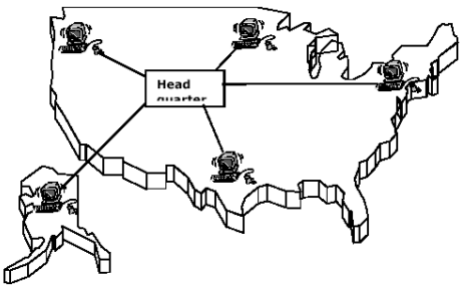
BAB II

LANDASAN TEORI

2.1 Manajemen Jaringan

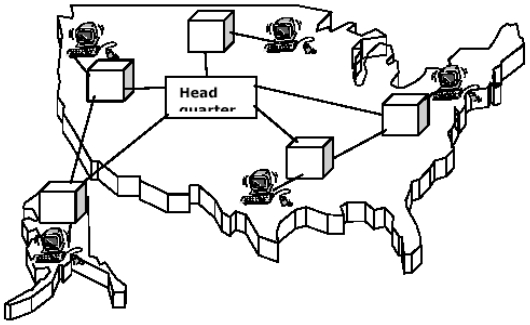
Pada suatu jaringan besar manajemen jaringan sangat dibutuhkan, karena manajemen jaringan memiliki kemampuan untuk memonitor, mengontrol serta merencanakan sistem dan jaringan itu sendiri. Dalam melakukan penerapan jaringan komputer, para manajer jaringan mengandalkan berbagai macam peralatan. Manajemen jaringan dapat diimplementasikan dengan menggunakan berbagai macam arsitektur yang didasarkan pada tipe dan ukurannya masing-masing. Pada dasarnya ada dua arsitektur yang dapat digunakan, manajemen terpusat *(centralized management) dan* manajemen menyebar *(distributed management).*

Manajemen Terpusat adalah arsitektur jaringan yang terpusat bersandar pada informasi dan kontrol untuk muncul pada sebuah lokasi tunggal yang tersentralisasi atau terpusat. Hal ini menyederhanakan manajemen jaringan yang ukurannya besar. Oleh karena *monitoring* dan kontrol terletak pada satu titik, maka manajer jaringan harus terletak pada titik yang sama. Supaya bisa memantau jaringan dengan efektif, maka stasiun manajemen yang terpusat harus menyelidiki peralatan/perlengkapan jaringan yang ditemukan dalam masing-masing jaringan. Hal ini memperkenalkan lalu lintas manajemen pada hubungan-hubungan *internetwork* (antar jaringan).



Gambar 2.1. Manajemen Terpusat

Manajemen Terdistribusi bertolak belakang dengan model terpusat, model terdistribusi menyebarkan informasi dan kontrol sedemikian sehingga masing-masing jaringan yang tersebar secara geografis bertanggung jawab bagi dirinya sendiri. Dalam sebuah model terdistribusi, manajemen ditekan ke bawah hingga mencapai tingkat yang terendah. Dalam model ini, kita memiliki manajer-manajer yang bertanggung jawab bagi daerah kekuasaan manajemennya masing-masing. Informasi penting yang berhubungan dengan keseluruhan antar jaringan diteruskan oleh manajer dari kumpulan manajer. Model ini baik sekali dalam mengembangkan jaringan yang berukuran besar.



Gambar 2.2. Manajemen Terdistribusi

2.2 Jaringan dan Topologi

Jaringan komputer adalah sebuah kumpulan komputer, printer dan peralatan lainnya yang terhubung. Informasi dan data bergerak melalui kabel-kabel sehingga memungkinkan pengguna jaringan komputer dapat saling bertukar dokumen dan data, mencetak pada printer yang sama dan bersama-sama menggunakan *hardware/software* yang terhubung dengan jaringan. Tiap komputer, printer atau perangkat yang terhubung dengan jaringan disebut ***node***. Sebuah jaringan komputer dapat memiliki dua, puluhan, ribuan atau bahkan jutaan *node*. Sebuah jaringan biasanya terdiri dari 2 atau lebih komputer yang saling berhubungan diantara satu dengan yang lain, dan saling berbagi sumber daya misalnya *CDROM*, Printer, pertukaran data atau memungkinkan untuk saling berkomunikasi secara elektronik. Komputer yang terhubung tersebut, dimungkinkan berhubungan dengan media kabel, saluran telepon, gelombang radio, satelit, atau sinar infra merah.

2.2.1 Manfaat Jaringan Komputer

a. *Sharing resources*, bertujuan agar seluruh program, peralatan lainnya dapat dimanfaatkan oleh setiap orang yang ada pada jaringan komputer tanpa terpengaruh oleh lokasi maupun pengaruh dari pemakai. Dengan kata lain, seorang *computer user* yang letaknya jauh sekalipun dapat memanfaatkan data maupun informasi yang lainnya tanpa mengalami kesulitan.
b. Dengan jaringan komputer kita akan mendapatkan reliabilitas yang tinggi dengan memiliki sumber-sumber alternatif persediaan. Misal semua data data disimpan atau disalin ke dua, tiga atau lebih komputer yang terkoneksi ke suatu jaringan. Sehingga bila salah satu komputer rusak, maka data masih dapat digunakan dengan menggunakan komputer lainnya.
c. Jaringan komputer dapat menghemat uang. Kita dapat membeli suatu barang di Negara lain tanpa harus mengeluarkan uang yang banyak. Hanya dengan menggunakan jaringan komputer kita dapat membeli barang tersebut.
d. Sistem jaringan komputer memberikan perlindungan terhadap data. Jaminan keamanan tersebut diberikan melalui pengaturan hak akses para pemakai, serta teknik perlindungan terhadap *harddisk* (media penyimpanan data) sehingga data mendapatkan perlindungan yang efektif.

2.2.2 Jenis Jaringan Komputer

Jaringan komputer dapat dibedakan berdasarkan cakupan geografisnya. Ada empat kategori utama jaringan komputer yaitu :

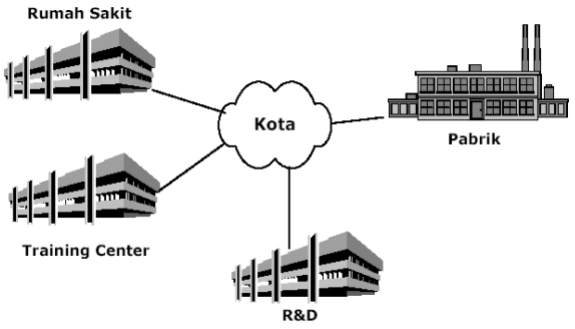
a. LAN (*Local Area Network*)
b. MAN (*Metropolitan Area Network*)
c. WAN (*Wide Area Network*)
d. GAN (*Global Area Network*)

2.2.2.a Local Area Network (LAN)

LAN digunakan untuk menghubungkan komputer yang berada di dalam suatu area yang kecil, misalnya di dalam suatu gedung perkantoran atau kampus. Jarak antar komputer yang dihubungkannya bisa mencapai 5 sampai 10 km. Suatu LAN biasanya bekerja pada kecepatan mulai 10 Mbps sampai 100 Mbps. LAN menjadi populer karena memungkinkan banyak pengguna untuk memakai sumber daya secara bersama-sama. Contoh dari sumber daya yang dapat digunakan itu misalnya suatu mainframe, *file server*, printer, dan sebagainya.

2.2.2.b Metropolitan Area Network (MAN)

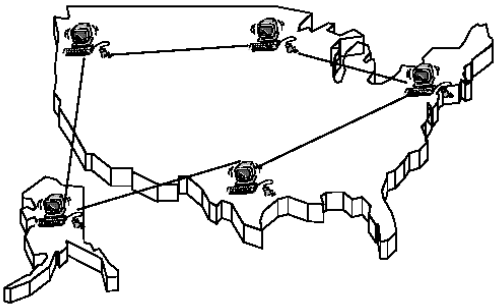
MAN merupakan suatu jaringan yang cakupannya meliputi suatu kota. MAN menghubungkan LAN-LAN yang lokasinya berjauhan. Jangkauan MAN bisa mencapai 10 km sampai beberapa ratus km. Suatu MAN biasanya bekerja pada kecepatan 1,5 sampai 150 Mbps. Pada gambar 2.3 anda dapat melihat suatu ilustrasi tentang MAN.



Gambar 2.3. Contoh MAN

2.2.2.c Wide Area Network (WAN)

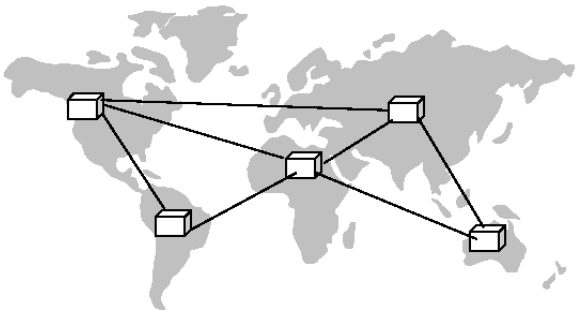
WAN dirancang untuk menghubungkan komputer-komputer yang terletak pada suatu cakupan geografis yang luas, seperti hubungan dari satu kota ke kota lain di dalam suatu negara. Cakupan WAN bisa meliputi 100 km sampai 1.000 km, dan kecepatan antar kota bisa bervariasi antara 1,5 Mbps sampai 2,4 Gbps. Dalam WAN, biaya untuk peralatan transmisi sangat tinggi, dan biasanya jaringan WAN dimiliki dan dioperasikan sebagai suatu jaringan publik. Para pelaku bisnis dapat menyewa sistem transmisi tersebut untuk menghubungkan kantor-kantor cabang yang dimilikinya. Gambar 2.4 mengilustrasikan suatu WAN.



