BAB II
DASAR TEORI

2.1 *Multi Protocol Label Switching* (MPLS)

2.1.1 Pengertian *Multi Protocol Label Switching* (MPLS)[1]

Multi Label Protocol Switching adalah sebuah teknologi untuk IP-*Routing* yang memiliki kecepatan tinggi. MPLS menyediakan layanan *connection oriented* yang berbasis IP *routing* dan *Control Protocols*. Teknologi ini baru diperkenalkan dan distandarisasi sebagai pendekatan untuk mengintegrasikan IP dengan layanan *Asynchronous Transfer Mode* (ATM) dan menggunakan metode yang telah direkomendasikan untuk IP melalui layanan ATM oleh ITU. MPLS memliki beberapa keunggulan antara lain :

- MPLS dapat digunakan untuk mengurangi banyaknya beban pengolahan trafik yang diolah yang terjadi dalam *router*, dikarenakan teknologi MPLS merupakan sebuah metode *forwarding* yang menggunakan teknik peningkatan pada sistem *forwarding* data pada koneksi tradisional dalam proses penyaluran paket data yang memiliki kapasitas yang besar, sehingga tingkat keefisienan pada teknologi MPLS dapat dikatakan lebih baik dan lebih tinggi.
- MPLS menyediakan mekanisme *Quality of Service* (QoS) pada jaringan.

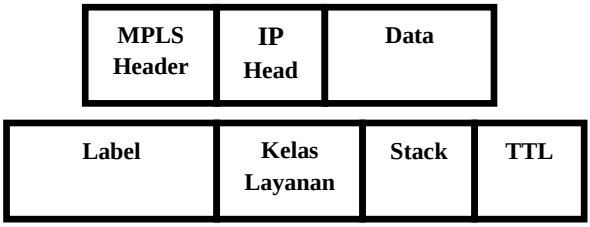
Multi Protocol Label Switching telah didesain sebagai teknologi pemercepat layanan IP *routing*, dan sekarang sesuai dengan perkembangan jaman, MPLS berkembang menjadi sebuah media yang sangat disukai dan sering diterapkan untuk menyediakan mekanisme QoS melalui IP. Banyak penyedia layanan dalam hal ini *Service Provider* yang sedang berusaha untuk menerapkan teknologi MPLS untuk menggantikan teknologi transportasi pengiriman paket yang lama, seperti SONET/SDH, ATM, dan lain sebagainya, dalam rangka untuk mencapai tingkat efisiensi yang lebih tinggi dan operasional yang lebih rendah sehingga dapat mengurangi biaya yang dikeluarkan[2].

MPLS memiliki jaringan koneksi virtual yang dibangun yang disebut *Label-Switched Path* (LSP), LSP merupakan koneksi antara *node* akhir *datagram* jaringan untuk menyediakan emulasi *connection-oriented*. Untuk membentuk LSP dibutuhkan suatu protokol pensinyalan, protokol ini menentukan *forwarding* paket berdasarkan label pada paket. Label yang pendek dan berukuran tetap mempercepat proses *forwarding* dan mempertinggi fleksibilitas pemilihan *path* pada saat pengiriman paket dari sumber menuju

tujuan. *Node* dalam jaringan MPLS disebut *Label-Switching Router*. Peran *router* MPLS dalam jaringan terbagi atas LSR *ingress* dan LSR *egress* yang akan mentranslasaikan suatu alamat IP tujuan menjadi label, melakukan pertukaran paket berdasarkan label oleh LSR *transit* dan kemudian LSR *egress* akan membuang label dan meneruskan paket ke *router* akhir pada sisi. Sebagai teknologi baru, MPLS mampu mengatasi permasalahan pada jaringan dan kekurangan *routing* IP tradisional dengan menggunakan penambahan sebuah label pada paket IP dan meneruskan paket tersebut berdasarkan pada label.

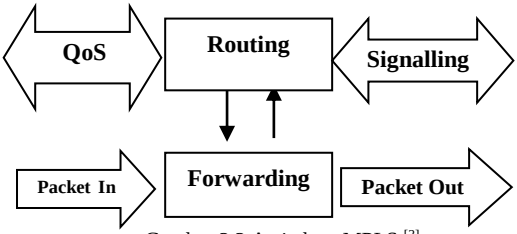
2.1.2 Arsitektur MPLS[3]

Jaringan MPLS terdiri dari sirkuit yang disebut *Label Switched Path*, *Label Switched Path* berfungsi untuk menghubungkan *node-node* yang disebut *Label Switched Roter*. Seluruh *Label Switched Path* terhubung dengan *Forwarding Equivalence Class* (FEC). Sesuai dengan gambar 2.1 yang merupakan kumpulan paket yang menerima perlakukan *forwarding* yang sama di sebuah *LSR*. *Header* MPLS terdiri dari 32 *bit* data, termasuk 20 *bit label*, 2 *bit* experimen dan 1 *bit* identifikasi *stack* serta 8 *bit* TTL [2].



Gambar 2.1 *Header* MPLS [3]

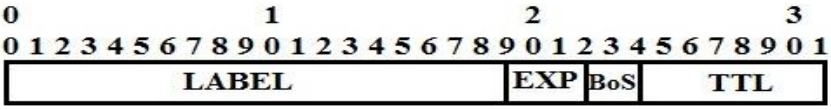
Jaringan MPLS terdiri atas sirkuit yang disebut *label-switched path* (LSP), yang menghubungkan titik – titik yang disebut *label-switching router* (LSR). LSR pertama disebut *ingres* sedangkan LSR terakhir disebut dengan *egress*. Untuk membentuk LSR, diperlukan suatu protokol pensinyalan. Protokol ini menentukan *forwarding* paket berdasarkan pada label yang ditambahkan pada paket dengan ukuran yang pendek dan berukuran tetap. Label ini berfungsi untuk mempercepat proses *forwarding* dan mempertinggi fleksibilitas pemilihan *path* pada pengiriman paket dalam jaringan. Hasilnya adalah jaringan datagram yang bersifat lebih *connection oriented* seperti yang ditampilkan pada gambar 2.2



Gambar 2.2 Arsitektur MPLS [3]

2.1.3 Label Struktur MPLS[1]

Struktur label MPLS memiliki struktur tertentu dengan data sebesar 32-*bits* seperti yang ditunjukkan pada gambar 2.3



