

BAB 3

METODE PENELITIAN

3.1 ALAT DAN PARAMETER YANG DITELITI

Penelitian berjudul “Analisis Performansi GRE Tunnel IPsec Dengan Metode *Failover* Pada *Open Source Router* VyOS” ini menggunakan beberapa perangkat lunak untuk membantu melakukan penelitian seperti *network emulator* GNS3 serta beberapa *tool* perangkat lunak seperti Wireshark dan D-ITG untuk mengumpulkan data.

3.1.1 Perangkat Keras

Perangkat keras yang digunakan terdiri dari satu komputer yang berfungsi untuk menjalankan GNS3 dan Wireshark dengan spesifikasi sebagai berikut :

Tabel 3. 1 Spesifikasi Komputer

| | |
|----------------|-----------------|
| Sistem Operasi | Linux Ubuntu |
| Processor | Intel I7 13700k |
| RAM | 16 GB |
| Harddisk | 2 TB |

3.1.2 Perangkat Lunak

a. *Graphical Network Simulator 3* (GNS3)

GNS3 merupakan perangkat lunak *network emulator* untuk merangkai topologi jaringan dan melakukan serangkaian pengujian jaringan, GNS3 yang digunakan merupakan versi 2.2.28.

b. Wireshark

Wireshark merupakan perangkat lunak *snifing tool* yang berfungsi untuk mengamati paket-paket yang ada ketika melakukan simulasi GRE *Tunnel* IPsec, Wireshark yang digunakan merupakan versi 3.6.1.

c. D-ITG

D-ITG merupakan perangkat lunak jaringan yang berfungsi untuk menguji parameter QoS seperti *throughput*, *delay*, dan *jitter* dengan mengirimkan trafik berupa TCP dari PC-Client 1 ke PC-Client 2, D-ITG yang digunakan merupakan versi 2.80.

3.1.3 Perangkat Virtual

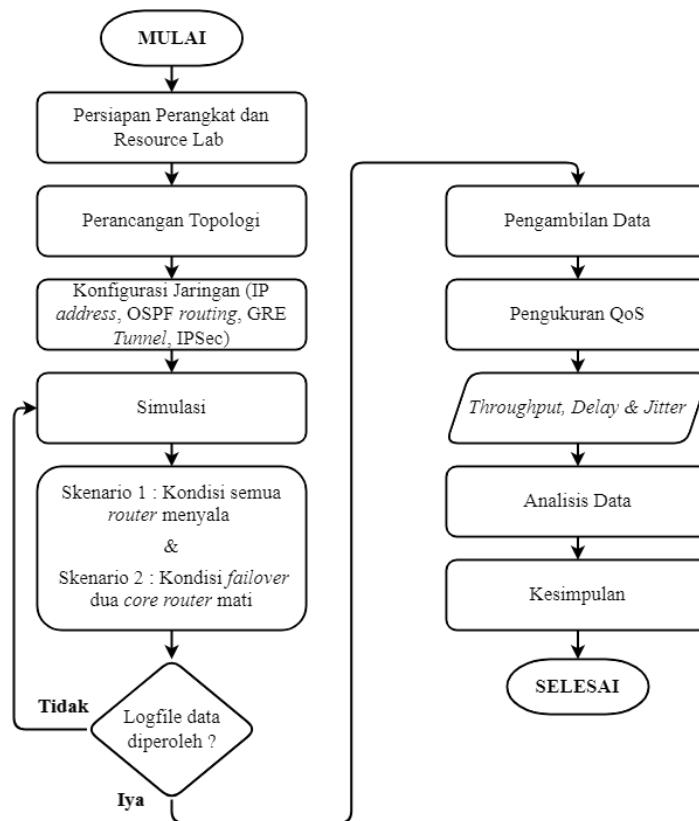
Perangkat virtual yang digunakan dalam penelitian ini yaitu 8 *router* VyOS dan 2 PC *client*. Pada tabel di bawah ini merupakan spesifikasi masing-masing perangkat.

Tabel 3. 2 Spesifikasi Perangkat Virtual

| Perangkat | Tipe | RAM |
|------------------|----------------------------|------------|
| <i>Router</i> | VyOS 1.3 | 512 MB |
| <i>Switch</i> | <i>Ethernet Switch</i> | 256-512 MB |
| <i>PC-Client</i> | <i>Ubuntu Server 14.06</i> | 512 MB |

3.2 ALUR PENELITIAN

Pada penelitian ini dilakukan beberapa tahapan seperti persiapan perangkat dan *resource lab*, perancangan topologi, konfigurasi jaringan, simulasi skenario pertama dan kedua, pengambilan data, pengukuran QoS, analisis data, trakhir kesimpulan dan saran. Di bawah ini merupakan gambar 3.1 *Flowchart* Penelitian.



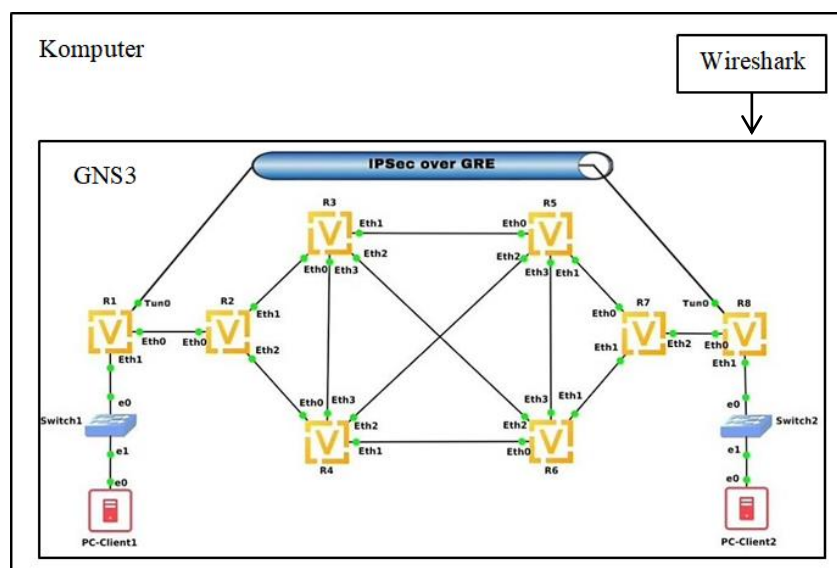
Gambar 3. 1 *Flowchart* Penelitian

Tahap pertama dimulai dengan mempersiapkan alat dan bahan serta perangkat yang dibutuhkan seperti komputer untuk menjalankan *network emulator* GNS3, tahap selanjutnya merancang topologi yang akan digunakan yaitu

topologi Mesh serta melakukan konfigurasi jaringan yang mencakup IP *address*, *routing* OSPF, GRE *Tunnel* dan IPSec. Setelah tahap konfigurasi selesai dilanjutkan dengan tahap simulasi yang dibagi menjadi dua skenario, skenario pertama dilakukan dengan kondisi semua *router* menyala, sedangkan skenario kedua dilakukan dengan kondisi *failover* yang nantinya pada saat proses komunikasi antar *host* berlangsung dua *core router* mati/down, bilamana pada tahap simulasi gagal maka akan kembali pada tahapan konfigurasi untuk melakukan pengecekan konfigurasi jaringan agar pada simulasi berikutnya simulasi dapat berhasil. Pada tahap selanjutnya tahap pengambilan data dengan menggunakan *tools* D-ITG serta data tersebut dapat diamati menggunakan Wireshark. Setelah data yang dibutuhkan pada penelitian ini terkumpul, penelitian ini dapat memasuki tahap analisis dan kesimpulan yang didapat setelah melakukan semua tahapan penelitian.

3.3 TOPOLOGI JARINGAN

Pada penelitian ini menggunakan topologi jaringan model mesh dengan delapan *router* VyOS, dua *ethernet switch* dan dua PC-client Ubuntu *server*. Koneksi antara perangkat di jaringan terkait langsung atau saling berhubungan satu sama lain menggunakan *routing* OSPF *single area* seperti pada gambar 3.2 di bawah ini.



Gambar 3. 2 Topologi Jaringan

Perangkat *router* yang berada di tengah yaitu R3, R4, R5, dan R6 berperan sebagai *core router*, sehingga hal ini mendukung terjadinya *failover* ketika

kondisi *core router* mengalami *down*. Untuk perangkat *router* R1 dan R8 nantinya akan berperan sebagai *tunnel*, hal ini menjadikan komunikasi antar *router* R1 dan R8 berkomunikasi melalui jalur virtual di atas jalur fisik pada *router* lainnya atau disebut *overlay network*. Setiap perangkat pada topologi akan mendapatkan alamat IP pada *interface* yang sudah ditentukan seperti pada tabel 3.3 di bawah.

