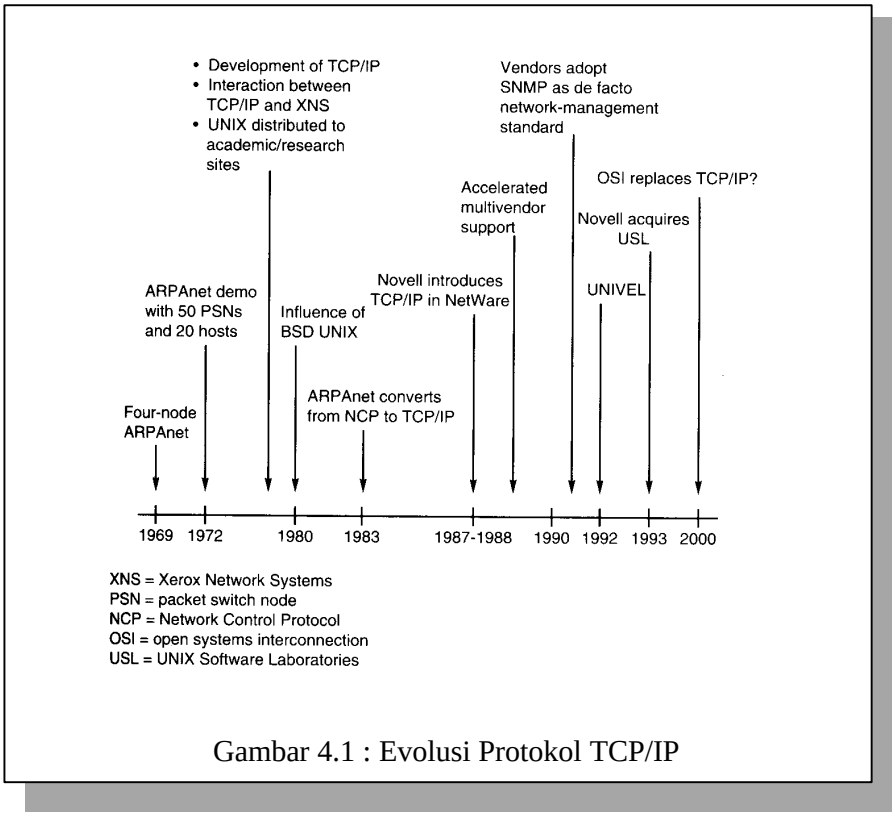
4 PROTOKOL TCP/IP

Gambaran yang umum tentang TCP/IP adalah gabungan dari dua protokol komunikasi yang dipakai untuk kominikasi data, TCP kepanjangan dari *transmission-control-protocol* dan IP kepanjangan dari *internet-protocol*. Kedua protokol tersebut dipakai untuk menyatakan sekelompok protokol yang memiliki kaitan dengan protokol TCP dan IP seperti *user-datagram-protocol* (UDP), *file transfer protocol* (FTP), *terminal-emulation-protocol* (TELNET), dll.



Gambar 4.1 : Evolusi Protokol TCP/IP

TCP/IP merupakan hasil riset dan pengembangan protokol pada percobaan jaringan *packet-switched* yang ditemukan oleh *defense-advanced-research-project-*

agency (DARPA*)* dengan nama ARPAnet pada tahun 1970, yang kemudian secara umum disebut protokol TCP/IP. Rangkaian protokol ini berisi beberapa protokol penting dan kemudian menjadi standart internet yang dikeluarkan oleh i*nternet-activities-board* (IAB). IAB adalah badan yang memberikan perhatian terhadap perkembangan protokol yang berhubungan dengan TCP/IP dan internet.

Pada gambar diatas menunjukkan peristiwa penting yang berkaitan dengan perkembangan protokol TCP/IP, awalnya dimulai dengan percobaan 4-node ARPAnet pada tahung 1969 dan dilanjutkan dengan demo pada tahun 1972. Pada periode 1978 samapai 1980 terjadi perkembangan yang besar karena interaksi antara peneliti TCP/IP, peneliti *xerox-network-system* (XNS) dan pemakai yang bekerja di fasilitas Xerox di Palo Alto menghasilkan XNS's RIP *(routing-information-protocol)* yang kemudian digunakan pada Unix BSD. Karena pada saat itu Unix BSD sedang populer maka RIP digunakan secara luas pada jaringan TCP/IP, akibat banyaknya kalangan akademis, fasilitas penelitian, dll yang menggunakan Unix BSD.

Saat ini Internet yang menggantikan ARPAnet tidak hanya terdiri dari satu jaringan tetapi merupakan komglomesari dari berbagai macam jaringan, tetapi protokol yang paling banyak dipakai adalah TCP/IP, walaupun ada beberapa jaringan menggunakan protokol berbeda, milanya BITNET dan CREN yang menggunakan protokol IBM's SNA.

4.1. ARSITEKTUR PROTOKOL 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 OSI dan model DoD. 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, *mocrowave-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 protocol 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,

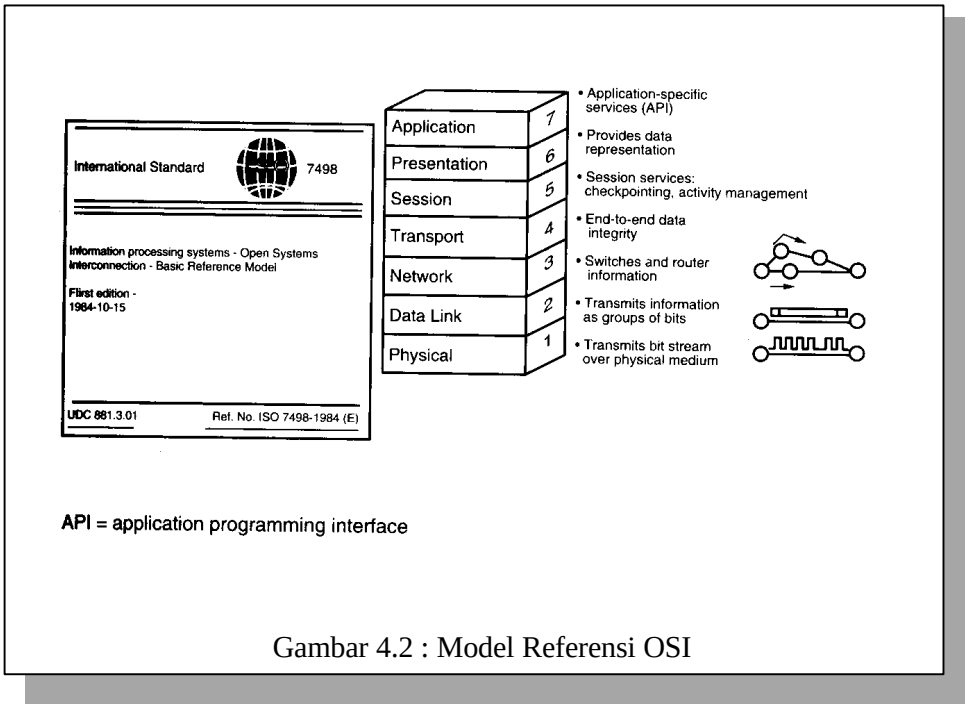
selanjutnya *network-protocol* menggunakan *physical-connections* untuk mengirim data.

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 *physical-connectios* 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).

4.1.1. Model OSI

Pada tahun 1978 *international-organization-of-standards* (ISO) menyusun standart OSI yang dapat dipakai untuk mengembangkan sistem terbuka dan sebagai referensi dengan sistem komunikasi data yang berbeda. Sistem jaringan yang dirancang menurut kerangka kerja dan spesifikasi OSI maka akan memiliki metode komunikasi yang saling kompatibel.