Gambar 2.3 Label MPLS[1]

- *Label*: Merupakan field yang terdiri dari 20 *bit* pertama pada label MPLS berupa Label Nilai dan nilai 16 pertama yang dibebaskan untuk penggunaan normal dikarenakan untuk penggunaan khusus. Nilai label tersebut contohnya alamat IP, besar data, jenis data dan lain-lain. Sistem mempelajari hop berikutnya dan operasi yang akan dilakukan, setelah menerima paket berlabel dan nilai label di bagian atas.
- *Experimental Use* (*EXP*): *bit* ini dimulai dari 20-22 yang dicadangkan untuk penggunaan eksperimental, dan digunakan hanya untuk QoS atau dapat juga merupakan hasil salinan dari bit – bit IP *precedence* pada paket IP. Secara teknis field ini digunakan untuk keperluan eksperimen.
- *BOS*: *Bit* 23 dikenal sebagai *Bottom of Stack bit*. Field ini digunakan untuk mengetahui label stack yang paling bawah. *Label stack* paling bawah memiliki nilai bit 1 dan yang lain diberi nilai 0. Hal ini sangat diperlukan pada proses label *stacking*. *Stack* adalah kumpulan label dan dapat terdiri dari satu label atau beberapa label.

Label	EXP	0	TTL
Label	EXP	0	TTL
. . . . .			
Label	EXP	1	TTL

Gambar 2.4 *Label Stacking*[1]

- *TTL*: *bit ke* - 8 (24-31) memiliki fungsi yang sama seperti pada header IP. Field ini biasanya merupakan hasil salinan dari IP TTL *header*. Nilai *time-to-live* mengalami penurunan sebesar 1 pada setiap hop yang membantu dalam proses pendeteksian dan penghentian *looping* dari paket PLS atau biasa disebut paket *storm*.

2.1.4 Enkaplsulasi pada MPLS[4]

Label *stack* MPLS terletak diantara layer 2 dan layer 3, sehingga label *stack* di MPLS sering disebut Shim Header. Inilah alasan mengapa MPLS sering disebut protokol di layer 2.5 karena bisa berada di layer 3 maupun di layer 2. MPLS tidak dapat disebut protokol layer 2 karena proses enkapsulasinya telah menggunakan label, padahal layer 2 itu memiliki sifat memformat paket menggunakan France. Sedangkan MPLS tidak dapat dikatakan sebagai protokol layer 3 karena masih menggunakan label dalam prosesnya, padahal layer 3 itu identik dengan hubungan end to end.

Enkapsulasi pada layer 2 dapat berupa PPP, *High-Level Data Link Control* (HDLC), dan lainnya. Dapat diambil contoh protokol transport yang digunakan adalah IPv4, dan enkapsulasi yang digunakan adalah PPP, jadi *label stack* ditempatkan setelah PPP tetapi sebelum *header* IPv4. Oleh karena itu kita harus memberikan nilai pada layer 2 untuk menunjukkan enkapsulasi jenis apa yang digunakan.

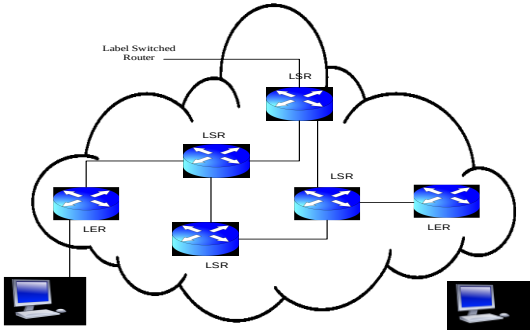
Tipe Enkapsulasi	Identitas Protokol Layer	Nilai
PPP	PPP Protocol Field	281
Ethernet/802.3	Ethertype Value	8847
HDLC	Protocol	8847
Frame Relay	NLPID (Network Level)	80

Tabel 2.1 Nilai masing – masing enkapsulasi layer 2[1]

2.1.5 *Label Switched Routers* (LSR)[1]

LSR adalah sebuah *router* yang memiliki kemampuan untuk memahami MPLS label dan bertanggung jawab untuk menerima dan mengirimkan paket label pada data link di jaringan MPLS MPLS ditampilkan pada gambar 2.5. Tiga operasi yang berhubungan dengan LSR adalah pop, push dan swap. Di Jaringan MPLS, ada tiga jenis LSR:

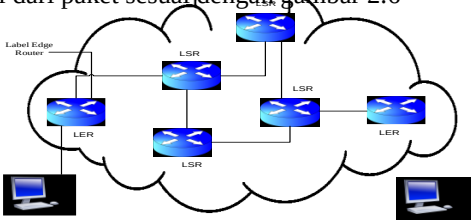
- LSR *Ingress*: menerima paket berlabel, menambahkan label ke paket yang digunakan dan mengirimkannya melalui data link.
- LSR *Egress*: menerima paket berlabel, menghapus label atau kumpulan label dan mengirimkannya melalui data link
- LSR *Intermediate*: melakukan operasi pada label paket masuk dan beralih paket pada link data yang benar.



Gambar 2.5 *Label Switched Routers* (LSR)[1]

2.1.6 *Label Edge Router*[1]

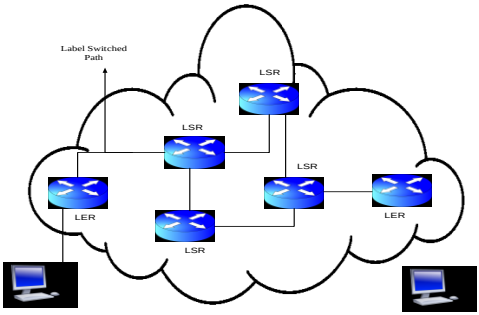
LERs bekerja sebagai penentu keputusan QoS pada jaringan MPLS. Dengan menggunakan nomor port pada lapisan-4 dari paket, kebijakan QoS dapat didirikan dan dikelola. LER bertanggung jawab untuk menambahkan atau menghapus label dari paket sesuai dengan gambar 2.6



Gambar 2.6 *Label Edge Routers* (LER)[1]

2.1.7 *Label Switched Paths* (LSP)[1]

LSP terdiri dari urutan LSR yang bertukar paket berlabel, melalui Jaringan MPLS. Dalam jaringan MPLS, LSR pertama dari LSP adalah LSR *ingress* untuk itu LSP, dan LSR terakhir dari LSP adalah LSR *egress*. LSR menengah adalah bekerja di antara masuknya dan LSR *egress* seperti yang ditampilkan pada gambar 2.6.