Tabel 3. 3 IP Address Perangkat

| Perangkat | Port | IP address | Prefix |
|-------------|------|-----------------|--------|
| R1 | Eth0 | 12.12.12.1 | /24 |
| | Eth1 | 192.168.100.1 | /24 |
| | Tun0 | 100.100.100.1 | /24 |
| R2 | Eth0 | 12.12.12.2 | /24 |
| | Eth1 | 23.23.23.2 | /24 |
| | Eth2 | 24.24.24.2 | /24 |
| R3 | Eth0 | 23.23.23.3 | /24 |
| | Eth1 | 35.35.35.3 | /24 |
| | Eth2 | 36.36.36.3 | /24 |
| | Eth3 | 34.34.34.3 | /24 |
| R4 | Eth0 | 24.24.24.4 | /24 |
| | Eth1 | 46.46.46.4 | /24 |
| | Eth2 | 45.45.45.4 | /24 |
| | Eth3 | 34.34.34.4 | /24 |
| R5 | Eth0 | 35.35.35.5 | /24 |
| | Eth1 | 57.57.57.5 | /24 |
| | Eth2 | 45.45.45.5 | /24 |
| | Eth3 | 56.56.56.5 | /24 |
| R6 | Eth0 | 46.46.46.6 | /24 |
| | Eth1 | 67.67.67.6 | /24 |
| | Eth2 | 36.36.36.6 | /24 |
| | Eth3 | 56.56.56.6 | /24 |
| R7 | Eth0 | 57.57.57.7 | /24 |
| | Eth1 | 67.67.67.7 | /24 |
| | Eth2 | 78.78.78.7 | /24 |
| R8 | Eth0 | 78.78.78.8 | /24 |
| | Eth1 | 192.168.200.1 | /24 |
| | Tun0 | 100.100.100.2 | /24 |
| PC-Client 1 | Eth0 | 192.168.100.100 | /24 |
| PC-Client 2 | Eth0 | 192.168.200.100 | /24 |

3.4 KONFIGURASI JARINGAN

Pada penelitian ini ada beberapa konfigurasi yang perlu dilakukan sebelum proses pengujian jaringan dilakukan, konfigurasi tersebut di antaranya yaitu *IP address*, *routing OSPF*, *GRE Tunnel*, dan *IPSec*.

3.4.1 IP Address

Konfigurasi *IP address* dilakukan pada semua perangkat yang ada pada jaringan sesuai tabel 3.3. Perintah yang digunakan ialah *conf* yang berfungsi agar *router* masuk kedalam mode konfigurasi, setelah itu perintah yang digunakan untuk setting *IP address* perangkat ialah *set int eth eth(port yang digunakan) address* (IP yang akan digunakan). Setelah semua *interface* mendapatkan *IP address*, selanjutnya perintah *commit* yang berfungsi untuk menerapkan konfigurasi yang sudah dimasukan. Di bawah ini merupakan tampilan pada *router R3* setelah semua *interface* mendapatkan *IP address*, perintah yang digunakan untuk menampilkan *IP address* yang ada yaitu *show interface*.

```
vyos@vyos# show interface
ethernet eth0 {
    address 23.23.23.3/24
    hw-id 0c:06:ef:7a:00:00
}
ethernet eth1 {
    address 35.35.35.3/24
    hw-id 0c:06:ef:7a:00:01
}
ethernet eth2 {
    address 36.36.36.3/24
    hw-id 0c:06:ef:7a:00:02
}
ethernet eth3 {
    address 34.34.34.3/24
    hw-id 0c:06:ef:7a:00:03
}
```

3.4.2 OSPF Routing

Konfigurasi *routing OSPF* dilakukan agar semua *router* dapat terhubung dan berkomunikasi satu sama lainnya. Perintah yang digunakan ialah *conf* yang berfungsi agar *router* masuk kedalam mode konfigurasi, setelah itu perintah yang digunakan untuk mengkonfigurasi protokol *routing OSPF* yaitu *set protocols ospf*

area (area yang digunakan yaitu 0 atau *singel area*) *network* (IP network yang digunakan). Setelah konfigurasi *routing* OSPF, selanjutnya perintah *commit* yang berfungsi untuk menerapkan konfigurasi yang sudah dimasukkan. Di bawah ini merupakan tampilan pada *router* R3 setelah dilakukan konfigurasi *routing* OSPF.

```
protocols {
  ospf {
    area 0 {
      network 23.23.23.0/24
      network 34.34.34.0/24
      network 36.36.36.0/24
      network 35.35.35.0/24
    }
  }
}
```

3.4.3 GRE *Tunnel*

Konfigurasi GRE *Tunnel* dilakukan karena berperan sebagai jalur virtual yang akan menghubungkan *router* R1 dan *router* R8 secara langsung secara *point to point*. Perintah yang digunakan ialah *conf* yang berfungsi agar *router* masuk kedalam mode konfigurasi, setelah itu perintah *set interfaces tunnel tun0 encapsulation gre* yang berfungsi menyediakan *interface* tun0, lalu perintah *set interface tunnel tun0 source-address* (IP address *router*), selanjutnya perintah *set interfaces tunnel tun0 remote* (IP address *router* tujuan), dan terakhir perintah *set interfaces tunnel tun0 address* (IP address *tunnel*). Di bawah ini merupakan tampilan pada *router* R1 (kiri) dan R8 (kanan) setelah dilakukan konfigurasi GRE *Tunnel*.

```
tunnel tun0 {
  address 100.100.100.1/24
  encapsulation gre
  remote 78.78.78.8
  source-address 12.12.12.1
  source-interface eth0
}
```

```
tunnel tun0 {
  address 100.100.100.2/24
  encapsulation gre
  remote 12.12.12.1
  source-address 78.78.78.8
  source-interface eth0
}
```

3.4.4 IPSec

Konfigurasi IPSec dilakukan pada *router* R1 dan *router* R8 agar komunikasi data yang berlangsung terenkripsi. Ada beberapa tahapan dalam melakukan konfigurasi IPSes pada *router* VyOS, yaitu :

a. IKE Group

Internet Key Exchange (IKE) Group merupakan protokol yang berfungsi dalam pembuatan dan pertukaran *criptographic key* yang digunakan untuk proses enkripsi dan dekripsi paket yang akan dikirimkan.

1. Perintah yang digunakan ialah *set vpn ipsec ipsec-interfaces interface eth(port yang digunakan)*.
2. Perintah *set vpn ipsec ike-group* (nama grup IKE : titan) *proposal 1 dh-group* (nomor grup yang digunakan : 5), perintah ini berperan untuk menentukan algoritma *Diffie-Hellman* yang akan membangkitkan dan menentukan kekuatan kunci yang digunakan dalam proses pertukaran kunci, semakin tinggi nomor *Diffie-Hellman group* maka semakin aman.
3. Perintah *set vpn ipsec ike-group* (nama grup IKE : titan) *proposal 1 encryption* (tipe enkripsi yang digunakan : aes128), perintah ini berperan untuk menentukan algoritma kriptografi yang akan digunakan yaitu *Advanced Encryption Standard (AES)* yang termasuk jenis kriptografi simetris di mana pada proses enkripsi dan dekripsi menggunakan kunci yang sama.
4. Perintah *set vpn ipsec ike-group* (nama grup IKE : titan) *proposal 1 hash* (tipe HMAC yang digunakan : sha1), perintah ini berperan untuk menentukan algoritma *Hash Messages Authentication Codes (HMAC)* yang berfungsi untuk memastikan data tidak berubah selama proses transmisi berlangsung.

b. ESP Group

Encapsulating Security Payload (ESP) Group merupakan salah satu protokol IPSec yang berfungsi untuk enkripsi data dengan protokol

IP bernilai 50, nilai pada protokol IP tersebut berperan sebagai identitas *header* pada paket data yang ditransmisikan, sehingga *header* dapat dikenali bahwa *header* tersebut berasal dari ESP.

1. Perintah yang digunakan ialah *set vpn ipsec esp-group* (nama grup ESP : titan) *proposal 1 encryption aes128*.
2. Perintah *set vpn ipsec esp-group* (nama grup ESP : titan) *proposal 1 hash sha1*.

c. IPsec site to site

Konfigurasi tahap ini dilakukan agar IPsec dapat mengenali atau autentikasi *router* yang akan dituju untuk membagikan *secret key* sehingga proses komunikasi menggunakan IPsec antar *router* dapat dilakukan.

1. Perintah yang digunakan ialah *set vpn ipsec site-to-site peer* (IP address router tujuan) *authentication mode pre-shared-secret*.
2. Perintah *set vpn ipsec site-to-site peer* (IP address router tujuan) *authentication pre-shared-secret* (*secret key* yang digunakan : titan), perintah ini berfungsi untuk menentukan *secret key* sehingga penerima IPsec dapat otentikasi paket yang dikirim oleh pengirim.
3. Perintah *set vpn ipsec site-to-site peer* (IP address router tujuan) *ike-group* (nama grup IKE : titan).
4. Perintah *set vpn ipsec site-to-site peer* (IP address router tujuan) *default-esp-group* (nama grup ESP : titan).
5. Perintah *set vpn ipsec site-to-site peer* (IP address router tujuan) *local-address* (IP address router).
6. Perintah *set vpn ipsec site-to-site peer* (IP address router tujuan) *tunnel 0 protocol gre*.

