Bab 2

LANDASAN TEORI

2.1 *Computer Network*

Computer Network (Anonymous, 2005c) adalah sebuah sistem komunikasi yang menghubungkan dua komputer atau lebih. Dalam hal ini, ditekankan bahwa *network* tidak hanya terdiri dari komputer saja, tetapi juga *peripheral-peripheral* lainnya seperti *printer*, *modem*, *plotter*, *scanner*, dan *peripheral* lainnya yang terhubung oleh beberapa medium seperti kabel, *fiber optic* maupun perangkat *wireless*.

Berdasarkan luas jangkauannya, *network* dapat dibagi menjadi 3 bagian (Cisco Systems, 2005a; Stallings, 2004, pp7 – pp8), yaitu :

1. *Local Area Network* (LAN)

   LAN adalah jaringan komputer yang mencakup satu lokal area seperti rumah, kantor atau kampus. Secara umum, LAN mencakup area maksimal 1 km^2.

2. *Metropolitan Area Network* (MAN)

   MAN adalah jaringan komputer besar yang menghubungkan jaringan antar kampus atau kantor bahkan antar kota yang berdekatan. MAN mencakup area lebih besar dibandingkan dengan LAN, tetapi lebih kecil dibandingkan dengan WAN.

3. *Wide Area network* (WAN)

   WAN adalah jaringan komputer yang sangat besar yang menghubungkan banyak LAN dan MAN dalam cakupan area yang sangat luas (> 100 km^2). WAN menghubungkan jaringan-jaringan komputer dalam jumlah yang sangat besar. Salah satu contoh nyata WAN adalah *internet*.

2.2 Topologi Jaringan

(Cisco Systems, 2005a) Topologi jaringan mendeskripsikan struktur dari suatu jaringan. Topologi jaringan terbagi menjadi 2 jenis yaitu topologi fisik dan topologi logik. Topologi fisik merupakan topologi jaringan yang memberikan gambaran tentang jalur kabel atau media, sedangkan topologi logik lebih menjelaskan bagaimana suatu media diakses untuk pengiriman data.

Secara umum, topologi fisik terbagi menjadi beberapa jenis (Cisco Systems, 2005a), seperti :

1. Topologi *bus*

   Topologi ini menggunakan sebuah kabel sebagai *backbone* yang memiliki alat terminal di kedua sisi ujungnya. *Host-host* terhubung langsung ke *backbone* ini.

2. Topologi *ring*

   Topologi ini menghubungkan satu *host* dengan *host* lainnya dan *host* terakhir terhubung ke *host* yang pertama. Topologi ini akan terlihat seperti sebuah lingkaran (*ring*).

3. Topologi *star*

Topologi ini menghubungkan *host-host* dengan satu alat jaringan secara terpusat.

4. Topologi *extended-star*

   Topologi ini menghubungkan beberapa jaringan yang menggunakan topologi *star*. Sebuah *line* dari masing-masing *hub* atau *switch* dihubungkan dengan sebuah alat jaringan secara terpusat. Keunggulan topologi ini adalah dapat dengan mudah untuk memperluas *segmen* atau area jaringan.

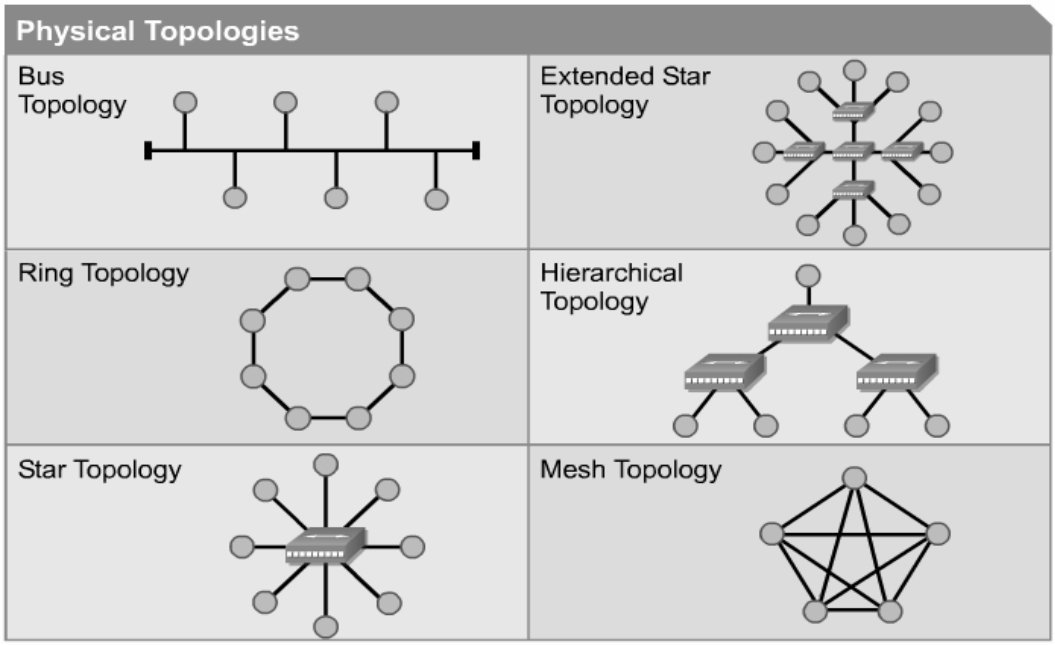
5. Topologi *hierarchical*

   Topologi ini mirip dengan topologi *extended-star*, akan tetapi topologi ini lebih memudahkan seorang *administrator* dalam mengatur *traffic* jaringan dari *segmen* yang berada di bawahnya secara hierarki.

6. Topologi *mesh*

   Topologi ini menghubungkan satu *host* ke semua *host* yang berada di jaringannya. Begitu juga dengan *host* yang kedua, juga terhubung ke semua *host* lainnya. Keunggulan topologi ini adalah jaringan yang *reliable*, karena bila satu *path* atau *line* terputus, tidak akan mempengaruhi jaringan karena masih tersedia *path-path* lainnya dalam pengiriman data.

Berikut ini menggambarkan bentuk-bentuk topologi jaringan yang telah dijelaskan di atas (Cisco Systems, 2005a).



Gambar 2.1 Topologi Jaringan Fisik

Topologi logik menjelaskan bagaimana *host-host* dapat saling berkomunikasi melalui media. Dua jenis topologi logik yang sering digunakan (Cisco Systems, 2005a) adalah :

1. *Broadcast*

   Topologi logik secara *broadcast* menjelaskan bahwa setiap *host* mengirimkan data ke semua *host* lainnya yang berada di jaringannya. Tidak ada aturan yang mengatur jalannya data di jaringan. Teknik ini lebih dikenal dengan istilah *first come first serve*. Salah satu teknologi yang menggunakan teknik ini adalah teknologi ethernet.

2. *Token Passing*

   Teknik ini mengatur jalannya data dan pemakaian jaringan dengan cara mengirimkan sebuah *token-elektronic* secara sekuensial ke setiap *host*. Ketika *host* mendapatkan *token-electronic* ini, maka *host* ini berhak untuk

menggunakan jaringan untuk mengirim data. Bila telah selesai menggunakan jaringan, *token-electronic* kembali dikirimkan ke *host-host* lainnya. Teknologi yang menggunakan teknik ini adalah *Token Ring* dan *Fiber Distributed Data Interface* (FDDI).

2.3 *Network Devices*

Alat yang paling dasar dalam membangun sebuah jaringan adalah media yang menghubungkan *host* dengan *host* lainnya. Fungsi dari media tersebut adalah sebagai wadah aliran data dalam jaringan. Secara garis besar, media jaringan terdapat 2 jenis yaitu *wire* dan *wireless* (Cisco Systems, 2005a). Beberapa jenis media yang tergolong *wire* adalah *twisted-pair cable*, *coaxial cable*, dan *fiber optic*. Sedangkan tipe *wireless* menggunakan atmosfer atau udara sebagai media. Pemilihan penggunaan berbagai tipe media ini bergantung pada jarak antar titik yang akan dihubungkan, biaya, kemudahan instalasi dan tingkat gangguan (*interference*) terhadap *noise*.

Beberapa penjelasan alat-alat jaringan yang berdasarkan Cisco Systems (2005a) diberikan sebagai berikut :

1. *Repeaters*

   Pada awalnya, alat ini muncul karena keterbatasan media dalam hal mentransfer data berupa sinyal. Semakin jauh sinyal ditransfer, sinyal akan semakin lemah. *Repeaters* mampu mengatasi hal ini dengan kinerjanya yaitu menerima sinyal, menguatkan kembali sinyal yang diterima dan memancarkan kembali sinyal tersebut.

