BAB III
METODE PENELITIAN

3.1 Alat dan Parameter yang Diteliti

Penelitian ini menggunakan suatu simulasi jaringan dalam menganalisis unjuk kerja "*DMVPN Dual Hub Single Cloud Berbasis Load Balancing Failover dengan Open-Source Router VyOS"*. Model simulasi yang diimplementasikan penelitian ini menggunakan program *network simulator GNS3* dan bantuan *software* pendukung lain seperti *Wireshark* dan D-ITG untuk mendapatkan data.

3.1.1 Perangkat Keras

Perangkat keras yang dipakai dalam penelitian ini terdiri dari satu laptop yang digunakan untuk menjalankan perangkat lunak *network emulator* dengan yang terdapat pada tabel 3.1.

Tabel 3.1 Spesifikasi Laptop

Sistem Operasi	Ubuntu DDE 20.04 LTS
Processor	AMD A8 6410
RAM	12GB
SSD	256GB

3.1.2 Perangkat Virtual

Perangkat yang digunakan dalam penelitian ini menggunakan model virtual yang terdiri dari, 7 *Router VyOS*, 3 *Switch*, dan 3 PC sebagai *client*. Spesifikasi masing-masing perangkat tercantum dalam tabel 3.2.

Tabel 3.2 Spesifikasi Perangkat Virtual

No	Perangkat	Tipe	RAM
1	*Router*	*1.3.0 Equuleus*	512 MB
2	*Switch*	*Ethernet Switch*	216-512 MB
3	PC-*Client*	*Ubuntu Server* 14.04.06 LTS	512 Mb

3.1.3 Perangkat Lunak

A. *Graphical Network Simulator* 3 (GNS3)

Software GNS3 merupakan *network emulator* yang nanti menggunakan versi 2.2.33.1 untuk merancang topologi, menjalankan topologi dan melakukan simulasi pengujian.

B. *Wireshark*

Tool sniffing ini berguna untuk mengamati *packet-packet* yang ada ketika menjalankan DMVPN. *Wireshark* akan menampilkan *protocol* apa saja yang berjalan pada suatu jaringan yang mengacu pada layer *Open System Interconnection* (OSI). Pada penelitian ini menggunakan *Wireshark* versi 3.2.3.

C. D-ITG

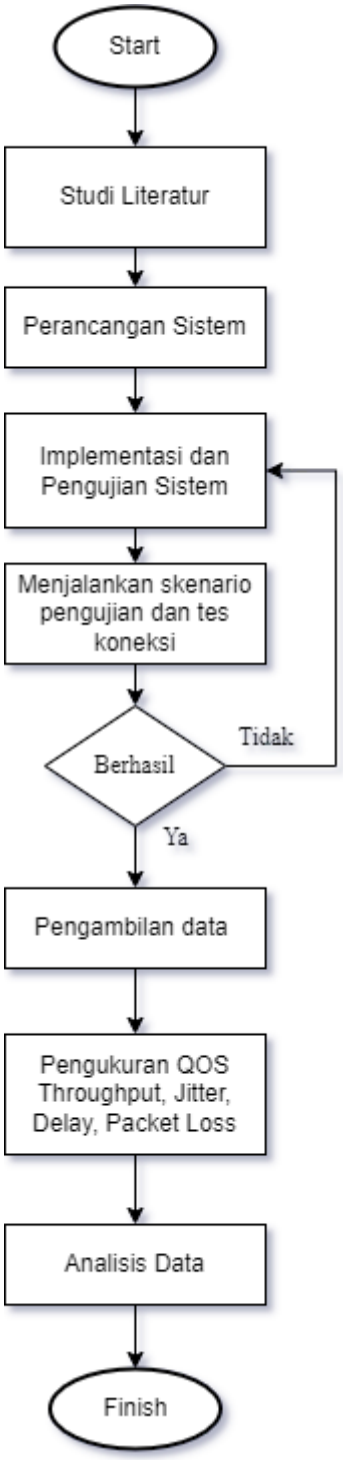
D-ITG versi 2.8.1 yang akan berperan sebagai *tool* untuk pengujian QoS pada DMVPN dengan mengirimkan sebuah trafik berupa TCP dan UDP pada jaringan. D-ITG akan menghasilkan *output* secara langsung berbagai parameter yang ada seperti *throughput*, *delay*, dan *jitter*.

D. VyOS

VyOS akan berperan sebagai sistem operasi yang digunakan oleh *router* untuk menjalankan DMVPN dengan memanfaatkan versi 1.3.0 *Equuleus*.

3.2 Alur Penelitian

Penelitian kali ini dibutuhkan beberapa tahapan untuk memenuhi aspek yang diperlukan. Dimulai dari mencari dan membaca literatur referensi penelitian, prosese perancangan aplikasi simulator dan menentukan skenario pengujian. Selanjutnya proses instalasi dan konfigurasi perangkat pada sistem operasi, *router* HUB, *router* SPOKE, dan *tool* pengujian D-ITG di sisi *client*. Kemudian dilakukan uji coba untuk memastikan konfigurasi berjalan dengan baik. Setelah itu melakukan proses pengujian untuk mengumpulkan data yang akan dianalisis. Berikut merupakan diagram alir yang menggambarkan alur kerja dari penelitian ini :



Gambar 3.1 Alur Pengerjaan

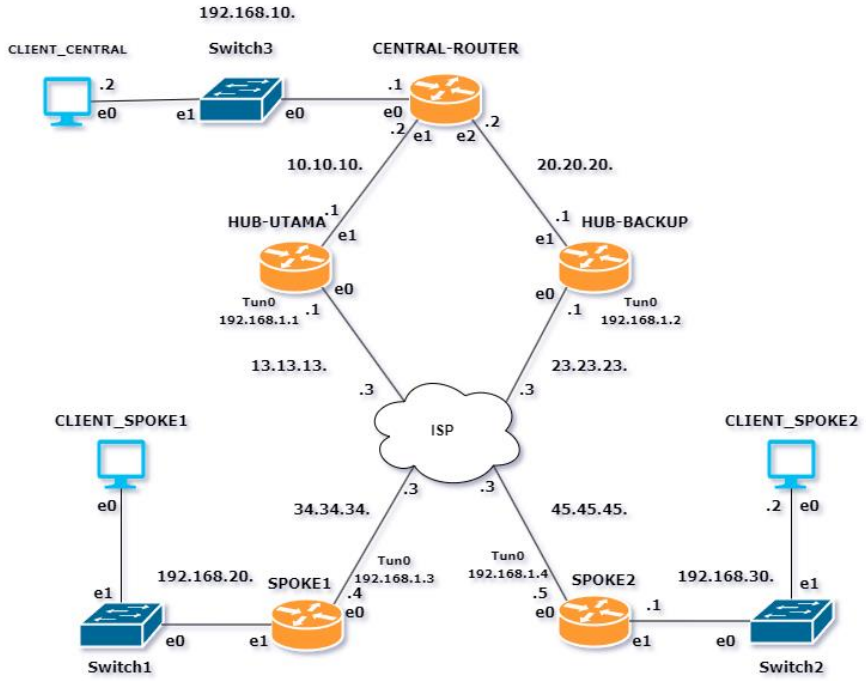
Gambar 3.1 merupakan alur kerja dari penelitian yang dilakukan. Alur kerja dimulai dengan studi literatur untuk mencari referensi terkait materi dan bahan yang akan dibahas pada penelitian ini. Selanjutnya melakukan perancangan sistem yang akan digunakan, mempersiapkan *resource* dan perangkat sistem yang diperlukan. Kemudian pada tahap implementasi jaringan dengan topologi sudah dirancang

sebelumnya yang disimulasikan menggunakan *software GNS3*. Setelah melakukan simulasi, tahap pengerjaan berlanjut ke konfigurasi jaringan. Pada tahap ini konfigurasi jaringan dilakukan pada perangkat, nantinya konfigurasi nanti akan memiliki beberapa skenario pengujian yaitu melakukan tes koneksi antar *spoke*, koneksi *hub* dan *spoke* kedua tes tersebut akan diulangi untuk skenario 2 dan 3 dengan mematikan salah satu *router hub* yang ada secara bergantian. Selanjutnya skenario 4 dengan mematikan kedua *router hub* dan skenario 5 yaitu pengujian waktu konvergensi dengan mematikan salah satu hub saat pengiriman data sedang berlangsung.

