BAB 2
DASAR TEORI

1.1 Kajian Pustaka

Dalam penelitian Salsa Rizkiana, Doan Perdana dan Ridha Negara pada tahun 2017 yang berjudul "Implementasi dan Analisis Performansi Layanan VPN pada Jaringan MPLS-TE menggunakan Protokol BGP dengan Metode QOS INTSERV". menjelaskan bahwa penggunaan teknologi *Traffic Enggineering* dengan menggunakan metode QoS *Intserv* pada jaringan MPLS VPN sudah terbukti bisa membuat perbaikan performansi layanan VoIP dan Video *call* dengan berbagai skenario penambahan *background traffic*. Hasil yang didapat pada skenario 0 Mbps itu tanpa *background traffic*, nilai *delay* juga memberikan perbaikan sebesar 7,42 ms atau 27,44% untuk layanan VoIP sedangkan pada layanan Video *call* hasil yang diperoleh lebih kecil dari pada layanan VoIP yaitu sebesar 3,2737 atau 11,14%. Dapat dilihat juga dari segi parameter *throughput* pada skenario 0 Mbps tanpa *background traffic* dengan menggunakan teknologi *Traffic Enggineering* pada jaringan MPLS VPN memberikan perbaikan sebesar 0.003 Mbit/s atau 6,02% untuk layanan VoIP sedangkan, pada layanan Video *call* sebesar 0,221 Mbit/s jika di persentasikan didapat 56,6%. Penambahan skenario *background traffic* 20-80 Mbps menghasilkan nilai *delay* dan *jitter* yang semakin besar atau berbanding lurus dengan *background traffic* sedangkan untuk nilai *throughput* menghasilkan nilai yang semakin kecil[3].

Dalam penelitian Laufi Dian Deode dan Wiwin Sulistyo pada tahun 2017 yang berjudul "Analisis QOS *Differentiated Service* pada jaringan MPLS menggunakan Algoritma *Threshold*". Menjelaskan bahwa penambahan WRED sebagai algoritma *threshold* pada jaringan MPLS, *Diffserv* mampu mempertahankan QoS pada layanan VoIP maupun Video *Streaming*. Dari hasil parameter QoS seperti *packet loss*, *delay*, *throughput* dan *jitter* yang didapat untuk layanan VoIP mampu mengurangi *packet loss* sebesar 43,1%, *delay* 0,005%, serta dapat memaksimalkan *throughput* sebesar 1,26% dan dapat juga mengurangi *jitter* sebesar 48,56%. Kemudian

untuk layanan Video *Streaming* mengurangi *packet loss* sebesar 15,93%, serta memaksimalkan *throughput* 1,6%. Hasil yang diperoleh dari layanan VoIP dan Video *streaming* paling bagus berdasarkan standar Tiphon[4].

Dalam penelitian Alifiah Pratiwi P.Wedda pada tahun 2015 yang berjudul "Implementasi dan analisis soft QoS (Diffserv) pada jaringan MPLS-TE untuk layanan Triple Play" meneliti tentang teknologi MPLS-TE dan menerapkan metode *differentiated* service pada jaringan dengan menggunakan *router* mikrotik sebagai MPLS router untuk melayani untuk layanan Triple play. Berdasarkan hasil pengujian yang telah dilakukan didapatkan hasil kesimpulan bahwa metode DiffServ dapat dikonfigurasikan pada jaringan MPLS-TE yang menggunakan *router* mikrotik RB 750 dan penggunaan teknologi DiffServ terbukti dapat membuat QoS layanan menjadi lebih baik, berdasarkan hasil pengukuran pada saat kondisi *link* dengan *background traffic* 0-80 Mbps parameter *delay* yang memberikan perbaikan sebesar 9,26657 ms atau 26% lebih baik untuk layanan VoIP, untuk layanan video *streaming* nilai *delay*nya mengalami penurunan 94,23880 ms atau 51,57%, dan untuk layanan FTP mengalami penurunan nilai *delay* sebesar 0,07728 ms atau sebesar 11,14% [5].

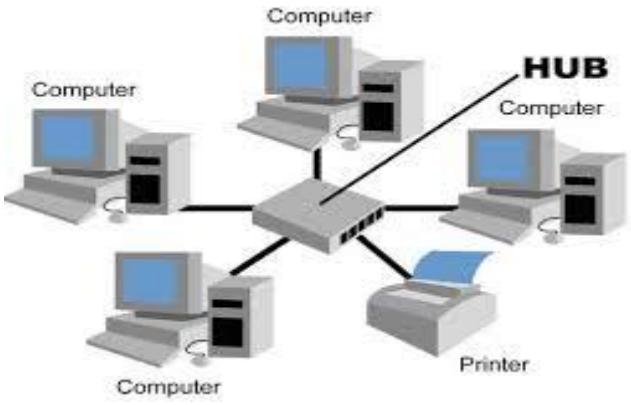
1.2 Jaringan Komputer

Jaringan komputer (*computer network*) adalah kumpulan komputer dan alat-alat lain yang saling dihubungkan bersama menggunakan media komunikasi tertentu. Informasi yang dapat melintas sepanjang media komunikasi, memungkinkan pengguna jaringan untuk dapat saling bertukar data atau menggunakan perangkat lunak maupun perangkat keras secara berbagi. Node merupakan salah satu titik sambungan yang dihubungkan pada jaringan di masing-masing komputer atau alat-alat lain. Ada tiga tipe jaringan dalam hubungannya dengan luas area yang dicakup yaitu:

- LAN (*Local Area Network*)
- MAN (*Metropolitan Area Network*)
- WAN (*Wide Area Network*)[6].

