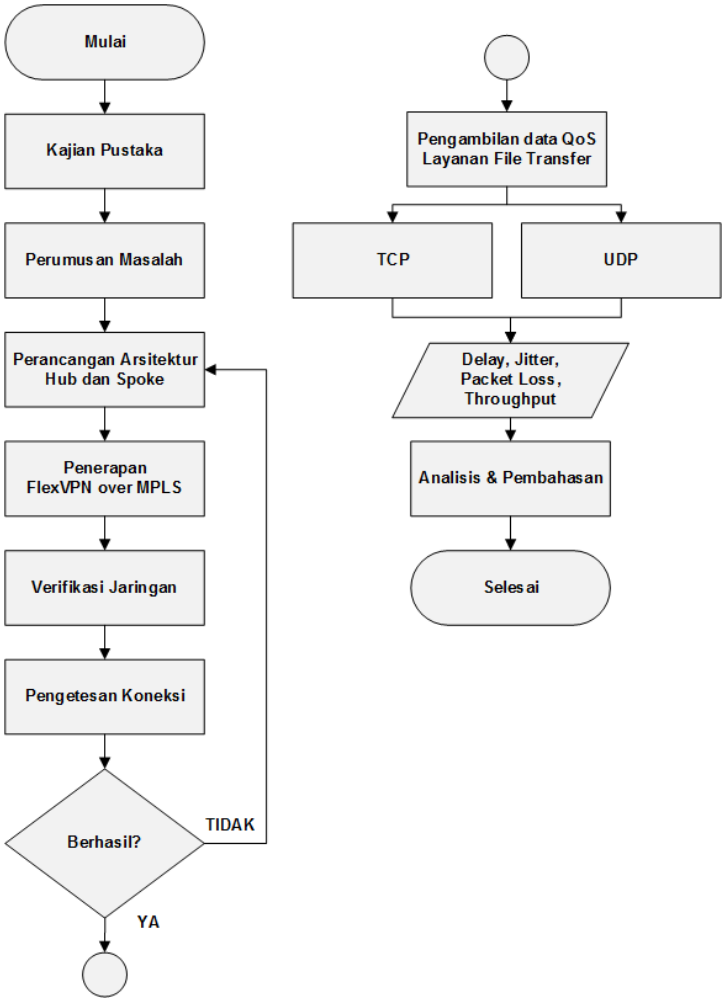
BAB 3
METODE PENELITIAN

3.1 Alur Penelitian

Alur penelitian berperan sebagai panduan langkah-langkah yang harus dilakukan untuk menyelesaikan penelitian supaya berjalan dengan sesuai rencana. Alur penelitian ini menggambarkan serangkaian tahapan yang direncanakan untuk dijalankan secara sistematis dan terperinci, sehingga dapat menuju pencapaian tujuan yang sudah direncanakan. Beberapa tahapan ini didesain untuk mengarahkan proses penelitian agar berjalan sesuai dengan rencana yang telah ditetapkan. Alur penelitian ini ditunjukkan berupa *flowchart* pada Gambar 3.1.



Gambar 3.1 Alur penelitian

Berdasarkan Gambar 3.1 penelitian ini dilakukan dengan meninjau beberapa referensi dan kajian pustaka yang sudah dikumpulkan untuk mendapatkan suatu permasalahan dari penelitian sebelumnya. Maksud dari kajian pustaka yaitu untuk membedakan penelitian yang dilakukan dengan penelitian sebelumnya. Perumusan masalah dapat diperoleh ketika sudah mendapatkan suatu skema berdasarkan referensi dan kajian pustaka dari penelitian sebelumnya. Perumusan masalah yang diangkat adalah bagaimana analisis kinerja jaringan FlexVPN *over* MPLS menggunakan parameter *quality of service* yang meliputi nilai dari parameter *delay, jitter*, *packet loss,* dan *throughput* terhadap penggunaan *transport protocol* TCP dan UDP pada layanan *file transfer.*

Tahapan perancangan sistem *hub* dan *spoke* sampai pengetesan koneksi merupakan tahapan yang dilakukan untuk mengimplementasikan FlexVPN *over* MPLS. Apabila jaringan FlexVPN *over* MPLS sudah terverifikasi maka dilanjutkan dengan pengambilan data *quality of service* pada layanan *file transfer* menggunakan *transport protocol* TCP dan UDP. Hasil pengambilan data tersebut menghasilkan beberapa parameter informasi meliputi nilai dari *delay*, *jitter*, *packet loss*, dan *throughput*. Hasil nilai parameter tesebut dapat dikaitkan dan dianalisis berdasarkan pengaruh jaringan FlexVPN *over* MPLS terkait *quality of service* penggunaan layanan *file transfer* terhadap *transport protocol* TCP dan UDP.

3.2 Metode Penelitian

Metode simulasi adalah metode yang digunakan untuk mengimplementasikan arsitektur *hub* dan *spoke* pada jaringan FlexVPN over MPLS menggunakan perangkat lunak EVE-NG. Perangkat yang digunakan meliputi 3 *router,* 2 *switch,* 2 virtual PC, dan internet. Proses komunikasi yang dilakukan meliputi router spoke1 dengan *router spoke*2 yang masing-masing terdapat satu virtual PC. *Router* hub berfungsi sebagai pusat informasi VPN dari jaringan FlexVPN *over* MPLS yang digunakan untuk mengkoneksikan antar *router spoke*. Pengujian melibatkan dua skenario yaitu TCP berupa *file transfer* dan UDP berupa video *streaming*. Hasil penelitian yang didapatkan berdasarkan simulasi berupa *delay, jitter, packet loss,* dan *throughput.* Tabel 3.1 menunjukkan perangkat simulasi yang digunakan dalam penelitian.