Gambar 4.2 : Model Referensi OSI

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

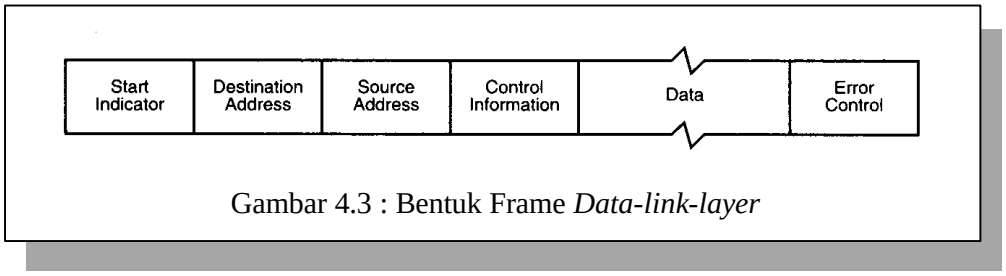
- Layer dibuat jika ketika diperlukan pemisahan level yang secara teori diperlukan.
- Masing-masing layer memiliki fungsi yang jelas.
- Setiap fungsi dari masing-masing layer telah ditentukan agar sesuai dengan standart protokol secara internasional.
- Batas kedua layer telah ditentukan untuk mengurangi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

Physical Layer

Physical-layer dipergunakan untuk mengtransmisikan data per bit melewati saluran komunikasi. Susunan bit tersebut mungkin mewakili pengiriman file atau rekord 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.).

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 *ethenet-frame*.
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 *error-control-field*, yang masing-masing memiliki fungsi tertentu.



Gambar 4.3 : 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.

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 *uniform-addresing-mecanism* (suatu mekanisme untuk pengalamatan sejenis).

Sebagai contoh jaringan lokal *Ethernet* dan *Token-ring* memiliki alamat *data-link* yang berbeda tipenya, untuk menghubungkan dua jaringan tersebut maka *diperlukan uniform-addressing-mechanism* yang dapat dimengerti oleh *Etrhernet* maupun *Token-ring*. Untuk jaringan yang berbasis *Novel-Netware* maka digunakan *internet-packet-exchange* (IPX) sebagai protokol *network-layer*, sedangkan jaringan berbasis TCP/IP digunakan *internet-protocol* (IP).

Transport Layer

Transport-layer menyediakan perbaikan untuk melayani *network-layer*. Lapisan ini membantu dalam meyakinkan pengiriman data dapat diandalkan dan menggabungkan data yang telah dikirim dari ujung ke ujung. Untuk meyakinkan pengiriman data dapat diandalkan *transport-layer* berdasarkan kepada mekanisme pengontrolan error yang disediakan oleh layer yang lebih rendah, jika layer yang dibawahnya tidah mampu untuk mengerjakan maka *transport-layer* akan bekerja lebih keras. Pada layer ini merupakan kesempatan terakhir untuk mengatasi error, tetapi pada kenyataannya *transport-layer* menyediakan pengiriman yang bebas error.

Transport-layer juga bertanggung jawab untuk membuat hubungan-hubungan secara logis pada sebuah hubungan jaringan, proses ini disebut *multiplexing* atau *time-*

sharing terjadi ketika nomer sambungan transport dibagi pada sambungan jaringan yang sama.

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 nomer transport dari *service-access-point*. Kadangkala untuk mengidentifikasi alamat tranport disebut dengan *socket* atau *port-number*.

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

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.

- ✓ Dialog control
- ✓ Token management
- ✓ Activity management

Pada beberapa protokol, umumnya hanya satu sisi yang mencoba untuk bekerja secara kritis, tetapi untuk menghindari kedua sisi mencoba bersama-sama untuk bekerja secara kirits maka dibutuhkan pengontrol mekanisme seperti menerapkan adanya *token*. Ketika menggunakan metode *token*, hanya sisi yang memegang *token* yang diijinkan untuk melakukan operasi. Untuk menentukan sisi mana yang memiliki *token* dan bagaimana *token* ditransfer diantara dua sisi disebut *token-management*.

Penggunaan kata token disini jangan membuat bingung dengan cora kerja *token-ring*, maka *token-management* merupakan konsep yang dimiliki layer yang lebih tinggi yaitu layer ke 5 pada model OSI, sedangkan cara kerja *token-ring* pada IBM dimiliki oleh layer 2 dan 1 pada model OSI.

Jika melakukan tugas transfer file antara 2 PC selama 1 jam, dan jaringan mengalami tabrakan sekitar setiap 30 menit, maka tidak akan didapatkan transfer file yang lengkap. Setelah masing-masing transfer tadi diabaikan akan dimulai lagi dari awal, untuk mengatasi masalah ini, maka file transfer dapat diperlakukan sebagai aktifitas tunggal dengan memasukkan *checkpoint* pada aliran data. Pada metode ini jika jaringan mengalami tabrakan *session-layer* dapat mensinkronkan pada *checkpoint* sebelumnya. Cara kerja dengan mengatur keseluruhan aktifitas ini disebut dengan *activity-management*.

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).

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 :

- ✓ Protokol menyediakan fasilitas penyedia file secara remote, seperti *open, close, read, save, write* dan membagi akses ke file tersebut.
- ✓ Akses transfer file dan database secara remote.
- ✓ Fasilitan menangani pesan untuk aplikasi *e-mail*.
- ✓ Fasilitas direktori global dan lokal pada sumber daya jaringan.
- ✓ Cara yang seragam untuk menangani berbagai monitoring sistim dan peralatan.
- ✓ Menjalankan perintah secara remote.
- ✓

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-control-protocol* (NCP) sedangkan untuk TCP/IP adalah FTP, SMTP dan TELNET.

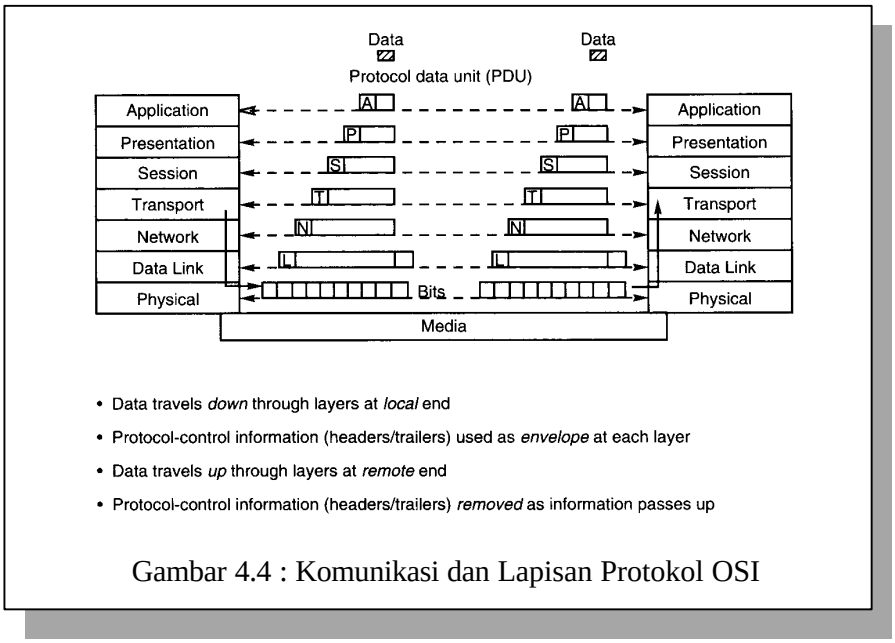
4.1.2. Protokol Envelope

Model referensi OSI dapat dipakai untuk membantu mengetahui pertukaran data diantara dua sistem PC, yang pada terminologi OSI disebut dengan *end-system*. Diumpamakan ada dua *end-system* seperti ditunjukkan pada gambar 4.4. sebuah pesan (data) dikirimkan dari *end-system* yang satu ke *end-system* yang lain. *Application-layer* memproses pesan tersebut dan menambahkan beberapa bit aplikasi sebagai informasi *header* pada pesan tersebut.

Didalam terminologi OSI bit aplikasi ini disebut dengan *protocol-control-information* (PCI) yang berisi informasi tentang proses yang telah dilakukan terhadap pesan tersebut. Didalam PCI tersebut dapat berisi informasi untuk mengindentifikasi *application-entities*, yaitu alamat *source* dan *destination* dari *application-layer*. PCI ditambah dengan pesan asli disebut dengan *aplication-protocol-data-unit* (APDU).

Selanjutnya APDU dikirim ke lapisan *presentation-layer*, maka *presentation-layer* menambahkan PCI ke data (APDU) yang diterima dari *aplication-layer*, dan menghasilkan *presentation-protocol-data-unit* (PPDU), pada posisi ini pesan asli (data) terbungkus pada APDU, selanjutnya APDU terbungkus oleh PPDU.

Setiap lapisan selalu menambahkan informasi dalam headernya kepada pesan yang diterima dari lapisan sebelumnya, proses ini mirip dengan mengambil data dan meletakkan didalam *envelope* (sampul pembungkus). Proses ini belangsung terus sampai turun kepada *physical-layer*.



