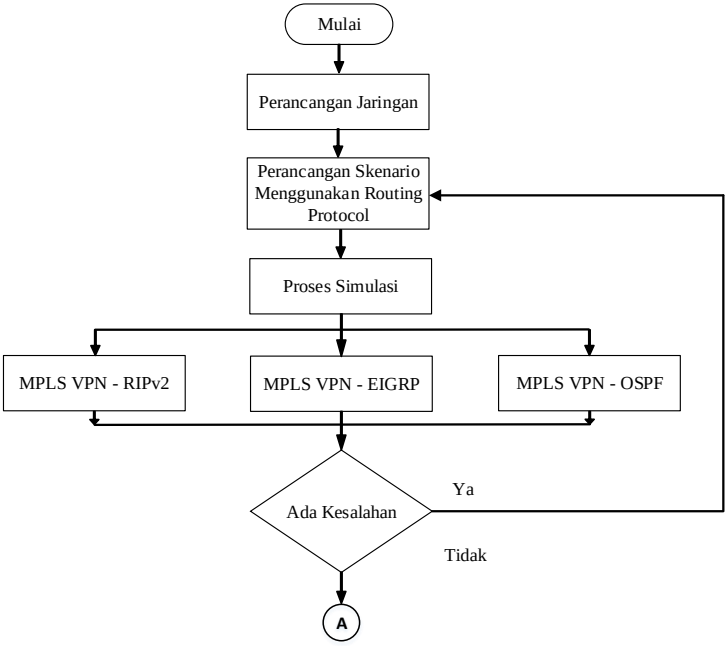
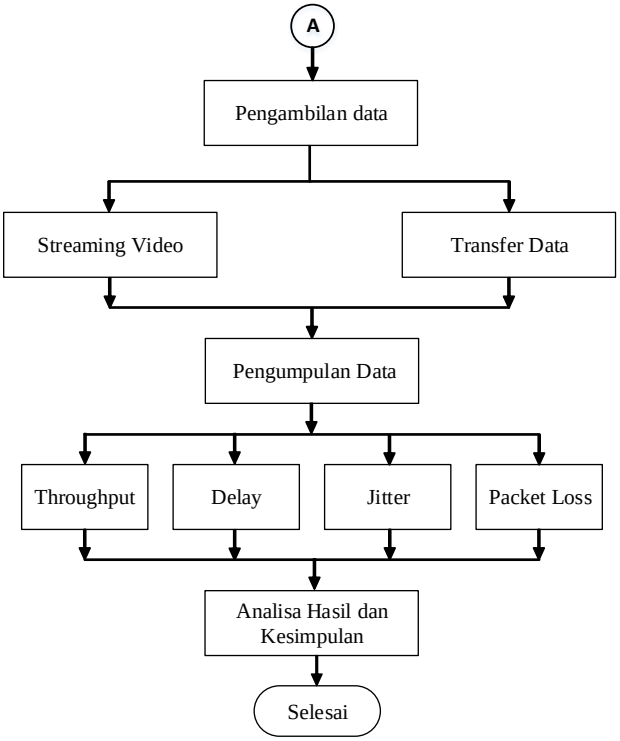
BAB III
PERANCANGAN DAN IMPLEMENTASI JARINGAN

Dalam merancang model penelitian skripsi "Analisis Perbandingan QOS Protokol EIGRP, OSPF, dan RIPv2 Pada Jaringan *Backbone* Menggunakan Teknologi MPLS VPN" terbagi menjadi beberapa tahapan yaitu Perancangan Topologi Jaringan, Pembangunan Topologi Jaringan, Pengumpulan Data, Analisa Hasil dan Kesimpulan terhadap hasil pengumpulan data parameter yang telah didapat pada proses simulasi. Pada BAB III ini akan menerangkan tentang konfigurasi jaringan MPLS VPN yang digunakan sebagai pembanding antara protokol *routing* EIGRP, protokol *routing* OSPF, dan protokol *routing* RIP. Dan juga berisi penjelasan topologi jaringan yang digunakan serta simulasi yang digunakan untuk pengambilan data sebagai tujuan dari penulisan penelitian.

Beberapa hal yang dibahas pada bab ini yaitu *Flowchart* pengerjaan penelitian skripsi, skenario pengambilan data, dan konfigurasi topologi jaringan

3.1 *Flowchart* Pengerjaan





Gambar 3.1 *Flowchart* Pengerjaan Keseluruhan

Pada subbab ini akan menampilkan *flowchart* pengerjaan simulasi yang digunakan pada penelitian ini secara keseluruhan. *Flowchart* yang menampilkan proses pengerjaan penelitian secara keseluruhan dapat dilihat pada gambar 3.1. *Flowchart* pengerjaan menggambarkan langkah-langkah untuk melakukan simulasi Analisis Perbandingan QOS Protokol EIGRP, OSPF, dan RIPv2 Pada Jaringan *Backbone* Menggunakan Teknologi MPLS VPN, *flowchart* dimulai dari tahap fase "mulai". Fase "mulai" merupakan fase pertama yang dilakukan pada penelitian ini. Setelah memasuki fase "mulai" maka selanjutnya akan memasuki fase tahap kedua yaitu "perancangan jaringan".

Pada tahap kedua yaitu "perancangan jaringan" yang dilakukan adalah membuat rancangan topologi jaringan yang akan digunakan dan dibuat simulasinya pada *software* GNS3. Jaringan ini dibuat secara manual sehingga

dapat memaksimalkan proses simulasi yang akan dikerjakan dalam penelitian ini. Dari tahap ini lalu masuk ke tahap ke tiga yaitu "pembuatan skenario" Dimana skenario yang akan diujikan dalam penelitian ini yaitu perbandingan antara *routing* protokol OSPF, *routing* protokol EIGRP, dan *routing* protokol RIPv2 pada jaringan MPLS VPN. Pemilihan *routing – routing* protokol tersebut yang digunakan dalam penelitian ini adalah karena *routing – routing* protokol tersebut merupakan *routing* protokol standar yang umum digunakan pada dunia kerja.