Tabel 3.1 Perangkat simulasi penelitian

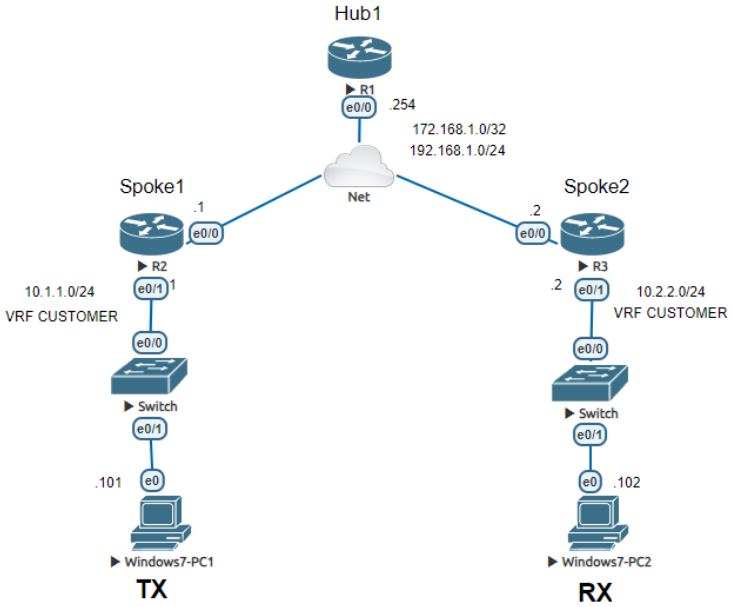
No	Perangkat	Jumlah	Keterangan
1	*Router*	3	Router yang digunakan yaitu Cisco C7300-ADVENTERPRISE9-15.5.2T.bin
2	Internet	1	Perantara antara router *hub* sebagai pusat informasi VPN dengan router *spoke*
3	*Switch*	2	*Switch* yang digunakan yaitu Cisco L2-ADVENTERPRISE-M-15.1-20140814.bin
4	Kabel *Ethernet*	7	Media penghubung keseluruhan perangkat yaitu antara *router*, *switch*, internet, dan PC
2	Virtual PC	2	Virtual PC menggunakan sistem operasi *Windows* 7

Tabel 3.2 menunjukkan pengalamatan IP *interface* yang digunakan dalam penelitian.

Tabel 3.2 Pengalamatan IP *interface*

Perangkat	*Port Interface*	IP *Address*
Router Hub	*Ethernet*0/0	192.168.1.254/24
	*Loopback*0	172.16.1.254/32
	Virtual-*Access*1	172.16.1.254/32
	Virtual-*Template*1	172.16.1.254/32
Router Spoke1	*Ethernet*0/0	192.168.1.1/24
	*Ethernet*0/1	10.1.1.1/24
	*Loopback*0	172.16.1.1/32
	*Tunnel*0	172.16.1.1/32
	Virtual-*Template*1	172.16.1.1/32
Router Spoke2	*Ethernet*0/0	192.168.1.2/24
	*Ethernet*0/1	10.2.2.2/24
	*Loopback*0	172.16.1.2/32
	*Tunnel*0	172.16.1.2/32
	Virtual-*Template*1	172.16.1.2/32

Gambar 3.2 menunjukkan arsitektur *hub* dan *spoke* digunakan pada jaringan FlexVPN *over* MPLS.



Gambar 3.2 Arsitektur *hub* dan *spoke* jaringan FlexVPN over MPLS

Berdasarkan Gambar 3.2 dapat dijelaskan bahwa sistem jaringan FlexVPN *over* MPLS pada penggunaan arsitektur *hub* dan *spoke* mempunyai *router spoke* yang membentuk asosiasi keamanan IKEv2 dan IPSec dengan *router hub*. MP-BGP digunakan untuk mengiklankan label untuk jaringan di setiap VRF berupa rute VPN dari *router hub* dengan *network* 10.0.0.0/8 sehingga *router spoke* mengetahui cara menjangkau *network* satu sama lain khususnya pada setiap VRF. Sementara asosiasi keamanan IKEv2 digunakan untuk pertukaran kebijakan informasi perutean dan keamanan antar peer layaknya klien dan server pada alamat *overlay* (virtual). IPsec digunakan sebagai mekanisme enkripsi paket data untuk memfasilitasi kebutuhan setiap VTI.

Apabila TX mengirimkan paket ke RX, proses yang terjadi diawali *switch* akan menerima paket dari sumber, kemudian memeriksa alamat tujuan yang tercantum dalam header paket untuk menentukan arah pengiriman. *Router spoke*1 menerima paket IP, menambahkan label yang cocok dengan rute ringkasan dari *router hub*, merangkum paket IP dengan VTI dan IPsec, dan meneruskannya melalui virtual *tunnel interface* ke *router hub*. *Router hub* akan mendekripsi paket IP dan mencari label yang diperlukan untuk mencapai 10.2.2.102 serta menambahkan label yang cocok dengan *network* 10.2.2.0/24, merangkum paket IP dengan VTI dan IPsec, dan mengirimkan paket ke *router spoke*2. *Router hub* juga mengirimkan paket pengalihan NHRP ke *router spoke*1 dan menambahkan label

yang cocok dengan awalan 10.1.1.0/24 tempat asal paket IP sehingga *router spoke*1 menerima dan memproses paket pengalihan NHRP yang memicu permintaan resolusi NHRP. *Router spoke*1 dapat membuat entri pemetaan NHRP dan mengaitkannya dengan VRF "CUSTOMER" untuk awalan 10.2.2.0/24 sehingga dapat mengirimkan permintaan resolusi NHRP ke *router hub*. Permintaan resolusi NHRP menyertakan ID permintaan yang diperlukan saat menerima balasan resolusi NHRP.