1.2.1 *Local Area Network* (LAN)

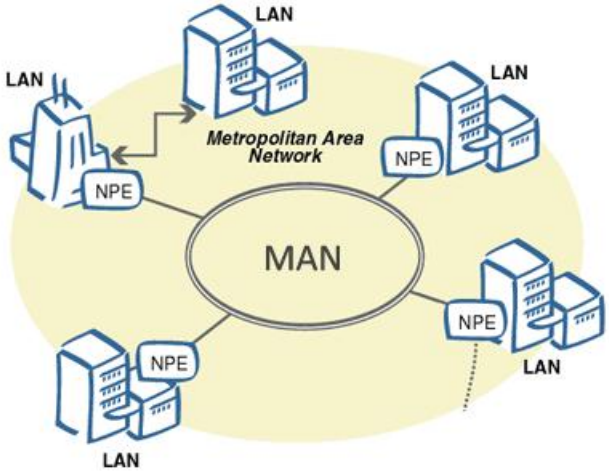
LAN (*Local Area Network*) adalah jaringan komputer yang dibangun pada area yang terbatas, seperti ruangan, rumah, kantor, gedung dan kampus. LAN mendukung kecepatan data yang cukup tinggi[7].



Gambar 2.1 Jaringan *Local Area Network* (LAN) [8]

1.2.2 *Metropolitan Area Network* (MAN)

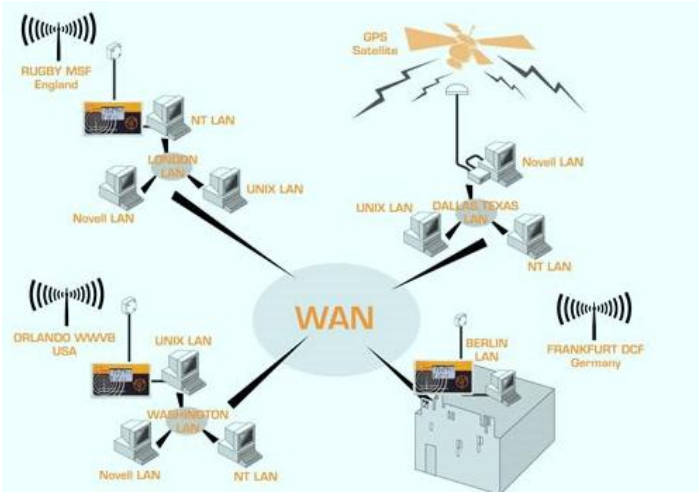
Metropolitan Area Network (MAN**)** merupakan jaringan komputer yang meliputi area sebuah kota. Teknologi yang digunakan oleh MAN areanya lebih besar dan komputer yang dapat dihubungkan pada jaringan pun jauh lebih banyak. MAN dapat memanfaatkan jaringan TV kabel jenis *coaxial* dan serat optik. Sehingga dapat mengangkut data berukuran gigabit dengan sangat cepat[7].



Gambar 2.2 Jaringan *Metropolitan Area Network* (MAN) [8]

1.2.3 *Wide Area Network* (WAN)

Wide Area Network (WAN) merupakan jaringan komputer yang meliputi area geografis sangat besar, seperti antarkota, antarnegara, antarbenua[8]. WAN dapat menghubungkan LAN atau MAN yang dipisahkan oleh jarak yang sangat jauh. Untuk menghubungkan kedua jarak yang berjauhan biasanya digunakan saluran telepon atau saluran komunikasi publik (umum)[7].



Gambar 2.3 Jaringan *Wide Area Network* (WAN) [8]

1.3 Model *Open System Interconnection* (OSI Layer)

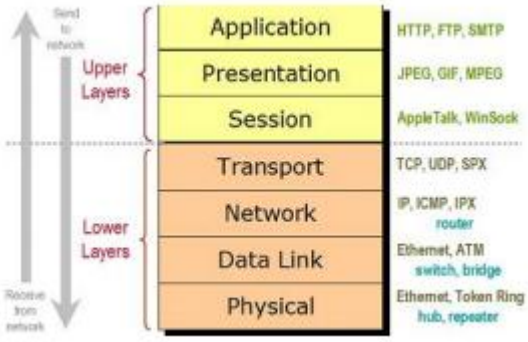
OSI layer merupakan sebuah model referensi berbentuk kerangka konseptual yang mendefinisikan standar konek didalam sebuah komputer. Adapun tujuan dibuatnya model referensi OSI layer ini agar menjadi rujukan untuk para *vendor* dan *developer* sehingga produk atau *software* yang dibuat dapat bersifat interporate, yang berarti dapat bekerja sama dengan sistem atau produk lainnya tanpa harus melakukan upaya khusus dari pengguna.