Setelah pengerjaan konfigurasi jaringan dan pengetesan skenario, maka selanjutnya adalah proses pengambilan data. Data yang akan diambil nantinya akan dijalankan dengan mengunakan D-ITG yang di*capture* dengan *wireshark*. Data l yang diambil akan mengacu ke beberapa parameter QoS yaitu *throughput*, *delay jitter* dan *packet loss.* Setelah semua data terkumpul, penelitian memasuki tahap analisis untuk menunjang kajian yang telah dilakukan. Setelah melakukan analisis penulis membuat kesimpulan yang mewakili seluruh tahapan yang telah dilewati sebagai pelengkap.

3.3 Topologi Jaringan

Topologi pada jaringan DMVPN secara fisik akan terlihat seperti model *star*, tapi ketika menerapkan jaringan DMVPN topologinya akan menjadi *full mesh* dengan komunikasinya tidak melewati media *interface* kabel, melainkan melalui sebuah jalur *tunnel*. Dimana jalur *tunnel* tersebut bersifat *point to multipoint* yang menghubungkan setiap *spoke* dan *hub*. Komunikasi antar *spoke* bisa secara langsung tanpa melalui *router hub* terlebih dahulu. Secara lebih jelasnya topologi bisa dilihat pada gambar 3.2.



Gambar 3.2 Topologi Jaringan

Tiap *node* pada DMVPN memiliki fungsi dan perannya masing-masing. *HUB_UTAMA* akan berfungsi sebagai pusat komunikasi dari DMVPN yang akan dihubungkan dengan *CENTRAL_ROUTER* yang berfungsi sebagai *router* penghubung pada kantor pusat dan juga sebagai penyeimbang trafik, kemudian terhubung dengan *CLIENT_CENTRAL* yang berperan sebagai kantor pusat. *HUB_BACKUP* memiliki fungsi seperti HUB_UTAMA namun terdapat fungsi lain yaitu untuk pembagi beban trafik apabila kedua *hub* menyala, lalu sebagai *hub* cadangan apabila HUB_UTAMA terjadi *down* dan memastikan proses pengiriman data tetap berjalan. Ketiganya akan terhubung ke ISP yang berperan sebagai internet dan terhubung dengan *SPOKE1* dan *SPOKE2* yang masing-masing berperan sebagai kantor cabang. Masing-masing *SPOKE* dan kantor pusat memiliki *client*. *Client-client* tersebut yang akan melakukan pengujian QoS dari DMVPN. Setiap *router* yang ada akan menjalankan DMVPN yaitu *router spoke* dan *hub* akan diberikan sebuah *interface tunnel* beserta IP *address*-nya, sedangkan untuk *router INTERNET* dan *client* akan dialokasikan sebuah IP *address* sesuai dengan *interface* yang ditetapkan pada *port* masing-masing. Pembagian alokasi dari setiap *address* yang ada terlampir dalam tabel 3.3.

Tabel 3.3 Alokasi IP *Address* Perangkat

Perangkat	***Port***	**IP *Address***	***Prefix***
Central_Router	E0	192.168.10.1	/24
	E1	10.10.10.2	/24
	E2	20.20.20.2	/24
HUB_UTAMA	E0	13.13.13.1	/24
	E1	10.10.10.1	/24
	Tun0	192.168.1.1	/24
HUB_BACKUP	E0	23.23.23.1	/24
	E1	20.20.20.1	/24
	Tun0	192.168.1.2	/24
INTERNET	E0	13.13.13.3	/24
	E1	23.23.23.3	/24
	E2	34.34.34.3	/24
	E3	45.45.45.3	/24
SPOKE1	E0	34.34.34.4	/24
	E1	192.168.20.1	/24
	Tun0	192.168.1.3	/24
SPOKE2	E0	45.45.45.5	/24
	E1	192.168.30.1	/24
	Tun0	192.168.1.4	/24
Client_Central	Eth0	192.168.10.2	/24
Client_SPOKE1	Eth0	192.168.20.2	/24
Client_SPOKE1	Eth0	192.168.30.2	/24

3.4 Konfigurasi Sistem

Konfigurasi DMVPN secara garis besar berfokus pada perangkat-perangkat *hub* dan *spoke*-nya. Perlu menjadi catatan bahwa walaupun pada topologi memiliki 2 *hub* dan *spoke* yang masing-masing berbeda, akan tetapi secara garis besar konfigurasinya sama, perbedaannya hanya terletak pada penempatan posisi IP *address* yang dipakai. Konfigurasi pada perangkat-perangkat tersebut mencakup : pembuatan jalur *interface tunnel*, konfiguasi *routing static*, konfigurasi *protocol*

NHRP beserta DMVPN, dan terakhir konfigurasi *routing* OSPF yang digunakan untuk menghubungkan setiap *client* yang terhubung kedalam *router* yang menjalankan DMVPN. Konfigurasi pada *Central_Router* sendiri hanya melakukan *routing* OSPF dan pengalamatan IP *address* karena pada dasarnya perangkat tersebut dipakai untuk menghubungkan kedua *router hub DMVPN*. Sedangkan router *INTERNET* hanya akan dikonfigurasikan *static default route. Load balancing* pada DMVPN akan otomatis berjalan saat semua *router* yang ada telah berhasil melakukan *step-step* yang diperlukan tadi. Jalur *load balancing* ini akan terbentuk pada komunikasi dari arah *client* yang berada di *router spoke* menuju *client* yang berada di *router central*. Konfigurasi ini bermaksud untuk menghubungkan bagian depan setiap *router hub* dan *spoke.* Berikut detail dari konfigurasi perangkat *router* utama dalam DMVPN :