Apabila *router hub* menerima permintaan resolusi NHRP dari *router spoke*1 maka *router hub* dapat mencari label untuk 10.2.2.0/24, dan meneruskan permintaan resolusi NHRP ke *router spoke*2. *Router Spoke*2 menerima permintaan resolusi NHRP dan membuat *virtual access interface* sehingga dapat memulai IKEv2 dan IPSec SA dengan *router spoke*1. NHRP menginstal rute untuk alamat IP *virtual access interface* untuk *router spoke*1. *Router spoke*2 mengirimkan balasan resolusi NHRP ke *router spoke*1 menggunakan *virtual access interface* berupa label yang digunakan *router spoke*1 untuk mengirim paket IP melalui *spoke-to-spoke tunnel*. NHRP mendapatkan label tersebut dari MPLS *Forwarding Infrastructure (MFI)*. Balasan Resolusi NHRP juga meliputi label *null implisit* untuk alamat IP *virtual access interface router spoke*2. *Router spoke1* menerima balasan resolusi NHRP pada *virtual access interface*nya. Balasan resolusi NHRP menyertakan ID permintaan yang ditambahkan oleh *router spoke*1 dalam permintaan resolusi NHRP. Cara tersebut digunakan *router spoke*1 mengetahui VRF mana yang menjadi milik balasan resolusi NHRP. *Router spoke*1 mencari entri NHRP di *cache* NHRP dan memasukkan rute di tabel *routing* VRF sehingga setiap client dari *spoke*1 dan *spoke*2 dapat melabeli paket *switch* secara langsung menggunakan *virtual access interface* untuk VRF.

Enkapsulasi MPLS dimulai saat paket tiba di *ingress router*. *Router* pertama yaitu *router spoke*1 berperan sebagai *ingress router* akan memberikan label kepada paket. Label tersebut memiliki makna dalam konteks LSP (*Label Switched Path*). LSP bertugas untuk mengarahkan setiap paketnya ke dalam *traffic* jaringan MPLS. Setelah label diterapkan, paket akan diteruskan ke *router* berikutnya dalam jalur LSP. *Router* akan membaca label dan meneruskan paket yang sudah diberi label tersebut ke dalam *traffic* jaringan sesuai tujuannya. Pada akhir jalur LSP, *router*

terakhir yaitu *spoke*2 berperan sebagai *egress router* akan mengidentifikasi label dan meneruskan paket tersebut ke tujuan akhir berdasarkan alamat tujuannya.

3.3 Skenario Penelitian

Simulasi pengujian melibatkan PC1 yang digunakan sebagai server/pengirim dengan PC2 yang digunakan sebagai penerima/*client*. Masing-masing virtual PC tersebut berada di perangkat *spoke*. Pengujian ini bertujuan untuk mengetahui kinerja *quality of service* yang dihasilkan dalam penggunaan arsitektur *hub* dan *spoke* khususnya pada jaringan FlexVPN *over* MPLS karena pada arsitektur tersebut mempunyai perangkat pusat atau *router hub* yang digunakan sebagai pusat ingformasi VPN serta digunakan untuk mengkoneksikan ke *router spoke* tanpa melibatkan *router hub* sebagi titik transit paket sehingga bentuk komunikasi *spoke-to-spoke* dapat terbentuk. Parameter penelitian yang digunakan berupa *throughput*, *delay*, *jitter*, dan *packet loss*. Berikut adalah skenario penelitian secara detail yang digunakan.

3.3.1 Skenario Layanan *File Transfer* (*TCP* dan *UDP*)

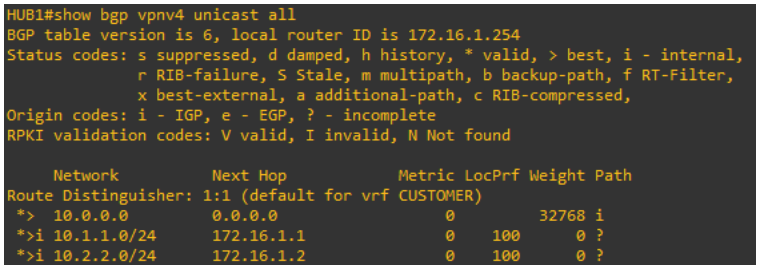
Skenario pengujian layanan *file transfer* menggunakan *transport protocol* TCP dan UDP dilakukan dengan melakukan *file transfer* yang dilakukan oleh sisi server/pengirim dan kemudian sisi *client*/penerima men*download* terkait *file* yang diberikan oleh sisi *server*/pengirim. *File* pengujian yang digunakan berformat ISO dengan ukuran file sebesar 20 MB, 40 MB, dan 80 MB. Perangkat lunak *filezilla* server digunakan untuk menyediakan *file* apa saja yang dikirimkan kepada *client*/penerima. Perangkat lunak *filezilla client* digunakan untuk melakukan proses *download* berupa *file* yang sudah disediakan oleh pihak server/pengirim. Pada saat proses *download*, tahap *capture data* dari sisi *client*/penerima dapat dilakukan menggunakan perangkat lunak *wireshark* sehingga setiap paket dapat ditangkap sampai proses download selesai dan hasil data akan didapatkan.

