BAB 2
DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian Mardianto pada tahun 2019 menganalisis QoS pada jaringan VPN dan MPLS VPN. Pada penelitian ini, hasil yang didapatkan bahwa pada perbandingan delay jaringan VPN dan MPLS VPN memiliki nilai delay sangat bagus. Perbandingan throughput MPLS VPN memiliki throughput yang stabil dibandingkan jaringan VPN. Pada perbandingan packet loss jaringan VPN dan MPLS VPN tidak memiliki packet loss [6].

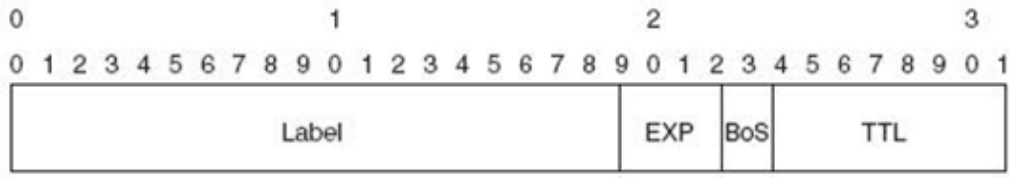
Penelitian M. Ikhsan Abdillah, Sunda Ariana, Syahril Rizal, pada tahun 2016 menganalisis performansi IPv6 *Over* MPLS. Dilihat dari hasil bahwa pada parameter *throughput* hasil simulasi yang memiliki performa tertinggi ada pada topologi EIGRP IPv6 *over* MPLS. Sedangkan untuk parameter *delay* hasil simulasi yang memiliki nilai performa *delay* terendah karena parameter *delay* nilai terendah adalah nilai yang terbaik yaitu pada topologi *Static* IPv6 over MPLS. Begitu pula dengan parameter terakhir yang diuji yaitu *packet loss*, nilai terendah adalah nilai yang terbaik pada parameter *packet loss* ini adalah topologi *Static* IPv6 *over* MPLS [7].

Penelitian AP Mungaran, R Munadi, D Perdana pada tahun 2018 menganalisis perbandingan QoS di *routing protocol* MPLS OSPF dan MPLS IS-IS menggunakan jaringan IPv6. Pada penelitian in hasil yang didapatkan bahwa *routing protocol* IS-IS yang diterapkan MPLS mendapatkan hasil QoS yang lebih baik daripada OSPF di jaringan IPv6. Dapat dilihat dari perbedaan *throughput* hingga 61 Kbps, *delay* 6 ms, *packet loss* 3% dan *jitter* sebesar 3 ms. Hal ini disebabkan *routing protocol* OSPF memiliki kompleksitas yang lebih tinggi karena pengenalan *neighbor* OSPF yang lebih rumit dibandingkan IS-IS [5].

2.2 *Multi-Protocol Label Switching* (MPLS)

Multi Protocol Label Switching (MPLS) adalah suatu metode *forwarding* (meneruskan data melalui suatu jaringan dengan menggunakan informasi dalam label yang diletakkan pada IP), sehingga memungkinkan *router* meneruskan paket dengan hanya melihat label dari paket itu tanpa melihat IP tujuannya [4].

Prinsip kerja MPLS ialah menggabungkan kecepatan *switching* pada *layer* 2 dengan kemampuan *routing* dan skalabilitas pada *layer* 3. Cara kerjanya adalah dengan menyelipkan *label* di antara *header layer* 2 dan *layer* 3 pada paket yang diteruskan. Label dihasilkan oleh Label-*Switching Router* dimana bertindak sebagai penghubung jaringan MPLS dengan jaringan luar. Label berisi informasi tujuan *node* selanjutnya ke mana paket harus dikirim. Kemudian paket diteruskan ke *node* berikutnya, di *node* ini label paket akan dilepas dan diberi label yang baru yang berisi tujuan berikutnya. Paket-paket diteruskan dalam path yang disebut LSP (*Label Switching Path*) [8].