Gambar 2.7 *Label Switched Paths* (LSP)[1]

2.1.8 *Forward Equivalence Class* (FEC)[1]

Sekelompok paket yang memiliki jalur transmisi yang sama dan *forwarding* mekanisme dikenal sebagai FEC. Paket milik FEC yang sama memiliki label yang sama. Tapi beberapa paket tidak termasuk mekanisme FEC dan *forwarding* yang sama karena berbeda Nilai EXP. *Ingress* LSR memutuskan milik paket yang FEC dan ini dilakukan hanya sekali dalam MPLS *network*.

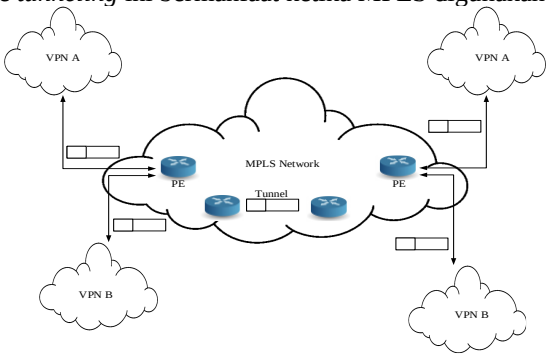
2.1.9 Cara Kerja Jaringan MPLS

MPLS memiliki rangkaian *node-node* yang bisa men-*switch* dan men-*route* berdasarkan label yang dipasang pada setiap paket. Domain MPLS terdiri dari serangkaian *node* MPLS yang saling terhubung satu sama lain. *Node* ini merupakan LSR. Label-labelnya menentukan aliran paket diantara kedua *endpoint* (titik akhir). Jalur khusus melalui jaringan LSR untuk setiap alirannya yang disebut FEC telah ditentukan. MPLS merupakan teknologi yang berorientasi sambungan. Setiap FEC memiliki karakterisasi lalu lintasnya yang menentukan persyaratan *QoS* untuk aliran tersebut. Karena LSR mengirim paket yang didasarkan pada nilai labelnya, maka proses pengirimannya lebih sederhana dari pada dengan *router* IP.

2.2 MPLS VPN

2.2.1. Pengertian MPLS VPN[6]

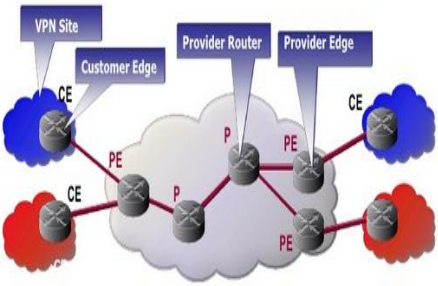
Teknologi MPLS dapat digunakan untuk VPN yang terdapat pada layer 2 dan layer 3 dari OSI layer. Teknologi *Frame Relay* dan ATM terdapat pada layer 2. IP *Tunneling* berbasis GRE atau *IPSec* terdapat pada layer 3 melalui jaringan IP. MPLS *Virtual Private Network* merupakan teknologi yang populer dan tersebar luas penggunaannya. Kepopularitasan dari teknologi MPLS VPN berkembang sejak teknologi tersebut ditemukan. Jaringan MPLS mampu memberikan dukungan untuk MPLS *tunnels*, yang digunakan untuk menetapkan VPN pada layer 2 pada *Frame Relay*, ATM dan lainnya. *Tunnels* ini mampu memberikan *virtual wire* yang disambungkan dari sumber ke tujuan pada VPN. Singkatnya enkapsulasi paket pada MPLS memberikan beberapa mekanisme *tunneling* yang digunakan sebagai transmisi paket melalui jaringan IP. Mekanisme *tunneling* ini bermanfaat ketika MPLS digunakan dengan VPN.



Gambar 2.8 MPLS *Tunnel* terhubung dengan beberapa VPN [6]

Dengan menggunakan MPLS VPN dapat membuat koneksi user tidak perlu dihubungkan *end-to-end* dengan jalur tersendiri, sehingga menghemat pembuatan link baru. *Overlapping* IP, proses *fast switching*, dan faktor keamanan data merupakan kelebihan dalam menggunakan MPLS VPN. Arsitektur MPLS VPN menyediakan kemampuan untuk menangani infrastrukut jaringan pribadi yang mengirimkan layanannya pada infrastruktur jaringan publik[6]. Dari sudut implementasi, pada jaringan MPLS VPN tidak perlu dilakukan konfigurasi pada setiap titik yang dilewati VPN, sehingga pengimplementasian lebih mudah[3] . MPLS *Tunnel* terhubung dengan beberapa VPN ditampilkan pada gambar 2.9.

2.2.2. Komponen MPLS VPN[7]



Gambar 2.9 Komponen MPLS VPN [7]

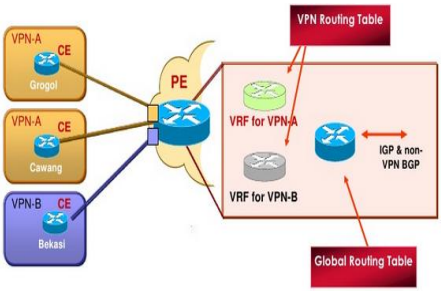
Berikut penjelasan dari gambar 2.9 :

- CE : *Customer Edge*, merupakan perangkat pelanggan yang secara langsung terhubung dengan *service provider*.
- PE : *Provider Edge*, merupakan perangkat yang berada di dalam jaringan *provider* yang terhubung dengan CE dan bertanggung jawab untuk memberikan akses layanan VPN
- P : *Provider*, merupakan perangkat yang berada di dalam jaringan *provider* yang tidak terhubung langsung dengan CE dan bertanggung jawab untuk fungsi *routing* dan *forwarding.*

