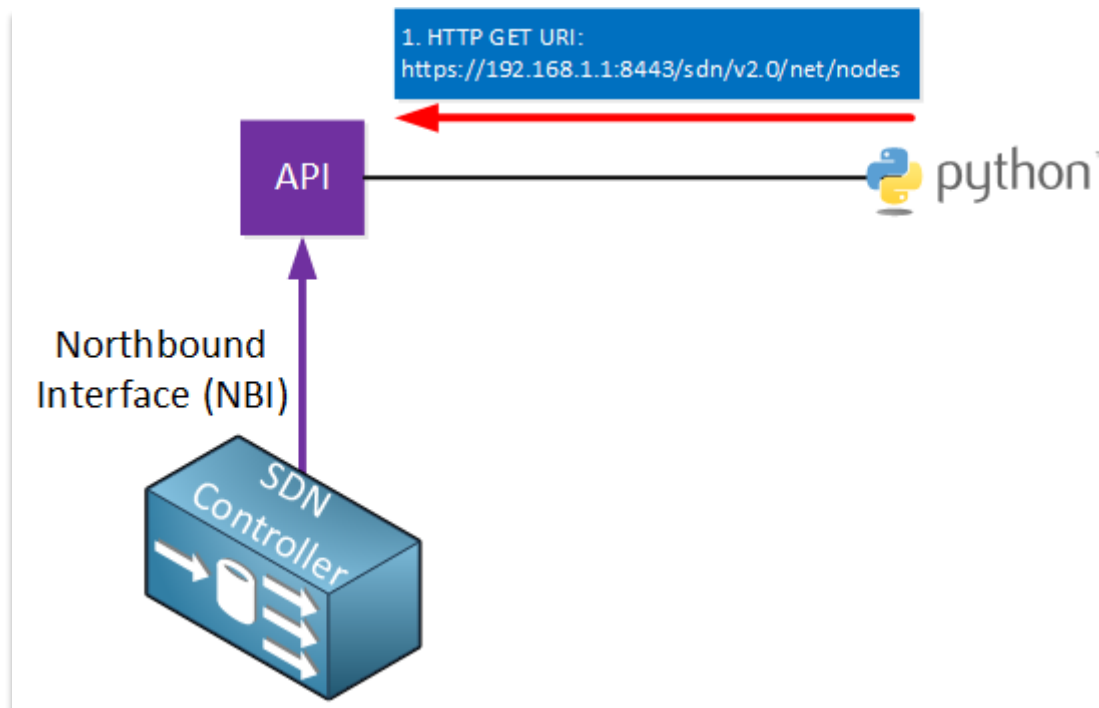
ANTARMUKA (INTERFACE) SDN

- **API** merupakan antarmuka perangkat lunak yang memungkinkan sebuah aplikasi memberikan akses ke aplikasi lainnya dengan fungsi dan struktur data yang telah didefinisikan.
- **Northbound interface** digunakan untuk mengakses **SDN controller** itu sendiri sehingga memungkinkan network administrator melakukan konfigurasi dan mengambil informasi dari **controller** tersebut.
- Hal ini dapat dilakukan melalui **Graphical User Interface (GUI)** atau melalui **API** sehingga memungkinkan aplikasi lainnya mengakses **SDN controller** atau melalui skrip yang ditulis untuk mengotomatisasi administrasi jaringan.

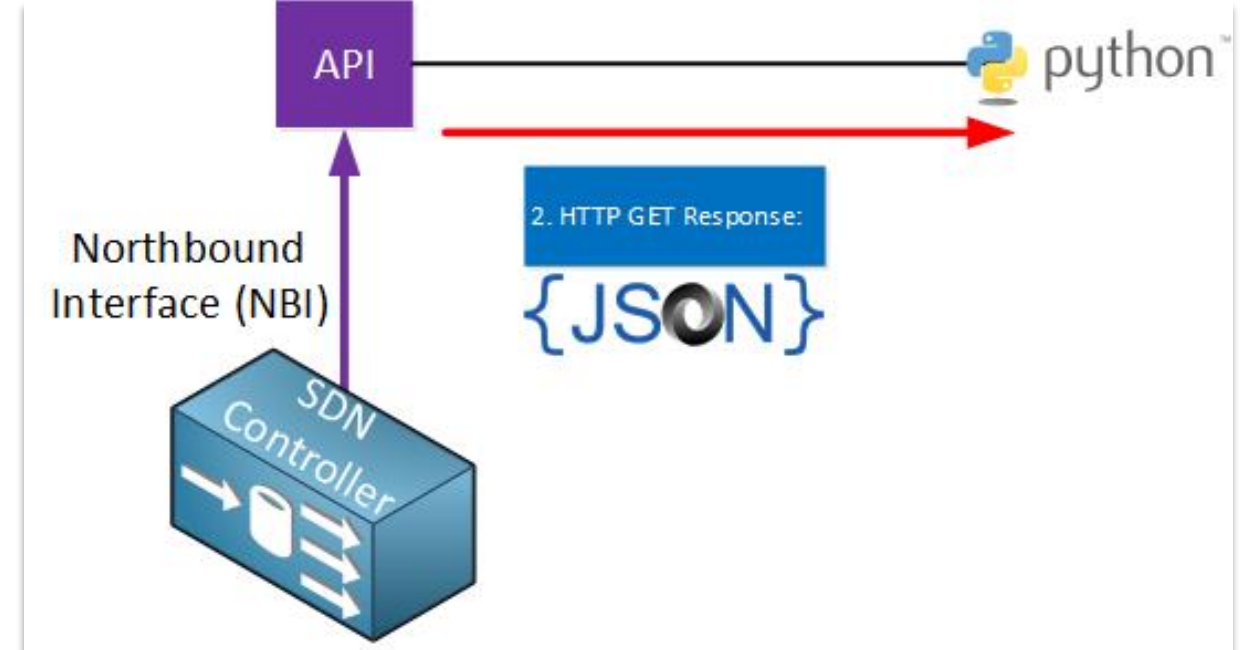
(Sumber: [NetworkLessons.com](https://www.networklessons.com))

SDN REST API

- **SDN controllers** umumnya menggunakan **Representational State Transfer (REST) API**.



HTTP GET request

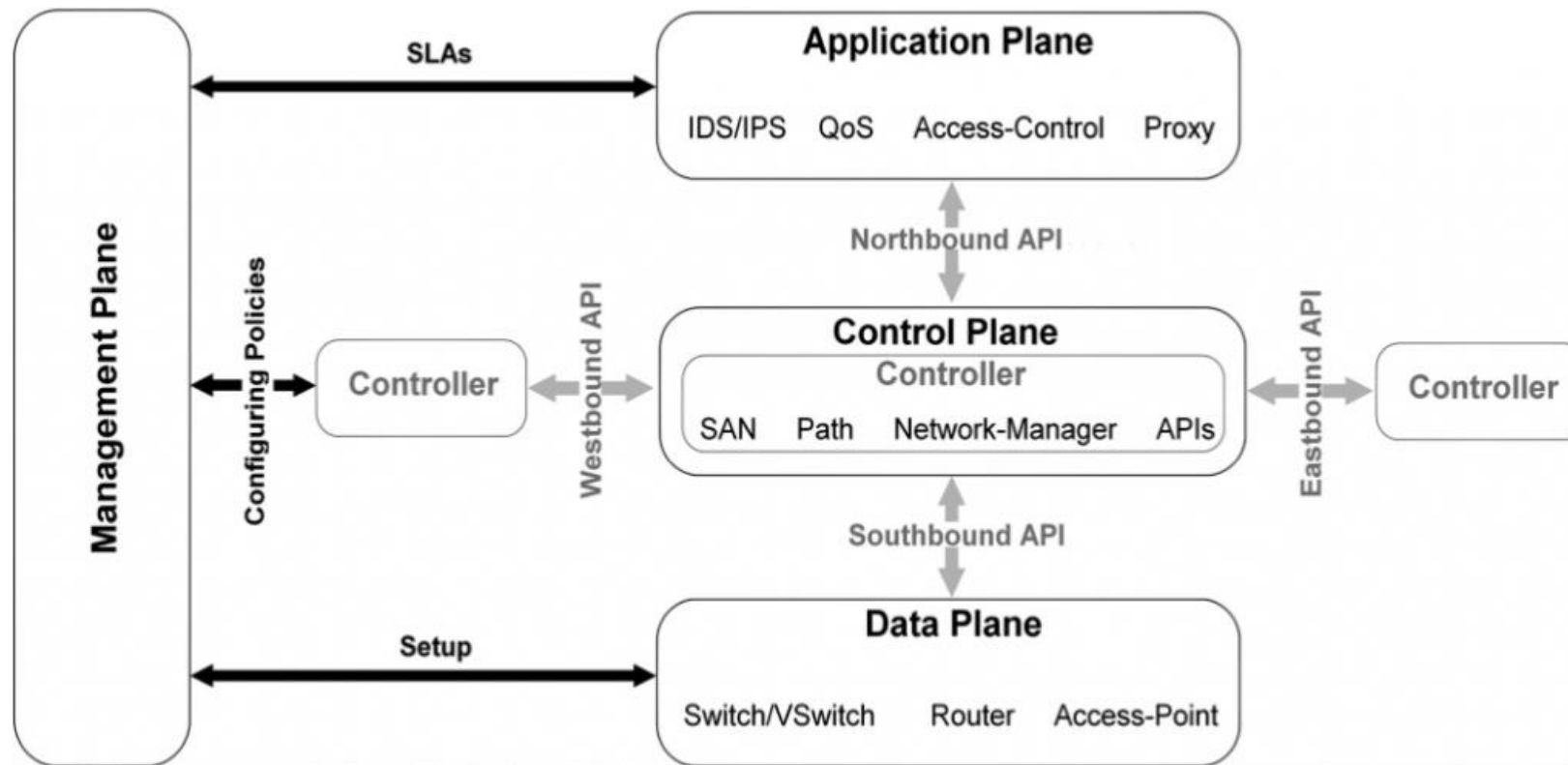


HTTP GET response

SDN REST API

- **REST API** menggunakan pesan (**messages**) **Hypertext Transfer Protocol (HTTP)** untuk mengirim dan menerima informasi antara **SDN controller** dan aplikasi lainnya.
 - a) **HTTP GET**: digunakan untuk mengambil informasi.
 - b) **HTTP POST/PUT**: digunakan untuk mengirim atau memperbaharui informasi.
- Ketika **SDN controller** menerima **HTTP GET request** maka ia akan membalas (**reply**) dengan **HTTP GET response** yang memuat informasi yang diminta.
- Informasi ini umumnya menggunakan format data seperti **JavaScript Object Notation (JSON)** dan **eXtensible Markup Language (XML)**.

ARSITEKTUR INTI SDN



(Sumber gambar: Benzekki et al., 2017)

ARSITEKTUR INTI SDN

SDN memiliki arsitektur inti yang dibagi menjadi 3 (tiga) lapisan:

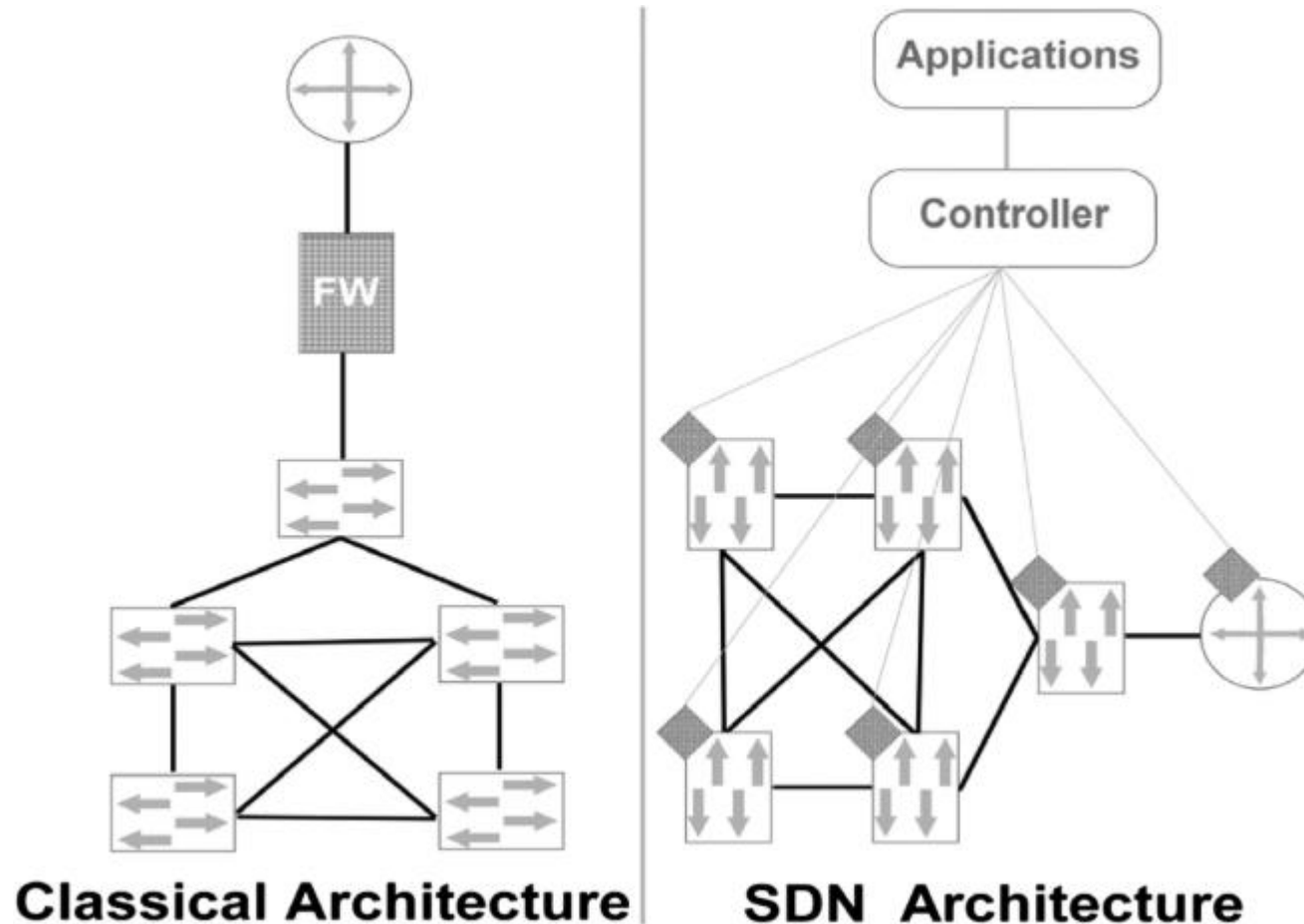
- a) Lapisan atas (**upper layer**) dari arsitektur SDN adalah **lapisan aplikasi** yang menetapkan aturan dan menawarkan berbagai layanan seperti **firewall, pengontrolan akses, Intrusion Detection System (IDS)/Intrusion Prevention System (IPS), kualitas layanan (Quality of Service/QoS), perutean dan layanan proxy**. Lapisan ini bertanggung jawab untuk mengabstraksi manajemen kontrol jaringan SDN melalui API yang diarahkan ke **Northbound** (Benzekki et al., 2017).
- b) Lapisan kedua yaitu **control plane** merupakan abstraksi dari topologi jaringan. **Controller** menjadi komponen utama yang bertanggung jawab untuk membuat tabel alur dan kebijakan penanganan data. Selain itu mengabstraksikan kompleksitas jaringan dan mengumpulkan informasi jaringan melalui **southbound API** serta mempertahankan pandangan holistik jaringan yang terkini. Komunikasi **southbound API** dapat dilakukan menggunakan dua skenario berbeda yaitu:
 - **in-band**, dimana trafik antara **controller** dan perangkat jaringan apa pun harus mematuhi aturan aliran yang ditentukan.
 - **out-of-band**, dimana trafik tidak mengikuti aturan aliran sehingga membutuhkan implementasi VLAN untuk mengisolasi aliran lalu lintas dari komunikasi.

ARSITEKTUR INTI SDN



- Terdapat pula **eastbound/westbound API** yang memungkinkan banyak **controller** untuk bertukar informasi mengenai aliran **data plane**.
- c) Lapisan terendah yaitu **data plane**, menyediakan perangkat jaringan seperti **switch fisik atau virtual, router, dan access point**, serta bertanggung jawab terkait semua aktivitas data termasuk penerusan (**forwarding**), fragmentasi, dan perakitan kembali (**reassembly**) (Benzekki et al., 2017).

PERBANDINGAN ARSITEKTUR SDN DENGAN KLASIK (TRADISIONAL)



(Sumber gambar: Benzekki et al., 2017)

PERBANDINGAN ARSITEKTUR SDN DENGAN KLASIK (TRADISIONAL)

No.	Karakteristik	Arsitektur SDN	Arsitektur Klasik
1.	Programmability	✓	
2.	Centralized control	✓	
3.	Error-prone configuration		✓
4.	Complex Network Control		✓
5.	Network Flexibility	✓	
6.	Improved Performance	✓	
7.	Easy Implementation	✓	
8.	Efficient Configuration	✓	
9.	Enhanced Management	✓	

(Sumber: Benzekki et al., 2017)

PENGENALAN NETWORK CONTROLLER

- Menurut [Cisco](#), **Network Controller** merupakan perangkat lunak yang mengatur fungsi jaringan dan menjadi perantara antara bisnis dengan infrastruktur jaringan.
- **Network Controller** mengelola dan mengatur semua aspek jaringan memastikan bahwa itu beroperasi dengan sempurna dan memenuhi tujuan bisnis.
- Jaringan tidak hanya menyediakan konektivitas dasar tetapi juga **keamanan, kolaborasi yang ditingkatkan, kualitas dan kontinuitas layanan**. Oleh karena itu, arsitektur, infrastruktur, dan manajemen jaringan sangat penting dan semakin terkait dengan keberhasilan bisnis.
- Cara kerja dari **Network Controller** adalah:
 - a) Memelihara inventaris dari perangkat dan status dari perangkat tersebut.
 - b) Mengotomatisasi operasi perangkat seperti konfigurasi dan pembaharuan **image**.
 - c) Menganalisa operasi jaringan, mengidentifikasi potensi masalah dan menyarankan perbaikan.
 - d) Menyediakan platform untuk integrasi dengan aplikasi lain seperti pelaporan.

KEUNTUNGAN NETWORK CONTROLLER

Menurut [Cisco](#), pemanfaatan **Network Controller** dapat memberikan keuntungan-keuntungan berikut:

1. **Mengurangi biaya operasional.**

Network Controllers mengotomatiskan operasi yang menghasilkan penskalaan yang lebih baik, mengurangi waktu, menghilangkan kesalahan, dan memangkas biaya.

2. **Meningkatkan ketersediaan.**

Network Controller mengumpulkan dan menganalisis lalu lintas jaringan untuk secara proaktif mendeteksi masalah potensial apa pun sebelum menjadi masalah nyata. Itu dapat melakukan analisis akar penyebab dan mengingatkan operasi TI yang dapat mengambil langkah korektif dan pencegahan untuk memastikan layanan jaringan selalu tersedia.

3. **Meningkatkan agility.**

Network Controller membuat jaringan gesit dan responsif terhadap kebutuhan bisnis sehingga dapat mengimplementasikan perubahan yang diperlukan pada semua perangkat jaringan secara konsisten dan universal tanpa perlu memakan waktu untuk mengkonfigurasi perangkat demi perangkat.

KEUNTUNGAN NETWORK CONTROLLER



4. Meningkatkan keamanan.

Analisis menyeluruh **Network Controller** terhadap lalu lintas jaringan dapat mengungkapkan potensi ancaman keamanan. **Controller** kemudian dapat bertindak untuk mengisolasi ancaman dan mencegah penyebarannya.

CISCO PACKET TRACER 8.x NETWORK CONTROLLER

- Mirip dengan **SDN Controller** yang ada di kenyataan seperti **Cisco Digital Network Architecture (DNA) Center** dan **Application Policy Infrastructure Controller Enterprise Module (APIC-EM)**.
- Digunakan untuk dapat mempraktikkan pendekatan manajemen jaringan terpusat menggunakan model **Network Controller**.
- **Network Controller** dapat dikelola menggunakan **Web Graphical User Interface (GUI)** atau menggunakan **Application Programming Interface (API)** baik yang diakses melalui **browser** di **Packet Tracer** maupun **PC riil**. Selain itu dapat pula diakses melalui **Python** atau **Postman** untuk menjalankan skrip otomatisasi infrastruktur yang dikembangkan sendiri.
- Tersedia **dashboard** terpusat untuk melihat status jaringan, mengidentifikasi & memecahkan masalah dengan cepat, dan mendorong perubahan konfigurasi ke semua perangkat yang dikelola.



NetworkController
PT-Controller

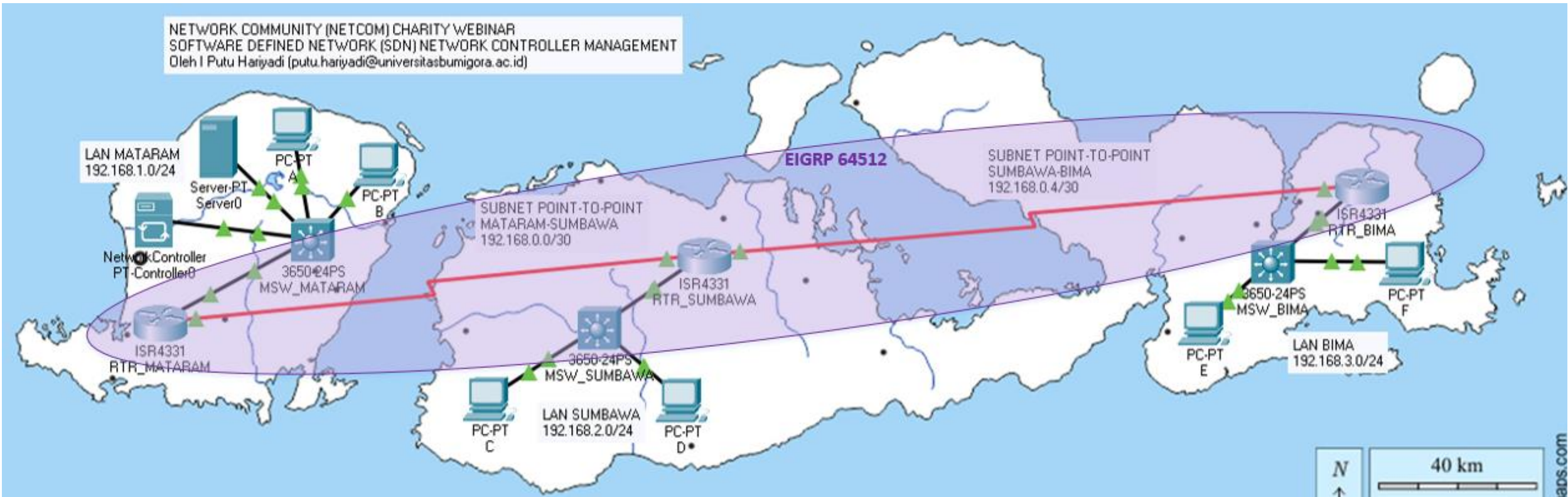
CISCO PACKET TRACER 8.x

- Dapat diunduh melalui situs **Cisco Networking Academy** dengan alamat <https://www.netacad.com/> atau melalui situs [SkillsForAll.com](https://www.skillsforall.com/).
- Kebutuhan sistem untuk menginstalasi dan menjalankan **Cisco Packet Tracer 8.x** adalah:

System Requirements	64 bit	32 bit
Operating System	Microsoft Windows 8.1, 10, 11 (64bit), Ubuntu 20.04 LTS (64bit) or macOS 10.14 or newer.	Microsoft Windows 8.1, 10 (32bit), 11
CPU	amd64(x86-64)	x86
Free RAM	4 GB	2 GB
Free Disk Space	1.4 GB	1.4 GB

RANCANGAN JARINGAN UJICOBA

NETWORK COMMUNITY (NETCOM) CHARITY WEBINAR
SOFTWARE DEFINED NETWORK (SDN) NETWORK CONTROLLER MANAGEMENT
Oleh I Putu Hariyadi (putu.hariyadi@universitasbumigora.ac.id)



RANCANGAN JARINGAN UJICOBA

- Terdapat sebuah perusahaan fiktif dengan nama **PT. Sabar Menanti** yang memiliki wilayah operasional di **Nusa Tenggara Barat (NTB)** dengan kantor pusat di **Mataram** dan kantor cabang di **Sumbawa** serta **Bima**.
- Setiap lokasi memiliki **Lokal Area Network (LAN)** yang terdiri dari 2 (dua) **PC Client** yang terhubung ke jaringan melalui **Multilayer Switch 3650** sebagai **network attachment**. Kecuali di LAN **Mataram** juga terdapat **Server Intranet** perusahaan dan **Network Controller**.
- Komunikasi antara kantor pusat dengan cabang melalui **router ISR4331** dengan **interface Serial Connection**.
- **Routing protocol** menggunakan **Enhanced Interior Gateway Protocol (EIGRP)** dengan **Autonomous System (AS) 64512**.

RANCANGAN PENGALAMATAN IP

PERANGKAT	HOSTNAME	INTERFACE	IP ADDRESS	GATEWAY
Network Controller	PT-Controller0	GigabitEthernet0	192.168.1.254/24	192.168.1.1
Server-PT	Server0	FastEthernet0	192.168.1.253/24	192.168.1.1
Router ISR4331	RTR_MATARAM	GigabitEthernet0/0/0	192.168.1.1/24	
		Serial0/1/0	192.168.0.1/30	
Multilayer Switch 3650	MSW_MATARAM	VLAN 1	192.168.1.2/24	192.168.1.1
Router ISR4331	RTR_SUMBAWA	GigabitEthernet0/0/0	192.168.2.1/24	
		Serial0/1/0	192.168.0.2/30	
		Serial0/1/1	192.168.0.5/30	
Multilayer Switch 3650	MSW_SUMBAWA	VLAN 1	192.168.2.2/24	192.168.2.1

RANCANGAN PENGALAMATAN IP

PERANGKAT	HOSTNAME	INTERFACE	IP ADDRESS	GATEWAY
Router ISR4331	RTR_BIMA	GigabitEthernet0/0/0	192.168.3.1/24	
		Serial0/1/1	192.168.0.6/30	
Multilayer Switch 3650	MSW_BIMA	VLAN 1	192.168.3.2/24	192.168.3.1
PC	A	FastEthernet0	DHCP Client	
	B	FastEthernet0		
	C	FastEthernet0		
	D	FastEthernet0		
	E	FastEthernet0		
	F	FastEthernet0		

KONFIGURASI PERANGKAT JARINGAN

DI KANTOR PUSAT MATARAM



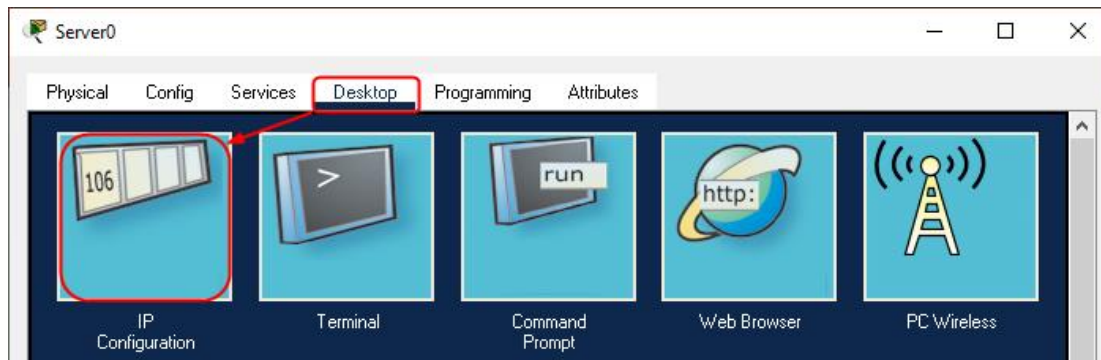
KONFIGURASI PC Server0 DI LAN MATARAM

1. Mengatur Pengalamatan IP secara Static pada PC Server0.
2. Mengaktifkan Service Domain Name System (DNS).
3. Mengatur Resource Records (RR).
4. Mengatur Service Syslog.
5. Mengatur Service Network Time Protocol (NTP).

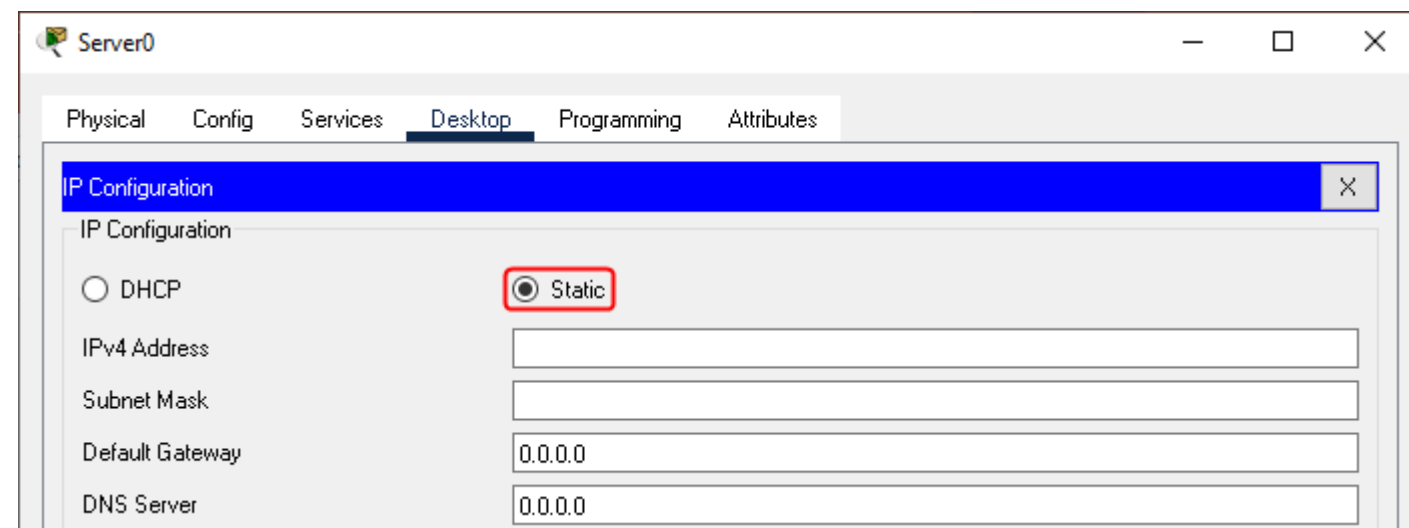


MENGATUR PENGALAMATAN IP SECARA STATIK

- Melalui **Logical Workspace** dari **Cisco Packet Tracer** klik pada **PC Server0**. Pilih tab **Desktop** pada kotak dialog yang tampil dan pilih **IP Configuration**.



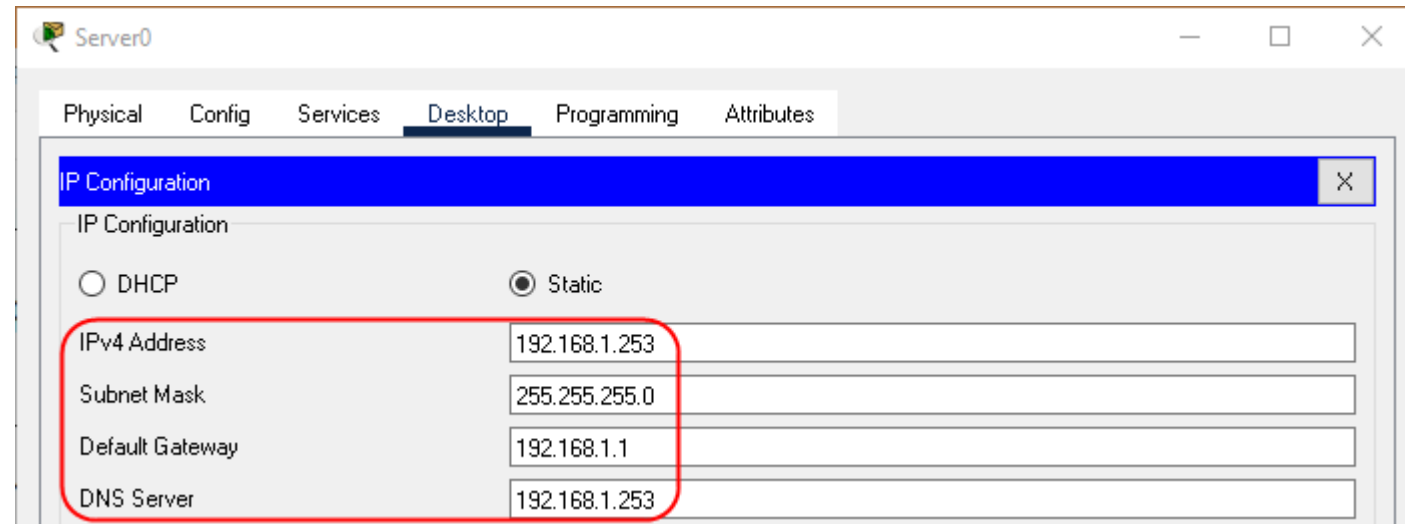
- Tampil kotak dialog **IP Configuration**, pilih **Static** untuk mengalokasikan pengalamatan IP secara manual atau statik.



MENGATUR PENGALAMATAN IP SECARA STATIK

- Lengkapi pengaturan dari parameter-parameter berikut:
 - a) **IPv4 Address** menggunakan **192.168.1.253**.
 - b) **Subnetmask** menggunakan **255.255.255.0**.
 - c) **Default Gateway** menggunakan **192.168.1.1**
 - d) **DNS Server** menggunakan **192.168.1.253**.

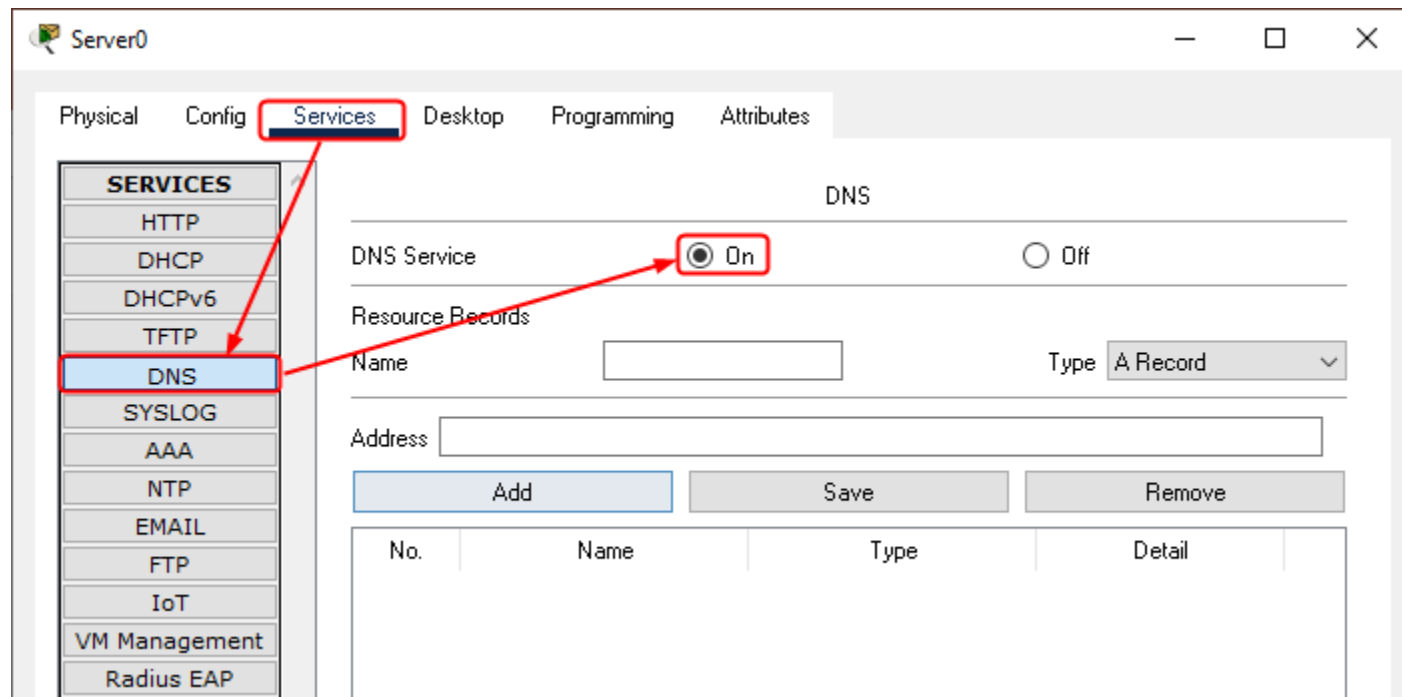
Hasil akhir pengaturan, seperti terlihat pada gambar berikut:



- Tutup kotak dialog **IP Configuration**.

MENGAKTIFKAN SERVICE DNS

- Pada kotak dialog *properties* dari **PC Server0**, pilih tab **Services** dan di panel menu sebelah kiri pilih **DNS**.
- Pada panel detail dari **DNS**, pilih **On** di pilihan parameter **DNS Service** untuk mengaktifkan *service* tersebut.

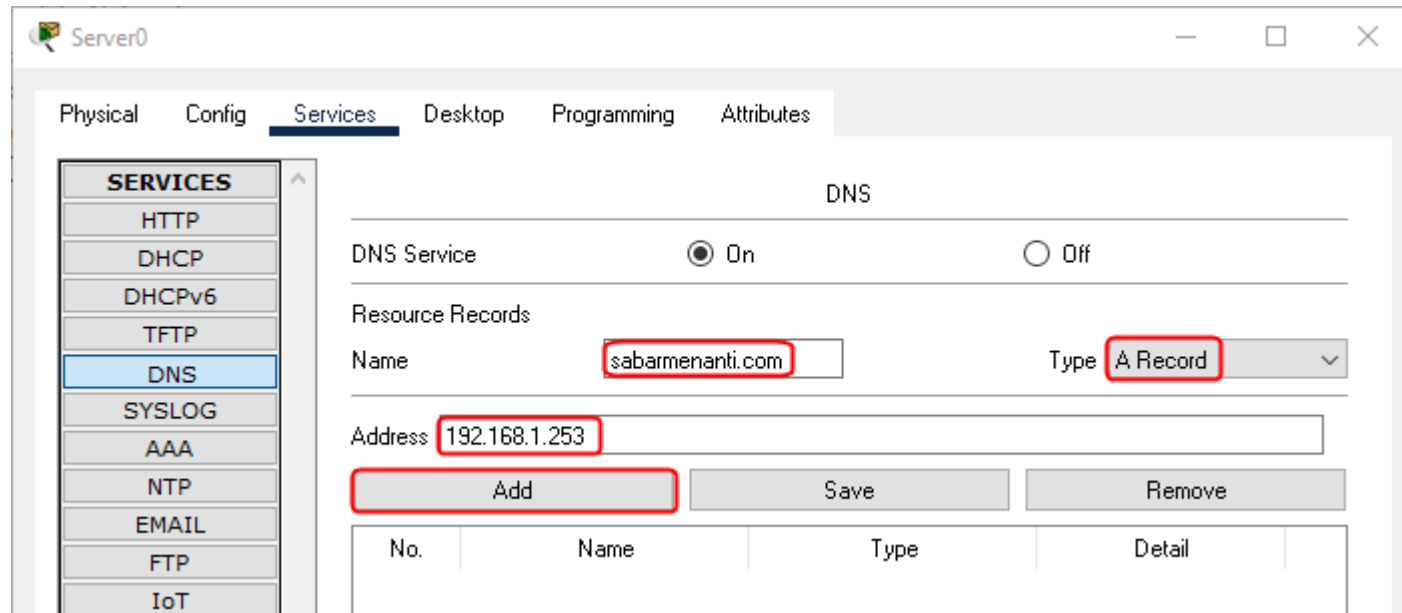


MENGATUR RESOURCE RECORDS (RR)

- Menambahkan RR untuk memetakan nama **domain sabarmenanti.com** ke alamat IP **192.168.1.253** yang merupakan alamat dari **PC Server0**.
- Lengkapi pengaturan dari parameter-parameter berikut:
 - a) **Name**, masukkan **sabarmenanti.com**.
 - b) **Type**, pilih **A Record**.
 - c) **Address**, masukkan **192.168.1.253**.

Hasil akhir pengaturan, seperti terlihat pada gambar berikut:

Tekan tombol **Add** untuk menyimpan pengaturan.



The screenshot shows the configuration window for 'Server0' with the 'Services' tab selected. Under the 'DNS' section, the 'DNS Service' is set to 'On'. In the 'Resource Records' section, a new record is being configured with the following details:

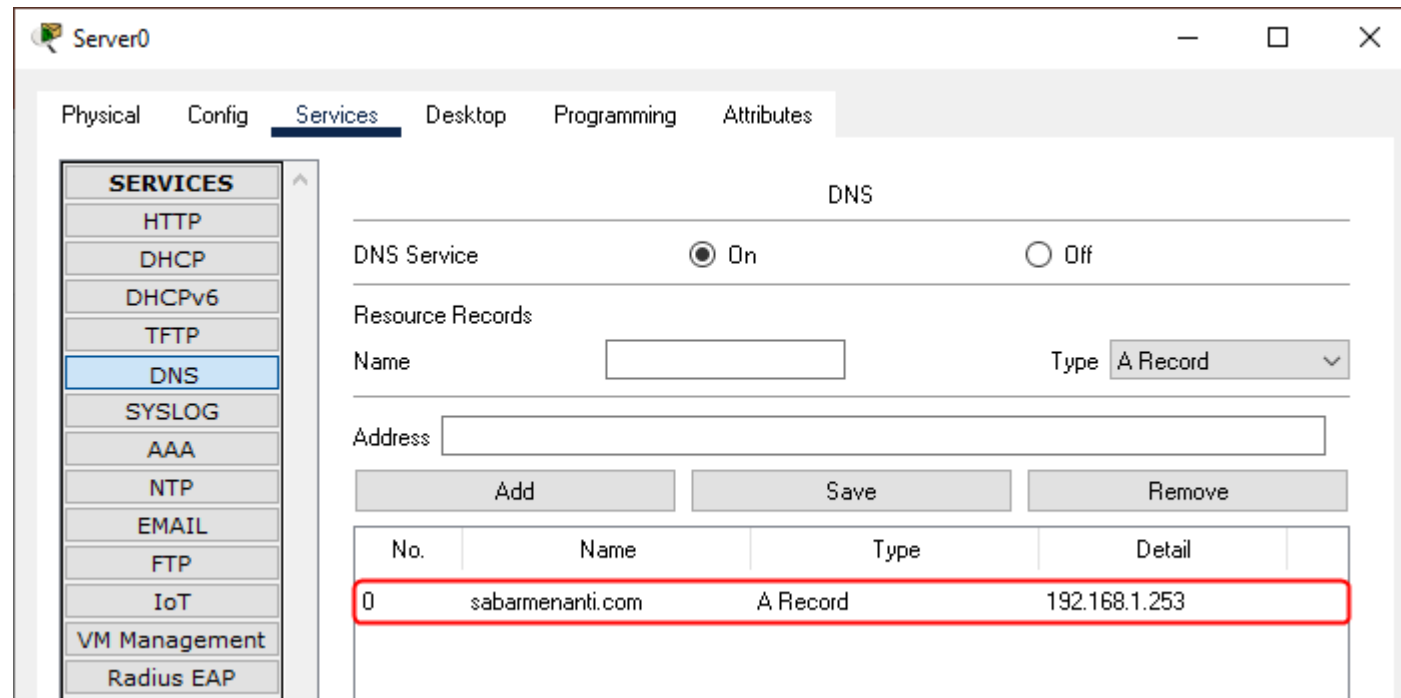
- Name: sabarmenanti.com
- Type: A Record
- Address: 192.168.1.253

The 'Add' button is highlighted with a red box, indicating the next step in the configuration process.