3.4.1 Konfigurasi *Router Hub*

Baris konfigurasi di bawah ini menunjukan bagaimana secara lengkap konfigurasi dari DMPVN pada *router* pusat. Langkah awal yang diperlukan adalah pemberian *ip address* pada step 1-9 yang digunakan untuk setiap *interface* fisik yang ada. Proses berikutnya adalah pembuatan jalur *tunnel* dan *ip address*, jalur *tunnel* yang telah terbuat tersebut kemudian diubah modelnya kedalam *generic routing encapsulation* seperti yang tertera pada baris 10 sampai 12. Sebelum DMVPN berjalan diperlukan terlebih dahulu sebuah *routing static default* ke arah *router* INTERNET. Hal tersebut dimaksudkan agar setiap perangkat yang ada terhubung terlebih dahulu bagian depannya. Setelah konfigurasi *interface* dan pemberian *routing* selesai berikutnya bisa dilanjutkan dengan mengkonfigurasi inti DMVPN yang tercakup dalam baris 23 sampai 25. Baris tersebut akan memerintahkan *router* agar alamat IP *local* yang sebelumnya dikonfigurasi dijadikan sebagai NBMA dengan sumber *interface*nya adalah *eth0* dan juga *interface* tersebut pengirimannya menggunakan *mode multicast.* Konfigurasi berikutnya pada DMVPN adalah memerintahkan *router* untuk mengaktifkan *protocol NHRP* yang akan memetakan alamat IP fisik pada *eth0* kedalam alamat *logical interface tunnel*. Konfigurasi tersebut juga akan membuat *interface tunnel* yang ada akan menghasilkan sifat pengiriman secara *multicast* dan dinamis, dan *redirect* yang berfungsi memerintahkan mengalihkan komunikasi langsung ke

dalam *spoke-spoke* jika ada *client* di dalam *router spoke*nya yang ingin terhubung dengan *spoke* yang lain. Perintah tersebut tercantum dalam baris 28 sampai 35. Step terakhir yang juga penting adalah membuat jaringan tempat *client* tiap-tiap *router* DMVPN yang ada, dengan kata lain jaringan yang ada di dalam masing-masing router DMVPN saling terhubung. Penghubungan jaringan tersebut memerlukan *routing* OSPF untuk meng*advertise network* yang ada di dalam *router* yang menjalankan DMVPN seperti yang ditunjukan pada baris 38 sampai 42. Selain itu perlu diingat bahwa karena DMVPN membuat komunikasi bersifat *multipoint* maka pada *routing* OSPF perlu dikonfigurasi agar pengiriman LSA yang dipakai untuk menghubungkan dengan *network* lain, dikirimkan secara *broadcast.* Satu hal penting lainnya adalah mengatur OSPF ini supaya menjadi DR dengan menaikkan prioritas *router*nya. Perintah tersebut bisa dilihat pada baris 13 sampai 20.

```
vyos@HUB-UTAMA:~$             show
configuration
interfaces {
    ethernet eth0 {
        address 13.13.13.1/24
        hw-id 0c:04:76:dc:00:00
    }
    ethernet eth1 {
        address 10.10.10.1/24
        hw-id 0c:04:76:dc:00:01
    }
    tunnel tun0 {
        address 192.168.1.1/24
        encapsulation gre
        ip {
            ospf {
                dead-interval 40
                hello-interval 10
                network broadcast
                priority 2
                retransmit-
interval 5
                transmit-delay 1
            }
        }
        multicast enable
        source-address 13.13.13.1
        source-interface eth0
    }
}
```

```
protocols {
    nhrp {
        tunnel tun0 {
            cisco-authentication
""
            holding-time 30
            multicast dynamic
            redirect
            shortcut
        }
    }
    ospf {
        area 0 {
            network 10.10.10.0/24
            network 192.168.1.0/24
        }
    }
    static {
        route 0.0.0.0/0 {
            next-hop 13.13.13.3 {
            }
        }
    }
}
```

3.4.2 Konfigurasi *Router Spoke*

Konfigurasi DMVPN pada sisi *router spoke* hampir sama dengan konfigurasi pada sisi *hub.* Namun menambahkan sedikit konfigurasi untuk mendaftakan alamat *tunnel* dari *router hub.* Dibawah ini merupakan konfigurasi pada *router Spoke 1*.

```
vyos@SPOKE1:~$ show configuration
interfaces {
    ethernet eth0 {
        address 34.34.34.4/24
        hw-id 0c:83:48:bc:00:00
    }
    ethernet eth1 {
        address 192.168.20.1/24
        hw-id 0c:83:48:bc:00:01
    }
    loopback lo {
    }
    tunnel tun0 {
        address 192.168.1.3/24
        encapsulation gre
        ip {
            ospf {
                dead-interval 40
                hello-interval 10
                network broadcast
                priority 0
                retransmit-interval
5
                transmit-delay 1
            }
        }
        multicast enable
        source-address 34.34.34.4
        source-interface eth0
    }
}
```

```
protocols {
    nhrp {
        tunnel tun0 {
            cisco-authentication ""
            holding-time 30
            map 192.168.1.1/24 {
                nbma-address
13.13.13.1
                register
            }
            map 192.168.1.2/24 {
                nbma-address
23.23.23.1
                register
            }
            multicast nhs
            redirect
            shortcut
        }
    }
    ospf {
        area 0 {
            network 192.168.1.0/24
            network 192.168.20.0/24
        }
    }
    static {
        route 0.0.0.0/0 {
            next-hop 34.34.34.3 {
            }
        }
    }
}
```

Konfigurasi tersebut mencakup konfigurasi *interface* fisik dan *tunnel* (baris 1-9), konfigurasi *routing static* (baris 42-45), konfigurasi *tunneling* beserta *routing* OSPF yang dijalankan (baris 10-16) dan baris (36-40). Perbedaanya disini pada sisi *spoke* perlu meregister atau mendaftarkan alamat dari *router hub* yang berperan sebagai pusat komunikasi. Proses pendaftaran tersebut ada di dalam konfigurasi NHRP pada baris 18-25 untuk HUB_ROUTER dan baris 27-29 untuk HUB_BACKUP dengan memetakan alamat IP *tunnel router hub* tetap diarahkan kedalam IP fisik *interface* fisik dari setiap router yang mengarah ke router INET.

3.4.3 Konfigurasi *Router Load Balancer*

Pada perangkat CENTRAL_ROUTER menuju hub terdapat dua jalur dimana jalur pertama adalah menuju HUB_UTAMA dan jalur kedua yaitu menuju HUB_BACKUP. Oleh karena itu peneliti mengaktifkan fitur *load balancing* untuk memaksimalkan kedua jalur tersebut. Secara *default*, *routing* OSPF akan mengaktifkan metode *per-destination load balancing* namun pada penelitian ini akan diubah menjadi *per-packet load balancing*.

```
vyos@CENTRAL-ROUTER:~$         show
configuration
interfaces {
    ethernet eth0 {
        address 192.168.10.1/24
        hw-id 0c:fd:55:49:00:00
    }
    ethernet eth1 {
        address 10.10.10.2/24
        hw-id 0c:fd:55:49:00:01
    }
    ethernet eth2 {
        address 20.20.20.2/24
        hw-id 0c:fd:55:49:00:02
    }
}
load-balancing {
    wan {
        flush-connections
        interface-health eth1 {
            failure-count 1
            nexthop 10.10.10.1
            success-count 1
        }
        interface-health eth2 {
            failure-count 1
            nexthop 20.20.20.1
            success-count 1
        }
        rule 1 {
            failover
            inbound-interface eth0
            interface eth1 {
                weight 1
            }
            interface eth2 {
                weight 1
            }
            per-packet-balancing
            protocol all
        }
    }
}
protocols {
    ospf {
        area 0 {
            network 10.10.10.0/24
            network 20.20.20.0/24
            network
192.168.10.0/24
        }
    }
}
```

Konfigurasi di atas terdapat konfigurasi *interface* fisik yang menuju kedua *hub* dan *client central* (baris 1-14), lalu konfigurasi OSPF (baris 42-45), dan yang terpenting adalah konfigurasi *load balancing* (baris 25-38) terdapat alamat *nexthop* yang merupakan alamat fisik dari HUB_UTAMA dan HUB_BACKUP yang menuju CENTRAL_ROUTER. selain itu juga terdapat konfigurasi *failover* dimana sudah ditentukan jalur dari jaringan lokal dan jalur menuju *router hub*. Nilai *weight* untuk keduanya disamakan karena kemampuan jalur *bandwidth* yang sama agar pembagian tugasnya merata. Perintah "*per-packet-balancing*" berfungsi untuk

mengubah konfigurasi *load balancing* dari yang defaultnya yaitu *per-destination* menjadi *per-packet*, dimana setiap paket yang dikirim akan dilewatkan melalui jalur yang berbeda secara bergantian.

