

BAB 4

PEGUJIAN DAN ANALISIS

4.1. Skenario Pengujian

Skenario pengujian jaringan yang dibuat didasarkan pada *topologi* yang telah dirancang pada Bab 3. Sekanarionya adalah sebagai berikut :

1. Pengujian konfigurasi jaringan, pengujian ini dilakukan dengan mengeksekusi *script* XML yang sudah dibuat. Kemudian hasilnya dievaluasi apakah sesuai dengan yang diinginkan.
2. Pengujian konektifitas, pengujian ini dilakukan dengan menggunakan perintah *ping* dari satu perangkat ke perangkat lainnya.
3. Pengujian NARB dilakukan sebelum LSP dibentuk, antara ES1 ke semua perangkat, ES2 ke semua perangkat, ES3 ke semua perangkat, dan ES4 ke semua perangkat.
4. Pengujian pembentukan LSP, dari pengujian ini dapat diketahui kapabilitas jaringan dalam melayani pengesetan LSP.
5. Pengujian *database*, dari pengujian ini akandiketahui data apa saja yang terdapat di dalam *database*.
6. Pengujian skalabilitas, dilakukan dengan meningkatkan kompleksitas *topology* jaringan dalam hal penambahan perangkat, penambahan *route* dan penambahan jumlah *end user* pada VLSR sumber dan tujuan. Dari pengujian ini akan diketahui jumlah maksimum *router* pada jalur tersingkat, konsistensi pemilihan *path* yang dilewati dan jumlah maksimum *end system* yang dipasang.

4.2 Pengujian Konfigurasi

Konfigurasi dibuat dalam bentuk *script* dengan menggunakan XML, sebagaimana yang telah dilakukan pada Bab 3. *Script* konfigurasi tersebut dijalankan pada *Virtual Network Experiment* (VNE) yang telah dilengkapi dengan

tools konfigurasi DRAGON. Hasil eksekusi ketika menjalankan konfigurasi pada masing-masing perangkat adalah sebagai berikut :

- Konfigurasi narb (Host)

```
narb: /a/uml/debian/uml_linux mem=512m con1=pts umid=narb
ubda=/home/dragon2/uml/narb,/a/uml/debian/Debian-4.0-x86-root_fs
eth0=daemon,,unix,/tmp/tmpB5DRaq/ctl
kernel: /a/uml/debian/uml_linux
filesystem: /a/uml/debian/Debian-4.0-x86-root_fs (cow)
memorysize: 512
dragonconfig: /usr/local/dragon/bin/genDragonConfig.pl --start --hostname narb --
element='narb' --loopback='192.168.1.10' --narbdomain='192.168.0.0' --narbintra='gre1' --
gretunnel='gre1, 192.168.1.6, 192.168.1.10, 10.1.6.1, 10.1.6.2, 10.1.6.0/30'
eth0: ifconfig eth0 192.168.1.10 netmask 255.255.0.0 broadcast 192.168.255.255
```

Konfigurasi DRAGON memperlihatkan bahwa *host* NARB, memiliki *ip address loopback* 192.168.1.10 dengan *domain* NARBnya adalah 192.168.0.0. Sedangkan *GRE intra domain* adalah GRE 1 dengan alamat *interface* 10.1.6.1, 10.1.6.2, 10.1.6.0/30. NIC nya menggunakan eth0 dengan IP address 192.168.1.10. yang tersambung dengan IP address VLSR2 192.168.1.6.

- Konfigurasi red-es1 (Host)

```
red-es1: /a/uml/debian/uml_linux mem=512m con1=pts umid=red-es1
ubda=/home/dragon2/uml/red-es1,/a/uml/debian/Debian-4.0-x86-root_fs
eth1=daemon,,unix,/tmp/tmp2Na3p6/ctl eth0=daemon,,unix,/tmp/tmpqDf7K_/ctl
kernel: /a/uml/debian/uml_linux
filesystem: /a/uml/debian/Debian-4.0-x86-root_fs (cow)
memorysize: 512
dragonconfig: /usr/local/dragon/bin/genDragonConfig.pl --start --hostname red-es1 --
element='p2p-csa' --loopback='192.168.1.2' --narb='192.168.1.10' --gretunnel='gre2,
192.168.1.4, 192.168.1.2, 10.1.1.2, 10.1.1.1, 10.1.1.0/30' --telink='gre2, 11.1.1.1, eth1'
eth0: ifconfig eth0 192.168.1.2 netmask 255.255.0.0 broadcast 192.168.255.255
eth1: ifconfig eth1 up
```

Konfigurasi DRAGON memperlihatkan *hostname* red-es1 , dengan *element p2p-csa* , alamat *loopback* 192.168.1.2, terhubung dengan perangkat NARB yang mempunyai ip address: 192.168.1.10, *gretunnel* adalah GRE2 dengan alamat 192.168.1.4, 192.168.1.2, 10.1.1.2, 10.1.1.1, 10.1.1.0/30. *Telink gre2* dengan alamat 11.1.1.1 yang terhubung pada eth1, Sedangkan konektor Ethernet yang digunakan adalah sebagai berikut :

- eth0: `ifconfig eth0 192.168.1.2 netmask 255.255.0.0 broadcast 192.168.255.255`
- eth1: `ifconfig eth1 up`
- Konfigurasi ES-2 (*Host*)

```
-----
red-es2: /a/uml/debian/uml_linux mem=512m con1=pts umid=ES-2
ubda=/home/dragon2/uml/red-es2,/a/uml/debian/Debian-4.0-x86-root_fs
eth1=daemon,,unix,/tmp/tmpnlpAAZ/ctl eth0=daemon,,unix,/tmp/tmpnbHMEb/ctl
kernel: /a/uml/debian/uml_linux
filesystem: /a/uml/debian/Debian-4.0-x86-root_fs (cow)
memorysize: 512
dragonconfig: /usr/local/dragon/bin/genDragonConfig.pl --start --hostname red-es2 --
element='p2p-csa' --loopback='192.168.1.9' --narb='192.168.1.10' --gretunnel='gre4,
192.168.1.8, 192.168.1.9, 10.1.5.1, 10.1.5.2, 10.1.5.0/30' --telink='gre4, 11.1.5.2, eth1'
eth0: ifconfig eth0 192.168.1.9 netmask 255.255.0.0 broadcast 192.168.255.255
eth1: ifconfig eth1 up
-----
```

Konfigurasi DRAGON untuk perangkat dengan *hostname* ES-2, memiliki element *p2p-csa*, alamat *loopback* 192.168.1.9 terhubung dengan NARB dengan alamat 192.168.1.10. GRE *tunnel* bernama GRE4 dengan alamat 192.168.1.8, 192.168.1.9, 10.1.5.1, 10.1.5.2, 10.1.5.0/30, telink GRE 4 dengan alamat 11.1.5.2 terhubung pada *socket ethernet* eth1. Konektor *Ethernet* yang digunakan adalah :

- eth0: `ifconfig eth0 192.168.1.9 netmask 255.255.0.0 broadcast 192.168.255.255`
- eth1: `ifconfig eth1 up`
- Konfigurasi ES-4 (*Host*)

```
-----
ES-4: /a/uml/debian/uml_linux mem=512m con1=pts umid=ES-4
ubda=/home/dragon2/uml/red-es4,/a/uml/debian/Debian-4.0-x86-root_fs eth1=daemon,,unix
eth0=daemon,,unix,/tmp/tmp4ZtWiH/ctl kernel: /a/uml/debian/uml_linuxfilesystem:
/a/uml/debian/Debian-4.0-x86-root_fs (cow)memorysize: 512
dragonconfig: /usr/local/dragon/bin/genDragonConfig.pl --start --hostname ES-4 --
element='p2p-csa' --loopback='192.168.1.13' --narb='192.168.1.10' --gretunnel='gre10,
192.168.1.3, 192.168.1.13, 10.1.10.2, 10.1.10.1, 10.1.10.0/30' --telink='gre10, 11.1.10.1,
eth1'
- eth0: ifconfig eth0 192.168.1.13 netmask 255.255.0.0 broadcast 192.168.255.255
- eth1: ifconfig eth1 up
-----
```

Konfigurasi ES-4 memiliki alamat *loop back* IP 192.168.1.13 terhubung dengan VLSR 4 dengan alamat IP 192.168.1.3, GRE tunnel 10.1.10.0/30 dan TE link pada GRE 10 dengan alamat 11.1.10.1

- Konfigurasi switch1 (Switch)

```

red-switch1: /a/uml/debian/uml_linux mem=512m con1=pts umid=switch1
ubda=/home/dragon2/uml/red-switch1,/a/uml/debian/Debian-4.0-x86-root_fs
eth8=daemon,,unix,/tmp/tmp4ZtWiH/ctl eth7=daemon,,unix,/tmp/tmpmT9keN/ctl
eth6=daemon,,unix,/tmp/tmpsf_Bb4/ctl eth5=daemon,,unix,/tmp/tmpuHIM31/ctl
eth4=daemon,,unix,/tmp/tmpYEOPfz/ctl eth3=daemon,,unix,/tmp/tmpoiXT7v/ctl
eth2=daemon,,unix,/tmp/tmpme9u1t/ctl eth1=daemon,,unix,/tmp/tmpLSwKUe/ctl
eth0=daemon,,unix,/tmp/tmpIRuh3G/ctl
kernel: /a/uml/debian/uml_linux
filesystem: /a/uml/debian/Debian-4.0-x86-root_fs (cow)
memorysize: 512
eth0: ifconfig eth0 up
eth1: ifconfig eth1 up
eth2: ifconfig eth2 up
eth3: ifconfig eth3 up
eth4: ifconfig eth4 up
eth5: ifconfig eth5 up
eth6: ifconfig eth6 up
eth7: ifconfig eth7 up
eth8: ifconfig eth8 up
br0: brctl addbr br0; brctl addif br0 eth0; brctl addif br0 eth1; brctl addif br0 eth2; brctl addif
br0 eth3; brctl addif br0 eth4; brctl addif br0 eth5; brctl addif br0 eth6; brctl addif br0 eth7;
brctl addif br0 eth8; ifconfig br0 up

```

Switch dikonfigurasi sesuai dengan jumlah perangkat yang akan dihubungkan, jumlah terminal ethernet pada konfigurasi yang dibuat adalah 9 dimulai dari eth0 sampai dengan eth8. melalui hubungan ini VLSR memberikan perintah penyambungan sesuai informasi yang diperoleh dari NARB.

- Konfigurasi vlsr1 (Host)

```

red-vlsr1: /a/uml/debian/uml_linux mem=512m con1=pts umid=vlsr1
ubda=/home/dragon2/uml/red-vlsr1,/a/uml/debian/Debian-4.0-x86-root_fs
eth4=daemon,,unix,/tmp/tmpUeeZBD/ctl eth3=daemon,,unix,/tmp/tmpqrYcl/ctl
eth2=daemon,,unix,/tmp/tmp21WtQ4/ctl eth1=daemon,,unix,/tmp/tmpnP27Wv/ctl
eth0=daemon,,unix,/tmp/tmpme9u1t/ctl
kernel: /a/uml/debian/uml_linux filesystem: /a/uml/debian/Debian-4.0-x86-root_fs (cow)
memorysize: 512
dragonconfig: /usr/local/dragon/bin/genDragonConfig.pl --start --hostname vlsr1 --
element='vlsr' --loopback='192.168.1.4' --narb='192.168.1.10' --gretunnel='gre2,
192.168.1.2, 192.168.1.4, 10.1.1.1, 10.1.1.2, 10.1.1.0/30, gre8, 192.168.1.3, 192.168.1.4,
10.1.8.1, 10.1.8.2, 10.1.8.0/30, gre3, 192.168.1.6, 192.168.1.4, 10.1.2.2, 10.1.2.1,
10.1.2.0/30, gre6, 192.168.1.8, 192.168.1.4, 10.1.3.2, 10.1.3.1, 10.1.3.0/30' --telink='gre2,
11.1.1.2, eth1, gre8, 11.1.8.2, eth4, gre3, 11.1.2.1, eth2, gre6, 11.1.3.1, eth3'
eth0: ifconfig eth0 192.168.1.4 netmask 255.255.0.0 broadcast 192.168.255.255
eth1: ifconfig eth1 up
eth2: ifconfig eth2 up
eth3: ifconfig eth3 up

```

eth4: ifconfig eth4 up

Konfigurasi DRAGON yang terbentuk pada VLSR1 , IP address loopback 192.168.1.4 terhubung dengan nrb 192.168.1.10, gretunnel yang terhubung ada tiga yaitu :