2. *Hubs*

Alat ini umumnya disebut *multiport repeaters*, karena secara umum fungsinya sama seperti *repeaters*. *Repeaters* hanya memiliki 2 *port* sedangkan *hubs* memiliki 4 hingga 24 *port*. *Hubs* terdiri dari 3 jenis, yaitu *passive hubs, active hubs, dan intelligent hubs*. *Passsive hubs* tidak memiliki kemampuan memanipulasi *traffic* data di jaringan. Fungsinya hanya meneruskan sinyal ke semua *port* lainnya. *Passive hubs* tidak membutuhkan tenaga listrik. *Active hubs* membutuhkan tenaga listrik untuk menguatkan sinyal yang diterima dan kemudian meneruskan ke *port-port* lainnya. *Intelligent hubs* memiliki fungsi tambahan dibandingkan dengan *active hubs* yaitu kemampuan mendiagnosa jaringan bila terjadi *colission*.

3. *Bridge*

Bridge memiliki tabel MAC-*address* dari semua *host* yang terhubung dengannya. *Bridge* akan mencocokkan MAC-*address* dari data yang diterima dengan tabel dan meneruskan data ke *port* yang sesuai dengan tujuan dari dari data tersebut. Berbeda dengan *hubs* yang meneruskan data ke semua *port* yang terhubung dengannya. Dalam hal ini, *bridge* berfungsi membagi satu *network* menjadi beberapa segmen yang lebih kecil. Dengan demikian, *bridge* mampu membatasi *collision domain*.

4. *Switch*

Alat ini umumnya disebut *multiport bridge*, karena secara umum fungsinya sama seperti *bridge*. *Bridge* pada umumnya hanya memiliki 2 *port* sedangkan *switch* memiliki 4 hingga 24 *port*. *Switch* memiliki 2 jenis yaitu *unmanageable switch* dan *manageable switch*. *Unmanageable switch*

merupakan *switch* yang tidak dapat dikonfigurasi dan fungsinya sama dengan *multiport bridge*, sedangkan *manageable switch* merupakan *swicth* yang dapat dikonfigurasi dengan kemampuan lebih dibandingkan dengan *unmanageable switch* seperti kemampuan *virtual local area network* (VLAN). VLAN merupakan teknologi segmentasi *network* di *layer* 2 (*data link layer*) dimana dapat mengurangi besarnya *broadcast domain*.

5. *Routers*

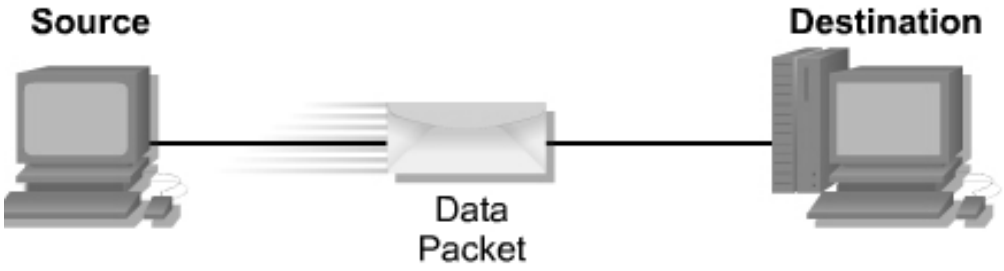
   Alat ini berfungsi untuk me-*routing* paket data antar *network* yang berbeda serta menghubungkan suatu jaringan ke *wide area network* (WAN). Karena kemampuannya mengarahkan (*routing)* paket data berdasarkan alamat IP, *router* menjadi alat yang cukup penting di dalam jaringan *internet*. Selain berfungsi sebagai *routing*, *routers* juga memiliki kemampuan lain seperti menghubungkan beberapa teknologi yang berbeda (misalnya teknologi *token ring* dengan *ethernet*) dan kemampuan sekuriti yaitu penerapan *access control list* (ACL). ACL merupakan sekumpulan *rule* yang berfungsi sebagai pengatur lalu lintas paket data masukan atau keluaran, dan memutuskan apakah paket data tersebut diizinkan lewat atau tidak.

2.4 Konsep *Networking Models*

2.4.1 Pengenalan *Layer*

(Cisco Systems, 2005a) Konsep *layer* digunakan untuk menjelaskan bagaimana komputer berkomunikasi satu sama lainnya. Konsep *layer* menjelaskan bagaimana jaringan komputer mendistribusikan informasi dari sumber ke tujuan. Ketika komputer mengirimkan informasi melalui

network, semua komunikasi diatur di sumber dan kemudian dikirim ke tempat tujuan. Berikut ini menggambarkan aliran data dikirim dari sumber ke tujuan.



Gambar 2.2 Aliran Data Dikirim dari Sumber ke Tujuan

Informasi yang ada dalam jaringan secara umum disebut sebagai data atau paket. Sebuah paket secara logik merupakan sekumpulan unit informasi yang bergerak di antara sistem komputer. Setiap kali data melewati *layer*, informasi ditambahkan dari setiap *layer* yang akan mengefektifkan komunikasi dengan *layer* penerima pada komputer tujuan.

Model *Open System Interconnection* (OSI) dan *Transmission Control Protocol*/*Internet Protocol* (TCP/IP) memiliki *layer* yang menjelaskan bagaimana data berkomunikasi dari komputer yang satu ke komputer yang lainnya. Model tersebut memiliki perbedaan pada jumlah dan fungsi *layer* yang dimilikinya. Tetapi, setiap model dapat digunakan untuk menjelaskan dan menyediakan keterangan lengkap tentang aliran informasi dari sumber ke tujuan.

2.4.2 *Open System Interconnection* (OSI) *Layer*

(Cisco Systems, 2005a) Pada awal tahun 1980an terjadi peningkatan pesat jumlah dan ukuran jaringan. Setelah terjadi hal tersebut dan teknologi berkembang pesat, disadari akan sulit sekali berkomunikasi dengan bahasa yang berbeda. Maksudnya adalah, alat-alat jaringan yang dikembangkan tidak memiliki standar aturan, sehingga alat-alat jaringan mengalami masalah dalam berkomunikasi antar alat jaringan yang berbeda.

Untuk mengatasi masalah komunikasi ini, *International Organization for Standardization* (ISO) mengembangkan model jaringan seperti *Digital Equipment Corporation net* (DECnet), *System Network Architecture*(SNA), dan TCP/IP untuk menetapkan aturan-aturan yang dapat diimplementasikan ke semua jaringan. Dengan model yang dikembangkan oleh ISO, *vendor* dapat membuat jaringan yang sesuai standar sehingga mampu berkomunikasi dengan alat jaringan yang berbeda.

Open System Interconnection (OSI) yang dikeluarkan tahun 1984 merupakan model jaringan yang dibuat oleh ISO. OSI menyediakan *vendor* dengan serangkaian standar yang menjamin kesesuaian yang lebih tinggi dengan teknologi jaringan lainnya.

Keuntungan dari model OSI *layer* menurut Cisco Systems (2005a) adalah :

1. Mengurangi kerumitan
2. Standarisasi *interface*
3. Mempermudah perancangan secara modular

4. Menjamin interoperabilitas teknologi yang berbeda
5. Perkembangan yang sangat cepat
6. Mempermudah pembelajaran dan pengajaran

OSI merupakan sebuah *framework* yang digunakan untuk mengerti bagaimana informasi berjalan dalam *network*. Model OSI menjelaskan bagaimana paket berjalan melalui berbagai macam *layer* ke *hardware* dalam sebuah *network*, bahkan bila pengirim dan penerima memiliki tipe media jaringan yang berbeda.

Model referensi OSI memiliki tujuh *layer* dengan fungsinya masing-masing. Sebuah data yang melewati model ini akan melalui tujuh *layer* tersebut secara berurutan tergantung dari arah data tersebut. *Layer-layer* tersebut berikut fungsinya mulai dari *layer* teratas adalah sebagai berikut (Cisco Systems, 2005a) :

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

   Menyediakan servis kepada aplikasi seperti *e-mail*, *transfer file*, dan lainnya.

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

   Menangani representasi data seperti *format*, *encoding* dan jenis kompresi.

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

   Bertugas untuk membuat, mengatur, dan memutuskan sesi antar aplikasi.

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

   Menyediakan *end-to-end connection* yang menjamin reliabilitas data.

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

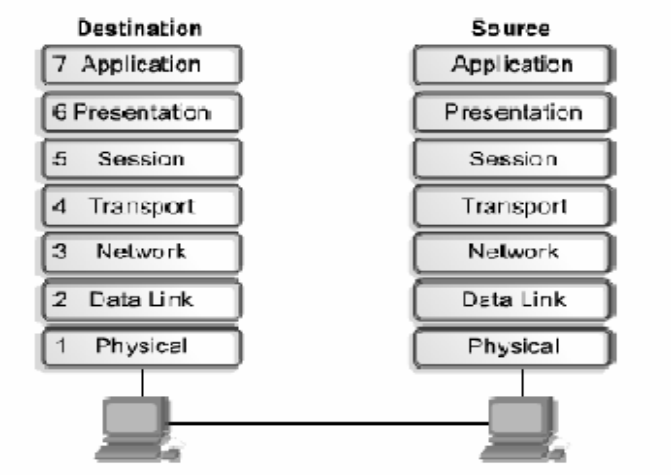
   Memberi alamat logik data dan menentukan jalur yang akan dilewatinya.

