

**MAKALAH
PROTOKOL TCP/IP
PERANCANGAN JARINGAN KOMPUTER**

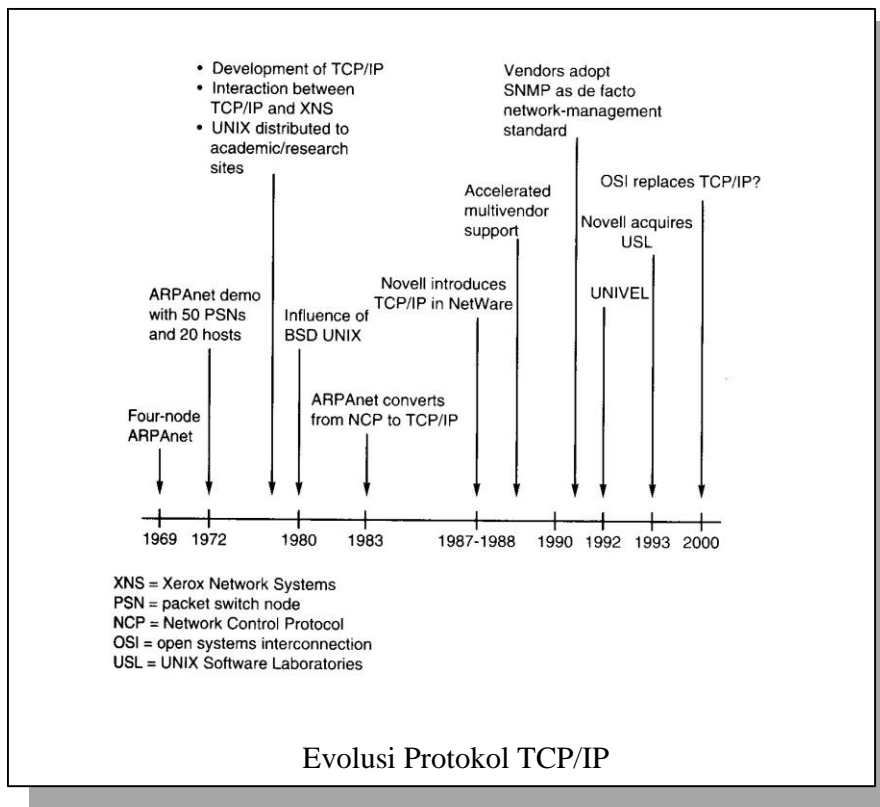


**Disusun Oleh :
Indri Setyaningsih
L200130132
Kelas A**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA**

PROTOKOL TCP/IP

Gambaran yang umum tentang TCP/IP adalah gabungan dari dua protokol komunikasi yang dipakai untuk komunikasi data, TCP kepanjangan dari *transmissioncontrol-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.



TCP/IP merupakan hasil riset dan pengembangan protokol pada percobaan jaringan *packet-switched* yang ditemukan oleh *defense-advanced-research-projectagency* (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 *internetactivities-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 tahun 1969 dan dilanjutkan dengan demo pada tahun 1972. Pada periode 1978 sampai 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 konglomerasi dari berbagai macam jaringan, tetapi protokol yang paling banyak dipakai adalah TCP/IP, walaupun ada beberapa jaringan menggunakan protokol berbeda, misalnya BITNET dan CREN yang menggunakan protokol IBM's SNA.

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.

