NETWORK
FUNDAMENTALS

Dasar jaringan
OSI layer
Perangkat jaringan
Ip address
Ethernet cable
Subnetting

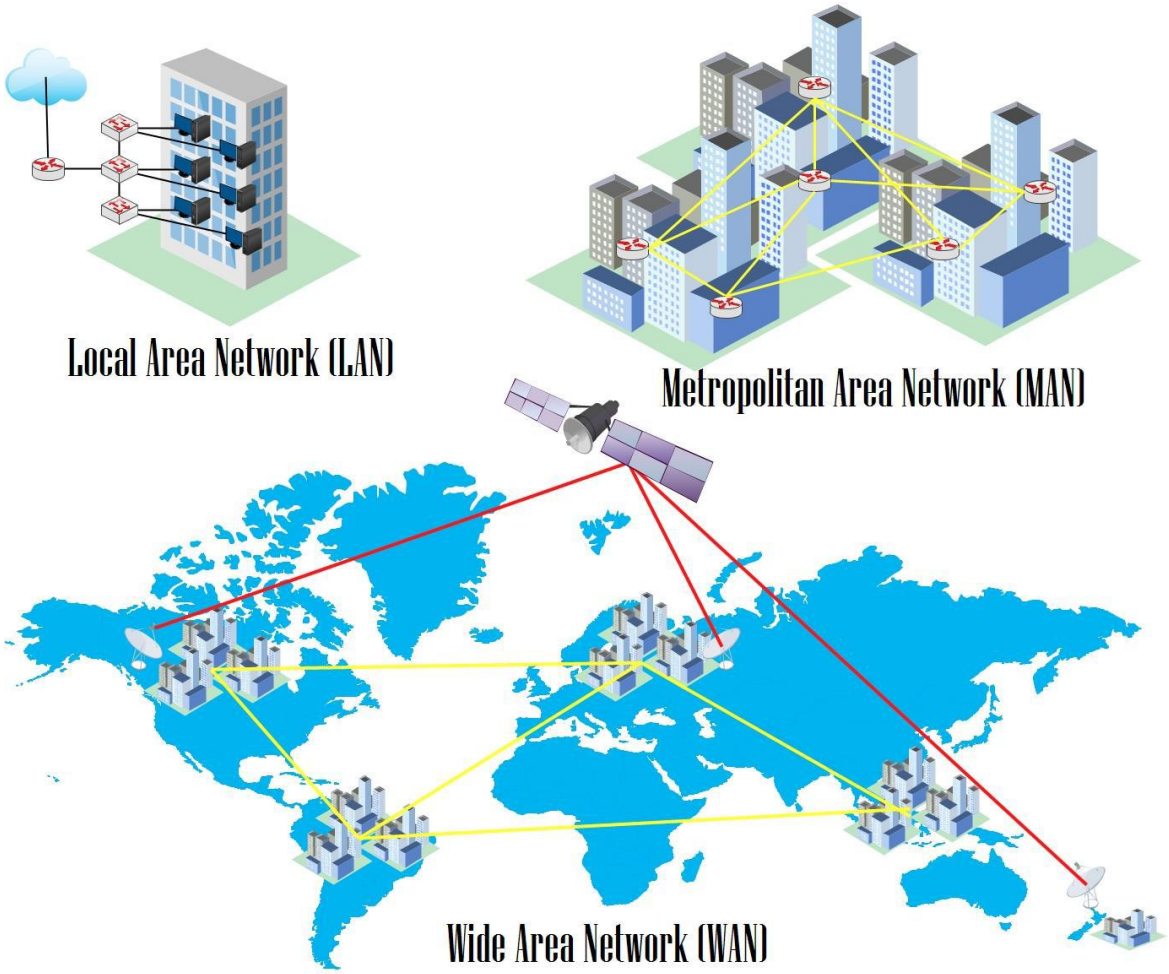
DASAR JARINGAN

Jaringan merupakan sekumpulan perangkat jaringan (Netwrok device) yang saling terhubung dengan perangkat endhost (end device) yang bertujuan agar seluruh perangkata dapat bertukar informasi satu sama lain nya.

Dan salah satu komponen dalam pembentukan jaringan adalah :

- Netwrok Device (perangkat jaringan) : Router, Switch, HUB dan lain sebagai nya
- End host (end device) : PC, laptop, mobile phone dan lain sebagai nya
- Interconection(penghubung) : NIC, konektor, media transisi (fiber optic, fast ethernet, wireless dll)

Macam – macam skala jaringan



- **LAN (Local Area Network) :** merupakan salah satu bentuk jaringan dengan skala yang sederhana yang biasa di gunakan di dalam suatu gedung, rumah, sekolah dan lain sebagainya yang untuk alat penghubung nya dengan kabel UTP
- **MAN** (**Metropolitan Area Network) :** merupakan skala jaringan dalam bentuk yang lebih besar dari LAN yang dapat di katakan sebagai kumpulan LAN dalam satu wilayah atau satu area yang dapat di katakan pada jaringan antar gedung dalam suatu area/wilayah yang sama, untuk media konetor biasa menggunakan kabel atau pun nirkabel
- **WAN** (**Wide Area Network) :** yaitu jaringan yang menghubung kan beberapa MAN yang dapat di gunakan pada jaringan untuk menghubungkan antara suatu negara dan benua, dan untuk media penghubung nya biasa menggunakan Fiber Optic

OSI layer

OSI (Open System Interconection) merupakan salah satu model umum suatu jaringan yang mana di gunakan untuk membangun suatu jaringan agar dapat saling terhubung walau berbeda vendor yang di keluarkan oleh OSI yang dengan *International Organization for Standardization* yang di bagi menjadi 7 layer , yang mana 7 layer OSI harus di gunakan oleh vendor – vendor dalam memproduksi perangkat jaringan atau yang lain nya, agar walau pun berbeda vendor mereka tetap dapat terhubung .

OSI (Open Source Interconnection) 7 Layer Model

Layer	Application/Example		Central Device/ Protocols			DOD4 Model
Application (7) Serves as the window for users and application processes to access the network services.	**End User layer** Program that opens what was sent or creates what is to be sent Resource sharing • Remote file access • Remote printer access • Directory services • Network management		**User Applications** SMTP		GATEWAY Can be used on all layers	Process
Presentation (6) Formats the data to be presented to the Application layer. It can be viewed as the "Translator" for the network.	**Syntax layer** encrypt & decrypt (if needed) Character code translation • Data conversion • Data compression • Data encryption • **Character Set Translation**		JPEG/ASCII EBDIC/TIFF/GIF PICT			
Session (5) Allows session establishment between processes running on different stations.	**Synch & send to ports** (logical ports) Session establishment, maintenance and termination • Session support - perform security, name recognition, logging, etc.		**Logical Ports** RPC/SQL/NFS NetBIOS names			
Transport (4) Ensures that messages are delivered error-free, in sequence, and with no losses or duplications.	**TCP** Host to Host, Flow Control Message segmentation • Message acknowledgement • Message traffic control • Session multiplexing	PACKET FILTERING	TCP/SPX/UDP			Host to Host
Network (3) Controls the operations of the subnet, deciding which physical path the data takes.	**Packets** ("letter", contains IP address) Routing • Subnet traffic control • Frame fragmentation • Logical-physical address mapping • Subnet usage accounting		**Routers** IP/IPX/ICMP			Internet
Data Link (2) Provides error-free transfer of data frames from one node to another over the Physical layer.	**Frames** ("envelopes", contains MAC address) [NIC card — Switch — NIC card] (end to end) Establishes & terminates the logical link between nodes • Frame traffic control • Frame sequencing • Frame acknowledgment • Frame delimiting • Frame error checking • Media access control		**Switch Bridge WAP** PPP/SLIP	Land Based Layers		Network
Physical (1) Concerned with the transmission and reception of the unstructured raw bit stream over the physical medium.	**Physical structure** Cables, hubs, etc. Data Encoding • Physical medium attachment • Transmission technique - Baseband or Broadband • Physical medium transmission Bits & Volts		**Hub**			

Sebagai Network Engginer kita perlu paham 3 layer lower layer yaitu layer 1 (physical), layer2 (data link), layer3(Network), dan berikut penjelasan dari 3 layer tersebut :

Layer 1 (Physical) : layer yang di fungsikan sebagi media transmisi untuk mendefinisikan alat trasnmisi jaringan, dengan satuan data unit bit, untuk pengalamatan dengan binnary (1 0), perangkat HUB yang mana perangkat tersebut akan menbroadcast data/informasi ke semua port yang terhubung dengan nya .

Layer2 (Data link) : salah satu layer yang berperan penting karna berfungsi untuk menentukan bit – bit data di kelompokan dalam menjadi satu format yang di sebut frame, dan pada layer ini data akan di kirim kan sesuai dengan MAC – address yang mana ia akan melihat tabel MAC – address, untuk perangkat yang bekerja pada layer2 seperti switch dan bridge yang mana ia hanya mengirim data pada network yang sama .

Layer3 (Network) : untuk fungsi utama pada layer ini ia akan menyidiakan fungsi routing yang mana dapat di kirim keluar dari segment network local menuju network yang berbeda, untuk pengalamatan pada layer ini ia menggunakan dengan ip address (192.168.X.X) yang mana ia akan menggirim data sesuai ip address tujuan dengan menggunkan perangakat yang berkerja pada layer ini yaitu router .

LAYER	NAME	DATA UNIT	PENGALAMATAN	PERANGKAT	MEMORY
LAYER 1	PHYSICAL	BIT	BINARY	HUB	-
LAYER 2	DATA LINK	FRAME	MAC – ADDRESS	SWITCH	MAC ADDRESS TABEL
LAYER 3	NETWORK	PACKET	IP – ADDRESS	ROUTER	ROUTING TABEL

LAYER	KONEKTIVITAS	PENGIRMAN DATA
LAYER 1	ANTAR NETWORK YANG SAMA	BROADCAST KE SEMUA PORT
LAYER 2	ANTAR NETWORK YANG SAMA	BERDASARKAN MAC –ADDRESS TUJUAN
LAYER 3	ANTAR NETWORK YANG BERBEDA	BERDASARKAN IP – ADDRESS

PHYSICAL TERMINATIONS

packetlife.net

Optical Terminations

ST (Straight Tip)

SC (Subscriber Connector)

LC (Local Connector)

MT-RJ

Wireless Antennas

RP-TNC

RP-SMA

Copper Terminations

RJ-45

RJ-11

RJ-21 (25-pair)

DE-9 (Female)

DB-25 (Male)

DB-60 (Male)

GBICs

1000Base-SX/LX

1000Base-T

Cisco GigaStack

1000Base-SX/LX SFP

1000Base-T SFP

X2 (10Gig)

by Jeremy Stretch

v1.1

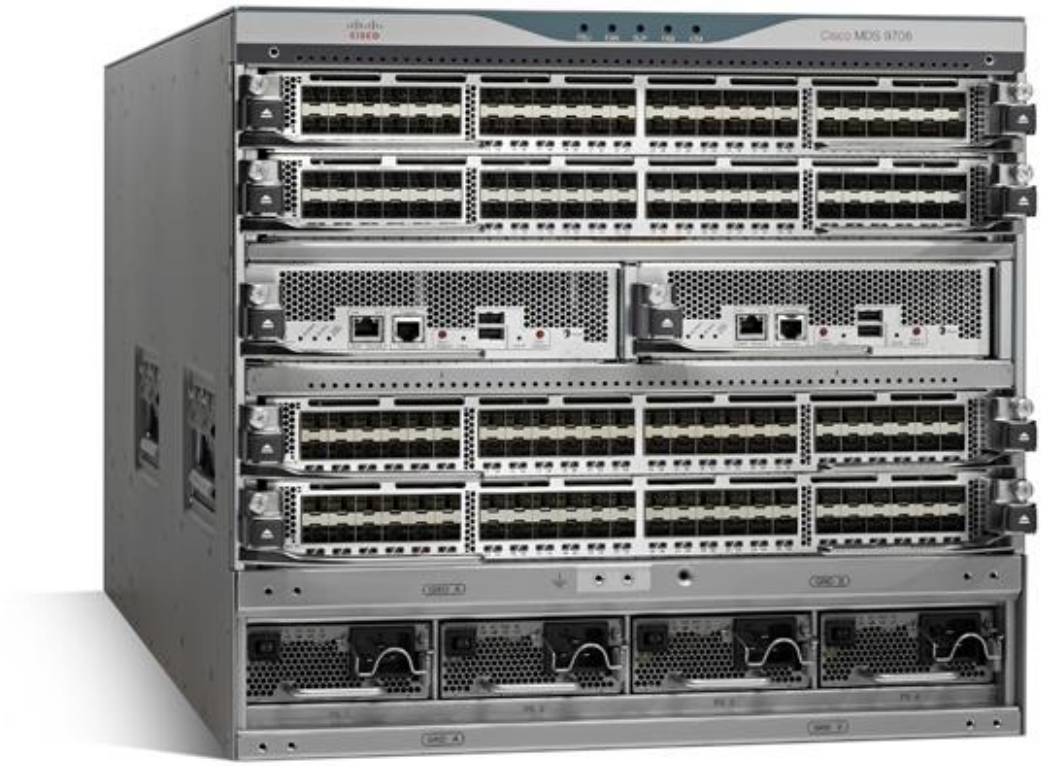
NETWORK DEVICE

Untuk jaringan sendiri terdapat beberapa device yang wajib kita tahu yang biasa kita akan gunakan untuk membuat suatu jaringan seperti Router, switch, hub, wireless dan lain sebagainya.

- ROUTER : fungsi router ini tidak lain yaitu ia di gunakan sebagai device yang menghubung kan suatu jaringan yang berbeda segment, dengan mensdistribusi kan ip address

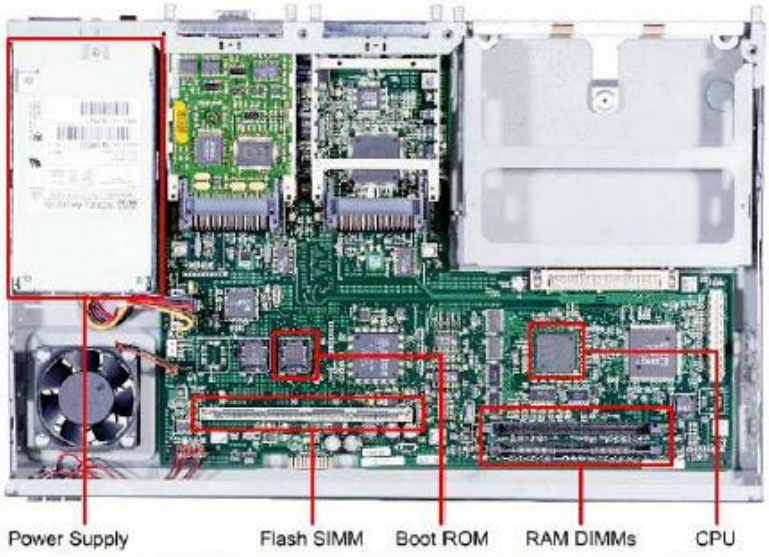
- SWITCH : fungsi perangkat ini merupakan salah satu perangkat yang di gunakan sebagai peghubung antar netwrok satu segment, yang mana ia akan mengirim kan packet/data melalui mac – address tabel

- MLS (Multi Layer Switch) : MLS juga merupakan suatu perangkat yang sering di gunakan pada suatu jaringan yang mana MLS ini masih termasuk perangkat yang tergolong switch tetapi perangkat ini dapat difungsi kan fungsi router juga, yang mana perangkat ini bergerak di 2 layer yaitu layer 2 dan 3

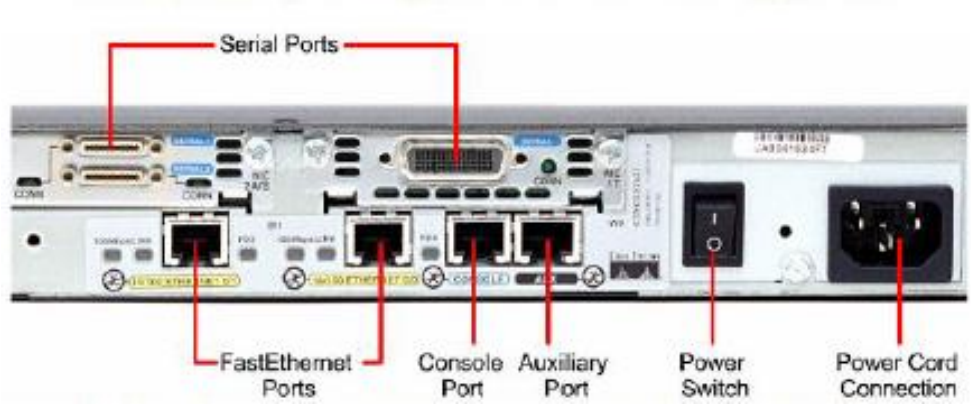


INTRODUCTION ROUTER CISCO

Sekilas untuk komponen internal pada perangkat router cisco, yang mana pada materi ini kita akan menggunakan router cisco 2699



Contoh bagian external router cisco 2u600



Dan untuk bagian utama pada suatu router di antara nya :

- **CPU** : CPU pada router dapat di gunakan untuk inisialisasi router , fungsi routing dan untuk mengontrol yang network interface, dan untuk router – router besar yang biasa di gunakan untuk core router memiliki beberapa CPU
- **RAM** : digunakan sebagai informasi routing table, fast switching cache, running configuration dan packet queque. RAM biasanya dibagi dua secara logik yaitu memori processor utama dan memory shared input/output (I/O). Memory shared I/O adalah berbagi antara berbagai interface I/O untuk menyimpan paket secara sementara. Isi RAM akan hilang begitu power dari Router dimatikan.
- **NVRAM** : di fungsi kan pada router sebagai penyedia storage untuk file starup configurastion, data masih ada walau pun router di matikan atau pun di restart
- **FLASH** : flash pada router di gunakan pada router untuk menagani IOS image, memberi akses software tanpa harus melepas chip pada prosesornya, data pun masih ada ketika router di matikan atau pun di restart, dapat menyimpan beberapa IOS router

Dan untuk proses booting pada router terdapat beberapa sesi

KONFIGURASI DASAR CISCO

Dalam CLI cisco terdapat beberapa user mode atau hak ases pada router atau pun switch, yang terbagi menjadi 3 :

- *User mode yang di tandai dengan* ">" : pada mode ini kita tidak dapat melakukan konfigurasi apapun
- *Privilange mode yang di tandai dengan "#"* : pada mode ini kita hanya dapat melihat konfigurasi dengan tidak dapat menambah konfigurasi
- *Global config yang di tandai dengan "(config)#"* : pada mode ini kita baru dapat melakukan konfigurasi entah itu menambah atau pun menghapus

Untuk masuk ke Privilange mode dari user mode ketikan "*enable*" dan setelah masuk ke privilange mode untuk masuk ke global config bisa ketikan "configure terminal"

```
Router>en
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
```

Dan berikut ini merupakan perintah paling dasar sebelum konfigurasi router atau pun switch di cisco.

Setelah masuk ke dalam mode Global config di situ lah kita dapat mulai meng konfigurasi-kan router atau pun switch

Dan untuk kembali ke user mode yang sebelum nya bisa dengan megetikan "exit" atau pun "exit"

```
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#
```

Cara mengganti hostname pada router atau pun switch

Mengganti hostname dalam router merupakan salah satu perintah dasar, yang biasa di gunakan agar kita tidak salah mengkonfigurasikan switch atau pun router, dengan cara masuk terlebih dahulu ke Global config

```
Router(config)#hostname bekasi
bekasi(config)#
```

Cara menyimpan konfigurasi pada router atau switch

Setelah kita telah banyak melakukan konfigurasi pada router atau pun switch maka jangan lupa untuk menyimpan konfigurasi tersebut di penyimpan di router atau yang biasa di sebut dengan *Nvram*

```
bekasi(config)#do write
Building configuration...
[OK]
```

Perlu di ketahui apabila kita ingin menyimpan atau pun melihat konfigurasi harus biasa menggunakan "do" seperti tadi do write, do show, dan lain sebgai nya,, tetapi apabila kalian masih berada di dalam privilange mode maka kita bisa langsung saja ketikan "write"

```
bekasi#write
Building configuration...
[OK]
```

Memasang password pada router atau switch

Pemberian password pada router atau pun switch sangat lah penting untuk yang bertujuan untuk keamanaan jaringan, agar para tangan – tangan jahil tidak bisa mengambil data – data kita

```
Router(config)#enable password 123
Router(config)#enable secret 456
```

Ada yang nama nya "enable password" dan juga "enable secret"

Enable password maka ia tidak terlalu aman karna pssword nya tidak

Terenkripsi

Dan "enable secret" terenskripsi pada saat di *do show run*

dan router atau switch hanya akan membaca password yang di enable secret

```
Router>enable :
Password:
Router#
```

Melihat infomasi interface router atau pun switch

Untuk melihat informasi interface kita seperti apa saja kah yang sudah ada di innterface kita dan juga salah satu cara dalam melakukan troubleshooting

```
bekasi(config)#do show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol

FastEthernet0/0        unassigned      YES unset  up                    up

FastEthernet0/0.30     30.30.30.1      YES manual up                    up

FastEthernet0/0.40     40.40.40.1      YES manual up                    up

FastEthernet0/1        unassigned      YES unset  up                    up

FastEthernet0/1.10     10.10.10.1      YES manual up                    up

FastEthernet0/1.20     20.20.20.1      YES manual up                    up

FastEthernet1/0        unassigned      YES unset  up                    up

FastEthernet1/0.50     50.50.50.1      YES manual up                    up

FastEthernet1/0.60     60.60.60.1      YES manual up                    up

FastEthernet1/1        unassigned      YES unset  administratively down down

Vlan1                  unassigned      YES unset  administratively down down
```

Melihat konfigurasi tang sedang berjalan

Fungsi dari melihat konfigurasi yang sedang berjalan di router atau pun di switch yang biasa di gunakan untuk troubleshooting

```
Router(config)#do sh run
Building configuration...

Current configuration : 620 bytes
!
version 12.4
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Router
!
!
!
enable secret 5 $1$mERr$DqFv/bNKU3CFm5jwSLasx/
enable password 123
!
!
!
!
!
!
ip cef
no ipv6 cef
!
!
!
!
!
!
!
```

```
!
!
!
!
!
spanning-tree mode pvst
!
!
!
!
!
!
interface FastEthernet0/0
no ip address
duplex auto
speed auto
shutdown
!
interface FastEthernet0/1
no ip address
duplex auto
speed auto
shutdown
!
interface Vlan1
no ip address
shutdown
!
ip classless
!
ip flow-export version 9
!
!
!
!
!
!
!
line con 0
!
line aux 0
!
line vty 0 4
login
!
!
!
end
```

Memasang MOTD (Messenger Of The Day)

```
Router(config)#banner motd z
Enter TEXT message. End with the character 'z'.
SELAMAT PAGI BOOS!! Z
```

Kemudian kita kembali ke user mode, maka akan keluar MOTD yang telah kita buat tadi

```
SELAMAT PAGI BOOS!!

Router>
```

Membersihkan konfigurasi

```
Router#write erase
Erasing the nvram filesystem will remove all configuration
files! Continue? [confirm]  (enter)
Router# delete flash:vlan.dat
Router#reload Proceed with reload? [confirm] (enter)
```

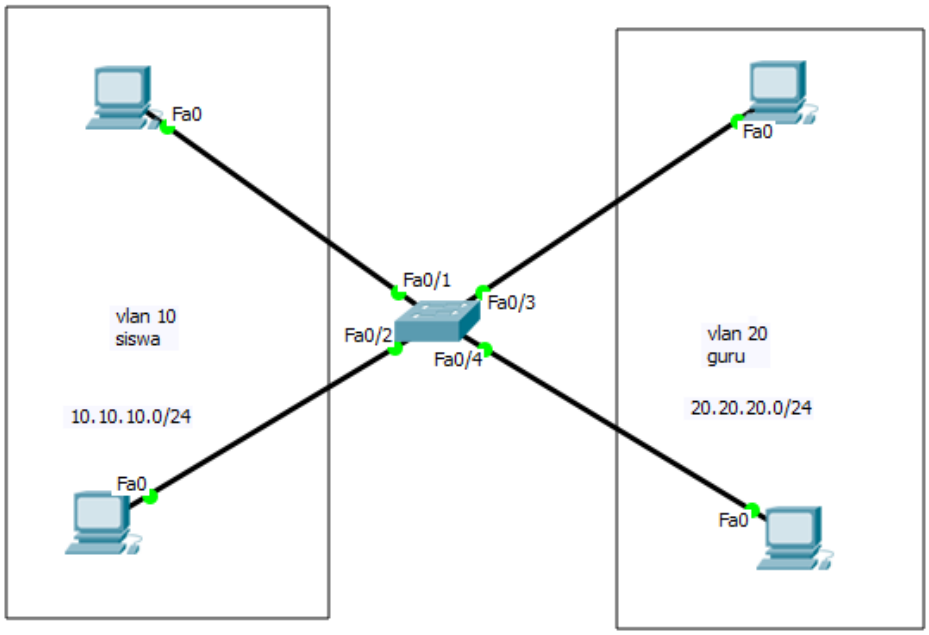
SWITCHING

Virtual LAN (VLAN)
TRUNKING
MLS
ETHERCHANNEL
Spanning Tree Protocol
Port Security
VTP
DHCP Vlan

Virtual LAN (VLAN)

Sesuai dengan nama VLAN yang mana ia akan di fungsi akan untuk membagi lingkup jaringan atau network yang mana ia akan membagi nya pada suatu switch beberapa network, yang akan di bagi sesuai dengan kebutuhan nya nanti, vlan juga biasa di gunakan di suatu jaringan untuk memudahkan dalam troubleshoting apabila terdapat suatu kesalahan jaringan karna network nya sudah kita bagi.

Untuk vlan sendiri di hanya di miliki oleh sebuah switch yang manageable yang mana switch tersebut dapat kita konfigurasi kan, beda dengan switch manageable yang mana port – port interface nya hanya dapat di gunakan pada network yang sama (satu network) yang mana ia tidak mendukung fitur VLAN yang ia akan membagi beberapa network .