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

   Menangani perpindahan data dari *network interface* menuju ke medium fisik.

7. *Physical Layer* (Layer 1)

   Mengatur bagaimana data dapat ditransmisikan dalam sebuah medium fisik.



Gambar 2.3 Alur Data Melewati 7-*layer* OSI

Setiap *layer* bertanggung jawab untuk meneruskan data dalam bentuk yang sesuai kepada *layer* di atas dan di bawahnya.

2.4.3 *Transmission Control Protocol / Internet Protocol* (TCP/IP)

2.4.3.1 Model TCP/IP *Layer*

(Cisco Systems, 2005a) U.S. Departement of Defense (DoD) menciptakan model TCP/IP, karena DoD ingin mendesain *network* yang dapat tetap berfungsi dalam kondisi apapun, termasuk perang nuklir. Dunia *network* terhubung dengan berbagai macam media yang berbeda-beda seperti *cooper wires, microwaves, optical fibers,* dan *satellite links*. DoD menghendaki transmisi paket setiap saat dalam kondisi apapun. Kesulitan ini membawa DoD ke dalam penciptaan TCP/IP.

Tidak seperti teknologi *networking* yang sebelumnya, TCP/IP dirancang dengan standar terbuka. Ini berarti semua orang bebas untuk menggunakan TCP/IP. Ini mempercepat perkembangan TCP/IP sebagai sebuah standar.

Sebagai sebuah protokol TCP/IP juga memiliki model referensi sendiri yang terdiri dari empat *layer* dengan keterangan sebagai berikut (Cisco Systems, 2005a):

1. *Application Layer*

   Layer ini berfungsi untuk menangani *high-level protocol,* masalah representasi data, proses *encoding*, dan *dialog control*; yang memungkinkan terjadinya komunikasi antar aplikasi jaringan. *Layer* ini berisi spesifikasi protokol-

protokol khusus yang menangani aplikasi umum seperti Telnet, *File Transfer Protocol* (FTP), *Domain Name System* (DNS), dan lain-lain.

2. *Transport Layer*

   Layer ini menyediakan layanan pengiriman dari sumber data menuju ke tujuan data dengan cara membuat *logical connection* antara keduanya. *Layer* ini bertugas untuk memecah data dan membangun kembali data yang diterima dari *application layer* ke dalam aliran data yang sama antara sumber dan pengirim data. *Transport layer* juga menangani masalah *reliability, flow control,* dan *error correction. Layer* ini terdiri dari dua protokol yaitu *Transmission Control Protocol* (TCP) dan *User Datagram Protocol* (UDP). Protokol TCP memiliki orientasi terhadap reliabilitas data. Sedangkan protokol UDP lebih berorientasi kepada kecepatan pengiriman data.

3. *Internet Layer*

   Layer ini memiliki tugas utama untuk memilih rute terbaik yang akan dilewati oleh sebuah paket data dalam sebuah jaringan. Selain itu, *layer* ini juga bertugas untuk melakukan *packet switching* untuk mendukung tugas utama tersebut. *Layer* ini terdiri dari *Internet Protocol* (IP), *Internet Control Message Protocol* (ICMP), *Address Resolution Protocol* (ARP), dan *Reverse Address Resolution Protocol* (RARP).

4. *Network Access Layer*

Layer ini bertugas untuk mengatur semua hal-hal yang diperlukan sebuah paket IP agar dapat dikirimkan melalui sebuah medium fisik jaringan. Termasuk di dalamnya detil teknologi LAN dan WAN.

2.4.3.2 *Transmission Control Protocol* (TCP)

Berdasarkan Cisco Systems (2005a), TCP merupakan bagian dari protokol TCP/IP yang digunakan bersama dengan IP untuk mengirim data dalam bentuk unit-unit pesan antara komputer ke *internet*. Pengiriman data ini dapat terjamin karena TCP memiliki dua proses data *acknowledgement* dimana TCP selalu meminta konfirmasi setiap kali selesai mengirim data, apakah data telah sampai di tempat tujuan. Kemudian TCP akan mengirimkan data urutan berikutnya atau melakukan *retransmission* yaitu pengiriman ulang data tersebut. Data yang dikirim dan diterima diatur berdasarkan nomor urut. TCP juga mengawasi unit data individual atau dikenal dengan nama paket, dimana pesan-pesan dibagi untuk efisiensi *routing* melewati *internet*.

Protokol TCP bertanggung jawab untuk pengiriman data dari sumber ke tujuan dengan benar. TCP juga bertugas mendeteksi kesalahan atau hilangnya data dan melakukan pengiriman kembali sampai data yang benar diterima dengan

lengkap. TCP menyediakan pelayanan seperti *connection oriented*, *reliable*, *byte stream service*. *Connection oriented* berarti dua aplikasi pengguna TCP harus melakukan pembentukan hubungan dalam bentuk pertukaran kontrol informasi sebelum transmisi data terjadi untuk dapat melakukan pertukaran data tersebut. *Reliable* berarti TCP menerapkan proses deteksi kesalahan paket dan pengiriman ulang. *Byte stream service* berarti paket dikirimkan dan sampai ke tempat tujuan secara berurutan.

2.4.3.3 *Internet Protocol* (IP)

Berdasarkan Cisco Systems (2005a), IP adalah protokol yang berorientasi pada data, yang mengatur bagaimana data dikirim dari satu komputer ke komputer lain dalam suatu jaringan komputer. Setiap perangkat keras (*host*) yang berada dalam jaringan *internet* setidaknya mempunyai satu alamat IP yang bersifat unik yang membedakan dari *host* lain. Alamat IP terdiri dari 32 bit di mana dalam penulisannya IP dibagi menjadi 4 bagian. Masing-masing bagian terdiri dari 8 bit dan dibatasi dengan titik. Contoh: “202.155.89.17”. Pengalamatan IP terdiri dari 2 bagian yaitu bagian *network number* dan *host number*. Bit-bit *network* ditandai dengan angka *binary* 1 dan bit-bit *host* ditandai dengan angka *binary* 0. Pembagian bit-bit *network* dan *host* ini ditentukan dengan s*ubnet mask*. Contoh: “202.155.89.17 / 255.255.255.0”, menyatakan bahwa 24 bit pertama menyatakan

network dari IP *address* tersebut, dan sisanya 8 bit merupakan bit-bit *host* bagi IP tersebut. Dari IP tersebut dapat disimpulkan bahwa *network* dari IP tersebut adalah 202.155.89.0, sedangkan 202.155.89.17 merupakan alamat IP dari host tersebut. Beberapa teknologi dalam pengalamatan IP adalah IP *subnetting*, dan *Variable-Length Subnet Mask* (VLSM).

Pengalamatan IP terbagi dalam lima kelas (Cisco Systems, 2005a), yaitu :

1. Kelas A

   Kelas A merupakan kelas yang memiliki jumlah *host number* yang terbanyak, karena hanya 8 bit pertama digunakan sebagai bit-bit *network* dan sisanya 24 bit digunakan sebagai bit-bit *host*.. Kelas ini biasa digunakan oleh perusahaan yang memiliki jaringan dalam skala yang besar. Alamat IP pada kelas A dimulai dari 1.0.0.0 sampai dengan 126.255.255.255.

2. Kelas B

   Kelas B memiliki 16 bit pertama sebagai bit-bit *network* dan 16 bit sisanya digunakan sebagai bit-bit *host*. Alamat IP kelas B digunakan untuk jaringan dengan skala menengah. Alamat IP pada kelas B berkisar antara 128.0.0.0 sampai dengan 192.167.255.255.

3. Kelas C

Kelas C memiliki 24 bit pertama sebagai bit-bit *network* dan 8 bit sisanya digunakan sebagai bit-bit *host*. Kelas ini memiliki jumlah *host address* yang paling sedikit dan digunakan untuk jaringan dengan skala kecil. Alamat pada kelas C berkisar antara 192.168.0.0 sampai dengan 223.255.255.255.

4. Kelas D

   Kelas D merupakan kelas khusus yang tidak dapat dipakai oleh publik karena satu blok kelas ini khusus dipakai untuk keperluan *multicast*. *Multicast* adalah jenis transmisi layaknya *broadcast*, namun dalam skala yang lebih kecil dan tertentu.

5. Kelas E

   Kelas E adalah kelas IP yang tidak digunakan dan khusus disimpan dengan tujuan sebagai kelas cadangan untuk keperluan di masa mendatang.