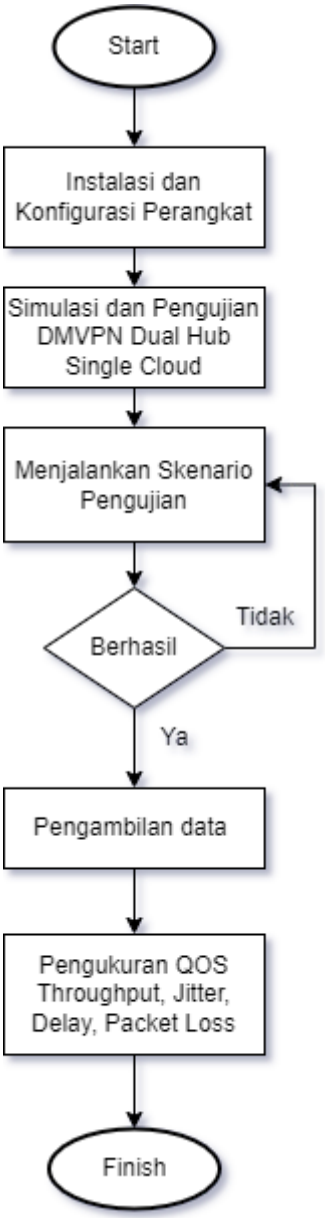
3.5 Proses Pengecekan Jaringan

Proses pengujian dari DMVPN apakah bekerja atau tidak akan dilakukan terlebih dahulu dengan mengirimkan paket *Internet Communication Message Protocol* (ICMP) berupa *ping* dari salah satu *router* ke setiap IP *tunnel* dan *traceroute* dari sisi *client* untuk memastikan bahwa DMVPN bekerja.

Apabila DMVPN telah sukses dijalankan maka *client* dari masing-masing SPOKE bisa saling terhubung satu sama lain. Ketika melakukan proses *traceroute*, jalur yang digunakan paket adalah langsung melalui *interface tunnel* yang dimiliki oleh SPOKE2, komunikasi yang langsung antara *router spoke* tersebut menujukan bahwa DMVPN Phase 3 telah berjalan.

3.6 Alur Simulasi dan Skenario Pengujian

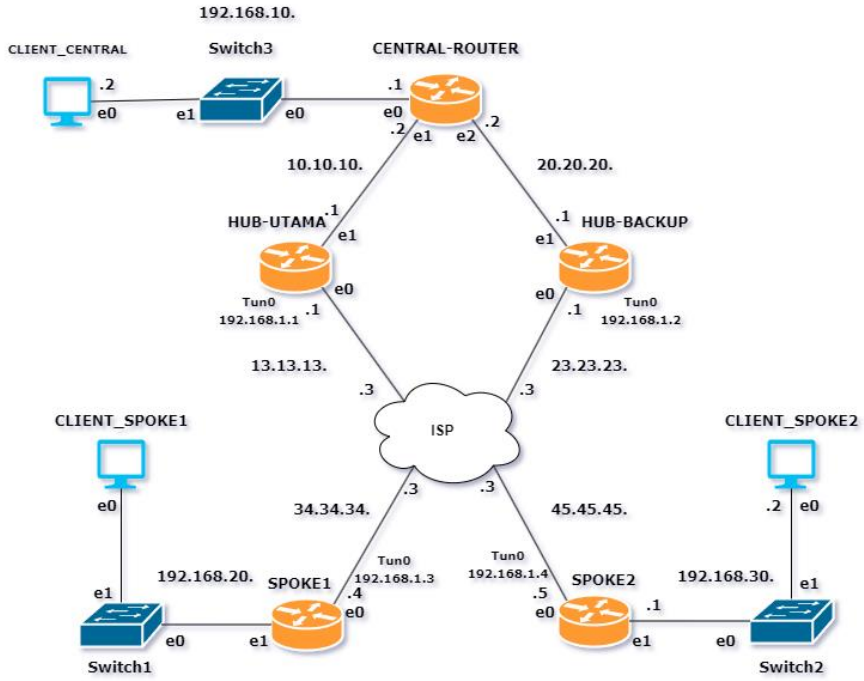
Alur simulasi dimulai dengan instalasi seluruh perangkat yang dibutuhkan kemudian melakukan konfigurasi baik pada *router hub*, *router spoke* maupun pada *client* untuk menkonfigurasi *software* D-ITG. Setelah konfigurasi perangkat selesai dan koneksi seluruh perangkat telah terhubung, dilanjutkan dengan proses pengujian. Untuk skenario pengujian QOS terdiri dari skenario 1 dimana terdapat kondisi saat kedua *hub* menyala, skenario 2 merupakan kondisi saat *hub* utama dimatikan dan hanya menggunakan 1 buah *hub backup*, skenario 3 merupakan kondisi saat *hub backup* dimatikan dan hanya menggunakan 1 buah *hub* utama, skenario 4 merupakan kondisi saat *hub utama* dan *hub backup* dimatikan. Sedangkan untuk skenario 5 yaitu *failover* dimana kondisi kedua *hub* menyala dan melakukan pengiriman data. Disaat proses pengiriman data berlangsung *hub* utama akan dimatikan sehingga penggunaan jalur komunikasi berpindah ke *hub backup*. Gambar 3.3 merupakan alur kerja dari simulasi pengujian yang dilakukan.



Gambar 3.3 *Flowchart* Alur Simulasi Pengujian

3.6.1 Skenario dimana Kedua Router HUB Menyala

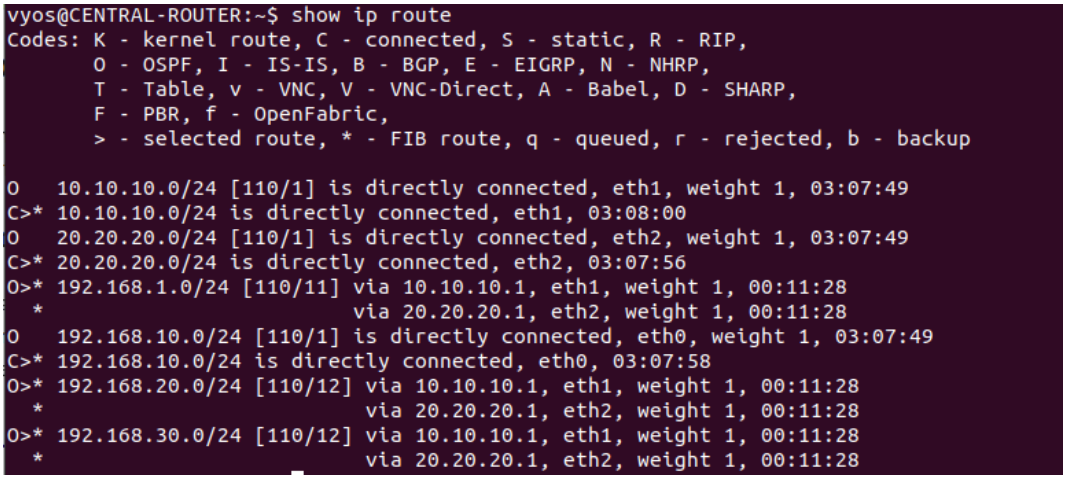
Secara garis besarnya baik simulasi dan pengujian akan menjalan 4 konsep. skenario pertama adalah melakukan pengujian ketika pusat komunikasi dari DMPVN yaitu semua *router* HUB berada dalam kondisi aktif.