- gre2, 192.168.1.2, 192.168.1.4, 10.1.1.1, 10.1.1.2, 10.1.1.0/30,
- gre3, 192.168.1.6, 192.168.1.4, 10.1.2.2, 10.1.2.1, 10.1.2.0/30,
- gre6, 192.168.1.8, 192.168.1.4, 10.1.3.2, 10.1.3.1, 10.1.3.0/30'

memiliki *TElink* pada GRE2 dengan alamat 11.1.1.2 melewati interface eth1, gre3, 11.1.2.1, eth2, gre6, 11.1.3.1, eth3. Konektor Ethernet yang digunakan :

- eth0: ifconfig eth0 192.168.1.4 netmask 255.255.0.0 broadcast 192.168.255.255
- eth1: ifconfig eth1 up
- eth2: ifconfig eth2 up
- eth3: ifconfig eth3 up

- Konfigurasi vlsr2 (Host)

```
red-vlsr2: /a/uml/debian/uml_linux mem=512m con1=pts umid=vlsr2
ubda=/home/dragon2/uml/red-vlsr2./a/uml/debian/Debian-4.0-x86-root_fs
eth2=daemon,,unix,/tmp/tmpVYOSpx/ctl eth1=daemon,,unix,/tmp/tmpzdrF9t/ctl
eth0=daemon,,unix,/tmp/tmpERJx0y/ctl
kernel: /a/uml/debian/uml_linux
filesystem: /a/uml/debian/Debian-4.0-x86-root_fs (cow)
memorysize: 512
dragonconfig: /usr/local/dragon/bin/genDragonConfig.pl --start --hostname vlsr2 --
element='vlsr' --loopback='192.168.1.6' --nrb='192.168.1.10' --narbintra='gre1' --
gretunnel='gre1, 192.168.1.10, 192.168.1.6, 10.1.6.2, 10.1.6.1, 10.1.6.0/30, gre3,
192.168.1.4, 192.168.1.6, 10.1.2.1, 10.1.2.2, 10.1.2.0/30, gre5, 192.168.1.8, 192.168.1.6,
10.1.4.2, 10.1.4.1, 10.1.4.0/30' --telink='gre3, 11.1.2.2, eth1, gre5, 11.1.4.1, eth2'
eth0: ifconfig eth0 192.168.1.6 netmask 255.255.0.0 broadcast 192.168.255.255
eth1: ifconfig eth1 up
eth2: ifconfig eth2 up
```

Konfigurasi DRAGON untuk perangkat dengan *hostname* red-vlsr2 memiliki IP address loopback 192.168.1.6 terhubung dengan *IP address* NARB 192.168.1.10, GRE *tunnel* nya adalah :

- GRE 1 antara 192.168.1.10 dengan 192.168.1.6 : 10.1.6.1, 10.1.6.2, 10.1.6.1, 10.1.6.0/30

- GRE 3 antara 192.168.1.4 dengan 192.168.1.6 : 10.1.2.1, 10.1.2.2, 10.1.2.0/30
- GRE 5 antar 192.168.1.8 dengan 192.168.1.6 : 10.1.4.2, 10.1.4.1, 10.1.4.0/30'

Sedangkan *Telink*-nya adalah gre3, 11.1.2.2, eth1, gre 5, 11.1.4.1, eth2, menggunakan interface konektor ethernet, eth0, eth1, eth2

- **Konfigurasi vlsr3 (Host)**

```
-----
red-vlsr3: /a/uml/debian/uml_linux mem=512m con1=pts umid=red-vlsr3
ubda=/home/dragon2/uml/red-vlsr3,/a/uml/debian/Debian-4.0-x86-root_fs
eth4=daemon,,unix,/tmp/tmpjk5HYC/ctl eth3=daemon,,unix,/tmp/tmpqrvYcl/ctl
eth2=daemon,,unix,/tmp/tmp8RSNBJ/ctl eth1=daemon,,unix,/tmp/tmp1_jS0Z/ctl
eth0=daemon,,unix,/tmp/tmpLSwKUe/ctl
kernel: /a/uml/debian/uml_linux
filesystem: /a/uml/debian/Debian-4.0-x86-root_fs (cow) memorysize: 512
dragonconfig: /usr/local/dragon/bin/genDragonConfig.pl --start --hostname vlsr3 --
element='vlsr' --loopback='192.168.1.8' --narb='192.168.1.10' --gretunnel='gre9,
192.168.1.3, 192.168.1.8, 10.1.9.1, 10.1.9.2, 10.1.9.0/30, gre4, 192.168.1.9, 192.168.1.8,
10.1.5.2, 10.1.5.1, 10.1.5.0/30, gre5, 192.168.1.6, 192.168.1.8, 10.1.4.1, 10.1.4.2,
10.1.4.0/30, gre6, 192.168.1.4, 192.168.1.8, 10.1.3.1, 10.1.3.2, 10.1.3.0/30' --telink='gre9,
11.1.9.2, eth4, gre4, 11.1.5.1, eth1, gre5, 11.1.4.2, eth2, gre6, 11.1.3.2, eth3'
eth0: ifconfig eth0 192.168.1.8 netmask 255.255.0.0 broadcast 192.168.255.255
eth1: ifconfig eth1 up
eth2: ifconfig eth2 up
eth3: ifconfig eth3 up
eth4: ifconfig eth4 up
-----
```

Konfigurasi DRAGON untuk perangkat dengan *hostname* VLSR3, memiliki alamat *loopback* 192.168.1.8 terhubung dengan NARB dengan alamat 192.168.1.10, memiliki *gretunnel* :

- gre4, 192.168.1.9, 192.168.1.8, 10.1.5.2, 10.1.5.1, 10.1.5.0/30,
- gre5, 192.168.1.6, 192.168.1.8, 10.1.4.1, 10.1.4.2, 10.1.4.0/30,
- gre6, 192.168.1.4, 192.168.1.8, 10.1.3.1, 10.1.3.2, 10.1.3.0/30

Sedangkan TE *link* pada GRE 4 dengan alamat 11.1.5.1 terhubung dengan interface eth1. Selanjutnya GRE 5 dengan alamat 11.1.4.2, eth2, dan GRE 6, 11.1.3.2, eth3. Sedangkan Konektor Ethernet yang digunakan :

- eth0: ifconfig eth0 192.168.1.8 netmask 255.255.0.0
broadcast 192.168.255.255

- eth1: ifconfig eth1 up
- eth2: ifconfig eth2 up
- eth3: ifconfig eth3 up

- Konfigurasi vlsr4 (Host)

```
-----
red-vlsr4: /a/uml/debian/uml_linux mem=512m con1=pts umid=vlsr4
ubda=/home/dragon2/uml/red-vlsr4,/a/uml/debian/Debian-4.0-x86-root_fs
eth3=daemon,,unix eth2=daemon,,unix,/tmp/tmpjk5HYC/ctl
eth1=daemon,,unix,/tmp/tmpUeeZBD/ctl eth0=daemon,,unix,/tmp/tmpmT9keN/ctl
kernel: /a/uml/debian/uml_linux
filesystem: /a/uml/debian/Debian-4.0-x86-root_fs (cow)
memorysize: 512
dragonconfig: /usr/local/dragon/bin/genDragonConfig.pl --start --hostname vlsr4 --
element='vlsr' --loopback='192.168.1.3' --narb='192.168.1.10' --gretunnel='gre8,
192.168.1.4, 192.168.1.3, 10.1.8.2, 10.1.8.1, 10.1.8.0/30, gre9, 192.168.1.8, 192.168.1.3,
10.1.9.2, 10.1.9.1, 10.1.9.0/30, gre10, 192.168.1.13, 192.168.1.3, 10.1.10.1, 10.1.10.2,
10.1.10.0/30' --telink='gre8, 11.1.8.1, eth1, gre9, 11.1.9.1, eth2, gre10, 11.1.10.2, eth3'
eth0: ifconfig eth0 192.168.1.3 netmask 255.255.0.0 broadcast 192.168.255.255
eth1: ifconfig eth1 up
eth2: ifconfig eth2 up
eth3: ifconfig eth3 up
-----
```

Konfigurasi DRAGON untuk host VLSR4 mempunyai IP *address loopback* 192.168.1.3 terhubung dengan NARB dengan IP address 192.168.1.10, mempunyai *GRE TUNNEL*, GRE 8, GRE 9, GRE 10.

Dari hasil eksekusi *script* konfigurasi yang telah dibuat dengan menggunakan XML, dapat dilihat bahwa jaringan yang terbentuk sesuai dengan hasil rancangan yang telah dibuat seperti Gambar 3.1. pada Bab 3.

4.3 Pengujian Konektifitas

Setelah konfigurasi berhasil, berikutnya dilakukan pengujian konektifitas antar perangkat yang telah dikonfigurasi. Pengujian ini dilakukan untuk membuktikan bahwa semua perangkat sudah terkoneksi. Sebelum dilakukan pengujian, semua perangkat diaktifkan terlebih dahulu dengan melakukan perintah *start all*, kemudian dilanjutkan dengan perintah *xterm all*. Setelah perintah tersebut dilakukan akan muncul terminal masing-masing perangkat. Pada masing-masing terminal dapat dilakukan perintah *ping* dari *host* tersebut ke *host* lainnya. Gambar 4.1 menunjukkan bahwa ping dilakukan pada ES1 yang memiliki

alamat IP 192.168.1.2. Ping dilakukan ke 4 alamat tujuan, yaitu 192.168.1.6, 192.168.1.8, 192.168.1.9 dan 192.168.1.10.

```

red-es1:~# ping 192.168.1.6
PING 192.168.1.6 (192.168.1.6) 56(84) bytes of data.
64 bytes from 192.168.1.6: icmp_seq=1 ttl=64 time=13.1 ms
64 bytes from 192.168.1.6: icmp_seq=2 ttl=64 time=0.756 ms
64 bytes from 192.168.1.6: icmp_seq=3 ttl=64 time=0.702 ms
64 bytes from 192.168.1.6: icmp_seq=4 ttl=64 time=0.841 ms
64 bytes from 192.168.1.6: icmp_seq=5 ttl=64 time=0.824 ms

[3]+ Stopped ping 192.168.1.6
red-es1:~# ping 192.168.1.8
PING 192.168.1.8 (192.168.1.8) 56(84) bytes of data.
64 bytes from 192.168.1.8: icmp_seq=1 ttl=64 time=14.1 ms
64 bytes from 192.168.1.8: icmp_seq=2 ttl=64 time=1.03 ms
64 bytes from 192.168.1.8: icmp_seq=3 ttl=64 time=0.724 ms
64 bytes from 192.168.1.8: icmp_seq=4 ttl=64 time=0.926 ms
64 bytes from 192.168.1.8: icmp_seq=5 ttl=64 time=0.992 ms
64 bytes from 192.168.1.8: icmp_seq=6 ttl=64 time=0.996 ms
64 bytes from 192.168.1.8: icmp_seq=7 ttl=64 time=0.986 ms

[4]+ Stopped ping 192.168.1.8
red-es1:~# ping 192.168.1.9
PING 192.168.1.9 (192.168.1.9) 56(84) bytes of data.
64 bytes from 192.168.1.9: icmp_seq=1 ttl=64 time=17.0 ms
64 bytes from 192.168.1.9: icmp_seq=2 ttl=64 time=0.723 ms
64 bytes from 192.168.1.9: icmp_seq=3 ttl=64 time=0.847 ms
64 bytes from 192.168.1.9: icmp_seq=4 ttl=64 time=0.785 ms
64 bytes from 192.168.1.9: icmp_seq=5 ttl=64 time=0.826 ms

[5]+ Stopped ping 192.168.1.9
red-es1:~# ping 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=64 time=13.7 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=64 time=0.746 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=64 time=0.682 ms
64 bytes from 192.168.1.10: icmp_seq=4 ttl=64 time=0.728 ms
64 bytes from 192.168.1.10: icmp_seq=5 ttl=64 time=0.718 ms

```

Gambar 4.1 Uji Konektifitas Jaringan GMPLS

Gambar 4.1 menunjukkan bahwa *node* asal yaitu 192.168.1.2 melakukan ping ke *node* tujuan dengan mengirimkan 56 kali 84 byte data. Bila alamat tujuan ditemukan, alamat tersebut membalas dengan mengirimkan 64 bytes data secara terus menerus dengan menggunakan protokol *Internet Control Message Protocol* (ICMP) yang dilengkapi nomor urut, *time to live* (ttl) dan waktu pengiriman. Untuk data selengkapnya dari hasil pengujian konektifitas ini dapat dilihat pada Tabel 4.1.