No.	Name	Type	Detail
-----	------	------	--------

MENGATUR RESOURCE RECORDS (RR)

- Hasil dari penambahan **RR** untuk nama **domain sabarmenanti.com**, seperti terlihat pada gambar berikut:



The screenshot shows the configuration window for 'Server0' with the 'Services' tab selected. Under the 'DNS' service, the 'DNS Service' is set to 'On'. The 'Resource Records' section shows a table with one entry:

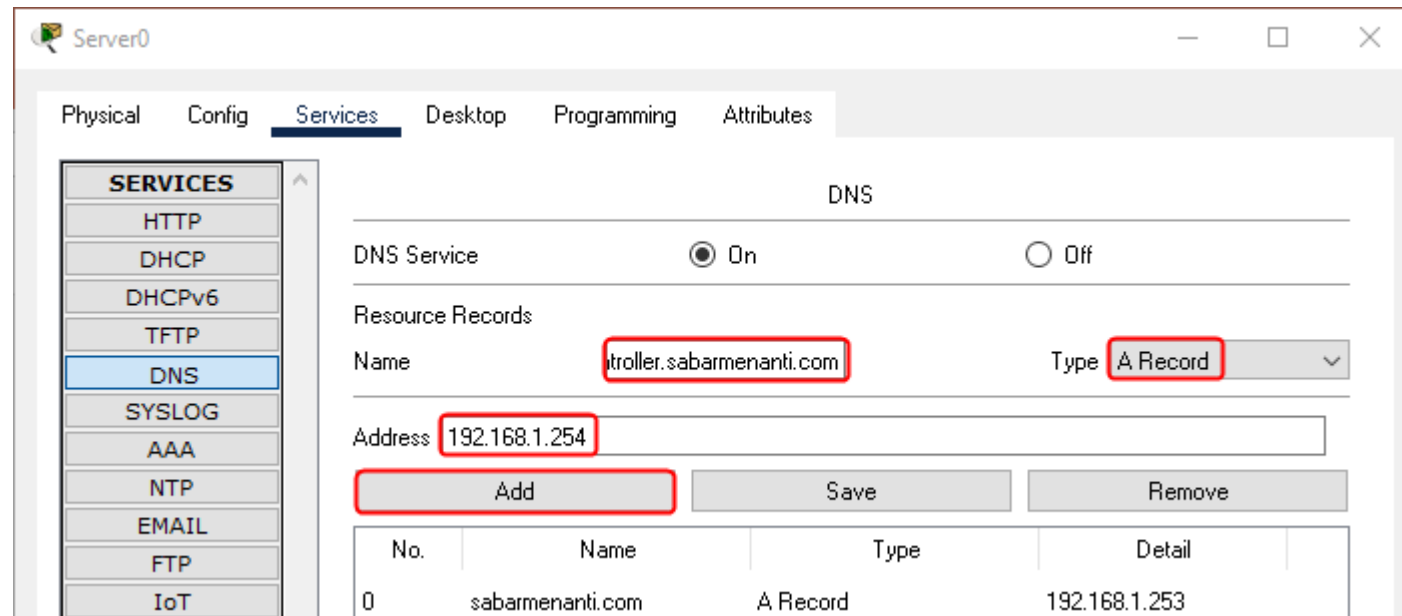
No.	Name	Type	Detail
0	sabarmenanti.com	A Record	192.168.1.253

MENGATUR RESOURCE RECORDS (RR)

- Menambahkan **RR** untuk memetakan nama **subdomain controller.sabarmenanti.com** ke alamat **IP 192.168.1.254** yang merupakan alamat dari **Network Controller PT-Controller0**.
- Lengkapi pengaturan dari parameter-parameter berikut:
 - a) **Name**, masukkan **controller.sabarmenanti.com**.
 - b) **Type**, pilih **A Record**.
 - c) **Address**, masukkan **192.168.1.254**.

Hasil akhir pengaturan, seperti terlihat pada gambar berikut:

Tekan tombol **Add** untuk menyimpan pengaturan.



The screenshot shows the Mikrotik WinBox interface for configuring DNS on a server named 'Server0'. The 'Services' tab is active, and the 'DNS' service is turned on. A new resource record is being added with the following details:

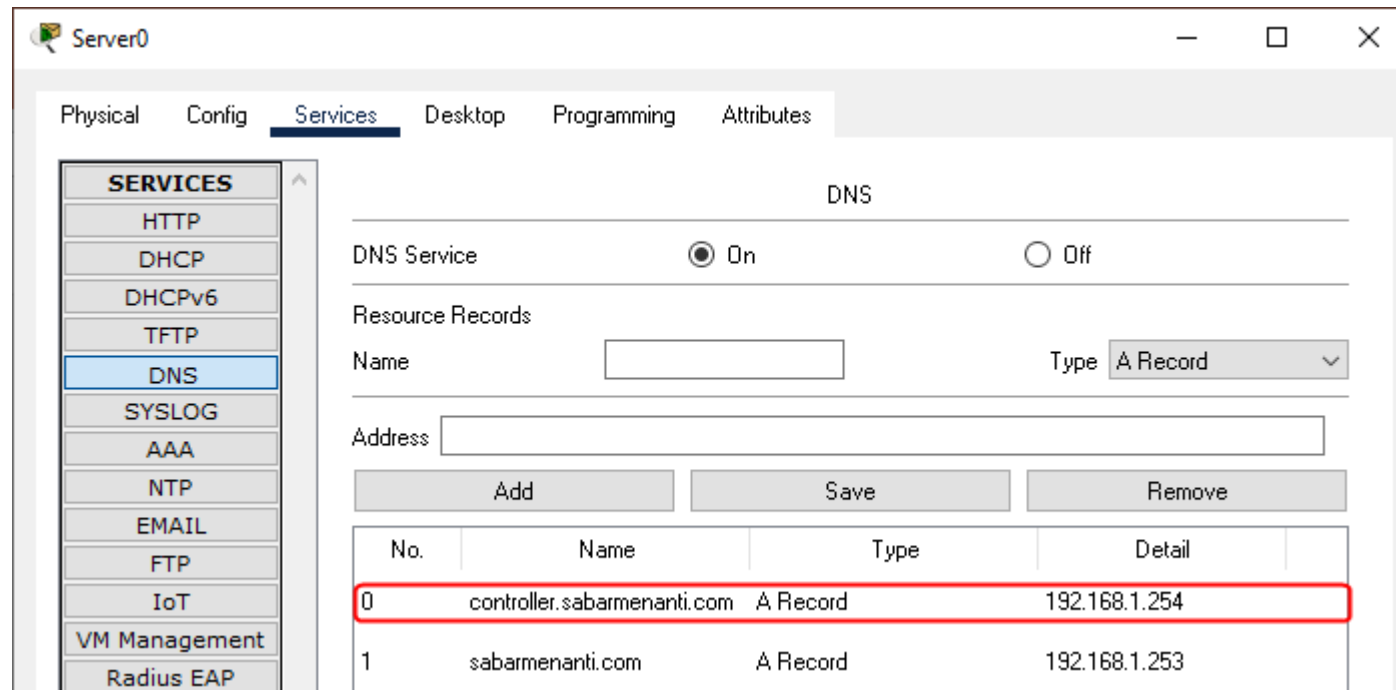
- Name:** controller.sabarmenanti.com
- Type:** A Record
- Address:** 192.168.1.254

The 'Add' button is highlighted with a red box. Below the configuration fields, a table shows the existing resource records:

No.	Name	Type	Detail
0	sabarmenanti.com	A Record	192.168.1.253

MENGATUR RESOURCE RECORDS (RR)

- Hasil dari penambahan RR untuk nama **subdomain controller.sabarmenanti.com**, seperti terlihat pada gambar berikut:

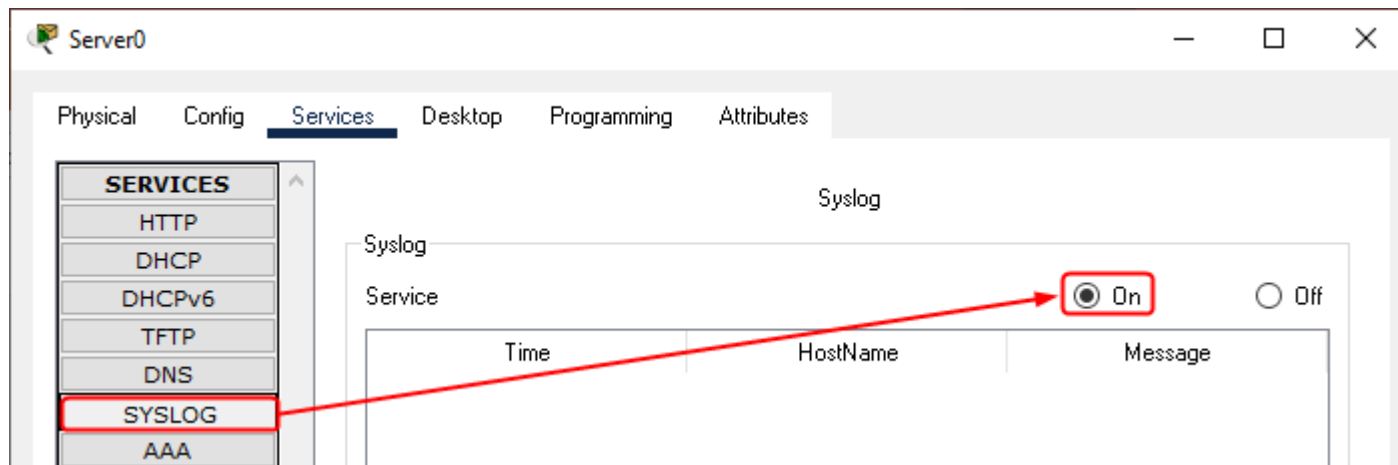


The screenshot shows the Mikrotik WinBox interface for configuring DNS on a server named 'Server0'. The 'Services' tab is active, and the 'DNS' service is turned on. Under 'Resource Records', a table lists the configured records. A red box highlights the first record, which is the newly added subdomain.

No.	Name	Type	Detail
0	controller.sabarmenanti.com	A Record	192.168.1.254
1	sabarmenanti.com	A Record	192.168.1.253

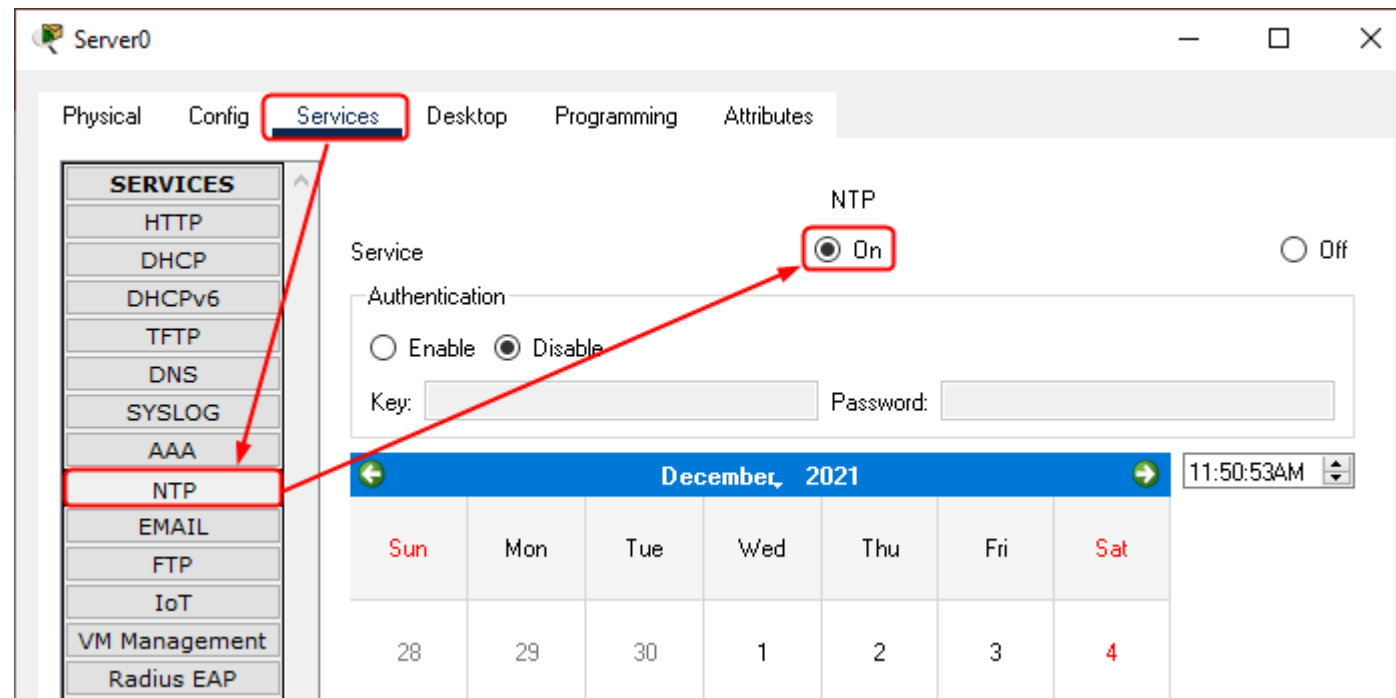
MENGATUR SERVICE SYSLOG

- **SYSLOG** merupakan standard untuk menyimpan pesan (**logging messages**) yang dihasilkan oleh **router** atau **switch** sehingga dapat diketahui aktivitas yang terjadi pada perangkat tersebut. Secara default pesan **syslog** dikirimkan ke **console** dari perangkat jaringan. Pilihan lainnya adalah mengirimkan pesan *log* tersebut ke **Server SYSLOG**.
- Pada panel menu sebelah kiri dari **Services**, pilih **SYSLOG**.
- Pada panel detail dari **SYSLOG**, pastikan terpilih **On** di pilihan parameter **Service** untuk mengaktifkan *service* tersebut sehingga **Server0** bertindak sebagai **Server SYSLOG**.



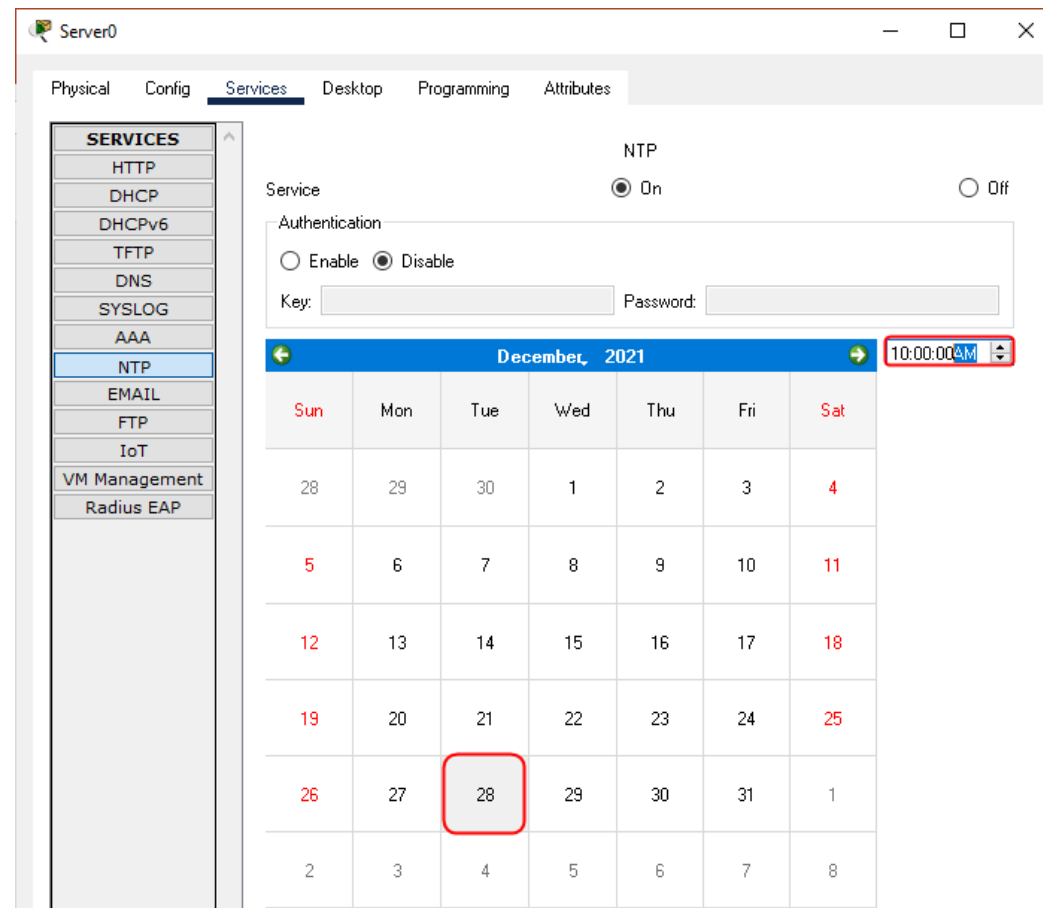
MENGATUR SERVICE NTP

- **NTP** merupakan protocol yang digunakan untuk melakukan sinkronisasi waktu pada sistem komputer ke sumber referensi waktu lainnya di jaringan. NTP menggunakan arsitektur **Client/Server**.
- Pada panel menu sebelah kiri dari **Services**, pilih **NTP**.
- Pada panel detail dari **NTP**, pastikan terpilih **On** di pilihan parameter **Service** untuk mengaktifkan *service* tersebut sehingga **Server0** bertindak sebagai **Server NTP**.



MENGATUR SERVICE NTP

- Lakukan penyesuaian pada tanggal dan waktu, sebagai contoh menggunakan 28 December 2021, jam 10:00:00AM, seperti terlihat pada gambar berikut:



Tutup kotak dialog *properties* dari PC Server0.

KONFIGURASI DASAR DI ROUTER RTR_MATARAM

1. Mengatur hostname.
2. Mengatur pengalamatan IP pada interface.
3. Mengatur password privilege mode.
4. Mengatur Secure Shell (SSH) Server.
5. Mengatur Server Dynamic Host Configuration Protocol (DHCP).
6. Mengatur routing protocol Enhanced Interior Gateway Routing Protocol (EIGRP).



MENGAKSES COMMAND LINE INTERFACE (CLI) DARI ROUTER RTR_MATARAM

- Melalui **Logical Workspace** dari **Cisco Packet Tracer**, klik pada **router RTR_MATARAM**.
- Pada kotak dialog *properties* dari **RTR_MATARAM** yang tampil, klik tab **CLI**.
- Apabila tampil **System Configuration Dialog** maka ketik **no** pada inputan “*Would you like to enter the initial configuration dialog? [yes/no].*” dan tekan **Enter**.
- Tekan **Enter** untuk memunculkan **prompt CLI Router>**.

```
at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

cisco ISR4331/K9 (1RU) processor with 1795999K/6147K bytes of memory.
Processor board ID FLM232010G0
3 Gigabit Ethernet interfaces
2 Serial interfaces
32768K bytes of non-volatile configuration memory.
4194304K bytes of physical memory.
3207167K bytes of flash memory at bootflash:.
0K bytes of WebUI ODM Files at webui:.

      --- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: no

Press RETURN to get started!

Router>
```

MENGATUR HOSTNAME

- Berpindah dari mode **user** ke mode **privilege**.

```
Router>enable
```

- Berpindah dari mode **privilege** ke mode **global configuration**.

```
Router#conf t
```

- Mengatur **hostname** dari *router* menggunakan **RTR_MATARAM**.

```
Router(config)#hostname RTR_MATARAM  
RTR_MATARAM(config)#
```

- Terlihat **hostname** telah berubah dari **Router** menjadi **RTR_MATARAM**.

MENGATUR PENGALAMATAN IP PADA INTERFACE

- Berpindah ke **interface configuration** dari **GigabitEthernet0/0/0** atau disingkat **g0/0/0**.

```
RTR_MATARAM(config)#int g0/0/0
```

- Mengatur pengalamatan IP pada **interface g0/0/0** dengan nilai **192.168.1.1/24**.

```
RTR_MATARAM(config-if)#ip address 192.168.1.1 255.255.255.0
```

- Mengaktifkan **interface**.

```
RTR_MATARAM(config-if)#no shutdown
```

- Berpindah ke **interface configuration** dari **Serial0/1/0** atau disingkat **s0/1/0**.

```
RTR_MATARAM(config)#int s0/1/0
```

- Mengatur pengalamatan IP pada **interface s0/1/0** dengan nilai **192.168.0.1/30**.

```
RTR_MATARAM(config-if)#ip address 192.168.0.1 255.255.255.252
```

- Mengaktifkan **interface**.

```
RTR_MATARAM(config-if)#no shutdown
```

MEMVERIFIKASI PENGATURAN PENGALAMATAN IP PADA INTERFACE

- Memverifikasi pengalamatan IP pada **interface g0/0/0** dan **s0/1/0** secara ringkas.

```
RTR_MATARAM(config-if)#do show ip int brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
GigabitEthernet0/0/0	192.168.1.1	YES	manual	up	up
GigabitEthernet0/0/1	unassigned	YES	NVRAM	administratively down	down
GigabitEthernet0/0/2	unassigned	YES	NVRAM	administratively down	down
Serial0/1/0	192.168.0.1	YES	manual	down	down
Serial0/1/1	unassigned	YES	NVRAM	administratively down	down
Vlan1	unassigned	YES	NVRAM	administratively down	down

- Terlihat **interface GigabitEthernet0/0/0** telah menggunakan alamat IP **192.168.1.1**. Sedangkan **interface Serial0/1/0** juga telah menggunakan alamat IP **192.168.0.1**.

MENGATUR PASSWORD PRIVILEGE MODE DAN SECURE SHELL (SSH) SERVER

- Berpindah ke satu mode sebelumnya yaitu dari **interface configuration** ke **global configuration mode**.

```
RTR_MATARAM(config-if)#exit
```

- Mengatur **password privilege mode** dengan sandi “**menanti**”.

```
RTR_MATARAM(config)#enable secret menanti
```

- Mengatur nama **domain** dengan nilai “**sabarmenanti.com**”.

```
RTR_MATARAM(config)#ip domain-name sabarmenanti.com
```

- Membuat **RSA key** dengan **modulus 1024**.

```
RTR_MATARAM(config)#crypto key generate rsa general-keys modulus 1024
% You already have RSA keys defined named RTR_MATARAM.sabarmenanti.com
% They will be replaced.
```

```
% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
*Mar 1 0:56:59.800: %SSH-5-ENABLED: SSH 1.99 has been enabled
```


MENGATUR SECURE SHELL (SSH) SERVER

- Membuat akun otentikasi pengguna yang mengakses melalui SSH dengan nama login “**sabar**” dan sandi login “**menanti**” yang disimpan pada **local database**.

```
RTR_MATARAM(config)#username sabar password menanti
```

- Mengaktifkan SSH pada **line virtual teletype (vty)** untuk 5 (lima) pengguna.

```
RTR_MATARAM(config)#line vty 0 4
```

- Mengotentikasi keseluruhan sesi **incoming virtual terminal** menggunakan **local username database**.

```
RTR_MATARAM(config-line)#login local
```

- Mengatur **SSH** sebagai protokol yang dapat digunakan ketika terkoneksi melalui **line vty**.

```
RTR_MATARAM(config-line)#transport input ssh
```

- Berpindah ke mode satu mode sebelumnya.

```
RTR_MATARAM(config-line)#exit
```

MENGATUR SERVER DHCP

- Membuat ruang alamat IP (**pool**) yang disewakan ke **DHCP Client** dengan nama **MATARAM**.
`RTR_MATARAM(config)#ip dhcp pool MATARAM`
- Mengatur alamat **network** dan **subnetmask** dari alamat IP yang akan disewakan ke **DHCP Client** yaitu **192.168.1.0/24**.
`RTR_MATARAM(dhcp-config)#network 192.168.1.0 255.255.255.0`
- Mengatur **default gateway** yang diperoleh **DHCP Client** yaitu **192.168.1.1**.
`RTR_MATARAM(dhcp-config)#default-router 192.168.1.1`
- Mengatur alamat **server DNS** yang diperoleh **DHCP client** yaitu **192.168.1.253**.
`RTR_MATARAM(dhcp-config)#dns-server 192.168.1.253`
- Berpindah ke satu mode sebelumnya.
`RTR_MATARAM(dhcp-config)#exit`
`RTR_MATARAM(config)#`

MENGATUR SERVER DHCP

- Mengatur agar alamat IP yang tidak disewakan ke **DHCP Client** oleh **Server DHCP** yaitu dari **192.168.1.1** sampai dengan **192.168.1.2**.

```
RTR_MATARAM(config)#ip dhcp excluded-address 192.168.1.1 192.168.1.2
```

- Mengatur agar alamat IP yang tidak disewakan ke **DHCP Client** oleh **Server DHCP** yaitu dari **192.168.1.253** sampai dengan **192.168.1.254**.

```
RTR_MATARAM(config)#ip dhcp excluded-address 192.168.1.253 192.168.1.254
```

MENGATUR ROUTING PROTOCOL EIGRP

- Mengaktifkan **routing protocol EIGRP** dengan **Autonomous System (AS) Number 64512**.

```
RTR_MATARAM(config)#router eigrp 64512
```

- Mengatur alamat jaringan dimana **router RTR_MATARAM** terhubung secara langsung yang dituliskan secara **classfull** yaitu **network 192.168.1.0** yang merupakan **LAN MATARAM**.

```
RTR_MATARAM(config-router)#network 192.168.1.0
```

- Mengatur alamat jaringan dimana **router RTR_MATARAM** terhubung secara langsung yang dituliskan secara **classfull** yaitu **network 192.168.0.0** yang merupakan **Subnet Point-to-Point MATARAM-SUMBAWA**.

```
RTR_MATARAM(config-router)#network 192.168.0.0
```

- Berpindah ke mode **privilege**.

```
RTR_MATARAM(config-router)#end  
RTR_MATARAM#
```

MEMVERIFIKASI ROUTING PROTOCOL YANG AKTIF

- Memverifikasi *routing protocol* yang aktif pada **router**.

```
RTR_MATARAM#show ip protocols
```

```
Routing Protocol is "eigrp 64512"
```

```
Outgoing update filter list for all interfaces is not set
```

```
Incoming update filter list for all interfaces is not set
```

```
Default networks flagged in outgoing updates
```

```
Default networks accepted from incoming updates
```

```
EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
```

```
EIGRP maximum hopcount 100
```

```
EIGRP maximum metric variance 1
```

```
Redistributing: eigrp 64512
```

```
Automatic network summarization is in effect
```

```
Automatic address summarization:
```

```
Maximum path: 4
```

```
Routing for Networks:
```

```
192.168.1.0
```

```
192.168.0.0
```

```
Routing Information Sources:
```

```
Gateway Distance Last Update
```

```
Distance: internal 90 external 170
```

MEMVERIFIKASI DAN MENYIMPAN KONFIGURASI

- Menampilkan informasi konfigurasi yang sedang berjalan atau aktif di **Random Access Memory (RAM)**.

```
RTR_MATARAM#show run
```

Tampil pesan `--More--` di bagian pojok kiri bawah dari *output* eksekusi perintah tersebut.

Tekan **spasi** untuk menampilkan *output* secara lengkap atau tekan **Enter** untuk menampilkan per baris. Untuk keluar dari *output* tersebut tekan sembarang tombol, sebagai contoh **q**.

- Menyimpan konfigurasi secara permanen dari **RAM** ke **Non Volatile Random Access Memory (NVRAM)**.

```
RTR_MATARAM#copy run start  
Destination filename [startup-config]? ←  
Building configuration...  
[OK]
```

Tekan **Enter** pada pesan konfirmasi “**Destination filename [startup-config]?**” yang menanyakan tentang nama *file* sebagai tujuan penyimpanan konfigurasi.

MEMVERIFIKASI DAN MENYIMPAN KONFIGURASI

- Menampilkan informasi konfigurasi tersimpan di NVRAM.

```
RTR_MATARAM#show start
```

Tampil pesan `--More--` di bagian pojok kiri bawah dari *output* eksekusi perintah tersebut.

Tekan **spasi** untuk menampilkan *output* secara lengkap atau tekan **Enter** untuk menampilkan per baris.

Untuk keluar dari *output* tersebut tekan sembarang tombol, sebagai contoh **q**.

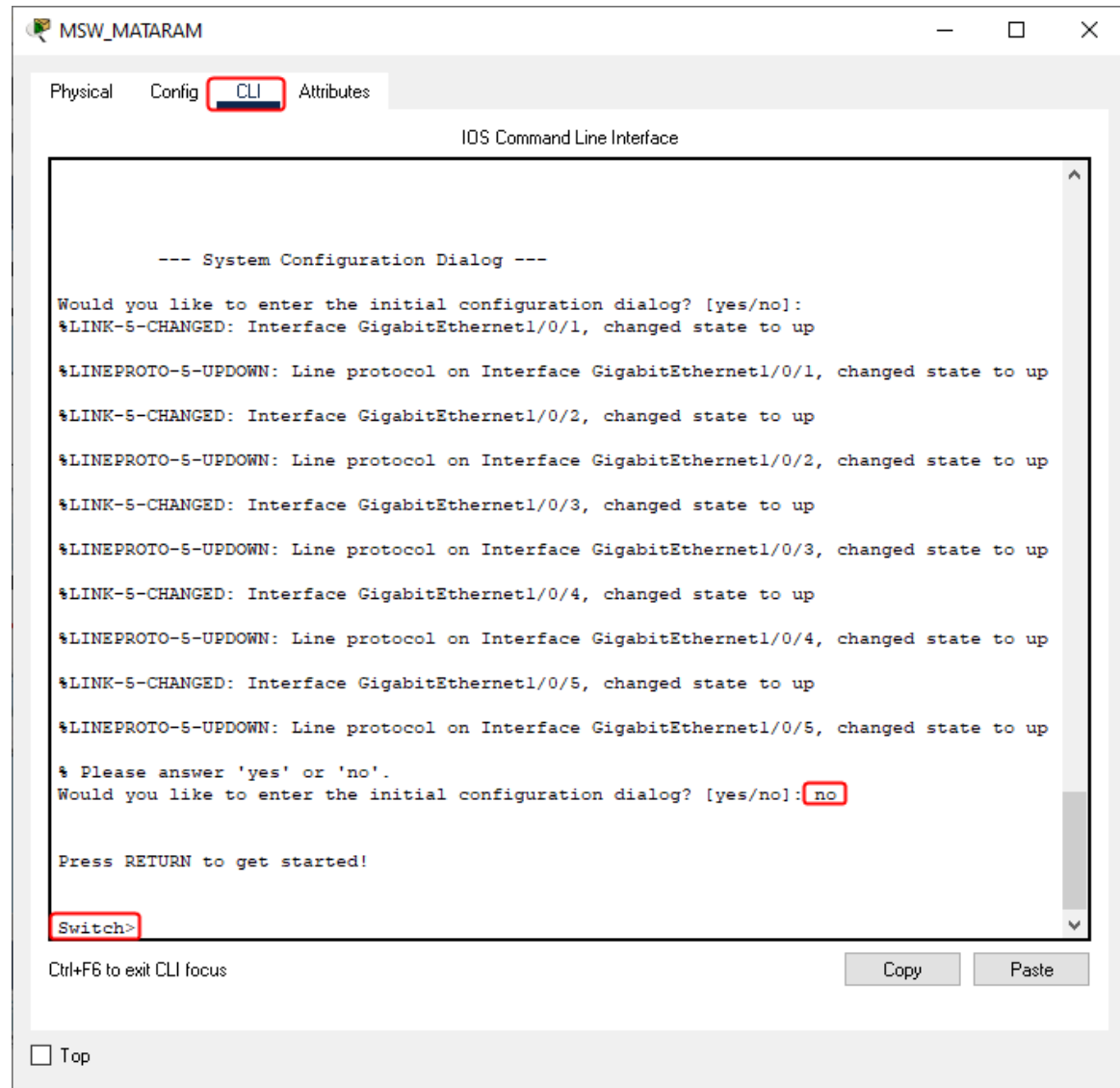
KONFIGURASI DASAR DI MULTILAYER SWITCH MSW_MATARAM

1. Mengatur Hostname.
2. Mengatur Pengalamatan IP pada Interface dan Default Gateway.
3. Mengatur Password Privilege Mode.
4. Mengatur Secure Shell (SSH) Server.



MENGAKSES COMMAND LINE INTERFACE (CLI) DARI MSW_MATARAM

- Melalui **Logical Workspace** dari **Cisco Packet Tracer**, klik pada **Multilayer Switch MSW_MATARAM**.
- Pada kotak dialog *properties* dari **MSW_MATARAM** yang tampil, klik tab **CLI**.
- Apabila tampil **System Configuration Dialog** maka ketik **no** pada inputan “*Would you like to enter the initial configuration dialog? [yes/no].*” dan tekan **Enter**.
- Tekan **Enter** untuk memunculkan **prompt CLI** `Switch>`.



MENGATUR HOSTNAME

- Berpindah dari mode **user** ke mode **privilege**.

```
Switch>enable
```

- Berpindah dari mode **privilege** ke mode **global configuration**.

```
Switch#conf t
```

- Mengatur **hostname** dari *multilayer switch* menggunakan **MSW_MATARAM**.

```
Switch(config)#hostname MSW_MATARAM  
MSW_MATARAM(config)#
```

- Terlihat **hostname** telah berubah dari **Switch** menjadi **MSW_MATARAM**.

MENGATUR PENGALAMATAN IP PADA INTERFACE DAN DEFAULT GATEWAY

- Berpindah ke **interface configuration** dari **VLAN 1**.

```
MSW_MATARAM(config)#int vlan 1
```

- Mengatur pengalamatan IP pada **interface VLAN 1** dengan nilai **192.168.1.2/24**.

```
MSW_MATARAM(config-if)#ip address 192.168.1.2 255.255.255.0
```

- Mengaktifkan **interface**.

```
MSW_MATARAM(config-if)#no shutdown
```

- Berpindah ke satu mode sebelumnya.

```
MSW_MATARAM(config-if)#exit
```

- Mengatur **default gateway** agar *multilayer switch* dapat berkomunikasi ke beda jaringan dengan nilai **192.168.1.1**.

```
MSW_MATARAM(config)#ip default-gateway 192.168.1.1
```

MEMVERIFIKASI PENGATURAN PENGALAMATAN IP PADA INTERFACE

- Memverifikasi pengalamatan IP pada **interface vlan 1**.

```
MSW_MATARAM(config)#do show int vlan1
Vlan1 is up, line protocol is up
Hardware is CPU Interface, address is 0001.c912.88eb (bia 0001.c912.88eb)
Internet address is 192.168.1.2/24
MTU 1500 bytes, BW 100000 Kbit, DLY 1000000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
ARP type: ARPA, ARP Timeout 04:00:00
Last input 21:40:21, output never, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
 1682 packets input, 530955 bytes, 0 no buffer
Received 0 broadcasts (0 IP multicast)
 0 runts, 0 giants, 0 throttles
 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
563859 packets output, 0 bytes, 0 underruns
 0 output errors, 23 interface resets
 0 output buffer failures, 0 output buffers swapped out
```

- Terlihat status **interface vlan 1** telah aktif (**up**) dan telah menggunakan alamat IP **192.168.1.2/24**.

MEMVERIFIKASI PENGATURAN DEFAULT GATEWAY

- Eksekusi perintah `do show run | include ip default-gateway` untuk memverifikasi hasil pengaturan **default gateway** pada *Multilayer Switch*.

```
MSW_MATARAM(config)#do show run | include ip default-gateway  
ip default-gateway 192.168.1.1
```

- Terlihat telah menggunakan IP **192.168.1.1** sebagai *default gateway*.

MENGATUR PASSWORD PRIVILEGE MODE DAN SECURE SHELL (SSH) SERVER

- Mengatur **password privilege mode** dengan sandi “**menanti**”.

```
MSW_MATARAM(config)#enable secret menanti
```

- Mengatur nama **domain** dengan nilai “**sabarmenanti.com**”.

```
MSW_MATARAM(config)#ip domain-name sabarmenanti.com
```

- Membuat **RSA key** dengan **modulus 1024**.

```
MSW_MATARAM(config)#crypto key generate rsa general-keys modulus 1024
% You already have RSA keys defined named MSW_MATARAM.sabarmenanti.com
% They will be replaced.
```

```
% The key modulus size is 1024 bits
```

```
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
```

```
*Mar 1 0:27:42.566: %SSH-5-ENABLED: SSH 1.99 has been enabled
```

MENGATUR SECURE SHELL (SSH) SERVER

- Membuat akun otentikasi pengguna yang mengakses melalui SSH dengan nama login “**sabar**” dan sandi login “**menanti**” yang disimpan pada **local database**.

```
MSW_MATARAM(config)#username sabar password menanti
```

- Mengaktifkan SSH pada **line virtual teletype (vty)** untuk 5 (lima) pengguna.

```
MSW_MATARAM(config)#line vty 0 4
```

- Mengotentikasi keseluruhan sesi **incoming virtual terminal** menggunakan **local username database**.

```
MSW_MATARAM(config-line)#login local
```

- Mengatur **SSH** sebagai protokol yang dapat digunakan ketika terkoneksi melalui **line vty**.

```
MSW_MATARAM(config-line)#transport input ssh
```

- Berpindah ke mode **privilege**.

```
MSW_MATARAM(config-line)#end
```

```
MSW_MATARAM#
```

MEMVERIFIKASI DAN MENYIMPAN KONFIGURASI

- Menampilkan informasi konfigurasi yang sedang berjalan atau aktif di **Random Access Memory (RAM)**.

```
MSW_MATARAM#show run
```

Tampil pesan `--More--` di bagian pojok kiri bawah dari *output* eksekusi perintah tersebut.

Tekan **spasi** untuk menampilkan *output* secara lengkap atau tekan **Enter** untuk menampilkan per baris. Untuk keluar dari *output* tersebut tekan sembarang tombol, sebagai contoh **q**.

- Menyimpan konfigurasi secara permanen dari **RAM** ke **Non Volatile Random Access Memory (NVRAM)**.

```
MSW_MATARAM#copy run start  
Destination filename [startup-config]? ←  
Building configuration...  
[OK]
```

Tekan **Enter** pada pesan konfirmasi “**Destination filename [startup-config]?**” yang menanyakan tentang nama *file* sebagai tujuan penyimpanan konfigurasi.

MEMVERIFIKASI DAN MENYIMPAN KONFIGURASI

- Menampilkan informasi konfigurasi tersimpan di NVRAM.