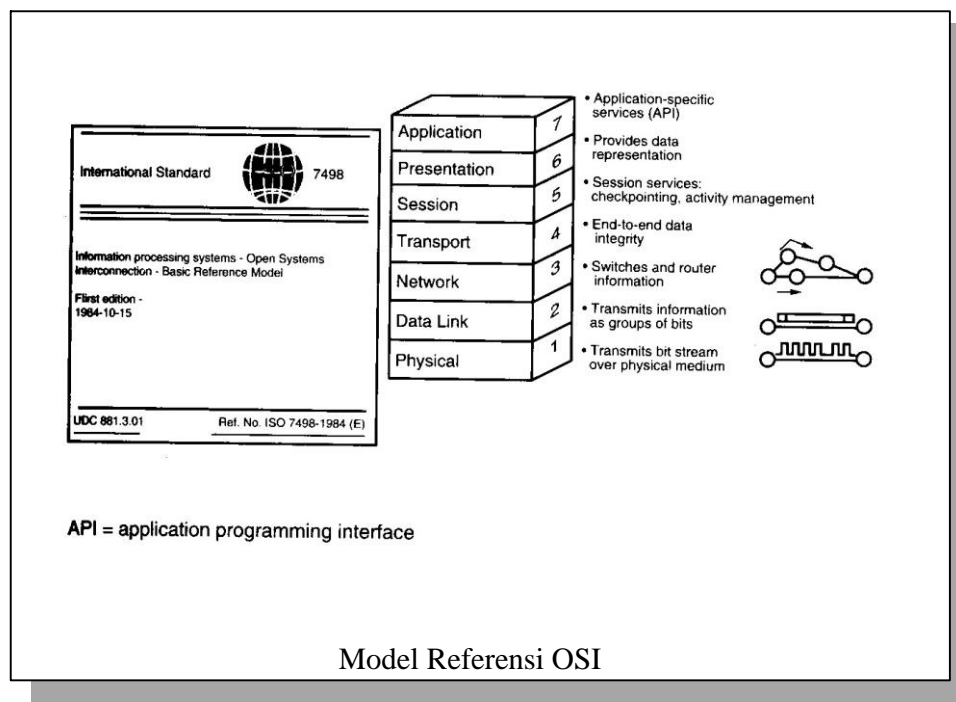
Applications menggunakan *network-protocol* yang dipakai sebagai dasar untuk berkomunikasi pada saat *network-application* berjalan pada jaringan, selanjutnya *network-protocol* menggunakan *physical-connections* untuk mengirim data.

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).

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.



Model OSI memiliki 7 layer yang bekerja dari layer teratas menuju ke bawah sesuai urutan : Application, 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.

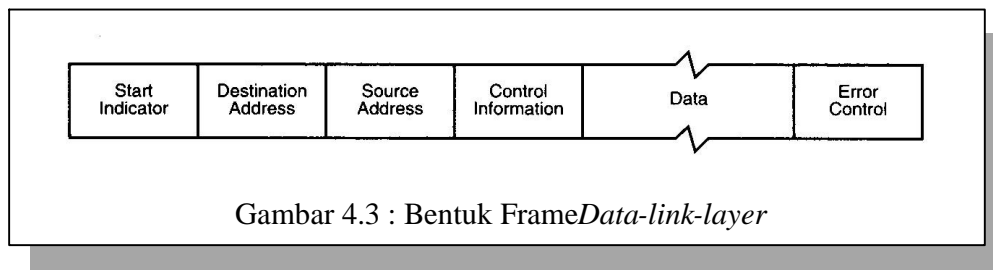
7. Physical Layer

Physical-layer dipergunakan untuk mengtransmisikan 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, elektrik dan prosedur antar muka dalam media fisik. Contoh: kabel UTP, Fiber, saluran transmisi, driver, sensor, pencatu, dll.

6. Data Link Layer

Data-link-layer dibentuk berdasarkan kemampuan transmisi dari *physicallayer*. Susunan bit yang akan dikirim atau diterima dikumpulkan dalam kelompok yang disebut *frame*. Dalam konteks LAN, frame dapat berarti *token-ring* atau *ethernetframe*.

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.



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. Contoh: Bluetooth, Wifi, Switch.

5. 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 *uniformaddressing-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 *Ethernet* 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). Contoh: Router.

4. 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 tidak 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 *timesharing* 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 *sequencedexchange-protocol* (SPX) dan *packet-exchange-protocol* (PXP), sedangkan pada TCP/IP menggunakan *transmission-control-protocol* (TCP).

3. 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

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*. Contoh: X Windows Server merupakan protocol yang menyediakan mekanisme client/server pada system operasi Unix, AppleTalk Session Protokol merupakan protocol yang menyediakan mekanisme client/server pada mesin-mesin Apple.

2. 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 metoda 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 menginterpretasikan data dapat dihilangkan. Contoh bahasa yang umum dipakai sesuai rekomendasi OSI adalah *abstract-syntax-representation, rev 1* (ASN 1). Contoh: Remote Desktop Connexion

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 :

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. Menjalankan perintah secara remote.

