

BAB III METODOLOGI PENELITIAN

3.1 Subjek dan Objek Penelitian

Subjek pada penelitian ini adalah Free Range Routing. Objek penelitian ini yaitu melakukan analisis performansi *routing protocol* OSPF dan EIGRP. Penelitian ini dilakukan menggunakan metode studi literatur dan simulasi langsung menggunakan *software* GNS3.

3.2 Alat dan Bahan Penelitian

Perangkat yang digunakan pada penelitian ini dibagi menjadi dua, yaitu *hardware* (perangkat keras) dan *software* (perangkat lunak).

3.2.1 Perangkat Keras (*hardware*)

Hardware yang digunakan yaitu seperti pada tabel 3.1 berikut ini :

Tabel 3.1 *Hardware* yang digunakan

No	Device	Jumlah	Fungsi
1	Laptop Lenovo (<i>AMD A9-9420 RADEON R5, 5 COMPUTE CORES 2C+3G, 3.00 GHz, (2 CPUs), 16384MB RAM, KINGSTON SSD SA400S37 480GB, ST1000LM035-1RK172 HDD 953869MB, Windows 11 Pro version 22H2 OS build 22621.2134</i>)	1	Melakukan simulasi dan analisis menggunakan GNS3

3.2.2 Perangkat Lunak (*software*)

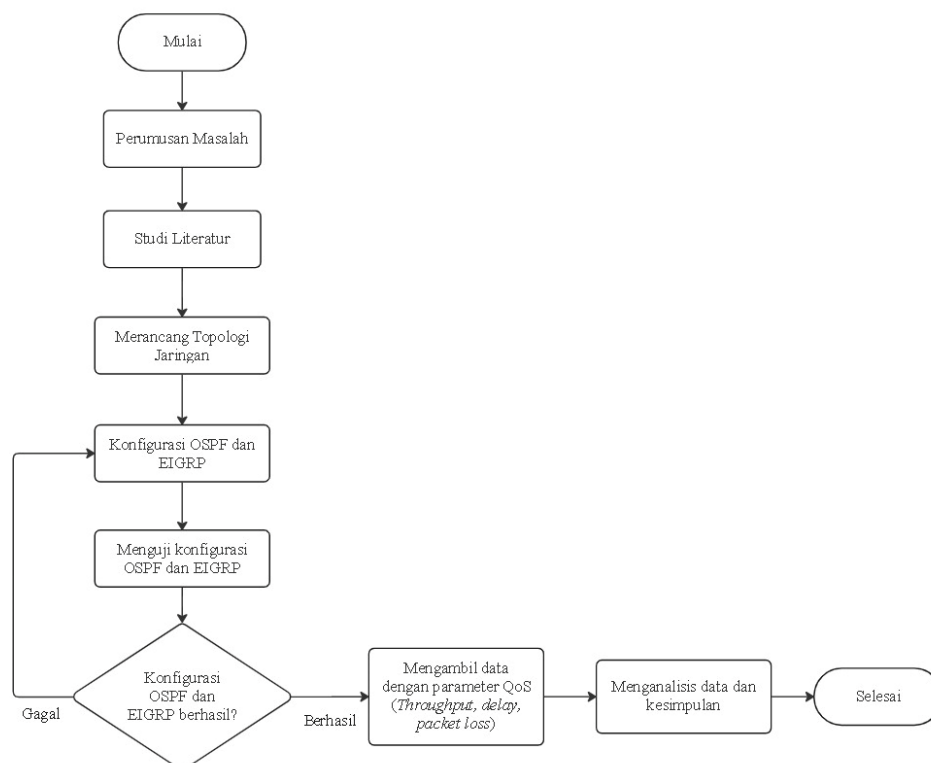
Software yang dibutuhkan yaitu seperti pada tabel 3.2 berikut ini :

Tabel 3.2 *Software* yang digunakan

No	Nama <i>software</i>	Versi	Fungsi
1	GNS3	2.2.36	Simulator pada jaringan yang dibuat
2	Wireshark	4.0.2	Melakukan <i>capture</i> paket jaringan
3	Alpine Linux	3.18	<i>Client</i> dan <i>Server</i>
4	Free Range Routing	8.2.2	OS jaringan untuk melakukan <i>routing protocol</i> yang digunakan

3.3 Diagram Alir/Proses Penelitian

Dalam penelitian ini, terdapat tahapan-tahapan sehingga hasil penelitian selesai sesuai harapan.