Model OSI layer juga menerapkan konsep yang dikenal dengan enkapsulasi. Enkapsulasi adalah metode membungkus data dari satu lapisan model OSI layer dalam struktur data baru sehingga, setiap lapisan dimodel OSI layer hanya akan melihat dan berurusan dengan formasi yang dibutuhkan untuk menangani serta memberikan data pada jaringan

komputer[10]. Model referensi OSI layer mempunyai prinsip-prinsip sebagai berikut:

a. Setiap lapisan memiliki fungsi yang dapat didefenisikan.
b. Batas-batas lapisan telah dirancang untuk mengurangi arus informasi dalam antarmuka.
c. Ketika tingkat tambahan abstraksi diperlukan, maka lapisan selanjutnya akan dibuat.
d. Setiap lapisan memiliki fungsi standar internasional.

Konfigurasi OSI layer berupa 7 lapisan yang disetiap lapisan akan mengenkapsulasi data sesuai protokol yang berlaku dalam lapisnya. Ilustrasi tentang OSI layer dapat dilihat seperti gambar 2.4 sebagai berikut.



Gambar 2.4 Protokol yang ada pada OSI Layer [9]

a. *Physical Layer*

Dalam lapisan ini sebenarnya tidak terdapat protokol yang spesifik, karena pada layer ini hanya mengirimkan bit data[9]. *Physical Layer* merupakan suatu lapisan atau layer yang pertama dalam model OSI. Layer ini bertanggung jawab untuk mentransmisikan bit data digital *physical layer* perangkat pengirim menuju ke *physical layer* perangkat penerima dengan media komunikasi jaringan. Data yang akan ditransmisikan dapat melalui beberapa macam media fisik seperti tegangan listrik, kabel, frekuensi radio, *infrared*, maupun sinyal cahaya biasa.

b. *Data Link Layer*

Pada layer ini bit-bit informasi diubah menjadi bentuk data yang disebut *frame*. Didalam layer ini terjadi koreksi kesalahan, *flow control*, dan menentukan bagaimana perangkat-perangkat jaringan seperti *hub*, *bridge*, *repeater* dan *switch* layer 2 beroperasi. *Data link layer* juga mengelola skema pengalamatan fisik seperti alamat MAC pada suatu jaringan[9].

c. *Network Layer*

Pada *layer network* ini terdapat pendefinisian alamat-alamat IP yang kemudian, ditetapkan jalur yang akan dapat digunakan untuk transfer data antara perangkat didalam suatu jaringan. Data yang sudah dienkapsulasi dilayer 3 disebut *datagram packet*. *Router* merupakan perangkat yang bekerja dilayer 3. *Router* juga berfungsi untuk menjalankan mekanisme *routing*. *Routing* dapat memungkinkan paket datagram dipindahkan antar komputer yang terhubung satu sama lain. Untuk mendukung proses *routing* ini, *network layer* menyimpan alamat logis seperti alamat IP untuk setiap perangkat pada jaringan. *Network* layer juga mengelola pemetaan antara alamat *logic* dan alamat fisik. Dalam jaringan IP, pemetaan ini dilakukan *Address Resolution Protocol* (ARP).

d. *Transport Layer*

Pada *transport layer* data akan dipecah ke dalam paket-paket data serta memberikan nomor urut ke paket-paket tersebut sehingga dapat disusun kembali pada sisi tujuan setelah diterima. Pada layer ini juga membuat sebuah tanda bahwa paket diterima dengan sukses dan mentransmisikan ulang terhadap paket-paket yang hilang ditengah jalan. Di layer ini terdapat 2 protokol penting yang bekerja yaitu protokol TCP dan UDP. *Transmission Control Protocol* (TCP) adalah protokol yang menyediakan layanan penuh lapisan transport untuk aplikasi sedangkan, *User Datagram Protocol* (UDP) adalah protokol yang melayani informasi yang bersifat *connectionless* dan *proces-to-proces* yang hanya menambahkan alamat *port*, *checksum error control* dan panjang informasi data pada layer diatasnya[9] .

e. *Session Layer*

Pada *session layer* ini, mendefinisikan bagaimana koneksi dapat dibuat, dipelihara atau dihancurkan. Dalam layer ini juga terdapat beberapa protokol yang bekerja. NETBIOS adalah salah satu protokol yang bekerja pada layer ini. Protokol ini memiliki fungsi sebagai peran yang memungkinkan user mengirim pesan tunggal secara serentak ke komputer lain yang terkoneksi[9] .

f. *Presentation Layer*

Layer ini mempunyai tanggung jawab untuk mendefinisikan sintaks yang digunakan *host* jaringan untuk komunikasi. Pada *presentation layer* ini juga melakukan proses enkripsi / dekripsi informasi atau data sehingga mampu digunakan pada lapisan aplikasi. Terdapat protokol yang dapat berjalan dalam layer ini seperti telnet, SMTP dan SNMP [9].