Gambar 4.4 : Komunikasi dan Lapisan Protokol OSI

Pada *data-link-layer* biasanya ditambahkan *trailer-field* yang ditambahkan pada data, yang berisi pengecekan kesalahan yang digunakan untuk melingkupi sebagai *fram*e dari *data-link-layer* yang dibangkitkan oleh mekanisme hardware *data-link-layer* (*network-interface-card*). *Trailer-filed* ini ditambahkan pada bagian akhir proses karena mekanisme hardware menghitung pengecekan kesalahan pada saat data akan dikirimkan keluar menuju saluran transmisi. Pengecekan kesalahan yang umum dipakai pada LAN dan WAN adalah CRC.

Pada *physical-layer* informasi dalam header yang ditambahkan dapat diambil dari indikasi yang disampaikan pada *receiver* tentang packet yang diterima. Pada jaringan *ethernet* digunakan 56-*bit* sebagai preamble/pembukaan; yang akan digunakan oleh *receiver* untuk mensinkronkan diri. Pada sisi receiver maka *physical-layer* menerima sederetan bit dan memilah bit sinkronisasi untuk *physical-layer* dan selanjutnya mengirimkan data kepada *data-link-layer*.

Pada *data-link-layer* data bit yang diterima dikelompokkan kedalam frame kemudian diperiksa dengan metoda CRC apakah tidak ada kesalahan. Sesudah *data-link-layer* memproses *frame* yang diterima, selanjutnya memilah menjadi *data-link* PCI yang berisi header *data-link* dan CRC, selanjutnya mengirimkan NPDU (*network*-PDU) ke *network-layer* untuk diproses.

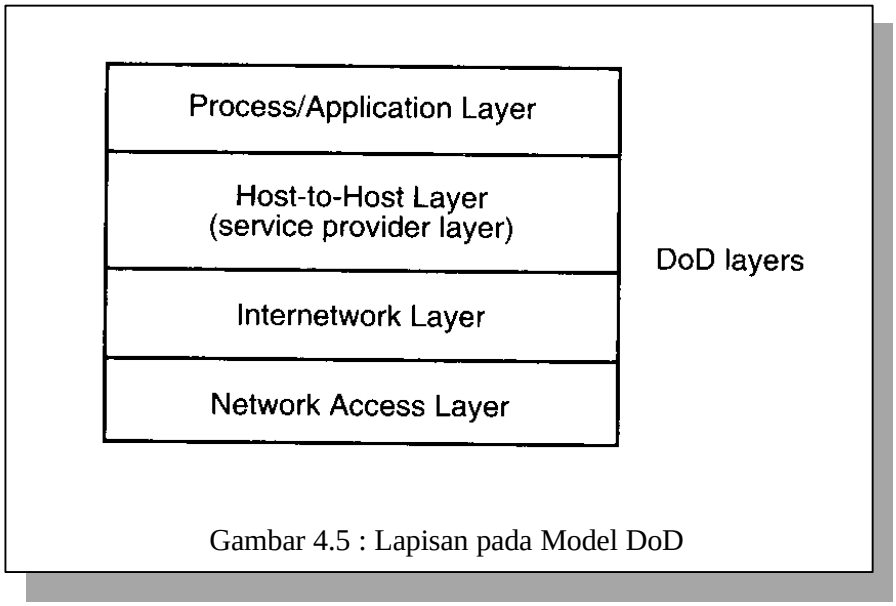
Setiap layer *end-system* pada receiver, PDU diproses berdasarkan informasi dalam header PCI untuk setiap lapisan tersebut. Sesudah PCI dihapus selanjutnya data (PDU) dikirimkan ke lapisan yang lebih atas, proses ini belangsung terus hingga *aplication-layer* selesai melakukan proses hingga data yang asli didapatkan.

Jika beberapa lapisan dipelajari maka pada *end-system* transmitter dan receiver (lokal dan remote), lapisan pada model OSI kelihatan seperti lapisan yang berkomunikasi dengan lapisan yang sepadan (*peer-layer*). Untuk itu lapisan dalam model OSI dijabarkan kedalam protokol *peer-to-peer*. Sebagai contoh adalah *transport-layer* pada kedua *end-system* jika berkomunikasi antar keduanya maka *transport-layer* membutuhkan lapisan *infra-structure* dibawahnya yaitu *network-layer, data-ling-layer,* dan *physical-layer*.

Model OSI menyediakan konsep wawasan berkomunikasi yang mudah untuk dimengerti, tetapi pada kenyatanannya setiap lapisan yang ada tidak mudah untuk dibentuk seperti pada model OSI. Contohnya jarak antara kedua lapisan *aplication-layer* dan *presentation-layer* tidak jelas, karena beberapa fungsi dari *presentation-layer* seperti *encoding* dan *decoding* data dapat dilakukan pada *aplication-layer* dengan memanggil bahasa program yang dapat melakukan fungsi *encoding* dan *decoding* data.

4.1.3. Model DoD

Walaupun model OSI telah dibuat tahun 1979, tetapi konsep lapisan untuk protokol telah ada jauh sebelum diformalisasi kedalam model OSI. Salah satu contoh protokol yang lebih dulu berhasil menggunakan konsep lapisan untuk protokol adalah TCP/IP, karena TCP/IP memiliki pertalian dengan *Departement of Defense*, sehingga lapisan protokol TCP/IP disebut dengan model DoD.



Gambar 4.5 : Lapisan pada Model DoD

Lapisan paling bawah adalah *network-access-layer* yang mewakili komponen penghubung dengan saluran fisik seperti kabel, transceiver, NIC, LAN *access-protocol* (seperti CSMA/CD untuk *ethernet* dan *token-access* untuk *token-ring*), *token-bus,* dan FDDI. Network-access-layer dipergunakan oleh *internet-layer*.

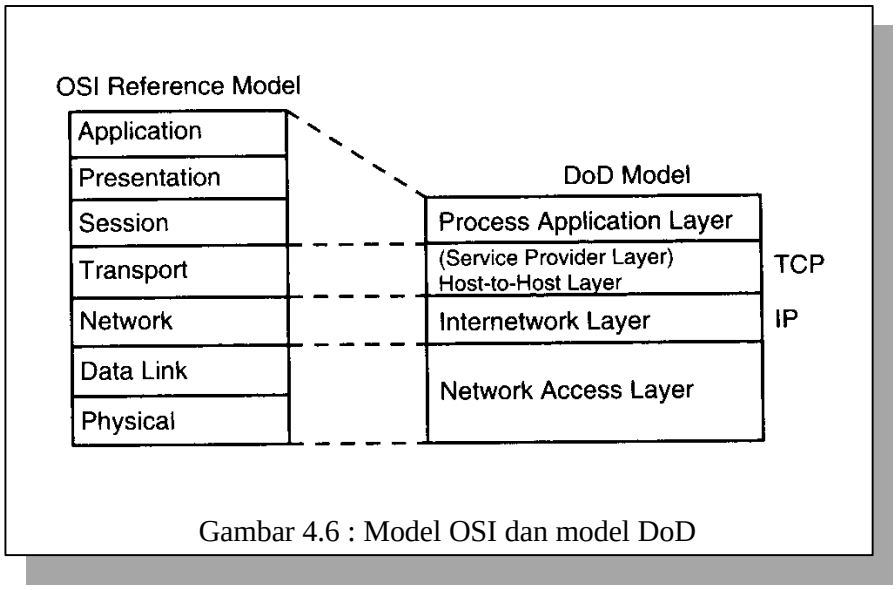
Internet-layer bertanggung jawab untuk menyediakan *logical-address* bagi *physical-network-interface*. Pada model DoD yang merupakan implementasi dari *internet-layer* adalah *internet-protocol* (IP) yang bertanggung jawab memetakan *logical-address* dan *physical-address* yang disediakan oleh *network-access-layer*, dengan menggunakan *address-resolution-protocol* (ARP) dan *Reverse-address-resolution-protocol* (RARP).

Segala masalah, *diagnostic-information*, dan keadaan yang berkenaan dengan protokol IP dilaporkan oleh protokol terpisah yang disebut *internet-control-message-protocol* (ICMP) yang juga beroperasi pada *internet-layer*. *Internet-layer* juga memiliki perhatian kepada *routing* (pengaturan lintasan) data paket diantara beberapa host dan jaringan. Layer lebih tinggi yang menggunakan *internet-layer* adalah *host-to-host-layer*.