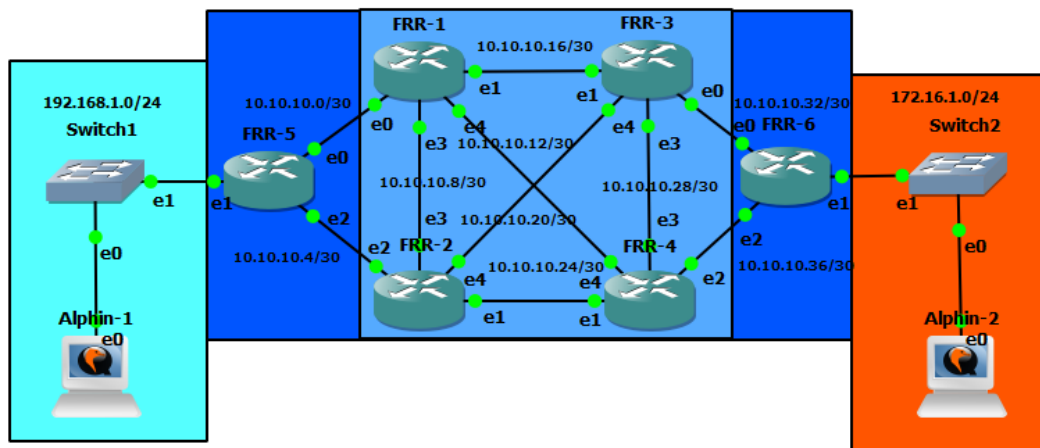


Gambar 3.1 Diagram Alir Penelitian

Pada gambar 3.1 adalah alur kerja dari tahapan pengerjaan penelitian secara keseluruhannya. Pada alur dari pengerjaan ini menggambarkan langkah-langkah dalam melakukan analisa performa *routing protocol* OSPF dan EIGRP. Tahap pertama dimulai dengan melakukan studi literature guna menambah pengetahuan mengenai penelitian terkait. Selanjutnya melakukan perancangan topologi yang akan disimulasikan. Setelah perancangan topologi sukses, langkah selanjutnya adalah melakukan simulasi dengan mengkonfigurasi Free Range Routing menggunakan *routing protocol* OSPF dan EIGRP. Setelah konfigurasi selesai dilakukan, selanjutnya adalah pengujian jaringan dengan melakukan ping dari komputer *client* ke *server*, jika pada proses pengujian gagal maka kembali melakukan proses konfigurasi *routing protocol* OSPF dan EIGRP. Setelah berhasil melakukan pengujian, selanjutnya melakukan pengambilan data dan analisis dari parameter QoS yaitu *throughput*, *delay* dan *packet loss*.

3.4 Skenario Penelitian

3.4.1 Rancangan Topologi Penelitian



Gambar 3.2 Topologi Jaringan

Gambar 3.2 diatas menjelaskan topologi jaringan dari penelitian yang akan dilakukan ini. Menggunakan enam FRR dan dua *Alpine Linux*, sebelumnya telah menggunakan *Tiny Core* akan tetapi pada *Tiny Core* konfigurasi yang telah dilakukan tidak dapat disimpan atau *save configuration* sehingga diganti dengan

menggunakan *Alpine Linux*. IP yang digunakan merupakan IPv4 sebagai *network address*. Dari gambar 3.2 diatas, topologi dari jaringan tersebut diberikan alamat IP yang telah disesuaikan dan disiapkan seperti tabel 3.3 dibawah ini :