Gambar 3.4 Skenario Pengujian Kedua *Router Hub* Menyala

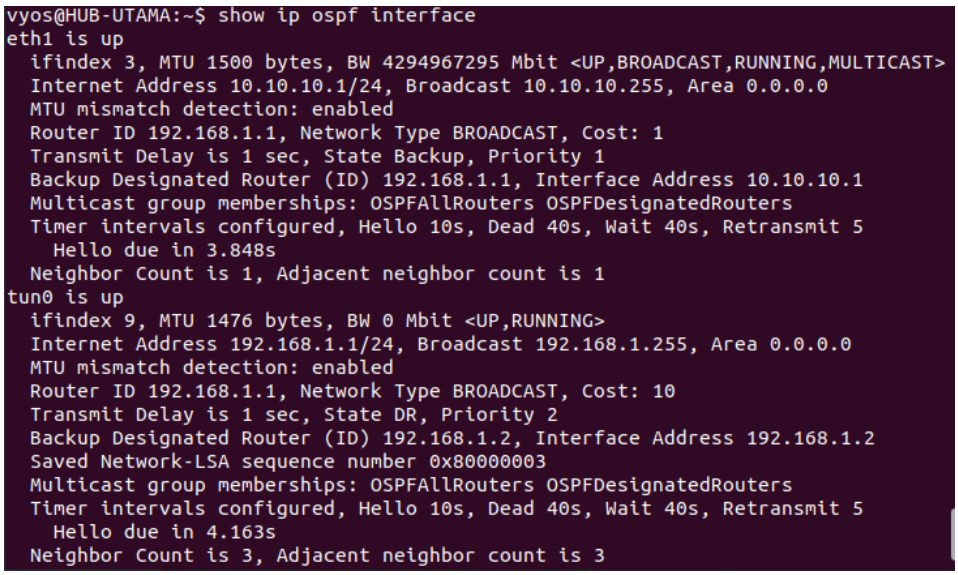
Gambar 3.4 merupakan konsep pertama DMVPN akan bekerja dengan menggunakan dua pusat HUB secara langsung, dan untuk komunikasi dari *client* cabang menuju *client* yang berada di *router* pusat akan menerapkan konsep *load balancing*. Pada konfigurasi ini menggunakan protokol *routing* OSPF, sehingga penentuan jalur pengiriman data ditentukan berdasarkan nilai *cost* pada *interface router*, dimana jalur dengan nilai *cost* yang paling rendah akan digunakan sebagai jalur pengiriman data. Sedangkan pada topologi *dual hub* ini nilai *cost*nya sama, maka penentuan jalur pengiriman data secara *default* ditentukan oleh protokol routing OSPF. Namun pada penelitian ini penentuan jalur pengiriman data ditentukan oleh *router central*.

Saat kondisi tersebut akan dilakukan pengiriman trafik dengan D-ITG dari *client* pusat ke *client* cabang dan antar *client* cabang. Konsep pertama DMVPN akan bekerja dengan menggunakan dua pusat HUB secara langsung, dan untuk komunikasi dari client pusat menuju *client* yang berada di router cabang akan menerapkan konsep *load balancing*.



Gambar 3.5 Tabel *Routing* dari Sisi *Central Router*

Load balancing bekerja atau tidaknya bisa dilihat dengan menggunakan bantuan tabel *routing* seperti yang terlampir pada gambar 3.5. Gambar tersebut menampilkan tabel *routing* yang telah dibuat dari sisi *router* HUB. Terlihat pada gambar bahwa ketika *router* ingin menuju alamat yang berada di router SPOKE1 dan SPOKE2, memiliki dua opsi jalur yaitu menggunakan jalur *interface tunnel* dari HUB_UTAMA dan HUB_BACKUP. Untuk nilai *cost* pada HUB-UTAMA *interface tunnel* 0 yang terhubung ke jaringan DMVPN secara *default* adalah 10 yang tertera pada gambar 3.6.



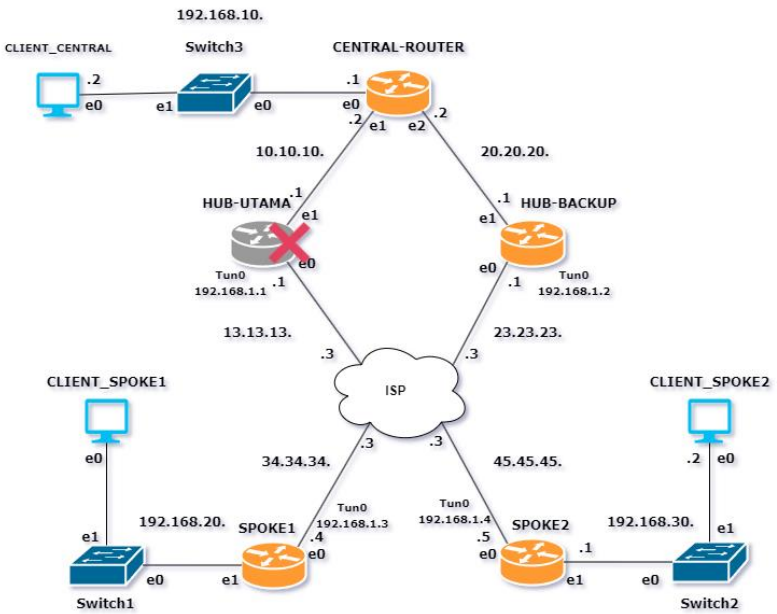
Gambar 3.6 Informasi *Routing* OSPF Pada *Hub* Utama

Sama seperti nilai *cost* pada HUB_UTAMA, nilai *cost* pada HUB_BACKUP juga 10, sehingga jalur yang melewati *interface tunnel* 0 memiliki 2 opsi jalur karena nilai *cost* yang sama. Selain itu yang membedakan yaitu nilai *priority* pada HUB_UTAMA adalah 2 sedangkan pada HUB_BACKUP yaitu 1. Apabila nilai *priority* lebih tinggi akan menjadikan *router* tersebut sebagai *Designated Router* (DR) dalam hal ini adalah HUB_ROUTER sedangkan HUB_BACKUP yang menjadi *Backup Designated Router* (BDR) seperti pada gambar 3.7.

```
vyos@HUB-BACKUP:~$ show ip ospf interface
eth1 is up
  ifindex 3, MTU 1500 bytes, BW 4294967295 Mbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 20.20.20.1/24, Broadcast 20.20.20.255, Area 0.0.0.0
  MTU mismatch detection: enabled
  Router ID 192.168.1.2, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State Backup, Priority 1
  Backup Designated Router (ID) 192.168.1.2, Interface Address 20.20.20.1
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 6.834s
  Neighbor Count is 1, Adjacent neighbor count is 1
tun0 is up
  ifindex 9, MTU 1476 bytes, BW 0 Mbit <UP,RUNNING>
  Internet Address 192.168.1.2/24, Broadcast 192.168.1.255, Area 0.0.0.0
  MTU mismatch detection: enabled
  Router ID 192.168.1.2, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State Backup, Priority 1
  Backup Designated Router (ID) 192.168.1.2, Interface Address 192.168.1.2
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 7.249s
  Neighbor Count is 3, Adjacent neighbor count is 3
```

Gambar 3.7 Informasi *Routing* OSPF Pada *Hub Backup*

3.6.2 Skenario HUB Tunggal



Gambar 3.8 Skenario Pengujian Hub Tunggal.