Buatlah topologi seperti di atas, yang mana kita akan ,membuat dua vlan yaitu vlan10 dengan nama siswa dan vlan20 dengan nama guru, dan jangan lupa pasang IP di setiap PC nya sesuai dengan topologi

Pasang VLAN

```
Switch>enable
Switch#config terminal
Switch(config)#vlan 10 : buat terlebih dahulu vlan nya
Switch(config-vlan)#int fa0/1 : masuk ke interface untuk di vlan 10
Switch(config-if)#switchport mode access :masuk mode access switch
Switch(config-if)#switchport access vlan 10 :access vlan 10
Switch(config-if)#int fa0/2
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 10
```

```
Switch(config-if)#vlan 20
Switch(config-vlan)#int fa0/3
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 20
Switch(config-if)#int fa0/4
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 20
```

Untuk penegecekan coba lah kita ping antar PC pada satu vlan

```
Pinging 10.10.10.2 with 32 bytes of data:

Reply from 10.10.10.2: bytes=32 time=1ms TTL=128
Reply from 10.10.10.2: bytes=32 time=0ms TTL=128
Reply from 10.10.10.2: bytes=32 time=1ms TTL=128
Reply from 10.10.10.2: bytes=32 time=0ms TTL=128

Ping statistics for 10.10.10.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

dan PC di vlan10 tidak akan bisa ping untuk ke beda vlan atau vlan20

```
PC>ping 20.20.20.1

Pinging 20.20.20.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 20.20.20.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Setelah kita buat vlan nya kita dapat mengecek vlan dan interface dalam vlan tersebut dengan " `show vlan brief` "

```
Switch#show vlan brief

VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                 Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                 Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                 Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                 Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                                 Gig0/1, Gig0/2
10   siswa                             active    Fa0/1, Fa0/2
20   guru                              active    Fa0/3, Fa0/4
1002 fddi-default                      active
1003 token-ring-default                active
1004 fddinet-default                   active
1005 trnet-default                     active
```

Bisa di lihat terdapat vlan 1, 10, 20 dan perlu diketahui vlan1 merupakan vlan default untuk switch dan sisa interface – interface yang lain nya masih berada di vlan1 dan juga kita sudah telah selsai memasukan interface – interface seperti pada topologi di atas pada vlan 10siswa atau pun vlan20 guru.

SWITCH VLAN TRUNKING

Pada lab kali ini kita akan mengkonfigurasi kan trunk yang dapat di fungsi kan untuk melewati trafic dari suatu switch ke switch atau router lain, dan pada lab kali ini kita akan mengkonfigurasi kan agar vlan yang sama dapat saling terhubung walaupun berbeda switch.

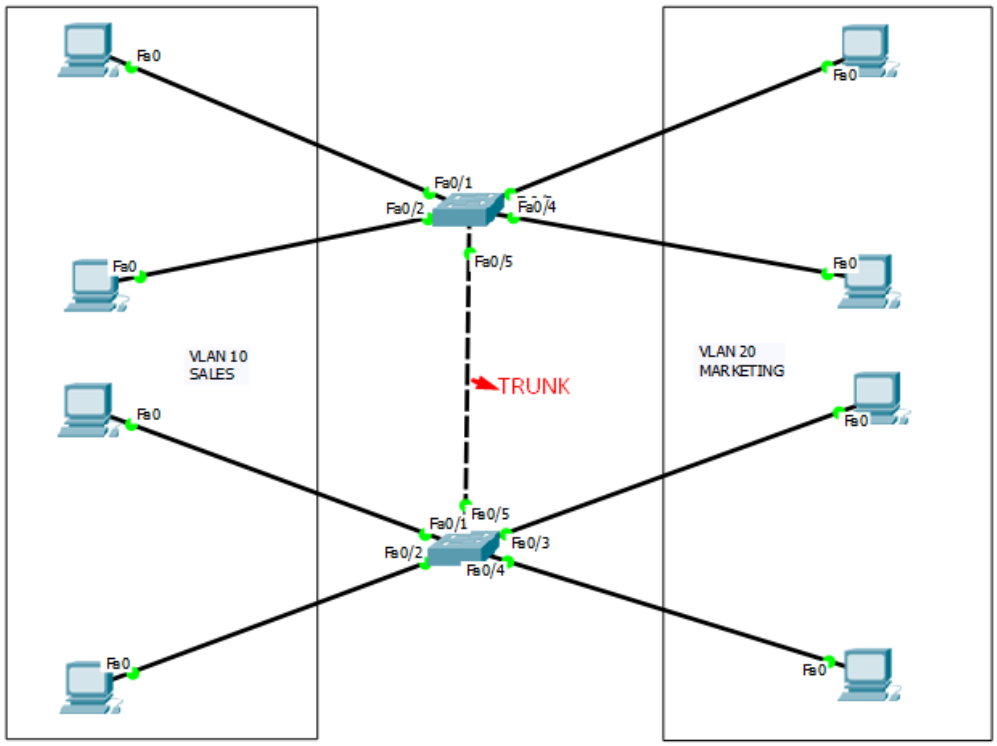
Ada 2 trunking protocol yang biasa digunakan:

- **ISL** = cisco proprietary, bekerja pada ethernet, token ring dan FDDI,

menambahi tag sebesar 30byte pada frame dan semua traffic VLAN ditag.

- **IEEE** 802.11Q (dot1q) = open standard, hanya bekerja pada ethernet,

menambahi tag sebesar 4byte pada frame.



Sesuai dengan topologi di atas kita akan membuat dua vlan (vlan 10 dan 20)di switch atas dan berikut juga di switch bawah, karna kedua vlan tersebut tidak terhubung dengan switch yang sama maka di butuhkan lah yang nama nya trunk.

Konfigurasikan VLAN di setiap switch sesuai dengan topologi.

```
Switch>enable
Switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#vlan 10
Switch(config-vlan)#name sales
Switch(config-vlan)#vlan 20
```

```
Switch(config-vlan)#name marketing
Switch(config-vlan)#int ra fa0/1-2 : bisa memakai range interface
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 10
Switch(config-if-range)#int ra fa0/3-4
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 20
```

Setelah di kedua switch di pasang vlan dan juga interface nya, maka setelah itu kita tinggal konfigurasikan trunk pada interface yang terhubung antar switch

Konfigurasi TRUNK

```
Switch(config)#int fa0/5
Switch(config-if)#switchport mode trunk : aktifkan mode trunk
Switch(config)#int fa0/5
Switch(config-if)#switchport mode trunk
```

Melihat interface trunk

```
Switch#show interfaces trunk
Port Mode Encapsulation Status Native vlan
Fa0/5 auto n-802.1q trunking 1

Port Vlans allowed on trunk
Fa0/5 1-1005

Port Vlans allowed and active in management domain
Fa0/5 1,10,20

Port Vlans in spanning tree forwarding state and not pruned
Fa0/5 1,10,20
```

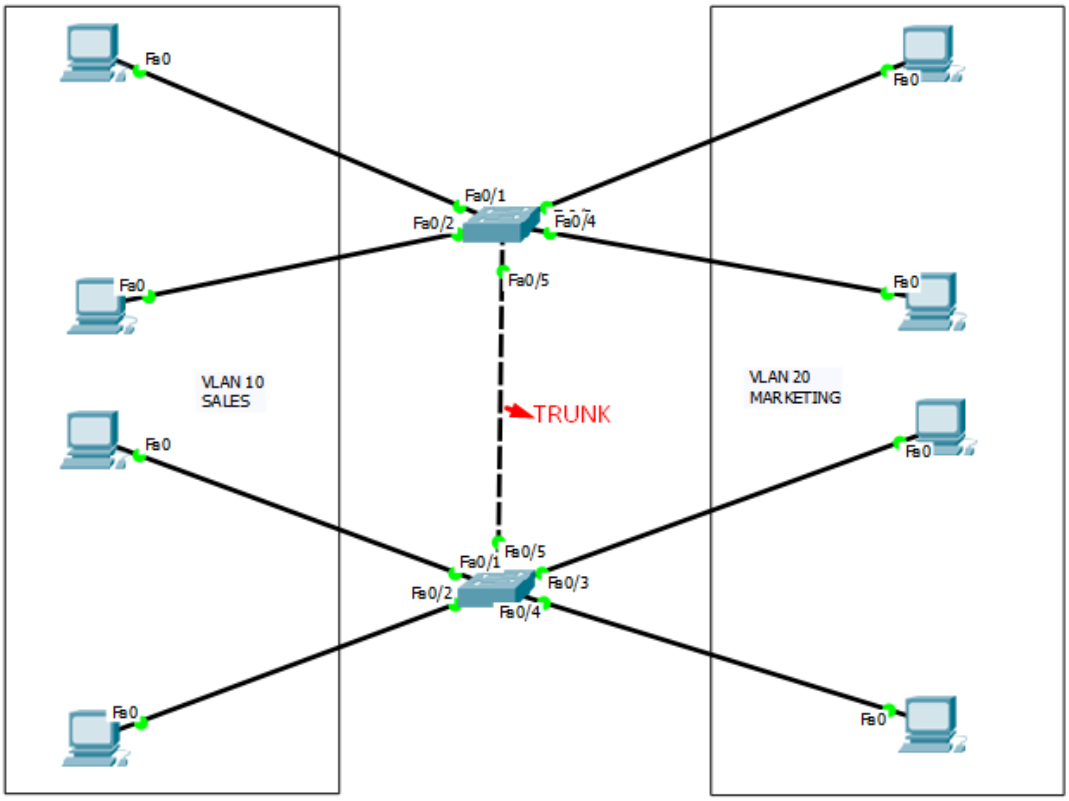
Test ping PC1 dengan PC yang sevlan tetapi berbeda switch

```
Packet Tracer PC Command Line 1.0
PC>ping 10.10.10.10

Pinging 10.10.10.10 with 32 bytes of data:

Reply from 10.10.10.10: bytes=32 time=0ms TTL=128
Reply from 10.10.10.10: bytes=32 time=1ms TTL=128
Reply from 10.10.10.10: bytes=32 time=0ms TTL=128
Reply from 10.10.10.10: bytes=32 time=0ms TTL=128

Ping statistics for 10.10.10.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

PC>
```

TRUNK ALLOWED

Pada default nya di saat kita mengkonfigurasi kan trunk pada interface antar switch maka trunk tersebut memboleh kan semua vlan (1 – 1005) untuk melewati trunk tersebut, dan agar pada trunk di interface kita hanya memboleh kan vlan tertentu saja maka kita dapat mengkonfigurasi kan *trunk allowed*

Sebelum kita memulai konfigurasi kita dapat melanjutkan pada topologi di lab sebelum nya,kemudian kita dapat melihat terlebih dahulu vlan yang di izinkan oleh trunk untuk melewati trunk tersebut



MELIHAT VLAN YANG DI IZINKAN TRUNK :

```
Switch#show int trunk
Port Mode Encapsulation Status Native vlan
Fa0/5 on 802.1q trunking 1

Port Vlans allowed on trunk
Fa0/5 1-1005

Port Vlans allowed and active in management domain
Fa0/5 1,10,20

Port Vlans in spanning tree forwarding state and not pruned
Fa0/5 1,10,20
```

Kemudian kita akan mengkonfigurasi kan *allowed trunk* dengan tujuan agar vlan yang di izin kan oleh trunk hanya vlan 10 dan 20 saja

KONFIGURASI ALLOWED TRUNK :

```
Switch(config)#int fa0/5
Switch(config-if)#switchport trunk allowed vlan 10,20
```

Untuk allowed ini kita konfigurasi kan di interface yang menjadi trunk di kedua switch

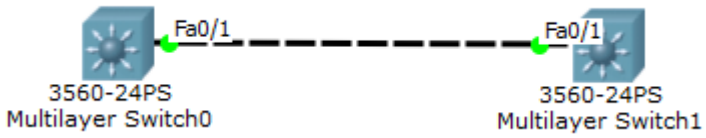
```
Switch(config)#int fa0/5
Switch(config-if)#switchport trunk allowed vlan 10,20
```

Kemudian setelah itu kita dapat melihat kembali di interface trunk apakah vlan yang di izinkan sudah berganti

```
Switch#sh int trunk
Port Mode Encapsulation Status Native vlan
Fa0/5 on 802.1q trunking 1

Port Vlans allowed on trunk
Fa0/5 10,20

Port Vlans allowed and active in management domain
Fa0/5 10,20

Port Vlans in spanning tree forwarding state and not pruned
Fa0/5 10,20
```

TRUNK MLS (Multi Layer Switch)

Trunk pada MLS (Multi Layer Switch) memiliki sedikit perbedaan di saat ingin mengkonfigurasi kan trunk yang mana ia tidak bisa langsung saja mengkonfigurasi kan trunk seperti pada switch – switch sebelum nya kita harus mengkonfigurasikan encapsulation pada trunk kemudian kita baru dapat mengkonfigurasi kan trunk



Untuk pengecekan kita dapat memulai dengan membuat trunk antar MLS dengan tidak membuat encapsulation terlebih dahulu

```
MLS1(config)#int fa0/1
MLS1(config-if)#switchport mode trunk
Command rejected: An interface whose trunk encapsulation is
"Auto" can not be configured to "trunk" mode.
```

Terlihat bahwa encapsulation mode auto tidak dapat menjadi trunk, oleh karna itu kita harus menkonfigurasikan encapsulation terlebih dahulu, setelah itu kita baru lah membuat interface trunk.

```
MLS1(config)#int fa0/1
MLS1(config-if)#switchport trunk encapsulation dot1q
MLS1(config-if)#switchport mode trunk
MLS2(config)#int fa0/1
MLS2(config-if)#switchport trunk encapsulation dot1q
MLS2(config-if)#switchport mode trunk
```

Setelah itu kita coba lihat apakah sudah terbuat interface trunk nya :

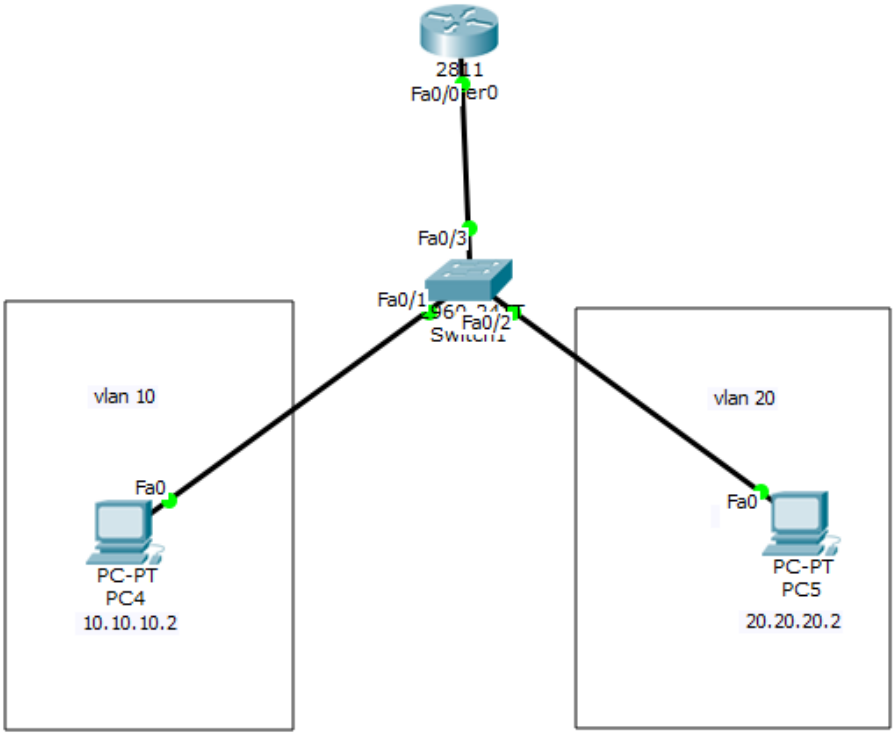
```
MLS1(config)#do sh int trunk
Port        Mode         Encapsulation  Status        Native vlan
Fa0/1       on           802.1q         trunking      1

Port        Vlans allowed on trunk
Fa0/1       1-1005

Port        Vlans allowed and active in management domain
Fa0/1       1

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/1       1
MLS1(config)#
```

Dapat di lihat bahwa interface fa0/1 sudah aktif menjadi trunk.

INTERVLAN ROUTING

Kalau pada lab sebelum nya suatu vlan tidak dapat saling terhubung dengan vlan yang berbeda di karakan network nya yang berbeda, dan pada lab kali ini kita akan mengkonfigurasi kan agar suatu vlan dapat saling terhubung dengan vlan yang berbeda (berbda network), yang mana kita akan mengkonfigurasi kan nya degan menggunakan router sebagai penghubung antar vlan yang berbeda.

Inter vlan di gunakan pada perangkat layer3 seperti router multi layer switch



Kita akan membuat dua vlan dalam sebuah switch yaitu vlan 10 dan 20, secara default nya PC di vlan 10 dan PC di vlan 20 tidak bisa saling terhubung atau dapat saling ping.

Buat vlan 10 dan 20 beserta interface nya :

```
Switch(config)#int fa0/1
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 10
Switch(config-if)#int fa0/2
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 20
```

Untuk membuat inter vlan dalam router di butuh kan juga trunk dari switch menuju ke router

Pasang interface trunk ke router :

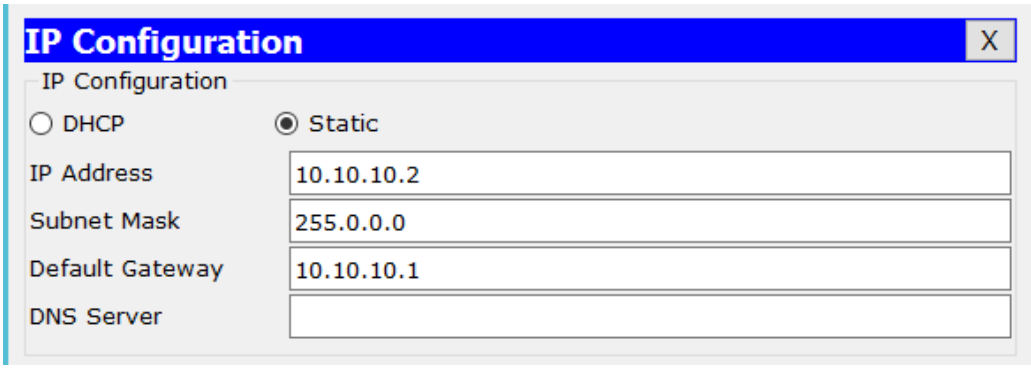
```
Switch(config)#int fa0/3
Switch(config-if)#switchport mode trunk
```

Pasang ip yang nanti nya untuk gateway antar vlan :

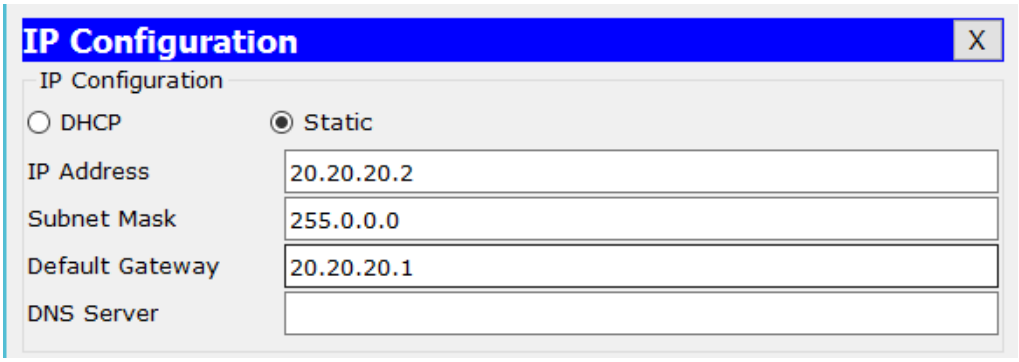
```
Router(config)#int fa0/0
Router(config-if)#no sh
Router(config)#int fa0/0.10  : masuk sub-interface
Router(config-subif)#encapsulation dot1Q 10
Router(config-subif)#ip ad 10.10.10.1 255.255.255.0  : gateway
Router(config-subif)#exit
Router(config)#
Router(config-if)#int fa0/0.20
Router(config-subif)#encapsulation dot1Q 20
Router(config-subif)#ip ad 20.20.20.1 255.255.255.0
Router(config-subif)#exit
```

Dan untuk pengetes an kita dapat mencoba dengan test ping antar pc yang berbeda vlan, dan jagan lupa pasang terlebih dahulu IP pada masing – masing PC, dan jagan lupa masuk an gateway nya dengan IP yang tadi di pasang di router di dalam sub.interface.

PC vlan 10 :



PC vlan 20 :



Test PING :

PC1 menuju PC2

```
PC>ping 20.20.20.2

Pinging 20.20.20.2 with 32 bytes of data:

Reply from 20.20.20.2: bytes=32 time=1ms TTL=127
Reply from 20.20.20.2: bytes=32 time=0ms TTL=127
Reply from 20.20.20.2: bytes=32 time=1ms TTL=127
Reply from 20.20.20.2: bytes=32 time=0ms TTL=127

Ping statistics for 20.20.20.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

PC2 menuju PC1

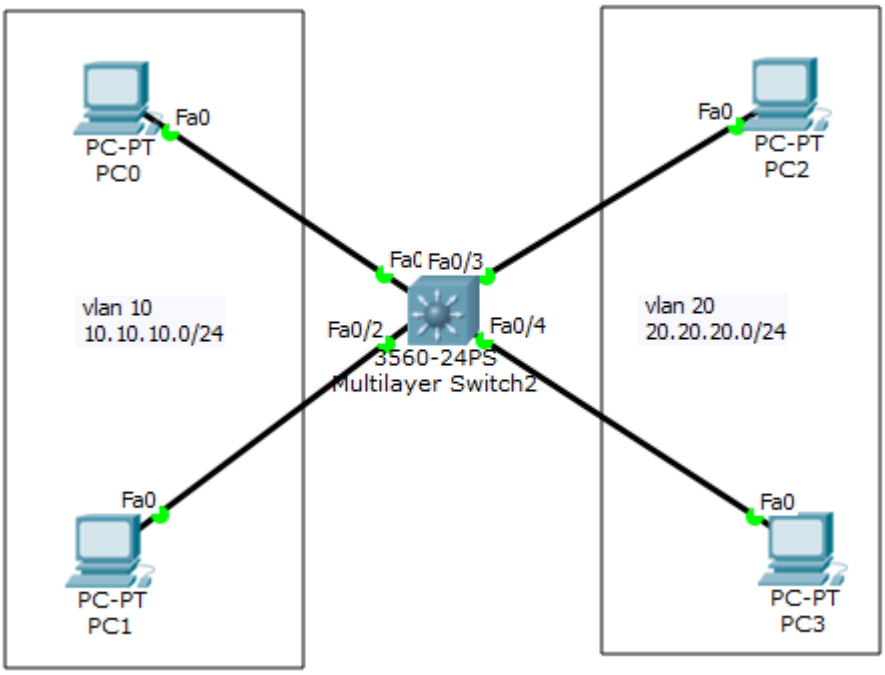
```
PC>ping 10.10.10.2

Pinging 10.10.10.2 with 32 bytes of data:

Reply from 10.10.10.2: bytes=32 time=0ms TTL=127
Reply from 10.10.10.2: bytes=32 time=0ms TTL=127
Reply from 10.10.10.2: bytes=32 time=1ms TTL=127
Reply from 10.10.10.2: bytes=32 time=0ms TTL=127

Ping statistics for 10.10.10.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Maka kedua vlan pun sudah dapat saling terhubung dengan mengkonfigurasi kan intervlan routing, yang mana kedua PC sudah dapat saling ping walau berbeda vlan / berbeda network .