Class A	**Network**	**Host**		
Octet	1	2	3	4

Class B	**Network**		**Host**	
Octet	1	2	3	4

Class C	**Network**			**Host**
Octet	1	2	3	4

Class D	**Host**			
Octet	1	2	3	4

Gambar 2.4 Struktur Kelas IP

Selain pembagian menurut alamat , alamat IP juga dibagi menjadi dua macam berdasarkan pemakaiannya di *internet* (Cisco Systems, 2005a):

1. *Private* IP *address*

   Private IP *address* merupakan alamat IP yang digunakan oleh sebuah komunitas, baik itu rumah ataupun sebuah perusahaan, yang tidak tersambung langsung ke *internet*. Alamat IP ini tidak bisa berkomunikasi langsung ke *internet*. Alamat IP ini tidak bisa berkomunikasi langsung dengan komputer lain pada jaringan *internet*, sehingga untuk dapat berkomunikasi dibutuhkan satu perantara yaitu *Internet Service Provider* (ISP) yang menyediakan jada layanan *internet*. IP yang tergolong *private* IP *address* adalah :

   Kelas A : 10.x.x.x

   Kelas B : 172.16.x.x s/d 172.31.x.x

   Kelas C : 192.168.x.x

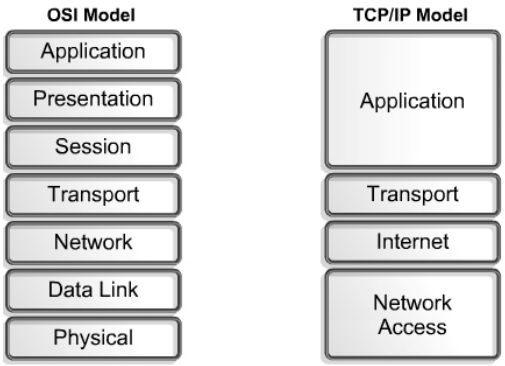
   Keterangan: x adalah nomor dimulai 0 sampai dengan 255.

2. *Public* IP *address*

   Public IP *address* adalah alamat IP yang digunakan untuk berkomunikasi antar komputer yang tersambung secara langsung dalam jaringan *internet*. Jenis IP address banyak digunakan oleh *Internet Service Provider* (ISP) dan lembaga-lembaga dunia yang mengatur lalu lintas di *internet*. Alamat *public* IP *address* adalah semua alamat IP

selain *private* IP *address* dan IP *loopback* (127.0.0.0 s/d 127.255.255.255).

2.4.4 Perbandingan OSI *Layer* dan TCP/IP *Layer*



Gambar 2.5 Perbandingan Model OSI *Layer* dan TCP/IP *Layer*

Persamaan antara model OSI dan model TCP/IP berdasarkan Cisco Systems (2005a) yaitu :

1. Keduanya memiliki *layer*.
2. Keduanya memiliki *application* layer, walaupun memiliki fungsi yang berbeda.
3. Keduanya memiliki *transport* layer dan *network layer* yang sebanding.
4. Keduanya harus diketahui oleh seorang *networking professional.*
5. Keduanya mengasumsikan paket sebagai pemilih. Ini berarti setiap paket dapat mengambil jalan yang berbeda-beda untuk mencapai tujuan yang sama. Ini bertolak belakang dengan *circuit-switched network* dimana semua paket mengambil jalan yang sama.

Menurut Cisco Systems (2005a), perbedaan antara model OSI dan TCP/IP:

1. TCP/IP menggabungkan *presentation layer* dan *session layer* OSI ke dalam *application layer*-nya.
2. TCP/IP menggabungkan *data link layer* dan *physical layer* OSI ke dalam *network access layer*.
3. TCP/IP lebih sederhana karena hanya memiliki 4 *layer*.
4. Protokol TCP/IP merupakan standar untuk pengembangan *internet*, sehingga model TCP/IP mendapatkan kredibilitas hanya karena protokolnya. Sebaliknya, jaringan tidak biasa dibangun dengan protokol OSI, meskipun OSI digunakan sebagai panduan.

2.5 Konsep *Routing*

2.5.1 *Path Determination*

Menurut Cisco Systems (2005a), *Path determination* terjadi pada i*nternet layer* model TCP/IP atau *network layer* model OSI. *Path determination* memungkinkan *router* mengevaluasi rute menuju alamat tujuan yang tersedia pada tabel *routing*. *Router* memgetahui rute tersebut melalui *static routing* atau *dynamic routing*. Rute yang dimasukkan secara manual oleh *administrator* adalah *static routes*. Rute yang didapat dari *router* lain melalui *routing protocol* adalah *dynamic routes*.

Router memakai *path determination* untuk memutuskan *port* mana yang akan dipakai untuk mengirim paket yang telah diterima ke tempat tujuan. Proses ini juga disebut sebagai paket *routing*. *Path*

determination dapat dimisalkan dengan seseorang yang sedang mengemudikan mobil dari satu kota ke kota yang lain. Pengemudi tersebut mempunyai peta yang dapat mengantarnya ke tempat tujuan, sama seperti *router* memiliki *routing table*. Pengemudi tersebut berjalan dari satu persimpangan ke persimpangan lain sama seperti paket yang berjalan dari *router* ke *router* lain pada setiap *hop* yang ada. Pada setiap persimpangan, pengemudi tersebut dapat menentukan rutenya sendiri dengan memilih kiri, kanan atau lurus. Sama halnya sebuah *router* memutuskan *port* mana yang dipakai untuk mengirimkan paket.

Keputusan pengemudi dipengaruhi oleh berbagai faktor seperti kepadatan lalu lintas, batas kecepatan di jalan, lebar jalur yang ada, adakah jalur bebas hambatan, atau apakah jalan yang ada sering ditutup. Terkadang lebih cepat mengambil rute yang panjang dengan jalan yang kecil tetapi tidak padat, daripada jalan yang besar dan padat. Mirip halnya dengan *router* yang mengambil keputusan *routing* berdasarkan *load*, *bandwith*, *delay*, *cost*, serta *realibility network link*.

2.5.2 *Routing*

Routing (Cisco Systems, 2005a) adalah proses yang digunakan sebuah *router* untuk meneruskan paket data ke alamat tujuan. Sebuah *router* mengambil keputusan penentuan jalur berdasarkan alamat tujuan dari paket data. Dalam mengambil keputusan yang tepat, *router* harus mempelajari rute *network* tujuan. Bila *router* dikonfigurasi menggunakan *dynamic routing*, *router* mendapatkan informasi *routing* dari *router-router*

sekitarnya. Bila *router* dikonfigurasi menggunakan *static routing*, *router* meneruskan paket data mengacu kepada *routing table* yang dimilikinya.

2.5.3 *Routed Protocol*

(Cisco System, 2005a) Protokol yang digunakan pada *network layer* untuk mengirim data dari satu *host* ke yang lainnya melalui *router* disebut *routed* atau *routable protocol*. *Routed protocol* memindahkan data melewati *network*.

Fungsi *routed protocol* meliputi:

1. Menyediakan informasi yang cukup dalam *network layer address* untuk memungkinkan *router* meneruskan ke alat jaringan berikutnya terutama ke tujuannya.
2. Menjelaskan format dan kegunaan suatu *fields* dalam paket data.

Internet Protocol (IP) dan Novell's *Internetwork Packet Exchange* (IPX) adalah contoh dari *routed protocols*. Contoh lainnya meliputi DECnet, AppleTalk, Banyan VINES, and Xerox *Network Systems* (XNS). IP adalah skema pengalamatan hirarki *network* yang paling banyak dipakai.

2.5.4 *Routing Protocol*

(Cisco Systems, 2005a) Router menggunakan *routing protocol* untuk bertukar informasi *routing table*. Dengan kata lain, *routing protocol* memungkinkan *router* untuk mengarahkan *routed protocol*. *Routing protocol* memungkinkan *router* untuk memilih jalur terbaik bagi data dari sumber ke tujuan.

Routing protocol mempunyai fungsi:

1. Sebagai proses untuk berbagai informasi *routing*
2. Memungkinkan *router* berkomunikasi dengan *router* lain untuk men-*update* dan me-*maintain routing table*.

Contoh dari *routing protocol* yang mendukung IP *routed protocol* meliputi *Routing Information Protocol* (RIP), *Interior Gateway Routing Protocol* (IGRP), *Open Shortest Path First* (OSPF), *Border Gateway Protocol* (BGP), dan *Enhanced* IGRP (EIGRP).

2.5.5 Pembagian *Routing Protocol*

Menurut Cisco Systems (2005a), *Routing protocol* dapat diklasifikasikan sebagai IGP atau EGP, yang menjelaskan apakah suatu grup *router* berada dibawah satu administrasi *Autonomous System* (AS) atau tidak. IGP dapat dikategorikan lagi menjadi *distance-vector* atau *link-state protocol*.