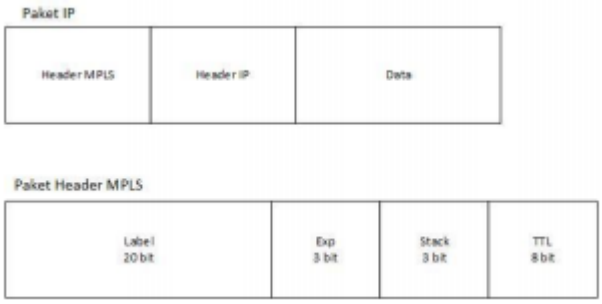
g. *Aplication Layer*

Layer ini merupakan lapisan paling atas dari model OSI dan yang ada pada komputer. *Application layer* menyediakan layanan yang dibutuhkan oleh aplikasi seperti *Hyper Text Tranfer Protocol* (HTTP). Protokol ini adalah protokol tanggung jawab untuk menyediakan sebuah interface antar protokol jaringan dengan aplikasi yang digunakan untuk mentransfer dokumen dan *web* dalam sebuah *web browser* melalui www. HTTP juga merupakan protokol yang meminta serta menjawab antar klien dan juga *server*. Selain itu juga terdapat beberapa protokol lain yang bekerja pada layer ini seperti FTP (*File Transfer Protocol*). Protokol ini merupakan standar untuk mentransfer file komputer antar mesin-mesin dalam sebuah jaringan internet. Ada juga protokol *Domain Name System* (DNS). Protokol ini adalah yang digunakan untuk memberikan suatu nama domain pada sebuah alamat IP agar lebih mudah diingat[9] .

2.4 *Multi Protocol Label Switching* (MPLS)

Seiring perkembangan teknologi MPLS merupakan penyempurna dari teknologi yang sebelumnya sudah banyak digunakan oleh para *network*

provider dalam menyediakan jasa penyewaan jaringan pribadi (*private*) dengan menggunakan infrastruktur jaringan bersama kepada customernya [2]. *Multi Protocol Label Switching* (MPLS) yaitu suatu metode *forwarding* bertujuan untuk meneruskan data melalui suatu jaringan dengan menggunakan informasi dalam label yang dilekatkan pada paket IP. MPLS merupakan penggabungan dari teknologi *switching* layer-2 dengan teknologi yang digunakan *routing* layer-3. Kemudian MPLS dapat menyederhanakan *routing* paket dan mengoptimalkan pemilihan jalur (*path*) yang melalui *core network* [10].



Gambar 2.5 Format MPLS *Header Packet*[10]

Pada gambar 2.5 di atas merupakan gambar format MPLS *header packet* dengan keterangan sebagai berikut:

a. *Label Value* (LABEL)

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

b. *Experimental Use* (EXP)

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

c. *Bottom of Stack* (STACK)

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

d. *Time to Live* (TTL)

Field ini biasanya merupakan hasil salinan dari IP TTL *header*. Nilai bit TTL akan berkurang 1 jika setiap paket melewati *hop* untuk menghindari terjadinya *packet storms*[10].

2.4.1 Komponen MPLS [10]

Adapun komponen yang terdapat pada MPLS sebagai berikut :

a. *Label Switched Path* (LSP)

   Label Switched Path merupakan jalur yang melalui satu atau serangkaian LSR dimana paket akan diteruskan oleh label *swapping* dari satu MPLS *node* menuju ke MPLS node yang lain.

b. *Label Switching Router*

   Label Switching Router termasuk MPLS *node* yang dapat meneruskan paket-paket pada layer 3.

c. MPLS *Edge Node* atau *Label Edge Router* (LER)

   Tugas dari MPLS *node* ialah menghubungkan sebuah MPLS domain dengan *node* yang berada diluar MPLS domain.

d. MPLS *Egress Node*

   MPLS *node* jenis ini juga yang akan mengatur trafik saat meninggalkan MPLS domain.

e. MPLS *Ingress Node*

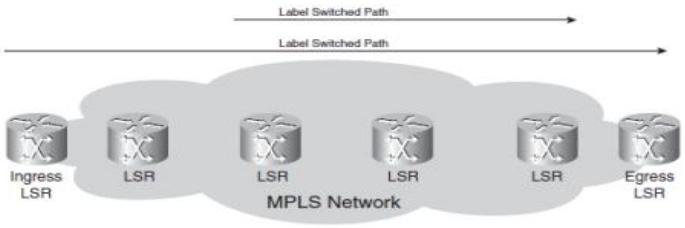
   Tujuan dari MPLS *Ingress node* ini yang berfungsi sebagai pengatur trafik saat akan memasuki MPLS domain.

f. MPLS Label

   Merupakan label yang ditempatkan sebagai MPLS *header*.

g. MPLS *Node*

   MPLS *node* ini sebagai *control protocol* yang akan meneruskan paket berdasarkan label. MPLS *node* adalah sebuah *router*. MPLS *node* bertindak sebagai *control protocol* yang bertugas meneruskan paket berdasarkan label.



Gambar 2.6 Komponen MPLS [10]

2.4.2 Arsitektur MPLS

Arsitektur pada MPLS mempunyai peran yang secara umum dapat dibedakan menjadi 3 yaitu:

1. P (*Provider*) yaitu *router backbone* yang melakukan *label switching* (LSR). Tidak melibatkan *routing* internet atau routing dari *costumer*.
2. PE (*Provider Edge Router*) yaitu *router* yang melakukan *label popping* (LER). *Router* yang dapat terhubung ke berbagai *service* seperti internet, L3VPN, L2VPN/VPLS, TE (*Traffic Engineering*).
3. CE (*Costumer Edge Router*) yaitu perangkat yang ada pada *customer* yang berkomunikasi dengan PE[11].