3.4 Verifikasi *Router Hub*

Verifikasi diperlukan untuk memastikan konfigurasi yang sudah dilakukan. Berikut adalah beberapa verifikasi pada *router hub*.

3.4.1 Verifikasi *show bgp vpnv4 unicast all*

Perintah *show bgp vpnv4 unicast all* berfungsi untuk menampilkan *database* informasi rute alamat VPNv4 secara lengkap dari tabel BGP. Verifikasi *show* bgp vpnv4 *unicast all* pada *router hub* ditunjukkan pada Gambar 3.3.

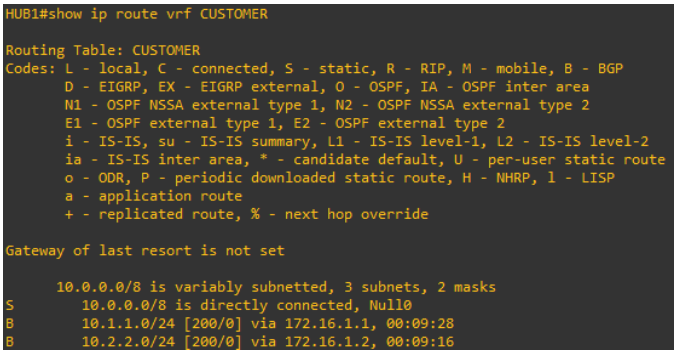


Gambar 3.3 Verifikasi *show bgp vpnv4 unicast all* (*router hub1*)

Berdasarkan Gambar 3.3 perintah "*show bgp vpnv4 unicast all*" berfungsi untuk menampilkan *database* informasi rute alamat VPNv4 secara lengkap dari tabel BGP. MP-BGP pada *router hub* merupakan pusat *peer grou*p untuk *router spoke*. Fungsinya adalah untuk mengaktifkan *address-family* VPNv4 dan mengiklankan label pada VTI dengan jaringan 10.0.0.0 di masing-masing VRF sehingga jalur *tunnel overlay* yang digunakan akan berpusat di pusat *peer group hub* dan antar *router spoke* tahu cara menjangkau jaringan satu sama lain berkat MP-BGP. *Network* VRF 10.1.1.0/24 dan 10.2.2.0/24 diiklankan oleh *routing static* pada masing-masing *spoke* sehingga *network* VRF tersebut dapat diintegrasikan oleh *MP-BGP* sebagai rute VPN melalui label pada VTI. *Address* 0.0.0.0 merupakan IKEv2 *keyring* yang melayani sekumpulan peer yang cocok dengan IKEv2 *profile*. IKEv2 *keyring* mendapatkan konteks VRF dari IKEv2 *profile*. Sementara, IKEv2 *authorization policy* digunakan untuk mengiklankan alamat IP *tunnel* melalui IKEv2. IP *loopback* digunakan untuk *pointing* dari ip *address* sebagai Virtual-*Template*1 *tunnel* dan tunnel0 dengan *protection* IPsec.

3.4.2 Verifikasi *show ip route vrf CUSTOMER*

Perintah *show ip route* vrf CUSTOMER berfungsi untuk menampilkan tabel *routing* yang terkait dengan VRF. Verifikasi *show ip route* vrf CUSTOMER pada *router hub* ditunjukkan pada Gambar 3.4.

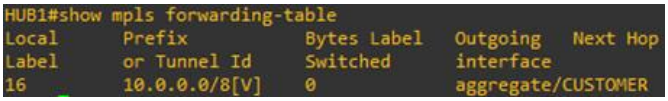


Gambar 3.4 *show ip route vrf CUSTOMER* (*router hub*1)

Berdasarkan Gambar 3.4 perintah "*show ip route* vrf CUSTOMER" berfungsi untuk menampilkan tabel *routing* yang terkait dengan VRF. Rute *static* mempunyai *network* 10.0.0.0/8 yang merupakan titik *endpoint* dari pusat informasi VPN. Berdasarkan *output* diatas *router spoke* mengiklankan network VRF-nya masing-masing menggunakan MP-BGP yaitu VRF CUSTOMER 10.1.1.0/24 via 172.16.1.1 dari *router spoke*1 dan VRF CUSTOMER 10.2.2.0/24 via 172.16.1.2 dari router *spoke*2. Masing-masing MP-BGP dari kedua *spoke* mempunyai titik *endpoint* pusat informasi VPN yang sama yaitu menuju ke hub. IKEv2 mempunyai kebijakan melakukan pembagian informasi routing antara peer VPN yaitu masing-masing VRF pada *spoke* dengan tujuan *tunnel overlay*nya menuju *hub*.

3.4.3 Verifikasi *show mpls forwarding-table*

Perintah *show mpls forwarding-table* berfungsi untuk menampilkan menampilkan isi tabel penerusan melalui MPLS. Verifikasi *mpls forwarding-table* pada *router hub* ditunjukkan pada Gambar 3.5.



Gambar 3.5 Verifikasi *show mpls forwarding-table* (*router hub*1)

Berdasarkan Gambar 3.5 perintah "*show mpls forwarding-table*" berfungsi untuk menampilkan isi tabel penerusan melalui MPLS. Berdasarkan *output* diatas tujuan *traffic* 10.0.0.0/8 pada pusat informasi *hub* sudah terlingkupi *tunnel* dan label MPLS.

3.5 Verifikasi *Router Spoke*1

Verifikasi diperlukan untuk memastikan konfigurasi yang sudah dilakukan. Berikut adalah beberapa verifikasi pada *router spoke*1.

