Bab 1. Arsitektur, Sejarah, Standarisasi dan Trend

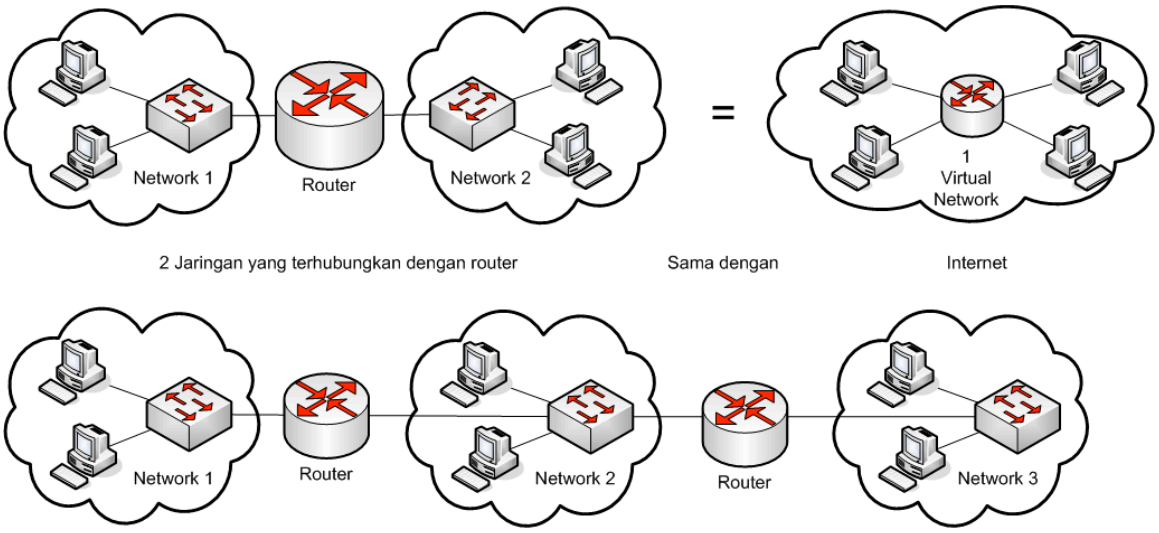
Zaman sekarang, Internet dan World Wide Web (WWW) sangat populer di seluruh dunia. Banyak masyarakat yang membutuhkan aplikasi yang berbasis Internet, seperti E-Mail dan akses Web melalui internet. Sehingga makin banyak aplikasi bisnis yang berkembang berjalan di atas internet. Transmission Control Protocol/Internet Protocol (TCP/IP) merupakan protokol yang melandasi internet dan jaringan dunia. Pada bab ini, akan dijelaskan tentang protokol TCP/IP, bagaimana internet terbentuk, dan bagaimana perkembangannya kedepan.

1.1. Model Arsitektur TCP/IP

Protokol TCP/IP terbentuk dari 2 komponen yaitu Transmission Control Protocol (TCP) dan Internet Protocol (IP).

1.1.1. Internetworking

Tujuan dari TCP/IP adalah untuk membangun suatu koneksi antar jaringan (*network*), dimana biasa disebut *internetwork,* atau *intenet*, yang menyediakan pelayanan komunikasi antar jaringan yang memiliki bentuk fisik yang beragam. Tujuan yang jelas adalah menghubungkan empunya (*hosts*) pada jaringan yang berbeda, atau mungkin terpisahkan secara geografis pada area yang luas.



Gambar 1.1 Contoh Internet – Dimana keduanya terlihat dalam sama sebagai 1 logikal jaringan

Internet dapat digolongkan menjadi beberapa group jaringan, antara lain:

- Backbone: Jaringan besar yang menghubungkan antar jaringan lainnya. Contoh : NSFNET yang merupakan jaringan backbone dunia di Amerika, EBONE yang merupakan jaringan backbone di Eropa, dan lainnya.
- Jaringan regional, contoh: jaringan antar kampus.
- Jaringan yang bersifat komersial dimana menyediakan koneksi menuju backbone kepada pelanggannya.

- Jaringan lokal, contoh: jaringan dalam sebuah kampus.

Aspek lain yang penting dari TCP/IP adalah membentuk suatu standarisasi dalam komunikasi. Tiap-tiap bentuk fisik suatu jaringan memiliki teknologi yang berbeda-beda, sehingga diperlukan pemrograman atau fungsi khusus untuk digunakan dalam komunikasi. TCP/IP memberikan fasilitas khusus yang bekerja diatas pemrograman atau fungsi khusus tersebut dari masing-masing fisik jaringan. Sehingga bentuk arsitektur dari fisik jaringan akan tersamarkan dari pengguna dan pembuat aplikasi jaringan. Dengan TCP/IP, pengguna tidak perlu lagi memikirkan bentuk fisik jaringan untuk melakukan sebuah komunikasi.

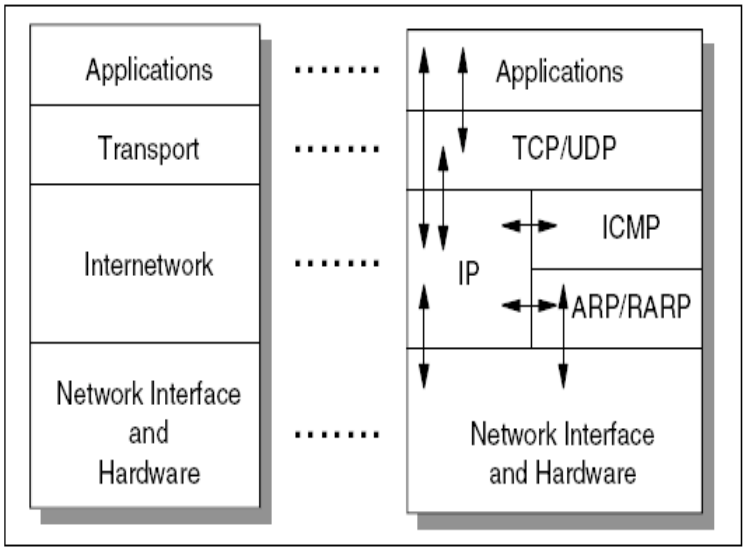
Sebagai contoh pada Gambar 1.1, untuk dapat berkomunikasi antar 2 jaringan, diperlukan komputer yang terhubung dalam suatu perangkat yang dapat meneruskan suatu paket data dari jaringan yang satu ke jaringan yang lain. Perangkat tersebut disebut ***Router***. Selain itu router juga digunakan sebagai pengarah jalur (*routing*).

Untuk dapat mengidentifikasikan host diperlukan sebuah alamat, disebut alamat IP (*IP address*). Apabila sebuah host memiliki beberapa perangkat jaringan (*interface*), seperti router, maka setiap interface harus memiliki sebuah IP address yang unik. IP address terdiri dari 2 bagian, yaitu :

```
IP address = <nomer jaringan><nomer host>
```

1.1.2. Lapisan (layer) pada Protokol TCP/IP

Seperti pada perangkat lunak, TCP/IP dibentuk dalam beberapa lapisan (*layer*). Dengan dibentuk dalam layer, akan mempermudah untuk pengembangan dan pengimplementasian. Antar layer dapat berkomunikasi ke atas maupun ke bawah dengan suatu penghubung interface. Tiap-tiap layer memiliki fungsi dan kegunaan yang berbeda dan saling mendukung layer diatasnya. Pada protokol TCP/IP dibagi menjadi 4 layer, tampak pada Gambar 1.2.



Gambar 1.2. Protokol TCP/IP

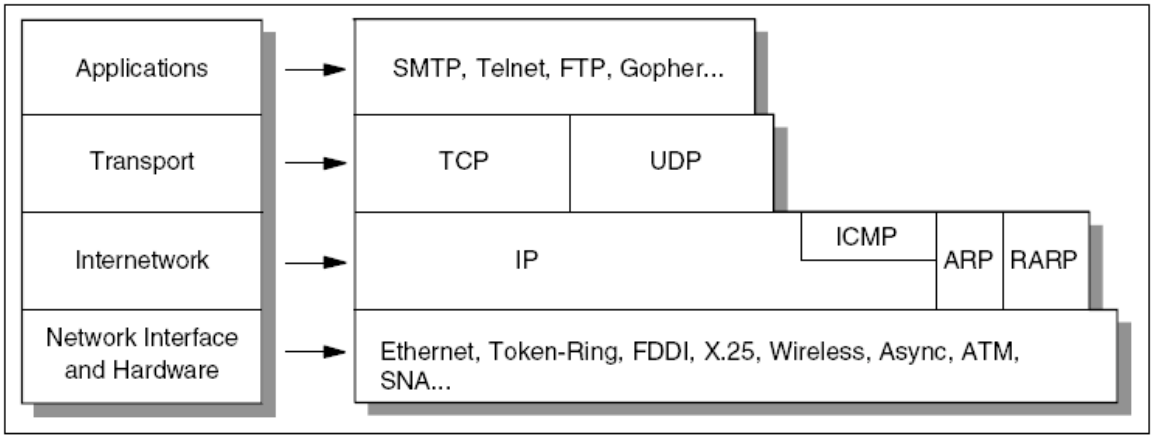
Layer Aplikasi (Aplications) Layer aplikasi digunakan pada program untuk berkomunikasi menggunakan TCP/IP. Contoh aplikasi antara lain Telnet dan File Transfer Protocol (FTP). Interface yang digunakan untuk saling berkomunikasi adalah nomer port dan socket.

Layer Transport Layer transport memberikan fungsi pengiriman data secara *end-to-end* ke sisi remote. Aplikasi yang beragam dapat melakukan komunikasi secara serentak (simulaneously). Protokol pada layer transport yang paling sering digunakan adalah Transmission Control Protocol (TCP), dimana memberikan fungsi pengiriman data secara *connection-oriented*, pencegahan duplikasi data, congestion control dan flow control. Protokol lainnya adalah User Datagram Protocol (UDP), dimana memberikan fungsi pengiriman *connectionless*, jalur yang tidak reliabel. UDP banyak digunakan pada aplikasi yang membutuhkan kecepatan tinggi dan dapat metoleransi terhadap kerusakan data.

Layer Internetwork Layer Internetwork biasa disebut juga layer internet atau layer network, dimana memberikan "vitual network" pada internet. Internet Protocol (IP) adalah protokol yang paling penting. IP memberikan fungsi routing pada jaringan dalam pengiriman data. Protokol lainnya antara lain : IP, ICMP, IGMP, ARP, RARP

Layer Network Interface Layer network interface disebut juga layer link atau layer datalink, yang merupakan perangkat keras pada jaringan. Contoh : IEEE802.2, X.25, ATM, FDDI, dan SNA.

Secara detail dapat digambarkan pada Gambar 1.3.



Gambar 1.3. Detail dari Model Arsitektur

1.1.3. Aplikasi TCP/IP

Level tertinggi pada layer TCP/IP adalah aplikasi. Dimana layer ini melakukan komunikasi sehingga dapat berinteraksi dengan pengguna.

Karakteristik dari protokol aplikasi antara lain:

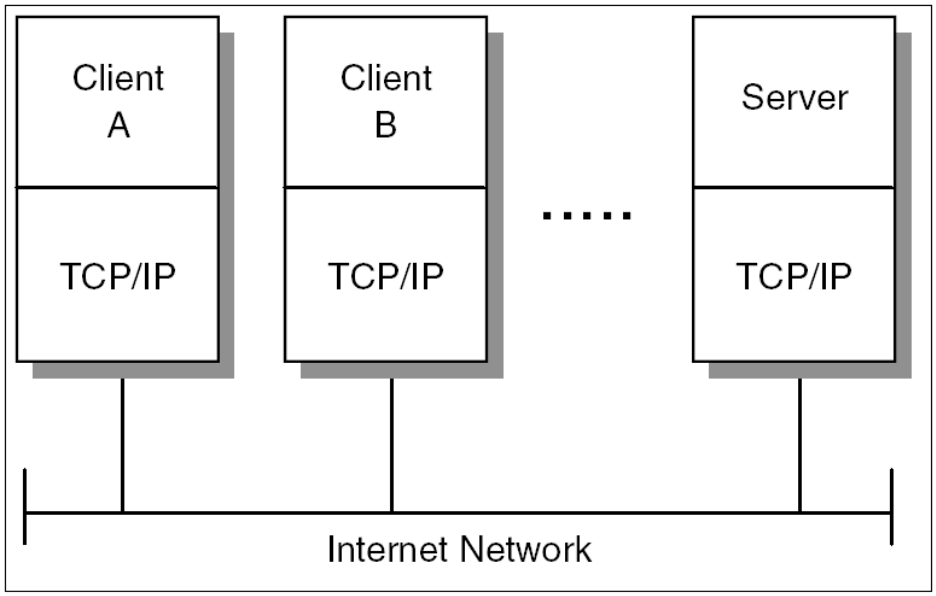
- Merupakan program aplikasi yang dibuat oleh pengguna, atau aplikasi yang merupakan standar dari produk TCP/IP. Contoh aplikasi yang merupakan produk dari TCP/IP antara lain :
  - TELNET, terminal interaktif untuk mengakses suatu remote pada internet.
  - FTP (File Transfer Protocol), transfer file berkecepatan tinggi antar disk.
  - SMTP (Simple Mail Transfer Protocol), sistem bersurat di internet
  - dll
- Menggunakan mekanisme TCP atau UDP.
- Menggunakan model interaksi client/server.

1.1.3.1. Model Client/Server

TCP adalah *peer-to-peer*, protokol yang bersifat *connection-oriented*. Tidak ada hubungan tuan dan budak (master/slave), tetapi banyak aplikasi yang bersifat client/server.

SERVER adalah aplikasi yang memberikan pelayanan kepada user internet. CLIENT adalah yang meminta pelayanan. Aplikasi bisa memiliki bagian server dan bagian client, dimana dapat berjalan secara bersamaan dalam 1 sistem.

Server merupakan progam yang dapat menerima permintaan (*request*), melakukan pelayanan yang diminta, kemudian mengembalikan sebagai *reply*. Server dapat melayani multi request bersamaan.



Gambar 1.4. Model Client-Server

Server bekerja dengan cara menunggu request pada port yang sudah terdaftar, sehingga client dapat dengan mudah mengirimkan data ke port pada server.

1.1.4. Bridge, Router dan Gateway

Ada beberapa cara untuk memberikan koneksi ke jaringan. Pada internetworking dapat dilakukan dengan router. Pada bagian ini akan dibedakan antara bridge, router dan gateway dalam mengakses jaringan.

Bridge Menghubungkan jaringan pada layer network interface dan meneruskan frame. Bridge juga berfungsi sebagai MAC relay.
Bridge juga transparant terhadap IP, artinya apabila suatu host mengirim IP datagram ke host yang lain, IP tidak akan di awasi oleh bridge dan langsung cross ke host yang dituju.

Router Menghubungkan jaringa pada layer internetwork dan mengarahkan jalur paket data.
Router mampu memilih jalur yang terbaik untuk pengiriman data, karena memiliki routing.
Dikarenakan router tidak transparant terhadap IP, maka router akan meneruskan paket berdasarkan alamat IP dari data.

Gateway Menghubungkan jaringan pada layer diatas router dan bridge. Gateway mendukung pemetaan alamat dari jaringan yang satu ke jaringan yang lain.
Gateway merupakan pintu keluar suatu host menuju ke jaringan diluar.

1.2. Sejarah Internet

Jaringan mulai dibangun pada kisaran tahun 60an dan 70an, dimana mulai banyak penelitian tentang paket-switching, collision-detection pada jaringan lokal, hirarki jaringan dan teknik komunikasi lainnya.

Semakin banyak yang mengembangkan jaringan, tapi hal ini mengakibatkan semakin banyak perbedaan dan membuat jaringan harus berdiri sendiri tidak bisa dihubungkan antar tipe jaringan yang berbeda. Sehingga untuk menggabungkan jaringan dari group yang berbeda tidak bisa terjadi. Terjadi banyak perbedaan dari interface, aplikasi dan protokol.

Situasi perbedaan ini mulai di teliti pada tahun 70an oleh group peneliti Amerika dari Defence Advanced Research Project Agency (DARPA). Mereka meneliti tentang internetworking, selain itu ada organisasi lain yang juga bergabung seperti ITU-T (dengan nama CCITT) dan ISO. Tujuan dari penelitian tersebut membuat suatu protokol, sehingga aplikasi yang berbeda dapat berjalan walaupun pada sistem yang berbeda.

Group resmi yang meneliti disebut ARPANET network research group, dimana telah melakukan meeting pada oktober 1971. Kemudian DARPA melanjukan penelitiannya tentang host-to-host protocol dengan menggunakan TCP/IP, sekitar tahun 1978. Implementasi awal internet pada tahun 1980, dimana ARPANET menggunakan TCP/IP. Pada tahun 1983, DARPA memutuskan agar semua komputer terkoneksi ke ARPANET menggunakan TCP/IP.

DARPA mengontak Bolt, Beranek, and Newman (BBN) untuk membangun TCP/IP untuk Berkeley UNIX di University of California di Berkeley, untuk mendistribusikan kode sumber bersama dengan sistem operasi Berkeley Software Development (BSD), pada tahun 1983 (4.2BSD). Mulai saat itu, TCP/IP menjadi terkenal di seluruh universitas dan badan penelitian dan menjadi protokol standar untuk komunikasi.

1.2.1. ARPANET

Suatu badan penelitian yang dibentuk oleh DARPA, dan merupakan “grand-daddy of packet switching”. ARPANET merupakan awal dari internet. ARPANET menggunakan komunikasi

56Kbps tetapi karena perkembangan akhirnya tidak mampu mengatasi trafik jaringan yang berkembang tersebut.

1.2.2. NFSNET

NSFNET, National Science Foundation (NSF) Network. Terdiri dari 3 bagian internetworking di Amerika, yaitu :

- Backbone, jaringan yang terbentuk dari jaringan tingkat menengah (mid-level) dan jaringan supercomputer.
- Jaringan tingkat menengah (mid-level) terdiri dari regional, berbasis disiplin dan jaringan konsorsium superkomputer.
- Jaringan kampus, akademik maupun komersial yang terhubung ke jaringan tingkat menengah.

1.2.3. Penggunaan Internet secara komersial

Penggunaan internet berawal dari Acceptable Use Policy (AUP) tahun 1992, dimana menyebutkan internet dapat digunakan untuk komersial. Internet Service Provider mulai membangun bisnis diantaranya PSINet dan UUNET, kemudian menyusul CERFNet dan membentuk Commercial Internet Exchange (CIX). Keberadaan internet makin berkembang dan semakin banyak public exchange point (IXP), dapat dilihat di : http://www.ep.net.

1.2.4. Internet2

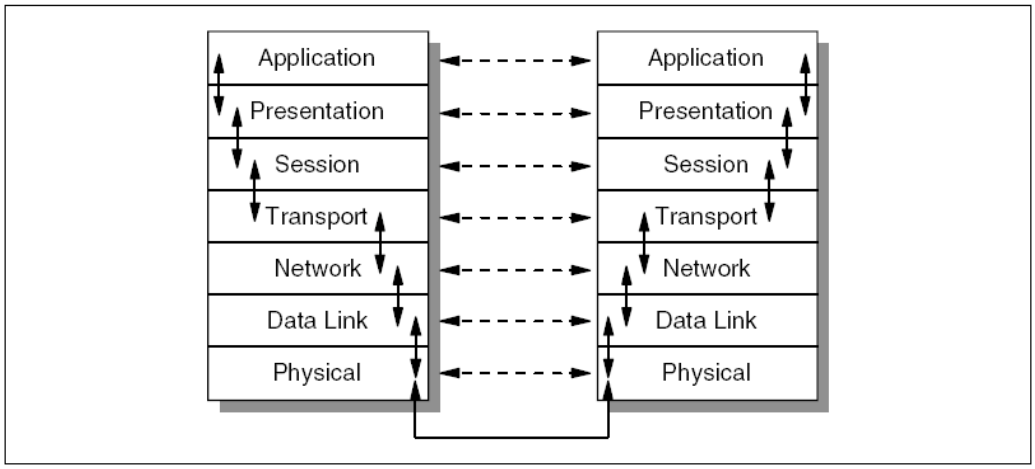
Perkembangan internet disusul dengan project **internet2** yang merupakan *Next Generation Internet* (NGI). Tujuan dari internet2 antara lain :

- Mendemostrasikan aplikasi baru yang dapat meningkatkan peneliti untuk melakukan kolaborasi dalam penelitian
- Membangun advanced communication infrastructures
- Menyediakan middleware dan perangkat development
- Mendukung QoS untuk penelitian dan komuniti pendidikan
- Mempromosikan next generation dari teknologi komunikasi
- Mengkoordinasi standarisasi
- Mengkapitalisasi sistem partner antara pemerintah dan sektor organisasi
- Melakukan perubahan jaringan dari internet ke internet2
- Mempelajari efek samping dari infrastruktur yang baru pada pendidikan tinggi dan komunitas internet

Informasi tentang internet2 dapat dilihat di http://www.internet2.edu

1.2.5. Model Referensi dari Open System Interconnection (OSI)

OSI (Open System Interconnection) model (ISO 7498) mendifinisikan 7 layer model dari komunikasi data.



Gambar 1.5. Model Referensi OSI

Tiap layer memiliki fungsi yang saling terhubung dengan layer di atasnya.

1.3. Standarisasi TCP/IP

TCP/IP semakin popular diantara developer dan pengguna, karena itu perlu adanya standarisasi. Standarisasi di kelola oleh Internet Architecture Board (IAB)

IAB mengacu pada Internet Engineering Task Force (IETF) untuk membuat standar baru. Dimana standarisasi menggunakan RFC. Untuk Internet Standar Process, menggunakan RFC 2026 – The Internet Standard Process – Revision 3, dimana didalamnya berisi tentang protokol, prosedur, dan konvensi yang digunakan dari oleh internet.

1.3.1. Request For Comment (RFC)

Internet Protocol suite masih dikembangkan dan perkembangannya menggunakan mekanisme *Request For Comment* (RFC). Protokol baru yang dikembangkan oleh peneliti akan diajukan dalam bentuk Internet Draft (ID). Kemudian akan di evaluasi oleh IAB. Apabila disetujui maka akan lahir RFC dengan seri baru untuk aplikasi atau protokol tersebut, sehingga developer dapat menggunakan standar tersebut.

1.3.2. Internet Standard

Proposal standar, draft standar, dan protokol standar merupakan bagian dari *Internet Standard Track*. Setelah proposal diakui maka proposal tersebut akan memiliki nomer, yang disebut standard number (STD). Contoh : Domain Name Systems (DNS) menggunakan STD 13 dan dijelaskan pada RFC 1034 dan 1035, sehingga dapat dituliskan "STD-13/RFC1034/RFC1035". Untuk info lengkapnya dapat diakses di http://www.ietf.org

1.4. Internet Masa Depan

Mencoba untuk memperkirakan penggunaan internet dimasa mendatang adalah tidak mudah. Karena itu pada bagian ini akan diberikan contoh kecil penggunaan internet untuk masa depan.

1.4.1. Aplikasi Multimedia

Penggunaan bandwidth semakin lama akan semakin efisien, banyak teknologi yang dapat digunakan untuk mengatur penggunaan bandwidth salah satunya Dense Wave Division Multiplexing (DWDM).

Penggunaan bandwidth banyak digunakan pada aplikasi multimedia, antara lain Voice over Internet Protocol (VoIP) dan masih banyak lagi lainnya, bahkan untuk video conference.

Sekarang untuk mendengarkan lagu dengan internet sudah dapat kita rasakan, dan dikedepannya akan dimungkinkan semua perangkat terkoneksi melalui internet dan masih banyak lagi lainnya. Atau mungkin anda sendiri akan diberi IP Address... ???

1.4.2. Penggunaan untuk komersial

Penggunaan teknologi Virtual Private Networking (VPN) semakin banyak digunakan oleh perusahaan. VPN digunakan untuk mengamankan komunikasi yang digunakan oleh sebuah perusahaan. Misal untuk Virtual meeting.

1.4.3. Wireless Internet

Penggunaan aplikasi tanpa kabel sangat meningkatkan mobilitas seseorang, sehingga kebutuhan internet wireless akan semakin populer. Dengan adanya teknologi bluetooth, Wifi IEEE802.11, Wi-MAX dan yang lainnya akan mendukung internet tanpa kabel.

Bab 2. Model Referensi OSI

OSI adalah referensi komunikasi dari Open System Interconnection. OSI model digunakan sebagai titik referensi untuk membahas spesifikasi protokol.

2.1. Layer pada OSI

OSI model terdiri dari 7 layer. Dimana bagian atas dari layernya (layer 7,6,dan 5) difokuskan untuk bentuk pelayanan dari suatu aplikasi. Sedangkan untuk layer bagian bawahnya (layer 4, 3, 2 dan 1) berorientasikan tentang aliran data dari ujung satu ke ujung yang lainnya.

Tabel 2.1. Model Referensi OSI

Nama layer	Fungsi	Contoh
Aplikasi (layer 7)	Aplikasi yang saling berkomunikasi antar komputer. Aplikasi layer mengacu pada pelayanan komunikasi pada suatu aplikasi.	Telnet, HTTP, FTP, WWW Browser, NFS, SMTP, SNMP
Presentasi (Layer 6)	Pada layer bertujuan untuk mendefinisikan format data, seperti ASCII text, binary dan JPEG.	JPEG, ASCII, TIFF, GIF, MPEG, MIDI
Sesi (Layer 5)	Sesi layer mendefinisikan bagaimana memulai, mengontrol dan mengakhiri suatu percakapan (biasa disebut session)	RPC, SQL, NFS, SCP
Transport (Layer 4)	Pada layer 4 ini bisa dipilih apakah menggunakan protokol yang mendukung error-recovery atau tidak. Melakukan multiplexing terhadap data yang datang, mengurutkan data yang datang apabila datangnya tidak berurutan.	TCP, UDP, SPX
Network (Layer 3)	Layer ini mendefinisikan pengiriman data dari ujung ke ujung. Untuk melakukan pengiriman pada layer ini juga melakukan pengalamatan. Mendifinisikan pengiriman jalur (routing).	IP, IPX, Appletalk DDP
Data Link (layer 2)	Layer ini mengatur pengiriman data dari interface yang berbeda. Semisal pengiriman data dari ethernet 802.3 menuju ke High-level Data Link Control (HDLC), pengiriman data WAN.	IEEE 802.2/802.3, HDLC, Frame relay, PPP, FDDI, ATM
Physical (Layer 1)	Layer ini mengatur tentang bentuk interface yang berbeda-beda dari sebuah media transmisi. Spesifikasi yang berbeda misal konektor, pin, penggunaan pin, arus listrik yang lewat, encoding, sumber cahaya dll	EIA/TIA-232, V35, EIA/TIA- 449, V.24, RJ45, Ethernet, NRZI, NRZ, B8ZS

2.2. Konsep dan Kegunaan Layer

Banyak kegunaan yang didapat dari pembagian fungsi menjadi yang lebih kecil atau yang disebut layer. Kegunaan yang pasti adalah mengurangi kompleksitas, sehingga dapat didefinisikan lebih detil.

Contoh kegunaannya antara lain:

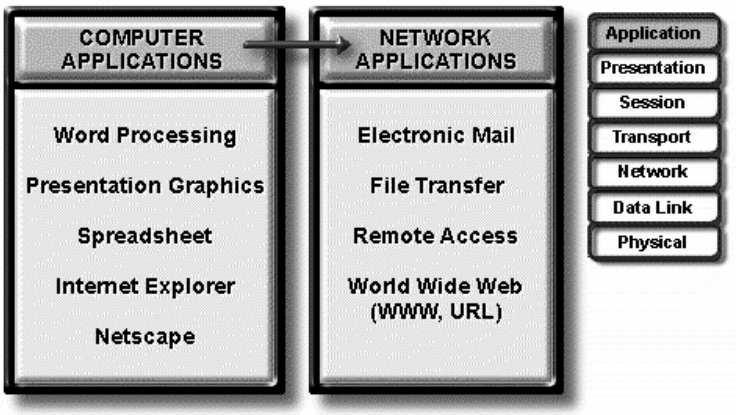
- Manusia dapat membahas dan mempelajari tentang protokol secara detil

- Membuat perangkat menjadi bentuk modular, sehingga pengguna dapat menggunakan hanya modul yang dibutuhkan
- Membuat lingkungan yang dapat saling terkoneksi
- Mengurangi kompleksitas pada pemrograman sehingga memudahkan produksi
- Tiap layer dapat diberikan pembuka dan penutup sesuai dengan layernya
- Untuk berkomunikasi dapat dengan segera menggunakan layer dibawahnya.

2.2.1. Layer Aplikasi

Pada layer ini berurusan dengan program komputer yang digunakan oleh user. Program komputer yang berhubungan hanya program yang melakukan akses jaringan, tetapi bila yang tidak berarti tidak berhubungan dengan OSI.
Contoh: Aplikasi word processing, aplikasi ini digunakan untuk pengolahan text sehingga program ini tidak berhubungan dengan OSI. Tetapi bila program tersebut ditambahkan fungsi jaringan misal pengiriman email, maka aplikasi layer baru berhubungan disini.

Sehingga bila digambar dapat digambar seperti Gambar 2.1.

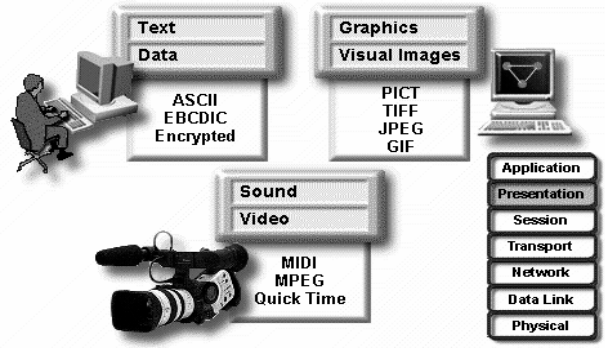


Gambar 2.1 Layer Aplikasi

2.2.2. Layer Presentasi

Pada layer ini bertugan untuk mengurusi format data yang dapat dipahami oleh berbagai macam media. Selain itu layer ini juga dapat mengkonversi format data, sehingga layer berikutnya dapat memafami format yang diperlukan untuk komunikasi.

Contoh format data yang didukung oleh layer presentasi antara lain : Text, Data, Graphic, Visual Image, Sound, Video. Bisa digambarkan seperti pada Gambar 2.2.

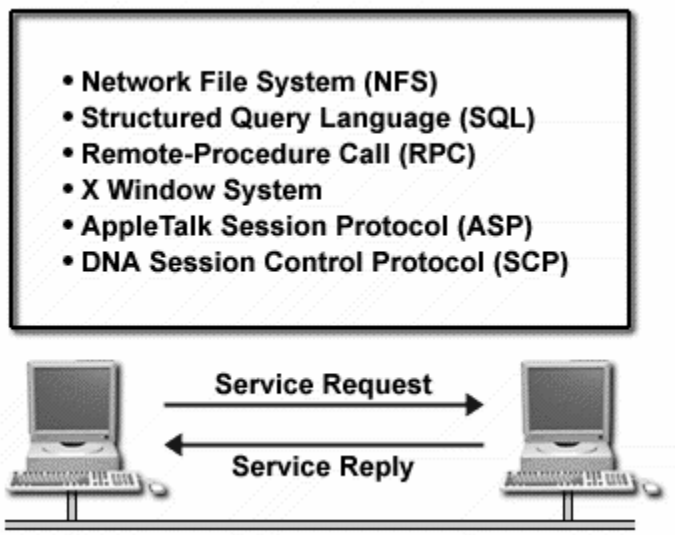


Gambar 2.2 Format data pada layer presentasi

Selain itu pada layer presentasi ini juga berfungsi sebagai enkripsi data.

2.2.3. Layer Sesi (Session)

Sesi layer mendefinisikan bagaimana memulai, mengontrol dan mengakhiri suatu percakapan (biasa disebut session). Contoh layer session : NFS, SQL, RPC, ASP, SCP

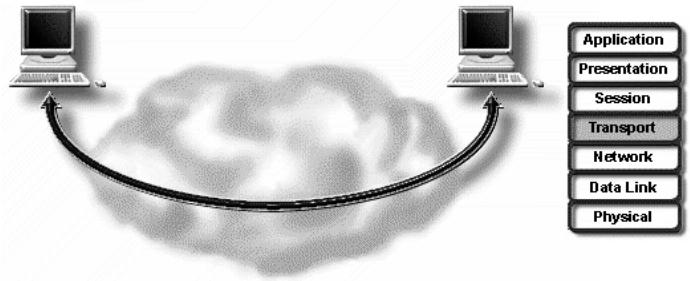


Gambar 2.3 Mengkoordinasi berbagai aplikasi pada saat berinteraksi antar komputer

2.2.4. Layer Transport

Pada layer 4 ini bisa dipilih apakah menggunakan protokol yang mendukung error-recovery atau tidak. Melakukan multiplexing terhadap data yang datang, mengurutkan data yang datang apabila datangnya tidak berurutan.

Pada layer ini juga komunikasi dari ujung ke ujung (end-to-end) diatur dengan beberapa cara, sehingga urusan data banyak dipengaruhi oleh layer 4 ini.



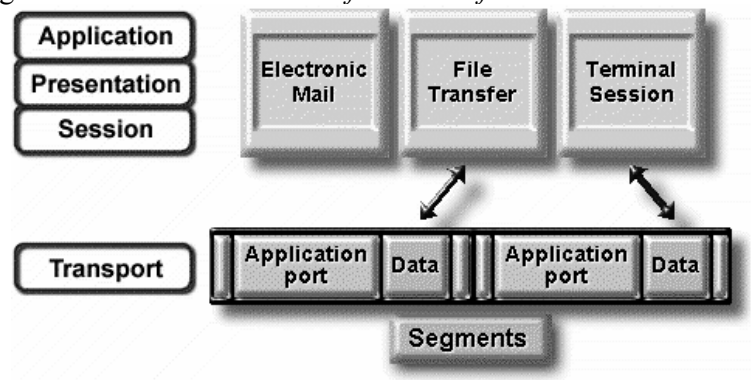
Gambar 2.4 Fungsi transport layer

Fungsi yang diberikan oleh layer transport :

- Melakukan segmentasi pada layer atasnya
- Melakukan koneksi end-to-end
- Mengirimkan segmen dari 1 host ke host yang lainnya
- Memastikan reliabilitas data

2.2.4.1. Melakukan segmentasi pada layer atasnya

Dengan menggunakan OSI model, berbagai macam jenis aplikasi yang berbeda dapat dikirimkan pada jenis transport yang sama. Transport yang terkirim berupa segmen per segmen. Sehingga data dikirim berdasarkan *first-come first served*.



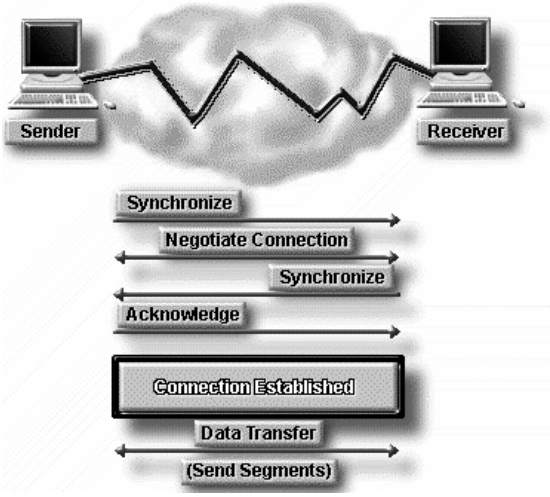
Gambar 2.5 Segmentasi pada layer transport

2.2.4.2. Melakukan koneksi end-to-end

Konsepnya, sebuah perangkat untuk melakukan komunikasi dengan perangkat lainnya, perangkat yang dituju harus menerima koneksi terlebih dahulu sebelum mengirimkan atau menerima data.

Proses yang dilakukan sebelum pengiriman data, seperti pada Gambar 2.6:

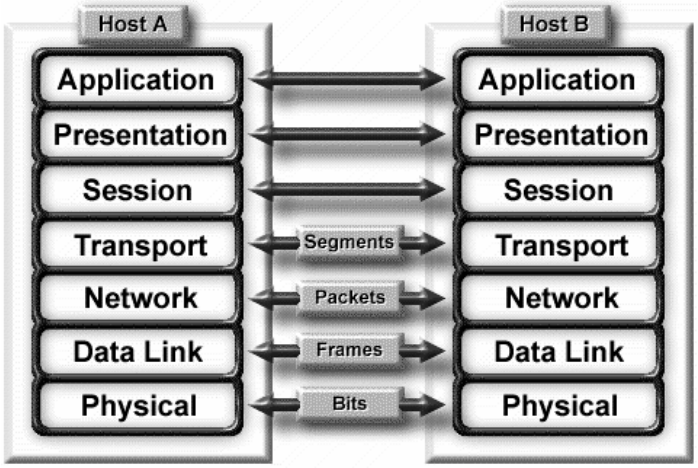
- Pengirim (*sender*) mengirimkan sinyal Synchronize terlebih dulu ke tujuan
- Penerima (*receiver*) mengirimkan balasan dengan sinyal Negotiate Connection
- Penerima mengirimkan Synchronize ulang, apa benar pengirim akan mengirimkan data
- Pengirim membalas dengan sinyal Acknowledge dimana artinya sudah siap untuk mengirimkan data
- ***Connection establish***
- Kemudian segmen dikirim



Gambar 2.6 Proses pembentukan koneksi

2.2.4.3. Mengirimkan segmen dari 1 host ke host yang lainnya

Proses pengiriman yang terjadi pada layer transport berupa segmen, sedangkan pada layer bawahnya berupa paket dan pada layer 2 berupa frame dan dirubah menjadi pengiriman bit pada layer 1. Hal tersebut dapat dilihat pada Gambar 2.7

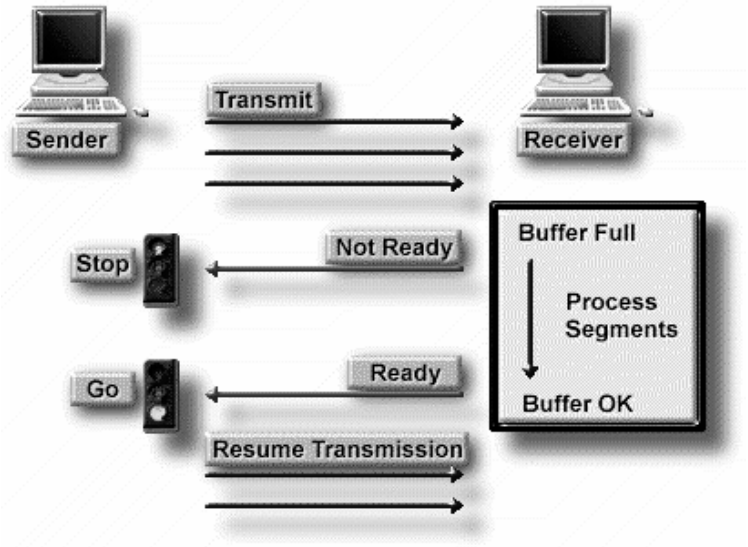


Gambar 2.7 Pengiriman segmen, paket, frame, dan bit

2.2.4.4. Memastikan reliabilitas data

Pada waktu pengiriman data sedang berjalan, kepadatan jalur bisa terjadi (*congestion*). Alasan terjadinya congestion antara lain: komputer berkecepatan tinggi mengirimkan data lebih cepat dari pada jaringannya, apabila beberapa komputer mengirimkan data ke tujuan yang sama secara simultan.