2.5 MPLS-TE

Label Multi-protocol Switching (MPLS) adalah bagian dari arsitektur *Internet Engineering Task* (IETF) dapat menyediakan *designation, routing, forwarding, and switching* yang efisien melalui suatu jaringan. Sedangkan *Traffic Engineering* adalah memindahkan *traffic* sehingga *traffic* dari *link* yang memiliki *congestion* dipindahkan ke *link* yang sedang tidak digunakan. *Traffic engineering* juga dapat diimplementasikan dengan cara mudah *tweaking* IP *metrics* dalam interface atau sesuatu yang serumit menjalankan sebuah ATM PVC *full-mesh* dan mengoptimalisasi jalur PVC berdasarkan permintaan *traffic* yang melewatinya[10].

2.6 *Voice over Internet Protocol* (VoIP)

Voice over IP atau sering disebut dengan VoIP tidak lain adalah sebuah protokol jaringan layer 3 yang menggunakan beragam protokol *point-to-point* layer 2 atau protokol-protokol *link layer,* seperti PPP,*Frame Relay,*

atau ATM untuk kebutuhan transpornya. VoIP memungkinkan berbagai *cisco router, access server,* dan *multiservice access concentrator* membawa dan mengirim trafik *voice* dan *fax* melintasi jaringan IP. Dalam VoIP terdapat digital *signal processors* (DSP) bertugas melakukan segmentasi (pemecahan) sinyal *voice* ke berbagai bentuk *frame* dan menyimpan dalam paket-paket *voice*. VoIP juga memberi jalan bagi beberapa *cisco router* dan *server* akses untuk mentransmisikan berbagai trafik *voice* melalui jaringan IP. Secara spesifik , VoIP dapat digunakan untuk memberikan :

- Fasilitas terminasi *telephony site-central* untuk trafik VoIP dari berbagai fasilitas atau perangkat *voice* kantor cabang (*remote*)
- *Gateway* PSTN untuk trafik telepon internet. VoIP umum digunakan sebagai *gateway* PSTN untuk berbagai aplikasi klien telepon internet berbasis H.323.

Keuntungan-keuntungan dari VoIP sebagai berikut:

- Penggunaan PBX *remote* melintasi WAN.
- *Offload* trafik *voice* dan trafik fax PSTN.
- Layanan-layanan universal untuk *voice-mail* dan *fax-mail*.
- Penyatuan *voice* dan data *trunking*.
- *Plain old telephone service* (POTS) *gateway telephony* Internet [12].

VoIP juga dapat berkomunikasi dengan sistem lain yang sedang beroperasi pada jaringan *packet-switch*. Agar dapat berkomunikasi maka dibutuhkanlah suatu standar komunikasi yang kompatibel satu sama lain. Dimana salah satu standar komunikasi pada VoIP menurut rekomendasi ITU-T yaitu H.323. Dalam standar H-323 memiliki beberapa komponen, protokol, dan prosedur yang menyediakan komunikasi multimedia melalui jaringan *packet-based.* Tujuan dari desain dan pengembangan H.323 adalah untuk memungkinkan interoperabilitas dengan tipe terminal multimedia lainnya. Pada terminal H.323 memungkinkan komunikasi *real time* dua arah berupa suara, video dan data. Standar H.323 terdiri dari 4 komponen fisik digunakan saat menghubungkan komunikasi multimedia *point-point* dan *point-to-multipoint* beberapa jenis jaringan:

- Terminal

- *Gateway*
- *Gatekeeper*
- *Multipoint Control Unit* (MCU)

Fungsi dan kemampuan dari terminal H.323 yaitu:

- *Audio Codec* adalah mengkodekan sinyal dari peralatan audio untuk transmisi dan menguraikan kode audio yang diterima. Fungsi-fungsi yang dibutuhkan antara lain mengkodekan dan menguraikan kode pada G.711 dan mengirim serta menerima format *a-law* dan *ɳ-law*. Sebagai tambahan *audio codec* ini juga dapat mengkode dan menguraikan kode pada G.726, G.728, G.729 dan G.723.1.
- *Video codec* merupakan fungsi tambahan pada terminal H.323.
- Data *Channel* dapat mendukung aplikasi-aplikasi seperti pengakses database, pengiriman *file*, dan *audiographics conferencing* merupakan kemampuan untuk memodifikasi gambar untuk beberapa pengguna secara bersama-sama dan direkomendasikan T.120
- *System Control Unit* yaitu menyediakan H.225 dan *call control* H.245, pengirim pesan, dan perintah-perintah pensinyalan.
- Media transmisi dapat membentuk format audio, video, data, *control stream*, dan *message* yang sesuai dengan antarmuka jaringan dan juga menerima dari antarmuka jaringan.
- *Network Interface* adalah suatu antarmuka yang *packet-based* untuk *Transmission Control Protocol* (TCP) dan *User Datagram Protocol* (UDP) pada layanan *unicast* maupun *multicast* [10].

2.7 Quality of Service (QoS)

Quality of Service (QoS) adalah kemampuan suatu jaringan dalam penyediaan layanan yang baik dengan menyediakan *bandwidth*, mengatasi *delay* dan *jitter*. Tujuan utama QoS adalah untuk menjamin aliran data bagi aplikasi hingga level tertentu, seperti tersedia cukup *bandwidth*, dapat mengendalikan *latency* dan *jitter*, serta mengurangi data *loss*. Ada 2 prinsip yang digunakan pada QoS sebagai berikut:

1. *Parameterized system* (*flow based*)

   Logika yang digunakan didalam *flow based* yaitu *resource network* yang dipesan berdasarkan pertukaran informasi kebutuhan aplikasi dengan jaringan. Prinsip ini sudah diterapkan dalam model lama yaitu *Integrated services*. *Integrated services* (IntServ) menggunakan protokol *Resource Reservation Protocol* (RSVP) yang dapat *merequest* dan memesan *resource* pada jaringan.