Beberapa contoh aplikasi tadi disebut dengan *application-programming interface* (API). Untuk menulis aplikasi yang berhubungan dengan jaringan dapat digunakan pustaka pemrograman API. Contoh aplikasi Netware pada *application-layer* adalah *netware-controlprotocol* (NCP) sedangkan untuk TCP/IP adalah FTP, SMTP dan TELNET.

PROTOKOL ENVELOPE

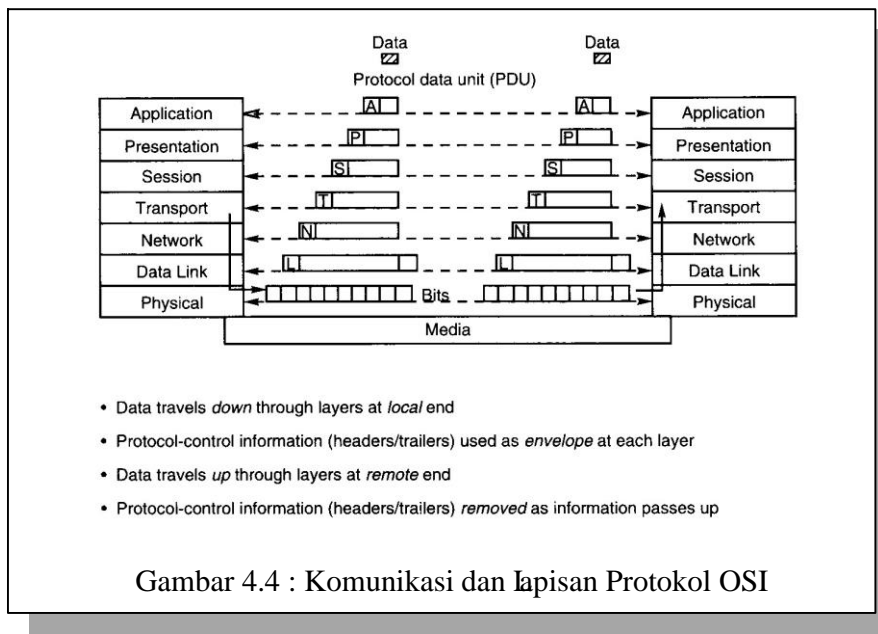
Model referensi OSI dapat dipakai untuk membantu mengetahui pertukaran data diantara dua sistem PC, yang pada terminologi OSI disebut dengan *end-system*. Diumpamakan ada dua *end-system* seperti ditunjukkan pada gambar 4.4. sebuah pesan (data) dikirimkan dari *end-system* yang satu ke *end-system* yang 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 mengidentifikasi *application-entities*, yaitu alamat *source* dan *destination* dari *application-layer*. PCI ditambah dengan pesan asli disebut dengan *application-protocol-data-unit* (APDU).

Selanjutnya APDU dikirim ke lapisan *presentation-layer*, maka *presentation-layer* menambahkan PCI ke data (APDU) yang diterima dari *application-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 berlangsung terus sampai turun kepada *physical-layer*.



Pada *data-link-layer* biasanya ditambahkan *trailer-field* yang ditambahkan pada data, yang berisi pengecekan kesalahan yang digunakan untuk melingkupi sebagai *frame* dari *data-link-layer* yang dibangkitkan oleh mekanisme hardware *datalink-layer* (*network-interface-card*). *Trailer-field* 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 *physicallayer* 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 *datalink-*

layer memproses *frame* yang diterima, selanjutnya memilah menjadi *data-link* PCI yang berisi header *data-link* dan CRC, selanjutnya mengirimkan NPDU (*networkPDU*) ke *network-layer* untuk diproses.