Tabel 4.1 memperlihatkan bahwa alamat masing-masing *device* dan alamat *interface* GRE terhubung, Pengujian dilakukan pada *host* ES1 dengan alamat 192.168.1.2 ke alamat lainnya yaitu ES2 (192.168.1.9), VLSR1 (192.168.1.4), VLSR2 (192.168.1.6), VLSR3 (192.168.1.9) dan NARB (192.168.1.10). Setelah dilakukan *ping* maka *host* tujuan melakukan *reply*. Dengan adanya respon ini berarti semua perangkat sudah terhubung. Selain itu ping juga dilakukan terhadap IP yang ada pada *interface GRE tunnel*, yaitu 10.1.1.1 dan 10.1.1.2. Semua IP tersebut juga merespon dengan *reply*. Hal yang sama juga terjadi pada pengujian di masing-masing *device*.

Sedangkan ping terhadap ERO yang diberi alamat TE *link*, yaitu 11.1.1.1 dan 11.1.1.2 ping memberikan *respon un reachable*, Ini menunjukkan alamat tersebut pengaturannya diluar aturan pengalamatan IP, alamat tersebut hanya diberikan NARB ketika memberikan respon terhadap pembuatan *tunnel*.

Tabel 4.1 Hasil *Test* Konektifitas antara ES1 dan perangkat lainnya

NO	Alamat Asal	Alamat Tujuan	Hasil Pengujian	Keterangan
1	192.168.1.2 (ES1)	192.168.1.4 (VLSR 1)	<i>reply</i>	
2		10.1.1.1 (GRE2)	<i>reply</i>	
3		10.1.1.2 (GRE2)	<i>replay</i>	
4		11.1.1.1 (TE LINK)	<i>unreachable</i>	
5		11.1.1.2 (TE LINK)	<i>unreachable</i>	
6		192.168.1.6 (VLSR2)	<i>reply</i>	
7		192.168.1.8 (VLSR3)	<i>reply</i>	
8		192.168.1.9 (ES22)	<i>reply</i>	
9		192.168.1.10 (NARB)	<i>reply</i>	
10		192.168.1.11(ES3)	<i>reply</i>	

4.5 Pengujian NARB

Pengujian NARB dilakukan untuk mengetahui TE *link* yang tersedia antara 2 *node*, pada penelitian ini dilakukan pengujian antara masing-masing titik, sehingga diketahui *path* antar *node* tersebut. Hasilnya dalam bentuk alamat GRE *tunnel* antara *node* sumber dan *node* tujuan pada *path* tersingkat.

4.5.1 Pengujian NARB antara *End System 1* (ES1) dan *End System 3* (ES3)

Pengujian NARB pertama dilakukan antara ES1 yang memiliki alamat IP 192.168.1.2 dengan *node* tujuan ES4 yang memiliki alamat 192.168.1.13. Gambar 4.2 memperlihatkan hasil pengujian NARB tersebut, dimana -V menunjukkan tag VLAN yang tersedia yaitu ada 100 Vtag yang dibawa oleh NARB, sedangkan -a memperlihatkan *vlan tag* yang digunakan, yakni 4Vtag.

Gambar 4.2 memperlihatkan bahwa *path* yang terbentuk merupakan hasil perhitungan RCE, dengan menggunakan algoritma SPF. Hasilnya berupa 6 alamat TE *link*, yang dapat divisualisasikan seperti pada Gambar 4.3.

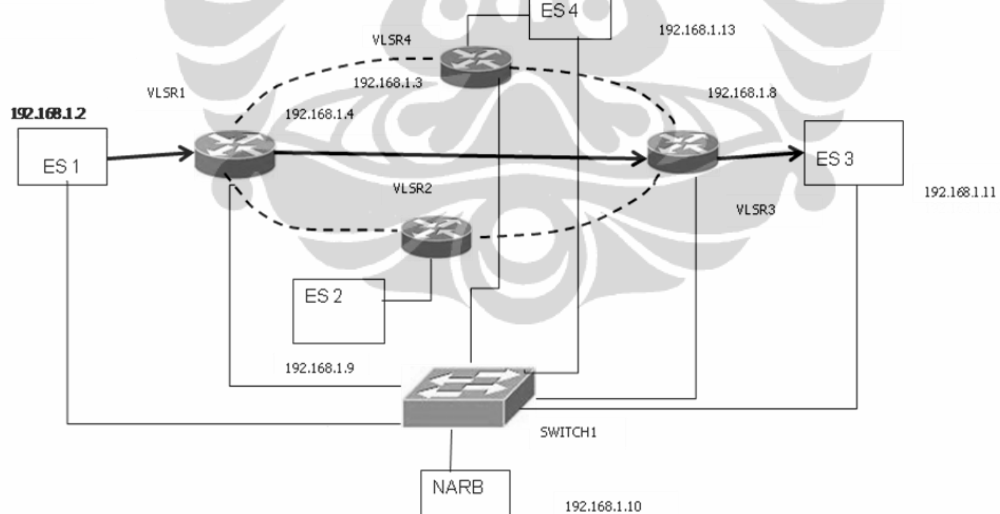
```

narb:~# narb_test -S 192.168.1.2 -D 192.168.1.11 -V -a
NARB@[2010/05/02 15:42:32] : Request successful! ERO returned...
NARB@[2010/05/02 15:42:32] : HOP-TYPE [strict]: 11.1.1.1 [UnumIfId: 262244(4,100
)]
NARB@[2010/05/02 15:42:32] : HOP-TYPE [strict]: 11.1.1.2 [UnumIfId: 262244(4,100
)]
NARB@[2010/05/02 15:42:32] : HOP-TYPE [strict]: 11.1.2.1 [UnumIfId: 262244(4,100
)]
NARB@[2010/05/02 15:42:32] : HOP-TYPE [strict]: 11.1.2.2 [UnumIfId: 262244(4,100
)]
NARB@[2010/05/02 15:42:32] : HOP-TYPE [strict]: 11.1.7.2 [UnumIfId: 262244(4,100
)]
NARB@[2010/05/02 15:42:32] : HOP-TYPE [strict]: 11.1.7.1 [UnumIfId: 262244(4,100
)]
NARB@[2010/05/02 15:42:32] : E2E VLAN TAG [ 100 ]
NARB@[2010/05/02 15:42:32] : ALL E2E VLAN TAGS: 100 101 102 103 104 105 106 107
108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127
128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147
148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167
168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187
188 189 190 191 192 193 194 195 196 197 198 199 200
narb:~# █

```

Gambar 4.2 Pengetesan NARB antara IP address 192.168.1.2 dengan 192.168.1.131

Gambar 4.3 menunjukkan bahwa path tersingkat antara ES1 dan ES2 yaitu garis yang ditebalkan yang jalurnya adalah ES1- VLSR 1 – VLSR 3 – ES 2, alamat TE antara ES1 dan VLSR1 adalah 11.1.1.1 - 11.1.1.2 , antara VLSR1 dan VLSR3 adalah 11.1.8.2 - 11.1.8.1 antara VLSR 3 dan ES 2 adalah 11.1.10.2.2 - 11.1.10.1



Gambar 4.3 Pengujian NARB antara ES1 dan ES3

4.5.2 Pengujian NARB antara *End System 1* (ES1) dengan Semua *Node* yang Terhubung.

Hasil pengujian NARB dari alamat sumber ES1,192.168.1.2 ke alamat *node* lainnya ditabulasikan pada Tabel 4.2. Pengujian dilakukan sama halnya

dengan pengujian sebelumnya hanya saja dengan mengganti alamat tujuan. Tabel ini menunjukkan jumlah *hop* yang dilewati jalur untuk mencapai alamat tujuan merupakan jumlah *hop* yang terkecil dan merupakan *path* tersingkat dari beberapa *path* yang tersedia. Misalnya antara ES1 dengan ES3, *path* yang tersedia adalah :

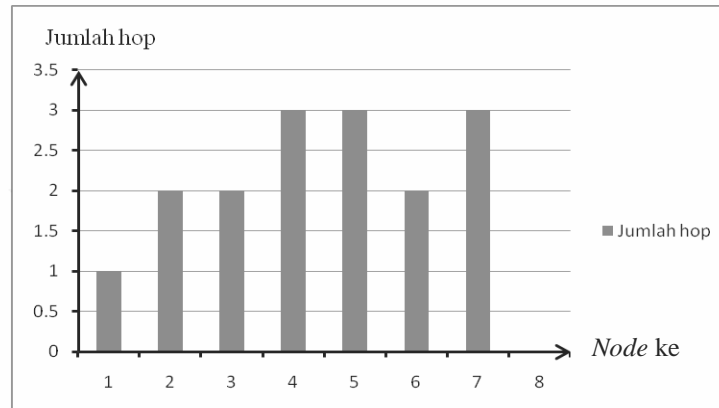
- ES1- VLSR1-VLSR3-ES3
- ES1-VLSR1-VLSR2-VLSR3-ES3
- ES1-VLSR1-VLSR4-VLSR3-ES3

Setelah dilakukan pengujian NARB maka *path* yang dipilih adalah ES1-VLSR1-VLSR3-ES3, hal ini ditunjukkan pada Tabel 4.2 data nomor 4.

Tabel 4.2 *Path* yang terbentuk antara alamat asal 192.168.1.2 ke alamat lainnya pada jaringan

No	Alamat Asal	Alamat Tujuan	Jumlah addres alamat <i>TE Link</i>	Jalur yang terbentuk
1	192.168.1.2	192.168.1.4	2	ES1-VLSR1
2	192.168.1.2	192.168.1.6	4	ES1-VLSR1-VLSR2
3	192.168.1.2	192.168.1.3	4	ES1-VLSR1-VLSR4
4	192.168.1.2	192.168.1.11	6	ES1-VLSR1-VLSR3-ES3
5	192.168.1.2	192.168.1.13	6	ES1-VLSR1-VLSR4-ES4
6	192.168.1.2	192.168.1.8	4	ES1-VLSR1-VLSR3
7	192.168.1.2	192.168.1.9	6	ES1-VLSR1-VLSR3-ES2
8	192.168.1.2	192.168.1.10	-	

Secara grafis dapat diperlihatkan kan pada Gambar 4.4, dimana pada gambar grafik tersebut terlihat bahwa masing -masing bar merupakan jalur tersingkat antara ES1 dan *node* tujuan yang dipilih, hal ini sesuai dengan topology jaringan yang dibangun. Selain itu juga terlihat bahwa *path* dari ES-1 yang terdekat adalah menuju VLSR 1 yang hanya terdiri dari 2 lamat TE link, yang berari bila paket dikirimkan ke VLSR 1 hanya memerlukan 1 *hop*.



Keterangan : node 1 : VLSR 1, node 2:VLSR2 ,node 3:VLSR4, node 4 : ES3, node 5 : ES4, node 6: VLSR3, node 7:ES2, node 8 : NARB

Gambar 4.4 Grafik jumlah hop terkecil antara node sumber (ES1) dengan node tujuan yang dipilih.

4.5.3 Pengujian NARB antara ES3 dan ES1

Pengujian berikutnya adalah untuk mengamati pemilihan jalur antara ES3 dengan ES 1. Pengujian ini bermaksud untuk mengetahui apakah jalur yang dipilih sama untuk kedua arah. Gambar 4.5, memperlihatkan bahwa pengujian menggunakan *node* ES3 sebagai *node* sumber yang memiliki alamat 192.168.1.9 dan ES1 sebagai node tujuan yang memiliki alamat 192.168.1.2. -V menunjukkan tag VLAN yang tersedia yaitu ada 100 Vtag yang dibawa oleh NARB, sedangkan -a memperlihatkan vlan tag yang digunakan, yakni 4 tag.