Protokol *Host-to-host* mengatur hubungan antara dua *host* pada jaringan. Pada model DoD *host-to-host-protocol* diimplementasikan menjadi *transmission-control-protocol* (TCP) dan *user-datagram-protocol* (UDP). Protokol TCP bertanggung jawab kepada keandalan hubungan, hubungan yang simultan, dan *full-duplex*. Keandalan hubungan dapat berarti TCP peduli kepada koreksi kesalahan transmisi dengan mengirim kembali bagian data yang mengalami error. *Process/Application-layer* yang menggunakan TCP tidak memiliki kepedulian kepada benar atau salahnya data yang diterima, karena urusan ini telah ditangani oleh TCP.

Process/Application-layer menyediakan aplikasi yang dapat menggunakan protokol pada lapisan *host-to-host-layer* (TCP dan UDP), contoh dari aplikasi ini adalah *file-transfer-protocol* (FTP), *terminal-emulation* (TELNET), *electronic-mail* (SMTP), dan *simple-network-management-protocol* (SNMP).

Jika membandingkan secara fungsionalitas maka model DoD dan model OSI memiliki kemiripan, seperti terlihat pada gambar berikut.

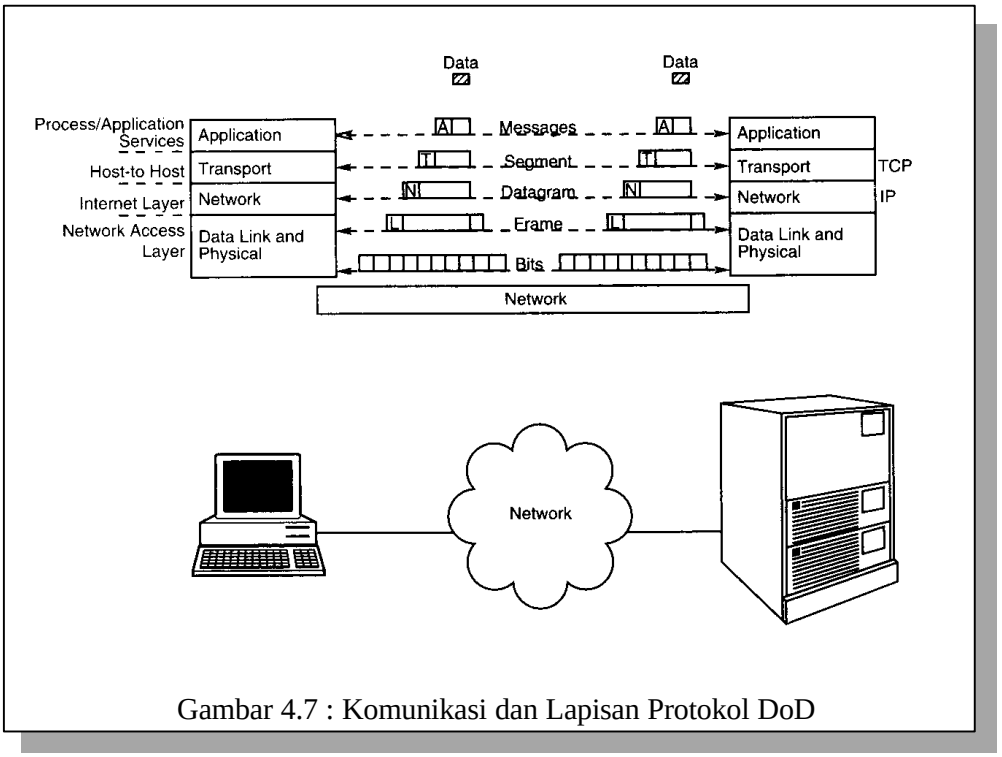


Gambar 4.6 : Model OSI dan model DoD

Network-access-layer pada model DoD berhubungan dengan dua lapisan pada model OSI yaitu : *physical-layer* dan *data-link-layer*. *Internet-layer* pada model DoD berhubungan dengan *network-layer* pada model OSI. *Host-to-host-layer* pada model DoD berhubungan dengan *transport-layer* pada model OSI. *Process/Applcation-layer* pada model DoD berhubungan dengan tiga lapisan pada model OSI yaitu : *session-layer*, *presentation-layer* dan *application-layer*.

4.1.4. Aliran Data pada Jaringan TCP/IP

Pada gambar 4.7 menunjukkan komunikasi diantara dia host menggunakan model DoD. Pada gambar 4.7 dibawah dapat dilihat kemiripan dengan gambar 4.4, yang menggambarkan kemiripan antara model OSI dan DoD.



Gambar 4.7 : Komunikasi dan Lapisan Protokol DoD

Data yang dikirimkan oleh host dibungkus oleh *header* protokol pada lapisan p*rocess/application*, kemudian data yang dihasilkan dibungkus pada lapisan *host-to-host* (TCP/UDP), kemudian data yang dihasilkan dibungkus lapisan *Internetwork* (IP), sampai akhirnya data dari protokol lapisan *Internetwork* dibungkus oleh protokol lapisan *network-access*.

Ketika data yang telah dibungkus diterima oleh host yang dituju, maka data tersebut dilepas pembungkusnya oleh oleh setiap lapisan DoD dan data diberikan kepada lapisan diatasnya sampai data yang asli didapatkan.

Perbedaan penting antara OSI dan DoD adalah perbedaan terminologi yang dipergunakan untuk menjelaskan data pada tiap lapisan. Pada model OSI dipergunakan terminologi *protocol-data-unit* (PDU) untuk menyatakan data pada tiap lapisan, sedangkan pada model DoD dipakai *message* untuk menyatakan data pada lapisan *process/application*, *segment* dipakai untuk menyatakan data pada lapisan

host-to-host, datagram dipakai untuk menyatakan data pada lapisan *internetwork* dan *frame* dipakai untuk menyatakan data pada lapisan *network-access*.

4.2. RANGKAIAN PROTOCOL TCP/IP

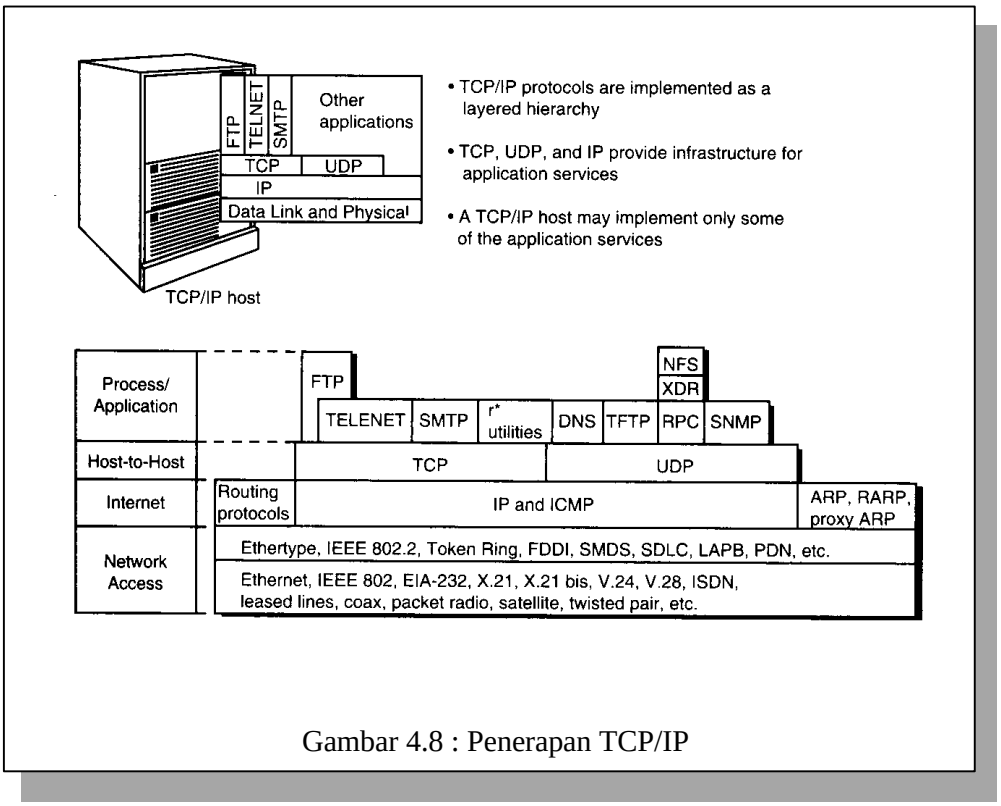
Protokol dan aplikasi TCP/IP atau sering disebut dengan protokol TCP/IP dan rangkaiannya (TCP/IP protocol suite) sebenarnya didefinisikan berdasarkan request-for-comment (RFCs) dan sejumlah standart yang ada. Pada tabel 4.1 dibawah menunjukkan beberapa RFCs dan nomer standart yang diapakai pada model DoD. Meskipun semua standart internet memiliki nomer RFC, tetapi tidak semua RFC merupakan bagian dari standart internet secara resmi, karena beberapa RFC ditujukan kepada eksperimental dan usulan, beberapa diantaranya berupa tutorial. RFC yang menjelaskan standart protokol secara resmi adalah RFC nomer 1600 sedangkan pada tabel dibawah ini menggambarkan beberapa RFC dan standart yang telah diketahui sebelumnya.