Gambar 2. 1 Format *MPLS* Header [4]

Gambar 2.1 merupakan gambar format MPLS *header* dengan rincian sebagai berikut :

- *Label Value* (LABEL)

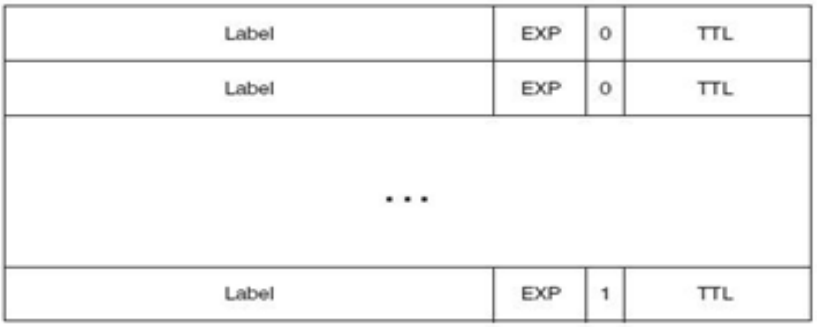
  Merupakan *field* yang terdiri dari 20 bit yang merupakan nilai dari label tersebut.

- *Experimental Use* (EXP)

  Secara teknis *field* ini digunakan untuk keperluan eksperimen. *Field* ini dapat digunakan sebagai indikator QoS atau juga merupakan hasil salinan dari bit-bit IP *precedence* pada paket IP.

- *Bottom of Stack* (BOS)

  Pada sebuah paket memungkinkan menggunakan lebih dari satu label. *Field* ini digunakan untuk mengetahui *label stack* yang paling bawah. Label yang paling bawah dalam *stack* memiliki nilai 1 bit sedangkan label yang lain diberi nilai 0.



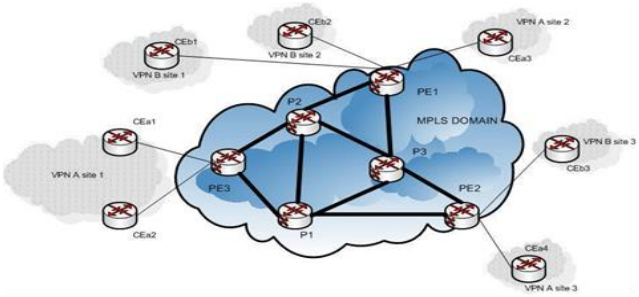
Gambar 2. 2 *Label Stacking* [4]**.**

- *Time to Live* (TTL)

  Field ini merupakan hasil salinan dari IP TTL *header*. Nilai bit TTL akan berkurang 1 pada setiap paket melalui hop.

2.3 MPLS VPN

Salah satu *feature* yang dapat digunakan pada jaringan MPLS adalah *Virtual Private Network* (VPN) yang sering disebut sebagai MPLS-VPN. VPN dapat dengan adanya fasilitas *tunnel* yang dapat melintasi jaringan MPLS. Fasilitas *tunnel* bisa membangun sebuah jalur *site* ke *site* lain secara *virtual* dan mempunyai tingkat keamanan yang cukup baik karena *site to site connection* tersebut hanya dapat diakses oleh *user* yang berada pada salah satu *site* (*private connection*) [5].



Gambar 2. 3 Contoh Topologi MPLS-VPN [9]

Pada gambar diatas dapat diilustrasikan koneksi *site to site* dengan meggunakan teknologi VPN pada suatu jaringan MPLS. Koneksi antar dua *site* dapat terbentuk melalui satu atau lebih *Provider Edge Router* (PE), koneksi dapat dilakukan melalui *Customer Edge Router* (CE) ke PE *router* bisa juga langsung terkoneksi ke PE *router* tanpa menggunakan *router* pada sisi *client*.

Secara umum digambarkan bahwa *provider* menyediakan suatu *core network* yang berbasis MPLS, *core network* ini nantinya akan menghubungkan beberapa *site* (sesuai dengan layanan koneksi yang diinginkan) dari satu *site* ke *site* lain secara *private*. Misal pada layanan VPN *customer* A atau disebut sebagai VPN A, akan mengkoneksikan 3 *site* yang terhubung di *core network* MPLS, tiga *site* ini akan terkoneksi melalui CE *router* milik *customer* (jika ada) ke PE *router* milik *provider*. Dengan itu koneksi VPN A antara 3 *site* ini, maka masing-masing *site* dapat terkoneksi secara *private* tanpa tercampur oleh trafik dari VPN lainnya [9].