Tabel 3.3 Tabel IP Address

Perangkat	Interface	Alamat IP
FRR-1	Loopback	10.0.0.1/32
	Ethernet 0	10.10.10.1/30
	Ethernet 1	10.10.10.5/30
	Ethernet 2	192.168.1.1/24
FRR-2	Loopback	10.0.0.2/32
	Ethernet 0	10.10.10.2/30
	Ethernet 1	10.10.10.17/30
	Ethernet 2	10.10.10.9/30
	Ethernet 3	10.10.10.13/30
FRR-3	Loopback	10.0.0.3/32
	Ethernet 0	10.10.10.6/30
	Ethernet 1	10.10.10.25/30
	Ethernet 2	10.10.10.10/30
	Ethernet 3	10.10.10.21/30
FRR-4	Loopback	10.0.0.4/32
	Ethernet 0	10.10.10.34/30
	Ethernet 1	10.10.10.17/30
	Ethernet 2	10.10.10.29/30
	Ethernet 3	10.10.10.14/30
FRR-5	Loopback	10.0.0.5/32
	Ethernet 0	10.10.10.38/30
	Ethernet 1	10.10.10.26/30
	Ethernet 2	10.10.10.30/30
	Ethernet 3	10.10.10.14/30

Perangkat	Interface	Alamat IP
FRR-6	Loopback	10.0.0.6/32
	Ethernet 0	10.10.10.33/30
	Ethernet 1	10.10.10.37/30
	Ethernet 2	172.16.1.1/24
PC1	Ethernet0	192.168.1.2/24
PC2	Ethernet0	172.16.1.2/24

3.4.2 Konfigurasi Simulasi Jaringan

Pada penelitian ini melakukan konfigurasi antara masing-masing router dengan *routing protocol* OSPF dan EIGRP sesuai dengan ketentuan topologi dan alamat IP diatas. Pastikan Free Range Routing telah disiapkan, selanjutnya masuk ke dalam router dan lakukan konfigurasi semua router sesuai alamat IP yang telah ditentukan seperti pada gambar 3.3, gambar 3.4 dan gambar 3.5 berikut ini.

```
frr# conf t
frr(config)# int lo
frr(config-if)# ip add 10.0.0.1/32
frr(config-if)# no sh
frr(config-if)# int eth1
frr(config-if)# ip add 192.168.1.1/24
frr(config-if)# no sh
frr(config-if)# int eth0
frr(config-if)# ip add 10.10.10.1/30
frr(config-if)# no sh
frr(config-if)# int eth2
frr(config-if)# ip add 10.10.10.5/30
frr(config-if)# no sh
frr(config-if)# ex
frr(config)# ex
```

Gambar 3.3 Konfigurasi Alamat IP

```

frr# conf t
frr(config)# router ospf
frr(config-router)# ospf router-id 10.0.0.2
frr(config-router)# network 10.0.0.2/32 area 0
frr(config-router)# network 10.10.10.0/30 area 1
frr(config-router)# network 10.10.10.8/30 area 0
frr(config-router)# network 10.10.10.16/30 area 0
frr(config-router)# network 10.10.10.12/30 area 0
frr(config-router)# ex
frr(config)# do show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, F - PBR,
       f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

C>* 10.0.0.1/32 [110/100] via 10.10.10.1, eth0, weight 1, 00:00:28
C  10.0.0.2/32 [110/0] is directly connected, lo, weight 1, 00:00:59
C>* 10.0.0.2/32 is directly connected, lo, 01:18:58
C  10.10.10.0/30 [110/100] is directly connected, eth0, weight 1, 00:00:37
C>* 10.10.10.0/30 is directly connected, eth0, 01:18:35
C>* 10.10.10.4/30 [110/200] via 10.10.10.1, eth0, weight 1, 00:00:28
C  10.10.10.8/30 [110/100] is directly connected, eth3, weight 1, 00:00:31
C>* 10.10.10.8/30 is directly connected, eth3, 01:17:26
C  10.10.10.12/30 [110/100] is directly connected, eth4, weight 1, 00:00:15
C>* 10.10.10.12/30 is directly connected, eth4, 01:16:53
C  10.10.10.16/30 [110/100] is directly connected, eth1, weight 1, 00:00:20
C>* 10.10.10.16/30 is directly connected, eth1, 01:17:56
C>* 192.168.1.0/24 [110/200] via 10.10.10.1, eth0, weight 1, 00:00:28
frr(config)#
frr(config)#