Setelah semua proses konfigurasi selesai, maka dapat diperiksa konfigurasinya menggunakan perintah *show interface*. Di bawah ini merupakan tampilan pada *router* R1 (kiri) dan *router* R8 (kanan) setelah dilakukan konfigurasi IPsec.

| | |
|---|---|
| <pre> vpn { ipsec { esp-group titan { proposal 1 { encryption aes128 hash sha1 } } ike-group titan { proposal 1 { dh-group 5 encryption aes128 hash sha1 } } } } site-to-site { peer 78.78.78.8 { authentication { mode pre-shared-secret pre-shared-secret titan } default-esp-group titan ike-group titan local-address 12.12.12.1 tunnel 0 { protocol gre } } } </pre> | <pre> vpn { ipsec { esp-group titan { proposal 1 { encryption aes128 hash sha1 } } ike-group titan { proposal 1 { dh-group 5 encryption aes128 hash sha1 } } } } site-to-site { peer 12.12.12.1 { authentication { mode pre-shared-secret pre-shared-secret titan } default-esp-group titan ike-group titan local-address 78.78.78.8 tunnel 0 { protocol gre } } } </pre> |
|---|---|

3.5 PROSES PENGUJIAN

Pada proses pengujian penelitian ini dibagi menjadi beberapa tahapan, yaitu pengecekan awal, skenario pada kondisi semua *router* menyala dan skenario pada kondisi *failover* dengan dua *core router* mati/*down*.

3.5.1 Pengecekan Awal Jaringan

Pengecekan awal ialah melakukan melakukan *ping* atau megirmkan paket *Internet Communication Message Protocol (ICMP)* dari *PC-Client 1* ke *PC-Client 2*. Di bawah ini merupakan gambar 3.3 yang menunjukkan proses *ping* berhasil ditandai dengan mendapatkan *reply* dari 192.168.200.100 yang merupakan *IP address PC-Client 2* itu sendiri.

```

root@titan:~# ping 192.168.200.100
PING 192.168.200.100 (192.168.200.100) 56(84) bytes of data.
64 bytes from 192.168.200.100: icmp_seq=1 ttl=62 time=3.24 ms
64 bytes from 192.168.200.100: icmp_seq=2 ttl=62 time=36.5 ms
64 bytes from 192.168.200.100: icmp_seq=3 ttl=62 time=8.73 ms
64 bytes from 192.168.200.100: icmp_seq=4 ttl=62 time=8.55 ms

```

Gambar 3. 3 Pengujian Ping PC-Client 1 ke PC-Client 2

Selain itu untuk mengetahui komunikasi paket dilakukan melalui jalur *interface tunnel* dapat melakukan pengecekan pada PC-Client 1 dengan memasukan perintah *traceroute* 192.168.200.100, perintah ini berfungsi untuk melacak jalur mana yang digunakan untuk mencapai PC-Client 2 dengan IP address 192.168.200.100. Di bawah ini merupakan gambar 3.4 yang menjukan proses komunikasi paket berhasil melewati jalur *interface tunnel* dengan IP address tunnel 100.100.100.2.

```

root@titan:~# traceroute 192.168.200.100
traceroute to 192.168.200.100 (192.168.200.100), 30 hops max, 60 byte packets
 1 192.168.100.1 (192.168.100.1)  1.090 ms  0.718 ms  0.561 ms
 2 100.100.100.2 (100.100.100.2) 7.129 ms  6.824 ms  6.510 ms
 3 192.168.200.100 (192.168.200.100) 6.367 ms  6.039 ms  5.702 ms

```

Gambar 3. 4 Traceroute PC-Client 1 ke PC-Client 2

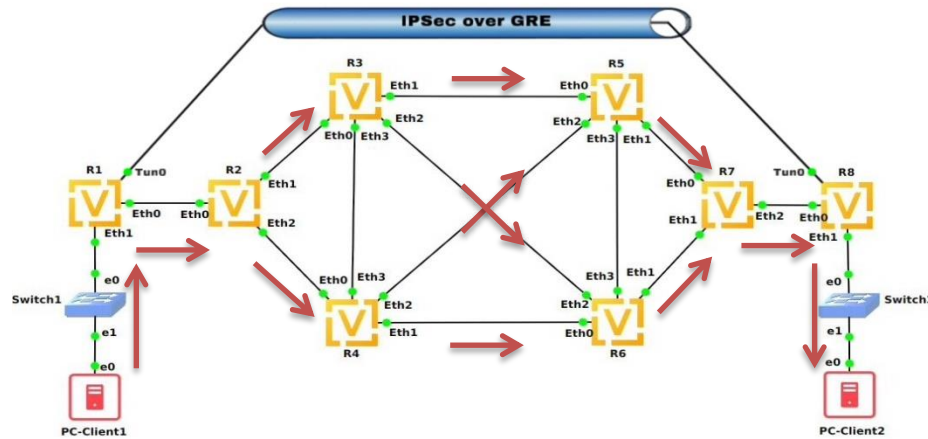
Selanjutnya untuk mengetahui komunikasi paket yang melalui *tunnel* telah terimplementasi IPsec dapat melakukan *capture* paket saat sedang melakukan *ping* dengan menggunakan perangkat lunak wireshark. Di bawah ini merupakan gambar 3.5 yang menunjukkan proses pertukaran data telah berhasil yang ditandai dengan tidak dapat terbacanya jenis protokol yang dikirimkan serta panjang paket yang dikirimkan, dapat disimpulkan bahwa paket telah terenkripsi oleh IPsec sehingga data akan dijamin kerahasiaannya.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-------------|------------|-------------|----------|--------|----------------------|
| 876 | 1000.760075 | 78.78.78.8 | 12.12.12.1 | ESP | 182 | ESP (SPI=0xc434bbb0) |
| 877 | 1001.761228 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc37643e5) |
| 878 | 1001.763284 | 78.78.78.8 | 12.12.12.1 | ESP | 182 | ESP (SPI=0xc434bbb0) |
| 879 | 1002.762813 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc37643e5) |
| 880 | 1002.764720 | 78.78.78.8 | 12.12.12.1 | ESP | 182 | ESP (SPI=0xc434bbb0) |
| 881 | 1003.765072 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc37643e5) |
| 882 | 1003.767052 | 78.78.78.8 | 12.12.12.1 | ESP | 182 | ESP (SPI=0xc434bbb0) |
| 883 | 1004.767126 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc37643e5) |
| 884 | 1004.769297 | 78.78.78.8 | 12.12.12.1 | ESP | 182 | ESP (SPI=0xc434bbb0) |
| 885 | 1005.768249 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc37643e5) |
| 886 | 1005.770311 | 78.78.78.8 | 12.12.12.1 | ESP | 182 | ESP (SPI=0xc434bbb0) |
| 887 | 1006.770093 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc37643e5) |
| 888 | 1006.772087 | 78.78.78.8 | 12.12.12.1 | ESP | 182 | ESP (SPI=0xc434bbb0) |
| 889 | 1007.771679 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc37643e5) |
| 890 | 1007.773746 | 78.78.78.8 | 12.12.12.1 | ESP | 182 | ESP (SPI=0xc434bbb0) |

Gambar 3. 5 Packet Capture Menggunakan Wireshark

3.5.2 Skenario 1

Pada skenario pertama pengujian GRE Tunnel berbasis IPsec dilakukan pada saat semua *router* pada jaringan dalam keadaan aktif dan normal, dengan ini proses komunikasi dari *PC-Client 1* ke *PC-Client 2* akan memiliki banyak opsi jalur yang berbeda untuk mencapai *router* tujuan yang disebut *redundancy*. Ilustrasi untuk skenario pertama seperti gambar 3. 11 di bawah ini.