2.2.3. Parameter MPLS VPN[7]

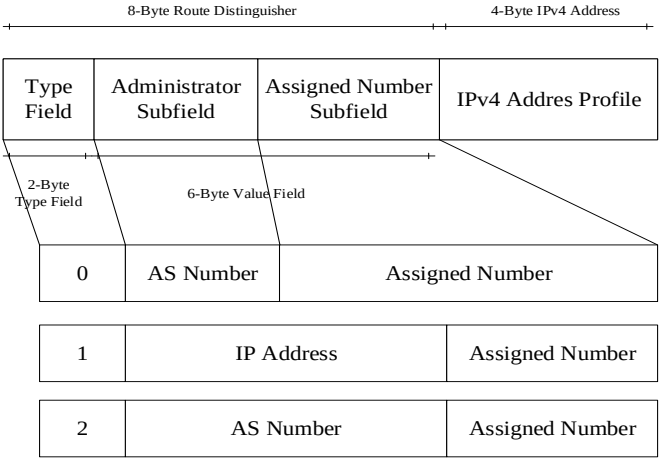
Terdapat beberapa parameter penting pada *router* PE yang berperan dalam membangun *layer*-3 VPN diantaranya :

1. VPN *Routing* dan *Forwarding Instance* (VRF)
   VRF merupakan suatu *virtual router*, setiap VPN membutuhkan VRF yang terpisah pada setiap *router* PE, VRF diasosiasikan dengan *interface/sub-interface* yang terhubung dengan CE. *Route* dalam VRF akan didistribusikan ke site yang lain (biasanya terhubung dengan PE lain) dari VPN yang sama.



Gambar 2.10 VPN *Routing* dan *Forwarding Instance*[7].

2. *Route Distinguisher* (RD)
   Route Distinguisher digunakan untuk merubah bentuk *non-unique* 32-*bit address* IPv4 *user* kedala 96-*bit* unik VPNv4 *address.* Seperti yang ditampilkan pada gambar 2.11. *Route Distinguisher* digunakan untuk identifikasi VRF *instances*, VRF *instances* yang berbeda, harus mempunyai RD yang berbeda, *Route Distinguisher* dikonfigurasi pada *router* PE untuk setiap VRF. Pada RD 8 byte = 64 bit, *Administrator field* berisi *autonomous system number (ASN)* dari IANA, *Assigned Number field* ditetapkan oleh *provider*, contoh RD adalah 100:1, 171.1.1.:1



Gambar 2.11 Format *Route Distinguisher*[7]

2.3 *Routing* IP[8]

2.3.1. Pengertian *Routing* IP

Routing IP adalah proses pemindahan paket dari satu *network* ke *network* lain dengan menggunakan beberapa *router*. Istilah *routing* digunakan untuk proses pengambilan sebuah paket dari sebuah alat dan mengirimkannya melalui *network* – *network* yang saling terhubung ke alat lain disebuah *network* yang berbeda. Apabila dalam sebuah *network* tidak memiliki *router*, maka proses *routing* tidak akan dapat dijalankan. Agar dapat melakukan *routing* paket, sebuah *router* harus mengetahui, hal-hal sebagai berikut :

- Alamat tujuan (*Destination Address)*
- *Router-router* tetangga (*Neighbour Router*)
- *Route* terbaik untuk setiap *network remote.*

Pada dasarnya sebuah *routing protocol* menentukan jalur (*path)* yang dilalui oleh sebuah paket melalui sebuah *internetwork*. Protokol *Routing* memiliki kemampuan untuk membentuk jaringan informasi yang disusun dalam tabel *routing* secara statik maupun dinamik. Jika tabel routing tersebut dibuat secara dinamik maka apabila terjadi perubahan pada jaringan *network* atau jaringan protokol *routing* maka akan mampu memperbaharui informasi pada tabel *routing* tersebut. *Routing* pada jaringan terbagi menjadi 2 macam :

- *Interior Gateway Protocol* (IGP)
- *Exterior Gateway Protocol* (EGP)
  Merupakan sebuah protokol *routing* yang berfungsi untuk menangani proses *routing* antar *autonomous system.*

Beberapa protokol routing yang sering digunakan diantaranya adalah :

- *Routing Internet Protocol* (RIP)
- *Interior Gateway Routing Protocol* (IGRP)
- *Enhanced Interior Gateway Routing Protocol* (EIGRP)
- *Open Shortest Path First* (OSPF)
- *Border Gateway Protocol* (BGP)

Untuk pembagiannya, protokol routing OSPF, EIGRP, RIP dan IGRP termasuk dalam IGP, sedangkan protokol routing BGP termasuk dalam EGP.

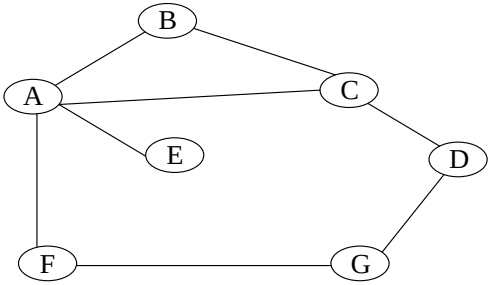
2.3.2. Jenis – Jenis *Routing* Protokol

Terdapat beberapa jenis *routing* protokol diantaranya :

1. *Routing* Statis (*Static Routing)*
   Merupakan *routing* yang terjadi jika konfigurasi dilakukan secara manual untuk menambahkan *route-route* di *routing table* setiap *router.*
2. *Routing Default* (*Default Routing)*
   Default routing mengirimkan paket-paket ke sebuah *network* tujuan yang *remote* yang tidak ada di *routing table.*
3. *Routing* Dinamis (*Dinamic Routing*)
   Routing dinamis adalah ketika *routing protocol* digunakan untuk menemukan *network* dan melakukan *update routing table* pada *router* karena secara otomatis akan meng-*update routing* yang terdapat pada *routing table.*

2.3.2.1. *Distance Vector Routing*

Distance Vevtor adalah *routing* dimana rute diberitahukan berdasarkan jarak dan arah, dengan satuan *metric*. *Distance Vektor* menggunakan algoritma *Bellman-Ford* untuk menentukan jalur terbaik untuk dilalui oleh data. Untuk mengetahui bagaimana algoritma dari *distance vektor* bekerja dapat dilihat dari gambar 2.12.