```
MSW_MATARAM#show start
```

Tampil pesan `--More--` di bagian pojok kiri bawah dari *output* eksekusi perintah tersebut.

Tekan **spasi** untuk menampilkan *output* secara lengkap atau tekan **Enter** untuk menampilkan per baris.

Untuk keluar dari *output* tersebut tekan sembarang tombol, sebagai contoh **q**.

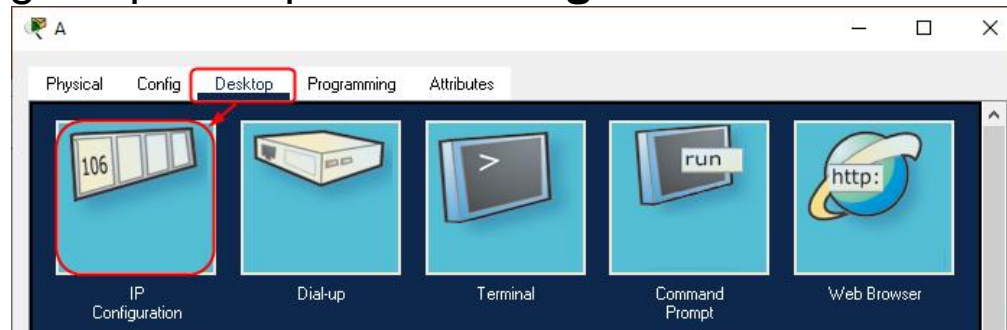
KONFIGURASI DI PC CLIENT LAN MATARAM

Mengatur alokasi pengalamatan IP secara dinamis yaitu sebagai **DHCP Client** pada **PC A** dan **PC B**.

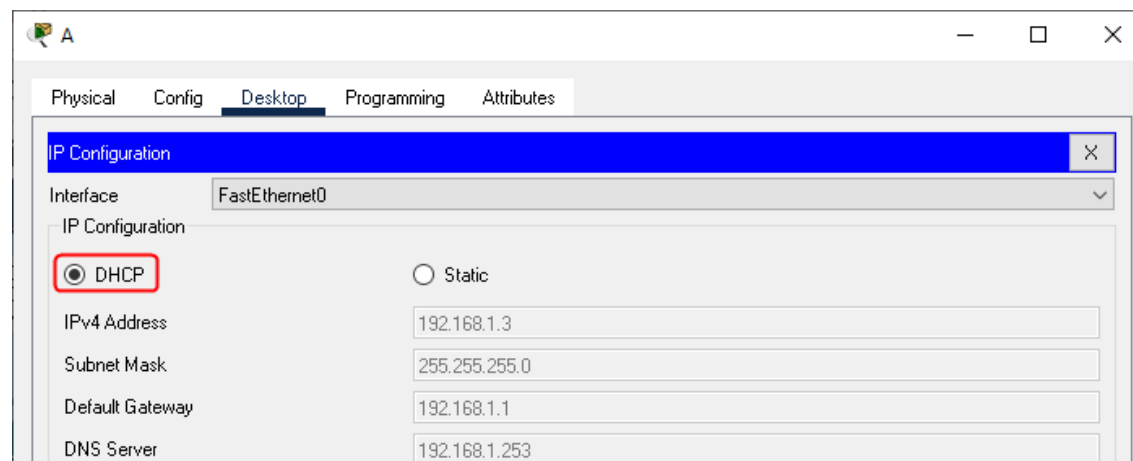


KONFIGURASI DHCP CLIENT PADA PC A DI LAN MATARAM

- Melalui **Logical Workspace** dari **Cisco Packet Tracer** klik pada **PC A**. Pilih tab **Desktop** pada kotak dialog yang tampil dan pilih **IP Configuration**.



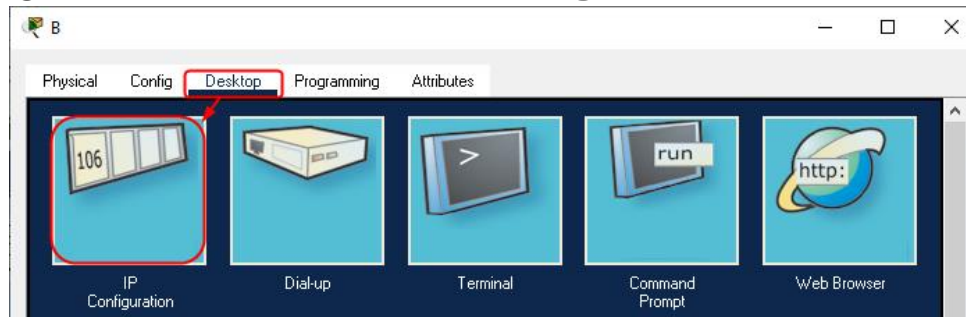
- Tampil kotak dialog **IP Configuration**, pilih **DHCP** untuk mengalokasikan pengalamatan IP secara dinamis.



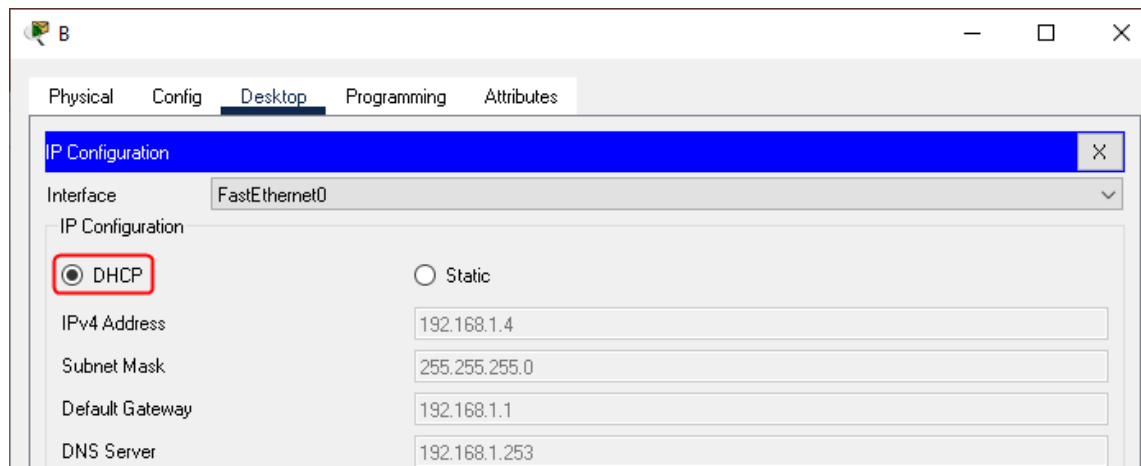
Terlihat **PC A** telah berhasil memperoleh pengalamatan IP dan parameter TCP/IP lainnya dari **Server DHCP**. Tutup kotak dialog **PC A**.

KONFIGURASI DHCP CLIENT PADA PC B DI LAN MATARAM

- Melalui **Logical Workspace** dari **Cisco Packet Tracer** klik pada **PC B**. Pilih tab **Desktop** pada kotak dialog yang tampil dan pilih **IP Configuration**.



- Tampil kotak dialog **IP Configuration**, pilih **DHCP** untuk mengalokasikan pengalamatan IP secara dinamis.



Terlihat **PC B** telah berhasil memperoleh pengalamatan IP dan parameter TCP/IP lainnya dari **Server DHCP**. Tutup kotak dialog **PC B**.

KONFIGURASI PERANGKAT JARINGAN

DI KANTOR CABANG SUMBAWA

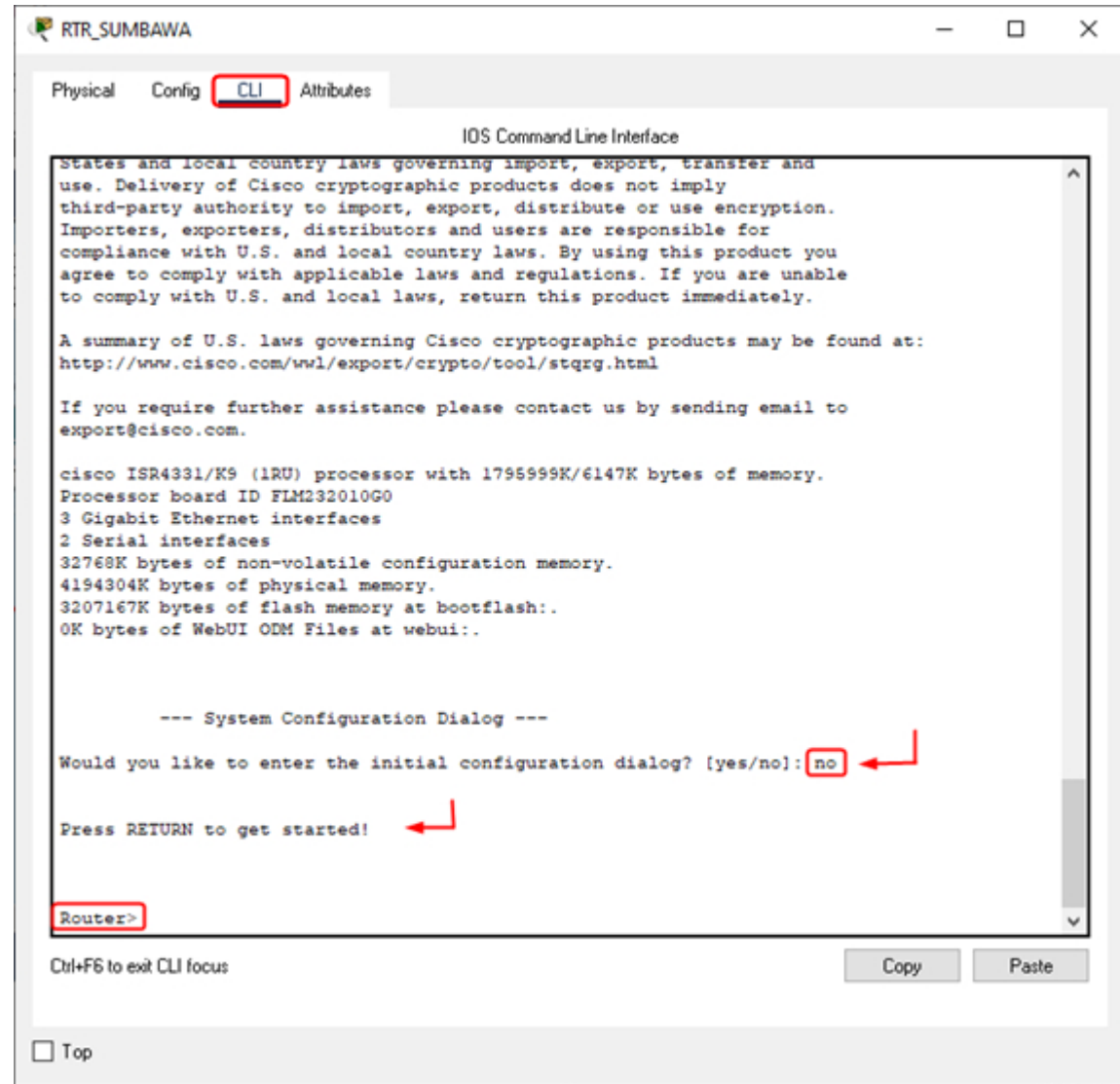


KONFIGURASI DASAR DI ROUTER RTR_SUMBAWA

1. Mengatur hostname.
2. Mengatur pengalamatan IP pada interface.
3. Mengatur password privilege mode.
4. Mengatur Secure Shell (SSH) Server.
5. Mengatur Server Dynamic Host Configuration Protocol (DHCP).
6. Mengatur routing protocol Enhanced Interior Gateway Routing Protocol (EIGRP).

MENGAKSES COMMAND LINE INTERFACE (CLI) DARI ROUTER RTR_SUMBAWA

- Melalui **Logical Workspace** dari **Cisco Packet Tracer**, klik pada router **RTR_SUMBAWA**.
- Pada kotak dialog *properties* dari **RTR_SUMBAWA** yang tampil, klik tab **CLI**.
- Apabila tampil **System Configuration Dialog** maka ketik **no** pada inputan “*Would you like to enter the initial configuration dialog? [yes/no].*” dan tekan **Enter**.
- Tekan **Enter** untuk memunculkan **prompt CLI Router>**.



MENGATUR HOSTNAME

- Berpindah dari mode **user** ke mode **privilege**.

```
Router>enable
```

- Berpindah dari mode **privilege** ke mode **global configuration**.

```
Router#conf t
```

- Mengatur **hostname** dari *router* menggunakan **RTR_SUMBAWA**.

```
Router(config)#hostname RTR_SUMBAWA  
RTR_SUMBAWA(config)#
```

- Terlihat **hostname** telah berubah dari **Router** menjadi **RTR_SUMBAWA**.

MENGATUR PENGALAMATAN IP PADA INTERFACE

- Berpindah ke **interface configuration** dari **GigabitEthernet0/0/0** atau disingkat **g0/0/0**.

```
RTR_SUMBAWA(config)#int g0/0/0
```

- Mengatur pengalamatan IP pada **interface g0/0/0** dengan nilai **192.168.2.1/24**.

```
RTR_SUMBAWA(config-if)#ip address 192.168.2.1 255.255.255.0
```

- Mengaktifkan **interface**.

```
RTR_SUMBAWA(config-if)#no shutdown
```

- Berpindah ke **interface configuration** dari **Serial0/1/0** atau disingkat **s0/1/0**.

```
RTR_SUMBAWA(config-if)#int s0/1/0
```

- Mengatur pengalamatan IP pada **interface s0/1/0** dengan nilai **192.168.0.2/30**.

```
RTR_SUMBAWA(config-if)#ip address 192.168.0.2 255.255.255.252
```

- Mengaktifkan **interface**.

```
RTR_SUMBAWA(config-if)#no shutdown
```

MENGATUR PENGALAMATAN IP PADA INTERFACE

- Berpindah ke **interface configuration** dari **Serial0/1/1** atau disingkat **s0/1/1**.

```
RTR_SUMBAWA(config-if)#int s0/1/1
```

- Mengatur pengalamatan IP pada **interface s0/1/1** dengan nilai **192.168.0.5/30**.

```
RTR_SUMBAWA(config-if)#ip address 192.168.0.5 255.255.255.252
```

- Mengaktifkan **interface**.

```
RTR_SUMBAWA(config-if)#no shutdown
```

MEMVERIFIKASI PENGATURAN PENGALAMATAN IP PADA INTERFACE

- Memverifikasi pengalamatan IP pada **interface g0/0/0** dan **s0/1/0** serta **s0/1/1** secara ringkas.

```
RTR_SUMBAWA(config-if) #do show ip int brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
GigabitEthernet0/0/0	192.168.2.1	YES	manual	up	up
GigabitEthernet0/0/1	unassigned	YES	NVRAM	administratively down	down
GigabitEthernet0/0/2	unassigned	YES	NVRAM	administratively down	down
Serial0/1/0	192.168.0.2	YES	manual	up	up
Serial0/1/1	192.168.0.5	YES	manual	down	down
Vlan1	unassigned	YES	NVRAM	administratively down	down

- Terlihat **interface GigabitEthernet0/0/0** telah menggunakan alamat **IP 192.168.2.1** dan **interface Serial0/1/0** juga telah menggunakan alamat **IP 192.168.0.2** Sedangkan **interface Serial0/1/1** telah menggunakan alamat **IP 192.168.0.5**.

MENGATUR PASSWORD PRIVILEGE MODE DAN SECURE SHELL (SSH) SERVER

- Berpindah ke satu mode sebelumnya yaitu dari **interface configuration** ke **global configuration mode**.

```
RTR_SUMBAWA(config-if)#exit
```

- Mengatur **password privilege mode** dengan sandi “**menanti**”.

```
RTR_SUMBAWA(config)#enable secret menanti
```

- Mengatur nama **domain** dengan nilai “**sabarmenanti.com**”.

```
RTR_SUMBAWA(config)#ip domain-name sabarmenanti.com
```

- Membuat **RSA key** dengan **modulus 1024**.

```
RTR_SUMBAWA(config)#crypto key generate rsa general-keys modulus 1024
% You already have RSA keys defined named RTR_SUMBAWA.sabarmenanti.com
% They will be replaced.
```

```
% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
*Mar 1 0:17:40.953: %SSH-5-ENABLED: SSH 1.99 has been enabled
```

MENGATUR SECURE SHELL (SSH) SERVER

- Membuat akun otentikasi pengguna yang mengakses melalui SSH dengan nama login “**sabar**” dan sandi login “**menanti**” yang disimpan pada **local database**.

```
RTR_SUMBAWA(config)#username sabar password menanti
```

- Mengaktifkan SSH pada **line virtual teletype (vty)** untuk 5 (lima) pengguna.

```
RTR_SUMBAWA(config)#line vty 0 4
```

- Mengotentikasi keseluruhan sesi **incoming virtual terminal** menggunakan **local username database**.

```
RTR_SUMBAWA(config-line)#login local
```

- Mengatur **SSH** sebagai protokol yang dapat digunakan ketika terkoneksi melalui **line vty**.

```
RTR_SUMBAWA(config-line)#transport input ssh
```

- Berpindah ke mode satu mode sebelumnya.

```
RTR_SUMBAWA(config-line)#exit
```

MENGATUR SERVER DHCP

- Membuat ruang alamat IP (**pool**) yang disewakan ke **DHCP Client** dengan nama **SUMBAWA**.

```
RTR_SUMBAWA(config)#ip dhcp pool SUMBAWA
```

- Mengatur alamat **network** dan **subnetmask** dari alamat IP yang akan disewakan ke **DHCP Client** yaitu **192.168.2.0/24**.

```
RTR_SUMBAWA(dhcp-config)#network 192.168.2.0 255.255.255.0
```

- Mengatur **default gateway** yang diperoleh **DHCP Client** yaitu **192.168.2.1**.

```
RTR_SUMBAWA(dhcp-config)#default-router 192.168.2.1
```

- Mengatur alamat **server DNS** yang diperoleh **DHCP client** yaitu **192.168.1.253**.

```
RTR_SUMBAWA(dhcp-config)#dns-server 192.168.1.253
```

- Berpindah ke satu mode sebelumnya.

```
RTR_SUMBAWA(dhcp-config)#exit  
RTR_SUMBAWA(config)#
```

MENGATUR SERVER DHCP

- Mengatur agar alamat IP yang tidak disewakan ke **DHCP Client** oleh **Server DHCP** yaitu dari **192.168.2.1** sampai dengan **192.168.2.2**.

```
RTR_SUMBAWA(config)#ip dhcp excluded-address 192.168.2.1 192.168.2.2
```

MENGATUR ROUTING PROTOCOL EIGRP

- Mengaktifkan **routing protocol EIGRP** dengan **Autonomous System (AS) Number 64512**.

```
RTR_SUMBAWA(config)#router eigrp 64512
```

- Mengatur alamat jaringan dimana **router RTR_SUMBAWA** terhubung secara langsung yang dituliskan secara **classfull** yaitu **network 192.168.2.0** yang merupakan **LAN SUMBAWA**.

```
RTR_SUMBAWA(config-router)#network 192.168.2.0
```

- Mengatur alamat jaringan dimana **router RTR_SUMBAWA** terhubung secara langsung yang dituliskan secara **classfull** yaitu **network 192.168.0.0** yang merupakan **Subnet Point-to-Point MATARAM-SUMBAWA** dan **Subnet Point-to-Point SUMBAWA-BIMA**.

```
RTR_SUMBAWA(config-router)#network 192.168.0.0
```

- Berpindah ke mode **privilege**.

```
RTR_SUMBAWA(config-router)#end  
RTR_SUMBAWA#
```


MEMVERIFIKASI ROUTING PROTOCOL YANG AKTIF

- Memverifikasi *routing protocol* yang aktif pada **router**.

```
RTR_SUMBAWA#show ip protocols
```

```
Routing Protocol is "eigrp 64512"
```

```
  Outgoing update filter list for all interfaces is not set
```

```
  Incoming update filter list for all interfaces is not set
```

```
  Default networks flagged in outgoing updates
```

```
  Default networks accepted from incoming updates
```

```
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
```

```
  EIGRP maximum hopcount 100
```

```
  EIGRP maximum metric variance 1
```

```
Redistributing: eigrp 64512
```

```
  Automatic network summarization is in effect
```

```
  Automatic address summarization:
```

```
    192.168.0.0/24 for GigabitEthernet0/0/0
```

```
      Summarizing with metric 2169856
```

```
Maximum path: 4
```

```
Routing for Networks:
```

```
  192.168.2.0
```

```
  192.168.0.0
```

```
Routing Information Sources:
```

```
  Gateway         Distance      Last Update
```

```
  192.168.0.1     90           1571512
```

```
Distance: internal 90 external 170
```

MEMVERIFIKASI DAN MENYIMPAN KONFIGURASI

- Menampilkan informasi konfigurasi yang sedang berjalan atau aktif di **Random Access Memory (RAM)**.

```
RTR_SUMBAWA#show run
```

Tampil pesan `--More--` di bagian pojok kiri bawah dari *output* eksekusi perintah tersebut.

Tekan **spasi** untuk menampilkan *output* secara lengkap atau tekan **Enter** untuk menampilkan per baris. Untuk keluar dari *output* tersebut tekan sembarang tombol, sebagai contoh **q**.

- Menyimpan konfigurasi secara permanen dari **RAM** ke **Non Volatile Random Access Memory (NVRAM)**.

```
RTR_SUMBAWA#copy run start  
Destination filename [startup-config]?  
Building configuration...  
[OK]
```

Tekan **Enter** pada pesan konfirmasi “**Destination filename [startup-config]?**” yang menanyakan tentang nama *file* sebagai tujuan penyimpanan konfigurasi.

MEMVERIFIKASI DAN MENYIMPAN KONFIGURASI

- Menampilkan informasi konfigurasi tersimpan di NVRAM.

```
RTR_SUMBAWA#show start
```

Tampil pesan `--More--` di bagian pojok kiri bawah dari *output* eksekusi perintah tersebut.

Tekan **spasi** untuk menampilkan *output* secara lengkap atau tekan **Enter** untuk menampilkan per baris.

Untuk keluar dari *output* tersebut tekan sembarang tombol, sebagai contoh **q**.

KONFIGURASI DASAR DI MULTILAYER SWITCH MSW_SUMBAWA

1. Mengatur Hostname.
2. Mengatur Pengalamatan IP pada Interface dan Default Gateway.
3. Mengatur Password Privilege Mode.
4. Mengatur Secure Shell (SSH) Server.



MENGAKSES COMMAND LINE INTERFACE (CLI) DARI MSW_MATARAM

- Melalui **Logical Workspace** dari **Cisco Packet Tracer**, klik pada **Multilayer Switch MSW_SUMBAWA**.
- Pada kotak dialog *properties* dari **MSW_SUMBAWA** yang tampil, klik tab **CLI**.
- Apabila tampil **System Configuration Dialog** maka ketik **no** pada inputan “*Would you like to enter the initial configuration dialog? [yes/no].*” dan tekan **Enter**.
- Tekan **Enter** untuk memunculkan **prompt CLI** Switch>.

```
MSW_SUMBAWA
Physical Config CLI Attributes
IOS Command Line Interface
Base ethernet MAC Address : 00:01:64:3A:9C:E4
Motherboard assembly number : 73-15899-06
Motherboard serial number : FDO20311WHP
Model revision number : N0
Motherboard revision number : A0
Model number : WS-C3650-24PS
System serial number : FDO2031Q0TD

--- System Configuration Dialog ---
Would you like to enter the initial configuration dialog? [yes/no]:
%LINK-5-CHANGED: Interface GigabitEthernet1/0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0/1, changed state to up
%LINK-5-CHANGED: Interface GigabitEthernet1/0/2, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0/2, changed state to up
%LINK-5-CHANGED: Interface GigabitEthernet1/0/3, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0/3, changed state to up
% Please answer 'yes' or 'no'.
Would you like to enter the initial configuration dialog? [yes/no]: no
Press RETURN to get started!
Switch>
```

MENGATUR HOSTNAME

- Berpindah dari mode **user** ke mode **privilege**.

```
Switch>enable
```

- Berpindah dari mode **privilege** ke mode **global configuration**.

```
Switch#conf t
```

- Mengatur **hostname** dari *multilayer switch* menggunakan **MSW_SUMBAWA**.

```
Switch(config)#hostname MSW_SUMBAWA  
MSW_SUMBAWA(config)#
```

- Terlihat **hostname** telah berubah dari **Switch** menjadi **MSW_SUMBAWA**.

MENGATUR PENGALAMATAN IP PADA INTERFACE DAN DEFAULT GATEWAY

- Berpindah ke **interface configuration** dari **VLAN 1**.

```
MSW_SUMBAWA(config)#int vlan 1
```

- Mengatur pengalamatan IP pada **interface VLAN 1** dengan nilai **192.168.2.2/24**.

```
MSW_SUMBAWA(config-if)#ip address 192.168.2.2 255.255.255.0
```

- Mengaktifkan **interface**.

```
MSW_SUMBAWA(config-if)#no shutdown
```

- Berpindah ke satu mode sebelumnya.

```
MSW_SUMBAWA(config-if)#exit
```

- Mengatur **default gateway** agar *multilayer switch* dapat berkomunikasi ke beda jaringan dengan nilai **192.168.2.1**.

```
MSW_SUMBAWA(config)#ip default-gateway 192.168.2.1
```

MEMVERIFIKASI PENGATURAN PENGALAMATAN IP PADA INTERFACE

- Memverifikasi pengalamatan IP pada **interface vlan 1**.

```
MSW SUMBAWA(config)#do show int vlan 1
Vlan1 is up, line protocol is up
Hardware is CPU Interface, address is 0001.643a.9ce4 (bia 0001.643a.9ce4)
Internet address is 192.168.2.2/24
MTU 1500 bytes, BW 100000 Kbit, DLY 1000000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
ARP type: ARPA, ARP Timeout 04:00:00
Last input 21:40:21, output never, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
 1682 packets input, 530955 bytes, 0 no buffer
  Received 0 broadcasts (0 IP multicast)
   0 runts, 0 giants, 0 throttles
   0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
 563859 packets output, 0 bytes, 0 underruns
   0 output errors, 23 interface resets
   0 output buffer failures, 0 output buffers swapped out
```

- Terlihat status **interface vlan 1** telah aktif (**up**) dan telah menggunakan alamat IP **192.168.2.2/24**.

MEMVERIFIKASI PENGATURAN DEFAULT GATEWAY

- Eksekusi perintah `do show run | include ip default-gateway` untuk memverifikasi hasil pengaturan **default gateway** pada *Multilayer Switch*.

```
MSW_SUMBAWA(config)#do show run | include ip default-gateway  
ip default-gateway 192.168.2.1
```

- Terlihat telah menggunakan IP **192.168.2.1** sebagai *default gateway*.

MENGATUR PASSWORD PRIVILEGE MODE DAN SECURE SHELL (SSH) SERVER

- Mengatur **password privilege mode** dengan sandi “**menanti**”.

```
MSW_SUMBAWA(config)#enable secret menanti
```

- Mengatur nama **domain** dengan nilai “**sabarmenanti.com**”.

```
MSW_SUMBAWA(config)#ip domain-name sabarmenanti.com
```

- Membuat **RSA key** dengan **modulus 1024**.

```
MSW_SUMBAWA(config)#crypto key generate rsa general-keys modulus 1024
% You already have RSA keys defined named MSW_SUMBAWA.sabarmenanti.com
% They will be replaced.
```

```
% The key modulus size is 1024 bits
```

```
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
```

```
*Mar 1 0:8:15.552: %SSH-5-ENABLED: SSH 1.99 has been enabled
```

MENGATUR SECURE SHELL (SSH) SERVER

- Membuat akun otentikasi pengguna yang mengakses melalui SSH dengan nama login “**sabar**” dan sandi login “**menanti**” yang disimpan pada **local database**.

```
MSW_SUMBAWA(config)#username sabar password menanti
```

- Mengaktifkan SSH pada **line virtual teletype (vty)** untuk 5 (lima) pengguna.

```
MSW_SUMBAWA(config)#line vty 0 4
```

- Mengotentikasi keseluruhan sesi **incoming virtual terminal** menggunakan **local username database**.

```
MSW_SUMBAWA(config-line)#login local
```

- Mengatur **SSH** sebagai protokol yang dapat digunakan ketika terkoneksi melalui **line vty**.

```
MSW_SUMBAWA(config-line)#transport input ssh
```

- Berpindah ke mode **privilege**.

```
MSW_SUMBAWA(config-line)#end
```

```
MSW_SUMBAWA#
```

MEMVERIFIKASI DAN MENYIMPAN KONFIGURASI

- Menampilkan informasi konfigurasi yang sedang berjalan atau aktif di **Random Access Memory (RAM)**.

```
MSW_SUMBAWA#show run
```

Tampil pesan `--More--` di bagian pojok kiri bawah dari *output* eksekusi perintah tersebut.

Tekan **spasi** untuk menampilkan *output* secara lengkap atau tekan **Enter** untuk menampilkan per baris. Untuk keluar dari *output* tersebut tekan sembarang tombol, sebagai contoh **q**.

- Menyimpan konfigurasi secara permanen dari **RAM** ke **Non Volatile Random Access Memory (NVRAM)**.

```
MSW_SUMBAWA#copy run start
Destination filename [startup-config]? ↵
Building configuration...
[OK]
```

Tekan **Enter** pada pesan konfirmasi “**Destination filename [startup-config]?**” yang menanyakan tentang nama *file* sebagai tujuan penyimpanan konfigurasi.

MEMVERIFIKASI DAN MENYIMPAN KONFIGURASI

- Menampilkan informasi konfigurasi tersimpan di NVRAM.

```
MSW_SUMBAWA#show start
```

Tampil pesan `--More--` di bagian pojok kiri bawah dari *output* eksekusi perintah tersebut.

Tekan **spasi** untuk menampilkan *output* secara lengkap atau tekan **Enter** untuk menampilkan per baris.

Untuk keluar dari *output* tersebut tekan sembarang tombol, sebagai contoh **q**.

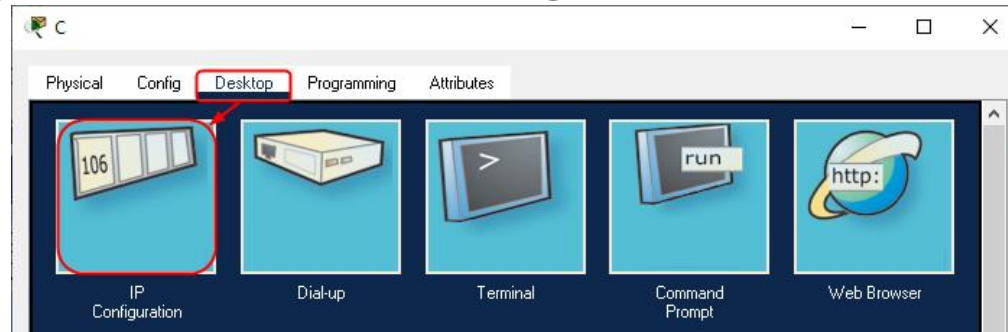
KONFIGURASI DI PC CLIENT LAN SUMBAWA

Mengatur alokasi pengalamatan IP secara dinamis yaitu sebagai **DHCP Client** pada **PC C** dan **PC D**.

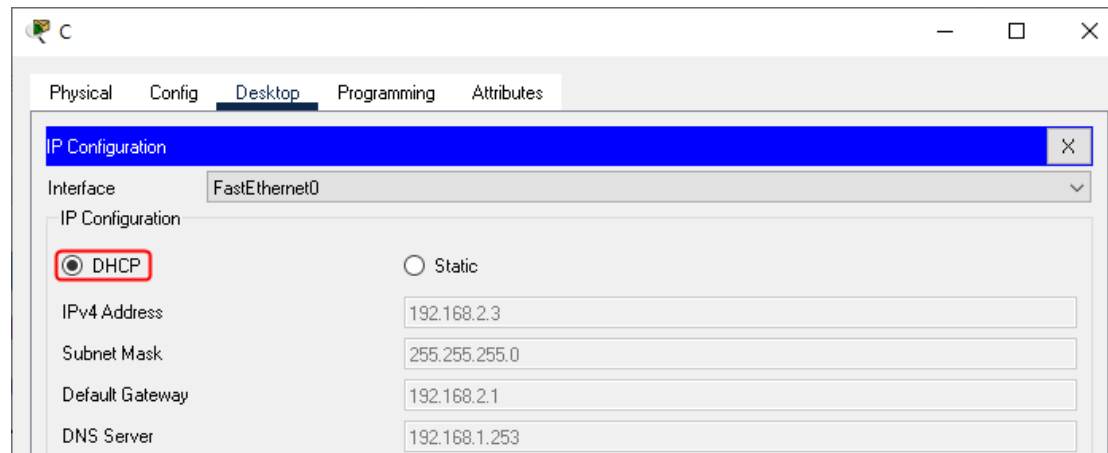


KONFIGURASI DHCP CLIENT PADA PC C DI LAN SUMBAWA

- Melalui **Logical Workspace** dari **Cisco Packet Tracer** klik pada **PC C**. Pilih tab **Desktop** pada kotak dialog yang tampil dan pilih **IP Configuration**.



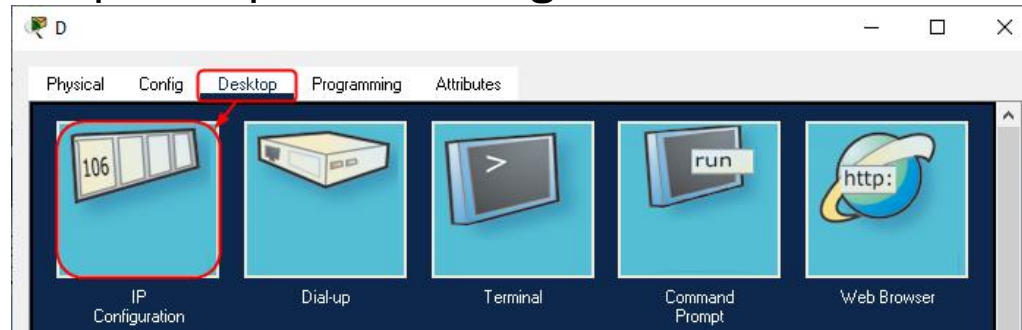
- Tampil kotak dialog **IP Configuration**, pilih **DHCP** untuk mengalokasikan pengalamatan IP secara dinamis.



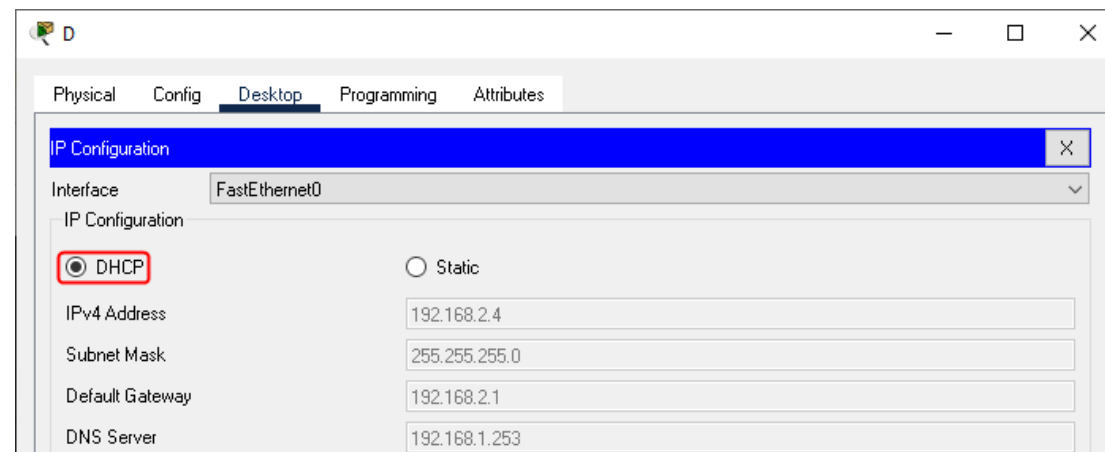
Terlihat **PC C** telah berhasil memperoleh pengalamatan IP dan parameter TCP/IP lainnya dari **Server DHCP**. Tutup kotak dialog **PC C**.

KONFIGURASI DHCP CLIENT PADA PC D DI LAN SUMBAWA

- Melalui **Logical Workspace** dari **Cisco Packet Tracer** klik pada **PC D**. Pilih tab **Desktop** pada kotak dialog yang tampil dan pilih **IP Configuration**.



- Tampil kotak dialog **IP Configuration**, pilih **DHCP** untuk mengalokasikan pengalamatan IP secara dinamis.



Terlihat **PC D** telah berhasil memperoleh pengalamatan IP dan parameter TCP/IP lainnya dari **Server DHCP**. Tutup kotak dialog **PC D**.

KONFIGURASI PERANGKAT JARINGAN

DI KANTOR CABANG BIMA

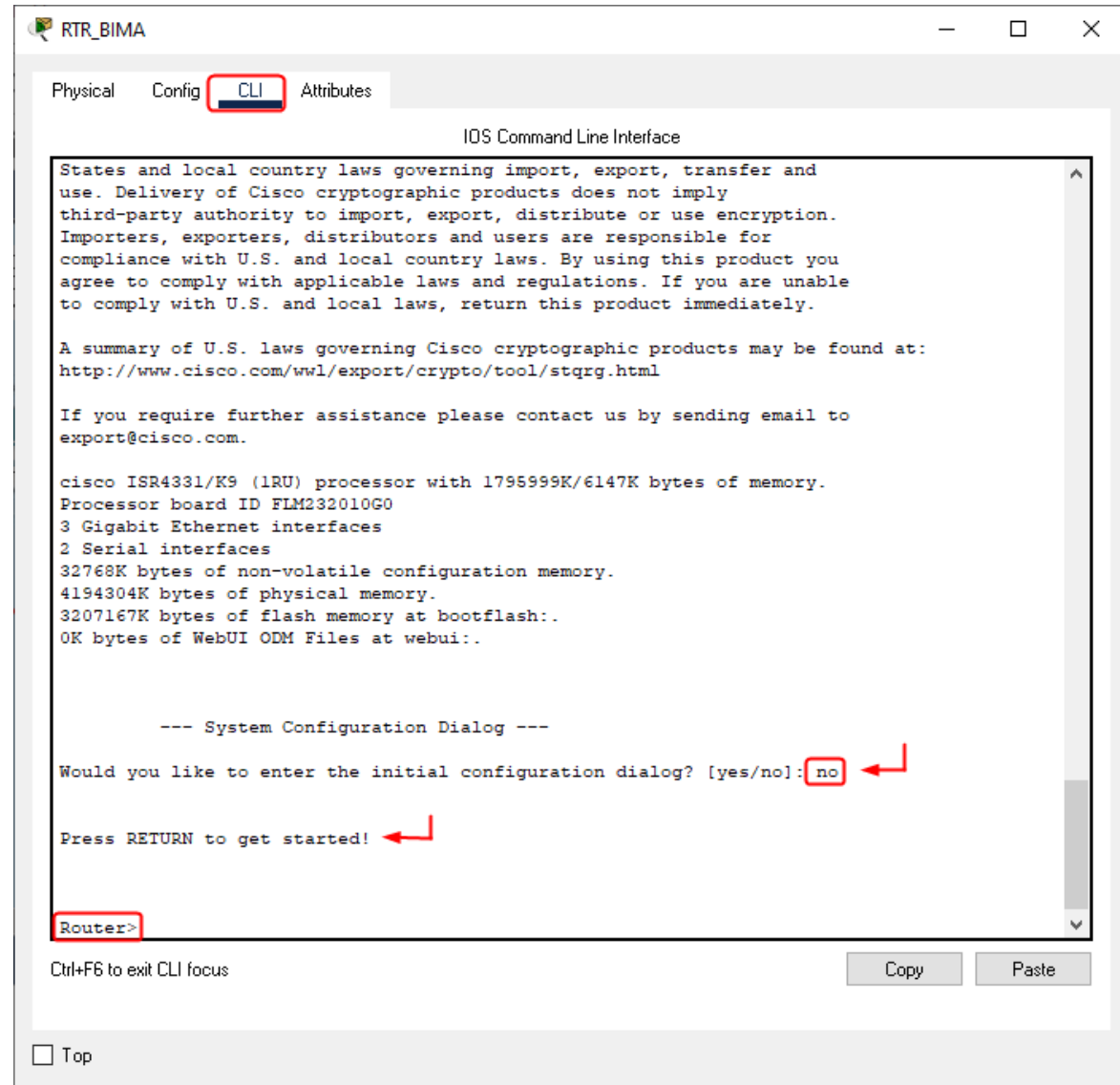


KONFIGURASI DASAR DI ROUTER RTR_BIMA

1. Mengatur hostname.
2. Mengatur pengalamatan IP pada interface.
3. Mengatur password privilege mode.
4. Mengatur Secure Shell (SSH) Server.
5. Mengatur Server Dynamic Host Configuration Protocol (DHCP).
6. Mengatur routing protocol Enhanced Interior Gateway Routing Protocol (EIGRP).

MENGAKSES COMMAND LINE INTERFACE (CLI) DARI ROUTER RTR_BIMA

- Melalui **Logical Workspace** dari **Cisco Packet Tracer**, klik pada **router RTR_BIMA**.
- Pada kotak dialog *properties* dari **RTR_BIMA** yang tampil, klik tab **CLI**.
- Apabila tampil **System Configuration Dialog** maka ketik **no** pada inputan "*Would you like to enter the initial configuration dialog? [yes/no]:*" dan tekan **Enter**.
- Tekan **Enter** untuk memunculkan **prompt CLI Router>**.



MENGATUR HOSTNAME

- Berpindah dari mode **user** ke mode **privilege**.
`Router>enable`
- Berpindah dari mode **privilege** ke mode **global configuration**.
`Router#conf t`
- Mengatur **hostname** dari *router* menggunakan **RTR_BIMA**.
`Router(config)#hostname RTR_BIMA`
`RTR_BIMA(config)#`
- Terlihat **hostname** telah berubah dari **Router** menjadi **RTR_BIMA**.

MENGATUR PENGALAMATAN IP PADA INTERFACE

- Berpindah ke **interface configuration** dari **GigabitEthernet0/0/0** atau disingkat **g0/0/0**.
`RTR_BIMA(config)#int g0/0/0`
- Mengatur pengalamatan IP pada **interface g0/0/0** dengan nilai **192.168.3.1/24**.
`RTR_BIMA(config-if)#ip address 192.168.3.1 255.255.255.0`
- Mengaktifkan **interface**.
`RTR_BIMA(config-if)#no shutdown`
- Berpindah ke **interface configuration** dari **Serial0/1/1** atau disingkat **s0/1/1** dengan nilai **192.168.0.6/30**.
`RTR_BIMA(config-if)#int s0/1/1`
- Mengatur pengalamatan IP pada **interface s0/1/1**.
`RTR_BIMA(config-if)#ip address 192.168.0.6 255.255.255.252`
- Mengaktifkan **interface**.
`RTR_BIMA(config-if)#no shutdown`

MEMVERIFIKASI PENGATURAN PENGALAMATAN IP PADA INTERFACE

- Memverifikasi pengalamatan IP pada **interface g0/0/0** dan **s0/1/1** secara ringkas.

```
RTR_BIMA(config-if) #do show ip int brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
GigabitEthernet0/0/0	192.168.3.1	YES	manual	up	up
GigabitEthernet0/0/1	unassigned	YES	NVRAM	administratively down	down
GigabitEthernet0/0/2	unassigned	YES	NVRAM	administratively down	down
Serial0/1/0	unassigned	YES	NVRAM	administratively down	down
Serial0/1/1	192.168.0.6	YES	manual	up	up
Vlan1	unassigned	YES	NVRAM	administratively down	down

- Terlihat **interface GigabitEthernet0/0/0** telah menggunakan alamat **IP 192.168.3.1**. Sedangkan **interface Serial0/1/1** juga telah menggunakan alamat **IP 192.168.0.6**.

MENGATUR PASSWORD PRIVILEGE MODE DAN SECURE SHELL (SSH) SERVER

- Berpindah ke satu mode sebelumnya yaitu dari **interface configuration** ke **global configuration mode**.

```
RTR_BIMA(config-if)#exit
```

- Mengatur **password privilege mode** dengan sandi “**menanti**”.

```
RTR_BIMA(config)#enable secret menanti
```

- Mengatur nama **domain** dengan nilai “**sabarmenanti.com**”.

```
RTR_BIMA(config)#ip domain-name sabarmenanti.com
```

- Membuat **RSA key** dengan **modulus 1024**.

```
RTR_BIMA(config)#crypto key generate rsa general-keys modulus 1024
% You already have RSA keys defined named RTR_BIMA.sabarmenanti.com
% They will be replaced.
```

```
% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
*Mar 1 0:7:49.503: %SSH-5-ENABLED: SSH 1.99 has been enabled
```

MENGATUR SECURE SHELL (SSH) SERVER

- Membuat akun otentikasi pengguna yang mengakses melalui SSH dengan nama login “**sabar**” dan sandi login “**menanti**” yang disimpan pada **local database**.

```
RTR_BIMA(config)#username sabar password menanti
```

- Mengaktifkan SSH pada **line virtual teletype (vty)** untuk 5 (lima) pengguna.

```
RTR_BIMA(config)#line vty 0 4
```

- Mengotentikasi keseluruhan sesi **incoming virtual terminal** menggunakan **local username database**.

```
RTR_BIMA(config-line)#login local
```

- Mengatur **SSH** sebagai protokol yang dapat digunakan ketika terkoneksi melalui **line vty**.

```
RTR_BIMA(config-line)#transport input ssh
```

- Berpindah ke mode satu mode sebelumnya.

```
RTR_BIMA(config-line)#exit  
RTR_BIMA(config)#
```


MENGATUR SERVER DHCP

- Membuat ruang alamat IP (**pool**) yang disewakan ke **DHCP Client** dengan nama **BIMA**.
- Mengatur alamat **network** dan **subnetmask** dari alamat IP yang akan disewakan ke **DHCP Client** yaitu **192.168.3.0/24**.

```
RTR_BIMA(config)#ip dhcp pool BIMA
```

```
RTR_BIMA(dhcp-config)#network 192.168.3.0 255.255.255.0
```

- Mengatur **default gateway** yang diperoleh **DHCP Client** yaitu **192.168.3.1**.

```
RTR_BIMA(dhcp-config)#default-router 192.168.3.1
```

- Mengatur alamat **server DNS** yang diperoleh **DHCP client** yaitu **192.168.1.253**.

```
RTR_BIMA(dhcp-config)#dns-server 192.168.1.253
```

- Berpindah ke satu mode sebelumnya.

```
RTR_BIMA(dhcp-config)#exit
```

```
RTR_BIMA(config)#
```

MENGATUR SERVER DHCP

- Mengatur agar alamat IP yang tidak disewakan ke **DHCP Client** oleh **Server DHCP** yaitu dari **192.168.3.1** sampai dengan **192.168.3.2**.

```
RTR_BIMA(config)#ip dhcp excluded-address 192.168.3.1 192.168.3.2
```

MENGATUR ROUTING PROTOCOL EIGRP

- Mengaktifkan **routing protocol EIGRP** dengan **Autonomous System (AS) Number 64512**.

```
RTR_BIMA(config)#router eigrp 64512
```

- Mengatur alamat jaringan dimana **router RTR_BIMA** terhubung secara langsung yang dituliskan secara **classfull** yaitu **network 192.168.3.0** yang merupakan **LAN BIMA**.

