BAB 2

LANDASAN TEORI

2.1 Teori Umum

2.1.1 Pengertian Jaringan Komputer

Jaringan komputer adalah sebuah sistem yang terdiri atas komputer, *software* dan perangkat jaringan lainnya yang saling bekerja bersama-sama untuk mencapai suatu kinerja jaringan yang sama. jaringan komputer dapat disebut juga himpunan *interkoneksi* sejumlah komputer *autonomous*. Dua buah komputer dikatakan terinterkoneksi bila keduanya dapat saling bertukar informasi. Tujuan dari jaringan komputer adalah:

- Membagi sumber daya, seperti berbagi pemakaian CPU,

  harddisk, memori, *printer*

- Akses informasi seperti, *web browsing*

- Komunikasi seperti, *chatting* dan *e-mail*

Agar dapat mencapai tujuannya, setiap bagian dari jaringan komputer meminta dan memberikan layanan (*service*). Pihak yang meminta atau menerima layanan disebut klien (*client*) dan yang memberikan atau mengirim layanan disebut pelayan (server). Arsitektur ini disebut dengan sistem *client-server*, dan digunakan pada hampir seluruh aplikasi jaringan computer.

(Sumber : *http://sengkang-flash.blospot.com/2011/01/definisi-jaringan-komputer.html*)

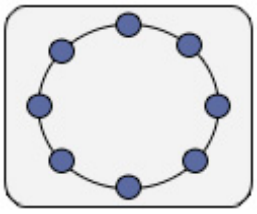
2.1.2 Klasifikasi Jaringan Komputer

2.1.2.1 Berdasarkan Topologi Jaringan

- **Ring**

Pada topologi ini setiap *node* saling berhubungan dengan *node* lainya sehingga berbentuk seperti lingkaran (*ring*). Topologi *token-ring* terlihat pada skema di bawah. Metode *token-ring* (sering disebut *ring* saja) adalah cara menghubungkan komputer sehingga berbentuk *ring* (lingkaran). Setiap simpul mempunyai tingkatan yang sama. Jaringan akan disebut sebagai *loop*, data dikirimkan kesetiap simpul dan setiap informasi yang diterima simpul diperiksa alamatnya apakah data itu untuknya atau bukan. Terdapat keuntungan dan kerugian dari tipe ini yaitu:

- ❖ Keuntungan : Hemat kabel
- ❖ Kerugian : Peka kesalahan, pengembangan jaringan lebih kaku

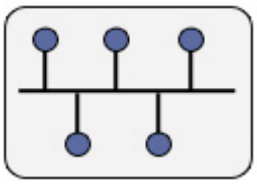


Gambar 2.1 Topologi *Ring*

(Sumber : http://smksantoyusup.wordpress.com/2010/04/22/topologi-jaringan-2/)

- **Bus**

Topologi *bus* disebut juga linear *bus* karena dihubungkan hanya melalui satu kabel yang linear seperti terlihat pada gambar 2.2. kabel yang umum digunakan adalah kabel koaksial.

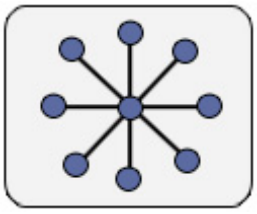


Gambar 2.2 Topologi *Bus*

(Sumber : http://smksantoyusup.wordpress.com/2010/04/22/topologi-jaringan-2/)

- **Star**

Hubungan antar *node* melalui suatu perangkat yang disebut *hub* atau *concentrator*. Setiap *node* dihubungkan dengan kabel ke *hub*.

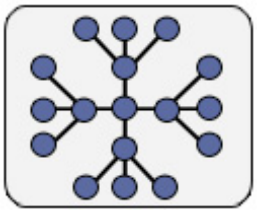


Gambar 2.3 Topologi *Star*

(Sumber : http://smksantoyusup.wordpress.com/2010/04/22/topologi-jaringan-2/)

- **Extended Star**

  Menggabungkan beberapa topologi *star* menjadi satu topologi. *Hub* atau *switch* yang digunakan untuk menghubungkan beberapa komputer pada satu jaringan dengan menggunakan topologi *star* dihubungkan lagi ke *hub* atau *switch* utama.

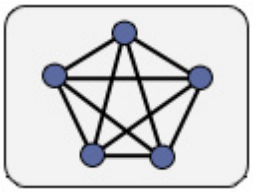


Gambar 2.4 Topologi *Extended Star*

(Sumber : http://smksantoyusup.wordpress.com/2010/04/22/topologi-jaringan-2/)

- **Mesh**

  Setiap komputer memiliki hubungan langsung dengan semua *host* lainnya dalam jaringan. Topologi ini juga merefleksikan internet yang memiliki banyak jalur ke satu titik.

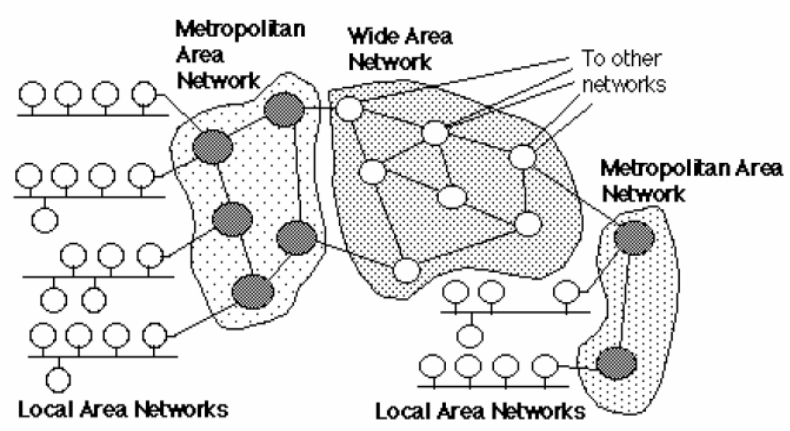


Gambar 2.5 Topologi Mesh

(Sumber : http://smksantoyusup.wordpress.com/2010/04/22/topologi-jaringan-2/)

2.1.2.2 Berdasarkan Luas Cakupan

Berdasarkan dari luas area yang dicakup, jaringan computer terbagimenjadi tiga ukuran, yaitu *Local Area Network* (LAN), *Metropolitan Area Network* (MAN), dan *Wide Area Network* (WAN). Pada gambar 2.6 akan menampilkan cakupan masing – masing area.



Gambar 2.6 Cakupan Daerah Suatu Jaringan (Sumber: http//cnap.binus.ac.id/)

1. **LAN**

   Jaringan yang lingkupnya paling kecil, biasanya mencakup rumah, gedung atau kampus.

2. **MAN**

   Merupakan jaringan yang mencakup sebuah area metropolitan, yaitu sebuah daerah yang lebih besar daripada LAN dalam sebuah area geografis, biasanya terkoneksi dalam satu kota yang jaraknya bisa mencapai 10 km.

3. **WAN**

Merupakan jaringan yang menghubungkan antar LAN yang mencakup jarak geografis yang sangat luas. Dibandingan LAN, WAN lebih pelan, karena membutuhkan permintaan koneksi ketika ingin mengirim data. WAN beroperasi pada *Layer* 1, 2 dan 3 (khususnya X.25 dan *Integrated Services Digital network* (ISDN)).

2.1.3 Peralatan Jaringan

- ***Router***

  Router berfungsi untuk memisahkan jaringan. Dengan menggunakan *routing protocol, router* dapat menentukan jalur terbaik untuk paket-paketnya. *Router* bekerja pada *Layer* 3 pada model OSI (*Network Layer*). *Router* dapat membagi *collision domain* dan *broadcast domain*.

- ***Switch***

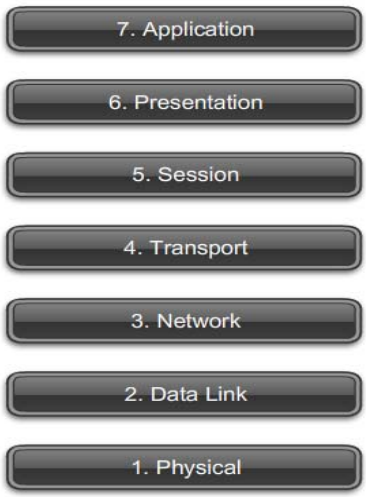
Switch adalah alat penghubung jaringan dengan *forwarding* berdasarkan alamat MAC. *Switch* membagi *collision domain* tetapi tidak membagi *broadcast domain*. *Switch* bekerja pada *layer 2* pada model OSI *(Data link Layer)* dan ada juga yang bekerja pada *layer* 3 (*Network layer*) pada model OSI. Perbedaan yang mendasar antara *switch layer* 2 dan *switch layer* 3 adalah kemampuan switch layer 3 dapat melakukan proses *routing*.

2.1.4 Konsep *Networking* Model

2.1.4.1 Model OSI *Layer*

Tujuan dari OSI *Layer* adalah :

1. Mengurangi kompleksitas dan mempercepat evolusi dalam dunia jaringan, karena masing – masing dapat fokus hanya pada satu layer saja tanpa perlu khawatir dapat mengganggu fungsi dari *layer* yang lain.
2. Menjamin interoperabilitas dan adanya standarisasi untuk berbagai *vendor* (seperti *router* Juniper dengan *router* Cisco, dapat berkomunikasi dengan adanya standarisasi).
3. Membuat perusahaan untuk lebih fokus terhadap salah satu bagian dari ke tujuh *layer* dibawahnya.



Gambar 2.7 Model OSI Layer (Sumber: http//cnap.binus.ac.id/)

Gambar 2.7 merupakan gambar dari model OSI. Model OSI terdiri dari 7 *layer*. *Layer* 7,6,5 disebut dengan *host layer*, maksudnya adalah proses dalam *layer* itu terjadi pada saat data masih di dalam komputer, sedangkan *layer* 4,3,2,1 disebut dengan *media layer*. Berikut penjelasan mengenai ke-7 *layer* tersebut : **(**http://cnap.binus.ac.id/ccna/**)**

1. ***Application Layer* (*Layer* 7)**

   Tugas dari *layer* ini adalah menyiapkan komunikasi *end-to-end.* Berperan sebagai *interface* (yang menghubungkan antara manusia dengan komputer). Protokol yang bekerja pada *layer* 7 adalah : HTTP, FTP, SMTP, Telnet, SNMP.

2. ***Presentation Layer* (Layer 6)**

   Layer ini bertugas untuk mendefinisikan format data, menampilkan data dan menangani kompresi dan enkripsi. Format data yang bekerja pada *layer* 6 adalah : ASCII, JPEG, GIF, MPEG, WAV, MIDI.

3. ***Session Layer* (*Layer* 5)**

   Tugas dari *layer* ini adalah :

   - Memulai dan mengakhiri suatu sesi antar dua *end system*.
   - Menjaga agar dua aplikasi atau lebih dapat berjalan secara bersamaan.
   - Menjaga sesi agar tetap terpisah, sehingga tidak saling tumpah tindih

4. *Transport Layer* (*Layer* 4)

Tugas dari *layer* ini adalah :

- Memikirkan bagaimana data dapat terkirim secara

  1. *Reliable* (dapat dipercaya)

     Mengutamakan pengiriman secara akurat. Contoh : *browsing*, *email*.

  2. *Unreliable*

     Mengutamakan kecepatan dalam mengirim data. Contoh : VoIP, *video streaming*.

- Dapat membuat dan menjelaskan layanan yang digunakan dengan melihat nomor *port*. Contoh : bila menggunakan port 80, artinya sedang melakukan *browsing*.
- Pada *layer* ini terjadi proses segmentasi (memecah data menjadi ukuran yang lebih kecil) dan juga proses *reassemble* (penyusunan kembali, data yang telah dipecah). Protokol yang bekerja pada *layer* 4 adalah : TCP, UDP.

5. *Network Layer* (*Layer* 3)

Layer ini berfungsi untuk mendefinisikan alamat-alamat IP , membuat *header* untuk paket-paket , dan mencari jalur terbaik lalu kemudian melakukan *routing* melalui *internetworking* dengan menggunakan *router* dan *switch Layer-3*. Protokol yang bekerja pada *layer* 3 adalah : IP, IPX, AppleTalk.

6. *Data Link Layer* (*Layer* 2)

Layer ini mendefinisikan bagaimana untuk mengirimkan data melalui suatu media, baik media kabel maupun nirkabel dengan *physical addressing.* Tugas utama dari *layer* ini adalah *error checking, flow control, Media Acces Control* untuk mengatur paket yang akan berjalan. Protokol yang bekerja pada *layer* 2 adalah : PPP, HDLC, Frame Relay, Ethernet, ATM.

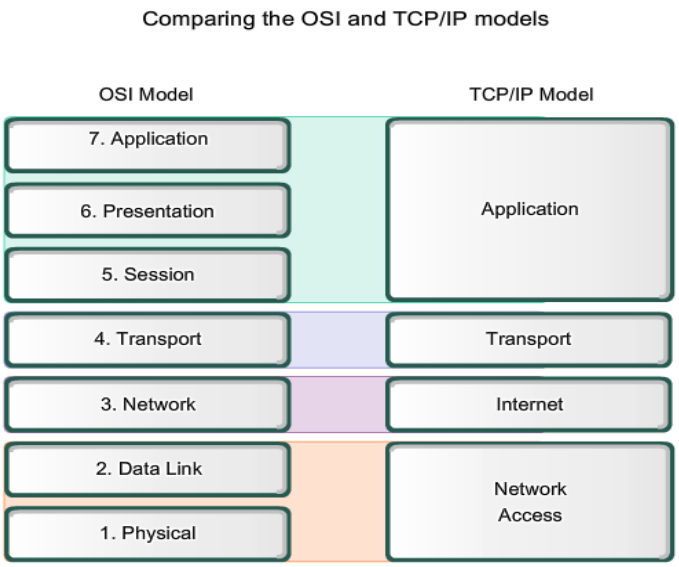
7. *Physical Layer* (Layer 1)

Layer ini berfungsi untuk mendefinisikan media transmisi jaringan, metode pensinyalan, sinkronisasi bit, arsitektur jaringan (seperti halnya *Ethernet* atau *Token Ring*), dan pengabelan. Selain itu, level ini juga mendefinisikan bagaimana *Network Interface Card* (NIC) dapat berinteraksi dengan media kabel atau radio**.** Protokol yang bekerja pada *layer* 1 adalah : Ethernet, V.35, RS-232.