Setelah tahap ketiga selesai kemudian masuk pada tahap keempat yaitu tahap "proses simulasi". Tahap ini yaitu memulai proses simulasi pada jaringan yang telah dibuat pada tahap kedua dengan skenario yang telah dibuat pada tahap ketiga sehingga didapatkan hasil yang akan dianalisa pada penelitian ini. Tahap keempat ini akan dilanjutkan pada tahap kelima dimana akan terbagi menjadi 3 jalur berbeda sesuai dengan *routing* protokol yang digunakan yaitu *routing* protokol OSPF, *routing* protokol EIGRP, dan *routing* protokol RIPv2 yang digunakan pada jaringan MPLS VPN.

Pada tahap kelima ini lalu dilanjutkan ke tahan keenam yaitu "ada kesalahan" di mana pada tahap ini akan dilihat apakah dari tahap keempat dan kelima terdapat kesalahan dalam menjalankan proses simulasi atau tidak, jika terdapat kesalahan maka dalam proses simulasi akan dihentikan dan akan dikembalikan kembali ke tahap keempat yaitu "proses simulasi" di mana sebelumnya akan di cek kembali apakah proses simulasi sudah berjalan secara benar atau belum. Jika tidak terdapat kesalahan dengan kata lain proses simulasi sudah sesuai dengan yang diharapkan, maka akan diteruskan menuju tahap selanjutnya yaitu tahap ketujuh yaitu "pengambilan data". Pada Tahap "pengumpulan data", data yang dikumpulkan didapat dari dua buah metode pengambilan data yaitu metode "*Streaming Video*" dan metode "*Transfer* Data". Dua buah metode ini digunakan untuk menguji kehandalan jaringan yang telah dirancang pada tahap kedua. Kemudian setelah tahap ini selesai lalu dialnjutkan dengan tahap "pengumpulan data".

Pada tahap ini akan difokuskan pada beberapa parameter yang diamati yaitu "*throughput*", "*delay*", "*jitter*", serta "*packet loss*". Data – data tersebut kemudian dikumpulkan sebagai hasil dari simulasi yang nantinya akan dianalisa. Jika data – data tersebut sudah cukup terkumpul maka kemudian akan masuk pada tahap selanjutnya yaitu tahap kesembilan "analisa hasil dan kesimpulan" lalu "selesai"

Pada penelitian ini menggunakan 3 buah skenario jaringan di mana masing – masing skenario menggunakan *routing* protokol yang berbeda. Jaringan terdiri dari sebuah *server* dan *client* dan dihubungkan dengan jaringan *backbone* berbasis MPLS VPN. Layanan yang akan diuji pada penelitian ini yaitu layanan video serta layanan transfer data yang akan dipertukarkan antara *server* dengan *client*. Kemudian pada jaringan tersebut ditambahkan *traffic generator* yang berguna untuk membanjiri jaringan pada simulasi agar serupa dengan *traffic* sesungguhnya. Dari kedua metode layanan yaitu video dan data maka akan didapatdata – data yang digunakan sebagai perbandingan nilai parameter seperti ““*throughput*”, “*delay*”, “*jitter*”, serta “*packet loss*” pada masing – masing skenario jaringan.

Tabel 3.1 Rancangan Skenario *Routing* P*border*rotokol OSPF, EIGRP dan RIPv2.

No.	Skenario	Layanan
1	MPLS VPN OSPF	Video
2	MPLS VPN OSPF	Data
3	MPLS VPN EIGRP	Video
4	MPLS VPN EIGRP	Data
5	MPLS VPN RIPv2	Video
6	MPLS VPN RIPv2	Data

3.2 Persiapan Penelitian

3.2.1. Perangkat Keras (*Hardware*)

Adapun perangkat yang dipakai dalam penelitian ini memiliki spesifikasi sebagai berikut :

a. 1 buah *Personal Computer* untuk menjalankan *software* GNS3 dengan spesifikasi :
   - *Processor* Intel Core i7
   - RAM DDR3 8 *Gigabyte*
   - VGA Nvidia 1 *Gigabyte*

- *Harddisk* Sata 500 *Gigabyte*
- 2 buah NIC (*Network Interface Card*)
- 1 buah LAN *Port On Board*
- *Operating System* Microsoft Windows 7 Ultimate

b. 2 buah *Personal Computer* sebagai *Client* atau *Server* dengan spesifikasi :
   - *Processor* Intel Core i7
   - RAM DDR3 8 *Gigabyte*
   - VGA Nvidia 1 *Gigabyte*
   - *Harddisk* Sata 500 *Gigabyte*
   - 1 buah LAN *Port On Board*
   - *Operating System* Microsoft Windows 7 Ultimate

c. 1 buah Laptop ASUS A43S sebagai *traffic generator* dengan spesifikasi
   - *Processor* Intel Core i3
   - RAM DDR3 4 *Gigabyte*
   - VGA Nvidia 1 *Gigabyte*
   - *Harddisk* Sata 500 *Gigabyte*
   - 1 buah LAN *Port On Board*
   - *Operating System* Microsoft Windows 7 Ultimate

d. Serta peralatan pendukung seperti :
   - 3 Buah Kabel UTP tipe *Cross* sebagai penghubung antar perangkat

3.2.2. Perangkat Lunak (*Software*)

Graphical Network Simulator 3 (GNS3) merupakan sebuah *software* pemodelan jaringan komputer berbasis GUI atau *Graphical User Interface* yang dimana program ini dapat mensimulasikan jaringan komputer baik berupa LAN atau WAN atau *router* secara lebih nyata sehingga tidak harus menggunakan perangkat asli. Secara umum GNS3 mirip dengan *software* Packet Tracer milik Cisco tetapi dapat memungkinkan membuat jaringan yang lebih kompleks karena menggunakan *Operating System* asli milik Cisco yaitu Cisco IOS (*Internetwork Operating System*) dan milik Juniper yaitu Junos (*Juniper Operating System*). GNS3 dapat dijalan di berbagai sistem operasi, seperti Windows, Linux, ataupun Mac.

