BAB 2
DASAR TEORI

2.1 KAJIAN PUSTAKA

Penelitian [4] membahas mengenai Analisis Performansi *Dynamic Multipoint Virtual Private Network* pada *Routing Protocol* BGP dengan FRRouting. Pada penelitian ini, hasil didapatkan dari nilai *Quality of Service* (QoS) dengan parameter *throughput, delay, jitter*, dan *packet loss* dengan standar TIPHON. Hasil pengukuran menunjukkan hampir semua parameter tergolong dalam kategori yang sangat bagus. Nilai *throughput* tertinggi ada pada 3551 Kbps, *delay* terkecilnya adalah 0,43 detik, nilai *jitter* terkecil adalah 0,324 ms, dan *packet loss* yang dihasilkan adalah 0%. Pada penelitian ini dapat disimpulkan bahwa DMVPN mampu berjalan dengan sangat baik.

Penelitian [5] membahas mengenai metode *Internet Protocol Secuity* (IPSec) dengan *Virtual Private Network* (VPN) untuk komunikasi data yang bertujuan untuk mengimplementasikan VPN dengan metode IPSec pada PT. Penas yang akan digunakan untuk berkomunikasi dengan kantor pusat yaitu PT. PPA secara cepat dan aman. Dalam penelitian ini menggunakan metode *Network Development Life Cycle* (NDCL), penelitian ini menunjukan keberhasilan dengan mengirimkan data dari PC *client* PT.Penas ke PC *client* PT.PPA dengan data yang sudah terenkripsi sehinggan komunikasi data menjadi lebih aman.

Penelitian [6] membahas mengenai Analisis dan Simulasi *Routing Border Gateway Protocol* (BGP) antar *Autonomous System* Menggunakan *Free Rang Routing* (FRR) yang bertujuan untuk membandingkan dua topologi *mesh* dengan topologi pertama dengan 4 router FRR dan topologi kedua dengan 6 router FRR dengan melakukan pengujian kedua topologi tersebut dengan skenario *failover*, protokol yang digunakan adalah UDP, sedangkan proses pengiriman trafiknya menggunakan D-ITG. Penelitian ini menghasilkan nilai *throughput, jitter, delay* dan *packet loss* masuk kedalam kategori sangat baik pada skenario tanpa *failover* dan kategori sangat baik hingga sedang pada skenario *failover* jika merujuk pada standar TIPHON ETSI.

Penelitian [8] membahas mengenai *Analyzing Generic Routing Encapsulation* (GRE) *and IP Security* (IPSec) T*unneling Protocols for Secured*

Communication over Public Networks, penelitian ini bertujuan untuk analisis kelebihan dan kelemahan protokol *tunneling*, dikarnakan perluasan akses internet terhubung membutuhkan keamanan dan privasi yang baik pada data di dalam jaringan, menerapkan protokol *tunneling* dapat membantu melawan serangan *cyber* terkait lapisan jaringan. Penelitian ini menghasilkan sekumpulan paket terenkripsi yang akan sulit didekripsi oleh penyerang. GRE *over* IPSec memberikan bentuk keamanan terbaik pada tingkat lapisan ketiga (*network layer*).

Penelitian [9] membahas mengenai simulasi dan analisa QoS dalam jaringan VPN *site to site* berbasis IPSec dengan *routing dynamic*, simulasi penelitian ini dilakukan pada emulator berbasis *web* bernama Eve-NG dengan mensimulasikan sebuah perusahaan dengan satu kantor pusat dan dua kantor cabang serta *server* yang berada pada *data center*, simulasi dilakukan pada layanan VoIP dengan *server* asterisk. Penelitian ini menunjukan keberhasilan Uji performansi QoS untuk *routing* OSPF, RIPv2 dan EIGRP dengan menunjukan pengujian semua parameter masih dalam batas kualitas standar ITU-T.