3.5.1 Verifikasi *show bgp vpnv4 unicast all*

Perintah *show bgp vpnv4 unicast all* berfungsi untuk menampilkan *database* informasi rute alamat VPNv4 secara lengkap dari tabel BGP. Verifikasi *show* bgp vpnv4 *unicast all* pada *router spoke*1 ditunjukkan pada Gambar 3.6.

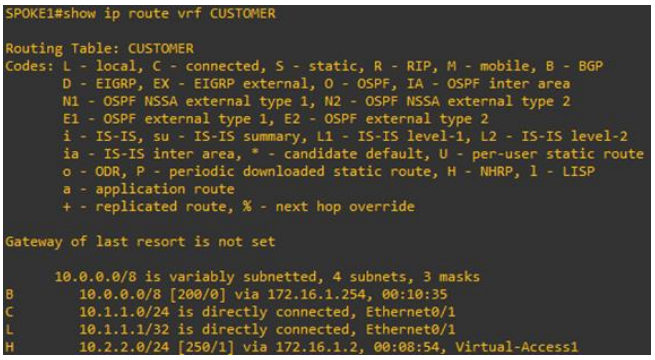
```
SPOKE1#show bgp vpnv4 unicast all
BGP table version is 4, local router ID is 172.16.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:1 (default for vrf CUSTOMER)
 *>i 10.0.0.0         172.16.1.254             0    100      0 i
 *>  10.1.1.0/24      0.0.0.0                  0         32768 ?
```

Gambar 3.6 Verifikasi *show bgp vpnv4 unicast all* (*router spoke*1)

Berdasarkan Gambar 3.6 perintah "*show bgp vpnv4 unicast all*" berfungsi untuk menampilkan *database* informasi rute alamat VPNv4 secara lengkap dari tabel BGP. Berdasarkan *output* diatas rute ringkasan 10.0.0.0 merupakan pusat *peer group* MP-BGP yaitu dari *router hub* sehingga *router spoke* tahu cara menjangkau jaringan satu sama lain melalui pengalihan label pada VTI. *Router spoke*1 mengiklankan *network* VRF CUSTOMER 10.1.1.0/24 pada *ethernet* 0/1 dan terhubung langsung dengan pusat *peer group hub* berkat MP-BGP. IP *loopback* digunakan untuk *pointing* dari ip address sebagai Virtual-*Template*1 *tunnel* dan *tunnel*0 dengan *protection* IPsec yang bersumber dari *interface Ethernet*0/0 dari *router hub*. *Address* 0.0.0.0 merupakan IKEv2 *keyring* yang melayani sekumpulan *peer* yang cocok dengan IKEv2 *profile*. IKEv2 *keyring* mendapatkan konteks VRF dari IKEv2 *profile*. Sementara, IKEv2 *authorization policy* digunakan untuk mengiklankan alamat IP *tunnel* melalui IKEv2.

3.5.2 Verifikasi *show ip route vrf CUSTOMER*

Perintah *show ip route* vrf CUSTOMER berfungsi untuk menampilkan tabel *routing* yang terkait dengan VRF. Verifikasi *show ip route* vrf CUSTOMER pada *router spoke*1 ditunjukkan pada Gambar 3.7.

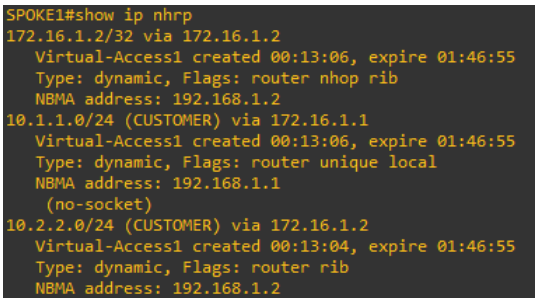


Gambar 3.7 Verifikasi *show* ip *route* vrf CUSTOMER (*router spoke*1)

Berdasarkan Gambar 3.7 perintah "*show* ip *route* vrf CUSTOMER" berfungsi untuk menampilkan tabel *routing* yang terkait dengan VRF. Pada tabel perutean diatas bahwa setiap *router spoke* memasang rute ringkasan MP-BGP dari pusat *network peer group hub* yang mana merupakan titik *endpoint* pusat informasi VPN. *Router spoke*1 mempunyai satu rute BGP yang merupakan jalur terhubung *tunnel overlay* dari *routing static* VRF ke 10.0.0.0/8 via ip *loopback* 172.16.1.254 yang mengarah ke *router hub* memanfaatkan VTI *static*. *Router spoke*1 mempunyai entri NHRP untuk *spoke-to-spoke traffic* ke *network* VRF CUSTOMER router spoke2 yaitu 10.2.2.0/24 via ip *loopback* 172.16.1.2 melalui Virtual-*Access*1. MPLS-NHRP pada VTI *dynamic* juga digunakan untuk melabeli *interface tunnel* yang terhubung ke *router hub*.

3.5.3 Verifikasi *show ip nhrp*

Perintah *show ip nhrp* berfungsi untuk memeriksa NHRP secara detail. Verifikasi *show ip route nhrp* pada *router spoke*1 ditunjukkan pada Gambar 3.8.