SVI (Switch Virtual interface)

SVI (Switch Virtual Interface) merupakan sebuah mekanisme untuk melakukan sejenis seperti inter vlan routing yang mana pada SVI kita dapat mengkonfigurasi kan ip address pada vlan untuk menjadi gateway pada client yang berada pada vlan tersebut agar dapat saling terhubung dengan beda vlan

Untuk perangkat switch kita harus menggunakan switch yang mendukung fungsi router atau yang dapat bergerak di doubel layer, dan switch yang dapat mengkonfigurasi kan SVI tersebut hanyalah switch MLS (Multi Layer Switch)



Sebelum kita mengkonfigurasi kan SVI pada MLS kita dapat memulai dengan mengkonfigurasi kan vlan terlebih dahulu sesuai dengan topologi

KONFIGURASI VLAN :

```
Switch>en
Switch#conf t
Enter configuration commands, one per line. End with
CNTL/Z.
Switch(config)#vlan 10
Switch(config-vlan)#vlan 20
Switch(config-vlan)#int ra fa0/1-2
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 10
Switch(config-if-range)#int ra fa0/3-4
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 20
```

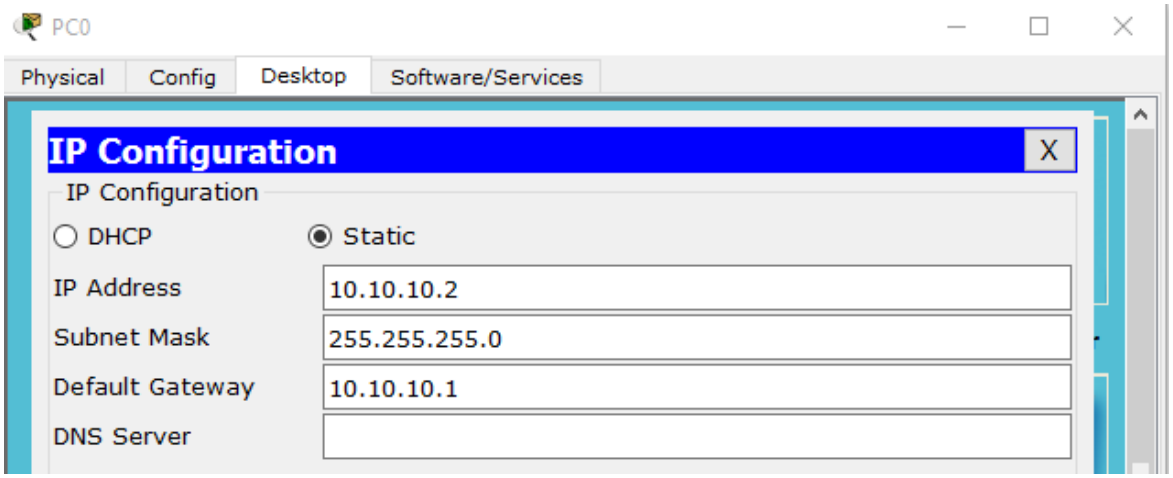
Pada tahapan ini maka client kita baru hanya dapat ping pada client yang satu vlan atau masih satu network, agar kita client dapat saling terhubung dengan vlan yang berbeda kita dapat mengkonfigurasi kan SVI dengan menambah kan ip address pada vlan yang nanti nya akan di gunakan client untuk menuju vlan lain

KONFIGURASI IP ADDRESS PADA VLAN :

```
Switch(config)#int vlan 10
Switch(config-if)#ip add 10.10.10.1 255.255.255.0
Switch(config-if)#int vlan 20
Switch(config-if)#ip add 20.20.20.1 255.255.255.0
```

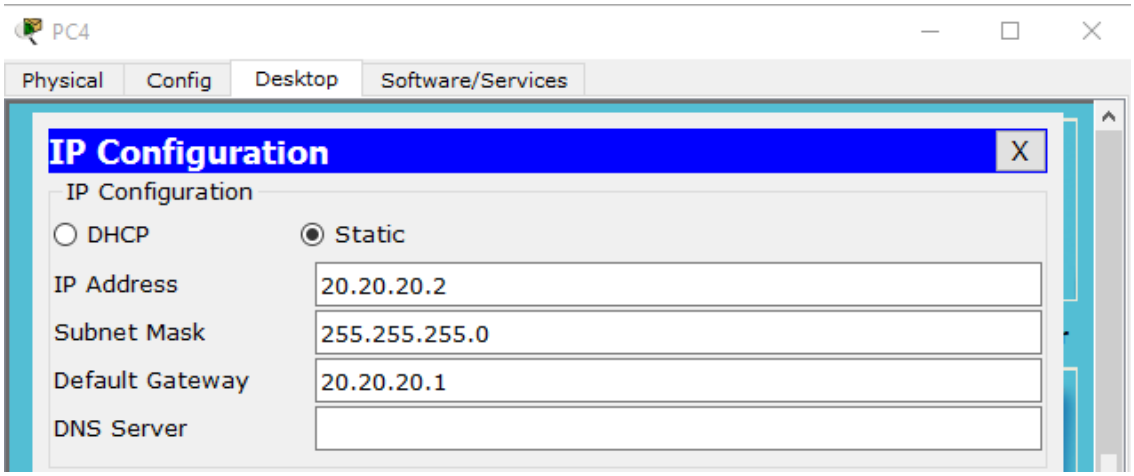
Setelah itu kita coba memberikan ip address pada masing – masing client dengan gateway ip vlan yang tadi kita konfiguasi kan sesuai dengan vlan

Client Vlan 10



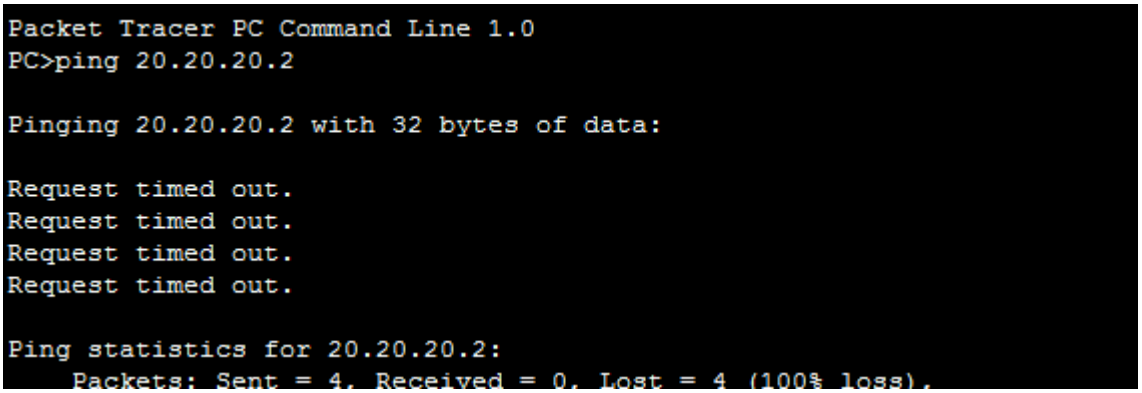
Konfigurasi kan ip pada client yang di vlan 20 dengan gateway yang tadi kita konfigurasi kan di ip pada vlan 20

Client vlan 20



Setelah kita konfigurasi kan ip pada setiap client beserta dengan gateway nya dengan ip pada vlan, kita dapat mencoba dengan melakukan tes ping dengan ping antar client yang berbeda vlan, apakah bisa ..??

Test ping client vlan 10 menuju vlan 20



Maka hasil nya pun akan Time out di karna kan kita belum mengaktifkan fungsi routing pada MLS, maka dari itu kita dapat mengkonfigurasi kan *//ip routing*

```
Switch(config)#ip routing
```

Maksud dari konfigurasi *iprouting* yaitu untuk mengaktifkan fungsi routing agar client yang berbeda network dapat saling terhubung dengan ip routing

VTP (Vlan Trunking protocol)

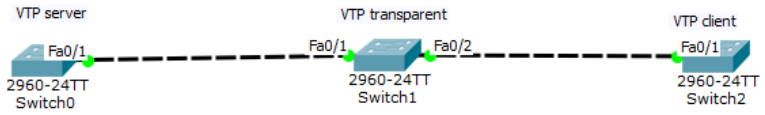
Vlan trunking protocol atau yang akrab di sebut dengan VTP ini adalah sebuah suatu cara agar kita dapat memanagement VLAN secara terpusat, yang di mana kita dapat lebih mudah menambakan atau pun mengurangi vlan dalam satu switch saja dan switch – switch yang lain nya akan mengikuti nya perubahan yang baru saja kita buat tadi.

Dalam VTP terbagi mejadi 3 mode untuk kita konfigurasikan :

1. **VTP mode server** : yaitu di mode ini switch yang akan kita konfigurasikan akan menjadi induk bagi para switch yang lain nya, yang mana apabila switch yang yang menjadi VTP mode server menambahkan vlan atau pun menghapus nya maka switch yang lain nya akan ikut mengupdate apa yang telah kita edit di switch yang di pasang mode server
2. **VTP mode client** : vtp mode yang akan menginduk kepada vtp server yang apa bila kita sudah satu domain dengan vtp server maka secara otomatis di switch kita yang sudah di pasang vtp mode client akan menambah kan sendiri, dan apa bila VTP server menghapus vlan maka client pun akan ikut terhapus juga
3. **VTP mode transparent** : vtp mode transparent ini ia dapat membuat vlan tetapi vlan yang di buat nya hanya lah bersifat local, yang mana iya hanya meneruskan saja, tetapi ia tidak mendapatkan update dari vtp server

	VTP Server	VTP Client	VTP Transparent
Create/Modify/Delete VLAN	Yes	No	Only local
Syncronizes itself	Yes	Yes	No
Forwards advertisements	Yes	Yes	Yes

Dan berikut topologi nya :



Kita akan mencoba dengan 3 switch yang memiliki mode server, transparent, client. Hal yang harus kalian seting adalah membuat trunk terlebih dahulu antar switch agar vtp client mendapat kan update dari server, setelah itu kita akan membuat domain untuk server dan password nya, yang kemudian di sesuaikan dengan vtp transparent dan client nya.

Switch VTP server :

```
VTPserver(config)#int fa0/1
VTPserver(config-if)#switchport mode trunk: pasang trunk antar switch
VTPserver(config)#vtp mode server: mode nya vtp server
Device mode already VTP SERVER
VTPserver(config)#vtp domain IDN: pasang domain untuk VTP
Domain name already set to IDN.
VTPserver(config)#vtp password 123 : set password untuk VTP
Password already set to 123
Buat vlan :
VTPserver(config)#vlan 10
VTPserver(config-vlan)#vlan 20
VTPserver(config-vlan)#vlan 30
VTPserver(config-vlan)#vlan 40
VTPserver(config-vlan)#vlan 50
```

Mengecek vlan :

```
VTPserver(config)#do sh vlan br

VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Fa0/2, Fa0/3, Fa0/4, Fa0/5
                                                 Fa0/6, Fa0/7, Fa0/8, Fa0/9
                                                 Fa0/10, Fa0/11, Fa0/12, Fa0/13
                                                 Fa0/14, Fa0/15, Fa0/16, Fa0/17
                                                 Fa0/18, Fa0/19, Fa0/20, Fa0/21
                                                 Fa0/22, Fa0/23, Fa0/24, Gig0/1
                                                 Gig0/2
10   VLAN0010                          active
20   VLAN0020                          active
30   VLAN0030                          active
40   VLAN0040                          active
50   VLAN0050                          active
1002 fddi-default                      active
1003 token-ring-default                active
1004 fddinet-default                   active
1005 trnet-default                     active
```

Switch VTP transparent :

```
VTPtransparent(config)#int ra fa0/1-2
VTPtransparent(config-if-range)#switchport mode trunk
VTPtransparent(config)#vtp mode transparent
Device mode already VTP TRANSPARENT.
VTPtransparent(config)#vtp domain IDN
Domain name already set to IDN.
VTPtransparent(config)#vtp password 123
Password already set to 123
```

Cek apakah vlan nya bertambah ? :

```
VTPtransparent(config)#do sh vlan br

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/3, Fa0/4, Fa0/5, Fa0/6
                                                Fa0/7, Fa0/8, Fa0/9, Fa0/10
                                                Fa0/11, Fa0/12, Fa0/13, Fa0/14
                                                Fa0/15, Fa0/16, Fa0/17, Fa0/18
                                                Fa0/19, Fa0/20, Fa0/21, Fa0/22
                                                Fa0/23, Fa0/24, Gig0/1, Gig0/2
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
```

Jawaban nya tentu tidak karna memang vtp mode transparent hanya bersifat local dan ia hanya meneruskan saja dari vtp server dan tidak mengupdate konfigurasi dari vtp server

Switch VTP client :

Pasang trunk untuk interface antar switch

```
VTPclient(config)#int fa0/1
VTPclient(config-if)#switchport mode trunk
```

Pasang mode untuk VTP client, lalu masuk an domain dan password :

```
VTPclient(config)#vtp mode client
Setting device to VTP CLIENT mode.
VTPclient(config)#vtp domain IDN
Domain name already set to IDN.
VTPclient(config)#vtp password 123
Setting device VLAN database password to 123
```

Coba kita lihat vlan nya apakah sudah ada :

```
VTPclient(config)#do sh vlan br

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/2, Fa0/3, Fa0/4, Fa0/5
                                                Fa0/6, Fa0/7, Fa0/8, Fa0/9
                                                Fa0/10, Fa0/11, Fa0/12, Fa0/13
                                                Fa0/14, Fa0/15, Fa0/16, Fa0/17
                                                Fa0/18, Fa0/19, Fa0/20, Fa0/21
                                                Fa0/22, Fa0/23, Fa0/24, Gig0/1
                                                Gig0/2
10   VLAN0010                         active
20   VLAN0020                         active
30   VLAN0030                         active
40   VLAN0040                         active
50   VLAN0050                         active
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
```

Maka vlan sudah ada di switch yang ber mode VTP client maka apbila di switch VTP server menghapus,menambah atau pun mengedit vlan – vlan tersebut maka VTP client akan mengupdate konfigurasi vlan pada VTP server

Dan pada switch VTP transparent ia tidak akan mengupdate tetapi di hanya memiliki jaringan lokal nya saja, yang mana ia dapat menambah vlan di switch nya ia saja

Test untuk vlan di switch VTP transaprent :

```
VTPtransparent(config)#vlan 60
VTPtransparent(config-vlan)#name test
```

Coba kita lihat apakah di switch yang VTP client atau VTP server bertambah vlan nya :

```
VTPserver(config)#do sh vlan br

VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Fa0/2, Fa0/3, Fa0/4, Fa0/5
                                                 Fa0/6, Fa0/7, Fa0/8, Fa0/9
                                                 Fa0/10, Fa0/11, Fa0/12, Fa0/13
                                                 Fa0/14, Fa0/15, Fa0/16, Fa0/17
                                                 Fa0/18, Fa0/19, Fa0/20, Fa0/21
                                                 Fa0/22, Fa0/23, Fa0/24, Gig0/1
                                                 Gig0/2
10   VLAN0010                          active
20   VLAN0020                          active
30   VLAN0030                          active
40   VLAN0040                          active
50   VLAN0050                          active
1002 fddi-default                      active
1003 token-ring-default                active
1004 fddinet-default                   active
1005 trnet-default                     active
```

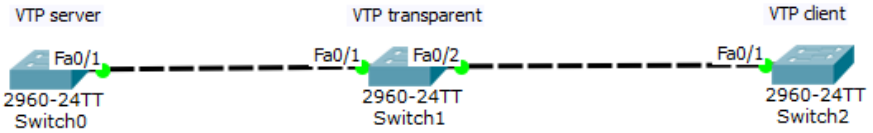
Dan di VTP client :

```
VTPclient(config)#do sh vlan br

VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Fa0/2, Fa0/3, Fa0/4, Fa0/5
                                                 Fa0/6, Fa0/7, Fa0/8, Fa0/9
                                                 Fa0/10, Fa0/11, Fa0/12, Fa0/13
                                                 Fa0/14, Fa0/15, Fa0/16, Fa0/17
                                                 Fa0/18, Fa0/19, Fa0/20, Fa0/21
                                                 Fa0/22, Fa0/23, Fa0/24, Gig0/1
                                                 Gig0/2
10   VLAN0010                          active
20   VLAN0020                          active
30   VLAN0030                          active
40   VLAN0040                          active
50   VLAN0050                          active
1002 fddi-default                      active
1003 token-ring-default                active
1004 fddinet-default                   active
1005 trnet-default                     active
```

Jadi kesimpulan nya guna dari VTP transparent di gunakan untuk switch yang untuk keamanan apabila terdapat seseorang yang masuk dalam sebuah switch dan mereka tidak langsung mengetahui seluruh jaringan di dalam nya.

VTP REVESION NUMBER

Vtp revision number adalah salah satu cara dalam pengelola sebuah vlan, dalam sebuah vtp domain tersebut memiliki masing – masing revison nya, yang mana revision itu akan otomatis bertambah apabila terdapat perubahan – perubahan pada vlan seperti menghapus, atau pun menambah vlan.



Pada lab kali ini akan ada yang di mana sebuah vtp client akan seolah – olah menjadi server.

Cek revision number nya :

```
VTPserver#sh vtp status
VTP Version : 2
Configuration Revision : 0 : masih 0 karna belum ada nya konfigurasi
Maximum VLANs supported locally : 255
Number of existing VLANs : 5
VTP Operating Mode : Server
VTP Domain Name :
VTP Pruning Mode : Disabled
VTP V2 Mode : Disabled
VTP Traps Generation : Disabled
MD5 digest : 0x7D 0x5A 0xA6 0x0E 0x9A 0x72 0xA0 0x3A
Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00
Local updater ID is 0.0.0.0 (no valid interface found)
```

Switch VTPserver :

```
VTPserver(config)#vtp mode server
Device mode already VTP SERVER.
VTPserver(config)#vtp domain IDN
Changing VTP domain name from NULL to IDN
VTPserver(config)#vtp password 123
Setting device VLAN database password to 123
```

Pasang trunk antar switch :

```
VTPserver(config)#int fa0/1
VTPserver(config-if)#switchport mode trunk
```

Buat VLAN :

```
VTPserver(config)#vlan 10
VTPserver(config-vlan)#vlan 20
VTPserver(config-vlan)#vlan 30
VTPserver(config-vlan)#vlan 40
VTPserver(config-vlan)#vlan 50
```

Cek kembali revision number nya :

```
VTPserver#sh vtp status
VTP Version : 2
Configuration Revision : 5: karna tadi kita menambahkan 5 vlan
Maximum VLANs supported locally : 255
Number of existing VLANs : 10
VTP Operating Mode : Server
VTP Domain Name : IDN
VTP Pruning Mode : Disabled
VTP V2 Mode : Disabled
VTP Traps Generation : Disabled
MD5 digest : 0x4A 0x11 0x28 0x8F 0xA2 0xCD 0x18 0xD1
Configuration last modified by 0.0.0.0 at 3-1-93 00:20:14
Local updater ID is 0.0.0.0 (no valid interface found)
```

switch VTP transparent

pasang trunk :

```
Switch(config)#int ra fa0/1-2
Switch(config-if-range)#switchport mode trunk
```

Konfigurasi VTP transparent :

```
Switch(config)#vtp mode transparent
Setting device to VTP TRANSPARENT mode.
Switch(config)#vtp domain IDN
Domain name already set to IDN.
Switch(config)#vtp password 123
Setting device VLAN database password to 123
```

Cek revision number :

```
VTPtransparent#sh vtp status
VTP Version : 2
Configuration Revision : 0
Maximum VLANs supported locally : 255
Number of existing VLANs : 5
VTP Operating Mode : Transparent
VTP Domain Name : IDN
VTP Pruning Mode : Disabled
VTP V2 Mode : Disabled
```

```
VTP Traps Generation : Disabled
MD5 digest : 0x74 0x00 0xC6 0xAF 0x89 0x05 0xE0 0xC2
Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00
```

Untuk VTP transparent ia tidak mengikuti dari VTP server karna sifat nya independent atau ia hanya untuk jaringan local nya, yang hanya meneruskan dari vtp server

Switch VTPclient :

Pasang trunk :

```
VTPclient(config)#int fa0/1
VTPclient(config-if)#switchport mode trunk
```

Konfigurasikan VTP client :

```
VTPclient(config)#vtp mode client
Setting device to VTP CLIENT mode.
VTVTPclient(config)#vtp domain IDN
Domain name already set to IDN.
VTPclient(config)#vtp password 123
Setting device VLAN database password to 123
```

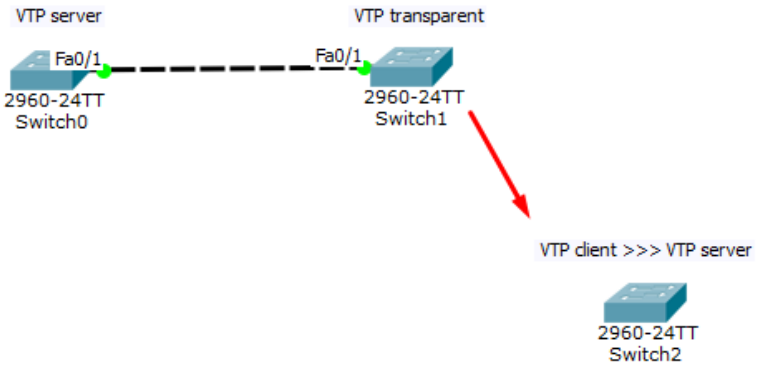
Cek apakah sudah ada vlan nya :

```
VTPclient(config)#do sh vlan br

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/2, Fa0/3, Fa0/4, Fa0/5
                                                Fa0/6, Fa0/7, Fa0/8, Fa0/9
                                                Fa0/10, Fa0/11, Fa0/12, Fa0/13
                                                Fa0/14, Fa0/15, Fa0/16, Fa0/17
                                                Fa0/18, Fa0/19, Fa0/20, Fa0/21
                                                Fa0/22, Fa0/23, Fa0/24, Gig0/1
                                                Gig0/2
10   VLAN0010                         active
20   VLAN0020                         active
30   VLAN0030                         active
40   VLAN0040                         active
50   VLAN0050                         active
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
VTPclient(config)#^Z
VTPclient#
%SYS-5-CONFIG_I: Configured from console by console
```

Setelah itu cek Revision number apakah sudah ada mengikuti sever :

```
VTPclient#sh vtp status
VTP Version : 2
Configuration Revision : 5 : ia akan mengikuti apabila client
Maximum VLANs supported locally : 255
Number of existing VLANs : 10
VTP Operating Mode : Client
VTP Domain Name : IDN
VTP Pruning Mode : Disabled
VTP V2 Mode : Disabled
VTP Traps Generation : Disabled
MD5 digest : 0x4A 0x11 0x28 0x8F 0xA2 0xCD 0x18 0xD1
Configuration last modified by 0.0.0.0 at 3-1-93 00:20:14
```

coba kita putus kan sambungan switch vtp client setelah itu ganti mode nya menjadi vtp server dengan domain dan password yang sama, dan buat vlan sampai 4, sampai revision number dari pada server pada topologi sebelum nya.



Ubah VTP client menjadi VTP server :

```
VTPclient(config)#vtp mode server
Setting device to VTP SERVER mode.
VTPclient(config)#vtp domain IDN
Domain name already set to IDN.
VTPclient(config)#vtp password 123
Password already set to 123
```

Tambah kan vlan agar revision number nya lebih banyak dari server sebelum nya :

```
VTPclient(config)#vlan 60
VTPclient(config-vlan)#vlan 70
VTPclient(config-vlan)#vlan 80
VTPclient(config-vlan)#vlan 90
```

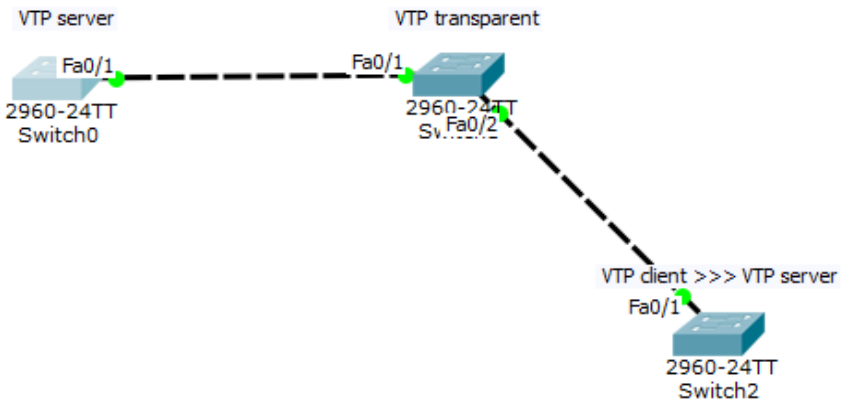
Setelah itu kita cek revision number nya apakah bertambah :

```
VTPclient#show vtp status
VTP Version : 2
Configuration Revision : 11
```

```
Maximum VLANs supported locally : 255
Number of existing VLANs : 14
VTP Operating Mode : Server
VTP Domain Name : IDN
VTP Pruning Mode : Disabled
VTP V2 Mode : Disabled
VTP Traps Generation : Disabled
MD5 digest : 0xD0 0xC4 0x65 0xE3 0xFF 0x78 0x27 0x01
Configuration last modified by 0.0.0.0 at 3-1-93 01:38:37
Local updater ID is 0.0.0.0 (no valid interface found)
```

Dan revision number nya telah bertambah dan lebih besar dari server sebelum nya.

Setelah itu coba kita ubah kembali mode nya dari server menjadi client, dan setelah itu sambungkan kembali ke topologi sebelum nya, dan lihat lah apa yang terjadi dengan vlan di server



cek vlan di vtp server :

```
VTPserver(config)#do sh vlan br

VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Fa0/2, Fa0/3, Fa0/4, Fa0/5
                                                 Fa0/6, Fa0/7, Fa0/8, Fa0/9
                                                 Fa0/10, Fa0/11, Fa0/12, Fa0/13
                                                 Fa0/14, Fa0/15, Fa0/16, Fa0/17
                                                 Fa0/18, Fa0/19, Fa0/20, Fa0/21
                                                 Fa0/22, Fa0/23, Fa0/24, Gig0/1
                                                 Gig0/2
10   VLAN0010                          active
20   VLAN0020                          active
30   VLAN0030                          active
40   VLAN0040                          active
50   VLAN0050                          active
60   VLAN0060                          active
70   VLAN0070                          active
80   VLAN0080                          active
90   VLAN0090                          active
1002 fddi-default                      active
1003 token-ring-default                active
1004 fddinet-default                   active
1005 trnet-default                     active
VTPserver(config)#
```

Maka vlan yang di server mengikuti switch yang bermode kan vtp client, di karna kan vtp akan mengikuti vtp yang revision number nya lebih besar, dan perlu di perhatikan pada saat memnyambung kan sebuah switch ke switch yang lain kita harus melihat terlebih dahulu revision number nya, agar tidak merusak jaringan vlan yang lain nya,

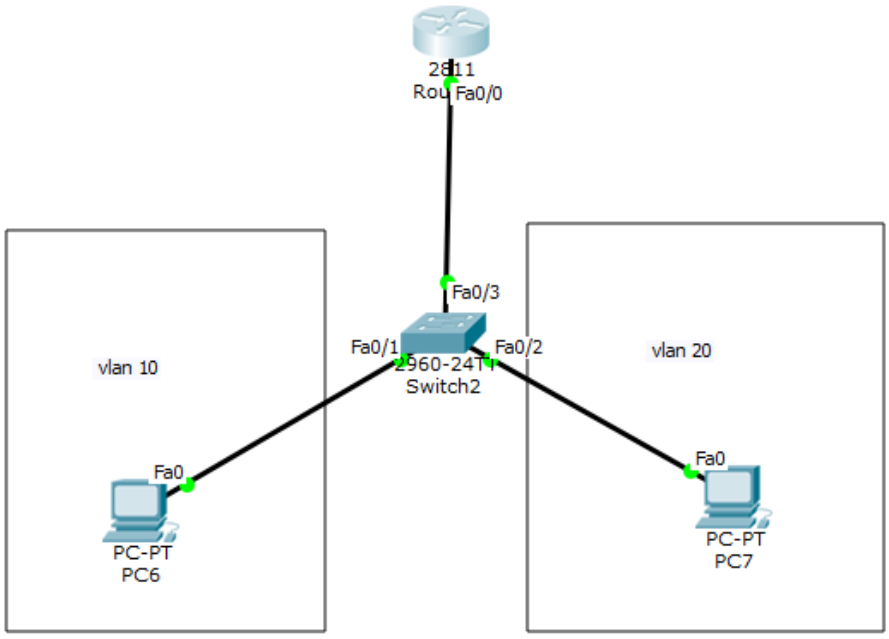
Dan salah satu cara untuk menghilangkan revision number tersebut dengan cara menganti mode nya menjadi transparent atau mengganti domain nya.

Maka revision number nya akan kembali menjadi 0.

```
VTPclient(config)#vtp mode transparent
Setting device to VTP TRANSPARENT mode.
VTPclient(config)#do sh vtp status
VTP Version : 2
Configuration Revision : 0
Maximum VLANs supported locally : 255
Number of existing VLANs : 14
VTP Operating Mode : Transparent
VTP Domain Name : IDN
VTP Pruning Mode : Disabled
VTP V2 Mode : Disabled
VTP Traps Generation : Disabled
MD5 digest : 0x26 0x1A 0xA5 0xCB 0xB4 0xE7 0x93 0xC8
Configuration last modified by 0.0.0.0 at 3-1-93 01:38:37
```

DHCP VLAN

Seperti yang telah kita ketahui DHCP berguna untuk memberikan ip secara otomatis, pada lab kali ini kita akan memberikan IP secara otomatis ke client agar client tidak perlu susah lagi memberi IP secara static/manual maka dengan DHCP router akan memberi IP secara otomatis ke setiap client menurut vlan dengan IP yang berbeda per vlan nya.



Pada lab kali ini kita akan menggunakan satu router, satu switch dan 2 client yang mana kedua client tersebut berada pada vlan yang berbeda dan network yang otomatis berbeda juga

Kita dapat memulai konfigurasi dengan mengkonfigurasi kan vlan terlebih dahulu pada switch dan juga membagi client sesuai pada vlan nya.