```
RTR_BIMA(config-router)#network 192.168.3.0
```

- Mengatur alamat jaringan dimana **router RTR_BIMA** terhubung secara langsung yang dituliskan secara **classfull** yaitu **network 192.168.0.0** yang merupakan **Subnet Point-to-Point SUMBAWA-BIMA**.

```
RTR_BIMA(config-router)#network 192.168.0.0
```

- Berpindah ke mode **privilege**.

```
RTR_BIMA(config-router)#end  
RTR_BIMA#
```

MEMVERIFIKASI ROUTING PROTOCOL YANG AKTIF

- Memverifikasi *routing protocol* yang aktif pada **router**.

```
RTR_BIMA#show ip protocols
```

```
Routing Protocol is "eigrp 64512"
```

```
  Outgoing update filter list for all interfaces is not set
```

```
  Incoming update filter list for all interfaces is not set
```

```
  Default networks flagged in outgoing updates
```

```
  Default networks accepted from incoming updates
```

```
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
```

```
  EIGRP maximum hopcount 100
```

```
  EIGRP maximum metric variance 1
```

```
Redistributing: eigrp 64512
```

```
  Automatic network summarization is in effect
```

```
  Automatic address summarization:
```

```
    192.168.0.0/24 for GigabitEthernet0/0/0
```

```
      Summarizing with metric 2169856
```

```
Maximum path: 4
```

```
Routing for Networks:
```

```
  192.168.3.0
```

```
  192.168.0.0
```

```
Routing Information Sources:
```

```
  Gateway         Distance      Last Update
```

```
  192.168.0.5     90           960568
```

```
Distance: internal 90 external 170
```

MEMVERIFIKASI DAN MENYIMPAN KONFIGURASI

- Menampilkan informasi konfigurasi yang sedang berjalan atau aktif di **Random Access Memory (RAM)**.

```
RTR_BIMA#show run
```

Tampil pesan `--More--` di bagian pojok kiri bawah dari *output* eksekusi perintah tersebut.

Tekan **spasi** untuk menampilkan *output* secara lengkap atau tekan **Enter** untuk menampilkan per baris. Untuk keluar dari *output* tersebut tekan sembarang tombol, sebagai contoh **q**.

- Menyimpan konfigurasi secara permanen dari **RAM** ke **Non Volatile Random Access Memory (NVRAM)**.

```
RTR_BIMA#copy run start  
Destination filename [startup-config]? ↵  
Building configuration...  
[OK]
```

Tekan **Enter** pada pesan konfirmasi “**Destination filename [startup-config]?**” yang menanyakan tentang nama *file* sebagai tujuan penyimpanan konfigurasi.

MEMVERIFIKASI DAN MENYIMPAN KONFIGURASI

- Menampilkan informasi konfigurasi tersimpan di NVRAM.

```
RTR_BIMA#show start
```

Tampil pesan `--More--` di bagian pojok kiri bawah dari *output* eksekusi perintah tersebut.

Tekan **spasi** untuk menampilkan *output* secara lengkap atau tekan **Enter** untuk menampilkan per baris.

Untuk keluar dari *output* tersebut tekan sembarang tombol, sebagai contoh **q**.

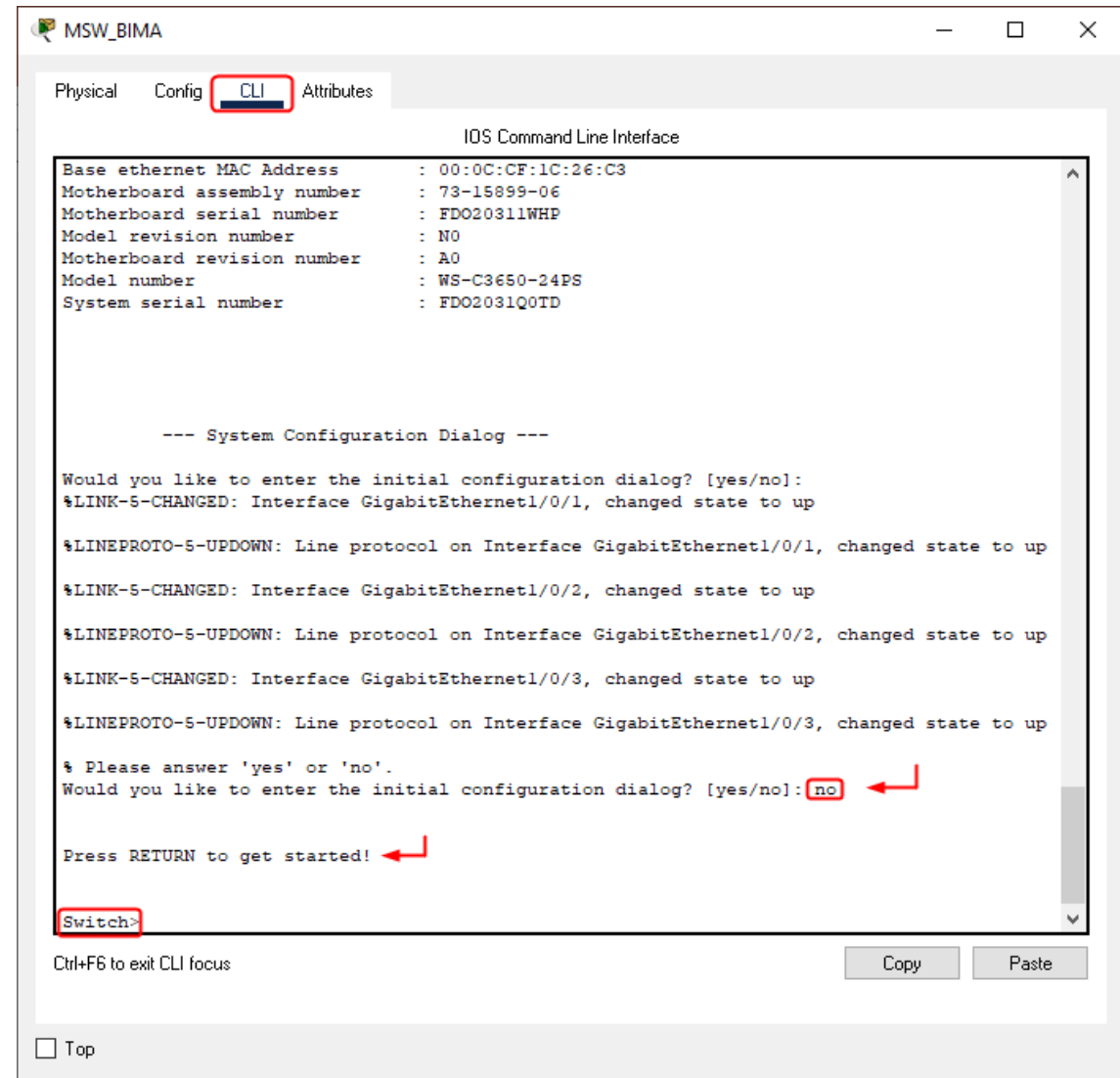
KONFIGURASI DASAR DI MULTILAYER SWITCH MSW_BIMA

1. Mengatur Hostname.
2. Mengatur Pengalamatan IP pada Interface dan Default Gateway.
3. Mengatur Password Privilege Mode.
4. Mengatur Secure Shell (SSH) Server.



MENGAKSES COMMAND LINE INTERFACE (CLI) DARI MSW_BIMA

- Melalui **Logical Workspace** dari **Cisco Packet Tracer**, klik pada **Multilayer Switch MSW_BIMA**.
- Pada kotak dialog *properties* dari **MSW_BIMA** yang tampil, klik tab **CLI**.
- Apabila tampil **System Configuration Dialog** maka ketik **no** pada inputan “*Would you like to enter the initial configuration dialog? [yes/no]:*” dan tekan **Enter**.
- Tekan **Enter** untuk memunculkan **prompt CLI Switch>**.



MENGATUR HOSTNAME

- Berpindah dari mode **user** ke mode **privilege**.
`Switch>enable`
- Berpindah dari mode **privilege** ke mode **global configuration**.
`Switch#conf t`
- Mengatur **hostname** dari *multilayer switch* menggunakan **MSW_BIMA**.
`Switch(config)#hostname MSW_BIMA`
`MSW_BIMA(config)#`
- Terlihat **hostname** telah berubah dari **Switch** menjadi **MSW_BIMA**.

MENGATUR PENGALAMATAN IP PADA INTERFACE DAN DEFAULT GATEWAY

- Berpindah ke **interface configuration** dari **VLAN 1**.

```
MSW_BIMA(config)#int vlan 1
```

- Mengatur pengalamatan IP pada **interface VLAN 1** dengan nilai **192.168.3.2/24**.

```
MSW_BIMA(config-if)#ip address 192.168.3.2 255.255.255.0
```

- Mengaktifkan **interface**.

```
MSW_BIMA(config-if)#no shutdown
```

- Berpindah ke satu mode sebelumnya.

```
MSW_BIMA(config-if)#exit
```

- Mengatur **default gateway** agar *multilayer switch* dapat berkomunikasi ke beda jaringan dengan nilai **192.168.3.1**.

```
MSW_BIMA(config)#ip default-gateway 192.168.3.1
```

MEMVERIFIKASI PENGATURAN PENGALAMATAN IP PADA INTERFACE

- Memverifikasi pengalamatan IP pada **interface vlan 1**.

```
MSW_BIMA(config)#do show int vlan 1
Vlan1 is up, line protocol is up
Hardware is CPU Interface, address is 000c.cf1c.26c3 (bia 000c.cf1c.26c3)
Internet address is 192.168.3.2/24
MTU 1500 bytes, BW 100000 Kbit, DLY 1000000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
ARP type: ARPA, ARP Timeout 04:00:00
Last input 21:40:21, output never, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
 1682 packets input, 530955 bytes, 0 no buffer
  Received 0 broadcasts (0 IP multicast)
   0 runts, 0 giants, 0 throttles
   0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
 563859 packets output, 0 bytes, 0 underruns
   0 output errors, 23 interface resets
   0 output buffer failures, 0 output buffers swapped out
```

- Terlihat status **interface vlan 1** telah aktif (**up**) dan telah menggunakan alamat IP **192.168.3.2/24**.

MEMVERIFIKASI PENGATURAN DEFAULT GATEWAY

- Eksekusi perintah `do show run | include ip default-gateway` untuk memverifikasi hasil pengaturan **default gateway** pada *Multilayer Switch*.

```
MSW_BIMA(config)#do show run | include ip default-gateway  
ip default-gateway 192.168.3.1
```

- Terlihat telah menggunakan IP **192.168.3.1** sebagai *default gateway*.

MENGATUR PASSWORD PRIVILEGE MODE DAN SECURE SHELL (SSH) SERVER

- Mengatur **password privilege mode** dengan sandi “**menanti**”.

```
MSW_BIMA(config)#enable secret menanti
```

- Mengatur nama **domain** dengan nilai “**sabarmenanti.com**”.

```
MSW_BIMA(config)#ip domain-name sabarmenanti.com
```

- Membuat **RSA key** dengan **modulus 1024**.

```
MSW_BIMA(config)#crypto key generate rsa general-keys modulus 1024
% You already have RSA keys defined named MSW_BIMA.sabarmenanti.com
% They will be replaced.
```

```
% The key modulus size is 1024 bits
```

```
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
```

```
*Mar 1 0:9:12.757: %SSH-5-ENABLED: SSH 1.99 has been enabled
```

MENGATUR SECURE SHELL (SSH) SERVER

- Membuat akun otentikasi pengguna yang mengakses melalui SSH dengan nama login “**sabar**” dan sandi login “**menanti**” yang disimpan pada **local database**.

```
MSW_BIMA(config)#username sabar password menanti
```

- Mengaktifkan SSH pada **line virtual teletype (vty)** untuk 5 (lima) pengguna.

```
MSW_BIMA(config)#line vty 0 4
```

- Mengotentikasi keseluruhan sesi **incoming virtual terminal** menggunakan **local username database**.

```
MSW_BIMA(config-line)#login local
```

- Mengatur **SSH** sebagai protokol yang dapat digunakan ketika terkoneksi melalui **line vty**.

```
MSW_BIMA(config-line)#transport input ssh
```

- Berpindah ke mode **privilege**.

```
MSW_BIMA(config-line)#end  
MSW_BIMA#
```

MEMVERIFIKASI DAN MENYIMPAN KONFIGURASI

- Menampilkan informasi konfigurasi yang sedang berjalan atau aktif di **Random Access Memory (RAM)**.

```
MSW_BIMA#show run
```

Tampil pesan `--More--` di bagian pojok kiri bawah dari *output* eksekusi perintah tersebut.

Tekan **spasi** untuk menampilkan *output* secara lengkap atau tekan **Enter** untuk menampilkan per baris. Untuk keluar dari *output* tersebut tekan sembarang tombol, sebagai contoh **q**.

- Menyimpan konfigurasi secara permanen dari **RAM** ke **Non Volatile Random Access Memory (NVRAM)**.

```
MSW_BIMA#copy run start
Destination filename [startup-config]? ↵
Building configuration...
[OK]
```

Tekan **Enter** pada pesan konfirmasi “**Destination filename [startup-config]?**” yang menanyakan tentang nama *file* sebagai tujuan penyimpanan konfigurasi.

MEMVERIFIKASI DAN MENYIMPAN KONFIGURASI

- Menampilkan informasi konfigurasi tersimpan di NVRAM.

```
MSW_BIMA#show start
```

Tampil pesan `--More--` di bagian pojok kiri bawah dari *output* eksekusi perintah tersebut.

Tekan **spasi** untuk menampilkan *output* secara lengkap atau tekan **Enter** untuk menampilkan per baris.

Untuk keluar dari *output* tersebut tekan sembarang tombol, sebagai contoh **q**.

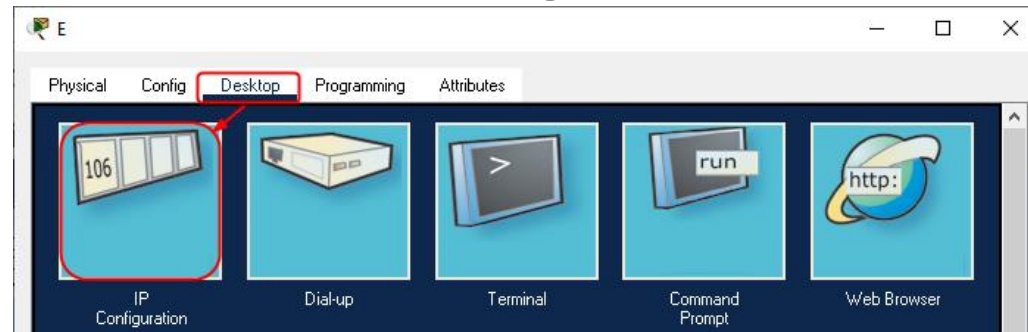
KONFIGURASI DI PC CLIENT LAN BIMA

Mengatur alokasi pengalamatan IP secara dinamis yaitu sebagai **DHCP Client** pada **PC E** dan **PC F**.

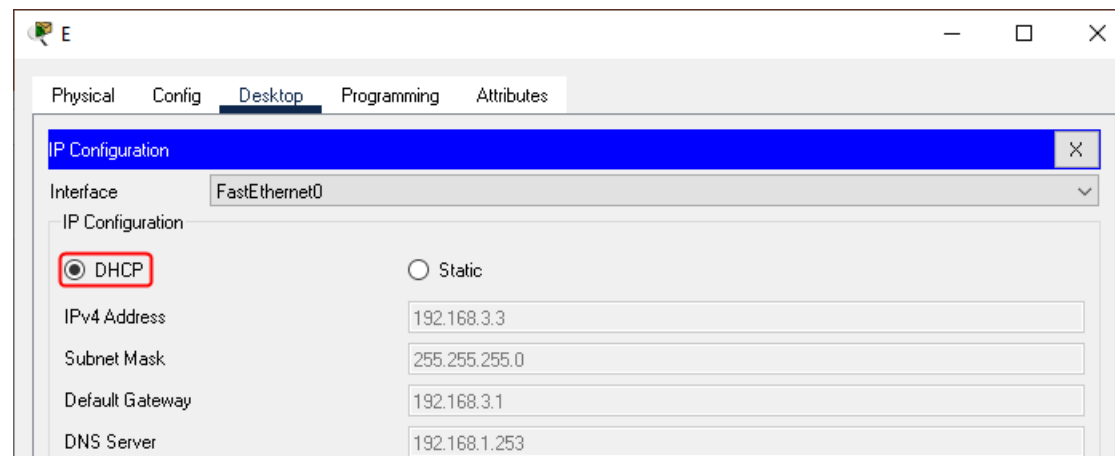


KONFIGURASI DHCP CLIENT PADA PC E DI LAN BIMA

- Melalui **Logical Workspace** dari **Cisco Packet Tracer** klik pada **PC E**. Pilih tab **Desktop** pada kotak dialog yang tampil dan pilih **IP Configuration**.



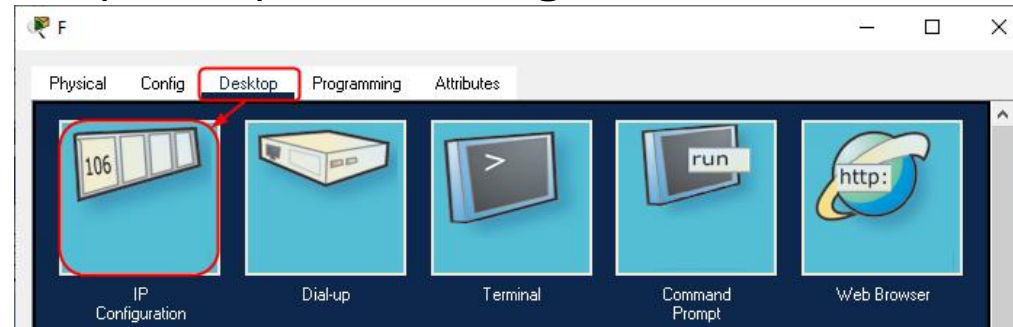
- Tampil kotak dialog **IP Configuration**, pilih **DHCP** untuk mengalokasikan pengalamatan IP secara dinamis.



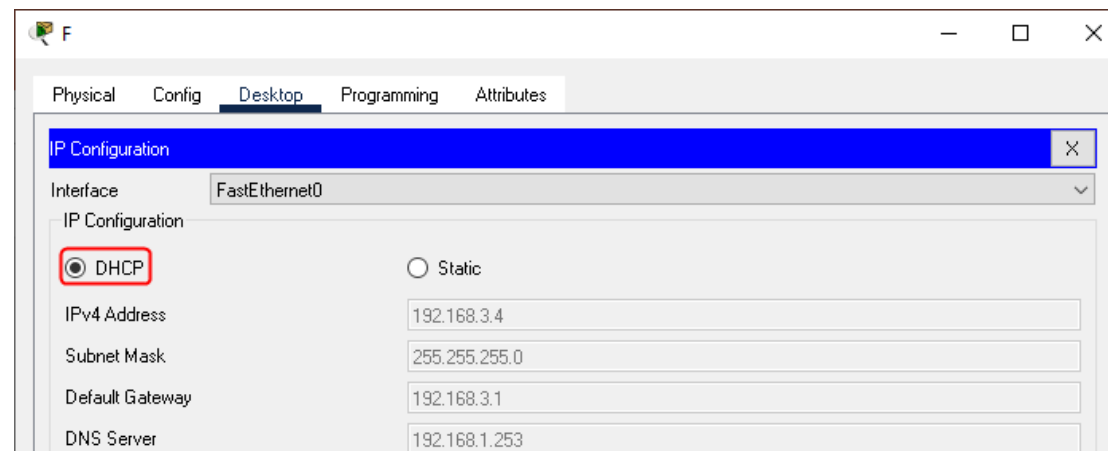
Terlihat **PC E** telah berhasil memperoleh pengalamatan IP dan parameter TCP/IP lainnya dari **Server DHCP**. Tutup kotak dialog **PC E**.

KONFIGURASI DHCP CLIENT PADA PC F DI LAN BIMA

- Melalui **Logical Workspace** dari **Cisco Packet Tracer** klik pada **PC F**. Pilih tab **Desktop** pada kotak dialog yang tampil dan pilih **IP Configuration**.



- Tampil kotak dialog **IP Configuration**, pilih **DHCP** untuk mengalokasikan pengalamatan IP secara dinamis.



Terlihat **PC F** telah berhasil memperoleh pengalamatan IP dan parameter TCP/IP lainnya dari **Server DHCP**. Tutup kotak dialog **PC F**.

VERIFIKASI TABEL ROUTING DAN KONEKSI ANTAR HOSTS



VERIFIKASI TABLE ROUTING DI RTR_MATARAM

- Melalui **Logical Workspace** dari **Cisco Packet Tracer**, klik pada **router RTR_MATARAM**. Pada kotak dialog *properties* dari **RTR_MATARAM** yang tampil, klik tab **CLI** dan tekan **Enter** untuk memunculkan **prompt CLI**.
- Berpindah dari mode **user** ke mode **privilege**.

```
RTR_MATARAM>enable  
Password:
```

- Tampil inputan inputan **Password:**, masukkan sandi dari **privilege mode** yaitu "**menanti**".
- Eksekusi perintah `show ip route` untuk menampilkan informasi tabel *routing* pada **router RTR_MATARAM**.

VERIFIKASI TABLE ROUTING DI RTR_MATARAM

- Hasil eksekusi dari perintah `show ip route` pada **router RTR_MATARAM**.

```
RTR_MATARAM#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

    192.168.0.0/24 is variably subnetted, 4 subnets, 3 masks
D    192.168.0.0/24 is a summary, 01:56:27, Null0
C    192.168.0.0/30 is directly connected, Serial0/1/0
L    192.168.0.1/32 is directly connected, Serial0/1/0
D    192.168.0.4/30 [90/2681856] via 192.168.0.2, 00:58:31, Serial0/1/0
    192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C    192.168.1.0/24 is directly connected, GigabitEthernet0/0/0
L    192.168.1.1/32 is directly connected, GigabitEthernet0/0/0
D    192.168.2.0/24 [90/2170112] via 192.168.0.2, 01:32:57, Serial0/1/0
D    192.168.3.0/24 [90/2682112] via 192.168.0.2, 00:41:59, Serial0/1/0
```

Terlihat **router RTR_MATARAM** telah memperoleh informasi untuk menjangkau subnet yang tidak terhubung langsung sebagai hasil dari pertukaran **EIGRP** antar **router** yang ditandai dengan kode “D”.

VERIFIKASI TABLE ROUTING DI RTR_SUMBAWA

- Melalui **Logical Workspace** dari **Cisco Packet Tracer**, klik pada **router RTR_SUMBAWA**. Pada kotak dialog *properties* dari **RTR_SUMBAWA** yang tampil, klik tab **CLI** dan tekan **Enter** untuk memunculkan **prompt CLI**.
- Berpindah dari mode **user** ke mode **privilege**.

```
RTR_SUMBAWA>enable  
Password:
```

- Tampil inputan inputan **Password:**, masukkan sandi dari **privilege mode** yaitu “**menanti**”.
- Eksekusi perintah `show ip route` untuk menampilkan informasi tabel *routing* pada **router RTR_SUMBAWA**.

VERIFIKASI TABLE ROUTING DI RTR_SUMBAWA

- Hasil eksekusi dari perintah `show ip route` pada **router RTR_SUMBAWA**.

```
RTR_SUMBAWA#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

    192.168.0.0/24 is variably subnetted, 5 subnets, 3 masks
D    192.168.0.0/24 is a summary, 01:27:35, Null0
C    192.168.0.0/30 is directly connected, Serial0/1/0
L    192.168.0.2/32 is directly connected, Serial0/1/0
C    192.168.0.4/30 is directly connected, Serial0/1/1
L    192.168.0.5/32 is directly connected, Serial0/1/1
D    192.168.1.0/24 [90/2170112] via 192.168.0.1, 01:37:59, Serial0/1/0
    192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C    192.168.2.0/24 is directly connected, GigabitEthernet0/0/0
L    192.168.2.1/32 is directly connected, GigabitEthernet0/0/0
D    192.168.3.0/24 [90/2170112] via 192.168.0.6, 00:36:38, Serial0/1/1
```

Terlihat **router RTR_SUMBAWA** telah memperoleh informasi untuk menjangkau subnet yang tidak terhubung langsung sebagai hasil dari pertukaran **EIGRP** antar **router** yang ditandai dengan kode “D”.

VERIFIKASI TABLE ROUTING DI RTR_BIMA

- Melalui **Logical Workspace** dari **Cisco Packet Tracer**, klik pada **router RTR_BIMA**. Pada kotak dialog *properties* dari **RTR_BIMA** yang tampil, klik tab **CLI** dan tekan **Enter** untuk memunculkan **prompt CLI**.

- Berpindah dari mode **user** ke mode **privilege**.

```
RTR_BIMA>enable  
Password:
```

- Tampil inputan inputan **Password:**, masukkan sandi dari **privilege mode** yaitu “**menanti**”.
- Eksekusi perintah `show ip route` untuk menampilkan informasi tabel *routing* pada **router RTR_BIMA**.

VERIFIKASI TABLE ROUTING DI RTR_BIMA

- Hasil eksekusi dari perintah `show ip route` pada **router RTR_BIMA**.

```
RTR_BIMA#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set













      192.168.0.0/24 is variably subnetted, 4 subnets, 3 masks
D       192.168.0.0/24 is a summary, 00:36:43, Null0
D       192.168.0.0/30 [90/2681856] via 192.168.0.5, 00:42:01, Serial0/1/1
C       192.168.0.4/30 is directly connected, Serial0/1/1
L       192.168.0.6/32 is directly connected, Serial0/1/1
D       192.168.1.0/24 [90/2682112] via 192.168.0.5, 00:42:01, Serial0/1/1
D       192.168.2.0/24 [90/2170112] via 192.168.0.5, 00:42:01, Serial0/1/1
      192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.3.0/24 is directly connected, GigabitEthernet0/0/0
L       192.168.3.1/32 is directly connected, GigabitEthernet0/0/0
```

Terlihat **router RTR_BIMA** telah memperoleh informasi untuk menjangkau subnet yang tidak terhubung langsung sebagai hasil dari pertukaran **EIGRP** antar **router** yang ditandai dengan kode “**D**”.

VERIFIKASI KONEKSI ANTAR HOSTS

- Lakukan verifikasi koneksi antar **hosts** menggunakan **Simple PDU** dari **PC A** ke:
 - **PC B** dan **Server0** di **LAN MATARAM**.
 - **PC C** dan **D** di **LAN SUMBAWA**.
 - **PC E** dan **F** di **LAN BIMA**.
- Hasil dari keseluruhan verifikasi yang dilakukan, seperti terlihat pada gambar berikut:

PDU List Window

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	A	B	ICMP		0.000	N	0	(edit)	
	Successful	A	Server0	ICMP		0.000	N	1	(edit)	
	Successful	A	C	ICMP		0.000	N	2	(edit)	
	Successful	A	D	ICMP		0.000	N	3	(edit)	
	Successful	A	E	ICMP		0.000	N	4	(edit)	
	Successful	A	F	ICMP		0.000	N	5	(edit)	

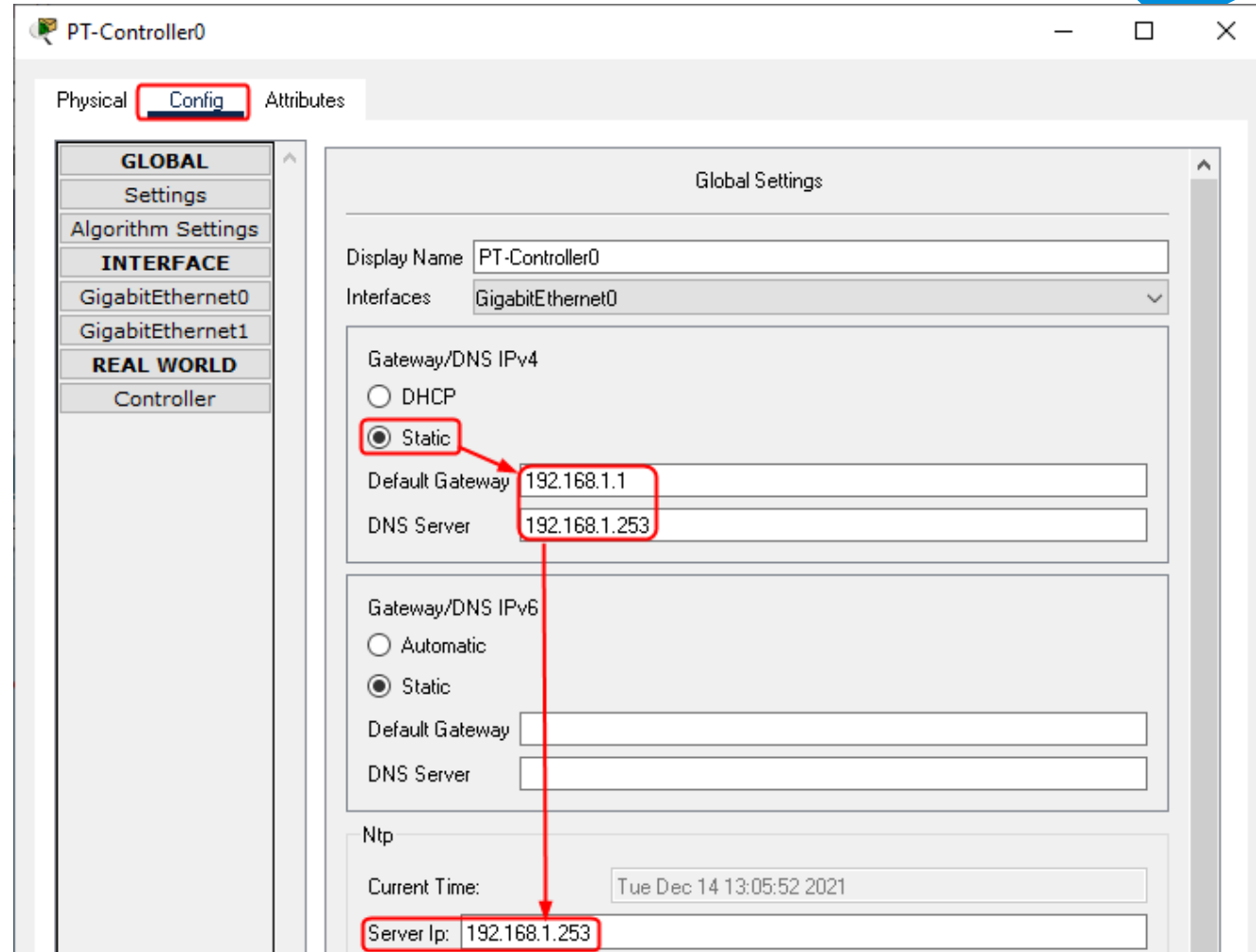
- Terlihat koneksi dari **PC A** ke seluruh **hosts** baik di jaringan lokal maupun beda jaringan berhasil dilakukan.

KONFIGURASI PT- Controller0 DI LAN MATARAM

Mengatur Pengalamatan IP secara Statik dan parameter Transmission Control Protocol/Internet Protocol (TCP/IP) lainnya pada **Network Controller PT-Controller0**.

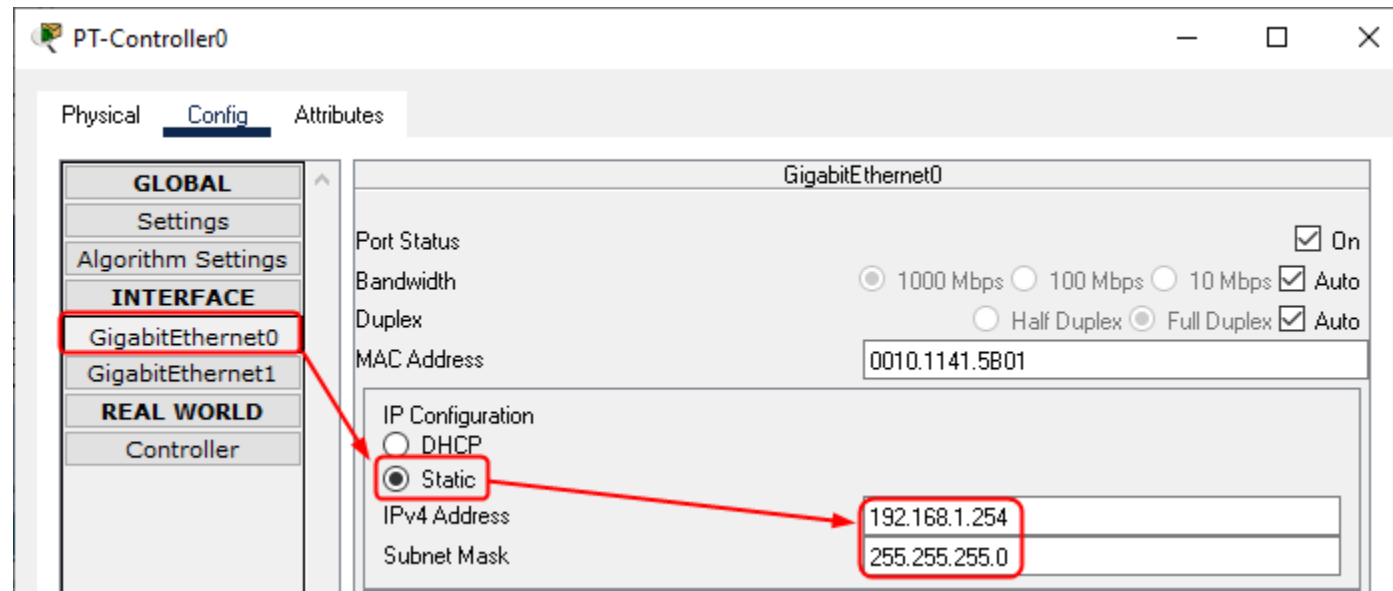
MENGATUR PARAMETER TCP/IP PADA GLOBAL SETTINGS

- Melalui **Logical Workspace** dari **Cisco Packet Tracer** klik pada **Network Controller PT-Controller0**. Pilih tab **Config** pada kotak dialog yang tampil.
- Pada bagian **Gateway/DNS IPv4**, pilih **Static** dan lengkapi pengaturan **Default Gateway** menggunakan **192.168.1.1** serta **DNS Server** menggunakan **192.168.1.253**.
- Selain itu pada bagian **Ntp**, lakukan pengaturan pada parameter **Server IP** menggunakan **192.168.1.253**.



MENGATUR PENGALAMATAN IP SECARA STATIK

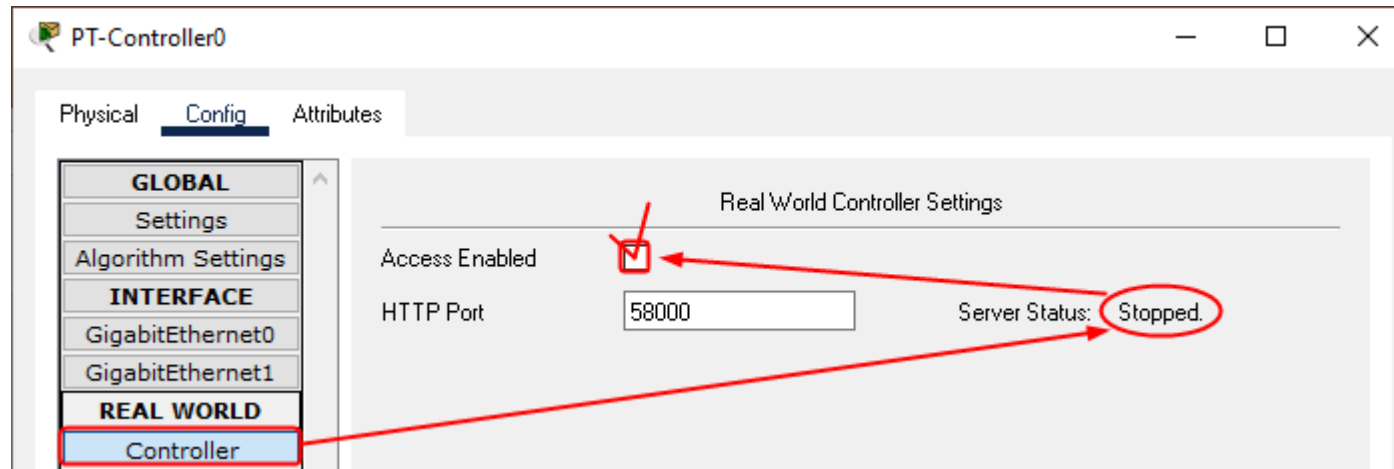
- Pada panel menu sebelah kiri, pilih **GigabitEthernet0**.
- Pada panel detail **GigabitEthernet0** sebelah kanan, pilih **Static** pada bagian **IP Configuration**.
- Lengkapi pengaturan parameter **IPv4 Address** menggunakan **192.168.1.254** dan **Subnetmask** menggunakan **255.255.255.0**.



- Tutup kotak dialog **PT-Controller0**.

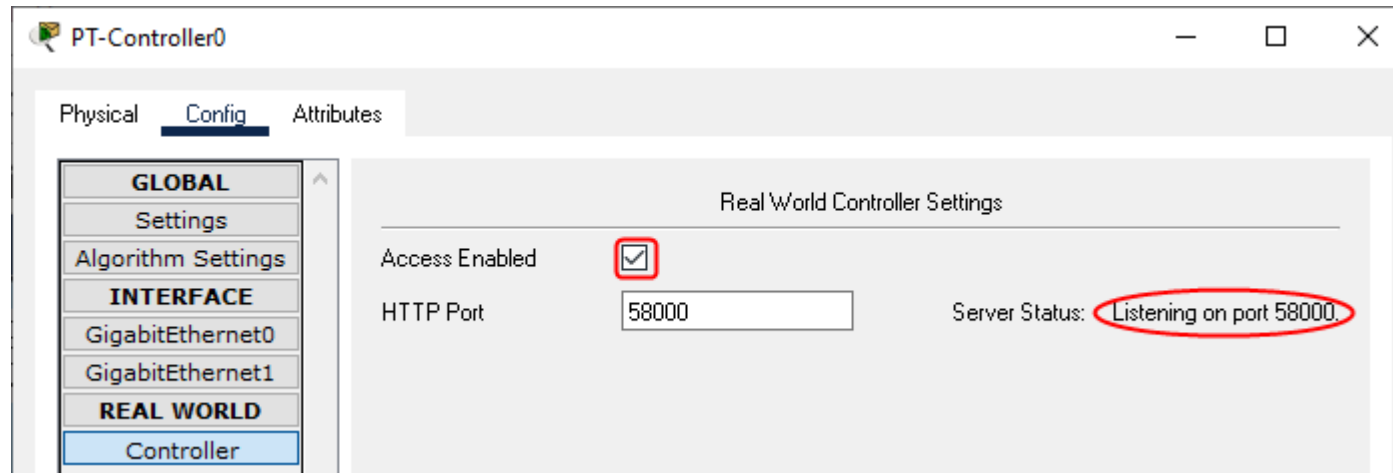
MENGATUR KONEKTIVITAS DARI PT-CONTROLLER0

- Pada panel menu sebelah kiri, pilih **Controller** dibawah menu **REAL WORLD**.
- Pada panel detail sebelah kanan dari **Real World Controller Settings** akan memperlihatkan **Server Status**.
- Apabila bernilai **Stopped** maka selanjutnya lakukan pengaktifan akses (konektivitas) dengan memilih **checkbox Access Enabled**.



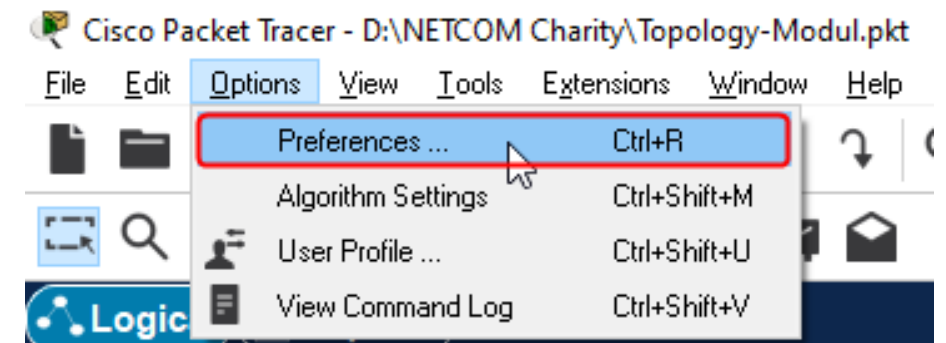
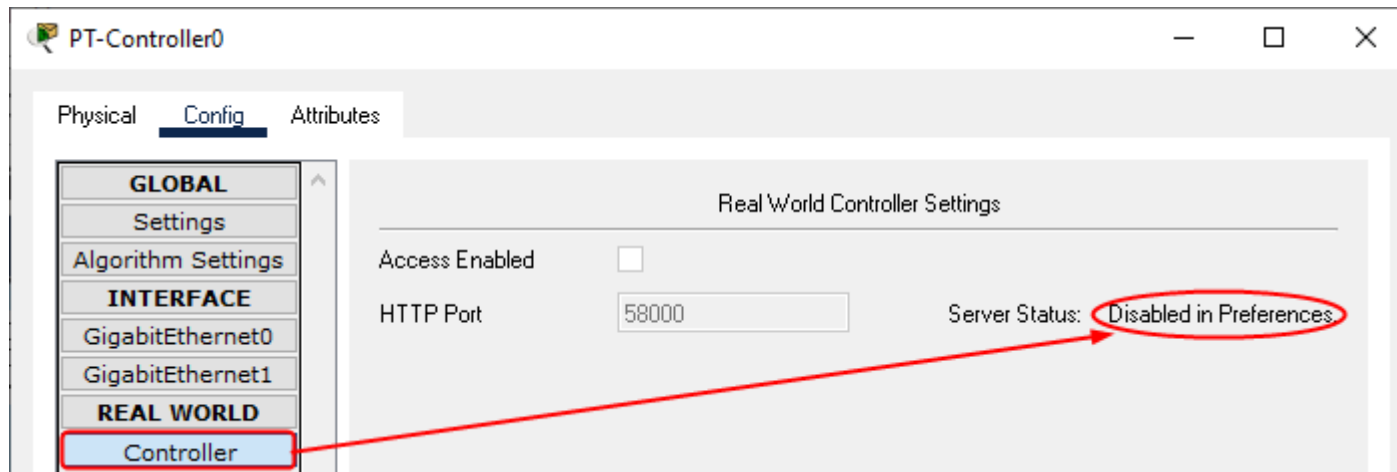
MENGATUR KONEKTIVITAS DARI PT-CONTROLLER0

- Hasil akhir pengaktifan **Access Enabled**, akan memperlihatkan pesan **Listening on port 58000** di bagian **Server Status**. Secara **default HTTP Port** menggunakan **58000**.



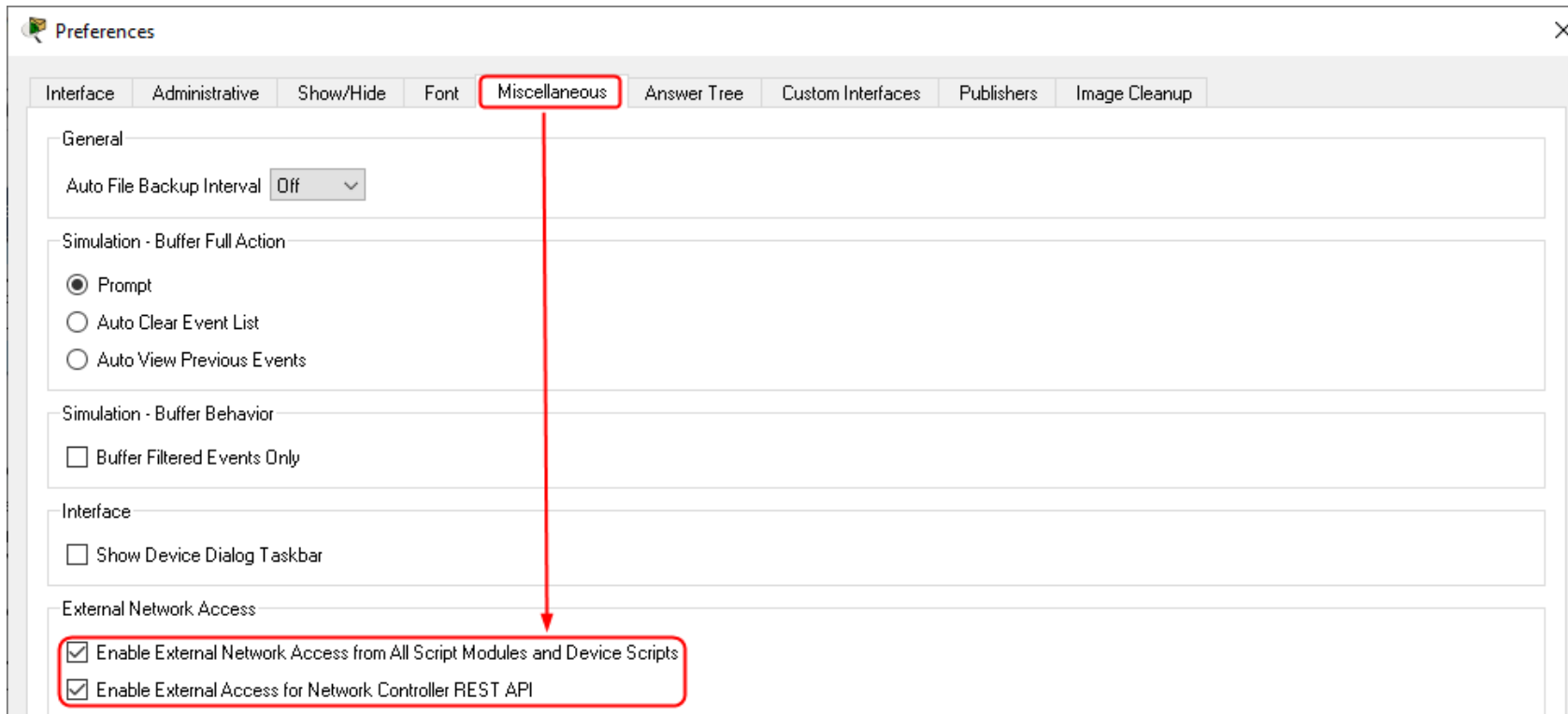
MENGATUR KONEKTIVITAS DARI PT-CONTROLLER0

- Sebaliknya jika **Server Status** bernilai **Disabled in Preferences** maka diperlukan pengaktifan akses dari eksternal melalui menu **Options** → **Preferences** dari **Cisco Packet Tracer**.



MENGATUR KONEKTIVITAS DARI PT-CONTROLLER0

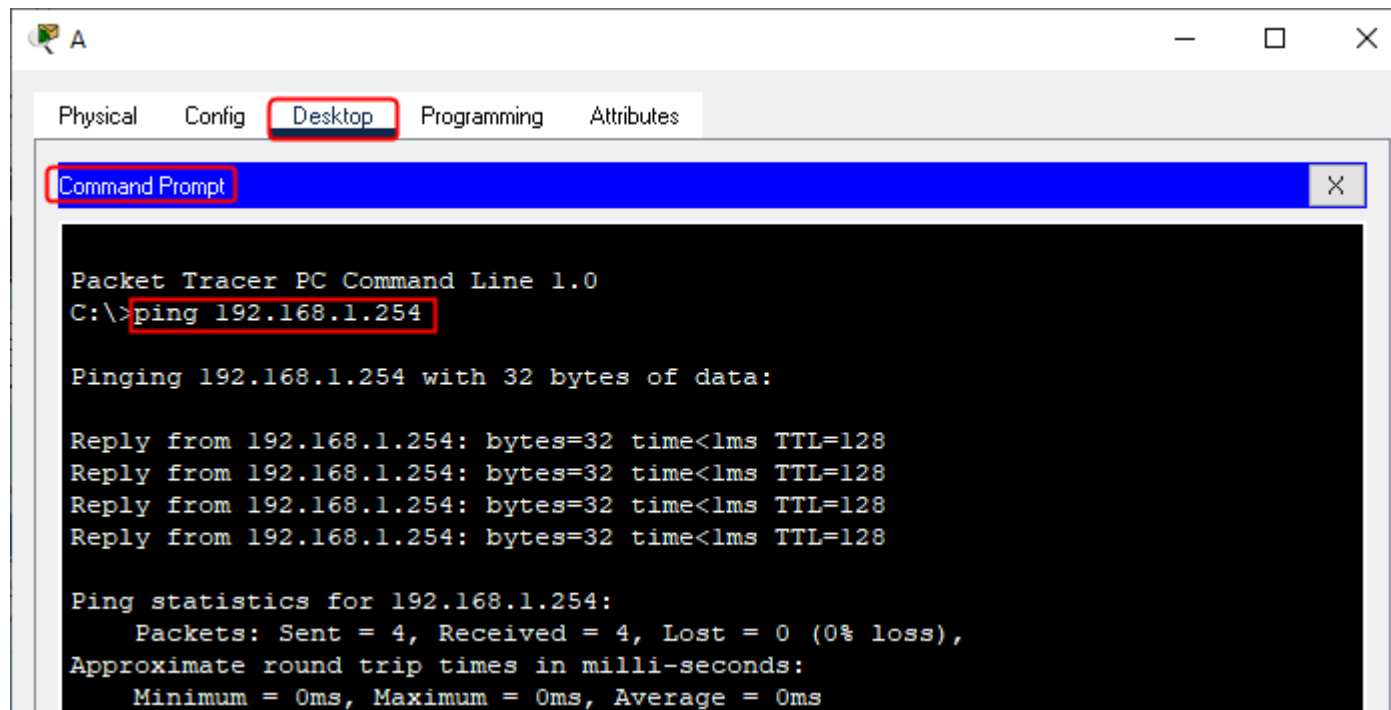
- Tampil kotak dialog **Preferences**, pilih tab **Miscellaneous**. Selanjutnya pada bagian **External Network Access**, centang checkbox **Enable External Access for Network Controller REST API**. Selain itu centang juga checkbox **Enable External Network Access from All Script Modules and Devices Scripts** untuk mengaktifkan akses ke jaringan eksternal dari seluruh skrip.



Tutup kotak dialog **Preferences**. Lakukan pengaktifan konektivitas menggunakan langkah sebelumnya yaitu pada **properties** dari **PT-Controller0**, pilih menu **Controller** dan centang checkbox **Access Enabled**.

MEMVERIFIKASI KONEKSI KE PT-Controller0

- Melalui **Logical Workspace** dari **Cisco Packet Tracer** klik pada **PC A**. Pilih tab **Desktop** pada kotak dialog yang tampil dan pilih **Command Prompt**.
- Tampil kotak dialog **Command Prompt**. Eksekusi perintah **ping 192.168.1.254** pada **prompt** yang tampil. Terlihat koneksi dari **PC A** ke **PT-Controller0** berhasil dilakukan.