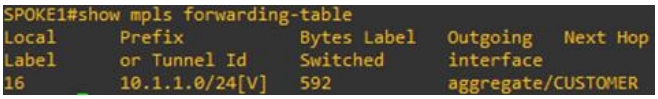


Gambar 3.8 Verifikasi *show* ip nhrp (*router spoke*1)

Berdasarkan Gambar 3.8 perintah "*show ip nhrp*" berfungsi untuk memeriksa NHRP secara detail. NHRP digunakan untuk menyelesaikan alamat *overlay* jarak jauh dan secara *dynamic* menemukan *end-point transport* yang diperlukan untuk membuat *tunnel* yang aman. NHRP memerlukan router untuk meminta *next-hop* IP dari IKEv2. NHRP membutuhkan VTI *dynamic* dalam mengefisiensikan pembentukan koneksi VPN melalui *tunnel* berdasarkan informasi dari *hub* terkait masing-masing *network* VRF. Hasilnya router *hub* akan membuat *database* NHRP antar *router spoke* sehingga tidak melibatkan *router hub* sebagai titik transit paket untuk sampai ke tujuan. Berdasarkan *output* diatas menghasilkan *router spoke* sudah dapat menjangkau jaringan satu sama lain berkat NHRP melalui Virtual-*Access*1. MPLS-NHRP juga digunakan pada VTI *dynamic* untuk melabeli *interface tunnel* yang terhubung ke *router spoke* lainnya.

3.5.4 Verifikasi *show mpls forwarding-table*

Perintah *show mpls forwarding-table* berfungsi untuk menampilkan menampilkan isi tabel penerusan melalui MPLS. Verifikasi *mpls forwarding-table* pada *router spoke1* ditunjukkan pada Gambar 3.9.

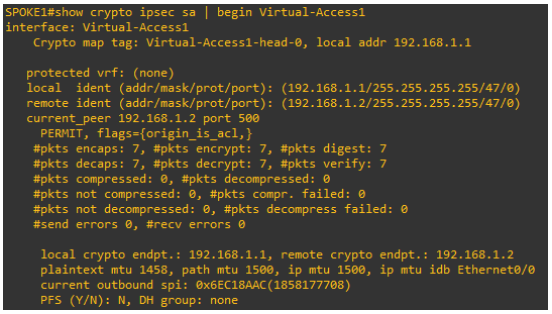


Gambar 3.9 Verifikasi *show mpls forwarding-table* (*router spoke*1)

Berdasarkan Gambar 3.9 perintah "*show mpls forwarding-table*" berfungsi untuk menampilkan isi tabel penerusan melalui MPLS. Berdasarkan *output* diatas tujuan *traffic* 10.1.1.0/24 pada perangkat *spoke* sudah terlingkupi *tunnel* dan label MPLS.

3.5.5 Verifikasi *show crypto ipsec sa | begin Virtual-Access1*

Perintah *show show crypto ipsec sa | begin* Virtual-*Access1* berfungsi untuk untuk menampilkan status IPsec *Security Association* pada *interface* Virtual-*Access*1. Verifikasi *show show crypto ipsec sa | begin* Virtual-*Access1* pada *router spoke*1 ditunjukkan pada Gambar 3.10.



Gambar 3.10 Verifikasi *show crypto* ipsec sa | *begin* Virtual-*Access*1 (*router spoke*1)

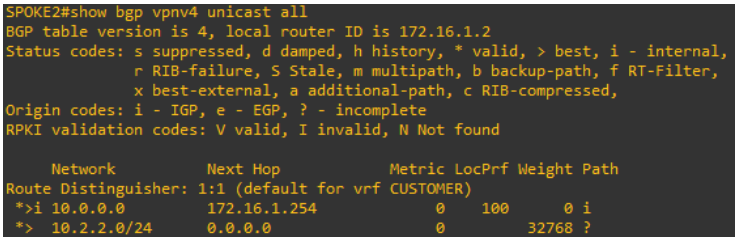
Berdasarkan Gambar 3.10 perintah "*show show crypto ipsec sa | begin* Virtual-*Access1*" berfungsi untuk menampilkan status IPsec SA pada *interface* Virtual-*Access*1. Berdasarkan *output* diatas *spoke-to-spoke traffic* sudah terkonfirmasi karena paket dikirim langsung di antara router *spoke* dan paket dienkripsi di antara *router spoke*. Pernyataan tersebut dapat dikonfirmasi karena output diatas menampilkan *current_peer* 192.168.1.2 yang merupakan *address* milik *router spoke*2.

3.6 Verifikasi *Router Spoke*2

Verifikasi diperlukan untuk memastikan konfigurasi yang sudah dilakukan. Berikut adalah beberapa verifikasi pada *router Spoke*2.

3.6.1 Verifikasi *show bgp vpnv4 unicast all*

Perintah *show bgp vpnv4 unicast all* berfungsi untuk menampilkan *database* informasi rute alamat VPNv4 secara lengkap dari tabel BGP. Verifikasi *show* bgp vpnv4 *unicast all* pada *router spoke*2 ditunjukkan pada Gambar 3.11.



Gambar 3.11 Verifikasi *show* bgp vpnv4 *unicast all* (*router spoke*2)

Berdasarkan Gambar 3.11 perintah "*show bgp vpnv4 unicast all*" berfungsi untuk menampilkan *database* informasi rute alamat VPNv4 secara lengkap dari tabel BGP. Berdasarkan output diatas rute ringkasan 10.0.0.0 merupakan pusat *peer group* MP-BGP yaitu dari *router hub* sehingga *router spoke* tahu cara menjangkau jaringan satu sama lain melalui pengalihan label pada VTI. *Router spoke*2 mengiklankan network VRF CUSTOMER 10.2.2.0/24 pada ethernet 0/1 dan terhubung langsung dengan pusat *peer group hub* berkat MP-BGP. IP *loopback* digunakan untuk pointing dari ip address sebagai Virtual-*Template*1 *tunnel* dan *tunnel*0 dengan *protection* IPsec yang bersumber dari *interface Ethernet*0/0 dari *router hub*. *Address* 0.0.0.0 merupakan IKEv2 *keyring* yang melayani sekumpulan *peer* yang cocok dengan IKEv2 *profile*. IKEv2 *keyring* mendapatkan konteks VRF dari IKEv2 *profile*. Sementara, IKEv2 *authorization policy* digunakan untuk mengiklankan alamat IP *tunnel* melalui IKEv2.