Gambar 2.4. Contoh WAN

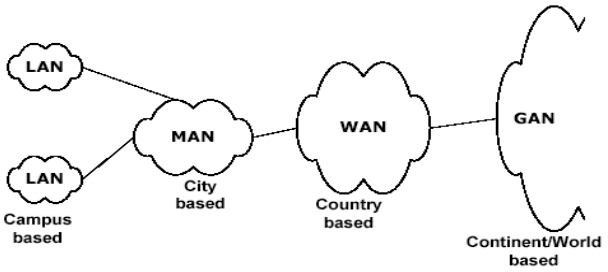
2.2.2.d Global Area Network (GAN)

GAN merupakan suatu jaringan yang menghubungkan negara-negara di seluruh dunia. Kecepatan GAN bervariasi mulai dari 1,5 Mbps sampai dengan 100 Gbps dan cakupannya mencakapi ribuan kilometer. Contoh yang sangat baik dari GAN ini adalah *Internet*. Gambar 2.5 memperlihatkan contoh suatu GAN.



Gambar 2.5. Ilustrasi GAN

LAN, MAN, WAN dan GAN dapat berinteraksi satu sama lain. Gambar 2.6 memperlihatkan interaksi antara jaringan-jaringan tersebut.



Gambar 2.6. Interaksi antara LAN, MAN, WAN, dan GAN

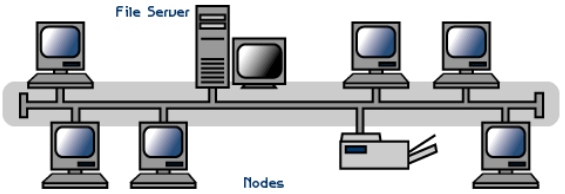
Interface yang digunakan antara jaringan-jaringan tersebut sudah ditentukan di dalam suatu *standard interface* internasional maupun regional. Standar-standar ini memungkinkan peralatan-peralatan yang berasal dari vendor yang sama berbeda dapat dihubungkan satu lain.

2.2.3 Topologi Jaringan Komputer

Topologi Secara fisik dari *network* adalah merujuk pada konfigurasi kabel, komputer, dan perangkat lainnya. Tidak usah bingung mengenai topologi fisik dan *topology logical*, dimana *topology logical* adalah metode yang digunakan untuk mengantarkan informasi diantara *workstation*.

1. *Linear Bus* - Garis Lurus

Topologi sebuah *Linear Bus*/Garis lurus terdiri dari satu jalur kabel utama dimana pada masing-masing ujungnya diberikan sebuah terminator. Semua *nodes* pada jaringan (*File server, workstation*, dan perangkat lainnya) tersambung sebuah kabel utama. Jaringan-jaringan *Ethernet* dan *Localtalk* menggunakan topologi linear ini.



Gambar 2.7. Topologi bus Garis Lurus / Linear

Keuntungan dari Topologi *Linear*/Garis lurus :

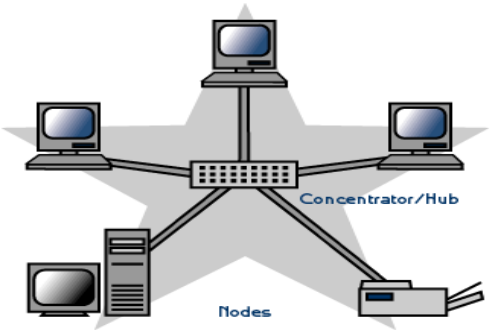
1. Mudah diinstalasi pada komputer atau perangkat lain ke dalam sebuah kabel utama.
2. Tidak terlalu banyak menggunakan kabel dibandingkan dengan topologi *star*/bintang.

Kekurangan dari Topologi *Linear*/Garis lurus :

1. Seluruh jaringan akan mati jika ada kerusakan pada kabel utama.
2. Membutuhkan terminator pada kedua sisi dari kabel utamanya.
3. Sangat sulit mengidentifikasi permasalahan jika jaringan sedang jatuh/mati.
4. Sangat tidak disarankan dipakai sebagai salah satu solusi pada penggunaan jaringan di gedung besar.

2. *Star* - Bintang

Topologi model ini didesain dimana setiap *node* (*file server, workstation*, dan perangkat lainnya) terkoneksi ke jaringan melewati sebuah *hub* atau *concentrator*. Data yang terkirim ke jaringan akan melewati *hub*/*concentrator* sebelum melanjutkan ke tempat tujuannya. *Hub* akan mengatur dan mengontrol keseluruhan fungsi jaringan. Dia juga bertindak sebagai *repeater*/penguat aliran data. Konfigurasi pada jaringan model ini menggunakan kabel *Tweisted pair*, dan dapat digunakan bersama kabel *koaksial* atau kabel fiber optik.



Gambar 2.8. Topologi Jaringan Bintang/Star

Keuntungan Topologi jaringan model Bintang/*Star* :

1. Mudah di pasang dan tidak mengganggu *node* yang lain.
2. Tidak mengakibatkan gangguan pada jaringan ketika akan memasang atau memindahkan perngkat jaringan lainnya.
3. Mudah untuk mendeteksi kesalahan dan memindahkan perangkat-perangkat lainnya.

Kekurangan Topologi jaringan model Bintang/*Star* :

1. Membutuhkan lebih banyak kabel daripada topologi jaringan garis lurus.

2. Membutuhkan hub atau concentrator, dan bilamana hub atau konsentrator tersebut jatuh atau rusak *node-node* yang terkoneksi tidak terdeteksi.
3. Lebih mahal daripada topologi *linear/*garis lurus, karena biaya untuk pengadaan hub dan konsentrator.

Protokol-protokol yang menggunakan konfigurasi bintang ini umumnya adalah *Ethernet* atau *LocalTalk*. *Token Ring* menggunakan topologi yang sama dengannya yang disebut Bintang dalam Lingkaran / *star-wired ring*.

2.3 Pengertian *Client/Server*

Sistem *client-server* mempunyai dua komponen utama yaitu *computer client* dan *computer server*. *Server* merupakan computer induk yang melakukan pemrosesan terbanyak untuk memenuhi permintaan-permintaan dari komputer *client* dan bertindak sebagai *server database* yang menyimpan data. Client adalah komputer yang melakukan pengiriman permintaan-permintaan data pada *server* kemudian menampilkan data tersebut pada *interface* aplikasi yang dimiliknya. Selain itu *client* juga mempunyai kemampuan untuk mengubah atau menghapus data.

Sistem tersebut merupakan *system client* komputer yang melibatkan proses-proses *client* yang meminta suatu pelayanan data kepada *computer server* yang menyediakan layanan data tersebut. Sehingga *client* maupun *server* sama-sama melakukan pekerjaan. Dengan adanya kombinasi *client* dan *server* ini maka kumpulan dari modul-modul program tidak dieksekusi dalam memori yang sama namun terbagi dalam computer *client-server.*

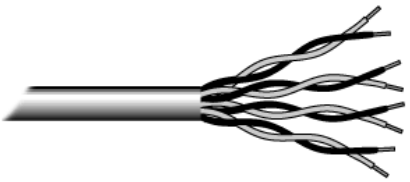
Hal ini menjadikan konfigurasi kedua komputer tersebut berbeda seperti kapasitas memori, kecepatan *processor* atau alat masukan dan alat keluaran yang disesuaikan dengan fungsi kerja dari elemen-elemen tersebut. Bagi *server* yang menjalankan tugas pengelolaan suatu *database* digunakan suatu konfigurasi yang

khusus menangani tugasnya tersebut, sedangkan komputer *client* menggunakan konfigurasi yang umum bagi sebuah komputer *desktop* yang terhubung ke jaringan.

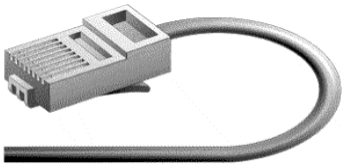
2.4 Media Transmisi Jaringan

Kabel adalah alat penghubung untuk mengirim informasi dari satu komputer ke komputer yang lain. Ada beberapa macam tipe kabel yang umumnya di gunakan bersama LAN. Dalam beberapa kasus, sebuah jaringan hanya menggunakan satu macam tipe kabel, di jaringan yang lain menggunakan kabel yang berbeda pula. Kabel yang dipilih adalah berdasarkan dengan topologi jaringan, protokol jaringan, dan ukurannya. Ini sangat penting untuk diketahui karena kesuksesan jaringan bergantung dari ke semua aspek tersebut. Salah satunya adalah kabel *Twisted Pair*, karena kabel ini paling sering digunakan

Kabel *twisted pair* ada dua tipe yaitu *shielded* and *unshielded. Unshielded twisted pair* (UTP) adalah yang paling populer dan umumnya merupakan pilihan yang terbaik untuk jaringan sederhana. Kualitas kabel UTP berbeda dengan telepon, Kabel ini mempunyai 4 pasang kabel di dalamnya. Setiap pasangan adalah kembar. Standar konektor untuk kabel jenis ini adalah konektor RJ-45, lihat gambar 2.11.