2. *Prioritized system* (*class based*)

   Setiap paket diidentifikasi berdasarkan *service* level yang diinginkan. *Differentiated service* (Diffserv) mengimplementasikan prinsip ini, contohnya adalah DSCP[13].

2.7.1 Teknik QoS

Ada 3 (tiga) teknik QoS yang umum digunakan, yaitu:

a. *Best Effort Service*

Best Effort Service digunakan untuk melakukan semua usaha agar dapat mengirimkan sebuah paket ke suatu tujuan. Teknik ini tidak memberikan jaminan bahwa paket dapat sampai ke tujuan yang diinginkan. Bila aplikasi yang digunakan rentan terhadap *network*, *delay*, fluktuasi *bandwidth*, dan perubahan kondisi jaringan, maka teknik *best effort* tidak cocok digunakan pada aplikasi tersebut.

b. *Integrated Service*

Integrated Service (IntServ) menyediakan aplikasi dengan tingkat jaminan layanan melalui negosiasi parameter jaringan secara *end to end*. Aplikasi akan meminta tingkat layanan yang dibutuhkan agar dapat beroperasi dan bergantung pada mekanisme QoS, sehingga dapat menyediakan sumber daya jaringan yang dimulai dari permulaan transmisi pada aplikasi tersebut. Aplikasi tidak akan mengirim trafik jika belum menerima tanda bahwa jaringan tersebut mampu menerima beban yang akan dikirimkan, dan mampu menyediakan QoS yang diminta secara *end to end*. Untuk mencegah terjadinya *over load* pada suatu jaringan, dilakukan suatu proses yang disebut sebagai *admission*

control. Jika QoS yang diminta tidak tersedia, maka jaringan tidak akan mengirimkan tanda ke aplikasi untuk melakukan pengiriman data. Jika aplikasi dapat melakukan pengiriman data, maka sumber daya pada jaringan yang sudah dipesan oleh aplikasi tersebut selesai.

Integrated Service (IntServ) ditunjukkan untuk aplikasi yang peka terhadap *delay* dan keterbatasan *bandwidth*, misal pada video *conference* dan VoIP. Arsitekturnya berdasarkan pada system pencadangan sumber daya per aliran trafik, dimana setiap aplikasi harus mengajukan permintaan *bandwidth* agar dapat melakukan transmisi. *IntServ* memiliki kekurangan pada hal skalabilitas. *IntServ* cocok digunakan untuk *voice* dan video, dan tidak cocok untuk aplikasi *web* yang memiliki aliran trafik yang banyak tetapi berdata kecil. Layanan *IntServ* terbagi menjadi 2 (dua) model, yaitu:

- *Guaranted Service* merupakan layanan dengan batas *bandwidth* dan *delay* yang jelas.
- *Controlled load Service* merupakan layanan dengan presentase *delay* statistik yang terjaga.

c. *Differentiated Service* (DiffServ)

Differentiated Service (DiffServ) menyediakan suatu set perangkat dengan klasifikasi dan mekanisme antrian terhadap protokol atau aplikasi dengan prioritas tertentu di atas jaringan yang berbeda. DiffServ bergantung pada kemampuan *edge router* untuk memberikan klasifikasi dari paket-paket yang berbeda tipe saat melewati suatu jaringan. Trafik jaringan diklasifikasikan berdasarkan alamat jaringan, protokol dan *port*, *ingress interface*, atau klasifikasi lain yang masih didukung oleh Standard *Access List* atau *Extended Access List*.

Keuntungan menggunakan teknik QoS ini adalah:

- *Scability*, berpengaruh pada penanganan jumlah *flow* dan protokol yang digunakan pada suatu jaringan. DiffServ mengumpulkan banyak *flow*, sehingga dapat menangani jumlah *flow* yang besar.

- *Ease of administering* akan memberi kebebasan pada *service provider* untuk memilih penerapan agar dapat menyediakan DiffServ dengan perubahan yang minimal terjadi pada infrastruktur tersebut.
- *Simplicity*, dimana penerapan DiffServ tidak menyimpang banyak dari dasar IP.
- *Measurable*[14].

2.7.2 Parameter – parameter QoS

Parameter –parameter yang ada didalam QoS terdiri dari *throughput*, *packet loss, jitter* dan *delay*.

1. *Packet Loss*

   Packet loss merupakan sebagian kegagalan paket mencapai tujuan atau bisa disebut dengan hilangnya jumlah paket dalam proses pengiriman menuju penerima. Kegagalan paket disebabkan antara lain:

a. Terjadinya tabrakan dalam jaringan dan memory yang terbatas.
b. Pada trafik terjadinya *overload* pada jaringan.
c. Kegagalan pada penerima disebabkan karena *overflow* yang terjadi karena *buffer*.

Nilai *packet loss* yang didapatkan jika semakin kecil jumlah data yang hilang, maka semakin baik[15]. Pada *packet loss* memiliki standard untuk kualitas sebagai berikut.