```

Gambar 3.4 Konfigurasi Routing OSPF

```

frr# conf t
frr(config)# router eigrp 10
frr(config-router)# network 10.0.0.1/32
frr(config-router)# network 10.10.10.0/30
frr(config-router)# network 10.10.10.4/30
frr(config-router)# network 192.168.1.0/24
frr(config-router)# ex
frr(config)# do sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, F - PBR,
       f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

E  10.0.0.1/32 [90/28160] is directly connected, lo, weight 1, 00:00:48
C>* 10.0.0.1/32 is directly connected, lo, 00:08:32
E  10.10.10.0/30 [90/28160] is directly connected, eth0, weight 1, 00:00:28
C>* 10.10.10.0/30 is directly connected, eth0, 00:08:32
E  10.10.10.4/30 [90/28160] is directly connected, eth2, weight 1, 00:00:23
C>* 10.10.10.4/30 is directly connected, eth2, 00:08:32
E  192.168.1.0/24 [90/28160] is directly connected, eth1, weight 1, 00:00:07
C>* 192.168.1.0/24 is directly connected, eth1, 00:08:32
frr(config)#

```

Gambar 3.5 Konfigurasi Routing EIGRP

Gambar 3.4 dan 3.5 menunjukkan konfigurasi antara *routing protocol* OSPF dan EIGRP, konfigurasi tersebut dapat diterapkan pada semua Free Range Routing sesuai topologi dan alamat IP yang telah ditentukan. Pada gambar 3.4 dapat dilihat pada perintah “*router ospf-id 10.0.0.2*” yang diartikan sebagai konfigurasi dengan protokol OSPF dan alamat IP tersebut menggunakan alamat IP *loopback* sebagai *process ID*, dan perintah *network* berfungsi sebagai alamat *network* yang sedang terhubung secara langsung dengan router yang sedang dikonfigurasi tersebut.

Gambar 3.5 menunjukkan perintah “*router eigrp 10*” yang dapat diartikan sebagai konfigurasi dengan protokol EIGRP dan angka *10* sebagai identitas dari router EIGRP itu sendiri supaya dapat saling terhubung. Sedangkan pada perintah *network* dimaksudkan untuk menghubungkan alamat *network* yang terhubung secara langsung dengan router tersebut.

3.4.3 Melakukan pengujian dan Pengambilan Data

Pada tahap penelitian ini akan melakukan pengujian pada *routing protocol* OSPF dan EIGRP apakah sudah saling terhubung satu sama lain dengan cara melakukan *test ping* dari komputer *client* ke *server* seperti pada gambar 3.6 berikut ini.

```
localhost:~# ping 172.16.1.2
PING 172.16.1.2 (172.16.1.2): 56 data bytes
64 bytes from 172.16.1.2: seq=0 ttl=60 time=159.224 ms
64 bytes from 172.16.1.2: seq=1 ttl=60 time=23.239 ms
64 bytes from 172.16.1.2: seq=2 ttl=60 time=11.542 ms
64 bytes from 172.16.1.2: seq=3 ttl=60 time=6.059 ms
64 bytes from 172.16.1.2: seq=4 ttl=60 time=9.316 ms
64 bytes from 172.16.1.2: seq=5 ttl=60 time=19.723 ms
64 bytes from 172.16.1.2: seq=6 ttl=60 time=10.310 ms
64 bytes from 172.16.1.2: seq=7 ttl=60 time=11.075 ms
64 bytes from 172.16.1.2: seq=8 ttl=60 time=12.932 ms
64 bytes from 172.16.1.2: seq=9 ttl=60 time=17.010 ms
64 bytes from 172.16.1.2: seq=10 ttl=60 time=8.112 ms
^C
--- 172.16.1.2 ping statistics ---
11 packets transmitted, 11 packets received, 0% packet loss
round-trip min/avg/max = 6.059/26.231/159.224 ms
localhost:~#
```