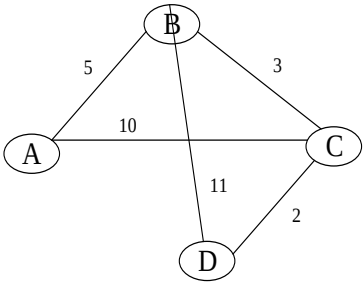
Gambar 2.12 *Routing Distance Vector*[5]

Pada algoritma Bellman-Ford, routing akan ditentukan berdasarkan jumlah *hop*, misalnya pada *router* A, *router* A akan menganggap *router* B,C dan E sebagai tetangganya, ketika *router* A akan menuju *router* C, maka *router* A akan terhubung secar langsung. Ketika *Router* A akan meunju *router* D maka *router A* akan menghitung berapa jumlah *hop* yang akan dilewati untuk menuju *router* D, pada skenario pertama *router* A dapat menuju D dengan melalui *hop* B dan *hop* C, pada skenario kedua *router* A akan melewati *hop* C. Pada kondisi

ini *router* A akan memilihi *router* C yang artinya *router* A hanya melewati satu *hop* saja, daripada melalui *router* B dan C yang memiliki jumlah 2 *hop*[5].

2.3.2.2. *Link State Routing*[5]

Pada umumnya *routing* protokol *link state* juga disebut *shortest-path-first* protokol. Dalam hal ini masing-masing *router* membangun tiga tabel terpisah. Tabel yang pertama adalah salah satu tabel menentukan topologi jaringan pada *router* yang terhubung secara langsung. Tabel yang kedua menentukan topologi dari seluruh kegiatan *internetwork* dan tabel yang ketiga digunakan sebagai tabel *routing*. *Router link state* mengetahui lebih banyak hal mengenai *internetwork* dibandingkan protokol *distance vector* lainnya. Salah satu contoh jenis *routing* protokol yang menggunakan metode *link state adalah Open Shortest Path First* (OSPF). OSPF adalah sebuah *routing* protokol yang benar-benar menerapkan metode *link state* dalam proses *routing* nya. Protokol *link state* akan mengirimkan pembaruan informasi yang terdiri dari keadaan *link* sebuah *route1r* ke semua *router* lain didalam suatu jaringan.[6] Untuk mengetahui bagaimana algoritma SPF atau Djkstra bekerja akan ditunjukkan pada gambar 2.13. Pada gambar tersebut *router* D berperan sebagai *router* yang baru terhubung kedalam jaringan yang ada.



Gambar 2.13 Contoh jaringan *Link State* [5]

Algoritma *routing link state* memiliki banyak keunggulan yaitu penstabilan yang cepat, tidak menghasilkan kepadatan trafik dan merespon dengan cepat dalam hal perubahan topologi atau kegagalan *router*, tetapi jumlah informasi yang disimpan pada setiap *router* (satu LSP untuk setiap *router* lainnya didalam jaringan) dapat cukup besar. [5]

2.4 *Open Shortest Path First* (OSPF) [3]

2.4.1. Pengertian *Open Shortest Path First*

OSPF merupakan *routing protocol* berbasis *link state*, termasuk dalam *Interior Gateway Protocol* (IGP). Menggunakan algoritma Djikstra untuk menghitung *shortest path first* (SPF). Menggunakan *cost* sebagai ruang metric. Setelah antar *router* bertukar informasi maka akan terbentuk database link state pada masing-masing *router*. OSPF mungkin merupakan IGP yang paling banyak digunakan. Menggunakan metode MD5 untuk autentufukasi antar *router* sebelum menerima *Link-state Advertisement* (LSA). Dari awal OSPF sudah mendukung CIDR dan VLSM, berbeda dengan RIP, bahkan untuk OSPFv3 sudah mendukung untuk Ipv6.

OSPF (*Open Shortest Path First*) merupakan protokol *Link State Routing* yang dijelaskan dalam RFC 1583. OSPF menggunakan algoritma *routing Link State*, yang dengan demikian, dapat membuat seleksi jalur yang lebih cerdas daripada protokol *Distance Vector Routing*. Ketika menentukan jalur terbaik ke tujuan, OSPF seperti semua *Routing Link State*, dapat mempertimbangkan salah satu atau semua metrik berikut: *bandwidth*, *delay*, keandalan dan beban. Semua *router* OSPF, jaringan, dan subnet secara logis dikelompokkan ke daerah. Jaringan OSPF mungkin terdiri dari satu daerah atau beberapa daerah diorganisasikan secara hirarki. Apakah area tunggal atau ganda daerah ada seluruh OSPF *internetwork* disebut sebagai salah satu domain routing (atau AS). Divisi hirarkis ini mengisolasi perubahan dan pembaruan rute lalu lintas ke daerah dan mengurangi *overhead* yang terlibat dengan mempertahankan tabel *routing* yang besar dan rute *recalculations* ketika perubahan terjadi. Sebagai protokol *routing Link State* OSPF memiliki banyak keuntungan dibandingkan *Distance Vector*. Sebagai contoh, OSPF menggunakan *multicast* bukan siaran untuk menyebarkan informasi rute ke *router* tetangga. *Update Route* dipicu, yang berarti mereka hanya dikirim bila perubahan terjadi pada jaringan (tidak periodik). *Update* dipicu dibanjiri dengan semua OSPF *router*, meningkatkan waktu konvergensi. Selain itu, setiap pembaruan hanya mencakup berubah informasi rute, tidak seluruh tabel. *Update* rute terisolasi hanya *router* di Daerah OSPF di mana perubahan terjadi. Tidak seperti protokol *Distance Vector*, seperti RIP dan IGRP, tidak mampu mendukung untuk jaringan besar karena tidak memiliki kemampuan menghitung hop maksimum dan membatasi diameter[9].