Tabel 2.1 Kategori *Packet Loss* [15]

Kategori Degradasi	*Packet Loss*
Sangat Bagus	0 %
Bagus	3%
Sedang	15%
Buruk	25%

Berikut rumus yang digunakan untuk mendapatkan jumlah *packet loss*:

$$Packet\ Loss\ = \frac{jumlah\ packet\ yang\ dikirim - jumlah\ paket\ yang\ diterima}{\text{jumlah paket yang dikirim}} X\ 100 \quad (2.1)$$

2. *Throughput*

   Throughput adalah kecepatan rata-rata pada data yang dikirimkan melalui kanal telekomunikasi. Sistem *throughput* atau kemampuan *throughput* yang dapat di artikan sebagai jumlah nilai rata-rata data

yang dikirimkan untuk semua jaringan tersebut. Nilai *throughput* semakin besar yang didapatkan semakin baik jaringan tersebut[15]. berikut ini rumus yang digunakan untuk mengetahui jumlah *throughput*:

$$Throughput = \frac{\text{jumlah data yang dikirim (bit)}}{\text{waktu pengiriman data (s)}} \quad (2.2)$$

3. *Delay*

Delay merupakan sebagai waktu tunda pada suatu data yang diproses pada jaringan atau waktu yang dibutuhkan data untuk menempuh jarak dari suatu *node* ke *node* lainnya sebagai tujuannya [15]. Dalam *delay* juga memiliki standard untuk kualitas sebagai berikut.

Tabel 2.2 Kategori *Delay* [15]

Kategori Degradasi	*Delay*
Sangat Bagus	<150 ms
Bagus	150 ms s/d 300 ms
Sedang	300 ms s/d 450 ms
Buruk	>450 ms

Berikut ini rumus yang digunakan untuk mendapatkan jumlah *delay*:

$$Delay = \frac{\text{jumlah paket diterima}}{\text{jumlah waktu pengamatan (s)}} \quad (2.3)$$

4. *Jitter*

Besarnya nilai *jitter* akan sangat dipengaruhi oleh variasi beban trafik dan besarnya tumbukan antar paket (*congestion*) yang ada dalam jaringan IP. Semakin beban trafik di dalam jaringan akan menyebabkan semakin besar pula peluang terjadinya *congestion* dengan demikian nilai *jitter* akan semakin besar. Dalam *jitter* juga memiliki standard untuk kualitas sebagai berikut.

Tabel 2.3 Kategori *Jitter* [15]

Klasifikasi Standar	*Delay* (ms)
Sangat bagus	0 ms
Bagus	0 ms s/d 75 ms
Sedang	75 ms s/d 125 ms
Buruk	>125 ms

berikut ini rumus yang digunakan untuk mengetahui jumlah *jitter*:

$$Jitter = \frac{\text{jumlah variasi delay (s)}}{\text{jumlah paket yang diterima}} \quad \text{.....24}$$

2.8 Routing Protokol IS-IS

Protocol Intermediate System to Intermediate System (IS-IS) adalah salah satu protokol IP *routing* dan merupakan *Interior Gateway Protocol* (IGP) untuk internet, yang digunakan untuk mendistribusikan informasi *routing* IP di seluruh *Autonomous System* (AS) tunggal pada sebuah jaringan Internet *Protocol*. IS-IS adalah *protocol routing link-state*, yang berarti bahwa *router* menukar informasi topologi dengan tetangga terdekat dari *router* tersebut. Informasi topologi tersebar di seluruh AS, sehingga setiap *router* di AS memiliki gambaran lengkap tentang topologi AS. Gambaran ini kemudian digunakan untuk menghitung jalur *end-to-end* melalui AS, biasanya menggunakan varian dari algoritma Dijkstra. Oleh karena itu, dalam protokol *routing link-state*, alamat *hop* berikutnya yang meneruskan data ditentukan dengan memilih jalur *end-to-end* terbaik ke tujuan akhir.

Keuntungan utama dari sebuah *link state routing protocol* adalah bahwa pengetahuan lengkap tentang topologi dapat memungkinkan *router* untuk menghitung *rute* yang memenuhi kriteria tertentu. Hal tersebut berguna untuk tujuan dari rekayasa lalu lintas, dimana *rute* dapat dibatasi untuk memenuhi persyaratan layanan kualitas tertentu. Kerugian utama dari sebuah *protocol routing link state* adalah bahwa ia tidak berskala baik karena lebih banyak *router* ditambahkan ke domain *routing*. Meningkatkan jumlah pada *router* merupakan suatu ukuran dan frekuensi pembaharuan topologi, dan juga lamanya waktu yang diperlukan untuk menghitung *rute end-to-end*. Kurangnya skalabilitas ini berarti bahwa *protocol routing link*

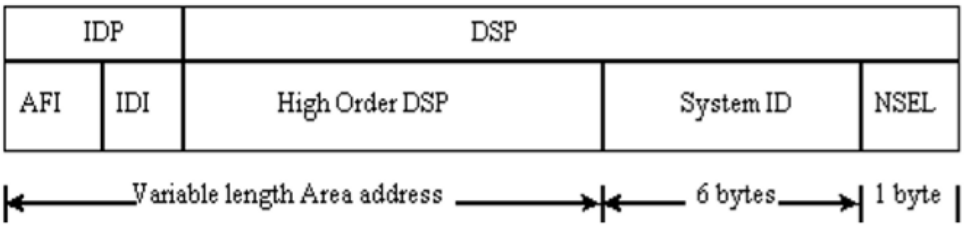
state tidak sesuai untuk di seluruh internet pada umumnya itulah alasan mengapa IGP hanya mengarahkan lalu lintas dalam satu AS tunggal.