2.1.4.2 Model TCP/IP *Layer*

Model Referensi *Transmission Control Protocol/Internet Protocol* (TCP/IP) diciptakan oleh Departemen Pertahanan Amerika (DARPA) karena mereka menginginkan jaringan yang dapat bertahan dalam kondisi apapun, sekalipun perang nuklir. *Department of Defense* (DOD) menginginkan jaringan yang dapat mengirimkan paket pada setiap saat, dalam kondisi apapun, dari satu titik ke titik lainnya. Dari keinginan tersebut lahirlah model

TCP/IP, dimana menjadi standar pertumbuhan internet. Model TCP/IP Memiliki 4 *layer*: *Layer Application*, *Layer Transport*, *Layer Internet*, dan *Layer Network Access*. Penting untuk diperhatikan bahwa beberapa *layer* pada Model TCP/IP memiliki nama yang sama dengan *layer* pada Model OSI. Jangan keliru antar kedua model tersebut. **(http://cnap.binus.ac.id/ccna/)**



Gambar 2.8 Model TCP/IP *Layer*

(Sumber: http//cnap.binus.ac.id/)

1. ***Layer Application*** adalah sebuah aplikasi yang mengirimkan data ke transport *Layer*. Misalnya FTP, *email* programs dan web *browsers*.
2. ***Layer Transport*** bertanggung jawab untuk komunikasi antara aplikasi. *Layer* ini mengatur aliran informasi dan mungkin menyediakan pemeriksaan *error*. Data dibagi kedalam beberapa paket yang dikirim ke internet *Layer* dengan sebuah

header. *Header* mengandung alamat tujuan, alamat sumber dan *checksum*. *Checksum* diperiksa oleh mesin penerima untuk melihat apakah paket tersebut ada yang hilang pada rute.

3. ***Layer Internetwork*** bertanggung jawab untuk komunikasi antara mesin. *Layer* ini meng-enkapsulasi paket dari *transport Layer* ke dalam IP *datagrams* dan menggunakan algoritma *routing* untuk menentukan kemana *datagram* harus dikirim. Masuknya *datagram* diproses dan diperiksa kesahannya sebelum melewatinya pada *Transport Layer*.
4. ***Layer networks interface*** adalah *level* yang paling bawah dari susunan TCP/IP. *Layer* ini adalah *device driver* yang memungkinkan *datagram* IP dikirim ke atau dari *phisycal network*. Jaringan dapaat berupa sebuah kabel, *Ethernet, frame relay, Token ring*, ISDN, ATM jaringan, radio, satelit atau alat lain yang dapat mentransfer data dari sistem ke sistem. *Layer network interface* adalah abstraksi yang memudahkan komunikasi antara *multitude arsitektur network.*

2.1.5 Protokol TCP/IP

Saat ini, *Internet* dan *World Wide Web* (WWW) adalah istilah yang umum bagi jutaan orang diseluruh dunia. Banyak orang bergantung pada aplikasi – aplikasi yang harus terkoneksi dengan *internet,* seperti surat elektronik dan website. Protokol *Transmission Control Protocol / Internet Protocol* (TCP/IP) merupakan mesin dari *internet* dan jaringan diseluruh dunia. Karena simpel dan

berkemampuan tinggi, TCP/IP terpilih menjadi satu – satunya protokol jaringan yang berada di dunia saat ini.

TCP dan IP dibangun oleh *Department of Defense* (DOD) untuk menghubungkan jaringan komputer yang dibuat oleh vendor berbeda kedalam sebuah jaringan (Internet). Hal tersebut awalnya berhasil karena hanya mengirimkan beberapa layanan dasar seperti : pengiriman *file*, surat elektronik dan *remote login* yang melewati banyak *client* dan *server*. IP menyediakan routing dari sebuah departemen ke jaringan perusahaan, lalu ke jaringan regional dan berakhir di global *internet.* **(**http://www.yale.edu/pclt/ comm/tcpip.htm**)**

Pada zaman komunikasi saat ini, sebuah jaringan harus tahan dari sebuah kerusakan. Oleh karena itu, DOD mendesain TCP/IP secara handal dan secara otomatis memperbaiki apabila ada kegagalan dari suatu node. Dengan desain seperti itu, cocok untuk diterapkan pada jaringan yang sangat besar dengan sedikit pengaturan terpusat.

2.1.5.1 Protokol TCP

TCP didefinisikan dalam RFC 793. TCP mempercayai IP untuk pengiriman data *end-to-end* termasuk masalah routing. TCP menjamin transmisi dan aliran data dari asal ke tujuan.

Karakteristik yang terdapat pada protokol TCP :

1. *Reliability*

   TCP menyediakan pengiriman data yang dapat diandalkan. Untuk dapat diandalkan, TCP menggunakan *field Sequence* dan *Acknowledgment* yang terdapat pada *header* TCP. Bila terdapat TCP *segment* yang rusak maka *segment* yang rusak tersebut akan dikirim ulang.

2. *Flow Control*

   Untuk mencegah data terlalu banyak dikirim dalam satu waktu, maka dilakukan *flow control* dengan *windowing.* TCP memanfaatkan *field Sequence* dan *Acknowledgment* dan *window* yang terdapat pada *header* TCP. Ukuran dari window berubah – ubah setiap waktu. Window awalnya berukuran kecil lalu kemudian membesar hingga terjadi *error*.

3. *Connection – oriented*

   Sebelum data dapat dikirim, terlebih dahulu melakukan pertukaran informasi antar dua *host.*

4. *Data Segmentation*

   TCP membagi data menjadi ukuran yang lebih kecil dan tidak lebih dari ukuran *maximum transmission unit* (MTU). Pada sisi penerima TCP akan melakukan *reassembly* ketika menerima *segment* dan juga dapat mengurutkan kembali *segment – segment* yang datang tidak berurutan.

2.1.5.2 Protokol IP

Layanan layer network yang diimplementasikan pada protokol TCP/IP adalah Internet Protokol (IP). IP versi 4 saat ini yang paling umum digunakan. IP versi 6 diciptakan dan telah diimplementasikan di beberapa tempat, umumnya di *Internet Service Provider*. IP dirancang sebagai protokol dengan tingkat *overhead* yang rendah, IP hanya menyediakan fungsi pengiriman paket dari sumber ke tujuan melalui sistem jaringan yang saling terhubung. IP tidak dirancang untuk mengatur aliran paket. Adapun karakteristik dasar dari IP versi 4 adalah :

1. *Connectionless*

   Paket IP dikirim tanpa memberitahu terlebih dahulu penerima bahwa paket tersebut akan datang. Oleh karena itu, IP tidak memerlukan pertukaran informasi dahulu sebelum IP dapat mengirim paket. Sehingga didalam header PDU tidak perlu ada penambahan *field.* Proses tersebut mengurangi terjadinya *overhead* pada IP.

   Pengiriman paket bersifat *connectionless* berdampak pada tidak berurutnya paket yang diterima ditujuan. Bila hal tersebut terjadi, layanan pada layer diatasnya (TCP) yang akan memecahkan masalah tersebut.

2. *Best-Effort (Unreliable)*

   Protokol IP tidak menyediakan layanan yang *reliable*. Bila dibandingkan dengan protokol yang *reliable*, maka header IP berukuran lebih kecil. Mengirimkan paket yang berukuran kecil

berdampak kecilnya overhead yang terjadi. *Overhead* yang kecil menyebabkan kecilnya terjadi delay dalam pengiriman.

Maksud *reliable* disini bukan berarti IP bekerja pada suatu saat, namun tidak bekerja sebagaimana mestinya pada saat yang lain. *Unreliable* disini berarti IP tidak memiliki kemampuan untuk mengatur, dan memperbaiki paket yang rusak maupun paket yang tidak terkirim.

3. *Media Independent*

IP versi 4 dan IP versi 6 tidak bergantung pada media yang digunakan, IP dapat berkomunikasi pada media kabel, fiber optik maupun sinyal radio. Terdapat karakteristik yang oleh layer *network* perhatikan yaitu ukuran maksimum dari PDU yang tiap media dapat kirimkan. Karakteristik tersebut dikenal sebagai *Maximum Transmission Unit* (MTU). Bagian dari pengaturan komunikasi antara layer *Data Link* dan layer *Network.* Layer *Data Link* melewatkan MTU naik ke layer *Network* dan menentukan seberapa besar ukuran pembuatan paket. Pada beberapa kasus, *intermediary device* seperti *router* akan membagi paket ketika akan dikirim dari satu media ke media lain dengan ukuran MTU yang lebih rendah. Proses itu disebut dengan istilah *fragmentation.*

2.1.5.2.1 Pengalamatan IP

Internet terdiri dari jutaan *host* dan dimana masing – masing diidentifikasi secara unik oleh pengalamatan pada layer *Network.* Untuk berharap setiap *host* dapat mengetahui alamat dari *host* yang

lain dapat menyebabkan performa dari peralatan jaringan yang dapat menurun. Membagi jaringan besar menjadi kumpulan grup yang lebih kecil dapat mengurangi *overhead* yang tidak perlu.

Untuk dapat membagi suatu jaringan, kita memerlukan pengalamatan yang terstruktur (hirarki), yang juga digunakan untu komunikasi data antar jaringan melalui internetwork.

IP versi 4 memiliki pengalamatan terstruktur, terdiri dari 32 bit yang ditulis dalam nilai – nilai desimal 4. Desimal tersebut terdiri dari 1 byte atau 8 bit. Setiap desimal dalam alamat IP disebut juga sebagai oktet.

IP versi 4 didefinisikan pada RFC 791, dimana dijelaskan juga pembagian kedalam kelas – kelas. Alamat IP terdiri dari dua bagian yaitu *network* ID dan *host* ID. Dimana *network* ID menentukan alamat jaringan dan *host* ID menentukan alamat *host* atau komputer. Untuk menentukan alamat kelas IP, dilakukan dengan memeriksa 4 bit pertama (bit yang paling kiri) dari alamat IP.

Tabel 2.1 Alamat Kelas IP

Kelas	Alamat *Bit* Pertama	Desimal
A	0xxx	1-126
B	10xx	128-191
C	110x	192-223
D	1110	224-239
E	1111	240-254

1. **Kelas A**

   Bit pertama alamat IP kelas A adalah 0, *network* ID 8 bit dan panjang *host* ID 24 bit. Kelas A digunakan untuk jaringan yang berskala besar, terdapat 126 jaringan dan tiap jaringan dapat menampung hingga 16 juta *host*. Alamat IP kelas A dimulai dari 1.0.0.0 sampai dengan 126.255.255.255. Alamat oktet awal 127 tidak boleh digunakan karena digunakan untuk mekanisme *Inter-process Communication* di dalam perangkat jaringan yang bersangkutan.

2. **Kelas B**

   Dua bit awal dari kelas B selalu diset 10 sehingga *byte* pertama kelas B bernilai antara 128 – 191. *Network* ID adalah 16 bit pertama dan *host* ID 16 bit sisanya. Kelas B digunakan untuk jaringan berskala menengah hingga besar, terdapat 16.384 jaringan dan tiap jaringan dapat menampung sekitar 65 ribu *host*. Alamat kelas B dimulai dari 128.0.0.0 sampai dengan 192.167.255.255.

3. **Kelas C**

   Tiga bit awal dari kelas C selalu diset 111, sehingga *byte* pertama kelas C bernilai antara 192 – 223. *Network* ID adalah 24 bit dan *host* ID 8 bit sisanya. Kelas C biasa digunakan untuk jaringan kecil, terdapat 2.097.152 jaringan dan tiap jaringan dapat menampung 256 *host*. Alamat kelas C dimulai dari 192.168.0.0 sampai dengan 223.255.255.255.

4. Kelas D

Empat bit awal dari kelas D selalu diset 1110, sehingga *byte* pertama kelas D bernilai antara 224 - 239. Kelas D digunakan untuk keperluan multicast, yaitu suatu metode pengiriman yang digunakan bila suatu *host* ingin berkomunikasi dengan beberapa *host* sekaligus, dengan hanya mengirim satu datagram saja. Alamat dari kelas D adalah 224.0.0.0 sampai dengan 239.255.255.255. Alokasi alamat tersebut ditujukan untuk keperluan sebuah grup, bukan untuk *host* seperti pada kelas A, B dan C.

5. Kelas E

Empat bit awal dari kelas E selalu diset 1111, sehingga *byte* pertama kelas E bernilai antara 240 – 254. Kelas E digunakan sebagai kelas eksperimental yang disiapkan untuk keperluan di masa mendatang.