KONFIGURASI VLAN :

```
Switch(config)#vlan 10
Switch(config-vlan)#int fa0/1
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 10
Switch(config-if)#vlan 20
Switch(config-vlan)#int fa0/2
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 20
```

Kemudian kita lanjut dengan mengkonfigurasi kan interface trunk pada interface switch yang mengarah ke router, dengan maksud agar router dapat melanjut kan trafic DHCP menuju client di tiap vlan nya yang telah di konfigurasi kan di router nanti

Pasang interface trunk :

```
Switch(config)#int fa0/3
Switch(config-if)#switchport mode trunk
```

Setelah mengkonfigurasi kan vlan dan juga interface trunk kita dapat lanjut konfigurasi di sisi router

Kita dapat memulai mengkonfigurasi kan intervlan routing, agar nanti nya kita PC/client dapat saling terhubng walau berbeda vlan

ROUTER :

Konfigurasi inter-vlan agar PC mendapat kan IP gateway dari router :

```
Router(config)#int fa0/0.10
Router(config-subif)#encapsulation dot1Q 10
Router(config-subif)#ip ad 10.10.10.1 255.255.255.0
Router(config-subif)#int fa0/0.20
Router(config-subif)#encapsulation dot1Q 20
Router(config-subif)#ip ad 20.20.20.1 255.255.255.0
```

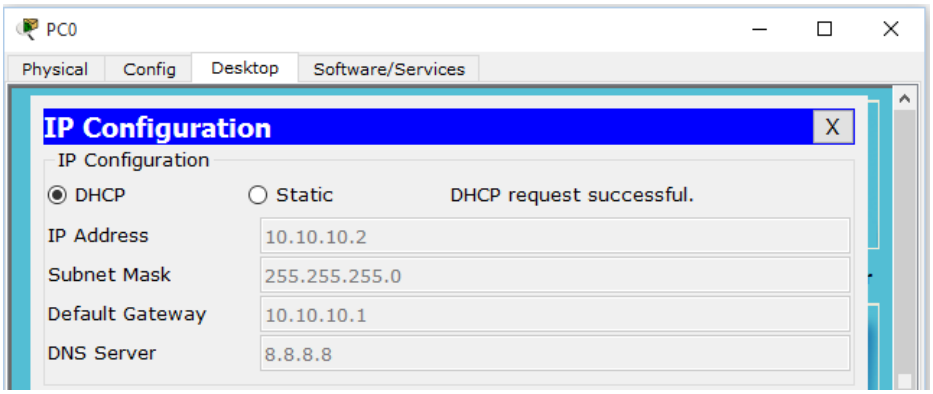
Setelah itu kita bisa dapat memulai mengkonfigurasi kan DHCP server pada router untuk membagi kan ip secara otomatis ke client menurut sesuai dengan vlan nya

Konfigurasi DHCP per vlan :

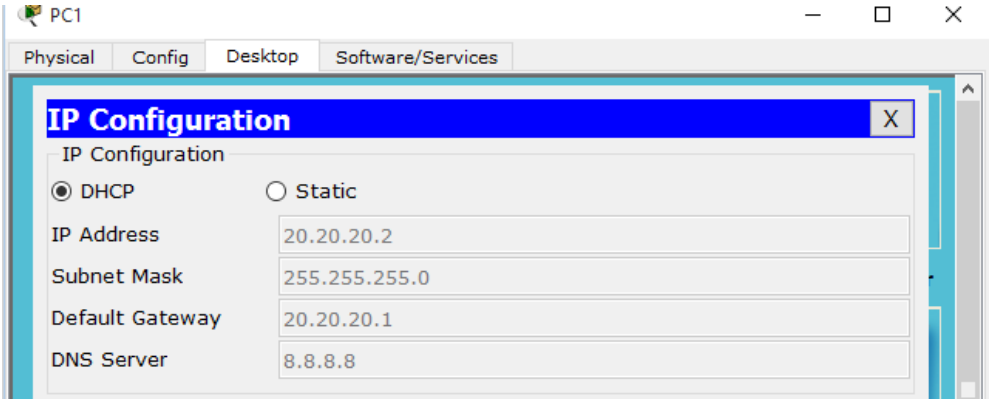
```
Router(config)#ip dhcp pool vlan10  :nama pool bebas
Router(dhcp-config)#network 10.10.10.0 255.255.255.0  :network IP
Router(dhcp-config)#default-router 10.10.10.1  :gateway dari
sub.interface
Router(dhcp-config)#dns-server 8.8.8.8  :DNS server
Router(dhcp-config)#ip dhcp pool vlan20
Router(dhcp-config)#network 20.20.20.0 255.255.255.0
Router(dhcp-config)#default-router 20.20.20.1
Router(dhcp-config)#dns-server 8.8.8.8
```

Setelah itu kita dapat mencoba melihat dengan merubah dari yang static menjadi dynamic

Client vlan 10 :



Client vlan20 :



Maka client/PC pun telah mendapat kan IP secara dynamic dari router

Verivikasi client yang mendapat kan IP secara DHCP dari router :

```
Router#show ip dhcp binding
IP address       Client-ID/              Lease expiration        Type
                 Hardware address
10.10.10.2       0010.11D0.9774           --                     Automatic
20.20.20.2       0001.9742.606C           --                     Automatic
Router#
```

Kita dapat meilahat terdapat IP yang di dapat kan client beserta MAC – address nya

Pada saat pemeberian IP DHCP kepada client untuk cisco maka dia akan memberikan IP dari yang terkecil, dan DHCP pun dapat kita konfigurasikan suatu pengecualian IP yang akan di beri kan kepada cliet, atau yang dapat di kata kan kita akan mengkonfigurasi kan IP yang tidak akan di beri kan ke client pada saat menkonfgurasi kan DHCP

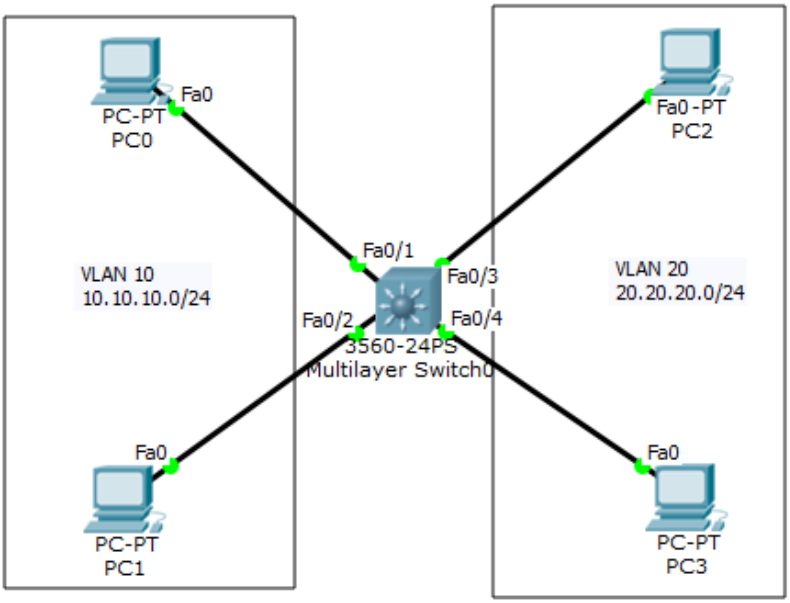
DHCP excluled address :

```
Router(config)#ip dhcp excluded-address 10.10.10.3
20.20.20.3
```

Maka IP 10.10.10.3 dan 20.20.20.3 tidak akan di beri kan kepada client

DHCP MLS

Perlu di ketahui untuk konfigurasi DHCP dapat juga di gunakan pada MLS dikarna kan perangakat MLS ini berjalan pada multi layer di layer 2 dan layer 3 yang ia Switch yang dapat menjalan kan fungsi routing juga



Topologi pada lab kali ini kita akan menggunakan satu perankat MLS yang di konfigurasikan VLAN10 dan VLAN 20 dengan ip sesuai dengan vlan nya, kemudian di konfigurasikan DHCP server pada MLS yang nanti nya setiap client mendapatkan IP yang berasal dari MLS secara otomatis/dyanmic .

KONFIGURASI VLAN :

```
Switch(config)#vlan 10
Switch(config-vlan)#vlan 20
Switch(config-vlan)#int ra fa0/1-2
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 10
Switch(config-if-range)#int ra fa0/3-4
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 20
```

Kemudian kita dapat lanjutkan dengan mengkonfigurasi kan SVI dengan maksud agar client yang berada pada sebuah vlan dapat saling terhubung dengan vlan yang lain (terhubung dengan network yang berbeda)

KONFIGURASI SVI :

```
Switch(config)#int vlan 10
Switch(config-if)#ip add 10.10.10.1 255.255.255.0
Switch(config-if)#int vlan 20
Switch(config-if)#ip add 20.20.20.1 255.255.255.0
Switch(config-if)#ip routing
```

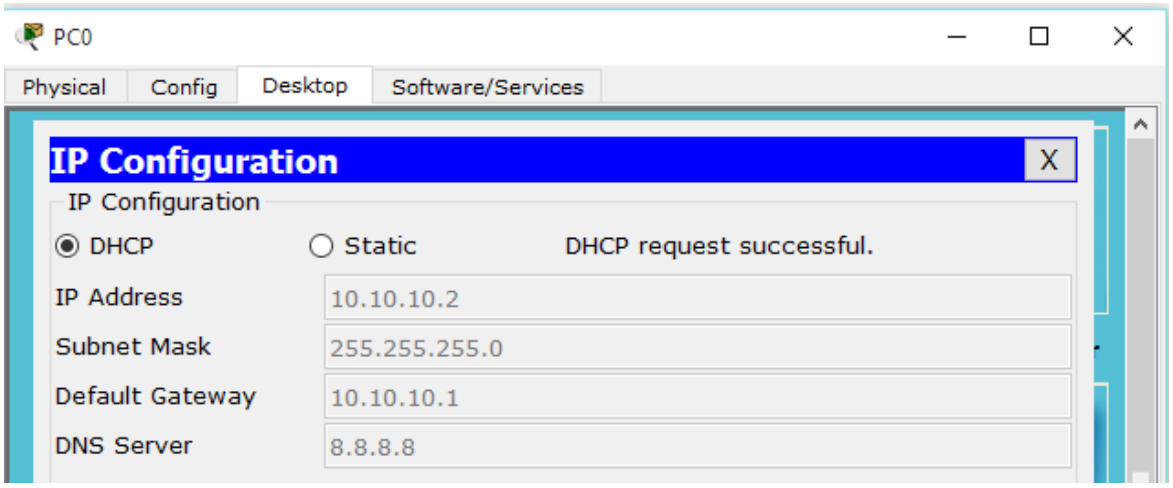
Setelah itu kita dapat lanjut dengan mengkonfigurasikan DHCP sever pada MLS

KONFIGURASI DHCP :

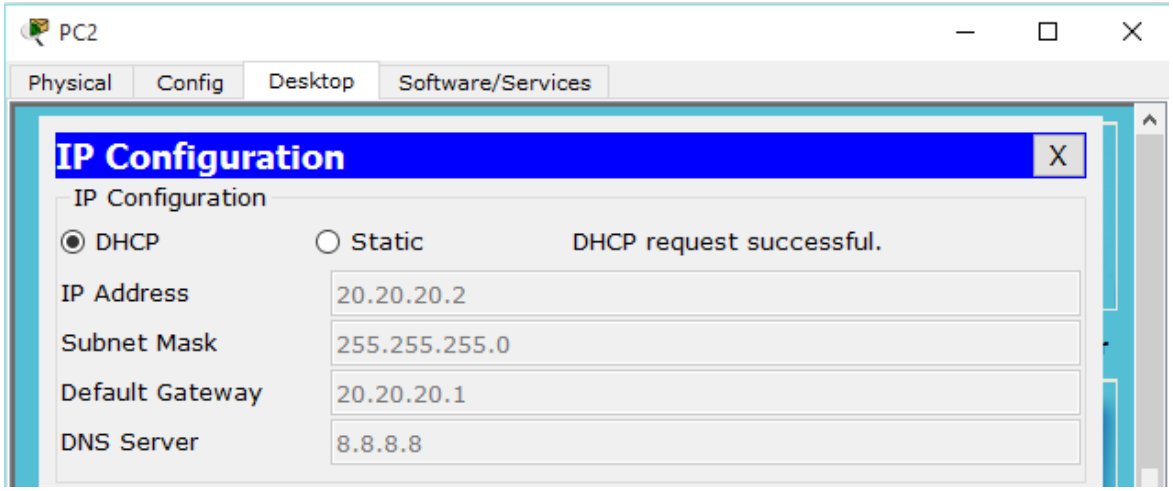
```
Switch(config)#ip dhcp pool vlan10
Switch(dhcp-config)#network 10.10.10.0 255.255.255.0
Switch(dhcp-config)#default-router 10.10.10.1
Switch(dhcp-config)#dns-server 8.8.8.8
Switch(dhcp-config)#ip dhcp pool vlan20
Switch(dhcp-config)#network 20.20.20.0 255.255.255.0
Switch(dhcp-config)#default-router 20.20.20.1
Switch(dhcp-config)#dns-server 8.8.8.8
```

Maka dengan ini kita telah mengkonfigurasi DHCP server, dan client/PC akan mendapat kan IP secara otomatis melalui DHCP sesuai dengan vlan nya.

Client Vlan10



Client Vlan20

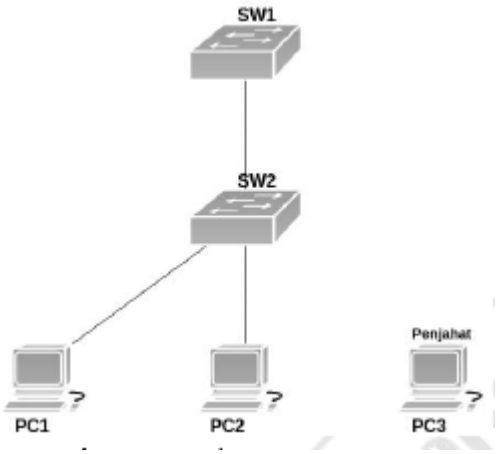


Maka dengan ini maka kita dapat DHCP telah berhasil dan juga sudah saling terhubung dengan client yang berbeda vlan

PORT – SECURITY

Port security merupakan sebuah fitur yang memungkinkan kita untuk mengamankan switch dari gangguan orang-orang yang tidak bertanggung jawab. Dengan mengaktifkan port security, nantinya interface pada switch bisa otomatis mati ketika ada orang yang tidak bertanggung jawab menghubungkan komputernya dengan switch.

Untuk praktik konfigurasi port security ini, kita akan menggunakan topologi seperti berikut



Berikut konfigurasi yang perlu kita lakukan di SW1 untuk mengaktifkan port Security

```
SW1(config)#interface fa0/1
SW1(config-if)#switchport port-security
Command rejected: FastEthernet0/1 is a dynamic port.
```

Perhatikan bahwa saat kita mencoba mengaktifkan port security, ada sebuah pesan error yang menunjukkan bahwa kita tidak bisa mengaktifkan port security pada dynamic port, sehingga kita harus merubah dulu mode port tersebut menjadi static

```
SW1(config)#interface fa0/1
SW1(config-if)#switchport mode access
SW1(config-if)#switchport port-security --> 1
SW1(config-if)#switchport port-security mac-address sticky --> 2
SW1(config-if)#switchport port-security maximum 2 --> 3
SW1(config-if)#switchport port-security violation shutdown --> 4
```

Perintah-perintah diatas digunakan untuk mengaktifkan port security pada interface fa0/1 SW1. Adapun penjelasan dari masing-masing perintah tersebut adalah sebagai berikut

1. Digunakan untuk mengaktifkan port security
2. Digunakan untuk mengkonfigurasikan metode dalam mendapatkan MAC Address. Ada dua metode yang dapat kita gunakana, yaitu static dan sticky. Sticky artinya switch akan mencatat MAC Address secara otomatis, MAC Address dari komputer pertama yang terhubung yang akan dicatat.
3. Digunakan untuk menentukan jumlah maximum device yang bisa connect
4. Digunakan untuk menentukan policy yang akan diterapkan saat ada device asing terhubung ke switch

Setelah mengaktifkan port security, kita coba lihat daftar mac address yang terhubung ke switch

```
SW1#show mac address-table
Mac Address Table
-------------------------------------------
Vlan Mac Address Type Ports
---- ----------- -------- -----
```

Perhatikan bahwa SW1 belum memiliki daftar mac address komputer yang terhubung dengan dirinya. Hal ini dikarenakan belum ada trafic sama sekali pada jaringan tersebut. Kita coba ping dari PC1 ke PC2 agar ada trafic yang beredar.

Setelah melakukan ping, kita coba lihat lagi tabel mac address di SW1

```
SW1#show mac address-table
Mac Address Table
-------------------------------------------
Vlan Mac Address Type Ports
---- ----------- -------- -----
1 0006.2aab.c874 STATIC Fa0/1
1 0009.7c83.00cc STATIC Fa0/1
```

Perhatikan bahwa saat ini ada dua mac address yang terdaftar di SW1. Kita

coba lihat status port security

```
SW1#show port-security
Secure Port MaxSecureAddr CurrentAddr SecurityViolation Security
Action
(Count) (Count) (Count)
---------------------------------------------------------------------
-
Fa0/1 2 2 0 Shutdown
---------------------------------------------------------------------
```

Perintah seperti diatas akan menunjukkan kepada kita status port security secara simpel. Untuk melihat status port security secara detail, gunakan perintah berikut

```
SW1#show port-security interface fa0/1
Port Security : Enabled
Port Status : Secure-up
Violation Mode : Shutdown
Aging Time : 0 mins
Aging Type : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses : 2
Total MAC Addresses : 2
Configured MAC Addresses : 0
Sticky MAC Addresses : 2
Last Source Address:Vlan : 0006.2AAB.C874:1
Security Violation Count : 0
```

Untuk melihat daftar mac address port security, kita bisa menggunakan perintah berikut

```
SW1#show port-security address
Secure Mac Address Table
------------------------------------------------------------------
Vlan Mac Address Type Ports Remaining Age
(mins)
---- ----------- ---- ----- -------------
1 0006.2AAB.C874 SecureSticky FastEthernet0/1 -
1 0009.7C83.00CC SecureSticky FastEthernet0/1 -
------------------------------------------------------------------
Total Addresses in System (excluding one mac per port) : 1
Max Addresses limit in System (excluding one mac per port) : 1024
```

Untuk pengujian, kita coba hubungkan PC penjahat ke switch, kemudian kita coba ping dari PC penjahat ke IP manapun, tujaunnya adalah agar ada trafic dari PC penjahat Sesaat setelah melakukan ping, maka akan ada peringatan yang menunjukkan bahwa interface pada SW1 berubah menjadi shutdown

```
SW1#
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to
administratively down
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1,
changed state to
down
```

Untuk lebih meyakinkan, kita coba lihat status port di SW1

```
SW1#show port-security interface fa0/1
Port Security : Enabled
Port Status : Secure-shutdown
Violation Mode : Shutdown
Aging Time : 0 mins
Aging Type : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses : 2
```

```
Total MAC Addresses : 2
Configured MAC Addresses : 0
Sticky MAC Addresses : 2
Last Source Address:Vlan : 000C.CF7E.98A7:1
Security Violation Count : 1
```

Perhatikan bahwa saat ini status dari interface fa0/1 adalah shutdown. Untuk mengaktifkannya kembali, kita harus shutdown kemudian no shutdown secara manual

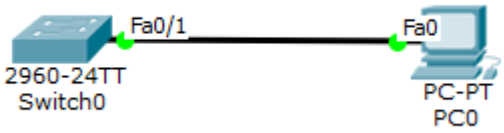
```
SW1(config)#interface fa0/1
SW1(config-if)#shutdown
SW1(config-if)#no shutdown
```

Pada contoh diatas, kita menggunakan violation shutdown, selain shutdown, ada dua violation lagi yang dapat kita gunakan pada port security. Berikut beberapa violation pada port security beserta penjelasannya

- **Shutdown** Interface akan shutdown saat ada PC asing yang konek
- **Protect** Data yang dikirimkan melalui interface tersebut tidak akan diforward (tidak dikirimkan)
- **Restrict** Sama halnya dengan protect, namun akan mengirimkan notifikasi SNMP

ENABLE SSH DAN TELNET

Telnet dan SSH merupakan sebuah protocol yang dapat kita gunakan untuk melakukan remote access pada sebuah perangkat yang pada lab kali ini kita akan melakukan konfigurasikan pada switch agar switch dapat di remote melalui SSH atau pun telnet.



Dalam pengaktifan telnet kita harus mengkonfigurasikan IP pada switch terlebih dahulu, tetapi dengan catatan karna switch memang sebenar nya tidak dapat di beri IP maka kita dapat memberi IP pada vlan di switch yaitu vlan 1/vlan default.

```
Switch(config)#int vlan 1
Switch(config-if)#no shutdown
Switch(config-if)#ip add 192.168.1.1 255.255.255.0
```

Setelah kita beri IP maka kita dapat test PING terlebih dahulu dari PC menuju switch

```
PC1>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:

Reply from 192.168.1.1: bytes=32 time=0ms TTL=255
Reply from 192.168.1.1: bytes=32 time=0ms TTL=255
Reply from 192.168.1.1: bytes=32 time=0ms TTL=255
Reply from 192.168.1.1: bytes=32 time=0ms TTL=255

Ping statistics for 192.168.1.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Apa bila sudah reply maka kita dapat langsung mengkonfigurasikan telnet pada switch dengan membuat user telnet nya

```
Switch(config)#username azhar password 123
Switch(config)#line vty 0 3
Switch(config-line)#login local
```

Line vty 0 3 merupakan jumlah user yang dapat login bersamaan untuk mengakses switch tersebut dan login local di gunakan agar user yang telah kita buat tadi di terapkan di line vty.

Setelah sudah maka kita dapat mencobaa melakukan telnet menuju switch dari PC

```
PC1>telnet 192.168.1.1
Trying 192.168.1.1 ...Open
```

```
User Access Verification

Username: azhar
Password:
Switch>
Switch>
```

Maka dapat dilihat bahwa kita sudah dapat mentelnet switch, kemudian kita dapat coba melakukan konfgurasi di switch

```
Switch>en
% No password set.
Switch>
```

Dapat di lihat bahwa ada pesan error yang berarti kita harus memasang password terlebih dahulu pada switch

```
Switch(config)#enable secret 456
```

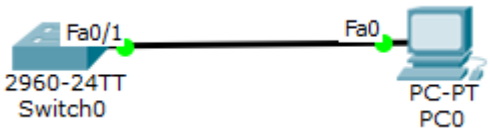
Maka setelah kita memasang password pada switch pada saat di telnet kita akan masukan password kembali dan sudah dapat melakukan konfigurasi pada switch melalui telnet.

```
PC>telnet 192.168.1.1
Trying 192.168.1.1 ...Open

User Access Verification

Username:azhar
Password:
Switch>en
Password:
Switch#conf t
Switch(config)#
```

Lalu bagaimana dengan SSH di switch..??

Pada dasar nya sebenar nya dilapangan telnet sudah jarang di pakai, hal ini di karna kan telnet tidak melakukan enskripsi terhadap packet yang di lewat kan, sehinggga packet kurang aman dan sangat mudah di ketahui oleh para orang yang kurang bertanggung jawab, maka dengan itu kita dapat melakukan remote access melalui SSH

Dan untuk topologi pada SSH kita sama menggunakan topologi pada telnet, tetapi kita hanya merubah service telnet menjadi SSH



KONFIGURASI SSH :

```
Switch(config)#username azhar1 password 123
Switch(config)#ip domain-name mq.com
Switch(config)#crypto key generate rsa
% Please define a hostname other than Switch.
```

Kita harus membuat username terlebih dahulu untuk mengaktifkan SSH apabila tidak akan keluar pesan error untuk membuat username

```
Switch(config)#ho SW1
SW1 (config)#crypto key generate rsa
The name for the keys will be: SW1.idn.id
Choose the size of the key modulus in the range of 360 to 2048
for your
General Purpose Keys. Choosing a key modulus greater than 512
may take
a few minutes.

How many bits in the modulus [512]:
% Generating 512 bit RSA keys, keys will be non-
exportable...[OK]

SW1 (config)#line vty 0 4
SW1 (config-line)#transport input ssh
SW1 (config-line)#login local
```

Dengan itu coba kita dapat mencoba meremote switch dengan SSH

```
PC>ssh -l ssh 192.168.1.1
Open
Password:
SW1>enable
Password:
SW1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW1 (config)#
```

Maka switch sudah dapat di remote access melalui SSH

SPANNING – TREE PROTOCOL (STP)

Spanning Tree Protocol (STP) merupakan protocol yang berfungsi mencegah loop pada switch ketika switch menggunakan lebih dari 1 link dengan maksud redundancy. STP secara defaultnya diset aktif pada Cisco Catalyst. STP merupakan open standard (IEEE 802.1D).

Ada beberapa jenis STP:

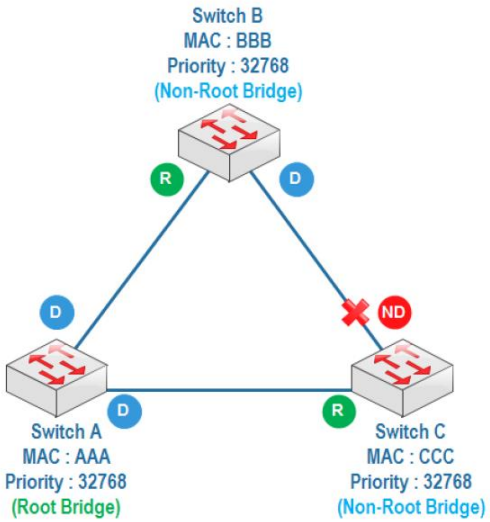
- Open Standard : STP (802.1D), Rapid STP (802.1W), Multiple Spanning Tree MST (802.1S)
- Cisco Proprietary : PVST (Per Vlan Spanning Tree), PVST+, Rapid PVST.

Ketika Switch0 mengirim packet data dengan destination yang tidak terdapat pada MAC address tabelnya, maka Switch0 akan membroadcast ke semua port sampai ke Switch1. Jika pada tabel MAC address Switch1 juga tidak terdapat destination tadi maka Switch1 akan kembali membroadcast ke Switch0 dan akan seperti itu sehingga network down.

Ada beberapa cara mengatasi hal tersebut:

- Hanya menggunakan 1 link (no redundancy)
- Shutdown salah satu interface, melakukan shutdown manual pada salah satu interface atau secara otomatis menggunakan STP.

STP akan membuat blocking atau shutdown pada salahsatu port untuk mencegah terjadinya loop. Ketika link utama down maka port yang sebelumnya blocking akan menjadi forward. Port blocking ditunjukkan dengan warna merah.



Cara kerja STP :

1. Ketika STP aktif, masing-masing switch akan mengirimkan frame khusus satu sama lain yang disebut *Bridge Protocol Data Unit (BPDU)*.

2. Menentukan Root Bridge

Switch dengan bridge id terendah akan menjadi root bridge. Bridge id = priority + MAC address. Dalam satu LAN hanya ada satu switch sebagai root bridge, switch lain menjadi non-root bridge. Default priority adalah 32768 dan bisa diubah.

3. Menentukan Root Port

Yang menjadi root port adalah path yang paling dekat dengan root bridge. Untuk setiap non-root bridge hanya punya 1 root port.

4. Menentukan designated port dan non-designated port

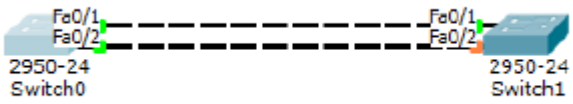
Designated port adalah port yang forward dan non designated port adalah port yang blocking. Untuk root bridge semua portnya adalah designated port. Switch dengan priority terendah, salah satu portnya akan menjadi nondesignated port atau port blocking. Jika priority sama maka akan dilihat MAC address terendah.

STP akan membuat blocking atau shutdown pada salahsatu port untuk mencegah terjadinya loop. Ketika link utama down maka port yang sebelumnya blocking akan menjadi forward. Port blocking ditunjukkan dengan warna merah.

STP menggunakan link cost calculation untuk menentukan root port pada non-root switch.

ROOT BRIDGE STP

Kali ini kita akan menetukan switch yang akan menjadi sebuah root bridge, dengan mengecilkan prioirty nya atau yang priority nya paling kecil dari yang lain nya.