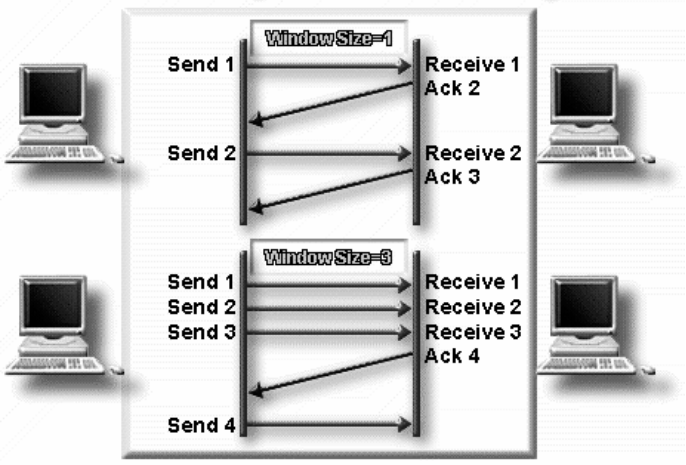
Untuk mengatasi hal tersebut setiap perangkat dilengkapi dengan yang namanya kontrol aliran (*flow control*). Dimana apabila ada pengirim yang mengirimkan data terlalu banyak, maka dari pihak penerima akan mengirmkan pesan ke pengirim bahwa jangan mengirim data lagi, karena data yang sebelumnya sedang di proses. Dan apabila telah selesai diproses, si penerima akan mengirimkan pesan ke pengirim untuk melanjutkan pengiriman data. Ilustrasi flow control dapat dilihat pada Gambar 2.8.



Gambar 2.8 Flow Control

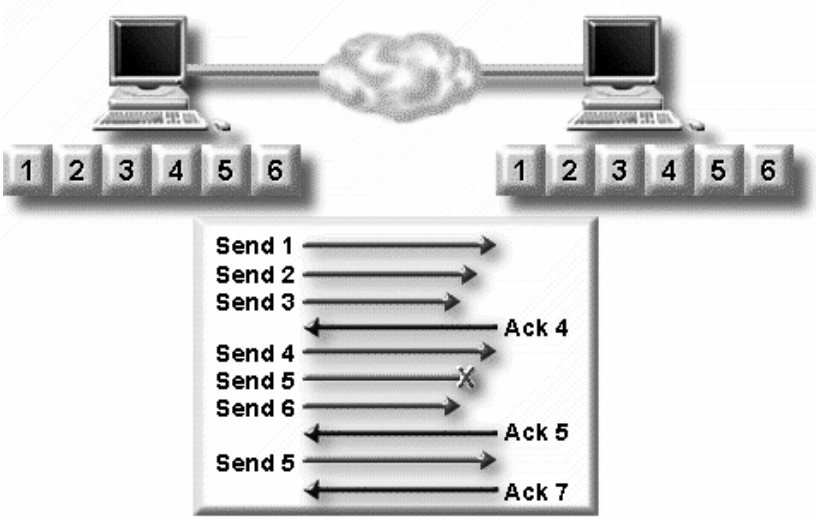
Dinamakan data yang reliabel artinya paket data datang sesuai dengan urutan pada saat dikirimkan. Protokol akan gagal apabila terjadi paket yang hilang, rusak, terjadi duplikasi, atau menerima paket data dengan urutan yang berbeda. Untuk memastikan data yang terkirim, si penerima harus mengirimkan acknowledge untuk setiap data yang diterima pada segmen.

Contoh: Pengirim mengirimkan data dengan format window segmen sebesar 1, maka penerima akan mengirimkan acknowledge no 2. Apabila pengirm mengirimkan data dengan format window segmen sebesar 3, maka penerima akan mengirimkan acknowledge no 4 apabila penerimaan data benar. Ilustrasi dapat dilihar di Gambar 2.9.



Gambar 2.9 Sistem windowing

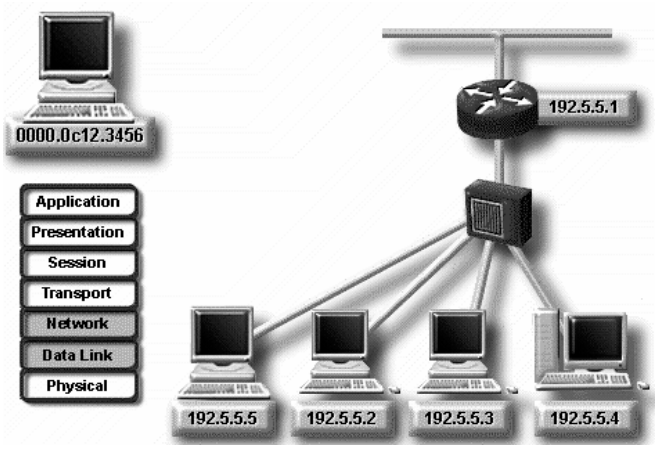
Teknik konfirmasi data dengan acknowledge bekerja mengirimkan informasi data mana yang terjadi kesalahan. Contoh pada Gambar 2.10 apabila data nomer 5 yang rusak maka si penerima akan memberikan acknowledge ke pengirim no 5, dan si pengirim akan mengirmkan ulang data segmen no 5.



Gambar 2.10 Acknowledge

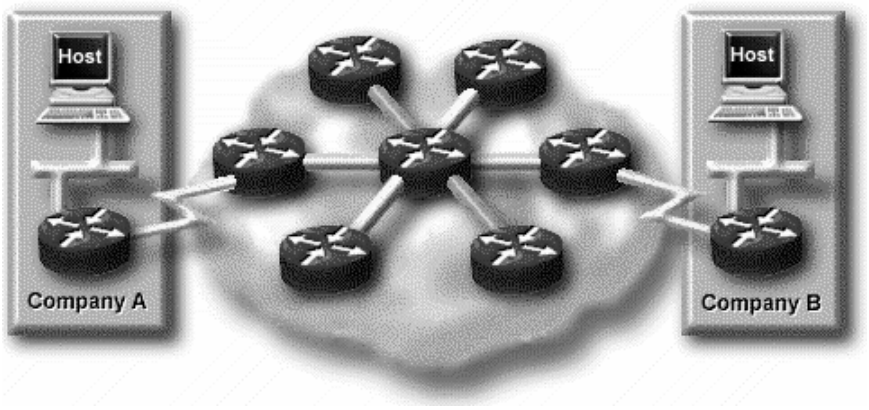
2.2.5. Layer Network

Fungsi utama dari layer network adalah pengalamatan dan routing. Pengalamatan pada layer network merupakan pengalamatan secara logical, Contoh penggunaan alamat IP seperti pada Gambar 2.11.



Gambar 2.11 Pengalamat logic dan fisik

Routing digunakan untuk pengarah jalur paket data yang akan dikirim. Dimana routing ada 2 macam yaitu Routed dan Routing Protocol.



Gambar 2.12 Untuk menuju ke tujuan lain menggunakan Routing

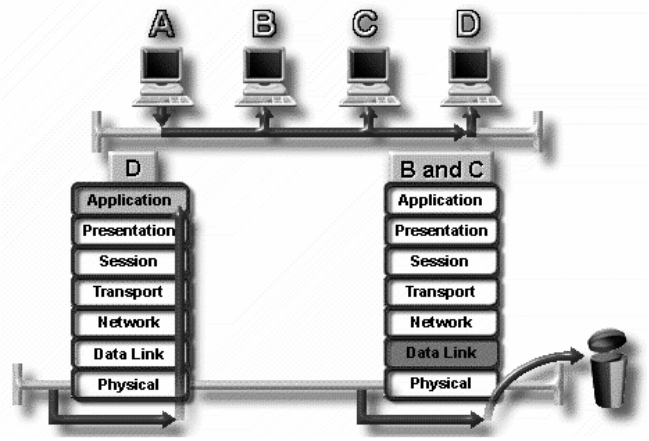
2.2.6. Layer Data Link

Fungsi yang diberikan pada layer data link antara lain :

- *Arbitration*, pemilihan media fisik
- *Addressing*, pengalamatan fisik
- *Error detection*, menentukan apakah data telah berhasil terkirim
- *Identify Data Encapsulation*, menentukan pola header pada suatu data

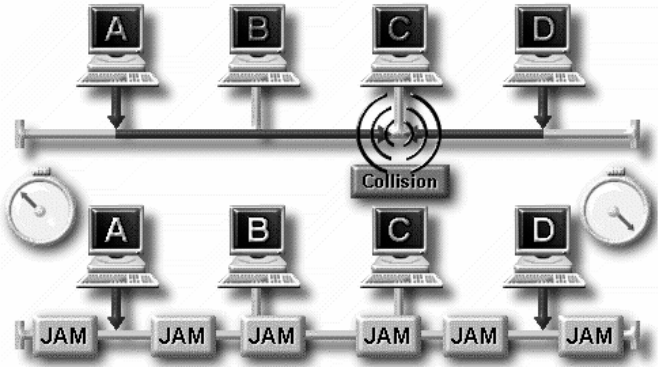
2.2.6.1. Arbitrasi

Penentuan waktu pengiriman data yang tepat apabila suatu media sudah terpakai, hal ini perlu melakukan suatu deteksi sinyal pembawa. Pada Ethernet menggunakan metode *Carrier Sense Multiple Access / Collision Detection* (CSMA/CD).



Gambar 2.13 CSMA/CD

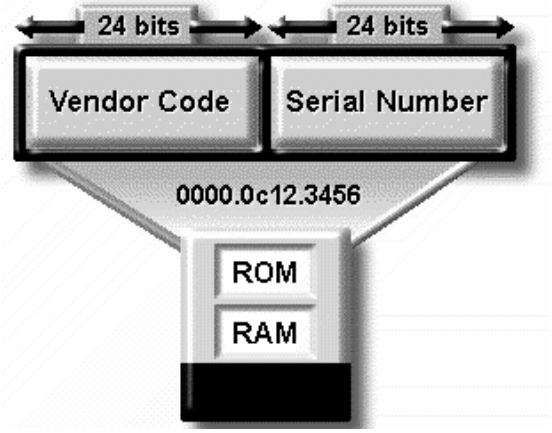
Pada jaringan yang dapat melakukan akses secara bersamaan simultan. Maka bila Host A mengirimkan data ke Host D, maka Host B dan C akan melakukan deteksi jalur, dan apabila jalur sedang dipakai maka Host B dan C akan menunggu terlebih dahulu. Hal ini dapat mencegah terjadinya *collision*. Ilustrasi seperti pada Gambar 2.14.



Gambar 2.14 Collision

2.2.6.2. Addressing

Pengalamatan yang dilakukan pada layer data link bersifat fisik, yaitu menggunakan Media Access Control (*MAC*). MAC ditanamkan pada interface suatu perangkat jaringan.MAC berukuran 48bit dengan format 12 heksadesimal.



Gambar 2.15 Media Access Control (MAC)

2.2.6.3. Error Detection

Teknik yang digunakan adalah *Frame Check Sequence* (FCS) dan *Cyclic Redundancy Check* (CRC).

2.2.6.4. Identify Data Encapsulation

Mengidentifikasikan format data yang lewat apakah termasuk ehternet, token ring, frame-relay dan sebagainya.

Tabel 2.2 Tipe Protokol Encoding

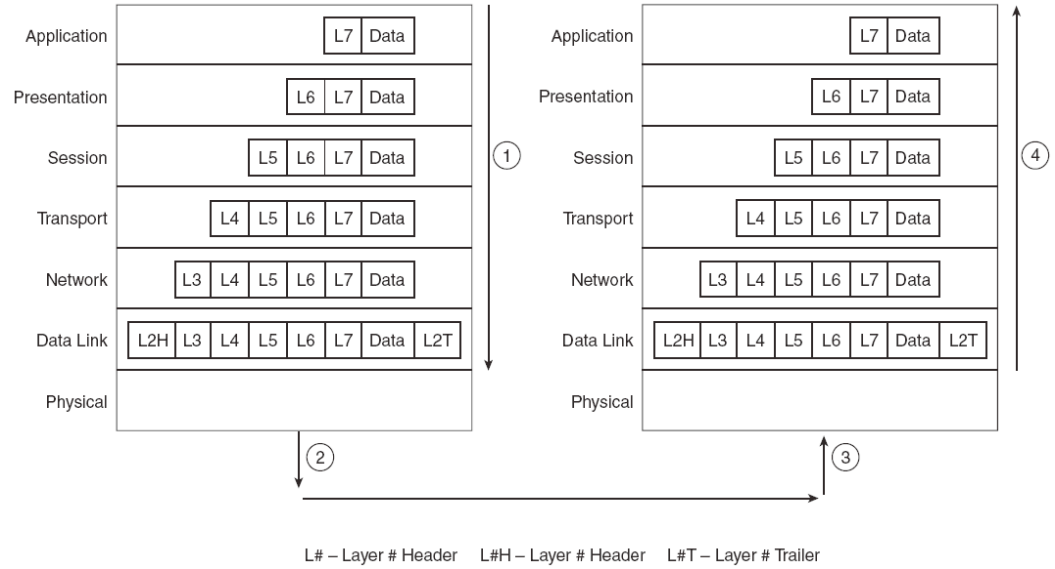
Protokol Data Link	Bagian (*Field*)	Header	Ukuran
802.3 Ethernet 802.5 Token Ring	DSAP	Header 802.2	1 byte
802.3 Ethernet 802.5 Token Ring	SSAP	Header 802.2	1 byte
802.3 Ethernet 802.5 Token Ring	Protocol Type	Header SNAP	2 byte
Ethernet (DIX)	Ethertype	Header Ethernet	2 byte
HDLC	Cisco proprietary	Extra Cisco Header	2 byte
Frame Relay RFC 2427	NLPID	RFC1490	1 byte
Frame Relay RFC 2427	L2 / L3 protocol ID	Q.933	2 byte / ID
Frame Relay RFC 2427	SNAP Protocol Type	Header SNAP	2 bye

2.3. Interaksi antar Layer pada OSI

Proses bagaimana komputer berinteraksi dengan menggunakan layer pada OSI, mempunyai dua fungsi umum, antara lain :

- Tiap layer memberikan pelayanan pada layer di atasnya sesuai dengan spesifikasi protokolnya
- Tiap layer mengirimkan informasi komunikasi melalui software dan hardware yang sama antar komputer.

Komunikasi antar komputer pada OSI layer dapat digambarkan seperti Gambar 2.16.



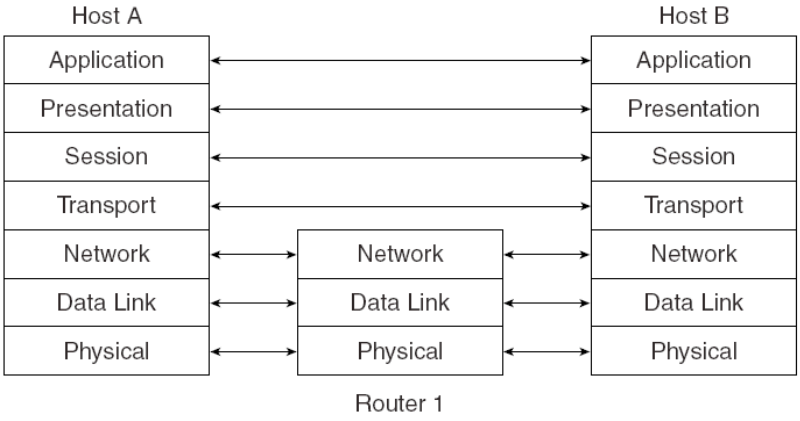
Gambar 2.16 Komunikasi antar Komputer pada OSI Layer

Sebuah data dibuat oleh aplikasi pada host A, contoh seseorang menuliskan email. Pada tiap layer ditambahkan header dan dilanjutkan ke layer berikutnya (langkah 1 Gambar 2.16). Contoh : pada layer transport menyalurkan data dan header yang ditambahkannya ke layer network, sedangkan pada layer network ditambahkan header alamat tujuannya supaya data bisa sampai pada komputer tujuannya.

Setelah aplikasi memuat data, software dan hardware pada komputer menambahkan header dan trailernya. Pada layer fisik dapat menggunakan medianya untuk mengirimkan sinyal untuk transmisi (langkah 2 Gambar 2.16).

Disisi penerima (langkah 3 Gambar 2.16), Host B mulai mengatur interaksi antar layer pada host B. Panah keatas (langkah 4 Gambar 2.16) menunjukkan proses pemecahan header dan trailer sehingga pada akhirnya data dapat diterima oleh pengguna di host B.

Apabila komunikasi yang terjadi antar 2 komputer masih harus melewati suatu media tertentu, semisal router. Maka bentuk dari interaksi OSI layer dapat dilihat seperti Gambar 2.17.



Gambar 2.17 Interaksi OSI Layer pada komunikasi melalui sebuah perantara, misal Router

2.4. Data Enkapsulasi

Konsep penempatan data dibalik suatu header dan trailer untuk tiap layer disebut enkapsulasi (*encapsulation*). Pada Gambar 2.16 terlihat pada tiap layer diberikan suatu header tambahan, kemudian ditambahkan lagi header pada layer berikutnya, sedangkan pada layer 2 selain ditambahkan header juga ditambahkan trailer. Pada layer 1 tidak menggunakan header dan trailer.

Pada pemrosesan layer 5, 6 dan 7 terkadang tidak diperlukan adanya header. Ini dikarenakan tidak ada informasi baru yang perlu diproses. Sehingga untuk layer tersebut bisa dianggap 1 proses.

Sehingga langkah-langkah untuk melakukan data enkapsulasi dapat dijabarkan sebagai berikut :

Langkah 1 **Membuat data** – artinya sebuah aplikasi memiliki data untuk dikirim

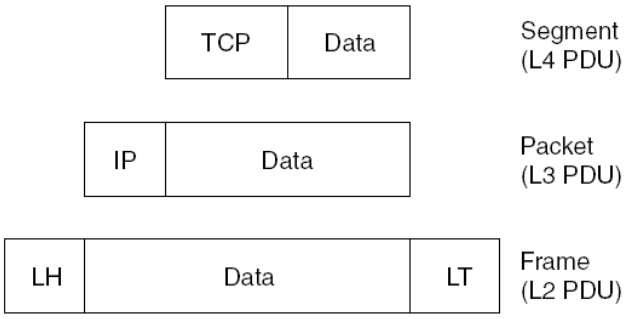
Langkah 2 **Paketkan data untuk di transportasikan** – artinya pada layer transport ditambahkan header dan masukkan data dibalik header. Pada proses ini terbentuk L4PDU.

Langkah 3 **Tambahkan alamat tujuan layer network pada data** – layer network membuat header network, dimana didalamnya terdapat juga alamat layer network, dan tempatkan L4PDU dibaliknya. Disini terbentuk L3PDU.

Langkah 4 **Tambahkan alamat tujuan layer data link pada data** – layer data link membuat header dan menempatkan L3PDU dibaliknya, kemudian menambahkan trailer setelahnya. Disini terbentu L2PDU.

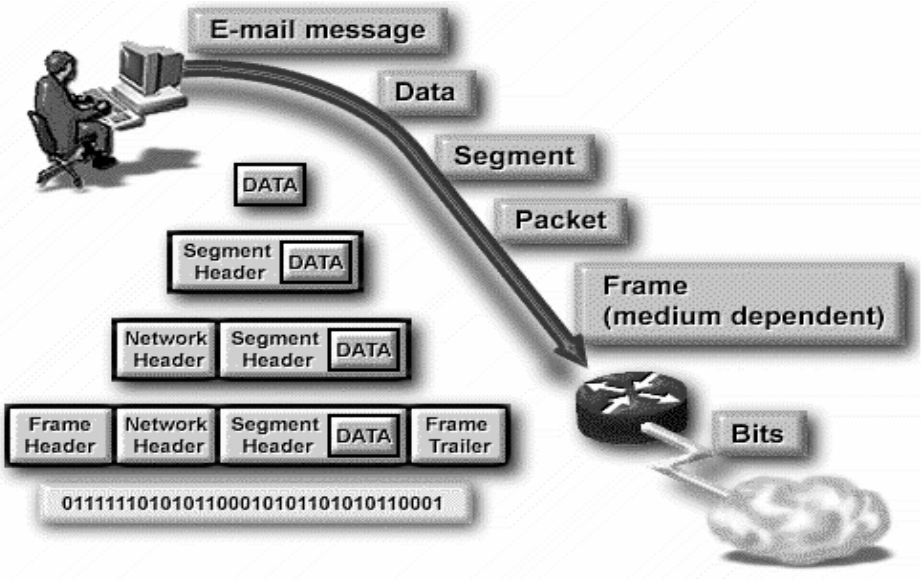
Langkah 5 **Transmit dalam bentu bit** – pada layer fisikal, lakukan encoding pada sinyal kemudian lakukan pengiriman frame.

Sehingga pemrosesannya akan mirip dengan model TCP/IP. Pada tiap layer terdapat LxPDU (Layer N Protocol Data Unit), dimana merupakan bentuk dari byte pada header-trailer pada data. Pada tiap-tiap layer juga terbentuk bentukan baru, pada layer 2 PDU termasuk header dan trailer disebut bentukan *frame*. Pada layer 3 disebut paket (*packet*) atau terkadang *datagram*. Sedangkan pada layer 4 disebut segmen (*segment*). Sehingga dapat digambarkan pada Gambar 2.18.



Gambar 2.18 Frame, Paket dan Segmen

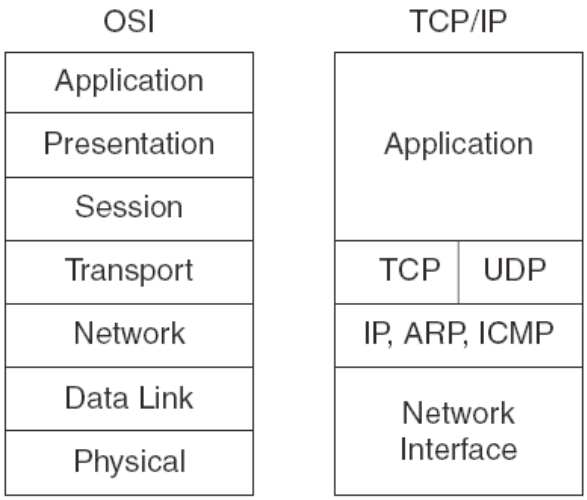
Sehingga bila pada contoh pengiriman email proses enkapsulasi yang terjadi dapat digambarkan pada Gambar 2.19.



Gambar 2.19 Proses enkapsulasi pada pengiriman E-Mail

2.5. Model referensi OSI dan TCP/IP

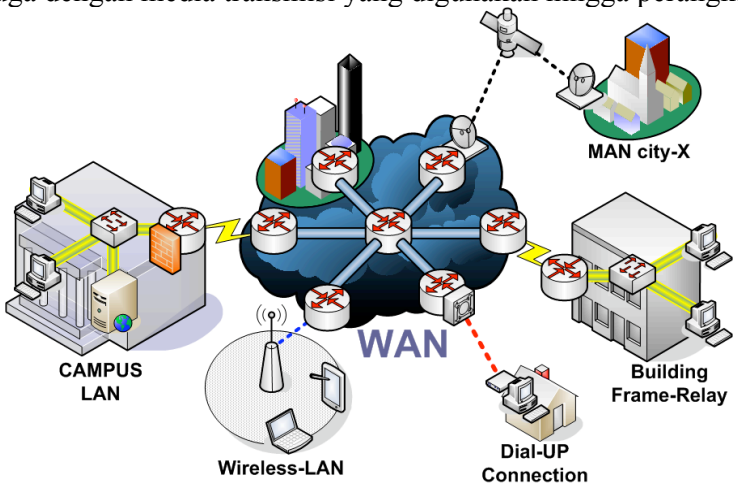
Apabila dibandingkan antara model OSI dan model TCP/IP dapat digambarkan pada Gambar 2.20.



Gambar 2.20 Perbandingan model OSI dan TCP/IP

Bab 3. Perangkat Jaringan

Bab ini berisikan tentang berbagai macam perangkat jaringan yang dapat dilalui oleh protokol TCP/IP, begitu juga dengan media transmisi yang digunakan hingga perangkat penyalurnya.



Gambar 3.1 Internetworking (WAN, MAN, LAN)

Distance Between CPUs	CPUs are in the same	Icon	Name
0.1 m	Printed circuit board Personal data asst.		Motherboard Personal Area Network (PAN)
1.0 m	Millimeter Mainframe		Computer System Network
10 m	Room		Local Area Network (LAN) Your classroom
100 m	Building		Local Area Network (LAN) Your school
1000 m = 1 km	Campus		Local Area Network (LAN) Stanford U.
10,000 m = 10 km	City		Metropolitan Area Network (MAN) San Francisco
100,000 m = 100 km	Country		Wide Area Network (WAN) Cisco Systems, Inc.
1,000,000 m = 1,000 km	Continent		Wide Area Network (WAN) Africa
10,000,000 m = 10,000 km	Planet		Wide Area Network (WAN) The internet
100,000,000 m = 100,000 km	Earth-moon system		Wide Area Network (WAN) Earth & artificial satellites
1,000,000,000 m = 1,000,000 km	Solar system		Solar Area Network (SAN)
71,000,000 km	Galaxy		Star Trek Area Network (STAN)

Gambar 3.2 Perbandingan Jaringan Komputer

3.1. Network Interface

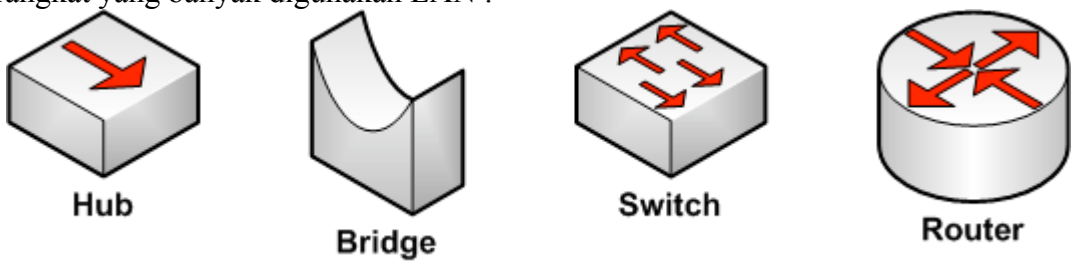
3.1.1. Local Area Network (LAN)

LAN adalah jaringan komputer yang mencover area lokal, seperti rumah, kantor atau group dari bangunan. LAN sekarang lebih banyak menggunakan teknologi berdasar IEEE 802.3 Ethernet switch, atau dengan Wi-Fi. Kebanyakan berjalan pada kecepatan 10, 100, atau 1000 Mbps.

Perbedaan yang menyolok antara Local Area Network (LAN) dengan Wide Area Network (WAN) adalah menggunakan data lebih banyak, hanya untuk daerah yang kecil, dan tidak memerlukan sewa jaringan.

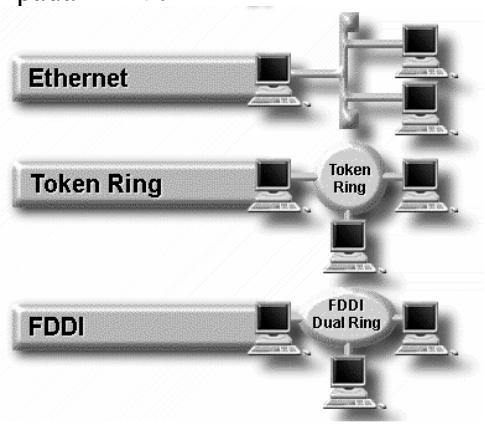
Walaupun sekarang ethernet switch yang paling banyak digunakan pada layer fisik dengan menggunakan TCP/IP sebagai protokol, setidaknya masih banyak perangkat lainnya yang dapat digunakan untuk membangun LAN. LAN dapat dihubungkan dengan LAN yang lain menggunakan router dan leased line untuk membentuk WAN. Selain itu dapat terkoneksi ke internet dan bisa terhubung dengan LAN yang lain dengan menggunakan tunnel dan teknologi VPN.

Perangkat yang banyak digunakan LAN :



Gambar 3.3 Perangkat LAN

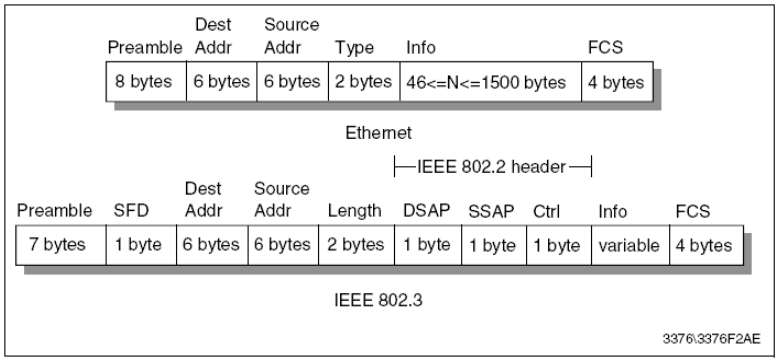
Teknologi yang digunakan pada LAN :



Gambar 3.4 Teknologi LAN

3.1.1.1. Ethernet dan IEEE 802.x Local Area Network

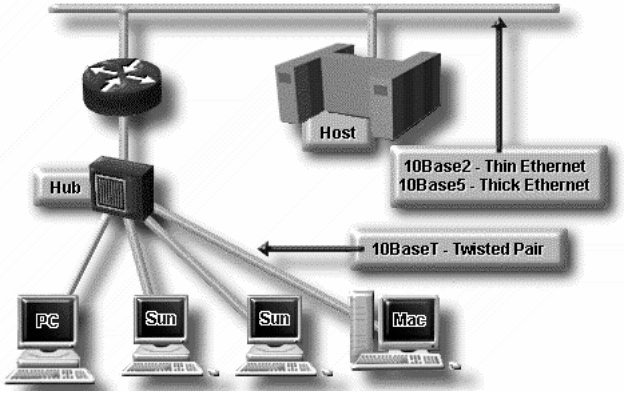
Perangkat jaringan yang paling banyak digunakan dengan standarisasi IEEE 802.3, format data dapat dilihat pada Gambar 3.5.



Gambar 3.5 Format frame untuk Ethernet dan IEEE 802.3

Pada layer data link digunakan IEEE 802.2 yaitu *Logical Link Controler (LLC)* dimana digunakan pada Media Access Control (MAC).

Beberapa teknologi Ethernet antara lain seperti pada Gambar 3.6.



Gambar 3.6 Ethernet IEEE 802.3

Untuk teknologi Ethernet digunakan format :

[x][y][z]

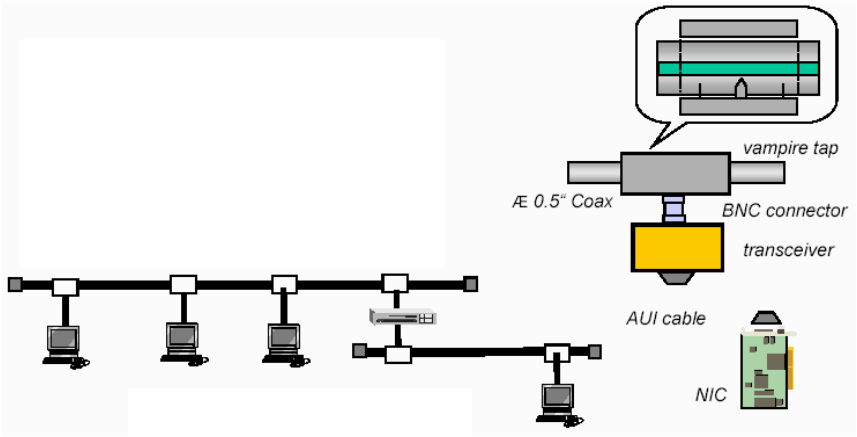
Contoh: 10BaseT, dimana artinya
10, adalah kecepatan dengan satuan Mbps. Selain 10 ada juga 100, 1000
Base, adalah teknologi yang digunakan berupa Baseband. Selain itu ada juga Broadband
T, adalah Twisted Pair, dimana media yang digunakan adalah kabel berpilin (*twisted pair*)

Ethernet

Coax

10Base-5

Disebut juga sebagai teknologi thick ethernet. Dimana perangkat yang digunakan seperti pada Gambar 3.7. Teknologi ini digunakan pada jaringan Token Ring (IEEE 802.5), dimana jaringan yang terbentuk seperti lingkaran.



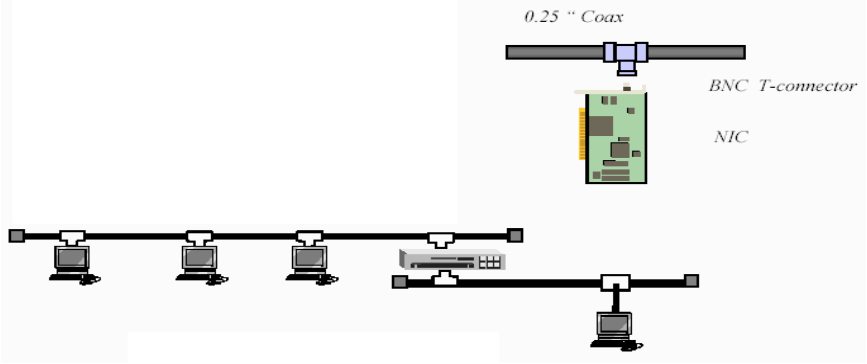
Gambar 3.7 Ethernet 10Base5

Keterangan :

- tap : tidak perlu memotong kabel
- transceiver : digunakan sebagai pengirim / penerima, collision detection, dan isolasi electric
- AUI : Attachment User Interface
- Digunakan untuk jaringan backbone
- Jarak maksimum untuk tiap segmen = 500m
- Jumlah maksimum host per segmen = 100
- Jarak minimum antar 2 station = 2.5m
- Jarak maksimum antar 2 station = 2.8km

10Base-2

Disebut juga sebagai teknologi thin ethernet. Dimana perangkat yang digunakan seperti pada Gambar 3.8.



Gambar 3.8 Ethernet 10Base2

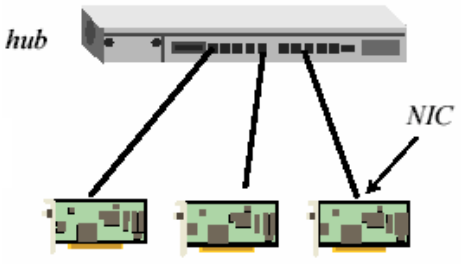
Keterangan :

- Menggunakan BNC konektor
- Digunakan pada LAN perkantoran
- Jarak maksimum segmen = 185m
- Jumlah maksimum station per segmen = 30
- Jarak minimum antar 2 station = 0.5m
- Jarak maksimum antar 2 station = 925m

Tembaga (cooper)

10Base-T

Teknologi jaringan untuk LAN dimana menggunakan hub sebagai repeater. Ilustrasi Ethernet 10BaseT seperti pada Gambar 3.9.



Gambar 3.9 Ethernet 10BaseT

Apabila menggunakan T berarti menggunakan media Twisted Pair, dan bila menggunakan F berarti menggunakan media Fiber Optic. Untuk perangkat disisi pengguna disebut juga *Network Interface Card* (NIC).

Fiber

10Base-F

Teknologi yang menggunakan fiber optic dan banyak digunakan untuk menghubungkan antar gedung. Jarak maksimum segmen yang diperbolehkan adalah 2000m.

Fast Ethernet

Copper

100Base-T2

Data dikirimkan melalui 2 pasang kabel tembaga

100Base-T4

Jaringan ethernet dengan kecepatan hingga 100 (fast ethernet). Jarak maksimum per segmen adalah 100m dengan menggunakan kabel twisted pair kategori 3.

100Base-Tx

Jaringan ehternet berkecepatan tinggi 100Mbps. Jarak maksimum persegmen adalah 100m full duplex. Jaringan ini menggunakan kabel twisted pair.

Fiber

100Base-FX

Jaringan ehternet berkecepatan tinggi 100Mbps. Jarak maksimum per segmen adalah 2000m full duplex dengan menggunakan media 2 kabel fiber optik.

100Base-SX

Jaringan ethernet menggunakan 2 kabel fiber optik untuk transmit dan receive dengan jarak maksimum 300m

100Base-BX

Jaringan ethernet menggunakan 1 kabel fiber optik dengan tipe singlemode.

Gigabit Ethernet

Fiber

1000Base-SX

Jaringan ethernet dengan kecepatan 1000Mbps. Dengan menggunakan media fiber optik dengan jarak maksimum per segmen 550m. Fiber optik yang digunakan adalah tipe multimode (50, 62.5 mikron)

1000Base-LX

Jaringan ethernet dengan kecepatan 1000Mbps. Dengan menggunakan media fiber optik dengan jarak maksimum per segmen hingga 5000m. Fiber optik yang digunakan adalah tipe singlemode (10 mikron) atau multimode (50, 62.5 mikron)

1000Base-CX

Jaringan ethernet dengan kecepatan 1000Mbps. Dengan menggunakan media kabel Twisted Pair yaitu 2 pasang STP. Jarak maksimum per segmen adalah 25m.

Cooper

1000Base-TX

Jaringan ethernet dengan kecepatan 1000Mbps. Dengan menggunakan media kabel Twisted Pair yaitu 4 pasang UTP. Jarak maksimum per segmen adalah 100m.

10Gigabit Ethernet

Fiber

LAN Phy

10GBase-SR

Jaringan 10Gigabit untuk jarak pendek (short-range), digunakan untuk jarak 26m hingga 82m. Bisa mencapai 300m apabila menggunakan 50um 2000MHz-km multimode FO

10GBase-LRM

Mencapai jarak 220m dengan menggunakan FDDI-grade 62.5 μm multimode FO.

10GBase-LR

Mencapai jarak 10km dengan menggunakan 1310 nm single-mode FO

10GBase-ER

Mencapai jarak 40km dengan menggunakan 1550 nm single-mode FO

10GBase-LX4

Jaringan 10Gigabit dengan menggunakan teknologi wavelength division multiplexing hingga mencapai jarak 240m – 300m. Bisa mencapai 10km dengan menggunakan FO single-mode dengan ukuran 1310nm.

WAN Phy

10GBase-SW, 10GBase-LW, dan 10GBase-EW digunakan untuk jaringan WAN, digunakan bersama dengan OC-192/STM-64 SDH/SONET.

Cooper

10GBase-CX4

Menggunakan 4 jalur kabel tembaga, hingga mencapai 15m.

10GBase-T

Menggunakan kabel UTP / STP dengan category 6 dan 7.