Penelitian [10] membahas mengenai perancangan jaringan *Virtual Private Network* berbasis *IP Security* menggunakan *router* Mikrotik, penelitian ini bertujuan agar suatu komunikasi dapat berlangsung aman dengan mengimplementasikan VPN IPSec sehingga terbentuknya jalur komunikasi bersifat *private*. Hasil dari penelitian ini yang sebelumnya data dikirimkan menggunakan Internet publik, tetapi data kurang aman karena data mudah terbaca oleh pihak lain. Sebagai solusinya dapat mengimplementasikan jaringan VPN IPsec sebagai jalur khusus untuk proses pengiriman dengan melakukan enksripsi data menggunakan IPSec sehingga data menjadi aman dan tidak mudah untuk dibaca oleh pihak lain.

Pada table 2.1 menunjukan perbandingan terhadap penelitian yang sebelumnya sudah dilakukan dan terkait dalam penelitian penulis.

Tabel 2. 1 Perbandingan dengan Penelitian Sebelumnya

Peneliti	Judul	Komponen Penelitian				
		Metode	Parameter pengujian	Jenis *Router*	Protokol *Routing*	Hasil
Nanda Iryani, Dyas Dendi Andika (2021)	Analisis Performansi *Dynamic Multipoint Virtual Private Network* pada *Routing Protocol* BGP dengan FRRouting	DMVPN	*Throughput, delay, jitter, packet loss*	FRRouting	BGP	Sangat bagus (standar TIPHON)
Maryanto, Maisyaroh, Budi Santoso (2018)	Metode Internet Protocol Security dengan Virtual Private Network untuk Komunikasi Data	IPSec	*Packet capture*	Mikrotik	*Static routing*	Data terenkripsi
Muhammad Sahal Nurhidayah, Dadiek Pranindito, Reni Dyah Wahyuningrum (2022)	Analisis dan Simulasi *Routing Border Gateway Protocol* antar *Autonomous System* Menggunakan *Free Rang Routing*	*Failover*	*Throughput, delay, jitter, packet loss*	FRRouting	BGP	Sangat baik – sedang (standar TIPHON)
Kingsley A. Ogudo (2019)	*Analyzing Generic Routing Encapsulation and IP Security* T*unneling Protocols for Secured Communication over Public Networks*	GRE *Tunnel*, IPSec	*Packet capture*	Cisco	EIGRP	Data terenkripsi

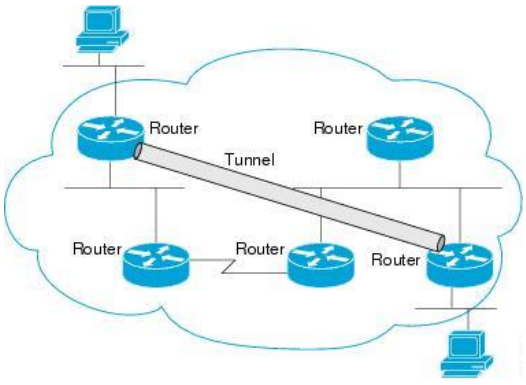
Peneliti	Judul	Komponen Penelitian				
		Metode	Parameter pengujian	Jenis *Router*	Protokol *Routing*	Hasil
Ahmad Firdausi, Hamam Wira Wardani (2020)	Simulasi dan Analisa QoS dalam Jaringan VPN *Site to Site* Berbasis IPSec dengan *Routing Dynamic*	IPSec	*Throughput, delay,packet loss*	Cisco	OSPF, RIPv2, EIGRP	Sesuai standar ITU-T
Ayu Purnama Sari, Sulistiyono, Naga Kemala (2020)	Perancangan Jaringan *Virtual Private Network* Berbasis *IP Security* Menggunakan *Router* Mikrotik	IPSec	*Packet capture*	Mikrotik	*Static routing*	Data terenkripsi
Titan Haryawan	Analisis Performansi GRE Tunnel IPSec dengan Metode Failover pada Open Source Router VyOS	GRE *Tunnel*, IPSec	*Throughput, delay, jitter* dan waktu konvergesi	VyOS	OSPF	*Throughput* dan *jiter* sangat baik, *delay* jelek (standar TIPHON). Waktu Konvergensi rata-rata 32,26 detik