```
Switch0#show spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    32769
              Address     0002.BA60.D273
              Cost        19
              Port        1(FastEthernet0/1)
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32769  (priority 32768 sys-id-ext 1)
              Address     00D0.FFDA.ECBC
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
-
Fa0/2            Altn BLK 19        128.2    P2p
Fa0/1            Root FWD 19        128.1    P2p

Switch0#
```

Secara otomatis, Switch0 menjadi root bridge di karna kan mac – address nya yang paling kecil salah satu cara agar menjadi root bridge dengan cara di lihat dari IP – loopback atau dilihat dari sedari priority nyamua portnya yang fordward (berwarna hijau), agar Switch1 yang menjadi root bridge, ubah priority pada Switch1.

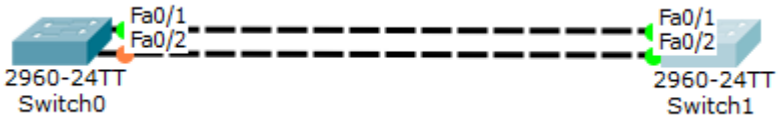
Ubah priority pada vlan nya dalam spanning – tree :

```
Switch(config)#spanning-tree vlan 1 priority 12288
```

Besar prioritry dapat di pilih dari 1 – 61440 , tetapi kita harus memasukan angka – angka nya yang lebih spesifik yang sudah ada pilihan nya yaitu :

0,4096,8192,12288,16384,20480,24576,28672,32768,36864,40960,
45056,49152,53248,57344,61440

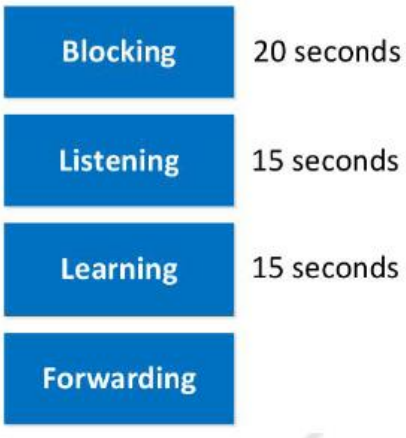
Dan priority default dari spanning – tree adalah 32768 dan 1 nya itu di tambah dari vlan nya



Maka switch yang menjadi root bridge yaitu yang priority nya lebih kecil .

STP PORT – FAST

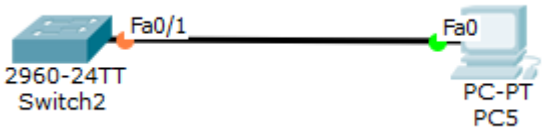
Spanning tree port fast merupakan salah satu fitur STP,yang mana di saat kita menancapkan kabel pada switch maka kita akan melewati beberapa sesi, sampai yang akhir nya menjadi forwading, dengan Spanning tree port fast ini kita akan di percepat dalam melewati beberapa proses tersebut.



Yang di mana nanti nya switch akan melewati step blocking sekitar 20 detik kemudian melewati step listening sampai 15 detik lalu learning sampai 15 detik dan kemudian sampai lah pada step forwading, dan apabila kita menginginkan agar dapat langsung melewati dari step blocking langsung ke step forward tanpa

harus melewati listening dan learning terlebih dahulu maka di butuhkan lah spanning tree port fast.

Port fast ini cocok di gunakan untuk port yang mengarah ke end host, tetapi tidak di rekomendsikan untuk port yang mengarah ke switch karena akan menonaktifkan fungsi STP dalam mencegah looping.



Langsung saja kita konfigurasikan di interface fa0/1 yang ingin di konfigurasikan spanning STP.

```
Switch(config)#int fa0/1
Switch(config-if)#spanning-tree portfast
```

Hanya dengan itu maka pada saat host mencolokan kembali ke port yang sudah di konfigurasikan ia akan langsung ke step forward atau kalau di di cisco packet tracer lampu nya akan langsung hijau tidak oren terlebih dahulu seperti pada gambar di atas.

ETHERCHANNEL

Kalau pada lab sebelum nya kita membahas tentang yang nama nya spanning tree protocol (STP) yang mana membuat beberapa interface kita block dan menyisakan satu interface agar tidak membuat looping, dan pada lab kali ini kita akan mengambung kan beberapa interface/link dan menggambungkan menjadi satu interface/link yang mana kita harus menoaktifkan STP yang mana tidak ada yang nama nya blocking port.

Dalam etherchanel terdapat 3 protocol :

- **LACP** (Link Aggregation Control Protocol) – open standard IEEE 802.1AD.

Yang mana ia telah open, pada perangkat yang lain yang terbagi menjai beberapa mode :

- ✓ **Active** : yang arti nya ia mengajak untuk di jadikan etherchannel LACP
- ✓ **Passive** : yang arti nya ia akan menuggu di ajak menjadi ethercahnnel

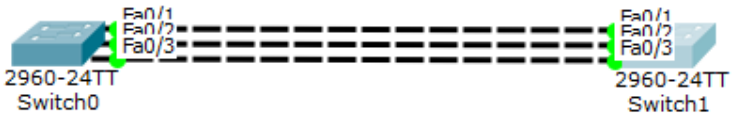
- **PAGP** (Port Aggregation Protocol) – cisco proprietary.

PAGP merupakan yang merupakan cisco proprietary yang pada saat ini hanya masih di miliki oleh cisco, dan pada PAGP terdapat beberapa mode :

- ✓ **Disarable** : yang arti nya ia akan mengajak untuk menjadi etherchannel
- ✓ **Auto** : yaitu sebalik nya ia akan menuggu untuk di jadikan etherchannel

- **Etherchannel layer 3** : yang mana ia akan menggunakan layer 3 yaitu IP yang mana kita akan menggunakan MPLS, dalam etherchannel layer3 hanya memiliki 1 mode :
- ✓ **ON** : mode ini sama saja dengan mengajak

Untuk etherchannel LACP dan PAGP kita akan mengunaka topologi yang sama, dengan dua switch.



Langsung saja kita ke konfigurasi.

etherchannel LACP

Konfiguarsikan trunk :

```
SW1(config)#int ra fa0/1-3
SW1(config-if-range)#switchport mode trunk
```

```
SW2(config)#int ra fa0/1-3
SW2(config-if-range)#switchport mode trunk
```

Konfiguarsikan LACP untuk yang satu nya ngajak dan yang satu nya nunggu atau bisa juga dengan ke dua nya sama – sama mengajak :

```
SW1(config-if-range)#channel-group 1 mode active
```

```
SW2(config-if-range)#channel-group 1 mode passive
```

Untuk melihat status etherchannel dengan :

SW1

```
SW1#show etherchannel summary
Flags:  D - down        P - in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------

1      Po1(SU)           LACP   Fa0/1(P) Fa0/2(P) Fa0/3(P)
SW1#
```

SW2

```
SW2#show etherchannel summary
Flags:  D - down        P - in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------

1      Po1(SU)           LACP   Fa0/1(P) Fa0/2(P) Fa0/3(P)
SW2#
```

Etherchannel PAGP

Konfigurasikan trunk :

```
SW1(config)#int ra fa0/1-3
SW1(config-if-range)#switchport mode trunk

SW2(config)#int ra fa0/1-3
SW2config-if-range)#switchport mode trunk
```

Konfigurasikan etherchannel :

```
SW1(config-if-range)#channel-group 1 mode desirable

SW2(config-if-range)#channel-group 1 mode auto
```

Setelah itu kita dapat melihat konfigureasi etherchannel :

SW1

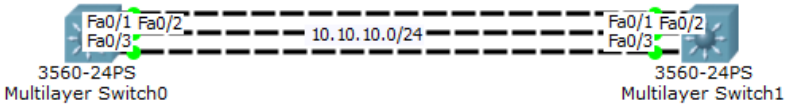
```
SW1#show etherchannel summary
Flags:  D - down        P - in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------

1      Po1(SU)           PAgP   Fa0/1(P) Fa0/2(P) Fa0/3(P)
```

SW2

```
SW2#show etherchannel summary
Flags:  D - down        P - in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------

1      Po1(SU)           PAgP   Fa0/1(P) Fa0/2(P) Fa0/3(P)
```

Etherchannel layer 3 :

Dalam etherchannel layer 3 kita akan mengunakan multi layer switch atau MPLS yang mana pada layer 3 kita akan mengunakan IP, dan akan menonaktifkan fungsi switch.
Pada lab kali ini kita akan menggunakan 2 perangkat MPLS.



Dalam etherchannel layer 3 kita tidak perlu membuat interface trunk, jadi kita langsung saja membuat etherchannel dengan mode on

```
SW1(config)#int ra fa0/1-3
SW1(config-if-range)#channel-group mode on

SW2(config)#int ra fa0/1-3
SW2(config-if-range)#channel-group 1 mode on
```

Setelah kita buat interface channel-group tersebut maka kita harus masuk ke interface tersebut kemudian menonaktifkan fungsi switch agar bisa di beri IP.

```
SW1(config)#int port-channel 1  :masuk ke interface port channel
SW1(config-if)#no switchport  :  nonaktifkan fungsi switch
SW1(config-if)#ip ad 10.10.10.1 255.255.255.0

SW2(config)#int port-channel 1
SW2(config-if)#no switchport
SW2(config-if)#ip ad 10.10.10.2 255.255.255.0
```

SWITCH STACKING

Switch stacking merupakan salah satu cara untuk menggambungkan beberapa switch menjadi satu yang dalam kata lain kita seolah – olah menggunakan satu switch saja, dan port nya pun bertambah

Apabila sudah di sambungkan dengan kabel stacking maka kita hanya sedikit mengkonfigurasikan reload pada switch

```
Switch(config)#do reload
```

Kemudian kita dapat coba cek interface pada switch

```
Switch(config)#do sh ip interface brief
```

maka interfacepun akan bertambah sesuai dengan port yang berada pada switch – switch yang di stacking

ROUTING

STATIC ROUTE
DYNAMIC ROUTE
OSPF EIGRP
STANDARD – EXTENDED ACCESS-LIST
HIGH AVAIBILITY
GRE TUNNEL
NAT

STATIC ROUTE 1

Static router merupakan suatu mekanisme dalam routing yang mana kita akan mengkonfigurasi kan suatu network agar dapat saling terhubung dengan network lain nya dengan mengkonfigurasi secara static/manual .

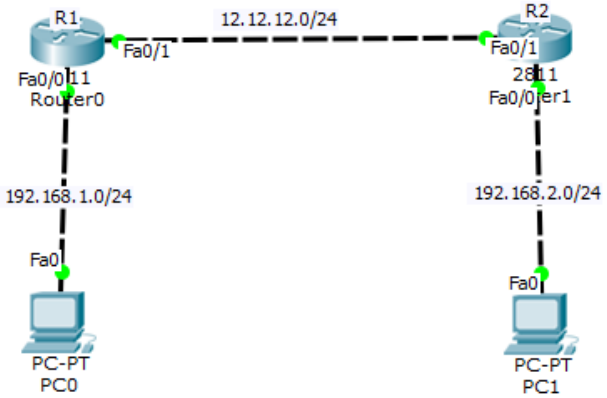
Untuk static route sendiri di konfigurasikan secara manual untuk menentukan jalur nya dan jalur nya, jadi semakin banyak routing banyak konfigurasi yang kita lakukan, salah satu keunggulan static route ia memiliki Administrative Distance (AD) 1 yang ia akan lebih di pilih dari pada routing protocol – protocol routing lain nya.

Dan di balik keunggulan static route sendiri juga memiliki beberapa kekurangan :

- No CPU cycles are used to calculate and communicate routes.
- The path a static route uses to send data is known.
- Konfigurasi dan maintenance yang memakan waktu
- Tidak cocok untuk network skala besar.
- Untuk jaringan kecil yang tidak akan terjadi perubahan topologi secara significant
- hanya mempunyai 1 exit path (karena hanya mempunyai satu neighbor).
- untuk unknown network menggunakan default route

untuk topologi untuk staic route juga kita akan menggunakan 2 router 2 host :

router dan 2 host



Kita akan mengkonfigurasikan static route agar ke dua PC dapat saling ping.

Konfigurasikan IP pada masing – masing interface sesuai pada topologi :

R1

```
R1(config)#int fa0/0
R1(config-if)#no sh
R1(config-if)#ip ad 192.168.1.1 255.255.255.0
R1(config-if)#int fa0/1
R1(config-if)#no sh
R1(config-if)#ip ad 12.12.12.1 255.255.255.0
```

R2

```
R2(config)#int fa0/0
R2(config-if)#no sh
R2(config-if)#ip ad 192.168.2.1 255.255.255.0
R2(config-if)#int fa0/1
R2(config-if)#no sh
R2(config-if)#ip ad 12.12.12.2 255.255.255.0
```

Sebelum kita mengkonfigurasikan static route kita dapat melihat tabel routing terlebih dahulu

R1

```
R1#show ip route
Codes: C - connected, S - static, I - IGRP, 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

     12.0.0.0/24 is subnetted, 1 subnets
C       12.12.12.0 is directly connected, FastEthernet0/1
C    192.168.1.0/24 is directly connected, FastEthernet0/0
```

R2

```
R2#sh ip route
Codes: C - connected, S - static, I - IGRP, 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

     12.0.0.0/24 is subnetted, 1 subnets
C       12.12.12.0 is directly connected, FastEthernet0/1
C    192.168.2.0/24 is directly connected, FastEthernet0/0
```

Kita dapat lihat bahwa pada tabel routing R1 belum mengenal network 192.168.2.0/24 yang berada di router R2 dan sebalik nya pada R2 juga pun belum mengenal network 192.168.1.0/24 yang berada pada router R1atau dalam kata lain setiap router memiliki tabel routing hanya yang langsung terhubung (Directly Conected)

Maka dari itu kita harus mengkonfigurasi kan static route agar setiap router mengatahui network yang tidak saling terhubung

Konfiguarsi static route :

Konfigurasikan IP untuk destination(tujuan) nya dengan subnetmask nya setelah itu gateway nya yang mengarah menuju link tersebut

```
R1(config)#ip route 192.168.2.0 255.255.255.0 12.12.12.2
```

```
R2(config)#ip route 192.168.1.0 255.255.255.0 12.12.12.1
```

Setelah sudah maka kita dapat cek kembali tabel routing di kedua router

R1

```
R1#sh ip route
Codes: C - connected, S - static, I - IGRP, 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

     12.0.0.0/24 is subnetted, 1 subnets
C       12.12.12.0 is directly connected, FastEthernet0/1
C    192.168.1.0/24 is directly connected, FastEthernet0/0
S    192.168.2.0/24 [1/0] via 12.12.12.2
R1#
```

R2

```
R2#sh ip route
Codes: C - connected, S - static, I - IGRP, 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

     12.0.0.0/24 is subnetted, 1 subnets
C       12.12.12.0 is directly connected, FastEthernet0/1
S    192.168.1.0/24 [1/0] via 12.12.12.1
C    192.168.2.0/24 is directly connected, FastEthernet0/0
```

Maka kedua network di kedua router sudah terdaftarkan di masing – masing router, dengan status code "*S*" yang berarti static.

Setelah itu coba kita dapat test ping antar PC kemudian kita lihat dengan traceroot jalur pengiriman data nya.

```
PC1>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Reply from 192.168.2.2: bytes=32 time=0ms TTL=126
Reply from 192.168.2.2: bytes=32 time=0ms TTL=126
Reply from 192.168.2.2: bytes=32 time=0ms TTL=126
Reply from 192.168.2.2: bytes=32 time=0ms TTL=126

Ping statistics for 192.168.2.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
PC2>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time=2ms TTL=126
Reply from 192.168.1.2: bytes=32 time=0ms TTL=126
```

```
Reply from 192.168.1.2: bytes=32 time=0ms TTL=126
Reply from 192.168.1.2: bytes=32 time=0ms TTL=126

Ping statistics for 192.168.1.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 2ms, Average = 0ms

PC>
```

Maka hasil nya pun reply, dan setelah kita test ping antar PC, maka untuk mengetahui jalur pengeriman data nya maka kita dapat tracert untuk menuju destination nya dengan tracerroot.

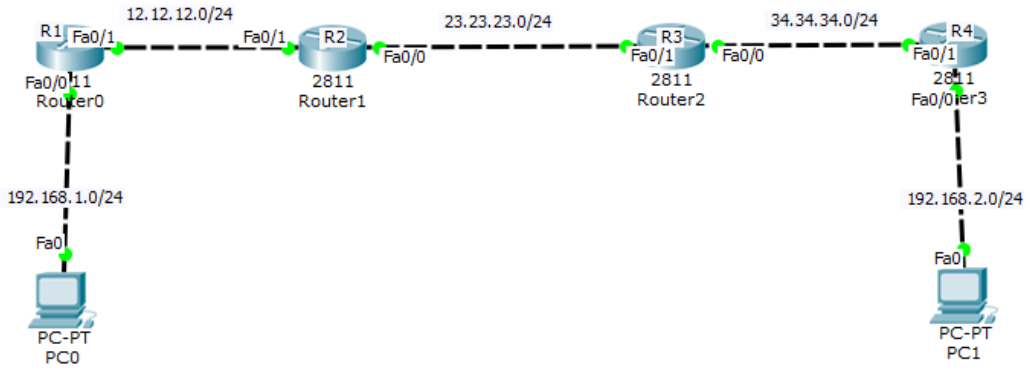
```
PC>tracert 192.168.2.2

Tracing route to 192.168.2.2 over a maximum of 30 hops:

1 1 ms 0 ms 0 ms 192.168.1.1
2 0 ms 0 ms 0 ms 12.12.12.2 :gateway (ip R1)
3 0 ms 0 ms 0 ms 192.168.2.2

Trace complete.
```

```
PC2>tracert 192.168.1.2

Tracing route to 192.168.1.2 over a maximum of 30 hops:

1 0 ms 0 ms 0 ms 192.168.2.1
2 0 ms 2 ms 0 ms 12.12.12.1
3 0 ms 0 ms 0 ms 192.168.1.2

Trace complete.
```

Maka kedua network sudah dapat saling terhubung dengan mengkonfigurasikan static route

STATIC ROUTE SCENARIO 2

Pada lab kali ini kita akan mengkonfigurasi kan sebuah static route dengan topologi yang lebih dari pada topoplogi sebelum nya, yaitu kita akan menggunakan 4 router yang mana semua berbeda network, agar setiap router yang berbeda network saling terhubung maka kita dapat mengkonfigurasi kan static route.



Kita akan mengkonfigurasi kan staic route menuju destination dengan gateway yang sesuai dengan destination nya agar setiap router memiliki tabel routing yang lengkap seluruh network pada setiap network pada masing – masing router

R1

```
R1(config)#int fa0/0
R1(config-if)#no sh
R1(config-if)#ip ad 192.168.1.1 255.255.255.0
R1(config-if)#int fa0/1
R1(config-if)#ip ad 12.12.12.1 255.255.255.0
R1(config-if)#no sh
R1(config)#ip route 23.23.23.0 255.255.255.0 12.12.12.2
R1(config)#ip route 34.34.34.0 255.255.255.0 12.12.12.2
R1(config)#ip route 192.168.2.0 255.255.255.0 12.12.12.2
```

R2

```
R2(config)#int fa0/0
R2(config-if)#no sh
R2(config-if)#ip ad 23.23.23.2 255.255.255.0
R2(config-if)#int fa0/1
R2(config-if)#no sh
R2(config-if)#ip ad 12.12.12.2 255.255.255.0
R2(config)#ip route 192.168.1.0 255.255.255.0 12.12.12.1
R2(config)#ip route 34.34.34.0 255.255.255.0 23.23.23.3
R2(config)#ip route 192.168.2.0 255.255.255.0 23.23.23.3
```

R3

```
R3(config)#int fa0/1
R3(config-if)#no sh
R3(config-if)#ip ad 23.23.23.3 255.255.255.0
R3(config-if)#int fa0/0
R3(config-if)#no sh
R3(config-if)#ip ad 34.34.34.3 255.255.255.0
R3(config)#ip route 192.168.1.0 255.255.255.0 23.23.23.2
R3(config)#ip route 12.12.12.0 255.255.255.0 23.23.23.2
R3(config)#ip route 192.168.2.0 255.255.255.0 34.34.34.4
```

R4

```
R4(config)#int fa0/1
R4(config-if)#ip ad 34.34.34.4 255.255.255.0
R4(config-if)#int fa 0/0
R4(config-if)#no sh
R4(config-if)#ip ad 192.168.2.1 255.255.255.0
R4(config)#ip route 192.168.1.0 255.255.255.0 34.34.34.3
R4(config)#ip route 12.12.12.0 255.255.255.0 34.34.34.3
R4(config)#ip route 23.23.23.0 255.255.255.0 34.34.34.3
```

Dan setelah itu kita dapat mencek tabel routing di setiap router dan pastikan kalau semua network tetatngga antar router sudah terdaftarkan dengan status nya "S" yang berarti static.

R1

```
R1#sh ip route
Codes: C - connected, S - static, I - IGRP, 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

     12.0.0.0/24 is subnetted, 1 subnets
C       12.12.12.0 is directly connected, FastEthernet0/1
     23.0.0.0/24 is subnetted, 1 subnets
S       23.23.23.0 [1/0] via 12.12.12.2
     34.0.0.0/24 is subnetted, 1 subnets
S       34.34.34.0 [1/0] via 12.12.12.2
C    192.168.1.0/24 is directly connected, FastEthernet0/0
S    192.168.2.0/24 [1/0] via 12.12.12.2
R1#
```

R2

```
R2#sh ip ro
Codes: C - connected, S - static, I - IGRP, 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

     12.0.0.0/24 is subnetted, 1 subnets
C       12.12.12.0 is directly connected, FastEthernet0/1
     23.0.0.0/24 is subnetted, 1 subnets
C       23.23.23.0 is directly connected, FastEthernet0/0
     34.0.0.0/24 is subnetted, 1 subnets
S       34.34.34.0 [1/0] via 23.23.23.3
S    192.168.1.0/24 [1/0] via 12.12.12.1
S    192.168.2.0/24 [1/0] via 23.23.23.3
```

R3

```
R3#sh ip route
Codes: C - connected, S - static, I - IGRP, 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

     12.0.0.0/24 is subnetted, 1 subnets
S       12.12.12.0 [1/0] via 23.23.23.2
     23.0.0.0/24 is subnetted, 1 subnets
C       23.23.23.0 is directly connected, FastEthernet0/1
     34.0.0.0/24 is subnetted, 1 subnets
C       34.34.34.0 is directly connected, FastEthernet0/0
S    192.168.1.0/24 [1/0] via 23.23.23.2
S    192.168.2.0/24 [1/0] via 34.34.34.4
```

R4

```
R4#sh ip route
Codes: C - connected, S - static, I - IGRP, 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

     12.0.0.0/24 is subnetted, 1 subnets
S       12.12.12.0 [1/0] via 34.34.34.3
     23.0.0.0/24 is subnetted, 1 subnets
S       23.23.23.0 [1/0] via 34.34.34.3
     34.0.0.0/24 is subnetted, 1 subnets
C       34.34.34.0 is directly connected, FastEthernet0/1
S    192.168.1.0/24 [1/0] via 34.34.34.3
C    192.168.2.0/24 is directly connected, FastEthernet0/0
```

Setelah sudah lengkap semua tabel routing di setiap router nya maka coba kita dengan test ping antar PC

PC1 ke PC2

```
PC1>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Reply from 192.168.2.2: bytes=32 time=1ms TTL=124
Reply from 192.168.2.2: bytes=32 time=11ms TTL=124
Reply from 192.168.2.2: bytes=32 time=0ms TTL=124
Reply from 192.168.2.2: bytes=32 time=0ms TTL=124

Ping statistics for 192.168.2.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 11ms, Average = 3ms
```

Untuk melihat arah jalur packet nya bisa kita tracerroot

```
PC>tracert 192.168.2.2

Tracing route to 192.168.2.2 over a maximum of 30 hops:

1 0 ms 0 ms 0 ms 192.168.1.1 : IP PC1
2 0 ms 0 ms 0 ms 12.12.12.2 :  gateway R2
3 0 ms 0 ms 0 ms 23.23.23.3 :  R3
4 0 ms 0 ms 0 ms 34.34.34.4 :  R4
5 0 ms 0 ms 0 ms 192.168.2.2 : ip PC2