Jenis jenis paket data yang dikirimkan pada *routing* protokol OSPF ada 5 macam yaitu :

a. *Hello Packet*

Hello Packet digunakan untuk menemukan serta membentuk suatu hubungan tetangga antara *router* OSPF. Untuk membentuk hubungan ini *router* OSPF akan mengirimkan paket berukuran kecil secara berkala ke jaringan. Paket inilah yang disebut dengan *Hello Packet*. Paket ini juga mengadpertensikan *router* mana saja yang akan menjadi tetangganya. Pada jaringan multi-access *Hello Packet* digunakan untuk memilih *Designated Router* (DR) dan *Back-up Designated Router* (BDR). DR dan BDR akan menjadi pusat komunikasi seputar informasi OSPF dalam jaringan tersebut. *Network Mask* pada format *Hello Packet* merupakan mask dari interface jaringan dari OSPF yang sedang berjalan. Subnet-Mask nya 0.0.0.0 (4 byte). *Hello Interval* biasanya *multicast* (224.0.0.5). Merupakan jumlah detik antara *Hello Packet*, biasanya 10 detik pada link *point-to-point* dan 30 detik pada NBMA / *link broadcast*. Options merupakan kemampuan opsional yang dimiliki *router*. RTR Prio digunakan dalam pemilihan DR dan BDR. *Router* dengan nilai *priority* tertinggi akan menjadi DR. *Router* dengan nilai *priority* di urutan kedua sebagai BDR. Secara default semua *router* OSPF memiliki nilai *priority* 1. Dengan *Range priority* mulai dai 0 hingga 255. Bila prioritasnya 0 berarti *router* tersebut tidak memenuhi syarat dalam pemilihan DR dan BDR, sedangkan nilai 255 menjamin sebuah *router* menjadi DR. Jjika dua buah *router* memiliki nilai priority sama, maka yang menjadi DR dan BDR adalah *router* yang memiliki nilai *router* ID tertinggi dalam jaringan. *Router Dead Interval* merupakan jumlah dalam hitungan detik sebelum tetangga dinyatakan *down*. Secara *default dead interval* adalah 4 kali *hello interval*. *Designated Router* bertujuan untuk mengurangi jumlah *flooding* pada media *multiaccess*. *Backup Designated Router* bertujuan sebagai cadangan dari DR. Selama *flooding* berlangsung, BDR tetap pasif. Neighbor berisi ID dari setiap *router* tetangga.

b. *Database Description* (DBD)

DBD digunakan selama pertukaran *database*. Paket DBD pertama digunakan untuk memilih hubungan *master* dan *slave* serta menetapkan urutan yang dipilih oleh *master*. Pemilihan *master* dan *slave* berdasarkan *router* ID tertinggi dari salah satu *router*. *Router* dengan *router* ID tertinggi akan menjadi master dan memulai sinkronisasi database. *Router* yang menjadi master akan melakukan pengiriman lebih dulu ke *router slave*. Peristiwa ini di istilahkan fase

Exstart State. Setelah fase *Exstart State* lewat, selanjutnya adalah fase *Exchange*. Pada fase ini kedua *router* akan saling mengirimkan *Database Description Packet*. Bila si penerima belum memiliki informasi yang terdapat dalam paket tersebut, maka *router* pengirim akan memasuki fase *Loading State*. Dimana fase ini *router* akan mengirimkan informasi state secara lengkap ke *router* tetangganya. Setelah selesai *router-router* OSPF akan memiliki informasi *state* yang lengkap dalam databasenya, ini disebut fase *Full State*.

c. *Link-State Request* (LSR)
   LSR akan dikirim jika bagian dari *database hilang* atau *out of date*. LSR juga digunakan setelah pertukaran DBD selesai untuk meminta LSAs yang telah terjadi selama pertukaran DBD.
d. *Link-State Update* (LSU)
   LSU mengimplementasikan *flooding* dari LSAs yang berisi *routing* dan informasi metric. LSU dikirim sebagai tanggapan dari LSR.
e. *Link-State Acknowledgement* (LSAck)
   OSPF membutuhkan pengakuan untuk menerima setiap LSA. Beberapa LSA dapat diakui dalam sebuah paket *single link-state acknowledgement*. Paket ini dikirim sebagai jawaban dari paket *update link state* serta memverifikasi bahwa paket *update* telah diterima dengan sukses. LSAck akan dikirim sebagai *multicast*. Jika *router* dalam keadaan DR atau BDR maka pengakukan dikirim ke alamat *multicast router* OSPF dari 224.0.0.5 sedangkan bila *router* dalam keadaan tidak DR atau BDR pengakuan akan dikirim kesemua alamat *multicast router* DR dari 224.0.0.6.

2.5 *Routing Information Protocol* (RIP)[10]

2.5.1. Pengertian RIP

RIP merupakan salah satu standarisasi dari *Distance Vector Protocol*, RIP didesain untuk digunakan pada jaringan kecil. RIP merupakan salah satu yang paling tepat dari *protocol Distance Vektor Routing* dan didukung oleh sistem yang kompleks.

Karakteristik RIP yang melekat pada *Distance Vektor* :

a. RIP mengirim sinyal pemberitahuan *routing periodic* setiap 30 detik
b. RIP mengirim sinyal pemberitahuan tabel routing setiapperiode
c. RIP menggunakan bentuk jarak metrik seperti hopcount

d. RIP menggunakan *Bellman-Ford Distance Vektor* algoritma untuk menentukan *path* mana yang lebih baik
e. RIP mendukung IP dan IPX routing
f. RIP utilitas merupakan UDP port 520
g. RIP rute memiliki jarak administrasi sebesar 120
h. RIP memiliki *hop count* maksimal sebesar 15 hop

RFC 1058 (versi 1) dan 2453 (versi 2) mendefinisikan RIP (*Routing Information Protocol*). Kedua versi dari protokol RIP awalnya dirancang di sekitar Xerox XNS protokol dan program diarahkan diintegrasikan ke dalam pelaksanaan Berkeley Unix. Protokol RIP dapat berjalan pada *host* akhir atau *router*. Meskipun berjalan di atas UDP dan IP, memanfaatkan port UDP 520, itu diklasifikasikan sebagai lapisan protokol jaringan karena fungsi dan operasi didasarkan pada penyampaian lapisan jaringan datagrams. Berfungsi sebagai IGP (*Interior Gateway Protocol*), RIP menyediakan penentuan rute dalam Sistem Otonom. RIP bekerja paling baik bila diterapkan pada jaringan berukuran kecil karena keterbatasan, RIP menunjukkan karakteristik sebagai berikut:

a. *Broadcast* berbasis-*Router* pada segmen yang sama update pertukaran rute melalui siaran.
b. IGP-RIP paling baik digunakan sebagai protokol *routing* IGP *interior*. IGP *routing* protokol tetap melacak rute internal untuk *internetwork* organisasi.
c. Dapat bekerja dengan baik pada jaringan berukuran kecil dikarenakan RIP mengimplementasikan siaran dan terbatas sehingga tidak cocok untuk menengah ke jaringan besar.
d. *Distance vector* protokol routing-RIP menggunakan hop sebagai jarak metrik untuk membuat yang terbaik seleksi jalur ke tujuan.
e. Nilai Metric adalah hop-A hop adalah satuan jarak yang digunakan oleh *router* RIP. Setiap jaringan *traverse datagram* dianggap sebagai hop tunggal. Jumlah maksimum hop adalah 15.
f. *Router* periodik update-RIP menggunakan timer periodik update untuk mengontrol siaran rute. Update Timer periodik adalah 30 detik.
g. Mengirim keluar seluruh tabel rute, apakah perubahan telah terjadi dalam jaringan-*Router* siaran *update* rute ke *router* tetangga secara berkala, termasuk semua (berubah dan tidak berubah) entri rute dari tabel rute mereka.

