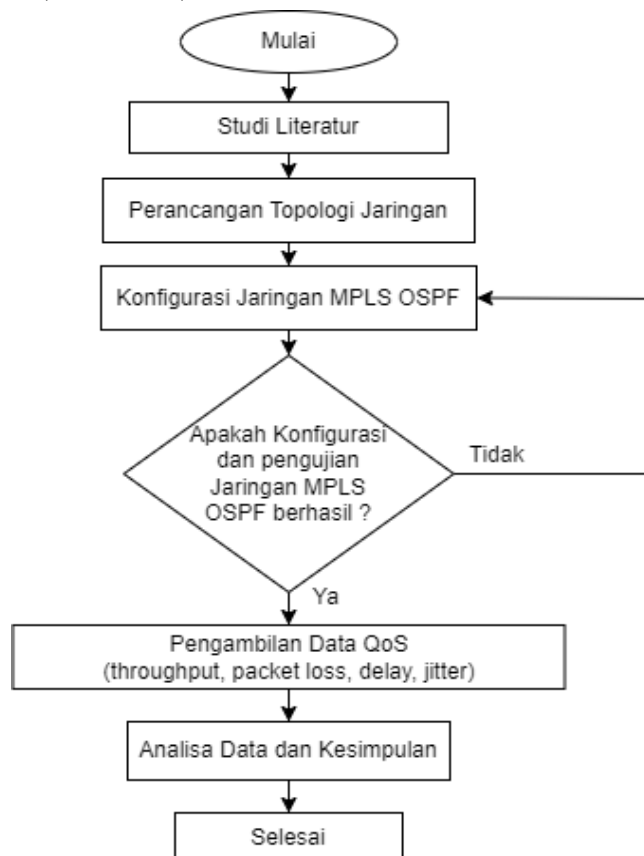


BAB III METODE PENELITIAN

3.1 PEMODELAN SISTEM

Penelitian ini menggunakan wireshark dalam menganalisis jaringan yang tersedia. Analisis dilakukan dengan menggunakan *tools Wireshark*, dan parameter yang dianalisis adalah *delay, jitter, throughput dan packet loss*. Pengolahan data dilakukan dengan cara membandingkan hasil dari pengukuran dengan menggunakan standar TIPHON (*Telecommunications and Internet Harmonization Over Network*). TIPHON merupakan standar penilaian parameter QoS yang dikeluarkan oleh badan standar ETSI (*European Telecommunications Standards Institute*). Kemudian dianalisis bagaimana kriteria jaringan tersebut dan diambil kesimpulan dari hasil parameter-parameter tersebut.