2.1.5.2.2 *Private* dan *Public* IP *Address*

1. *Private* IP address

Hampir seluruh alamat pada IPv4 merupakan alamat publik yang dapat digunakan pada jaringan internet, namun terdapat juga blok alamat yang digunakan untuk keperluan terbatas atau tidak terhubung dengan internet. Alamat tersebut disebut sebagai alamat *Private.*

Blok alamat *private* adalah :

- 10.0.0.0 – 10.255.255.255

- 172.16.0.0 – 172.31.255.255
- 192.168.0.0 – 192.168.255.255

Host - host yang tidak memerlukan akses ke internet dapat menggunakan alamat *private* sebanyak apapun. Namun, jaringan internal tetap harus didesain dengan pengalamatan yang baik dan terstruktur sehingga alamat yang digunakan tetap unik untuk network internal tersebut.

Host yang berada di jaringan yang berbeda dapat menggunakan alamat *private* yang sama. Paket yang menggunakan alamat tersebut sebagai *souce* dan *destination* tidak akan muncul di jaringan internet. *Router* atau firewall yang terletak di ujung jaringan tersebut harus memblok atau menterjemahkan alamat – alamat tersebut.

2. *Public Address*

Umumnya alamat IPv4 merupakan alamat publik. Alamat tersebut didesain untuk digunakan pada *host* yang dapat diakses oleh *host* lain melalui internet.

2.1.5.2.3 *Network Address Translation* (NAT)

Dengan NAT, alamat *private* dapat diterjemahkan menjadi alamat publik, sehingga suatu host pada jaringan *private* dapat mengakses layanan yang berada di internet. NAT diimplementasikan pada ujung dari suatu jaringan *private*. NAT memungkinkan *host – host* untuk meminjam alamat publik agar dapat berkomunikasi dengan jaringan

di luar jaringan *private* tersebut. **(**http://www.dahlan.web.id/files/Network%20Address%20Translation.pdf**)**

2.1.5.2.4 IP *Subnetting*

Subnetting adalah teknik membuat banyak jaringan dari suatu alamat blok IP. Karena kita menggunakan *router* untuk membuat jaringan yang berbeda untuk dapat terhubung, maka setiap *interface* pada *router* tersebut harus memiliki alamat IP yang unik.

Kita membuat *subnet* dengan cara meminjam satu atau lebih *host* bit sebagai *network* bit. Semakin banyak kita meminjam *host* bit, maka semakin banyak *subnet* yang dapat dibuat. Untuk setiap bit yang dipinjam, kita menggandakan jumlah *subnetwork* yang tersedia. Contohnya, bila kita meminjam 1 bit, kita dapat mendefinisikan menjadi 2 bit. Namun, semakin banyak kita meminjam bit, semakin sedikit alamat yang dapat digunakan oleh *host* per *subnet*.

2.1.5.2.5 *Subnet Mask*

Subnet mask digunakan bersamaan dengan alamat IP untuk mendefinisikan *subnet* mana dari sebuah alamat IP berada dengan mengidentifikasi *host* bit dan *network* bit. *Router* hanya memeriksa *network* bit dalam sebuah alamat IP yang diindikasikan oleh *subnet mask*,ketika menjalankan fungsi routing. Subnet mask terdiri dari 32 bit sama seperti alamat IPv4. Bila tidak melakukan *subnetting* maka *default subnet mask*nya adalah sebagai berikut :

Tabel 2.2 *Default Subnet Mask*

Kelas	Desimal	*Binary*
A	255.0.0.0	11111111.00000000.00000000.00000000
B	255.255.0.0	11111111.11111111.00000000.00000000
C	255.255.255.0	11111111.11111111.11111111.00000000

2.1.6 *Routing*

Pada saat pengiriman paket, paket tersebut dapat melewati jaringan yang berbeda. *Intermediary device,* seperti *router* adalah perangkat jaringan yang digunakan untuk menghubungkan antara jaringan tersebut. Selain itu, peran dari *router* adalah untuk memilih jalur terbaik dan membawa paket ke tujuan, proses tersebut disebut dengan *routing.* **(**http://cnap.binus.ac.id/ccna/**)**

Pada proses *routing* yang melalui jaringan yang berbeda, paket tersebut akan melewati beberapa *intermediary device*. Setiap perangkat atau *device* yang dilalui paket untuk dapat sampai ke tujuan disebut dengan *hop*.

Router memiliki *routing table*, yang berisi :

1. Daftar jaringan yang terhubung langsung dengan *router* tersebut (*directly connected network*).
2. Jalur menuju jaringan yang tidak terhubung langsung dengan *router* tersebut (*remote network*).
3. Alamat *default route* (0.0.0.0).

Routing terbagi dengan dua cara, yaitu :

1. ***Static Route***

Static route digunakan dalam sebuah jaringan yang hanya terdiri dari beberapa *router* saja atau dipakai untuk jaringan kecil dan jaringan yang terhubung ke internet hanya melalui satu *Internet service provider*. Digunakan *static route* karena hanya *Internet service provider* tersebut yang menjadi jalan keluar untuk akses ke internet.

Dalam *static route*, pengisian dan pemeliharaan *routing table* dilakukan secara manual oleh administrator. Kelebihan dalam *static* route yaitu tidak memerlukan *bandwith* jaringan yang besar akan tetapi jika salah satu jalur routing-nya terputus maka router tidak bisa mencari alternative jalan baru untuk meneruskan paket data yang dikirim.

2. ***Dynamic Route***

Dynamic Route mempelajari rute sendiri yang terbaik yang akan ditempuhnya untuk meneruskan paket dari sebuah jaringan ke jaringan lainnya. Administrator tidak menentukan rute yang harus ditempuh oleh paket-paket tersebut. Administrator hanya menentukan bagaimana cara *router* mempelajari paket dan kemudian router mempelajarinya sendiri. Rute pada *dynamic routing* berubah sesuai dengan informasi yang didapatkan oleh *router*.

Dynamic route ini digunakan apabila jaringan memiliki lebih dari satu kemungkinan rute untuk tujuan yang sama. Sebuah *dynamic routing* dibangun berdasarkan informasi yang dikumpulkan oleh *routing protocol*.

Protokol ini didesain untuk mendistribusikan informasi secara dinamis yang mengikuti perubahan kondisi jaringan. *Routing protocol* mengatasi situasi *routing* yang kompleks secara cepat dan akurat. *Routing protocol* dirancang tidak hanya untuk mengubah ke rute backup bila rute utama putus, namun juga dirancang untuk menentukan rute mana yang terbaik untuk mencapai tujuan tersebut.

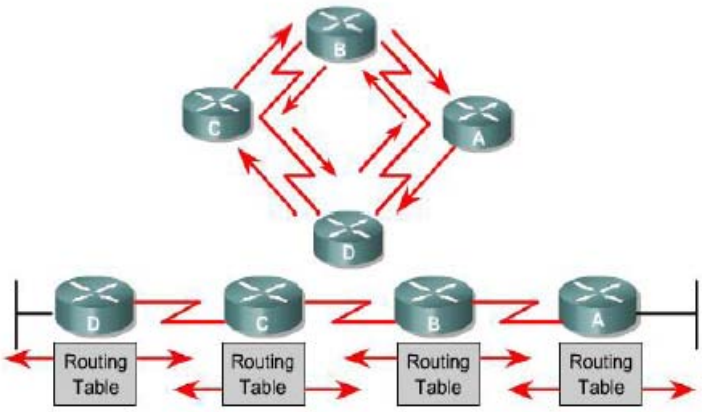
Pengisian dan pemeliharaan *routing table* tidak dilakukan secara manual oleh administrator. Router saling bertukar informasi agar dapat mengetahui alamat tujuan dan menerima *routing table*. Pemeliharaan jalur dilakukan berdasarkan pada jarak terpendek antara perangkat pengirim dan perangkat tujuan.

Dynamic routing protocol terdiri dari beberapa kategori, yaitu :

1. *Distance Vector Route Protocol* (DVRP)

   Routing protocol ini hanya tahu mengenai jarak dan arah. Jarak yang dimaksud dengan jumlah dari *hop count*, sedangkan arah merupakan *next hop router* atau *exit interface.*

   Contoh *distance vector* adalah *Routing Information Protocol* (RIP) *version* 1, RIP *version* 2, *Interior Gateway Routing Protocol* (IGRP), *Enhanced Interior Gateway Routing Protocol* (EIGRP).



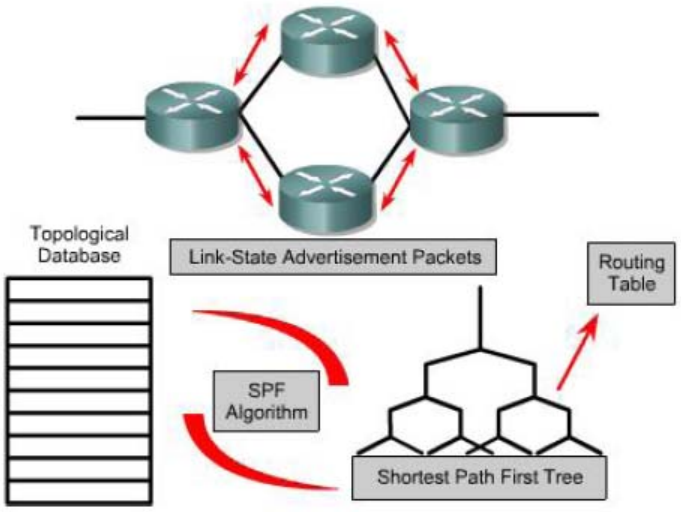
Gambar 2.9 Konsep *Distance Vektor*

(Sumber: http//cnap.binus.ac.id/)

2. *Link State Routing Protocol* (LSRP)

Routing protocol ini lebih *modern* dibanding *distance vector*. Algoritma pada *Link State Routing Protocol* ini menghitung dan menggunakan jalan yang terpendek ke *router* lain. Kelebihan *routing protocol* jenis ini adalah informasi akan di*update* dikirim jika ada perubahan topologi jaringan, lebih cepat untuk konvergen, tidak rentan terhadap *routing loop*, dan lebih sedikit menghabiskan *bandwidth* dibanding *distance vector*,. Sedangkan kelemahannya antara lain lebih sulit untuk dikonfigurasi dan membutuhkan lebih banyak memori dan *processing power*mengambil pandangan umum seluruh topologi jaringan.

Contoh *Link State Routing Protocol* adalah OSPF dan IS-IS.



Gambar 2.10 Konsep *Link-State*

(Sumber: http//cnap.binus.ac.id/)

3. *Hybrid Routing Protocol*

Hybrid routing protocol adalah merupakan kombinasi dari *distance vector* dan *link-state routing protocol*, dimana bekerja dengan cara berbagi informasi mengenai seluruh jaringan dengan *router* tetangga. *Hybrid routing protocol* ini hadir setelah Cisco System membuat *routing protocol* EIGRP (*Enhanced Interior Gateway Routing Protocol*) yang merupakan pengembangan dari IGRP klasik yang bersifat *open standart*. EIGRP dari Cisco ini bersifat *proprietary*, dengan kata lain hanya dapat digunakan oleh perangkat jaringan buatan Cisco (http://cnap.binus.ac.id/ccna/)

2.1.6.1 *Routing Protocol*

2.1.6.1.1 *Autonomous System*

Autonomous system adalah kumpulan jaringan yang berada pada kontrol administrasi yang sama, biasanya sebuah perusahaan atau organisasi yang sama memiliki *autonomous system* yang sama juga (http://cnap.binus.ac.id/ccna/).

2.1.6.1.2 *Routing Information Protocol* (RIP)

Routing Information Protocol (RIP) adalah *routing protocol* yang mencari jalur terbaik menggunakan *hop count* sebagai *metric*. Jumlah maksimal hop yang diperbolehkan adalah 15, bila mencapai *hop* ke-16 maka akan terjadi *destination unreachable* (http://cnap.binus.ac.id/ccna/).

Secara *default* periode *update* dilakukan secara *broadcast* atau *multicast* setiap 30 detik.

Di dalam RIP terdapat 3 jenis waktu, yaitu :

1. *Default Invalid Timer*

   Lamanya waktu sejak suatu *router* tidak pernah mengirimkan paket *update* hingga dinyatakan *invalid* dalam *routing table* di *router* tetangganya. Namun informasinya belum dihapus (update + 150 detik = 180 detik).

2. *Flush Timer*

   Waktu yang diperlukan ketika suatu *router* menghapus informasi tentang *router* tetangganya dari *routing table*nya sejak dinyatakan *invalid* (240 detik).

3. *Holddown Timer*

   Adalah lamanya waktu dimana informasi yang *invalid* masih disimpan oleh suatu router hingga suatu router dinyatakan valid kembali (180 detik).

RIP memiliki 3 versi yaitu RIPv1, RIPv2, dan RIPng.

1. RIPv1

RIPv1 menggunakan *classfull routing*, tidak mendukung *subnetting* dan tidak mendukung *Variable Length Subnet Mark* (VLSM). Penyebaran informasi RIPv1 secara *broadcast*. RIPv1 didefinisikan pada RFC 1058

2. RIPv2

RIPv2 hadir sekitar tahun 1994 yang mampu menggunakan *classless inter-domain* routing. RIPv2 mendukung VLSM, *subnetting,* dan authentikasi. Penyebaran informasi RIPv2 secara *multicast*. RIPv2 didefinisikan pada RFC 2453

3. RIPng

RIPng merupakan protokol RIP untuk IPv6. RIPng didefinisikan pada RFC 2080.

2.1.6.1.3 *Interior Gateway Routing Protocol (*IGRP*)*

*Interior Gateway Routing Protocol (*IGRP*)* adalah protokol yang diciptakan untuk mengatasi kekurangan RIP. *Metric*-nya berupa gabungan *bandwith, delay* dan *load*. *Routing update* yang dilakukan IGRP secara *broadcast* dan tiap 90 detik. Jumlah maksimal hop yang diperbolehkan adalah 255.