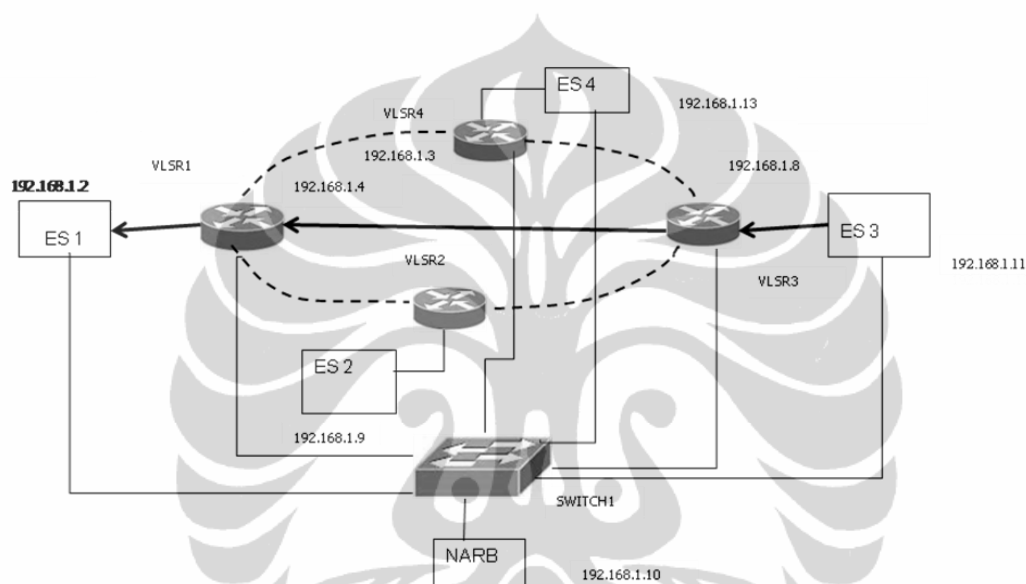
```

UML - narb
168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187
188 189 190 191 192 193 194 195 196 197 198 199 200
narb:~# narb_test -S 192.168.1.9 -D 192.168.1.2 -V -a
NARB@[2010/05/11 03:42:08] : Request successful! ERO returned...
NARB@[2010/05/11 03:42:08] : HOP-TYPE [strict]: 11.1.5.2 [UnumIfId: 262244(4,100
)]
NARB@[2010/05/11 03:42:08] : HOP-TYPE [strict]: 11.1.5.1 [UnumIfId: 262244(4,100
)]
NARB@[2010/05/11 03:42:08] : HOP-TYPE [strict]: 11.1.3.2 [UnumIfId: 262244(4,100
)]
NARB@[2010/05/11 03:42:08] : HOP-TYPE [strict]: 11.1.3.1 [UnumIfId: 262244(4,100
)]
NARB@[2010/05/11 03:42:08] : HOP-TYPE [strict]: 11.1.1.2 [UnumIfId: 262244(4,100
)]
NARB@[2010/05/11 03:42:08] : HOP-TYPE [strict]: 11.1.1.1 [UnumIfId: 262244(4,100
)]
NARB@[2010/05/11 03:42:08] : E2E VLAN TAG [ 100 ]
NARB@[2010/05/11 03:42:08] : ALL E2E VLAN TAGS: 100 101 102 103 104 105 106 107
108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127
128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147
148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167
168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187
188 189 190 191 192 193 194 195 196 197 198 199 200
narb:~#

```

Gambar 4.5 Pengetesan NARB antara ES3 dengan ES 1

Jalur TE *link* yang terbentuk dapat dilihat pada Gambar 4.6, gambar tersebut memperlihatkan bahwa *path* yang terbentuk antara ES3 dan ES1 yaitu ES3-VLSR3-VLSR1-ES1 yang merupakan path yang tersingkat. Bila dibandingkan dengan path antara ES1 dengan ES3 terbukti path yang terbentuk bersifat *bidirectional*. Hal ini berarti keterbatasan sumberdaya kedua arah sama. Gambar 4.5 merupakan hasil perhitungan RCE yang menggunakan algoritma SPF. Hasilnya berupa 6 alamat TE *link*, yang dapat divisualisasikan seperti Gambar 4.6.

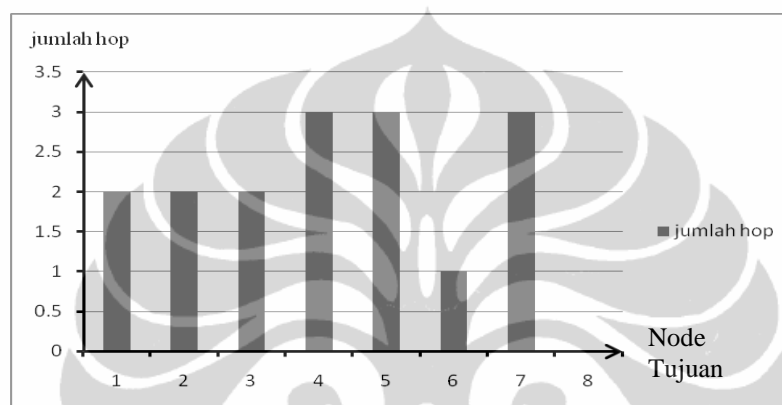


Gambar 4.6 NARB *test* antara ES3 dan ES1

Gambar 4.6 memperlihatkan bahwa *path* tersingkat antara ES3 dan ES1 yaitu garis yang ditebalkan yang jalurnya adalah ES3- VLSR 3 – VLSR 1 – ES 1, alamat TE antara ES1 dan VLSR1 adalah 11.1.5.1 - 11.1.5.2 , antara VLSR1 dan VLSR3 adalah 11.1.3.2 - 11.1.3.1 antara VLSR 3 dan ES 2 adalah 11.1.1.2 - 11.1.1.1 . Sedangkan *path* yang terbentuk antara ES 3 yang memiliki alamat IP 192.168.1.11 dengan perangkat lainnya ditabulasi pada Tabel 4.3. Sama halnya dengan Tabel 4.2. jalur yang terbentuk juga merupakan jalur dengan jumlah *hop* paling sedikit.

Tabel 4.3 *Path* yang terbentuk antara alamat asal 192.168.1.11 ke alamat lainnya pada jaringan

No	Alamat Asal	Alamat Tujuan	Jumlah alamat	Jalur yang terbentuk
1	192.168.1.11	192.168.1.4	4	ES3-VLSR3-VLSR1
2	192.168.1.11	192.168.1.6	4	ES3-VLSR3-VLSR2
3	192.168.1.11	192.168.1.3	4	ES3-VLSR3-VLSR4
4	192.168.1.11	192.168.1.9	6	ES3-VLSR3-VLSR2-ES2
5	192.168.1.11	192.168.1.13	6	ES3-VLSR3-VLSR4-ES4
6	192.168.1.11	192.168.1.8	2	ES3 – VLSR3
7	192.168.1.11	192.168.1.2	6	ES3-VLSR3-VLSR1-ES1
8	192.168.1.11	192.168.1.10	0	Tidak memiliki TE <i>link</i>



Keterangan : node1:VLSR1, node 2: VLSR2, node3: VLSR4, node4: ES2, node5: ES4, node6 : VLSR3, node7: ES1, node8:NARB

Gambar 4.7 Grafik jumlah *hop* dari *node* sumber ES2 ke *node* lain

Secara grafis dapat diperlihatkan kan pada Gambar 4.7 dimana pada gambar grafik tersebut terlihat bahwa masing -masing bar merupakan jalur tersingkat antara ES3 dan *node* tujuan yang dipilih, hal ini sesuai dengan topology jaringan yang dibangun.

4.5.4 Pengujian NARB antara ES2 dan VLSR4

Pengujian berikutnya dilakukan antara ES2 dan VLSR4. Pengujian ini bertujuan untuk melihat apakah NARB dapat menentukan jalur tersingkat ke tujuan yang berupa *router*. Gambar 4.7 memperlihatkan *Explicit Route Object* (ERO) yang terbentuk antara ES2 dengan VLSR4, terdiri dari enam alamat *TE link*. Dari jumlah alamat yang ada tersebut menunjukkan bahwa *path* yang terbentuk melewati tiga *hop*, yang merupakan *path* tersingkat diantara perangkat tersebut. Pengujian ini juga membuktikan bahwa NARB mampu untuk menentukan jalur tersingkat yang menuju *router*.

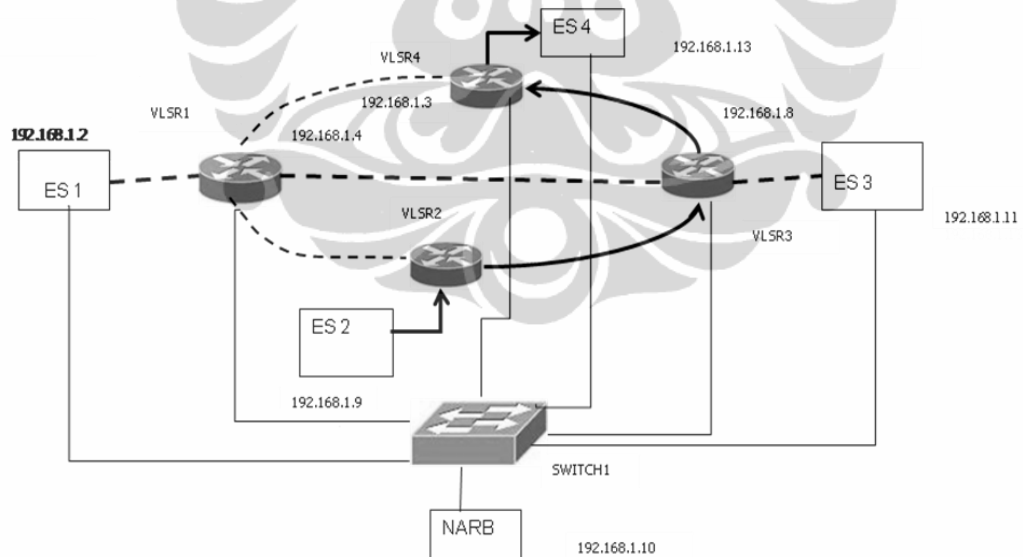
```

UML - narb
168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187
188 189 190 191 192 193 194 195 196 197 198 199 200
narb:~# narb_test -S 192.168.1.11 -D 192.168.1.3 -v -a
NARB@[2010/05/11 03:47:01] : Request successful! ERO returned...
NARB@[2010/05/11 03:47:01] : HOP-TYPE [strict]: 11.1.7.1 [UnumIfId: 262244(4,100
)]
NARB@[2010/05/11 03:47:01] : HOP-TYPE [strict]: 11.1.7.2 [UnumIfId: 262244(4,100
)]
NARB@[2010/05/11 03:47:01] : HOP-TYPE [strict]: 11.1.4.1 [UnumIfId: 262244(4,100
)]
NARB@[2010/05/11 03:47:01] : HOP-TYPE [strict]: 11.1.4.2 [UnumIfId: 262244(4,100
)]
NARB@[2010/05/11 03:47:01] : HOP-TYPE [strict]: 11.1.9.2 [UnumIfId: 262244(4,100
)]
NARB@[2010/05/11 03:47:01] : HOP-TYPE [strict]: 11.1.9.1 [UnumIfId: 262244(4,100
)]
NARB@[2010/05/11 03:47:01] : E2E VLAN TAG [ 100 ]
NARB@[2010/05/11 03:47:01] : ALL E2E VLAN TAGS: 100 101 102 103 104 105 106 107
108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127
128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147
148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167
168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 167
188 189 190 191 192 193 194 195 196 197 198 199 200
narb:~# █

```

Gambar 4.8 Pengetesan NARB antara ES3 dengan VLSR 4

Jika divisualisasikan *path* tersebut dapat dilihat pada Gambar 4.9 dengan jalur yang dilewati ES3-VLSR2-VLSR3-VLSR4, yaitu jalur yang garisnya ditebalkan. Sama seperti kasus sebelumnya hasil perhitungan RCE selalu mencari jalur terdekat dengan patokan jumlah hop yang paling sedikit.



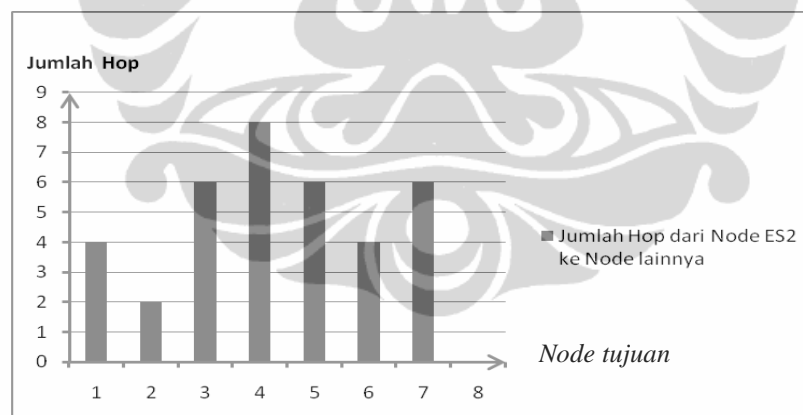
Gambar 4.9 Pengujian NARB antara ES2 dan VLSR4

Sedangkan jalur tersingkat antara ES2 dengan *node* lainnya dapat dilihat pada Tabel 4.4.