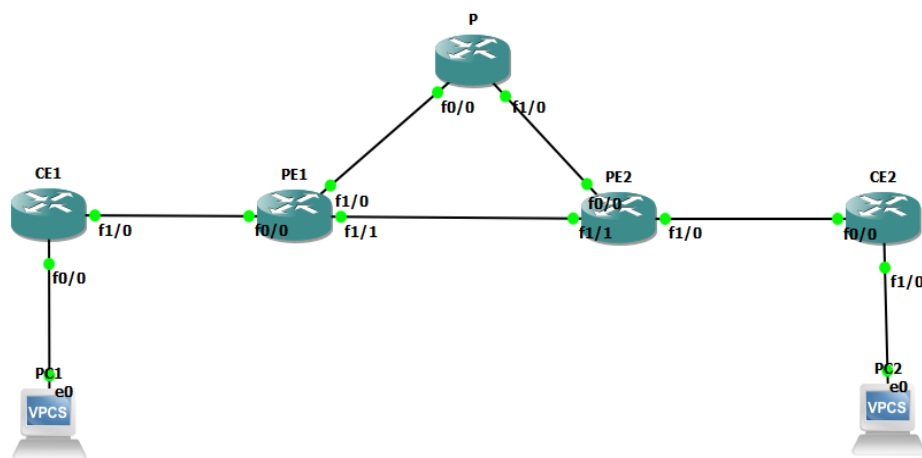
3.2 ALUR PENELITIAN



Gambar 3.1 Diagram Alur Penelitian

Gambar 3.1 menunjukkan tahapan dalam perencanaan dan simulasi penelitian yang akan dilakukan. Alur kerja dimulai dengan melakukan studi literatur yang mengangkat topik tentang perancangan jaringan ipv4 dan ipv6 serta penggunaan jaringan MPLS. Pada tahap ini juga dilakukan pengumpulan artikel penelitian selanjutnya yang telah dilakukan terkait dengan protokol *routing* OSPF. Apabila bahan keseluruhan telah terpenuhi selanjutnya melakukan perancangan topologi jaringan yang akan digunakan. Pada perancangan topologi jaringan meliputi kebutuhan *hardware* dan *software*. Setelah itu dilanjutkan dengan memulai tahap simulasi atau pengujian dengan OSPF MPLS pada IPv4 dan IPv6. Setelah tahap simulasi selesai selanjutnya proses pengambilan data dilakukan dengan menggunakan software wireshark dengan sesi waktu 15, 30 dan 60 detik, masing-masing sesi diulang selama lima kali. Apabila berhasil maka penelitian akan diteruskan ke tahapan selanjutnya, apabila gagal maka akan dilakukan simulasi ulang. Pengambilan data dilakukan setelah diuji berdasarkan skenario, maka didapat nilai QoS (*Quality of Service*) berupa nilai *delay*, *jitter*, *packet loss* dan *throughput*. Setelah itu akan dilakukan pengecekan ulang jika terjadi RTO (*Request Time Out*) atau *destination host unreachable* pada saat pengujian. Dan akan dilakukan analisa untuk menyimpulkan hasilnya.

3.3 PERANCANGAN SISTEM



Gambar 3.2 Desain Topologi Jaringan

3.3.1 Perancangan Desain Topologi Jaringan

Langkah awal dalam penelitian yang dilakukan adalah merancang desain topologi jaringan, adapun desain topologi jaringan dapat dilihat pada gambar 3.2. Topologi jaringan yang akan diuji coba dibuat menggunakan perangkat lunak simulasi GNS3. Pada desain topologi jaringan yang digunakan terdapat lima router yang saling terhubung dengan memiliki 1 *router provider* (P), 2 *router provider edge* (PE), 2 *router customer edge* (CE) dan dua buah PC yang berfungsi sebagai *client* dan *server*.

Tabel 3.1 Addressing Topologi Jaringan

Router	Interface	IP Address	Subnet Mask
PC1	e0	192.168.10.2	255.255.255.0
PC2	e0	192.168.20.2	255.255.255.0
CE1	Fa0/0	192.168.10.1	255.255.255.0
	Fa1/0	172.16.1.1	255.255.255.0
CE2	Fa0/0	172.16.1.2	255.255.255.0
	Fa1/0	192.168.20.1	255.255.255.0
PE1	Fa0/0	172.16.1.2	255.255.255.0
	Fa1/0	17.10.11.1	255.255.255.240
	Fa1/1	17.20.25.1	255.255.255.240
	Lo/0	12.12.1.1	255.255.255.0
PE2	Fa0/0	17.30.30.2	255.255.255.240
	Fa1/0	172.16.2.1	255.255.255.0
	Fa1/1	17.20.25.2	255.255.255.240
	Lo/0	12.12.3.1	255.255.255.0
P	Fa0/0	17.10.11.2	255.255.255.240
	Fa1/0	17.30.30.1	255.255.255.240
	Lo/0	12.12.2.1	255.255.255.0

Langkah selanjutnya adalah mengkonfigurasi alamat IP pada setiap *router*, alamat IP yang digunakan di setiap *router* dapat dilihat pada Tabel 3.1.

Setelahnya, dilakukan pengujian konektivitas dengan menggunakan *ping point-to-point* antar *router*. Berikut hasil konfigurasi IP *address*:

```
PE1(config)#int f0/0
PE1(config-if)#ip add 172.16.1.2 255.255.255.0
PE1(config-if)#no sh
```

3.3.2 Perancangan Routing *Open Shortest Path First*

Open Shortest Path First (OSPF) adalah protokol routing yang digunakan untuk IP dan termasuk dalam kategori protokol link-state. OSPF menggunakan algoritma Dijkstra untuk memilih jalur terpendek menuju setiap tujuan. Dalam konteks komunikasi perangkat pada jaringan MPLS, IP *loopback* digunakan pada masing-masing perangkat. Sebelum membuat routing OSPF, langkah awalnya adalah membuat *interface* pada *bridge* dengan nama *loopback*. Interface *loopback* adalah suatu *interface* yang sebenarnya tidak ada secara fisik dan diciptakan untuk mengatasi masalah pada jaringan. Keunggulan dari IP *loopback* adalah kestabilan koneksi, karena IP ini tidak terhubung ke suatu jaringan tertentu atau *interface* fisik, sehingga tidak akan mengalami *down* seperti halnya *interface* yang terhubung ke jaringan tertentu.