Konsep kedua adalah menerapkan DMVPN yang hanya terdiri dari satu buah *router* HUB dengan mencoba mematikan HUB utama yang berarti DMVPN hanya bertumpu pada HUB_BACKUP seperti pada gambar 3.8. Kondisi tersebut mengakibat jaringan hanya bekerja menggunakan satu buah *router central* sebagai pusat komunikasi, ini artinya DMVPN hanya bekerja dengan menggunakan HUB_BACKUP sebagai pusat dari semua proses komunikasi yang ada. Skenario ketika salah satu *router* HUB mati akan menyebabkan *load balancing* yang terjadi saat proses komunikasi dari *client* yang berada di cabang menuju *client* yang berada di pusat maupun sebaliknya akan otomatis hilang dikarenakan kondisi jaringan yang saat ini bersifat *single point of failure*. Hal tersebut bisa diverifikasi langsung dengan melihat tabel *routing* sekali lagi yang berada di *router* SPOKE1, akan terlihat apabila jalur ke alamat jaringan *client* yang berada di *router* CENTRAL hanya memiliki satu buah jalur saja seperti pada gambar 3.9.

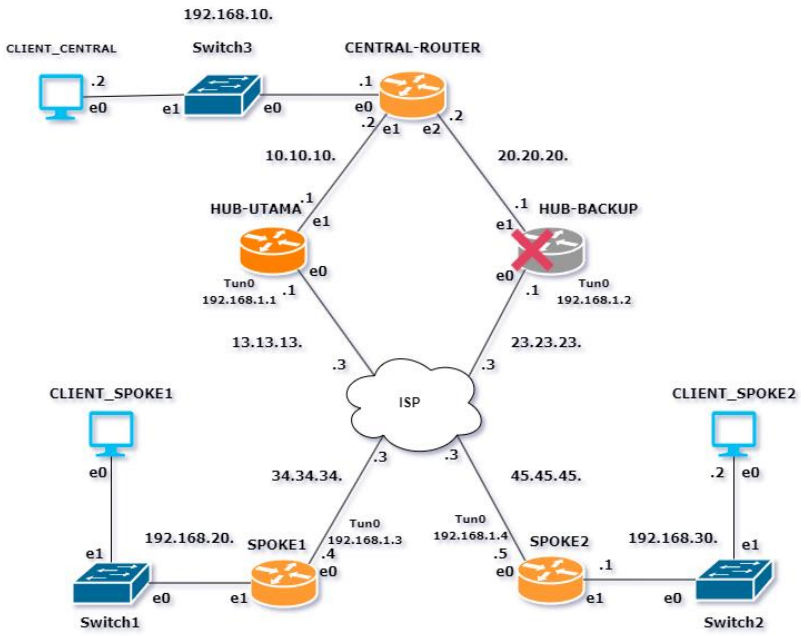
```
vyos@CENTRAL-ROUTER:~$ show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup

O   10.10.10.0/24 [110/1] is directly connected, eth1, weight 1, 05:43:30
C>* 10.10.10.0/24 is directly connected, eth1, 05:47:09
O   20.20.20.0/24 [110/1] is directly connected, eth2, weight 1, 05:32:32
C>* 20.20.20.0/24 is directly connected, eth2, 05:47:07
O>* 192.168.1.0/24 [110/11] via 20.20.20.1, eth2, weight 1, 00:08:21
O   192.168.10.0/24 [110/1] is directly connected, eth0, weight 1, 05:47:02
C>* 192.168.10.0/24 is directly connected, eth0, 05:47:08
O>* 192.168.20.0/24 [110/12] via 20.20.20.1, eth2, weight 1, 00:08:20
O>* 192.168.30.0/24 [110/12] via 20.20.20.1, eth2, weight 1, 00:08:20
vyos@CENTRAL-ROUTER:~$
```

Gambar 3.9 Tabel *Routing* dari Sisi Router SPOKE1 saat HUB_UTAMA dimatikan

```
vyos@HUB-BACKUP:~$ show ip ospf interface
eth1 is up
  ifindex 3, MTU 1500 bytes, BW 4294967295 Mbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 20.20.20.1/24, Broadcast 20.20.20.255, Area 0.0.0.0
  MTU mismatch detection: enabled
  Router ID 192.168.1.2, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State Backup, Priority 1
  Backup Designated Router (ID) 192.168.1.2, Interface Address 20.20.20.1
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 5.635s
  Neighbor Count is 1, Adjacent neighbor count is 1
tun0 is up
  ifindex 9, MTU 1476 bytes, BW 0 Mbit <UP,RUNNING>
  Internet Address 192.168.1.2/24, Broadcast 192.168.1.255, Area 0.0.0.0
  MTU mismatch detection: enabled
  Router ID 192.168.1.2, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State DR, Priority 1
  No backup designated router on this network
  Saved Network-LSA sequence number 0x80000002
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 6.049s
  Neighbor Count is 2, Adjacent neighbor count is 2
```

Gambar 3.10 Informasi Ruting OSPF pada *Hub Backup* Menyala

Gambar 3.10 merupakan informasi routing ospf HUB_BACKUP saat HUB_UTAMA dimatikan, nilai *cost* pada HUB_BACKUP tetap 10. Karena sebelumnya menjadi BDR dan *router* DR *down*, maka HUB_BACKUP secara otomatis menjadi DR meskipun nilai *priority* nya tetap 1.

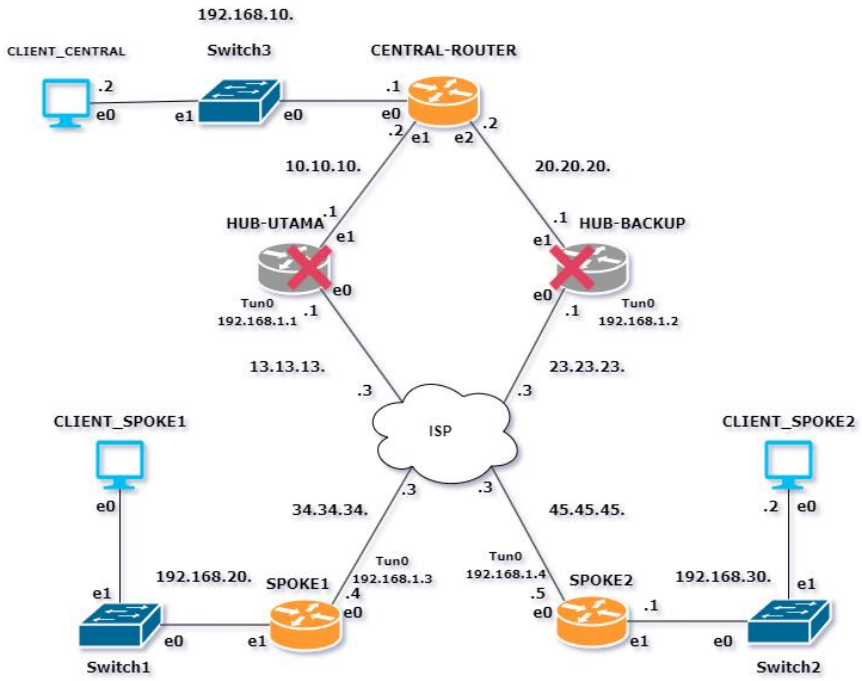


Gambar 3.11 Skenario Pengujian Hub Tunggal

Pengujian selanjutnya akan mematikan HUB-BACKUP dan jalur komunikasi akan melewati HUB-UTAMA. Gambar 3.11 mengilustrasikan bahwa

HUB-BACKUP dalam kondisi mati. pengujian ini bertujuan untuk membandingkan performa antara HUB-UTAMA dan HUB-BACKUP.