Dari tabel 2.1 di atas merupakan perbandingan dengan penelitian yang dilakukan sebelumnya. Kesamaan pada penelitian ini yaitu menggunakan metode Teknologi GRE *Tunnel* dengan dienkripsinya data menggunakan IPSec seperti pada penelitian [8], menggunakan protokol *routing* OSPF seperti pada penelitian [9]. Sedangkan pembeda pada penelitian ini yaitu peneliti menambahkan waktu konvergensi pada parameter pengujian selain parameter QoS yang meliputi *throughput, delay dan jitter*, selanjutnya pembeda pada penelitian ini yaitu peneliti menggunakan *open source router* VyOS, berbeda dengan penelitian [4] dan [6] yaitu menggunakan *open source router* FFR atau *Free Range Routing.*

2.2 DASAR TEORI

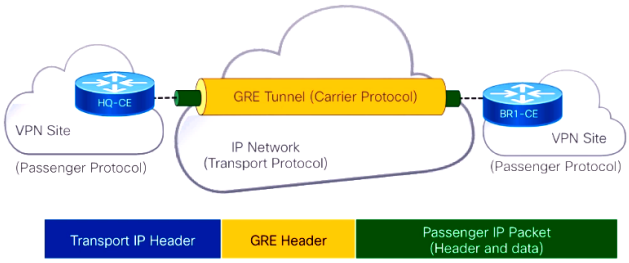
2.2.1 Teknologi *Tunneling*

Teknologi *tunneling* adalah teknologi yang tugasnya mengelola dan menyediakan koneksi *point-to-point* dari sumber ke tujuan. Disebut *tunnel* (terowongan) karena pada dasarnya koneksi *point-to-point* dibentuk dengan melalui jaringan publik tetapi koneksi tersebut tidak memperdulikan data milik orang lain yang melewati jaringan publik tersebut tetapi koneksinya hanya untuk transportasi data dari pembuatnya. Ini seperti menggunakan jalur bus yang pada dasarnya menggunakan jalan raya akan tetapi memuat jalur terpisah khusus untuk bus. *Tunneling* merupakan *overlay network* yaitu jalur yang dibuat secara virtual di atas jalur fisik pada jaringan komputer [11]. Di bawah ini merupakan gambar ilustrasi tunneling [12].



Gambar 2. 1 Ilustrasi *Tunneling*

Generic Routing Encapsulation (GRE) adalah protokol *tunneling* yang berada pada *layer* 3 model OSI yaitu *network layer*. GRE *Tunnel* awalnya dikembangkan oleh Cisco dengan menyediakan pendekatan umum sederhana untuk membawa paket dari satu protokol ke protokol lain menggunakan enkapsulasi. GRE dapat digunakan sebagai protokol transportasi untuk protokol lainnya yang berbeda. GRE merangkum muatan sebagai paket yang perlu dikirimkan ke jaringan tujuan dalam paket IP eksternal. Setelah mencapai ujung terowongan, GRE yang dienkapsulasi ini dihapus dan muatannya diteruskan ke tujuan. GRE *Tunnel* ini sama seperti IPIP dan EoIP yang awal mulanya dikembangkan sebagai *stateless tunnel* dan dapat meneruskan hanya Ipv4 dan IPv6 paket (*ethernet* tipe 800 dan 86dd) [4]. Di bawah ini merupakan gambar ilustrasi GRE Tunnel [13].