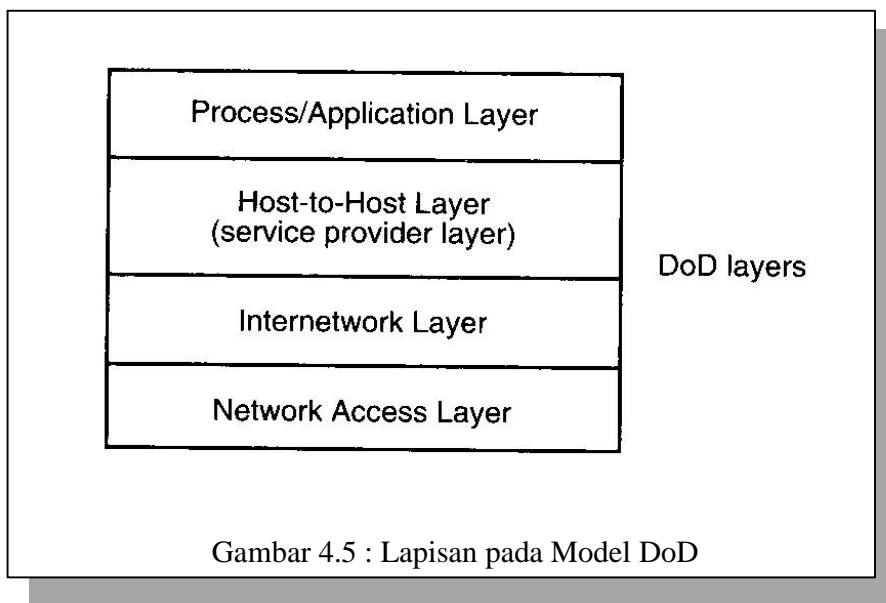
Setiap *layer end-system* pada receiver, PDU diproses berdasarkan informasi dalam header PCI untuk setiap lapisan tersebut. Sesudah PCI dihapus selanjutnya data (PDU) dikirimkan ke lapisan yang lebih atas, proses ini berlangsung terus hingga *aplication-layer* selesai melakukan proses hingga data yang 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-link-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 *aplicationlayer* dan *presentation-layer* tidak jelas, karena beberapa fungsi dari *presentationlayer* seperti *encoding* dan *decoding* data dapat dilakukan pada *aplication-layer* dengan memanggil bahasa program yang dapat melakukan fungsi *encoding* dan *decoding* data.

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.



Lapisan paling bawah adalah *network-access-layer* yang mewakili komponen penghubung dengan saluran fisik seperti kabel, transceiver, NIC, LAN *accessprotocol* (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-addressresolution-protocol* (RARP).

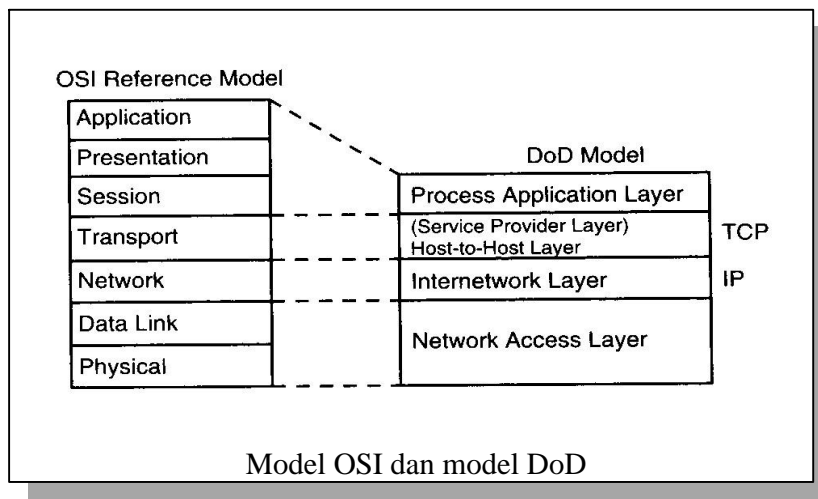
Segala masalah, *diagnostic-information*, dan keadaan yang berkenaan dengan protokol IP dilaporkan oleh protokol terpisah yang disebut *internet-control-messageprotocol* (ICMP) yang juga beroperasi pada *internet-layer*. *Internet-layer* juga memiliki perhatian kepada *routing* (pengaturan lintasan)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

Protokol *Host-to-host* mengatur hubungan antara dua *host* pada jaringan. Pada model DoD *host-to-host-protocol* diimplementasikan menjadi *transmission-controlprotocol* (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/Applicationlayer* 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.



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/Application-layer* pada model DoD berhubungan dengan tiga lapisan pada model OSI yaitu : *sessionlayer*, *presentation-layer* dan *application-layer*.