Gambar 2.9. Kabel Unshielded twisted pair



Gambar 2.10. Konektor RG 45

Kabel *Shielded Twisted Pair (STP),* sangat sensitif terhadap sinyal radio dan listrik. kabel ini baik sangat baik digunakan dimana lingkungan pengaruh listrik kurang, kabel ini biasanya digunakan pada jaringan yang menggunakan topologi *Token Ring*

2.5 Protokol

Agar komputer-komputer dalam suatu jaringan dapat berkomunikasi, maka dibutuhkan suatu protokol atau suatu aturan standar komunikasi baik antar komputer maupun antara jaringan komputer. Saat ini terdapat banyak jenis protokol jaringan komputer, seperti IPX/SPX yang biasa digunakan oleh **NOVELL** Netware, **NETBIOS / NETBEUI** biasa digunakan oleh jaringan Microsoft *LAN Manager* ataupun *Microsoft Window Networking*, *AppleTalk* yang biasa digunakan oleh jaringan komputer *Apple Macintosh*, dsb. Ada satu protokol yang dikembangkan oleh **DARPA** *(Defense Advanced Research Projects Agency)*, dalam pengembangan dari **ARPANET** dan juga digunakan oleh jaringan komputer berbasis sistem operasi **UNIX** yaitu protokol **TCP/IP** *(Transmission Control Protocol/Internet Protocol).* Protokol TCP/IP ini menjadi standar protokol yang digunakan pada jaringan Internet, karena TCP/IP dikembangkan untuk dapat diterapkan di hampir segala jenis *platform* komputer, biasa dikenal dengan konsep *open system*.

2.5.1 Protokol TCP/IP

2.5.1.a *Transmission Control Protocol* (TCP)

Transmission Control Protocol (TCP) melakukan transmisi data per segmen, artinya paket data dipecah dalam jumlah yang sesuai dengan besaran paket, kemudian dikirim satu per satu hingga selesai. Agar pengiriman data sampai dengan baik, maka pada setiap paket pengiriman TCP akan menyertakan nomor seri (*sequence number*). Komputer *client* yang menerima paket tersebut harus mengirim balik sebuah sinyal *ACKnowledge* dalam satu periode yang ditentukan. Bila pada waktunya komputer *client* belum juga memberikan *ACK*, maka akan terjadi *time out* yang menandakan pengiriman paket gagal dan harus

diulang kembali. Model protokol TCP disebut sebagai *connection oriented protocol*.

Port merupakan pintu masuk datagram dan paket data. *Port* data dibuat mulai dari 0 sampai 65.536. *Port* 0 sampai 1024 diesdiakan untuk layanan standar, seperti FTP, TELNET, Mail, Web dan lainnya. *Port* ini lebih dikenal dengan nama *well-know port*. Dapat dilihat contoh port pada tabel 2.1.

Tabel 2.1. Contoh Beberapa Nomor Port Umum

No Port	Keterangan
21	FTP
110	POP3
23	Telnet
25	SMTP
80	HTTP/Web
161	SNMP
162	SNMP Trap

2.5.1.b *Internet Protocol (IP)*

Internet protokol menggunakan IP *address* sebagai identitas. Pengiriman data akan dibungkus dalam paket dengan label berupa IP *address* si pengirim dan IP address penerima. Apabila IP penerima melihat pengiriman paket tersebut dengan identitas IP *address* yang sesuai, maka datagram tersebut akan diambil dan disalurkan ke TCP melalui *port*. IP *address* dibagi menjadi dua bagian, yaitu:

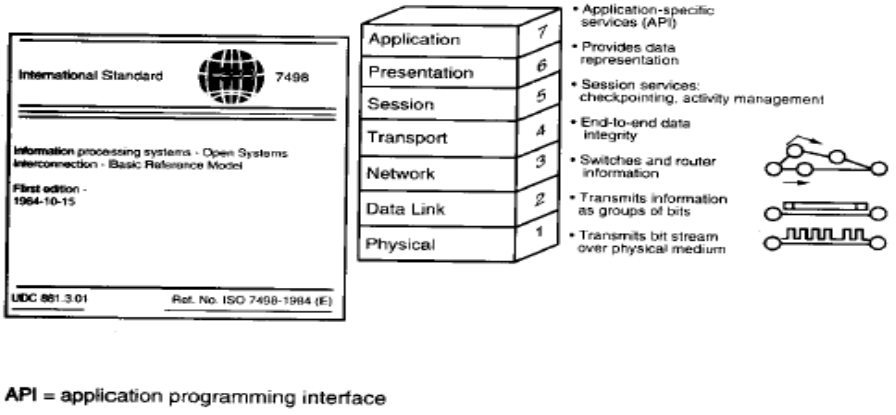
1. *Network ID* (menyatakan identitas jaringan)
2. *HOST ID* (menyatakan identitas komputer)

Sedangkan untuk penulisan IP *address* terbagi atas empat oktet, yang masing-masing mempunyai nilai maksimal 255 atau lebih jelasnya dapat dilihat pada sub bab berikut dalam Tugas Akhir ini.

2.5.1.c Arsitektur TCP/IP

Dalam mempelajari arsitektur protokol ini maka konsep *layer* (lapisan) yang termasuk keluarga TCP/IP masih diperlukan, karena berhubungan dengan elemen didalam protokol lain yang dipakai pada aplikasi TCP/IP. Elemen protokol yang akan dibahas dalam hubungan ini adalah model OSI. Model tersebut masih diperlukan dalam memahami konsep susunan elemen pembentuk protokol TCP/IP. Jaringan TCP/IP dapat dijabarkan berdasarkan 3 elemen jaringan, yaitu *physical-connectios, protocols* dan *applications*. *Physical-connection* menyediakan media yang dilewati data biner pada saat dikirimkan, media ini dapat berupa kabel *coaxial*, kabel *telephone*, *leased-line*, gelombang *infrared*, gelombang radio, *microwave-link, satellite-link,* dll. *Physical-connectios* ini merupakan level terendah secara fungsional dalam jaringan. Untuk mengoperasikan jaringan maka diperlukan sekumpulan standart tentang tatacara yang diikuti semua peralatan agar dapat saling berkomunikasi dan bekerja sama, sekumpulan standart inilah yang disebut protokol yang dapat menyediakan berbagai fungsi komunikasi pada jaringan. *Applications* menggunakan *network-protocol* yang dipakai sebagai dasar untuk berkomunikasi pada saat *network-application* berjalan pada jaringan, selanjutnya *network-protocol* menggunakan *physical-connections* untuk mengirim data. Dalam memberikan gambaran jaringan beroperasi yang merupakan gabungan elemen *physical-connectios, protocols* dan *applications*, maka dapat dilihat elemen jaringan ini menurut hirarki : *applications* berada pada level teratas dan *physicalconnectios* berada paling bawah, maka *protocol* menjadi jembatan diantara keduanya. Untuk mengerti lebih dalam tentang hirarki diantara elemen jaringan dan fungsi yang dijalankan, maka dapat dipakai suatu ukuran atau model untuk menggambarkan masing-masing bagian dan

fungsinya. Suatu model yang paling umum dipakai sebagai acuan adalah *open-system-interconnection* (OSI).



Gambar 2.11 . Model Referensi OSI

Model OSI memiliki 7 *layer* yang bekerja dari *layer* teratas menuju kebawah sesuai urutan : *aplication, presentation, session, transport, network, data-link, physical*. Ketujuh *layer* tersebut disusun berdasarkan lima prinsip yang harus diikuti untuk menentukan layer dalam komunikasi, yaitu :

1) *Layer* dibuat jika ketika diperlukan pemisahan level yang secara teori diperlukan.
2) Masing-masing *layer* memiliki fungsi yang jelas.
3) Setiap fungsi dari masing-masing *layer* telah ditentukan agar sesuai dengan standart protokol secara internasional.
4) Batas kedua *layer* telah ditentukan untuk mengurangi informasi menerobos antarmuka *layer.*
5) Setiap *layer* ditentukan dengan jelas fungsinya, tetapi jumlah *layer* sebaiknya sekecil mungkin untuk menghindari arsitektur yang luas.

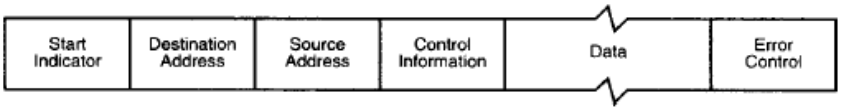
1. Physical Layer

Physical-layer dipergunakan untuk mentransmisikan data per bit melewati saluran komunikasi. Susunan bit tersebut mungkin mewakili pengiriman *file* atau *record* dari *file database*, *physical-layer* mengabaikan arti

(melupakan) susunan bit tersebut, untuk selanjutnya akan dikodekan menjadi digit **1** dan **0** atau menjadi bentuk analog. *Physical-layer* menangani proses mekanis, elektris dan prosedur antar muka dalam media fisik (saluran transmisi, driver, sensor, pencatu, dll.).