Gambar 3. 6 Ilustrasi Skenario Pertama

Di bawah ini merupakan tampilan tabel *routing* dengan perintah *run show ip route* yang menampilkan opsi jalur yang digunakan oleh *router* pada saat berkomunikasi.

```
vyos@vyos# run 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, D - SHARP,
F - PBR, f - OpenFabric,
> - selected route, * - FIB route, q - queued, r - rejected, b - backup

O 12.12.12.0/24 [110/1] is directly connected, eth0, weight 1, 00:10:23
C>* 12.12.12.0/24 is directly connected, eth0, 00:10:25
O 23.23.23.0/24 [110/1] is directly connected, eth1, weight 1, 00:10:23
C>* 23.23.23.0/24 is directly connected, eth1, 00:10:26
O 24.24.24.0/24 [110/1] is directly connected, eth2, weight 1, 00:10:23
C>* 24.24.24.0/24 is directly connected, eth2, 00:10:25
O>* 34.34.34.0/24 [110/2] via 23.23.23.3, eth1, weight 1, 00:09:32
* via 24.24.24.4, eth2, weight 1, 00:09:32
O>* 35.35.35.0/24 [110/2] via 23.23.23.3, eth1, weight 1, 00:09:32
O>* 36.36.36.0/24 [110/2] via 23.23.23.3, eth1, weight 1, 00:09:32
O>* 45.45.45.0/24 [110/2] via 24.24.24.4, eth2, weight 1, 00:09:37
O>* 46.46.46.0/24 [110/2] via 24.24.24.4, eth2, weight 1, 00:09:37
O>* 56.56.56.0/24 [110/3] via 23.23.23.3, eth1, weight 1, 00:09:32
* via 24.24.24.4, eth2, weight 1, 00:09:32
O>* 57.57.57.0/24 [110/3] via 23.23.23.3, eth1, weight 1, 00:09:32
* via 24.24.24.4, eth2, weight 1, 00:09:32
O>* 67.67.67.0/24 [110/3] via 23.23.23.3, eth1, weight 1, 00:09:32
* via 24.24.24.4, eth2, weight 1, 00:09:32
O>* 78.78.78.0/24 [110/4] via 23.23.23.3, eth1, weight 1, 00:09:32
* via 24.24.24.4, eth2, weight 1, 00:09:32
O>* 192.168.100.0/24 [110/2] via 12.12.12.1, eth0, weight 1, 00:09:30
O>* 192.168.200.0/24 [110/5] via 23.23.23.3, eth1, weight 1, 00:09:29
* via 24.24.24.4, eth2, weight 1, 00:09:29
```

Gambar 3. 7 Tabel *Routing* Skenario Pertama Pada *Router* R2

Dapat dilihat dari gambar 3.7 di atas yang merupakan tabel *routing* pada *router* R2 pada kondisi skenario pertama, rute *gateway* dengan alamat IP 34.34.34.0, IP 56.56.56.0, IP 57.57.57.0, IP 67.67.67.0, dan IP 78.78.78.0 dapat diakses melalui dua rute yaitu melalui alamat IP 23.23.23.3 dengan *port eth1* dan alamat IP 24.24.24.4 dengan *port eth2*. Hal ini dikarenakan pada skenario pertama semua *router* berjalan dengan normal, sehingga memiliki dua opsi rute untuk proses komunikasi.

```

vyos@vyos# run 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, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup

O>* 12.12.12.0/24 [110/4] via 57.57.57.5, eth0, weight 1, 00:00:39
   *                via 67.67.67.6, eth1, weight 1, 00:00:39
O>* 23.23.23.0/24 [110/3] via 57.57.57.5, eth0, weight 1, 00:00:09
   *                via 67.67.67.6, eth1, weight 1, 00:00:09
O>* 24.24.24.0/24 [110/3] via 57.57.57.5, eth0, weight 1, 00:00:39
   *                via 67.67.67.6, eth1, weight 1, 00:00:39
O>* 34.34.34.0/24 [110/3] via 57.57.57.5, eth0, weight 1, 00:00:39
   *                via 67.67.67.6, eth1, weight 1, 00:00:39
O>* 35.35.35.0/24 [110/2] via 57.57.57.5, eth0, weight 1, 00:22:49
O>* 36.36.36.0/24 [110/2] via 67.67.67.6, eth1, weight 1, 00:00:39
O>* 45.45.45.0/24 [110/2] via 57.57.57.5, eth0, weight 1, 00:22:49
O>* 46.46.46.0/24 [110/2] via 67.67.67.6, eth1, weight 1, 00:00:39
O>* 56.56.56.0/24 [110/2] via 57.57.57.5, eth0, weight 1, 00:00:39
   *                via 67.67.67.6, eth1, weight 1, 00:00:39
O  57.57.57.0/24 [110/1] is directly connected, eth0, weight 1, 00:23:39
C>* 57.57.57.0/24 is directly connected, eth0, 00:23:40
O  67.67.67.0/24 [110/1] is directly connected, eth1, weight 1, 00:23:39
C>* 67.67.67.0/24 is directly connected, eth1, 00:23:41
O  78.78.78.0/24 [110/1] is directly connected, eth2, weight 1, 00:23:39
C>* 78.78.78.0/24 is directly connected, eth2, 00:23:40
O>* 192.168.100.0/24 [110/5] via 57.57.57.5, eth0, weight 1, 00:00:39
   *                via 67.67.67.6, eth1, weight 1, 00:00:39
O>* 192.168.200.0/24 [110/2] via 78.78.78.8, eth2, weight 1, 00:22:45

```

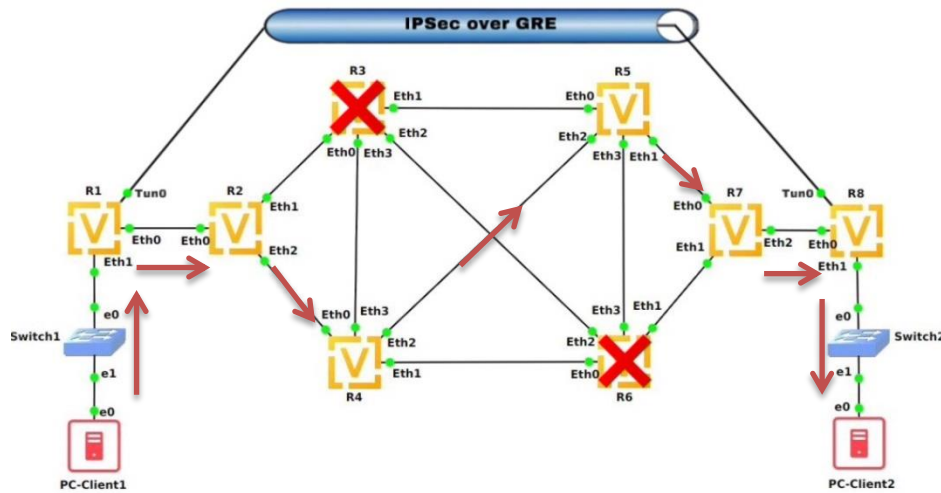
Gambar 3. 8 Tabel *Routing* Skenario Pertama Pada *Router* R7

Dapat dilihat dari gambar 3.8 di atas yang merupakan tabel *routing* pada *router* R7 pada kondisi skenario pertama, rute *gateway* dengan alamat IP 12.12.12.0, IP 23.23.23.0, IP 24.24.24.0, IP 34.34.34.0, dan IP 56.56.56.0 dapat diakses melalui dua rute yaitu melalui alamat IP 57.57.57.5 dengan *port eth0* dan alamat IP 67.67.67.6 dengan *port eth1*. Hal ini dikarenakan pada skenario pertama semua *router* berjalan dengan normal, sehingga memiliki dua opsi rute untuk proses komunikasi.

3.5.3 Skenario 2

Pada skenario kedua pengujian *GRE Tunnel* berbasis *IPSec* dilakukan pada kondisi dua *core router* mengalami *down* ketika proses pengiriman data berlangsung, sehingga pada kondisi ini terjadi konsep *failover* dengan berpindahannya jalur komunikasi dari *PC-Client 1* ke *PC-Client 2* secara otomatis melalui jalur *backup/cadangan* dengan *router* yang tersedia.

Gambar 3.9 di bawah ini merupakan ilustrasi untuk skenario kedua ketika router R3 dan R6 mengalami *down*.



Gambar 3. 9 Ilustrasi Skenario Kedua *Router R3 dan R6 Down*

Di bawah ini merupakan tampilan tabel *routing* setelah *router R3 dan R6* mengalami *down*.

```

vyos@vyos# run 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, D - SHARP,
F - PBR, f - OpenFabric,
> - selected route, * - FIB route, q - queued, r - rejected, b - backup

0 12.12.12.0/24 [110/1] is directly connected, eth0, weight 1, 00:15:06
C>* 12.12.12.0/24 is directly connected, eth0, 00:15:08
0 23.23.23.0/24 [110/1] is directly connected, eth1, weight 1, 00:00:05
C>* 23.23.23.0/24 is directly connected, eth1, 00:15:09
0 24.24.24.0/24 [110/1] is directly connected, eth2, weight 1, 00:15:06
C>* 24.24.24.0/24 is directly connected, eth2, 00:15:08
O>* 34.34.34.0/24 [110/2] via 24.24.24.4, eth2, weight 1, 00:00:08
O>* 35.35.35.0/24 [110/3] via 24.24.24.4, eth2, weight 1, 00:00:08
O>* 45.45.45.0/24 [110/2] via 24.24.24.4, eth2, weight 1, 00:14:20
O>* 46.46.46.0/24 [110/2] via 24.24.24.4, eth2, weight 1, 00:14:20
O>* 56.56.56.0/24 [110/3] via 24.24.24.4, eth2, weight 1, 00:00:08
O>* 57.57.57.0/24 [110/3] via 24.24.24.4, eth2, weight 1, 00:00:08
O>* 67.67.67.0/24 [110/4] via 24.24.24.4, eth2, weight 1, 00:00:06
O>* 78.78.78.0/24 [110/4] via 24.24.24.4, eth2, weight 1, 00:00:08
O>* 192.168.100.0/24 [110/2] via 12.12.12.1, eth0, weight 1, 00:14:13
O>* 192.168.200.0/24 [110/5] via 24.24.24.4, eth2, weight 1, 00:00:08