IGRP telah mengatasi beberapa kekurangan dari RIP, tetapi IGRP tidak mendukung VLSM. Maka dari itu, Cisco telah membuat EIGRP untuk memperbaiki masalah ini (http://cnap.binus.ac.id/ccna/)

2.1.6.1.4 *Enhanced Interior Gateway Routing Protocol* (EIGRP)

Enhanced Interior Gateway Routing Protocol (EIGRP) adalah protokol dengan optimalisasi untuk meminimalkan ketidakstabilan *routing* yang terjadi setelah perubahan topologi, serta penggunaan dan pengolahan daya *bandwith* pada router. EIGRP menggunakan algoritma *Diffusing Update Algorithm* (DUAL) untuk mencari jalur terbaik (http://cnap. binus.ac.id/ccna/).

Di dalam EIGRP tidak ada *periodic update*, tetapi menggunakan *trigerred update*, yaitu waktu untuk melakukan

update *routing table* saat ada perubahan topologi (ketika ada jalur yang putus atau memang ada perubahan topologi). Jumlah maksimal hop yang diperbolehkan adalah 255.

EIGRP merupakan *proprietary* Cisco yang merupakan kelemahan dari EIGRP karena hanya berjalan pada *vendor* Cisco saja, tidak bisa dari *vendor* yang lain. EIGRP menggunakan beberapa istilah, yaitu :

1. ***Successor***

   Istilah yang digunakan untuk jalur terbaik berdasarkan *metric*..

2. ***Feasible Successor***

   Istilah yang digunakan untuk jalur yang akan digunakan untuk *backup route.*

3. ***Neighbor table***

   Istilah yang digunakan untuk tabel yang berisi alamat dan *interface* untuk mengakses ke *router* sebelah atau *directly connected.*

4. ***Topology table***

   Istilah yang digunakan untuk tabel yang berisi semua tujuan dari *router* sekitarnya.

5. ***Reliable transport protocol* (RTP)**

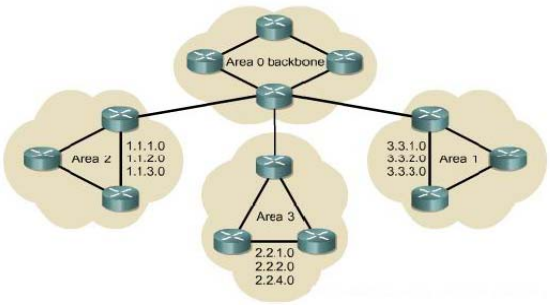
   Protokol yang digunakan EIGRP untuk mengirim dan menerima paket.

2.1.6.1.5 *Open Shortest-Path First* (OSPF)

Open Shortest-Path First (OSPF) merupakan jenis *link state routing protocol* yang melakukan perhitungan jalur terpendek menggunakan *bandwith* (http://cnap.binus.ac.id/ccna/).

Tipe Paket OSPF :

1. ***Hello packet*** – Paket hello digunakan untuk membangun dan memelihara *adjacency* dengan *router* OSPF lainnya.
2. **DBD** – *Database Description* (DBD) berisi daftar-daftar dari *database link state router* pengirim dan digunakan oleh *router* penerima untuk memeriksa dan dibandingkan dengan *database link state local*.
3. **LSR** – *Receiving Routers* kemudian bisa meminta informasi lebih lanjut tentang isi di dalam DBD dengan mengirim *Link-State Request* (LSR)
4. **LSU** – *Link State Update* (LSU) paket digunakan untuk me-*reply* ke LSRs serta mengumumkan informasi baru. LSUs berisi tujuh jenis *Link-State Advertisements* (LSAs) yang berbeda.
5. **LSAck** – Ketika sebuah LSU diterima, *router* mengirim sebuah *Link-state Acknowledgement* (LSAck) sebagai konfirmasi penerimaan LSU.

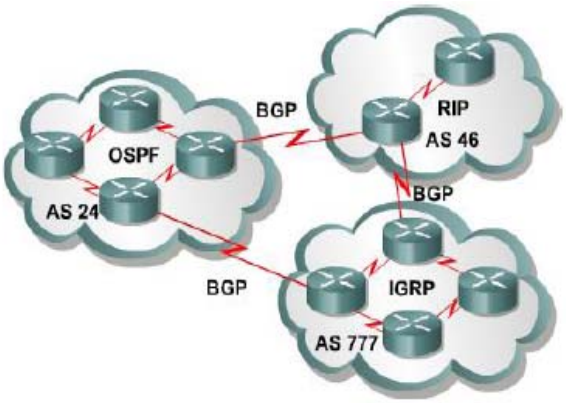


Gambar 2.11 Area Pada OSPF

(Sumber: http//cnap.binus.ac.id/)

2.1.6.1.6 *Border Gateway Protocol* (BGP)

Border Gateway Protocol atau lebih familiar dikenal dengan nama BGP merupakan sebuah protokol *routing inter-Autonomous System*. Fungsi utama sistem BGP adalah untuk bertukar informasi *network* yang dapat 'dijangkau' (*reachability*) oleh sistem BGP lain, termasuk di dalamnya informasi-informasi yang terdapat dalam list *autonomous system* (AS). BGP berjalan melalui sebuah protokol *transport*, yaitu TCP.



Gambar 2.12 BGP

(Sumber: http//cnap.binus.ac.id/)

2.2 Teori Khusus

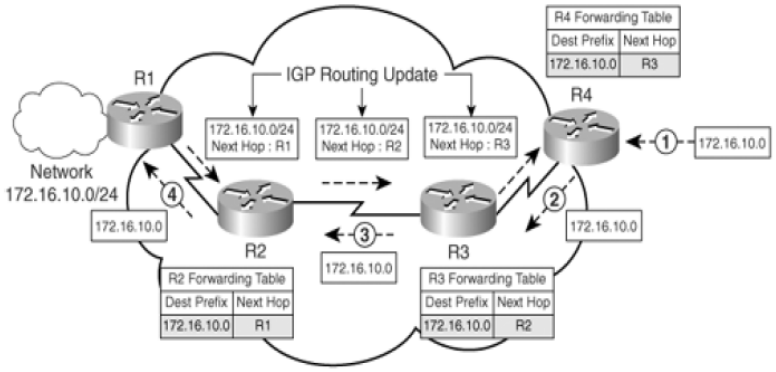
2.2.1 *Multiprotocol Label Switching* (MPLS)

2.2.1.1 Pendahuluan

Menurut *Cisco Systems Learning* (2006), *Multiprotocol Label Switching* (MPLS) adalah sebuah metode dengan performa tinggi untuk meneruskan paket melewati suatu jaringan. MPLS mengizinkan *router* yang berada di *edge network* untuk menyisipkan label yang simple kedalam sebuah paket. Praktek ini mengizinkan perangkat MPLS (ATM *switch* maupun *router* yang ada di tengah *Internet service provider core*) untuk menyisipkan label di setiap paket.

2.2.1.2 *Packet Forwarding* pada jaringan IP Tradisional *Versus* MPLS

Pada jaringan IP tradisional, *routing protocol* digunakan untuk mendistribusikan informasi *routing* di Layer 3. Proses penerusan paket dilakukan berdasarkan alamat tujuan. Oleh karena itu, ketika sebuah paket diterima suatu *router*, maka *router* tersebut akan menentukan *next-hop address* menggunakan alamat IP tujuan dengan informasi yang terdapat pada tabel *routing*. Proses ini akan terus berulang pada tiap *hop* (*router*) dari sumber ke tujuan. **(**http://cnap.binus.ac.id/ccna/**)**



Gambar 2.13 Operasi IP *Forwarding* Tradisional

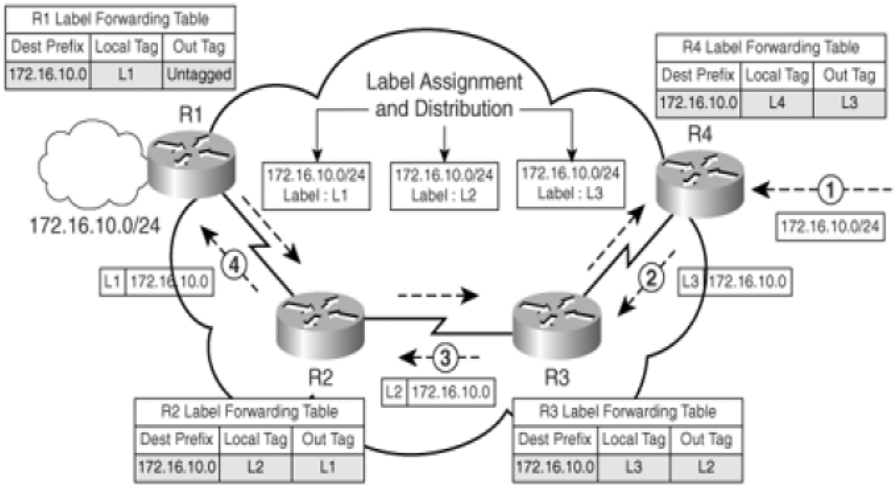
(Sumber:

http://www.cisco.com/en/US/products/ps6557/prod_presentation_list.html)

Berdasarkan Gambar 2.15 proses penerusan paket adalah sebagai berikut:

1. R4 menerima sebuah paket data yang ditujukan untuk jaringan 172.16.10.0
2. R4 mencari rute untuk jaringan 172.16.10.0 pada label routing dan paket diteruskan ke *next-hop*, *router* R3.
3. R3 menerima paket data tersebut dengan tujuan 172.16.10.0 lalu mencari rute untuk jaringan 172.16.10.0 dan kemudian meneruskannya ke *router* R2.
4. R2 menerima paket data tersebut dengan tujuan 172.16.10.0 lalu mencari rute untuk jaringan 172.16.10.0 dan meneruskannya ke *router* R1.
5. Karena router Rl terhubung langsung ke jaringan 172.16.10.0, Rl akan meneruskan paket tersebut ke *interface* yang tepat.

Sedangkan pada jaringan MPLS, paket data diteruskan berdasarkan label. Label mungkin akan disesuaikan dengan alamat IP tujuan atau dengan parameter lainnya, misalnya kelas-kelas QoS dan alamat sumber. (http://cnap.binus.ac.id/ccna/)



Gambar 2.14 Operasi Paket *Forwarding* Pada Jaringan MPLS

(Sumber: http://www.cisco.com/en/US/products/ps6557/prod_presentation_list.html)

Berdasarkan Gambar 2.16, proses penerusan paket adalah sebagai berikut :

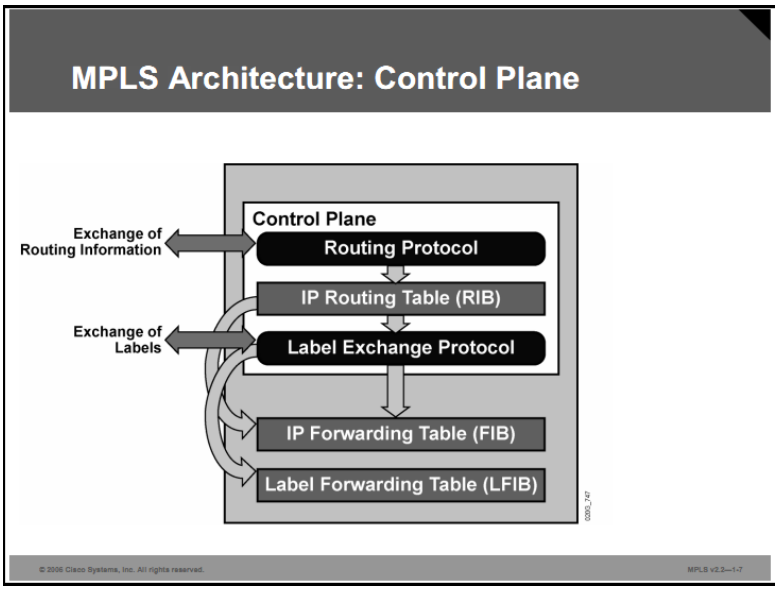
1. R4 menerima sebuah paket data dan jaringan 172.16.10.0 dan mengidentifikasi bahwa rute ke tujuan adalah jaringan MPLS. Oleh karena itu, R4 meneruskan paket tersebut ke *next-hop router* R3 setelah memakaikan sebuah label L3 pada paket tersebut.
2. R3 menerima paket yang berlabel tersebut dengan label L3 dan menukar L3 dengan L2 dan meneruskan paket tersebut ke R2.

3. R2 menerima paket yang berlabel tersebut dengan label L2 dan menukar L2 dengan LI dan meneruskan paket tersebut ke Rl.
4. Rl *router* yang bertindak sebagai batas antara jaringan berbasis IP dan MPLS; oleh karena itu, Rl melepaskan label pada paket dan meneruskan paket IP tersebut ke jaringan 172.16.10.0.