2. Data Link Layer

Data-link-layer dibentuk berdasarkan kemampuan transmisi dari *physical layer*. Susunan bit yang akan dikirim atau diterima dikumpulkan dalam kelompok yang disebut *frame.* Dalam konteks LAN, *frame* dapat berarti *token-ring* atau *ethenetframe.* Awal dan akhir *frame* di tandai dengan susunan bit khusus, sehingga *frame* tersusun dalam susunan bit yang terdiri atas *address-field, control-field, data-field,* dan *errorcontrol-field,* yang masing-masing memiliki fungsi tertentu.



Gambar 2.12. Bentuk Frame Data-link-layer

Address-field berisi alamat node pengirim *(source)* dan penerima *(destination). Control-field* dipakai untuk menandai adanya perbedaan jenis dari *data-link-frame,* termasuk frame data dan frame yang dipakai untuk mengatur *data-link-channel. Data-field* berisi data asli yang dikirimkan bersama dalam frame. *Error-control-field* dipakai untuk mendeteksi adanya pada *data-link-frame. Data-link-layer* merupakan layer pertama yang terlihat memiliki perhatian kepada pendeteksian error. *Error-control-field* umunya berisi hasil pengecekan secara hardware yang dipergunakan untuk mendeteksi adanya.

3. Network Layer

Network-layer dibentuk berdasarkan hubungan *node-to-node* yang disediakan oleh *data-link-layer*. Pelayanan *data-link* secara *node-to-node* menuju jaringan akan menjadi meningkat dengan adanya layer ini, sehingga

data-link-layer dapat menambah pelayanan untuk rute lintasan sejumlah *packet* (bagian dari informasi yang berada pada *network-layer*) diantara beberapa *node* dihubungkan melewati jaringan yang kompleks secara berubah-ubah. Disamping melayani proses *routing*, *network-layer* membantu menghilangkan kemacetan dengan cara mengatur aliran data. Disamping itu *network-layer* dapat membuat kemungkinan agar dua jaringan dapat dihubungkan menerapkan *uniformaddresing-mecanism* (suatu mekanisme untuk pengalamatan sejenis). Sebagai contoh jaringan lokal *Ethernet* dan *Token-ring* memiliki alamat *datalink* yang berbeda tipenya, untuk menghubungkan dua jaringan tersebut maka *diperlukan uniform-addressing-mechanism* yang dapat dimengerti oleh *Etrhernet* maupun *Token-ring*.

4. Transport Layer

Transport-layer adalah *layer* menengah dalam model OSI, 3 *layer* dibawahnya menyatakan bagian *subnet* dari model jaringan, sedangkan 3 *layer* diatasnya biasanya dipergunakan untuk proses *softwering* pada *node*. *Transport-layer* biasanya dipergunakan pula pada *node* yang tugasnya untuk merubah *subnet* yang tidak bisa diandalkan menjadi jaringan yang dapat lebih diandalkan. Karea adanya *multiplexing*, beberapa elemen *software* atau pada OSI disebut dengan *protocol-entity* untuk membagi *address* (alamat) *network-layer* yang sama. Untuk mengidentifikasi setiap elemen software didalam *transport-layer* diperlukan bentuk umum dalam pengalamatan, yang disebut dengan *transport-address* yang biasanya merupakan kombinasi alamat *network-address* dan nomor transport dari *service-access-point*. Kadangkala untuk mengidentifikasi alamat tranport disebut dengan *socket* atau *port-number*.

Contoh : *Transport-protocol* pada *Netware* menggunakan *squencedexchange-protocol* (SPX) dan *packet-exchange-protocol* (PXP), sedangkan pada TCP/IP menggunakan *transmission-control-protocol* (TCP).

5. Session-Layer

Session-layer menggunakan *transport-layer* untuk menyediakan perbaikan servis pada *session*. Contoh dari *session* termasuk pencatatan pada *host* saat *user* sedang pada jaringan atau *session* sedang menyusun untuk kegunaan transfer *file*. *Session* pada saat tersambung secara umum dapat menyediakan komunikasi dua arah *(full-duplex)*, tetapi pada beberapa aplikasi kadangkala hanya memerlukan komunikasi satu arah *(half-duplex)*. *Session-layer* memiliki *dialog-control* yang dapat menyediakan komunikasi satu arah atau dua arah.

1. *Dialog control*
2. *Token management*
3. *Activity management*

6. Presentation Layer

Presentation-layer mengatur cara untuk mewakili data, beberapa cara untuk mewakili data untuk file teks dan angka adalah menggunakan ASCII atau EBDIC. Jika dua sisi yang terlibat dalam komunikasi menggunakan meotda berbeda untuk mewakili data, maka tidak akan saling mengerti satu sama lain. *Presentation-layer* menggunakan *syntax* dan *semantic* yang umum untuk mewakili data. Jika semua *node* memakai dan mengerti bahasa yang umum dipakai maka kesalah pahaman dalam menginterprestasikan data dapat dihilangkan. Contoh bahasa yang umum dipakai sesuai rekomendasi OSI adalah *abstract-syntax-representaion, rev 1* (ASN 1).

7. Application Layer

Application-layer berisi beberapa protokol dan fungsi yang diperlukan oleh pemakai aplikasi untuk melakukan jenis komunikasi yang diinginkan.

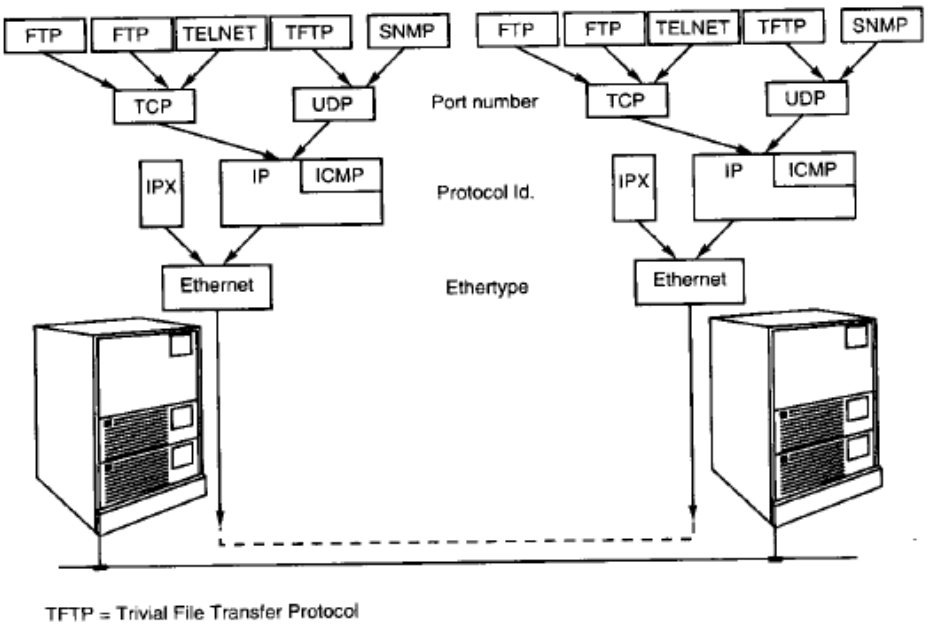
Contoh fungsi utama yang diperlukan adalah :

1. Protokol menyediakan fasilitas penyedia *file* secara *remote,* seperti *open, close, read, save, write* dan membagi akses ke file tersebut.
2. Akses transfer file dan database secara remote.
3. Fasilitas menangani pesan untuk aplikasi *e-mail*.
4. Fasilitas direktori global dan lokal pada sumber daya jaringan.
5. Cara yang seragam untuk menangani berbagai monitoring sistem dan peralatan.
6. Menjalankan perintah secara *remote.*
7. Beberapa contoh aplikasi tadi disebut dengan *Application-Programing Interface* (API). Untuk menulis aplikasi yang berhubungan dengan jaringan dapat digunakan pustaka pemprograman API.

Contoh aplikasi Netware pada *application-layer* adalah *netware-controlprotocol* (NCP) sedangkan untuk TCP/IP adalah FTP, SMTP dan TELNET.

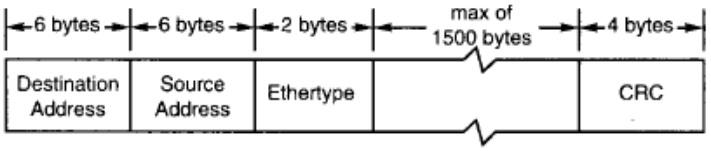
2.5.1.d *Multiplexing* dan *Demultiplexing*

Pada gambar 2.13 menunjukkan komunikasi TCP/IP antara dua *host*, masing masing *host* menjalankan penyedia aplikasi TCP/IP seperti FTP, TELNET, (TFTP) dan SMTP.