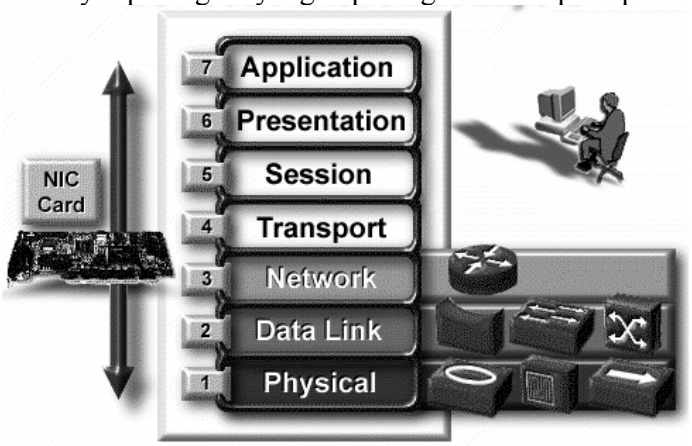
Hub, Switch dan Router

Perangkat yang digunakan untuk teknologi ini antara lain:

- Hub, Repeater: perangkat ini bekerja pada layer 1

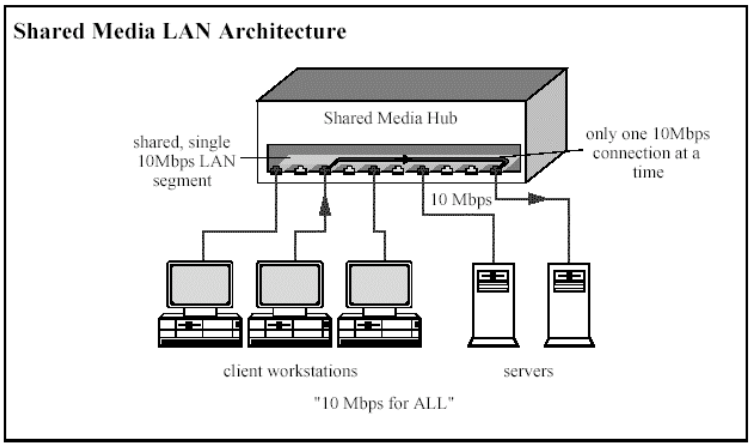
- Switch, bridge: perangkat ini bekerja pada layer 2
- Router: perangkat ini bekerja pada layer 3

Sehingga menurut OSI layer perangkat yang dapat digunakan seperti pada Gambar 3.10.

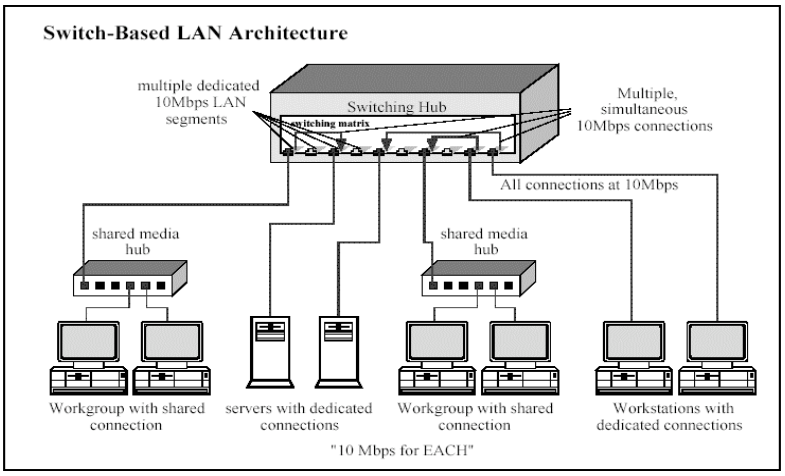


Gambar 3.10 Perangkat Jaringan sesuai dengan Layer

Perbedaan cara kerja Hub dan Switch dapat dilihat pada Gambar 3.11 dan Gambar 3.12.



Gambar 3.11 Cara kerja HUB

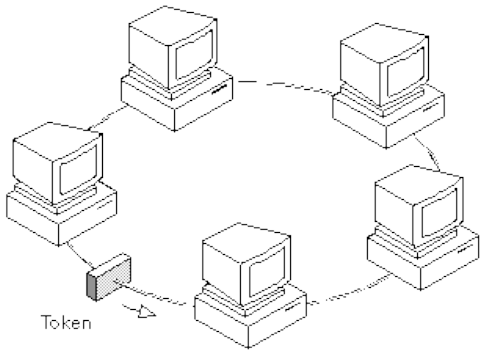


Gambar 3.12 Cara kerja Switch

3.1.1.2. Token Ring

Token Ring dikembangkan oleh IBM pada tahun 1980 dan menjadi standar IEEE 802.5. Menjadi berkembang setelah melebihi kemampuan dari 10Base-T.

Token ring merupakan jaringan bertopologi star, dengan Multistation Access Unit (MAU) sebagai pusat jaringan. MAU berfungsi seperti HUB hanya saja data bergerak dengan 1 arah. Data bergerak seperti lingkaran pada MAU.



Gambar 3.13 Token Ring

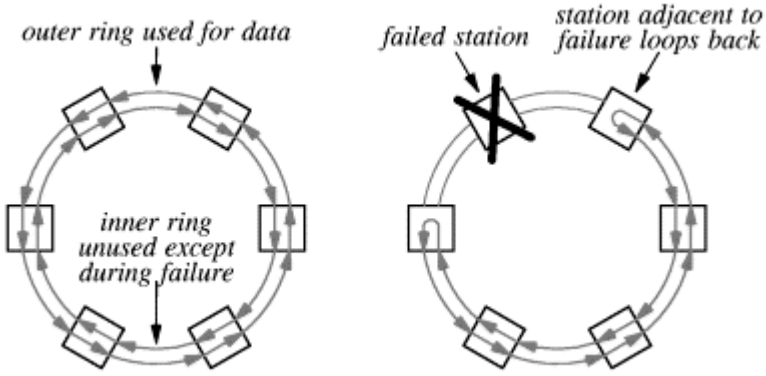
Untuk mengakses jaringan diperlukan yang namanya token. Token dilempar ke jaringan dan akan menerima data dengan dikirimkan kembali ke token si pengirim.

Dengan adanya teknologi switch pada Ethernet, token ring menjadi tidak banyak digunakan.

3.1.1.3. Fiber Distribution Data Interface (FDDI)

FDDI merupakan standar untuk jaringan fiber optik dengan kecepatan 100Mbps. Pada OSI Model FDDI diilustrasikan seperti pada Gambar 3.14. RFC yang menerangkan FDDI adalah RFC 1188.

FDDI bekerja dengan menggunakan 2 jalur berbentuk RING, dimana apabila terjadi kerusakan pada suatu station maka pada station sebelumnya akan membuat loopback sehingga jaringan tidak terputus.



Gambar 3.14 Cara kerja FDDI

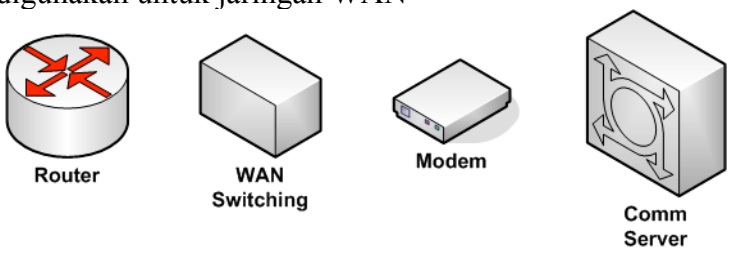
3.1.2. Wide Area Network (WAN)

WAN adalah jaringan komputer dimana memiliki cakupan daerah yang lebih luas. Contoh dari WAN adalah internet.

Jenis koneksi WAN dapat dibedakan menjadi

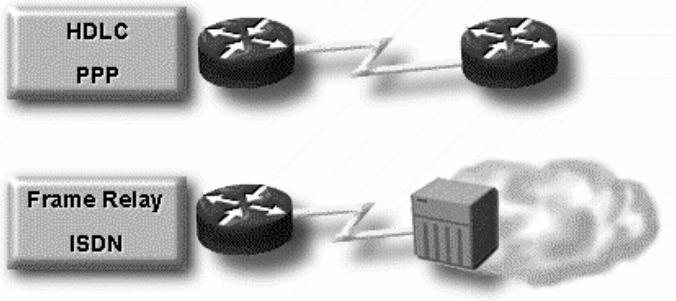
Akronim WAN	Nama WAN	Bandwidth Maksimum	Penggunaan	Tipe Pelayanan (service)
POTS	Plain Old Telephone Service	4 KHz analog	Standar	Circuit Switching
ISDN	Integrated Services Digital Network	128 Kbps	Data dan voice secara bersamaan	
X.25	X.25		Radio Paket, workshore	Packet-Swithing
Frame Relay	Frame Relay	1.544Mbps	Flexible workshore	
ATM	Asynchronous Transfer Mode	622Mbps	High Power network	Cell-Swithing
SMDS	Switched Multimegabit Data Service	1.544 & 44.736Mbps	MAN, variant dari ATM	
T1, T3, E1, E3	T1, T3, E1, E3	1.544 & 44.736 Mbps	Telecomunication	Dedicated Digital
xDSL	Digital Subcriber Line	384 kbps	Teknologi baru melalui line telepon	
Dial-up Modem	Modem	56kbps	Teknologi lama yang menggunakan jalur telepon	Lainnya
Cable Modem	Cable Modem	10Mbps	TV Kabel	
Terresterial Wireless	Wireless	<5Mbps	Microwave & link dengan laser	
Satellite Wireless	Wireless	<5Mbps	Microwave & link dengan laser	
SONET	Synchronous Optical Network	9.992Mbps	Jaringan cepat menggunakan FO	

Perangkat yang digunakan untuk jaringan WAN



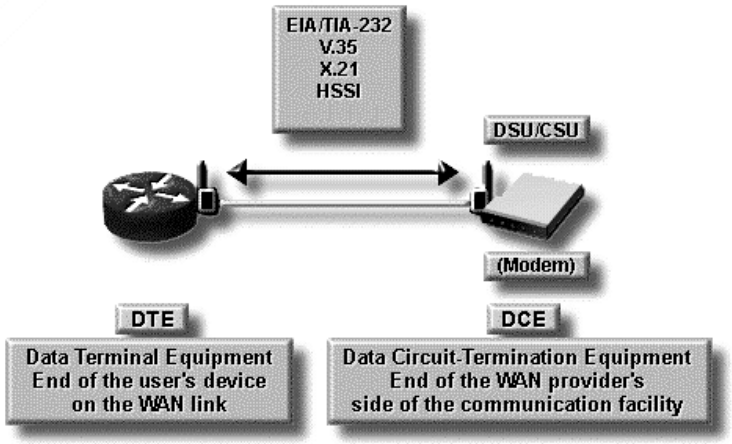
Gambar 3.15 Perangkat WAN

Cara menghubungkan perangkat WAN ada 2 macam yaitu, menghubungkan langsung secara point-to-point atau melalui perangkat swithing lainnya.



Gambar 3.16 Cara menghubungkan perangkat WAN

Sedangkan pada bentuk fisiknya perangkat WAN akan disambungkan seperti berikut



Gambar 3.17 Bentuk sambungan fisik perangkat WAN

Contoh perangkat WAN :

3.1.2.1. Serial Line IP (SLIP)

SLIP merupakan standar yang digunakan pada jaringan point-to-point dengan koneksi serial dimana berjalan protokol TCP/IP, diterangkan pada RFC 1055. Protokol ini telah diganti oleh Point-to-Point Protocol (PPP). Contoh koneksi yang menggunakan SLIP adalah hubungan antar PC dengan menggunakan null-modem

3.1.2.2. Point-to-Point Protocol (PPP)

PPP diterangkan di standard protocol nomer 51, dan RFC 1661 dan RFC 1662. PPP memiliki 3 komponen inti, yaitu :

1. Menggunakan enkapsulasi datagram melalui link serial
2. Link Control Protocol digunakan untuk menyambungkan, menkonfigurasi, dan testing koneksi data link
3. Network Control Protocol digunakan untuk menghubungkan protokol yang berbeda.

Phase yang dilakukan untuk membuat koneksi dengan PPP yaitu:

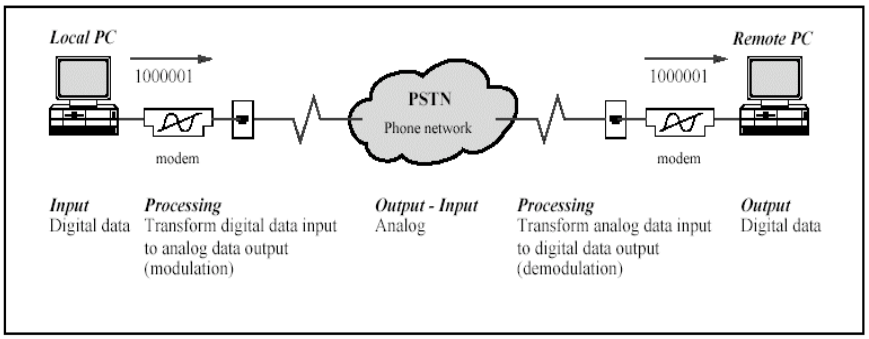
1. Pembentukan link dan negosiasi konfigurasi
2. Mengukur kualiti dari link
3. Authentikasi
4. Negosiasi configurasi protokol layer Network
5. Pemutusan link

Untuk media yang lainnya PPP menggunakan enkapsulasi melalui PPP.

Perangkat yang biasa digunakan pada komunikasi PPP antara lain modem.

Gambar 3.18 Modem

Komunikasi yang dilakukan dengan modem dapat dilakukan seperti Gambar 3.19.



Gambar 3.19 Koneksi menggunakan Modem

3.1.2.3. Integrated Services Digital Network (ISDN)

Komunikasi ini menggunakan enkapsulasi PPP melalui ISDN, dimana dibahas pada RFC1618.

ISDN Basic Rate Interface (BRI) mendukung 2 B-Channel dengan kapasitas 64kbps dan 16kbps D-Channel digunakan untuk kontrol informasi. B-Channel hanya bisa digunakan untuk voice saja atau data saja.

ISDN Primary Rate Interface (PRI) mendukung beberapa B-Channel (biasanya 30) dan 64kbps D-Channel.

Perangkat ISDN menggunakan tipe perangkat DCE/DTE.

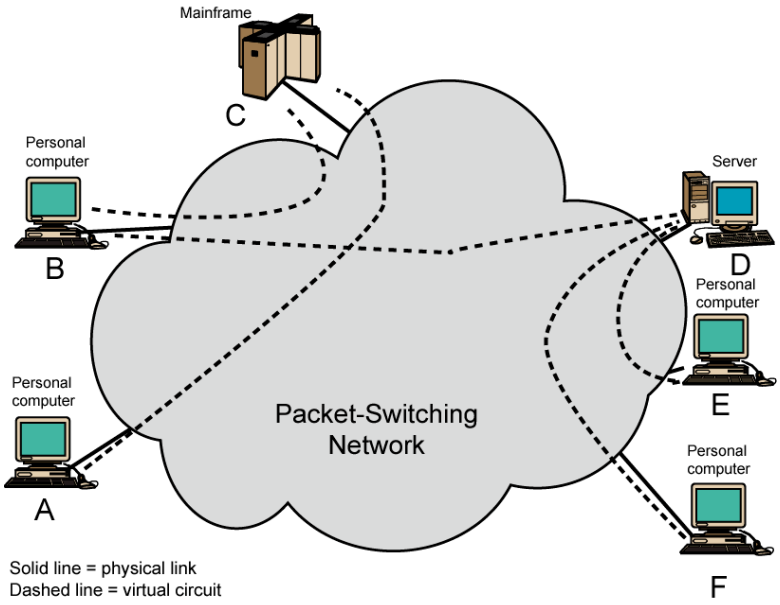
3.1.2.4. X.25

Enkapsulasi IP melalui X.25 didokumentasikan di RFC1356. X.25 merupakan interface penghubung antara host dengan packet switching, dan banyak digunakan pada ISDN.

Layer pada X.25:

- Physical
  - Merupakan interface antar station dengan node
  - DTE pada perangkat user
  - DCE pada node
  - Menggunakan X.21
  - Merupakan sequence dari frame
- Link
  - Link Access Protocol Balance (LAPB), merupakan bagian dari HDLC
- Packet
  - Merupakan eksternal virtual circuit
  - Merupakan logical circuit antar subcriber

Penggunaan X.25 dapat dilihat pada Gambar 3.20.



Gambar 3.20 Penggunaan X.25

3.1.2.5. Frame Relay

Frame Relay merupakan pengembangan dari X.25.

Karakteristik frame relay :

- Call Control dilakukan pada koneksi logical.
- Multiplexing dan switching dilakukan di layer 2
- Tidak ada flow control dan error control pada setiap hop
- Flow control dan error control dilakukan di layer atasnya
- Menggunakan single data frame.

3.1.2.6. PPP over SONET dan SDH Circuit

Synchronous Optical Network disingkat SONET, Synchronous Digital Hierarchy disingkat SDH link, koneksi PPP over SONET atau SDH didokumentasikan di RFC1619.

Kecepatan dasar dari PPP over SONET/SDH adalah STS-3c/STM-1 pada kecepatan 155.52 Mbps.

3.1.2.7. Asynchronous Transfer Mode (ATM)

ATM mengirimkan data secara potongan diskrit. Memiliki koneksi multi logical melalui koneksi fisik tunggal. Paket ATM yang terkirim pada koneksi logic disebut cell. ATM mampu meminimalis error dan flow control. ATM memiliki data rate 25.6Mbps sampe 622.08Mbps.

3.2. Media Transmisi

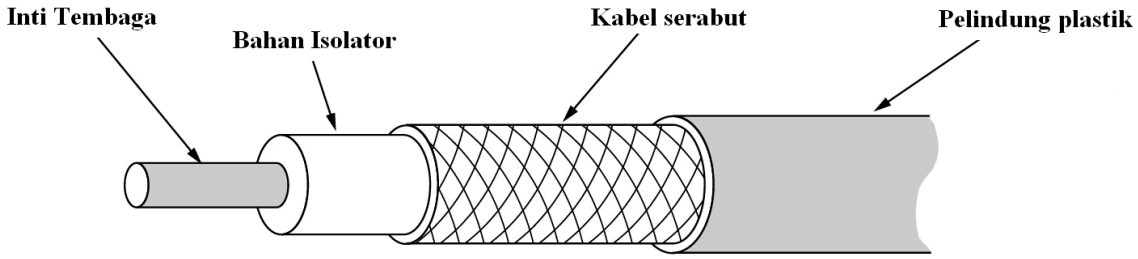
3.2.1. Media Terarah (Guided Transmission Data)

Suatu media yang digunakan untuk mengirimkan data, dimana arah ujung yang satu dengan ujung yang lainnya sudah jelas, contoh : kabel.

3.2.1.1. Coaxial

Kabel data yang menggunakan material tembaga dimana terdapat 2 bagian yaitu :

- Kabel inti ditengah
- Kabel serabut disisi samping dengan dipisahkan oleh suatu isolator

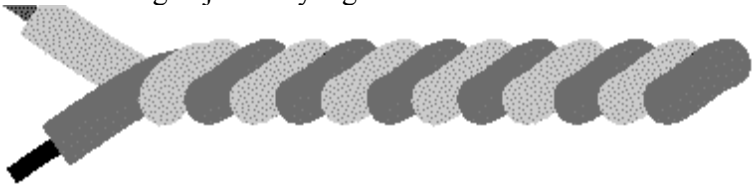


Gambar 3.21 Kabel Coaxial

Kabel ini menggunakan konektor Bayonet Nut Connector (BNC)

3.2.1.2. Twisted Pair

Kabel berpilin (*Twisted Pair*), menggunakan kabel berpasangan dimana tujuannya untuk menghilangkan efek *crosstalk*. Banyak digunakan untuk jaringan LAN, dikarenakan mampu mengirimkan bandwidth dengan jumlah yang besar.



Gambar 3.22 Twisted Pair

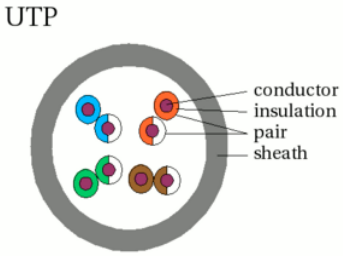
Kabel ini menggunakan konektor seri *Registered Jack* (RJ), dan tergantung dari jenis kategorinya. Untuk kategori 2 menggunakan RJ11 sedangkan untuk kategori 5 keatas menggunakan RJ45.

Tabel 3.1 Daftar Kategori Kabel Berpilin

Kategori (Category)	Data rate maksimum	Penggunaan
CAT 1	1 Mbps (1MHz)	Analog voice, ISDN
CAT 2	4 Mbps	Token Ring
CAT 3	16 Mbps	Voice dan data 10BaseT
CAT 4	20 Mbps	16 Mbps Token Ring
CAT 5	100Mbps 1000Mbps (4 pasang)	ATM
CAT 5E	1000Mbps	Ethernet
CAT 6	Mencapai 400MHz	Superfast broadband
CAT 6E	Mencapai 500MHz	10GBaseT
CAT 7	Mencapai 1.2GHz	Full Motion Video Teleradiology

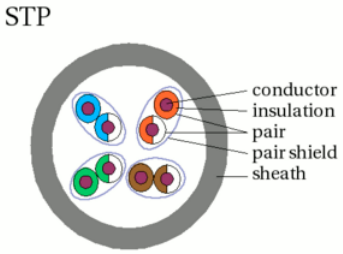
Jenis kabel berpilin menurut pelindungnya dibagi menjadi :

- Unshielded Twisted Pair (UTP)



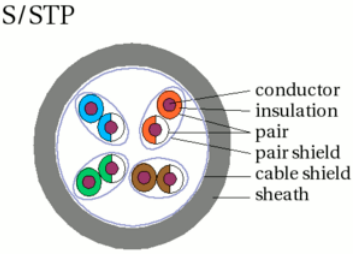
Gambar 3.23 UTP

- Shielded Twisted Pair (STP)



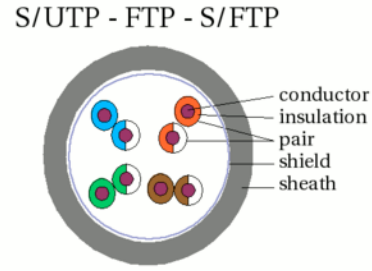
Gambar 3.24 STP

- Screened Shielded Twisted Pair (S/STP)



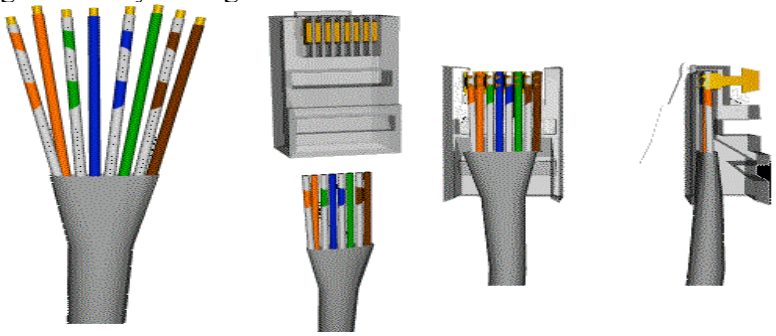
Gambar 3.25 S/STP

- Screened Unshielded Twisted Pair (S/UTP) / Foiled Twisted Pair (FTP)

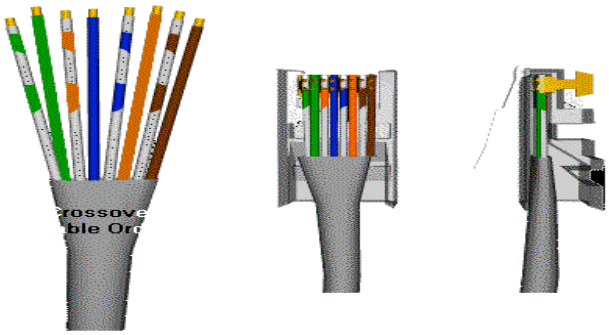


Gambar 3.26 S/UTP

Untuk pemasangan kabelnya mengikuti aturan TIA/EIA-586-A/B



Gambar 3.27 TIA/EIA-586-B

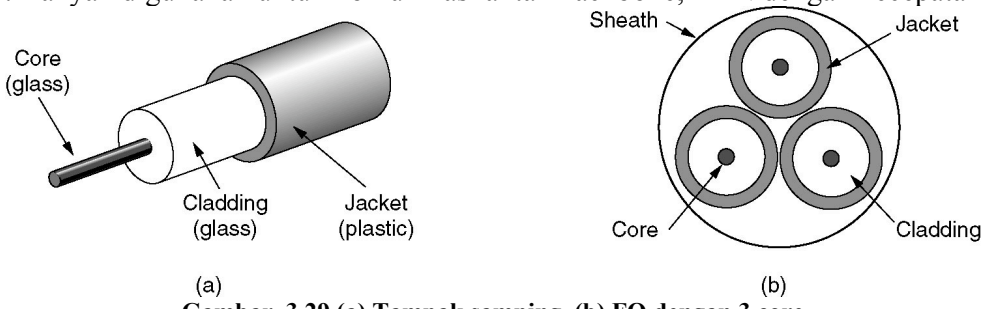


Gambar 3.28 TIA/EIA-586-A

Apabila kedua ujung menggunakan aturan yang sama, kabel tersebut disebut *Straight-Through*, sedangkan bila berbeda disebut *Cross-Over*.

3.2.1.3. Fiber Optic

Jenis kabel yang satu ini tidak menggunakan tembaga (*cooper*), melainkan serat optik. Dimana sinyal yang dialirkan berupa berkas cahaya. Mampu mengirimkan bandwidth lebih banyak. Banyak digunakan untuk komunikasi antar Backbone, LAN dengan kecepatan tinggi.

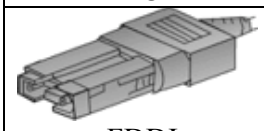
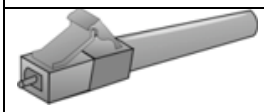
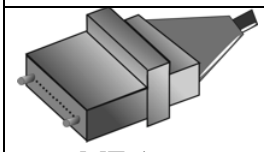
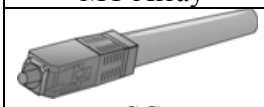
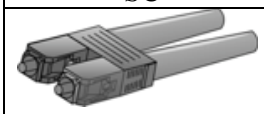
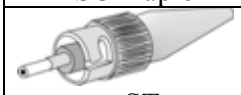


Gambar 3.29 (a) Tampak samping, (b) FO dengan 3 core

Berdasarkan jumlah sumber cahaya yang masuk pada core FO, kabel FO dibagi menjadi 2 yaitu:

- Multimode, jumlah sumber lebih dari 1. Menggunakan diameter core dengan ukuran 50 micron – 100 micron
- Singlemode, jumlah sumber 1. Menggunakan diameter core dengan ukuran 2 – 8 micron

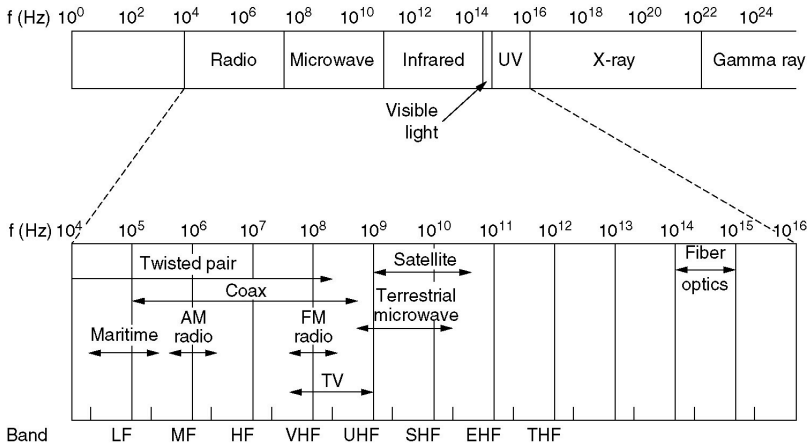
Tabel 3.2 Tipe Konektor FO

Connector	Insertion Loss	Repeatability	Tipe Fiber	Kegunaan
 FC	0.50-1.00 dB	0.20 dB	SM, MM	Datacom, Telecommunications
 FDDI	0.20-0.70 dB	0.20 dB	SM, MM	Fiber Optic Network
 LC	0.15 db (SM) 0.10 dB (MM)	0.2 dB	SM, MM	High Density Interconnection
 MT Array	0.30-1.00 dB	0.25 dB	SM, MM	High Density Interconnection
 SC	0.20-0.45 dB	0.10 dB	SM, MM	Datacom
 SC Duplex	0.20-0.45 dB	0.10 dB	SM, MM	Datacom
 ST	Typ. 0.40 dB (SM) Typ. 0.50 dB (MM)	Typ. 0.40 dB (SM) Typ. 0.20 dB (MM)	SM, MM	Inter-/Intra-Building, Security, Navy

3.2.2. Media Tidak Terarah (Un-Guided Transmission Data)

Suatu media yang digunakan untuk mengirimkan data, dimana arah ujung yang satu dengan ujung yang lainnya tersebar, contoh : nirkabel (*wireless*).

Komunikasi ini mengirimkan sinyal ke udara berdasarkan spektrum elektromagnetik

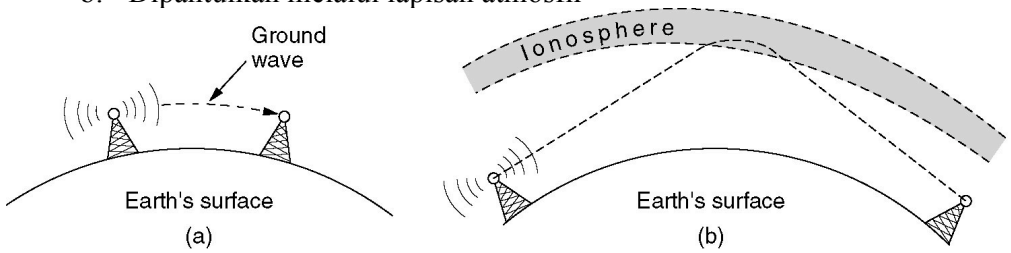


Gambar 3.30 Spektrum Elektromagnetik

3.2.2.1. Transmisi Radio

Perkembangan teknologi komunikasi radio sangat pesat, penggunaan wireless-LAN sudah semakin populer. Untuk mengirimkan data menggunakan komunikasi radio ada beberapa cara yaitu :

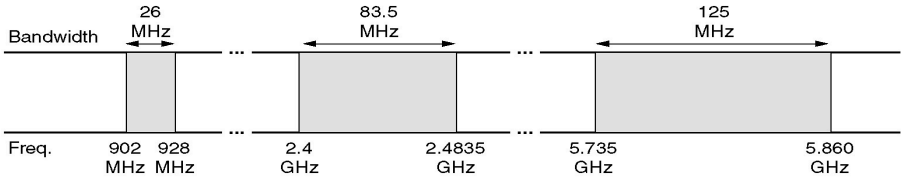
a. Memancarkan langsung, sesuai dengan permukaan bumi
b. Dipantulkan melalui lapisan atmosfir



Gambar 3.31 Komunikasi radio

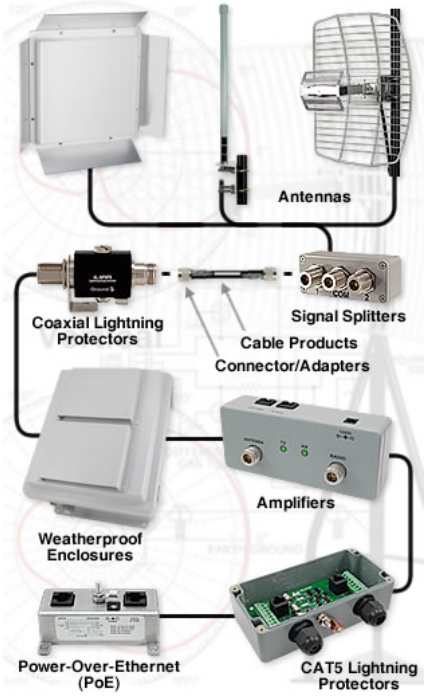
Komunikasi radio ini menggunakan frekuensi khusus supaya tidak mengakibatkan *interference* dengan penggunaan frekuensi lainnya, frekuensi yang boleh digunakan disebut ISM band. ISM singkatan dari *Industrial, Scientific and Medical*. Frekuensi yang bisa digunakan antara lain :

- 900 MHz
- 2.4 GHz
- 5.8 GHz



Gambar 3.32 ISM Band

Contoh penggunaan perangkat Wireless-LAN seperti pada Gambar 3.33.

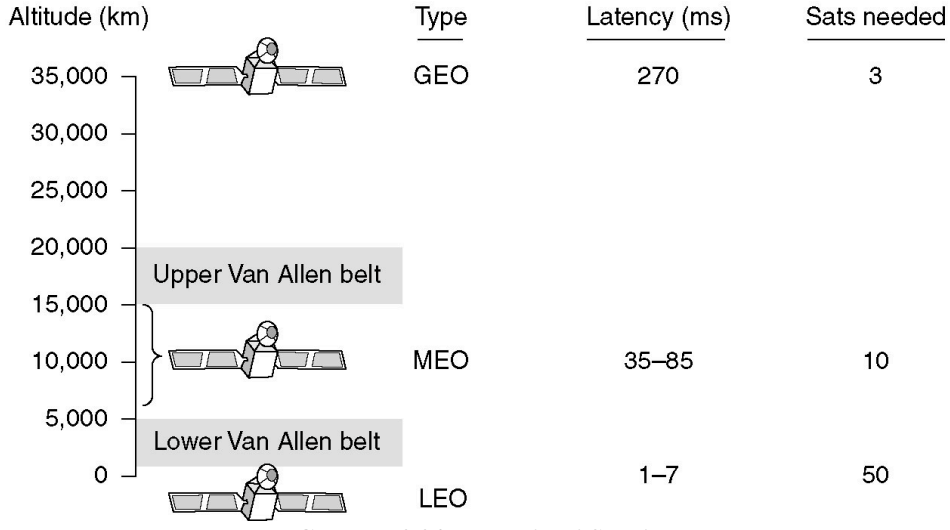


Gambar 3.33 Perangkat Wireless-LAN

3.2.2.2. Komunikasi Satelit

Komunikasi ini digunakan untuk komunikasi jarak jauh atau antar benua. Dimana untuk menghubungkannya diperlukan teknologi satelit. Menurut jaraknya satelit bisa dikategorikan menjadi :

- Geostationary
- Medium-Earth Orbit
- Low-Earth Orbit



Gambar 3.34 Komunikasi Satelit

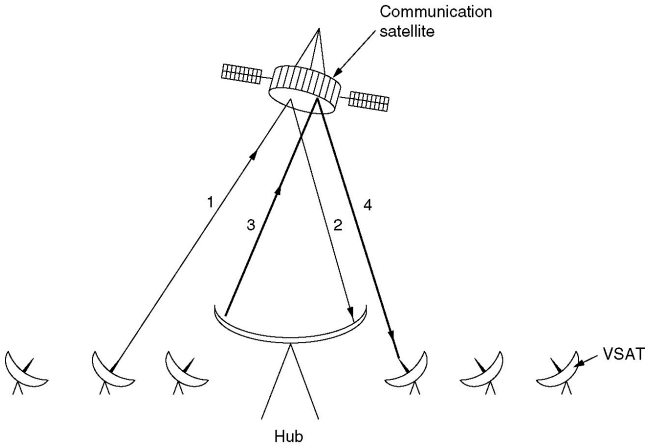
Komunikasi satelit menggunakan frekuensi / band.

Tabel 3.3 Frekuensi Kerja Satelit

Band	Downlink	Uplink	Bandwidth	Permasalahan
L	1.5 GHz	1.6GHz	15 MHz	Bandwidth rendah, saluran penuh

S	1.9 GHz	2.2 GHz	70 MHz	Bnadwidth rendah, saluran penuh
C	4.0 GHz	6 GHz	500 MHz	Interferensi Teresterial
Ku	11 GHz	14 GHz	500 MHz	Hujan
Ka	20 GHz	30 GHz	3500 MHz	Hujan, harga perangkat

Untuk menghubungi *site* yang lain, bisa dilakukan dengan *Very Small Aperture Terminal* (VSAT). VSAT adalah stasiun bumi 2 arah dengan antena parabola dengan diameter sekitar 3 – 10 meter.



Gambar 3.35 Komunikasi satelit dengan VSAT

Bab 4. Internet Protocol

IP adalah standard protokol dengan nomer STD 5. Standar ini juga termasuk untuk ICMP, dan IGMP. Spesifikasi untuk IP dapat dilihat di RFC 791, 950, 919, dan 992 dengan update pada RFC 2474. IP juga termasuk dalam protokol internetworking.

4.1. Pengalamatan IP

Alamat IP merupakan representasi dari 32 bit bilangan unsigned biner. Ditampilkan dalam bentuk desimal dengan titik. Contoh 10.252.102.23 merupakan contoh valid dari IP.

4.1.1. Alamat IP (IP Address)

Pengalamatan IP dapat di lihat di RFC 1166 – Internet Number. Untuk mengidentifikasi suatu host pada internet, maka tiap host diberi IP address, atau internet address. Apabila host tersebut tersambung dengan lebih dari 1 jaringan maka disebut *multi-homed* dimana memiliki 1 IP address untuk masing-masing interface. IP Address terdiri dari :

```
IP Address = <nomer network><nomer host>
```

Nomer network diatur oleh suatu badan yaitu *Regional Internet Registries* (RIR), yaitu :

- American Registry for Internet Number (ARIN), bertanggung jawab untuk daerah Amerika Utara, Amerika Selatan, Karibia, dan bagian sahara dari Afrika
- Reseaux IP Europeens (RIPE), bertanggung jawab untuk daerah Eropa, Timur Tengah dan bagian Afrika
- Asia Pasific Network Information Center (APNIC), bertanggung jawab untuk daerah Asia Pasific

IP address merupakan 32 bit bilangan biner dimana bisa dituliskan dengan bilangan desimal dengan dibagi menjadi 4 kolom dan dipisahkan dengan titik.

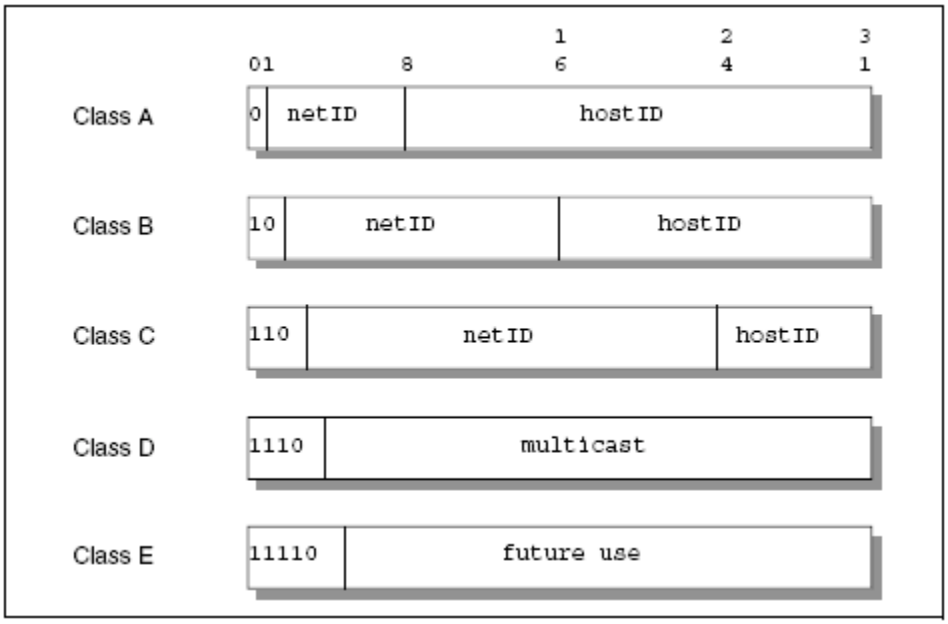
Bilangan biner dari IP address 128.2.7.9 adalah :
10000000 00000010 00000111 00001001

Penggunaan IP address adalah unik, artinya tidak diperbolehkan menggunakan IP address yang sama dalam satu jaringan.