Untuk *System Requirements* GNS3 1.3.11 yang dipakai pada penelitian kali ini adalah :

- OS : Windows 7 (64 bit) dan seterusnya.
- *Processor* : 2 *or more logical cores*
- RAM : 4 *Gigabyte*
- *Harddisk* : Ruang kosong sebesar 1 *Gigabyte*

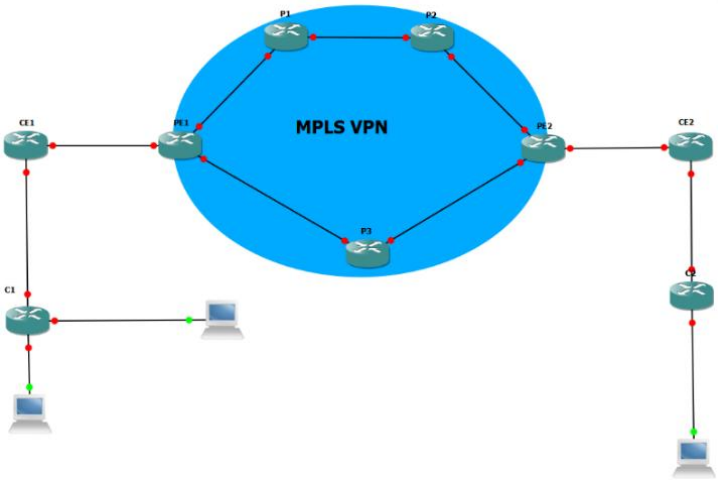
Untuk *software* analisa yang dipakai dalam penelitian kali ini menggunakan *software Wireshark.*

Wireshark merupakan *software* aplikasi *Network Analyzer* yang dapat digunakan untuk memonitor dan menangkap paket – paket data atau informasi dalam satu jaringan komputer secara *real time* berbasis GUI. Dari paket – paket data yang ditangkap tersebut, *Wireshark* akan berusaha untuk menampilkan semua informasi dalam paket data tersebut secara lengkap dan sedetail mungkin dengan format yang dapat dipahami oleh penggunanya. *Wireshark* dapat melakukan paket *filtering*, paket *colour coding*, inspeksi paket data secara individu, dan fitur – fitur lain. *Wireshark* dapat dijalankan di berbagai sistem operasi seperti Windows, Linux, dan Mac.

3.3 Pembuatan Topologi Jaringan

Topologi jaring yang dipakai dalam penelitian kali ini merupakan topologi jaringan komputer yang dirancang penulis untuk diimplementasikan teknologi MPLS VPN. Jaringan MPLS VPN ini menggunakan *routing* protokol EIGRP, OSPF, dan RIPv2 sebagai penghubung antar *router* agar dapat saling terkoneksi.

Topologi yang digunakan pada penelitian kali ini dapat dilihat pada gambar 3.4 dimana terdapat jaringan MPLS VPN dengan memiliki 3 *router provider*, 2 *router provider edge* dan 2 *router customer edge* dan 2 *router customer* yang dari *router costumer* tersebut akan terhubung dengan *client/server.* Ditambahkan juga *traffic generator* yang membanjiri jaringan dengan data agar serupa dengan jaringan sungguhan.



Gambar 3.2 Topologi Jaringan

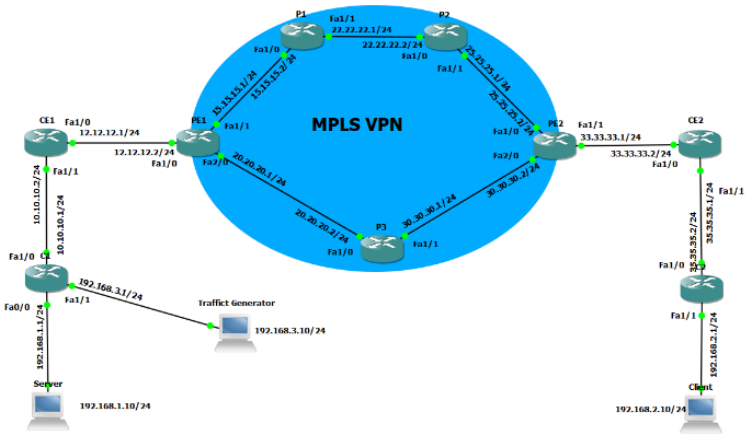
3.3.1. Pembuatan Skenario

Pada subbab ini akan dibahas pembuatan skenario jaringan yang dipakai dalam penelitian kali ini. Topologi jaringan yang dibuat dibagi menjadi 3 buah topologi sesuai dengan *routing* protokol pada jaringan MPLS VPN :

a. Topologi pertama menggunakan *routing* protokol OSPF pada jaringan MPLS VPN
b. Topologi yang kedua menggunakan *routing* protokol EIGRP pada jaringan MPLS VPN
c. Topologi ketiga menggunakan *routing* protokol RIPv2 pada jaringan MPLS VPN

Layanan yang dipakai pada masing – masing skenario pada penelitian ini ada 2 buah layanan yaitu :

a. Layanan *Streaming* Video
b. Layanan *Transfer* Data



Gambar 3.3 Topologi jaringan MPLS VPN dengan *routing* protokol OSPF, EIGRP dan RIPv2

Gambar 3.5 merupakan konfigurasi secara lengkap topologi MPLS VPN yang digunakan pada penelitian ini. Konfigurasi topologi juga menggunakan *routing* protokol BGP dan pada bagian jaringan MPLS menggunakan *routing* protokol OSPF karena jaringan MPLS hanya dapat menggunakan *routing* protokol berbasis *link state*.