Gambar 2. 2 Ilustrasi GRE *Tunnel*

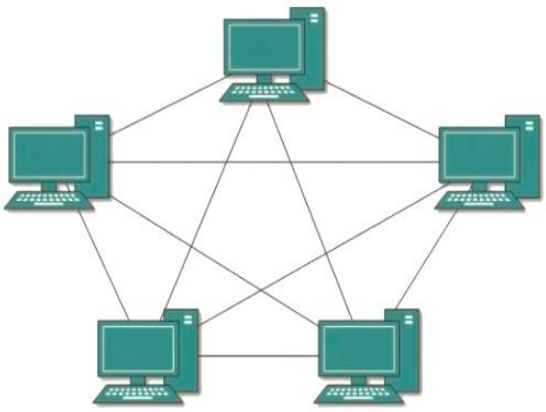
2.2.2 Teknologi VPN

Virtual Private Network atau disingkat VPN merupakan teknologi komunikasi yang memungkinkan terhubung ke *public network* lalu menggunakannya untuk bergabung dengan *private network*. Koneksi VPN bersifat *virtual* dan *private*, sehingga hanya pengguna tertentu yang dapat mengakses koneksi VPN tersebut. VPN sangat dibutuhkan baik oleh swasta (perusahaan) dan organisasi pemerintah. Hal ini dikarenakan dalam VPN terdapat proses enkripsi menjamin data yang melalui *tunnel* tidak dapat dibaca oleh orang lain. Proses enkripsi dilakukan dengan mengubah informasi yang ada dalam *tunnel* menjadi *ciphertext* yaitu teks yang diacak sehingga tidak ada artinya sama sekali. Untuk dapat membuatnya kembali memiliki arti, maka akan dilakukannya proses dekripsi. Pengirim dan penerima telah menyepakati sebuah algoritma yang akan digunakan untuk melakukan proses enkripsi dan dekripsinya. Sehingga data yang dikirim aman sampai tempat tujuan, hal ini dikarenakan orang lain di luar *tunnel* tidak memiliki algoritma untuk melakukan dekripsi. [11,14]. Namun Menurut penelitian [30] implementasi VPN dapat mempengaruhi nilai *delay* dikarenakan adanya proses enkripsi dan dekripsi.

Internet Protocol Security atau disingkat IPSec merupakan protokol yang digunakan untuk melindungi transmisi *datagram* melalui jaringan berbasis TCP/IP. IPSec dikembangkan oleh *Internet Engineering Task Force* (IETF). IPSec bekerja dalam lapisan *network* dari model lapisan *Open Systems Interconnection* (OSI). IPSec berfungsi untuk melindungi dan memvalidasi komunikasi IP antar *host*, dan beroperasi pada lalu lintas IPv6 dan IPv4. IPSec sebenarnya adalah fitur IPv6, akan tetapi beberapa pengembang telah menerapkannya ke IPv4 [5].

2.2.3 Topologi Mesh

Topologi Mesh adalah jaringan komputer di mana bentuk-bentuk koneksi antara perangkat komputasi di jaringan terkait langsung satu sama lain. Setiap perangkat komputasi dalam jaringan saling berhubungan atau disebut *dedicated link*, sehingga dapat saling berkomunikasi secara langsung. Topologi Mesh biasanya tidak terlalu besar dan dirancang untuk jaringan yang membutuhkan komunikasi berkecepatan tinggi antar perangkat. Topologi jaringan Mesh sangat sulit diatur dan menggunakan banyak kabel, sehingga relatif jarang digunakan. Jika salah satu komputer di topologi Mesh *crash*, komputer lain tidak akan terpengaruh. Proses pembuatan jaringan topologi Mesh menggunakan Persamaan N x (N-1) : 2. N adalah jumlah komputer. Oleh karena itu, jika ada 5 komputer dalam jaringan topologi ini, jumlah kabel yang digunakan adalah 5 x (5-1) : 2 = 10 sambungan. Juga, semua perangkat komputasi memerlukan *port* I/O ekspresi N1 (5-1 = 4). Di bawah ini merupakan gambar ilustrasi tolologi Mesh [15].