2.2.1.3 Arsitektur MPLS

Menurut *Cisco System Learning*(2006), Fungsionalitas MPLS dibagi menjadi dua bagian utama blok arsitektur, yaitu:

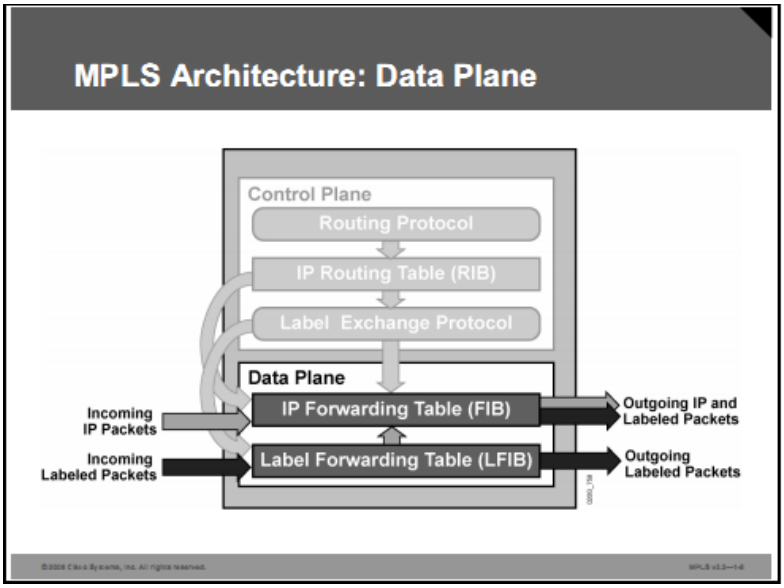
1. *Control Plane* – menjaga pertukaran informasi *routing* dan pertukaran label diantara perangkat jaringan. C*ontrol plane* membangun *routing table (Routing Information Base[RIB])* berdasarkan *routing protocol* untuk pengaturan *routing* di layer 3. Contoh fungsi *control plane* adalah pertukaran informasi protokol *routing*, seperti OSPF dan BGP. Selain itu, semua fungsi yang berhubungan dengan pertukaran label antar *router-router* tetangga.



Gambar 2.15 Arsitektur *Control Plane*

(Sumber: Implementing Cisco MPLS Volume 1 : Introducing Basic MPLS Concepts)

2. *Data Plane* - bertugas untuk menjaga penerusan paket-paket data berdasarkan suatu tujuan alamat IP atau label. *Data plane* disebut juga *forwarding plane*. *Data plane* adalah penerus paket sederhana dimana hanya meneruskan suatu tipe dari *routing* protokol atau pertukaran protokol label yang akan digunakan. *Data plane* mengirimkan paket ke *interface* yang tepat berdasarkan informasi yang berasal dari tabel LFIB atau FIB.



Gambar 2.16 Arsitektur Data Plane

(Sumber: Implementing Cisco MPLS Volume 1 : Introducing Basic MPLS Concepts)

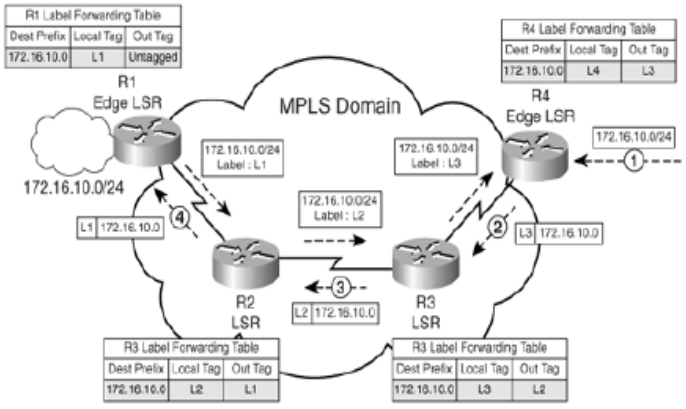
2.2.1.4 Istilah-Istilah Dalam MPLS

Menurut Cisco System Learning(2006), Beberapa istilah penting dalam MPLS yang akan digunakan terus dalam skripsi ini, yaitu :

1. *Forwarding Equivalent Class* (FEC) - merupakan sekumpulan paket-paket yang akan mendapatkan perlakuan *forwarding* yang sama (melewati jalur yang sama).
2. MPLS *Label Switch Router* (LSR) - bertugas dalam *label switching*; LSR menerima *labeled packet* dan menukar *label* tersebut dengan *outgoing label* dan meneruskan *labeled packet* baru tersebut dari *interface* yang tepat. Berdasarkan lokasinya dalam *domain* MPLS, LSR bisa bertugas dalam *label imposition*

(addition, disebut juga push) atau pun *label disposition* (removal, disebut juga pop).

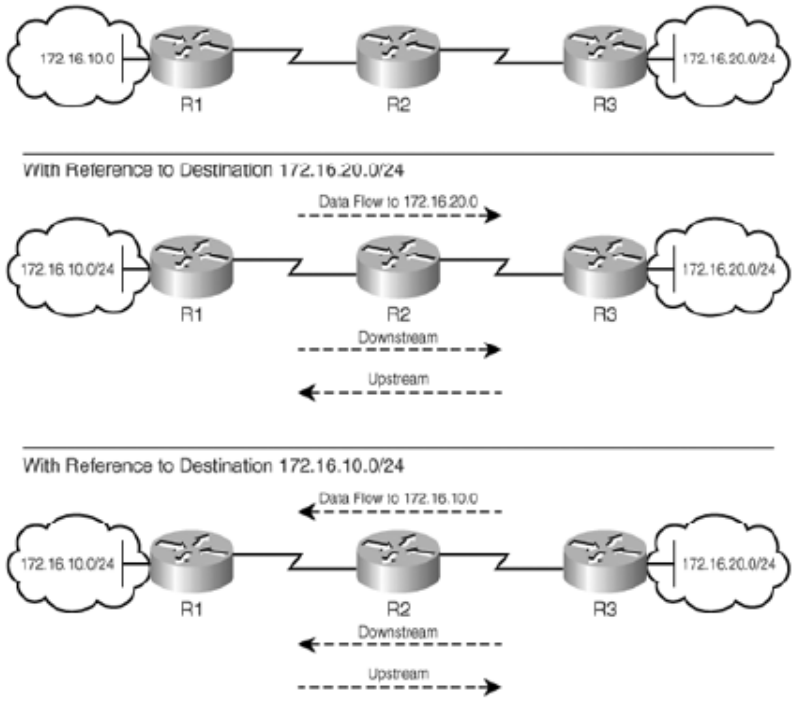
3. MPLS *Edge-Label Switch Router* (E-LSR) – sebuah LSR pada perbatasan domain MPLS. Ingress E-LSR bertugas dalam *label imposition* dan meneruskan paket melalui jaringan *MPLS-enabled*. Egress E-LSR bertugas dalam *label disposition* dan meneruskan paket *IP* ke tujuan.



Gambar 2.17 LSR dan E-LSR

(Sumber:http://www.cisco.com/en/US/products/ps6557/prod_presentation_list.html)

4. MPLS *Label Switched Path* (LSP) – jalur pengiriman paket dari sumber ke tujuan pada jaringan MPLS-*enabled*
5. *Upstream and Downstream* – konsep dari *upstream* dan *downstream* merupakan poros untuk memahami operasi dari distribusi *label* (*control plane*) dan penerusan paket data dalam sebuah *domain* MPLS.



Gambar 2.18 *Upstream* dan *Downstream*

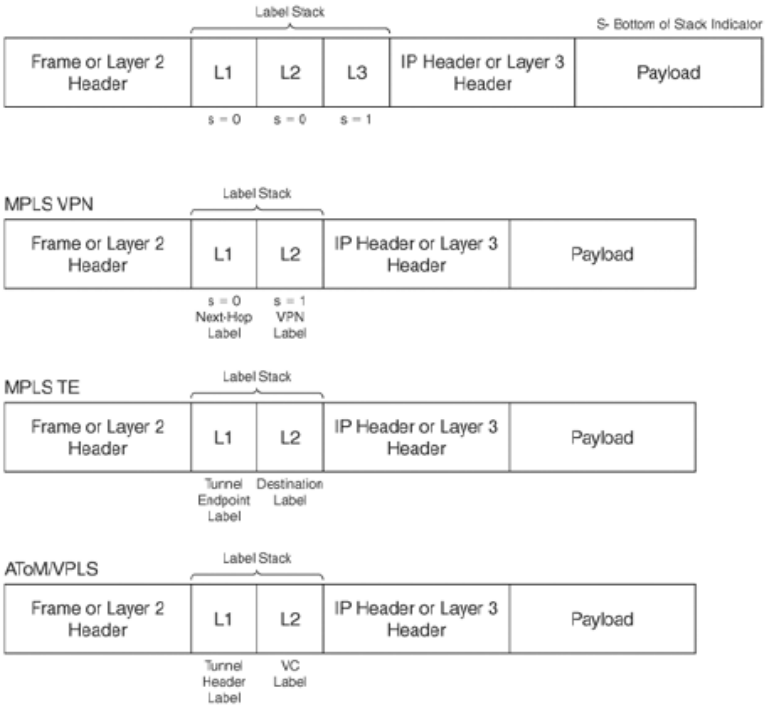
(Sumber:

http://www.cisco.com/en/US/products/ps6557/prod_presentation_list.html)

Sebuah *label* MPLS terdiri dari bagian-bagian berikut ini:

1. 20-bit *label value* – nomor yang ditetapkan oleh *router* untuk mengidentifikasikan *prefix* yang diminta.
2. 3-bit *experimental field* – mendefinisikan QoS yang diberikan pada FEC yang telah diberi *label*.
3. 1-bit *bottom-of-stack indicator* – jika E-LSR menambahkan lebih dari satu *label* pada sebuah paket IP, maka akan terbentuk *label stack*. Oleh karena itu, *bottom-of-stack*

indicator bertugas untuk mengenal apakah sebuah *label* yang dijumpai merupakan *label* terbawah dalam *label stack.*



Gambar 2.19 MPLS *Label Stack*

(Sumber:http://www.cisco.com/en/US/products/ps6557/prod_presentation_list.html)

4. 8-bit *Time-to-Live field* – memiliki fungsi yang sama dengan IP TTL, di mana paket akan dibuang jika TTL sebuah paket adalah 0. Ketika sebuah *labeled packet* melewati sebuah LSR, nilai TTL-nya akan dikurangi 1.

2.2.2 MPLS *Virtual Private Network* (MPLS VPN)

2.2.2.1 Pendahuluan

Menurut Cisco System Learning(2006), Teknologi MPLS sudah banyak diadopsi oleh para *Internet service provider* (ISP) bersamaan dengan teknologi VPN untuk menghubungkan antar cabang perusahaan.

Di sini akan dijelaskan sedikit pondasi dan menunjukkan bagaimana cara untuk menyediakan layanan VPN ke pelanggan.

2.2.2.2 Kategori VPN

VPN pada umumnya digunakan oleh ISP untuk menggunakan infrastruktur fisik dalam mengimplementasikan *point-to-point link* antar cabang perusahaan. Jaringan pelanggan yang diimplementasi dengan VPN akan berada pada pengawasan pelanggan yang disebut dengan *customer sites* yang terhubung satu sama lain melalui jaringan ISP. Biaya pengimplementasian tergantung pada jumlah *site* yang akan dihubungkan. (De Ghein, 2007, P213)

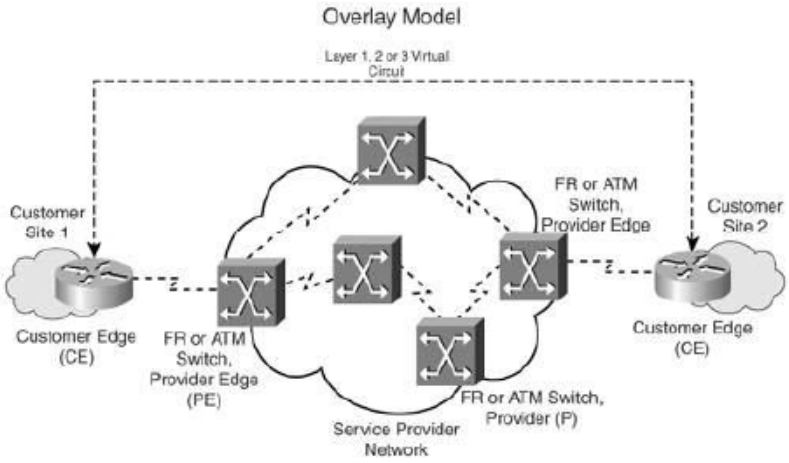
Frame Relay dan ATM merupakan teknologi pertama yang mengadopsi VPN. Pada umumnya, VPN terdiri dari 2 wilayah, yaitu:

1. Jaringan *customer*, terdiri dari *router-router* pada setiap *site* pelanggan yang disebut dengan *customer edge* (CE) router.
2. Jaringan *provider*, digunakan oleh ISP untuk menawarkan *dedicated point-to-point links* melalui jaringannya. *Router* yang terhubung langsung dengan CE disebut dengan *provider edge* (PE) *router*. Selain itu juga terdapat *router* pada jaringan *backbone*-nya yang disebut dengan *provider* (P) *router*.

   Berdasarkan partisipasi ISP terhadap *routing* di pelanggan, implementasi VPN dapat dibagi menjadi:

   1. *Overlay* VPN - Pada model ini *provider* menghubungkan antar cabang perusahaan dengan menggunakan jaringan

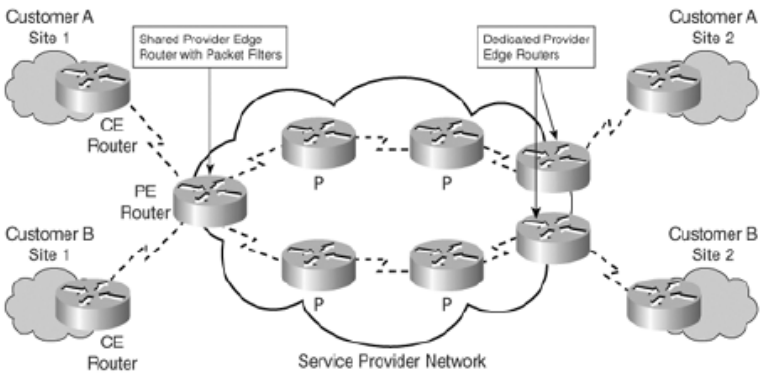
pribadi yang *emulated*, SP tidak mencampuri proses *routing* di sisi pelanggan. ISP hanya bertugas untuk menyediakan layanan data dengan menggunakan virtual *point-to-point link* yang dikenal dengan istilah *Layer 2 Virtual Circuit*.