Tabel 3.2 *IP Address* yang digunakan *Provider*

Router	**IP**	**Subnet Mask**	***Interface***
P1	15.15.15.2	255.255.255.0	Fast Ethernet 1/0
	22.22.22.1	255.255.255.0	Fast Ethernet 1/1
	4.4.4.4	255.255.255.255	Loopback 0
P2	22.22.22.2	255.255.255.0	Fast Ethernet 1/0
	25.25.25.1	255.255.255.0	Fast Ethernet 1/1
	5.5.5.5	255.255.255.255	Loopback 0
P3	20.20.20.2	255.255.255.0	Fast Ethernet 1/0
	30.30.30.1	255.255.255.0	Fast Ethernet 1/1
	6.6.6.6	255.255.255.255	Loopback 0

Tabel 3.2 merupakan list konfigurasi dari alamat *IP* yang digunakan pada jaringan *provider* dimana pada jaringan *provider* juga digunakan konfigurasi *VPN* (*Virtual Private Network*). Pada konfigurasi jaringan *provider*

menggunakan *routing* protokol OSPF sebagai penghubung antar *interface router* dari *provider* ke *provider* serta dari *interface provider* ke *provider edge* yang saling terhubung. Pada setiap *interface* fisik di *router provider* dan pada setiap *interface* fisik di *router provider edge* yang terhubung langsung dengan *router provider* ditambahkan perintah untuk menghidupkan layanan mpls pada masing – masing *interface router*.

Tabel 3.3 *IP Address* yang digunakan *provider edge*

Router	**IP**	**Subnet Mask**	***Interface***
PE1	12.12.12.2	255.255.255.0	Fast Ethernet 1/0
	15.15.15.1	255.255.255.0	Fast Ethernet 1/1
	20.20.20.1	255.255.255.0	Fast Ethernet 2/0
	3.3.3.3	255.255.255.255	Loopback 0
PE2	25.25.25.2	255.255.255.0	Fast Ethernet 1/0
	33.33.33.1	255.255.255.0	Fast Ethernet 1/1
	30.30.30.2	255.255.255.0	Fast Ethernet 2/0
	7.7.7.7	255.255.255.255	Loopback 0

Tabel 3.3 merupakan konfigurasi dari alamat *IP* yang digunakan pada *router provider edge*. Pada konfigurasi yang dilakukan di *provider edge* tersebut hal – hal yang perlu diperhatikan adalah menambahkan perintah pada *interface* yang terhubung antara *provider edge* dan *provider* untuk menyalakan layanan mpls pada masing – masing *interface*. Sedang *interface* pada *provider edge* yang tidak terhubung dengan *router provider* tidak ditambahkan perintah tersebut. Untuk *interface provider edge* yang terhubung dengan *provider* dihubungkan dengan *routing* protokol OSPF seperti yang digunakan pada hubungan antar *router provider* karena merupakan satu kesatuan jaringan MPLS. Sedang untuk *interface provider edge* yang terhubung dengan *interface costumer edge* menggunakan *routing* protokol yang berbeda sesuai dengan skenario yaitu menggunakan *routing* protokol OSPF, EIGRP, dan RIPv2 tetapi menggunakan penambahan VRF yang bernama "Costumer_A" untuk menghubungkan dengan *router costumer edge* agar dapat dikenali sedangkan pada *router costumer edge* tidak ditambahkan penamaan VRF. Penamaan VRF digunakan karena pada *interface* pada *provider edge* yang terhubung dengan *costumer edge* ditambahkan VRF sehingga *routing* protokol pada *interface* tersebut harus juga ditambahkan penaaman VRF.

Tabel 3.4 *IP Address* yang digunakan *customer edge*

Router	**IP**	**Subnet Mask**	***Interface***
CE1	12.12.12.1	255.255.255.0	Fast Ethernet 1/0
	10.10.10.2	255.255.255.0	Fast Ethernet 1/1
	2.2.2.2	255.255.255.255	Loopback 0
CE2	33.33.33.2	255.255.255.0	Fast Ethernet 1/0
	35.35.35.1	255.255.255.0	Fast Ethernet 1/1
	8.8.8.8	255.255.255.255	Loopback 0

Tabel 3.4 merupakan konfigurasi dari alamat *IP* yang digunakan pada *router customer edge*. Pada konfigurasi bagian ini menggunakan *routing* protokol yang berbeda sesuai dengan skenario yang telah dirancang yaitu menggunakan *routing* protokol OSPF, EIGRP, dan RIPv2 sesuai dengan skenario yang akan dijalankan. *Routing* protokol ini akan diterapkan pada *interface costumer edge* yang terhubung dengan *interface provider edge* serta *interface* yang terhubung dengan *interface* pada *router costumer*.