4.1.2. Pembagian Kelas Alamat IP (Class-based IP address)

Bit pertama dari alamat IP memberikan spesifikasi terhadap sisa alamat dari IP. Selain itu juga dapat memisahkan suatu alamat IP dari jaringan. Network. Alamat Network (*network address*) biasa disebut juga sebagai *netID*, sedangkan untuk alamat host (*host address*) biasa disebut juga sebagai *hostID*.

Ada 5 kelas pembagian IP address yaitu :



Gambar 4.1 Pembagian Kelas pada IP

Dimana :

- Kelas A : Menggunakan 7 bit alamat network dan 24 bit untuk alamat host. Dengan ini memungkinkan adanya 2^7-2 (126) jaringan dengan 2^{24}-2 (16777214) host, atau lebih dari 2 juta alamat.
- Kelas B : Menggunakan 14 bit alamat network dan 16 bit untuk alamat host. Dengan ini memungkinkan adanya 2^{14}-2 (16382) jaringan dengan 2^{16}-2 (65534) host, atau sekitar 1 juga alamat.
- Kelas C : Menggunakan 21 bit alamat network dan 8 bit untuk alamat host. Dengan ini memungkin adanya 2^{21}-2 (2097150) jaringan dengan 2^8-2 (254) host, atau sekitar setengah juta alamat.
- Kelas D : Alamat ini digunakan untuk multicast
- Kelas E : Digunakan untuk selanjutnya.

Kelas A digunakan untuk jaringan yang memiliki jumlah host yang sangat banyak. Sedangkan kelas C digunakan untuk jaringan kecil dengan jumlah host tidak sampai 254. sedangkan untuk jaringan dengan jumlah host lebih dari 254 harus menggunakan kelas B.

4.1.3. Alamat IP yang perlu diperhatikan

- Alamat dengan semua bit = 0, digunakan untuk alamat jaringan (network address). Contoh 192.168.1.0
- Alamat dengan semua bit = 1, digunakan untuk alamat broadcast (broadcast address). Contoh 192.168.1.255
- Alamat loopback, alamat dengan IP 127.0.0.0 digunakan sebagai alamat loopback dari sistem lokal.

4.2. IP Subnet

Perkembangan internet yang semakin pesat, menyebabkan penggunaan IP semakin banyak, dan jumlah IP yang tersedia semakin lama semakin habis. Selain itu untuk pengaturan jaringan juga semakin besar karena jaringannya yang semakin besar. Untuk itu perlu dilakukan "pengecilan" jaringan yaitu dengan cara membuat subnet (*subneting*).

Sehingga bentuk dasar dari IP berubah dengan pertambahan *subnetwork* atau nomer subnet, menjadi

```
<nomer jaringan><nomer subnet><nomer host>
```

Jaringan bisa dibagi menjadi beberapa jaringan kecil dengan membagi IP address dengan pembaginya yang disebut sebagai *subnetmask* atau biasa disebut *netmask*. Netmask memiliki format sama seperti IP address.

Contoh penggunaan subnetmask :

- Dengan menggunakan subnetmask 255.255.255.0, artinya jaringan kita mempunyai 2^8-2 (254) jumlah host.
- Dengan menggunakan subnetmask 255.255.255.240, artinya pada kolom terakhir pada subnet tersebut 240 bila dirubah menjadi biner menjadi 11110000. Bit 0 menandakan jumlah host kita, yaitu 2^4-2 (14) host.

4.2.1. Tipe dari subneting

Ada 2 tipe subneting yaitu static subneting dan variable length subneting.

4.2.1.1. Static subneting

Subneting yang digunakan hanya memperhatikan dari kelas dari IP address. Contoh untuk jaringan kelas C yang hanya memiliki 4 host digunakan subneting 255.255.255.0. Dalam hal penggunaan ini akan memudahkan karena apabila ada penambahan host tidak perlu lagi merubah subnetmask, tetapi akan melakukan pemborosan sebanyak 250 alamat IP.

4.2.1.2. Variable Length Subneting Mask (VLSM)

Subneting yang digunakan berdasarkan jumlah host. Sehingga akan semakin banyak jaringan yang bisa dipisahkan.

4.2.1.3. Gabungan antara static subneting dan variable length subneting

Penggunaan subneting biasanya menggunakan static subneting. Tetapi karena suatu keperluan sebagian kecil jaringan tersebut menggunakan variable length subneting. Sehingga diperlukan router untuk menggabungkan kedua jaringan tersebut.

4.2.2. Cara perhitungan subnet

4.2.2.1. Menggunakan static subneting

Suatu jaringan menggunakan kelas A, menggunakan IP 10.252.102.23.

```
00001010 11111100 01100110 00010111          Alamat 32 bit
   10       252      102      23        Alamat desimal
```

Artinya 10 sebagai alamat network dan 252.102.23 sebagai alamat host.
Kemudian administrator menentukan bahwa bit 8 sampe dengan bit ke 24 merupakan alamat subnet. Artinya menggunakan subnetmask 255.255.255.0 (11111111 11111111 11111111 00000000 dalam notasi bit). Dengan aturan bit 0 dan 1 maka jaringan tersebut memiliki 2^{16}-2

(65534) subnet dengan masing-masing subnet memiliki jumlah host maksimum sebanyak 2^8-2 (254).

4.2.2.2. Menggunakan variable length subneting

Suatu jaringan menggunakan kelas C, dengan IP address 165.214.32.0. Jaringan tersebut ingin membagi jaringannya menjadi 5 subnet dengan rincian :

- Subnet #1 : 50 host
- Subnet #2 : 50 host
- Subnet #3 : 50 host
- Subnet #4 : 30 host
- Subnet #5 : 30 host

Hal ini tidak bisa dicapai dengan menggunakan static subneting. Untuk contoh ini, apabila menggunakan subneting 255.255.255.192 maka hanya akan terdapat 4 subnet dengan masing-masing subnet memiliki 64 host, yang dibutuhkan 5 subnet. Apabila menggunakan subnet 255.255.255.224, memang bisa memiliki sampe 8 subnet tetapi tiap subnetnya hanya memiliki jumlah host maksimal 32 host, padahal yang diinginkan ada beberapa subnet dengan 50 host.

Solusinya adalah dengan membagi subnet menjadi 4 subnet dengan menggunakan subnetmask 255.255.255.192 dan subnet yang terakhir dibagi lagi dengan menggunakan subnetmask 255.255.255.224. Sehingga akan didapatkan 5 subnet, dengan subnet pertama sampe ketiga bisa mendapatkan maksimal 64 host dan subnet ke empat dan kelima memiliki 32 host.

4.3. IP Routing

Fungsi utama dari sebuah IP adalah *IP routing*. Fungsi ini memberikan mekanisme pada router untuk menyambungkan beberapa jaringan fisik yang berbeda. Sebuah perangkat dapat difungsikan sebagai host maupun router.

Ada 2 tipe IP routing yaitu : direct dan indirect.

4.3.1. Tipe Routing

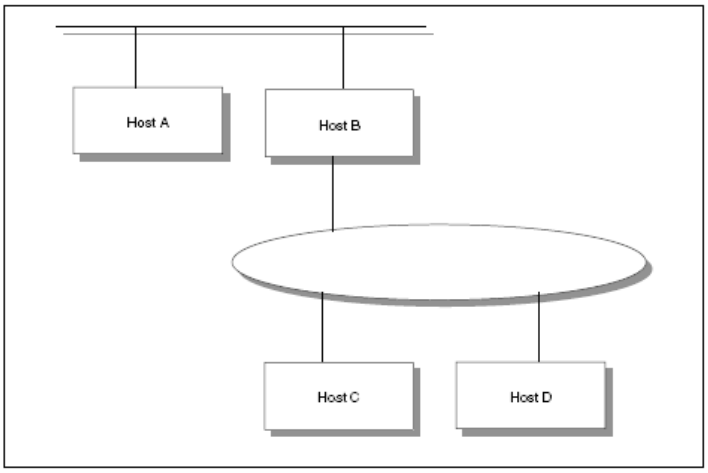
4.3.1.1. Direct Routing

Apabila host kita dengan tujuan berada dalam 1 jaringan. Maka data kita bila dikirimkan ketujuan akan langsung dikirimkan dengan mengenkapsulasi IP datagram pada layer phisical. Hal ini disebut dengan Direct Routing.

4.3.1.2. Indirect Routing

Apabila kita ingin mengirimkan suatu data ketujuan lain, dimana tujuan tersebut berada di jaringan yang berbeda dengan kita. Maka untuk itu dibutuhkan 1 IP address lagi yang digunakan sebagai IP gateway. Alamat pada gateway pertama (hop pertama) disebut indirect route dalam algoritma IP routing. Alamat dari gateway pertama yang hanya diperlukan oleh pengirim untuk mengirimkan data ke tujuan yang berada di jaringan yang berbeda.

Pada Gambar 4.2 akan diperlihatkan perbedaan direct dan indirect routing.



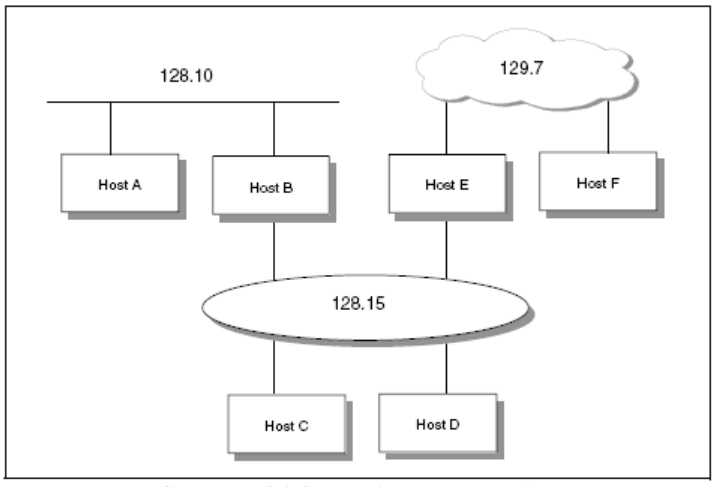
Gambar 4.2 Direct dan Indirect Route – Host C memiliki direct route terhadap Host B dan D, dan memiliki indirect route terhadap host A melalui gateway B

4.3.2. Table Routing

Menentukan arah dari berbagai direct route dapat dilihat dari list akan interface. Sedangkan untuk list jaringan dan gatewaynya dapat dikonfigurasi kemudian. List tersebut digunakan untuk fasilitas IP routing. Informasi tersebut disimpan dalam suatu tabel yang disebut tabel arah (*Routing Table*).

Tipe informasi yang ada pada table routing antara lain :

1. Direct route yang didapat dari interface yang terpasang
2. Indirect route yang dapat dicapai melalui sebuah atau beberapa gateway
3. Default route, yang merupakan arah akhir apabila tidak bisa terhubung melalui direct maupun indirect route.



Gambar 4.3 Skenario Table Routing

Gambar 4.3 menyajikan contoh suatu jaringan. Table Routing dari host D akan berisikan :

```
Destination    Router      Interface
129.7.0.0      E           Lan0
128.15.0.0     D           Lan0
128.10.0.0     B           Lan0
```

```
Default         B           Lan0
127.0.0.1       Loopback    Lo
```

Host D terhubung pada jaringan 128.15.0.0 maka digunakan direct route untuk jaringan ini. Untuk menghubungi jaringan 129.7.0.0 dan 128.10.0.0, diperlukan indirect route melalui E dan B.

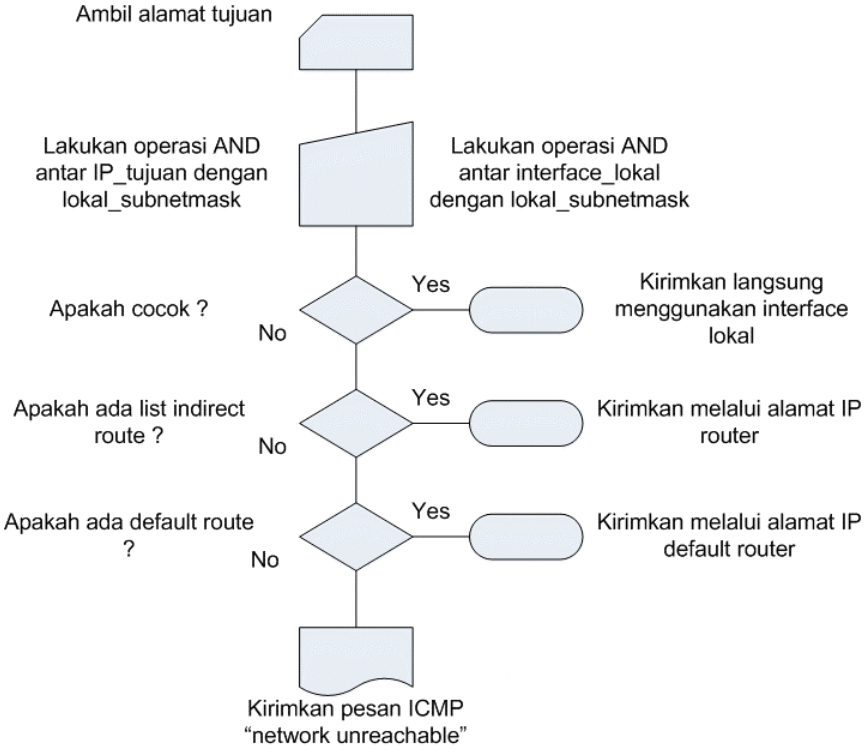
Sedangkan table routing untuk host F, berisikan :

```
Destination     Router      Interface
129.7.0.0       F           Wan0
Default         E           Wan0
127.0.0.1       Loopback    Lo
```

Karena jaringan selain 129.7.0.0 harus dicapai melalui E, maka host F hanya menggunakan default route melalui E.

4.3.3. Algoritma IP routing

Algoritma routing digambarkan pada Gambar 4.4.

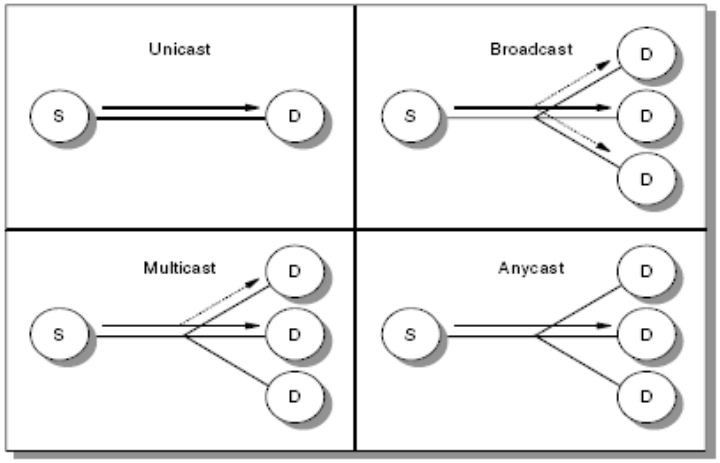


Gambar 4.4 Algoritma Routing

4.4. Metode Pengiriman – Unicast, Broadcast, Multicast dan Anycast

Pengiriman data pada IP address umumnya adalah 1 paket pengiriman, hal ini disebut *Unicast*. Koneksi unicast adalah koneksi dengan hubungan one-to-one antara 1 alamat pengirim dan 1 alamat penerima.

Untuk penerima dengan jumlah lebih dari 1 ada beberapa cara pengiriman yaitu broadcast, multicast dan anycast. Dapat dilihat pada



Gambar 4.5 Mode pengiriman data

4.4.1. Broadcast

Pengiriman data dengan tujuan semua alamat yang berada dalam 1 jaringan, mode pengiriman data seperti ini disebut *Broadcast*. Aplikasi yang menggunakan metode ini akan mengirimkan ke alamat broadcast. Contoh 192.168.0.255, apabila mengirimkan data ke alamat ini maka semua host yang berada dalam jaringan tersebut akan menerima data.

4.4.2. Multicast

Pengiriman data dengan tujuan alamat group dalam 1 jaringan, mode pengiriman data ini disebut *Multicast*. Alamat ini menggunakan kelas D, sehingga beberapa host akan didaftarkan dengan menggunakan alamat kelas D ini. Apabila ada pengirim yang mengirimkan data ke alamat kelas D ini akan diteruskan menuju ke host-host yang sudah terdaftar di IP kelas D ini.

4.4.3. Anycast

Apabila suatu pelayanan menggunakan beberapa IP address yang berbeda, kemudian apabila ada pengirim mengirimkan data menuju ke pelayanan tersebut maka akan diteruskan ke salah satu alamat IP tersebut, mode pengiriman ini disebut *Anycast*. Contoh: Apabila ada 5 server dengan aplikasi FTP yang sama, maka apabila ada user mengakses pelayanan FTP tersebut akan diarahkan ke salah satu dari 5 server tersebut.

4.5. IP Private - Intranet

Kebutuhan IP address beriringan dengan meningkatnya penggunaan internet. Karena jumlah IP address yang digunakan semakin lama semakin habis. Untuk mengatasi permasalahan ini dilakukan penggunaan IP Private.

IP Private ini diatur dalam RFC 1918 – Address alocation for Private Internets. RFC ini menjelaskan penggunaan IP address yang harus unik secara global. Dan penggunaan beberapa bagian dari IP address tersebut yang digunakan untuk tidak terhubung langsung ke internet. Alamat IP ini digunakan untuk jalur intranet. Alamat-alamat IP address tersebut adalah :

- 10.0.0.0 : digunakan untuk jaringan kelas A
- 172.16.0.0 – 172.31.0.0 : digunakan untuk jaringan kelas B
- 192.168.0.0 – 192.168.255.0 : digunakan untuk jaringan kelas C

Jaringan yang menggunakan alamat tersebut tidak akan diroutingkan dalam internet.

4.6. Classless Inter-Domain Routing (CIDR)

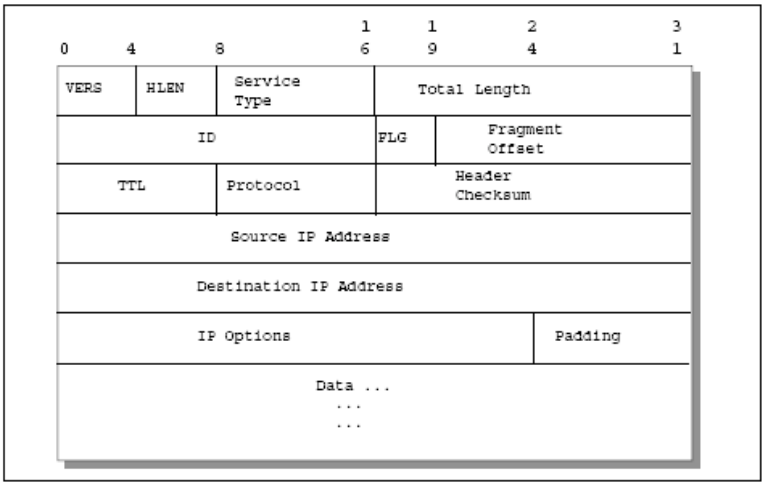
Apabila kita membutuhkan IP address dengan jumlah host 500 dengan kelas IP C, maka kita harus memiliki 2 subnet. Karena untuk kelas C maksimal host adalah 254. Untuk masing-masing subnet tersebut harus dimasukkan kedalam table routing pada perangkat router di jaringan tersebut.

Hal tersebut mengakibatkan jumlah entri dalam table routing akan semakin membengkak dan akan menguras sumber daya perangkat. Untuk mengatasi hal tersebut dapat digunakan Classless Inter-Domain Routing (CIDR). CIDR adalah routing yang tidak memperhatikan kelas dari alamat IP. CIDR dibahas pada RFC 1518 sampai 1520.

Contoh : Untuk mengkoneksikan 500 host dengan alamat IP kelas C diperlukan 2 subnet. IP address yang digunakan adalah 192.168.0.0/255.255.255.0 dengan 192.168.1.0/255.255.255.0, sehingga table routing pada perangkat router juga ada 2 subnet. Dengan menggunakan CIDR table routing pada perangkat cukup dengan menggunakan alamat 192.168.0.0/255.255.252.0 dengan ini hanya diperlukan 1 entri table routing untuk terkoneksi dengan jaringan tersebut.

4.7. IP Datagram

Unit yang dikirim dalam jaringan IP adalah *IP datagram*. Dimana didalamnya terdapat header dan data yang berhubungan dengan layer diatasnya.



Gambar 4.6 Format IP Datagram

Dimana :

- VERS : versi dari IP yang digunakan. Versi 4 artinya menggunakan IPv4, 6 artinya IPv6.
- HLEN : panjang dari IP header
- Service : no urut quality of service (QoS)
- Total Length : jumlah dari IP datagram
- ID : nomer data dari pengirim apabila terjadi fragmentasi
- Flags : penanda fragmentasi
- Fragment offset : no urut data fragmen bisa data telah di fragmentasi
- Time to Live (TTL) : lama waktu data boleh berada di jaringan, satuan detik
- Protocol : nomer dari jenis protokol yang digunakan

- Header checksum : digunakan untuk pengecekan apabila data rusak
- Source IP address : 32 bit alamat pengirim
- Destination IP Address : 32 bit alamat tujuan
- IP options : digunakan apabila data diperlukan pengolahan tambahan
- Padding : digunakan untuk membulatkan jumlah kolom IP options menjadi 32
- Data : data yang dikirimkan berikut header di layer atasnya.

4.7.1. Fragmentasi

Dalam perjalanannya menuju tujuan, data akan melewati berbagai macam interface yang berbeda. Dimana masing-masing interface memiliki kemampuan yang berbeda untuk mengirimkan frame data. Kemampuan ini disebut *Maximum Transfer Unit* (MTU). Batas maksimum data dapat ditempatkan dalam 1 frame.

IP dapat memisahkan data yang terkirim menjadi sebesar MTU. Proses pemisahan ini disebut fragmentasi (*fragmentation*).

Bab 5. Internetworking

5.1. Internet Control Message Protocol (ICMP)

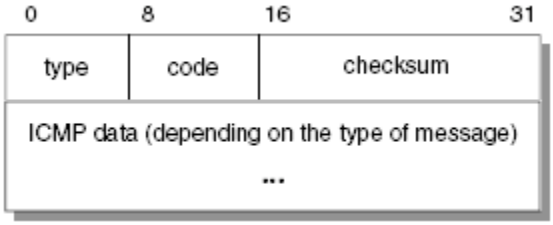
Ketika router atau host tujuan menginformasikan sesuatu kerusakan pada IP datagram, protokol yang digunakan adalah *Internet Control Message Protocol* (ICMP). Karakterisitk dari ICMP antara lain :

- ICMP menggunakan IP
- ICMP melaporkan kerusakan
- ICMP tidak dapat melaporkan kerusakan dengan menggunakan pesan ICMP, untuk menghindari pengulangan
- Untuk data yang terfragmentasi, pesan ICMP hanya mengirimkan pesan kerusakan pada fragmentasi pertama
- Pesan ICMP tidak merespon dengan mengirimkan data secara broadcast atau multicast
- ICMP tidak akan merespon kepada IP datagram yang tidak memiliki header IP pengirim
- Pesan ICMP dapat membuat proses kerusakan pada IP datagram

Spesifikasi ICMP dapat dilihat pada RFC 792 dengan update RFC 950.

5.1.1. Pesan ICMP

Pesan ICMP dikirimkan dalam IP Datagram. Pada IP header, protokol akan berisikan no 1 (ICMP). Dan type of service (TOS) bernilai 0 (routine). Format ICMP dapat dilihat pada Gambar 5.1.



Gambar 5.1 Format Pesan ICMP

Keterangan :

- Type : jenis pesan :
  - 0 : Echo reply
  - 3 : Destination Unreacheable
  - 4 : Source quench
  - 5 : Redirect
  - 8 : Echo
  - 9 : Router Advertisement
  - 10 : Router Solicitation
  - 11 : Time exceeded
  - 12 : Parameter problem
  - 13 : Timestamp request
  - 14 : Timestamp reply
  - 15 : Information request (kadaluwarsa)
  - 16 : Information reply (kadaluwarsa)

    - 17 : Address mask request
    - 18 : Address mask reply
    - 30 : Traceroute
    - 31 : Datagram conversion error
    - 32 : Mobile host redirect
    - 33 : IPv6 Where-are-you
    - 34 : IPv6 I-Am-Here
    - 35 : Mobile registration request
    - 36 : Mobile registration reply
    - 37 : Domain name request
    - 38 : Domain name reply
    - 39 : SKIP
    - 40 : Photuris
- Code : berisikan balasan laporan kerusakan dari pesan ICMP yang dikirim.
- Checksum : digunakan untuk pengecekan kerusakan pesan ICMP yang dikirim.
- Data : berisikan pesan ICMP yang dikirimkan.

Penjelasan tentang jenis pesan ICMP

5.1.1.1. Echo (8) dan Echo reply (0)

Echo digunakan untuk mengecek keaktifan dari suatu host. Dimana apabila host tersebut aktif akan dibales dengan pesan Echo Reply.

5.1.1.2. Destination Unreachable (3)

Pesan ini berasal dari suatu router dimana memberitahukan bahwa host tujuan tidak dapat dicapai (unreachable).

5.1.1.3. Source Quench (4)

Pesan ini berasal dari suatu router dimana router tidak memiliki ruang buffer untuk meneruskan datagram.

5.1.1.4. Redirect (5)

Pesan ini berasal dari router, dimana host tersebut harus mengirimkan datagram berikutnya kepada router yang berada pada jaringan yang dituju oleh pesan ICMP.

5.1.1.5. Router Advertisement (9) dan Router Solicitation (10)

Pesan ini digunakan oleh router yang mempunyai protokol discover. Dimana router akan memberikan IP address kepada jaringannya dan host yang menerima IP address tersebut akan membalas dengan pesan Router Solicitation.

5.1.1.6. Time Exceeded (11)

Pesan ini berasal dari router, dimana pesan yang dikirim sudah kehabisan waktu sesuai batas TTL.

5.1.1.7. Parameter Problem (12)

Pesan ini diakibatkan pada proses persiapan untuk mengirimkan pesan ICMP ada kesalahan.

5.1.1.8. Timestamp Request (13) dan Timestamp Reply (14)

Pesan ini digunakan untuk proses debug.

5.1.1.9. Information Request (15) dan Information Reply (16)

Pesan yang digunakan untuk mendapatkan IP address, pesan ini sudah digantikan oleh ARP dan RARP.

5.1.1.10. Address Mask Request (17) dan Address Mask Reply (18)

Pesan ini digunakan untuk mendapatkan subnetmask dari suatu jaringan.

5.1.2. Aplikasi ICMP

Contoh aplikasi yang menggunakan protokol ICMP antara lain adalah : PING dan TRACEROUTE.

5.1.2.1. PING

Ping adalah program tersederhana dari aplikasi TCP/IP. Ping mengirimkan IP datagram ke suatu host dan mengukur waktu round trip dan menerima respon. Ping merupakan singkat an dari *Packet InterNet Groper*.

Ping menggunakan pesan ICMP echo dan echo reply.

Ping dapat juga digunakan untuk memastikan installasi IP address di suatu host. Langkah-langkah yang dapat dilakukan yaitu :

- Ping loopback : test terhadap software TCP/IP
- Ping IP alamatku : test perangkat jaringan di host tersebut
- Ping alamat IP suatu host lain : test apakah jalur sudah benar
- Ping nama dari suatu host : test apakah sistem DNS sudah berjalan.

5.1.2.2. TRACEROUTE

Aplikasi traceroute melacak jalur mana saja yang dilalui untuk menuju ke suatu host tujuan.

Cara kerja traceroute dengan mengirimkan pesan dengan TTL = 1. Dimana apabila sudah mencapai suatu target jumlah TTL akan menjadi 0, dan ini akan memberikan pesan ke pengirim dengan pesan time exceeded, sehingga host akan mengirimkan lagi pesan ICMP dengan nilai TTL diperbesar. Proses ini dilakukan terus hingga mencapai host yang dituju.

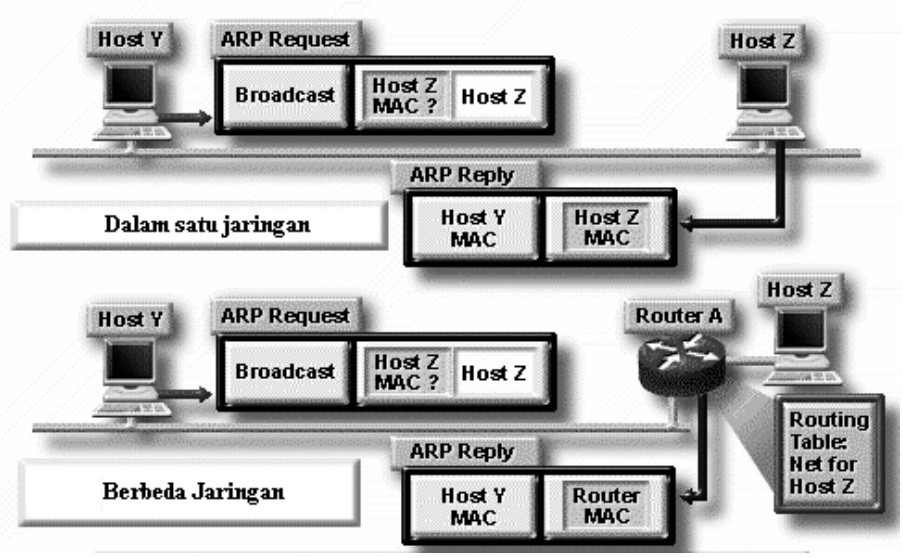
5.2. Internet Group Management Protocol (IGMP)

IGMP digunakan untuk mengecek apakah suatu host dapat bergabung dengan IP Multicast. Protokol IGMP memberikan fasilitas kepada router untuk melakukan cek kepada host yang tertarik untuk menggunakan sistem multicast.
Spesifikasi IGMP dapat dilihat pada RFC 1112 dengan update pada RFC 2236.

5.3. Address Resolution Protocol (ARP)

Protokol ARP digunakan untuk merubah protokol pengalamatan pada layer yang lebih atas (IP Address) menjadi alamat fisik jaringan.

Spesifikasi ARP dapat dilihat di RFC 826.



Gambar 5.2 Cara kerja protokol ARP

Cara kerja protokol ARP :
Host Y melakukan broadcast dengan mengirimkan pesan ARP Request, apabila host yang dituju berada dalam satu jaringan maka host tersebut akan mengirimkan pesat ARP Reply yang berisikan informasi MAC.
Bila host yang dituju berada dalam jaringan yang berbeda maka yang akan mengirimkan ARP Reply adalah Router yang memisahkan jaringan tersebut.

5.4. Reverse Address Resolution Protocol (RARP)

Protokol RARP digunakan untuk merubah protokol pengalatan pada layer yang lebih rendah (Alamat MAC) menjadi alamat IP.

5.5. Bootstrap Protocol (BOOTP)

Bootstrap Protocol (BOOTP) dapat membuat sebuah client / workstation untuk melakukan initialisasi (proses booting pada komputer) dengan IP Stack yang minimal sehingga mendapatkan IP Address, alamat Gateway, dan alamat Name server dari sebuah BOOTP server.

BOOTP spesifikasi bisa dilihat di RFC 951 – bootstrap protocol.

Proses BOOTP antara lain :

1. Client mendeteksi alamat fisik jaringan pada sistemnya sendiri, biasanya berada di ROM pada interface.
2. BOOTP celint mengirimkan informasi alamat fisik jaringannya ke server dengan menggunakan protokol UDP pada port 67.
3. Server menerima pesan dari client dan mencatat informasi alamat fisik client, kemudian membandingkan dengan data yang ada diserver. Apabila data yang dicari ada maka server akan memberikan IP address kepada client melalui port 68 protokol UDP.
4. Ketika client menerima reply dari server, client akan mencatat record alamat IP kemudian melakukan proses bootstrap.

5.6. Dynamic Host Configuration Protocol (DHCP)

DHCP memberikan framework untuk disampaikan kepada client yang berisikan informasi tentang konfigurasi jaringan. DHCP bekerja berdasarkan protokol BOOTP, dimana ditambahkan fungsi untuk mengalokasikan penggunaan IP address dan konfigurasi jaringan lainnya.

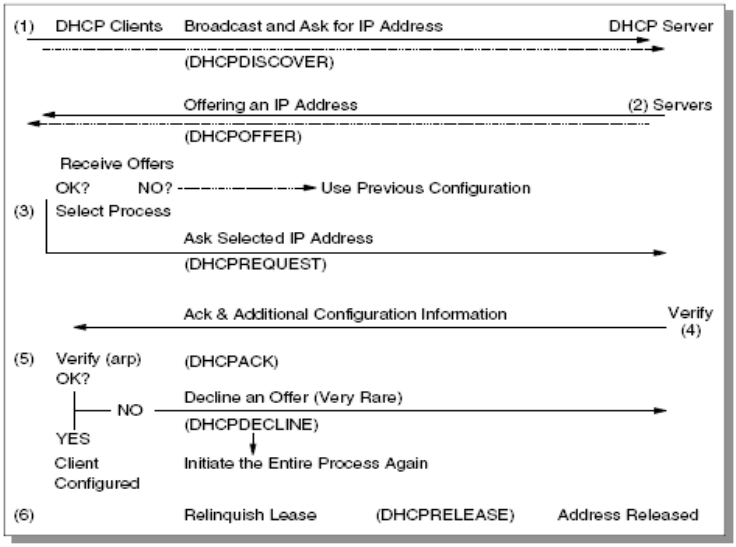
Spesifikasi DHCP dapat dilihat pada RFC 2131 – Dynamic Host Configuration Protocol, dan RFC 2132 – DHCP options and BOOTP vendor extension.

DHCP melakukan transaksi dengan melihat pada jenis pesan yang dikirimkan. Pesan-pesan tersebut antara lain :

- DHCPDISCOVER : broadcast oleh client untuk menemukan server
- DHCPOFFER : respon dari server karena menerima DHCPDISCOVER dan menawarkan IP address kepada client
- DHCPREQUEST : pesan dari client untuk mendapatkan informasi jaringan
- DHCPACK : acknowledge dari server
- DHCPNACK : negative acknowledge dari server yang menyatakan waktu sewa dari client sudah kadaluwarsa
- DHCPDECLINE : pesan dari client yang menyatakan bahwa dia sedang menggunakan informasi dari server
- DHCPRELEASE : pesan dari client bahwa client sudah tidak menggunakan lagi informasi dari server
- DHCPINFORM : pesan dari client bahwa dia sudah menggunakan informasi jaringan secara manual.

5.6.1. Proses alokasi alamat jaringan

Bagian ini menjelaskan interaksi antara client dan server, dimana client tidak mengetahui alamat IP nya. Di asumsikan DHCP server memiliki 1 blok alamat jaringan dimana dapat digunakan pada jaringan tersebut. Setiap server memiliki sebuah database yang berisikan info IP address dan sewa (leases) penggunaan jaringan pada suatu tempat penyimpanan yang permanen.



Gambar 5.3 Interaksi DHCP client dan DHCP server

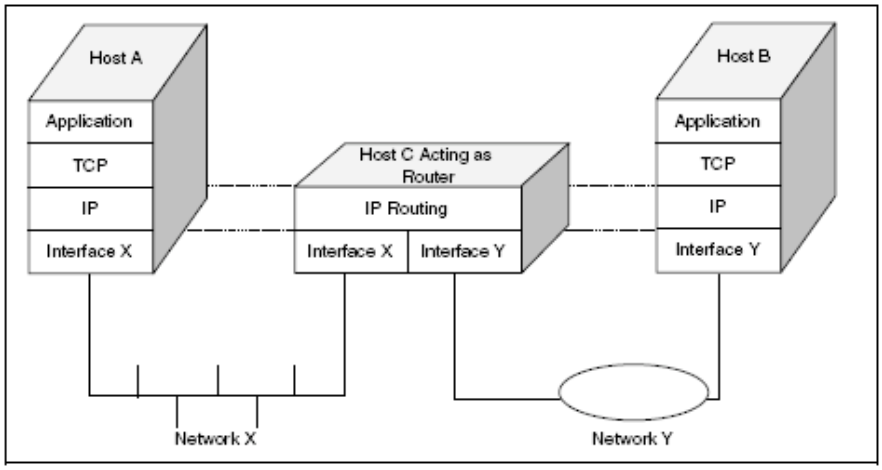
Berikut keterangan dari interaksi antara DHCP client dan DHCP server :

1. Client melakukan broadcast DHCPDISCOVER pada jaringan lokal.
2. Server merespon dengan pesan DHCPOFFER, dimana informasi ini juga memberikan informasi tentang IP address.
3. DHCP client menerima 1 atau lebih pesan DHCPOFFER dari 1 atau lebih DHCP server. Client memilih salah satu informasi itu dan mengirimkan pesan DHCPREQUEST dan informasi jaringan mana yang dipilih.
4. Server menerima pesan DHCPREQUEST tersebut dan membalas dengan mengirimkan pesan DHCPACK dengan mengirimkan informasi lengkap.
5. Client menerima DHCPACK dan melakukan konfigurasi terhadap interface jaringannya.
6. Apabila client sudah tidak menginginkan lagi alamat IP tersebut, client akan mengirimkan pesan DHCPRELEASE.

Bab 6. Protokol Routing

Salah satu fungsi dari protokol IP adalah membentuk koneksi dari berbagai macam bentuk interface yang berbeda. Sistem yang melakukan tugas tersebut disebut IP router. Tipe dari perangkat ini terpasang dua atau lebih bentuk interface dan meneruskan datagram antar jaringan.

Ketika mengirim data ke tujuan, suatu host akan melewati sebuah router terlebih dahulu. Kemudian router akan meneruskan data tersebut hingga tujuannya. Data tersebut mengalir dari router satu ke router yang lain hingga mencapai host tujuannya. Tiap router melakukan pemilihan jalan untuk menuju ke hop berikutnya.