```
Packet Tracer PC Command Line 1.0
C:\>ping 192.168.1.254

Pinging 192.168.1.254 with 32 bytes of data:

Reply from 192.168.1.254: bytes=32 time<lms TTL=128
Reply from 192.168.1.254: bytes=32 time<lms TTL=128
Reply from 192.168.1.254: bytes=32 time<lms TTL=128
Reply from 192.168.1.254: bytes=32 time<lms TTL=128

Ping statistics for 192.168.1.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

MEMVERIFIKASI KONEKSI KE PT-Controller0

- Dengan cara yang sama, lakukan verifikasi koneksi ke **PT-Controller0** menggunakan perintah **ping controller.sabarmenanti.com**.

```
C:\>ping controller.sabarmenanti.com

Pinging 192.168.1.254 with 32 bytes of data:

Reply from 192.168.1.254: bytes=32 time<lms TTL=128
Reply from 192.168.1.254: bytes=32 time<lms TTL=128
Reply from 192.168.1.254: bytes=32 time<lms TTL=128
Reply from 192.168.1.254: bytes=32 time<lms TTL=128

Ping statistics for 192.168.1.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

- Terlihat koneksi berhasil dilakukan. Tutup kotak dialog **Command Prompt**.

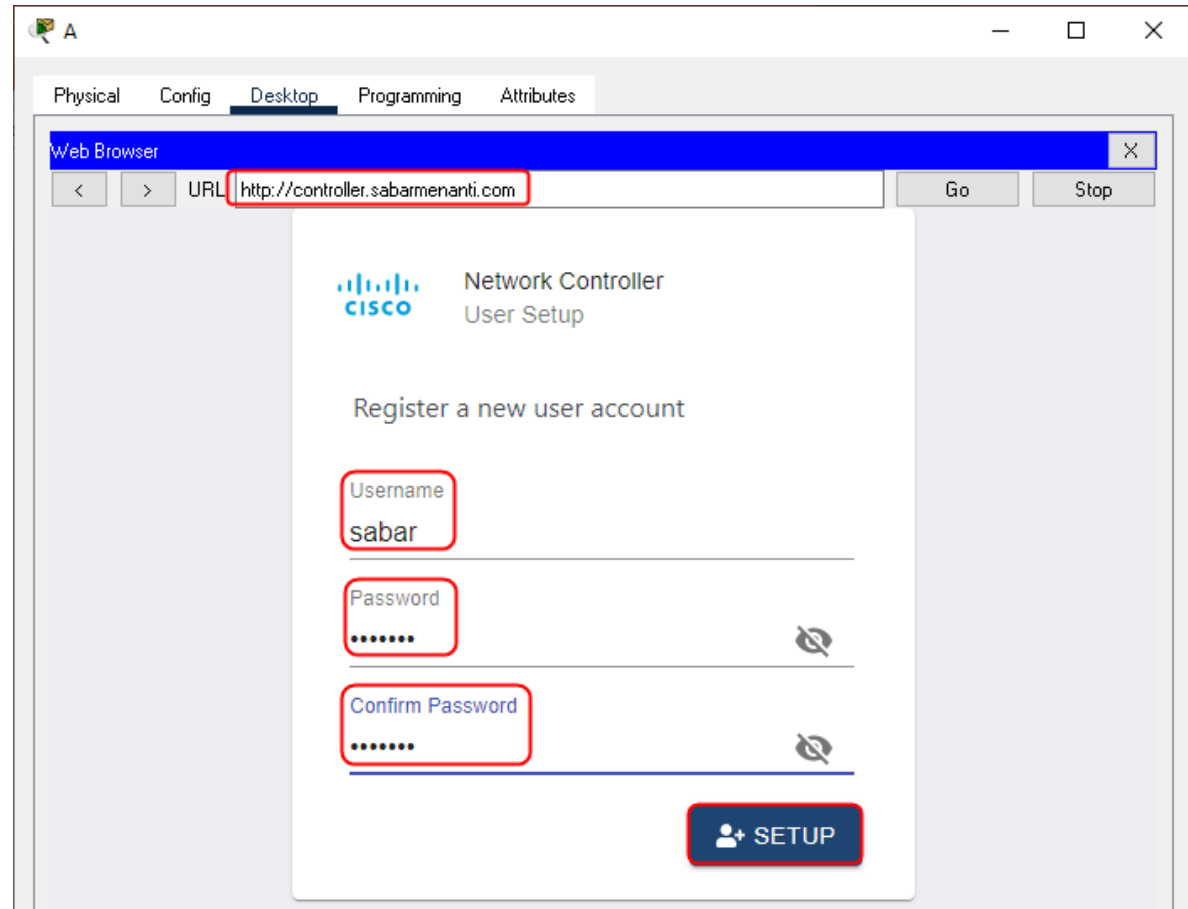
MEMBUAT AKUN PENGGUNA DAN LOGIN KE SDN CONTROLLER

- Pilih **Web Browser** pada tab **Desktop** dari kotak dialog *properties PC A*. Pada inputan URL dari **Web Browser**, masukkan alamat **controller.sabarmenanti.com** dan tekan tombol **Go**.

Tampil halaman **Register a new user account** untuk membuat akun pengguna baru di **Network Controller**. Lengkapi parameter-parameter berikut:

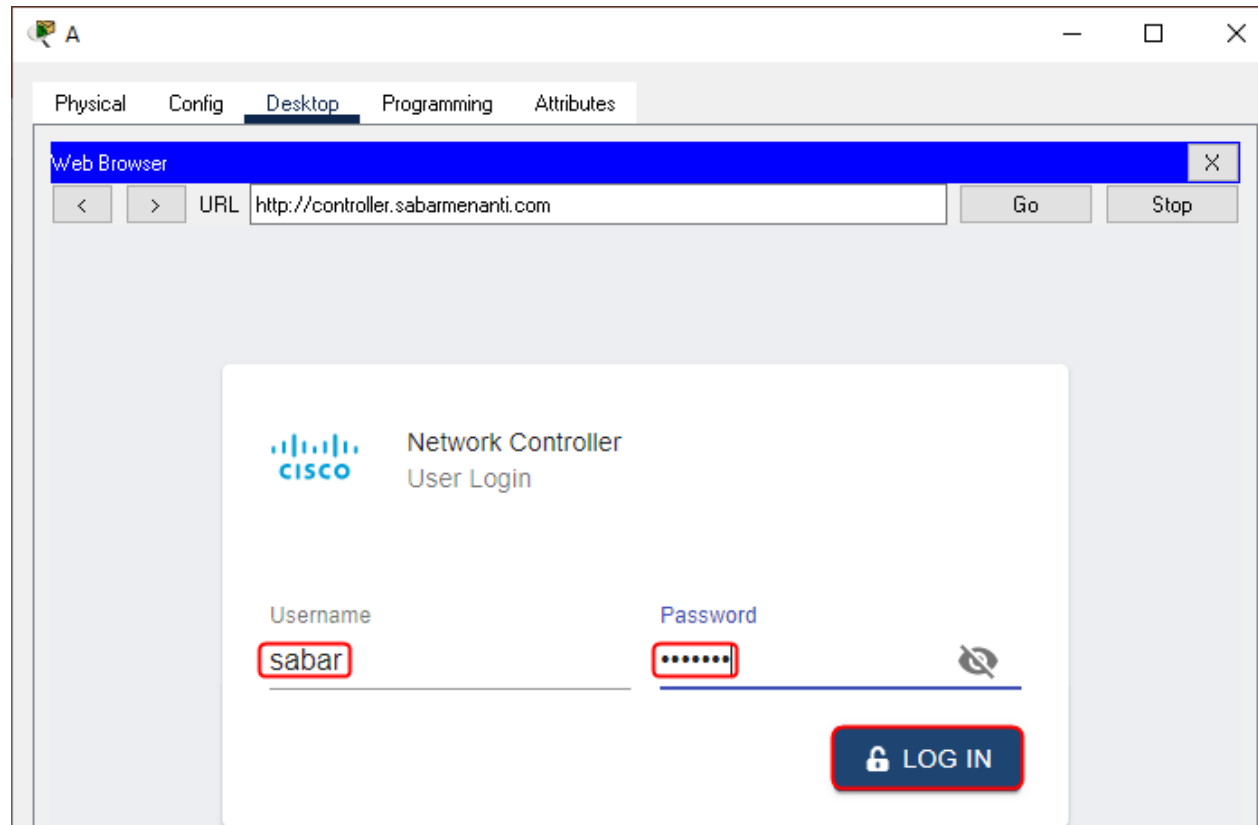
- **Username**, masukkan nama login, sebagai contoh **sabar**.
- **Password dan Confirm Password**, masukkan sandi login untuk nama login yang dibuat, sebagai contoh **menanti**.

Tekan tombol **SETUP** untuk memproses pembuatan akun baru.



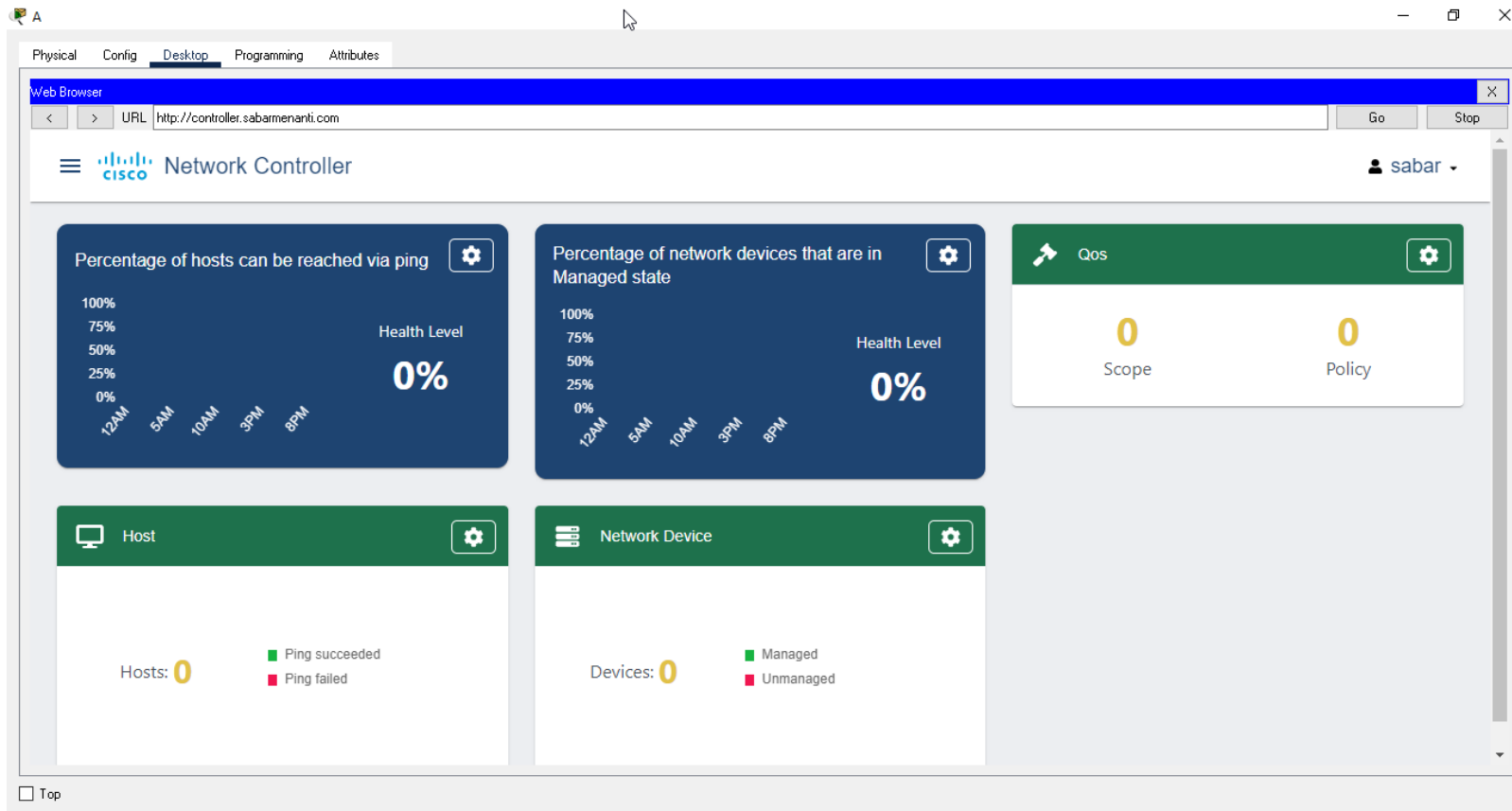
LOGIN KE SDN CONTROLLER

- Tampil halaman **User Login** dari **Network Controller**. Masukkan **username sabar** dengan **password menanti** dan tekan tombol **LOG IN**.



DASHBOARD DARI SDN CONTROLLER

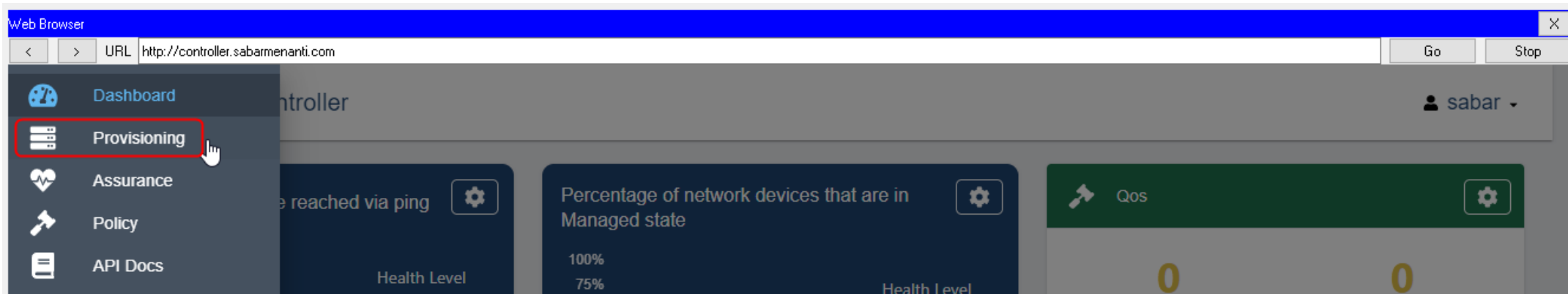
- Apabila proses otentikasi login berhasil dilakukan maka akan tampil halaman **Dashboard** dari **Network Controller**.



Terlihat ringkasan informasi terkait **persentase hosts** yang dapat dijangkau melalui ping, **persentase perangkat jaringan** yang dikelola, **Quality of Service (QoS)** dan **Host** serta **Network Device** yang masih bernilai **0**.

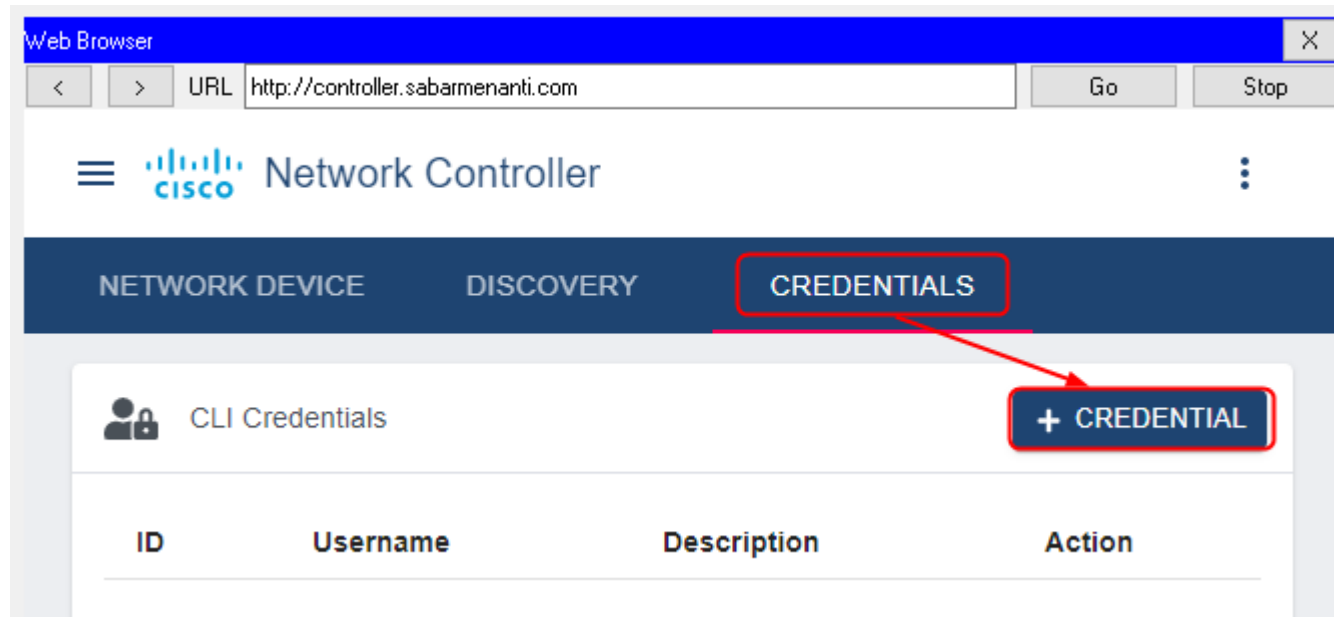
TOPOLOGY DISCOVERY DENGAN SDN CONTROLLER

- **Cisco Discovery Protocol (CDP)** merupakan protokol yang digunakan oleh **PT-Controller0** untuk menemukan secara otomatis keseluruhan perangkat jaringan yang terdapat pada topologi yaitu **3 (tiga) router (RTR_MATARAM, RTR_SUMBAWA, RTR_BIMA)** dan **3 (tiga) multilayer switch (MSW_MATARAM, MSW_SUMBAWA, MSW_BIMA)**. Selain itu **PT-Controller0** juga dapat menemukan 7 (tujuh) hosts yaitu **Server0, PC A, B, C, D, E dan F**.
- Untuk mengakses keseluruhan perangkat jaringan pada topologi maka diperlukan penambahan **Credentials**. Klik tombol menu pada sebelah kiri dari logo **Cisco** yang terdapat pada bagian *header* dari **Network Controller Graphical User Interface (GUI)** dan pilih **Provisioning**.



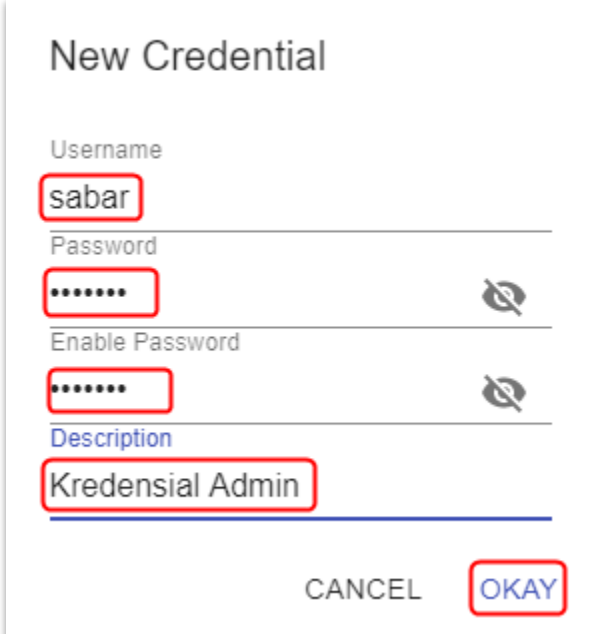
TOPOLOGY DISCOVERY DENGAN SDN CONTROLLER

- Tampil halaman **Provisioning** dan pilih menu **CREDENTIALS** di bagian navigasi **Header**. Klik tombol **+ CREDENTIAL** untuk menambahkan kredensial baru.



TOPOLOGY DISCOVERY DENGAN SDN CONTROLLER

- Tampil kotak dialog **New Credential**. Lengkapi pengaturan dari parameter-parameter berikut:
 - a) **Username**, masukkan nama login pengguna yaitu **sabar**.
 - b) **Password**, masukkan sandi dari nama login pengguna yaitu **menanti**.
 - c) **Enable Password**, masukkan **password privilege mode** yaitu **menanti**.
 - d) **Description**, masukkan deskripsi sebagai keterangan dair kredensial yang ditambahkan yaitu **Kredensial Admin**.
- Tekan **OKAY** untuk memproses pembuatan kredensial baru.



New Credential

Username
sabar

Password

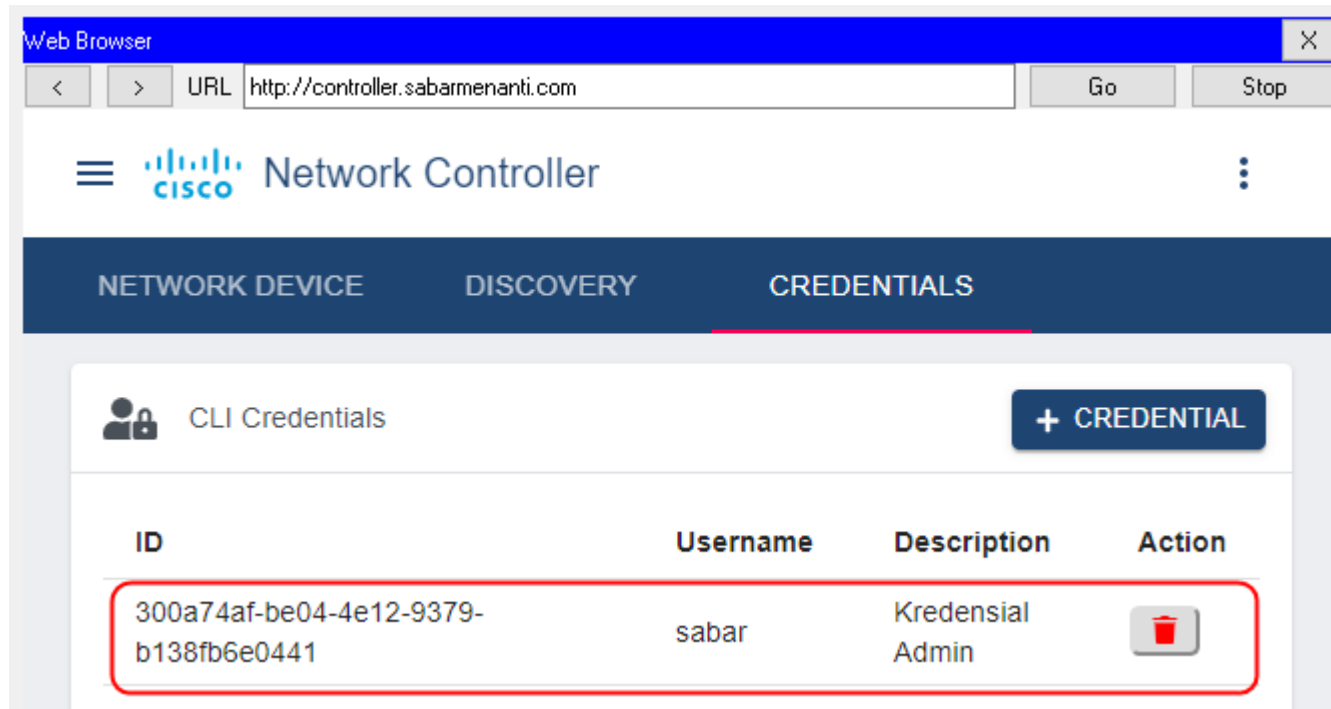
Enable Password

Description
Kredensial Admin


CANCEL OKAY

TOPOLOGY DISCOVERY DENGAN SDN CONTROLLER

- Hasil dari pembuatan kredensial baru, seperti terlihat pada gambar berikut:



The screenshot shows a web browser window displaying the Cisco Network Controller interface. The URL bar shows `http://controller.sabarmenanti.com`. The page title is "Network Controller" with the Cisco logo. The navigation menu includes "NETWORK DEVICE", "DISCOVERY", and "CREDENTIALS". The "CREDENTIALS" section is active, showing "CLI Credentials" with a "+ CREDENTIAL" button. A table lists the credentials, with one entry highlighted by a red box:

ID	Username	Description	Action
300a74af-be04-4e12-9379-b138fb6e0441	sabar	Kredensial Admin	

TOPOLOGY DISCOVERY DENGAN SDN CONTROLLER

- Klik menu **DISCOVERY** di bagian navigasi *header* dan klik tombol **+ DISCOVERY** untuk menambahkan *discovery* baru. Lengkapi pengaturan dari parameter-parameter berikut:
 - a) **Name**, masukkan nama pengenal untuk **discovery** baru yang ditambahkan yaitu **MSW_MATARAM**.
 - b) **IP Address**, masukkan alamat IP dari **Multilayer Switch MSW_MATARAM** yaitu **192.168.1.2**
 - c) **CLI Credentials List**, pilih **sabar – Kredensial Admin**.
- Klik **ADD** untuk menyimpan.
- Tampil pesan **Status** yaitu **In Progress**. Klik tombol **Fast Forward** untuk mempercepat pemrosesan.

New Discovery

Discovery Type: CDP

Name: MSW_MATARAM

IP Address: 192.168.1.2

Timeout: 5

Retry: 3

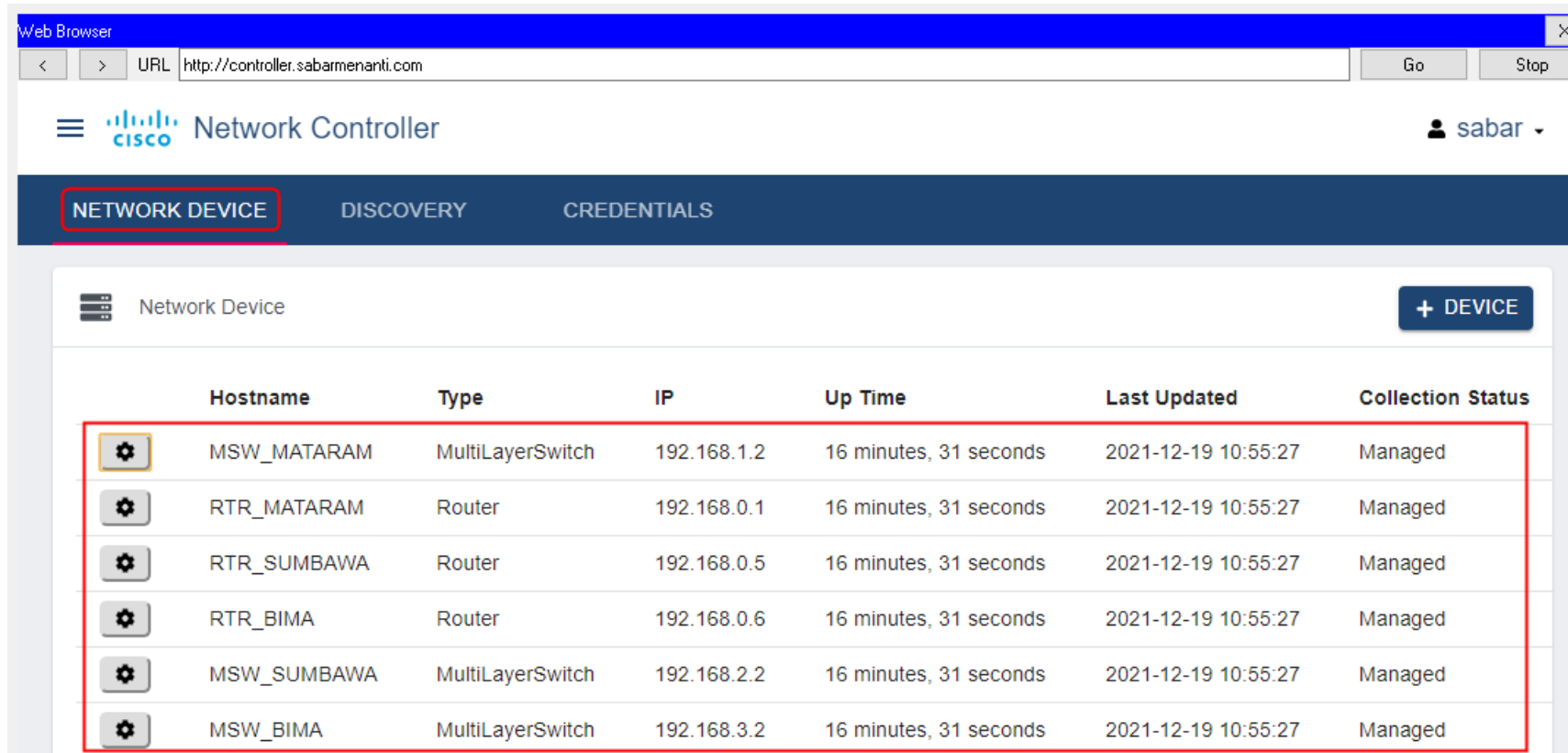
CDP Level: 16

sabar - Kredensial Admin

CANCEL ADD

TOPOLOGY DISCOVERY DENGAN SDN CONTROLLER

- Klik menu **NETWORK DEVICE** di bagian navigasi *header* maka selanjutnya akan tampil daftar informasi perangkat jaringan yang telah berhasil ditemukan (**discovery**). Terlihat terdapat 6 (enam) perangkat jaringan yang ditemukan meliputi **3 (tiga) router** dan **3 (tiga) multilayerswitch**.

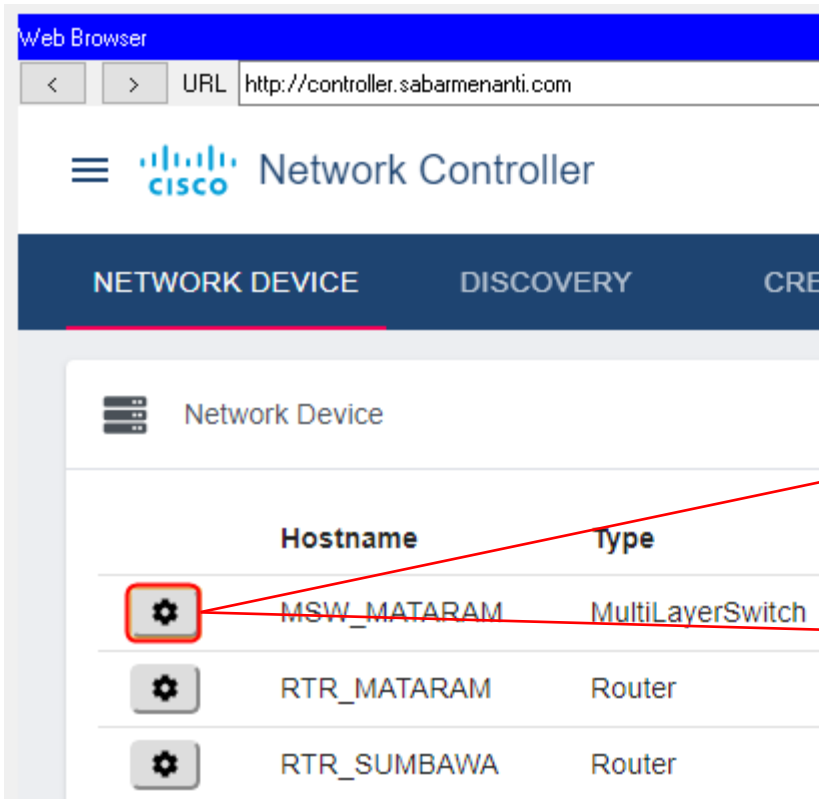


The screenshot shows the Cisco Network Controller web interface. The 'NETWORK DEVICE' menu item is highlighted with a red box. Below the navigation bar, a table displays the discovered network devices. The table has columns for Hostname, Type, IP, Up Time, Last Updated, and Collection Status. A red box highlights the first six rows of the table, which correspond to the three routers and three multilayer switches mentioned in the text.

	Hostname	Type	IP	Up Time	Last Updated	Collection Status
⚙️	MSW_MATARAM	MultiLayerSwitch	192.168.1.2	16 minutes, 31 seconds	2021-12-19 10:55:27	Managed
⚙️	RTR_MATARAM	Router	192.168.0.1	16 minutes, 31 seconds	2021-12-19 10:55:27	Managed
⚙️	RTR_SUMBAWA	Router	192.168.0.5	16 minutes, 31 seconds	2021-12-19 10:55:27	Managed
⚙️	RTR_BIMA	Router	192.168.0.6	16 minutes, 31 seconds	2021-12-19 10:55:27	Managed
⚙️	MSW_SUMBAWA	MultiLayerSwitch	192.168.2.2	16 minutes, 31 seconds	2021-12-19 10:55:27	Managed
⚙️	MSW_BIMA	MultiLayerSwitch	192.168.3.2	16 minutes, 31 seconds	2021-12-19 10:55:27	Managed

PENGUMPULAN INFORMASI DENGAN SDN CONTROLLER

- Klik pada **icon Gear** dari salah satu daftar perangkat jaringan yang tampil di **NETWORK DEVICE**, sebagai contoh **MSW_MATARAM** maka akan menampilkan informasi lebih detail terkait perangkat tersebut.



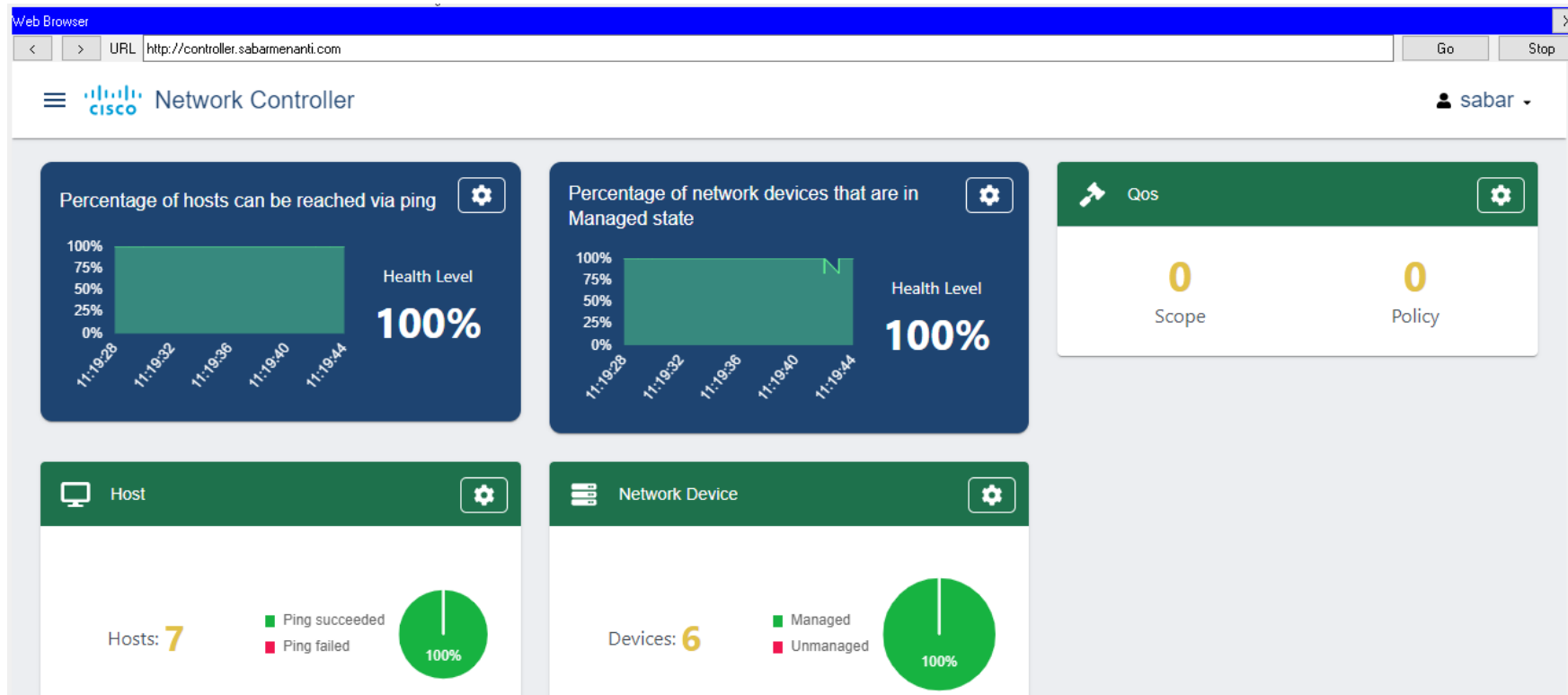
Update Network Device

Device Detail	
Hostname	MSW_MATARAM
ID	CAT10100RK5-uuid
Interface Count	29
Software Version	16.3.2
MAC Address	0001.C912.88EB
Management IP Address	192.168.1.2
Platform ID	3650
Product ID	3650-24PS
Serial Number	CAT10100RK5-
Type	MultiLayerSwitch
UpTime	25 minutes, 2 seconds

Terlihat informasi terkait **Software Version** dan **Product ID** termasuk detail informasi lainnya seperti konektivitas dari perangkat jaringan tersebut.

PENGUMPULAN INFORMASI DENGAN SDN CONTROLLER

- Klik pada **banner Network Controller** di bagian **header** untuk membawa kembali ke **Dashboard** dari **SDN Controller**.



Terlihat ringkasan informasi terkait **persentase hosts** yang dapat dijangkau melalui ping yaitu **100%**, **persentase perangkat jaringan** yang dikelola yaitu **100%**, **Quality of Service (QoS)** dengan jumlah **Scope** dan **Policy** yang masih bernilai **0** dan **Host** sejumlah **7 hosts** yang berhasil dicek koneksinya dengan ping serta **Network Device** sejumlah **6 perangkat yang dikelola**.

Klik **icon Gear** di sebelah kanan dari setiap bagian tersebut untuk menampilkan informasi lebih detail.

PENGUMPULAN INFORMASI DENGAN SDN CONTROLLER

- Informasi terkait **Network Health** dari setiap kategori perangkat dan permasalahan jaringan (**Network Issues**) yang ditemukan oleh **SDN Controller**, dapat diakses dengan memilih menu sebelah kiri dari **logo Cisco** di bagian *header* dan memilih menu **Assurance**.

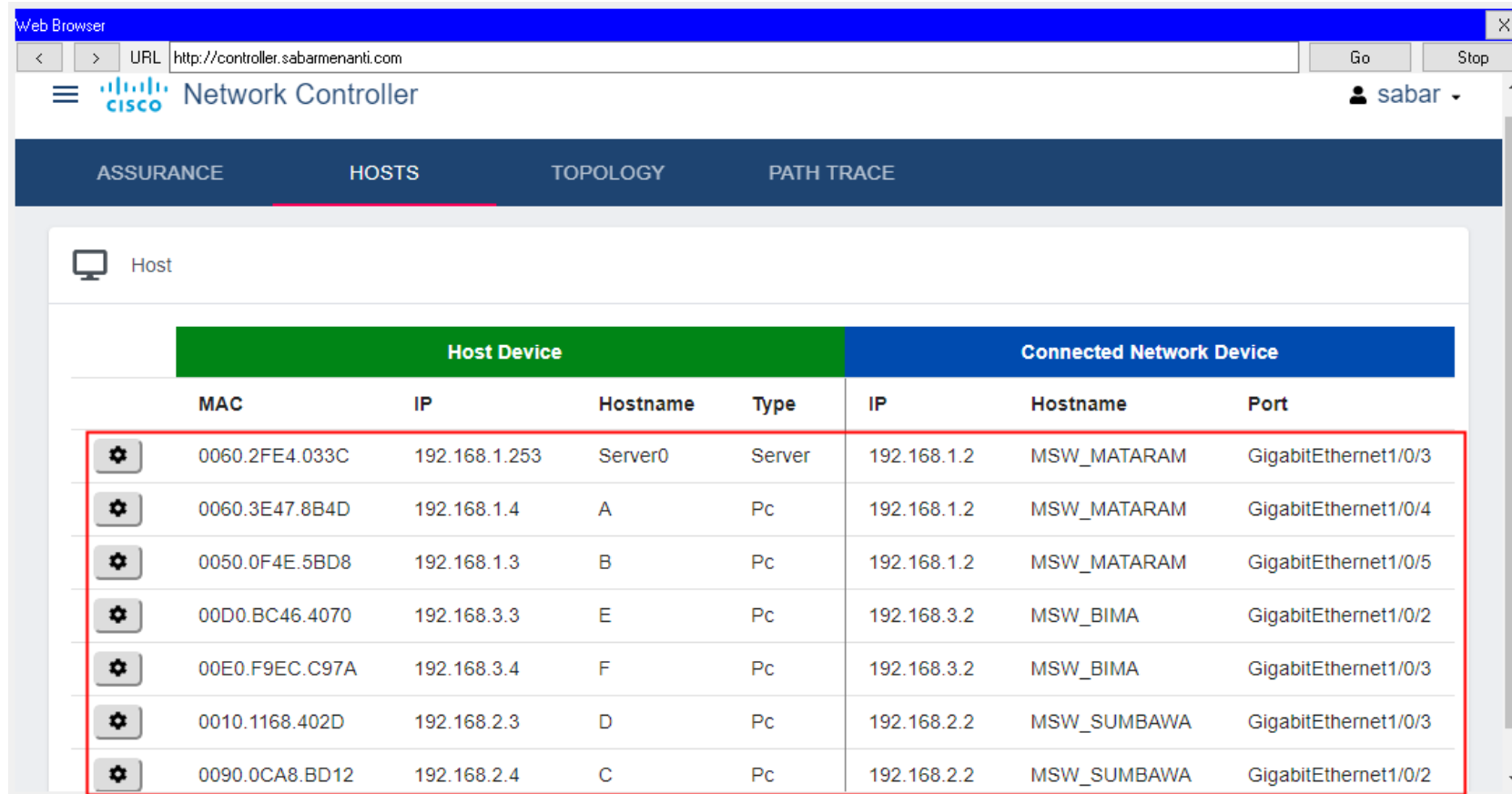
The screenshot displays the Cisco Network Controller Assurance interface. On the left, a navigation menu is visible with the 'Assurance' option highlighted in a red box. The main dashboard area is divided into several sections:

- Assurance**: Overview section showing overall network health.
- Routers**: A card showing 'Healthy Percentage: 100%' and 'Healthy Ratio: 3:3'. It includes a pie chart with 100% green (Managed) and 0% red (Unmanaged).
- Switches**: A card showing 'Healthy Percentage: 100%' and 'Healthy Ratio: 3:3'. It includes a pie chart with 100% green (Managed) and 0% red (Unmanaged).
- Network Health**: A central section with two line graphs. The top graph for 'Routers' shows a health level of 100% over time (11:54:43 to 11:54:59). The bottom graph for 'Switches' also shows a health level of 100%.
- Network Issues**: A table with columns for Date, Name, Description, and Source, currently empty.

Pada bagian **Network Issues**, masih belum terlihat adanya permasalahan jaringan.

PENGUMPULAN INFORMASI DENGAN SDN CONTROLLER

- Pada halaman **Assurance** yang tampil, pilih menu **HOSTS** untuk melihat daftar informasi konektivitas **Layer 2** dan **Layer 3** dari setiap **hosts** yang ditemukan oleh **SDN Controller**.



The screenshot shows the Cisco Network Controller Assurance interface. The 'HOSTS' menu is selected. Below the navigation bar, there is a 'Host' section with a table of host devices and their connections. The table has two main columns: 'Host Device' and 'Connected Network Device'. The 'Host Device' column includes MAC, IP, Hostname, and Type. The 'Connected Network Device' column includes IP, Hostname, and Port. A red box highlights the first seven rows of the table.

Host Device				Connected Network Device		
MAC	IP	Hostname	Type	IP	Hostname	Port
0060.2FE4.033C	192.168.1.253	Server0	Server	192.168.1.2	MSW_MATARAM	GigabitEthernet1/0/3
0060.3E47.8B4D	192.168.1.4	A	Pc	192.168.1.2	MSW_MATARAM	GigabitEthernet1/0/4
0050.0F4E.5BD8	192.168.1.3	B	Pc	192.168.1.2	MSW_MATARAM	GigabitEthernet1/0/5
00D0.BC46.4070	192.168.3.3	E	Pc	192.168.3.2	MSW_BIMA	GigabitEthernet1/0/2
00E0.F9EC.C97A	192.168.3.4	F	Pc	192.168.3.2	MSW_BIMA	GigabitEthernet1/0/3
0010.1168.402D	192.168.2.3	D	Pc	192.168.2.2	MSW_SUMBAWA	GigabitEthernet1/0/3
0090.0CA8.BD12	192.168.2.4	C	Pc	192.168.2.2	MSW_SUMBAWA	GigabitEthernet1/0/2

Terlihat semua **hosts** telah teridentifikasi. Klik **icon Gear** untuk menampilkan informasi detail dari setiap **host**.

VISUALISASI TOPOLOGI JARINGAN DI SDN CONTROLLER