Berikut hasil konfigurasi IP *loopback*:

```
PE1(config-if)#int lo0
PE1(config-if)#ip add 12.12.1.1 255.255.255.0
PE1(config-if)#no sh
```

Berikut hasil konfigurasi protokol routing OSPF:

```
Berikut hasil konfigurasi protokol routing OSPF:
PE1(config)#router ospf 1
PE1(config-router)#net 12.12.1.1 0.0.0.255 area 0
PE1(config-router)#net 172.16.1.0 0.0.0.255 area 0
PE1(config-router)#net 17.10.11.0 0.0.0.15 area 0
PE1(config-router)#net 17.20.25.0 0.0.0.15 area 0
```

```
PE1(config-router)#do wr
Building configuration...
[OK]
```

3.3.3 Perancangan Jaringan Multi Protocol Label Switching

Multi Protocol Label Switching adalah teknologi pengiriman paket pada jaringan backbone (utama) dengan kecepatan tinggi. Prinsip dasar *label switching* adalah adanya penambahan *label* khusus dan unik pada setiap paket data dan *label* tersebut digunakan untuk mengarahkan paket melalui jaringan. Setelah melakukan konfigurasi *routing* OSPF, langkah berikutnya adalah membangun jaringan MPLS. Penting untuk memastikan bahwa *routing* ke IP *Loopback* pada setiap router MPLS telah terkonfigurasi dengan benar. Setelah semua IP *Loopback* terdaftar dalam *routing*, konfigurasi jaringan MPLS dimulai dengan menambahkan *LDP interface*. *LDP interface* digunakan untuk menentukan antarmuka dari router yang terhubung ke router lainnya dalam jaringan MPLS.

Berikut hasil konfigurasi MPLS:

```
PE1(config)#ip cef
PE1(config)#mpls ip
PE1(config)#mpls label protocol ldp
PE1(config)#mpls ldp router-id f0/0
PE1(config)#mpls ldp router-id f1/0
PE1(config)#mpls ldp router-id f1/1
PE1(config)#int f0/0
PE1(config-if)#mpls ip
PE1(config-if)#mpls label protocol ldp
PE1(config-if)#int f1/0
PE1(config-if)#mpls ip
PE1(config-if)#mpls label protocol ldp
PE1(config-if)#int f1/1
PE1(config-if)#mpls ip
```

```
PE1(config-if)#mpls label protocol ldp
PE1(config-if)#do wr
Building configuration...
[OK]
```

3.4 SKENARIO PENGUJIAN

Pengujian yang dilakukan pada penelitian ini menggunakan dua skenario, yakni: pengujian pada jaringan MPLS OSPF untuk IPV4 dan pengujian pada jaringan MPLS OSPF untuk IPV6. Proses pengambilan data dilakukan dengan menggunakan *software Wireshark* dan dengan sesi waktu 15 detik, 30 detik dan 60 detik, masing-masing sesi diulang sebanyak 5 kali. Pengujian dilakukan untuk memahami kinerja jaringan dari IPv4 dan IPv6 serta mengamati stabilitas kinerja melalui hasil rata-rata dari beberapa pengujian. Hasil pengujian kemudian dianalisis menggunakan beberapa parameter QoS untuk mengevaluasi kualitas jaringan. Beberapa standar dapat diajukan acuan untuk mengetahui bagus tidaknya nilai dalam parameter QoS. Standar kategori performansi jaringan yang digunakan sesuai dengan versi TIPHON (*Telecommunications and Internet Protocol Harmonization Over Networks*).