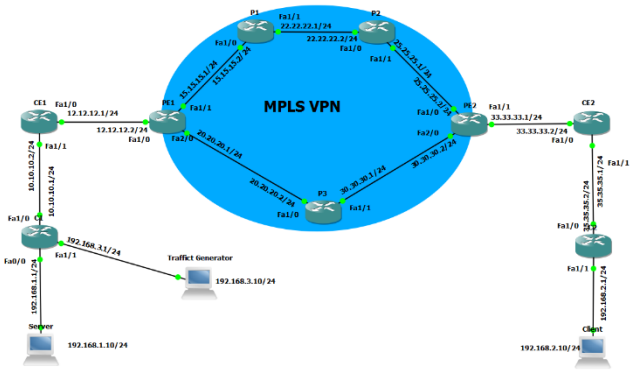
Tabel 3.5 *IP Address* yang digunakan *customer*

Router	**IP**	**Subnet Mask**	***Interface***
C1	10.10.10.1	255.255.255.0	Fast Ethernet 1/0
	192.168.3.1	255.255.255.0	Fast Ethernet 1/1
	192.168.1.1	255.255.255.0	Fast Ethernet 0/0
	1.1.1.1	255.255.255.255	Loopback 0
C2	35.35.35.2	255.255.255.0	Fast Ethernet 1/0
	192.168.2.1	255.255.255.0	Fast Ethernet 1/1
	9.9.9.9	255.255.255.255	Loopback 0

Tabel 3.5 merupakan konfigurasi dari alamat *IP* yang digunakan pada *router customer*. Untuk konfigurasi yang digunakan hampir sama dengan konfigurasi pada *costumer edge* dimana *interface* yang terhubung dengan *router costumer edge* menggunakan *routing* protokol sesuai dengan skenario yang dijalankan yaitu menggunakan *routing* protokol OSPF, EIGRP, dan RIPv2. Untuk *interface* yang terhubung dengan *client/server* maupun *traffic generator* tidak perlu menggunakan *routing* protokol.

3.3.2. Konfigurasi Topologi

Secara keseluruhan komponen jaringan yang digunakan pada *software* simulasi GNS3 seperti pada gambar 3.6.



Gambar 3.4 Komponen Jaringan Pada *GNS3*

Konfigurasi yang digunakan yaitu

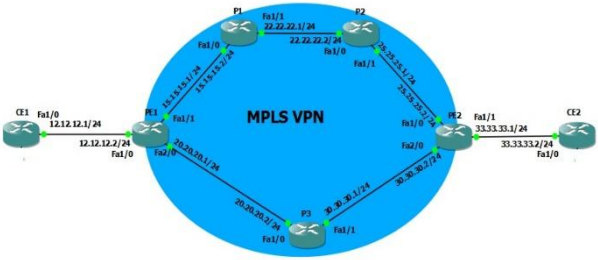
- 3 buah *Router Provider* (P)
- 2 buah *Router Provider Edge* (PE)
- 2 buah *Router Customer Edge* (CE)
- 2 buah *Router Customer* (C)
- 2 buah *Personal Computer* sebagai *Client/Server*
- 1 buah *Personal Computer Traffic Generator*

3.3.3. Konfigurasi MPLS VPN

Sesuai dengan judul penelitian ini yaitu ” Analisis Perbandingan QOS Protokol EIGRP, OSPF, dan RIPv2 Pada Jaringan *Backbone* Menggunakan Teknologi MPLS VPN” maka konfigurasi jaringan MPLS VPN menjadi salah satu parameter yang penting dalam keberhasilan pada penelitian skripsi. Semua skenario yang telah dirancang menggunakan VPN dalam berkomunikasi antara *client* dan *server*. Dalam penelitian ini digunakan nama “Customer_A” sebagai nama VPN pada topologi jaringan yang dipakai. Untuk semua *interface* yang dipakai dalam penelitian kali ini adalah *fastethernet* untuk menghubungkan antar *router* pada topologi jaringan MPLS VPN. Baik pada jaringan antar *router provider*, *router provider edge*, *router customer edge*, dan *router*

customer semua terhubung menggunakan *interface fastethenet* dalam simulasi penelitian.

Berikut adalah beberapa konfigurasi yang digunakan dalam pembuatan simulasi baik pada topologi OSPF.



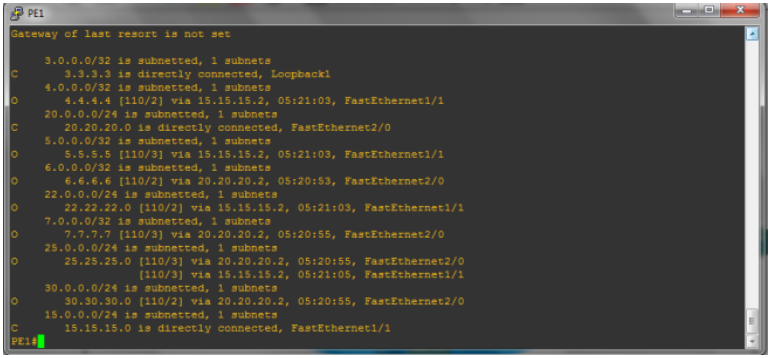
Gambar 3.5 Topologi jaringan MPLS VPN

3.3.4. Konfigurasi Topologi MPLS VPN

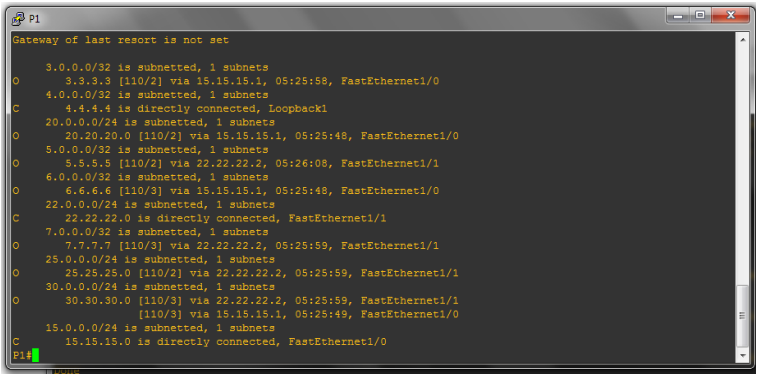
3.3.4.1. *Setting Router* OSPF

Langkah – langkah yang pertama dilakukan yaitu mengatur semua ip *interface* sesuai dengan tabel ip yang dibuat. Baik berupa *interface* fisik maupun *interface* logic seperti ip *loopback*. Selanjutnya yaitu menyeting *routing* protokol OSPF pada jaringan MPLS VPN yang ditandai dengan warna biru pada gambar 3.7 karena jaringan MPLS VPN tersebut hanya dapat diseting dengan *routing* protokol berbasis *link state* salah satunya yaitu OSPF. Penyetingan *routing* protokol OSPF dilakukan pada *router* provider dan *router* provider edge tetapi hanya pada *interface* yang terhubung dengan *router* provider.