Gambar 3.6 Melakukan *Test Ping* dari *Client* ke *Server*

Pada gambar 3.6 terlihat bahwa pengujian *ping* dari *client* ke *server* telah berhasil dengan ditandai adanya *ttl*. Selanjutnya dilakukan pengambilan data dengan cara *client* mengirimkan paket data menuju ke *server*. Parameter yang dipergunakan yaitu *throughput*, *delay*, dan *packet loss*. Tabel 3.4 akan menunjukkan ukuran paket data yang dikirimkan dari *client* ke *server* untuk mengetahui nilai parameter yang akan diujikan, masing-masing ukuran paket data dilakukan sepuluh kali percobaan pengujian pada protokol TCP dan UDP.

Tabel 3.4 Ukuran Paket Data dan Parameter Pengujian

Ukuran Paket	Parameter QoS
10 MB	<i>Throughput, delay, dan packet loss</i>
20 MB	
30 MB	
40 MB	
50 MB	

```
localhost:~# iperf3 -s
-----
Server listening on 5201 (test #1)
-----
```

Gambar 3.7 Alpine Linux Sebagai Server

```
localhost:~# iperf3 -c 172.16.1.2 -n 10m -u
Connecting to host 172.16.1.2, port 5201
[ 5] local 192.168.1.2 port 44786 connected to 172.16.1.2 port 5201
[ ID] Interval      Transfer      Bitrate      Total Datagrams
[ 5] 0.00-1.06    sec    115 KBytes    885 Kbits/sec    81
[ 5] 1.06-2.68    sec    137 KBytes    692 Kbits/sec    97
[ 5] 2.68-3.05    sec    119 KBytes    2.66 Mbits/sec   84
[ 5] 3.05-4.00    sec    141 KBytes    1.22 Mbits/sec  100
[ 5] 4.00-5.00    sec    129 KBytes    1.05 Mbits/sec   91
[ 5] 5.00-6.00    sec    127 KBytes    1.04 Mbits/sec   90
[ 5] 6.00-7.30    sec    99.0 KBytes    624 Kbits/sec   70
[ 5] 7.30-8.01    sec    156 KBytes    1.79 Mbits/sec  110
[ 5] 8.01-9.02    sec    73.5 KBytes    599 Kbits/sec   52
[ 5] 9.02-10.90   sec    170 KBytes    740 Kbits/sec  120
[ 5] 10.90-11.20  sec    36.8 KBytes    992 Kbits/sec   26
```

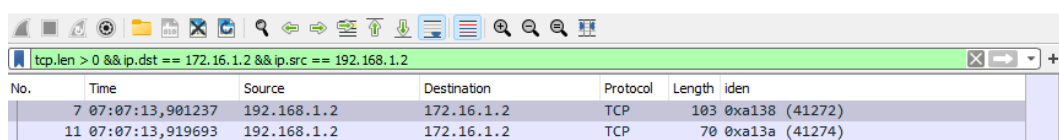
Gambar 3.8 Client Mengirimkan Data dengan Protokol UDP


```
localhost:~# iperf3 -c 172.16.1.2 -n 10m
Connecting to host 172.16.1.2, port 5201
[ 5] local 192.168.1.2 port 58088 connected to 172.16.1.2 port 5201
[ ID] Interval           Transfer     Bitrate      Retr   Cwnd
[ 5] 0.00-1.23 sec      195 KBytes  1.30 Mbits/sec    0    33.9 KBytes
[ 5] 1.23-2.09 sec      153 KBytes  1.44 Mbits/sec    0    69.3 KBytes
[ 5] 2.09-3.00 sec      154 KBytes  1.39 Mbits/sec    0    74.9 KBytes
[ 5] 3.00-4.03 sec      255 KBytes  2.04 Mbits/sec    0    86.3 KBytes
[ 5] 4.03-5.03 sec      255 KBytes  2.08 Mbits/sec    0    91.9 KBytes
[ 5] 5.03-6.02 sec      0.00 Bytes  0.00 bits/sec     0    91.9 KBytes
[ 5] 6.02-7.01 sec      255 KBytes  2.12 Mbits/sec    0   112 KBytes
```