Gambar 6.1 Operasi routing sebuah pada IP

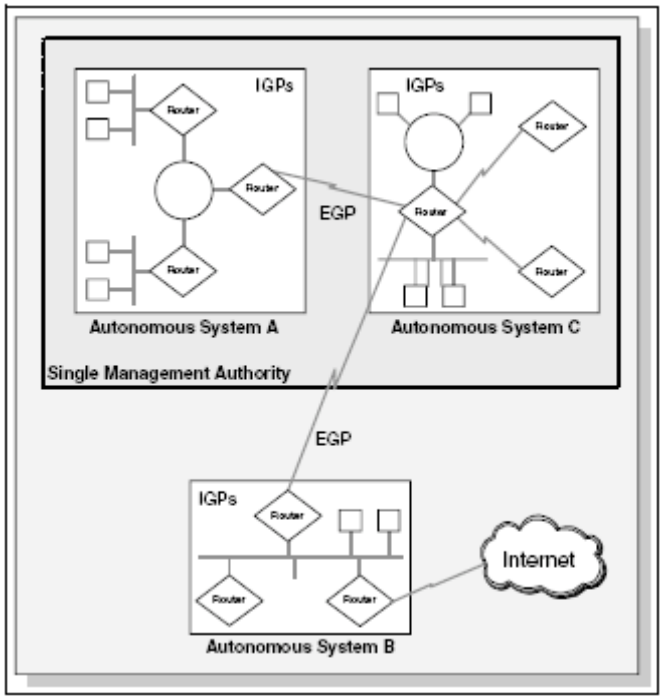
Gambar 6.1 menunjukkan sebuah jaringan dimana host C meneruskan paket data antara jaringan X dan jaringan Y

Routing table pada tiap perangkat digunakan untuk meneruskan paket data pada jaringan tiap segmen.

Protocol routing mempunyai kemampuan untuk membangun informasi dalam routing table secara dinamik. Apabila terjadi perubahan jaringan routing protokol mampu memperbaharui informasi routing tersebut.

6.1. Autonomous System

Definisi dari Autonomous System (AS) merupakan bagian dari memahami Routing Protocol. AS merupakan bagian logical dari Jaringan IP yang besar. AS biasanya dimiliki oleh sebuah organisasi jaringan. AS di administrasi oleh sebuah managemen resmi. AS dapat dikoneksikan dengan AS lainnya, baik public maupun private. Ilustrasi tentang AS dapat dilihat pada Gambar 6.2.



Gambar 6.2 Autonomous System

Beberapa routing protocol digunakan untuk menentukan jalur pada sistem AS. Yang lainnya digunakan untuk interkoneksi pada suatu set autonoumous system, yaitu :

- Interior Gateway Protocol (IGP) : dengan IGP router dapat saling tukar informasi routing antar AS. Contoh protokol ini antara lain Open Shortest Path First (OSPF) dan Routing Information Protocol (RIP).
- External Gateway Protocol (EGP) : dengan EGP router dapat saling tukar hasil akhir (*summary*) antar AS. Contoh protokol ini antara lain Border Gateway Protocol (BGP)

6.2. Tipe IP Routing dan Algoritma IP Routing

Algoritma routing digunakan untuk membangun dan mengatur table routing pada perangkat. Terdapat 2 cara untuk membangun table routing, yaitu :

- Static Routing : routing ini dibangun berdasarkan definisi dari adminstrator.
- Dynamic Routing : algoritma ini dapat membuat perangkat router untuk dapat menentukan jalur routingnya secara otomatis, dengan cara menjelajah jaringan tersebut dan bertukar informari routing antar router. Terdapat 3 kategori tentang algoritma dinamik, yaitu :
    - Distance Vector
    - Link State
    - Hybrid

6.2.1. Static Routing

Routing static adalah entri suatu route yang dilakukan oleh seorang administrator untuk mengatur jalur dari sebuah paket data. Entri routing table bisa dilakukan dengan program yang terdapat pada perangkat tersebut.

6.2.2. Distance Vector Routing

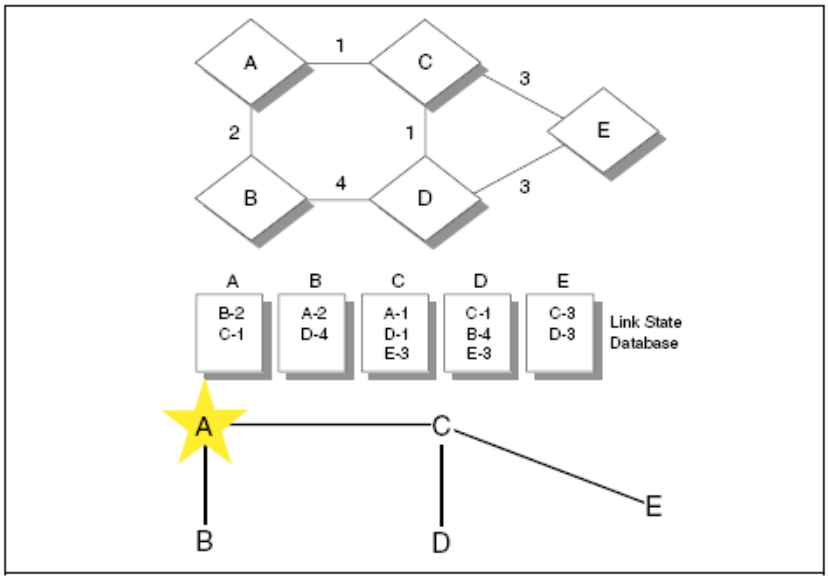
Routing ini menggunakan algoritma Bellman-Ford. Dimana tiap router pada jaringan memiliki informasi jalur mana yang terpendek untuk menghubungi segmen berikutnya. Kemudian antar router akan saling mengirimkan informasi tersebut, dan akhirnya jalur yang lebih pendek akan lebih sering dipilih untuk menjadi jalur menuju ke host tujuan.

Protokol yang menggunakan algoritma ini yaitu RIP.

6.2.3. Link State Routing

Routing ini menggunakan teknik link state, dimana artinya tiap router akan mengolek informasi tentang interface, bandwidth, roundtrip dan sebagainya. Kemudian antar router akan saling menukar informasi, nilai yang paling efisien yang akan diambil sebagai jalur dan di entri ke dalam table routing. Informasi state yang ditukarkan disebut *Link State Advertisement* (LSA).

Dengan menggunakan algoritma pengambilan keputusan *Shortest Path First* (SPF), informasi LSA tersebut akan diatur sedemikian rupa hingga membentu suatu jalur routing. Ilustrasi SPF dapat dilihat pada Gambar 6.3.



Gambar 6.3 Shortest Path First

Routing protokol yang menggunakan algoritma antara lain OSPF.

6.2.4. Hybrid Routing

Routing merupakan gabungan dari Distance Vector dan Link State routing. Contoh penggunaan algoritma ini adalah EIGRP.

6.3. Routing Information Protocol (RIP)

Routing protokol yang menggunakan algoritma distance vector, yaitu algortima Bellman-Ford. Pertama kali dikenalkan pada tahun 1969 dan merupakan algoritma routing yang pertama pada ARPANET.

Versi awal dari routing protokol ini dibuat oleh Xerox Parc's PARC Universal Packet Internetworking dengan nama Gateway Internet Protocol. Kemudian diganti nama menjadi Router Information Protocol (RIP) yang merupakan bagian Xerox network Services.

Versi dari RIP yang mendukung teknologi IP dimasukkan dalam BSD system sebagai routed daemon.

Spesifikasi RIP dapat dilihat di RFC 1058.

RIP yang merupakan routing protokol dengan algoritma distance vector, yang menghitung jumlah hop (count hop) sebagai routing metric. Jumlah maksimum dari hop yang diperbolehkan adalah 15 hop. Tiap RIP router saling tukar informasi routing tiap 30 detik, melalui UDP port 520. Untuk menghindari loop routing, digunakan teknik *split horizon with poison reverse*. RIP merupakan routing protocol yang paling mudah untuk di konfigurasi.

RIP memiliki 3 versi yaitu RIPv1, RIPv2, RIPng

- RIPv1 didefinisikan pada RFC 1058, dimana menggunakan classful routing, tidak menggunakan subnet. Tidak mendukung Variable Length Subnet Mask (VLSM).
- RIPv2 hadir sekitar tahun 1994, dengan memperbaiki kemampuan akan Classless Inter-Domain Routing. Didefinisikan pada RFC 2453.
- RIPng merupakan protokol RIP untuk IPv6. Didefinisikan pada RFC 2080.

6.4. Open Shortest Path First (OSPF)

OSPF merupakan routing protocol berbasis link state, termasuk dalam interior Gateway Protocol (IGP). Menggunakan algoritma Dijkstra untuk menghitung shortest path first (SPF). Menggunakan cost sebagai routing metric. Setelah antar router bertukar informasi maka akan terbentuk database link state pada masing-masing router.

OSPF mungkin merupakan IGP yang paling banyak digunakan. Menggunakan metode MD5 untuk autentikasi antar router sebelum menerima Link-state Advertisement (LSA). Dari awak OSPF sudah mendukung CIDR dan VLSM, berbeda dengan RIP. Bahkan untuk OSPFv3 sudah mendukung untuk IPv6.

Router dalam broadcast domain yang sama akan melakukan *adjacencies* untuk mendeteksi satu sama lainnya. Pendeteksian dilakukan dengan mendengarkan "Hello Packet". Hal ini disebut 2 way state. Router OSPF mengirimkan "Hello Packet" dengan cara unicast dan multicast. Alamat multicast 224.0.0.5 dan 224.0.0.6 digunakan OSPF, sehingga OSPF tidak menggunakan TCP atau UDP melainkan IP protocol 89.

6.5. Enhanced Interior Gateway Routing Protocol (EIGRP)

EIGRP merupakan routing protocol yang dibuat CISCO. EIGRP termasuk routing protocol dengan algoritma hybrid.

EIGRP menggunakan beberapa terminologi, yaitu :

- Successor : istilah yang digunakan untuk jalur yang digunakan untuk meneruskan paket data.
- Feasible Successor : istilah yang digunakan untuk jalur yang akan digunakan untuk meneruskan data apabila successor mengalami kerusakan.

- Neighbor table : istilah yang digunakan untuk tabel yang berisi alamat dan interface untuk mengakses ke router sebelah
- Topology table : istilah yang digunakan untuk tabel yang berisi semua tujuan dari router sekitarnya.
- Reliable transport protocol : EIGRP dapat menjamin urutan pengiriman data.

Perangkat EIGRP bertukar informasi hello packet untuk memastikan daerah sekitar. Pada bandwidth yang besar router saling bertukar informasi setiap 5 detik, dan 60 detik pada bandwidth yang lebih rendah.

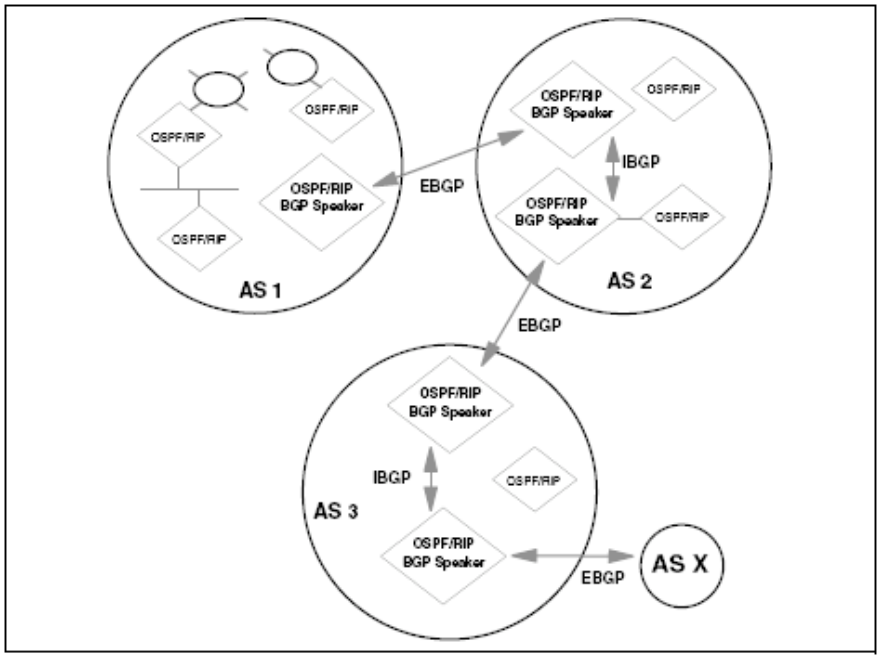
6.6. Border Gateway Protocol (BGP)

BGP adalah router untuk jaringan external. BGP digunakan untuk menghidari routing loop pada jaringan internet.

Standar BGP menggunakan RFC 1771 yang berisi tentang BGP versi 4.

6.6.1. Konsep dan terminologi BGP

Konsep dan terminologi dapat dilihat pada Gambar 6.4.



Gambar 6.4 Komponen BGP

- BGP Speaker : Router yang mendukung BGP
- BGP Neighbor (pasangan) : Sepasang router BGP yang saling tukar informasi. Ada 2 jenis tipe tetangga (neighbor) :
    - Internal (IBGP) neighbor : pasangan BGP yang menggunakan AS yang sama.
    - External (EBGP) neighbor : pasangan BGP yang menggunakan AS yang berbeda.
- BGP session : sesi dari 2 BGP yang sedang terkoneksi
- Tipe traffik :
    - Lokal : trafik lokal ke AS
    - Transit : semua trafik yang bukan lokal
- Tipe AS :

    - Stub : bagian AS yang terkoneksi hanya 1 koneksi dengan AS.
    - Multihomed : bagian ini terkoneksi dengan 2 atau lebih AS, tetapi tidak meneruskan trafik transit.
    - Transit : bagian ini terkoneksi dengan 2 atau lebih AS, dan meneruskan paket lokal dan transit
- Nomer AS : 16 bit nomer yang unik
- AS path : jalur yang dilalui oleh routing dengan nomer AS
- Routing Policy : aturan yang harus dipatuhi tentang bagaimana meneruskan paket.
- Network Layer Reachability Information (NLRI) : digunakan untuk advertise router.
- Routes dan Path : entri tabel routing

6.6.2. Operasional BGP

BGP neighbor, peer, melakukan koneksi sesuai dengan konfigurasi manual pada perangkat router dan membuat jalur TCP dengan port 179. BGP speaker akan mengirimkan 19 byte pesan *keepalive* untuk menjaga konektivitas (dilakukan tiap 60 detik).

Pada waktu BGP berjalan pada dalam sistem AS, melakukan pengolahan informasi routing IBGP hingga mencapai *administrative distance* 200. Ketika BGP berjalan diantara sistem AS, maka akan melakukan pengolahan informasi routing EBGP hingga mencapai *administrative distance* 20. BGP router yang mengolah trafik IBGP disebut transit router. Router yang berada pada sisi luar dari sistem AS dan menggunakan EBGP akan saling tukar informasi dengan router ISP.

Semakin bertambahnya jaringan akan mengakibatkan jumlah table routing yang semakin banyak pada router BGP. Untuk mengatasi hal tersebut dapat dilakukan : *route reflector* (RFC 2796) dan *Confederation* (RFC 3065).

Router reflector akan mengurangi jumlah koneksi yang dibutuhkan AS. Dengan sebuah router (atau dua router untuk redundansi) dapat dijadikan sebagai router reflector (duplikasi router), sehingga router yang lainnya dapat digunakan sebagai peer.

Confederation digunakan untuk jaringan AS dengan skala besar, dan dapat membuat jalan potong sehingga internal routing pada AS akan mudah di manaj. Confederation dapat dijalankan bersamaan dengan router reflector.

6.7. Proses Routing di sistem UNIX

Selain dengan menggunakan mesin Router. Routing protocol juga dapat dijalankan pada sistem UNIX dengan bantuan beberapa aplikasi antara lain :

- RouteD : mendukung interior routing dengan mengimplementasikan RIP
- GateD : mendukung interior dan eksterior routing dengan mengimplementasikan OSPF, RIPv2, BGP-4
- Quagga : mendukung interior dan ekterior routing dengan mengimplementasikan OSPFv3, RIPv1, RIPv2, RIPng, BGP4

Bab 7. Transport Layer

Pada bab ini akan dijelaskan tentang fungsi dari 2 protokol penting pada layer transport, yaitu :

- User Datagram Protocol (UDP)
- Transmission Control Protocol (TCP)

7.1. Port dan Socket

7.1.1. Port

Port digunakan untuk melakukan proses komunikasi dengan proses lain pada jaringan TCP/IP. Port menggunakan nomer 16 bit, digunakan untuk komunikasi host-to-host. Tipe port ada 2 macam yaitu :

- Well-known : port yang sudah dimiliki oleh server. Contoh : telnet menggunakan port 23. Well-known port memiliki range dari 1 hingga 1023. Port Well-known diatur oleh Internet Assigned Number Authority (IANA) dan dapat digunakan oleh proses sistem dengan user tertentu yang mendapatkan akses.
- Ephemeral : client tidak menggunakan port well-known karena untuk berkomunikasi dengan server, mereka sudah melakukan perjanjian terlebih dahulu untuk menggunakan port mana. Ephemeral port memiliki range dari 1023 hingga 65535.

Untuk 1 nomer port tidak bisa digunakan oleh 2 aplikasi yang berbeda dalam waktu yang bersamaan.

7.1.2. Socket

Interface socket merupakan bagian dari *Application Programming Interface* (API) yang digunakan untuk protokol komunikasi.

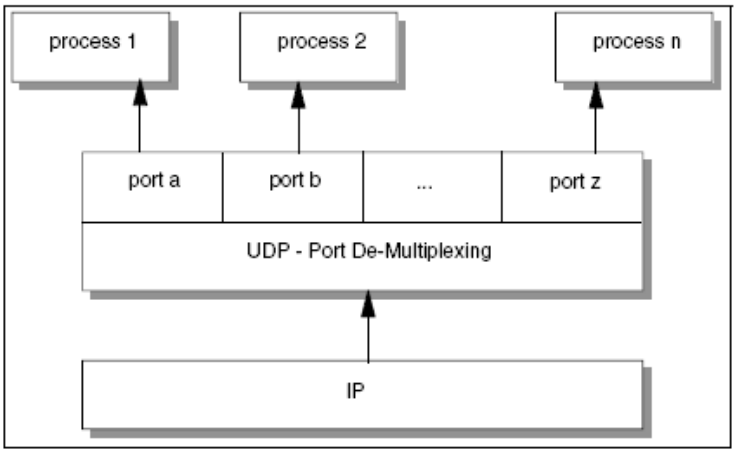
Terminologi yang digunakan :

- Socket merupakan tipe spesial dari *file handle*, dimana digunakan oleh sistem operasi untuk mengakses jaringan.
- Alamat soket adalah : <protocol, local address, local process> contoh : <tcp, 193.44.234.3, 12345>
- Pembicaraan (*conversation*) : link komunikasi antar 2 proses
- Asosiasi (*Association*) : kejadian komunikasi antar 2 proses <protocol, local-address, local-process, foreign-address, foreign-process>
    - Contoh : <tcp, 193.44.234.4, 1500, 193.44.234.5, 21>
- Setengah Asosiasi (*half-association*) : < protocol, local-address, local-process> atau <protocol, foreign-address, foreign-process>
- Half-association disebut juga transport address.

7.2. User Datagram Protocol (UDP)

UDP merupakan standar protokol dengan STD nomer 6. Spesifikasi UDP dapat dilihat pada RFC 768 – User Datagram Protocol.

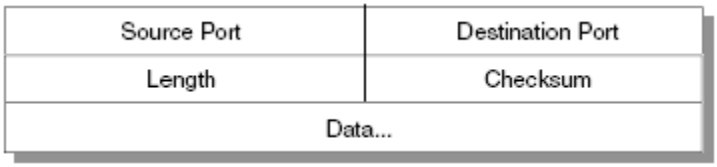
UDP pada dasarnya adalah interface untuk aplikasi IP. Dimana UDP tidak memiliki fungis reliabilitas data, flow control, dan error-recovery untuk komunikasi IP. UDP memiliki proses seperti multiplexing/demultiplexing untuk mengirimkan datagram, dari port menuju IP datagram. Karena itu UDP juga disebut sebagai connectionless-oriented protocol.



Gambar 7.1 Proses Demultiplexing berbasis port pada UDP

7.2.1. Format Datagram UDP

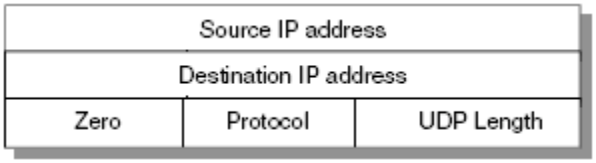
UDP datagram memiliki 16 byte seperti pada Gambar 7.2.



Gambar 7.2 Format Datagram UDP

Dimana :

- Source Port : port yang digunakan untuk mengirimkan data.
- Destination Port : port yang digunakan untuk tujuan data.
- Length : panjang data paket keseluruhan
- Checksum : 16 bit komplemen-1 dari pseudo-ip-header yang merupakan error check dari paket data



Gambar 7.3 Pseudo IP Header – UDP

7.2.2. Aplikasi yang menggunakan UDP

Aplikasi yang menggunkan protokol UDP antara lain :

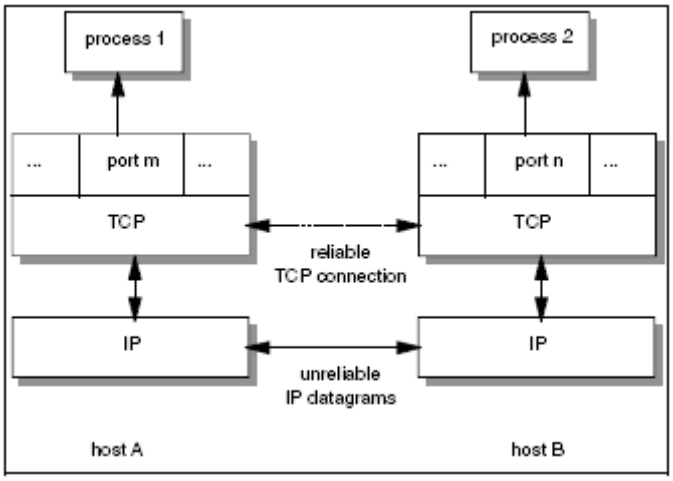
- Trivial File Transfer Protocol (TFTP)
- Domain Name System (DNS) name server
- Remote Procedure Call (RPC) pada Network File System (NFS)
- Simple Network Management Protocol (SNMP)
- Lighweight Directory Access Protocol (LDAP)

7.3. Transmission Control Protocol (TCP)

TCP merupakan standar protokol dengan STD nomer 7. Spesifikasi TCP dapat dilihat pada RFC 793 – Transmission Control Protocol.

TCP memberikan fasilitas untuk aplikasi dibandingkan UDP, karena TCP memberikan error recovery, flow control, dan reliabilitas. TCP biasa disebut juga sebagai protokol berbasis connection-oriented.

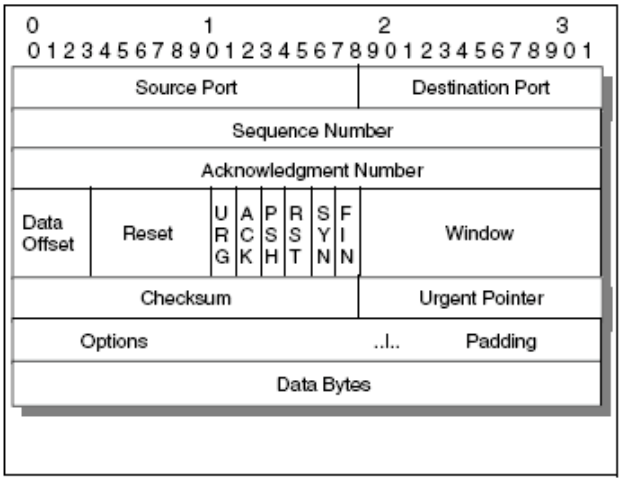
2 Proses komunikasi menggunakan koneksi TCP disebut InterProcess Communication (IPC). IPC diilustrasikan seperti pada Gambar 7.4.



Gambar 7.4 IPC

7.3.1. Format Segmen TCP

Format TCP dapat dilihat pada Gambar 7.5.



Gambar 7.5 Format TCP

Dimana :

- Source Port : 16 bit nomer port. Digunakan untuk menerima reply
- Destination port : 16 bit nomer port tujuan
- Sequence Number : nomwer awal data pada segmen
- Acknowledge number : apabila ACK diset maka ini menjadi nomer urut data yang akan diterima
- Data offset : nomer dimana bagian data mulai

- Reserved : untuk kegunaan masa depan, diset 0
- URG : mengaktifkan titik yang darurat pada suatu segmen
- ACK : kolom acknowledge
- PSH : fungsi push
- RST : mereset suatu koneksi
- SYN : untuk mensinkronisasi nomer urutan
- FIN : batas akhir data
- Window : nomer window untuk proses windowing
- Checksum : nomer yang digunakan untuk mengecek validitas pengirim dan penerima
- Urgent Pointer : menunjuk pada titik yang darurat pada suatu segmen
- Options : digunakna untuk pilihan lain pada datagram
- Padding : digunakan untuk membulatkan data pada bagian options

7.3.2. Interface Pemrograman pada aplikasi TCP

Fungsi yang digunakan pada komunikasi TCP antara lain :

- Open : membuka koneksi dengan memasukkan beberapa parameter antara lain :
    - Actif / Pasif
    - Informasi soket tujuan
    - Nomer port lokal
    - Nilai timeout
- Send : mengirimkan buffer data ke tujuan
- Receive : Menerima dan mengcopy data kepada buffer milik pengguna
- Close : menutup koneksi
- Status : melihat informasi
- Abort : membatalkan semua kegiatan send atau receive

7.3.3. Aplikasi yang menggunakan TCP

Hampir keseluruhan aplikasi jaringan menggunakan TCP, standar aplikasi yang menggunakan TCP antara lain :

- Telnet
- File Transfer Protocol (FTP)
- Simple Mail Transfer Protocol (SMTP)
- Hyper-Text Transfer Protocol (HTTP)

Bab 8. Struktur dan Pemrograman untuk Layer Aplikasi

Layer tertinggi adalah layer aplikasi. Layer ini saling berkomunikasi antar host dan merupakan interface yang tampak oleh user pada protokol TCP/IP

8.1. Karakteristik dari Aplikasi

Pada layer aplikasi terdapat beberapa karakterisik yang sama yaitu :

- Merupakan aplikasi yang ditulis oleh user (*user-written*) atau aplikasi sudah merupakan standar dengan didalamnya sudah terdapat produk TCP/IP. Aplikasi TCP/IP set yang terdapat antara lain :
  - TELNET, digunakan untuk mengakses remote host melalui terminal yang interaktif
  - FTP (File Transfer Protocol) digunakan untuk transfer file antar disk
  - SMTP (Simple Mail Transfer Protocol) digunakan sebagai sistem surat di internet
- Menggunakan sistem transpor UDP atau TCP
- Menggunakan model client-server

8.2. Pemrograman dengan Socket API

Application Programming Interface (API) dapat digunakan oleh user untuk dapat membuat suatu aplikasi. Sedangkan untuk fasilitas jaringannya dapat menggunakan API bagian *SOCKET*. Dalam bagian ini akan dijelaskan contoh API yang digunakan untuk jaringan.

Pada bab ini dijelaskan bagaimana menggunakan bahasa pemrograman C untuk kepentingan jaringan pada mesin linux.

Contoh kompilasi dan menjalankan program :
gcc –o program source.c
./program

8.2.1. Struktur dan Penanganan Data

Sebelum menggunakan pemrograman socket diperlukan suatu variable struktur untuk menyimpan informasi tentang jaringan. Struktur yang diperlukan antara lain :

- sockaddr
- sockaddr_in

Contoh penggunaannya yaitu :

```
struct sockaddr {
       unsigned short sa_family; // address family, AF_xxx
       char sa_data[14]; // 14 bytes of protocol address
};
```

Dimana, sa_family digunakan untuk penentuan jenis family yang digunakan pada bab ini menggunakan AF_INET artinya menggunakan family INTERNETWORKING. Sedangkan untuk sa_data digunakan untuk informasi tujuan dan port yang digunakan.

Untuk menggunakan struktur tersebut diperlukan 1 lagi struktur yaitu sockadd_in dimana arti "in" adalah internet

```
struct sockaddr_in {
       short int sin_family; // Address family
       unsigned short int sin_port; // Port number
       struct in_addr sin_addr; // Internet address
       unsigned char sin_zero[8]; // Same size as struct sockaddr
};
```

Dengan struktur ini maka programmer akan dengan mudah mengontrol data. Pada bagian `sin_zero` digunakan sebagai pelengkap dimana harus diset dengan nilai 0, hal tersebut dapat digunakan fungsi `memset()`.

Untuk menggunakan alamat IP perlu juga sebuah variabel struktur yaitu struktur *in_addr*, dimana struktur *in_addr* adalah sebagai berikut :

```
// Internet address (a structure for historical reasons)
struct in_addr {
       unsigned long s_addr; // that's a 32-bit long, or 4 bytes
};
```

Sehingga untuk penggunanya dapat dilakukan dengan cara, membuat sebuah variable contoh *ina* dan bertipe *struct sockaddr_in* maka *ina.sin_addr.s_addr* dapat digunakan sebagai objek untuk alamat IP.

8.2.1.1. Perubahan variable

Perubahan awal yang dapat digunakan adalah perubahan dari short (2 byte) menjadi long (4 byte). Kemudian perubahan lainnya adalah perubahan dari *host* menjadi *network*. Sehingga masing-masing perubahan bisa disingkat menjadi 1 huruf yaitu , s, l, n, dan h.

Fungsi yang dapat digunakan untuk perubahan tersebut antara lain :

- htons() : perubahan host ke network dengan sistem short
- htonl() : perubahan host ke network dengan sistem long
- ntohs() : perubahan network ke host dengan sistem short
- ntohl() : perubahan network ke host dengan sistem long

8.2.1.2. Penanganan alamat IP

Ada beberapa cara untuk memasukkan alamat IP kedalam suatu variable pada pemrograman socket.

Apabila kita sudah memiliki variable *struct sockadd_in ina*, dan kita memiliki alamat IP "10.252.102.23". Maka dengan fungsi *inet_addr()*, akan dapat merubah alamat IP menjadi *unsigned long*. Contoh penggunaan :

```
ina.sin_addr.s_addr = inet_addr("10.252.102.23");
```

selain itu ada cara yang lainnya, yaitu dengan menggunakan inet_aton :

```
#include <sys/socket.h>
#include <netinet/in.h>
```

```
#include <arpa/inet.h>

int inet_aton(const char *cp, struct in_addr *inp);
```

Dan contoh penggunaannya adalah sebagai berikut :

```
struct sockaddr_in my_addr;

my_addr.sin_family = AF_INET; // host byte order
my_addr.sin_port = htons(MYPORT); // short, network byte order
inet_aton("10.252.102.53", &(my_addr.sin_addr));
memset(&(my_addr.sin_zero), '\0', 8); // zero the rest of the struct
```

Sehingga apabila kita ingin menampilkan isi variabel tersebut dapat dilakukan dengan fungsi tambahan *inet_ntoa* (network to ascii).

```
printf("%s", inet_ntoa(ina.sin_addr));
```

Contoh lengkapnya :

```
char *a1, *a2;
.
.
a1 = inet_ntoa(ina1.sin_addr); // this is 192.168.4.14
a2 = inet_ntoa(ina2.sin_addr); // this is 10.12.110.57
printf("address 1: %s\n",a1);
printf("address 2: %s\n",a2);
```

akan menghasilkan

```
address 1: 10.12.110.57
address 2: 10.12.110.57
```

8.2.2. System Call

System call adalah fungsi-fungsi dalam pemrograman. Fungsi-fungsi tersebut digunakan untuk menjalankan dan mengakses jaringan.

8.2.2.1. socket()

Penggunaan :

```
#include <sys/types.h>
#include <sys/socket.h>

int socket(int domain, int type, int protocol);
```

Fungsi ini digunakan untuk inisialisasi dalam penggunaan socket. Dimana *domain* berisikan AF_INET, sedangkat *type* berisikan SOCK_STREAM atau SOCK_DGRAM dan *protocol* berisikan angka 0.

SOCK_STREAM digunakan apabila menggunakan protokol TCP dan SOCK_DGRAM digunakan untuk protokol UDP.

Selain isi diatas, masih banyak lagi lainnya dan bisa dilihat pada *manual page*.

8.2.2.2. bind()

Penggunaan :

```
#include <sys/types.h>
#include <sys/socket.h>

int bind(int sockfd, struct sockaddr *my_addr, int addrlen);
```

fungsi *bind* digunakan untuk melakukan asosiasi terhadap alamat IP dan port. Variabel *sockfd* didapat dari fungsi *socket()*.

Contoh :

```
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

#define MYPORT 3490

main()
{
        int sockfd;
        struct sockaddr_in my_addr;

        sockfd = socket(AF_INET, SOCK_STREAM, 0); // do some error checking!

        my_addr.sin_family = AF_INET; // host byte order
        my_addr.sin_port = htons(MYPORT); // short, network byte order
        my_addr.sin_addr.s_addr = inet_addr("10.12.110.57");
        memset(&(my_addr.sin_zero), '\0', 8); // zero the rest of the struct

        // don't forget your error checking for bind():
        bind(sockfd, (struct sockaddr *)&my_addr, sizeof(struct sockaddr));
        .
        .
        .
```

Apabila ingin menggunakan alamat IP mesin kita, dapat digunakan :

```
my_addr.sin_port = 0; // choose an unused port at random
my_addr.sin_addr.s_addr = INADDR_ANY; // use my IP address
```

8.2.2.3. connect()

Penggunaan :

```
#include <sys/types.h>
#include <sys/socket.h>

int connect(int sockfd, struct sockaddr *serv_addr, int addrlen);
```

fungsi *connect* digunakan untuk mengakses suatu remote host.

Contoh :

```
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
```

```
#define DEST_IP "10.12.110.57"
#define DEST_PORT 23

main()
{
        int sockfd;
        struct sockaddr_in dest_addr; // will hold the destination addr

        sockfd = socket(AF_INET, SOCK_STREAM, 0); // do some error checking!

        dest_addr.sin_family = AF_INET; // host byte order
        dest_addr.sin_port = htons(DEST_PORT); // short, network byte order
        dest_addr.sin_addr.s_addr = inet_addr(DEST_IP);
        memset(&(dest_addr.sin_zero), '\0', 8); // zero the rest of the struct

        // don't forget to error check the connect()!
        connect(sockfd, (struct sockaddr *)&dest_addr, sizeof(struct sockaddr));
        .
        .
        .
```

8.2.2.4. listen()

Penggunaan :

```
int listen(int sockfd, int backlog);
```

Fungsi dari perintah *listen* digunakan untuk menunggu koneksi dari suatu host.

8.2.2.5. accept()

Penggunaan :

```
#include <sys/socket.h>

int accept(int sockfd, void *addr, int *addrlen);
```

Fungsi dari accept digunakan setelah fungsi listen. Dimana socket akan meneruskan ke variable socket yang baru setelah suatu host menghubungi. Accept akan membentuk socket baru dan bisa diproses untuk send atau recv.

Contoh :

```
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

#define MYPORT 3490 // the port users will be connecting to

#define BACKLOG 10 // how many pending connections queue will hold

main()
{
        int sockfd, new_fd; // listen on sock_fd, new connection on new_fd
        struct sockaddr_in my_addr; // my address information
        struct sockaddr_in their_addr; // connector's address information
        int sin_size;

        sockfd = socket(AF_INET, SOCK_STREAM, 0); // do some error checking!

        my_addr.sin_family = AF_INET; // host byte order
        my_addr.sin_port = htons(MYPORT); // short, network byte order
        my_addr.sin_addr.s_addr = INADDR_ANY; // auto-fill with my IP
```

```
memset(&(my_addr.sin_zero), '\0', 8); // zero the rest of the struct

// don't forget your error checking for these calls:
bind(sockfd, (struct sockaddr *)&my_addr, sizeof(struct sockaddr));

listen(sockfd, BACKLOG);
sin_size = sizeof(struct sockaddr_in);

new_fd = accept(sockfd, (struct sockaddr *)&their_addr, &sin_size);
.
.
.
```

8.2.2.6. send() dan recv()

Penggunaan :

```
int send(int sockfd, const void *msg, int len, int flags);

int recv(int sockfd, void *buf, int len, unsigned int flags);
```

Funsi dari send dan recv adalah untuk pertukaran data. Fungsi *send()* dan *recv()* digunakan untuk data dengan protokol yang berbasis connection-oriented, sedangkan untuk protokol yang berbasis connectionless-oriented menggunakan *sendto()* dan *recvfrom()*.

Pointer **msg* merupakan isi dari data yang akan dikirim, begitu juga dengan **buf* merupakan pointer yang berisi data yang diterima. Variabel *len* digunakan sebagai panjang data tersebut.

Contoh :

```
char *msg = "Beej was here!";
int len, bytes_sent;
.
.
len = strlen(msg);
bytes_sent = send(sockfd, msg, len, 0);
.
.
.
```

8.2.2.7. sendto() dan recvfrom()

Penggunaan :

```
int sendto(int sockfd, const void *msg, int len, unsigned int flags, const struct
sockaddr *to, int tolen);

int recvfrom(int sockfd, void *buf, int len, unsigned int flags, struct sockaddr
*from, int *fromlen);
```

Fungsi dari sendto dan recvfrom adalah untuk pertukaran data dengan protokol DGRAM. Fungsi tersebut hampir sama dengan fungsi send dan recv dimana terdapat variabel tambahan yaitu *struct sockaddr *to*, dan *int toleni.*

8.2.2.8. close() dan shutdown()

Penggunaan :

```
close(sockfd);
```

```
int shutdown(int sockfd, int how);
```

Fungsi *close()* dan *shutdown()* digunakan untuk menutup koneksi setelah melakukan pertukaran data. Shutdown digunakan apabila diinginkan suatu kondisi tertentu, variabel tersebut ditambahkan pada variable *how*. Varibel tersebut mempunya nilai dan arti tertentu yaitu :

- 0 – Setelah ditutup, hanya diperbolehkan menerima
- 1 – Setelah ditutup, hanya diperbolehkan mengirim
- 2 – Seteleh ditutup, menerima dan mengirim tidak diperbolehkan (sama dengan *close()*)

8.2.2.9. getpeername()

Penggunaan :

```
#include <sys/socket.h>

int getpeername(int sockfd, struct sockaddr *addr, int *addrlen);
```

Fungsi *getpeername()* digunakan untuk mengetahui informasi tentang tujuan.

8.2.2.10. gethostname()

Penggunaan :

```
#include <unistd.h>

int gethostname(char *hostname, size_t size);
```

Fungsi *gethostname()* digunakan untuk mengetahui informasi tentang mesin jaringan kita.