Tabel 4.4 *Path* antara ES2 dan Node lainnya

No	Sumber	Tujuan	Jumlah alamat	Jalur yang terbentuk
1	192.168.1.9	192.168.1.4	4	ES2- VLSR2-VLSR4
2	192.168.1.9	192.168.1.6	2	ES2-VLSR2
3	192.168.1.9	192.168.1.2	6	ES2-VLSR2-VLSR1-ES1
4	192.168.1.9	192.168.1.13	8	ES2-VLSR2-VLSR3-VLSR4-
5	192.168.1.9	192.168.1.3	6	ES2-VLSR2-VLSR3-VLSR4
6	192.168.1.9	192.168.1.8	4	ES2-VLSR2-VLSR3
7	192.168.1.9	192.168.1.11	6	ES2-VLSR2-
8	192.168.1.9	192.168.1.10	-	Tidak terdapat TE Link

Tabel tersebut ditampilkan secara grafis pada Gambar 4.10. Pada gambar path yang tersingkat adalah menuju alamat 192.168.1.6 yang merupakan alamat dari VLSR 2. Secara grafis dapat diperlihatkan kan pada Gambar 4.7 dimana pada gambar grafik tersebut terlihat bahwa masing -masing bar merupakan jalur tersingkat antara ES3 dan *node* tujuan yang dipilih, hal ini sesuai dengan topology jaringan yang dibangun.



Keterangan *node* tujuan : 1. VLSR1, 2. VLSR2, 3. ES1, 4.ES4, 5.VLSR4, 6.VLSR3, 7.ES3, 8.NARB

Gambar 4.10 Grafik jumlah hop dari node sumber (ES2) ke node tujuan

4.5.5 Pengujian NARB antara ES4 dan ES1

Pengujian berikutnya dilakukan antara ES4 dan ES1, Gambar 4.11 memperlihatkan *Explicit Route Object* yang terbentuk pada pengujian tersebut, terdiri dari enam alamat *TE link*. Dari jumlah alamat yang ada tersebut

menunjukkan bahwa *path* yang terbentuk melewati tiga *hop*, yang merupakan *path* tersingkat diantara perangkat tersebut.

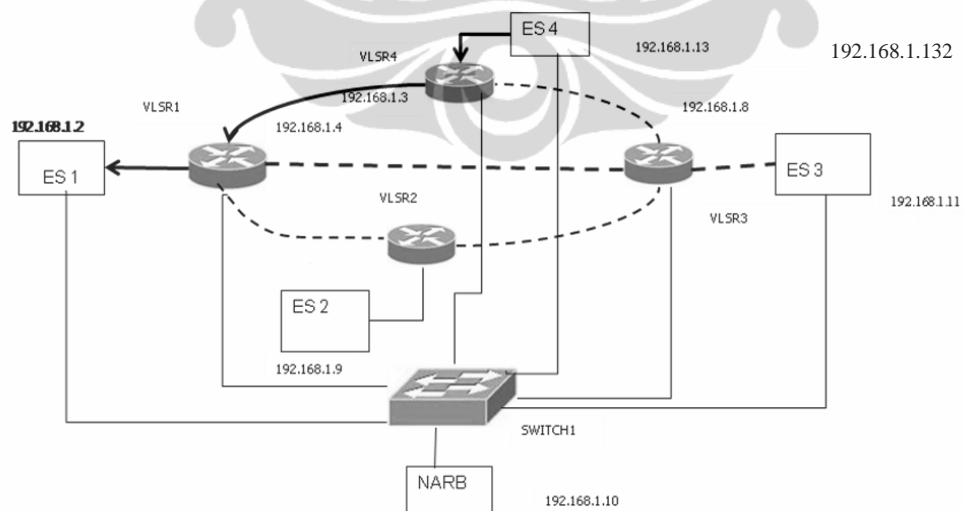
```

UML - narb
168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187
188 189 190 191 192 193 194 195 196 197 198 199 200
narb:~# narb_test -S 192.168.1.13 -D 192.168.1.2 -v -a
NARB@[2010/05/11 03:35:43] : Request successful! ERO returned...
NARB@[2010/05/11 03:35:43] : HOP-TYPE [strict]: 11.1.10.1 [UnumIfId: 262244(4,10
0)]
NARB@[2010/05/11 03:35:43] : HOP-TYPE [strict]: 11.1.10.2 [UnumIfId: 262244(4,10
0)]
NARB@[2010/05/11 03:35:43] : HOP-TYPE [strict]: 11.1.8.1 [UnumIfId: 262244(4,100
)]
NARB@[2010/05/11 03:35:43] : HOP-TYPE [strict]: 11.1.8.2 [UnumIfId: 262244(4,100
)]
NARB@[2010/05/11 03:35:43] : HOP-TYPE [strict]: 11.1.1.2 [UnumIfId: 262244(4,100
)]
NARB@[2010/05/11 03:35:43] : HOP-TYPE [strict]: 11.1.1.1 [UnumIfId: 262244(4,100
)]
NARB@[2010/05/11 03:35:43] : E2E VLAN TAG [ 100 ]
NARB@[2010/05/11 03:35:43] : ALL E2E VLAN TAGS: 100 101 102 103 104 105 106 107
108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127
128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147
148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167
168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187
188 189 190 191 192 193 194 195 196 197 198 199 200
narb:~#

```

Gambar 4.11 Pengujian NARB antara ES4 dan ES1

Jika divisualisasikan *path* tersebut dapat dilihat pada gambar 4.12 dengan jalur yang dilewati ES3-VLSR2-VLSR3-VLSR4, yaitu jalur yang garisnya ditebalkan. Sama seperti kasus sebelumnya hasil perhitungan RCE selalu mencari jalur terdekat dengan patokan jumlah *hop* yang paling sedikit.



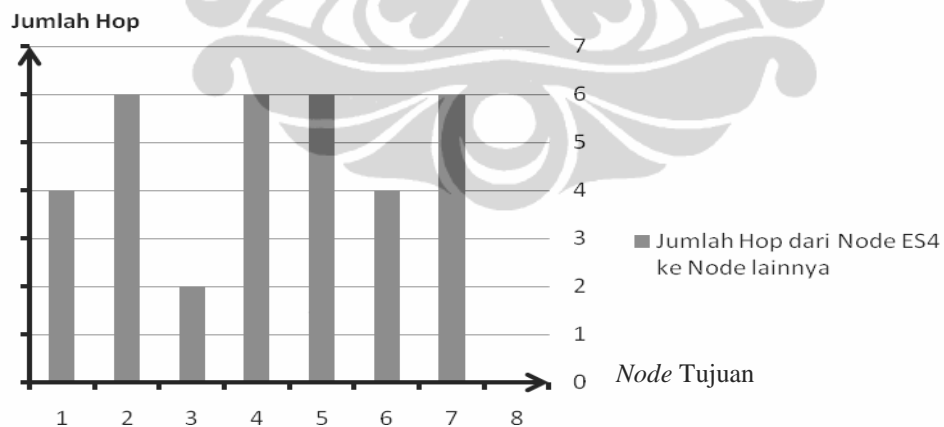
Gambar 4.12 Pengujian NARB antar ES4 dan ES1

Sedangkan antara ES3 dengan *node* lainnya dapat dilihat pada Tabel 4.5.

Tabel 4.5 *Path* antar ES4 dengan *node* lainnya

No	Source	Tujuan	Jumlah	Jalur yang terbentuk
1	192.168.1.13	192.168.1.4	4	ES4-VLSR4-VLSR1
2	192.168.1.13	192.168.1.6	6	ES4 – VLSR4-VLSR3-VLSR2
3	192.168.1.13	192.168.1.3	2	ES4-VLSR4
4	192.168.1.13	192.168.1.11	6	ES4-VLSR4-VLSR2-ES3
5	192.168.1.13	192.168.1.9	6	ES4-VLSR4-VLSR3-ES2
6	192.168.1.13	192.168.1.8	4	ES4-VLSR4-VLSR3
7	192.168.1.13	192.168.1.2	6	ES4-VLSR4-VLSR1-ES1
8	192.168.1.13	192.168.1.10	-	-

Tabel 4.5 memperlihatkan bahwa jalur yang terbentuk antara ES 4 dengan *node* lainnya merupakan jalur tersingkat yang dilewati, sedangkan untuk jalur antara ES4 dan NARB dengan alamat 192.168.1.10, tidak terbentuk TE *link*-nya, karena NARB tidak digunakan untuk lalu lintas trafik.



Keterangan : node tujuan 1. VLSR1, 2. VLSR2, 3.ES4, 4.ES3, 5.ES2, 6.VLSR3, 7.ES1, 8.NARB

Gambar 4.13 Grafik jumlah *hop* dari *node* sumber(ES4) ke *node* tujuan

4.6 Pengujian Pembentukan LSP

Pembentukan LSP dilakukan per *hop*, dengan melakukan *telnet* ke 2611, langkah pertamanya adalah melakukan *edit* LSP, *set* alamat asal dan alamat tujuan *set bandwidth*, *swcap encoding*, *gpid Vtag*. Bila pengesetan berhasil maka LSP akan terbentuk. Sebaliknya kalau ada kesalahan akan muncul pesan *error*, . atau bila tidak tersedia pada NARB akan muncul pesan *commit*.

4.5.1 Pembentukan LSP dengan Bandwidth 1 Giga bit/s

Tujuan akhir dari penelitian ini adalah untuk membangun LSP dengan menggunakan *script* konfigurasi yang telah dibuat menggunakan XML. Sebagaimana telah dijelaskan pada bab 3, bahwa pembentukan LSP dimulai dari pengaturan pemesanan sumberdaya yang diinginkan, yang dibatasi oleh ketersediaan sumberdaya yang ada pada NARB, setelah pengaturan selesai dan NARB memberikan respon maka LSP terbentuk. Sinyal protokol RSVP pada pembentukan LSP melewati GRE, yang telah di konfigurasi. Pada data yang diperoleh saat pengaturan LSP, sinyal RSVP mengalir melalui GRE2, GRE 6, GRE 4. Hal ini dapat dilihat pada gambar 4.14, 4.15 dan 4.16. Dimana pada gambar 4.14 dapat dilihat log VLSR1 memperlihatkan data-data pemesanan sender *template* berisi alamat IP sumber, *identifier ID*, *T-Spec*, *ADSPEC*, *Label Request*, *Explicit route*.

```

13:43:43.048 interface system index is 10
13:43:43.048 ***** new message received *****
13:43:43.048 gre3 received MSG from 10.1.2.2 : PATH 1 1 ttl:50 length:208
SESSION:192.168.1.4/2000/100772032
RSVP_HOP:Hop address: 10.1.2.2LIH: 3 TLV: Type: 1 Length: 8 Value: IPv4: 18461
542B
TIME_VALUES:30000
SENDER_TEMPLATE:192.168.1.6/1000
SENDER_TSPEC: Generalized r:1000.000 b:1000.000 p:1000.000 m:100 M:1500
ADSPEC:length:40 hops: 1 bw: inf lat: 0 MTU: 2147483647
LABEL_REQUEST:LSP_Enc: 2 Sw_Type: 51 G_Pid: 33
EXPLICIT_ROUTE:UNumIfID:{11.1.2.1,262244}--
DRAGON_EXT_INFO: [(1: Service Confirmation ID: ucid=855503833, seqnum=214748364
9 (5: MonNodeList: -192.168.1.6 )]
SESSION_ATTRIBUTE: SetupPriority: 7 HoldingPriority: 7 Flags: 0 NameLength: 4 S
essionName: test
13:43:43.048 found Session: 192.168.1.4/2000/100772032
13:43:43.049 received PATH for 192.168.1.4/2000.100772032
13:43:43.049 found Session: 192.168.1.4/2000/100772032
13:43:43.049 MPLS: explicit route for 192.168.1.4 is IP4:192.168.1.4/32--
13:43:43.050 routing result after adjustment: rsvp-api
13:43:43.050 found PSB:192.168.1.6/1000 from PHopSB:10.1.2.2[3] via gre3
13:43:43.050 PSB:updateRoutingInfo done, gateway is 0.0.0.0 , new lif count: 1
RSVPD.log (END)

```

Gambar 4.14 Log RSVPD pada VLSR 1

Dari data yang terlihat pada Gambar 4.14 yang diambil pada VLSR1, GRE3 telah menerima pesan dari 10.1.1.2 (GRE2) : PATH 1 1, ttl =50 panjang