Gambar 2. 3 Ilustrasi Topologi Mesh

2.2.4 VyOS

VyOS adalah perangkat virtual *open source* yang dapat digunakan untuk *routing*, *firewall*, dan *Virtual Private Network* (VPN) yang mencakup IPSec, DMVPN, OpenVPN dan lain-lain. VyOS terusun dari karnel Linux bebasis Debian yang memiliki antarmuka CLI serta konfigurasi bersifat *stateful*. Di dalam VyOS terdapat *routing* protokol seperti *Open shortest path first* atau OSPF, *Border Gateway Protocol atau* BGP, dan *Routing Information Protocol* atau RIP. Sistem pada perangkat ini dapat berjalan di platform virtual maupun fisikal [16]. Di bawah ini merupakan gambar dari logo VyOS [17].

Gambar 2. 4 Logo VyOS

2.2.5 *Transmission Control Protocol* (TCP)

Transmission Control Protocol atau disingkat TCP merupakan protokol yang memungkinkan komputer yang ada pada jaringan berkomunikasi atau melakukan pertukaran data. TCP berada pada lapisan *transport* pada model *OSI layer*. 75% protokol yang digunakan untuk layanan internet saat ini yaitu TCP karena sifatnya yang *connection oriented* yaitu sebelum data dapat ditransmisikan harus melakukan negosiasi untuk menciptakan sesi koneksi terlebih dahulu. Selain itu TCP memiliki sifat *reliable* yaitu data ditransfer ke tujuannya dalam keadaan sesuai urutannya seperti ketika data tersebut diterima, namun kekurangan pada protokol ini ketika trafik pada jaringan sedang padat akan berdampak terjadinya tabrakan atau *congestion* yang tinggi dan menyebakbakn *time out* sehingga akan melakukan pengiriman ulang karena sifatnya yang *connection oriented* [18].

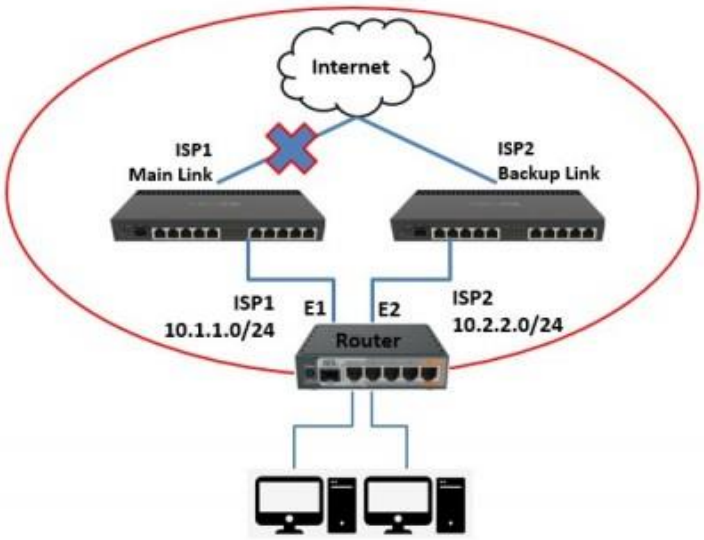
2.2.6 *Open Shortest Path First* (OSPF)

Open Shortest Path First atau OSPF dikembangkan oleh *Internet Engineering Task Force* (IETF) merupakan protokol *link state routing* dan berfungsi untuk menghubungkan antara *router* yang berada pada sistem otonom (AS) sehingga protokol routing ini tergolong IGP. OSPF memiliki kemampuan konvergensi yang tepat waktu dengan tingkat kehilangan paket minimal [19]. Untuk mengelola jaringan skala besar, OSPF menggunakan konsep area dalam implementasinya yaitu mengkelompokan *router* yang dikenal, area 0 atau *backbone* merupakan area penting yang wajib ada pada *routing* OSPF. OSPF menggunakan algoritma *dijkstra* untuk mengimplementasikan protokol *routing*. Pada awal OSPF *router* dihidupkan, maka *router* akan membuat *adjacency* atau membuat pertemanan antar *router* lainnya dengan mengirimkan *hello packet*