8.2.2.11. DNS – Mengirim ke “whitehouse.gov”, Dijawab “198.137.240.92”

Penggunaan :

```
#include <netdb.h>

struct hostent *gethostbyname(const char *name);
```

Struktur hostent memiliki objek didalam antara lain :

```
struct hostent {
        char *h_name;
        char **h_aliases;
        int h_addrtype;
        int h_length;
        char **h_addr_list;
};
#define h_addr h_addr_list[0]
```

Dimana :

- h_name – nama resmi dari suatu host
- h_aliases – NULL , nama alternatif dari suatu host
- h_addrtype – type dari alamat, contoh AF_INET
- h_length – panjang dari data alamat IP
- h_addr_list – ZERO, sekumpulan IP dengan nama tersebut

- h_addr – alamat pertama dari h_addr_list

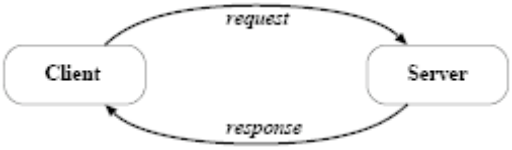
Untuk mendapatkan hasil dari struktur hostent digunakan fungsi *gethostbyname()*. Cara penggunaan dapat dilihat pada contoh program.

Contoh program :

```
/*
** getip.c - a hostname lookup demo
*/

#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <netdb.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>

int main(int argc, char *argv[])
{
        struct hostent *h;
        if (argc != 2) { // error check the command line
                fprintf(stderr,"usage: getip address\n");
                exit(1);
        }

        if ((h=gethostbyname(argv[1])) == NULL) { // get the host info
                herror("gethostbyname");
                exit(1);
        }

        printf("Host name : %s\n", h->h_name);
        printf("IP Address : %s\n", inet_ntoa(*((struct in_addr *)h->h_addr)));
        return 0;
}
```

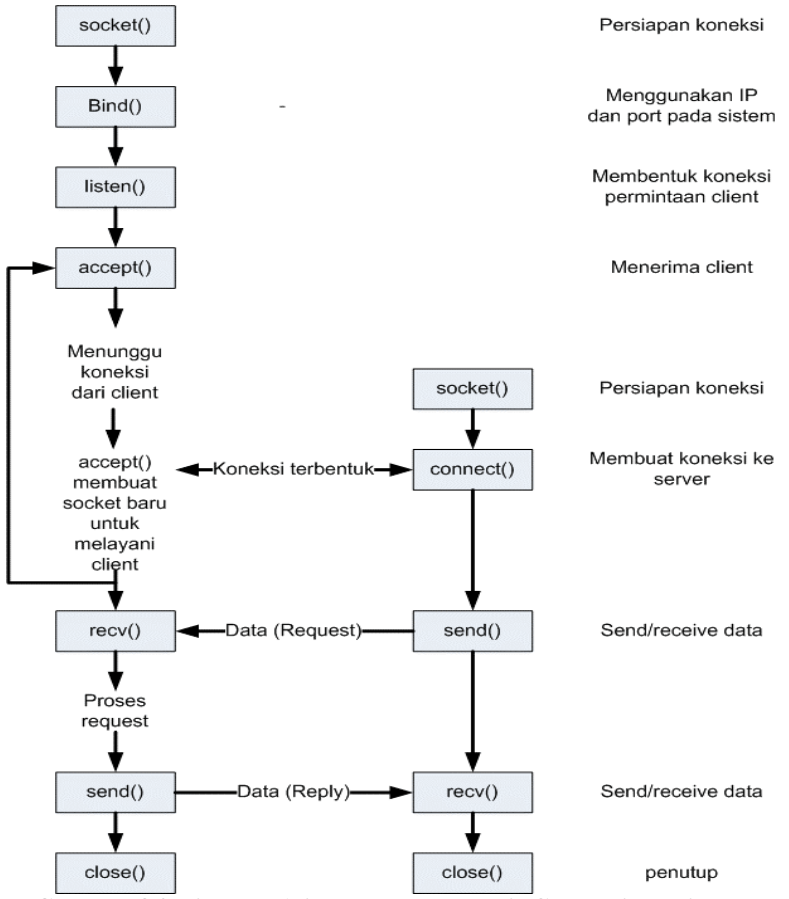
8.2.3. Skenario penggunaan pemrograman socket

Pemrograman socket menggunakan sistem client-server, dimana proses client berbicara dengan proses server dan sebaliknya. Contoh, client dengan aplikasi **telnet** akan menghubungi server yang menjalankan aplikasi **telnetd**.

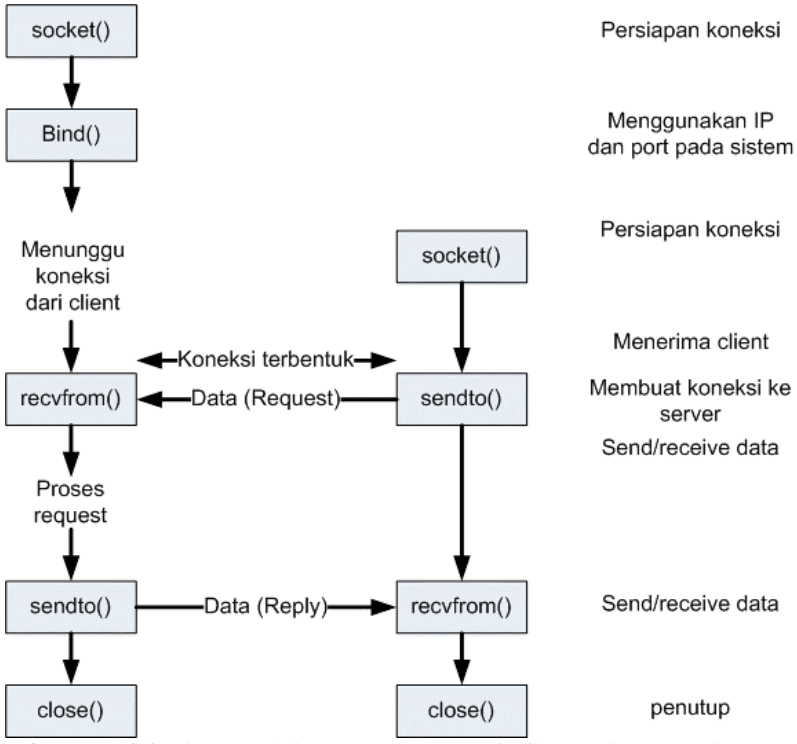


Gambar 8.1 Client-Server

Diagram alir yang digunakan tampak pada



Gambar 8.2 Diagram Alir Program Berbasis Connection-oriented



Gambar 8.3 Diagram Alir Program Berbasis Connectionless-oriented

8.2.3.1. Contoh Server Berbasis STREAM

```
/*
** server.c - a stream socket server demo
*/

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <sys/wait.h>
#include <signal.h>

#define MYPORT 3490 // the port users will be connecting to
#define BACKLOG 10 // how many pending connections queue will hold

void sigchld_handler(int s)
{
       while(wait(NULL) > 0);
}

int main(void)
{
       int sockfd, new_fd; // listen on sock_fd, new connection on new_fd
       struct sockaddr_in my_addr; // my address information
       struct sockaddr_in their_addr; // connector’s address information
       int sin_size;
       struct sigaction sa;
       int yes=1;

       if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
              perror("socket");
              exit(1);
       }

       if (setsockopt(sockfd,SOL_SOCKET,SO_REUSEADDR,&yes,sizeof(int)) == -1) {
              perror("setsockopt");
              exit(1);
       }

       my_addr.sin_family = AF_INET; // host byte order
       my_addr.sin_port = htons(MYPORT); // short, network byte order
       my_addr.sin_addr.s_addr = INADDR_ANY; // automatically fill with my IP
       memset(&(my_addr.sin_zero), ’\0’, 8); // zero the rest of the struct

       if (bind(sockfd, (struct sockaddr *)&my_addr, sizeof(struct sockaddr))== -1)
{
              perror("bind");
              exit(1);
       }

       if (listen(sockfd, BACKLOG) == -1) {
              perror("listen");
              exit(1);
       }

       sa.sa_handler = sigchld_handler; // reap all dead processes
       sigemptyset(&sa.sa_mask);
       sa.sa_flags = SA_RESTART;
       if (sigaction(SIGCHLD, &sa, NULL) == -1) {
              perror("sigaction");
              exit(1);
       }
```

```
        while(1) { // main accept() loop
                sin_size = sizeof(struct sockaddr_in);
                if ((new_fd = accept(sockfd, (struct sockaddr *)&their_addr,
        &sin_size)) == -1) {
                        perror("accept");
                        continue;
                }

                printf("server: got connection from %s\n",
                inet_ntoa(their_addr.sin_addr));
                if (!fork()) { // this is the child process
                        close(sockfd); // child doesn't need the listener
                        if (send(new_fd, "Hello, world!\n", 14, 0) == -1)
                                perror("send");
                        close(new_fd);
                        exit(0);
                }
                close(new_fd); // parent doesn't need this
        }
        return 0;
}
```

Untuk mencoba program server tersebut jalankan:

telnet server 3490

8.2.3.2. Contoh Client Berbasis STREAM

```
/*
** client.c - a stream socket client demo
*/
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <netdb.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>

#define PORT 3490 // the port client will be connecting to
#define MAXDATASIZE 100 // max number of bytes we can get at once

int main(int argc, char *argv[])
{
        int sockfd, numbytes;
        char buf[MAXDATASIZE];
        struct hostent *he;
        struct sockaddr_in their_addr; // connector's address information

        if (argc != 2) {
                fprintf(stderr,"usage: client hostname\n");
                exit(1);
        }

        if ((he=gethostbyname(argv[1])) == NULL) { // get the host info
                perror("gethostbyname");
                exit(1);
        }

        if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
                perror("socket");
                exit(1);
        }

        their_addr.sin_family = AF_INET; // host byte order
        their_addr.sin_port = htons(PORT); // short, network byte order
```

```
        their_addr.sin_addr = *((struct in_addr *)he->h_addr);
        memset(&(their_addr.sin_zero), '\0', 8); // zero the rest of the struct

        if (connect(sockfd, (struct sockaddr *)&their_addr,sizeof(struct sockaddr))
== -1) {
                perror("connect");
                exit(1);
        }

        if ((numbytes=recv(sockfd, buf, MAXDATASIZE-1, 0)) == -1) {
                perror("recv");
                exit(1);
        }

        buf[numbytes] = '\0';
        printf("Received: %s",buf);

        close(sockfd);
        return 0;
}
```

Program ini mencari server dengan port 3490 dan menerima string dari server dan menampilkan ke layar.

8.2.3.3. Socket dengan DATAGRAM

Program **listener** akan bersiap pada sebuah mesin dan akan menunggu paket yang menuju ke port 4950. Program **talker** akan mengirim paket menuju ke port tersebut.

Listing program listernet :

```
/*
** listener.c - a datagram sockets "server" demo
*/
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>

#define MYPORT 4950 // the port users will be connecting to
#define MAXBUFLEN 100

int main(void)
{
        int sockfd;
        struct sockaddr_in my_addr; // my address information
        struct sockaddr_in their_addr; // connector's address information
        int addr_len, numbytes;
        char buf[MAXBUFLEN];

        if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
                perror("socket");
                exit(1);
        }

        my_addr.sin_family = AF_INET; // host byte order
        my_addr.sin_port = htons(MYPORT); // short, network byte order
        my_addr.sin_addr.s_addr = INADDR_ANY; // automatically fill with my IP
        memset(&(my_addr.sin_zero), '\0', 8); // zero the rest of the struct

        if (bind(sockfd, (struct sockaddr *)&my_addr,sizeof(struct sockaddr)) == -1)
{
```

```
                perror("bind");
                exit(1);
        }

        addr_len = sizeof(struct sockaddr);
        if ((numbytes=recvfrom(sockfd,buf, MAXBUFLEN-1, 0,(struct sockaddr
*)&their_addr, &addr_len)) == -1) {
                perror("recvfrom");
                exit(1);
        }

        printf("got packet from %s\n",inet_ntoa(their_addr.sin_addr));
        printf("packet is %d bytes long\n",numbytes);

        buf[numbytes] = '\0';
        printf("packet contains \"%s\"\n",buf);
        close(sockfd);

        return 0;
}
```

Listing program talker:

```
/*
** talker.c - a datagram "client" demo
*/
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>

#define MYPORT 4950 // the port users will be connecting to

int main(int argc, char *argv[])
{
        int sockfd;
        struct sockaddr_in their_addr; // connector's address information
        struct hostent *he;
        int numbytes;

        if (argc != 3) {
                fprintf(stderr,"usage: talker hostname message\n");
                exit(1);
        }

        if ((he=gethostbyname(argv[1])) == NULL) { // get the host info
                perror("gethostbyname");
                exit(1);
        }
        if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
                perror("socket");
                exit(1);
        }

        their_addr.sin_family = AF_INET; // host byte order
        their_addr.sin_port = htons(MYPORT); // short, network byte order
        their_addr.sin_addr = *((struct in_addr *)he->h_addr);
        memset(&(their_addr.sin_zero), '\0', 8); // zero the rest of the struct

        if ((numbytes=sendto(sockfd, argv[2], strlen(argv[2]), 0,(struct sockaddr
*)&their_addr, sizeof(struct sockaddr))) == -1) {
                perror("sendto");
                exit(1);
        }
```

```
        printf("sent %d bytes to %s\n", numbytes,
        inet_ntoa(their_addr.sin_addr));

        close(sockfd);
        return 0;
}
```

8.2.4. Socket lanjutan

Pada bagian ini dijelaskan tentang penggunaan beberapa fungsi yang dapat mendukung kerja dari program jaringan menggunakan pemrograman socket.

8.2.4.1. Blocking

Suatu aplikasi server dapat menerima paket data secara bersamaan, untuk itu perlu dilakukan pelepasan suatu pembatas atau yang disebut non-blocking. Sehingga server bisa menerima data secara bersamaan.

Pada initialisasi socket(), socket secara awal memiliki nilai awal blocking. Untuk membuat mejadi bersifat non-blocking dilakukan dengan cara memanggil fungsi *fcntl()*. Hal ini dapat dilihat pada contoh berikut :

```
#include <unistd.h>
#include <fcntl.h>
.
.
sockfd = socket(AF_INET, SOCK_STREAM, 0);
fcntl(sockfd, F_SETFL, O_NONBLOCK);
.
.
```

8.2.4.2. select() – Synchronous I/O Multiplexing

Dengan fungsi select, aplikasi akan dapat memilah dan memroses data pada waktu yang bersamaan. Contoh penggunaan select()

```
#include <sys/time.h>
#include <sys/types.h>
#include <unistd.h>

int select(int numfds, fd_set *readfds, fd_set *writefds, fd_set *exceptfds, struct
timeval *timeout);
```

Untuk memperjelas berikut adalah contoh program dimana akan menunggu dalam 2.5 detik apakah ada data yang masuk dari inputan keyboard.

```
/*
** select.c - a select() demo
*/
#include <stdio.h>
#include <sys/time.h>
#include <sys/types.h>
#include <unistd.h>

#define STDIN 0 // file descriptor for standard input

int main(void)
{
        struct timeval tv;
        fd_set readfds;
```

```
        tv.tv_sec = 2;
        tv.tv_usec = 500000;

        FD_ZERO(&readfds);
        FD_SET(STDIN, &readfds);

        // don't care about writefds and exceptfds:
        select(STDIN+1, &readfds, NULL, NULL, &tv);

        if (FD_ISSET(STDIN, &readfds))
                printf("A key was pressed!\n");
        else
                printf("Timed out.\n");

        return 0;
}
```

Contoh penggunaan select() pada alikasi multiperson chat server

```
/*
** selectserver.c - a cheezy multiperson chat server
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>

#define PORT 9034 // port we're listening on

int main(void)
{
  fd_set master; // master file descriptor list
  fd_set read_fds; // temp file descriptor list for select()
  struct sockaddr_in myaddr; // server address
  struct sockaddr_in remoteaddr; // client address
  int fdmax; // maximum file descriptor number
  int listener; // listening socket descriptor
  int newfd; // newly accept()ed socket descriptor
  char buf[256]; // buffer for client data
  int nbytes;
  int yes=1; // for setsockopt() SO_REUSEADDR, below
  int addrlen;
  int i, j;

  FD_ZERO(&master); // clear the master and temp sets
  FD_ZERO(&read_fds);

  // get the listener
  if ((listener = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
    perror("socket");
    exit(1);
  }

  // lose the pesky "address already in use" error message
  if (setsockopt(listener, SOL_SOCKET, SO_REUSEADDR, &yes, sizeof(int)) == -1) {
    perror("setsockopt");
    exit(1);
  }

  // bind
  myaddr.sin_family = AF_INET;
  myaddr.sin_addr.s_addr = INADDR_ANY;
  myaddr.sin_port = htons(PORT);
```

```
  memset(&(myaddr.sin_zero), '\0', 8);

  if (bind(listener, (struct sockaddr *)&myaddr, sizeof(myaddr)) == -1) {
    perror("bind");
    exit(1);
  }

  // listen
  if (listen(listener, 10) == -1) {
    perror("listen");
    exit(1);
  }

  // add the listener to the master set
  FD_SET(listener, &master);

  // keep track of the biggest file descriptor
  fdmax = listener; // so far, it's this one

  // main loop
  for(;;) {
    read_fds = master; // copy it
      if (select(fdmax+1, &read_fds, NULL, NULL, NULL) == -1) {
        perror("select");
        exit(1);
      }

      // run through the existing connections looking for data to read
      for(i = 0; i <= fdmax; i++) {
        if (FD_ISSET(i, &read_fds)) { // we got one!!
          if (i == listener) {
            // handle new connections
            addrlen = sizeof(remoteaddr);
            if ((newfd = accept(listener, (struct sockaddr *)&remoteaddr, &addrlen))
== -1) {
              perror("accept");
            } else {
              FD_SET(newfd, &master); // add to master set
              if (newfd > fdmax) { // keep track of the maximum
                fdmax = newfd;
              }

              printf("selectserver: new connection from %s on socket %d\n",
inet_ntoa(remoteaddr.sin_addr), newfd);
            }
          } else {
            // handle data from a client
             if ((nbytes = recv(i, buf, sizeof(buf), 0)) <= 0) {
               // got error or connection closed by client
               if (nbytes == 0) {
                 // connection closed
                 printf("selectserver: socket %d hung up\n", i);
               } else {
                 perror("recv");
               }
               close(i); // bye!

               FD_CLR(i, &master); // remove from master set
             } else {
               // we got some data from a client
               for(j = 0; j <= fdmax; j++) {
                 // send to everyone!
                 if (FD_ISSET(j, &master)) {
                   // except the listener and ourselves
                   if (j != listener && j != i) {
                     if (send(j, buf, nbytes, 0) == -1) {
                       perror("send");
                     }
                   }
```

```
                }
              }
            }
          } // it's SO UGLY!
        }
      }
    }

    return 0;
}
```

8.3. Remote Procedure Call (RPC)

RPC adalah suatu protokol yang memperbolehkan suatu program komputer yang memberikan suatu subroutin kepada komputer yang lain untuk menjalankan suatu perintah tanpa melalui programmner membuat program terlebih dahulu.

Bab 9. Protokol Penamaan dan Direktori

Protokol TCP/IP memiliki banyak jenis aplikasi, tetapi semuanya itu merupakan bentuk dari utilitas jaringan. Semuanya itu menjadi penting dalam suatu perusahaan untuk menggunakan jaringan. Jaringan ada untuk diakses dan melayani pengguna, baik dari dalam maupun dari luar. Dibutuhkan server untuk melayani aplikasi, data dan sumber lainnya. Server tersebut dimungkinkan dapat berjalan di aneka macam perangkat keras, dari berbagai macam vendor dan juga berbagai macam jenis sistem operasi. Pada bab ini akan dijelaskan metode untuk pengaksesan suatu sumber dan aplikasi pada jaringan terdistribusi.

9.1. Domain Name System (DNS)

DNS dijelaskan pada standar protocol dengan no STD 13. Dan dijelaskan pada RFC 1034, dan RFC 1035.

Pada awal internet, seorang pengguna hanya bisa mengakses internet dengan menggunakan alamat IP. Sehingga pengguna harus dapat menghafalkan berbagai macam alamat IP seperti layaknya menghafalkan no telp. Contoh untuk mengakses suatu server, pengguna harus tahu alamat IP dari server tersebut, dengan cara TELNET 202.154.187.5. Kemudian dikembangkan suatu sistem penamaan sehingga pengguna cukup mengakses internet dengan sebuah nama unik, contoh TELNET www. Dimana IP 202.154.187.5 dipetakan dengan nama www.

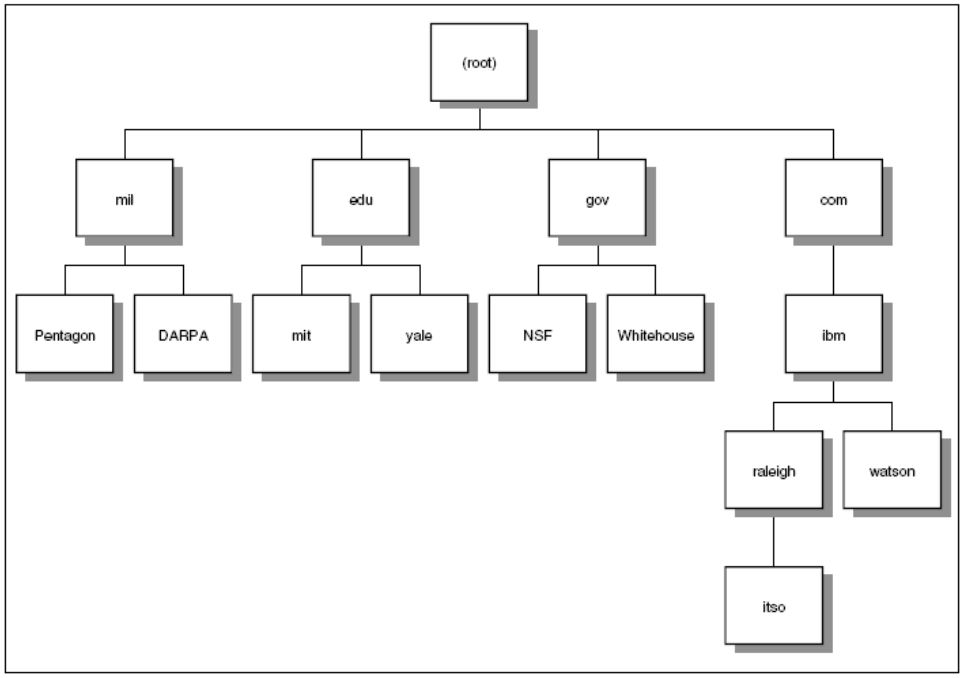
Karena perkembangan internet sangat cepat, maka dikembangkan sistem Domain Name System (DNS). Dimana cukup dengan sebuah host yang melakukan pemetaan suatu nama terhadap IP, sehingga host lain cukup mengakses host tersebut dan menanyakan suatu nama dan dibalaskan alamat IP kepada host penanya. Sehingga host penanya tidak perlu memiliki database pemetaan tersebut.

9.1.1. Hirarki Penamaan

Penamaan suatu domain dibentuk dalam suatu bentuk pohon hirarki. Dimana hal ini mempermudah untuk pengontrolan suatu nama domain. Contoh :

```
small.itso.raleigh.ibm.com
```

Small merupakan nama dari host, itso.raleigh.ibm.com merupakan nama domain dengan level terendah, dan merupakan subdomain dari raleigh.ibm.com, dan juga merupakan subdomain dari ibm.com, dan juga merupakan subdomain dari domain com yang juga merupakan top-level domain. Hal tersebut terlihat seperti pada Gambar 9.1.



Gambar 9.1 DNS – Hirarki Penamaan

9.1.2. Fully Qualified Domain Names (FQDN)

Ketika menggunakan DNS, pengguna dapat mengakses suatu site hanya dengan bagian kecil dari suatu domain. Semisal untuk mengakses website resmi kampus dari jaringan LAN kampus, pengguna cukup mengetikkan www. Padahal nama lengkap dari server tersebut adalah www.eepis-its.edu. Nama www.eepis-its.edu merupakan FQDN.

9.1.3. Domain generik

Tiga karakter dari top-level domain disebut juga domain generik atau domain organisasional. Tabel 9.1 menunjukkan contoh dari Top-Level Domain.

Tabel 9.1 Top-Level Domain

Nama Domain	Arti
com	Organisasi komersial (company)
edu	Institusi edukasi atau pendidikan
gov	Institusi pemerintahan
int	Organisasi internasional
mil	Militer AS
net	Pusat layanan jaringan
org	Organisasi non-profit
Kode-negara	2 digit kode negara

Dikarenakan internet berawal di Amerika Serikat, kebanyakan top-level domain merupakan milik dari badan di AS. Namun pada saat ini hanya gov dan mil yang dikhususkan digunakan di AS.

9.1.4. Domain Negara

Tiap negara memiliki domain sendiri dengan menggunakan 2 karakter huruf yang merupakan singkatan dari nama negaranya. Karakter yang digunakan sesuai dengan ISO 3166. Contoh: Indonesia menggunakan domain .id.

9.1.5. Pemetaan Nama Domain ke Alamat IP

Yang mengontrol pemetaan nama adalah *nameserver*. Nameserver adalah sebuah program server dimana memegang master atau duplikat dari database pemetaan nama ke alamat IP. Fungsi nameserver adalah menjawab permintaan dari program client tentang suatu nama domain. Nama program client disebut *name resolver*.

9.1.6. Pemetaan Alamat IP ke Nama Domain – pointer query

Untuk pemetaan alamat IP ke nama domain tidak berbentuk hirarki melainkan dalam format domain in.addr-arpa (ARPA digunakan karena internet berawal dari ARPAnet).

Penggunaan in.addr-arpa adalah pemetaan terbalik dari suatu alamat IP. Contoh: IP dengan alamat 129.34.139.30, pada database ditulis dengan 30.139.34.129.in-addr.arpa. Kemudian dicari nama host yang cocok. Sistem ini disebut *pointer query*.

9.1.7. Pendistribusian Nama Domain

Pengaturan nama suatu domain dapat dilakukan di jaringan lokal, hal ini disebabkan cara kerja DNS menggunakan sistem *zones of authority* atau yang biasa disingkat *zones*. Dimana dengan sistem zones ini suatu nameserver dapat mendelegasikan suatu nama domain ke nameserver lainnya yang terhubung melalui internetworking.

Pada nameserver root, nameserver mendelegasikan suatu domain ke suatu nameserver. Contoh : domain eepis-its.edu, dimana nameserver .edu di eduacause.net mendelegasikan nama eepis-its ke nameserver di jaringan kampus PENS. Nama domain eepis-its didelegasikan ke nameserver ns1.eepis-its.edu (202.154.187.2) dan ns2.eepis-its.edu (202.154.187.3). Dan pada nameserver ns1 dan ns2 dicatat nama-nama host dari jaringan eepis-its.edu.

9.1.8. Domain Name Resolution

Proses yang dilakukan pada penanyaan nama domain antara lain :

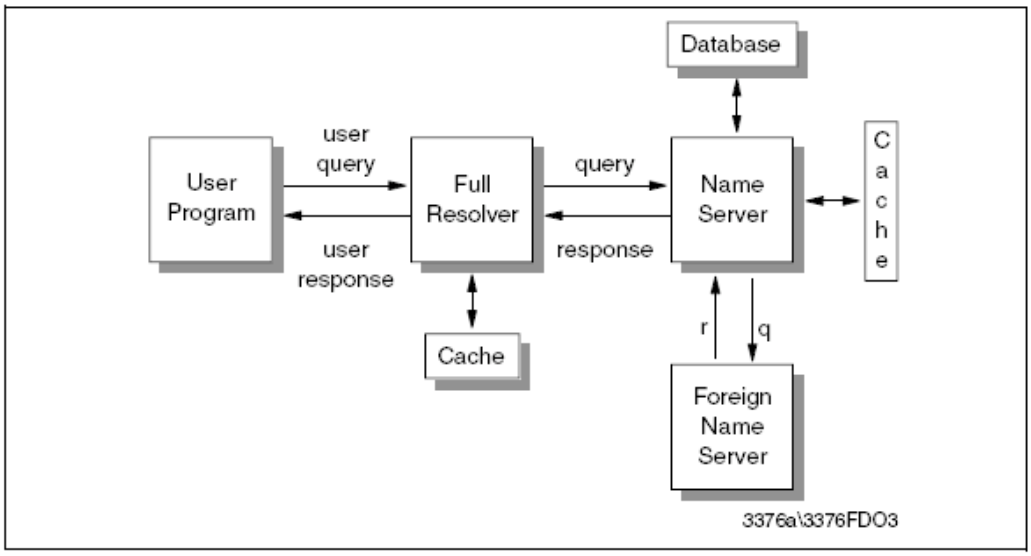
1. Suatu program menggunakan gethostbyname().
2. Resolver menanyakan ke suatu nameserver
3. Nameserver mengecek apakah ada jawaban di database lokal atau di penyimpanan sementara (cache). Apabila tidak diketemukan nameserver akan meneruskan ke nameserver lainnya sesuai dengan hirarki nama domain.
4. Program pada pengguna menerima jawaban berupa alamat IP atau pesan error jika terjadi kesalahan.

Proses diatas disebut Domain Name Resolution, yang merupakan aplikasi berbasis server-client. Fungsi client dilakukan oleh resolver secara transparan terhadap pengguna. Sedangkan fungsi server dilakukan oleh Nameserver.

Pengiriman ini menggunakan jalur UDP dan TCP.

9.1.9. Domain name full resolver

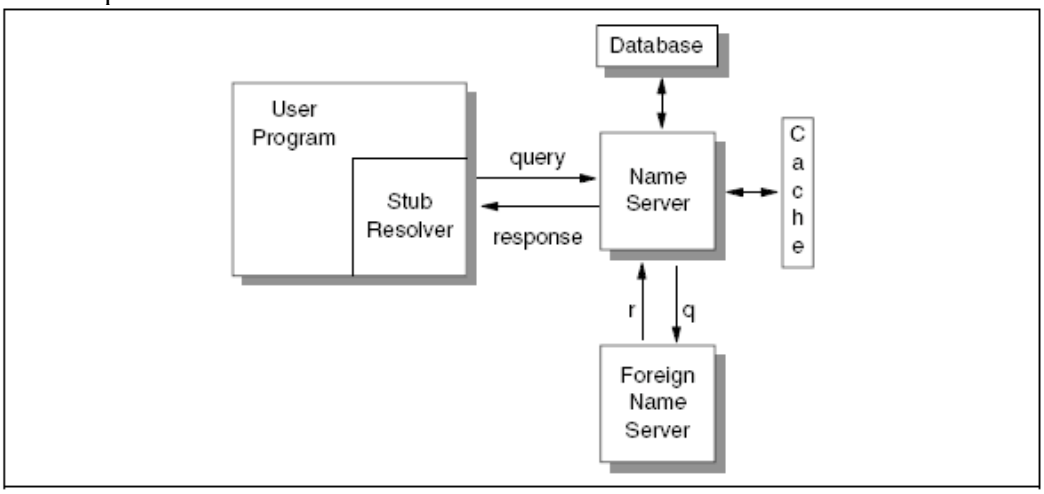
Dikatakan full resolver apabila dilakukan DNS resolution dari program pengguna, dan di query ke suatu nameserver dari program resolver untuk di proses. Sistem full resolver terlihat pada Gambar 9.2.



Gambar 9.2 DNS – menggunakan full resolver untuk domain name resolution

9.1.10. Domain name stub resolver

Sebuah program yang dilengkapi dengan subrutin pemrosesan nama domain dan dapat melakukan query ke nameserver disebut domain name stub resolver. Dimana pada UNIX, stub resolver dilakukan dengan subrutin gethostbyname() dan gethostbyaddr(). Stub resolver dapat dilihat pada



Gambar 9.3 DNS – menggunakan stub resolver untuk domain name resolution

9.1.11. Operasi Domain Name Server

Tipe dari nameserver antara lain :

Primary Nameserver menggunakan zones dari disk dan memiliki autorisasi terhadap keseluruhan zone

Secondary Nameserver ini memiliki autorisasi terhadap keseluruhan zone tapi data zone diambil dari nameserver primary dengan menggunakan proses *zones*

transfer.

Caching-only Sebuah nameserver yang tidak memiliki autorisasi dan data zone. Tetapi hanya melakukan penerusan query ke suatu nameserver yang sudah dicatat

9.1.12. Resource Record dari Domain Name System

Database dari DNS disebut dengan resource record (RR), dimana didalamnya dimulai dengan Start of Authority (SOA), dimana SOA mencatat nama dari domain. Kemudian ada penunjukan nameserver (NS) yang akan menjawab nama dari domain tersebut.

Format resource record :

Tabel 9.2 Format Resource Record dari DNS

Nama	TTL	Class	Tipe	RData

Dimana :

- Nama : nama dari domain
- TTL : Time-to-live, lama waktu suatu nama akan berada dalam cache. Satuan yang digunakan detik, contoh 86400 adalah 1 hari.
- Class : mengidentifikasikan nama protokol, contoh IN (sistem Internet)
- Tipe : mengidentifikasi tipe dari resource record

Tabel 9.3 Tipe dari RR

Tipe	Nilai	Arti
A	1	Alamat host
CNAME	5	Canonical Name, nama alias dari suatu host
HINFO	13	CPU dan OS yang digunakan suatu host, bersifat komentar
MX	15	Mail Exchange untuk suatu domain
NS	2	Nameserver yang memiliki authority untuk suatu domain
PTR	12	Pointer untuk nama domain
SOA	6	Start of Authority
WKS	11	Well-Known Services, memberikan spesifik dari suatu layanan di jaringan tersebut

- RData : nilainya bergantung dari tipenya, contoh:
    - A Alamat IP
    - CNAME nama domain
    - MX 16 bit prioritas diikuti dengan nama domain
    - NS nama host
    - PTR nama domain

9.1.13. Transport

Pesan DNS dikirimkan melalui UDP dan TCP

- UDP : port 53
    - Digunakan untuk transfer zone antar nameserver, dengan panjang pesan 512 byte.
- TCP : port 53
    - Panjang total frame dari pesan

9.1.14. Aplikasi DNS

DNS di implementasikan pada :

host Mendapatkan alamat IP dari suatu nama host atau mendapatkan nama host dari suatu alamat IP

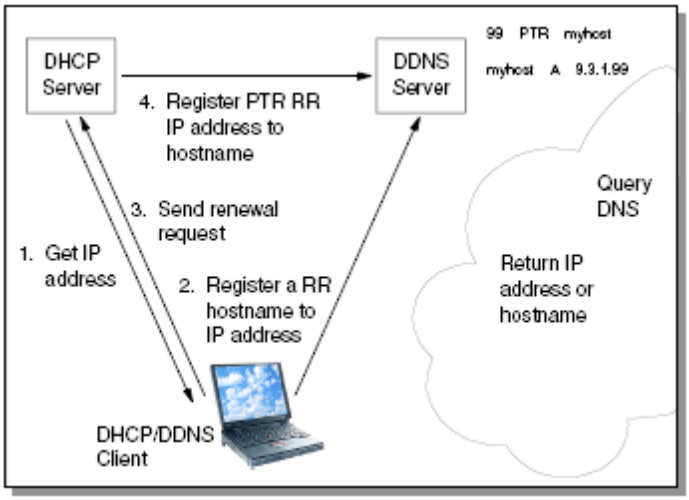
nslookup Mencari informasi tentang node jaringan, dan memeriksa isi database dari nameserver

dig Mencari informasi yang lebih lengkap dari suatu nama domain. DIG singkatan dari Domain Internet Groper

Bind Aplikasi nameserver

9.2. Dynamic Domain Name System (DDNS)

DDNS digunakan pada client yang menggunakan sistem DHCP, dimana DHCP server mengirimkan pesan kepada nameserver untuk mencatat IP dan nama host. Cara kerja DDNS dapat dilihat pada Gambar 9.4.



Gambar 9.4 DDNS

Dimana :

1. Client mendapatkan alamat IP dari DHCP server
2. Client mengirimkan nama host dengan alamat IP menuju DHCP server
3. Mengirim permintaan pembaruan pada saat proses DHCP
4. Mendaftarkan PTR RR alamat IP ke nama host

9.3. Network Information System (NIS)

NIS bukan merupakan standar internet. NIS digunakan untuk berbagi informasi pada lingkungan unix. Informasi yang dapat dibagi antara lain /etc/passwd, /etc/group dan /etc/hosts.

NIS memiliki kelebihan antara lain :

- Memberikan konsistensi ID pengguna dan ID group pada jaringan yang besar
- Mempersingkat waktu untuk mengelola ID pengguna, ID group dan kepemilikan NFS baik oleh pengguna itu sendiri maupun sistem administrator

Sistem NIS terdiri dari :

NIS master server Mengelola peta atau database dari password pengguna

NIS slave server Cadangan dari NIS master server

NIS client Sistem yang dilayani oleh NIS server

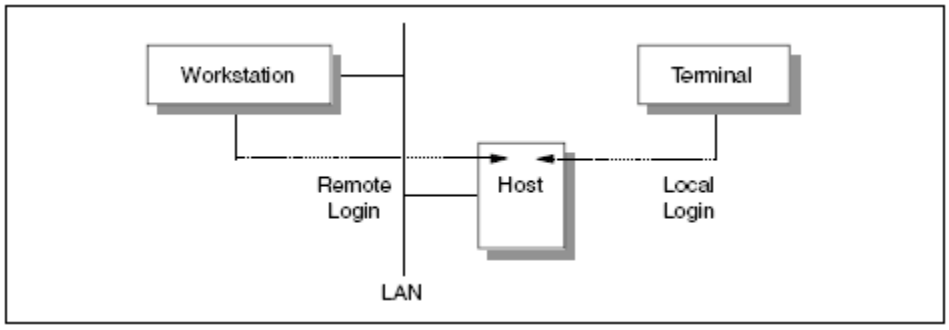
Bab 10. Eksekusi Jarak Jauh

Salah satu dasar mekanisme jaringan komputer adalah dapat melakukan perintah komputer secara jarak jauh. Pengguna dapat menjalankan aplikasi programnya pada komputer yang letaknya terpisah secara jauh. Salah satu aplikasi yang dapat melakukan aksi jarak jauh adalah TELNET.

10.1. TELNET

Telnet merupakan protokol standar dengan STD nomer 8. Dijelaskan pada RFC 854 – TELNET protocol spesification dan RFC 855 – TELNET options Spesifications.

TELNET memberikan interface pada suatu program di salah satu host (TELNET client) untuk mengakses sumber daya yang berada pada host yang lainnya (TELNET server) sehingga client akan merasakan melakukan kegiatan seperti pada hostnya sendiri. Terlihat seperti pada Gambar 10.1.



Gambar 10.1 TELNET – melakukan login jarak jauh dengan TELNET

Sebagai contoh, seorang pengguna menggunakan sebuah workstation pada LAN melakukan akses ke suatu host yang juga terhubung pada LAN sehingga merasa seperti menggunakan terminal pada host.