h. Terbatas pada diameter jaringan maksimum 15 hop. Jumlah maksimal jaringan datagram dapat melintasi adalah 15 (hop). Setiap nilai di atas ini dianggap tidak terjangkau.

Sebagai sebuah *routing protocol* berbasis *distance-vector*, RIP mengimplementasikan Bellman-Ford (juga disebut sebagai Ford-Fulkerson) algoritma untuk menentukan pilihan jalan terbaik. RIP menyeleksi jalur terbaik ke tujuan dengan jarak terpendek, diukur dalam hop dengan jarak maksimum 15. Pembaruan dikirim menggunakan update berkala, termasuk semua informasi dalam tabel rute, bahkan ketika tidak ada perubahan yang terjadi dalam jaringan. Meskipun banyak protokol *routing* lain ada saat ini dengan lebih cerdas dan pemilihan rute yang efisien, RIP v1 tetap yang paling populer protokol yang digunakan[9].

2.5.2. Versi dari RIP

RIP memiliki dua versi yaitu RIPv1 dan RIPv2, yang digunakan dalam penelitian ini merupakan RIPv2. RIPv2 (RFC 2543) termasuk dalam kategori *classless*, dan dengan begitu termasuk *subnet mask* dengan pemberitahuan *routing* tabel. Berikut merupakan perbandingan antara RIPv1 dan RIPv2.

Tabel 2.2 Perbandingan *Routing Protocol* RIPV1 dan RIPv2

	RIPv1	RIPv2
Protokol *Routing*	*Classful*	*Classless*
Mendukung VLSM	Tidak	Ya
Membawa Informasi *Subnet Mask* Dalam *Routing*	Tidak	Ya
Mendukung Autentikasi	Tidak	Ya
Mendukung *Manual Route Summarization*	Tidak	Ya
Tipe Addressing	*Broadcast*	*Multicast*

RIPv2 sangat mendukung VLSMs, yang mengabaikan *discontiguous network* dan berbagai jenis *subnet mask* yang ada. Peningkatan baru yang diberikan oleh RIPv2 adalah :

a. Pemberitahuan *routing* yang dikirm melalui *multicast*, menggunakan alamat 224.0.0.9
b. Pengesahan yang terenkripisi dapat dikonfigurasi diantara *router* RIPv2.
c. Mendukung untuk *Route tagging*.
d. RIPv2 dapat digunakan dengan RIPv1 standar.
e. *Route*s RIPv1 hanya akan mengirim paket versi 1.
f. *Routers* RIPv1 akan menerima pemberitahuan kedua tipe yaitu RIPv1 dan RIPv2.
g. *Routers* RIPv2 akan mengirim dan menerima pemberitahuan dengan tipe 2.

Pada *routing* protokol RIPv2 juga mengirimkan paket yang akan dipertukarkan antar router yaitu paket *Hello* dimana paket ini akan melakukan update informasi dalam hitungan detik. Selama 30 detik (waktu *default update timer*), untuk *update routing* table secara periodik, dimana sebuah router mengirimkan sebuah *copy* lengkap dari *routing* tablenya ke *router* tetangga .

2.6 *Enhanced Interior Gateway Routing Protocol* (EIGRP)[14]

2.6.1. Pengertian EIGRP

Enhanced Interior Gateway Routing protocol (EIGRP) adalah sebuah *protocol proprietary* (milik) Cisco yang bekerja pada *router* Cisco dan pada prosesor route internal yang terdapat pada *switch layer core* dan *switch layer* distributor Cisco.

2.6.2. Fitur dan Operasi EIGRP

Enhanced IGRP (EIGRP) adalah sebuah *protocol distance-vector* yang classless dan yang sudah ditingkatkan (*enhanced*), yang memberikan kita keunggulan yang nyata dibandingkan protocol propriertary Cisco lainnya, yaitu *Interior Gateway Routing Protocol* (IGRP). Inilah pada dasarnya mengapa disebut *Enhanced* IGRP. Seperti IGRP, EIGRP menggunakan konsep dari sebuah *autonomous system* untuk menggambarkan kumpulan dari *router-router* yang *contiguous* (berentetan sebelah-menyebelah) yang menjalankan *routing* protokol yang sama dan berbagi informasi routing. Tetapi tidak seperti IGRP, EIGRP memasukan subnet mask ke dalam update route-nya. Dan seperti yang sekarang diketahui, pengumuman (*advertisement*) dari informasi subnet memungkinkan penggunakan VLSM dan melakukan *summarization* (perangkuman) ketika merancang network.

EIGRP kadang-kadang disebut sebagai *routing protocol* hybrid karena mempunyai karakteristik – karakteristik baik dari protocol *distance-vector* maupun dari *protocol link-state*. Sebagai contoh, EIGRP tidak mengirimkan paket – paket *link-state* seperti dilakukan OSPF, melainkan mengirimkan update *distance-vector* yang tradisional yang berisi informasi tentang network – network ditambah dengan biaya untuk mencapai mereka dari prespektif *router* yang melakukan pengumuman tersebut. Sebuah EIGRP memiliki karakteristik – karakteristik *link-state* seperti mensinkronisasikan routing table antara *router* – *router* tetangga pada saat dimulai dijalankan (*startup*), dan kemudian mengirimkan update-update yang spesifik hanya jika topologi network berubah. Ini membuat EIGRP sesuai untuk network – network yang sangat besar. EIGRP mempunyai sebuah jumlah hop maksimum.