Gambar 3.9 *Client* Mengirimkan Data dengan Protokol TCP

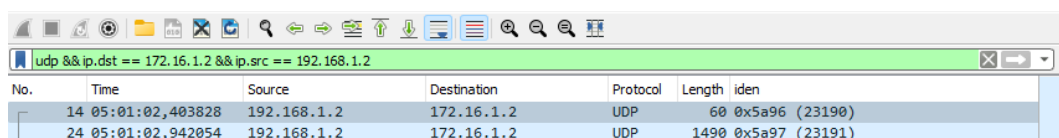
Gambar 3.8 dan 3.9 dilakukan pengujian dengan mengirimkan paket data dari komputer *client* ke *server* sesuai ukuran paket yang telah ditentukan dan pada gambar 3.7 sebagai *server*-nya. Pada gambar 3.8 melakukan pengiriman data dengan menggunakan protokol UDP dan pada gambar 3.9 melakukan pengiriman paket data dengan menggunakan protkol TCP.

Ketika proses pengiriman paket data sedang berlangsung dari komputer *client* ke *server* dilakukan penangkapan paket atau *packet capture* untuk dilakukan analisa. Penangkapan paket tersebut menggunakan *software wireshark* yang dilakukan pada komputer *client* dan komputer *server* untuk melihat proses pengiriman paket data tersebut. Ketika proses penangkapan paket data telah selesai selanjutnya melakukan filtering paket baik yang terjadi dikomputer *client* maupun dikomputer *server*. Hal ini dilakukan untuk mencari nilai-nilai parameter yang akan di analisa. Berikut gambar 3.10 dan gambar 3.11 pemfilteran paket data dikomputer *client* dan *server*.



No.	Time	Source	Destination	Protocol	Length	iden
7	07:07:13,901237	192.168.1.2	172.16.1.2	TCP	103	0xa138 (41272)
11	07:07:13,919693	192.168.1.2	172.16.1.2	TCP	70	0xa13a (41274)

Gambar 3.10 Pemfilteran Paket Data pada Protokol TCP



No.	Time	Source	Destination	Protocol	Length	iden
14	05:01:02,403828	192.168.1.2	172.16.1.2	UDP	60	0x5a96 (23190)
24	05:01:02,942054	192.168.1.2	172.16.1.2	UDP	1490	0x5a97 (23191)

Gambar 3.11 Pemfilteran Paket Data pada Protokol UDP

Pada gambar 3.10 dan gambar 3.11 terlihat perintah “*ip.dst==172.16.1.2&&ip.src==192.168.1.2*” yang artinya *ip.dst* sebagai alamat IP penerima atau *server* dan *ip.src* sebagai alamat pengirim atau *client*. Sedangkan untuk perintah *tcp* untuk melakukan filtering paket data dengan protokol TCP dan perintah *udp* untuk melakukan filtering paket data dengan protokol UDP.

3.5 Hipotesis Penelitian

Kualitas performa dari *routing protocol* pada saat melakukan pengiriman paket data dapat dipengaruhi oleh banyaknya transfer data yang terjadi secara bersamaan. (H0) Kualitas performa dari *routing protocol* dapat dikatakan buruk jika tidak memenuhi standar klasifikasi yang ada pada layanan pengujian. (H1) Kualitas performa dari *routing protocol* dapat dikatakan bagus jika memenuhi standar klasifikasi yang ada pada layanan pengujian.