Trace complete.
```

Dengan ini maka routing static telah sukses kita konfigurasi kan, perlu di perhatikan pada umum nya routing static mendapat kan trouble shoot pada bagian pengisian gateway pada router untuk menuju destination

ROUTING DYNAMIC EIGRP

Kalau pada lab sebelum nya kita telah membahas tentang routing dengan static/manual, maka pada lab kali ini kita akan membahas salah satu protocol routing dynamic yaitu EIGRP *(Enhanced Interior Gateway Protocol)*

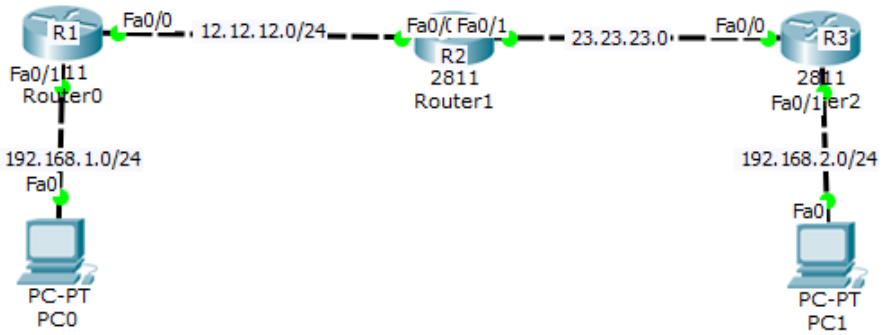
EIGRP merupakan salah satu protocol dalam dynamic route yang hanya di miliki oleh cisco, yang dalam kata lain routing EIGRP ini merupakan salah satu dari cisco propitary yang mana ia hanya dapat di gunakan pada perankat cisco saja

Routing EIGRP ini memiliki administrative distance sebanyak 90, update dalam routing EIGRP mengunakan multicast : 224.0.0.10, jumlah maksimal hop count nya 255 (default 100), memiliki konvergensi yang cepat, pengirirman hello packet di kirim setiap 5 second (dead interval 15 second), mendukung equal dan unequal cost load balancing.

Keuntungan routing EIGRP yaitu Terdapat backup route jika best route down (successor=primary, feasible successor=backup) dan ia mendukung VLSM.

Routing EIGRP menggunakan autonomous system number (ASN) untuk mengidentifikasi router – router yang sharing informasi route, atau yang dapat di artikan hanya router yang memiliki ASN yang bisa shraring informasi route.

Untuk topolgi pada lab kali ini kita akan menggunakan 3 router dan 2 client yang mana kita akan menghubung kan beberapa network yang berbeda dengan menggunakan routing EIGRP .



Sebelum kita konfigurasi routing pada router maka kita harus konfigurasikan terlebih dahulu ip address pada setiap router sesuai pada topology :

```
R1(config)#int fa0/1
R1(config-if)#no sh
R1(config-if)#ip ad 192.168.1.1 255.255.255.0
R1(config-if)#int fa0/0
R1(config-if)#no sh
R1(config-if)#ip ad 12.12.12.1 255.255.255.0
```

R2

```
R2(config-if)#int fa0/0
R2(config-if)#no sh
R2(config-if)#ip ad 12.12.12.2 255.255.255.0
R2(config-if)#int fa0/1
R2(config-if)#no sh
R2(config-if)#ip ad 23.23.23.2 255.255.255.0
```

R3

```
R3(config)#int fa0/1
R3(config-if)#no sh
R3(config-if)#ip ad 192.168.2.1 255.255.255.0
R3(config-if)#int fa0/0
R3(config-if)#no sh
R3(config-if)#ip ad 23.23.23.3 255.255.255.0
```

Setelah mengkonfigurasi kan ip address pada setiap interface maka kita dapat memulai mengkonfigurasi kan routing EIGRP

Karna default dari routing EIGRP classfull apabila kita ingin mengkonfigurasi kan dengan IP clasess maka kita harus mengkonfigurasikan "*no auto summary*", dan kita harus sama saat memasukan AS number nya apabila berbeda maka router tidak akan bisa bertukar informasi routing nya

Konfigurasi routing EIGRP:

```
R1(config)#router eigrp 10
R1(config-router)#no auto-summary
R1(config-router)#network 192.168.1.0
R1(config-router)#network 12.12.12.0
```

```
R2(config)#router eigrp 10
R2(config-router)#no auto-summary
R2(config-router)#network 12.12.12.0
R2(config-router)#network 23.23.23.0
```

```
R3(config)#router eigrp 10
R3(config-router)#no auto-summary
R3(config-router)#network 192.168.2.0
R3(config-router)#network 23.23.23.0
```

Setelah itu kita dapat meilihat tabel routing dari masing – masing router pastikan di setiap router memiliki tabel routing yang lengkap pada semua network, yang akan memiliki status "D" yang berarti EIGRP :

R1

```
R1#sh ip route
Codes: C - connected, S - static, I - IGRP, 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

     12.0.0.0/24 is subnetted, 1 subnets
C       12.12.12.0 is directly connected, FastEthernet0/0
     23.0.0.0/24 is subnetted, 1 subnets
D       23.23.23.0 [90/30720] via 12.12.12.2, 00:04:38, FastEthernet0/0
C    192.168.1.0/24 is directly connected, FastEthernet0/1
D    192.168.2.0/24 [90/33280] via 12.12.12.2, 00:03:34, FastEthernet0/0
```

R2

```
R2#sh ip route
Codes: C - connected, S - static, I - IGRP, 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

     12.0.0.0/24 is subnetted, 1 subnets
C       12.12.12.0 is directly connected, FastEthernet0/0
     23.0.0.0/24 is subnetted, 1 subnets
C       23.23.23.0 is directly connected, FastEthernet0/1
D    192.168.1.0/24 [90/30720] via 12.12.12.1, 00:05:26, FastEthernet0/0
D    192.168.2.0/24 [90/30720] via 23.23.23.3, 00:04:13, FastEthernet0/1
```

R3

```
R3#sh ip route
Codes: C - connected, S - static, I - IGRP, 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

     12.0.0.0/24 is subnetted, 1 subnets
D       12.12.12.0 [90/30720] via 23.23.23.2, 00:04:31, FastEthernet0/0
     23.0.0.0/24 is subnetted, 1 subnets
C       23.23.23.0 is directly connected, FastEthernet0/0
D    192.168.1.0/24 [90/33280] via 23.23.23.2, 00:04:31, FastEthernet0/0
C    192.168.2.0/24 is directly connected, FastEthernet0/1
```

Setelah masing – masing router sudah memiliki tabel routing yang kengkap maka kita dapat coba test ping antar PC1 dengan PC2 apakah sudah dapat saling terhubung atau reply :

PC1 > PC2

```
PC1>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Reply from 192.168.2.2: bytes=32 time=1ms TTL=125
Reply from 192.168.2.2: bytes=32 time=0ms TTL=125
Reply from 192.168.2.2: bytes=32 time=0ms TTL=125
Reply from 192.168.2.2: bytes=32 time=0ms TTL=125

Ping statistics for 192.168.2.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

PC2 > PC1

```
PC2>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time=0ms TTL=125
Reply from 192.168.1.2: bytes=32 time=0ms TTL=125
Reply from 192.168.1.2: bytes=32 time=0ms TTL=125
Reply from 192.168.1.2: bytes=32 time=0ms TTL=125

Ping statistics for 192.168.1.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Apabila reply maka konfigurasi pada routing EIGRP sudah berhasil kita lakukan, dan antar router sudah dapat bertukar data.

ROUTING DYNAMIC OSPF

Pada lab kali ini kita akan membahas routing protocol lain nya dalam routing dynamic yaitu OSPF (Open Shortest Path First), routing protocol OSPF ini termasuk bagian dari *link state* yang mana ia akan mengirim sebuah data atau packet melalui jalur yang bandwith terbesar atau nilai cost yang kecil, untuk jumlah administratif distance berjumlah 110.

OSPF ini sekarang merupakan protocol yang sudah banyak di pakai perusahan dalam routing untuk jaringan yang berskala besar di karna kan mudah nya mengkonfigurasikan nya dan juga yang bersifat open vendor/yang dapat di konfigurasi kan di setiap vendor.

Untuk perhitungan cost pada OSPF dapat di rumuskan dengan

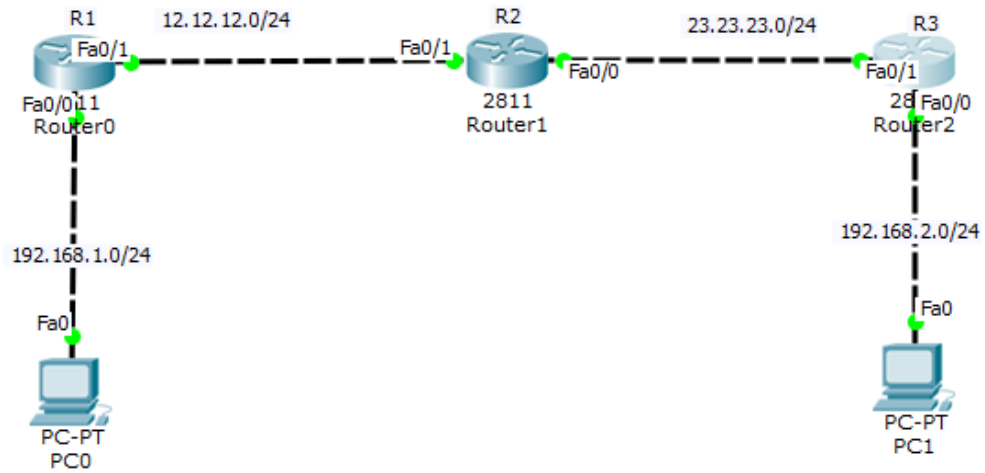
$$\frac{\text{Reference Bandwith}}{\text{Bandwith}}$$

Reference bandwith adalah ketetapan bandwith yaitu 100mb, yang kemudian dibagi bandwith yang sesuai dengan bandwith pada kabel yang di pakai pada router :

- Gigabyte Ethernet : 1000MB
- Fast Ethernet : 100MB
- Etherner : 10MB

Maka hasil dari pembagian tersebut merupakan cost nya dari suatu link OSPF tersebut

Untuk topologi pada lab kali ini kita akan menggunakan 3 router dan 2 client yang mana kita akan mengkonfigurasi kan nya dengan menggunakan routing OSPF



Seperti pada lab sebelum nya kita dapat mengkonfigurasi kan ip address terlebih dahulu pada setiap interface nya

```
R1(config)#int fa0/0
R1(config-if)#no sh
R1(config-if)#ip ad 192.168.1.1 255.255.255.0

R1(config-if)#int fa0/1
R1(config-if)#no sh
R1(config-if)#ip ad 12.12.12.1 255.255.255.0
```

```
R2(config)#int fa0/1
R2(config-if)#no sh
R2(config-if)#ip ad 12.12.12.2 255.255.255.0

R2(config-if)#int fa0/0
R2(config-if)#no sh
R2(config-if)#ip ad 23.23.23.2 255.255.255.0
```

```
R3(config)#int fa0/0
R3(config-if)#no sh
R3(config-if)#ip ad 192.168.2.1 255.255.255.0

R3(config-if)#int fa0/1
R3(config-if)#no sh
R3(config-if)#ip ad 23.23.23.3 255.255.255.0
```

Sebelum kita mulai merouting setiap router maka coba kita lihat terlebih dahulu tabel routing dari setiap router tersebut

R1

```
R1#sh ip route
Codes: C - connected, S - static, I - IGRP, 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

     12.0.0.0/24 is subnetted, 1 subnets
C       12.12.12.0 is directly connected, FastEthernet0/1
C    192.168.1.0/24 is directly connected, FastEthernet0/0
```

R2

```
R2#sh ip route
Codes: C - connected, S - static, I - IGRP, 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

     12.0.0.0/24 is subnetted, 1 subnets
C       12.12.12.0 is directly connected, FastEthernet0/1
     23.0.0.0/24 is subnetted, 1 subnets
C       23.23.23.0 is directly connected, FastEthernet0/0
```

R3

```
R3#sh ip route
Codes: C - connected, S - static, I - IGRP, 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

     23.0.0.0/24 is subnetted, 1 subnets
C       23.23.23.0 is directly connected, FastEthernet0/1
C    192.168.2.0/24 is directly connected, FastEthernet0/0
```

Di saat sebelum routing maka di setiap router hanya memiliki network nya masing – masing saja atau network yang terhubung pada router tersebut.

Pada routing Dynamic OSPF kita akan memasukan network yang terhubung pada router tersebut atau yang terdapat pada tabel routing sebelum di routing, kemudian kita masukan wildcart mask nya

Cara mencari wildcart mask yaitu 255.255.255.255 di kurang subnet mask, karna pada lab kali ini kita mengunakan prefix 24 maka subnetmask nya yaitu 255.255.255.0 kemudian apabila dikurang dengan 255.255.255.255 maka hasil nya adalah 0.0.0.255 maka itu adalah wildcart mask yang akan kita gunakan.

Dan untuk lab OSPF pada kali ini kita hanya akan mengunakan 1 area yaitu area backbone atau *area0*.

R1

```
R1(config)#router ospf 1
R1(config-router)#network 12.12.12.0 0.0.0.255 area 0
R1(config-router)#network 192.168.1.0 0.0.0.255 area 0
```

R2

```
R2(config)#router ospf 1
R2(config-router)#network 12.12.12.0 0.0.0.255 area 0
R2(config-router)#network 23.23.23.0 0.0.0.255 area 0
```

R3

```
R3(config)#router ospf 1
R3(config-router)#network 23.23.23.0 0.0.0.255 area 0
R3(config-router)#network 192.168.2.0 0.0.0.255 area 0
```

Maka dengan ini maka konfigurasi routing OSPF sudah selsai maka coba kita lihat kembali tabel routing pada setiap router apakah sudah terdaftar network router yang lain, dan pastikan status nya yaitu "O" yang berarti OSPF.

R1

```
R1#sh ip route
Codes: C - connected, S - static, I - IGRP, 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

     12.0.0.0/24 is subnetted, 1 subnets
C       12.12.12.0 is directly connected, FastEthernet0/1
     23.0.0.0/24 is subnetted, 1 subnets
O       23.23.23.0 [110/2] via 12.12.12.2, 00:16:27, FastEthernet0/1
C    192.168.1.0/24 is directly connected, FastEthernet0/0
O    192.168.2.0/24 [110/3] via 12.12.12.2, 00:16:17, FastEthernet0/1
```

R2

```
R2#sh ip route
Codes: C - connected, S - static, I - IGRP, 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

     12.0.0.0/24 is subnetted, 1 subnets
C       12.12.12.0 is directly connected, FastEthernet0/1
     23.0.0.0/24 is subnetted, 1 subnets
C       23.23.23.0 is directly connected, FastEthernet0/0
O    192.168.1.0/24 [110/2] via 12.12.12.1, 00:18:53, FastEthernet0/1
O    192.168.2.0/24 [110/2] via 23.23.23.3, 00:17:48, FastEthernet0/0
```

R3

```
R3#sh ip route
Codes: C - connected, S - static, I - IGRP, 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

     12.0.0.0/24 is subnetted, 1 subnets
O       12.12.12.0 [110/2] via 23.23.23.2, 00:20:22, FastEthernet0/1
     23.0.0.0/24 is subnetted, 1 subnets
C       23.23.23.0 is directly connected, FastEthernet0/1
O    192.168.1.0/24 [110/3] via 23.23.23.2, 00:20:22, FastEthernet0/1
C    192.168.2.0/24 is directly connected, FastEthernet0/0
```

Untuk verivikasi kita dapat lakukan test ping antar PC1 dan PC2

PC1 > PC2

```
PC>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Reply from 192.168.2.2: bytes=32 time=0ms TTL=125
Reply from 192.168.2.2: bytes=32 time=0ms TTL=125
Reply from 192.168.2.2: bytes=32 time=1ms TTL=125
Reply from 192.168.2.2: bytes=32 time=0ms TTL=125

Ping statistics for 192.168.2.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

PC2 > PC1

```
PC>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time=1ms TTL=125
Reply from 192.168.1.2: bytes=32 time=0ms TTL=125
Reply from 192.168.1.2: bytes=32 time=1ms TTL=125
Reply from 192.168.1.2: bytes=32 time=0ms TTL=125

Ping statistics for 192.168.1.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

maka dengan ini kita telah selesai mengkonfigurasi kan routing OSPF.

ACCESS – LIST (ACL)

Pada lab kali ini kita akan membahas tentang yang nama nya access – list, yang mana fungsi dari access – list bisa di katakan sebagai filtering sebuah packet yang melewati router, jadi router akan memfilter packet – packet mana saja yang di boleh kan atau di larang melewati router .

Dalam access – list sendiri terbagi menjadi 2 jenis yaitu :

- ***Standard Access – list*** : pada standard access – list ini ACL akan di konfigurasikan pada router yang terdekat dengan destination, pada ACL standard ini kita hanya dapat memblock sebuah network, subnet, dan host untuk action dalam memfilter nya hanya terdapat deny (dilarang) dan permit (dibolehkan)
- ***Extended Access – list*** : hampir sama dengan standard ACL, yang membedakan pada ACL extended ini memiliki fitur yang lebih banyak dalam memfilter untuk konfigurasi ACL dengan menggunakan extended tidak harus yang terdekat dengan destination, karna pada extended ACL ia dapat memfilter source ataupun destination dan port ataupun protocol – protocol

Standard ACL	Extended ACL
ACL Number range 1-99	ACL Number range 100-199
Can block a network, host and subnet	Can allow or deny a network, host, subnet and service
All service are blocked	Select service can be blocked
Implemented closest to the destination	Implemented closest to the destination
Filtering based on source IP address only	Filtering based on source IP address, destination IP, protocol and port number

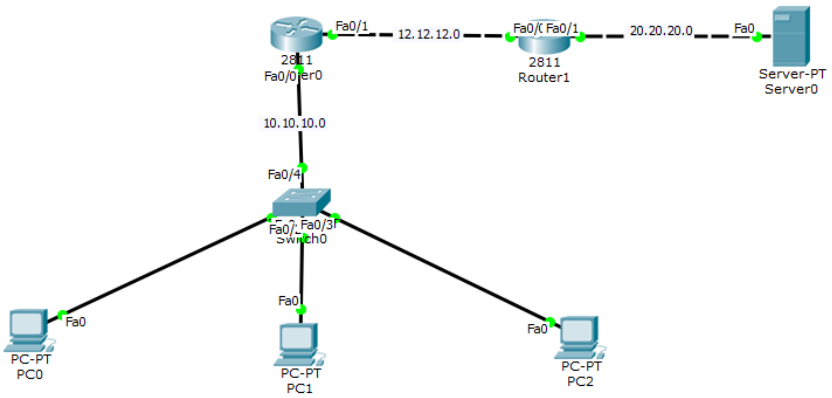
Untuk default dari ACL sendiri yaitu *deny*(dilarang/menolak) jadi di saat kita mengkonfigurasi kan ACL maka otomatis semua trafick yang lewat akan terblock atau deny, yang pada dasar nya kita hanya mengkonfigurasi kan hanya beberapa host saja yang di block, tetapi semua nya akan ikut terblock, itu di karna kan default dari ACL deny, maka dari itu kita harus mengkonfigurasi kan *permit any* (bolehkan semua) pada rule terkahir .

STANDARD ACCESS – LIST

Standard ACL :

- Standard ACL hanya dapat melakukan filtering berdasarkan IP host atau IP network source nya saja.
- Standard ACL mengunnakan ACL number 1 – 99
- Konfigurasi sedekat mungkin dengan destination
- Direction in dan out nya ditentukan berdasarkan arah packet nya dari source menuju destination

pada lab kali ini kita akan mengkonfigurasi kan dengan menggunkan 2 router,1switch, 1server, 3client



Tujuan pada lab ini kita akan mendeny(memblock) network 10.10.10.0 agar tidak dapat terhubung dengan network server yaitu 20.20.20.0 tetapi network 10.10.10.0 masih dapat terhubung ke network 12.12.12.0 maka kita akan memfilter nya dengan menggunakan ACL stadard .

Sebelum kita mengkonfigurasi kan ACL maka kita dapat mengkonfigurasi kan IP address pada router sesuai topologi dan konfigurasi trunk pada switch yang mengarah ke router.

SW1

```
SW1(config)#int fa0/4
SW1(config-if)#switchport mode trunk
```

R1

```
R1(config)#int fa0/0
R1(config-if)#no sh
R1(config-if)#ip ad 10.10.10.1 255.255.255.0
R1(config-if)#int fa0/1
R1(config-if)#no sh
R1(config-if)#ip ad 12.12.12.1 255.255.255.0
```

R2

```
R2(config)#int fa0/0
R2(config-if)#no sh
R2(config-if)#ip ad 12.12.12.2 255.255.255.0
R2(config-if)#int fa0/1
R2(config-if)#no sh
R2(config-if)#ip ad 20.20.20.2 255.255.255.0
```

Karna antar router kita memiliki network yang berbeda – beda maka dengan itu kita harus mengkonfigurasi kan juga routing pada setiap router, agar lebih mudah kita dapat membuat routing dynamic EIGRP

R1

```
R1(config)#router eigrp 10
R1(config-router)#network 10.10.10.0
R1(config-router)#network 12.12.12.0
```

R2

```
R2(config)#router eigrp 10
R2(config-router)#network 12.12.12.0
R2(config-router)#network 20.20.20.0
```

Setelah setiap perangkat sudah di konfigurasikan IP masing – masing dan sudah di routing maka kita dapat langsung mengkonfigurasikan ACL pada router terdekat dengan destination.

Sebelum nya pastikan bahwa client – client di switch sudah dapat PING ke server

Karna pada lab kali ini destination kita adalah network 20.20.20.0 maka kita dapat mengkonfigurasikan nya di R2 dan memberi nya di interafce yang mengarah ke server yaitu out apabila source nya dari network 10.10.10.0

Konfigurasikan ACL pada R2 :

```
R2(config)#access-list 1 deny 10.10.10.0 0.0.0.255
R2(config)#access-list 1 permit any
```

Setelah kita membuat ACL maka ACL tersebut kita harus masukan atau tanamkan pada interface nya yang terdekat pada destination di R2, di interface fa0/1 yang mana apabila kita analisa apabila source address yang berasal dari network 10.10.10.0 maka interface nya kita konfigurasikan dengan out

```
R2(config)#int fa0/1
R2(config-if)#ip access-group 1 out
```

untuk verivikasi kita dapat melakukan ping dari PC1 menuju server

Test PING dari PC1 menuju ke server

```
PC1>ping 20.20.20.1

Pinging 20.20.20.1 with 32 bytes of data:

Reply from 12.12.12.2: Destination host unreachable.
Reply from 12.12.12.2: Destination host unreachable.
Reply from 12.12.12.2: Destination host unreachable.
Reply from 12.12.12.2: Destination host unreachable.

Ping statistics for 20.20.20.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

PC2 menuju ke server

```
PC2>ping 20.20.20.1

Pinging 20.20.20.1 with 32 bytes of data:

Reply from 12.12.12.2: Destination host unreachable.
Reply from 12.12.12.2: Destination host unreachable.
Reply from 12.12.12.2: Destination host unreachable.
Reply from 12.12.12.2: Destination host unreachable.

Ping statistics for 20.20.20.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

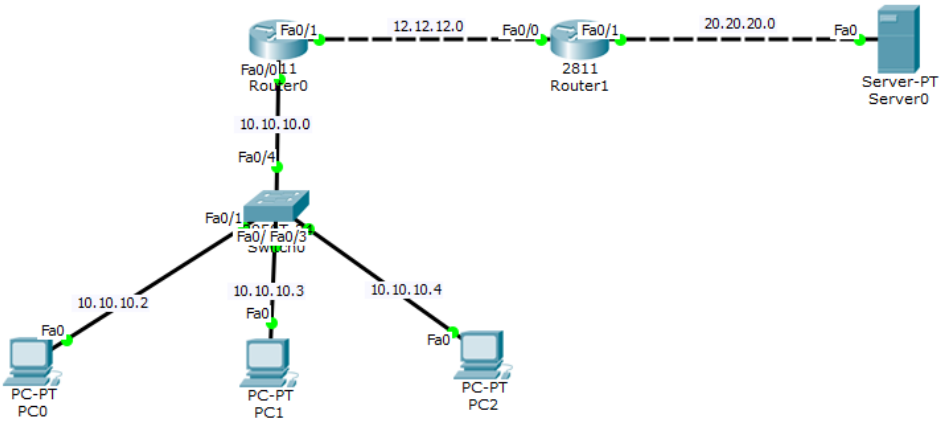
PC3 menuju server

```
PC3>ping 20.20.20.1

Pinging 20.20.20.1 with 32 bytes of data:

Reply from 12.12.12.2: Destination host unreachable.
Reply from 12.12.12.2: Destination host unreachable.
Reply from 12.12.12.2: Destination host unreachable.
Reply from 12.12.12.2: Destination host unreachable.

Ping statistics for 20.20.20.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Maka client dari network 10.10.10.0 akan terdeny apabila menuju network 20.20.20.0 yang mengarah kan server

ACCESS – LIST STANDARD SKENARIO2

Pada lab kali ACL standard skenario2 kita akan memblock source yang berasal dari salah satu host / satu client saja, kalau pada lab ACL standard sebelum nya kita memblock seluruh host dengan memblock network nya, maka pada lab kali ini kita hanya memblock salah satu saja yang tidak dapat terhubung sedangkan yang lain nya masih dapat terhubung.



Tujuan pada lab kali ini yaitu agar host yang ber-IP 10.10.10.1 tidak dapat mengakses network 20.20.20.0 tetapi host yang lain nya atau yang berIP 10.10.10.2 dan 10.10.10.3 masih dapat mengakses network 20.20.20.0

Lalu bagaimana kah cara nya ..??, maka kalau pada lab sebelum nya kita mendeny sebuah network maka pada lab kali ini kita akan mendeny host nya kemudian kita masukan IP dari host yang ingin kita deny tadi.

Sebelum nya kita konfigurasikan terlebih dahulu IP pada setiap perangkat sesuai dengan yang di topologi. Seperti pada router, PC, server, dan jangan lupa allowed trunk pada switch.

R1

```
R1(config-if)#no sh
R1(config-if)#ip ad 10.10.10.1 255.255.255.0
R1(config-if)#int fa0/1
R1(config-if)#no sh
R1(config-if)#ip ad 12.12.12.1 255.255.255.0
```

R2

```
R2(config)#int fa0/0
R2(config-if)#no sh
R2(config-if)#ip ad 12.12.12.2 255.255.255.0
R2(config-if)#int fa0/1
R2(config-if)#no sh
R2(config-if)#ip ad 20.20.20.2 255.255.255.0
```

SW1

```
SW1(config)#int fa0/4
SW1(config-if)#switchport mode trunk
```

Setelah kita mengkonfigurasikan IP pada setiap perangkat nya tentu nya network pada setiap router nya pun berbeda maka di perlukan lah routing agar host di SW1 dapat terhubung ke server yang berbeda router, agar lebih mudah kita dapat merouting antar router ini dengan routing dynamic OSPF.