208, diikuti dengan spesifikai yang diminta. *Explicit_route* yang tersedia adalah UnumIFID 11.1.2.1, yang merupakan alamat TE link.

```

red-vlsr2>
Vty connection is timed out.
Connection closed by foreign host.
red-vlsr2:~# cd /var/log
red-vlsr2:/var/log# less -n -i +G RSVPD.log
13:46:17.751 interface system index is 10
13:46:17.751 ***** new message received *****
13:46:17.751 gre5 received MSG from 10.1.4.2 : PATH 1 1 ttl:50 length:208
SESSION:192.168.1.6/2000/134326464
RSVP_HOP:Hop address: 10.1.4.2LIH: 3 TLV: Type: 1 Length: 8 Value: IPv4: 18461
5938
TIME_VALUES:30000
SENDER_TEMPLATE:192.168.1.8/1000
SENDER_TSPEC: Generalized r:1000.000 b:1000.000 p:1000.000 m:100 M:1500
ADSPEC:length:40 hops: 1 bw: inf lat: 0 MTU: 2147483647
LABEL_REQUEST:LSP_Enc: 2 Sw_Type: 51 G_Pid: 33
EXPLICIT_ROUTE:UNumIFID:{11.1.4.1.262244}--
DRAGON_EXT_INFO: [(1: Service Confirmation ID: ucId=320312482, seqnum=214748364
9)(5: MonNodeList: -192.168.1.8-)]
SESSION_ATTRIBUTE: SetupPriority: 7 HoldingPriority: 7 Flags: 0 NameLength: 4 S
essionName: test
13:46:17.752 found Session: 192.168.1.6/2000/134326464
13:46:17.752 received PATH for 192.168.1.6/2000/134326464
13:46:17.752 found Session: 192.168.1.6/2000/134326464
13:46:17.753 MPLS: explicit route for 192.168.1.6 is IP4:192.168.1.6/32--
13:46:17.753 routing result after adjustment: rsvp-api
13:46:17.753 found PSB:192.168.1.8/1000 from PHopSB:10.1.4.2[3] via gre5
13:46:17.753 PSB:updateRoutingInfo done, gateway is 0.0.0.0, new lif count: 1
RSVPD.log (END)

```

Gambar 4.15 Log pada VLSR 2

```

red-vlsr3:~# cd /var/log
red-vlsr3:/var/log# less -n -i +G RSVPD.log
13:36:29.699 MPLS: explicit route for 192.168.1.8 is IP4:192.168.1.8/32--
13:36:29.699 routing result after adjustment: rsvp-api
13:36:29.699 found PSB:192.168.1.9/1000 from PHopSB:10.1.5.2[2] via gre4
13:36:29.699 PSB:updateRoutingInfo done, gateway is 0.0.0.0, new lif count: 1
13:36:37.344 gre5 sends MSG to 10.1.4.1 : PATH 1 1 ttl:50 length:208
SESSION:192.168.1.6/2000/134326464
RSVP_HOP:Hop address: 10.1.4.2LIH: 3 TLV: Type: 1 Length: 8 Value: IPv4: 18461
5938
TIME_VALUES:30000
SENDER_TEMPLATE:192.168.1.8/1000
SENDER_TSPEC: Generalized r:1000.000 b:1000.000 p:1000.000 m:100 M:1500
ADSPEC:length:40 hops: 1 bw: inf lat: 0 MTU: 2147483647
LABEL_REQUEST:LSP_Enc: 2 Sw_Type: 51 G_Pid: 33
EXPLICIT_ROUTE:UNumIFID:{11.1.4.1.262244}--
DRAGON_EXT_INFO: [(1: Service Confirmation ID: ucId=320312482, seqnum=214748364
9)(5: MonNodeList: -192.168.1.8-)]
SESSION_ATTRIBUTE: SetupPriority: 7 HoldingPriority: 7 Flags: 0 NameLength: 4 S
essionName: test
13:36:41.955 creating Srefresh message, looking for 1 IDs at Hop:10.1.5.2 via gr
e4
13:36:41.957 adding ID 0
13:36:41.957 gre4 sends MSG to 10.1.5.2 : SREFRESH 1 1 ttl:15 length:20
MESSAGE_ID_LIST:flags:0 epoch:10566394 id:0
RSVPD.log (END)

```

Gambar 4.16 Log RSVPD pada VLSR 3

Sedangkan proses pembentukan LSP tersebut dapat dilihat pada gambar 4.17, yang menunjukkan pembentukan LSP dari ES1 ke node VLSR1.

```

red-es1> sh lsp
**LSP status summary**
Name      Status   Dir    Source (IP/LSP ID)  Destination (IP/Tunnel ID)
-----
teste1e2  Edit    <=>   192.168.1.2        192.168.1.4
                1000                2000
red-es1> commit lsp teste1e2
red-es1> sh lsp
**LSP status summary**
Name      Status   Dir    Source (IP/LSP ID)  Destination (IP/Tunnel ID)
-----
teste1e2  In service <=> 192.168.1.2        192.168.1.4
                1000                2000
red-es1>

```

Gambar 4.17 Pembentukan LSP antara es1 dengan VLSR1

Sebagaimana prosedur dalam pembentukan LSP, agar status LSP menjadi *in service* harus dilakukan perintah *commit*. Pada VLSR 1, juga dilakukan pembentukan LSP, jumlah LSP yang dibentuk disesuaikan dengan kebutuhan. Pada penelitian ini jumlah LSP yang dibuat disesuaikan dengan hubungan fisiknya. Hal ini dapat dilihat pada Gambar 4.18. Gambar menunjukkan pembentukan LSP dari ES1 ke ES2 dan ES3.

```
red-vlsr1> sh lsp
```

LSP status summary				
Name	Status	Dir	Source (IP/LSP ID)	Destination (IP/
teste1e2	In service	=>	192.168.1.2 1000	192.168.1.4 2000
testrir2	In service	<=>	192.168.1.4 1001	192.168.1.6 2001
testr1r3	In service	<=>	192.168.1.4 1002	192.168.1.8 2002

```
red-vlsr3> sh lsp
```

LSP status summary				
Name	Status	Dir	Source (IP/LSP ID)	Destination (IP
testr1r3	In service	=>	192.168.1.4 1002	192.168.1.8 2002
testr2r3	In service	=>	192.168.1.6 1003	192.168.1.8 2003
testr3e2	In service	<=>	192.168.1.8 1004	192.168.1.9 2004

Gambar 4.18 Status LSP

4.5.2 Pembentukan LSP dengan Batasan *Bandwidth* 149,3 Mbps

Berikutnya dilakukan pembentukan LSP dengan bandwith 149,3 Mbps. Sebagaimana prosedur sebelumnya pembentkan LSP dilakukan dengan mengatur alamat asal, alamat tujuan, *bandwidth*, kapabilitas *switch*, *encoding* dan Generalized Payload ID (GPID), serta *vtag*. Setelah pengaturan selesai, dapat dilihat LSP yang berhasil dibentuk pada Gambar 4.19. Gambar tersebut menunjukkan bahwa LSP dalam kondisi *in service*, ini berarti LSP sudah terbentuk. Kemudian pada VLSR 1 di lihat log aktifitas pembentukan LSP yang dapat dilihat pada Gambar 4.19. Dimana LSP yang terbentuk memiliki parameter *bandwidth* 149,3 Mbps, Jenis Encoding Ethernet, GPID Ethernet, dengan kapabilitas *switch* l2sc.

```

UML - red-es1
test Edit <=> 192.168.1.2 192.168.1.4
1000 2000
red-es1> commit lsp test
red-es1> sh lsp
**LSP status summary**
Name Status Dir Source (IP/LSP ID) Destination (IP/Tunnel ID)
-----
test In service <=> 192.168.1.2 192.168.1.4
1000 2000
red-es1> sh lsp test
Src 192.168.1.2/1000, dest 192.168.1.4/2000
GRI: 2012345314-2147483649
Generic TSPEC R=eth0.3M, B=eth0.3M, P=eth0.3M, m=100, M=1500
Encoding ethernet, Switching 12sc, G-Pid ethernet
Ingress Local ID Type: none id, Value: 1000
Egress Local ID Type: none id, Value: 2000.
No E2E LSP VLAN Tag configured.
Status: In service
red-es1>
red-es1>
Vty connection is timed out.
Connection closed by foreign host.
red-es1>#

UML - red-es2
Based on Zebra (version 0.94).
Copyright 1996-2002 Kunihiro Ishiguro.
Copyright 2003-2004 the Dragon Team.
User Access Verification
Password:
red-es2> sh lsp
**LSP status summary**
Name Status Dir Source (IP/LSP ID) Destination (IP/Tunnel ID)
-----
test2 In service => 192.168.1.8 192.168.1.9
1000 2000
red-es2> sh lsp test2
Src 192.168.1.8/1000, dest 192.168.1.9/2000
Generic TSPEC R=eth149.3M, B=eth149.3M, P=eth149.3M, m=100, M=1500
Encoding ethernet, Switching 12sc, G-Pid ethernet
No ingress Local ID configured.
No egress Local ID configured.
E2E LSP VLAN Tag: 100.
Status: In service
red-es2>

UML - red-vlsr1
1000 2000
red-vlsr1> commit lsp test1
red-vlsr1> sh lsp
**LSP status summary**
Name Status Dir Source (IP/LSP ID) Destination (IP/Tunnel ID)
-----
test In service => 192.168.1.2 192.168.1.4
1000 2000
test1 In service <=> 192.168.1.4 192.168.1.8
1000 2000
red-vlsr1> sh lsp test1
Src 192.168.1.4/1000, dest 192.168.1.8/2000
GRI: 20250474-2147483649
Generic TSPEC R=eth149.3M, B=eth149.3M, P=eth149.3M, m=100, M=1500
Encoding ethernet, Switching 12sc, G-Pid ethernet
Ingress Local ID Type: none id, Value: 1000
Egress Local ID Type: none id, Value: 2000.
E2E LSP VLAN Tag: 100.
Status: In service
red-vlsr1>
red-vlsr1>
Vty connection is timed out.
Connection closed by foreign host.
red-vlsr1>#

UML - red-vlsr3
Encoding ethernet, Switching 12sc, G-Pid ethernet
Ingress Local ID Type: none id, Value: 1000
Egress Local ID Type: none id, Value: 2000.
E2E LSP VLAN Tag: 100.
Status: In service
red-vlsr3> sh lsp
**LSP status summary**
Name Status Dir Source (IP/LSP ID) Destination (IP/Tunnel ID)
-----
test1 In service => 192.168.1.4 192.168.1.8
1000 2000
test2 In service <=> 192.168.1.8 192.168.1.9
1000 2000
red-vlsr3> sh lsp test2
Src 192.168.1.8/1000, dest 192.168.1.9/2000
GRI: 1828344679-2147483649
Generic TSPEC R=eth149.3M, B=eth149.3M, P=eth149.3M, m=100, M=1500
Encoding ethernet, Switching 12sc, G-Pid ethernet
Ingress Local ID Type: none id, Value: 1000
Egress Local ID Type: none id, Value: 2000.
E2E LSP VLAN Tag: 100.
Status: In service
red-vlsr3>#

```

Gambar 4.19 Status LSP yang terbentuk

4.7 Pengujian Protokol OSPF

4.6.1 IP OSPF Tetangga

Untuk melihat apakah protokol OSPF bekerja dapat dilakukan dengan melihat IP tetangga dari NARB yang hasilnya dapat dilihat pada Gambar 4.20. Gambar tersebut memperlihatkan tetangga terdekat dari NARB adalah VLSR 2 dengan alamat IP 192.168.1.6. Menggunakan alamat GRE 1, yaitu 10.1.6.1 dan 10.1.6.2.

```

narb> show ip ospf neighbor
Neighbor ID Pri State Dead Time Address Interface RXmtL RqstL DBsmL
192.168.1.6 1 Full/DROther 00:00:31 10.1.6.1 gre1:10.1.6.2 0 0 0

```

Gambar 4.20 Hasil Penghitungan IP OSPF tetangga terdekat

4.6.4 IP OSPF-TE Database

Database yang ada di NARB diperlihatkan pada Gambar 4.21, dimana terlihat dua jenis area yaitu *area router*, *area link*. Masing-masing area mempunyai alamat sesuai dengan konfigurasi yang telah dibuat yaitu 192.166.1.2,