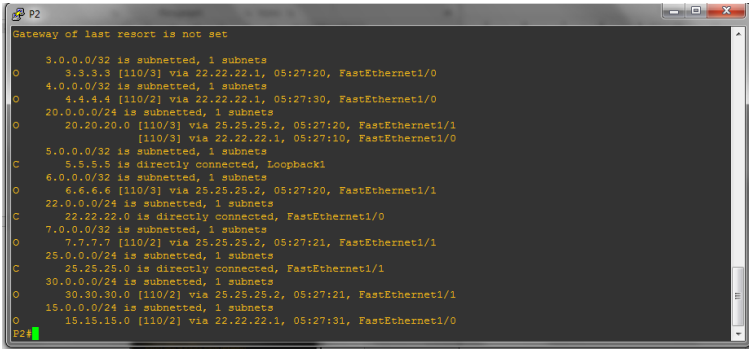
Fungsi dari penyettingan routing protokol OSPF ini adalah agar dapat menjembatani tiap *router* yang memiliki *network* berbeda agar dapat saling berkomunikasi sehingga setiap *interface* yang berada dalam kawasan MPLS VPN dapat saling terhubung sehingga jaringan MPLS VPN dapat terbentuk. Setelah selesai mengkonfigurasi masing – masing *router* maka hasilnya dapat dilihat seperti pada gambar di bawah ini.



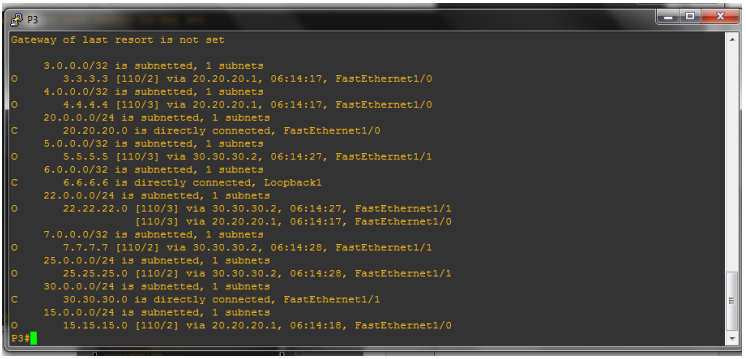
Gambar 3.6 Tampilan *routing* protokol pada PE1



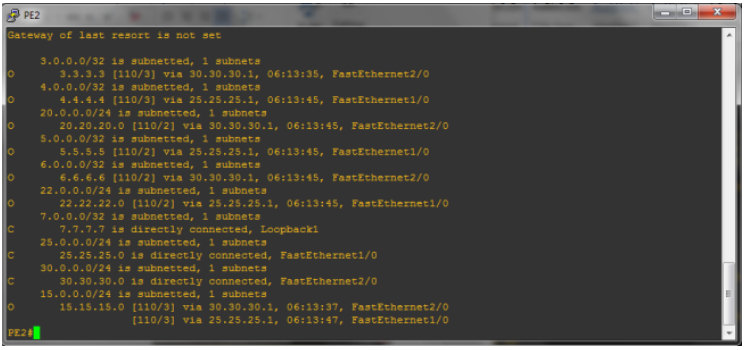
Gambar 3.7 Tampilan *routing* protokol pada P1



Gambar 3.8 Tampilan *routing* protokol pada P2



Gambar 3.9 Tampilan *routing* protokol pada P3



Gambar 3.10 Tampilan *routing* protokol pada PE2

Untuk mengetahui apakah *routing* protokol OSPF telah saling terhubung biasanya akan keluar pemberitahuan ADJCHG atau *adjencency* yang berarti *routing* protokol OSPF telah berhasil saling terhubung antar *router* yang sudah dikonfigurasi.

Konfigurasi *routing* protokol OSPF ini akan dipakai pada semua skenario jaringan dikarenakan merupakan konfigurasi *routing* protokol dasar yang dipakai pada jaringan MPLS VPN pada penelitian ini.

3.3.4.2. *Setting* MPLS IP

Layanan MPLS (*Multi Protocol Label Switching*) dapat diaktifkan pada *interface* dengan perintah “mpls ip”. Tetapi layanan ini harus diaktifkan di area MPLS untuk mendukung terbentuknya jaringan MPLS. Pada konfigurasi MPLS dilakukan pada setiap *interface router provider* yang terhubung dengan

interface router provider lainnya dan *interface router provider* yang terhubung dengan *interface* pada *router provider edge* dan sebaliknya yang terhubung secara langsung. Untuk lebih jelasnya dapat dilihat pada gambar 3.7 dimana pada area biru terdapat *interface* yang saling terhubung. Di dalam *interface – interface* tersebut yang akan diaktifkan layanan MPLS. Fungsi dari penyetingan MPLS ini adalah pada saat transfer data antar *interface* yang sudah mengaktifkan layanan MPLS berlangsung lebih cepat karena menggunakan teknik "*labelling*" sehingga dapat meningkatkan performasi jaringan.

Konfigurasi layanan MPLS ini akan dipakai pada semua skenario jaringan dikarenakan merupakan merupakan konfigurasi dasar pada jaringan MPLS pada penelitian ini.