Distance-vector routing menentukan jarak, arah dan *vector* ke setiap *link* pada *internetwork*. Jarak dapat berupa *hop count* ke suatu *link*. *Router* dengan algoritma *distance-vector* mengirim semua atau sebagian *routing table* ke *router* yang bersebelahan dengan dirinya (*router* tetangga) dalam suatu periode tertentu walaupun tidak ada perubahan pada *network*.

Contoh *distance-vector protocol* adalah :

1. *Routing Information Protocol* (RIP) – IGP yang paling umum pada internet, RIP menggunakan *hop count* sebagai satu-satunya *routing metric*.

2. *Interior Gateway Routing Protocol* (IGRP) – Dikembangkan oleh Cisco untuk mengatasi masalah *routing* pada *network* yang besar dan heterogen.
3. *Enhanced IGRP* (EIGRP) – Cisco *proprietary* IGP, mempunyai banyak fitur *link-state routing protocol*. Oleh karena itu disebut sebagai *balanced-hybrid protocol*, tetapi sebenarnya adalah *advanced distance-vector routing protocol*.

Link-state routing protocol didesain untuk mengatasi keterbatasan *distance-vector routing protocol*. *Link-state routing protocol* menanggapi dengan cepat perubahan *network* dengan mengirimkan *link-state advertisement* (LSA) hanya pada saat terjadi perubahan. LSA tersebut dikirimkan ke semua *router* yang ada pada *network*. Algoritma *link-state* biasanya menggunakan *database* yang ada untuk membuat *routing table entries* yang menggambarakan jalur terpendek (*shortest path*).

Contoh *link-state* routing protocol adalah :

1. *Open Shortest Path First* (OSPF). Merupakan *link-state routing protocol* yang dikembangkan oleh *Internet Engineering Task Force* (IETF) pada tahun 1988.
2. *Intermediate System-to-Intermediate System* (IS-IS). Merupakan *link-state routing protocol* yang digunakan untuk *routed protocol* selain IP.

2.5.6 *Routing Table*

Routing table (Cisco Systems, 2005a) berisi berbagai informasi yang diperlukan untuk dapat meneruskan paket data antar *network*. Informasi tersebut didapat dari *routing protocol* ataupun *static routing* dan akan bervariasi tergantung pada *routing protocol* yang digunakan. *Routing table* dibuat dan di-*maintain* dengan menganalisa *routing update*.

2.5.7 *Routing Metric*

Setiap *routing protocol* mempunyai *routing metric* yang berbeda yang digunakan untuk menentukan jalur terbaik ke alamat tujuan (Cisco Systems, 2005a). Algoritma *routing* menghasilkan suatu angka, disebut *metric value*, untuk setiap jalur yang ada. Nilai yang kecil menentukan jalur yang akan dipilih. *Routing metric* dapat berdasarkan satu karakteristik jalur, atau kalkulasi dari beberapa karakteristik. *Metric* yang sering dipakai oleh *routing protocol* yaitu *bandwidth*, *delay*, *load*, *reliability*, *hop count*, *ticks*, dan *cost* .

2.6 Cisco *Internetwork Operating System* (IOS)

2.6.1 Pengenalan Cisco IOS

(Cisco Systems, 2005b) Teknologi Cisco dibuat berdasar Cisco *Internetwork Operating System* (IOS). Cisco IOS adalah sistem operasi yang mengatur fungsi *routing* dan *switching* dari alat-alat jaringan.

Sama seperti suatu komputer, *router* atau *switch* tidak dapat berfungsi tanpa sistem operasi, dimana sistem operasi pada alat-alat

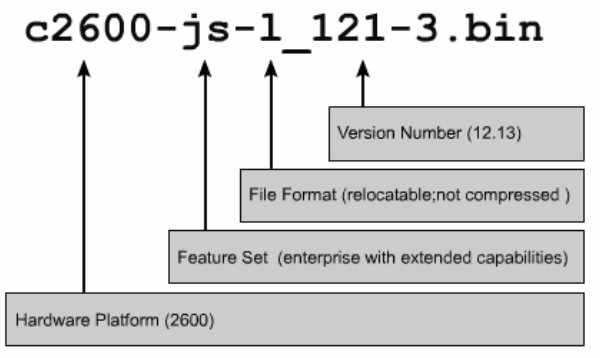
jaringan Cisco adalah Cisco IOS. Cisco IOS memiliki arsitektur *embedded-operating-system* pada semua *router* Cisco dan juga merupakan sistem operasi dari *switch* Cisco Catalyst. Tanpa sistem operasi, perangkat keras tidak mempunyai satu kemampuan apapun. Berikut ini adalah servis jaringan disediakan Cisco IOS berdasarkan Cisco Systems (2005b) :

- Fungsi dasar *routing* dan *switching*.
- Akses ke *network* re*source* yang *reliable* dan aman.
- *Network scalability*.

Cisco menyediakan IOS *images* untuk berbagai alat jaringan pada berbagai *platform* produk jaringan. Untuk mengoptimalkan Cisco IOS yang diperlukan di berbagai *platform*, Cisco berupaya mengembangkan barbagai macam Cisco IOS *software image*, dimana setiap *image* menawarkan fitur yang berbeda untuk berbagai *platform*, *memory resource* yang tersedia, serta kebutuhan pelanggan.

Walaupun terdapat berbagai IOS *images* untuk model alat jaringan Cisco yang berbeda, struktur perintah dasar konfigurasi adalah sama. Berdasarkan Cisco Systems (2005b) tata cara penamaan untuk membedakan Cisco IOS terdiri dari bagian-bagian :

- *Platform* dimana IOS *image* akan berjalan.
- Fitur khusus yang ada pada IOS *image*
- File format, apakah terkompresi atau *relacatable*.
- Nomor versi.



Gambar 2.6 Tata Cara Penamaan Cisco IOS

2.6.2 *Command-line interface* (CLI)

Berdasarkan Cisco Systems (2005b), Cisco IOS *software* menggunakan *command-line interface* (CLI) sebagai lingkungan console tradisional. CLI dapat diakses melalui beberapa metode. Salah satu cara untuk mengakses CLI adalah melalui console *session*. Suatu console menggunakan koneksi serial langsung yang berkecepatan rendah dari komputer atau terminal ke koneksi console yang terdapat pada *router*. Salah satu cara lain mengakses CLI yaitu dengan menggunakan koneksi *dial-up* memakai *modem* atau *null modem* yang dihubungkan ke *port* AUX pada *router*. Kedua cara tersebut tidak memerlukan *router* untuk memiliki konfigurasi *network services*. Metode lain untuk mengakses CLI *session*s adalah dengan *telnet* ke *router*. Untuk membangun suatu sesi *telnet* ke *router*., setidaknya satu *interface router* harus dikonfigurasi dengan *IP address*, dan *virtual terminal sessions* harus dikonfigurasi untuk *login* dan *password*.

Gambar 2.7 Gambar user interface untuk router atau switch

2.6.3 *Configuration Mode* Cisco IOS

Cisco CLI menggunakan struktur yang hirarki (Cisco Systems, 2005b). Struktur ini membutuhkan *user* untuk masuk ke beberapa mode tertentu untuk melakukan suatu tugas tertentu. Contohnya, untuk mengkonfigurasi *router interface*, *user* harus masuk ke *interface configuration mode*. Pada *interface configuration mode*, semua konfigurasi yang dilakukan hanya berlaku pada *interface* tersebut. Setiap mode ditandai dengan suatu *prompt* yang khusus dan hanya memperbolehkan perintah tertentu yang terdapat pada mode tersebut.

IOS menyediakan *command interpreter services* yang dikenal dengan *command executive* (EXEC). Setelah setiap perintah dimasukkan, EXEC akan memvalidasi dan mengeksekusi perintah tersebut.

Sebagai fitur keamanan dari Cisco, IOS *software* membagi EXEC *session*s menjadi dua level akses.

- *User* EXEC mode. Hanya memperbolehkan perintah monitoring dasar yang sangat terbatas. Sering disebut sebgai “view only” mode.

User EXEC level tidak memperbolehkan semua perintah yang dapat merubah konfigurasi *router*. *User* EXEC mode dapat diidentifikasi dengan “>” *prompt*.

- *Priviledged* EXEC mode. Mode yang dapat mengakses semua perintah *router*, Mode ini dapay dikonfigurasi untuk meminta *password* sebelum *user* dapar mengakses. Sebagai tambahan keamanan, mode ini juga dapat dikonfigurasi untuk meminta *user* ID. Dengan begitu, hanya *user* yang berwenang yang dapat mengakses *router*. Perintah untuk konfigurasi dan manajemen memerlukan administrator jaringan berada pada *priviledged* EXEC level. *Global configuration mode* dan semua mode konfigurasi yang spesifik hanya dapat dicapai melalui *priviledged* EXEC mode. *Priviledged* EXEC mode dapat diidentifikasi dengan “#” *prompt*.