secara *multicast* (224.0.0.5), selanjutnya *hello packet* akan dibalas dengan paket LSDB (*Link-State Database*) yang berisi alamat *network router* tetangga. Ketika ada *router* yang tidak berfungsi, maka *router* tetangga akan mengirimkan paket LSR (*Link-State Request*) untuk mengetahui jalur alternatif sebagai pengganti *router* yang tidak berfungsi tersebut, paket LSR akan dibalas dengan paket LSU (*Link-State update*) yang berisi *update* alamat *network router*. Setiap paket yang diterima akan dibalas juga dengan paket LSAck (*Link-State Acknowledgement*) yang berarti paket telah diterima oleh *router* penerima [20].

2.2.7 *Failover*

Failover merupakan metode untuk mengalihkan jalur komunikasi secara otomatis dengan menciptakan *main link* dan *backup link*, berbeda dengan *switchover* yaitu berpindahnya secara manual atau adanya campur tangan manusia. Pada saat kondisi perangkat pada jaringan dalam keaadan normal maka hanya *main link* atau jalur utama yang digunakan untuk komunikasi, sedangkan jalur yang lainnya merupakan *backup link* atau jalur cadangan cadangan ketika jalur utama terputus yang pemindahaan koneksinya berjalan secara otomatis. *Failover* bertujuan agar komunikasi data pada jaringan dapat terus berjalan tanpa terputus sehingga tidak menyebabkan meningkatnya *downtime, downtime* adalah periode waktu ketika sebuah sistem tidak berfungsi sebagaimana mestinya. [21]. Di bawah ini merupakan gambar ilustrasi *failover* [22].



Gambar 2. 5 Ilustrasi *Failover*

2.2.8 *Quality of Services* (QoS)

QoS merupakan cara untuk mengetahui *bandwidth, delay, jitter* dan *packet loss* pada lalu lintas jaringan komputer. *Quality of Service* dibuat untuk membantu pengguna agar menjadi lebih produktif dengan memastikan bahwa pengguna mendapatkan kinerja yang andal. QoS mengacu pada kemampuan jaringan untuk menyediakan layanan yang lebih baik melalui lalu lintas jaringan tertentu melalui teknologi yang berbeda. Kualitas layanan merupakan tantangan utama dalam jaringan Internet pada umumnya. Tujuan dari mekanisme QoS yaitu untuk mempengaruhi setidaknya satu dari empat parameter dasar QoS yang telah diidentifikasi yaitu *througput, delay, jitter* dan *packet loss* [23].

a. *Throughput*

Throughput adalah jumlah keseluruhan kedatangan paket yang berhasil diamati di tujuan selama interval waktu tertentu dibagi dengan periode interval waktu tersebut. *Throughput* adalah kapasitas aktual jaringan untuk membawa data. *Throughput* selalu terkait dengan *bandwidth*, karena *throughput* sering disebut dalam praktiknya sebagai *bandwidth* [23]. Di bawah ini merupakan parameter standarisasi throughput [24].

Tabel 2. 2 Klasifikasi Standarisasi *Throughput*

Kategori	Besar *Throughput*	*Indeks*
Sangat bagus	>2,1 Mbps	4
Bagus	1200 Kbps - 2,1 Mbps	3
Sedang	700 - 1200 Kbps	2
Jelek	338 - 700 Kbps	1

b. *Delay*

Delay adalah penundaan suatu paket yang disebabkan oleh proses penerusan paket dari satu titik ke titik lain. *Delay* bisa dipengaruhi oleh beberapa faktor seperti jarak, media fisik, tabrakan (*congestion*) atau juga waktu proses yang lama [23]. Di bawah ini merupakan parameter standarisasi delay [24].