2.4 *Protocol Routing*

Protocol routing adalah aturan yang mempertukarkan informasi *routing* yang nantinya akan membentuk tabel *routing*, sedangkan *routing* adalah aksi pengiriman-pengiriman paket data berdasarkan tabel *routing*. Semua *routing protocol* bertujuan mencari rute tersingkat untuk mencapai tujuan [5].

Routing adalah proses menentukan rute dari *host* asal ke *host* tujuan. *Routing* merupakan proses memindahkan data dari satu *network* ke *network* lain dengan cara mem-*forward* paket data via *gateway*. *Routing* menentukan kemana datagram akan dikirim agar mencapai tujuan yang diinginkan. Adapun informasi yang dibutuhkan *router* dalam melakukan *routing* yaitu: 1)alamat tujuan, 2)mengenal sumber informasi, 3)menemukan rute, 4)pemilihan rute dan 5)menjaga informasi *routing* [10].

2.5 *Open Shortest Path First* (OSPF)

Open Shortest Path First (OSPF) adalah sebuah *routing* protokol standar untuk jaringan Internet Protokol (IP). OSPF menggunakan algoritma *Link State Routing* dan merupakan salah satu jenis Protokol *Interior Gateway Protocol* (IGP) bekerja pada sistem *autonomous* (AS).

OSPF bekerja berdasarkan algoritma *Shortest Path First* yang dikembangkan berdasarkan algoritma Dijkstra. OSPF mendistribusikan informasi *routing*-nya di dalam *router-router* yang tergabung ke dalam suatu AS [2].

2.6 *Intermediate System-Intermediate System* (IS-IS)

IS-IS merupakan salah satu *routing* protokol yang menggunakan metode *link state* sebagai metode pengumpul rutenya. IS-IS juga akan melakukan pengumpulan informasi dan status dari semua *link* yang ada dalam jaringannya. Kemudian informasi tersebut dibentuk menjadi sebuah informasi rute yang dapat digunakan untuk meneruskan data. IS-IS adalah sebuah *Interior Gateway Protocol* (IGP) yang berarti protokol *routing* yang digunakan hanya dalam sebuah *domain* atau jaringan administrasi yang sama. Setiap *router* kemudian membangun sebuah gambaran tentang topologi jaringan. IS-IS menggunakan Algoritma *djikstra* untuk menghitung jalan terbaik melalui jaringan.

Routing protokol IS-IS memiliki karakteristik untuk membagi *domain* pada jaringannya yang disebut dengan *level*. *Level* disini terbagi menjadi 3, yaitu : *Level* 1, *level* 2 dan *level* 1/2. *Routing* protokol IS-IS menghitungulang rute *path* terbaik ketika terjadi perubahan pada topologi jaringannya dan juga men-*support multiple path* yang mempunyai nilai *cost* yang sama [3].

2.7 *Quality of Service* (QoS)

Quality of Service (QoS) adalah kemampuan suatu jaringan untuk menyediakan layanan yang baik dengan menyediakan bandwidth, mengatasi *jitter* dan *delay*. Untuk menentukan kualitas QoS dibutuhkan beberapa parameter pendukung. Parameter QoS adalah :

a. Packet Loss

Packet loss merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang, dapat terjadi karena *collision* dan *congestion* pada jaringan dan hal ini berpengaruh pada semua aplikasi karena retransmisi akan mengurangi efisiensi jaringan secara keseluruhan meskipun jumlah *bandwidth* cukup tersedia untuk aplikasi tersebut [11]. Secara matematis diekspresikan dengan persamaan yang ditunjukkan pada persamaan 2.1

$$Packet\ Loss = \frac{(Paket\ data\ dikirim - paket\ data\ diterima)}{paket\ data\ yang\ dikirim} x\ 100\ \% \quad (2.1)$$

Klasifikasi standarisasi *packet loss* ditunjukkan pada Tabel 2.1 [11]