IS-IS pada awalnya dirancang sebagai *protocol routing* untuk CLNP, namun telah diperluas untuk mencakup *routing* IP, versi *extended* terkadang disebut *Integrated* IS-IS. Setiap *router* IS-IS mendistribusikan informasi tentang negara setempat (antarmuka yang dapat digunakan dan tetangga yang terjangkau, serta biaya untuk menggunakan setiap antarmuka) ke *router* lain menggunakan pesan *link state* PDU (LSP). Setiap *router* menggunakan pesan yang diterima untuk membangun database identik yang menggambarkan topologi AS. Pada database ini, setiap *router* menghitung tabel routingnya sendiri dengan menggunakan algoritma *Shortest Path First* (SPF) atau Dijkstra. Tabel *routing* ini berisi semua tujuan yang diketahui oleh *protocol routing* terkait dengan alamat IP *hop* berikutnya dan antarmuka keluar.

a. protokol menghitung ulang *rute* saat topologi jaringan berubah, menggunakan algoritma Dijkstra, dan meminimalkan lalu lintas *protocol routing* yang dihasilkannya.
b. Protokol ini menyediakan dukungan untuk beberapa jalur dengan biaya yang sama.
c. Protokol ini menyediakan hirarki *multilevel* (dua tingkat untuk IS-IS) yang disebut *routing area*,sehingga informasi tentang topologi didalam area yang didefenisikan dari AS kemudian disembunyikan dari *router* di luar area ini. Hal ini memungkinkan tingkat tambahan perlindungan *routing* dan pengurangan lalu lintas *protocol routing*.
d. Semua pertukaran protokol dapat diautentikasi sehingga hanya *router* terpercaya yang dapat bergabung dalam pertukaran perutean untuk AS [16].

2.8.1 Struktur Alamat NSAP

Router Cisco dapat mengarahkan data CLNS yang menggunakan pengalamatan sesuai dengan standar ISO 10589. Struktur NSAP diilustrasikan pada gambar 2.7.



Gambar 2.7 Struktur pengamatan NSAP

Alamat OSI NSAP bisa sampai 20 oktet dan terdiri dari bagian berikut, seperti yang ditunjukkan pada Gambar 2.7 yaitu:

a. *Authority and Format ID* (AFI) menentukan format alamat dan otoritas yang menugaskan alamat tersebut.
b. *Inter Domain ID* (IDI) mengidentifikasi domain ini.
c. AFI dan IDI bersama-sama membentuk bagian *InterDomain Part* (IDP) dari alamat NSAP. Hal ini dapat disamakan dengan jaringan utama IP *classful*.
d. *High-Order Domain-Specific Part* (DSP) (HODSP) digunakan untuk membagi domain menjadi beberapa area. Hal ini dapat dianggap OSI setara dengan subnet di IP.
e. ID sistem mengidentifikasi OSI individu (pada ES atau IS). Di OSI, sebuah perangkat memiliki alamat, sama seperti pada protokol DECnet. Ini berbeda dengan IP, dimana sebuah interface memiliki alamat. OSI tidak menentukan panjang tetap untuk ID sistem, namun menentukan bahwa itu konsisten untuk semua perangkat. Perangkat lunak Cisco IOS memperbaiki ID sistem sebagai 6 byte sebelum pemilih NSAP 1-byte (NSEL).
f. NSEL (juga dikenal sebagai N-*selector*, identifier layanan atau ID proses) mengidentifikasi sebuah proses pada perangkat. Ini adalah setara yang longgar dari port atau soket di IP. NSEL panjangnya 1 byte. Hal ini tidak digunakan dalam keputusan *routing*. Bila NSEL diatur ke 00, alamat tersebut mengidentifikasi perangkat itu sendiri alamat level jaringannya. Dalam kasus ini, NSAP dikenal sebagai judul entitas jaringan (NET).

g. HODSP, ID sistem, dan NSEL bersama-sama membentuk bagian spesifikasi-domain (DSP) dari alamat NSAP[17].

2.9 GNS3

Graphical Network Simulator atau yang sering dikenal dengan GNS adalah salah satu aplikasi *emulator* yang dapat berfungsi sebagai mengemulasikan *router* dan membuat jaringan (*network*) topologi untuk keperluan praktik. Selain itu aplikasi dari GNS3 jauh lebih *real* bila dibandingkan dengan apliklasi yang lainnya[18].

2.10 Wireshark

Wireshark adalah sebuah *tool* yang ditunjukan untuk penganalisisan paket data jaringan. *Wireshark* akan melakukan pengamatan paket yang beredar pada suatu jaringan secara *real-time* dan kemudian menangkap data serta menampilkan selengkap mungkin. Dengan menggunakan *wireshark* detail paket dapat diamati dengan melihat nilai parameter yang muncul. *Wireshark* juga memungkinkan penggunanya untuk mengamati data dari jaringan yang sedang beroperasi dan langsung memilah data. Informasi detail bagi masing-masing paket, termasuk *full-header* dan porsi data bisa diperoleh. *Wireshark* mempunyai beberapa fitur termasuk *display fileter language* dan memiliki kemampuan untuk merekonstruksi kembali sebuah aliran pada sesi TCP[19].