Tabel 2. 3 Klasifikasi Standarisasi *Delay*

Kategori	Besar *delay*	*Indeks*
Sangat bagus	<150 ms	4
Bagus	150-300 ms	3
Sedang	300-450 ms	2
Jelek	>450 ms	1

c. *Jitter*

Jitter adalah variasi *delay* antar paket yang terjadi dalam suatu jaringan IP. Tingkat *jitter* sangat dipengaruhi oleh beban trafik dan besarnya tabrakan (*congestion*) antar paket dalam suatu jaringan IP. Semakin berat beban lalu lintas di jaringan, semakin besar kemungkinan antrian dan semakin tinggi nilai *jitter* yang berdampak pada semakin rendah nilai QoS. Untuk mendapatkan nilai kualitas layanan jaringan yang baik perlu meminimalkan nilai *jitter* [23]. Di bawah ini merupakan parameter standarisasi jitter [24].

Tabel 2. 4 Klasifikasi Standarisasi *Jitter*

Kategori	Besar *jitter*	*Indeks*
Sangat bagus	0 ms	4
Bagus	0 ms-75 ms	3
Sedang	75 ms-125 ms	2
Jelek	125 ms-225 ms	1

2.2.9 Konvergensi

Konvergensi merupakan proses pada *router* yang bertujuan mengumpulkan informasi mengenai kondisi jaringan, mencari rute terbaik sesuai algoritma pada *routing protocol* yang digunakan, dan melakukan pembaharuan pada *routing* tabel. Konvergensi dapat terjadi ketika jumlah *router* bertambah atau dapat terjadi karena kegagalan link, sehingga terjadi perubahan pada setiap *router* untuk menghitung *metric* dan melakukan *update routing* tabel yang baru berdasarkan informasi terbaru. Konvergensi dibagi dua, yaitu konvergensi *failover* dan konvergensi *recovery*. Konvergensi *failover* adalah lama *dynamic routing* mendapatkan jalur *routing* ketika jalur utama yang digunakan tidak dapat dilewati, untuk menguji konvergensi *failover* dilakukan dengan cara mematikan jalur yang

sebelumnya berjalan dengan normal, kemudian mengukur waktu yang diperlukan untuk jaringan mencapai kondisi konvergensi. Sedangkan konvergensi *recovery* adalah lama dynamic *routing* mendapatkan jalur utama kembali jika jalur utama dihidupkan pada kondisi normal, untuk menguji konvergensi *recovery* dilakukan dengan cara jalur yang tadinya mati akan dihidupkan kembali kemudian mengukur waktu yang diperlukan untuk jaringan mencapai kondisi konvergensi [25]. Semakin banyaknya *router down* dan kegagalan *link* atau disebut *multiple link failure* pada jaringan maka akan memperbesar pengaruh terhadap kondisi jaringan sehingga hal ini berdampak pada bertambahnya waktu untuk proses konverensi [26].

Pada *routing* OSPF terdapat standar parameter untuk lamanya waktu konvergensi apabila terjadinya kondisi berubahnya topologi atau terdapat *router* beserta *link* yang mengalami *down* sehingga terjadinya *failover* yaitu di mana berpindahnya rute utama yang mengalami *down* melalu rute *backup* sehingga komunikasi atau pengiriman data dapat terus berlanjut. Di bawah ini tabel 2.5 merupakan standar waktu konvergensi pada OSPF [27].