Tabel 4.1
Standart yang dipakai pada DoD

Nama Standart	Nomer Standart	RFC	Lapisan model DoD
File Transfer Protocol (FTP)	9	959	Process/Application
Terminal Emulation Network Protocol (TELNET)	8	854, 855	Process/Application
Trivial File Transfer Protocol (TFTP)	33	1350	Process/Application
Simple Mail Transfer Protocol (SMTP)	10	821	Process/Application
Simple Network Management Protocol (SNMP)	15	1157	Process/Application
Domain Name System	13	1034, 1035	Process/Application
Mail Routing and the Domain System (DNS-MX)	14	974	Process/Application
Transmission Control Protocol (TCP)	7	793	Host-to-Host
User Datagram Protocol (UDP)	6	768	Host-to-Host
Internet Protocol (IP)	5	791	Internet
IP subnet extension	5	950	Internet
IP Broadcast Datagram	5	919	Internet
IP Broadcast Datagram with Subnets	5	922	Internet
Internet Control Message Protocol (ICMP)	5	792	Internet

4.2.1. Penerapan TCP/IP

Rangkaian protokol TCP/IP berserta kelengkapannya termasuk aplikasinya dapat di gunakan untuk berbagai teknologi jaringan fisik, seperti WAN, LAN, radio, saluran ISDN, *satellite-link*, dll. Pada gambar 4.8 menunjukkan beberapa bagian penyedia aplikasi yang terdapat pada TCP/IP dan berbagai macan *network-access-rptocol* yang menunjang TCP/IP. Aplikasi yang disediakan terlihat berhubungan dengan model DoD. Penerapannya dapat mendukung beberapa aplikasi yang disediakan atau beberapa protokol, seperti ARP, RARP, proxy ARP, dan routing-protocol, sebagai contoh pada gambar tersebut lapisan host TCP/IP hanya mendukung sebagai penyedia palikasi FTP, TELNET dan SMTP.



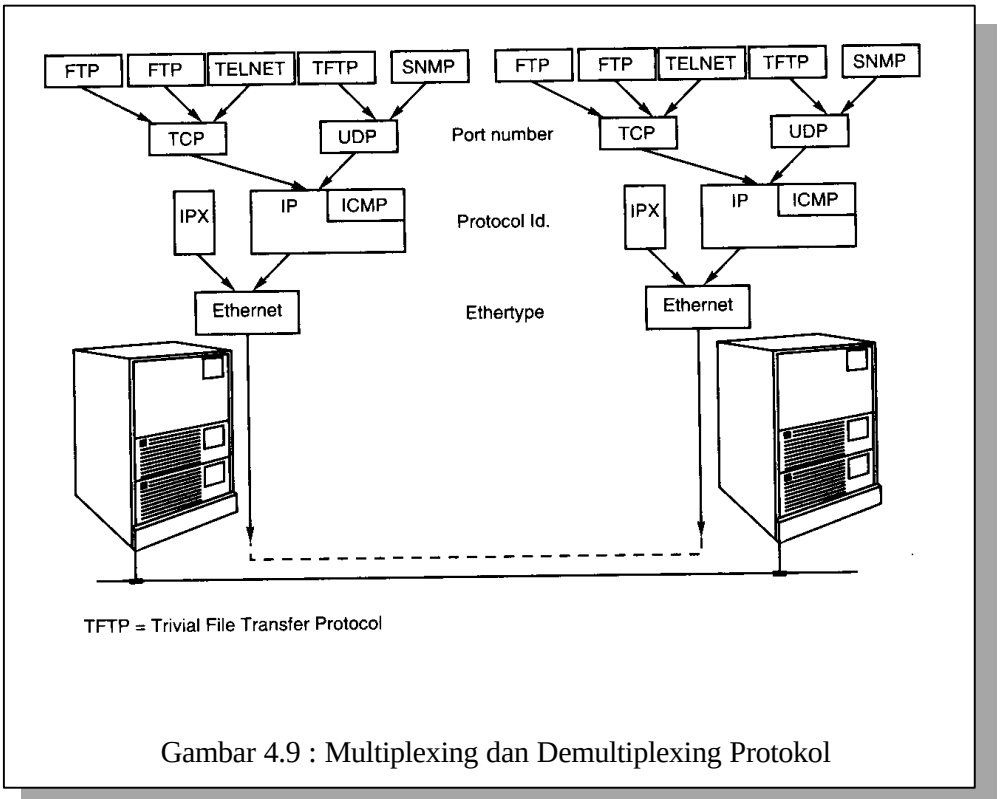
Gambar 4.8 : Penerapan TCP/IP

Penyedia aplikasi yang dijabarkan dalam lapisan process/application dapat dibentuk oleh *transport* atau protokol lapisan *network* yang berbeda. Contohnya seperti SMTP secara umum dikaitkan dengan TCP/IP tetapi dapat berjalan pula pada protokol IPX (RFC 1420), protokol Apple-talk (RFC(1419) dan protokol OSI (RFC1418). Hal yang sama terjadi pada beberapa aplikasi yang berdasarkan non-TCP/IP, seperti *netware-core-protocol* (NCP) yang secara umum berjalan pada

protokol jaringan IPX dan X.500 yang dipakai pada protokol OSI, dapat berjalan pula pada TCP/IP. Netware/IP adalah produk dari *Novell* yang menyeriakan service NCP pada TCP/IP, dan *ISO-development-environment* (ISODE) menyediakan service X.500 pada TCP/IP.

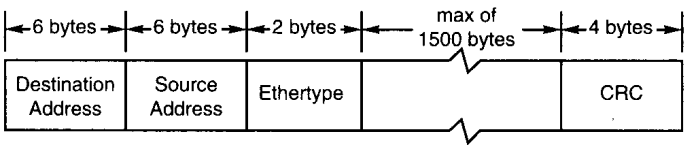
4.2.2. Multiplexing dan Demultiplexing

Pada gambar 4.9 menunjukkan komunikasi TCP/IP antara dua *host*, masing-masing *host* menjalankan penyedia aplikasi TCP/IP seperti FTP, TELNET, (TFTP) dan SMTP.



Gambar 4.9 : Multiplexing dan Demultiplexing Protokol

Sebagai contoh pada gambar diatas 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 4.10) yang merupakan bagian dari *frame*-nya Ethernet yang dipergunakan sebagai protokol multiplexing dan demultiplexing pada *data-link-layer*.