Tabel 2.1 Klasifikasi standarisasi *packet loss*

No	Kategori	Besar (%)
1	Sangat Baik	0-2
2	Baik	3-14
3	Cukup Baik	15-25
4	Tidak Direkomendasikan	>25

b. *Delay*

Delay adalah waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. *Delay* dapat dipengaruhi oleh jarak, media fisik, kongesti atau juga waktu proses yang lama [11]. Persamaan *delay* rata-rata dapat ditunjukkan pada persamaan 2.2

$$Delay = \frac{\text{waktu penerimaan paket} - \text{waktu pengiriman paket}}{\Sigma\ \text{paket yang diterima}} \quad (2.2)$$

Semakin kecil *delay* maka semakin bagus performansi sebuah jaringan. *Delay* tergantung pada banyak faktor, termasuk *bandwidth*, antrian di tiap *router*, proses di tiap *router*, dan jarak media fisik yang menghubungkan tiap *router*.

Klasifikasi standarisasi *delay* berdasarkan [11] ditunjukkan pada Tabel 2.2

Tabel 2.2 Klasifikasi standarisasi *delay*

No	Kategori	Besar (ms)
1	Sangat Baik	< 150
2	Baik	150 - 300
3	Cukup Baik	300- 450
4	Tidak Direkomendasikan	> 450

c. *Jitter*

Jitter atau variasi *delay*, berhubungan erat dengan *latency*, yang menunjukkan banyaknya variasi *delay* pada transmisi data di jaringan. *Delay* antrian pada *router* dan *switch* menyebabkan *jitter*. Hal ini diakibatkan oleh variasi-variasi panjang antrian, waktu pengolahan data, dan waktu penghimpunan ulang paket-paket di akhir perjalanan *jitter* [11]. Rumus menghitung *jitter* ditunjukkan pada persamaan 2.3

$$Jitter = \frac{(delay2-delay1)+(delay3-delay2)+\cdots+(delay\ n-delay\ (n-1))}{Total\ Paket\ yang\ diterima} \quad (2.3)$$

Klasifikasi standarisasi *jitter* ditunjukkan pada Tabel 2.3 [11]

Tabel 2.3 Klasifikasi standarisasi *jitter*

No	Kategori	Besar (ms)
1	Sangat Baik	0
2	Baik	1 - 75
3	Cukup Baik	76 - 125
4	Tidak Direkomendasikan	>126

d. *Throughput*

Merupakan kecepatan (*rate)* transfer data efektif, yang diukur dalam *byte per second*. *Throughput* juga merupakan jumlah total kedataangan paket yang sukses yang diamati pada tujuan selama *interval* waktu tertentu dibagi oleh durasi *interval* waktu tersebut. *Throughput* maksimal dari suatu titik atau jaringan komunikasi menunjukkan kapasitasnya [11].

Secara matematis *throughput* dapat dituliskan sebagai berikut:

$$Throughput = \frac{\Sigma \text{ Ukuran Paket Diterima}}{\Sigma \text{ Durasi Pengukuran}} \quad (2.4)$$

2.8 *Internet Protocol versi* 6

Internet Protocol versi 6 (IPv6) adalah protokol internet versi baru yang didesain sebagai pengganti dari *Internet Protocol* versi 4 (IPv4). IPv6 dikembangkan oleh IETF untuk dapat memenuhi kebutuhan IP yang diperlukan, selain itu IPv6 juga dikembangkan untuk mengatasi atau menyempurnakan kekurangan-kekurangan dari teknologi pendahulunya, yaitu IPv4. Kelebihan utama dari IPv6 adalah pengalamatannya yang luas, yaitu 128bit. Dengan demikian ada 2128 atau sekitar 3, 4 x 1038 alamat IPv6 yang berlimpah tersedia, sehingga dapat memenuhi kebutuhan IP saat ini maupun di masa mendatang. IPv6 ini dapat mengatasi masalah pada NAT (Network Address Translation) yang mengurangi atau menghalangi penggunaan dari aplikasi realtime yang membutuhkan hubungan dua arah.
Pada IPv6 mendukung hirarki pengalamatan dan jumlah pengalamatan node yang lebih banyak, sehingga konfigurasi alamat lebih sederhana. Kelebihan dari IPv6 yang lain ialah kapabilitas untuk QOS (Quality Of Service), autentifikasi, dan privasi [12]