3.6.2 Verifikasi *show ip route* vrf CUSTOMER

Perintah *show ip route* vrf CUSTOMER berfungsi untuk menampilkan tabel *routing* yang terkait dengan VRF. Verifikasi *show ip route* vrf CUSTOMER pada *router spoke*2 ditunjukkan pada Gambar 3.12.

```
SPOKE2#show ip route vrf CUSTOMER

Routing Table: CUSTOMER
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 4 subnets, 3 masks
B        10.0.0.0/8 [200/0] via 172.16.1.254, 00:12:45
H        10.1.1.0/24 [250/1] via 172.16.1.1, 00:11:17, Virtual-Access1
C        10.2.2.0/24 is directly connected, Ethernet0/1
L        10.2.2.2/32 is directly connected, Ethernet0/1
```

Gambar 3.12 Verifikasi *show* ip *route* vrf CUSTOMER (*router spoke*2)

Berdasarkan Gambar 3.12 perintah *show* ip *route* vrf CUSTOMER berfungsi untuk menampilkan tabel *routing* yang terkait dengan VRF. Pada tabel perutean diatas bahwa setiap *router spoke* memasang rute ringkasan BGP dari *hub* yang mana merupakan titik *endpoint* pusat informasi VPN. *Router spoke*2 mempunyai satu rute BGP yang merupakan jalur terhubung *tunnel overlay* dari

routing static VRF ke 10.0.0.0/8 via ip *loopback* 172.16.1.254 yang mengarah ke *router hub* memanfaatkan VTI *static*. *Router Spoke*2 mempunyai entri NHRP untuk *spoke-to-spoke traffic* ke *network* VRF CUSTOMER *router spoke*1 yaitu 10.1.1.0/24 via ip *loopback* 172.16.1.1 melalui Virtual-*Access*1. MPLS-NHRP pada VTI *dynamic* juga digunakan untuk melabeli interface *tunnel* yang terhubung ke *router hub*.

3.6.3 Verifikasi *show ip nhrp*

Perintah *show ip nhrp* berfungsi untuk memeriksa NHRP secara detail. Verifikasi *show ip route nhrp* pada *router spoke*2 ditunjukkan pada Gambar 3.13.

```
SPOKE2#show ip nhrp
172.16.1.1/32 via 172.16.1.1
   Virtual-Access1 created 00:15:10, expire 01:44:51
   Type: dynamic, Flags: router nhop rib nho
   NBMA address: 192.168.1.1
10.1.1.0/24 (CUSTOMER) via 172.16.1.1
   Virtual-Access1 created 00:15:08, expire 01:44:51
   Type: dynamic, Flags: router rib
   NBMA address: 192.168.1.1
10.2.2.0/24 (CUSTOMER) via 172.16.1.2
   Virtual-Access1 created 00:15:10, expire 01:44:51
   Type: dynamic, Flags: router unique local
   NBMA address: 192.168.1.2
    (no-socket)
```

Gambar 3.13 Verifikasi *show* ip nhrp (*router spoke*2)

Berdasarkan Gambar 3.13 perintah "*show* ip nhrp" berfungsi untuk memeriksa NHRP secara detail. NHRP digunakan untuk menyelesaikan alamat *overlay* jarak jauh dan secara *dynamic* menemukan *end-point transport* yang diperlukan untuk membuat tunnel yang aman. NHRP memerlukan *router* untuk meminta *next-hop* IP dari IKEv2 sebagai pengalihan. NHRP membutuhkan VTI *dynamic* dalam mengefisiensikan pembentukan koneksi VPN melalui *tunnel* berdasarkan informasi dari hub terkait masing-masing VRF. Hasilnya *router hub* akan membuat *database* NHRP antar *router spoke* sehingga tidak melibatkan *router hub* sebagai titik transit paket untuk sampai ke tujuan. Berdasarkan *output* diatas menghasilkan router *spoke* sudah dapat menjangkau jaringan satu sama lain berkat NHRP melalui Virtual-*Access*1. MPLS-NHRP juga digunakan pada VTI *dynamic* untuk melabeli interface *tunnel* yang terhubung ke *router spoke* lainnya.

3.6.4 Verifikasi *show* mpls *forwarding-table*

Perintah *show mpls forwarding-table* berfungsi untuk menampilkan menampilkan isi tabel penerusan melalui MPLS. Verifikasi *mpls forwarding-table* pada *router spoke2* ditunjukkan pada Gambar 3.14.

Gambar 3.14 Verifikasi *show* mpls *forwarding-table* (*router spoke*2)

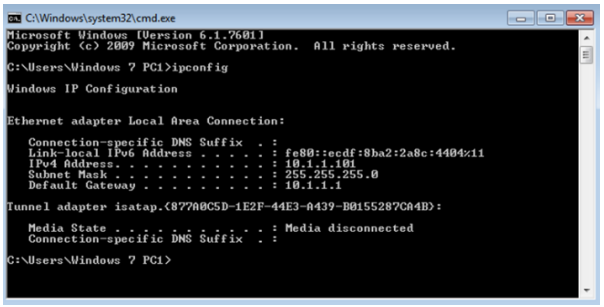
Berdasarkan Gambar 3.14 perintah "*show mpls forwarding-table*" berfungsi untuk menampilkan isi tabel penerusan melalui MPLS. Berdasarkan *output* diatas tujuan traffic 10.2.2.0/24 pada perangkat *spoke2* sudah terlingkupi *tunnel* dan label MPLS.