Gambar 2.20 *Overlay* VPN

(Sumber: http://www.cisco.com/en/US/products/ps6557/prod_presentation_list.html)

2. Peer-to-Peer VPN – Dikembangkan untuk mengatasi kelemahan pada model *Overlay* dan mengoptimalkan transportasi data melewati jaringan *backbone* ISP. Oleh karena itu, ISP juga ikut aktif dalam proses *routing* di sisi pelanggan.



Gambar 2.21 *Peer-to-Peer* VPN

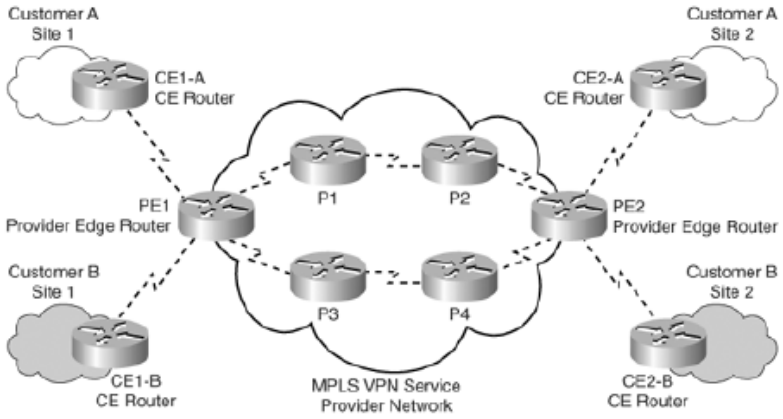
(Sumber:

http://www.cisco.com/en/US/products/ps6557/prod_presentation_list.html)

2.2.2.3 Arsitektur dan Terminologi MPLS VPN

Menurut Cisco System Learning(2006), Pada arsitektur MPLS VPN, *edge router* membawa informasi *routing* pelanggan dan mengoptimalkan proses *routing* pada pelanggan, sedangkan data diteruskan ke cabang-cabang melalui jaringan *backbone* ISP yang berbasis MPLS. Model MPLS VPN juga mencegah pengalamatan yang tumpang-tindih atau *overlapping*.

Domain jaringan MPLS VPN, seperti jaringan VPN tradisional, terdiri dari jaringan pelanggan dan *provider*. Model jaringan MPLS VPN mirip dengan model *peer-to-peer* VPN. Bagaimanapun juga, trafik pelanggan terisolasi pada router PE yang sama yang menyediakan konektivitas ke dalam jaringan ISP bagi banyak pelanggan. Komponen-komponen dari jaringan MPLS VPN dapat dilihat pada gambar 2.23.



Gambar 2.22 Arsitektur Jaringan MPLS VPN

(Sumber:

http://www.cisco.com/en/US/products/ps6557/prod_presentation_list.html)

Komponen-komponen utama arsitektur MPLS VPN adalah :

1. Jaringan pelanggan, biasanya merupakan wilayah kekuasaan pelanggan. Jaringan pelanggan untuk Customer A adalah CEl-A dan CE2-A bersama dengan perangkat - perangkat yang terdapat pada sisi 1 dan 2 Customer A.
2. *Router* CE, merupakan *router* yang terdapat pada jaringan pelanggan yang terhubung langsung dengan jaringan ISP. Pada gambar 2.22, *router-router* CE Customer A adalah CEl-A dan CE2-A, dan *router-router* CE Customer B adalah CE1-B dan CE2-B.
3. Jaringan *provider*, merupakan wilayah kekuasaan *provider* yang terdiri dari *router-router* PE dan P. Jaringan ini mengontrol *routing traffic* antarsisi pelanggan. Pada gambar

2.22, jaringan *provider* terdiri dari *router-router* PE1, PE2, PI, P2, P3, dan P4.

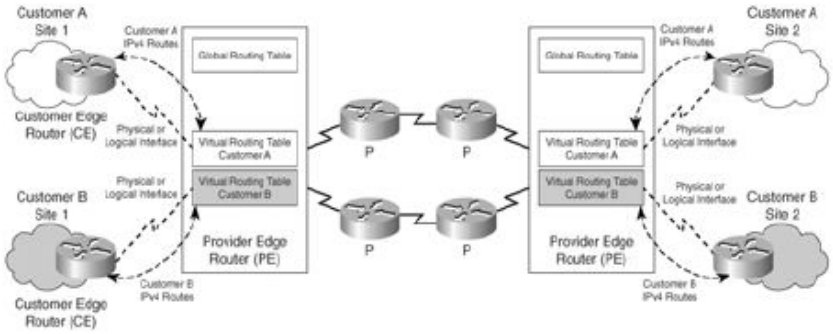
4. *Router* PE, merupakan *router* yang terdapat pada jaringan *provider* yang terhubung langsung ke *router* CE. Pada gambar 2.22, PE1 dan PE2 adalah *router* PE.
5. *Router* P, merupakan *router* yang terdapat pada jaringan *backbone* ISP yang terhubung langsung baik dengan *router* PE maupun *router* P. Pada gambar 2.23, *router* P1, P2, P3, dan P4 adalah *router* P.

2.2.2.4 Model *Routing* Pada Jaringan MPLS VPN

Menurut Cisco System Learning(2006), Implementasi dari MPLS VPN sangatlah mirip dengan implementasi model *peer-to-peer router dedicated*. Dari sisi *router* CE, hanya update IPv4 dan data, yang diteruskan ke router PE. Router CE tidak perlu dikonfigurasi sebagai *router* yang MPLS-*enabled* untuk menjadi bagian dari *domain* MPLS VPN. Yang diperlukan *router* CE hanyalah *routing protocol* yang memungkinkannya untuk menukar informasi *routing* IPv4 dengan *router* PE.

Pada implementasi MPLS VPN, *router* PE mempunyai banyak fungsu. Pertama, *router* PE harus bisa mengisolasi trafik pelanggan jika terdapat lebih dari satu pelanggan yang terhubung ke *router* PE. Oleh karena itu, setiap pelanggan diberi *routing table* independen yang mirip dengan *router* PE. *Routing* bisa melewati jaringan *backbone* ISP karena menggunakan proses *routing* yang terdapat pada *global routing table*.

Router-router P menyediakan *label switching* antara *router-router* PE dan tidak menyadari adanya rute-rute VPN. *Router-router* CE pada jaringan pelanggan tidak peduli dengan *router* P dan, oleh sebab itu, topologi bagian dalam jaringan ISP adalah tidak terlihat bagi pelanggan.



Gambar 2.23 Fungsionalitas Router PE

(Sumber:

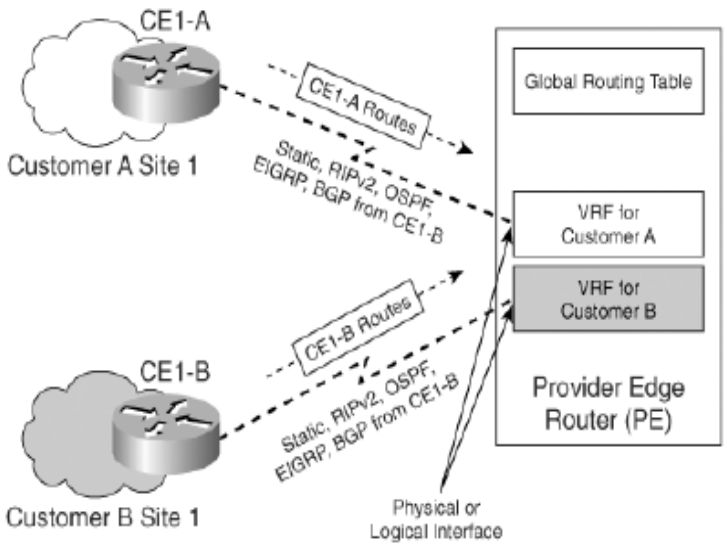
http://www.cisco.com/en/US/products/ps6557/prod_presentation_list.html)

Router-router PE hanya bertugas dalam *label switching* paket. Mereka tidak membawa rute-rute VPN dan tidak ikut serta dalam *routing* MPLS VPN. *Router-router* PE menukar rute-rute IPv4 dengan *router-router* CE menggunakan konteks *individual routing protocol*. Untuk memungkinkan jaringan melayani banyak VPN pelanggan, *multiprotocol* BGP (MP-BGP) harus dikonfigurasi pada *router-router* PE untuk membawa rute-rute pelanggan.

2.2.2.5 *Virtual Routing and Forwarding* (VRF)

Menurut Cisco System Learning(2006), Pengisolasian pelanggan dilakukan oleh *router* PE dengan menggunakan label *Virtual Routing and Forwarding* (VRF). Pada intinya, ini sama dengan menggunakan beberapa *router* untuk menangani pelanggan-pelanggan yang terhubung ke jaringan *provider*. Fungsi dari tabel VRF mirip dengan *label* routing global, kecuali bahwa tabel VRF berisi semua rute yang menuju ke VPN khusus. Jumlah dari VRF terbatas oleh jumlah *interface* yang terdapat pada suatu router, dan sebuah *interface* tunggal (logika maupun fisik) hanya bisa diasosiasikan dengan sebuah VRF. *Interface* yang akan diasosiasikan dengan VRF harus bisa mendukung *Cisco Express Forwarding* (CEF).

VRF berisi tabel *routing* IP sama dengan tabel *routing* IP global, sebuah tabel CEF, daftar *interface-interface* yang merupakan bagian dari VRF, dan sejumlah peraturan yang membatasi pertukaran *routing protocol* pada *router-router* CE.



Gambar 2.24 Implementasi VRF Pada Router PE

(Sumber:

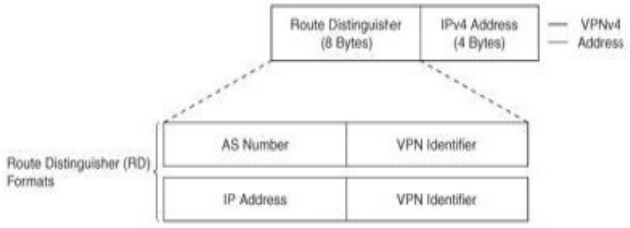
http://www.cisco.com/en/US/products/ps6557/prod_presentation_list.html)

2.2.2.6 *Route Distinguisher* (RD)

Menurut Cisco System Learning(2006), *Route Distinguisher* (RD) berfungsi untuk memungkinkan memindahkan data antar kedua sisi pelanggan melewati jaringan *backbone* ISP.

Format RD adalah *64-bit unique identifier* yang digabungkan dengan 32-bit *customer prefix* atau route yang diperoleh dari router CE, yang membentuk 96-bit address yang bisa dibawa melewati *router-router* PE pada *domain* MPLS. Oleh karena itu, sebuah RD yang unik dikonfigurasi untuk setiap VRF pada *router* PE. Pengalamatan yang dibentuk oleh 96-bit tersebut disebut dengan *VPN version 4 (*VPNv4*) address*.

Pengalamatan VPNv4 ditukarkan di antara *router-router* PE pada jaringan ISP digabung dengan pengalamatan IPv4. Jika ISP tidak memiliki nomor AS BGP, format pengalamatan IPv4 bisa digunakan, dan jika jaringan ISP memiliki nomor AS, format dari nomor AS bisa digunakan.



Gambar 2.25 *Route Distinguisher*

(Sumber:

http://www.cisco.com/en/US/products/ps6557/prod_presentation_list.html)

2.2.2.7 *Multiprotocol* BGP (MP-BGP)

Menurut Cisco System Learning(2006), Protokol yang digunakan untuk menukar rute-rute VPNv4 adalah *multiprotocol BGP (*MP-BGP*)*. *Router-router* PE harus menjalankan protokol *routing* IGP, yang pada saat ini Cisco mendukung OSPFv2 dan IS-IS pada jaringan MPLS ISP. MP-BGP juga bertugas untuk memberi label VPN, serta memungkinkan penggunaan pengalamatan VPNv4 pada lingkungan *router* MPLS VPN yang memungkinkan *overlapping* pengalamatan dengan beberapa pelanggan.

2.2.2.8 *Route Targets* (RT)

Menurut Cisco System Learning(2006), *Route Targets* (RT) merupakan pengenal tambahan yang digunakan pada *domain* MPLS VPN yang mengidentifikasikan keanggotaan VPN dari rute-rute yang dipelajari pada sisi tersebut. RT diimplementasikan dengan cara meng-*encoding* 16-bit urutan teratas dari BGP *extended community* (total 64-bit) dengan sebuah nilai yang berhubungan dengan keanggotaan VPN pada sisi tertentu. Ketika sebuah rute VPN yang dipelajari dari sebuah *router* CE disuntikkan ke BGP VPNv4, sebuah daftar atribut-atribut *route target extended community* akan diasosiasikan dengannya. *Export route target* digunakan sebagai identifikasi dari keanggotaan VPN dan diasosiasikan ke setiap VRF. *Import route target* diasosiasikan dengan setiap VRF dan mengidentifikasi rute-rute VPNv4 yang akan diimpor ke VRF untuk pelanggan tertentu. Format dari RT mirip dengan format RD. Interaksi antara nilai-nilai RT dan RD pada *domain* MPLS VPN sebagai *update* diterjemahkan sebagai sebuah *update* MP-BGP.