Gambar 2.13. Multiplexing dan Demultiplexing Protokol

Sebagai contoh pada gambar 2.13 *Ethernet* ditunjang oleh protokol IP dan ICMP serta beberapa protokol potensial lainnya seperti IPX dan *apple-talk* IDP. Agar Ethernet dapat membedakan *data-packet* yang datang dari *interface* jaringan yang didesain untuk IP atau ICMP, maka Ethernet menggunakan 2-byte e*thertipe-field* (terlihat pada gamabr 2.14) yang merupakan bagian dari *frame*-nya *Ethernet* yang dipergunakan sebagai protokol *multiplexing* dan *demultiplexing* pada *data-link-layer*.



Gambar 2.14. 2-byte Ethertype Field

Pada tabel 2.2 ditampilkan beberapa nilai dari *ethertype-field* yang dipakai pada beberapa protokol jaringan.

Tabel 2.2. Nilai Ethertype untuk Multiplexing/Demultiplexing pada Data-Link-Layer

Nilai Ethertype (desimal)	Nile Ethertype (hexadesimal)	Protokol atau organisasi yang mengusulkan
1-1500	0-05DC	IEEE 802.3 length field
1536	0600	XEROX NS IDP
2048	0800	DoD IP
2049	08001	X.75 Internet
2053	0805	X.25 Level 3
2054	0806	ARP
2055	0807	XNS Compatibility
2989	0BAD	Banyan Systems
32821	8035	Reverse ARP
32773	8005	HP Probe
32776	8008	AT&T
24586-24590	6010-6014	3COM Corporation
32923	809B	AppleTalk
33079-33080	8137-8138	Novell, Inc
36864	9000	Loopback

Jika lapisan IP menerima *packet* dari *ethernet*, maka lapisan IP dapat membedakan bahwa *packet* tersebut untuk diproses oleh modul protokol TCP atau UDP. Untuk melakukan hal ini digunakan *protocol-Id* berisi 8-bit yang berada pada paket IP. Pada tabel 2.3 ditampilkan beberapa nilai *protocol-Id*, yang dapat dilihat pula pada *Server Netware* pada *file sys:etc/protocol* atau pada file */etc/protocols* pada host Unix.

Tabel 2.3. Nilai Protocol-Id untuk Multiplexing/Demultiplexing pada Network-Layer1

Protocol ID	Layer pada paket IP
0	Reserved
1	Internet Control Message Protocol (ICMP)
2	Internet Group Management Protocol
4	IP in IP encapsulation
5	Steaming IP
6	Transmission Control Protocol (TCP)
8	Exterior Gateway Protocol (EGP)
9	Any provate interior gateway protocol (misalnya : CISCO's IGP)
11	Network Voice Protocol (NVP-II)
12	Parc Universal Protocol (PUP)
16	Chaos protocol
17	User Datagram Protocol (UDP)
21	Packet Radio Measurement (PRM)
22	Xerox NS IDP (XNS-IDP)
29	ISO Transport Protocol Class 4 (ISO-TP4)
30	Bulk Transfer Protocol (NETBLT)
36	Express Transport Protocol (XTP)
37	Datagram Delivery Protocol (DDP)
75	Packet Video Protocol (PVP)
80	ISO Internet Protocol (ISO-IP)
83	Vines

Pada saat modul protokol TCP atau UDP menerima *packet* dari lapisan IP, UDP atau TCP dapat membedakan *packet* tersebut apakah diproses pada aplikasi yang disediakan yaitu FTP, TELNET, SMTP atau SMTP. Modul protokol TCP dan UDP mengerjakan hal tersebut dengan memerikasa *port-number-field* yang terdiri atas 16-bit didalam masing-masing *packet*. Pada tabel 2.4 diberikan beberapa nilai *port-number* yang digunakan untuk multiplexing/demultiplexing pada modul protokol TCP, dan pada tabel 2.5 diberikan beberapap nilai *port-number* untuk multiplexing/demultiplexing pada modul protokol UDP.

Dikarenakan modul protokol TCP dan UDP adalah berbeda satu sama lain, maka port-number masing-masing berbeda, akan tetapi perlu dicatat bahwa pada tabel 2.4 dan 2.5 terdapat aplikasi TCP/IP dengan port-number yang berbeda, hal ini disebabkan aplikasi yang disediakan berlaku sama untuk TCP dan UDP. Nilai *port-number* untuk modul protokol TCP dan UDP pada suatu

sistem dapat dilihat pada file *sys:/ sys:/etc/services* untuk *server Netware* dan */etc/services* pada host Unix.

Tabel 2.4. Nilai Port-Number untuk Multiplexing/Demultiplexing pada TCP

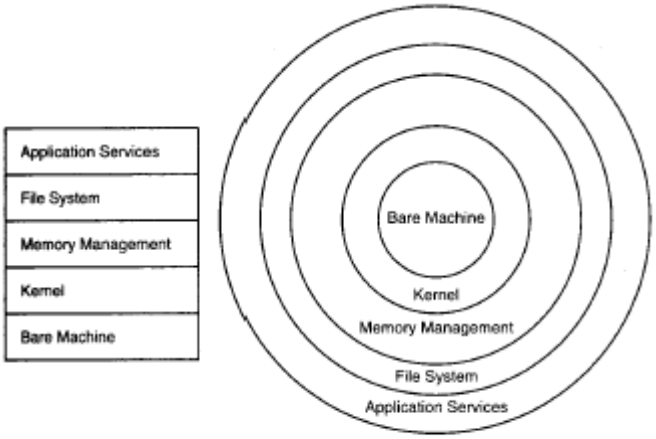
Nomer Port	Servis aplikasi yang disediakan
0	Reserved
1	TCP Port Service Multiplexor
2	Management Utility
4	Compression Process
5	Remote Job Entry
7	Echo
9	Dischard
11	Active Users (systat)
13	Daytime
17	Quote of the Day (QUOTD)
20	FTP data port
21	FTP control port
23	TELNET
25	SMTP
35	Any prtivate printer server
37	Time
39	Resource Location Protocol
42	Host name server (name server)
43	Who Is (nickname)
49	Login Host Protocol
52	XNZ Time Protocol
53	Domain Name Server (domain)
54	XNS clearing house
66	Oracle SQL*NET (sql*net)
67	Bootstrap Protocol Server (bootps)
68	Bootstrap Protocol Client (bootpc)
70	Ghoper Protocol
79	Finger Protocol
80	World Wide Web HTTP
dst...	dst...

Tabel 2.5. Nilai Port-Number untuk Multiplexing/Demultiplexing pada UDP

Nomer Port	Servis aplikasi yang disediakan
0	Reserved
2	Management Utility
4	Compression Process
5	Remote Job Entry
7	Echo
9	Dischard
11	Active Users (systat)
13	Daytime
17	Quote of the Day (QUOTD)
35	Any prtivate printer server
37	Time
39	Resource Location Protocol
42	Host name server (name server)
43	Who Is (nickname)
49	Login Host Protocol
52	XNZ Time Protocol
53	Domain Name Server (domain)
54	XNS clearing house
66	Oracle SQL*NET (sql*net)
67	Bootstrap Protocol Server (bootps)
68	Bootstrap Protocol Client (bootpc)
69	Trivial File Transfer Protocol (TFTP)
70	Ghoper Protocol
79	Finger Protocol
80	World Wide Web HTTP
dst...	dst...

2.5.1.e Sistem Operasi Host dan Implementasi TCP/IP

Unjuk kerja dalam penerapan TCP/IP, termasuk prosedur konfigurasi dan kemudahan dalam pengoperasiannya tergantung kepada *operating-system* (OS) apa yang dipakai untuk menjalankan TCP/IP. Untuk menunjukkan interaksi antara aplikasi dan protokol TCP/IP, beberapa vendor memberikan gambaran OS sebagai bagian dari model OSI, walaupun sebenarnya OS tidak begitu sesuai dengan model OSI, karena model OSI didesain untuk menyatakan bagian dari fungsi-fungsi komunikasi. OS memiliki model sendiri berupa *multiple-layer*, seperti ditunjukkan pada gambar 2.16 dibawah.. OS adalah program yang berisi pengatur sumber daya pada PC, sedangkan sistem jaringan komunikasi yang dimodelkan dalam OSI merupakan salah satu dari sumber daya yang diatur dalam OS. Pada beberapa jenis OS yang ada sistem