Tabel 2. 5 Standar Waktu Konvergensi OSPF

OSPF	Parameter	Standar Waktu Konvergensi (ms)
SPF	*Initial*	5000
	Min-delay	10000
	Max-delay	10000
LSA	*Initial*	0
	Min-delay	5000
	Max-delay	5000
LSA *arrival*		1000

Pada tabel 2.5 di atas terdapat *Shortest Path First* (SPF) dan *Link State Advertisements* (LSA) yaitu suatu algoritma pada routing OSPF yang bekerja saat terjadinya *router* atau *link down*, paramater *initial* yaitu berapa lama waktu tunggu untuk memulai perhitungan setelah terjadinya perubahan pada topologi, parameter *min-delay* yaitu berapa lama waktu tunggu minimum antara perhitungan pertama dan kedua, dan *max-delay* yaitu waktu maksimal untuk melakukan perhitungan.

Pada OSPF terdapat algoritma SPF dan LSA yang bekerja bersamaan pada saat kondisi *failover* sehingga jaringan melakukan proses konvergensi, di mana pada proses konvergensi tersebut ada beberapa tahapan SPF yaitu proses *initial* dengan waktu selama 5000 *millisecond* atau 5 detik, proses *min-delay* dengan waktu selama 10000 *millisecond* atau 10 detik, dan proses *max-delay* dengan waktu selama 10000 *millisecond* atau 10 detik. Selanjutnya tahapan LSA yaitu proses *initial* dengan waktu selama 0 *millisecond* atau 0 detik, proses *min-delay* dengan waktu selama 5000 *millisecond* atau 5 detik, dan proses *max-delay* dengan waktu selama 5000 *millisecond* atau 5 detik Sehingga untuk standar waktu konvergensi pada *routing* OSPF setelah kedua tahapan konvergensi selesai yaitu SPF dan LSA tidak lebih dari 10 detik.

2.2.10 *Graphic Network Simulator 3* (GNS3)

Graphic Network Simlator 3 atau GNS3 adalah perangkat lunak simulasi jaringan komputer berbasis antarmuka grafis yang mudah dipahami seperti Cisco Packet Tracer. Namun pada GNS3 ini memungkinkan emulasi jaringan yang kompleks secara nyata karena menggunakan sistem operasi asli perangkat jaringan seperti Cisco dan Juniper. Kekurangan GNS3 tidak menyediakan fiture perangkat seperti *router* dan perangkat pendukung jaringan [28].

2.2.11 *Distributed Internet Traffic Generator* (D-ITG)

D-ITG merupakan platform yang dapat menghasilkan lalu lintas IPv4 dan IPv6 dengan meniru beban kerja pada internet secara akurat dapat mengukur parameter QoS seperti *throughput, delay, jitter, packet,* dll. DITG dapat meniru lalu lintas untuk berbagai aplikasi umum seperti Telnet, VoIP G.711, G.723, G.729, deteksi aktivitas suara, DNS RTP terkompresi, dan permainan jaringan. Pada lapisan *transport*, DITG saat ini mendukung UDP atau *User Datagram Protocol*, TCP atau *Transmission Control Protocol*, DCCP atau *Datagram Congestion Control Protocol*, SCTP atau *Stream Control Transmission Protocol*, dan ICMP atau *Internet Control Message Protocol* [29].

2.2.12 Wireshark

Wireshark adalah perangkat lunak capture paket bersifat *open source* yang berguna untuk menganalisis dan mengumpulkan lalu lintas data di Internet. Wireshark sering digunakan sebagai alat pemecahan masalah pada jaringan.

Wireshark mendukung berbagai format file *capture/trace* paket, termasuk format *".cap*" dan format *".erf*". Selain itu, alat deskripsi yang dibangun memiliki kemampuan untuk menampilkan paket terenkripsi dari beberapa protokol yang umum digunakan pada jaringan Internet saat ini, yaitu WEP dan WPA/WPA2. Wireshark sangat berguna untuk analisis jaringan yang cara kerjanya yaitu menangkap paket data dari protokol yang berbeda dari berbagai jenis jaringan yang biasa ditemukan di lalu lintas Internet. Paket-paket tersebut ditangkap kemudian ditampilkan dalam jendela *capture* secara *real time* [30].