```

Gambar 3. 10 Tabel *Routing* Skenario Kedua Pada *Router R2* Ketika *R3 dan R6 Down*

Dapat dilihat dari gambar 3.10 di atas yang merupakan tabel *routing* pada *router R2* dengan kondisi *router R3 dan R6 down*, semua alamat IP *gateway* hanya dapat diakses melalui satu rute yaitu melalui alamat IP 24.24.24.4 dengan *port eth2*. Hal ini dikarenakan pada skenario kedua *router R3* mengalami *down* sehingga hanya ada satu rute yang dapat dilewati untuk proses komunikasi yaitu *router R4*.

```

vyos@vyos# run 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, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup

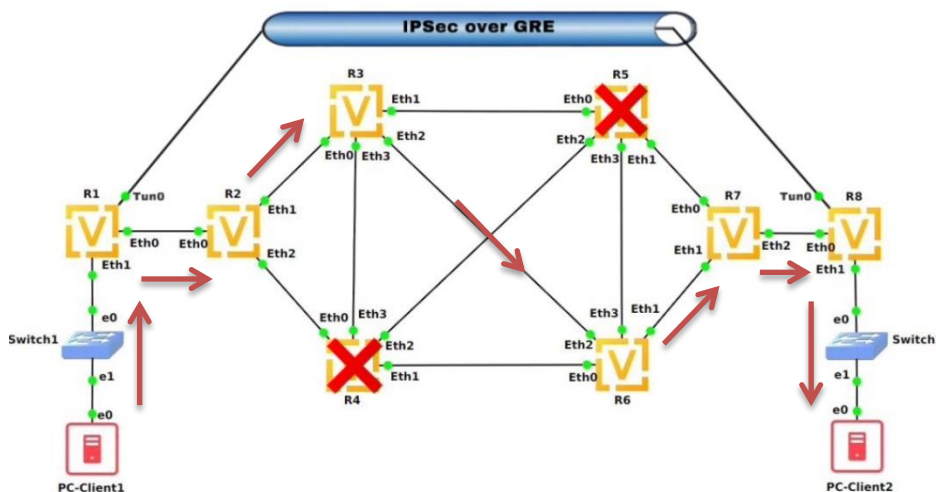
O>* 12.12.12.0/24 [110/4] via 57.57.57.5, eth0, weight 1, 00:01:55
O>* 23.23.23.0/24 [110/4] via 57.57.57.5, eth0, weight 1, 00:01:55
O>* 24.24.24.0/24 [110/3] via 57.57.57.5, eth0, weight 1, 00:01:55
O>* 34.34.34.0/24 [110/3] via 57.57.57.5, eth0, weight 1, 00:01:55
O>* 35.35.35.0/24 [110/2] via 57.57.57.5, eth0, weight 1, 00:16:05
O>* 45.45.45.0/24 [110/2] via 57.57.57.5, eth0, weight 1, 00:16:05
O>* 46.46.46.0/24 [110/3] via 57.57.57.5, eth0, weight 1, 00:01:55
O>* 56.56.56.0/24 [110/2] via 57.57.57.5, eth0, weight 1, 00:01:55
O 57.57.57.0/24 [110/1] is directly connected, eth0, weight 1, 00:16:55
C>* 57.57.57.0/24 is directly connected, eth0, 00:16:56
O 67.67.67.0/24 [110/1] is directly connected, eth1, weight 1, 00:16:55
C>* 67.67.67.0/24 is directly connected, eth1, 00:16:57
O 78.78.78.0/24 [110/1] is directly connected, eth2, weight 1, 00:16:55
C>* 78.78.78.0/24 is directly connected, eth2, 00:16:56
O>* 192.168.100.0/24 [110/5] via 57.57.57.5, eth0, weight 1, 00:01:55
O>* 192.168.200.0/24 [110/2] via 78.78.78.8, eth2, weight 1, 00:16:01

```

Gambar 3. 11 Tabel *Routing* Skenario Kedua Pada *Router* R7 Ketika R3 dan R6 *Down*

Dapat dilihat dari gambar 3.11 di atas yang merupakan tabel *routing* pada *router* R7 dengan kondisi *router* R3 dan R6 *down*, semua alamat IP *gateway* hanya dapat diakses melalui satu rute yaitu melalui alamat IP 57.57.57.5 dengan *port eth0*. Hal ini dikarenakan pada skenario kedua *router* R6 mengalami *down* sehingga hanya ada satu rute yang dapat dilewati untuk proses komunikasi yaitu *router* R5.

Gambar 3.12 di bawah ini merupakan ilustrasi untuk skenario kedua ketika *router* R4 dan R5 mengalami *down*.



Gambar 3. 12 Ilustrasi Skenario Kedua *Router* R4 dan R5 *Down*

Di bawah ini merupakan tampilan tabel *routing* setelah *router* R4 dan R5 mengalami *down*.

```

vyos@vyos:~$ 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, D - SHARP,
        F - PBR, f - OpenFabric,
        > - selected route, * - FIB route, q - queued, r - rejected, b - backup

12.12.12.0/24 [110/1] is directly connected, eth0, weight 1, 00:23:04
>* 12.12.12.0/24 is directly connected, eth0, 00:23:05
23.23.23.0/24 [110/1] is directly connected, eth1, weight 1, 00:23:04
>* 23.23.23.0/24 is directly connected, eth1, 00:23:06
24.24.24.0/24 [110/1] is directly connected, eth2, weight 1, 00:01:15
>* 24.24.24.0/24 is directly connected, eth2, 00:23:05
>* 34.34.34.0/24 [110/2] via 23.23.23.3, eth1, weight 1, 00:01:24
>* 35.35.35.0/24 [110/2] via 23.23.23.3, eth1, weight 1, 00:22:19
>* 36.36.36.0/24 [110/2] via 23.23.23.3, eth1, weight 1, 00:22:19
>* 45.45.45.0/24 [110/3] via 23.23.23.3, eth1, weight 1, 00:01:14
>* 56.56.56.0/24 [110/3] via 23.23.23.3, eth1, weight 1, 00:01:24
>* 57.57.57.0/24 [110/3] via 23.23.23.3, eth1, weight 1, 00:01:24
>* 67.67.67.0/24 [110/4] via 23.23.23.3, eth1, weight 1, 00:01:24
>* 78.78.78.0/24 [110/4] via 23.23.23.3, eth1, weight 1, 00:01:24
>* 192.168.100.0/24 [110/2] via 12.12.12.1, eth0, weight 1, 00:22:12
>* 192.168.200.0/24 [110/5] via 23.23.23.3, eth1, weight 1, 00:01:24
vyos@vyos:~$

```

Gambar 3. 13 Tabel *Routing* Skenario Kedua Pada *Router R2* Ketika *R4* dan *R5 Down*

Dapat dilihat dari gambar 3.13 di atas yang merupakan tabel *routing* pada *router R2* dengan kondisi *router R4* dan *R5 down*, semua alamat IP *gateway* hanya dapat diakses melalui satu rute yaitu melalui alamat IP 23.23.23.3 dengan *port eth1*. Hal ini dikarenakan pada skenario kedua *router R4* mengalami *down* sehingga hanya ada satu rute yang dapat dilewati untuk proses komunikasi yaitu *router R3*.

```

vyos@vyos:~$ 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, D - SHARP,
        F - PBR, f - OpenFabric,
        > - selected route, * - FIB route, q - queued, r - rejected, b - backup

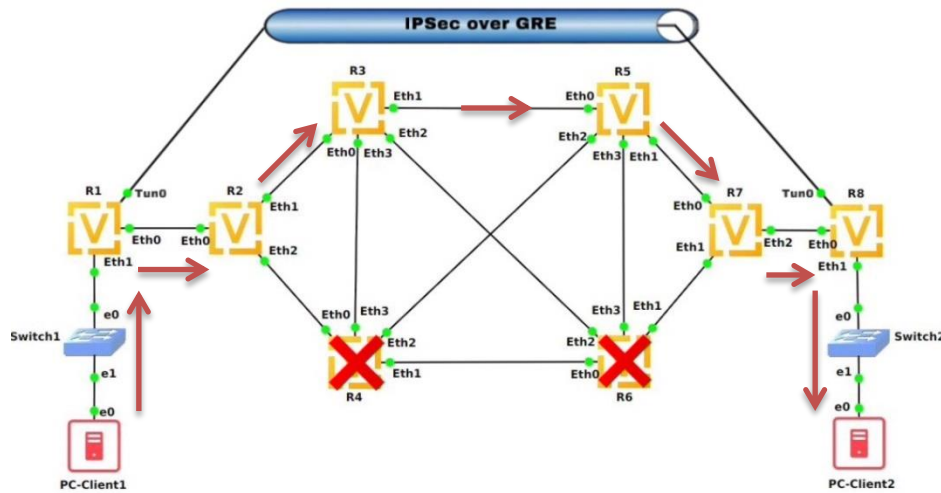
>* 12.12.12.0/24 [110/4] via 67.67.67.6, eth1, weight 1, 00:00:37
>* 23.23.23.0/24 [110/4] via 67.67.67.6, eth1, weight 1, 00:00:37
>* 24.24.24.0/24 [110/3] via 67.67.67.6, eth1, weight 1, 00:00:37
>* 34.34.34.0/24 [110/3] via 67.67.67.6, eth1, weight 1, 00:00:37
>* 36.36.36.0/24 [110/2] via 67.67.67.6, eth1, weight 1, 00:02:58
>* 45.45.45.0/24 [110/3] via 67.67.67.6, eth1, weight 1, 00:00:37
>* 46.46.46.0/24 [110/2] via 67.67.67.6, eth1, weight 1, 00:02:58
>* 56.56.56.0/24 [110/2] via 67.67.67.6, eth1, weight 1, 00:00:37
>* 57.57.57.0/24 [110/1] is directly connected, eth0, weight 1, 00:29:56
>* 57.57.57.0/24 is directly connected, eth0, 00:29:58
67.67.67.0/24 [110/1] is directly connected, eth1, weight 1, 00:29:56
>* 67.67.67.0/24 is directly connected, eth1, 00:29:59
78.78.78.0/24 [110/1] is directly connected, eth2, weight 1, 00:29:56
>* 78.78.78.0/24 is directly connected, eth2, 00:29:57
>* 192.168.100.0/24 [110/5] via 67.67.67.6, eth1, weight 1, 00:00:37
>* 192.168.200.0/24 [110/2] via 78.78.78.8, eth2, weight 1, 00:29:04
vyos@vyos:~$