3.3.4.3. *Setting* BGP

Pada penyettingan BGP (*Border Gateway Protocol*) ini berfungsi agar dapat menghubungkan *Autonomus System* yang berbeda dimana disini *autonomus system* yang coba di bangun adalah *autonomus system* antara *routing* protokol OSPF yang berada pada area OSPF dengan *routing* protokol yang berada di luar area MPLS sesuai dengan skenario yang dibuat yaitu *routing* protokol OPSF, EIGRP , atau RIPv2. Penyetingan dilakukan pada setiap *provider edge* dikarenakan *provider edge* merupakan gerbang antara sisi *customer* dengan *provider.*

Konfigurasi BGP ini akan dipakai pada semua skenario jaringan dikarenakan merupakan merupakan konfigurasi BGP dasar pada jaringan MPLS pada penelitian ini.

3.3.4.4. *Setting* VRF

Pada penyetingan VRF (*Virtual Routing and Forwading*) pada jaringan MPLS VPN diperlukan agar beberapa *routing* protokol yang berbeda pada *router* dapat berdiri sendiri dan tidak saling bercampur. Untuk itu dibutuhkan sebuah *router* virtual Sehingga jika ada beberapa *costumer* dapat memiliki *tabel routing* tersendiri sesuai dengan kebutuhannya. Pada penyetingan *router* virtual ditambahkan perintah "rd", rd yang dimaksud di sini adalah "*Route-Distinguisher*" yang merupakan identifikasi nomor unik dari setiap tabel *routing* untuk tiap vrf. Untuk mengaktifkan vrf dilakukan pada *router provider edge* baik PE1 dan PE2 di *interface* yang terhubung dengan sisi *costumer*. VRF diaktifkan pada router PE karena pada *router* PE merupakan gerbang

penghubung antara wilayah MPLS dengan wilayah *user* dan juga pada *router* PE merupakan tempat dibangunnya jaringan LSP.

Konfigurasi VRF ini akan dipakai pada semua skenario jaringan dikarenakan merupakan merupakan konfigurasi dasar pada jaringan MPLS pada penelitian ini. Setelah selesai mengkonfigurasi VRF pada sisi *provider edge* maka langkah selanjutnya yaitu mengkonfigurasi *routing* protokol pada *router customer edge* sesuai dengan skenario yang digunakan. Sehingga perintah – perintah yang telah dijabarkan sebelum ini merupakan perintah dasar pada semua skenario jaringan MPLS VPN yang digunakan pada penelitian ini. Salah satu contoh konfigurasi yang digunakan jika menjalankan skenario OSPF.

Untuk skenario ospf maka pada sisi *costumer edge* konfigurasi *routing* protokol yang digunakan ospf tetapi dengan *id* yang berbeda dengan *id* pada *routing* protokol ospf di jaringan MPLS. Untuk *routing* protokol yang berada pada *interface provider edge* yang terhubung dengan *interface* pada *router costumer edge* memiliki *id* yang sama dengan yang terdapat pada *router costumer edge* tetapi ditambahkan VRF yang telah dikonfigurasi pada perintah sebelumnya dikarenakan *interface* yang terdaftar pada saat membuat *routing* protokol merupakan *interface* yang telah dipasang layanan VRF.

3.3.4.5. *Setting Redistribute*

Pada bagian ini perintah yang akan dijalankan memiliki fungsi untuk penghubung antar *routing* protokol yang berbeda. Sistem *Redistribute* ini memiliki fungsi sebagai jembatan agar *routing* protokol yang berbeda dalam suatu jaringan dapat saling mengenali sehingga dapat bertukar *tabel routing* antar *routing* protokol. Perintah Redistribute ini akan dipasang pada *routing* protokol PE1 di bagian *routing* protokol BGP dan *routing* protokol yang terdapat pada *interface provider edge* yang terhubung dengan *interface costumer edge* sesuai dengan skenario yang dijalankan.

3.3.4.6. *Setting Routing* Protokol Pada Sisi *Customer*

Pada sisi *costumer* hingga pada sisi *router provider edge* yang terhubung dengan *customer edge* akan dihubungkan dengan *routing* protokol sesuai dengan skenario yang dibuat. Penomoran *routing* protokol sesuai dengan id *routing* protokol yang berada pada *interface provider edge* tetapi tanpa penambahan penamaan vrf. Karena vrf hanya dikonfigurasi pada *router provider edge*. Setelah selesai semua konfigurasi yang dijalankan maka langkah

selanjutnya adalah pengujian koneksi jaringan. Pengujian ini dapat dilakukan dengan perintah "PING" dengan alamat tujuan yang sudah ditentukan.

3.4 Standar Parameter

Ada beberapa standar yang digunakan sebagai rujukan dalam penelitian ini. Standar yang digunakan dikeluarkan oleh beberapa lembaga yang berwenang seperti ITU dan Cisco :

a. Untuk *video streaming*, *jitter* yang bagus bernilai kurang dari 75 ms (berdasarkan ITU-T G. 1010), tetapi dari Cisco tidak merekomendasikan angka yang pasti mengenai *jitter*. Namun dapat digunakan nilai acuan bagai berikut

Tabel 3.6 Penggolongan *Jitter*

Kategori	***Jitter***
Sangat Bagus	**0 ms**
Bagus	**0 s/d 75 ms**
Sedang	**76 s/d 125 ms**
Jelek	**125 s/d 225 ms**

b. Untuk *video streaming*, nilai *delay* yang baik bernilai antara 0 – 150 ms berdasarkan pada aplikasi (ITU-T standar) dan untuk Cisco menyarankan tidak lebih dari 4 detik.

c. Untuk *packet loss* dapat dilihat pada tabel berikut sesuai dengan rekomendasi Cisco.