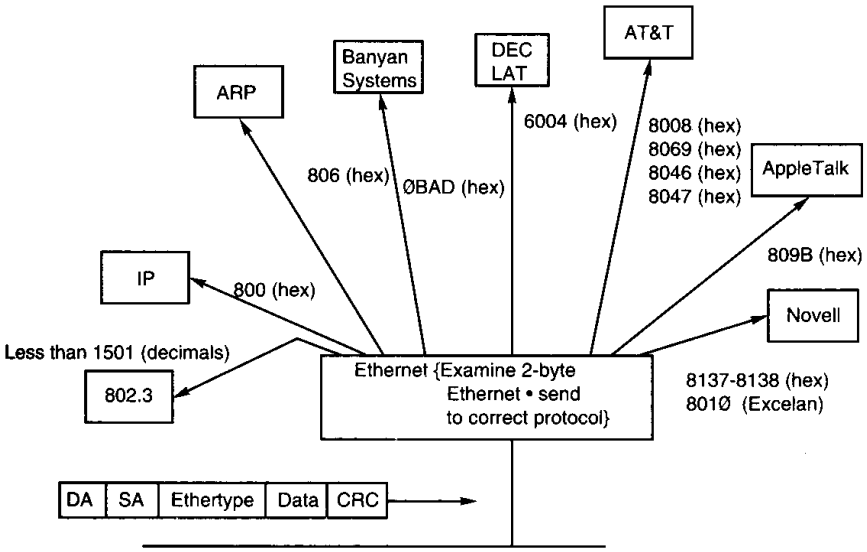
Gambar 4.10 : 2-byte Ethertype Field

Pada tabel 4.2 dibawah ditampilkan beberapa nilai dari ethertype-field yang dipakai pada beberapa protokol jaringan.

Tabel 4.2
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

Ethertype-field pada *data-link-layer* memberikan kemungkinan untuk mengadakan *multiplexing* pada transmitter untuk beberapa protokol jaringan dan melakukan *demultiplexing* pada receiver, seperti telihat pada gambar 4.11.



Gambar 4.11 : Multiplexing dan Demultiplexing menggunakan Ethertype

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 4.3 dibawah ditampilkan beberapap nila *protocol-Id*, yang dapat dilihat pula pada Server Netware pada file *sys:etc/protocol* atau pada file */etc/protocols* pada host Unix.

Tabel 4.3
Nilai Protocol-Id untuk Multiplexing/Demultiplexing pada Network-Layer

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 4.4 diberikan beberapa nilai *port-number* yang digunakan untuk multiplexing/demultiplexing pada modul protokol TCP, dan pada tabel 4.5 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 4.4 dan 4.5 terdapat aplikasi TCP/IP dengan port-number yang berbeda, hal ini disebabkan aplikasi yang disediakan berlaku sama untuk TCP dan UDP.

Nilay port-number untuk modul protokol TCP dan UDP pada suatu sistem dapat dilihat pada file *sys:/etc/services* untuk server Netware dan */etc/services* pada host Unix.

Tabel 4.4
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...

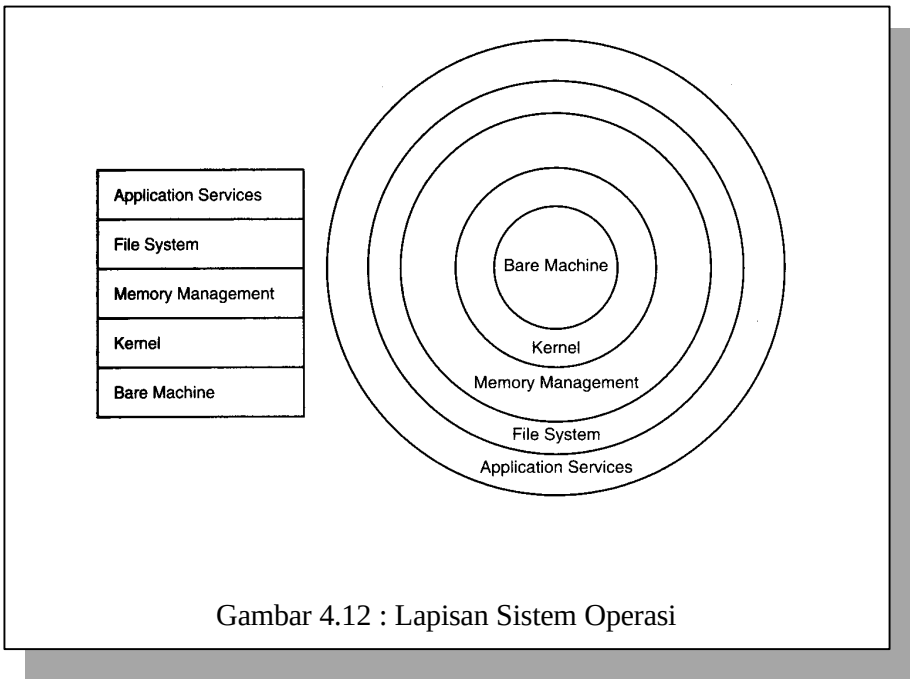
Tabel 4.5
Nilai Port-Number untuk Multiplexing/Demultiplexing pada UDP

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...

4.2.3. 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 4.12 dibawah. Jika dibandingkan dengan gambar 4.2 maka terlihat jelas perbedaannya. 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 jaringan komunikasi merupakan bagian fasilitas yang diberikan dari OS dalam mengatur I/O.



Gambar 4.12 : 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 :

- ✓ Bagian dari kernel OS (inti OS)
- ✓ *Device-driver* (pengendali peralatan)
- ✓ Proses yang terjadi dalam aplikasi

Pada beberapa OS seperti Unix dan Netware, maka TCP/IP di terapkan 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-link-libraries* (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.

4.3. IP Address

IP Address teridri dari 32 bit yang merupakan alamat yang dibuat untuk merepresentasikan alamat *Internet Layer*, yang dalam penulisannya maka *IP Address* dibagi atas 4 segmen, tiap segmen terdiri dari 8 bit. Jika direpresentasikan dalam bilangan biner maka variasi address yang dapat digunakan oleh *host* dalam jaringan TCP/IP, adalah dari 00000000.00000000.00000000.00000000 sampai dengan 11111111.11111111.11111111.11111111. Jadi, jaringan TCP/IP secara teoritis mampu mengintegrasikan sebanyak 2^{32} (4 milyar lebih) komputer. Pada kenyataannya ada address-address khusus yang dipakai untuk keperluan tertentu, sehingga tidak boleh dipakai oleh *host*. Setiap *interface* yang dalam *node* yang melakukan operasi pada *Internet layer* (*IP stack*) akan memiliki *IP Address.*

4.3.1. Pembagian Kelas Address.

IP Address yang berjumlah 4 milyar lebih itu diklasifikasikan menjadi beberapa kelas tertentu. Pembagian kelas-kelas ini ditujukan untuk mempermudah alokasi *IP Address*, baik untuk *host*/jaringan tertentu maupun untuk keperluan tertentu. Untuk memudahkan pembacaan dan penulisan, *IP Address* biasanya direpresentasikan dalam bilangan desimal. Jadi, range *address* di atas dapat diubah menjadi *address* 0.0.0.0 sampai *address* 255.255.255.255. Nilai desimal dari *IP Address* inilah yang dikenal dalam pemakaian sehari-hari. Beberapa contoh *IP Address* adalah :

167.205.9.35
167.205.169.113
10.252.1.1

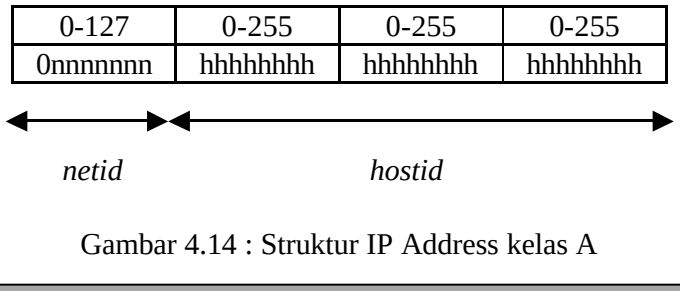
Ilustrasi *IP Address* dalam bilangan desimal dan biner dapat dilihat pada gambar berikut :

Desimal	167	205	9	35
Biner	10100111	11001101	00001001	00100011

Gambar 4.13 : IP Address dalam bilangan Desimal dan Biner

IP Address terdiri dari 2 bagian yaitu *network bit (netid)* dan *Host bit (hostid), netid* menyatakan alamat jaringan dan *hostid* menyatakan alamat host. *Netid* berperan dalam identifikasi suatu *network* dari *network* yang lain, sedangkan *hostid* berperan dalam identifikasi *host* dalam suatu *nertwork*. Jadi, seluruh *host* yang tersambung dalam jaringan yang sama memiliki *netid* yang sama. Sebagian dari bit-bit bagian awal dari *IP Address* merupakan *netid*, sedangkan sisanya untuk *hostid*. Garis pemisah anatara bagian *netid* dan *hostid* tidak tetap, bergantung kepada kelas *network.* Ada 3 kelas address yang utama dalam TCP/IP, yakni kelas A, kelas B dan kelas C. Perangkat lunak *Internet Protocol* menentukan pembagian jenis kelas ini dengan menguji beberapa bit pertama dari *IP Address.* penentuan kelas ini dilakukan dengan cara berikut :