```

Gambar 3. 14 Tabel *Routing* Skenario Kedua Pada *Router R7* Ketika *R4* dan *R5 Down*

Dapat dilihat dari gambar 3.14 di atas yang merupakan tabel *routing* pada *router R7* dengan kondisi *router R4* dan *R5 down*, semua alamat IP *gateway* hanya dapat diakses melalui satu rute yaitu melalui alamat IP 67.67.67.6 dengan *port eth1*. Hal ini dikarenakan pada skenario kedua *router R5* mengalami *down* sehingga hanya ada satu rute yang dapat dilewati untuk proses komunikasi yaitu *router R6*.

Gambar 3.15 di bawah ini merupakan ilustrasi untuk skenario kedua ketika router R4 dan R6 mengalami *down*.



Gambar 3. 15 Ilustrasi Skenario Kedua *Router R4 dan R6 Down*

Di bawah ini merupakan tampilan tabel *routing* setelah *router R4 dan R6* mengalami *down*.

```
yos@vyos:~$ 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, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup

 12.12.12.0/24 [110/1] is directly connected, eth0, weight 1, 00:23:04
 >* 12.12.12.0/24 is directly connected, eth0, 00:23:05
 23.23.23.0/24 [110/1] is directly connected, eth1, weight 1, 00:23:04
 >* 23.23.23.0/24 is directly connected, eth1, 00:23:06
 24.24.24.0/24 [110/1] is directly connected, eth2, weight 1, 00:01:15
 >* 24.24.24.0/24 is directly connected, eth2, 00:23:05
 >* 34.34.34.0/24 [110/2] via 23.23.23.3, eth1, weight 1, 00:01:24
 >* 35.35.35.0/24 [110/2] via 23.23.23.3, eth1, weight 1, 00:22:19
 >* 36.36.36.0/24 [110/2] via 23.23.23.3, eth1, weight 1, 00:22:19
 >* 45.45.45.0/24 [110/3] via 23.23.23.3, eth1, weight 1, 00:01:14
 >* 56.56.56.0/24 [110/3] via 23.23.23.3, eth1, weight 1, 00:01:24
 >* 57.57.57.0/24 [110/3] via 23.23.23.3, eth1, weight 1, 00:01:24
 >* 67.67.67.0/24 [110/4] via 23.23.23.3, eth1, weight 1, 00:01:24
 >* 78.78.78.0/24 [110/4] via 23.23.23.3, eth1, weight 1, 00:01:24
 >* 192.168.100.0/24 [110/2] via 12.12.12.1, eth0, weight 1, 00:22:12
 >* 192.168.200.0/24 [110/5] via 23.23.23.3, eth1, weight 1, 00:01:24
 yos@vyos:~$
```

Gambar 3. 16 Tabel *Routing* Skenario Kedua Pada *Router R2* Ketika *R4 dan R6 Down*

Dapat dilihat dari gambar 3.16 di atas yang merupakan tabel *routing* pada *router R2* dengan kondisi *router R4 dan R6 down*, semua alamat IP *gateway* hanya dapat diakses melalui satu rute yaitu melalui alamat IP 23.23.23.3 dengan *port eth1*. Hal ini dikarenakan pada skenario kedua *router R4* mengalami *down* sehingga hanya ada satu rute yang dapat dilewati untuk proses komunikasi yaitu *router R3*.


```

vyos@vyos:~$ 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, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup

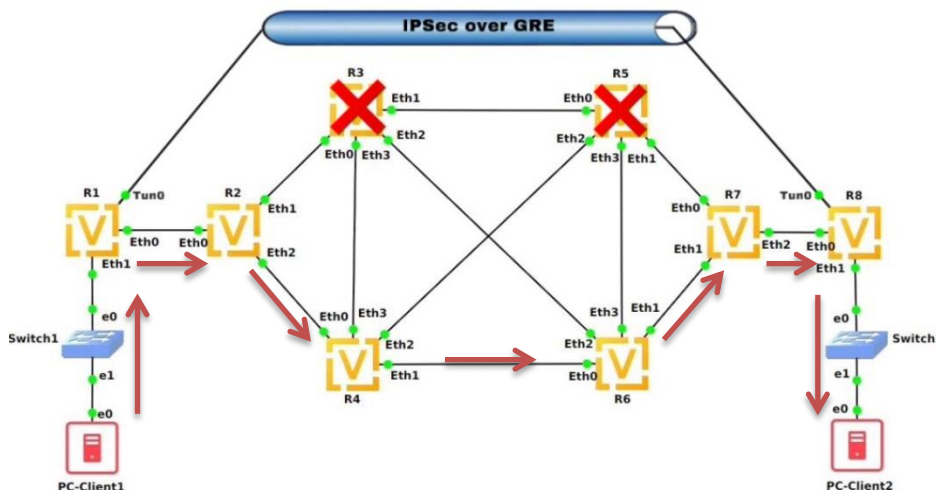
0>* 12.12.12.0/24 [110/4] via 57.57.57.5, eth0, weight 1, 00:04:16
0>* 23.23.23.0/24 [110/3] via 57.57.57.5, eth0, weight 1, 00:04:16
0>* 24.24.24.0/24 [110/4] via 57.57.57.5, eth0, weight 1, 00:04:16
0>* 34.34.34.0/24 [110/3] via 57.57.57.5, eth0, weight 1, 00:04:16
0>* 35.35.35.0/24 [110/2] via 57.57.57.5, eth0, weight 1, 00:25:11
0>* 36.36.36.0/24 [110/3] via 57.57.57.5, eth0, weight 1, 00:04:16
0>* 45.45.45.0/24 [110/2] via 57.57.57.5, eth0, weight 1, 00:04:10
0>* 56.56.56.0/24 [110/2] via 57.57.57.5, eth0, weight 1, 00:04:16
0 57.57.57.0/24 [110/1] is directly connected, eth0, weight 1, 00:25:55
C> 57.57.57.0/24 is directly connected, eth0, 00:25:57
0 67.67.67.0/24 [110/1] is directly connected, eth1, weight 1, 00:25:55
C> 67.67.67.0/24 is directly connected, eth1, 00:25:58
0 78.78.78.0/24 [110/1] is directly connected, eth2, weight 1, 00:25:55
C> 78.78.78.0/24 is directly connected, eth2, 00:25:56
0>* 192.168.100.0/24 [110/5] via 57.57.57.5, eth0, weight 1, 00:04:16
0>* 192.168.200.0/24 [110/2] via 78.78.78.8, eth2, weight 1, 00:25:03

```

Gambar 3. 17 Tabel *Routing* Skenario Kedua Pada *Router* R7 Ketika R4 dan R6 Down

Dapat dilihat dari gambar 3.17 di atas yang merupakan tabel *routing* pada *router* R7 dengan kondisi *router* R4 dan R6 *down*, semua alamat IP *gateway* hanya dapat diakses melalui satu rute yaitu melalui alamat IP 57.57.57.5 dengan *port eth0*. Hal ini dikarenakan pada skenario kedua *router* R6 mengalami *down* sehingga hanya ada satu rute yang dapat dilewati untuk proses komunikasi yaitu *router* R5.

Gambar 3.18 di bawah ini merupakan ilustrasi untuk skenario kedua ketika *router* R3 dan R5 mengalami *down*.