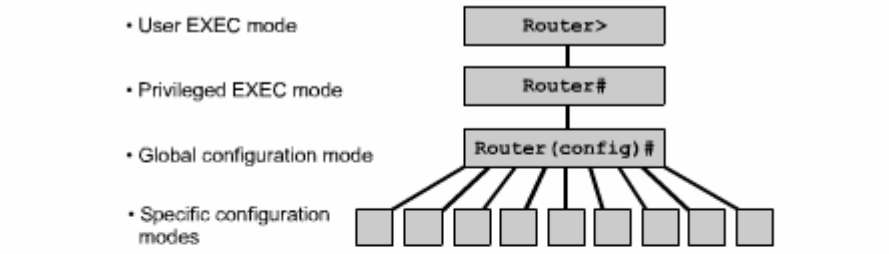
Pada lingkungan CLI, apabila suatu *password* dikonfigurasi dan *router* meminta *password* tersebut, sebagai keamanan maka Cisco *networking device* tidak akan memperlihatkan *password* yang dimasukkan.

Semua perubahan konfigurasi di CLI pada Cisco *router* dilakukan pada *global configuration mode*. *Global configuration mode*, sering disingkat *global config*, adalah mode konfigurasi yang utama. Perintah pada *global configuration mode* digunakan di *router* untuk melakukan konfigurasi yang mempengaruhi sistem secara keseluruhan.

Mode yang lebih spesifik diperlukan untuk perubahan konfigurasi yang lain. Mode khusus tersebut adalah bagian dari *global configuration*

mode. Berdasarkan Cisco Systems (2005b), beberapa mode yang dapat diakses melalui *global configuration mode* adalah :

- *Interface mode*
- *Line mode*
- *Router mode*
- *Subinterface mode*
- *Controller mode*



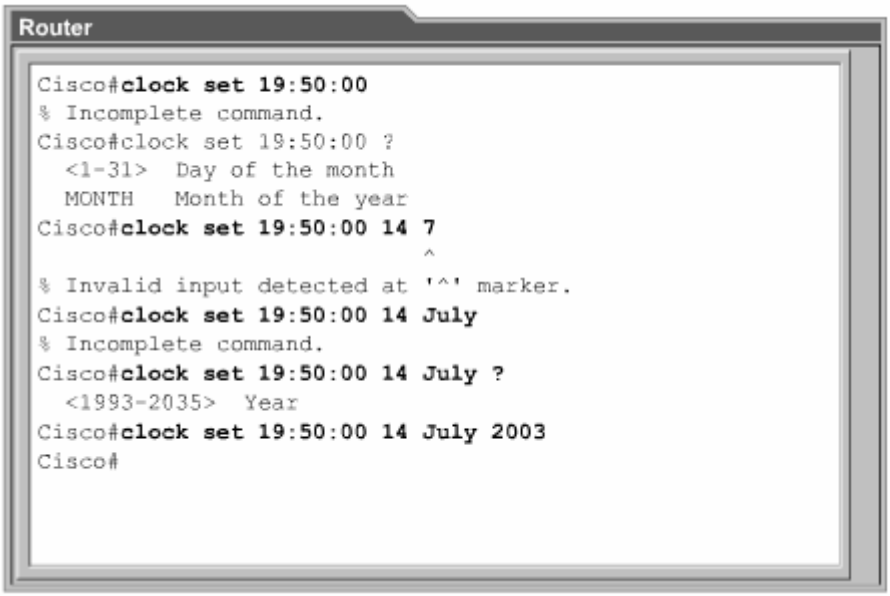
Configuration Mode	Prompt
Interface	`Router(config-if)#`
Subinterface	`Router(config-subif)#`
Controller	`Router(config-controller)#`
Map-list	`Router(config-map-list)#`
Map-class	`Router(config-map-class)#`
Line	`Router(config-line)#`
Router	`Router(config-router)#`
IPX-router	`Router(config-ipx-router)#`
Route-map	`Router(config-route-map)#`

Gambar 2.8 Struktur hirarki mode konfigurasi pada CLI Cisco IOS

Pada saat memasuki mode khusus tersebut, *router prompt* akan berubah untuk menandakan mode konfigurasi yang sedang dijalankan. Semua perubahan konfigurasi yang dilakukan hanya berlaku untuk *interface* atau proses yang terdapat pada mode tersebut.

2.6.4 *Error - Help* pada Cisco IOS

(Cisco Systems, 2005b) *Command* line *error* sering muncul terutama karena kesalahan pengetikan. Apabila suatu *keyword* perintah salah dalam penulisannya, *user interfaces* pada IOS (CLI) menyediakan suatu bentuk penanda *error* yang berupa tanda (^). Simbol "^" muncul pada tempat yang terdapat kesalahan dalam suatu baris perintah, *keyword* atau argument yang ditulis. Penanda lokasi *error* dan sistem *help* yang interaktif membantu *user* untuk menemukan dan membetulkan *error* sintaks. Dengan memasukkan tanda tanya (“?”) maka akan diperlihatkan semua pilihan perintah yang tersedia pada mode tersebut.



Gambar 2.9 *Command line error indicator* dan *help response* pada CLI Cisco IOS

2.7 Cisco *Discovry Protocol* (CDP)

Menurut Cisco Systems (2005b), Cisco Discovery Protocol (CDP) adalah protokol pada *layer* 2 OSI yang menghubungkan media fisik di *layer* bawah dengan *network layer protocol* diatas, dan merupakan *proprietary* dari Cisco. CDP

digunakan untuk mendapatkan informasi mengenai alat jaringan tetangga, seperti tipe ala-alat jaringan, terkoneksi dengan *interfaces* apa, dan nomor model. CDP merupakan media dan protokol yang *independent*, dan hanya berjalan pada semua peralatan Cisco melalui *Subnetwork Access Protocol* (SNAP). Kegunaan utama CDP adalah untuk mengetahui alat jaringan Cisco yang terhubung dengan alat jaringan lokal, dimana keduanya menjalankan CDP.

2.8 *Access Control List*

Berdasarkan Cisco Systems (2005b), *Access control list* (ACL) adalah salah satu kemampuan *router* untuk menyaring data yang melalui *router*. ACL berupa sederetan aturan yang menyatakan paket data diizinkan lewat atau ditolak. ACL dapat diterapkan ke semua *routed network protocol*, seperti *Internet Protocol*(IP) dan *Internet Packet Exchange*(IPX). ACL dikonfigurasi di *router*.

Penyaringan ACL akan menentukan suatu paket akan diizinkan lewat atau akan diblok. Router akan memutuskan hal tersebut dengan memeriksa paket tersebut dan mencocokannya dengan serangkaian aturan pada ACL. Yang diperiksa oleh ACL meliputi alamat sumber, alamat tujuan, protokol, dan nomor *port*.

Untuk mengatur lalu lintas jaringan, diperlukan penerapan ACL yang terpisah antara arah masuk dan arah keluar (*inbound* dan *outbound*), juga harus diterapkan pada tiap *interface* yang dimiliki oleh *router*. Jadi jumlah ACL yang dapat diterapkan pada *router* berjumlah dua untuk arah masuk dan arah keluar dikali jumlah *interface* yang ada (dua kali jumlah *interface*).

Tujuan dari penggunaan ACL menurut Cisco Systems (2005b) adalah:

1. Membatasi lalu lintas jaringan dan meningkatkan kinerja jaringan.
2. Menyediakan fasilitas pengaturan lalu lintas jaringan.
3. Menyediakan keamanan jaringan tingkat dasar (*basic level*).
4. Memutuskan tipe-tipe lalu lintas mana yang akan diblok dan yang akan diteruskan.
5. Memperbolehkan *administrator* untuk mengontrol hak akses *client* di dalam jaringan.
6. Bila ACL tidak diterapkan pada sebuah *router*, maka semua paket akan diteruskan dengan kata lain tidak ada penyaringan terhadap paket data.

ACL harus digunakan pada *router firewall*, yang biasanya diletakan di antara jaringan *eksternal* dan jaringan *internal*. Hal ini dilakukan untuk memberikan keamanan terhadap jaringan *internal*.

2.9 *Virtual Local Area Network*

Teknologi *Virtual Local Area Network* (VLAN) memungkinkan pembagian satu *physical network* menjadi beberapa *logical network* (Cisco Systems, 2005c). VLAN merupakan teknologi di *data link layer* pada *OSI layer*. Dengan adanya teknologi VLAN, *switch* mampu melakukan segmentasi LAN seperti halnya sebuah *router*, namun tidak memiliki kemampuan me-*routing* paket data, sehingga dalam melakukan *inter*-VLAN *routing* tetap dibutuhkan sebuah *router*.