192.168.1.4, 192.168.1.6, 192.168.1.8, 192.168.9. yang masing-masing adalah IP *address* dari VLSR 1, VLSR 2, VLSR3, VLSR4.

```

narb> show ip ospf-te database
OSPF-TE link state database, area 0.0.0.0
Type ID Adv Rtr Seq Age Cksum Len 0x8000 3 0xc7bc 28
Area-Router ID TLV 1.168.1.2 192.168.1.2 0xc7bc 28
Area-Router ID TLV 1.168.1.4 192.168.1.4 0xbdc1 28
Area-Router ID TLV 1.168.1.6 192.168.1.6 0xb1c7 28
Area-Router ID TLV 1.168.1.8 192.168.1.8 0xa7cc 28
Area-Router ID TLV 1.168.1.9 192.168.1.9 0x9dd1 28
Area-Link TLV 1.0.0.7 192.168.1.2 0xf7b3 156
Area-Link TLV 1.0.0.7 192.168.1.9 0x2970 156
Area-Link TLV 1.0.0.9 192.168.1.4 0x67cf 184
Area-Link TLV 1.0.0.9 192.168.1.6 0xc967 184
Area-Link TLV 1.0.0.9 192.168.1.8 0xfc28 184
Area-Link TLV 1.0.0.10 192.168.1.4 0xfd31 184
Area-Link TLV 1.0.0.10 192.168.1.6 0xa483 184
Area-Link TLV 1.0.0.10 192.168.1.8 0x66c2 184
Area-Link TLV 1.0.0.11 192.168.1.4 0x54c6 184
Area-Link TLV 1.0.0.11 192.168.1.8 0xed3e 184

```

Gambar 4.21 Database pada NARB

4.6.5 Database Detail IP OSPF

Database detail NARB dapat dilihat pada Gambar 4.22. Pada gambar terlihat bahwa salah satu data di dalam *database* yaitu Area Link Time Length Value (TLV) dengan alamat *interface* 1.0.0.11, memiliki alata TE link 11.1.32-11.1.3.1. Selain itu juga terdapat data tentang 8 *bandwidth unreserve*, 8 *bandwidth* LSP masing-masing maksimal 125 Mbps, kapabilitas *switch* L2SC.

```

Area-Link TLV 1.0.0.11 192.168.1.8 0x8000 2 0xdb47 184
Link: 160 octets of data
Link-Type: Point-to-point (1)
Link-ID: 192.168.1.4
Local Interface IP Address(es): 11.1.3.2
Remote Interface IP Address(es): 11.1.3.1
Traffic Engineering Metric: 10
Maximum Reservable Bandwidth: 125000000 (Bytes/sec)
Unreserved Bandwidth (pri 0): 125000000 (Bytes/sec)
Unreserved Bandwidth (pri 1): 125000000 (Bytes/sec)
Unreserved Bandwidth (pri 2): 125000000 (Bytes/sec)
Unreserved Bandwidth (pri 3): 125000000 (Bytes/sec)
Unreserved Bandwidth (pri 4): 125000000 (Bytes/sec)
Unreserved Bandwidth (pri 5): 125000000 (Bytes/sec)
Unreserved Bandwidth (pri 6): 125000000 (Bytes/sec)
Unreserved Bandwidth (pri 7): 125000000 (Bytes/sec)
Interface Switching Capability Descriptor: 12sc ethernet
Max LSP Bandwidth 0: 125000000 (Bytes/sec)
Max LSP Bandwidth 1: 125000000 (Bytes/sec)
Max LSP Bandwidth 2: 125000000 (Bytes/sec)
Max LSP Bandwidth 3: 125000000 (Bytes/sec)
Max LSP Bandwidth 4: 125000000 (Bytes/sec)
Max LSP Bandwidth 5: 125000000 (Bytes/sec)
Max LSP Bandwidth 6: 125000000 (Bytes/sec)
Max LSP Bandwidth 7: 125000000 (Bytes/sec)
-- L2SC specific information--

```

Gambar 4.22 Database detail pada NARB

4.7 Pengujian Skalabilitas Jaringan

Pengujian Skalabilitas ini bertujuan untuk mengetahui kemampuan RCE melakukan perhitungan jalur tersingkat. Pada pengujian ini dilakukan pengujian pada jaringan Gambar 3.1, dengan peningkatan jumlah VLSR, Jumlah End Sistem, dan jumlah Jalur yang tersedia.

4.7.1 Dengan Menambah Jumlah VLSR

Dengan menambah jumlah VLSR pada Gambar 3.5 dilakukan pengujian skalabilitas jaringan dari sisi kemampuan jaringan dalam menangani pertambahan jumlah VLSR, hasil pengujian dapat dilihat pada Tabel 4.4. Pada penelitian ini *node* sumbernya adalah ES-1. Pengujian menggunakan NARB_TEST, dilakukan dengan penambahan VLSR satu per satu pada VLSR 3. Setelah dilakukan penambahan terus menerus diketahui bahwa jumlah VLSR dibatasi sebanyak 35. Bila melebihi angka tersebut jaringan tidak bisa di aktifkan. Ketika jumlah VLSR

Tabel 4.4 *Path* antara ES2 dan VLSR n

No	Jumlah VLSR antara sumber dan tujuan pada jalur tersingkat	Jumlah alamat TE link (ERO)	Jumlah hop yang bisa dihitung	Keterangan
1	1	2	1	
2	2	4	2	
3	3	6	3	
4	4	8	4	
5	5	10	5	
.	.	.	.	jumlah ERO selalu 2 kali jumlah VLSR
28	28	56	28	
29	29	58	29	
30	30	60	30	
31	31	62	31	
32	32	64	32	
33	33	66	33	
34	34	68	34	
35	35	70	35	
36	36	-	-	Gagal

melebihi 35, maka sistem akan merespon dengan pesan yang memberi tahu bahwa *delay*nya adalah 0,25 μ dt. Selain itu path yang dilewati selalu melewati jalur tersingkat yang tersedia. Hal ini membuktikan bahwa NARB menggunakan algoritma *link state* SPF.

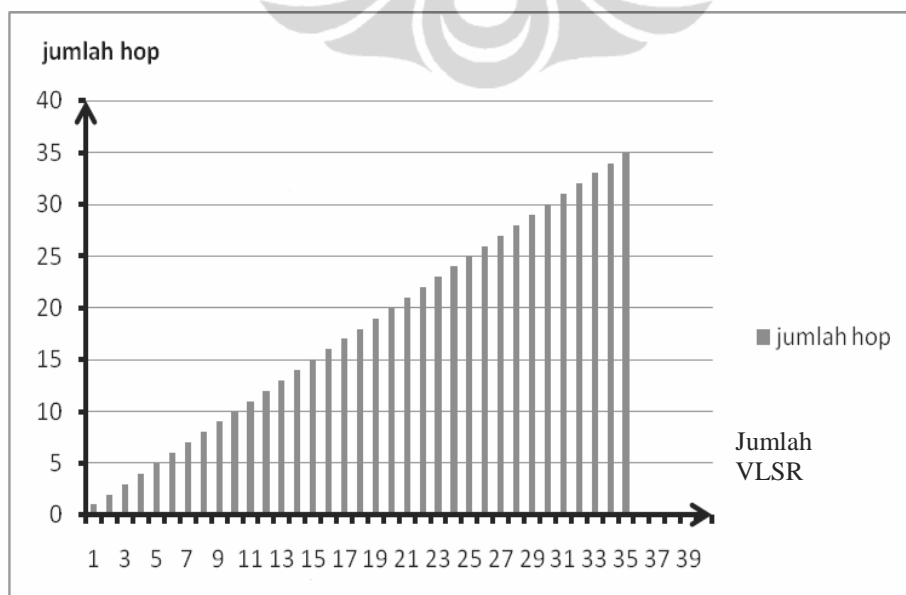
Secara grafis Tabel 4.4 dapat dilihat pada Gambar 4.26, dimana gambar tersebut memperlihatkan setelah pada jumlah 35, sistem VNE tidak mampu lagi melakukan fungsinya dan memberikan pesan seperti gambar 4.23

```

after: <class 'pexpect.EOF'>
match: None
match_index: None
exitstatus: None
flag_eof: True
pid: 23113
child fd: 180
closed: False
timeout: 30
delimiter: <class 'pexpect.EOF'>
logfile: <open file u'/home/dragon2/uml/vlsr-40.log', mode 'w' at 0xa4b1d40>
logfile_read: None
logfile_send: None
maxread: 2000
ignorecase: False
searchwindowsize: None
delaybeforesend: 0.05
delayafterclose: 0.1
delayafterterminate: 0.1
So long, and thanks for all the fish!
dragon2@zulhelman-laptop:~/VNE-snapshot-2009Jan12/configs$

```

Gambar 4.23 Pesan yang muncul ketika terpasang 36 VLSR



Gambar 4.24 Jumlah hop pada Jalur tersingkat yang dipilih fungsi jumlah VLSR

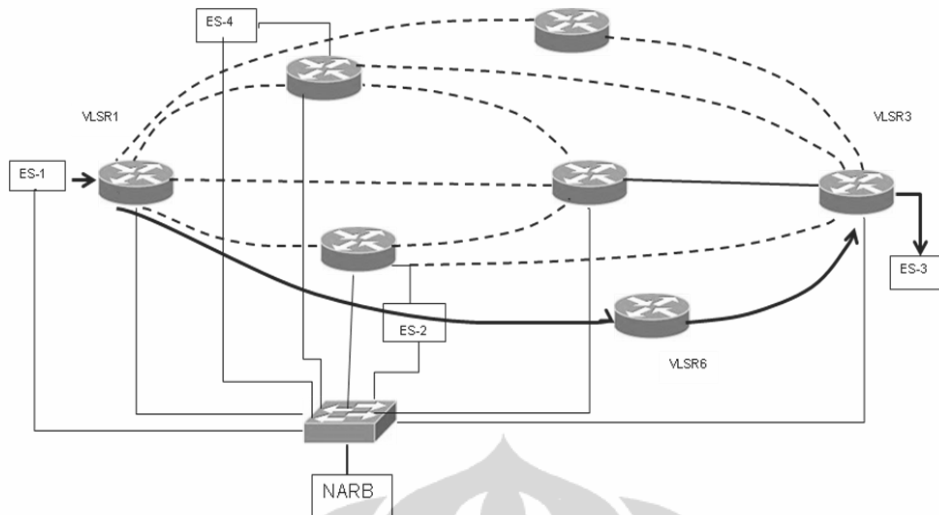
4.7.2 Dengan Menambah Jumlah Jalur Alternatif

Pengujian ini dimaksudkan untuk mengetahui ketahanan (*Robustness*) protokol OSPF-TE terhadap perubahanan topologi, skenarionya adalah dengan menambahkan path dengan jumlah *hop* yang sama secara terus menerus. Setelah dilakukan konfigurasi seperti gambar 3.6 dengan penambahan *path* sebanyak 17 dengan jumlah *hop* yang sama, dilakukan NARB Test, ternyata NARB memilih jalur tersingkat yang konsisten yaitu ES1- VLSR 1- VLSR 2- VLSR 5, dan ES 3. datanya dapat dilihat pada lampiran 5. Gambar 4.24 memperlihatkan *path* yang dipilih oleh protokol OSPF-TE.

```
Debian GNU/Linux 4.0 narb-1 tty1
narb-1 login: root
narb-1:~# narb_test -S 192.168.1.2 -D 192.168.1.11
NARB@[2010/06/10 13:30:52] : Request successful! ERO returned...
NARB@[2010/06/10 13:30:52] : HOP-TYPE [strict]: 11.1.1.1
NARB@[2010/06/10 13:30:52] : HOP-TYPE [strict]: 11.1.1.2
NARB@[2010/06/10 13:30:52] : HOP-TYPE [strict]: 11.1.13.2
NARB@[2010/06/10 13:30:52] : HOP-TYPE [strict]: 11.1.13.1
NARB@[2010/06/10 13:30:52] : HOP-TYPE [strict]: 11.1.12.2
NARB@[2010/06/10 13:30:52] : HOP-TYPE [strict]: 11.1.12.1
NARB@[2010/06/10 13:30:52] : HOP-TYPE [strict]: 11.1.5.1
NARB@[2010/06/10 13:30:52] : HOP-TYPE [strict]: 11.1.5.2
narb-1:~#
```

Gambar 4.24 Jalur tersingkat yang dipilih fungsi jumlah *path* yang tersedia

Dari data yang terlihat pada Gambar 4.24 diperoleh path yang dipilih adalah ES1-VLSR1-VLSR6-VLSR5-ES3 secara konsisten sampai penambahan 17 *path* dengan bobot yang sama. Disini dapat dilihat bahwa protokol OSPF-TE yang digunakan DRAGON memiliki *Robustness* yang baik. Jalur yang dipilih dapat dilihat pada Gambar 4.25, yaitu pada garis yang ditebalkan.