2.2.2.9 *Address Family* (AF)

Sebuah *Address Familv* (AF) adalah protokol *Network Layer* yang terbatas. Sebuah *Address Family Identifier* (AFI) membawa sebuah identitas dari protokol *Network Layer* yang berhubungan dengan pengalamatan jaringan pada atribut-atribut *multiprotocol* di BGP.

2.2.3 *Traffic Engineering* (TE)

2.2.3.1 Pendahuluan

Ketika berbicara tentang pertumbuhan dan pengembangan jaringan, terdapat dua teknik yang dapat dilakukan, yaitu *network engineering* dan *traffic engineering.*

Network engineering adalah proses memanipulasi jaringan yang kita miliki agar sesuai dengan trafik yang ada. Kita membuat perkiraan akan trafik yang lewat pada jaringan kita, lalu kita menambahkan jalur baru yang sesuai maupun peralatan jaringan seperti *router, switch* dan yang lainnya. *Network Engineering* biasanya selesai dalam jangka waktu yang lama karena waktu untuk instalasi jalur yang baru maupun instalasi peralatan jaringan. (Eric Osborne dan Ajay Simha,2002)

Traffic engineering adalah proses memanipulasi trafik agar sesuai dengan jaringan yang kita miliki. Tidak peduli seberapa keras kita berusaha, trafik jaringan tidak pernah akan sama dengan perkiraan kita. Terkadang suatu trafik meningkat melebihi prediksi sedangkan kita tidak dapat melakukan *upgrade* agar jaringan kita menjadi lebih cepat. Selain itu, akan terjadi kemacetan pada jalur utama (*best path*) sehingga menyebabkan jalur lain tidak digunakan. (Eric Osborne dan Ajay Simha,2002)

Traffic engineering diciptakan bukan hanya untuk teknologi MPLS, namun sudah terlebih dahulu ada pada teknologi ATM. Hal sederhana seperti mengubah *metric* pada sebuah *routing protocol* juga dapat disebut sebagai *traffic engineering. Traffic engineering* dengan MPLS dapat sama efektifnya seperti ATM, namun tanpa terjadi kekurangan seperti pada *IP over ATM.*

2.2.3.2 *Traffic Engineering* sebelum MPLS

IP traffic engineering populer namun sedikit kasar, cara untuk mengontrol jalur yang dilewati oleh IP melalui jaringan kita dengan cara merubah *cost* di suatu jalur. Karena tidak ada cara untuk mengatur jalur mana yang diambil oleh suatu trafik berdasarkan dari arah datangnya trafik, namun hanya ada dari arah ke mana trafik tersebut pergi.

ATM di lain sisi, mengizinkan kita untuk membuat PVC yang melewati jaringan dari sumber trafik ke tujuan. Hal tersebut berarti kita memiliki hak dalam mengatur trafik yang lebih baik Beberapa ISP besar menggunakan ATM untuk mengatur trafik pada jaringan mereka. Mereka melakukannya dengan membentuk ATM PVC yang *full mesh* antar *router* dan secara berkala mengubah dan mengatur PVC tersebut berdasarkan pengamatan trafik dari *router – router* mereka. Namun masalah yang muncul pada *router* yang membentuk *full-mesh* akan terjadi $O(N^2)$ *flooding* dan ketika sebuah link mati akan menyebabkan

$O(N^3)$ *flooding* yang menyebabkan masalah di beberapa jaringan berskala besar.

2.2.3.3 *Traffic Engineering* dengan MPLS

Tiga contoh penerapan MPLS-TE di kehidupan nyata adalah :

- Mengoptimalkan penggunaan dari jaringan kita.
- Menangani kemacetan trafik yang tidak diperkirakan sebelumnya.
- Menangani jalur dan *node* yang rusak.

Mengoptimalkan penggunaan jaringan dapat kita lakukan dengan membuat *full-mesh* dari MPLS TE-LSP diantara *router – router* yang ada, lalu memutuskan jumlah bandwidth yang akan digunakan diantara sepasang router, Kemudian biarkan LSP tersebut mencari jalur terbaik berdasarkan jumlah *bandwidth* yang mereka butuhkan. Dengan membuat TE-LSP menjadi *full-mesh* kita telah memanfaatkan dengan baik infrastruktur yang kita miliki, sehingga dapat menunda pembuatan jalur baru untuk beberapa saat yang tentunya dapat menghemat pengeluaran.

Pendekatan lain dalam membangun MPLS-TE adalah untuk menangani kemacetan yang tidak diperkirakan sebelumnya. Daripada membangun sebuah topologi *full-mesh* LSP antar *router*, lebih baik kita membiarkan IGP untuk meneruskan trafik sesuai keinginan IGP dan

membuat TE-LSP setelah kemacetan terjadi. Dengan begitu, kita tetap membiarkan jaringan kita hanya terdiri dari IGP *routing*, karena IGP *routing* lebih sederhana bila dibandingkan dengan *full-mesh* TE-LSP. Bila terjadi peningkatan trafik yang dapat menimbulkan kemacetan di suatu jalur dan jalur yang lain kosong, kita dapat membangun *tunnel* MPLS–TE untuk memindahkan trafik dari jalur yang macet ke jalur yang kosong yang mana IGP tidak memilih jalur kosong tersebut.

Fungsi ketiga dari MPLS-TE adalah untuk *quick recovery* bila terjadi kerusakan jalur dan *node*. MPLS-TE memiliki komponen yang disebut dengan *Fast Reroute* (FRR) yang berfungsi untuk mengurangi *packet loss* secara drastis apabila sebuah jalur atau *node* rusak.

2.2.3.4 Cara Kerja *Traffic Engineering*

Cara kerja dari *traffic engineering* terbagi menjadi tiga tahapan :

- *Information distribution*

  MPLS TE memungkinkan *router* untuk membangun jalur dengan menggunakan informasi selain jalur terpendek, yaitu dengan menggunakan informasi yang didistribusikan sehingga *router* dapat lebih pintar dalam melakukan kalkulasi jalur.

  MPLS TE menggunakan OSPF atau IS-IS untuk mendistribusikan informasi mengenai *resource* yang tersedia di

jaringan. Tiap informasi tersebut akan didistribusikan dalam bentuk *per-interface*. Tiga hal penting yang didistribusikan adalah

- Ketersediaan *bandwidth* per interface
- *Attribute flag* per interface
- *Administrative weight* per interface

Ketiga hal tersebut akan didistribusikan dalam keadaan sebagai berikut :

- Ketika suatu jalur *up* atau *down*
- Ketika ada konfigurasi yang berubah
- Ketika secara periodik IGP menyebarkan informasi
- Ketika *bandwidth* berubah secara signifikan

- *Path calculation and setup*

  Ketika kita melakukan penentuan jalur dalam pembuatan *tunnel*, kita dapat menggunakan dua cara yaitu, eksplisit dan dinamis. Dengan cara eksplisit, kita harus mendefinisikan arah jalur dari *tunnel* yang akan kita buat untuk dilewatkan trafik data. Sedangkan bila dilakukan dengan cara dinamis, maka jalur yang akan digunakan oleh suatu *tunnel* akan dihitung terlebih dahulu oleh *head-end router*. *Head-end router* tersebut akan melihat *database* dari MPLS TE yang dipelajari dari *routing protocol* seperti OSPF atau IS-IS. Proses dalam IOS Cisco yang berperan dalam perhitungan dari jalur TE disebut PCALC.

- *Forwarding traffic down a tunnel*

  Terdapat tiga metode untuk mengalirkan trafik melalui suatu *tunnel*. Ketiga metode itu adalah :

  - *Static route*
  - *Policy routing*
  - *Autoroute*

2.2.4 *Diffserv-Aware Traffic Engineering* (DS-TE)

2.2.4.1 Pengenalan *Quality of Services* (QoS)

Dalam suatu jaringan harus menyediakan keamanan, dapat diramalkan, terukur dan harus terjamin layanannya. Seorang admin dan perancang jaringan dapat meningkatkan performa dari suatu jaringan apabila ia dapat mengatur *delay*, variasi dari *delay* (*jitter*), ketersediaan *bandwidth* dan parameter *packet loss* dengan teknik *quality of service* (QoS). (Eric Osborne dan Ajay Simha,2002)

Terdapat dua arsitektur QoS yang digunakan saat ini :

- *Integrated Services* (*IntServ*)
- *Differentiated Services* (*Diffserv*)

IntServ dapat menyediakan QoS untuk paket IP. Suatu aplikasi mengirimkan sinyal ke jaringan bahwa mereka memerlukan QoS dalam pengiriman paket lalu kemudian bandwidth di pesan untuk aplikasi

tersebut, akan tetapi *IntServ* tidak dirancang untuk jaringan berskala besar, sehingga *IntServ* hanya cocok bagi jaringan berukuran kecil – menengah. Sedangkan *Diffserv* menyediakan skalabilitas dan fleksibilitas dalam implementasi QoS di suatu jaringan, sehingga *Diffserv* dapat digunakan pada jaringan berskala besar seperti *Internet Service Provider*. Perangkat jaringan mengetahui pembagian kelas trafik dan menyediakan QoS yang berbeda untuk kelas trafik yang berbeda (Eric Osborne dan Ajay Simha,2002)

2.2.4.2 Arsitektur *Diffserv*

Diffserv mempunyai dua komponen utama :

- *Traffic conditioning* **–** terdiri dari *classification, policing, marking dan shaping*. Hal tersebut hanya dilakukan di *edge router*.
- *Per – hop behavior* **–** terdiri dari *queuing, scheduling, dan mekanisme dropping*. Hal tersebut dilakukan di setiap hop.

Cisco IOS menyediakan banyak tools untuk mengaplikasikan komponen – komponen *Diffserv* diatas. Kita dapat melakukannya dengan cara lama seperti metode *per-platform* atau cara yang lebih baru *Modular QoS CLI* (MQC). Pada skripsi ini, akan digunakan metode MQC.

Berikut adalah penjelasan dari arsitektur *Diffserv* :

- *Classification*

Tahap pertama dalam mengaplikasikan arsitektur *Diffserv* adalah dengan cara mengklasifikasi paket. *Classification* adalah proses untuk pengurutan paket – paket, sehingga setelah diurutkan akan didapat trafik yang berbeda.

- *Classifying IP packet*

  Pengklasifikasian paket IP dilakukan secara langsung, yaitu dengan mencocokan dengan yang ada di IP header, seperti *source* IP, *destination* IP dan nilai DSCP.

- *Classifying MPLS packet*

  Pengklasifikasian paket MPLS dilakukan dengan mencocokan dengan nilai EXP dari *label stack* terluar.

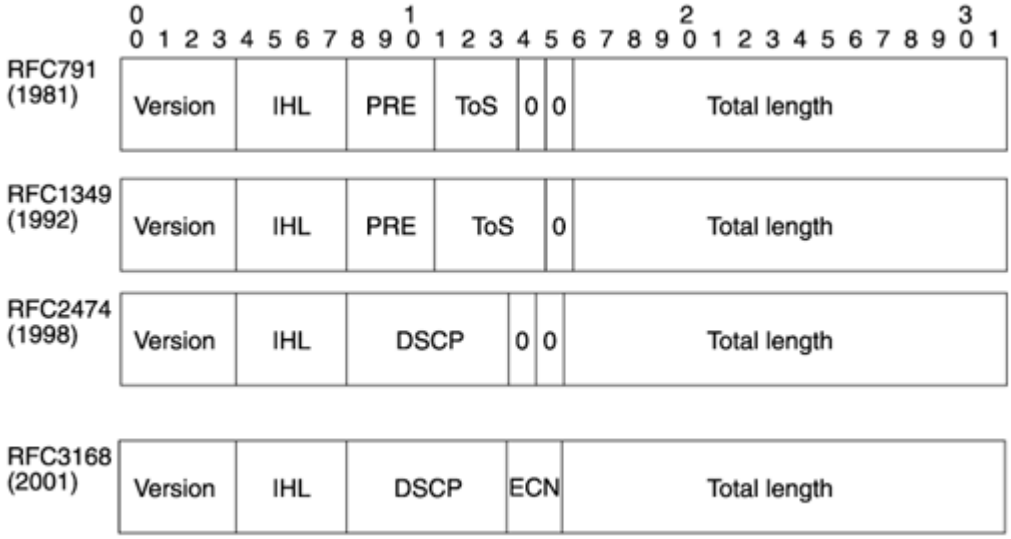
- *Policing*

  Policing berfungsi untuk memeriksa apakah suatu trafik sudah sesuai dengan ketentuan yang telah disetujui sebelumnya dan mengijinkan untuk membuang trafik tersebut bila melanggar ketentuan atau melakukan *marking* kembali dengan nilai DSCP yang baru. Dalam proses *policy* tidak dilakukan proses *buffering* sehingga tidak berdampak pada *delay*. *Policing* dilakukan di *edge network.*

- *Marking*

  Marking pada QoS telah berevolusi dari waktu ke waktu. Di dalam *header* IP terdapat sebuah *byte* yang disebut *type of service*

(ToS) *byte*. 8 bit pada *byte* tersebut dengan seiring waktu terus mengalami evolusi.