3.6.3 Skenario Kedua HUB Mati

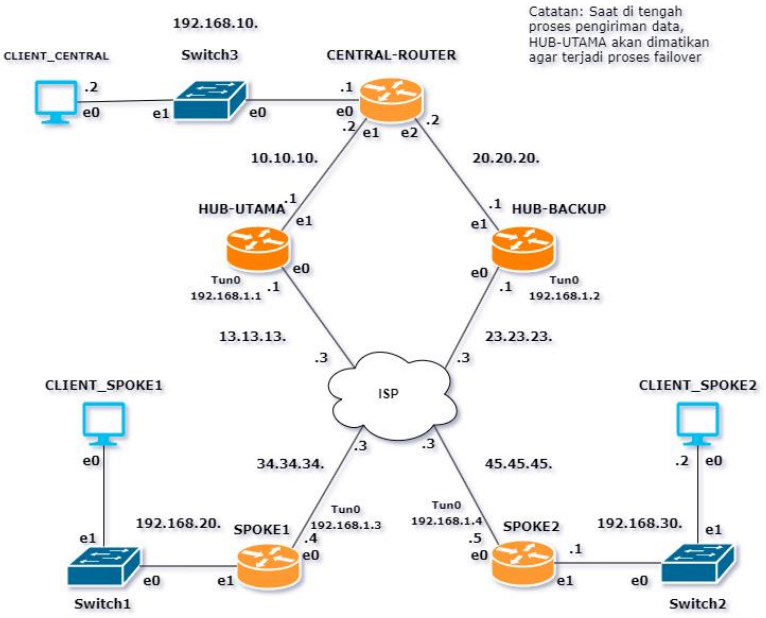


Gambar 3.12 Skenario Pengujian Kedua *Hub* Kondisi Mati

Gambar 3.12 menunjukan dimana kedua *router hub* mati. Pengujian ini bertujuan membuktikan apakah masing-masing *spoke* masih dapat terhubung meskipun pusat dari DMVPN terputus.

3.6.4 Skenario *Failover*

Skenario pengujian yang dilakukan berikutnya adalah hampir sama seperti model skenario ketika semua *hub* menyala, hanya saja yang menjadi catatan adalah saat ditengah proses komunikasi sedang berlangsung akan mematikan salah satu *router* HUB yang ada. Pengujian ini akan menghasilkan sebuah kondisi jaringan *failover* dimana ketika salah satu *link* mati maka proses komunikasi otomatis akan berpindah menuju perangkat yang menyala seperti yang terdapat pada gambar 3.13.



Gambar 3.13 Skenario Pengujian *Failover*

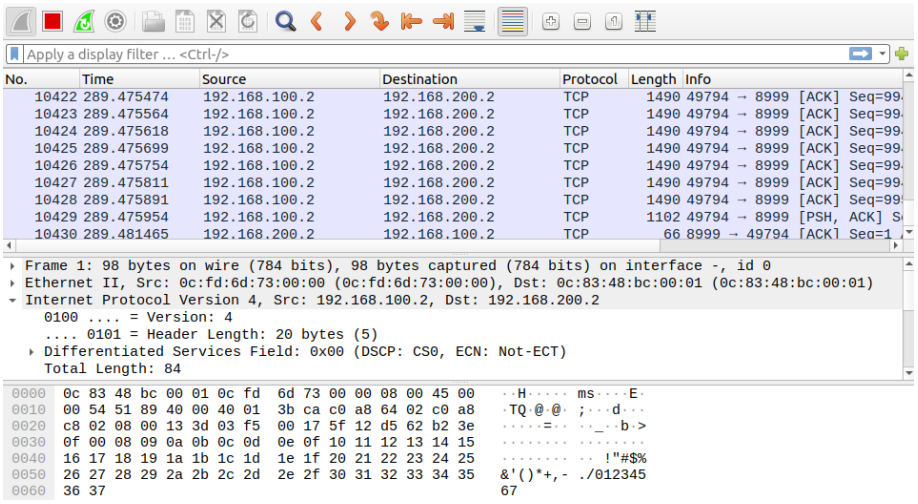
3.7 Pengambilan Data

Pengambilan data akan mengacu kepada kondisi skenario pengujian yang sudah ada baik dari skenario ketika semua *hub* menyala sampai skenario ketika *hub* dimatikan saat proses pengambilan data berlangsung. Secara garis besar pengambilan data yang ada akan terbagi menjadi ke dalam dua jenis, yaitu pengambilan data untuk nilai QoS dan yang kedua adalah pengambilan data untuk nilai konvergensi.

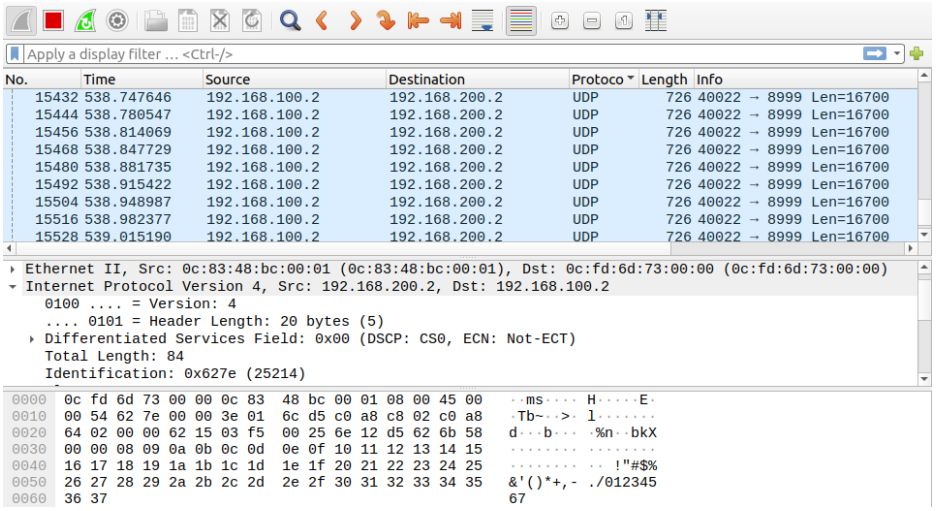
3.7.1 Pengambilan Data Parameter QoS

Parameter QoS yang akan diambil menggunakan pengujian jenis skenario pertama dan jenis kedua yaitu kondisi pengambilan data ketika semua *router* HUB aktif dan bekerja dan kondisi ketika hanya satu buah *router* HUB saja yang aktif bekerja. Pengambilan data parameter QoS yang mencakup *throughput*, *delay*, *jitter* dan *packet loss* diujikan dengan mengirimkan dua buah jenis protokol yang berbeda yaitu TCP dan UDP. Protokol keduanya akan dibuat oleh dengan bantuan perangkat lunak D-ITG. D-ITG akan mengirimkan trafik TCP atau UDP dari *client* yang akan diujikan. Pengujian performansi beserta pengambilan data untuk menguji untuk kerja dari DMVPN akan memanfaatkan *client* pada masing masing *router* cabang dan pusat. *Client* tersebut akan di*install software* D-ITG dan bersamaan menggunakan *wireshark* untuk meng-*capture* paket-paket apa saja yang

berjalan seperti yang tercantum pada gambar 3.14 dan 3.15. Pengunaan D-ITG di masing-masing *client* akan menghasilkan informasi-informasi mengenai kualitas jaringan yang ada.