Setiap segmen yang terbentuk memiliki id-nya tersendiri untuk membedakan VLAN yang satu dengan yang lainya. Secara *default*, semua *port* di *switch* Cisco merupakan VLAN 1 (id satu) yang merupakan VLAN manajemen *switch*. Implementasi VLAN dapat dilakukan dengan 2 metode, yaitu *static* VLAN

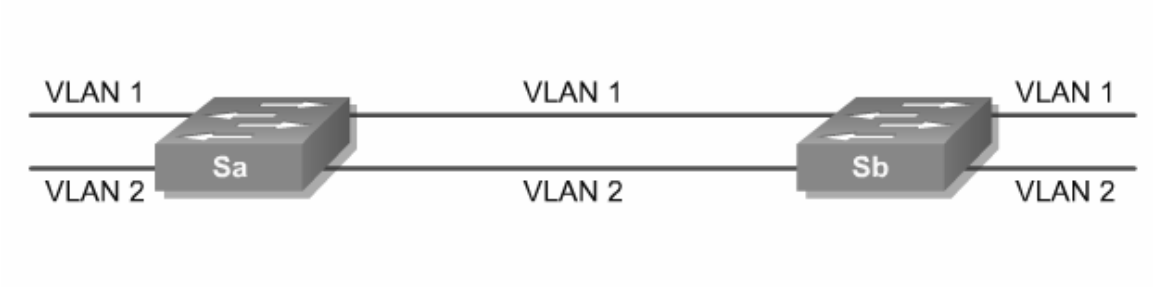
dan *dynamic* VLAN. *Static* VLAN sering disebut juga dengan *port-based* VLAN dimana *administrator* menentukan *port-port* mana saja yang di-*assign* ke VLAN id tertentu. Dengan demikian bila sebuah komputer terhubung ke jaringan melalui *port switch* tersebut, akan menjadi anggota dari VLAN id yang telah ditentukan. Sedangkan *dynamic* VLAN ditentukan berdasarkan MAC *address* atau protokol yang digunakan komputer dalam jaringan.

Berdasarkan Cisco Systems (2005b), VLAN menawarkan banyak keuntungan, seperti:

1. Kemudahan dalam memindahkan *workstation* tanpa merubah topologi yang telah ada
2. Kemudahan dalam menambah *workstation* baru
3. Kemudahan dalam mengatur trafik jaringan
4. Meningkatkan keamanan jaringan

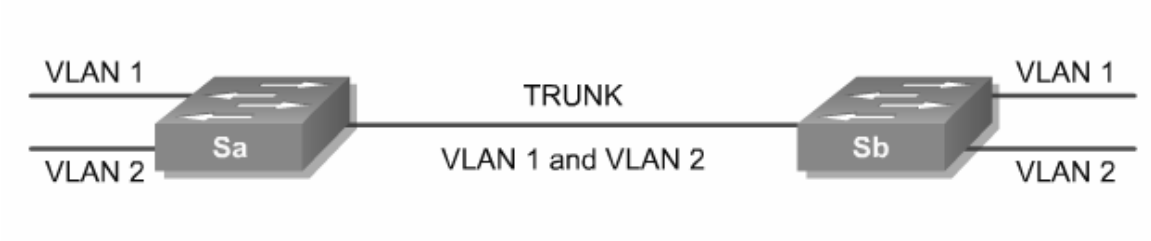
2.10 *Trunking*

Dalam melakukan komunikasi sesama VLAN antar *switch*, cara paling sederhana yang dapat ditempuh adalah dengan menghubungkan *switch* pertama dengan *switch* kedua dimana kedua *port* yang terhubung memiliki VLAN-*id* yang sama. Gambar berikut menampilkan topologi fisik dalam melakukan komunikasi sesama VLAN antar *switch* (Cisco Systems, 2005c).



Gambar 2.10 Topologi Fisik Komunikasi Sesama VLAN Antar *Switch*

Setiap *link* yang terbentuk hanya dapat dilalui oleh VLAN-*id* tertentu. Dengan demikian bila *switch* terbagi 2 VLAN yang berbeda, maka dibutuhkan 2 *port switch* untuk menghubungkan *switch* tersebut dengan *switch* lainnya. Jika *switch* terbagi 4 VLAN yang berbeda, maka dibutuhkan 4 *port switch* untuk menghubungkan *switch* tersebut dengan *switch* lainnya. Untuk mengatasi pemborosan *port* dalam melakukan komunikasi VLAN antar *switch*, digunakanlah konsep *trunking*. Pada mulanya konsep *trunking* diimplementasikan dalam penyiaran radio dimana teknologi ini mampu mentransmisikan beberapa channel dengan frekuensi berbeda melalui satu *line* komunikasi. Dalam penerapannya pada komunikasi VLAN antar *switch*, *trunking* berlaku sebagai media penghubung beberapa VLAN sekaligus dalam satu *line*. *Trunking* menggunakan metode *frame tagging* dalam pengiriman paket data. Setiap paket data disisipkan VLAN-*id* untuk membedakan VLAN. Gambar berikut menampilkan topologi fisik dalam penggunaan *trunking* sebagai media komunikasi VLAN antar *switch* (Cisco Systems, 2005).



Gambar 2.11 Topologi Fisik Komunikasi Sesama VLAN Antar *Switch* Dengan *Trunking*

2.11 VLAN *Trunking Protocol*

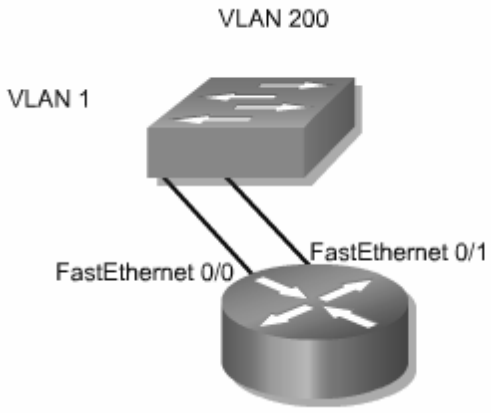
(Cisco Systems, 2005c) Setiap *switch* Cisco memiliki VLAN *database*-nya masing-masing untuk menyimpan informasi tentang VLAN yang diterapkan di *switch* tersebut. Pada mulanya, setiap *switch* Cisco dikonfigurasi satu per satu sedemikian rupa sehingga VLAN *database* pada setiap *switch* adalah sama. Seiring perkembangan jaringan yang kian membesar, *switch* Cisco juga semakin bertambah dan konfigurasi tetap dilakukan secara manual dan satu per satu. Hal ini dapat menimbulkan kelalaian dalam mengkonfigurasi *switch-switch* yang kian membanyak. Bila terjadi penambahan satu VLAN, sang *administrator* harus kembali mengkonfigurasi setiap *switch-switch* secara manual dan satu per satu. Untuk mengatasi masalah ini, digunakanlah teknologi VLAN *Trunking Protocol* (VTP). VTP memberikan sarana pemberitaan informasi VLAN ke *switch* Cisco yang berada di dalam *domain* manajemen yang sama. Satu *switch* Cisco hanya memiliki satu nama *domain*.

Berdasarkan Cisco Systems (2005c), VTP terbagi atas 3 jenis yaitu *server*, *client*, atau transparan. VTP *server* dapat memodifikasi VLAN *database* dan mengirimkan informasi VLAN ke semua *port trunking* di *switch* Cisco. VTP *client* tidak dapat memodifikasi VLAN *database*. VTP *client* hanya berfungsi menerima informasi VLAN dari VTP *server*, meng*update* VLAN *database* dan mengirimkan

informasi VLAN *database* kembali ke semua *port trunking* di *switch* Cisco. VTP transparan tidak dapat meng-*update* informasi VLAN yang diterima dari VTP *server*. VTP transparan hanya meneruskan informasi VLAN ke semua *port trunking* di *switch* Cisco.

2.12 *Inter*-VLAN *Routing*

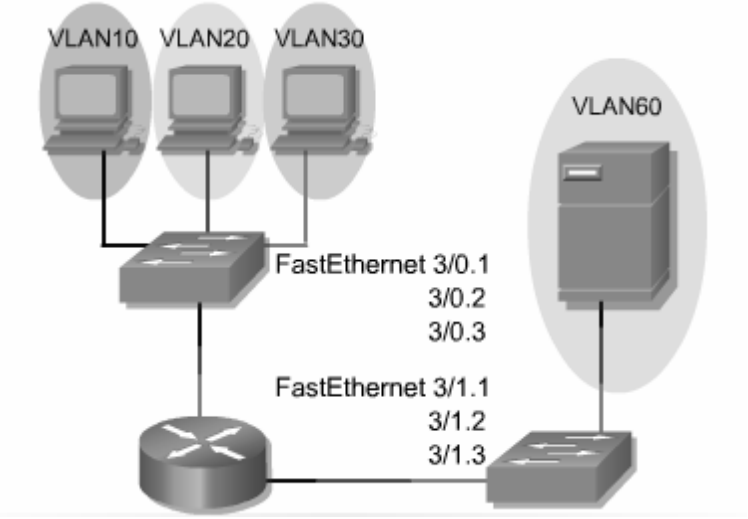
(Cisco Systems, 2005c) VLAN merupakan teknologi segmentasi jaringan di *data link layer* dan dalam melakukan komunikasi antar VLAN dibutuhkan alat jaringan berbasis *layer network* yaitu *router*. Komunikasi antar VLAN dapat ditempuh dengan menghubungkan *switch* dengan *router* per VLAN. Bila di *switch* terbagi menjadi 2 VLAN, maka jumlah *line* yang dihubungkan ke *router* adalah dua mewakili masing-masing VLAN. Gambar berikut menampilkan topologi fisik dalam melakukan komunikasi antar VLAN menggunakan *router*.