- Pilih menu **TOPOLOGY** untuk menampilkan visualisasi topologi yang dibuat secara dinamis oleh **SDN Controller**. Lakukan **Click and Drag** untuk menata kembali visualisasi perangkat jaringan pada topologi.

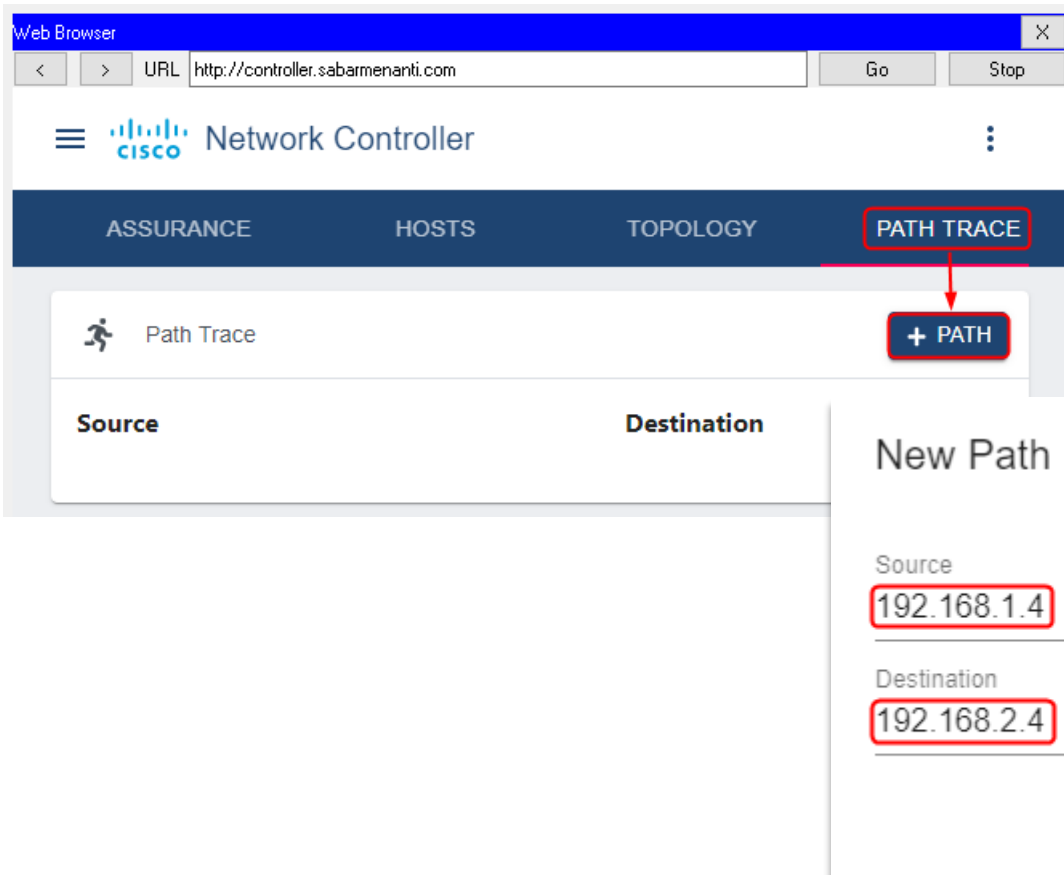
The screenshot shows the Cisco Network Controller web interface. The browser address bar displays 'http://controller.sabarmenanti.com'. The main navigation bar includes 'ASSURANCE', 'HOSTS', 'TOPOLOGY', and 'PATH TRACE'. The 'TOPOLOGY' tab is active, showing a network diagram with nodes: MSW_MATARAM, RTR_MATARAM, MSW_SUMBAWA, RTR_SUMBAWA, MSW_BIMA, and RTR_BIMA. A right-hand panel titled 'Node' displays details for a selected device:

Device Detail	
Collection Status	Managed
Error Description	
Hostname	RTR_SUMBAWA
ID	FDO1302H69Y-uuid
Interface Count	6
Inventory Status Detail	Managed
IOS Version	

- Klik pada perangkat jaringan untuk menampilkan informasi detail terkait perangkat tersebut.

PATH TRACE DI SDN CONTROLLER

- Pilih menu **PATH TRACE** untuk menampilkan rute perjalanan paket dari sumber ke tujuan tertentu pada **SDN Controller**. Klik **+ PATH** untuk menambahkan penelusuran rute perjalanan paket.



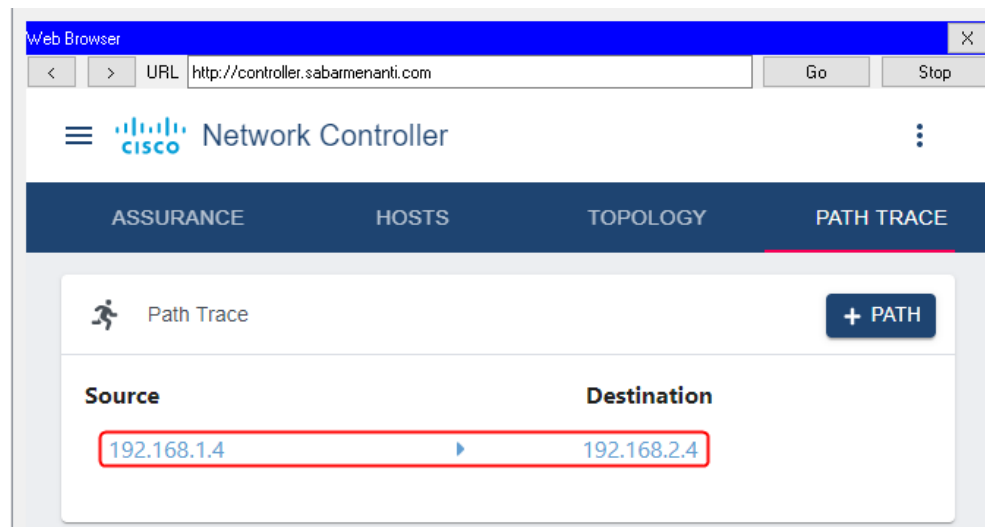
The screenshot shows the Cisco Network Controller web interface. The top navigation bar includes 'ASSURANCE', 'HOSTS', 'TOPOLOGY', and 'PATH TRACE'. The 'PATH TRACE' menu is highlighted. Below it, there is a 'Path Trace' section with a '+ PATH' button. A dialog box titled 'New Path' is open, showing input fields for 'Source' (192.168.1.4) and 'Destination' (192.168.2.4). The 'OKAY' button is highlighted.

Tampil kotak dialog **New Path**. Lengkapi inputan parameter **Source** dengan alamat IP PC sumber. Sedangkan inputan parameter **Destination** dengan alamat IP PC tujuan.

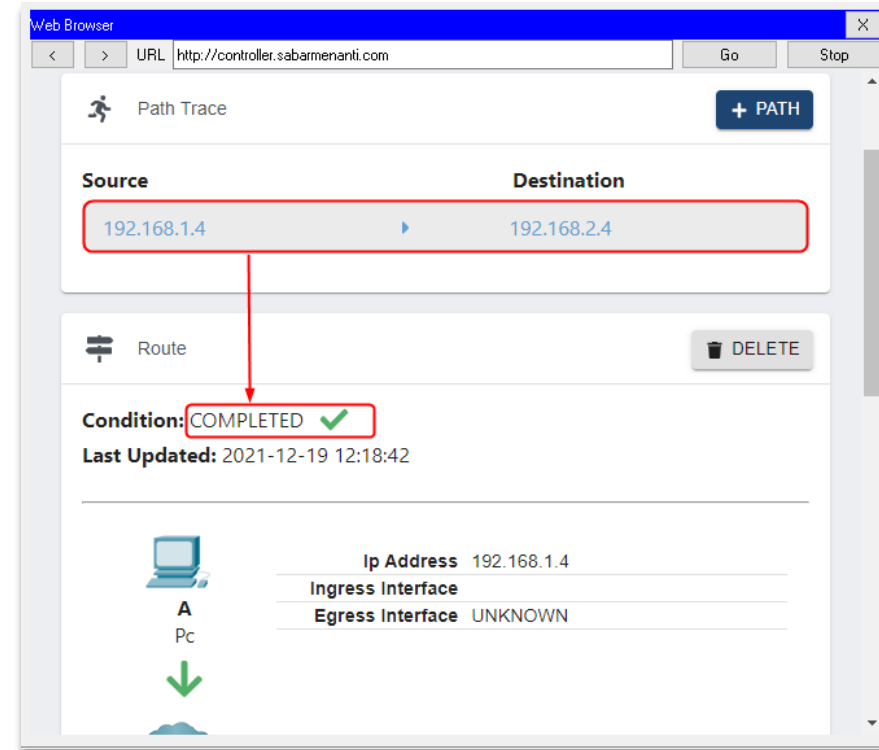
Sebagai contoh dari **PC A** dengan alamat IP **192.168.1.4** di **LAN MATARAM** menuju ke **PC C** dengan alamat IP **192.168.2.4** di **LAN SUMBAWA**. Sesuaikan alamat IP dari setiap PC jika memiliki nilai berbeda. Klik **OKAY** untuk memproses.

PATH TRACE DI SDN CONTROLLER

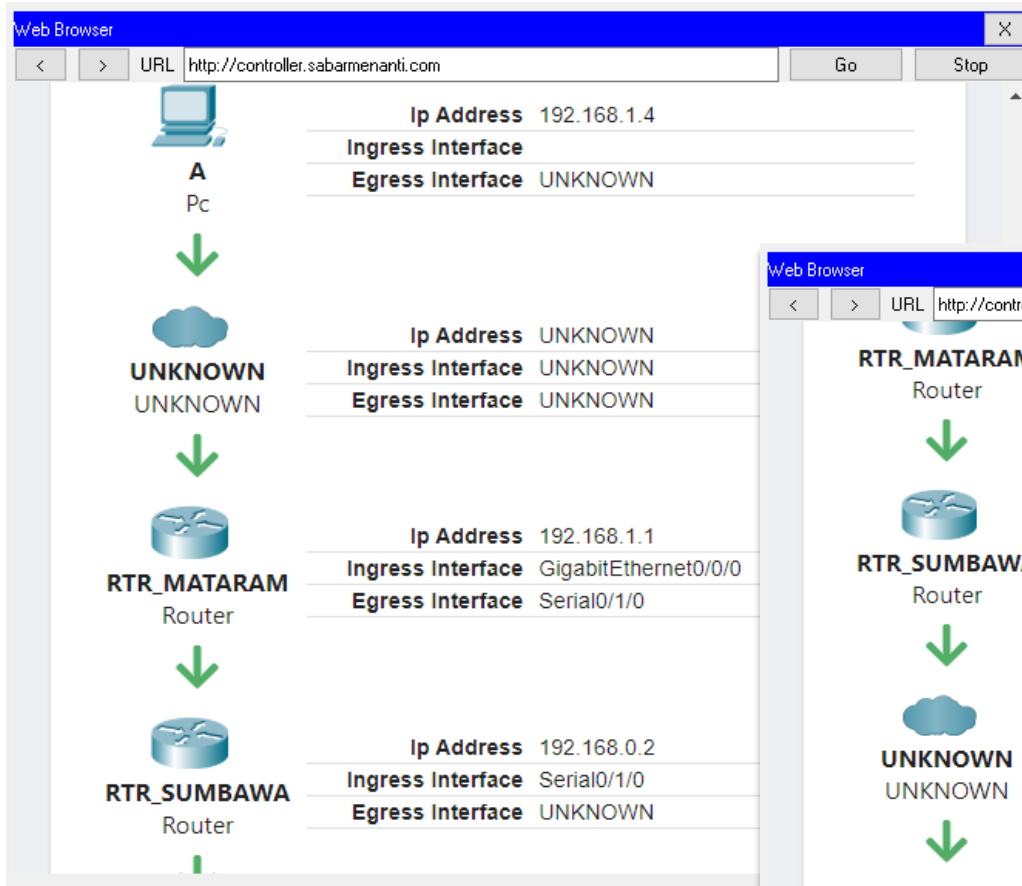
- Hasil penambahan **PATH** dari **PC A** ke **PC C**, seperti terlihat pada gambar berikut:



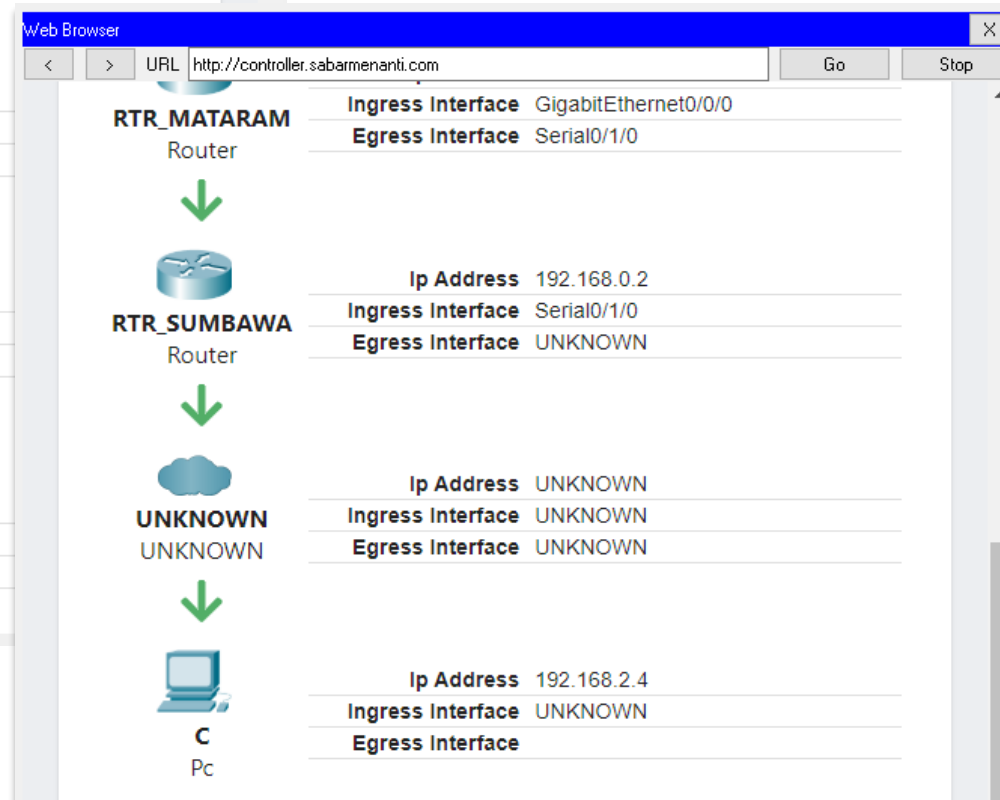
Klik pada **PATH** baru yang telah ditambahkan untuk menampilkan informasi detail terkait penelusuran perjalanan paket data dari **PC A** ke **PC C**.



PATH TRACE DI SDN CONTROLLER



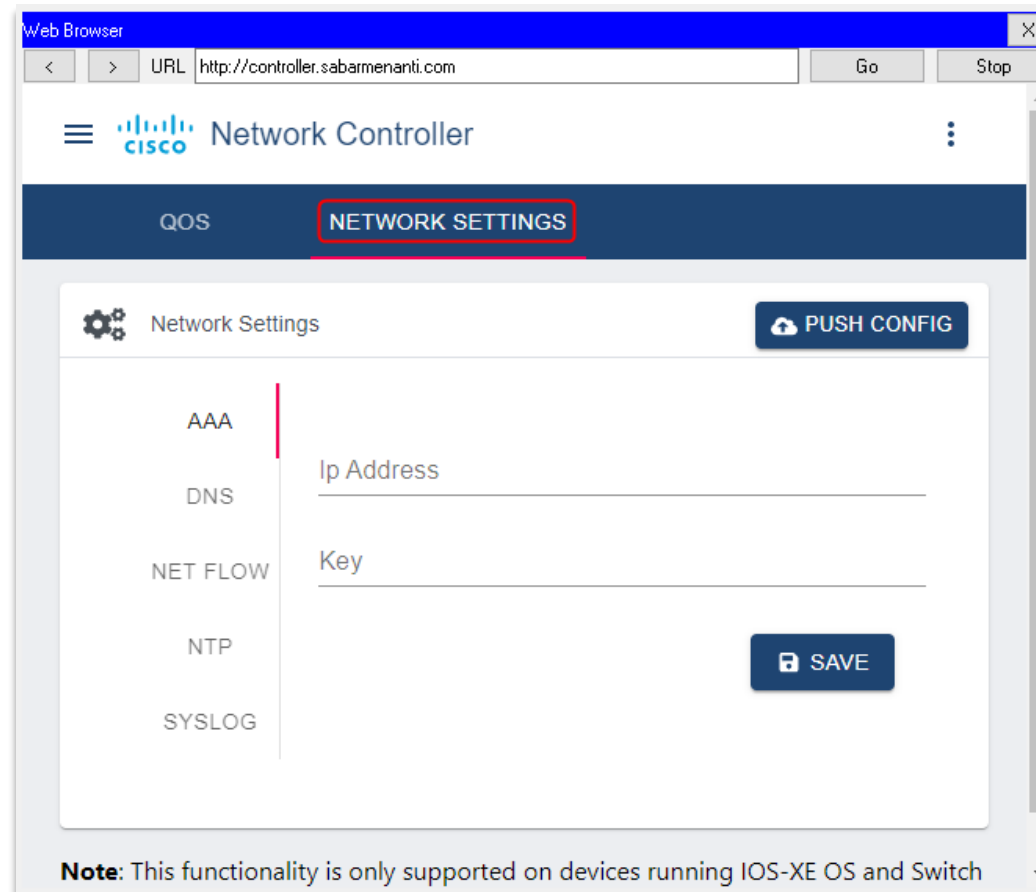
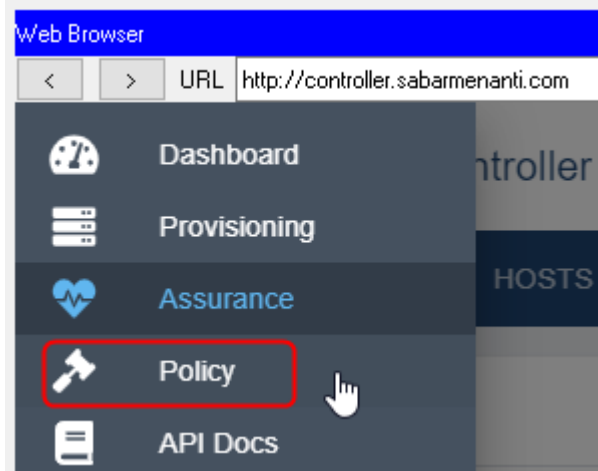
- Terlihat hanya informasi **layer 3** yaitu **Router** yang ditampilkan. Sedangkan **Switch** ditampilkan sebagai perangkat **UNKNOWN** karena bekerja di **layer 2**.



- Rute perjalanan paket data yang bersumber dari **PC A** ke **PC C** yaitu paket pada awalnya akan dikirimkan ke perangkat **UNKNOWN (MSW_MATARAM)** dan kemudian diteruskan ke **RTR_MATARAM**. Dari **RTR_MATARAM**, paket akan dikirimkan ke **RTR_SUMBAWA**. Setelah itu diteruskan ke perangkat **UNKNOWN (MSW_SUMBAWA)** sehingga sampai ke **PC C**.

KONFIGURASI JARINGAN DI SDN CONTROLLER

- Konfigurasi jaringan di **SDN Controller** dapat dilakukan dengan memilih menu sebelah kiri dari **logo Cisco** di bagian *header* dan memilih menu **Policy**. Pada halaman yang tampil, pilih **NETWORK SETTINGS**.



- Terlihat terdapat pengaturan **Authentication Authorization Accounting (AAA), DNS, NET FLOW, NTP** dan **SYSLOG**.
- Setelah dilakukan pengaturan pada setiap komponen tersebut maka dapat dilakukan **PUSH CONFIG** ke perangkat jaringan yang mendukung.

KONFIGURASI JARINGAN DI SDN CONTROLLER

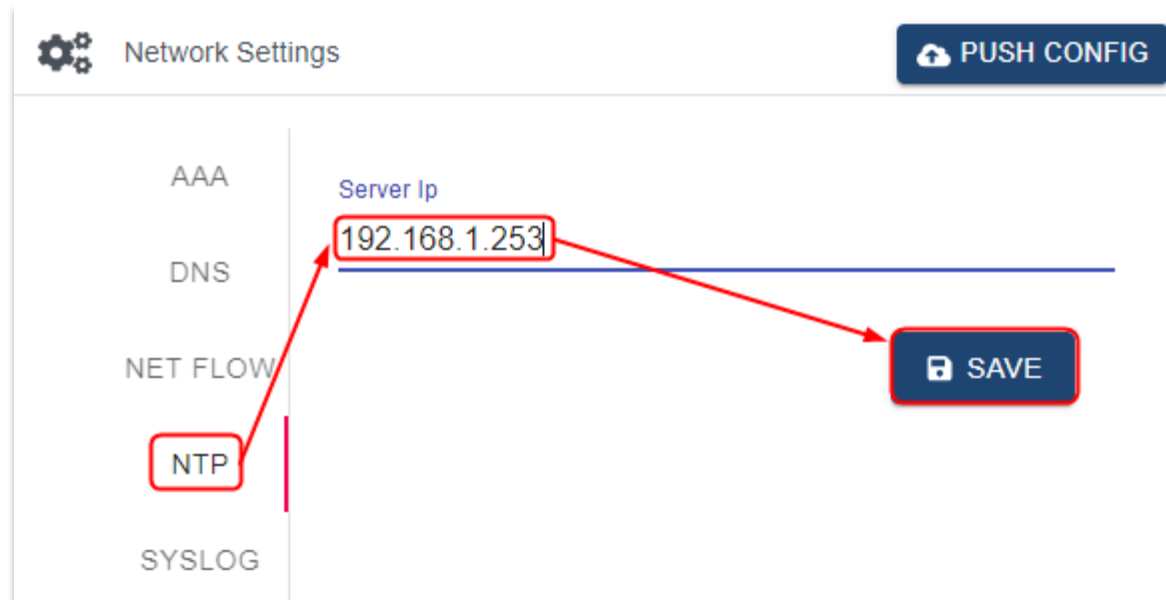
- Pilih **DNS** dan lakukan pengaturan parameter **Domain Name** berupa nama domain yang ingin diterapkan pada perangkat jaringan, sebagai contoh **sabar.com**. Selain itu lakukan pengaturan IP Address berupa alamat IP dari Server DNS yang ingin diterapkan pada perangkat jaringan, sebagai contoh **192.168.1.253**. Tekan tombol **Save** untuk menyimpan pengaturan. Tampil notifikasi “**Saved Successfully**”.



The screenshot shows the 'Network Settings' interface. On the left, there is a vertical menu with options: AAA, DNS, NET FLOW, NTP, and SYSLOG. The 'DNS' option is selected and highlighted with a red box. To the right of the menu, the 'Domain Name' field contains 'sabar.com' and the 'Ip Address' field contains '192.168.1.253'. Both fields are highlighted with red boxes. A red arrow points from the 'DNS' menu item to the 'Domain Name' field, and another red arrow points from the 'Ip Address' field to the 'SAVE' button. The 'SAVE' button is a dark blue button with a white lock icon and the text 'SAVE'. In the top right corner of the settings panel, there is a 'PUSH CONFIG' button with a cloud icon.

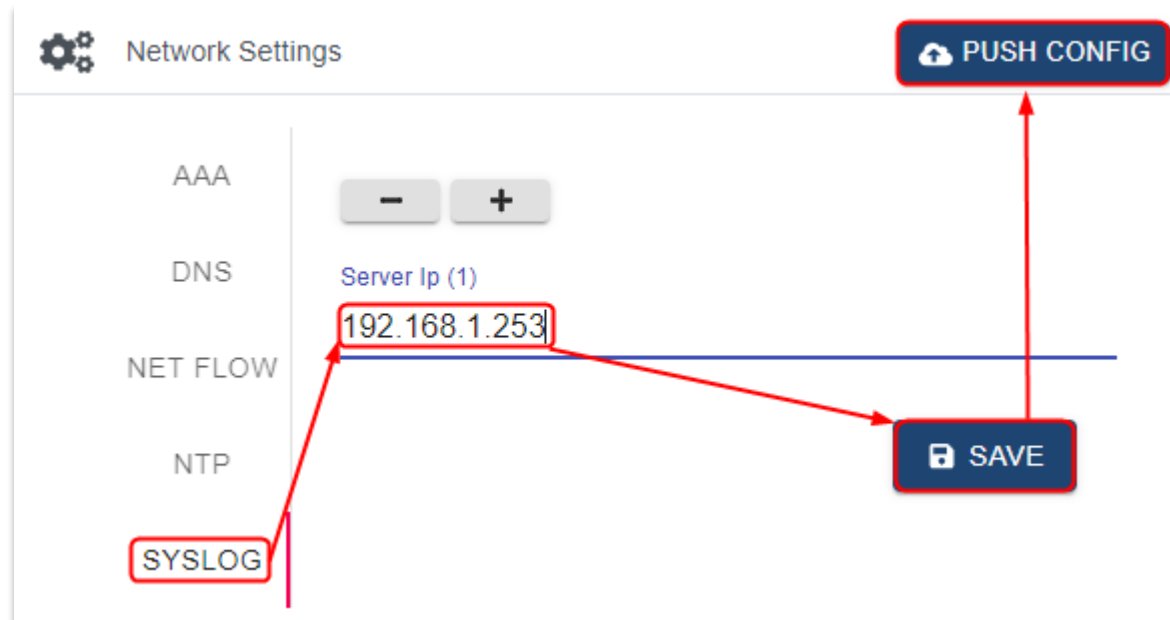
KONFIGURASI JARINGAN DI SDN CONTROLLER

- Pilih **NTP** dan lakukan pengaturan parameter **Server IP** berupa alamat IP dari **Server NTP** yang ingin diterapkan pada perangkat jaringan, sebagai contoh **192.168.1.253**. Tekan tombol **Save** untuk menyimpan pengaturan. Tampil notifikasi “**Saved Successfully**”.



KONFIGURASI JARINGAN DI SDN CONTROLLER

- Pilih **SYSLOG** dan lakukan pengaturan parameter **Server IP** berupa alamat IP dari **Server SYSLOG** yang ingin diterapkan pada perangkat jaringan, sebagai contoh **192.168.1.253**. Tekan tombol **Save** untuk menyimpan pengaturan. Tampil notifikasi “**Saved Successfully**”.



- Tekan tombol **PUSH CONFIG** untuk menerapkan pengaturan **DNS**, **NTP** dan **SYSLOG** tersebut ke perangkat jaringan.

KONFIGURASI JARINGAN DI SDN CONTROLLER

- Tampil kotak dialog **Push All Network Settings** yang menampilkan ringkasan informasi pengaturan konfigurasi jaringan yaitu **DNS**, **NTP** dan **SYSLOG** yang telah dilakukan.
- Klik **OKAY** untuk menindaklanjuti proses **PUSH CONFIG** ke perangkat jaringan.
- Tampil notifikasi “**Saved Successfully**”.

Push All Network Settings

DNS		
Domain Name	192.168.1.253	
Ip Address	sabar.com	
NETFLOW		
Reflection Ip	-	
Port Number	-	
NTP		
Server Ip	192.168.1.253	
SYSLOG		
Server Ip (1)	192.168.1.253	

CANCEL **OKAY**

VERIFIKASI PUSH CONFIG DNS

- Melalui CLI dari **router RTR_MATARAM, RTR_SUMBAWA** dan **RTR_BIMA**, lakukan eksekusi perintah `show run | begin ip domain` pada **mode privilege** untuk memverifikasi **PUSH CONFIG DNS**.

```
RTR_MATARAM#show run | begin ip domain
ip domain-name sabar.com
ip name-server 192.168.1.253
```

```
RTR_SUMBAWA#show run | begin ip domain
ip domain-name sabar.com
ip name-server 192.168.1.253
```

```
RTR_BIMA#show run | begin ip domain
ip domain-name sabar.com
ip name-server 192.168.1.253
```

VERIFIKASI PUSH CONFIG DNS

- Melalui CLI dari **Multilayer Switch MSW_MATARAM, MSW_SUMBAWA, MSW_BIMA**, lakukan eksekusi perintah `show run | begin ip domain` pada **mode privilege** untuk memverifikasi **PUSH CONFIG DNS**.

```
MSW_MATARAM#show run | begin ip domain
ip domain-name sabar.com
ip name-server 192.168.1.253
```

```
MSW_SUMBAWA#show run | begin ip domain
ip domain-name sabar.com
ip name-server 192.168.1.253
```

```
MSW_BIMA#show run | begin ip domain
ip domain-name sabar.com
ip name-server 192.168.1.253
```

VERIFIKASI PUSH CONFIG NTP

- Melalui CLI dari setiap **Router** dan **Multilayer Switch**, lakukan eksekusi perintah berikut pada **mode privilege**:

- a) `show run | include ntp` digunakan untuk memverifikasi **PUSH CONFIG NTP**.
- b) `show ntp associations` digunakan untuk menampilkan informasi status asosiasi NTP.
- c) `show clock` digunakan untuk menampilkan informasi waktu dan tanggal setelah pengaturan NTP.

- Hasil eksekusi di **router RTR_MATARAM**:

```
RTR_MATARAM#show run | include ntp
ntp server 192.168.1.253
RTR_MATARAM#show ntp associations
```

```
address          ref clock      st   when   poll   reach  delay      offset      disp
*~192.168.1.253  .INIT.        0    0      16    377    0.00      0.00      0.12
* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

```
RTR_MATARAM#show clock
14:6:17.664 UTC Tue Dec 28 2021
```

VERIFIKASI PUSH CONFIG NTP

- Hasil eksekusi di router RTR_SUMBAWA:

```
RTR_SUMBAWA#show run | include ntp
ntp server 192.168.1.253
RTR_SUMBAWA#show ntp associations
```

address	ref clock	st	when	poll	reach	delay	offset	disp
*~192.168.1.253	.INIT.	0	4	16	217	4.00	0.00	0.12

* sys.peer, # selected, + candidate, - outlier, x falseticker, ~ configured

```
RTR_SUMBAWA#show clock
10:9:48.462 UTC Tue Dec 28 2021
```

- Hasil eksekusi di router RTR_BIMA:

```
RTR_BIMA#show run | include ntp
ntp server 192.168.1.253
RTR_BIMA#show ntp associations
```

address	ref clock	st	when	poll	reach	delay	offset	disp
*~192.168.1.253	.INIT.	0	3	16	377	95.00	-5.00	0.12

* sys.peer, # selected, + candidate, - outlier, x falseticker, ~ configured

```
RTR_BIMA#show clock
10:10:53.594 UTC Tue Dec 28 2021
```

VERIFIKASI PUSH CONFIG NTP

- Hasil eksekusi di **Multilayer Switch MSW_MATARAM**:

```
MSW_MATARAM#show run | include ntp
```

```
ntp server 192.168.1.253
```

```
MSW_MATARAM#show ntp associations
```

address	ref clock	st	when	poll	reach	delay	offset	disp
*~192.168.1.253	.INIT.	0	0	16	337	0.00	0.00	0.12

* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured

```
MSW_MATARAM#show clock
```

```
10:15:9.44 UTC Tue Dec 28 2021
```

- Hasil eksekusi di **Multilayer Switch MSW_SUMBAWA**:

```
MSW_SUMBAWA#show run | include ntp
```

```
ntp server 192.168.1.253
```

```
MSW_SUMBAWA#show ntp associations
```

address	ref clock	st	when	poll	reach	delay	offset	disp
*~192.168.1.253	.INIT.	0	11	16	377	30.00	-13.00	0.12

* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured

```
MSW_SUMBAWA#show clock
```

```
10:16:6.445 UTC Tue Dec 28 2021
```

VERIFIKASI PUSH CONFIG NTP

- Hasil eksekusi di **Multilayer Switch MSW_BIMA**:

```
MSW_BIMA#show run | include ntp
```

```
ntp server 192.168.1.253
```

```
MSW_BIMA#show ntp associations
```

address	ref clock	st	when	poll	reach	delay	offset	disp
*~192.168.1.253	.INIT.	0	12	16	377	10.00	1.00	0.11

* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured

```
MSW_BIMA#show clock
```

```
10:17:10.842 UTC Tue Dec 28 2021
```


VERIFIKASI PUSH CONFIG SYSLOG

- Melalui CLI dari setiap **Router** dan **Multilayer Switch**, lakukan eksekusi perintah `show run | include logging` pada **mode privilege** untuk memverifikasi **PUSH CONFIG SYSLOG**.

```
RTR_MATARAM#show run | include logging  
logging 192.168.1.253
```

```
MSW_MATARAM#show run | include logging  
logging 192.168.1.253
```

```
RTR_SUMBAWA#show run | include logging  
logging 192.168.1.253
```

```
MSW_SUMBAWA#show run | include logging  
logging 192.168.1.253
```

```
RTR_BIMA#show run | include logging  
logging 192.168.1.253
```

```
MSW_BIMA#show run | include logging  
logging 192.168.1.253
```

PENGUJIAN SYSLOG

- Pengujian **SYSLOG** dilakukan dengan menonaktifkan **interface GigabitEthernet0/0/0** atau **g0/0/0** melalui CLI dari **router RTR_BIMA**.

```
RTR_BIMA#conf t
```

```
Enter configuration commands, one per line. End with CNTL/Z.
```

```
RTR_BIMA(config)#int g0/0/0
```

```
RTR_BIMA(config-if)#shutdown
```

```
RTR_BIMA(config-if)#
```

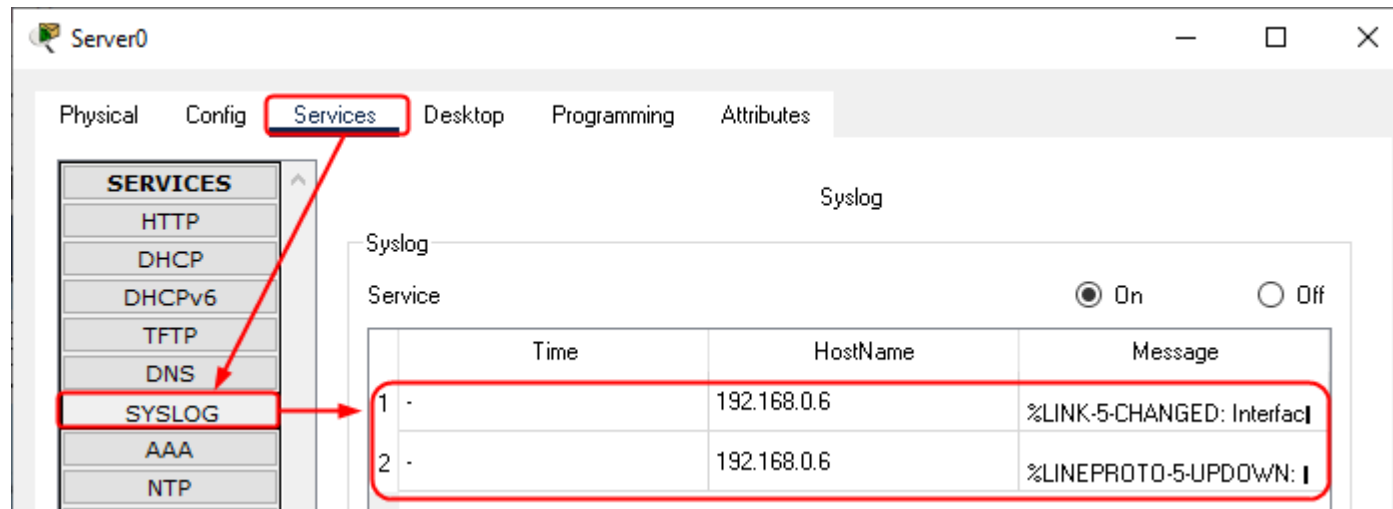
```
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to administratively down
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/0, changed state to down
```

- Terlihat pesan **console** yang menampilkan informasi terkait **interface g0/0/0** yang dinonaktifkan.