Tahapan yang dilakukan pada skenario pengujian untuk pengambilan data:

1. Pada konfigurasi jaringan yang sudah dibuat menggunakan *ipterm server* dan *client*, serta *iperf* yang digunakan untuk menaikkan trafik pada jaringan.
2. Gunakan GNS3 VM2. GNS3 *Virtual Machine* memungkinkan pengguna untuk membuat dan menguji topologi jaringan yang kompleks tanpa perlu perangkat keras fisik. GNS3 VM2 mengintegrasikan berbagai perangkat jaringan virtual dan fisik, seperti *router*, *switch* dan *firewall* dalam satu topologi, memungkinkan untuk mengembangkan dan menguji konfigurasi jaringan tanpa perlu mengganggu infrastruktur fisik yang ada. GNS3 VM2 diinstal sebagai bagian dari instalasi GNS3 pada komputer

tujuan, yang memberikan fleksibilitas dan kinerja yang lebih baik dan memastikan simulasi jaringan yang lebih lancar dan responsif.

```
GNS3 server version: 2.2.45
Release channel: 2.2
VM version: 0.15.0
Ubuntu version: focal
Qemu version: 4.2.1
Virtualization: virtualbox
kvm
KVM support available: True
Uptime: up 0 minutes

IP: 192.168.56.102 PORT: 80

To log in using SSH: ssh gns3@192.168.56.102
Password: gns3

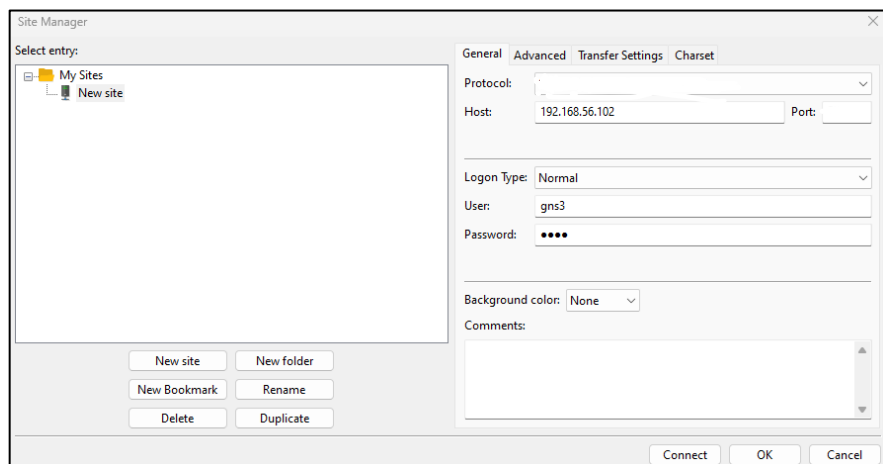
To launch the Web-Ui: http://192.168.56.102

Images and projects are stored in '/opt/gns3'
```

Gambar 3.3 Informasi pada GNS3 VM2

Pada gambar 3.3 menunjukkan informasi seperti IP, *port* dan *password* yang akan digunakan dalam konfigurasi selanjutnya.

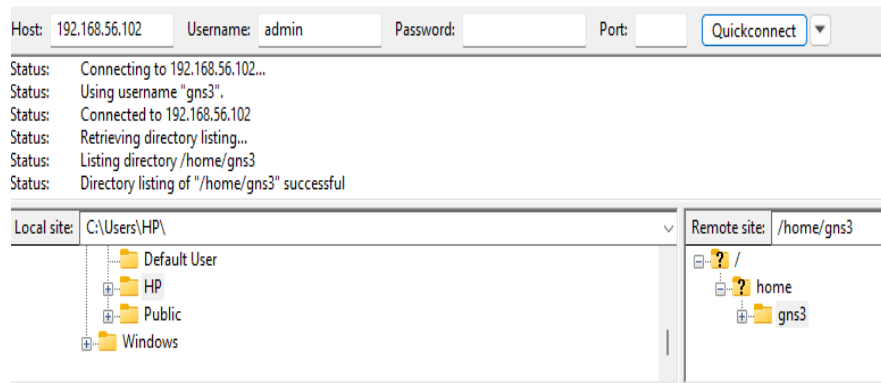
3. Selanjutnya, buka FileZilla. FileZilla dapat dijalankan pada berbagai platform yang memungkinkan untuk dengan mudah mengelola *transfer file* seperti mengunggah dan mengunduh *file*.