3.7 Verifikasi Sebelum Pengujian

Tujuan dari verifikasi yaitu untuk memastikan bahwa apa yang direncanakan sebelum pengambilan data sudah dilakukan dan benar. Verifikasi pertama dilakukan mengecek IP *address* di virtual PC1 dan virtual PC2 melalui *command prompt*. Verifikasi kedua dilakukan pengetasan ping virtual PC1 ke virtual PC2 melalui *command prompt*. Verifikasi ketiga dilakukan pengecekan jalur paket yang dilalui dari server/pengirim ke *client*/penerima.

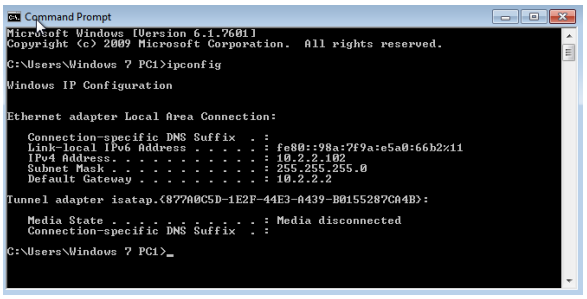
3.7.1 Verifikasi Konfigurasi Virtual PC

Gambar 3.15 berikut menunjukkan hasil pengecekan IP pada virtual PC1 melalui *command prompt*.



Gambar 3.15 Verifikasi IP *address* PC1

Berdasarkan Gambar 3.15 sudah terkonfirmasi virtual PC1 mempunyai ip address 10.1.1.101 dengan *subnet mask* /24 atau 255.255.255.0. *Default gateway* pada virtual PC1 menggunakan ip *address* pada *router spoke*1 dengan *interface Ethernet*0/1 yaitu 10.1.1.1 Pada sisi *router spoke*1 merupakan VRF CUSTOMER A dengan *network* 10.1.1.0/24. Gambar 3.16 berikut menunjukkan hasil pengecekan IP pada virtual PC2 melalui *command prompt.*

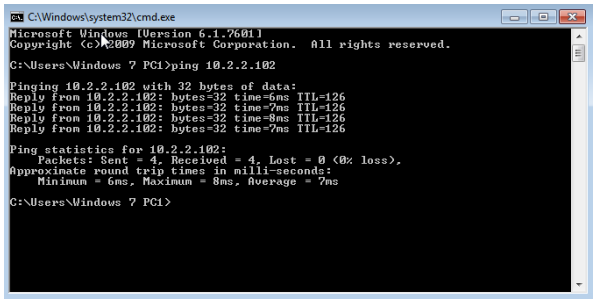


Gambar 3.16 Verifikasi IP *address* PC2

Berdasarkan Gambar 3.16 sudah terkonfirmasi virtual PC2 mempunyai ip address 10.2.2.102 dengan *subnet mask* /24 atau 255.255.255.0. Default gateway pada virtual PC1 menggunakan ip *address* pada *router spoke*2 dengan *interface ethernet*0/1 yaitu 10.2.2.2. Pada sisi *router spoke*2 merupakan VRF CUSTOMER A dengan *network* 10.2.2.0/24.

3.7.2 Verifikasi Pengetesan Ping PC1 ke PC2

Gambar 3.17 berikut menunjukkan hasil pengetesan ping virtual PC1 ke virtual PC2 melalui *command prompt*.



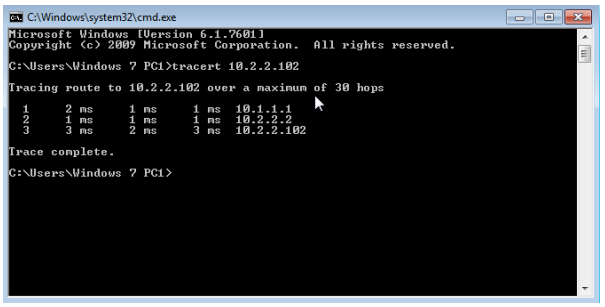
Gambar 3.17 Verifikasi Pengetesan ping PC1 ke PC2

Berdasarkan Gambar 3.17 sudah terkonfirmasi virtual PC1 dengan ip *address* 10.1.1.101/24 sudah terkoneksi dengan virtual PC2 yang mempunyai ip

address 10.2.2.102/24 sehingga kedua virtual PC tersebut sudah dapat melakukan komunikasi.

3.7.3 Verifikasi Pengetesan *Tracert* PC1 ke PC2

Gambar 3.18 berikut menunjukkan hasil pengetesan *tracert* virtual PC1 ke virtual PC2 melalui *command prompt*.



Gambar 3.18 Verifikasi rute paket

Berdasarkan Gambar 3.18 sudah terkonfirmasi bahwa rute paket yang dilalui tidak melewati *router hub* karena *router* tersebut merupakan pusat informasi VPN yang digunakan mengkoneksikan antara *router spoke* sehingga terbentuk komunikasi *spoke-to-spoke* yang mana tidak melibatkan *router hub* sebagai titik transit paket untuk sampai ke penerima.