PENGUJIAN SYSLOG

- Pesan log yang sama juga akan terlihat pada **Server0** dengan mengakses tab **Services** → **SYSLOG**.



- Klik dua kali pada **entri Message** untuk menampilkan informasi dengan lebih detail.

NETWORK ISSUES DI SDN CONTROLLER

- Pada bagian **Assurance** → **Network Issues** dari **SDN Controller** juga akan menampilkan informasi terkait permasalahan konektivitas dalam menjangkau **Multilayer Switch MSW_BIMA**, **PC E** dan **PC F**. Selain itu pula juga ditampilkan informasi persentase **Network Health** untuk **Switches** sebesar **66%** dan **Hosts** sebesar **75%**.

The screenshot displays the Cisco Network Controller Assurance interface. The top navigation bar includes 'ASSURANCE', 'HOSTS', 'TOPOLOGY', and 'PATH TRACE'. The main content area is divided into three sections:

- Assurance:** Shows 'Routers' with a healthy percentage of 100% and a healthy ratio of 3:3. Below it, 'Switches' are shown with a healthy percentage of 66% and a healthy ratio of 2:3. A pie chart indicates 66% managed (green) and 33% unmanaged (red) switches.
- Network Health:** A line graph shows the health level of 3 routers at 100% and 3 switches at 66% over time. A summary box at the bottom indicates 'Switches 3 Health Level 66%'.
- Network Issues:** A table lists connectivity issues:

Date	Name	Description	Source
2021-12-19 11:12	Connectivity	Device is down	MSW_BIMA
2021-12-19 11:12	Connectivity	Device is down	E
2021-12-19 11:12	Connectivity	Device is down	F

PENGUJIAN SYSLOG

- Lakukan pengaktifan kembali **interface GigabitEthernet0/0/0** atau **g0/0/0** melalui CLI dari **router RTR_BIMA**.

```
RTR_BIMA#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
RTR_BIMA(config)#int g0/0/0
RTR_BIMA(config-if)#no shutdown

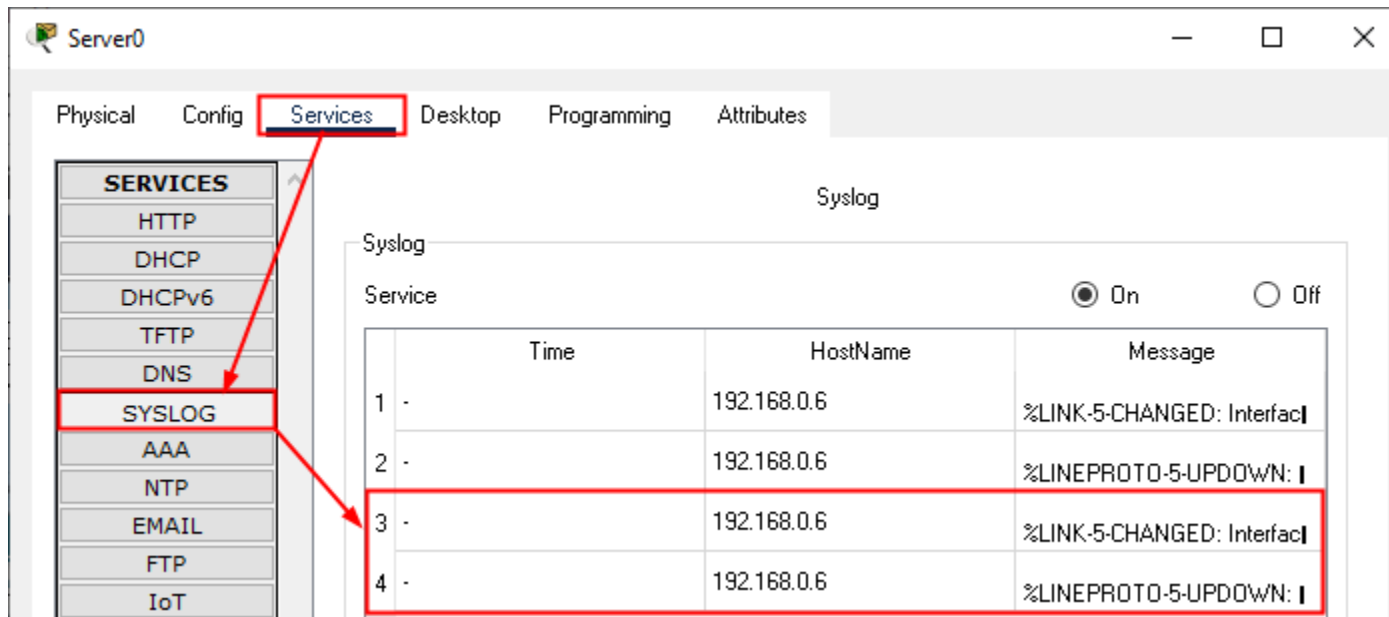
RTR_BIMA(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/0, changed state to up
```

- Terlihat pesan **console** yang menampilkan informasi terkait **interface g0/0/0** yang diaktifkan.

PENGUJIAN SYSLOG

- Pesan log yang sama juga akan terlihat pada **Server0** dengan mengakses tab **Services** → **SYSLOG**.



- Klik dua kali pada **entri Message** untuk menampilkan informasi dengan lebih detail.
- Tutup kotak dialog **Server0**.

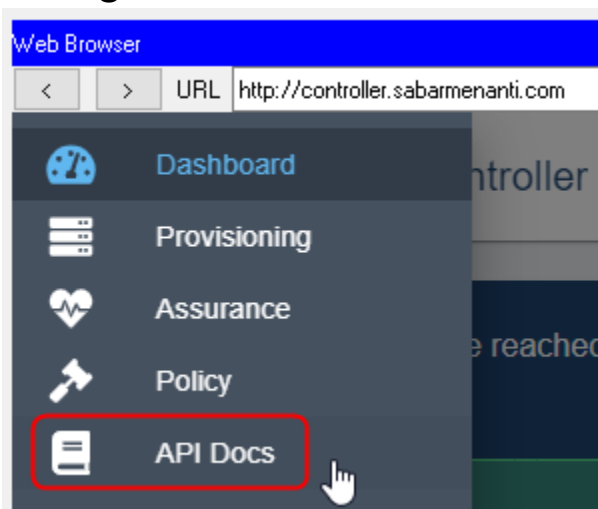


IMPLEMENTASI REST API DENGAN SDN CONTROLLER

- Dokumentasi REST API Network Controller
- Request Token Otentikasi (Service Ticket)
- Python Script Untuk Memperoleh Service Ticket Dari SDN Controller
- Python Script Untuk Mengambil Network Health Di SDN Controller
- Python Script Untuk Mengambil Network Issues Di SDN Controller
- Python Script Untuk Mengirim Notifikasi Network Issues Ke Cisco Webex Room
- Python Script Untuk Mengirim Notifikasi Network Issues Ke Telegram

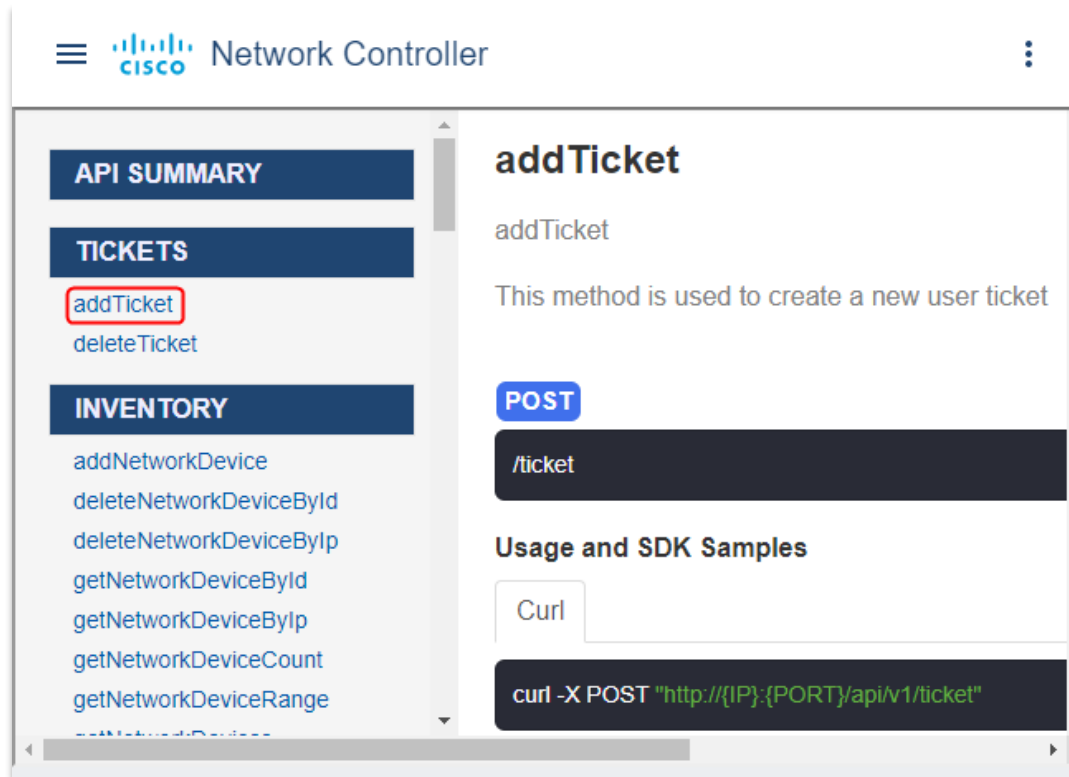
DOKUMENTASI REST API NETWORK CONTROLLER

- API memungkinkan pengguna untuk mengakses **Network Controller** menggunakan antarmuka REST melalui HTTP. Dokumentasi **REST API** untuk **Network Controller** dapat diakses melalui **web browser** dari salah satu PC, sebagai contoh **PC A**. Pada inputan **URL** masukkan alamat <http://controller.sabarmenanti.com> dan tekan tombol **Go**.
- **Login** menggunakan **Username sabar** dengan **Password menanti**. Tampil halaman **Dashboard**. Pada bagian *header*, pilih menu di sebelah kiri dari **logo Cisco** dan pilih menu **API Docs**. Selanjutnya akan tampil halaman **API and SDK Documentation** untuk **Network Controller**. Lakukan eksplorasi menggunakan panel navigasi sebelah kiri.



REQUEST TOKEN OTENTIKASI (SERVICE TICKET)

- Melalui dokumentasi **API**, klik pada **addTicket** sebagai metode yang digunakan untuk membuat **ticket** baru bagi pengguna. **Ticket** tersebut nantinya digunakan sebagai token otentikasi untuk mengirim permintaan (**request**) **REST** berikutnya ke **Network Controller**.

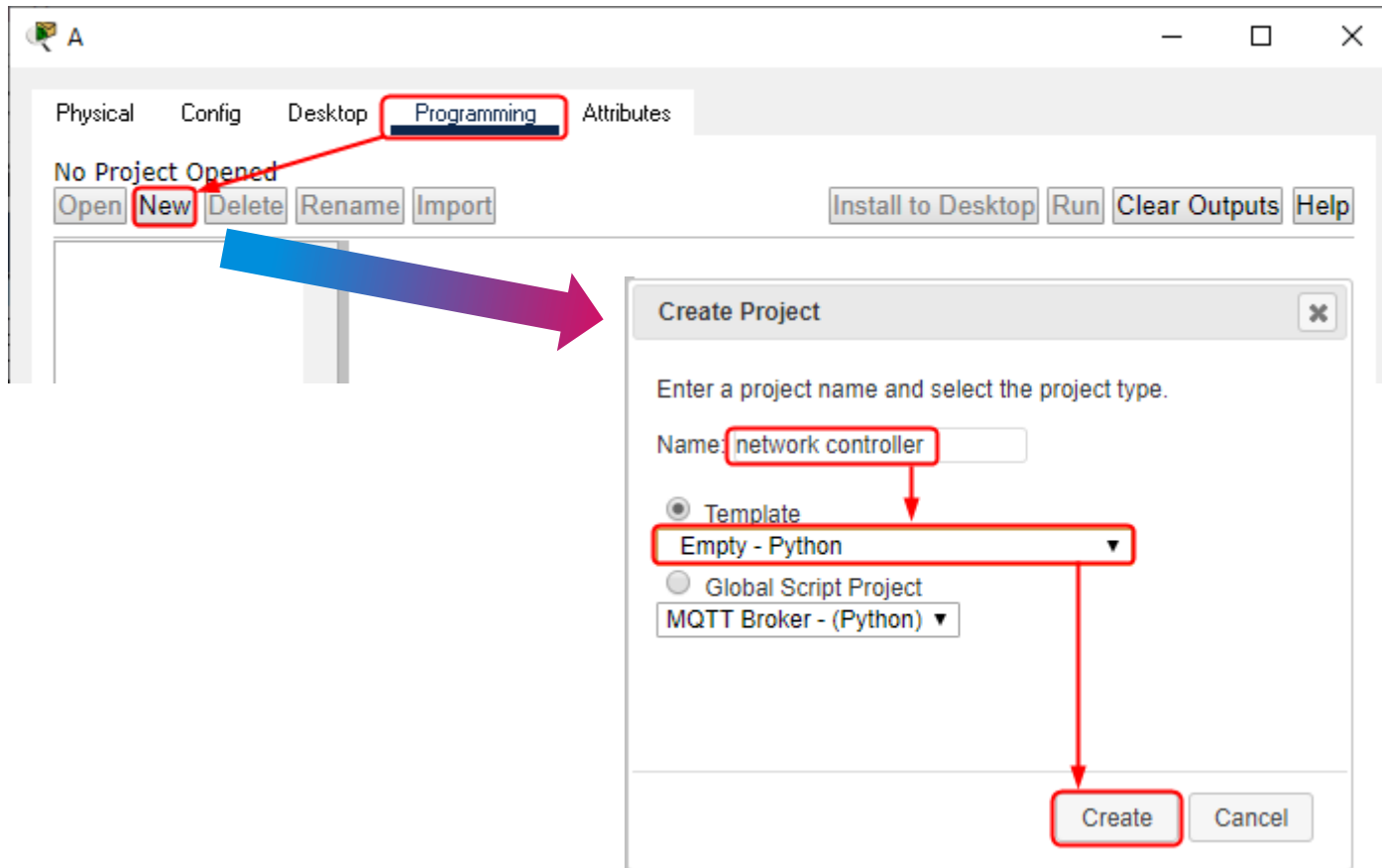


The screenshot shows the Cisco Network Controller API documentation. The left sidebar contains a navigation menu with sections: API SUMMARY, TICKETS, and INVENTORY. Under the TICKETS section, the 'addTicket' method is highlighted with a red box. The main content area displays the 'addTicket' method details, including its description: 'This method is used to create a new user ticket'. Below the description, the HTTP method is listed as 'POST' with a blue background. The endpoint is shown as '/ticket'. Under the 'Usage and SDK Samples' section, there is a 'Curl' tab and a code block containing the command: `curl -X POST "http://{IP}:{PORT}/api/v1/ticket"`.

Method ini memiliki 2 (dua) parameter yaitu **username** dan **password** untuk mengakses ke **Network Controller**. Pada studi kasus ini **username** menggunakan **sabar** dan **password** menggunakan **menanti**.

PYTHON SCRIPT UNTUK MEMPEROLEH SERVICE TICKET DARI SDN CONTROLLER

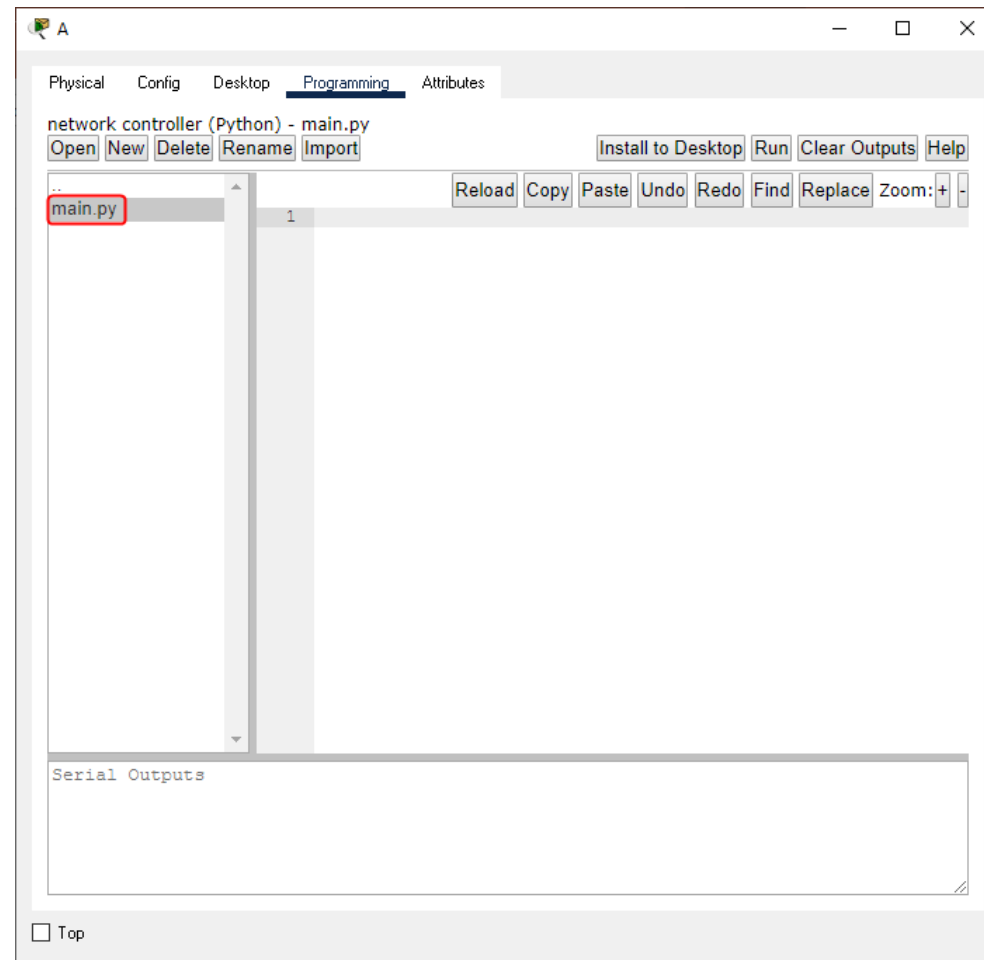
- Pada kotak dialog properties dari **PC A**, pilih tab **Programming**. Klik tombol **New** untuk membuat **Project**.



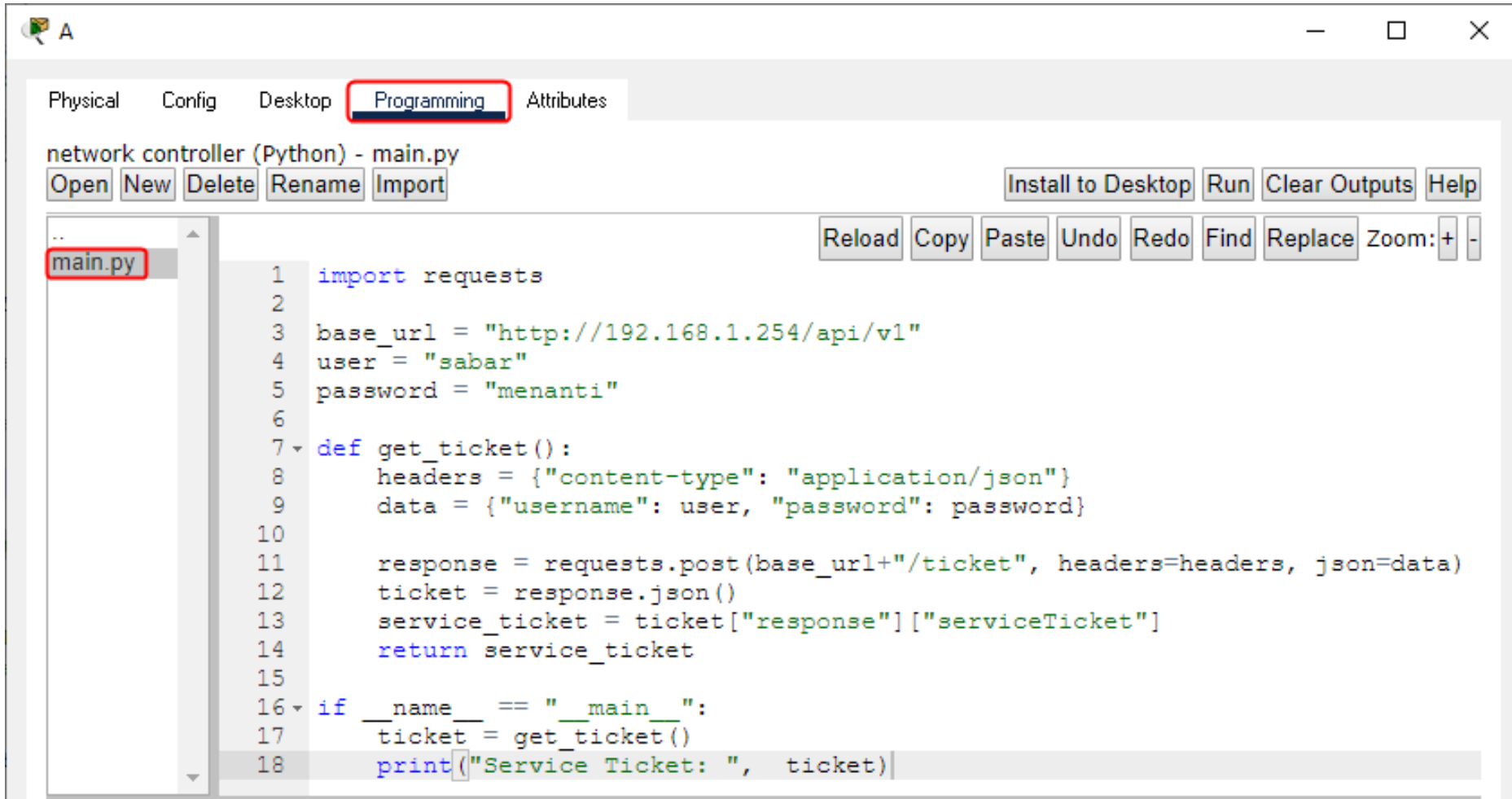
Tampil kotak dialog **Create Project**. Pada inputan **Name:**, masukkan nama **project**, sebagai contoh **network controller**. Sedangkan di bagian **Template**, pilih **Empty - Python**. Tekan tombol **Create** untuk memproses pembuatan *project*.

PYTHON SCRIPT UNTUK MEMPEROLEH SERVICE TICKET DARI SDN CONTROLLER

- Klik dua kali pada file **main.py** yang telah tersedia setelah pembuatan *project*.



PYTHON SCRIPT UNTUK MEMPEROLEH SERVICE TICKET DARI SDN CONTROLLER



```
1 import requests
2
3 base_url = "http://192.168.1.254/api/v1"
4 user = "sabar"
5 password = "menanti"
6
7 def get_ticket():
8     headers = {"content-type": "application/json"}
9     data = {"username": user, "password": password}
10
11     response = requests.post(base_url+"/ticket", headers=headers, json=data)
12     ticket = response.json()
13     service_ticket = ticket["response"]["serviceTicket"]
14     return service_ticket
15
16 if __name__ == "__main__":
17     ticket = get_ticket()
18     print("Service Ticket: ", ticket)
```

Ketik kode program **Python** berikut pada file **main.py** yang digunakan untuk mengambil **service ticket** dari **Network Controller**. Kode program juga dapat diambil dari **Github** (<https://github.com/iputuhariyadi/sdn>).

PYTHON SCRIPT UNTUK MEMPEROLEH SERVICE TICKET DARI SDN CONTROLLER

Penjelasan baris kode program:

- Baris 1 digunakan untuk melakukan `import modul requests` yang dibutuhkan ketika mengirimkan permintaan (**request**) API ke **Network Controller**.
- Baris 3 memuat deklarasi **variable base_url** dengan nilai "<http://192.168.1.254/api/v1>" yang merupakan **Uniform Resource Locator (URL)** untuk mengidentifikasi sumber daya yang dapat dimanipulasi oleh *client* di **Network Controller**. URL tersebut terdiri dari:
 - ✓ **Scheme** yaitu **HTTP**.
 - ✓ **Authority** yaitu **192.168.1.254** yang merupakan alamat IP dari **Network Controller**.
 - ✓ **Path** yang merepresentasikan lokasi dari sumber daya yang akan dimanipulasi pada server yaitu **/api/v1**.
- Baris 4 dan 5 memuat deklarasi **variable user** dengan nilai "**sabar**" dan **password** dengan nilai "**menanti**" yang digunakan untuk melalui proses otentikasi *login* ke **Network Controller**.

PYTHON SCRIPT UNTUK MEMPEROLEH SERVICE TICKET DARI SDN CONTROLLER

- Baris 7 memuat deklarasi pembuatan fungsi dengan nama **get_ticket**. Fungsi tersebut digunakan untuk mengirimkan **API request** dengan **method POST** dengan parameter berupa **url** dari API untuk memperoleh **service ticket Network Controller** yaitu <https://192.168.1.254/api/v1/ticket>.
- Sedangkan parameter **headers** memuat nilai yang diambil dari deklarasi **variable headers** pada baris 8 yaitu **key Content-type** bernilai **application/json** yang mengindikasikan format **request body** yang dikirim dari API Client yaitu **JSON**.
- Selain itu parameter **json** memuat nilai yang diambil dari deklarasi **variable data** pada baris 9 untuk akun otentikasi login ke **Network Controller**. **Variable data** terdiri dari 2 (dua) **key** yaitu **username** dengan nilai yang diambil dari **variable user** dan **key password** dengan nilai yang diambil dari **variable password**.
- Baris 11 digunakan untuk mengirimkan **API request** dengan **method POST** dengan parameter berupa **base_url**, **headers** dan **json** serta hasilnya disimpan pada **variable response**.

PYTHON SCRIPT UNTUK MEMPEROLEH SERVICE TICKET DARI SDN CONTROLLER

- Baris 12 digunakan untuk mengembalikan objek JSON dari nilai yang terkandung di dalam **variable response** dan hasilnya disimpan pada **variable ticket**.
- Baris 13 digunakan untuk mengambil nilai **serviceTicket** yang terdapat dalam **variable ticket** dan hasilnya disimpan pada **service_ticket**.
- Baris 14: **keyword return** berfungsi untuk mengembalikan nilai dari fungsi ketika dipanggil berupa nilai dari **variable service_ticket**.
- Baris 16 memuat kondisi yang digunakan untuk mengeksekusi kode di dalam pernyataan **if** hanya ketika program dijalankan langsung oleh **interpreter Python**.
- Baris 17 digunakan untuk memanggil fungsi **get_ticket** dan hasilnya disimpan pada **variable ticket**.
- Baris 18 digunakan untuk menampilkan isi dari **variable ticket** menggunakan fungsi **print()**.

PYTHON SCRIPT UNTUK MEMPEROLEH SERVICE TICKET DARI SDN CONTROLLER



```
network controller (Python) - main.py
Open New Delete Rename Import Install to Desktop Run Clear Outputs Help
Reload Copy Paste Undo Redo Find Replace Zoom: + -
main.py
1 import requests
2
3 base_url = "http://192.168.1.254/api/v1"
4 user = "sabar"
5 password = "menanti"
6
7 def get_ticket():
8     headers = {"content-type": "application/json"}
9     data = {"username": user, "password": password}
10
11     response = requests.post(base_url+"/ticket", headers=headers, json=data)
12     ticket = response.json()
13     service_ticket = ticket["response"]["serviceTicket"]
14     return service_ticket
15
16 if __name__ == "__main__":
17     ticket = get_ticket()
18     print("Service Ticket: ", ticket)
```

Starting network controller (Python)...
('Service Ticket: ', 'NC-89-e2d836bca5a94fca9d08-nbi')
network controller (Python) finished running.

OUTPUT

- Klik tombol **Run** untuk mengeksekusi kode program Python.
- Terlihat pada bagian **OUTPUT**, skrip telah berhasil dieksekusi dan menampilkan nilai **Service Ticket**.

PYTHON SCRIPT UNTUK MENGAMBIL NETWORK HEALTH DI SDN CONTROLLER

```
1 import requests
2
3 base_url = "http://192.168.1.254/api/v1"
4 user = "sabar"
5 password = "menanti"
6
7 def get_ticket():
8     headers = {"content-type": "application/json"}
9     data = {"username": user, "password": password}
10
11     response = requests.post(base_url+"/ticket", headers=headers, json=data)
12     ticket = response.json()
13     service_ticket = ticket["response"]["serviceTicket"]
14     return service_ticket
15
16 def get_network_health():
17     ticket = get_ticket()
18     headers = {"X-Auth-Token": ticket}
19     response = requests.get(base_url+"/assurance/health", headers=headers)
20     health = response.json()
21     network_health = health['response'][0]['networkDevices']['totalPercentage']
22     return network_health
23
24 if __name__ == "__main__":
25     network_health = get_network_health()
26     print("Persentase Network Health: "+ network_health +"%")
```

- Pengambilan informasi **Network Health** pada **Network Controller** dapat dilakukan dengan menyesuaikan kode program **Python** pada **file main.py** yaitu membuat deklarasi fungsi **get_network_health** seperti terlihat pada **baris 16-22** dan menyesuaikan instruksi yang dieksekusi oleh kondisi **if** pada **baris 25 dan 26**.

PYTHON SCRIPT UNTUK MENGAMBIL NETWORK HEALTH DI SDN CONTROLLER

Penjelasan baris kode program:

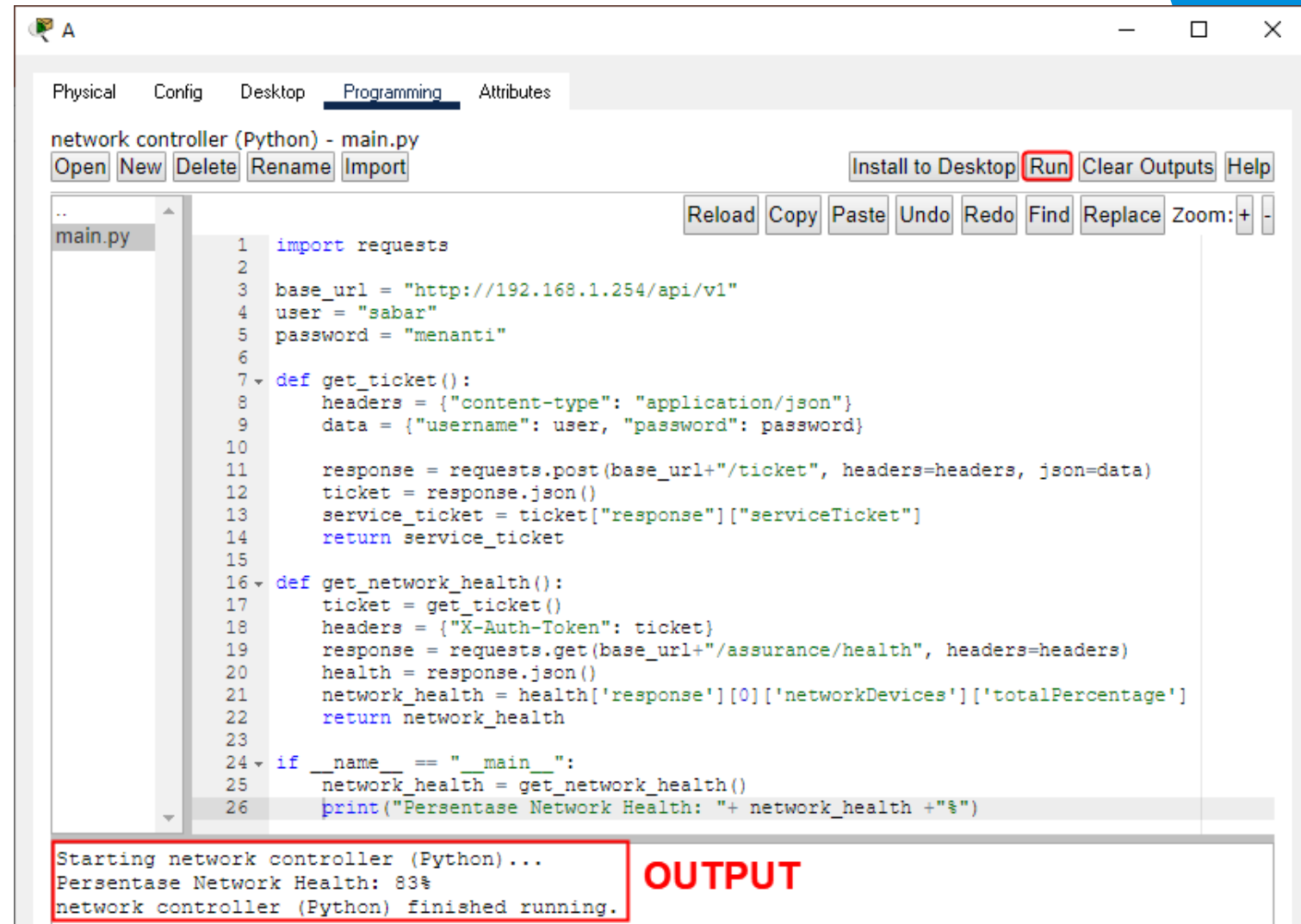
- Baris 16 sampai dengan 22 memuat deklarasi pembuatan fungsi dengan nama **get_network_health**.
- Baris 17 digunakan untuk memanggil fungsi **get_ticket** sehingga menghasilkan **ticket** otentikasi **Network Controller** dan hasilnya disimpan pada **variable ticket**.
- Baris 18 memuat deklarasi **variable headers** dengan **key x-auth-token** dengan **value** atau nilai yang diambil dari **variable ticket**.
- Baris 19 digunakan untuk mengirimkan **API request** dengan **method GET** dengan parameter berupa **base_url** dengan **path /assurance/health** untuk mengambil keseluruhan informasi **Network Health** dari **Network Controller** dan **headers** serta hasilnya disimpan pada **variable response**.
- Baris 20 digunakan untuk mengembalikan objek JSON dari nilai yang terkandung di dalam **variable response** dan hasilnya disimpan pada **variable health**.

PYTHON SCRIPT UNTUK MENGAMBIL NETWORK HEALTH DI SDN CONTROLLER

- Baris 21 digunakan untuk mengambil nilai **persentase Network Health** untuk kategori perangkat jaringan (**networkDevices**) yang terkandung di dalam **variable health** dan hasilnya disimpan pada variable **network_health**.
- Baris 22 digunakan untuk mengatur nilai **variable network_health** sebagai nilai balik dari pemanggilan fungsi **get_network_health**.
- Baris 24 memuat kondisi yang digunakan untuk mengeksekusi kode di dalam pernyataan **if** hanya ketika program dijalankan langsung oleh **interpreter Python**.
- Baris 25 digunakan untuk memanggil fungsi **get_network_health** dan hasilnya disimpan pada **variable network_health**.
- Baris 26 digunakan untuk menampilkan isi dari **variable network_health** menggunakan fungsi **print()**.

PYTHON SCRIPT UNTUK MENGAMBIL NETWORK HEALTH DI SDN CONTROLLER

- Klik tombol **Run** untuk mengeksekusi kode program Python.
- Terlihat pada bagian **OUTPUT**, skrip telah berhasil dieksekusi dan menampilkan nilai **Persentase Network Health sebesar 83%**.



```
network controller (Python) - main.py
Open New Delete Rename Import Install to Desktop Run Clear Outputs Help
Reload Copy Paste Undo Redo Find Replace Zoom: + -

..
main.py
1 import requests
2
3 base_url = "http://192.168.1.254/api/v1"
4 user = "sabar"
5 password = "menanti"
6
7 def get_ticket():
8     headers = {"content-type": "application/json"}
9     data = {"username": user, "password": password}
10
11     response = requests.post(base_url+"/ticket", headers=headers, json=data)
12     ticket = response.json()
13     service_ticket = ticket["response"]["serviceTicket"]
14     return service_ticket
15
16 def get_network_health():
17     ticket = get_ticket()
18     headers = {"X-Auth-Token": ticket}
19     response = requests.get(base_url+"/assurance/health", headers=headers)
20     health = response.json()
21     network_health = health['response'][0]['networkDevices']['totalPercentage']
22     return network_health
23
24 if __name__ == "__main__":
25     network_health = get_network_health()
26     print("Persentase Network Health: "+ network_health +"%")

Starting network controller (Python)...
Persentase Network Health: 83%
network controller (Python) finished running.
```

OUTPUT

PYTHON SCRIPT UNTUK MENGAMBIL NETWORK ISSUES DI SDN CONTROLLER

- Pengambilan informasi **Network Issues** pada **Network Controller** dapat dilakukan dengan menyesuaikan kode program **Python** pada file **main.py** yaitu membuat deklarasi fungsi **get_network_issues** seperti terlihat pada **baris 24-38** dan menyesuaikan instruksi yang dieksekusi oleh kondisi **if** pada **baris 41 dan 46**.

```
24 def get_network_issues():
25     ticket = get_ticket()
26     headers = {'Accept': 'application/json', 'X-Auth-Token': ticket}
27     issues = requests.get(url = base_url + "/assurance/health-issues", headers=headers)
28     issue_details = issues.json()
29     devices = len(issue_details['response'])
30     output = "Peringatan! Terjadi gangguan akses ke "+ str(devices) + " perangkat berikut:\n"
31     output += "-"*60 + "\n"
32     output += "NO. | PERANGKAT | WAKTU | DESKRIPSI\n"
33     output += "-"*60 + "\n"
34     number=1
35     for device in issue_details['response']:
36         output += ""+ str(number) + ". | "+ device['issueSource'] + " | "+ device['issueTimestamp'] + " | "+ device['issueDescription'] + "\n"
37         number +=1
38     return output
39
40 if __name__ == "__main__":
41     network_health = get_network_health()
42     if int(network_health) < 100:
43         issues = get_network_issues()
44         print(issues)
45     else:
46         print("Persentase Network Health: "+ network_health +"%")
```

PYTHON SCRIPT UNTUK MENGAMBIL NETWORK ISSUES DI SDN CONTROLLER

Penjelasan baris kode program:

- Baris 24 sampai dengan 38 memuat deklarasi pembuatan fungsi dengan nama **get_network_issues**.
- Baris 25 digunakan untuk memanggil fungsi **get_ticket** sehingga menghasilkan **ticket** otentikasi **Network Controller** dan hasilnya disimpan pada **variable ticket**.
- Baris 26 memuat deklarasi **variable headers** yang terdiri dari 2 (dua) **key** yaitu **key Accept** dengan nilai **application/json** yang mengatur jenis luaran yang diminta sebagai balasan dari **API Server** yaitu **JSON** dan **key x-auth-token** dengan nilai yang diambil dari **variable ticket**.
- Baris 27 digunakan untuk mengirimkan **API request** dengan **method GET** dengan parameter berupa **base_url** dengan **path /assurance/health-issues** untuk mengambil keseluruhan informasi permasalahan (**issues**) terkait **Network Health** dari **Network Controller** dan parameter **headers** serta hasilnya disimpan pada **variable issues**.
- Baris 28 digunakan untuk mengembalikan objek **JSON** dari nilai yang terkandung di dalam **variable issues** dan hasilnya disimpan pada **variable issue_details**.

PYTHON SCRIPT UNTUK MENGAMBIL NETWORK ISSUES DI SDN CONTROLLER

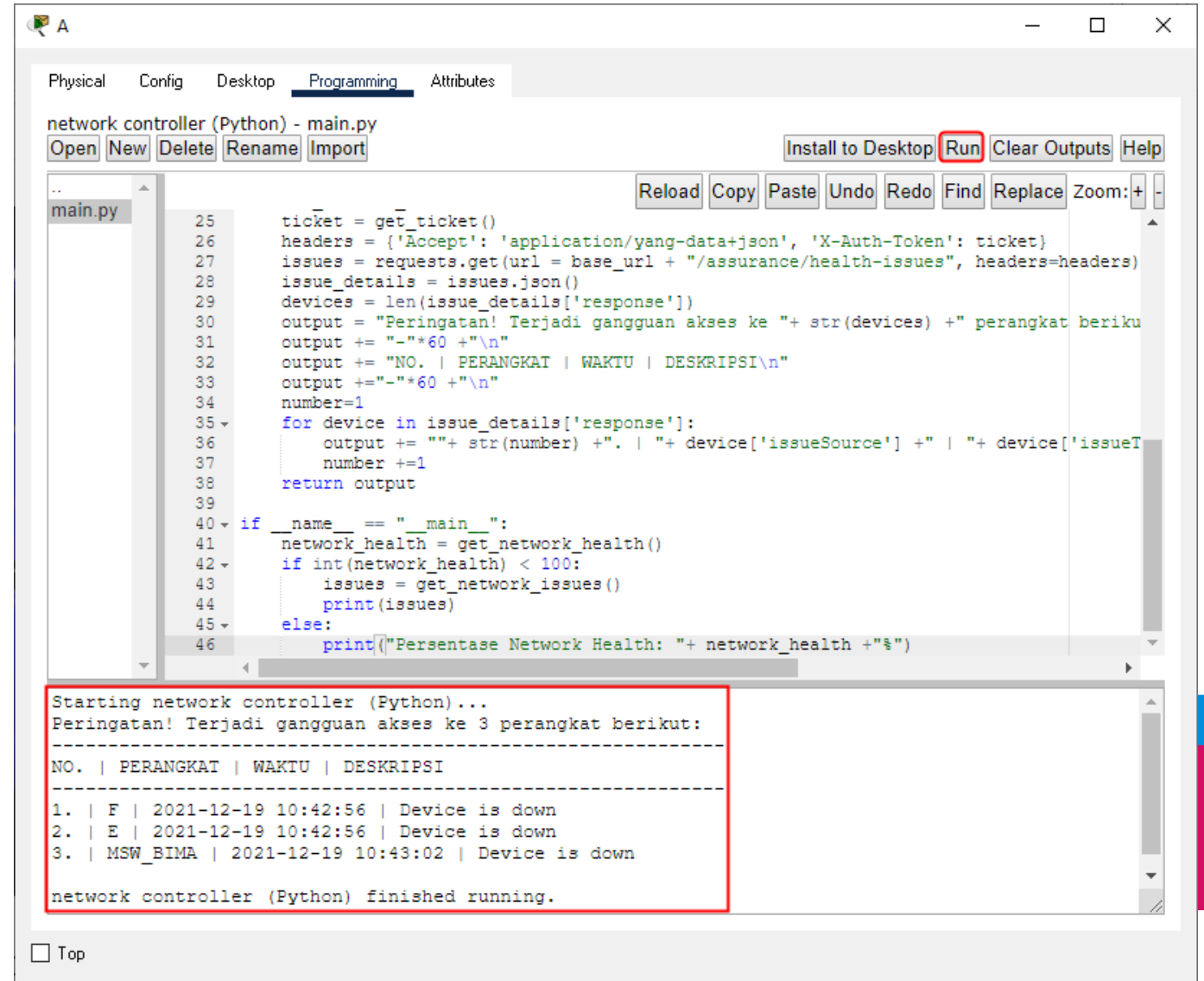
- Baris 29 digunakan untuk mengambil informasi jumlah perangkat yang mengalami **Network Issues** dan menyimpan hasilnya pada **variable devices**.
- Baris 30 sampai 37 digunakan untuk memformat penampilan data perangkat yang mengalami **Network Issues** dan menyimpan hasilnya pada **variable output**.
- Baris 38 digunakan untuk mengatur nilai **variable output** sebagai nilai balik dari pemanggilan fungsi **get_network_issues**.
- Baris 40 memuat kondisi yang digunakan untuk mengeksekusi kode di dalam pernyataan **if** hanya ketika program dijalankan langsung oleh **interpreter Python**.
- Baris 41 digunakan untuk memanggil fungsi **get_network_health** dan hasilnya disimpan pada **variable network_health**.