Gambar 2.15. Lapisan Sistem Operasi

Sebenarnya OS tidak diperuntukkan hanya menjalankan TCP/IP, karena dapat di gabungkan dalam ROM atau digabungkan dalam aplikasi, pada aplikasi komersial TCP/IP akan langsung berinteraksi dengan OS. Beberapa jenis interaksi antara TCP/IP dengan OS dapat diklasifikasi sebagai :

a. Bagian dari kernel OS (inti OS)

b. *Device-driver* (pengendali peralatan)
c. Proses yang terjadi dalam aplikasi

Pada beberapa OS seperti Unix dan Netware, maka TCP/IP diterapkan sebagai bagian dari kernel OS, sehingga dalam menerapkan TCP/IP seperti ini lebih cepat untuk mengakses kernel OS untuk fungsi-fungsi komunikasi. OS yang dilengkapi dengan penerapan TCP/IP sebagai *device-driver* misalnya OS/2, VMS, MS Windows, Windows NT, dan MS DOS. Pada MS Windows penerapan TCP/IP disusun dalam *virtual-device-driver* VxD sebagai pengendali 32-bit yang menggunakan *extended-memory* untuk menghindari penggunaan *base-memory* yang berlaku pada *processor* Intel, *base-memory* adalah memory dibawah 640 kB. OS yang menerapkan TCP/IP sebagai bagian dari proses dalam aplikasi misalnya IBM *mainframe-operating-system* jenis MVS dan VM, MS Windows, dan MS DOS. Pada penerapan TCP/IP dalam MS Windows menggunakan *dynamic-linklibraries* (DLL). *Interface* standart untuk DDL ini disebut *Winsock* (*Windows socket*), yang telah dipersiapkan oleh beberapa vendor antara lain Microsoft, Novell, Bean &Whiteside, FTP Software dan Net Manage. Beberapa interface *Winsock* versi *public-domain* juga dapat dipergunakan, misalnya *Trumpet Winsock*. Pada penerapan TCP/IP dalam MS DOS menggunakan progam *terminate-and-stay-resident* (TSR). Kalau TSR disimpan atau dijalankan pada *base-memory* akan menyebabkan persaingan dengan aplikasi lain dalam menggunakan memory, maka TSR di letakkan pada *upper-memory* menggunakan perintah DOS yaitu LOAD HIGH.

2.5.1.f IP Address

Untuk mempermudah proses pembagiannya, IP *address* dikelompokkan dalam kelas-kelas. Dasar pertimbangan pembagian IP *address* ke dalam kelas-kelas adalah untuk memudahkan pendistibusian pendaftaran IP *address*. Dengan memberikan sebuah *ruang nomor jaringan* (beberapa blok IP *address*) kepada ISP *(Internet Service Provider)* di suatu area diasumsikan penanganan komunitas lokal tersebut akan lebih baik, dibandingkan dengan jika setiap

pemakai individual harus meminta IP address ke otoritas pusat, yaitu *Internet Assigned Numbers Authority* (IANA). IP *address* ini dikelompokkan dalam lima kelas : Kelas A, Kelas B, Kelas C, Kelas D, dan kelas E. Perbedaan pada tiap kelas tersebut adalah pada ukuran dan jumlahnya. IP Kelas A dipakai oleh sedikit jaringan namun jaringan ini memiliki anggota yang besar. Kelas C dipakai oleh banyak jaringan, namun anggota masing-masing jaringan sedikit. Kelas D dan E juga didefinisikan, tetapi tidak digunakan dalam penggunaan normal. Kelas D diperuntukkan bagi jaringan *multicast*, dan Kelas E untuk keperluan eksperimental.

Pembagian kelas-kelas IP *address* didasarkan pada dua hal : *network ID* dan *host ID* dari suatu IP *address*. Setiap IP *address* selalu merupakan sebuah pasangan dari *network-ID* (identitas jaringan) dan host-ID (identitas *host* dalam jarinangan tersebut). *Network-ID* adalah bagian dari IP *address* yang digunakan untuk menunjukkan jaringan tempat komputer ini berada. Sedangkah *host-ID* ialah bagian dari IP *address* yang digunakan untuk menunjukkan *workstation*, *server, router*, dan semua *host* TCP/IP lainnya dalam jaringan tersebut. Dalam satu jaringan, *host-ID* ini harus unik (tidak boleh ada yang sama).

1. Kelas A

Karakteristik :

Format	: 0nnnnnnn hhhhhhh hhhhhhh hhhhhhh
Bit pertama	: 0
Panjang NetID	: 8 bit
Panjang HostID	: 24 bit
Byte pertama	: 0 – 127
Jumlah	: 126 Kelas A (0 dan 127 dicadangkan)
Range IP	: 1.xxx.xxx.xxx sampai 126.xxx.xxx.xxx
Jumlah IP	: 16.777.214 IP address pada tiap kelas A

IP address kelas A diberikan untuk jaringan dengan jumlah *host* yang sangat besar. Bit pertama dari IP *address* kelas A selalu di Set 0 (nol) sehingga byte terdepan dari IP *address* kelas A selalul bernilai antara angka 0 dan 127. Pada *IP address* kelas A, *network ID* ialah delapan bit pertama, sedangkan *host ID* ialah

24 bit berikutnya. Dengan demikian, cara membaca IP *address* kelas A, misalnya 113.46.5.6 ialah :

Network ID = 113
Host ID = 46.5.6

Sehingga IP *address* di atas berarti *host* nomor 46.5.6 pada *network* nomor 113. Dengan panjang host ID yang 24 bit, *network* dengan IP *address* kelas A ini dapat menampung sekitar 16 juta host.

2. Kelas B

Karakteristik :

Format	: 10nnnnnn nnnnnnn hhhhhhh hhhhhhh
2 bit pertama	: 10
Panjang NetID	: 16 bit
Panjang hosted	: 16 bit
Byte pertama	: 128-191
Jumlah	: 16.384 Kelas B
Range IP	: 128.0.xxx.xxx sampai 191.155.xxx.xxx
Jumlah IP	: 65.532 IP address pada tiap Kelas B

IP address kelas B biasanya dialokasikan untuk jaringan berukuran sedang dan besar. Dua bit pertama dari IP *address* kelas B selalu di set 10 (satu nol) sehingga byte terdepan dari IP *address* kelas B selalu bernilai antara 128 hingga 191. Pada IP *address* kelas B, network ID ialah enam belas bit pertama, sedangkan *host* ID ialah 16 bit berikutnya. Dengan demikian, cara membaca IP address kelas A, misalnya 132.92.121.1 ialah :

Network ID = 132.92
Host ID = 121.1

Sehingga IP *address* di atas berarti *host* nomor 121.1 Pada *network* nomor 132.92 Dengan panjang *host* ID yang 16 bit, network dengan IP *address* kelas B ini dapat menampung sekitar 65000 host.

3. Kelas C

Karakteristik :

Format	: 110nnnnn nnnnnnnn nnnnnnnn hhhhhhhh
3 bit pertama	: 110
Panjang NetID	: 24 bit
Panjang HostID	: 8 bit
Byte pertama	: 192-223
Kelas	: 2.097.152 kelas C
Range IP	: 192.0.0.xxx sampai 223.255.255.xxx
Jumlah IP	: 254 IP address pada tiap Kelas C

IP *address* kelas C awalnya digunakan untuk jaringan berukuran kecil (misalnya LAN). Tiga bit pertama dari IP *address* kelas C selalu berisi 111. Bersama 21 bit berikutnya, angka ini membentuk *network* ID 24 bit. Host-ID ialah 8 bit terakhir. Dengan konfigurasi ini, bisa dibentuk sekitar dua juta *network* dengan masing-masing network memiliki 256 IP *address.*

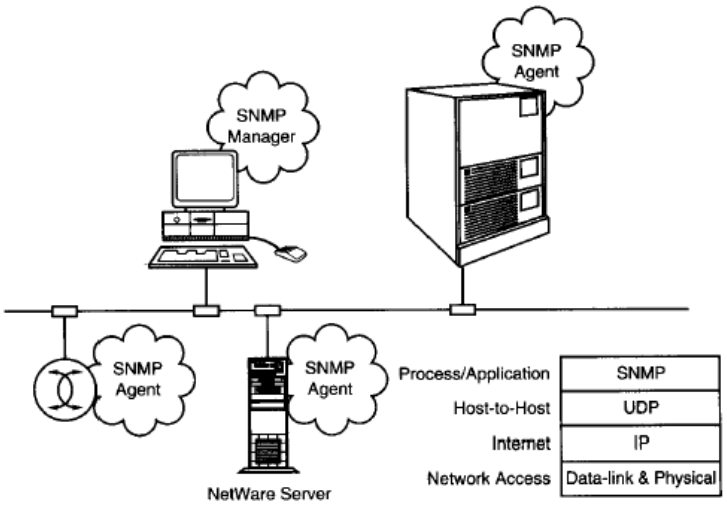
2.5.2 Protokol UDP (*User Datagram Protocol*)

Protokol UDP bersifat *connectionless* yang artinya UDP tidak mempunyai fasilitas *handshaking* sehingga tidak menjamin suatu data terkirim atau diterima dengan benar. Alasan penggunaan UDP adalah jenis data yang dikirimkan relatif kecil, dibandingkan dengan harus mengirimkan ulang data tersebut. Aplikasi yang menggunakan UDP bersifat *query-response*, jika suatu respon tidak diterima dalam jangka waktu tertentu maka suatu aplikasi akan mengirimkan ulang *query*-nya.