Gambar 2.26 Evolusi dari *header* IP

(Sumber: Traffic Engineering with MPLS : Quality of Service with MPLS TE)

Pada awalnya, *header* IP memiliki 3 bit *precedence* dan 3 bit ToS, dan 2 bit yang tidak digunakan. Bit *precedence* digunakan untuk membuat keputusan mengenai perlakuan terhadap suatu paket. Nilai *precedence* 0 – 5 digunakan untuk data dari pelanggan. Nilai *precedence 6-7* di *reserved* untuk mengatur trafik jaringan. Pada RFC 1349, 1 bit yang berada pada *unused* bit diberikan pada ToS bit, sehingga didalam *header* IP menjadi 3 *precedence* bit, 4 ToS bit, dan 1 *unused* bit.

ToS bit tidak pernah dikembangkan dengan baik. Tujuan awal dari ToS bit adalah dapat melakukan *marking* terhadap paket yang memiliki ciri, *low delay*, *high throughput*, atau *high-reliability path*, akan tetapi layanan arsitekturnya tidak pernah dirancang atau dibangun untuk nilai ToS bit.

RFC 2474 dan 2475 mendefinisikan ulang keseluruhan ToS *byte*. ToS *byte* sekarang berisi 6 bit yang berisi informasi DSCP bit. Sisa dua bit dari ToS *byte* digunakan untuk mekanisme TCP yang disebut dengan *Explicit Congestion Notification* (ECN), yang didefinisikan pada RFC 3168.

Ketika berbicara mengenai QoS dan ToS *byte*, beberapa orang menggunakan istilah IP *Precedence* sedangkan yang lain menggunakan istilah Diffserv. Mapping antara DSCP bit dan IP Precedencce bit dapat dilihat pada tabel berikut :

Tabel 2.3 *Mapping* bit DSCP ke IP *Precedence*

IP Precedence (*Decimal*)	**IP Precedence (*Bit*)**	**DSCP (*Decimal*)**	**DSCP (*Bit*)**
0	000	0	000000
1	001	8	001000
2	010	16	010000
3	011	24	011000
4	100	32	100000

5	101	40	101000
6	110	48	110000
7	111	56	111000

Untuk mengubah nilai IP *Precedence* menjadi nilai DSCP hanya dengan mengkalikan nilai IP *Precedence* dengan 8. Kedelapan nilai IP Precedence disebut *classes*, dan nilai DSCP bit yang memetakan nilai IP Precedence disebut sebagai *Class Selector Code Point* (CSCP), terkadang disingkat menjadi CS.

Sebagai tambahan untuk delapan class selector, pada RFC 2579 dan 2598 ditambahkan 13 nilai DSCP tambahan, yaitu 12 nilai Assured Forwarding (AF) dan sebuah nilai Expedited Forwarding (EF)

Tabel 2.4 Tambahan nilai DSCP pada RFC 2597 dan 2598

Nama	**DSCP (*Decimal)***	**DSCP (*Bit*)**
Default	0	000000
AF11	10	001010
AF12	12	001100
AF13	14	001110
AF21	18	010010
AF22	20	010100
AF23	22	010110
AF31	26	011010

AF32	28	011100
AF33	30	011110
AF41	34	100010
AF42	36	100100
AF43	38	100110
EF	46	101110

Terdapat 12 nilai AF, semuanya dalam format AF*xy*, dimana nilai *x* adalah nomor *class* dan *y* adalah *drop precedence*. Terdapat empat kelas (AF1*y* – AF4*y*) masing – masing memiliki tiga *drop precedence* (AF*x*1 – AF*x3*). AF adalah metode untuk menyediakan *low packet loss* dengan *traffic rate* yang diberikan, tetapi tidak menjamin *latency*.

EF adalah perilaku yang didefinisikan untuk meminta *low-delay, low-jitter, low-loss service*. EF biasanya diimplementasikan menggunakan LLQ. EF hanya didefinisikan dalam satu kelas, karena bila terdapat lebih dari satu kelas, kedua kelas tersebut akan berebut *resource* yang sama. (Eric Osborne dan Ajay Simha,2002)

- *Queuing*

Queuing atau antrian adalah sebuah proses pengurutan paket yang terkait dengan *output buffers*. *Queuing* hanya bekerja pada

interface yang mengalami *congestion* dan apabila *congestion* tidak terjadi maka *queuing* juga aktif.

Banyak teknik *queuing* dapat diaplikasikan pada jaringan MPLS, bergantung *platform* dan versi dari perangkat jaringan :

- *First In First Out* (FIFO)

  FIFO berada di setiap platform dan setiap *interface* dan secara *default* berada di semua *interface*.

- *Modified Deficit Round Robin* (MDDR) (hanya untuk *platform* GSR)

- *Class-based Weighted Fair Queuing* (CBWFQ) (umumnya untuk *platform* non-GSR)

- *Low-Latency Queuing* (LLQ)

  MDRR, CBWFQ, dan LLQ dikonfigurasi dengan MQC. Tinggal mencocokan MPLS EXP dalam *class-map* dan lakukan konfigurasi atau jaminan *latency* dengan perintah *bandwidth* atau *priority.*

- *Dropping*

  Merupakan salah satu bagian *Diffserv* PHB. *Dropping* sangatlah penting, yaitu untuk membuang paket – paket berdasarkan

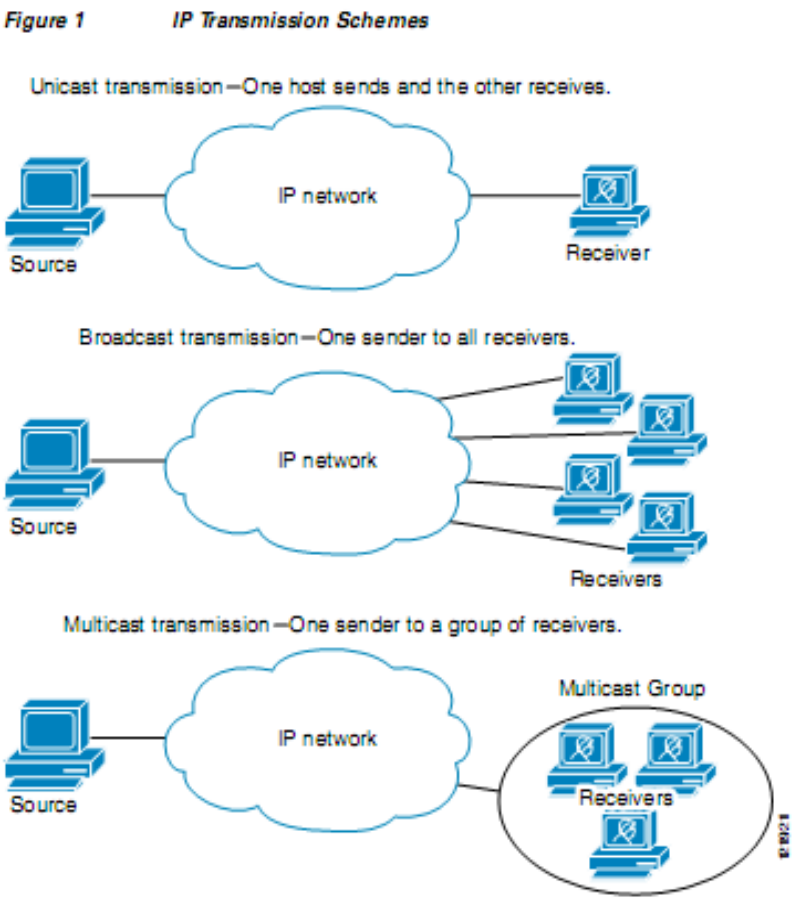
antrian paket – paket yang telah mencapai 100% dari panjang antrian maksimal.

Manajemen terhadap *queuing* FIFO menggunakan kebijakan *tail-drop*, dimana akan melakukan *dropping* terhadap setiap paket yang datang ketika antrian sedang penuh.

Weighted Random Early Detection (WRED) adalah mekanisme *Diffserv* yang diimplementasikan hampir di semua *platform* Cisco. WRED bekerja pada MPLS EXP sama seperti *IP Precedence*.

2.2.5 *Multicast*

Multicast adalah sebuah teknik dimana sebuah data dikirimkan melalui jaringan ke sekumpulan komputer yang tergabung ke dalam sebuah grup tertentu yang disebut sebagai *multicast group*. Alamat IP *multicast* terdapat dalam kelompok IP kelas D, yang mempunyai jangkauan alamat IP dari 224.0.0.0/4 sampai dengan 239.255.255.255 Penerapan *multicast* mempunyai beberap protokol yang juga sudah ditentukan oleh IANA (*internet Assigned Numbers Authority*) yang disebut sebagai *well-known address.*



Gambar 2.27 Konsep *Multicast*

(Sumber: http//cnap.binus.ac.id/)

2.2.5.1 Protokol IP *multicast*

IP *multicast* adalah metode pengiriman IP kepada penerima yang tergabung dalam suatu grup yang dilakukan dalam sekali pengiriman. IP *multicast* adalah teknik pengiriman data *one-to-many* dan *many-to-many*. Hal ini berarti pengiriman IP *multicast* dapat dilakukan dari satu pengirim ke banyak penerima dan dari banyak pengirim ke banyak

penerima. *Multicast* menggunakkan infrastruktur jaringan secara efisien dengan hanya membutuhkan pengirim atau sumber untuk mengirimkan paket data dalam satu kali pengiriman saja, walaupun jaringan tersebut membutuhkan pengiriman kepada jumlah penerima yang besar. Node yang berada dalam jaringan yaitu *switch* dan *router*, mengatur penduplikasian paket data untuk dapat mencapaikan paket ke banyak penerima.

Protokol tingkat bawah yang paling umum digunakkan adalah *User Datagram Protocol* (UDP). Berdasarkan karakteristiknya, UDP masih terdapat kekurangan. Karena UDP belum sekompleks protokol-protokol pengiriman data *multicast* lainnya, maka data yang dikirimkan oleh UDP dapat hilang atau rusak. Ada pula jenis-jenis dari ptotokol IP *multicast* adalah :

- *Internet Group Management Protocol (IGMP)*
- *Protocol Independent Multicast (PIM)*
- *Distance Vector Multicast Routing Protocol (DVMRP)*
- *Multicast Open Shortest Path First (MOSPF)*
- *Multicast BGP (MBGP)*
- *Multicast Source Discovery Protocol (MSDP)*
- *Multicast Listener Discovery (MLD)*
- *GARP Multicast Registration Protocol (GMRP)*
- *Multicast DNS (mDNS)*

Pada skripsi ini digunakkan *Protocol Independent Multicast (PIM)* dan IGMP. *PIM* adalah kumpulan *routing protocol multicast,* yang masing-masing digunakkan dalam situasi dan kondisi yang berbeda. Ada empat jenis protokol PIM yaitu, *Sparse Mode* (SM), *Dense Mode* (DM), *Sparse Dense Mode* (SDM) dan *Bidirectional* (Bidir). Berikut ini adalah penjelasan tentang *routing protocol* pada PIM

- **Sparse Mode (SM)**

  PIM-SM menggunakan model *join* dimana paket *multicast* hanya akan diteruskan ke suatu *interface* jika *host* yang hendak menerima telah bergabung dalam grup atau terdapat permintaan terhadap paket tersebut

  Dalam protokol ini terdapat titik pusat (*central point*) yang digunakan oleh seluruh sumber pengirim dalam mengirimkan paketnya. Setiap pengirim paket melakukan proses pengiriman dengan memilih jalur terbaik ke *central point*. Kemudian *central point* mendistribusikan paket tersebut keseluruh penerima yang tergabung dalam grup tujuan menggunakan jalur terbaik. Titik pusat ini disebut *Rendezvous Point* (RP). Dalam sebuah jaringan, bisa terdapat lebih dari satu RP, namun hanya ada satu RP untuk satu grup *multicast*.

- **Dense Mode (DM)**

PIM-DM menggunakan Model *Push* untuk mengirimkan paket *multicast* ke setiap "ujung" dari jaringan. Penerapan konfigurasi PIM-DM akan menjadi efisien jika dalam setiap subnet dalam jaringan terdapat anggota *multicast*.

Konsep PIM Dense Mode :

- ❖ Protokol PIM- DM akan mengirimkan paket *multicast* ke semua *interface* dalam jaringan, di mana proses ini disebut *flooding*.
- ❖ *Router – router* yang tidak memiliki anggota di *interface*-nya akan mengirimkan *prune*. Proses ini akan berulang setiap 3 menit.
- ❖ Mekanisme *flooding* dan *prune* ini akan digunakan *router* oleh *router* untuk membangun tabel *multicast forwarding* mereka.

- **Sparse Dense Mode (SDM)**

Pemilihan mode akan lebih efesien jika pemilihan mode tersebut dilakukan berdasarkan *per-group*, bukan *per-interface*. Kemampuan ini difasilitasi dengan adanya konfigurasi *sparse-dense mode*. Penerapan konfigurasi ini memungkinkan sebuah grup dapat mengikuti *sparse* dense *mode* bergantung pada eksistensi *rendezous point* dalam jaringan.

Jika suatu jaringan terdapat sebuah RP maka akan menggunakan *Sparse Mode* dan sebaliknya jika tidak memakai RP maka akan menggunakan *Dense Mode*

- **Bidirectional (Bidir)**

Bidirectional PIM (Bidir-PIM) merupakan penyempurnaan dari protokol PIM yang dirancang untuk komunikasi yang efektif *many-to many* dalam satu domain PIM tunggal. Kelompok *multicast* dalam mode *bidirectional* dapat berkembang dengan jumalah yang semaunya di dalam *source* dengan jumlah yang minimal di *aditional overhead.*