PYTHON SCRIPT UNTUK MENGAMBIL NETWORK ISSUES DI SDN CONTROLLER

- Baris 42 sampai 46 memuat kondisi yang digunakan untuk mengeksekusi kode di dalam pernyataan **if** hanya ketika nilai dari **variable network_health** lebih kecil dari **100** yaitu memanggil fungsi **get_network_issue** pada baris 43 dan menyimpan hasilnya pada **variable issues** serta menampilkan isi variable tersebut menggunakan fungsi **print()** pada baris 44. Sebaliknya akan mengeksekusi kode di dalam pernyataan **else** yaitu menampilkan isi dari **variable network_health** menggunakan fungsi **print()** pada baris 46.

PYTHON SCRIPT UNTUK MENGAMBIL NETWORK ISSUES DI SDN CONTROLLER

- Klik tombol **Run** untuk mengeksekusi kode program **Python**.
- Terlihat pada bagian **OUTPUT**, skrip telah berhasil dieksekusi dan menampilkan informasi terkait **Network Issues** yaitu gangguan akses ke 3 (tiga) perangkat meliputi **PC E**, **F** dan **MSW_BIMA**.



```
network controller (Python) - main.py
Open New Delete Rename Import Install to Desktop Run Clear Outputs Help
Reload Copy Paste Undo Redo Find Replace Zoom: + -
main.py
25 ticket = get_ticket()
26 headers = {'Accept': 'application/yang-data+json', 'X-Auth-Token': ticket}
27 issues = requests.get(url = base_url + "/assurance/health-issues", headers=headers)
28 issue_details = issues.json()
29 devices = len(issue_details['response'])
30 output = "Peringatan! Terjadi gangguan akses ke "+ str(devices) +" perangkat beriku
31 output += "-"*60 +"\n"
32 output += "NO. | PERANGKAT | WAKTU | DESKRIPSI\n"
33 output += "-"*60 +"\n"
34 number=1
35 for device in issue_details['response']:
36     output += ""+ str(number) +". | "+ device['issueSource'] +" | "+ device['issueT
37     number +=1
38 return output
39
40 if __name__ == "__main__":
41     network_health = get_network_health()
42     if int(network_health) < 100:
43         issues = get_network_issues()
44         print(issues)
45     else:
46         print("Persentase Network Health: "+ network_health +"%")

Starting network controller (Python)...
Peringatan! Terjadi gangguan akses ke 3 perangkat berikut:
-----
NO. | PERANGKAT | WAKTU | DESKRIPSI
-----
1. | F | 2021-12-19 10:42:56 | Device is down
2. | E | 2021-12-19 10:42:56 | Device is down
3. | MSW_BIMA | 2021-12-19 10:43:02 | Device is down
network controller (Python) finished running.
```

PYTHON SCRIPT UNTUK MENGIKIRIM NOTIFIKASI NETWORK ISSUES KE CISCO WEBEX ROOM

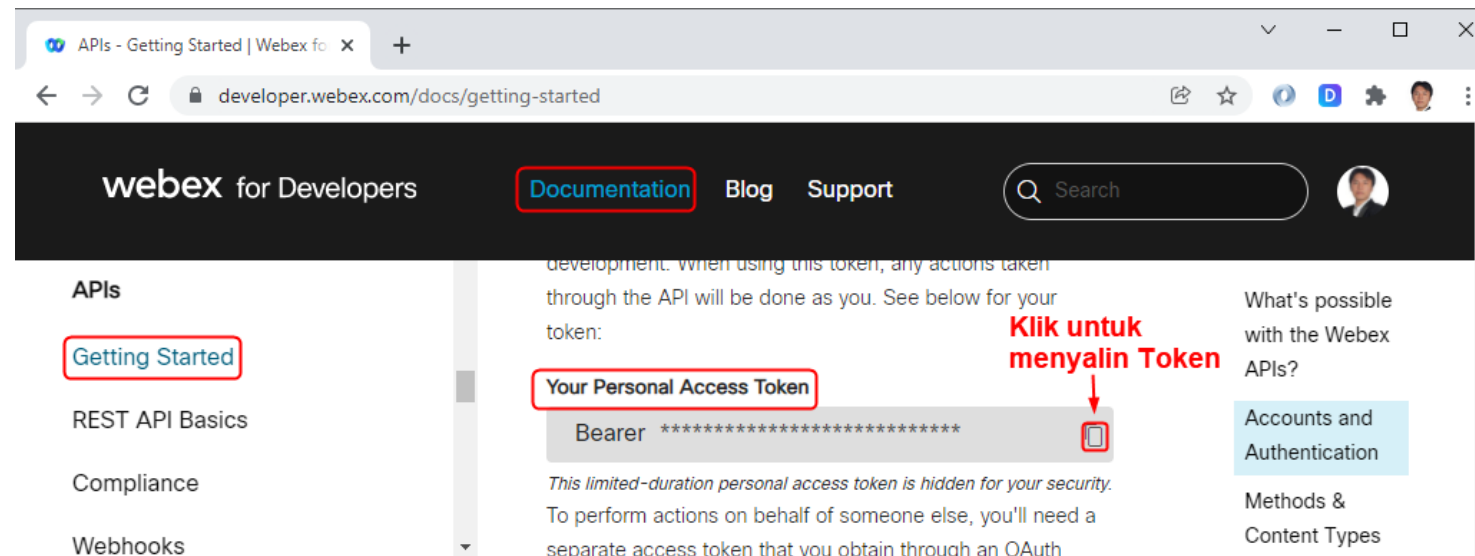
- Pengiriman notifikasi terkait **Network Issues** pada **Network Controller** ke **Cisco Webex** dapat dilakukan dengan menyesuaikan kode program **Python** pada **file main.py**. Terdapat 4 (empat) bagian yang memerlukan penyesuaian yaitu:
- Bagian pertama mengimport seluruh fungsi pada **library time**, seperti terlihat pada baris 2 dan mengimport module **realhttp** agar dapat melakukan akses ke jaringan eksternal yaitu **Cisco Webex** dan **Telegram**, seperti terlihat pada baris 3.
- Bagian kedua mendeklarasikan **variable** terkait komunikasi ke **Cisco Webex**, seperti terlihat pada baris 9 sampai dengan 11.

```
1 import requests
2 from time import *
3 import realhttp
4
5 base_url = "http://192.168.1.254/api/v1"
6 user = "sabar"
7 password = "menanti"
8
9 webex_url = "https://webexapis.com/v1/messages"
10 accessToken = "Your Access Token"
11 roomId = "Your Room ID"
12
13 def get_ticket():
```

PYTHON SCRIPT UNTUK MENGIKIRIM NOTIFIKASI NETWORK ISSUES KE CISCO WEBEX ROOM

Terdapat 3 (tiga) **variable** yang dideklarasikan yaitu:

- Variable **webex_url** dengan nilai <https://webexapis.com/v1/messages> yang merupakan **API URL** untuk mengirim pesan ke **Webex room**,
- Variable **accessToken** yang menampung **Token Cisco Webex**. **Token** dapat diperoleh dengan mengakses situs developer.webex.com dan **login** menggunakan akun **Cisco Webex** yang dimiliki. Apabila telah berhasil **login**, pilih menu **Documentation** kemudian pada panel menu sebelah kiri pilih **Getting Started** maka akan tampil halaman **Getting Started with the Webex API**. *Scroll* ke bawah ke bagian **Your Personal Access Token**. Salinkan **Token** tersebut sebagai nilai dari *variable* ini.



PYTHON SCRIPT UNTUK MENGIKIRIM NOTIFIKASI NETWORK ISSUES KE CISCO WEBEX ROOM

- c) Variable **roomId** yang menampung **Room ID** pada **Cisco Webex** sebagai tujuan pengiriman pesan. **Room ID** dari **room** yang dimiliki pada **Cisco Webex** dapat diperoleh dengan mengakses URL <https://developer.webex.com/docs/api/v1/rooms/list-rooms>. Kemudian klik tombol **Run** maka di bagian **Response** akan terlihat **id** dari **room** yang ada. Lakukan penyalinan nilai dari **id** dari salah satu **Room** tersebut sebagai nilai dari **variable** ini.

Sebagai contoh terlihat cuplikan **room id** dari **NETCOM Charity** yaitu **Y2....**

Pastikan melakukan penyesuaian nilai **variable accessToken** dan **roomId** dari **Cisco Webex** sesuai dengan yang dimiliki.

The screenshot shows the Cisco Webex Developer API documentation for the 'List Rooms' endpoint. The browser address bar is highlighted with a red box, showing the URL: `developer.webex.com/docs/api/v1/rooms/list-rooms`. The page content includes a list of API endpoints for rooms, with the 'List Rooms' endpoint (GET) highlighted. A 'Run' button is visible, and the 'Response' section shows a JSON object with an 'id' field highlighted in red, containing the value 'Y21zY29zcGFyazovL3Vy'.

PYTHON SCRIPT UNTUK MENGIKIRIM NOTIFIKASI NETWORK ISSUES KE CISCO WEBEX ROOM

- Bagian ketiga membuat fungsi dengan nama **onHTTPDone** dengan 3 (tiga) parameter yaitu **status**, **data** dan **replyHeader**, seperti terlihat pada baris 46 sampai dengan 50.

```
46- def onHTTPDone(status, data, replyHeader):
47-     if status == 200:
48-         print("Pesan Network Issues sukses dikirim!")
49-     else:
50-         print("Pesan Network Issues gagal dikirim!")
51
52- if __name__ == "__main__":
53-     network_health = get_network_health()
54-     if int(network_health) < 100:
55-         issues = get_network_issues()
56-         print(issues)
57-         http = realhttp.RealHTTPClient()
58-
59-         headers = {"Authorization":"Bearer "+accessToken, "Content-Type":"application/json"}
60-         message = {"roomId":roomId, "markdown":'***Permasalahan Jaringan :**'+'\n'+ '>'+ issues}
61-         http.postWithHeader(webex_url, message, headers)
62-
63-         http.onDone(onHTTPDone)
64-         while True:
65-             sleep(5)
66-     else:
67-         print("Persentase Network Health: "+ network_health +"%")
```

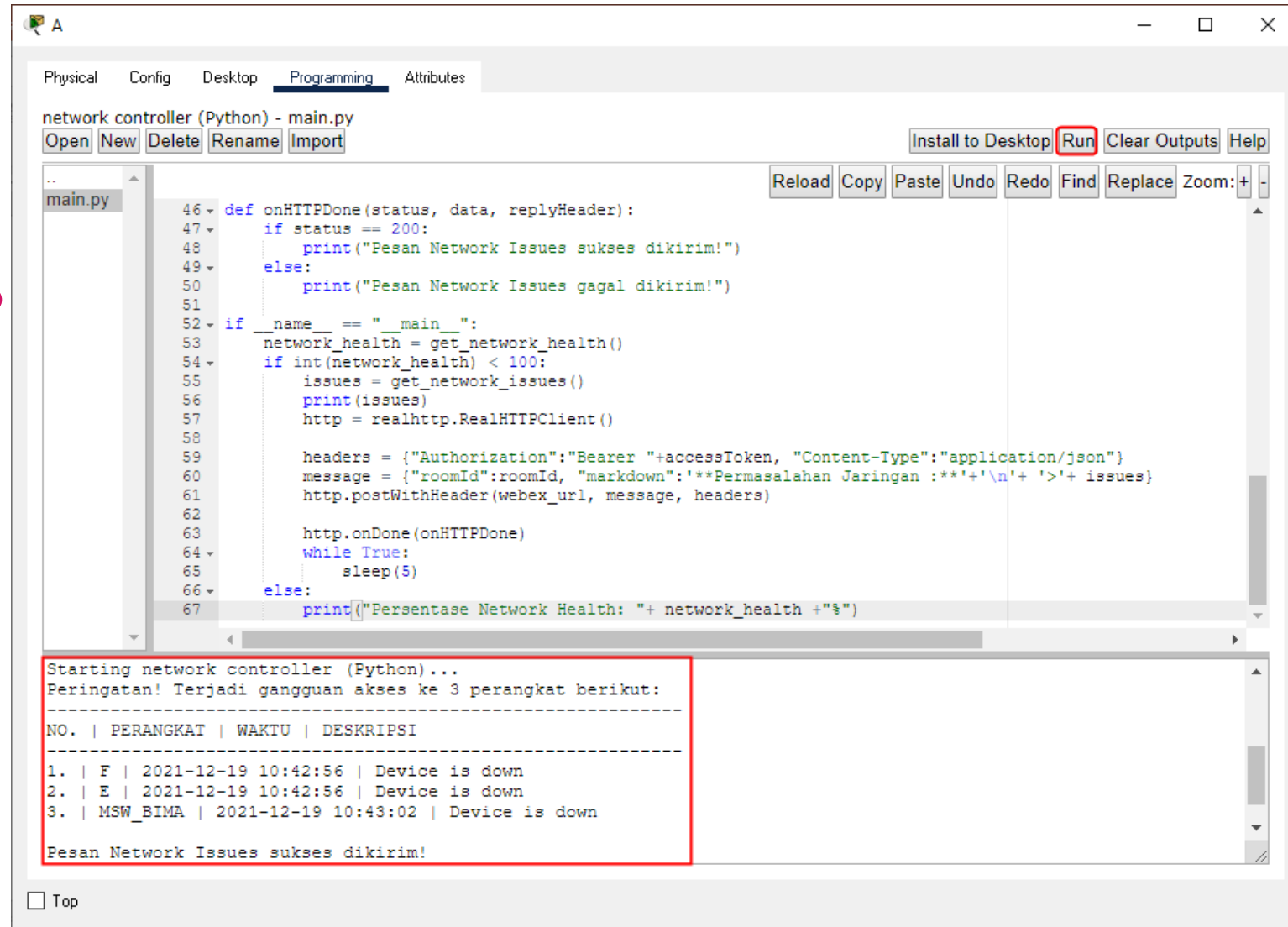
Fungsi ini akan dipanggil menggunakan method **http.onDone** sebagai **callback** ketika permintaan HTTP telah selesai dieksekusi, seperti terlihat pada **baris 63**. Variable **status** memuat respon berupa **HTTP Status Code**. Jika bernilai **200** maka bermakna **OK** atau sukses.

PYTHON SCRIPT UNTUK MENGIKIRIM NOTIFIKASI NETWORK ISSUES KE CISCO WEBEX ROOM

- Bagian ke-empat membuat kode program terkait **Real HTTP Client** dan **API Request ke Cisco Webex**, seperti terlihat pada baris 57 sampai dengan 65.
- Baris 57 digunakan untuk membuat **instance Real HTTP Client** dan hasilnya disimpan pada variable **http**.
- Baris 59 memuat deklarasi **variable headers** yang terdiri dari 2 (dua) **key** yaitu **key Authorization** dengan nilai berupa prefix “**Bearer** “ diikuti dengan nilai yang diambil dari **variable accessToken** dan **key Accept** bernilai **application/json** yang mengatur jenis luaran yang diminta sebagai balasan dari **API Server** yaitu **JSON**.
- Baris 60 memuat deklarasi **variable message** yang terdiri dari 2 (dua) **key** yaitu **key roomId** dengan nilai yang diambil dari **variable roomId** dan **key markdown** dengan nilai berupa pesan menggunakan prefix “**Permasalahan Jaringan**” diikuti dengan pesan peringatan serta nilai yang diambil dari **variable issues**.
- Baris 61 digunakan untuk mengirimkan **API request** dengan **method POST** sehingga dapat mengirimkan pesan ke **Cisco Webex Room** tertentu dan memiliki 3 (tiga) parameter dimana setiap nilainya diambil masing-masing dari **variable webex_url**, **message** dan **headers**.
- Baris 64 dan 65 digunakan untuk menanggunghkan eksekusi sesuai dengan jumlah detik yang diberikan yaitu 5 detik.

PYTHON SCRIPT NOTIFIKASI NETWORK ISSUES KE CISCO WEBEX

- Klik tombol **Run** untuk mengeksekusi kode program **Python**.
- Terlihat pada bagian **OUTPUT**, skrip telah berhasil dieksekusi dan menampilkan informasi terkait **Network Issues** yaitu gangguan akses ke 3 (tiga) perangkat meliputi **PC E, F** dan **MSW_BIMA** serta **Pesan Network Issues sukses dikirim ke Cisco Webex Room**.



The screenshot shows a Python IDE window titled "network controller (Python) - main.py". The code in the editor is as follows:

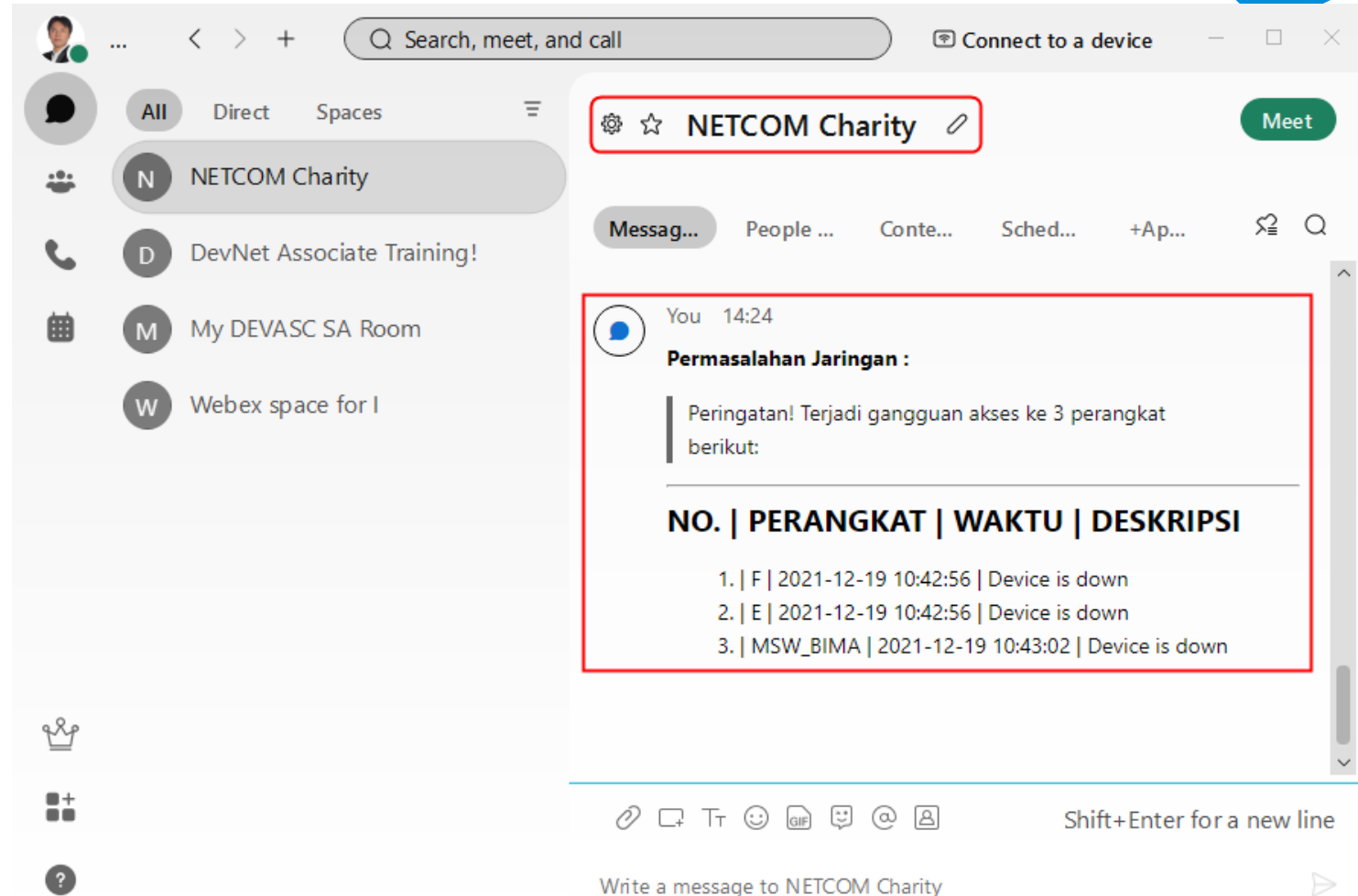
```
46 def onHTTPDone(status, data, replyHeader):
47     if status == 200:
48         print("Pesan Network Issues sukses dikirim!")
49     else:
50         print("Pesan Network Issues gagal dikirim!")
51
52 if __name__ == "__main__":
53     network_health = get_network_health()
54     if int(network_health) < 100:
55         issues = get_network_issues()
56         print(issues)
57         http = realhttp.RealHTTPClient()
58
59         headers = {"Authorization": "Bearer "+accessToken, "Content-Type": "application/json"}
60         message = {"roomId": roomId, "markdown": "***\n'+ '>'+ issues}
61         http.postWithHeader(webex_url, message, headers)
62
63         http.onDone(onHTTPDone)
64         while True:
65             sleep(5)
66     else:
67         print("Persentase Network Health: "+ network_health +"%")
```

The terminal output window shows the following text:

```
Starting network controller (Python)...
Peringatan! Terjadi gangguan akses ke 3 perangkat berikut:
-----
NO. | PERANGKAT | WAKTU | DESKRIPSI
-----
1. | F | 2021-12-19 10:42:56 | Device is down
2. | E | 2021-12-19 10:42:56 | Device is down
3. | MSW_BIMA | 2021-12-19 10:43:02 | Device is down
-----
Pesan Network Issues sukses dikirim!
```

PYTHON SCRIPT NOTIFIKASI NETWORK ISSUES KE CISCO WEBEX ROOM

- Hasil verifikasi menggunakan **Cisco Webex Desktop for Windows** memperlihatkan keberhasilan penerimaan pesan notifikasi terkait **Network Issues** yang dikirim menggunakan skrip **Python** dari dalam **Cisco Packet Tracer** ke **Webex Room NETCOM Charity**.



The screenshot shows a Cisco Webex chat interface. The chat room is named "NETCOM Charity". A message from "You" at 14:24 is displayed, containing a network issue notification. The notification text is as follows:

Permasalahan Jaringan :

Peringatan! Terjadi gangguan akses ke 3 perangkat berikut:

NO.	PERANGKAT	WAKTU	DESKRIPSI
1.	F	2021-12-19 10:42:56	Device is down
2.	E	2021-12-19 10:42:56	Device is down
3.	MSW_BIMA	2021-12-19 10:43:02	Device is down

PYTHON SCRIPT UNTUK MENGIKIRIM NOTIFIKASI NETWORK ISSUES KE TELEGRAM

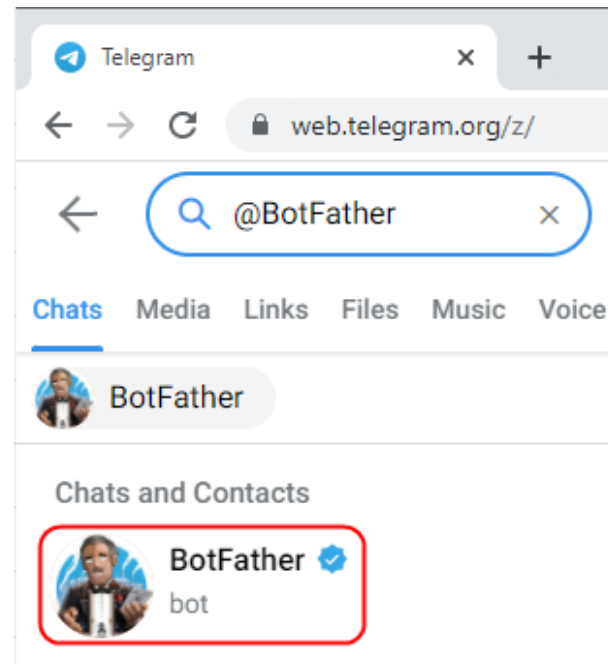
- Pengiriman notifikasi terkait **Network Issues** pada **Network Controller** ke **Telegram** dapat dilakukan dengan menyesuaikan kode program **Python** pada **file main.py**. Terdapat 4 (empat) bagian yang memerlukan penyesuaian yaitu:
- Bagian pertama mendeklarasikan **variable** terkait komunikasi ke **Telegram**, seperti terlihat pada baris 13 dan 14.

```
1 import requests
2 from time import *
3 import realhttp
4
5 base_url = "http://192.168.1.254/api/v1"
6 user = "sabar"
7 password = "menanti"
8
9 webex_url = "https://webexapis.com/v1/messages"
10 accessToken = "Your Access Token"
11 roomId = "Your Room ID"
12
13 bot_token = "Your Telegram Bot Token"
14 bot_chatID = "Your Telegram Bot Chat ID"
15
16 def get_ticket():
```

PYTHON SCRIPT UNTUK MENGIKIRIM NOTIFIKASI NETWORK ISSUES KE TELEGRAM

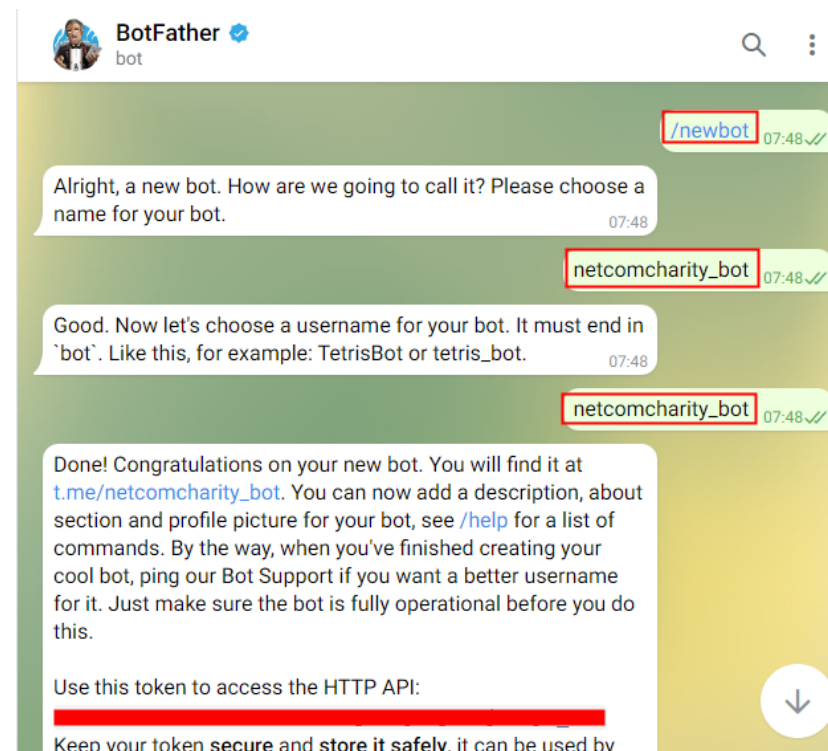
Terdapat 2 (dua) **variable** yang dideklarasikan yaitu:

- Variable bot_token** yang menampung **Token** dari **Bot Telegram** yang telah dibuat. Pembuatan **bot** dapat dilakukan dengan mengakses **Telegram Web Client** pada alamat <https://web.telegram.org> dan menggunakan **@BotFather**.



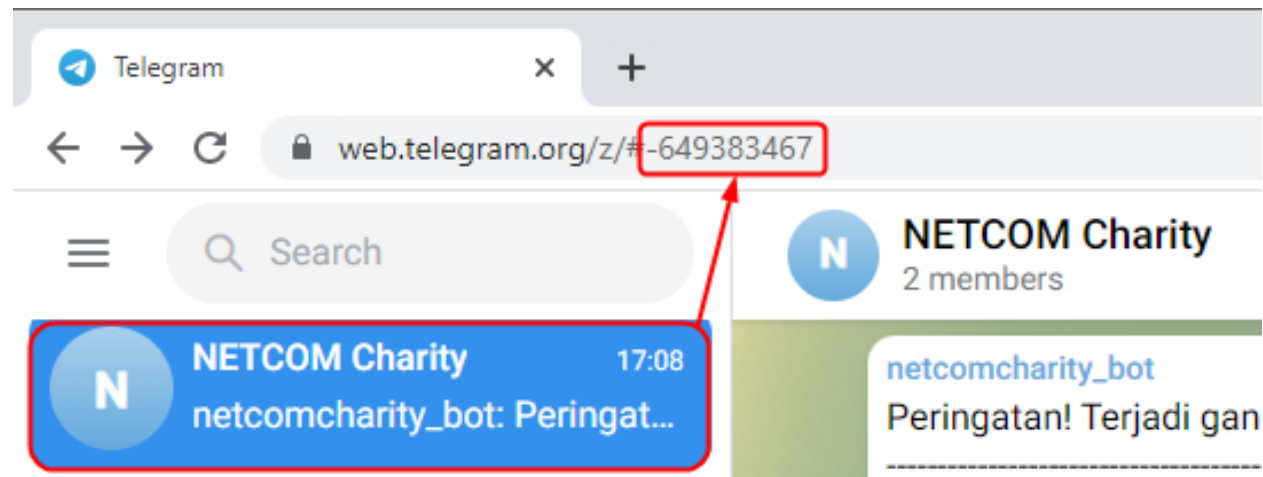
PYTHON SCRIPT UNTUK MENGIKIRIM NOTIFIKASI NETWORK ISSUES KE TELEGRAM

Ketik perintah **/newbot**. Muncul respon yang meminta untuk memasukkan **nama bot** dan **username bot**, sebagai contoh masukkan **netcomcharity_bot**. Apabila proses pembuatan **bot** berhasil dilakukan maka muncul respon yang menginformasikan terkait **token** untuk akses API dari **bot** yang dibuat. Salinkan **Token** tersebut sebagai nilai dari **variable bot_token** ini.



PYTHON SCRIPT UNTUK MENGIKIRIM NOTIFIKASI NETWORK ISSUES KE TELEGRAM

- b) **Variable bot_chatID** yang menampung nomor identifikasi (ID) dari **chat group** sebagai tujuan pengiriman pesan. Salah satu cara untuk mengetahui **chat ID** dari **group chat Telegram** tertentu adalah dengan mengakses **Telegram Web Client** dengan alamat <https://web.telegram.org> dan memilih **chat group** yang ingin diketahui **chat ID**-nya. Sebagai contoh cuplikan dari **chat ID** yang dimiliki oleh **group chat NETCOM Charity** yaitu **-649383467**. Salinkan **chat ID** tersebut sebagai nilai dari **variable bot_chatID**.



PYTHON SCRIPT UNTUK MENGIKIRIM NOTIFIKASI NETWORK ISSUES KE TELEGRAM

- Bagian kedua, membuat fungsi dengan nama **escape_underscore** dengan satu parameter yaitu **txt** yang digunakan untuk mengganti tanda **underscore** dengan **dash** pada pesan yang dikirim melalui **Telegram**, pada baris 56 sampai dengan 57. Apabila tidak diganti maka pengiriman pesan akan mengalami kegagalan.

```
56 def escape_underscore(txt):
57     return txt.replace("_", "-")
58
59 if __name__ == "__main__":
60     network_health = get_network_health()
61     if int(network_health) < 100:
62         issues = get_network_issues()
63         print(issues)
64         http = realhttp.RealHTTPClient()
65
66         #headers = {"Authorization": "Bearer "+accessToken, "Content-Type": "application/json"}
67         #message = {"roomId": roomId, "markdown": '**Permasalahan Jaringan :**'+'\n'+>'+ issues}
68         #http.postWithHeader(webex_url, message, headers)
69
70         issues = escape_underscore(issues)
71         send_text = 'https://api.telegram.org/bot' + bot_token + '/sendMessage?chat_id=' + bot_chatID + '&parse_mode=Markdown&text=' + issues
72         http.get(send_text)
73
74         http.onDone (onHTTPDone)
75     while True:
76         sleep(5)
77 else:
78     print("Persentase Network Health: "+ network_health +"%")
```

- Bagian ketiga, menonaktifkan kode program terkait **Cisco Webex** mulai baris 66 sampai dengan 68 dengan menambahkan tanda **#** diawal setiap baris tersebut sehingga dianggap sebagai komentar.

PYTHON SCRIPT UNTUK MENGIKIRIM NOTIFIKASI NETWORK ISSUES KE TELEGRAM

- Bagian ke-empat, membuat kode program untuk melakukan **API Request** ke **Telegram** sehingga dapat mengirimkan pesan ke **group chat** tertentu, mulai baris 70 sampai dengan 72.
- Baris 70 digunakan untuk mengganti tanda **underscore** dengan tanda **dash** pada nilai dari **variable issues** melalui pemanggilan fungsi **escape_underscore** dan menyimpan hasilnya kembali ke **variable issues**.
- Baris 71 memuat deklarasi **variable send_text** dengan nilai yang merupakan **URL** untuk mengidentifikasi sumber daya yang dimanipulasi oleh client di **Telegram**. **URL** tersebut memiliki format penulisan sebagai berikut:

```
https://api.telegram.org/bot$token/sendMessage?chat_id=$chatId&parse_mode=Markdown&text=$message
```

- a) **https://api.telegram.org/** merupakan **API Server** dari **Telegram**.
- b) **bot\$token/sendMessage** merupakan sumber daya (**resources**) yang akan dimanipulasi pada server. Ganti **\$token** dengan nilai **token** dari **bot** yang telah dibuat. Sebagai contoh di kode program **Python** yang dibuat nilai **token** disimpan pada **variable bot_token** yang telah dideklarasikan pada baris 13.

PYTHON SCRIPT UNTUK MENGIKIRIM NOTIFIKASI NETWORK ISSUES KE TELEGRAM

- c) `?chat_id=$chatId&parse_mode=Markdown&text=$message` merupakan **query** yang memiliki 3 (tiga) parameter yaitu **chat_id** yang digunakan untuk menampung **ID** dari **group chat Telegram** dan **parse_mode** bernilai **Markdown** untuk pemformatan pesan serta **text** yang digunakan untuk menampung pesan yang akan dikirim.
- Ganti `$chatId` dengan nilai **ID** dari **group chat** yang telah dibuat. Sebagai contoh di kode program **Python** yang dibuat nilai **chatId** disimpan pada **variable** `bot_chatID` yang telah dideklarasikan pada baris 14.
- Ganti `$message` dengan pesan yang akan dikirim. Sebagai contoh di kode program **Python** yang dibuat nilai **message** disimpan pada **variable** `issues` yang telah dideklarasikan pada baris 70.
- Baris 72 digunakan untuk mengirimkan **API request** dengan method **GET** untuk mengirimkan pesan ke **group chat Telegram** tertentu dan memiliki satu parameter yaitu **send_text**.

PYTHON SCRIPT UNTUK MENGIRIM NOTIFIKASI NETWORK ISSUES KE TELEGRAM

- Klik tombol **Run** untuk mengeksekusi kode program **Python**.
- Terlihat pada bagian **OUTPUT**, skrip telah berhasil dieksekusi dan menampilkan informasi terkait **Network Issues** yaitu gangguan akses ke 3 (tiga) perangkat meliputi **PC E, F** dan **MSW_BIMA** serta **Pesan Network Issues sukses dikirim ke Telegram**.



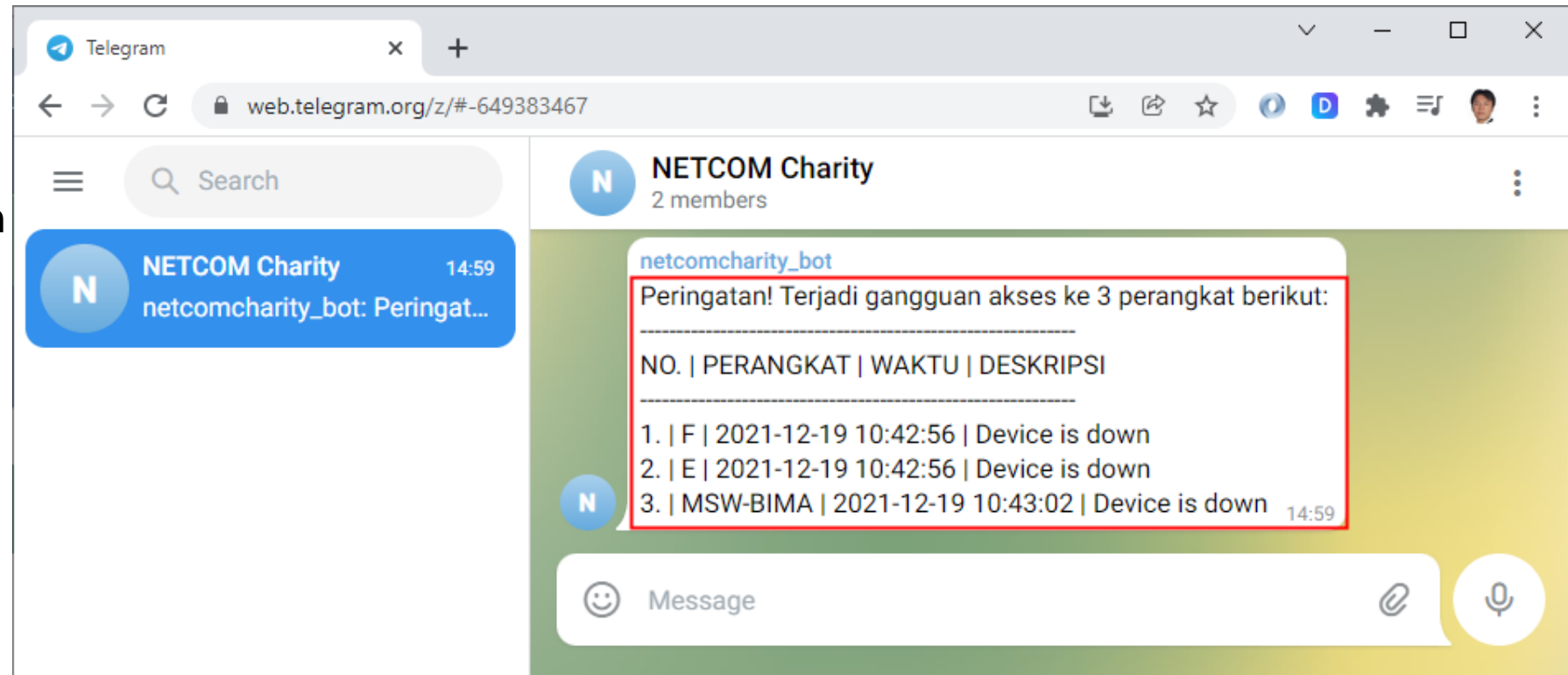
```
network controller (Python) - main.py
Open New Delete Rename Import
Install to Desktop Run Clear Outputs Help
Reload Copy Paste Undo Redo Find Replace Zoom: + -

..
main.py
58
59 if __name__ == "__main__":
60     network_health = get_network_health()
61     if int(network_health) < 100:
62         issues = get_network_issues()
63         print(issues)
64         http = realhttp.RealHTTPClient()
65
66         #headers = {"Authorization": "Bearer "+accessToken, "Content-Type": "application/json"}
67         #message = {"roomId": roomId, "markdown": '**Permasalahan Jaringan :**'+'\n'+>'+ issues
68         #http.postWithHeader(webex_url, message, headers)
69
70         issues = escape_underscore(issues)
71         send_text = 'https://api.telegram.org/bot' + bot_token + '/sendMessage?chat_id=' + bot_c
72         http.get(send_text)
73
74         http.onDone(onHTTPDone)
75         while True:
76             sleep(5)
77     else:
78         print("Persentase Network Health: "+ network_health +"%")

Starting network controller (Python)...
Peringatan! Terjadi gangguan akses ke 3 perangkat berikut:
-----
NO. | PERANGKAT | WAKTU | DESKRIPSI
-----
1. | F | 2021-12-19 10:42:56 | Device is down
2. | E | 2021-12-19 10:42:56 | Device is down
3. | MSW_BIMA | 2021-12-19 10:43:02 | Device is down
-----
Pesan Network Issues sukses dikirim!
```


PYTHON SCRIPT NOTIFIKASI NETWORK ISSUES KE TELEGRAM

- Hasil verifikasi menggunakan **Telegram Web Client** memperlihatkan keberhasilan penerimaan pesan notifikasi terkait **Network Issues** yang dikirim menggunakan skrip **Python** dari dalam **Cisco Packet Tracer** ke **Group NETCOM Charity**.





ADA PERTANYAAN?

REFERENSI

- Kamal Benzekki, Abdeslam El Fergougui, Abdelbaki Elbelrhiti Elalaoui. (2017). Software-defined networking (SDN): a survey. *Security and Communication Networks*
- NetworkLessons.com, Introduction to SDN (Software Defined Networking), <https://networklessons.com/cisco/ccna-routing-switching-icnd2-200-105/introduction-to-sdn-software-defined-networking>
- Cisco, What Is a Network Controller?, <https://www.cisco.com/c/en/us/solutions/enterprise-networks/what-is-a-network-controller.html>
- Cisco Networking Academy, Cisco Packet Tracer 8.1.0 Frequently Asked Questions, <https://www.netacad.com/sites/default/files/cisco-packet-tracer-faq.pdf>
- Cisco Networking Academy, DevNet Associate v1.0, <https://www.netacad.com>
- Cisco Packet Tracer Python API Documentation
- Webex for Developer, <https://developer.webex.com/>
- Telegram Bot API, <https://core.telegram.org/bots/api>

TERIMAKASIH