Gambar 3. 18 Ilustrasi Skenario Kedua *Router* R3 dan R5 Down

Di bawah ini merupakan tampilan tabel *routing* setelah *router* R3 dan R5 mengalami *down*.

```

vyos@vyos:~$ 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, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup

0 12.12.12.0/24 [110/1] is directly connected, eth0, weight 1, 00:29:26
C>* 12.12.12.0/24 is directly connected, eth0, 00:29:27
0 23.23.23.0/24 [110/1] is directly connected, eth1, weight 1, 00:00:05
C>* 23.23.23.0/24 is directly connected, eth1, 00:29:28
0 24.24.24.0/24 [110/1] is directly connected, eth2, weight 1, 00:07:37
C>* 24.24.24.0/24 is directly connected, eth2, 00:29:27
O>* 34.34.34.0/24 [110/2] via 24.24.24.4, eth2, weight 1, 00:00:07
O>* 36.36.36.0/24 [110/3] via 24.24.24.4, eth2, weight 1, 00:00:07
O>* 45.45.45.0/24 [110/2] via 24.24.24.4, eth2, weight 1, 00:02:27
O>* 46.46.46.0/24 [110/2] via 24.24.24.4, eth2, weight 1, 00:02:27
O>* 56.56.56.0/24 [110/3] via 24.24.24.4, eth2, weight 1, 00:00:07
O>* 57.57.57.0/24 [110/4] via 24.24.24.4, eth2, weight 1, 00:00:05
O>* 67.67.67.0/24 [110/3] via 24.24.24.4, eth2, weight 1, 00:00:07
O>* 78.78.78.0/24 [110/4] via 24.24.24.4, eth2, weight 1, 00:00:07
O>* 192.168.100.0/24 [110/2] via 12.12.12.1, eth0, weight 1, 00:28:34
O>* 192.168.200.0/24 [110/5] via 24.24.24.4, eth2, weight 1, 00:00:07
vyos@vyos:~$

```

Gambar 3. 19 Tabel *Routing* Skenario Kedua Pada *Router R2* Ketika *R3* dan *R5 Down*

Dapat dilihat dari gambar 3.19 di atas yang merupakan tabel *routing* pada *router R2* dengan kondisi *router R3* dan *R5 down*, semua alamat IP *gateway* hanya dapat diakses melalui satu rute yaitu melalui alamat IP 24.24.24.4 dengan *port eth2*. Hal ini dikarenakan pada skenario kedua *router R3* mengalami *down* sehingga hanya ada satu rute yang dapat dilewati untuk proses komunikasi yaitu *router R4*.

```

vyos@vyos:~$ 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, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup

>* 12.12.12.0/24 [110/4] via 67.67.67.6, eth1, weight 1, 00:00:37
>* 23.23.23.0/24 [110/4] via 67.67.67.6, eth1, weight 1, 00:00:37
>* 24.24.24.0/24 [110/3] via 67.67.67.6, eth1, weight 1, 00:00:37
>* 34.34.34.0/24 [110/3] via 67.67.67.6, eth1, weight 1, 00:00:37
>* 36.36.36.0/24 [110/2] via 67.67.67.6, eth1, weight 1, 00:02:58
>* 45.45.45.0/24 [110/3] via 67.67.67.6, eth1, weight 1, 00:00:37
>* 46.46.46.0/24 [110/2] via 67.67.67.6, eth1, weight 1, 00:02:58
>* 56.56.56.0/24 [110/2] via 67.67.67.6, eth1, weight 1, 00:00:37
>* 57.57.57.0/24 [110/1] is directly connected, eth0, weight 1, 00:29:56
>* 57.57.57.0/24 is directly connected, eth0, 00:29:58
>* 67.67.67.0/24 [110/1] is directly connected, eth1, weight 1, 00:29:56
>* 67.67.67.0/24 is directly connected, eth1, 00:29:59
>* 78.78.78.0/24 [110/1] is directly connected, eth2, weight 1, 00:29:56
>* 78.78.78.0/24 is directly connected, eth2, 00:29:57
>* 192.168.100.0/24 [110/5] via 67.67.67.6, eth1, weight 1, 00:00:37
>* 192.168.200.0/24 [110/2] via 78.78.78.8, eth2, weight 1, 00:29:04
vyos@vyos:~$

```

Gambar 3. 20 Tabel *Routing* Skenario Kedua Pada *Router R7* Ketika *R3* dan *R5 Down*

Dapat dilihat dari gambar 3.20 di atas yang merupakan tabel *routing* pada *router R7* dengan kondisi *router R3* dan *R5 down*, semua alamat IP *gateway* hanya dapat diakses melalui satu rute yaitu melalui alamat IP 67.67.67.6 dengan *port eth1*. Hal ini dikarenakan pada skenario kedua *router R5* mengalami *down* sehingga hanya ada satu rute yang dapat dilewati untuk proses komunikasi yaitu *router R6*.

3.6 PENGAMBILAN DATA

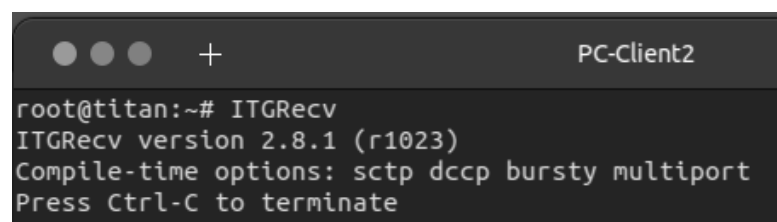
3.6.1 Nilai Parameter QoS

Pada proses pengambilan nilai parameter *Quality of Services* (QoS) yang terdiri dari *throughput*, *delay* dan *jitter* sendiri didapatkan dari skenario pertama pada saat kondisi semua *router* menyala dan skenario kedua pada saat kondisi dua *core router* mengalami *down*. Perangkat lunak D-ITG nantinya digunakan untuk mengirimkan trafik dari *PC-client* 1 ke *PC-Client* 2 yang mempermudah proses pengambilan data. Pada tabel 3.4 merupakan skenario proses pengambilan data dengan protokol TCP yang terbagi lagi menjadi beberapa besaran data beserta lama waktu pengirimannya, pengambilan data ini masing-masing dilakukan sebanyak 30 kali pengujian.

Tabel 3. 4 Skenario Pengambilan Data

| Skenario Pertama dan Skenario Kedua | | | |
|-------------------------------------|-----------------|---------------------|------------------|
| Protokol | Besar Data (MB) | Lama Pengiriman (s) | Jumlah Pengujian |
| TCP | 10 | 20 | 30 |
| | 20 | 20 | 30 |
| | 30 | 20 | 30 |
| | 35 | 20 | 30 |
| | 40 | 20 | 30 |
| | 50 | 20 | 30 |
| | 100 | 20 | 30 |
| | 1000 | 20 | 30 |

Proses pengambilan data menggunakan perangkat lunak D-ITG sendiri memerlukan beberapa perintah konfigurasi yang dimasukkan kedalam *PC-Client* 1 dan *PC-Client* 2. Langkah pertama melakukan konfigurasi pada *PC-Client* 2 yang berperan sebagai penerima trafik data dengan memasukkan perintah *ITGRecv* seperti pada gambar 3.21 di bawah ini.



```
root@titan:~# ITGRecv
ITGRecv version 2.8.1 (r1023)
Compile-time options: sctp dccp bursty multiport
Press Ctrl-C to terminate
```

Gambar 3. 21 *PC-Client* 2 Sebagai Penerima Trafik Data

Selanjutnya langkah kedua melakukan konfigurasi pada *PC-Client 1* yang berperan sebagai pengirim trafik data dengan memasukkan *script* bernama *pengambilandata.sh* yang berisikan perintah *ITGSend -a* (IP address penerima :192.168.200.100) *-c* (besaran data yang dikirimkan : 16.700 Byte) *-C* (Jumlah paket yang dikirimkan dalam satu detik : 30) *-t* (lama waktu data dikirimkan : 20000 milliseconds) *-T* (jenis data yang dikirimkan : TCP) *-l* (lokasi folder data yang akan dikirimkan bernama *sender_tcp_10mb-Data-1*) *-x* (lokasi folder data yang akan diterima bernama *receiver_tcp_10mb_Data-1*). Di bawah ini merupakan *script* yang akan dieksekusi untuk mengirimkan trafik dari *PC-Client 1* ke *PC-Client 2*.

```

GNU nano 2.2.6      File: pengambilandata.sh      Modified
# /bin/bash
Send=ITGSend
IP=192.168.200.100
#PENGIRIMAN PROTOKOL TCP
#running dari pc 1

#10MB TCP
for ((c=1; c<=30; c++))
do
echo "Pengambilan data TCP 10MB ke $c"
$Send -a $IP -c 16700 -C 30 -t 20000 -T TCP -l /home/titanfix/hasdat/10mb/
sender2/sender_tcp_10mb_Data-$c -x /home/titanfix/10mb/terima2/
receiver_tcp_10mb_Data-$c
sleep 15
done
#Jeda 10 detik
echo "Jeda dulu 10 detik, masuk ke 20MB"
sleep 10

```

Gambar 3. 22 Script ITGSend pada *PC-Client 1*

Pada gambar 3.22 di atas dapat dilihat bahwa *PC-Client 1* akan mengirimkan trafik menggunakan protokol TCP dengan besaran data sebesar 16700 Bytes, dengan jumlah paket yang dikirimkan perdetiknya berjumlah 30, serta *interval* waktu pengiriman data dilakukan selama 20000 milidetik atau 20 detik. Sehingga besaran data yang akan diterima *PC-Client 2* sebesar 10 MB.