SNMP mempergunakan *port* layanan UDP dari "*well-known*" nya TCP/IP yaitu 161 dan 162. Port 161 dipergunakan oleh *manager* (untuk menuju *agent*) dan *port* 162 dipergunakan oleh *agent* (untuk menuju *manager*)

2.6 Simple Network Management Protocol (SNMP)

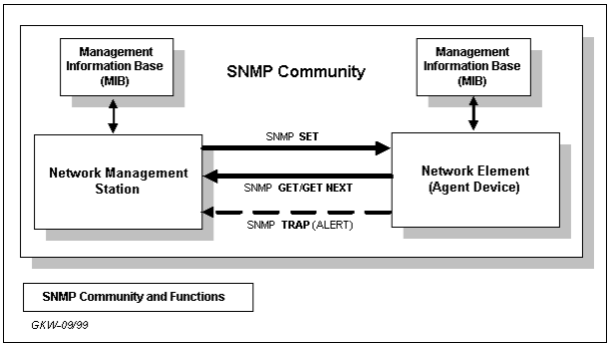
SNMP merupakan sistem yang dikembangkan untuk menyediakan peralatan manajemen jaringan yang mendasar dan yang mudah diimplementasikan untuk rangkaian protokol TCP/IP. Pada gambar dibawah ditunjukkan bahwa SNMP *manager* adalah suatu *host* khusus dalam jaringan yang dapat dipergunakan untuk mengirimkan pertanyaan yang berkisar untuk manajemen jaringan kepada *host/node* yang lain.



Gambar 2.16. Pemakaian SNMP

Node yang dikelola bisa berupa *Bridge, Router atau Host*, yang masing masing menjalankan program yang disebut dengan SNMP *agent*. Program SNMP *agent* ini menangkap pertanyaan yang dikirim oleh SNMP *manager* dan mengirimkan kembali informasi yang diinginkan. SNMP menggunakan UDP sebagai protokol transport untuk mengirimkan pertanyaan dan menerima jawaban dari SNMP *agent*.

SNMP merupakan bagian dari *Application layer* pada struktur protokol TCP/IP. *Simple Network Management Protocol* (SNMP) memudahkan seorang *administrator* dalam mengatur performansi jaringan, mencari dan menyelesaikan masalah dalam suatu jaringan. Dan membuat rencana perubahan dalam jaringan. Elemen *network management* terdiri dari NMS (*Network Management Station*) yang merupakan pusat pengendali untuk pengumpulan dan analisa data *network management.*



Gambar 2.17. Interaksi NMS dengan Agen

Pada SNMP terdapat 3 perintah dasar yang digunakan untuk mengidentifikasi suatu *host* pada jaringan komputer, yaitu:

1. *Read/Get*, digunakan oleh NMS untuk mengambil suatu *item* dari agen MIB.
2. *Write/Set*, digunakan oleh NMS untuk mengkonfigurasi harga suatu variable pada agent MIB.
3. *Trap*, digunakan oleh agen untuk mengirim peringatan kepada NMS.

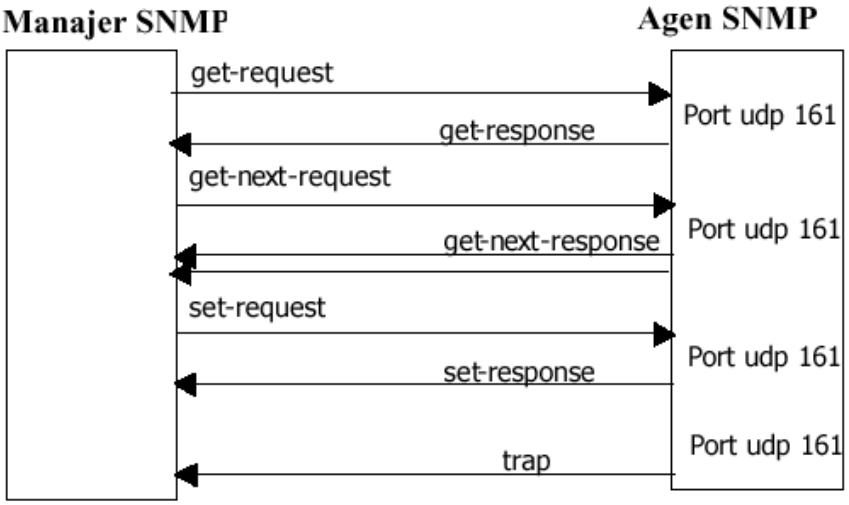
SNMP juga mempunyai elemen-elemen yang penting, yaitu :

1. *Manager* adalah pelaksana dan manajemen jaringan. Pada kenyataannya *manager* ini merupakan komputer biasa yang ada pada jaringan yang mengoperaksikan perangkat lunak untuk manajemen jaringan. Manajer ini terdiri atas satu proses atau lebih yang berkomunikasi dengan agen-agennya dan dalam jaringan. Manajer akan mengumpulkan informasi dari agen dari jaringan yang diminta oleh administrator saja bukan semua informasi yang dimiliki agen.
2. MIB atau *Manager Information Base*, dapat dikatakan sebagai struktur basis data variabel dari elemen jaringan yang dikelola. Struktrur ini bersifat hirarki dan memiliki aturan sedemikian rupa sehingga informasi setiap variabel dapat dikelola atau ditetapkan dengan mudah.
3. *Agent* merupakan perangkat lunak yang dijalankan disetiap elemen jaringan yang dikelola. Setiap agen mempunyai basis data variabel yang bersifat

lokal yang menerangkan keadaan dan berkas aktivitasnya dan pengaruhnya terhadap operasi.

2.6.1 PDU (*Protocol Data Unit*) SNMP

PDU merupakan unit data yang terdiri atas sebuah *header* dan beberapa data yang ditempelkan. Analogi dari PDU seperti sebuah benda yang mengandung variabel-variabel. Variabel ini memiliki nama dan nilai. Protokol SNMP menggunakanan operasi yang relatif sederhana dan PDU dalam jumlah terbatas untuk menjalankan fungsinya. Lima PDU yang telah didefinisikan dalam standart adalah sebagai berikut : *Get Request, Get-Next Request, Get Response, Set Request, Set Response, Trap*.



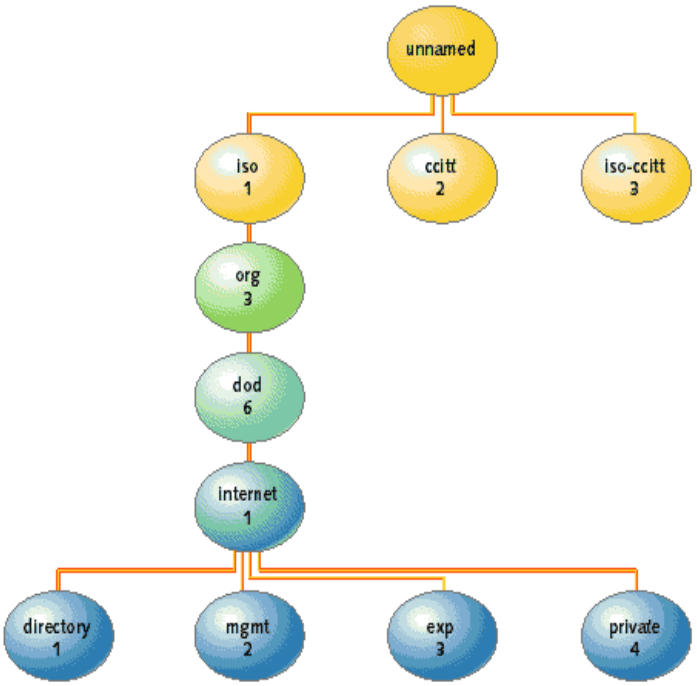
Gambar 2.18. Interaksi Manager dan Agent

Struktur Informasi dalam SNMP Informasi dalam SNMP disimpan dalam bentuk variabel- variabel yang didefinisikan dalam MIB. Ada variabel yang berjenis teks, bilangan bulat atau *integer*, waktu, dan sebagainya. Jenis-jenis ini juga telah didefinisikan ke dalam beberapa tipe data atau variabel.