Gambar 3.4 Tampilan pada FileZilla

Gambar 3.4 adalah tampilan pada FileZilla untuk memasukkan informasi berupa *host*, *port*, *username* serta *password* yang digunakan.

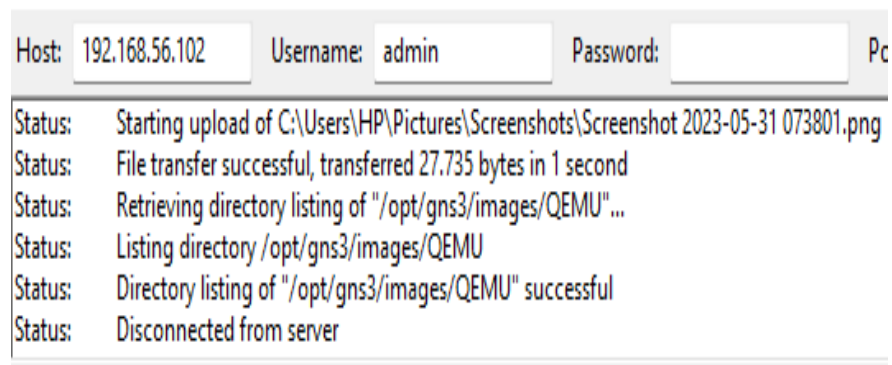
4. Tahap selanjutnya klik *connect*, dan status akan berubah menjadi *connected to 192.168.56.102*.



Gambar 3.5 Tampilan Status

Gambar 3.5 menunjukkan tampilan pada saat *host* sudah terhubung dan status berubah menjadi “*connected*”.

5. Selanjutnya, *upload file* yang akan dikirimkan dan status akan berubah menjadi “*file transfer successful*”.



Gambar 3.6 Tampilan status setelah upload file selesai

Gambar 3.6 berisi tampilan setelah *upload file* berhasil dilakukan dan status berubah menjadi “*file transfer successful*”. Data/file yang dikirimkan dengan ukuran 27.735 bytes (27,4 KB) dengan format png.

6. Setelah *transfer file* selesai dilakukan, pengambilan data akan dilakukan dengan menggunakan *software wireshark* sebagai *network analyzer*. *Wireshark* akan dipasangkan pada *interface f1/0 router CE2* yang mengarah pada *VPCS* sebagai *client*.

1. Mengamati besarnya *delay*

Ketika nilai *delay* yang dihasilkan semakin kecil dalam sebuah jaringan tersebut akan semakin baik, begitu juga sebaliknya, apabila nilai *delay* yang dihasilkan semakin besar maka kualitas jaringan tersebut akan semakin buruk. Semakin besar nilai *delay* yang dihasilkan akan menyebabkan semakin lama paket data akan di terima atau dengan kata lain kinerja jaringan tersebut akan menjadi lebih lambat.

2. Mengamati jumlah *packet loss*

Semakin kecil nilai *packet loss* yang dihasilkan dari jaringan internet maka kualitas jaringan tersebut akan semakin baik. Begitu juga sebaliknya, ketika nilai dari *packet loss* yang dihasilkan semakin besar, maka kualitas dari jaringan tersebut bisa dikatakan semakin buruk. Karena dengan semakin besar dari nilai *packet loss* maka paket data yang di terima akan semakin berkurang atau hilang.

3. Mengamati besarnya nilai *jitter*

Ketika nilai *jitter* yang dihasilkan dari monitoring menggunakan aplikasi *wireshark* dalam sebuah jaringan adalah kecil, maka kualitas jaringan tersebut akan semakin baik. Sebaliknya, ketika nilai *jitter* yang dihasilkan adalah besar maka kualitas jaringan tersebut akan semakin buruk.

4. Mengamati besarnya *throughput*

Semakin tinggi nilai *throughput* maka kualitas jaringan tersebut semakin baik karena dapat memaksimalkan penggunaan *bandwidth* pada jaringan sehingga lebih efisien. Sebaliknya semakin kecil nilai *throughput* yang dihasilkan maka kualitas jaringan tersebut akan semakin buruk.