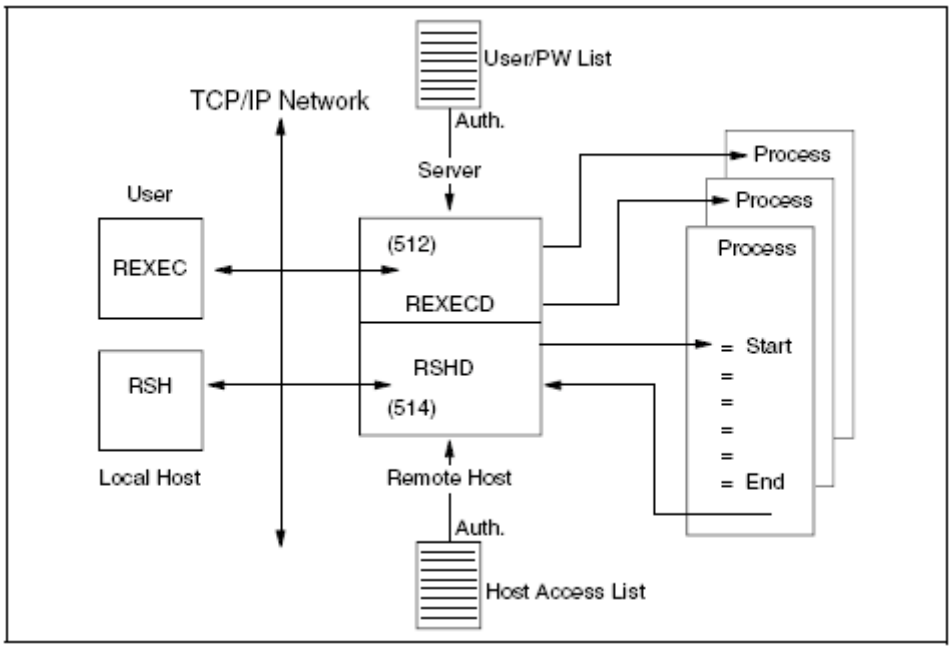
Kebanyakan telnet tidak memberikan fasilitas grafik interface.

10.2. Remote Execution Command protocol (REXEC dan RSH)

Remote EXEcution Command Daemon (REXECD) adalah merupakan server yang memperbolehkan menjalankan suatu perintah yang dikirimkan oleh suatu host melalui jaringan TCP/IP, client menggunakan aplikasi REXEC atau menggunakan Remote Shell Protocol (RSH) untuk mentransfer suatu kegiatan dari host satu ke host yang lainnya.

REXECD merupakan server (atau daemon). Dimana tugasnya menangani perintah dari host lainnya, kemudian meneruskan perintah tersebut ke virtual machine untuk dilakukan action perintah. Daemon memberikan login secara otomatis apabila nama user dan password setelah dimasukkan.

REXEC menggunakan TCP port 512, sedangkan RSH menggunakan Tcp 514. Dijelaskan seperti pada Gambar 10.2

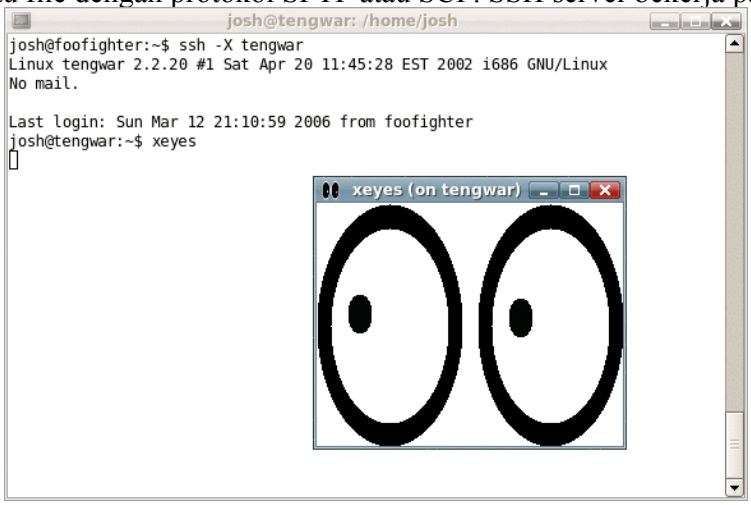


Gambar 10.2 Prinsip REXEC dan REXECD

10.3. Secure Shell (SSH)

Pada dunia komputer, secure shell atau SSH adalah protokol standar yang membentuk jalur yang aman pada komunikasi antar komputer. SSH menggunakan teknik enkripsi public key pada sistem authentikasi pengguna untuk mengakses komputer yang lain. SSH memberikan sistem enkripsi pada jalur yang digunakan, sehingga memberikan tingkat keamanan data yang tinggi.

SSH biasa digunakan untuk melakukan remote login dan menjalankan perintah pada komputer remote, tetapi SSH juga dapat digunakan sebagai tunnel jaringan, melakukan penerusan pada port TCP, dan koneksi X11. Selain itu dapat juga digunakan untuk mentransfer suatu file dengan protokol SFTP atau SCP. SSH server bekerja pada port 22.



Gambar 10.3 Contoh penggunaan SSH

10.3.1. Sejarah SSH

Pada tahun 1995, Tatu Ylonen, peneliti dari Helsinki University of Technology, Finlandia, mendesign suatu protokol keamanan yang bisa mengamankan dari teknik password sniffing. Keberhasilan SSH menggantikan protokol rlogin, TELNET, dan rsh. Dimana protokol-protokol tersebut tidak memberikan fasilitas keamanan authentikasi dan kerahasiaan data. Ylonen mempublikasikan protokol ini secara freeware pada juli 1995.

Pada Desember 1995, Ylonen mendirikan SSH Communications Security yang digunakan untuk memasarkan dan mendevelop SSH, dan SSH berkembang menjadi protokol proprietary.

Pada 1996, SSH-1 mengalami revisi menjadi SSH-2 dengan menggunakan algoritma yang lebih aman.

Pada tahun 1999, beberapa komunitas menginginkan adanya versi SSH yang berbasis open source, sehingga dibentuk yang namanya OpenSSH.

10.3.2. Penggunaan SSH

SSH banyak digunakan untuk :

- Dengan SSH client yang digunakan untuk pengontrolan server secara jarak jauh.
- Dengan kombinasi SFTP dapat melakukan transfer file
- Dengan kombinasi rsync dapat digunakan sebagai mirror, backup
- Dengan kombinasi SCP digunakan untuk aplikasi rcp dengan kemampuan keamanan data
- Penerus Port atau tunneling

10.4. Virtual Network Computing (VNC)

VNC adalah sistem yang digunakan untuk melakukan pembagian sumber untuk desktop, dimana menggunakan protokol RFB (Remote Frame Buffer) yang digunakan untuk mengatur komputer lain secara jarak jauh. VNC mengirimkan informasi penekanan tombol keyboard dan klik pada mouse sehingga dapat mengontrol komputer lain pada jaringan dan menampilkan layar pada komputer pengontrol.

VNC bersifat platform-independent, artinya VNC viewer dapat terhubung dengan VNC server walau berbeda sistem operasi. Terdapat berbagai macam VNC server-client dan dalam bentuk java. VNC dapat dikontrol dari beberapa client sekaligus dalam saat yang bersamaan. VNC banyak digunakan dalam hal remote technical support, akses file dari komputer di rumah ke komputer tempat kerja.

VNC pertama kali dikembangkan di AT&T, dan bersifat opensource dengan lisensi GPL.

10.4.1. Cara Kerja VNC

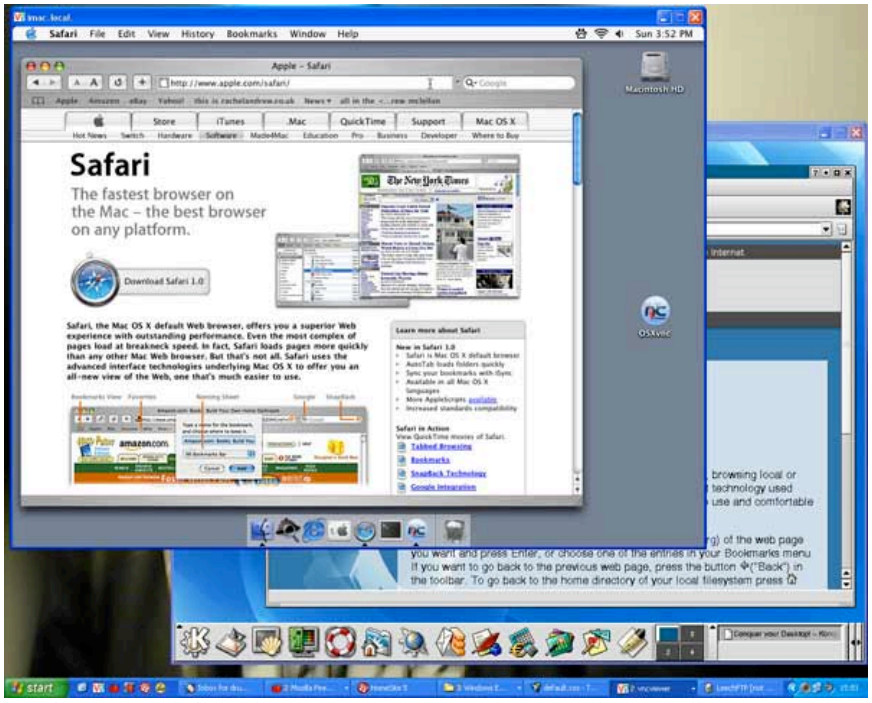
VNC memiliki 2 bagian yaitu, client dan server. Server adalah program yang membagi sumber dan layar pada komputer, dan Clinet (viewer) adalah program yang melihat dan melakukan interaksi dengan server.

VNC menggunakan protokol yang sederhana berdasarkan cara kerja graphic yaitu “letakkan kotak pada posisi x,y yang diberikan”. Server mengirimkan framebuffer

sebesar kotak yang ditentukan kepada client. Sehingga untuk mengirimkan gambar hanya diperlukan untuk bagian yang bergerak saja, tetapi bila terjadi pergerakkan gambar yang menuntut sepenuh layar, maka gambar yang dikirimkan juga sebesar gambar sepenuh layar tersebut.

VNC menggunakan port 5900 hingga 5906, tiap port mewakili dari port pada layar X-windows (port 5900 hingga 5906 untuk layar 0 hingga 6). Untuk viewer berupa java diimplementasikan pada RealVNC pada port 5800 hingga 5806. Port tersebut dapat dirubah.

Pada komputer Windows, komputer hanya dapat menggunakan 1 layar tidak seperti Unix. Sehingga hanya menggunakan port 5900.



Gambar 10.4 VNC di Windows mengakses VNC di MAC dan Linux

10.5. Remote Desktop Protocol (RDP)

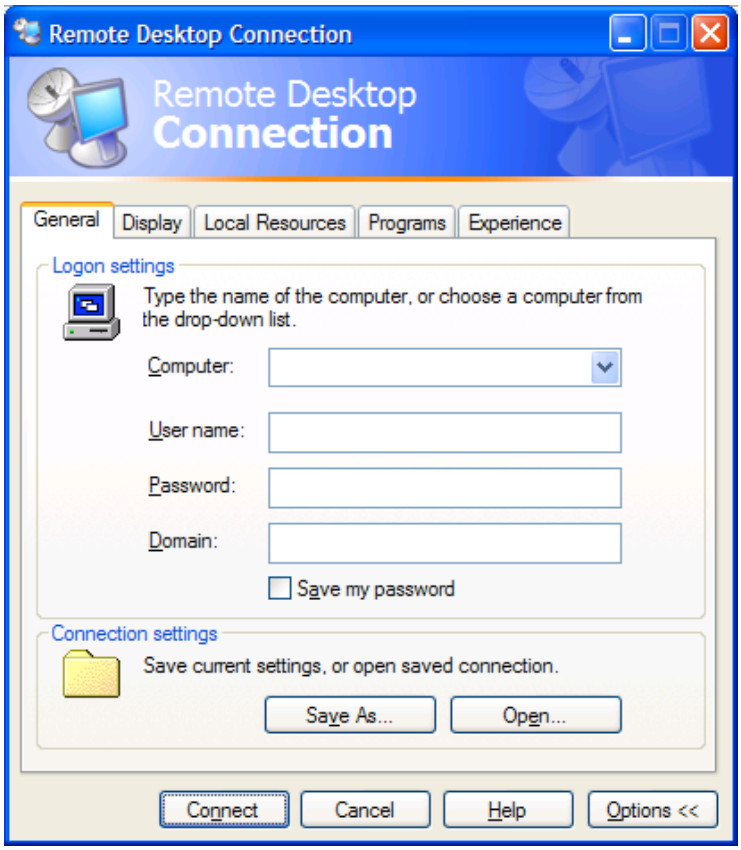
RDP adalah protokol multi-channel yang memperbolehkan user untuk terkoneksi dengan Microsoft Terminal Services. Untuk client dapat dilakukan dari sistem operasi Windows, dan sistem operasi yang lainnya seperti Linux, FreeBSD, Mac OS X. Pada bagian server aplikasi menggunakan port 3389.

Versi awal dari RDP adalah versi 4.0, dimana digunakan pada Terminal Services pada sistem operasi Windows NT 4.0 Server, Terminal Server Edition. Pada Windows 2000 menjadi versi 5.0 dengan tambahan fitur seperti dapat melakukan mencetak pada printer yang terpasang di komputer lokal. Versi 5.1 berada di Windows XP Proffesional, dimana mampu menampilkan grafik 24-Bit dan suara. Versi 5.2 terdapat di Windows 2003, dimana memiliki fitur console mode connection. Dan pada windows Vista akan menggunakan versi 6.0

10.5.1. Fitur

- Mendukung penggunaan warna 24bit
- Enkripsi 128bit
- Mendukung Transport Layer Security
- Menggunakan aplikasi audio tetapi didengarkan di komputer lokal
- File System Redirection
- Printer Redirection
- Port Redirection
- Clipboard dapat digunakan pada komputer lokal atau komputer remote
- Berbagi sumber harddisk dengan komputer remote

10.5.2. Contoh Aplikasi



Gambar 10.5 Remote Desktop Connection

Bab 11. Protokol Transfer File

Protokol TCP/IP memiliki beberapa aplikasi, terutama yang berhubungan dengan memodifikasi file. Ada 2 mekanisme untuk melakukan transfer file, mekanisme yang pertama melakukan pengiriman file dari komputer lain ke komputer lokal, dan mekanisme yang lain adalah menggunakan mekanisme file sistem, dimana ada suatu mekanisme yang memperbolehkan suatu pengguna untuk melakukan perubahan terhadap file yang berada di komputer yang lain.

Contoh protokol yang menggunakan mekanisme pertama adalah FTP dan TFTP, sedangkan yang menggunakan mekanisme kedua adalah NFS.

11.1. File Transfer Protocol (FTP)

FTP merupakan protokol standar dengan STD nomer 9. Dijelaskan pada RFC 959 – File Transfer Protocol (FTP) dan diupdate dengan RFC 2228 – FTP security extention.

Melakukan duplikat file dari komputer yang satu dengan komputer yang lain dengan dapat dilakukan 2 arah. Client dapat mengirim file menuju ke server atau dapat meminta suatu file dari server.

Untuk mengakses file di server, pengguna diharuskan untuk mengidentifikasikan dirinya terlebih dahulu. Dan server akan melakukan proses authentikasi untuk pengguna tersebut.

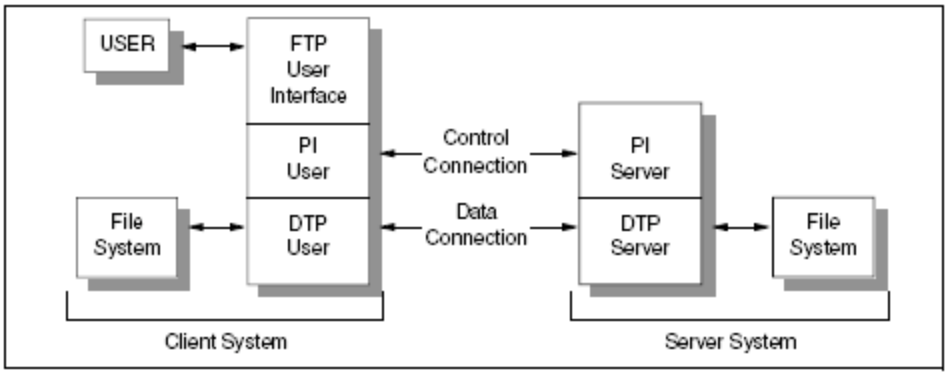
FTP menggunakan koneksi berbasis connection-oriented, sehingga dari kedua sisi harus memiliki koneksi TCP/IP.

11.1.1. Sekilas tentang FTP

FTP menggunakan TCP sebagai protokol transport. FTP server menerima koneksi pada port 20 dan 21. Diperlukan 2 koneksi, yaitu untuk login dengan menggunakan protokol TELNET, dan yang satunya digunakan untuk transfer file.

Pada kedua sisi jaringan, aplikasi FTP dilengkapi dengan protocol interpreter (PI), data transfer process (DTP), dan tampilan antar muka.

Sehingga prinsip kerja protokol FTP adalah, user interface melakukan perintah melalui PI dan dilanjutkan ke sisi server. Untuk melakukan transfer file PI memberikan perintah kepada DTP untuk mengirimkan file. Dapat dilihat pada Gambar 11.1.



Gambar 11.1 FTP – Prinsip kerja FTP

11.1.2. Operasional FTP

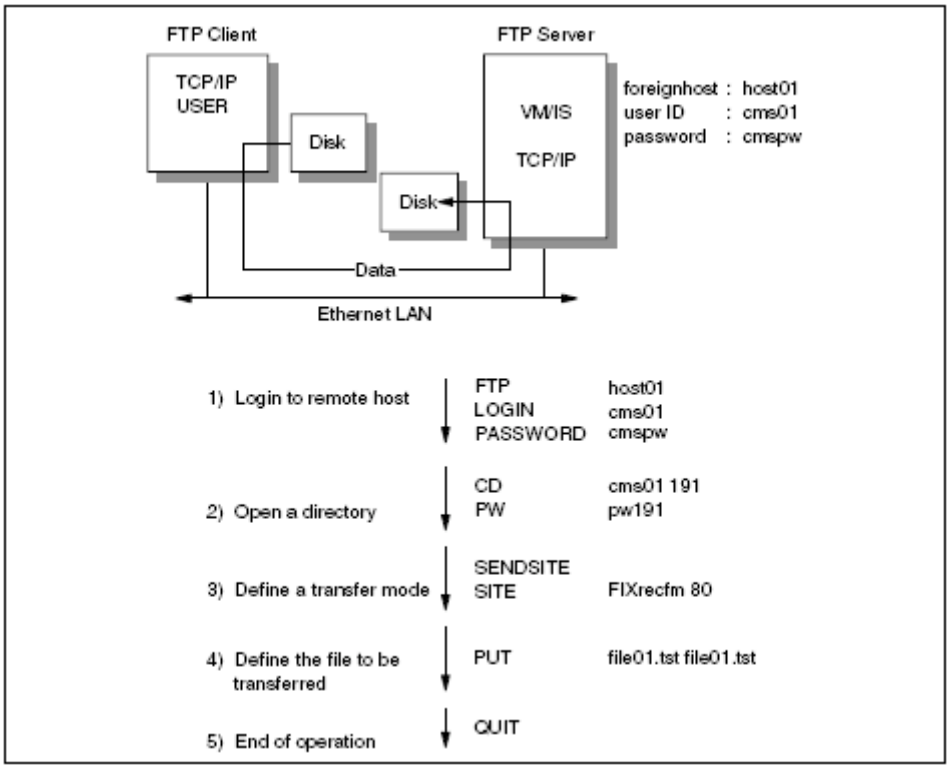
Ketika melakukan FTP, pengguna akan melakukan beberapa atau semua operasional yang ada, yaitu :

- Melakukan koneksi ke host lain
  - Dengan perintah **Open** dan memasukkan user dan password dengan perintah **User** dan **Pass**.
- Memilah direktori
  - Dengan perintah **cd** dan menunjuk ke direktori yang dituju
- Melihat list dari file
  - Dengan perintah **dir** atau **ls**
- Memilih cara transfer file
  - Dengan perintah **bin** atau **ascii**
- Mentransfer file
  - Dengan perintah **get** untuk mengambil file, **mget** untuk mengambil file dengan jumlah lebih dari 1, **put**,mengirim file, dan **mput** mengirim file dengan jumlah lebih dari 1.
- Menggunakan mode passive
  - Dengan perintah **passive** client yang berada di balik firewall dapat melakukan FTP seolah-olah berasal dari luar firewall.
- Menutup koneksi
  - Dengan perintah **quit, bye,** atau **logout**

11.1.3. Skenario FTP

Seorang pengguna pada jaringan LAN, akan mengirimkan file dengan FTP, yang akan dilakukan adalah seperti Gambar 11.2.

User akan mengakses server dengan nama host01, dimana pada host tersebut pengguna terdaftar sebagai usernama cms01 dengan password cmspw. Kemudian user tersebut akan memilah direktori dan memilih jenis mode transfer yang akan dipakai. Direktori yang dipakai adalah 191 dan mode yang digunakan FIXrecfm80. Kemudian pengguna mengirim file dengan perintah PUT. Nama file yang dikirim adalah file01.tst. Dan terakhir menutup koneksi dengan perintah QUIT.



Gambar 11.2 FTP – Skenario FTP

11.1.4. Contoh Penggunaan FTP

Contoh penggunaan FTP dapat dilihat pada Gambar 11.3.

```
[C:\SAMPLES]ftp host01.itsc.raleigh.ibm.com
Connected to host01.itsc.raleigh.ibm.com.
220 host01 FTP server (Version 4.1 Sat Nov 23 12:52:09 CST 1991) ready.
Name (rs60002): cms01
331 Password required for cms01.
Password: xxxxxx
230 User cms01 logged in.
ftp> put file01.tst file01.tst
200 PORT command successful.
150 Opening data connection for file01.tst (1252 bytes).
226 Transfer complete.
local: file01.tst remote: file01.tst
1285 bytes received in 0.062 seconds (20 Kbytes/s)
ftp> close
221 Goodbye.
ftp> quit
```

Gambar 11.3 FTP – Contoh penggunaan FTP

11.1.5. Anonymous FTP

Anonymous FTP adalah model FTP yang tanpa menggunakan authentikasi pada pengguna. Sehingga pengguna siapa pun bisa melakukan transfer, tapi biasanya hanya diperbolehkan untuk mengambil suatu file.

11.2. Trivial File Transfer Protocol (TFTP)

TFTP merupakan standar protokol dengan STD nomer 33. Dijelaskan pada RFC 1350 – The TFTP Protocol. Dan diupdate pada RFC 1785, 2347, 2348, dan 2349.

Transfer TFTP adalah transfer file antar disk (disk-to-disk), dengan menggunakan API SENDFILE.

TFTP menggunkan protokol UDP. TFTP client melakukan inisialisasi dengna mengirim permintaan untuk read/write melalui port 69, kemudian server dan client melakukan negosiasi tentang port yang akan digunakan untuk melakukan transfer file.

11.2.1. Penggunaan TFTP

Perintah TFTP <hostname> membawa pengguna pada prompt interaktiv, dimana dapat melanjutkan dengan sub perintah, antara lain

Connect <host>	Menentukan tujuan
Mode <ascii/binary>	Menentukan mode pengiriman
Get <nama file remote> [<nama file lokal>]	Mengambil file
Put <nama file remote> [<nama file lokal>]	Menaruh file
Verbose	
Quit	Keluar TFTP

11.3. Network File System (NFS)

SUN Microsystems Network File System (NFS) adalah protokol yang dapat membagi sumber daya melalui jaringan. NFS dibuat untuk dapat independent dari jenis mesin, jenis sistem operasi, dan jenis protokol transport yang digunakan. Hal ini dilakukan dengan menggunakan RPC.

NFS dijelaskan pada RFC 1813 – NFS: NFS Version 3 Protocol dan RFC 3010 – NFS Version 4 Protocol.

11.3.1. Konsep NFS

NFS memperbolehkan user yang telah diijinkan untuk mengakses file-file yang berada di remote host seperti mengakses file yang berada di lokal. Protokol yang digunakan :

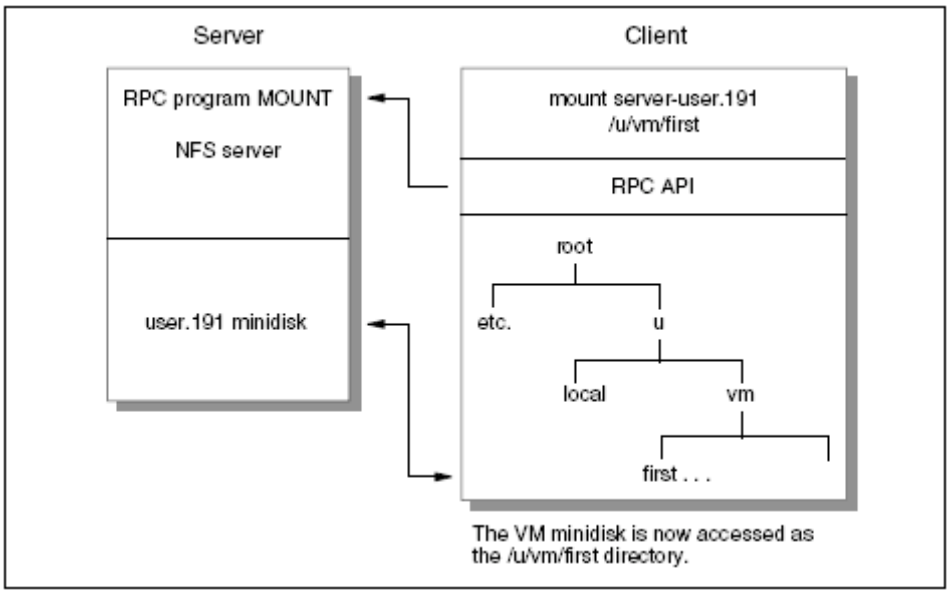
1. Protokol mount menentukan host remote dan jenis file sistem yang akan diakses dan menempatkan di suatu direktori
2. Protokol NFS melakukan I/O pada remote file sistem

Protokol mount dan protokol NFS bekerja dengan menggunakan RPC dan mengirim dengan protokol TCP dan UDP.

11.3.1.1. Protokol Mount

Protokol ini digunakan untuk membuat link dengan cara me-mount pada suatu direktori

Perintah yang digunakan adalah **mount**.



Gambar 11.4 Protokol Mount

Untuk mengakses pengguna harus menjalankan program mount terlebih dahulu, contoh :

```
# mount //remote/share /mnt
```

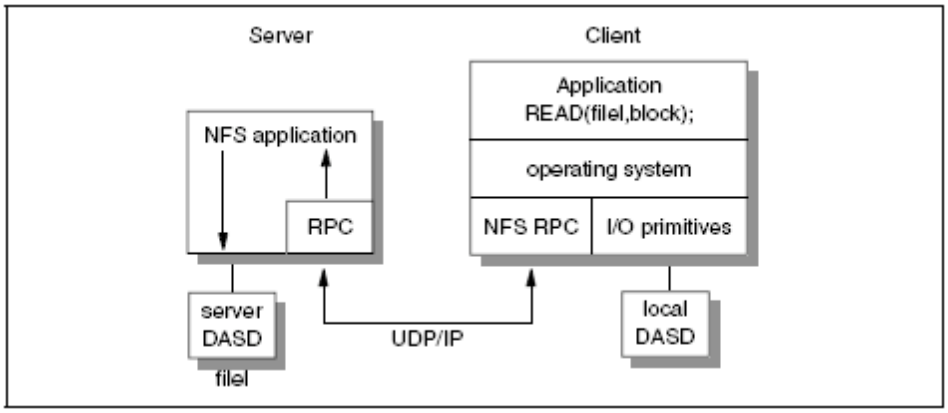
Peritah tersebut digunakan untuk mengakses server dengan nama remote dan memiliki direktori yang dibagikan dengan nama share, kemudian di mount di direktori /mnt pada komputer lokal.

Setelah selesai menggunakan atau memodifikasi file, pengguna harus melakukan pelepasan dengan perintah **umount**, contoh :

```
# umount /mnt
```

11.3.1.2. Protokol NFS

NFS adalah program RPC yang memberikan I/O kepada remote host, setelah di lakukan permintaan oleh program mount.



Gambar 11.5 Protokol NFS

11.3.2. NFS versi 4

Merupakan perbaikan dari versi 3, dengan beberapa fitur tambahan antara lain :

- Mengurangi transfer informasi yang dibutuhkan oleh protokol mount
- Keamanan pada layer RPC
- Mendukung RPCSEC_GSS
- Mendukung kerberos
- Bentuk baru dari file handle
- Gabungan perintah lookup dan read
- Mendukung format file 32bit

Bab 12. Aplikasi Surat (Mail)

Electronic-Mail (E-Mail) merupakan aplikasi TCP/IP yang paling banyak digunakan. Bab ini membahas protokol yang mendukung aplikasi email.

12.1. Simple Mail Transport Protocol (SMTP)

SMTP merupakan protokol dasar yang bertugas untuk menukarkan email (mail exchange) antar host yang berbasis TCP/IP. Standar dari protokol ini ada 3 yaitu :

- Standar yang digunakan untuk pertukaran email antar komputer (STD 10/RFC 821), disebut standar SMTP
- Standar yang digunakan untuk format pesan (STD 11) dengan dijabarkan pada RFC 822 yang berisi tentang sintak mail dan RFC 1049 yang berisi tentang penggunaan file yang bukan berupa ASCII text (email menggunakan 7bit ASCII) supaya dapat digunakan pada badan email. Standar ini disebut MAIL
- Standar yang digunakan untuk menjalurkan email berdasarkan domain name system (DNS), dijabarkan pada RFC 974 dengan nama DNS-MX

Standar diatas digunakan untuk email yang menggunakan format bahasa Inggris, sedangkan standar penggunaan email yang mendukung penggunaan bahasa lain antara lain :

- Multipurpose Internet Mail Exchange (MIME) dijabarkan pada RFC 2045 hingga 2049.
- Pelayanan tambahan dari SMTP berupa : pemberitahuan service extension pada SMTP client, penggunaan 8bit format data, batas ukuran email.

12.1.1. Cara kerja SMTP

SMTP bekerja berdasarkan pengiriman end-to-end, dimana SMTP client akan menghubungi SMTP server untuk segera mengirimkan email. SMTP server melayani pengguna melalui port 25.

Dimana setiap pesan harus memiliki :

- Header atau amplop, yang dijabarkan pada RFC 822.
- Kontent, yang berisi tentang isi dari surat yang akan dikirimkan.

12.1.1.1. Format mail header

Pengguna tidak perlu kebingungan tentang mail header, karena semuanya sudah diatur oleh SMTP.

Format dari mail header adalah

```
Bagian-nama : Bagian-isi
```

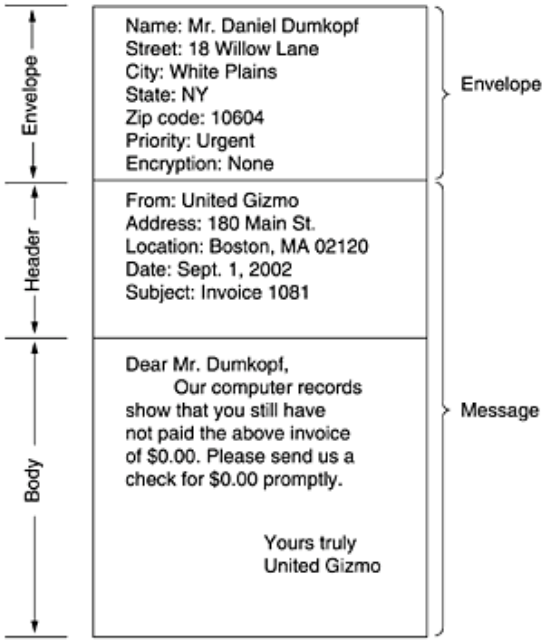
Contoh penggunaan mail header :

```
To: Sukaridhoto <dhoto@eepis-its.edu>
```

Contoh bagian header yang sering digunakan antara lain

Tabel 12.1 SMTP – Header yang sering digunakan

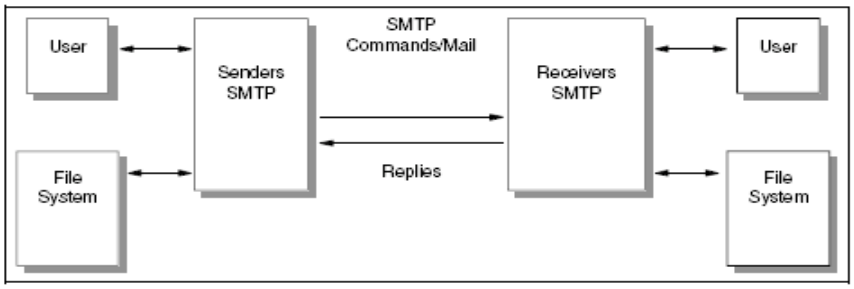
Kata kunci	Nilai
to	Tujuan dari email
cc	Tujuan kedua dari email (carbon-copy)
from	Pengirim email
reply-to	Alamat pengembalian email
return-path	Alamat host untuk pengembalian email
Subject	Subjek tentang email yang diisikan oleh pengguna



Gambar 12.1 Envelope, Header, Body

12.1.1.2. Mail Exchange

Model SMTP dapat dilihat pada Gambar 12.2. Dari hasil pengguna meminta mail. SMTP pengirim melakukan koneksi 2 arah dengan SMTP penerima. SMTP dapat berupa tujuan akhir atau penerus (mail gateway). SMTP pengirm akan membangkitkan perintah untuk melakukan reply to pada SMTP penerima.

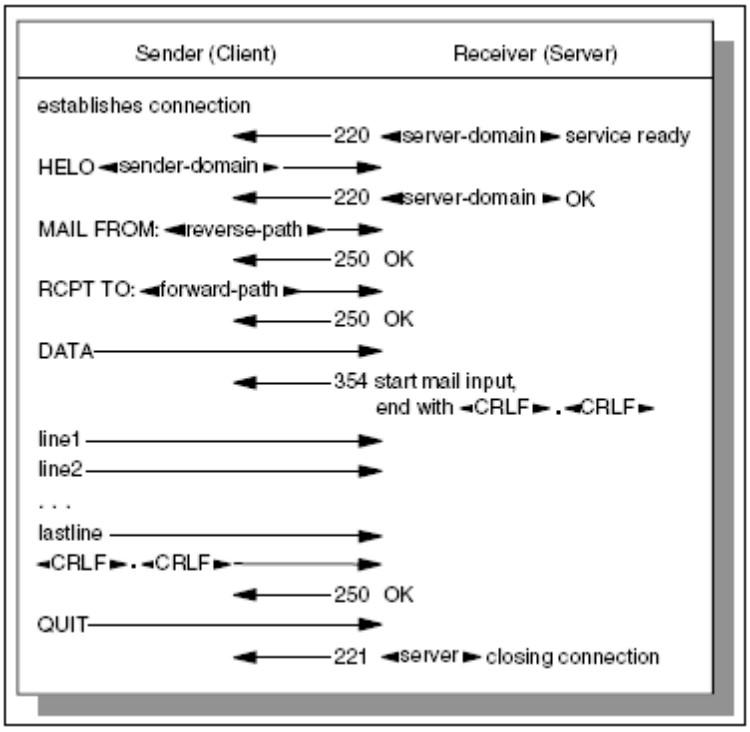


Gambar 12.2 Model SMTP

Diagram alir pertukaran surat SMTP

Pertukaran email yang terjadi adalah sebagai berikut :

1. SMTP Pengirim melakukan koneksi TCP/IP dengan SMTP penerima dan menunggu server untuk mengirim pesan 220 yang menandakan pelayanan terhadap pesan sudah siap atau pesan 421 pelayanan tidak siap
2. HELO (kependekan dari hello) dikirim oleh server dengan menunjukkan nama domain.
3. Pengirim akan memulai memberikan perintah kepada SMTP dimana apabila SMTP mendukung perintah tersebut akan membalas dengan pesan 250 OK
4. Memberikan informasi kepada SMTP tentang tujuan dari email dengan perintah RCPT TO dilanjutkan dengan alamat email yang dituju.
5. Setelah tujuan diset, dilanjutkan dengan perintah DATA yang menunjukkan bahwa baris berikutnya adalah isi dari email dengan diakhiri dengan <CRLF>.<CRLF>
6. Client mengisikan data sesuai dengan pesan yang akan dikirimkan hingga mengisikan <CRLF>.<CRLF>
7. Pengirimkan akan menghentikan kegiatan dengan memberi perintah QUIT.



Gambar 12.3 Aliran SMTP

Dapat dicontohkan dengan :

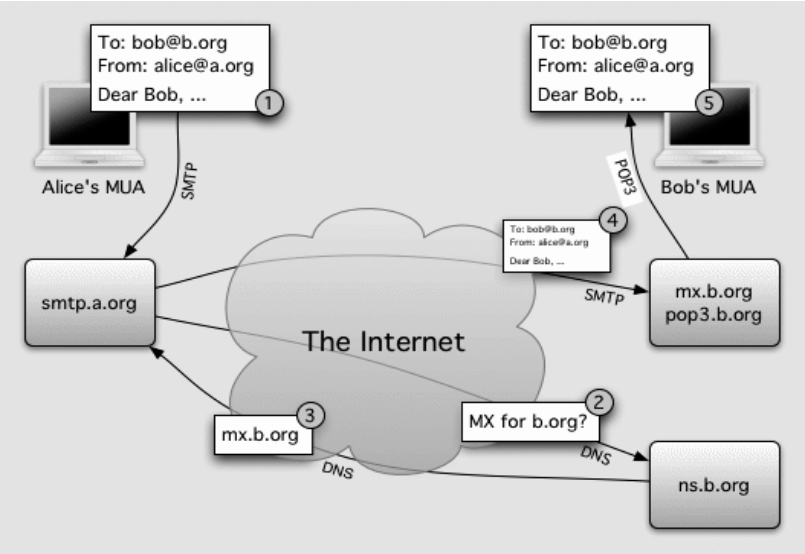
```
R: 220 delta.aus.edu Simple Mail Transfer Service Ready
S: HELO stockholm.ibm.com
R: 250 delta.aus.edu
S: MAIL FROM:<abc@stockholm.ibm.com>
R: 250 OK
S: RCPT TO:<xyz@delta.aus.edu>
R: 250 OK
S: RCPT TO:<opq@delta.aus.edu>
R: 550 No such user here
S: RCPT TO:<rst@delta.aus.edu>
R: 250 OK
S: DATA
R: 354 Start mail input, end with <CRLF>.<CRLF>
S: Date: 23 Jan 89  18:05:23
S: From: Alex B. Carver <abc@stockholm.ibm.com>
S: Subject: Important meeting
S: To:  <xyz@delta.aus.edu>
S: To:  <opq@delta.aus.edu>
S: cc:  <rst@delta.aus.edu>
S:
S: Blah blah blah
S: etc.....
S: .
R: 250 OK
S: QUIT
R: 221 delta.aus.edu Service closing transmission channel
```

Gambar 12.4 Contoh penggunaan SMTP

12.1.2. SMTP dan Domain Name System

Apabila jaringan menggunakan DNS, maka SMTP tidak dapat hanya dengan mudah mengirimkan suatu email ke TEST.IBM.COM hanya dengan membuka koneksi TCP ke TEST.IBM.COM. Yang dilakukan pertama kali adalah melakukan query ke server name dan mendapatkan hasil ke arah mana tujuan tersebut.