Gambar 4.25 Jalur tersingkat yang dipilih fungsi jumlah path yang tersedia

Berikutnya dilakukan pengujian dengan membentuk LSP pada *path* antara VLSR 1 dan VLSR 2. dengan asumsi jalur tersebut telah digunakan. Setelah dilakukan NARB Test antara ES 1 dan ES 3, dari data yang diperoleh diketahui bahwa NARB konsisten memilih jalur tersingkat yaitu ES1- VLSR 1- VLSR 2- VLSR 5, dan ES 3. Hasil pengujiannya dapat dilihat pada Gambar 4.26.

```

vlsr-6(edit-lsp-test)# set vtag a
vlsr-6(edit-lsp-test)# set vtag any
vlsr-6(edit-lsp-test)# exit
vlsr-6> sh l
local-id lsp
vlsr-6> sh ls
vlsr-6> sh lsp
**LSP status summary**
Name      Status  Dir  Source (IP/LSP ID)  Destination (IP/Tunnel ID)
-----
test      Edit   (<=>)  192.168,1,14      192.168,1,4
                1000              2000

vlsr-6> commit lsp test
vlsr-6> sh lsp test
Src 192.168,1,14/1000, dest 192.168,1,4/2000
GRI: 677796681-2147483649
Generic TSPEC R=gige, B=gige, P=gige, m=100, M=1500
Encoding ethernet, Switching l2sc, G-Pid ethernet
Ingress Local ID Type; none id, Value: 1000
Egress Local ID Type; none id, Value: 2000,
EZE LSP VLAN Tag: 100,
Status: In service
vlsr-6>

```

```

Debian GNU/Linux 4,0 narb-1 tty1
narb-1 login: root
narb-1:~# narb_test -S 192.168,1,2 -D 192.168,1,11
NARBe[2010/06/10 23:59:44] : Request successful! ERO returned
NARBe[2010/06/10 23:59:44] : HOP-TYPE [strict]: 11,1,1,1
NARBe[2010/06/10 23:59:44] : HOP-TYPE [strict]: 11,1,1,2
NARBe[2010/06/10 23:59:44] : HOP-TYPE [strict]: 11,1,13,2
NARBe[2010/06/10 23:59:44] : HOP-TYPE [strict]: 11,1,13,1
NARBe[2010/06/10 23:59:44] : HOP-TYPE [strict]: 11,1,12,2
NARBe[2010/06/10 23:59:44] : HOP-TYPE [strict]: 11,1,12,1
NARBe[2010/06/10 23:59:44] : HOP-TYPE [strict]: 11,1,5,1
NARBe[2010/06/10 23:59:44] : HOP-TYPE [strict]: 11,1,5,2
narb-1:~#

```

Gambar 4.26 Jalur tersingkat yang dipilih fungsi jumlah path yang tersedia saat LSP antara VLSR 1 dan VLSR 6 dalam kondisi *Inservices*

Pada gambar terlihat antara VLSR 1 (192.168.1.4) dan VLSR 6 (192.168.1.14) sudah terbentuk LSP, kemudian dilakukan NARB_Test ke ES 3(192.168.1.11). Hasil pengujiannya menunjukkan jalur yang dipilih tidak berubah.

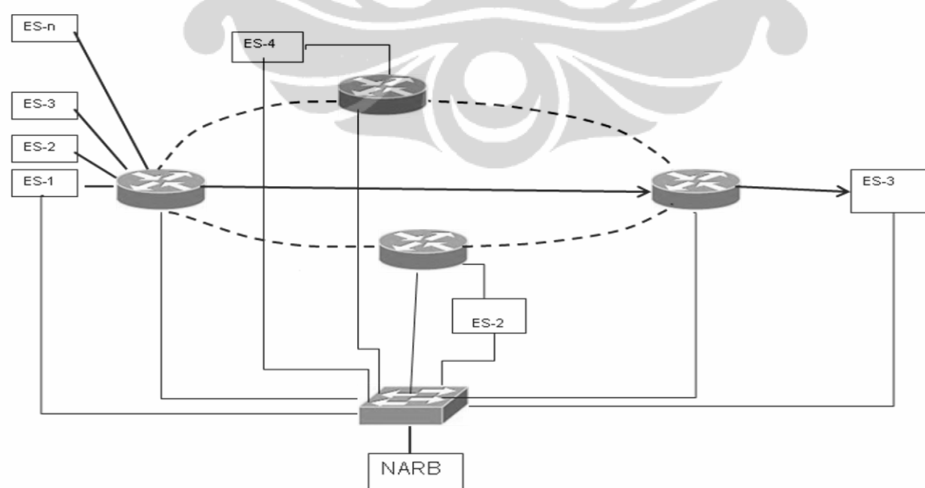
4.7.3 Dengan Menambah Jumlah ES

Pengujian ini bermaksud untuk mengetahui jumlah maksimum *end system* dipasang pada Router. Ketika jumlah *end system* sampai 15, salah satu hasil pengujiannya dapat dilihat pada Gambar 4.27, dimana sumbernya ES1 dengan alamat 192.168.1.2 sedangkan tujuannya adalah ES3 dengan alamat 192.168.1.11

```
narb-1 login: root
narb-1:~# narb_test -S 192.168.1.2 -D 192.168.1.11
NARB@[2010/06/13 03:11:01] : Request successful! ERO returned...
NARB@[2010/06/13 03:11:01] : HOP-TYPE [strict]: 11.1.1.1
NARB@[2010/06/13 03:11:01] : HOP-TYPE [strict]: 11.1.1.2
NARB@[2010/06/13 03:11:01] : HOP-TYPE [strict]: 11.1.8.2
NARB@[2010/06/13 03:11:01] : HOP-TYPE [strict]: 11.1.8.1
NARB@[2010/06/13 03:11:01] : HOP-TYPE [strict]: 11.1.5.1
NARB@[2010/06/13 03:11:01] : HOP-TYPE [strict]: 11.1.5.2
narb-1:~#
```

Gambar 4.27 Hasil Pengujian antara ES1 dengan ES3

Sedangkan visualisasi hasil pengujiannya dapat dilihat pada Gambar 4.28.



Gambar 4.28 Visualisasi Hasil Pengujian antara ES1 dengan ES3 ketika terdapat 15 ES pada VLSR 1.

Pada gambar terlihat bahwa terdapat 6 *TE link address* yang berarti terdapat 3 *hop* antara ES1 dengan ES3. Ini merupakan 3 *hop* tersingkat antara ES-1 dan ES-3 dari jalur yang tersedia yaitu ES1-VLSR1-VLSR3-ES3, *path* lainnya adalah :

- ES1-VLSR1-VLSR4-VLSR3-ES3
- ES1-VLSR1-VLSR2-VLSR3-ES3

Bila ES nya ditambah lagi NARB tidak mengenali perangkat lagi dengan pesan yang terlihat pada Gambar 4.29.

```

narb-1 login: root
narb-1:~# narb_test -S 192.168.1.2 -D 192.168.1.11
NARB@[2010/06/13 05:06:54] : Request failed : Unknown Source Address
narb-1:~# narb_test -S 192.168.1.2 -D 192.168.1.4
NARB@[2010/06/13 05:06:43] : Request failed : Unknown Source Address
narb-1:~# narb_test -S 192.168.1.4 -D 192.168.1.11
NARB@[2010/06/13 05:07:14] : Request failed : Unknown Source Address
narb-1:~# narb_test -S 192.168.1.4 -D 192.168.1.11
NARB@[2010/06/13 05:07:28] : Request failed : Unknown Source Address
narb-1:~# narb_test -S 192.168.1.5 -D 192.168.1.11
NARB@[2010/06/13 05:07:44] : Request failed : Unknown Source Address
narb-1:~# narb_test -S 192.168.1.5 -D 192.168.1.11
NARB@[2010/06/13 05:08:08] : Request failed : Unknown Source Address
narb-1:~# narb_test -S 192.168.1.12 -D 192.168.1.11
NARB@[2010/06/13 05:08:48] : Request failed : Unknown Source Address
narb-1:~# narb_test -S 192.168.1.8 -D 192.168.1.11
NARB@[2010/06/13 05:09:13] : Request successful! ERO returned...
NARB@[2010/06/13 05:09:13] : HOP-TYPE [strict]: 11.1.9.2
NARB@[2010/06/13 05:09:13] : HOP-TYPE [strict]: 11.1.9.1
NARB@[2010/06/13 05:09:13] : HOP-TYPE [strict]: 11.1.5.1
NARB@[2010/06/13 05:09:13] : HOP-TYPE [strict]: 11.1.5.2
narb-1:~# narb_test -S 192.168.1.6 -D 192.168.1.11
NARB@[2010/06/13 05:10:33] : Request successful! ERO returned...
NARB@[2010/06/13 05:10:33] : HOP-TYPE [strict]: 11.1.4.1
NARB@[2010/06/13 05:10:33] : HOP-TYPE [strict]: 11.1.4.2
NARB@[2010/06/13 05:10:33] : HOP-TYPE [strict]: 11.1.9.2
NARB@[2010/06/13 05:10:33] : HOP-TYPE [strict]: 11.1.9.1
NARB@[2010/06/13 05:10:33] : HOP-TYPE [strict]: 11.1.5.1
NARB@[2010/06/13 05:10:33] : HOP-TYPE [strict]: 11.1.5.2
narb-1:~# narb_test -S 192.168.1.4 -D 192.168.1.11
NARB@[2010/06/13 05:12:08] : Request failed : Unknown Source Address
narb-1:~# █
  
```

Gambar 4.29 Pesan kesalahan bila NARB tidak dapat menghitung TE link

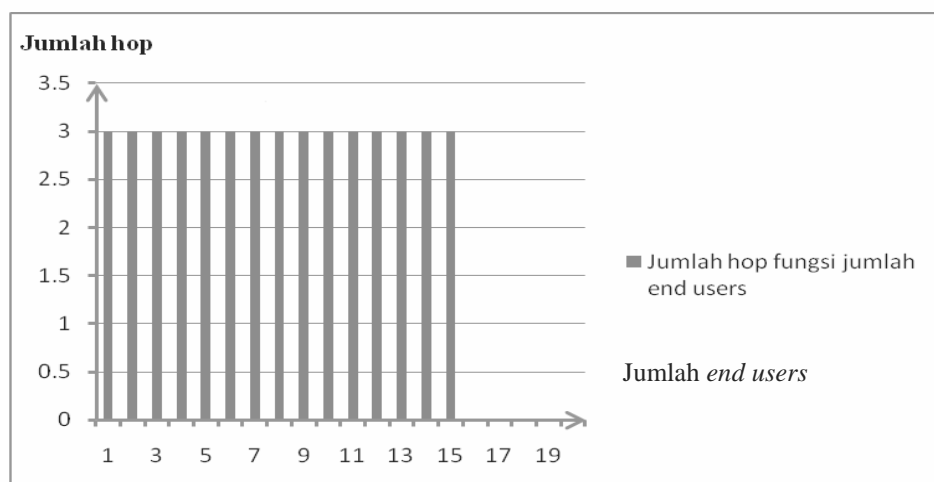
Data hasil pengujian selengkapnya dapat dilihat pada Tabel 4.5. Tabel tersebut menunjukkan bahwa jumlah maksimum ES yang diijinkan dipasang pada router sebanyak 15 *end system*. Data diambil dengan penambahan 5 *end system* secara terus menerus. Penambahan dilakukan melalui script XML, ketika jumlah *end system* mencapai 20 NARB_TEST memberikan respon tidak mengenal source address seperti pesan pada Gambar 4.29. Kemudian dilakukan pengurangan satu persatu sampai pengujian berhasil, sebagaimana yang telah dicatat pada Tabel 4.5. Pengujian berhasil dilakukan saat *end system* berjumlah 15.

Kemudian data yang ada pada Tabel 4.5 ditampilkan dalam bentuk grafik seperti yang ditunjukkan pada Gambar 4.30. Grafik melihat jumlah *hop* yang dipilih fungsi jumlah *end system*.

Tabel 4.5 Skalabilitas *End Systems* pada Router

No	Jumlah ES	Jumlah TE link	Jumlah Hop	Keterangan
1	1	6	3	
2	2	6	3	
3	3	6	3	
4	4	6	3	
5	5	6	3	
6	6	6	3	
7	7	6	3	
8	8	6	3	
9	9	6	3	
10	10	6	3	
11	11	6	3	
12	12	6	3	
13	13	6	3	
14	14	6	3	
15	15	6	3	
16	16	0	0	NARB tidak mengenali alamat sumber
17	17	0	0	
18	18	0	0	
19	19	0	0	
20	20	0	0	

Gambar 5.30 merupakan grafik dari Tabel 4.5, dimana pada saat jumlah end user sudah mencapai 16, sistem tidak mengenali lagi node sumbernya.



Gambar 4.30 Jumlah hop fungsi pertambahan end users