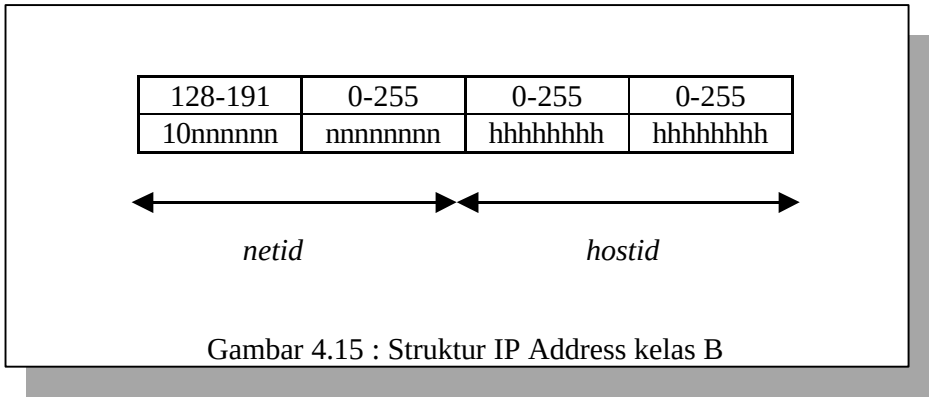
- Jika bit pertama dari *IP Address* adalah 0, address merupakan *network* kelas A. Bit ini dan 7 bit berikutnya (8 bit pertama) merupakan *netid* sedangkan 24 bit terakhir merupakan *hostid*. Dengan demikian hanya ada 128 network kelas A, yakni dari nomor 0.xxx.xxx.xxx sampai 127.xxx.xxx.xxx, tetapi setiap network dapat menampung lebih dari 16 juta *host* (256^3). Sebagai ilustrasi dapat dilihat pada gambar berikut :



Gambar 4.14 : Struktur IP Address kelas A

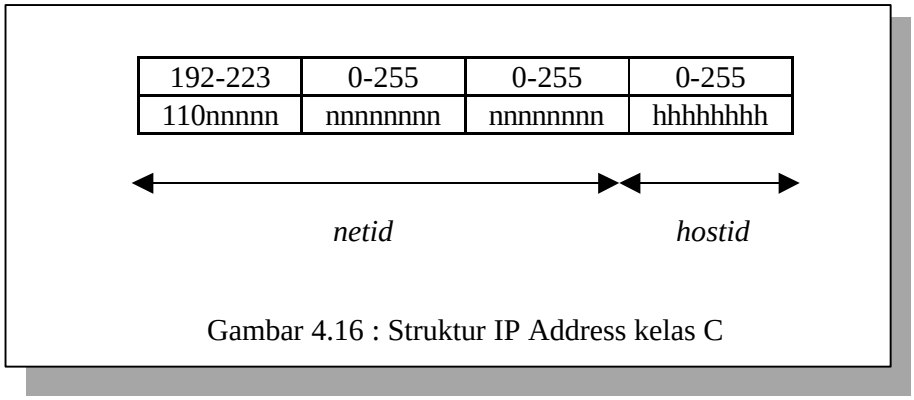
- Jika 2 bit pertama dari *IP Address* adalah 10, address merupakan *network* kelas B. Dua bit ini dan 14 bit berikutnya (16 bit pertama) merupakan *netid* sedangkan 16 bit terakhir merupakan *hostid*. Dengan demikian terdapat lebih dari 16 ribu *network* kelas B (64 x 256), yakni dari *network* 128.0.xxx.xxx - 191.255.xxx.xxx.

Setiap *network* kelas B mampu menampung lebih dari 65 ribu *host* (256^2). Ilustrasinya dapat dilihat pada gambar berikut.



Gambar 4.15 : Struktur IP Address kelas B

- Jika 3 bit pertama dari *IP Address* adalah 110, address merupakan *network* kelas C. Tiga bit ini dan 21 bit berikutnya (24 bit pertama) merupakan *netid* sedangkan 8 bit terakhir merupakan *hostid*. Dengan demikian terdapat lebih dari 2 juta *network* kelas C (32 x 256 x 256), yakni dari nomor 192.0.0.xxx sampai 223.255.255.xxx. Setiap *network* kelas C hanya mampu menampung sekitar 256 *host*. Ilustrasinya dapat dilihat pada gambar berikut.



Gambar 4.16 : Struktur IP Address kelas C

Kelas berikutnya dari *IP Address* adalah kelas D dan E. Kelas D merupakan *IP Address* yang dialokasikan sebagai *multicast address.* Alokasi nomor untuk address *multicast* ini adalah 224.0.0.0 samapai dengan 239.255.255.255. Kelas inilah yang digunakan sebagai address pada *multicast backbone* untuk aplikasi *video conference*. Sedangkan kelas E merupakan sisanya (240.0.0.0 sampai 255.255.255) yang dialokasikan untuk eksperimen. Sebelum membahas *multicast address*, akan dibahas lebih dahulu beberapa address khusus untuk tujuan tertentu.

4.3.2. Address Khusus

Selain address yang dipergunakan untuk mengenal *host*, ada beberapa jenis address yang digunakan untuk keperluan khusus dan tidak boleh digunakan untuk pengenal *host*. Address tersebut adalah :

Network Address

Address ini digunakan untuk mengenali suatu network pada jaringan Internet. Misalkan untuk *host* dengan *IP Address* kelas B 167.205.9.35. Tanpa memakai subnet, *network address* dari *host* ini adalah 167.205.0.0. Address ini didapat dengan membuat seluruh bit *host* pada 2 segmen terakhir menjadi 0. Tujuannya adalah untuk menyederhanakan informasi *routing* pada Internet. *Router* cukup melihat network address (167.205) untuk menentukan ke router mana datagram tersebut harus dikirimkan.

Contoh untuk kelas C, *network address* untuk *IP Address* 202.152.1.250 adalah 202.152.1.0. Analogi yang baik untuk menjelaskan fungsi *network address* ini adalah dalam pengolahan surat pada kantor pos. Petugas penyortir surat pada kantor pos cukup melihat kota tujuan pada alamat surat (tidak perlu membaca seluruh alamat) untuk menentukan jalur mana yang harus ditempuh surat tersebut. Pekerjaan *"routing"* surat-surat menjadi lebih cepat. Demikian juga halnya dengan *router* di Internet pada saat melakukan *routing* atas datagram.

Broadcast Address

Address ini digunakan untuk mengirim/menerima informasi yang harus diketaui oleh seluruh *host* yang ada pada suatu *network.* Seperti diketahui, setiap datagram *IP* memiliki *header* alamat tujuan berupa *IP Address* dari *host* yang akan dituju oleh datagram tersebut. Dengan adanya alamat ini, maka hanya *host* tujuan saja yang memproses datagram tersebut, sedangkan *host* lain akan mengabaikannya. Bagaimana jika suatu *host* ingin mengirim datagram kepada seluruh *host* yang ada pada networknya? Tidak efisien jika ia harus membuat replikasi datagram sebanyak jumlah *host* tujuan.

Pemakaian *bandwidth* akan meningkat dan beban kerja *host* pengirim bertambah, padahal isi datagram-datagram tersebut sama. Oleh karena itu, dibuat konsep *broadcast address. Host* cukup mengirim ke alamat *broadcast,* maka seluruh *host* yang ada pada *network* akan menerima datagram tersebut. Konsekuensinya, seluruh *host* pada *network* yang sama harus memiliki *broadcast address* yang sama dan address tersebut tidak boleh digunakan sebagai *IP Address* untuk *host* tertentu. Jadi, sebenarnya setiap *host* memiliki 2 address untuk menerima datagram : pertama adalah *IP Address* yang bersifat unik/tersendiri dan kedua adalah *bradcast address* pada network tempat host tersebut berada.

Broadcast address diperoleh dengan membuat bit-bit *host* pada *IP Address* menjadi bit 1. Jadi, untuk *host* dengan *IP Address* 167.205.9.35 atau 167.205.240.2, adalah 167.205.255.255 (2 segmen terakhir dari *IP Address* tersebut dibuat berharga

11111111.11111111, sehingga secara desimal terbaca 255.255). Jenis informasi yang di*broadcast* biasanya adalah informasi *routing*.