Ada sejumlah fitur yang kuat dan membuat EIGRP jauh lebih baik dibandingkan IGRP dan protocol-protocol lainnya. Yang utamanya adalah sebagai berikut :

a. Mendukung IP, IPX, dan *AppleTalk* melalui modul-modul yang bersifat protocol-dependent (bergantung pada protocol)
b. Pencarian *network* tetangga (*neighbor discovery*) yang dilakukan dengan efisien.
c. Komunikasi melalui *Reliable Transport Protocol* (RTP)
d. Pemilihan jalur terbaik melalui *Diffussing Update Algorithm* (DUAL)

Pada *routing* protokol EIGRP mengirimkan beberapa paket yang akan dipertukarkan antar *router* dimana paket – paket ini digunakan untuk membangun konduktivitas jaringan pada protokol EIGRP. Beberapa paket tersebut yaitu :

a. *Hello Packet*
   Hello Packet dikirim secara *multicast* melalui *ip address* 224.0.0.10. *Hello pacet* digunakan untuk mengetahui jalur ke arah router lain apakah masih tersambung atau tidak. *Hello packet* secara default dikirimkan setiap 15 detik secara simultan. Jika *router* lain tidak merespon paket ini melebihi *hold time* yaitu 45 detik maka jalur ke *router* lain tersebut akan dianggap mati dan DUAL akan mengkalkulasikan uang dan mencari jalur yang lain.
b. *Acknowledgement Packet*
   Acknowledgment (Ack) paket merupakan paket yang mirip dengan *Hello packet* tapi memiliki tidak memiliki data di dalamnya.

Berfungsinya sebagai balasan atau respon apabila suatu paket telah diterima oleh *router* tujuan. *Acknowledgment* paket digunakan sebagai balasan *Update* paket, *Query* paket dan *Reply* paket, sedang pada *Hello* paket dan Ack paket tidak memerlukan balasan. Ack paket dikirim secara *unicast.*

c. *Update Packet*

Update Packet digunakan untuk menyampaikan tujuan yang dapat dijangkau oleh *router*. Ketika sebuah *router* baru ditemukan, *update* paket dikirim secara *unicast* sehingga *router* dapat membangun tabel topologi. Dalam kasus lain, *update* paket dikirim secara *multicast* untuk perubahan *link-cost.*

d. *Query Packet*

Query Packet adalah sebuah paket yang berisi *request* atau pertanyaan yang dilakukan secara *multicast* yang akan digunakan untuk meminta sebuah rute. Selama mengirimkan *query* paket, setiap *router* akan melanjutkan paket tersebut sampai pada *router* yang akan mengirimkan *reply* paket sebagai informasi rute agar *router* pengirim dapat menuju jaringan tersebut. *Query* paket dikirim secara *multicast*.

e. *Reply Packet*

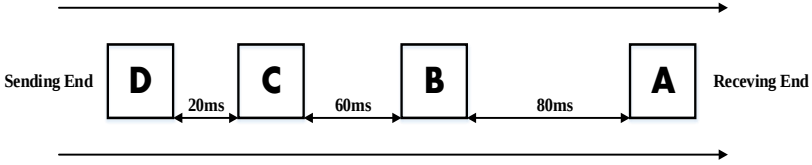
Reply Packet dikirim sebagai respon dari *query* paket. *Reply* paket dikirim secara *unicast* kepada *router* yang mengirimkan *query* paket.

2.7 *Quality Of Service*

2.7.1. Pengertian *Quality Of Service* [13]

Quality of Service (QoS), sebagaimana dijelaskan dalam rekomendasi CCITT E.800 adalah : "Efek kolektif dari kinerja layanan yang menentukan derajat kepuasan seorang pengguna terhadap suatu layanan" Jika dilihat dari ketersediaan suatu jaringan, terdapat karakteristik kuantitatif yang dapat dikontrol untuk menyediakan suatu layanan dengan kualitas tertentu. Pengertian *Quality Of Service* adalah kemampuan untuk menyediakan jaminan dan performa layanan pada suatu jaringan. Terdapat banyak hal yang bisa terjadi pada paket dikirim ketika mereka melakukan perjalanan dari asal ke tujuan maupun dari tujuan ke asal, masalah – masalah tersebut sering disebut sebagai parameter – parameter *QoS*, yang meliputi *Throughput, Delay*, *Jitter*, dan *Packet Loss*[12]. Kinerja jaringan dapat dilihat dari beberapa nilai parameter – parameter kualitas layanan yaitu *throughput*, *delay*, *jitter*, dan *packetloss*.

- *Throughput* : Aspek utama *throughput* yaitu berkisar pada ketersediaan *bandwidth* yang cukup untuk suatu aplikasi. Hal ini menentukan besarnya trafik yang dapat diperoleh aplikasi saat melewati jaringan. Aspek penting lainnya adalah *error* (pada umumnya berhubungan dengan *link error rate*) dan *losses* (pada umumnya berhubungan dengan kapasitas *buffer*).
- *Delay* merupakan rata – rata waktu yang dibutuhkan untuk mengirimkan suatu paket data dari sumber (pengirim) ke tujuan (penerima) dalam sebuah route. *Delay* maksimum yang direkomendasikan oleh ITU untuk aplikasi suara adalah 150 ms, dan yang masih bisa diterima pengguna adalah 250ms
- *Jitter* merupakan variasi *delay* yang terjadi akibat adanya selisih waktu atau interval antar kedatangan paket di penerima. Diakibatkan oleh panjang *queue* dalam suatu waktu pengolahan data, *reassemble* paket – paket data di akhir pengiriman akibat kegagalan sebelumnya dan proses pengiriman paket dalam media sehingga dapat dikatakan juga sebagai variasi delay jaringan.



Gambar 2.14 Ilustrasi *Jitter* Suatu Paket Data[11]

Jitter dapat diilustrasikan seperti pada gambar 2.14, satu *source* mengirimkan paket data A-B-C-D, setiap paket dikirimkan ke *destination* dengan variasi *delay* (*jitter*) yang berbeda – beda, antara paket A dan B terdapat variasi delay sebesar 80 ms, antara paket B dan C sebesar 60 ms, dan antara paket C dan D sebesar 20 ms.

- *Packet Loss* : Kehilangan paket ketika terjadi *peak load* dan *congestion* (kemacetan transmisi paket akibat padatnya *traffic* yang harus dilayani) dalam batas waktu tertentu.