Tabel 3.7 Penggolongan *Packet Loss*

Kategori	***Packet Loss***
Sangat Bagus	**0%**
Bagus	**3%**
Sedang	**15%**
Jelek	**25%**

3.5 Metodologi Pengambilan Data

Pengambilan data pada penelitian ini, akan terjadi pertukaran data antara *client* dan *server* pada saat berkomunikasi. Paket akan dikirim dari *server* menuju *client*. Pengambilan data akan dilakukan pada sisi *client* dengan menggunakan software Wireshark sebagai *network analyzer*. *Traffict Generator* akan ditempatkan di sisi *client* tapi berada pada *network* yang lain. Dengan kata lain akan terhubung dengan *router* yang sama tetapi berbeda *network* ipnya. *Traffict Generator* akan membanjiri *server* dengan paket data sehingga seolah – olah jaringan *server* sedang sibuk. Berikut cara pengambilan data yang dilakukan.

1. Pengambilan data dilakukan dengan memfilter paket data yang lewat pada aplikasi Wireshark. Untuk *video streaming* dan *transfer data* maka filternya yaitu ip sumber dan jenis paket data yang digunakan, Untuk filter yang digunakan pada saat menganalisa layanan *transfer data* adalah "(ip.src==192.168.2.10)&&!(ip.src==192.168.1.10)&&tcp" sedang untuk filter yang digunakan pada saat menganalisa layanan *video streaming* adalah "(ip.src==192.168.2.10)&&!(ip.src==192.168.1.10)&&udp"
2. Untuk *video streaming*, jenis paket data yang digunakan yaitu UDP, tetapi karena layanan *video streaming* berjalan di RTP maka pada paket UDP akan di *decode* (dirubah) menjadi RTP. Protokol UDP di sini hanya sebagai *transport* saja.
3. Untuk mengetahui nilai *throughput*, dapat dilihat pada bagian "*summary*" pada Wireshark.
4. Untuk melihat nilai *packet loss*, pada *video streaming* dapat dilihat pada bagian *stream analysis* pada bagian *telephony*, sedang pada *transfer data* dapat kita filter dengan perintah "tcp.ack"
5. Untuk memudahkan pengamatan, hasil pengamatan dari Wireshark dapat dirubah menjadi bentuk excel dengan cara menyimpan *file* hasil Wireshark dengan format ".csv" dan dapat dibuka pada Microsoft Excel.
6. Untuk melihat nilai *delay* dan *jitter* pada *video streaming* maka hasil dari *stream analysis* tadi dapat kita analisis dahulu dengan menggunakan Microsoft Excel.

7. Sedangkan pada bagian *transfer data*, dapat kita analisis juga dengan menggunakan Microsoft Excel dengan menambahkan beberapa rumus terlebih dahulu.

3.6 Skenario Pengujian dan Pengambilan Data

Skenario yang digunakan pada penelitian ini ada tiga macam, yakni

1. Skenario dengan menggunakan *routing* protokol OSPF
   Skenario OSPF adalah skenario atau topologi yang sistem *routing*nya menggunakan *routing* protokol OSPF dari *router* PE menuju *router customer* (C).
2. Skenario dengan menggunakan *routing* protokol EIGRP
   Skenario EIGRP adalah skenario topologi yang sistem *routing*-nya menggunakan *routing* protokol EIGRP dari *router* PE menuju *router customer* (C).
3. Skenario dengan menggunakan *routing* protokol RIPv2.
   Skenario RIPv2 adalah skenario topologi yang sistem *routing*-nya menggunakan *routing* protokol RIPv2 dari *router* PE menuju *router customer*(C).

Customer di sini merupakan *router* yang terhubung dengan perangkat akhir dalam hal penelitian ini berupa komputer yang berfungsi sebagai *client* atau *server*. Pada setiap skenario akan menjalankan dua buah layanan yaitu

1. Layanan *Video Streaming* dengan ditambahkan *Traffic Generator*.
   Pada layanan *video streaming* akan ditambahkan beban trafik sebesar 15 Mbit, 30 Mbit, dan 45 Mbit dengan pengambilan data pada asing asing beban trafik sebanyak 5 kali.
2. Layanan *Transfer Data* dengan ditambahkan *Traffic Generator*
   Pada layanan *transfer data* akan ditambahkan beban trafik sebesar 150 Kbit, 250 Kbit, dan 350 Kbit dengan pengambilan data pada asing asing beban trafik sebanyak 5 kali.

Tabel 3.8 Skenario Pengambilan Data

No	Skenario	Layanan	Beban Traffik	Parameter
1	MPLS VPN OSPF	*Video Streaming*	15 mb	***Througput, Delay, Jitter,* dan *Packet Loss*.**
			30 mb	
			45 mb	

		Transfer Data	150 kb	
			250 kb	
			350 kb	
2	MPLS VPN EIGRP	*Video Streaming*	15 mb	***Througput, Delay, Jitter,* dan *Packet Loss*.**
			30 mb	
			45 mb	
		Transfer Data	150 kb	
			250 kb	
			350 kb	
3	MPLS VPN RIPv2	*Video Streaming*	15 mb	***Througput, Delay, Jitter,* dan *Packet Loss*.**
			30 mb	
			45 mb	
		Transfer Data	150 kb	
			250 kb	
			350 kb	

Sehingga pada setiap skenario akan menjalankan layanan *video streaming* kemudian dilanjutkan dengan layanan *transfer data*. Dan pada masing – masing layanan akan ditambahkan beban trafik secara bertahap sehingga akan didapat 90 hasil data yang ditangkap oleh *Wireshark*. 45 hasil data *video streaming* dan 45 hasil data *transfer data*. Dari hasil data tersebut akan dianalisis untuk mencari parameter *throughput*, *delay*, *jitter*, dan *packet loss*.