Gambar 3.14 Tampilan *Wireshark* saat Pengiriman Paket TCP



Gambar 3.15 Tampilan *Wireshark* saat Pengiriman paket UDP

Mekanisme pengambilan data dengan D-ITG menggunakan model *client-server* pada setiap *client* yang diujikan dengan masing masingnya mempunyai peran sebagai yang mengirimkan data dan yang menerima data. *Client* yang berperan sebagai penerima data hanya perlu memasukan *ITGRecv* untuk mendeklarasikan dirinya sebagai penerima data dengan contoh ilustrasi sebagai berikut:

```
root@CLIENT-SPOKE2:~# ITGRecv
ITGRecv version 2.8.1 (r1023)
Compile-time options: sctp dccp bursty multiport
Press Ctrl-C to terminate
Listening on UDP port : 8999
Finish on UDP port : 8999
```

Berbeda dengan sisi penerima yang hanya memerlukan perintah sederhana, pada sisi pengirim perlu memasukkan beberapa parameter-parameter yang diperlukan untuk pengiriman trafik yang digunakan sebagai proses komunikasi. Beberapa parameter wajib yang perlu dideklarasikan oleh pengirim mencakup alamat tujuan dan juga parameter log yang nantinya dipakai untuk mengambil nilai QoS jaringan.

```
root@CLIENT-SPOKE1:~# ITGSend -a 192.168.30.2 -c 10485760
-C 30 -t 20000 -T UDP -l /home/anand/send.log -x
/home/anand/receive.log
ITGSend version 2.8.1 (r1023)
Compile-time options: sctp dccp bursty multiport
Started sending packets of flow ID: 1
Finished sending packets of flow ID: 1
```

Pengujian D-ITG disisi pengirim perlu memasukkan perintah yang didalamnya mencakup beberapa parameter seperti IP *Address* yang diikuti dengan *flag* -a. Secara bawaan D-ITG akan menggunakan mengirimkan paket dengan tipe protokol UDP, untuk menspesifikan tipe protokol data yang dikirimkan juga perlu dengan menggunakan *flag* -T di ikuti dengan protokol yang diikuti. Durasi pengiriman juga perlu dideklarasikan dengan *flag* -t dengan diikuti dengan lama waktu yang diinginkan dalam satuan *millisecond.* Parameter penting lain yang perlu dimasukan adalah -l yang difungsikan agar pengujian memberikan sebuah log berisi informasi performa jaringan seperti *throughput, jitter, delay* dan *packet* pada sisi pengirim sedangkan -x akan menyimpan log informasi di sisi penerima . Data tersebut yang akan digunakan sebagai nilai parameter QoS. Aspek selanjutnya yang perlu disertakan dalam proses pengiriman traffik tersebut adalah -c yang berfungsi memberikan beban besar paket yang akan dikirimkan. Beban besar paket tersebut bisa divariasikan dengan satuannya adalah *bytes.* Apabila ukuran paket yang akan dikirimkan menggunakan satuan *megabytes* maka harus dikonversi terlebih dahulu menjadi *bytes* dengan dikalikan 1024^2. Kemudian *flag* -C berfungsi untuk

menentukan jumlah data yang dikirim dalam satu detik. Pengujian performansi untuk menghasilkan nilai QoS dari jaringan akan menggabungkan semua parameter yang ada dengan skenario pengambilan datanya tercantum pada tabel 3.4.

Tabel 3.4 Skenario Pengambilan Data untuk Protokol TCP dan UDP

Besar Data (MB)	Lama Pengiriman (s)	Jumlah Pengujian	Konversi Besar Data (Bytes)
10	20	30	10.485.760
20	20	30	20.971.520
30	20	30	31.457.280
40	20	30	41.943.040
50	20	30	52.428.800

Pengambilan data dengan D-ITG akan melibatkan empat skenario yang terdiri dari dua komponen protokol data yang berbeda yaitu TCP dan UDP. Masing-masing protokol yang ada memiliki skenario parameter yang sama dengan jumlah besar data yang akan dikirimkan paling kecil adalah 10MB sampai yang terbesar adalah 50MB. Besar data yang dikirimkan tersebut akan dikirimkan selama 20 detik yang tentunya dilakukan pengambilan data atau pengujiannya sebanyak 30 kali.

Pengambilan data QoS yang memerlukan jumlah sebanyak 30 kali tentunya akan melelahkan dan juga memakan banyak waktu apabila dilakukan secara manual, oleh sebab itu disini peneliti mencoba mengotomatisasi proses pengambilan data atau lebih tepatnya proses pengiriman traffik D-ITG yang dilakukan oleh *sender* tersebut dengan membuat sebuah program automatisasi.

```
#/bin/bash
Send=ITGSend
IP=192.168.20.2

#PENGIRIMAN PROTOKOL UDP

#10MB udp
for ((c=1; c<=30; c++))
do
echo "Pengambilan data UDP 10MB"
$Send -a $IP -c 10485760 -C 30 -t 20000 -T UDP -l /home/anand/test2/sk3/udp/10m$
sleep 5
done
```

Gambar 3.16 Program Otomatisasi Pengambilan Data

Pengambilan data akan dibuat ke dalam sebuah *script* otomatisasi sehingga mempercepat sekaligus membuat pengambilan lebih efisien. Gambar 3.16

mencantumkan bagaimana kode program tersebut dimana kode dituliskan menggunakan *python 3.* Jumlah 30 pengambilan atau pengujian yang diperlukan akan menggunakan fungsi perulangan yang ada dengan badan perulangan mencakup bagaimana perintah-perintah yang akan dimasukkan pengirim saat pengambilan data sesuai dengan skenario yang ada.

3.7.2 Pengambilan Nilai Konvergensi

Nilai konvergensi jaringan didapatkan melalui pengujian dari skenario *failover*, yaitu skenario memutuskan sebuah *link* saat pengiriman paket sedang berlangsung. Berbeda dengan pengambilan data QoS yang melibatkan skenario pengujian pertama sampai keempat serta menggunakan bantuan *software* D-ITG. Pengambilan data nilai konvergensi memanfaatkan *tool ping* yang dikirimkan dari masing-masing *client* baik itu saat proses komunikasi antar SPOKEnya atau komunikasi *client* SPOKE menuju *client* HUB, pada saat proses pengiriman trafik *ping* berlangsung tersebut HUB utama akan dimatikan untuk menguji berapa lama waktu konvergensi jaringan dari kondisi yang *down* sampai bisa lanjut berkomunikasi lagi.