Hasil dari proses pengiriman trafik terdapat pada *PC-Client 2* yang berperan sebagai penerima dan dapat dilihat dengan masuk kedalam *folder* */home/titanfix/10mb/terima2*. Semua *file* dalam *folder* berisikan *log file* dengan parameter-parameter QoS yang ada yaitu *throughput*, *delay* dan *jitter* dengan

memasukan perintah *ITGDecs*, sehingga akan menampilkan *log file* seperti gambar 3.23 di bawah ini.

```
***** TOTAL RESULTS *****
Number of flows      =          1
Total time           =    19.979519 s
Total packets        =         598
Minimum delay        =     1.191634 s
Maximum delay        =     1.197134 s
Average delay        =     1.193534 s
Average jitter       =     0.000990 s
Delay standard deviation = 0.001724 s
Bytes received       =    9986600
Average bitrate      = 3998.734904 Kbit/s
Average packet rate  = 29.930650 pkt/s
Packets dropped      =          0 (0.00 %)
Average loss-burst size =          0 pkt
Error lines          =          0
```

Gambar 3. 23 Log File

3.6.2 Nilai Konvergensi

Pada proses pengambilan nilai konvergensi diterapkan dari skenario kedua saat kondisi *failover* dua *core router* mati ketika proses pengiriman data berlangsung yaitu ketika *router R3 dan R6 down, router R4 dan R5 down, router R4 dan R6 down, dan router R3 dan R5 down*. Pengambilan nilai konvergensi memanfaatkan *ping tool* yang dikirimkan dari *PC-Client 1* ke *PC-Client 2* yang nantinya saat proses pengiriman trafik *ping* berlangsung, dua dari *core router* akan dimatikan yang bertujuan untuk mengetahui berapa lama waktu konvergensi jaringan dari kondisi *down* sampai bisa terhubung kembali menggunakan *stopwatch*. Di bawah ini merupakan gambar 3.24 yang menunjukkan proses konvergensi pada saat kondisi jaringan mengalami *failover*.

```
PC-Client1
root@dydean:~# ping 192.168.200.100
PING 192.168.200.100 (192.168.200.100) 56(84) bytes of data:
64 bytes from 192.168.200.100: icmp_seq=1 ttl=62 time=8.46 ms
64 bytes from 192.168.200.100: icmp_seq=2 ttl=62 time=8.65 ms
64 bytes from 192.168.200.100: icmp_seq=3 ttl=62 time=8.25 ms
64 bytes from 192.168.200.100: icmp_seq=4 ttl=62 time=8.77 ms
64 bytes from 192.168.200.100: icmp_seq=5 ttl=62 time=8.65 ms

64 bytes from 192.168.200.100: icmp_seq=16 ttl=62 time=9.05 ms
64 bytes from 192.168.200.100: icmp_seq=17 ttl=62 time=36.5 ms
64 bytes from 192.168.200.100: icmp_seq=18 ttl=62 time=8.87 ms
64 bytes from 192.168.200.100: icmp_seq=19 ttl=62 time=4.36 ms
64 bytes from 192.168.200.100: icmp_seq=20 ttl=62 time=8.82 ms
```

Gambar 3. 24 Proses Konvergensi

Proses *failover* dapat diamati melalui hasil *packet capture* menggunakan *wireshark* seperti gambar 3. 25 di bawah ini, ketika *PC-client 1* melakukan

perintah “ping” ke PC-client 2. Saat proses pengiriman trafik *ICMP* berlangsung dengan kondisi semua *router* menyala, maka jalur yang dilewati melalui *router* R4 ke *router* R6. Namun ketika *router* R4 dan *router* R6 dimatikan maka proses *packet capture* terhenti dan akan mencari jalur alternatif yang masih tersedia yaitu jalur pada *router* R3 dan *router* R5 untuk mengirimkan trafik *ICMP* seperti pada gambar 3.26 di bawah ini.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------|------------|-------------|----------|--------|----------------------|
| 19 | 12.314584 | 46.46.46.4 | 224.0.0.5 | OSPF | 82 | Hello Packet |
| 20 | 13.018780 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 21 | 14.020943 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 22 | 15.022462 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 23 | 16.023317 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 24 | 17.026748 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 25 | 18.026781 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 26 | 19.029503 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 27 | 20.029720 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 28 | 21.032911 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 29 | 21.487822 | 46.46.46.4 | 224.0.0.5 | OSPF | 82 | Hello Packet |
| 30 | 22.033164 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 31 | 22.314738 | 46.46.46.4 | 224.0.0.5 | OSPF | 82 | Hello Packet |
| 32 | 23.035963 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 33 | 24.037501 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 34 | 25.039247 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 35 | 26.040893 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 36 | 27.043082 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 37 | 28.044802 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 38 | 29.046469 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 39 | 30.047763 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 40 | 31.049659 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 41 | 31.488158 | 46.46.46.4 | 224.0.0.5 | OSPF | 82 | Hello Packet |
| 42 | 32.050321 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 43 | 32.315318 | 46.46.46.4 | 224.0.0.5 | OSPF | 82 | Hello Packet |
| 44 | 33.053300 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 45 | 34.055440 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 46 | 35.057205 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 47 | 36.059060 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 48 | 37.060900 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 49 | 38.062785 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 50 | 39.064090 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 51 | 40.065458 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |

Gambar 3. 25 Packet Capture Router R4 ke R6

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------|------------|-------------|----------|--------|----------------------|
| 62 | 60.002686 | 35.35.35.5 | 224.0.0.5 | OSPF | 82 | Hello Packet |
| 63 | 60.094123 | 35.35.35.3 | 224.0.0.5 | OSPF | 82 | Hello Packet |
| 64 | 70.002987 | 35.35.35.5 | 224.0.0.5 | OSPF | 82 | Hello Packet |
| 65 | 79.004532 | 35.35.35.3 | 224.0.0.5 | OSPF | 82 | Hello Packet |
| 66 | 80.003257 | 35.35.35.5 | 224.0.0.5 | OSPF | 82 | Hello Packet |
| 67 | 89.004963 | 35.35.35.3 | 224.0.0.5 | OSPF | 82 | Hello Packet |
| 68 | 89.745362 | 35.35.35.5 | 224.0.0.5 | OSPF | 154 | LS Update |
| 69 | 89.886245 | 35.35.35.5 | 224.0.0.5 | OSPF | 262 | LS Update |
| 70 | 89.888821 | 35.35.35.3 | 35.35.35.5 | OSPF | 98 | LS Acknowledge |
| 71 | 90.003486 | 35.35.35.5 | 224.0.0.5 | OSPF | 82 | Hello Packet |
| 72 | 90.007019 | 35.35.35.3 | 224.0.0.5 | OSPF | 262 | LS Update |
| 73 | 90.007298 | 35.35.35.5 | 35.35.35.3 | OSPF | 98 | LS Acknowledge |
| 74 | 90.067019 | 35.35.35.3 | 224.0.0.5 | OSPF | 122 | LS Update |
| 75 | 90.117035 | 35.35.35.3 | 224.0.0.5 | OSPF | 118 | LS Acknowledge |
| 76 | 90.192635 | 35.35.35.3 | 224.0.0.5 | OSPF | 262 | LS Update |
| 77 | 90.193334 | 35.35.35.5 | 35.35.35.3 | OSPF | 98 | LS Acknowledge |
| 78 | 90.288750 | 35.35.35.5 | 224.0.0.5 | OSPF | 98 | LS Acknowledge |
| 79 | 90.319631 | 35.35.35.5 | 224.0.0.5 | OSPF | 166 | LS Update |
| 80 | 90.498974 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 81 | 90.502782 | 78.78.78.8 | 12.12.12.1 | ESP | 182 | ESP (SPI=0xc36cce60) |
| 82 | 91.117112 | 35.35.35.3 | 224.0.0.5 | OSPF | 78 | LS Acknowledge |
| 83 | 91.500946 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 84 | 91.504827 | 78.78.78.8 | 12.12.12.1 | ESP | 182 | ESP (SPI=0xc36cce60) |
| 85 | 92.502862 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 86 | 92.507034 | 78.78.78.8 | 12.12.12.1 | ESP | 182 | ESP (SPI=0xc36cce60) |
| 87 | 93.502716 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 88 | 93.503479 | 78.78.78.8 | 12.12.12.1 | ESP | 182 | ESP (SPI=0xc36cce60) |
| 89 | 94.505700 | 12.12.12.1 | 78.78.78.8 | ESP | 182 | ESP (SPI=0xc3d90ce4) |
| 90 | 94.509620 | 78.78.78.8 | 12.12.12.1 | ESP | 182 | ESP (SPI=0xc36cce60) |

Gambar 3. 26 Packet Capture Router R3 ke R5