R1

```
R1(config)#router ospf 1
R1(config-router)#network 10.10.10.0 0.0.0.255 area 0
R1(config-router)#network 12.12.12.0 0.0.0.255 area 0
```

R2

```
R2(config)#router ospf 1
R2(config-router)#network 12.12.12.0 0.0.0.255 area 0
R2(config-router)#network 20.20.20.0 0.0.0.255 area 0
```

Setelah sudah maka pastikan setiap client dapat ping ke server, maka setelah itu kita dapat langsung mengkonfigurasi kan ACL pada router yang paling terdekat pada destination yaitu R2

IP host yang akan kita deny adalah IP dari PC1 yaitu 10.10.10.2 dan untuk PC yang lain nya tetap dapat mengakses network 20.20.20.0

Karna default dari access – list ini adalah deny maka apabila kita tidak membuat konfigurasi ACL untuk mempermit yang lain nya maka selain dari PC1 maka ia pun akan ikut terdeny, maka kita harus juga membuat konfigurasi ACL untuk mempermit

```
R2(config)#access-list 1 deny host 10.10.10.2
R2(config)#access-list 1 permit any
R2(config)#int fa0/1
R2(config-if)#ip access-group 1 out
```

Setelah sudah maka kita dapat melihat tabel ACL nya :

```
R2(config)#do sh access-list
Standard IP access list 1
10 deny host 10.10.10.2
20 permit any
```

Dapat di lihat rule untuk ACL pun perhatikan bahwa ACL akan membaca rule dari atas ke bawah jadi apabila kita membuat konfigurasi untuk mempermit terlebih dahulu maka konfiguraasi deny nya pun tidak akan di baca oleh ACL karna sudah tertiban oleh konfigurasi permit yang tadi.

Maka setelah itu kita dapat mencoba test ping dari PC1 dengan IP 10.10.10.2 menuju network 20.20.20.0

```
PC1>ping 20.20.20.1

Pinging 20.20.20.1 with 32 bytes of data:

Reply from 12.12.12.2: Destination host unreachable.
Reply from 12.12.12.2: Destination host unreachable.
Reply from 12.12.12.2: Destination host unreachable.
Reply from 12.12.12.2: Destination host unreachable.

Ping statistics for 20.20.20.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Kemudian kita dapat mencoba test ping dari PC2 dan PC3 menuju network 20.20.20.0

PC2

```
PC2>ping 20.20.20.1

Pinging 20.20.20.1 with 32 bytes of data:

Reply from 20.20.20.1: bytes=32 time=15ms TTL=126
Reply from 20.20.20.1: bytes=32 time=0ms TTL=126
Reply from 20.20.20.1: bytes=32 time=1ms TTL=126
Reply from 20.20.20.1: bytes=32 time=0ms TTL=126

Ping statistics for 20.20.20.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 15ms, Average = 4ms
```

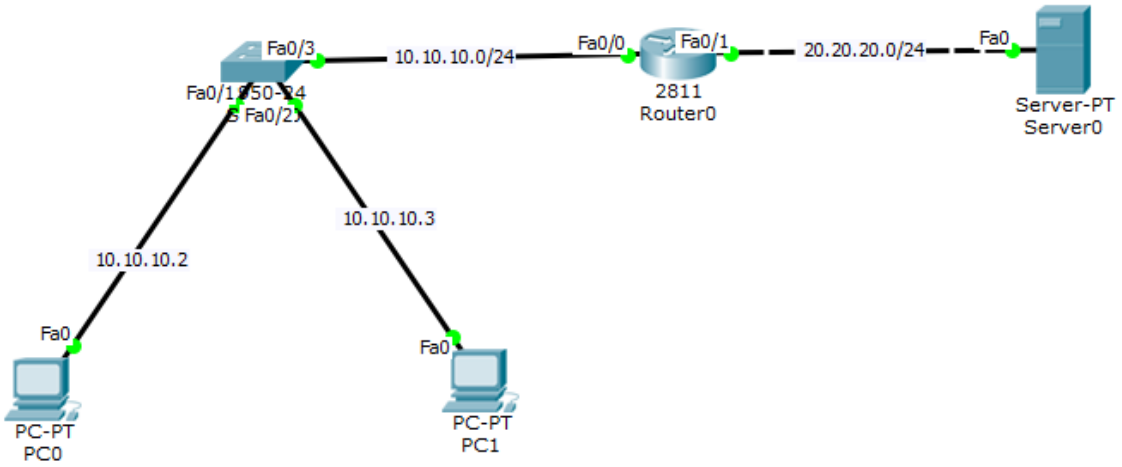
PC3

```
PC3>ping 20.20.20.1

Pinging 20.20.20.1 with 32 bytes of data:

Reply from 20.20.20.1: bytes=32 time=1ms TTL=126
Reply from 20.20.20.1: bytes=32 time=0ms TTL=126
Reply from 20.20.20.1: bytes=32 time=0ms TTL=126
Reply from 20.20.20.1: bytes=32 time=0ms TTL=126

Ping statistics for 20.20.20.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Maka PC 2 dan 3 pun akan reply karna kita hanya membuat ACL untuk PC1 saja yang akan di deny dan PC 2 dan 3 kita permit untuk menuju destination network 20.20.20.0

EXTENDED ACCESS – LIST

Pada lab kali ini kita akan membahas salah satu jenis access –list yaitu extended access – list yang mana salah satu perbedaan extended acl ini memiliki fiutr yang lebih mendalam dari pada standard access – list, apabila dalam standard acl kita hanya dapat memfilter yang berasal dari source saja, maka pada extended acl ini dapat memfilter seperti destination, protocol, port dan lain sebagainya atau dapat di katakan extended acl ini merupakan acl yang lebih spesifik dari pada standard acl .



Pada topologi kali ini kita akan memfilter salah sartu client dengan IP 10.10.10.2, agar tidak dapat melakukan ping ke network server, dan untuk client ip 10.10.10.3 kita akan memfilter agar tidak dapat mengakses http .

Jadi kita akan memfilter menggunakan ACL extended untuk source address 10.10.10.2 tidak dapat melakukan ping maka kita konfigurasikan *deny protocol ICMP* kemudian untuk client 10.10.10.3 kita akan konfigurasikan *deny http / tcp 80*

Sebelum kita mengonfigurasi kan kita dapat mengkonfigurasi kan ip address terlebih dahulu pada setiap interface pada masing – masing device sesuai pada topologi.

Router

```
Router(config)#int fa0/0
Router(config-if)#no sh
Router(config-if)#ip add 10.10.10.1 255.255.255.0
Router(config-if)#int fa0/1
Router(config-if)#no sh
Router(config-if)#ip add 20.20.20.1 255.255.255.0
Router(config-if)#exit
```

Setelah mengkonfigurasikan ip address kita dapat memulai dengan mengkonfigurasikan ACL extended untuk client dengan ip 10.10.10.2 deny icmp dan client 10.10.10.3 deny tcp 80

```
Router(config)#access-list 100 deny icmp host 10.10.10.2
20.20.20.2 0.0.0.255
Router(config)#access-list 100 deny tcp host 10.10.10.3
20.20.20.2 0.0.0.255 eq 80
Router(config)#access-list 100 permit ip any any
```

100 = pengklasifikasian access – list extended(100–199) atau standard(1-99)
Deny = action dari ACL permit(izinkan) ataukah deny(dilarang)
Icmp = protocol yang akan di filter
Host = jenis yang akan kita filter perhost atau network dan lain sebagainya
10.10.10.2 = source address (sumber ip address)
20.20.20.2 = destination (tujuan ip address)
0.0.0.0255 = wilcard mask
Eq = untuk penghususan port dalam tcp atau udp
Permit ip any any = di karna kan default dari ACL deny maka agar apabila terdapat client lain yang terhubng tidak ikut terfilter atau deny maka kita harus menambahkan permit ip any any

Setelah kita mengkonfigurasi kan ACL kita dapat tanamkan konfigurasi ACL ke dalam interface

```
Router(config)#int fa0/1
Router(config-if)#ip access-group 100 out
```

Untuk verifvikasi kita dapat melakukan test ping dari client 10.10.10.2 kemudian test akses web

Client 10.10.10.2

```
PC>ping 20.20.20.2

Pinging 20.20.20.2 with 32 bytes of data:

Reply from 10.10.10.1: Destination host unreachable.
Reply from 10.10.10.1: Destination host unreachable.
Reply from 10.10.10.1: Destination host unreachable.
Reply from 10.10.10.1: Destination host unreachable.

Ping statistics for 20.20.20.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

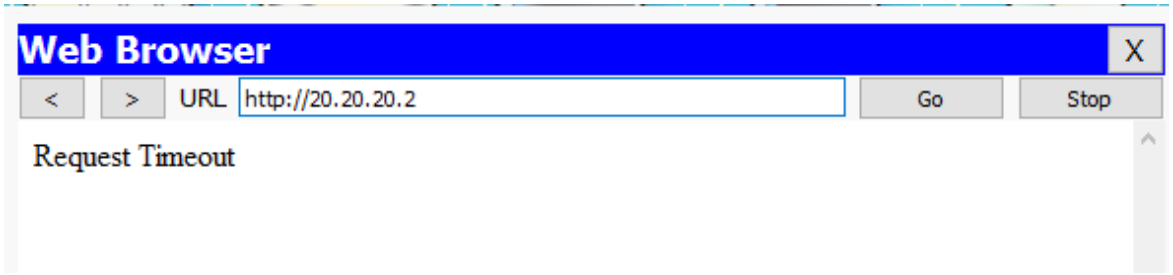
Akses web (HTTP)

Maka client 10.10.10.2 tidak dapat melakukan ping tetapi tetap dapat melakukan akses http, di karna kan kita tadi mendeny protocol ICMP/ping

Kemudian kita dapat lakukan ping dari client 10.10.10.3 dan lakukan akses HTTP

```
PC>ping 20.20.20.2

Pinging 20.20.20.2 with 32 bytes of data:

Reply from 20.20.20.2: bytes=32 time=0ms TTL=127
Reply from 20.20.20.2: bytes=32 time=0ms TTL=127
Reply from 20.20.20.2: bytes=32 time=0ms TTL=127
Reply from 20.20.20.2: bytes=32 time=2ms TTL=127

Ping statistics for 20.20.20.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 2ms, Average = 0ms
```

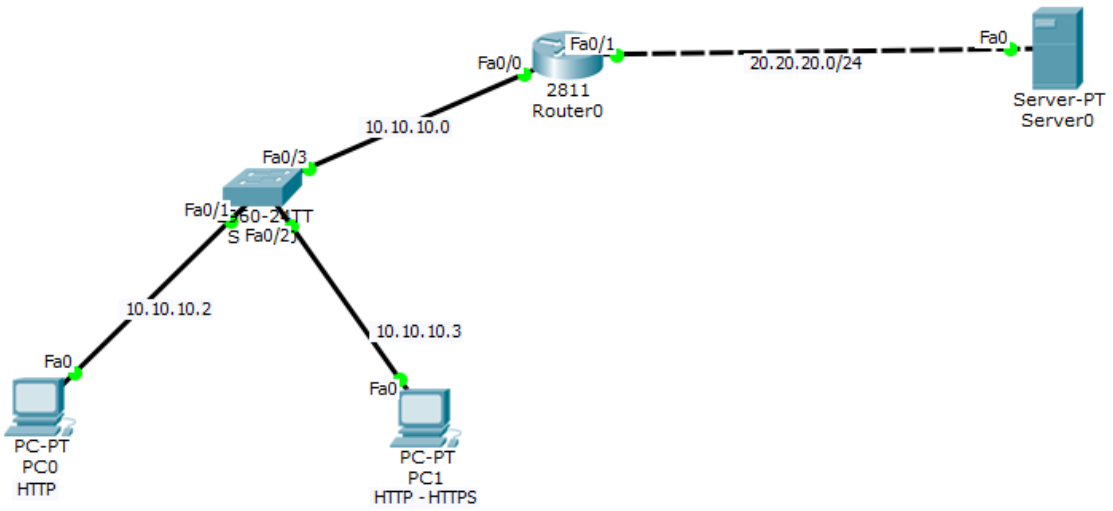
Kemudian akses HTTP



EXTENDED ACCESS – LIST SCENARIO2

Pada lab kali ini kita akan mengkonfigurasikan dengan manggunkan range yang mana pada lab kali ini kita akan mendeny 2 protocol sekaligus yaitu http dan https

Untuk topologi pada lab kali ini kita akan menggunakan 2 client, 1router, dan 1server



Kita akan mendeny HTTP pada client1 10.10.10.2 dan mendeny HTTP dan HTTPS pada ip 10.10.10.3 menggunakan range pada ACL

Sebelum mengkonfigurasikan ACL kita dapat mengkonfigurasikan ip addres terelebih dahulu pada router

```
Router(config)#int fa0/0
Router(config-if)#no sh
Router(config-if)#ip add 10.10.10.1 255.255.255.0
Router(config-if)#int fa0/1
Router(config-if)#no sh
Router(config-if)#ip add 20.20.20.1 255.255.255.0
Router(config-if)#exit
```

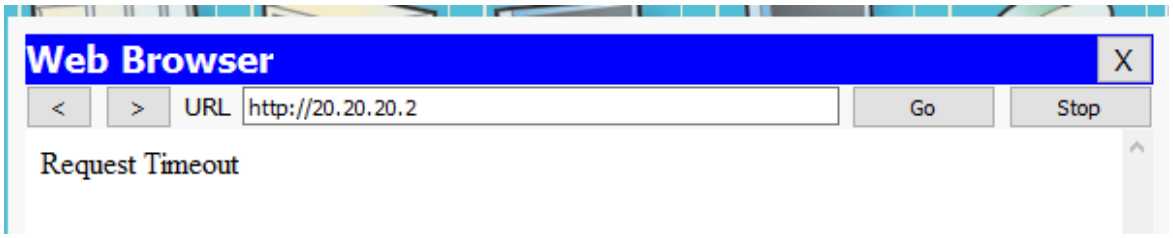
Kemudian kita dapat lanjut dengan mengkonfigurasikan ACL extended dengan memulai dengan mengkonfigurasiakn rule untuk memfilter client 10.10.10.2 dengan deny HTTP dan dilanjutkan 10.10.10.3 deny HTTP dan HTTP dengan menggunakan range

```
Router(config)#access-list 100 deny tcp host 10.10.10.2
20.20.20.2 0.0.0.255 eq 80
Router(config)#access-list 100 deny tcp host 10.10.10.3
20.20.20.2 0.0.0.255 range 80 443
Router(config)#access-list 100 permit ip any any
```

Setelah itu kita dapat konfigurasi kan ACL yang telah kita buat tadi kedalam interface router apakah itu in ataukah out.

```
Router(config)#int fa0/0
Router(config-if)#ip access-group 100 out
```

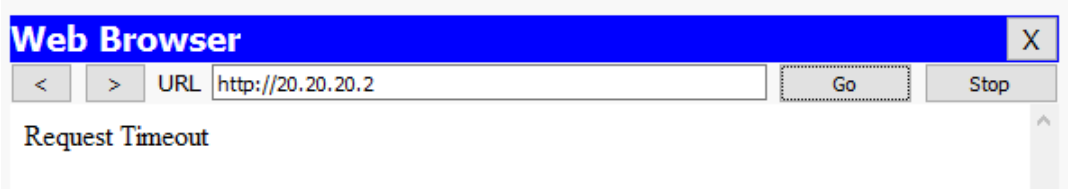
Untuk verivikasi kita dapat lakukan akses HTTP pada client1



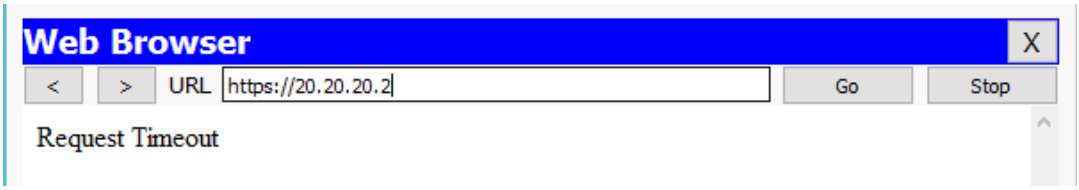
Maka ia tidak bisa, lalu bagaimana dengan HTTPS

Untuk HTTPS maka ia masih dapat di akses di karanakan konfigurasi ACL kita tadi hanya mendeny protocol port HTTP tidak dengan HTTPS

Lalu bagaimana dengan client2 apakah dapat mengakses HTTP dan HTTPS



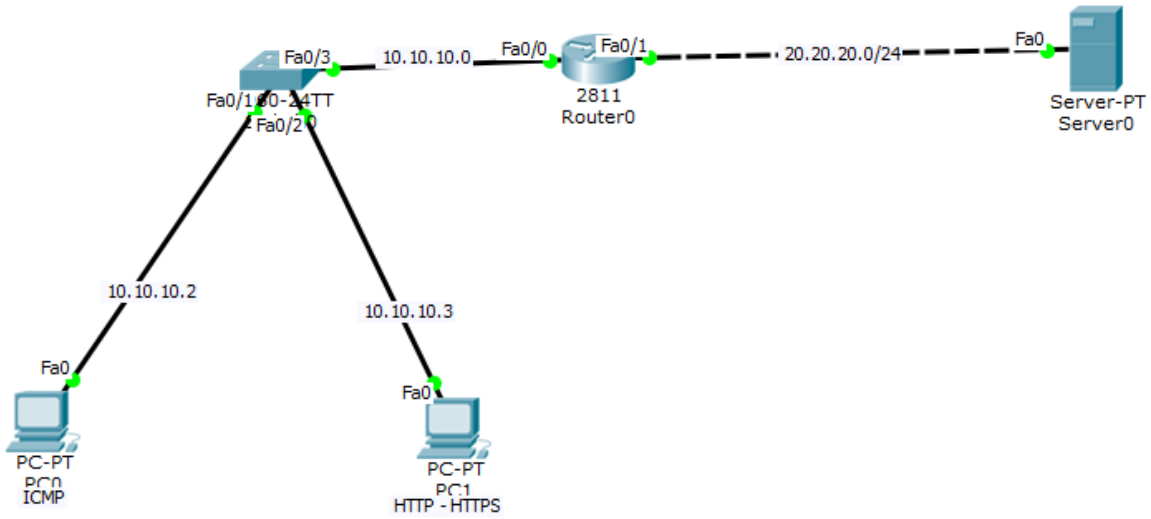
Lalu HTTPS



NAMED ACCESS – LIST

Pada lab kali ini kita akan mengkonfigurasikan ACL dengan menggunakan name, yang mana pada ACL ini kita dapat menetukan sequenece number atau dalam kata lain kita dapat mencustome urutan rule kita sendiri .

Apabila pada lab sebelum nya kita "*sh ip access-list*" maka ia akan terlihat rule pada ACL, yang mana ACL ia akan membaca rule dari atas ke bawah, maka apabila terdapat salah satu konfigurasi yang kurang maka kita tidak dapat menghapus salah satu konfigurasi dalam ACL tersebut, kecuali kita menghapus ACL tersebut dan mengkonfigurasi nya dari awal kembali.



Kita akan mengkonfigurasikan extended ACL dengan menggunakan ACL name, dengan mendeny client1 dengan IP 10.10.10.2 deny ICMP dan client2 deny 10.10.10.3 deny HTTP dan HTTPs tetapi kita tidak menggunakan range tetapi dengan menambahkan rule

Sebelumnya kita dapat mengkonfigurasi kan pada interface router

```
Router(config)#int fa0/0
Router(config-if)#no sh
Router(config-if)#ip add 10.10.10.1 255.255.255.0
Router(config-if)#int fa0/1
Router(config-if)#no sh
Router(config-if)#ip add 20.20.20.1 255.255.255.0
Router(config-if)#exit
```

Kita akan mendeny ICMP untuk client1 dan HTTP untuk client2 kemudian permit any setelah itu kita dapat menyisipkan rule untu k deny HTTPS pada client2

```
Router(config)#ip access-list extended mq
Router(config-ext-nacl)#10 deny icmp host 10.10.10.2
20.20.20.2 0.0.0.255
Router(config-ext-nacl)#15 deny tcp host 10.10.10.3
20.20.20.2 0.0.0.255 eq 80
Router(config-ext-nacl)#20 permit ip any any
```

Extended : ACL yang akan di gunakan standard atau extended
Mq : nama ACL yang kita gunakan (bebas)
10 : sequence number (urutan rule)
Deny : action ACL (deny/permit)
ICMP : protocol yang ingin di filter (tcp/udp, ICMP dll)
Host : jenis filter host, network dan lain sebagai nya
10.10.10.2 : source address
20.20.20.2 : destination address
0.0.0.255 : wilcard mask
Eq 80 : identifikasi number port

maka dengan ini kita telah mengkonfigurasi kan deny ICMP(ping),HTTTP, dan permit any, tetapi kita belum mengkonfigurasikan deny untuk HTTPS.

pada dasar nya apabila kita menggunakan ACL dengan konfigurasi biasa seperti pada lab sebelum nya kita tidak dapat menyisipkan rule di antara rule – rule yang lain dan apabila kita menambahkan rule dengan ACL maka rule tersebut sudah tidak terbaca di karna kan ACL membaca dari atas kebawah dan rule tersebut berada pada rule "*permit ip any any*" maka rule deny sudah tidak dapat terbaca kembali oleh ACL

sh ip access-list

```
Router#sh ip access-lists
Extended IP access list mq
10 deny icmp host 10.10.10.2 20.20.20.0 0.0.0.255
15 deny tcp host 10.10.10.3 20.20.20.0 0.0.0.255 eq www
25 permit ip any any
```

Kita akan menyisipkan rule di antara sequence 10 dan 15 untuk deny HTTPS

```
Router(config)#ip access-list extended mq
Router(config-ext-nacl)#13 deny tcp host 10.10.10.3
20.20.20.2 0.0.0.255 eq 443
```

Kemudian kita dapat lakukan "*sh ip access- list*"

```
Router#sh ip access-lists
Extended IP access list mq
10 deny icmp host 10.10.10.2 20.20.20.0 0.0.0.255
13 deny tcp host 10.10.10.3 20.20.20.0 0.0.0.255 eq 443
15 deny tcp host 10.10.10.3 20.20.20.0 0.0.0.255 eq www
25 permit ip any any
```

Maka kita telah menyisipkan salah satu rule kedalam ACL , yang mana merupakan salah satu kelebihan dalam penggunaan ACL name

Setelah itu jangan lupa ACL yang telah kita konfigurasikan di tanamkan dalam interface router

```
Router(config)#int fa0/1
Router(config-if)#ip access-group mq out
```

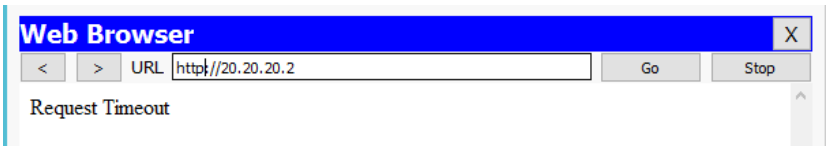
Kalau pada ACL biasa kita menggunakan number ACL nya, pada ACL name kita menggunakan nama dari ACL tersebut yang telah kita konfig

Untuk verivikasi dapat di lakukan dengan test ping dari client1 dan aksess HTTP dan HTTPS pada client2

Client1

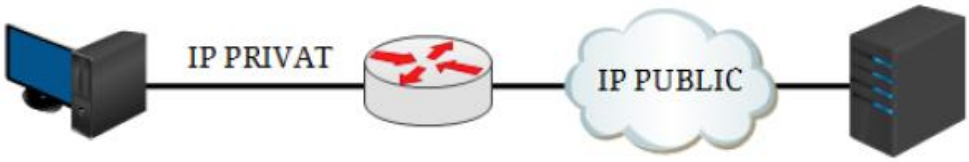
```
PC>ping 20.20.20.2

Pinging 20.20.20.2 with 32 bytes of data:

Reply from 10.10.10.1: Destination host unreachable.
Reply from 10.10.10.1: Destination host unreachable.
Reply from 10.10.10.1: Destination host unreachable.
Reply from 10.10.10.1: Destination host unreachable.

Ping statistics for 20.20.20.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Client2



NAT (NETWORK ADDRESS TRANSLATION)

- Berfungsi untuk menerjemahkan atau merubah IP seperti dari IP privat menjadi IP public
- Ip privat sendiri tidak dapat di gunakan dalam internet, maka dari itu kita harus menerjemahkan Ip privat tersebut ke dalam ip public dengan menggunakan NAT
- Dapat digunakan apabila terdapat suatu server local yang ingin di akses mengunakan internet maka digunakan IP public
- Dapat digunakan apabila ingin koneksi VPN menuju kantor mengunakan IP public



Dalam konfigurasi NAT interface dibagi mejadi 2 kategori :

- Inside : traffic yang masuk ke interface yang berasal dari local network
- Outside : traffic yang keluar dari interface router yang menuju ke destination (internet).

Nat pada cisco terbagi dari beberapa tipe :

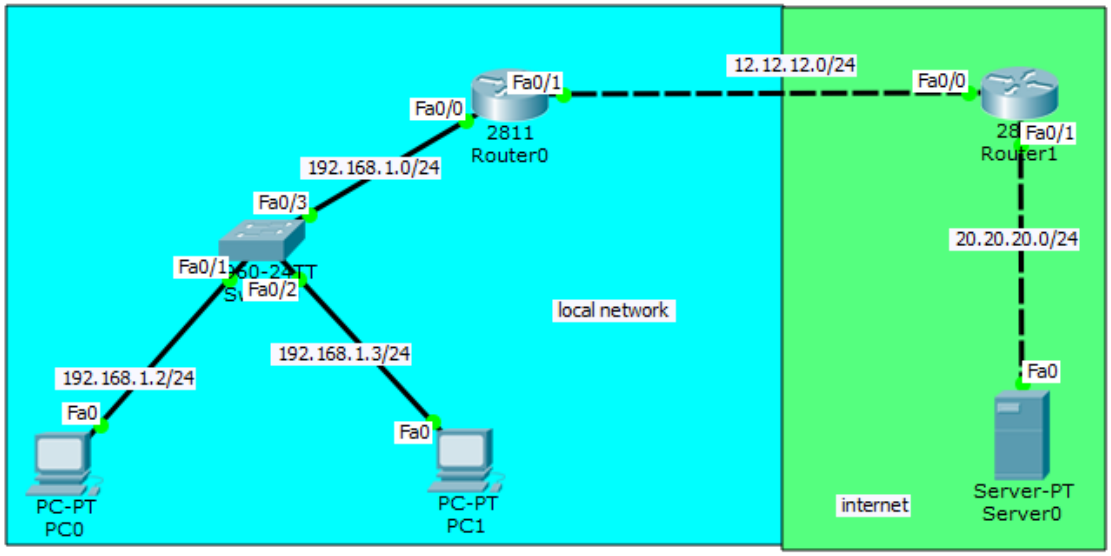
- Static NAT : satu IP privat yang ditranslasikan ke satu IP public (one to one mapping)
- Dynamic NAT : jumlah IP public yang di sediakan harus sejumlah ip privat yang ditranslasikan, dan NAT jenis ini jarang di gunakan
- Overloading/port address translation (PAT) : Akses ineternet menggunakan 1 IP public , dan ini yang sedang banyak di gunakan pada saat ini

STATIC NAT

Pada static NAT kita akan mentranslasikan suatu ip privat ke ip public secara static, yang mana kita akan mengkonfigurasi kan satu ip privat yang kita ubah ke ip public dengan kita sendiri yang merubah nya sesuai ip yang kita inginkan secara manual, atau dapat di katakan static nat merupakan one to one mapping.