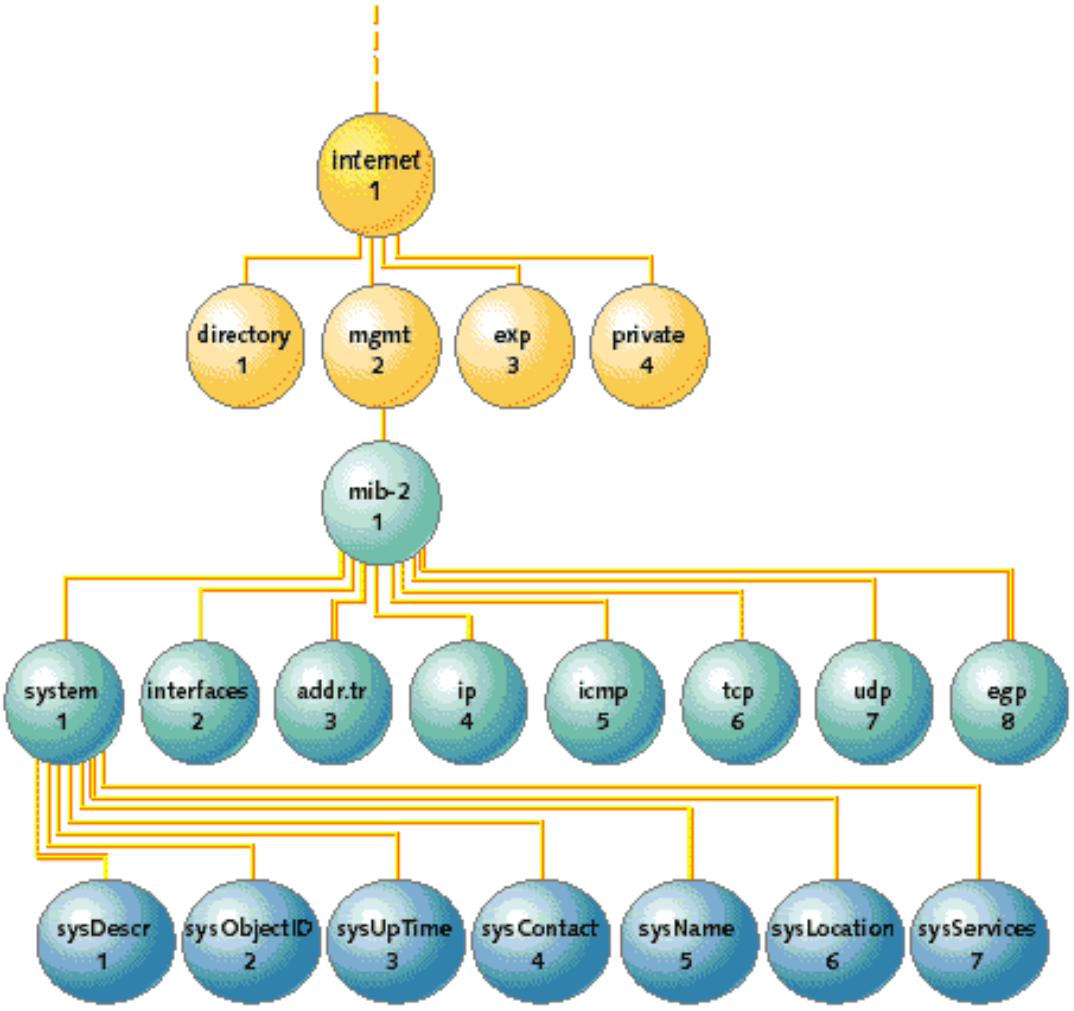
2.7 Management Information Base (MIB)

Salah satu konsep terpenting dalam manajemen SNMP adalah konsep tentang *Management Information Base* (MIB) agen. MIB adalah struktur basis data variabel dari elemen jaringan yang dikelola. Semua informasi yang diakses atau dimodifikasi melalui agen sama halnya dengan diakses melalui MIB. Sebagai contoh, supaya bisa membaca sebuah nilai dari agen, *manager* SNMP memeriksa MIB. Supaya bisa menuliskan sebuah nilai, maka *manager* SNMP menuliskannya pada MIB. Dan MIB terdiri dari objek-objek yang diatur, dan masing-masing memiliki sebuah pengidentifikasi yang unik.

Masing-masing objek yang diatur itu memiliki pengidentifikasi yang unik, yang disebut *Object Identifier* (OID). Berkenaan dengan sebuah objek MIB, OID haruslah digunakan. Semua agen standar SNMP diperlukan untuk mendukung serangkaian standar objek yang diatur. Ini akan memungkinkan manajer SNMP mengakses objek MIB melalui serangkaian standar OID.



Gambar 2.19. Hirarki OID MIB



Gambar 2.20. Hirarki OID MIB - Sys Group

Untuk setiap objek MIB akan muncul definisi MIB yang menetukan objek yang diatur. Definisi tersebut mencakup sejumlah atribut, termasuk pengidentifikasi objek, tipe *syntax*, izin akses, deskripsi, dan informasi contoh. Definisi MIB diciptakan dengan menggunakan syntax spesial yang disebut *Abstract Syntax Notation One* (ASN.1). ASN.1 adalah deskripsi data *machine-independent* yang pertama kali dikembangkan unutk penggunaan di dalam rangkaian protocol OSI. Untuk SNMP, hanya subrangkaian ASN.1 ditentukan di dalam *Structure of Management Information* (SMI), bagian dari SNMP RFC (*Request For Comment*).

2.8 Bahasa Pemrograman PERL

PERL merupakan singkatan dari *"Practical Extraction and Report Language"* adalah sebuah bahasa *interpreter* sekaligus kompiler. Dalam hal ini PERL akan mendeteksi setiap baris untuk mencari *syntax error* sebelum program dijalankan (*run*). PERL diciptakan oleh Larry Wall pada tahun 1986, awalnya hanya beroperasi pada sistem operasi UNIX, namun saat ini PERL telah diterapkan pada berbagai *platform* seperti Windows, OS/2, Macintosh dan Linux.

PERL adalah sebuah bahasa pemrograman yang menggabungkan kemampuan dari bahasa C, utilitas *sed* dan *awk* bahkan fitur *shell*. PERL juga mirip dengan bahasa C sehingga *programer* C pun mampu mempelajari PERL dengan cepat dan mudah. Salah satu contoh kemiripannya adalah adanya tanda titik koma atau *semicolon* " ; " pada akhir suatu perintah. Berikut salah satu contoh sederhana program PERL:

```
#!usr/bin/perl
Print "Selamat datang didunia PERL !\n"
```

Simpan file tersebut dengan nama tes.pl, kemudian dalam shell UNIX ubah hak akses file dengan pernyataan:

```
Chmod 755 tes.pl
```

Dari shell UNIX, script dapat dijalankan dengan perintah "perl tes.pl". sehingga akan menampilkan : Selamat datang didunia PERL ! pada layar.

2.9 CGI (*Common Gateway Interface*)

CGI adalah suatu metode dimana sebuah *web server* dapat memperoleh data dari atau mengirim data ke *database*, dokumen dan program-program yang lain, serta merepresentasikan data tersebut melalui web. Dengan kata lain *Common Gateway Interface* adalah bahasa pemrograman untuk web, dan dapat ditulis dengan sembarang bahasa pemrograman, misalnya C/C++, Fortran, PERL, TCL, Visual Basic. CGI mendefinisikan bagaimana *web server* dan *web browser* mengatasi informasi dari *form* HTML pada suatu dokumen web.

Sebagaimana bahasa pemrograman, dalam membuat program CGI juga terdapat beberapa persyaratan agar memudahkan pembuatannya, yaitu:

1. Server yang menangani CGI
2. Browser dan akses Internet
3. Bahasa pemrograman
4. *Software* khusus

Software khusus hanya dibutuhkan apabila *programmer* bekerja di luar *platform UNIX*, karena CGI bekerja di bawah *platform UNIX*. Selain beberapa persyaratan diatas, CGI juga membutuhkan beberapa media, yaitu:

a. *Text Editor*, digunakan untuk menulis script CGI yang dengan cara menulis dokumen HTML atau JavaScript, karena sama-sama menggunakan teks, kecuali dalam penggunaan file ekstensi yaitu ".pl" atau ".cgi".
b. *Path Compiler* PERL pada *Host Web*, umumnya dalam penulisan script CGI pada baris pertama dituliskan "#!/usr/bin/perl". Pernyataan tersebut menunjukkan di mana letak *compiler* PERL.
c. Selain hal tersebut, yang perlu diperhatikan adalah di mana program CGI akan ditempatkan. Program CGI pada linux ditempatkan di direktori "/srv/www/htdocs/cgi-bin".

Pada *Common Gateway Inteface* terdapat bagian utama yang sangat penting, yaitu MIME (*Multipurpose Internet Mail Extensions*) Header. Dengan CGI untuk setiap *file* yang masuk ke *server* harus memiliki informasi yang baik agar *file* tersebut dapat ditangani dengan baik dan informasi tersebut diletakkan pada *MIME Header.* Daftar MIME *Header* dapat dilihat dalam tabel 2.6 (*dikutip dari:Andi Yogyakarta,2001*).

Tabel 2.6. MIME Header

Tipe File	Ekstensi
text/plain	txt, text
text/html	html, htm
application/zip	zip
application/x-sh	sh
application/x-perl	pl
audio/x-wav	wav
image/gif	gif
image/tiff	tiff, tif
image/jpeg	jpeg, jpg, jpe
video/x-msvideo	avi
video/quicktime	qt, mov, moov
video/mpeg	mpeg, mpg, mpe

2.10 Socket

Socket adalah mekanisme komunikasi yang memungkinkan terjadinya pertukaran data atau melakukan suatu proses baik dalam satu *platform* maupun multi *platform* pada setiap lingkungan sistem operasi. *Socket interface* juga telah tersedia untuk Microsoft™ Windows yang sering disebut sebagai WinSock. Dalam pemrograman jaringan, setelah membuat sebuah pipa data antara dua proses, maka masing-masing proses akan memegang satu ujung. Setiap ujung dari pipa data ini disebut *socket*. Jadi *socket* adalah *end-point* komunikasi antara *client* dan *server*. Mekanisme *socket* dapat diimplementasikan pada multi *client* dengan satu *server*.