Multicast Address

Kelas address A, B dan C adalah address yang digunakan untuk komunikasi antar *host*, yang menggunakan datagram-datagram *unicast*. Artinya, datagram/paket memiliki address tujuan berupa satu *host* tertentu. Hanya host yang memiliki address tujuan berupa satu *host* tertentu. Hanya host yang memiliki *IP Address* sama dengan *destination address* pada datagram yang akan menerima datagram tersebut, sedangkan *host* lain akan mengabaikannya. Jika datagram ditujukan untuk seluruh *host* pada suatu jaringan, maka *field* address tujuan ini akan berisi alamat *broadcast* dari jaringan yang bersangkutan. Dari dua mode pengiriman ini *(unicast dan broadcast),* muncul pula mode ke tiga. Diperlukan suatu mode khusus jika suatu *host* ingin berkomunikasi dengan beberapa *host* sekaligus *(host group),* dengan hanya mengirinkan suatu datagram saja. Namun berbeda dengan mode *broadcast*, hanya *host-host* yang tergantung dalam suatu group saja yang akan menerima datagram ini, sedangkan *host* lain tidak akan terpengaruh. Oleh karena itu, dikenalkan konsep *multicast.* Pada konsep ini, setiap *group* yang menjalankan aplikasi bersama mendapatkan satu *multicast address.* Struktur kelas *multicast address* dapat dilihat pada gambar berikut.

224-239	0-255	0-255	0-255
1110xxxx	xxxxxxxx	xxxxxxxx	xxxxxxxx

Gambar 4.17 : Struktur IP Address kelas *Multicast Address*

Untuk keperluan *multicast*, maka sejumlah *IP Address* dialokasikan sebagai *multicast address*, yaitu jika suatu struktur *IP Address* memiliki bentuk 1110xxx.xxxxxxxx.xxxxxxxx.xxxxxxxx (bentuk desimal 224.0.0.0 sampai 239.255.255.255), maka *IP Address* merupakan *multicast address*. Alokasi ini ditujukan untuk keperluan *group*, bukan untuk *host* seperti pada kelas A, B dan C. Anggota *group* adalah *host-host* yang ingin bergabung dalam *group* tersebut. Anggota ini juga tidak terbatas pada jaringan di satu *subnet*, namun bisa mencapai seluruh dunia. Karena menyerupai suatu *backbone*, maka jaringan *muticast* ini dikenal pula sebagai *Multicast Backbone (Mbone).*

4.3.3. Subnetting

Untuk beberapa alasan yang menyangkut efisiensi *IP Address*, mengatasi masalah topologi network dan organisasi, network administrator biasanya melakukan subnetting. Esensi dari subnetting adalah “memindahkan” garis pemisah antara bagian network dan bagian host dari suatu *IP Address*. Beberapa bit dari bagian host dialokasikan menjadi bit tambahan pada bagian network. Address satu network menurut struktur baku dipecah menajdi beberapa subnetwork. Cara ini menciptakan sejumlah network tambahan, tetapi mengurangi jumlah maksimum host yang ada dalam tiap network tersebut.

Subnetting juga dilakukan untuk mengatasi perbedaan hardware dan media fisik yang digunakan dalam suatu network. Router IP dapat mengintegrasikan berbagai network dengan media fisik yang berbeda hanya jika setiap network memiliki address network yang unik. Selain itu, dengan subnetting, seorang Network Administrator dapat mendelegasikan pengaturan host address seluruh departemen dari suatu perusahaan besar kepada setiap departemen, untuk memudahkannya dalam mengatur keseluruhan network.

44	132	1	20
00101100	10000100	00000001	00010100

IP Address

255	255	0	0
11111111	11111111	00000000	00000000

Subnet Mask

44	132	0	0
00101100	10000100	00000000	00000000

Network Address

44	132	255	255
00101100	10000100	11111111	11111111

Broadcast Address

Gambar 4.18 : Subnetting 16 bit pada IP Adress kelas A

Suatu subnet didefinisikan dengan mengimplementasikan masking bit (subnet mask) kepada *IP Address*. Struktur subnet mask sama dengan struktur IP Address, yakni terdiri dari 32 bit yang dibagi atas 4 segmen. Bit-bit dari *IP Address* yang ”ditutupi” (masking) oleh bit-bit subnet mask yang aktif dan bersesuaian akan diinterprestasikan sebagai network bit. Bit 1 pada subnet mask berarti mengaktifkan

masking (on), sedangkan bit 0 tidak aktif (off). Sebagai contoh kasus, mari kita ambil satu *IP Address* kelas A dengan nomor 44.132.1.20. Ilustrasinya dapat dilihat pada gambar 4.18 diatas.

Dengan aturan standard, nomor *netid* untuk *IP Address* ini adalah 44 dan nomor *hostid* adalah 132.1.20. Network tersebut dapat menampung maksimum lebih dari 16 juta host yang terhubung langsung. Misalkan pada address ini akan diimplementasikan *subnet mask* sebanyak 16 bit 255.255.0.0. (Hexa = FF.FF.00.00 atau Biner = 11111111.11111111.00000000.00000000). Perhatikan bahwa pada 16 bit pertama dari *subnet mask* tersebut berharga 1, sedangkan 16 bit berikutnya 0. Dengan demikian, 16 bit pertama dari suatu *IP Address* yang dikenakan *subnet mask* tersebut akan dianggap sebagai *netid* (*network bit)*. Nomor network akan berubah menjadi 44.132 dan nomor host menjadi 1.20. Kapasitas maksimum host yang langsung terhubung pada network menjadi sekitar 65 ribu *host*.

Subnet mask di atas identik dengan standars *IP Address* kelas B. Dengan menerapkan *subnet mask* tersebut pada satu network kelas A, maka dapat dibuat 256 network baru dengan kapasitas masing-masing subnet setara network kelas B. Penerapan subnet yang lebih jauh seperti 255.255.255.0 (24 bit) pada kelas A akan menghasilkan jumlah network yang lebih besar (lebih dari 65 ribu network) dengan kapasitas masing-masing subnet sebesar 256 host. Network kelas C juga dapat dibagi-bagi lagi menjadi beberapa subnet dengan menerapkan *subnet mask* yang lebih tinggi seperti untuk 25 bit (255.255.255.128), 26 bit (255.255.255.192), 27 bit (255.255.255.224) dan seterusnya.

Subnetting dilakukan pada saat konfigurasi *interface*. Penerapan *subnet mask* pada *IP Address* akan mendefinisikan 2 buah address baru, yakni *Network Address* dan *Broadcast Address*. *Network Address* didefinisikan dengan menset seluruh bit host berharga 0, sedangkan *Broadcat Address* dengan menset bit host berharga 1. Seperti yang telah dijelaskan pada bagian sebelumnya, *Network Address* adalah alamat network yang berguna pada informasi routing. Suatu *host* yang tidak perlu mengetahui address seluruh *host* yang ada pada network yang lain. Informasi yang dibutuhkannya hanyalah address dari network yang akan dihubungi serta *gateway* untuk mencapai network tersebut. Ilustrasi mengenai *Subnetting, Network Address* dan *Broadcast Address* dapat dilihat pada tabel 4.6 dibawah. Dari tabel tersebut dapat disimpulkan bagaimana nomor network standars dari suatu IP Address diubah menjadi nomor *subnet/subnet address* melalaui *subnetting.*

Subnetting hanya berlaku pada network lokal. Bagi network di luar network lokal maka nomor network yang dikenali tetap nomor network standard menurut kelas *IP Address*.

Tabel 4.6 :
Beberapa kombinasi IP Address, Network, Netmask dan Broadcast

IP Address	Network Address	Subnet Mask	Broadcast Address	Interpretasi
44.132.1.20	44.0.0.0	255.255.0.0 (16 bit)	44.132.255.255	Host 1.20 pada subnet 44.132.0.0
81.150.2.3	81.0.0.0	255.255.255.0 (24 bit)	81.50.2.255	Host 3 pada subnet 81.50.2.0
167.205.2.100	167.205.0.0	255.255.255.128 (25 bit)	167.205.2.127	Host 100 pada subnet 157.205.2.0
167.205.2.130	167.205.0.0	255.255.255.192 (26 bit)	167.205.2.191	Host 2 pada subnet 167.205.2.128