Kita akan mengkonfigurasikan ACL terlebih dahulu untuk menandai suatu ip privat yang akan kita translasikan menjadi Ip public

Untuk topologi pada lab kali ini kita akan menggunakan 2 router, 1 switch, 1 server, dan 2 host yang mana kita akan membuat router dan server merupakan seolah – olah internet, dan salah satu PC agar dapat terhubung ke internet kita akan mengkonfigurasi kan NAT tidak dengan routing agar Ip privat pada PC tersbeut di ganti menjadi ip public.



Kita akan mentranslasikan IP 192.168.1.2 yang berada pada host di SW1.

Sebelum kita mengkonfigurasikan NAT maka terlebih dahulu konfigurasikan IP pada setiap perangkat seperti pada topologi dan trunk pada switch.

R1

```
R1(config)#int fa0/0
R1(config-if)#no sh
R1(config-if)#ip ad 192.168.1.1 255.255.255.0
R1(config-if)#int fa0/1
R1(config-if)#no sh
R1(config-if)#ip ad 12.12.12.1 255.255.255.0
```

R2

```
R2(config)#int fa0/0
R2(config-if)#no sh
R2(config-if)#ip ad 12.12.12.2 255.255.255.0
R2(config-if)#int fa0/1
R2(config-if)#no sh
R2(config-if)#ip ad 20.20.20.1 255.255.255.0
```

SW1

```
Switch(config)#int fa0/3
Switch(config-if)#switchport mode trunk
```

Setelah setiap device sudah di beri Ip dan switch sudah di trunk, kita dapat mengkonfigurasikan default route pada R1

```
R1(config)#ip route 0.0.0.0 0.0.0.0 12.12.12.2
```

Konfigurasikan NAT static

```
R1(config)#ip nat inside source static 192.168.1.2 12.12.12.12
R1(config)#int fa0/0
R1(config-if)#ip nat inside
R1(config-if)#int fa0/1
R1(config-if)#ip nat outside
```

Setelah sudah kita konfigurasikan NAT static pada host yang ber IP 192.168.1.2 maka kita dapat mencoba ping dari host/PC tersebut menuju server/internet.

```
PC1>ping 20.20.20.2

Pinging 20.20.20.2 with 32 bytes of data:

Reply from 20.20.20.2: bytes=32 time=0ms TTL=126
Reply from 20.20.20.2: bytes=32 time=0ms TTL=126
Reply from 20.20.20.2: bytes=32 time=0ms TTL=126
Reply from 20.20.20.2: bytes=32 time=1ms TTL=126

Ping statistics for 20.20.20.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Maka hasil nya pun ia akan reply, lalu bagaimanakah dengan host yang ber IP 192.168.1.3 ?

```
PC2>ping 20.20.20.2

Pinging 20.20.20.2 with 32 bytes of data:

Request timed out.
Request timed out.
```

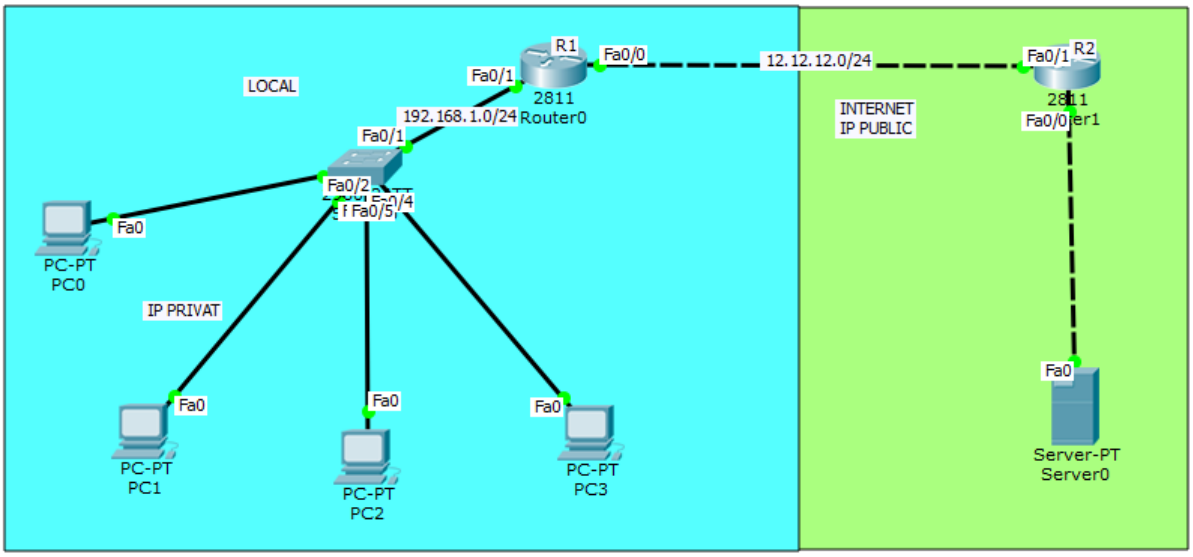
```
Request timed out.
Request timed out.

Ping statistics for 20.20.20.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Maka ia pun akan time out, karna yang tadi kita konfigurasikan hanya lah NAT untuk host yang ber IP 192.168.1.2, maka host yang ber IP 192.168.1.3 masih tidak bisa mengakses internet atau PING server, apabila kita ingin host tersebut dapat mengakses juga maka kita harus mengkonfgiurasikan NAT juag seperti tadi dengan IP 192.168.1.3 atau IP host teresebut .

DYNAMIC NAT OVERLOAD

Apabila pada lab sebelum nya kita telah mengkonfigurasi kan NAT secara static yang mana apabila ada beberapa PC yang ingin internetan maka kita akan mengkonfigurasi kan nat nya satu per satu di setiap PC tersebut, maka apabila sangat banyak client yang ingin yang internet kita menjadi lebih tidak efisien dalam mengkonfigurasikan nya maka pada lab kali ini di dynamic nat overload kita akan mengkonfigurasi kan nat dengan cara lebih mudah dengan menggunakan dynamic Nat overload yang dalam istilah lain PAT (Port Address Translation)

Untuk topologi pada lab kali ini tidak jauh dari lab sebelum nya, yang memebedakan kita akan menggunakan client yang lebih banyak dari sebelum nya



Konfigurasikan ip address pada tiap – tiap perangkat sesuai topologi

R1 :

```
R1(config)#int fa0/0
R1(config-if)#no sh
R1(config-if)#ip add 12.12.12.1 255.255.255.0
R1(config-if)#int fa0/1
R1(config-if)#no sh
R1(config-if)#ip add 192.168.1.1 255.255.255.0
```

R2 :

```
R2(config)#int fa0/1
R2(config-if)#no sh
R2(config-if)#ip add 12.12.12.2 255.255.255.0
R2(config-if)#int fa0/0
R2(config-if)#no sh
R2(config-if)#ip add 20.20.20.2 255.255.255.0
```

Trunk pada switch :

```
Switch(config)#int fa0/1
Switch(config-if)#switchport mode trunk
```

Dalam konfigurasi NAT dynamic kita dapat mengkonfigurasi kan ACL terlebih dahulu untuk source pada NAT nanti nya dengan ACL *permity any*

```
R1(config)#access-list 1 permit any
```

Setelah itu kita dapat mengkonfigurasi kan NAT overload, dengan memasukan ACL yang telah kita buat tadi pada source list

```
R1(config)#ip nat inside source list 1 interface
fastEthernet 0/0 overload
```

Setelah itu kita dapat mengkonfigurasi kan untuk NAT inside dan outside pada tiap interface nya

```
R1(config)#int fa0/0
R1(config-if)#ip nat outside
R1(config-if)#int fa0/1
R1(config-if)#ip nat inside
```

Sampai di sini kita telah selesai mengkonfigurasi kan NAT, hanya tinggal default route menuju internet dengan menggunakan ip route dengan gateway ip yang mengarah ke internet

```
R1(config)#ip route 0.0.0.0 0.0.0.0 12.12.12.2
```

Dan untuk verivikasi kita dapat lakukan test ping dari client- client yang berada pada jaringan local meuju internet atau server

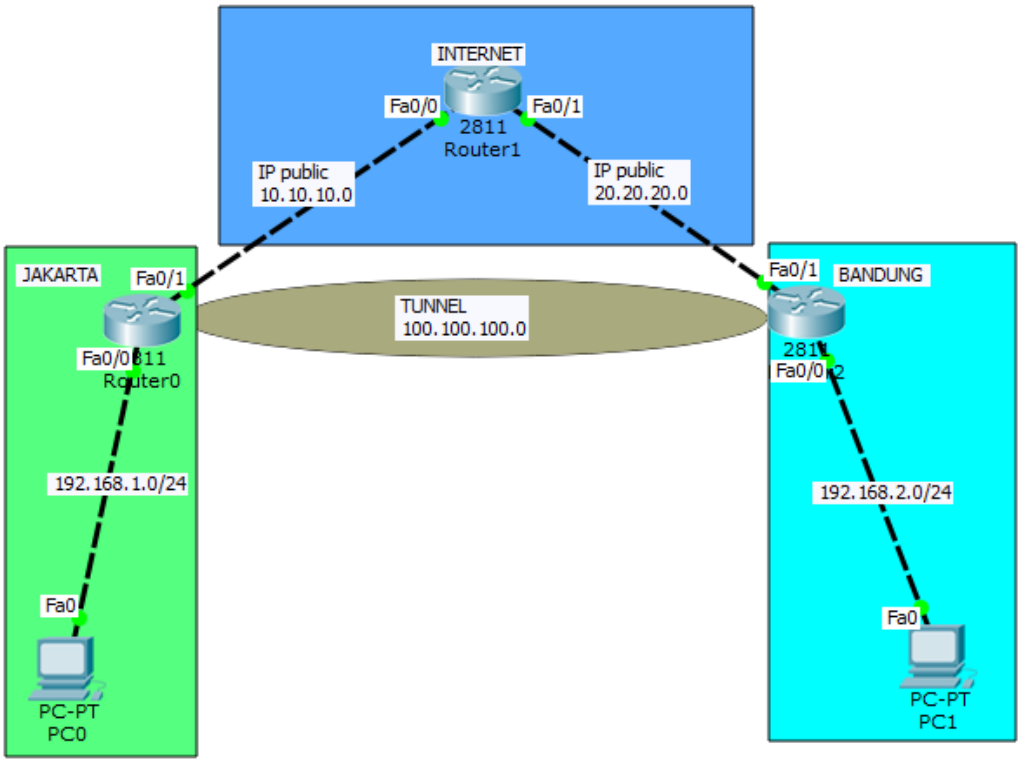
```
PC>ping 20.20.20.1

Pinging 20.20.20.1 with 32 bytes of data:

Reply from 20.20.20.1: bytes=32 time=0ms TTL=126
Reply from 20.20.20.1: bytes=32 time=1ms TTL=126
Reply from 20.20.20.1: bytes=32 time=0ms TTL=126
Reply from 20.20.20.1: bytes=32 time=0ms TTL=126

Ping statistics for 20.20.20.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

TUNNEL GRE

Apa itu tunnel..??

Tunnel merupakan salah satu cara pengiriman data melalui jalur tersendiri/privat yang kita buat dalam intrenet, agar suatu packet dapat terkirim dalam lingkup yang jauh, sebagai contoh di saat kita memiliki beberapa kantor cabang di berbagai daerah maka pada saat kita mengirim data ke antar kantor yang lain nya maka tidak di mungkin kan kita menggunakan kabel karna itu akan memakan biaya yang sangat mahal dan tidak efsien dalam penggunaan ny, maka di butuh kan tunnel atau suatu terowongan agar kita dapat mengirim data hanya dengan menggunakan internet saja dan mengkonfigurasikan tunnel pada suatu jalur di internet maka packet pun akan terkirim melalui jalur data internet yaitu melalu tunnel.

pada inti nya tunnel merupakan sebuah jalur privat atau yang dapat di seperti terowongan yang berada di dalam internet.



Tujuan lab pada kali ini kita akan menghubungkan kedua router yaitu dari router kantor di bandung dan di jakarta agar dapat terhubung melalui tunnel yaitu melalui network 100.100.100.0 bukan melewati 10.10.10.0 dan 20.20.20.0.

Sebelum kita mengkonfigurasikan tunnel kita harus mengkonfigurasikan IP terlebih dahulu.

R1(jakarta)

```
R1(config)#int fa0/0
R1(config-if)#no sh
R1(config-if)#ip ad 192.168.1.1 255.255.255.0
R1(config-if)#int fa0/1
R1(config-if)#no sh
R1(config-if)#ip ad 10.10.10.1 255.255.255.0
```

R2(bandung)

```
R2(config)#int fa0/0
R2(config-if)#no sh
R2(config-if)#ip ad 192.168.2.1 255.255.255.0
R2(config-if)#int fa0/1
R2(config-if)#no sh
R2(config-if)#ip ad 20.20.20.2 255.255.255.0
```

R3(internet)

```
R3(config)#int fa0/0
R3(config-if)#no sh
R3(config-if)#ip ad 10.10.10.3 255.255.255.0
R3(config-if)#int fa0/1
R3(config-if)#no sh
R3(config-if)#ip ad 20.20.20.3 255.255.255.0
```

Setelah setiap router di beri IP sesuai pada topologi maka kita dapat, merouting router – router tersebut agar dapat saling terhubung terlebih dahulu, tetapi kita tidak merouting local network di setiap router nya, karna internet tidak boleh mengetahui local address.

Kita akan merouting dengan routing dynamic eigrp

R1(jakarta)

```
R1(config)#router eigrp 10
R1(config-router)#network 10.10.10.0
```

R2(bandung)

```
R2(config)#router eigrp 10
R2(config-router)#network 20.20.20.0
```

R3(internet)

```
R3(config)#router eigrp 10
```

```
R3(config-router)#network 10.10.10.0
R3(config-router)#network 20.20.20.0
```

Setelah setiap router sudah di routing maka coba pastikan antar router sudah dapat ping atau saling terhubung.

R1(jakarta)

```
R1#ping 20.20.20.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 20.20.20.2, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
0/0/2 ms
```

R2(bandung)

```
R2#ping 10.10.10.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.1, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
0/0/0 ms
```

Setelah setiap router sudah dapat terhubung maka kita dapat langsung membuat tunnel dengan membuat IP tunnel telebih dahulu di interface tunnel, setelah itu konfigurasikan tunnel source dan tunnel destination nya.

R1(jakarata)

```
R1(config)#interface tunnel 0
R1(config-if)#ip ad 100.100.100.1 255.255.25.0
R1(config-if)#tunnel source fa0/1
R1(config-if)#tunnel destination 20.20.20.1
```

R2(bandung)

```
R2(config)#interface tunnel 0
R2(config-if)#ip ad 100.100.100.2 255.255.255.0
R2(config-if)#tunnel source fa0/1
R2(config-if)#tunnel destination 10.10.10.1
```

Sampai sini coba kita test PING antar host yang berada pada router yang di jakarta dan di bandung.

```
PC1JKT>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Reply from 192.168.1.1: Destination host unreachable.
```

```
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.

Ping statistics for 192.168.2.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

```
PC2BNDNG>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.2.1: Destination host unreachable.
Reply from 192.168.2.1: Destination host unreachable.
Reply from 192.168.2.1: Destination host unreachable.
Reply from 192.168.2.1: Destination host unreachable.

Ping statistics for 192.168.1.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Maka hasil pada saat mengirim packet pun akan unreachable, yang mana paket tertolak maka kita harus meanmbahkan routing di dalam tunnel untuk menuju antar local network.

Kita dapat membuat routing diantara tunnel dengan local network dengan static routing.

```
R1(config)#ip route 192.168.2.0 255.255.255.0 100.100.100.2
```

```
R2(config)#ip route 192.168.1.0 255.255.255.0 100.100.100.1
```

Isikan gateway dengan memasukan interface tunnel antar router, setelag itu kita dapat coba test ping kembali antar host di router jakarta dan bandung.

```
PC1JKT>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Reply from 192.168.2.2: bytes=32 time=1ms TTL=126
Reply from 192.168.2.2: bytes=32 time=0ms TTL=126
Reply from 192.168.2.2: bytes=32 time=0ms TTL=126
Reply from 192.168.2.2: bytes=32 time=0ms TTL=126

Ping statistics for 192.168.2.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

```
PC2BNDG>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:
```

```
Reply from 192.168.1.2: bytes=32 time=0ms TTL=126
Reply from 192.168.1.2: bytes=32 time=0ms TTL=126
Reply from 192.168.1.2: bytes=32 time=0ms TTL=126
Reply from 192.168.1.2: bytes=32 time=0ms TTL=126

Ping statistics for 192.168.1.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

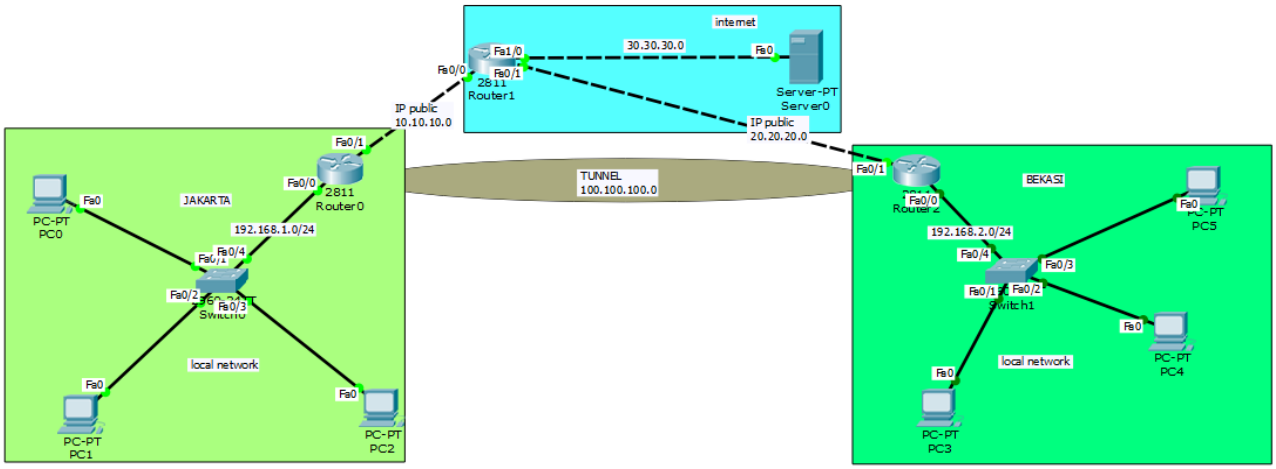
Maka setelah di routing di tunnel maka ia akan reply dan sudah dapat terhubung, kemudian kita dapat coba traceroot untuk meilhat jalur manakah yang di lewati oleh packet tersebut.

```
PC1JKT>tracert 192.168.2.2

Tracing route to 192.168.2.2 over a maximum of 30 hops:

1 0 ms 0 ms 1 ms 192.168.1.1
2 0 ms 0 ms 0 ms 100.100.100.2
3 0 ms 0 ms 0 ms 192.168.2.2
```

Maka ia pun akan melewati interface tunnel bukan melewati IP public atau yang memlalui router internet.

TUNNEL GRE WITH NAT

Kalau pada lab sebelum nya kita hanya mengkonfigurasikan tunnel saja, maka pada lab kali ini kita akan mengkonfigurasikan tunnel dengan menggunakan NAT juga agar host pada tiap – tiap router memiliki akses internet.



Tujuan pada lab kali ini kita akan membuat tunnel antar router yang ada di jakarta dan di bandung, tetapi pada host yang berada di router – router tersebut mendapatkan akses internet dengan menggunakan NAT.

Sebelum mulai mengkonfigurasikan maka kita dapat mencoba untuk mengkonfigurasikan IP terlebih dahulu di setiap router dan perangkat lain nya sesuai seperti yang di topologi

R1

```
R1(config)#int fa0/0
R1(config-if)#no sh
R1(config-if)#ip ad 192.168.1.1 255.255.255.0
R1(config-if)#int fa0/1
R1(config-if)#no sh
R1(config-if)#ip ad 10.10.10.1 255.255.255.0
```

R2

```
R2(config)#int fa0/0
R2(config-if)#no sh
R2(config-if)#ip ad 192.168.2.1 255.255.255.0
R2(config-if)#int fa0/1
R2(config-if)#no sh
R2(config-if)#ip ad 20.20.20.2 255.255.255.0
```

R3

```
R3(config)#int fa0/0
R3(config-if)#no sh
R3(config-if)#ip ad 10.10.10.3 255.255.255.0
R3(config-if)#int fa0/1
```

```
R3(config-if)#no sh
R3(config-if)#ip ad 20.20.20.3 255.255.255.0
R3(config-if)#int fa1/0
R3(config-if)#no sh
R3(config-if)#ip ad 30.30.30.3 255.255.255.0
```

Setelah setiap router sudah di beri IP masing – masing maka kita dapat meng-konfigurasi kan routing dynamic antar router jakarta, bekasi dan internet dengan dynnamic routing

```
R1(config)#router eigrp 10
R1(config-router)#network 10.10.10.0
```

```
R2(config)#router eigrp 10
R2(config-router)#network 20.20.20.0
```

```
R3(config)#router eigrp 10
R3(config-router)#network 10.10.10.0
R3(config-router)#network 20.20.20.0
R3(config-router)#network 30.30.30.0
```

Setelah sudah kita membuat routing kita dapat test dengan PING antar router terlebih dahulu

```
R1#ping 20.20.20.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 20.20.20.2, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
0/3/19 ms
```

```
R2#ping 10.10.10.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.1, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
0/0/0 ms
```

Kalau sudah dapat di PING kita dapat coba membuat NAT terlebih dahulu agar setiap host memiliki akses internet melalui NAT di setiap router nya.

Buat default route untuk NAT, kita dapat membuat NAT dengan NAT dynamic overload di karna kan kita memiliki beberapa client pada setiap router nya.

R1

```
R1(config)#ip route 0.0.0.0 0.0.0.0 10.10.10.3
R1(config)#access-list 1 permit any
R1(config)#ip nat inside source list 1 interface fa0/1 overload
R1(config)#int fa0/1
R1(config-if)#ip nat outside
R1(config-if)#int fa0/0
R1(config-if)#ip nat inside
```

R2

```
R2(config)#ip route 0.0.0.0 0.0.0.0 20.20.20.3
R2(config)#access-list 1 permit any
R2(config)#ip nat inside source list 1 interface fa0/1 overload
R2(config)#int fa0/1
R2(config-if)#ip nat outside
R2(config-if)#int fa0/0
R2(config-if)#ip nat inside
```

Setelah sudah kita buat NAT untuk host pada masing – masing router maka kita dapat coba test ping menuju internet.

```
PC1JKT>ping 30.30.30.3

Pinging 30.30.30.3 with 32 bytes of data:

Reply from 30.30.30.3: bytes=32 time=58ms TTL=254
Reply from 30.30.30.3: bytes=32 time=0ms TTL=254
Reply from 30.30.30.3: bytes=32 time=34ms TTL=254
Reply from 30.30.30.3: bytes=32 time=0ms TTL=254

Ping statistics for 30.30.30.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 58ms, Average = 23ms
```

```
PC2BKS>ping 30.30.30.3

Pinging 30.30.30.3 with 32 bytes of data:

Reply from 30.30.30.3: bytes=32 time=1ms TTL=254
Reply from 30.30.30.3: bytes=32 time=0ms TTL=254
Reply from 30.30.30.3: bytes=32 time=0ms TTL=254
Reply from 30.30.30.3: bytes=32 time=0ms TTL=254

Ping statistics for 30.30.30.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Maka PC di setiap router sudah memiliki akses internet maka kita dapat langsung mengkonfigurasikan tunnel pada R1(jakarta) dan R2(bekasi)

```
R1(config)#int tunnel 0
R1(config-if)#ip address 100.100.100.1 255.255.255.0
R1(config-if)#tunnel source fa0/1
R1(config-if)#tunnel destination 20.20.20.2
```

```
R2(config)#int tunnel 0
R2(config-if)#ip add 100.100.100.2 255.255.255.0
R2(config-if)#tunnel source fa0/1
R2(config-if)#tunnel destination 10.10.10.1
```

Setelah kita buat interface tunnel dan membuat tunnel nya maka kita belum dapat terhubung host antar router, maka kita harus membuat routing antar host dengan IP tunnel sebagai gateway, dan kita dapat membuat routing dengan menggunakan static route.

```
R1(config)#ip route 192.168.2.0 255.255.255.0 100.100.100.2
```

```
R2(config)#ip route 192.168.1.0 255.255.255.0 100.100.100.1
```

Maka dengan ini host antar router di jakarta dan di bekasi sudah saling terhubung menggunakan tunnel, setelah itu coba kita test ping dari host yang berada di R1(jakarta) menuju host yang berada di R2(bekasi)

```
PC1JKT>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Reply from 192.168.2.2: bytes=32 time=1ms TTL=126
Reply from 192.168.2.2: bytes=32 time=0ms TTL=126
Reply from 192.168.2.2: bytes=32 time=10ms TTL=126
Reply from 192.168.2.2: bytes=32 time=0ms TTL=126

Ping statistics for 192.168.2.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 10ms, Average = 2ms
```

Dan sebalik nya dari host yang berada di R2(bekasi) menuju host yang berada di R1(jakarta)

```
PC2BKS>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time=0ms TTL=126
Reply from 192.168.1.2: bytes=32 time=0ms TTL=126
Reply from 192.168.1.2: bytes=32 time=0ms TTL=126
```

```
Reply from 192.168.1.2: bytes=32 time=0ms TTL=126

Ping statistics for 192.168.1.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Maka dengan ini setiap host di kedua router sudah dapat terhubung melalui tunnel dan memiliki akses internet melalui NAT dynamic overload