Gambar 2.12 Topologi Fisik Komunikasi Antar VLAN Menggunakan *Router*

Hal ini tentu saja tidak efisien jika *switch* terbagi jumlah VLAN yang banyak, karena akan memboroskan penggunaan *port-port* di *switch* dan *router*. Untuk menghindari hal tersebut, digunakan teknologi *trunking*. Di sisi *router*,

setiap *interface* mampu mendukung banyak *sub-interfaces* untuk membedakan masing-masing VLAN. Gambar berikut menampilkan penggunaan topologi fisik dalam melakukan komunikasi antar VLAN menggunakan router dengan *trunking* (Cisco Systems, 2005).



Gambar 2.13 Topologi Fisik Komunikasi Antar VLAN Dengan *Trunking* dan Media *Router*

2.13 Apache *Web Server*

Apache (Anonymous, 2005a) adalah *web server* yang didistribusikan secara gratis dibawah NCSA (National Center for Computing Application) HTTPD. Apache mulai dikembangkan karena beberapa masalah keamanan yang terdapat pada NCSA *server* dan pemikiran untuk mendapatkan UNIX *web server* dengan fitur khusus yang gratis. Apache dibentuk oleh sekitar 20 *programmer*, dinamakan grup Apache, yang mengembangkan kumpulan *patches* untuk NCSA *server*. Nama Apache muncul dari "A PAtCHy *server*". Apache didesain untuk kecepatan, realibilitas dan memperbaiki masalah keamanan di NCSA HTTPD. Versi awal Apache dirilis untuk *platform* UNIX, akan tetapi sekarang Apache dapat ditemui di berbagai *platform* lainnya, seperti *platform* Windows.

2.14 MySQL

MySQL adalah *structured query language database server*. Tata bahasa pemrograman SQL di aplikasi MySQL menggunakan tata bahasa (*syntax*) SQL standar. MySQL digunakan karena memilki kecepatan yang baik, reliabilitas dan kemudahan.

MySQL mengimplementasikan konsep *client-server* yang terdiri dari *daemon* mysqld dan beragam jenis aplikasi *client* dan *library*.

MySQL adalah sumber perangkat lunak yang terbuka (*Open Source Software*). *Open Source Software* dapat diartikan bahwa ada kemungkinan untuk pemakai menambah atau memodifikasi program tanpa ada kewajiban membayar royalti.

2.15 Macromedia Flash

Menurut Hakim (2004, pp1-3), Flash merupakan program animasi profesional yang digunakan untuk membuat animasi, dari animasi sederhana sampai animasi kompleks, meliputi multimedia dan aplikasi *web* yang dinamis dan interaktif seperti *e-commerce* atau toko *on-line* di *internet*.

Animasi atau *movie* Flash terdiri dari grafik, teks, animasi dan aplikasi untuk situs *web*, serta video, gambar, dan suara yang diimpor dari aplikasi diluarnya. *Movie* Flash juga bisa memasukkan unsur interaktif dengan Actionscript (bahasa pemrograman di Flash) sehingga user dapat berinteraksi dengan *movie* melalui *keyboard* atau *mouse*.

2.16 PHP Hypertext Preprocessor

Berdasarkan Welling dan Thomson (2001, p2), PHP adalah sebuah bahasa *server-side scripting* yang didesain khusus untuk *web*. Didalam halaman HTML, dapat disertakan kode PHP yang akan dijalankan setiap kali halaman tersebut dibuka. Kode PHP diinterpretasikan di *web server* dan menghasilkan HTML atau *output* lain. PHP adalah produk *open source*. *Source code* PHP dapat diakses, digunakan, dirubah, dan didistribusikan kembali tanpa dikenakan biaya. Pada awalnya PHP adalah singkatan dari Personal Home Page, tetapi kemudian dirubah pada salah satu baris GNU recursive naming convention dan sekarang menjadi PHP Hypertext Preprocessor.

2.17 *State Transition Diagram* (STD)

2.17.1 Pengertian *State Transition Diagram* (STD)

STD digunakan untuk menggambarkan sifat dinamis dari suatu objek. STD mengilustrasikan berbagai *state* yang dimiliki suatu objek, *event* yang menyebabkan perubahan *state*, serta aturan yang ada untuk transisi antar *state* pada suatu objek. Dengan kata lain, STD menjelaskan *state* apa dari objek yang dapat melakukan transisi ke *state* lain (Whitten, Bentley, Dittman, 2001, p655 dan p668).

2.17.2 Pembuatan *State Transition Diagram* (STD)

Ada dua cara pembuatan STD, yaitu :

1. Perhatikan semua *state* yang mungkin muncul dan perhatikan semua hubungan yang mungkin diantara *state-state* tersebut.

2. Mulai dari *state* awal, perhatikan *state* apa yang dapat diteruskan darinya, kemudian dilanjutkan sampai semua jaringan tergambarkan.

Dalam pembuatan STD terdapat beberapa aturan, yaitu :

1. Identifikasi semua *state* yang mungkin, atau *state initial*
2. Gambarkan kotak untuk mewakili setiap *state*.
3. Hubungkan *state* satu dengan *state* lain dengan anak panah untuk memperlihatkan transisi.
4. Setiap *state* harus menuju ke *state* lainnya.
5. Namakan panah transisi dengan *event* yang menyebabkan transisi tersebut terjadi.
6. Buatlah aksi yang ada pada setiap kotak.
7. Pikirkan kemungkinan sistem akan *event* yang tidak terduga.
8. Pelajari diagram untuk melihat apakah perlu diuraikan.
9. Lakukan pembahasan untuk kecepatan dan konsisten
    a. Apakah semua *state* telah tergambarkan.
    b. Apakah semua *state* dapat dituju.
    c. Apakah semua *state* dapat menuju ke *state* lain (kecuali *final state*).
    d. Apakah sistem berjalan normal pada semua kejadian.

2.18 *Entity Relationship Diagram* (ERD)

2.18.1 Pengertian *Entity Relationship Diagram* (ERD)

ERD (Whitten, Bentley, dan Dittman, 2001, p260) merupakan suatu model yang menggambarkan data yang ada dalam bentuk *entity*, serta hubungan yang ada antar *entity* tersebut. Komponen utama yang terdapat pada ERD (Pressman, 2001, p307): objek data (*entity*), atribut, hubungan (*relationship*), dan berbagai tipe indikator lainnya.

2.18.2 Hubungan data pada *Entity Relationship Diagram* (ERD)

Hubungan antar data pada ERD, digambarkan dengan berbagai simbol yang menunjukkan *cardinality* (Whitten, Bentley, dan Dittman, 2001 pp264-265; Pressman, 2001, pp305-307). *Cardinality* merupakan jumlah minimum dan maksimum objek data yang berelasi dengan suatu objek data yang lain. Hubungan yang mungkin ada yaitu :

- *One-to-one* (1:1) – suatu objek A berhubungan dengan satu dan hanya satu objek B, dan objek B berhubungan dengan satu objek A.
- *One-to-many* (1:m) – satu objek A dapat berhubungan dengan satu atau banyak objek B, tetapi objek B hanya berhubungan dengan satu objek A. Contoh : seorang ibu dapat mempunyai banyak anak, tetapi satu anak hanya mempunyai satu ibu.
- *Many-to-many* (m:m) – suatu objek A dapat berhubungan dengan satu atau lebih objek B, sedangkan objek B dapat berhubungan dengan satu atau lebih objek A. Contoh : seorang paman dapat mempunyai

beberapa keponakan, dan seorang keponakan dapat mempunyai banyak paman.

Adapun notasi *cardinality* yang umum adalah sebagai berikut :

Tabel 2.1 Notasi *Cardinality* ERD

Arti *Cardinality*	Minimum *instances*	Maksimum *instances*	Notasi
Satu dan hanya satu	1	1	
Nol atau satu	0	1	
Satu atau lebih	1	*Many* (>1)	
Nol atau lebih	0	*Many* (>1)	
Lebih dari satu	>1	>1	