SMTP akan mencari record pada DNS dengan tanda MX, dan akan mengirimkan ke email ke host yang tercatat pada host tersebut



Gambar 12.5 Cara kerja Email

12.2. Multipurpose Internet Mail Extensions (MIME)

MIME adalah standar internet yang menyambung format email supaya mendukung format text dengan format selain US-ASCII, non-text attachment, multi-part pada badan pesan, dan informasi pada header. Keseluruhan email yang ditulis oleh pengguna akan dikirim melalui SMTP dengan format MIME. Selain digunakan pada sistem email MIME juga digunakan pada protokol lainnya seperti HTTP pada world wide web. MIME dijabarkan pada RFC 2045, RFC 2046 dan RFC 2049. Dasar internet untuk protokol email , SMTP, hanya mendukung 7bit ASCII, karena itu ditambah dukungan dengan MIME supaya bisa mendukung yang lainnya.

12.2.1. Header yang terdapat pada MIME

```
MIME-Version: 1.0
From: Steve Hayes <steve@hayessj.bedfont.uk.ibm.com>
To:   Matthias Enders <enders@itso180.itso.ral.ibm.com>
Subject: Multipart message
Content-type: multipart/mixed; boundary="1995021309105517"

This section is called the preamble. It is after the header but before the first
boundary. Mail readers which understand multipart messages must ignore this.
--1995021309105517

The first part. There is no header, so this is text/plain with
charset=us-ascii by default. The immediately preceding <CRLF> is part of the
<CRLF><CRLF> sequence that ends the null header. The one at the end is part of the
next boundary, so this part consists of five lines of text with four <CRLF>s.
--1995021309105517
Content-type: text/plain; charset=us-ascii
Comments: this header explicitly states the defaults

One line of text this time, but it ends in a line break.

--1995021309105517
Content-Type: multipart/alternative; boundary=_
Comments:     An encapsulated multipart message!

Again, this preamble is ignored. The multipart body contains a still image and a
video image encoded in Base64. See 11.2.3.5, "Base64 encoding" on page 413. One
feature is that the character "_" which is allowed in multipart boundaries never
occurs in Base64 encoding so we can use a very simple boundary!
--_
Content-type: text/plain

This message contains images which cannot be displayed at your terminal.
This is a shame because they're very nice.

--_
Content-type: image/jpeg
Content-transfer-encoding: base64
Comments: This photograph is to be shown if the user's system cannot display MPEG
videos. Only part of the data is shown in this book
because the reader is unlikely to be wearing MIME-compliant spectacles.

Qk1OAAAAAAAAAE4EAABAAAAAQAEAAAPAAAAAABAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAQAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAB4VjQSAAAAAAAAgAAAkgAAAJKAAKcAAACgAIAAqpIAAMHBwQDJyckA
/9uqAKpJAAD/SQAAAG0AAFVtAACqbQAA/20AAAAkAABVkgAAqiQAAP+SAAAAtgAAVbYAAKq2AAD/
<base64 data continues for another 1365 lines>
--_
Content-type: video/mpeg
Content-transfer-encoding: base64

AAABswoAeBn//+CEAAABsgAAAOgAAAG4AAAAAAAAAQAAT/////wAAAGy//8AAAEBQ/Z1IwwBGWCX
+pqMiJQDjAKywS/1NRrtXcTCLgzVQymqqHAf0sL1sMgMq4SWLCwOTYRdgyAyrhNYsLhhF3DLjAGg
BdwDXBv3yMV8/4tzrp3zsAWIGAJg1IBKTeFFI2IsgutIdfuSaAGCTsBVnWdz8afdMMAMgKgMEkPE
<base64 data continues for another 1839 lines>
--_--
That was the end of the nested multipart message.  This is the epilogue.
Like the preamble it is ignored.
--1995021309105517--
And that was the end of the main multipart message. That's all folks!
```

Gambar 12.6 Contoh MIME

12.2.1.1. MIME-Version

Versi yang digunakan pada MIME

```
MIME-Version: 1.0
```

12.2.1.2. Content-Type

Tipe yang digunakan pada pesan

```
Content-Type: text/plain
```

Tabel 12.2 Contoh Content-type

Tipe	Subtipe	Deskripsi
Text	Plain	Unformated text
	Enriched	Text yang memiliki format
Image	Gif	Gambar dengan format GIF
	Jpeg	Gambar dengan format JPEG
Audio	Basic	Suara
Video	Mpeg	Film dengan format MPEG
Application	Octet-Stream	Sequence yang tidak terinterpreted
	Postscript	Dokumen for postscript
Message	RFC822	MIME RFC 822
	Partial	Pesan yang dipisah
	External-body	Pesan yang ditarik dari jaringan
Multipart	Mixed	Independent
	Alternative	Pesan yang sama beda format
	Parallel	Bagian yang harus dilihat secara bersamaan
	Digest	Tiap bagian merupakan bagian RFC 822

12.2.1.3. Content-Transfer-Encoding

Metode yang digunakan untuk pengiriman pada email, yaitu :

- 7bit
- Quoted-printable
- Base64

12.2.1.4. Encoded-Word

Digunakan bila menggunakan karakter lain

12.2.1.5. Multipart-Messages

Pemisah bagian pesan

```
Content-type: multipart/mixed; boundary="frontier"
MIME-version: 1.0

This is a multi-part message in MIME format.
--frontier
Content-type: text/plain
```

```
This is the body of the message.
--frontier
Content-type: text/html; encoding=UTF-8
Content-transfer-encoding: base64

PGh0bWw+CiAgPGhlYWQ+CiAgPC9oZWFkPgogIDxib2R5PgogICAgPHA+VGhpcyBpcyB0aGUg
Ym9keSBvZiB0aGUgbWVzc2FnZS48L3A+CiAgPC9ib2R5Pgo8L2h0bWw+Cg==
--frontier--
```

12.3. Post-Office-Protocol (POP)

Para pengguna email, akan menggunakan protokol POP untuk mengambil email yang berada di server. Protokol yang digunakan sekarang adalah versi 3 sehingga disebut POP3.

POP3 berkembang dari protokol sebelumnya yang disebut POP (biasa disebut POP1) dan POP2.

Protokol POP3 didesign untuk pengguna dengan jaringan yang sebentar-bentar harus dimatikan. Sehingga pengguna dapat menggunakan email tanpa harus terkoneksi secara terus-menerus. Walaupun pada POP3 terdapat pilihan “leave messages on server”, pengguna email biasanya akan mengkoneksikan, mengambil email dan menyimpan pada PC, menghapus email di server dan memutus koneksi.

POP3 server melayani pengguna melalui port 110.

```
S: <wait for connection on TCP port 110>
C: <open connection>
S:    +OK POP3 server ready <1896.697170952@dbc.mtview.ca.us>
C:    APOP mrose c4c9334bac560ecc979e58001b3e22fb
S:    +OK mrose's maildrop has 2 messages (320 octets)
C:    STAT
S:    +OK 2 320
C:    LIST
S:    +OK 2 messages (320 octets)
S:    1 120
S:    2 200
S:    .
C:    RETR 1
S:    +OK 120 octets
S:    <the POP3 server sends message 1>
S:    .
C:    DELE 1
S:    +OK message 1 deleted
C:    RETR 2
S:    +OK 200 octets
S:    <the POP3 server sends message 2>
S:    .
C:    DELE 2
S:    +OK message 2 deleted
C:    QUIT
S:    +OK dewey POP3 server signing off (maildrop empty)
C:  <close connection>
S:  <wait for next connection>
```

Gambar 12.7 Contoh Penggunaan POP3

12.4. Internet Message Access Protocol version 4 (IMAP4)

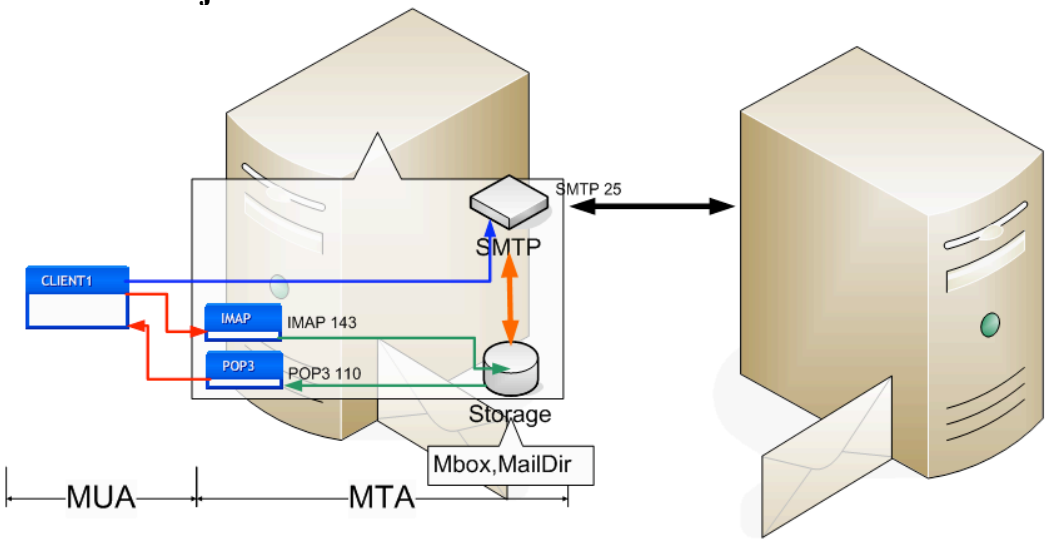
IMAP4 adalah protokol yang dapat digunakan oleh pengguna untuk membaca email di suatu server. IMAP4 dijabarkan pada RFC 3501.

Contoh penggunaan telnet pada IMAP

```
[]:[8:48am]:[/home/rnejdl] > telnet mail 143
Trying 66.13.175.242...
Connected to tethys.ringofsaturn.com.
Escape character is '^]'.
* OK [CAPABILITY IMAP4REV1 LITERAL+ SASL-IR LOGIN-REFERRALS AUTH=LOGIN] tethys.r
ingofsaturn.com IMAP4rev1 2004.350 at Sun, 8 Aug 2004 13:51:21 -0500 (CDT)
a001 CAPABILITY
* CAPABILITY IMAP4REV1 LITERAL+ IDLE NAMESPACE MAILBOX-REFERRALS BINARY UNSELECT
 SCAN SORT THREAD=REFERENCES THREAD=ORDEREDSUBJECT MULTIAPPEND SASL-IR LOGIN-REF
ERRALS AUTH=LOGIN
a001 OK CAPABILITY completed
a002 logout
* BYE tethys.ringofsaturn.com IMAP4rev1 server terminating connection
a002 OK LOGOUT completed
Connection closed by foreign host.
```

Gambar 12.8 Telnet IMAP

12.5. Cara kerja Email



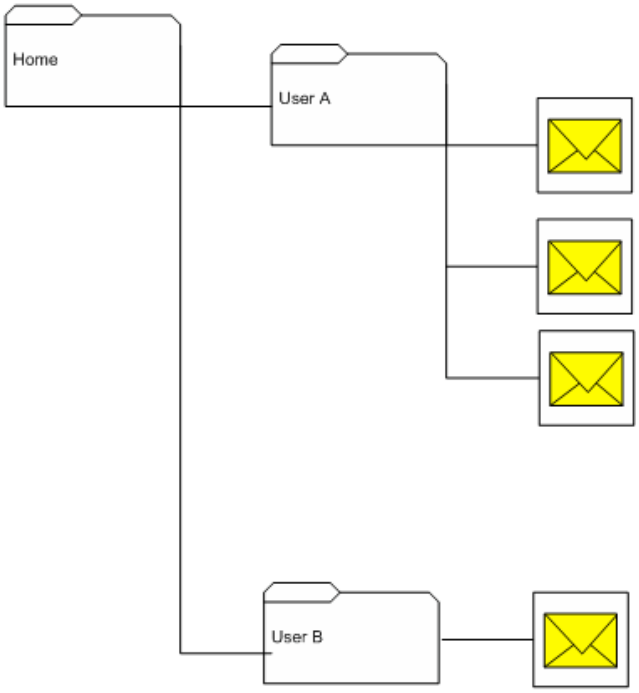
Gambar 12.9 Cara kerja EMAIL

Client menggunakan MUA (Mail User Agent) untuk membaca email dengan cara POP3 atau IMAP4. Dan untuk mengirimkan email melalui protokol SMTP.

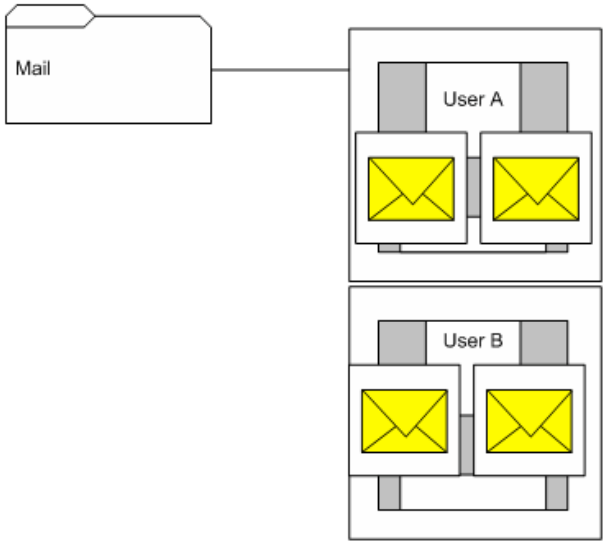
Antar mail server atau MTA (Mail Transfer Agent) saling bertukar email melalui protokol SMTP, dan menyimpan email dalam format Mbox atau Maildir.

Mbox adalah tipe penyimpanan email dimana email disimpan dalam 1 file untuk masing-masing user.

Maildir adalah tipe penyimpanan email dimana email disimpan dalam 1 folder untuk masing-masing user.



Gambar 12.10 Maildir



Gambar 12.11 Mbox

Bab 13. World Wide Web

Bab ini menjelaskan beberapa protokol dan aplikasi yang menjadikan internet mudah digunakan dan populer. Sebagai bukti, trafik world wide web yang menggunakan layanan hypertext transfer protocol (HTTP), dapat melebihi penggunaan protokol lainnya seperti TELNET dan FTP dalam penggunaan bandwidth. Dapat dipastikan setiap sistem operasi modern telah dilengkapi dengan aplikasi web browser, bahkan beberapa dilengkapi dengan web server. Dengan itu akan semakin mudah bagi pengguna dan dunia bisnis untuk dapat saling bertukar informasi di dunia jaringan komputer.

World wide web pertama kali dikembangkan pada tahun 1989 oleh Tim Berners Lee di European Laboratory untuk Particle Physic. Digunakan untuk berbagi dokumen dengan para ilmuwan.

Pada tahun 1993, penggunaan web semakin semarak, dengan dikembangkannya web browser berbasis grafik user interface oleh National Center of Supercomputing Applications (NCSA) yang disebut mosaic. Sehingga pengguna semakin mudah untuk melakukan akses web.

13.1. Hypertext Transfer Protocol (HTTP)

HTTP adalah suatu metode yang digunakan untuk transfer suatu informasi melalui world wide web. Didesign untuk memberikan cara untuk mempublikasikan dan mengambil halaman HTML.

Pengembangan HTTP dikoordinir oleh World Wide Web Concortium berkolaborasi dengan Internet Engineering Task Force, menghasilkan RFC 2616 yang berisikan tentang HTTP/1.1.

HTTP merupakan protokol yang digunakan untuk request/respon antara client dan server. Bentuk dari client adalah web browser, spider atau bentuk lainnya yang direferensi sebagai user agent. Dan tujuan server, dimana menyimpan atau membuat sumber daya seperti file HTML dan file gambar, disebut origin server. Diantara server dan client bisa terdapat penghubung (intermediate) antara lain proxy, gateway atau tunnel.

HTTP client memulai requestnya dengan menggunakan TCP sebagai layer transportnya dengan mengakses port 80 pada server.

Sumber daya yang diakses melalui HTTP disebut Uniform Resource Identifiers (URI) dengan mengakses suatu Uniform Resource Locators (URL).

13.1.1. Request Message

Pesan request terdiri dari :

- Request line, seperti GET / images/logo.gif HTTP/1.1, dimana artinya mengakses file logi.gif pada direktori images.

- Header, seperti Accept-Language : en
- Baris kosong
- Pilihan badan pesan.

Request line dan header diikuti dengan CRLF (Carriage Return yang diikuti dengan Line Feed).

13.1.2. Request Method

HTTP mendifinisikan 8 metode yaitu :

HEAD
: Meminta respon yang mirip dengan GET hanya saja tanpa dilanjutkan dengan badan pesan. Metode ini digunakan untuk mengambil informasi meta.

GET
: Meminta sumber daya yang spesifik, dan digunakan untuk mengakses halaman web.

POST
: Mensubmit suatu data untuk diproses. Data dimasukkan kedalam badan pesan

PUT
: Melakukan upload suatu resource ke suatu site

DELETE
: Menghapus suatu resource

TRACE
: Melakukan echo back terhadap suatu resource, sehingga client dapat melihat intermediate yang ada.

OPTIONS
: Mengembalikan metode HTTP dari server, digunakan untuk melihat resource dari suatu web server

CONNECT
: Digunakan untuk proxy apabila mengakses suatu site yang mendukung SSL

Safe Methods

Metode yang didefiniskan safe antara lain GET dan HEAD, digunakan hanya untuk pengambilan data dan tanpa melakukan perubahan disisi server. Metode yang didefinisikan unsafe antara lain POST, PUT dan DELETE, harus ditampilkan kepada pengguna dengan cara yang khusus, biasanya dalam bentuk tombol dan bukan link, dan dapat membuat pengguna lebih memperhatikan data yang akan dikirimkan.

13.1.3. Versi HTTP

HTTP berkembang mengikuti perkembangan perangkat lunak. Sehingga versi yang pernah digunakan adalah :

0.9
: Kadaluwarsa dan hanya mendukung metode GET

HTTP/1.0
: Protokol yang digunakan pertama kali digunakan

HTTP/1.1

Versi yang sekarang digunakan dengan mendukung persistent connection dan bekerja dengan proxy.

HTTP/1.2

Versi yang akan datang, sedang dikembangkan oleh W3C dan akan digunakan sebagai HTTP Extension Framework.

13.1.4. Kode Status (Code Status)

Reson pertama kali yang muncul pada saat mengakses suatu web dan digunakan sebagai kode informasi status yang digunakan pada client. Contoh kode status antara lain :

- Informasional (1xx), informasi yang digunakan untuk mengambil informasi
    - 100 Continue
    - 101 Switching protocol
- Sukses (2xx), akses yang berhasil
    - 200 OK
    - 201 Created
    - 202 Accepted
    - 203 Non-authoritative information
    - 204 No-Content
    - 205 Reset Content
    - 206 Partial Content
- Redirection (3xx), informasi ini memberitahukan kepada user agent untuk melakukan request tambahan supaya mencapai akses.
    - 300 Multiple choices
    - 301 Moved permanently
    - 302 Moved temporarily
    - 303 See Other
    - 304 Not Modified
    - 305 Use Proxy
- Client Error (4xx), terjadi kesalahan pada client
    - 400 Bad request
    - 401 Unauthorized
    - 402 Payment Required
    - 403 Forbidden
    - 404 Not found
    - 405 Method not allowed
    - 406 Not acceptable
    - 407 Proxy Authentication Required
    - 408 Request Timeout
    - 409 Conflict
    - 410 Gone
    - 411 Length Required
    - 412 Precondition failed
    - 413 Request entity too large
    - 414 Request URI too long
    - 415 Unsupported media

- Server error (5xx), informasi ini memberitahukan kepada client bahwa terjadi kesalahan di server.
    - 500 Internal server error
    - 501 Not implemented
    - 502 Bad Gateway
    - 503 Service unavailable
    - 504 Gateway timeout
    - 505 HTTP version not supported

13.1.5. Contoh

Berikut ini merupakan contoh komunikasi HTTP antar client dan server. Server berjalan di www.sample.com, dengan port 80.

Client request

```
GET /index.html HTTP/1.1
Host: www.example.com
```

Gambar 13.1 Client mengakses HTTP

Kemudian server akan memberikan respon

Server respon

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Server: Apache/1.3.27 (Unix)  (Red-Hat/Linux)
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Etag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
Content-Length: 438
Connection: close
Content-Type: text/html; charset=UTF-8
```

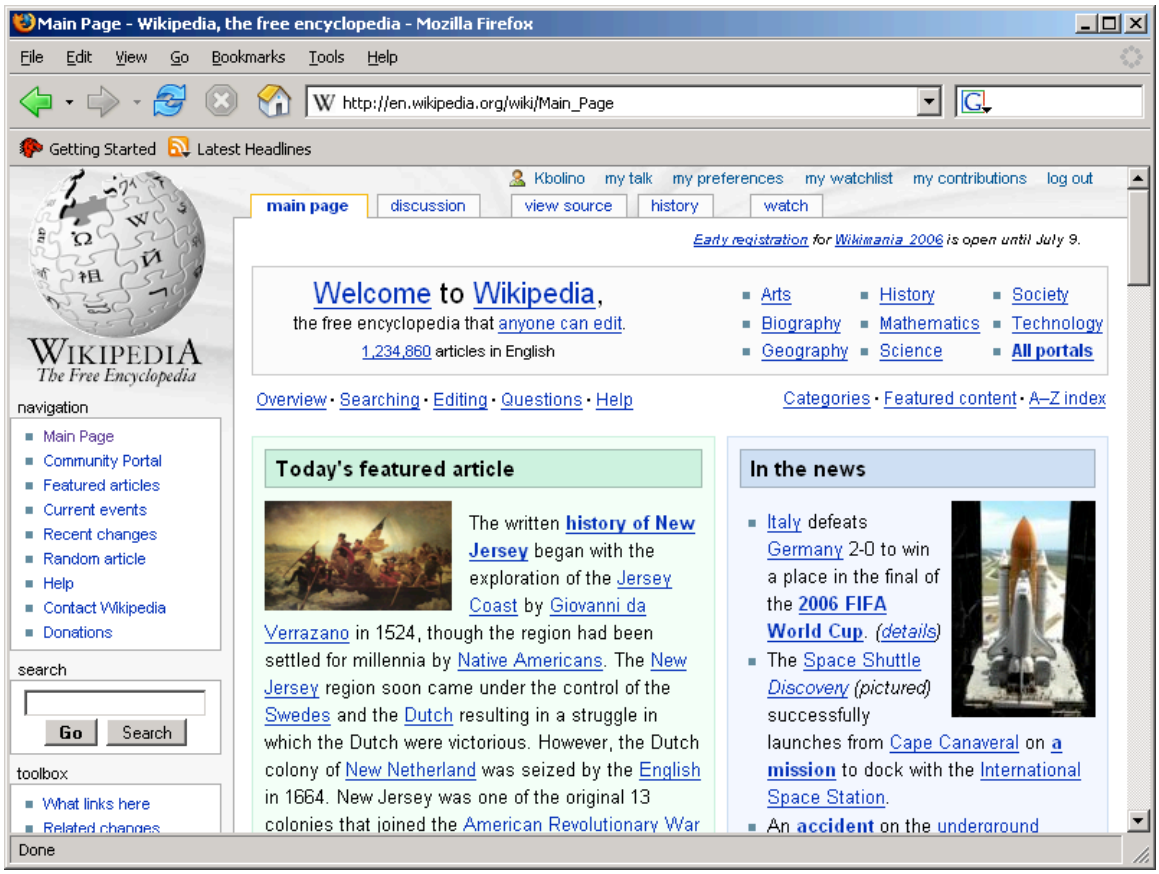
Gambar 13.2 Respon dari server

13.2. Web Browser

Web browser adalah aplikasi perangkat lunak yang membantu pengguna untuk dapat melakukan interaksi dengan tulisan, gambar dan informasi lainnya yang terdapat di suatu halaman web pada suatu website pada World Wide Web. Tulisan dan gambar dapat berupa hyperlink pada halaman lain pada website yang sama atau berbeda.

Web browser terdapat di personal computer dengan aplikasi Microsoft Internet Explorer, Mozilla Firefox, Appe Safari, Netscape dan Opera (rangking menurut survey 2006). Web browser merupakan HTTP user agent.

Web Browser berkomunikasi dengan menggunakan protokol HTTP pada suatu URL. Kebanyakan browser sudah mendukung protokol lainnya seperti FTP (File Transfer Protocol), RTSP (Real Time Sreaming Protocol) dan HTTPS (Versi HTTP yang mendukung enkripsi SSL).



Gambar 13.3 Contoh dari web browser (Mozilla-Firefox)

13.2.1. Sejarah

Tim Berners-Lee menggunakan NeXTcube sebagai aplikasi web server pertama kali pada tahun 1990, dan memperkenalkan pada CERN pada tahun 1991. Sehingga semenjak tahun tersebut pengembangan web browser semakin meningkat.

Web browser pertama adalah Silversmith, diciptakan oleh John Bottoms pada tahun 1987, menggunakan sistem SGML. Kemudian disusul oleh ViolaWWW yang berbasis HyperCard.

Perkembangan web browser meledak semenjak terciptanya NCSA Mozaic, yang merupakan web browser dengan GUI pertama kali, dikeluarkan pada September 1993. Marc Andreessen yang merupakan kepala proyek tersebut keluar dari NCSA dan membuat perusahaan dengan nama Netscape Communications Corporation.

Netscape mengeluarkan produk dengan nama Navigator pada tahun 1994, dan menguasai pasar dunia. Kemudian diikuti oleh Microsoft dengan mengeluarkan produk web browser dengan nama Internet Explorer, yang dibeli dari perusahaan Spyglass Inc. Hal ini yang menimbukan perang web browser, perang antara Microsoft dan Netscape.

Perang berlanjut dengan masing-masing perusahaan memberikan fitur-fitur tambahan seperti Cascading Style Sheet (CSS) dari Microsoft dan JavaScript Style Sheet(JSSS) dari Netscape. Kemudian Netscape semakin kalah dibandingkan dengan Microsoft dengan dalih penggunaan web browser yang sudah menjadi satu dengan sistem operasi OEM.

Akhirnya Netscape membuat produknya menjadi Open Source dengan membuat projek Mozilla. Perusahaan Netscape kemudian dibeli oleh America Online pada tahun 1998, hal ini manarik developer sehingga pada tahun 2002 mengeluarkan Mozilla 1.0. Projek ini semakin berkembang dan pada tahun 2004 keluar produk dengan nama Mozilla-Firefox dengan versi 1.0. Pada tahun 2005 keluar versi 1.5, versi 2 dijadwalkan akan keluar pada tahun 2006 dan sudah dipersiapkan produk Firefox 3. Sekarang Firefox merupakan web browser yang banyak digunakan, hampir 10% dari Traffik Internet.

Opera, web browser yang dapat dijalankan di perangkat genggam dan PC keluar pada tahun 1996.

Lynx merupakan web browser favorit bagi pengguna shell di unix.

Macintosh mengeluarkan Apple Safari yang merupakan web browser yang dikembangkan dari projek Konqueror. Safari digunakan pada sistem operasi Mac OS X.

13.2.2. Fitur

Standar web browser harus mendukung fasilitas sebagai berikut :

- HTTP dan HTTPS
- HTML, XML dan XHTML
- Format gambar termasuk GIF, PNG, JPEG, dan SVG
- Cascading Style Sheet (CSS)
- Java Script (Dynamic HTML) dan XMLHttpRequest
- Cookie
- Digital Certificate
- Favicons
- RSS, Atom

Sedangkan fitur fundamental yang harus didukung antara lain :

- Bookmark
- Caching dari isi web
- Mendukung media lain melalui plugin, contoh Macromedia Flash

Fasilitas tambahan seperti

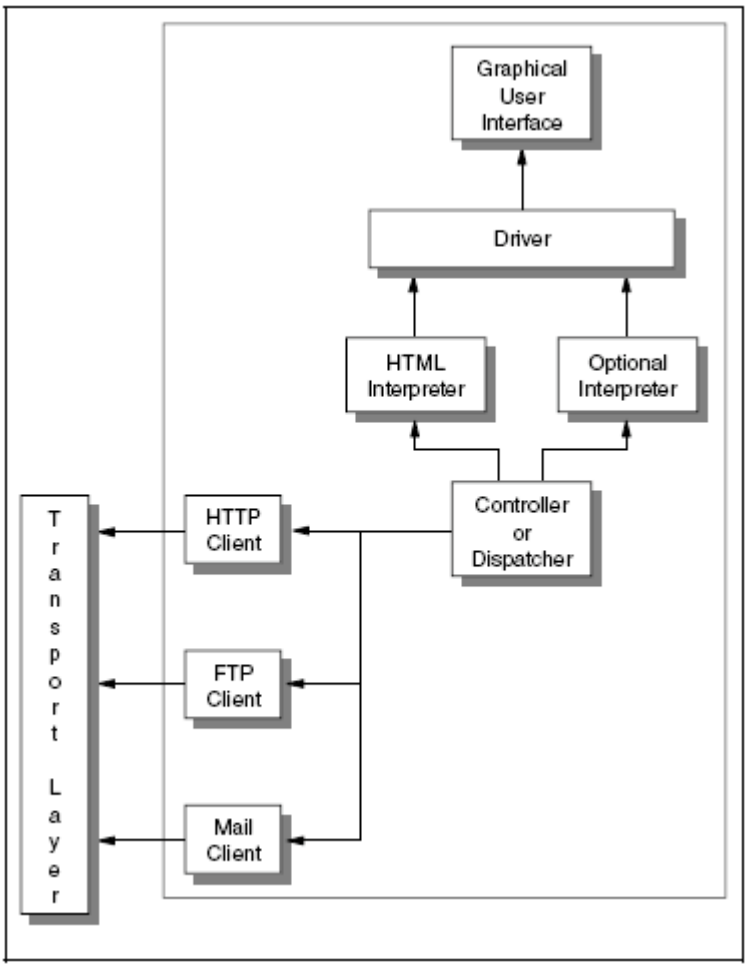
- Autocompletition dari URL
- Browsing secara Tabular
- Navigasi spasial
- Navigasi Caret
- Screen Reader

Fasilitas penghilang pengganggu

- Pop-Up advertisement
- Filter iklan
- Pertahanan terhadap phising

13.2.3. Struktur Web Browser

Secara keseluruhan web browser memiliki struktur seperti pada Gambar 13.4.



Gambar 13.4 Struktur Web Browser

13.3. Web Server

Pengertian web server dapat diartikan sebagai berikut :

1. Komputer yang memiliki tanggung jawab untuk menerima HTTP request dari client, yang biasanya menggunakan web browser dan melayani dalam bentuk halaman web, dimana biasanya berupa dokumen HTML dan objek link seperti gambar dll.
2. Program komputer yang melayani HTTP.

13.3.1. Fitur

Banyak program web server yang beredar, tetapi pada dasarnya memiliki fitur yang sama yaitu :

1. HTTP: merespon permintaan HTTP dan memberikan jawaban HTTP dengan memberikan dokumen HTML dan memberikan informasi kesalahan bila terjardi kesalahan.
2. Logging: web server memiliki fasilitas logging tentang informasi client yang melakukan request, respon yang diberikan oleh server, disimpan pada suatu file

log. Dari file log tersebut webmaster dapat membuat analisa statistik dengan menjalan aplikasi log analyzer.

Pada prakteknya webserver juga memberikan fasilitas lainnya yaitu :

1. Configurability, dapat dilakukan konfigurasi bahkan dengan aplikasi eksternal
2. Authentication, memberikan fasilitas authorisasi (meminta informasi username dan password), sebelum mengakses suatu atau semua resource
3. Dapat menangani tidak hanya konten static tetapi juga konten dinamik yang diberikan dari berbagai interface (SSI, CGI, SCGI, FastCGI, PHP, ASP, ASP.NET, ServerAPI, dll)
4. Mendukung Modular, memberikan fasilitas diluar program inti, dan ditempatkan dalam bentuk modular, sehingga server bisa memanggilnya apabila diperlukan.
5. HTTPS protokol HTTP dengan keamanan enkripsi dari SSL maupun TLS. Menggunakan koneksi pada port 443.
6. Kompresi terhadap konten dengan menggunakan fasilitas gzip, sehingga bisa mengurangi penggunaan bandwidth
7. Virtual Host, membentuk multi web server walau hanya menggunakan 1 alamat IP
8. Mendukung file dengan ukuran besar
9. Bandwidth Throttling, dapat mengatur penggunaan bandwidth terhadap pengakses

13.3.2. Tipe Konten

Konten yang diberikan oleh webserver dapat dikatakan :

- statik, apabila berasal dari file yang terdapat pada filesistem
- dinamik, apabila berasal dari suatu program atau script yang dipanggil oleh web server.

Memberikan layanan statik dapat diakses lebih cepat dari pada layanan dinamik, terlebih lagi bila konten tersebut harus mengakses database terlebih dahulu.

13.3.3. Translasi Path

Web server melakukan peralihan jalur dari URL menuju ke filesistem, dimana URL pada web server relatif terhadap direktori document root.

Contoh client mengakses suatu alamat

http://www.example.com/path/file.html

Web browser akan merubah menjadi HTTP/1.1 request

GET /path/file.html HTTP/1.1
Host : www.example.com

Web server pada www.example.com akan menambahkan path tersebut pada akar direktori. Pada mesin Unix biasanya terletak di /var/www/htdocs, sehingga menjadi

/var/www/htdocs/**path/file.html**

Web server akan membaca file tersebut, apabila file tersebut dapat ditemukan maka akan dikirimkan kepada client sebagai HTTP respon.

13.3.4. Konkuren (concurency)

Aplikasi program web server menggunakan teknik pemrograman konkuren. Bahkan dikombinasikan dengan finite state machine dan non-blocking I/O, untuk melayani permintaan HTTP.

13.3.5. Sejarah

Pada tahun 1998 Tim Berners-Lee mengusulkan kepada CERN (Pusat penelitian nuklir di Eropa) sebuah proyek dengan tujuan mempermudah pertukaran informasi antar peneliti dengan menggunakan sistem hypertext. Hasil dari proyek ini adalah 2 buah program, yaitu browser dengan nama WorldWideWeb dan Web server, yang jalan di mesin NeXTSTEP.

Gambar 13.5 Mesin webserver pertama

13.3.6. Perangkat Lunak

Top rangking program aplikasi web server adalah :

- Apache HTTP Server dari Apache Software Foundation
- Internet Information Services (IIS) dari Microsoft
- Sun Java System Web Server dari Sun Microsystem, dalam bentuk Sun ONE web server, iPlanet web server, dan Netscape Enterprice Server
- Zeus Web Server dari Zeus Technology

13.4. Konten

Web server melayani statik konten dan dinamik konten.

13.4.1. Konten Statik (Static Content)

Konten yang diambil secara langsung dari suatu file pada filesistem. Contoh dari konten statik antara lain :

- Hypertext Markup Language (HTML)

- Extensible Markup Language (XML)

13.4.2. Client-Side Dynamic Content

Fungsi dinamis dari aplikasi dijalankan disisi client. Contoh :

- Program dan Applet, contoh Java Applet yang berjalan menggunakan Java Virtual Machine (JVM).
- Java Script, merupakan komponen dinamis dari web browser

13.4.3. Server-Side Dynamic Content

Dengan mengakses fungsi yang terdapat di webserver sehingga memperoleh hasil yang sesuai request disebut dengan server-side dynamic content. Contoh :

- Common Gateway Interface (CGI), dengan menggunakan pemrogam PERL dapat dibuat aplikasi yang sesuai dengan keinginan client
- API dari webserver tertentu, contoh Netscape Server API (NSAPI), dan Microsoft internet Information Server API (ISAPI)
- Servlet, menjalankan aplikasi applet disisi server
- Server-Side Includes (SSI), digunakan oleh webserver yang mendukung teknologi JAVA sehingga dapat merubah beberapa bagian kecil dari HTML
- Java Server Page (JSP), mengenerate halaman HTML dari suatu aplikasi
- PHP Hypertext Preprocessor (PHP), aplikasi modular yang ditambahkan kepada webserver untuk membentuk suatu halaman HTML yang disesuaikan dengan input.

Bab 14. Manajemen Jaringan

Dengan berkembangnya jaringan TCP/IP yang sangat pesat, maka diperlukan juga suatu manajemen untuk mengatur jaringan.

Internet Architecture Board (IAB) merekomendasikan RFC 1052 yang berisikan tentang :

- Simple Network Management Protocol (SNMP)
- ISO Common Management Information Service / Common Management Information Protocol (CMIS / CMIP)

Dan IAB menyarankan untuk menggunakan SNMP.

14.1. Simple network Management Protocol (SNMP)

SNMP merupakan salah protokol resmi dari Internet Protocol suite yang dibuat oleh Internet Engineering Task Force (IETF). SNMP merupakan contoh dari layer 7 aplikasi yang digunakan oleh network management system untuk memonitor perangkat jaringan sehingga dapat memberikan informarsi yang dibutuhkan bagi pengelolanya.

14.1.1. Management Information Base (MIBs)

MIB merupakan database yang digunakan untuk manajemen perangkat pada jaringan. Database tersebut berisikan objek entiti dari perangkat jaringan (seperti router atau switch). Objek pada MIB didefinisikan menggunakan Abstract Syntax Notation One (ASN 1), dan diberi nama “Structure of Management Information Version 2 (SMIv2). Software yang digunakan untuk parsing disebut MIB compiler.

RFC yang membahas antara lain RFC1155 – Structure and identification of Management Information for TCP/IP base internets, RFC1213 – Management Information Base for Network Management of TCP/IP-based internets, dan RFC 1157 – A Simple Network Management Protocol.

SNMP, komunikasi yang terjadi antara management station (contoh: console) dengan management object (seperti router, gateway dan switch), menggunakan MIB. Component yang berkerja untuk mengambil data disebut SNMP agent, merupakan software yang dapat berkomunikasi dengan SNMP Manager.

14.1.2. Arsitektur SNMP

Framework dari SNMP terdiri dari :

Master Agent

Master agent merupakan perangkat lunak yang berjalan pada perangkat yang mendukung SNMP, dimana bertujuan untuk merespon permintaan dari SNMP dari **management station**. Master agent kemudian meneruskan kepada **subagent** untuk memberikan informasi tentang manajemen dengan fungsi tertentu.

Subagent
Subagent merupakan perangkat lunak yang berjalan pada perangkat yang mendukung SNMP dan mengimplementasikan MIB. Subagent memiliki kemampuan :

- Mengumpulkan informasi dari objek yang dimanaj
- Mengkonfigurasi informasi dari objek yang dimanaj
- Merespon terhadap permintaan manajer
- Membangkitkan alarm atau trap

Management Station
Management station merupakan client dan melakukan permintaan dan mendapakan trap dari SNMP server.

14.1.3. Protokol SNMP

PDU dari SNMP (versi 1) antara lain :

1. GET REQUEST – digunakan untuk mendapatkan informasi manajemen
2. GETNEXT REQUEST – digunakan secara iteratif untuk mendapatkan sekuen dari informasi manajemen
3. GET RESPONSE
4. SET – digunakan untuk melakukan perubahan terhadap subsistem
5. TRAP – digunakan untuk melakukan pelaporan terhadap subsistem manajemen

Untuk versi berikutnya ditambahkan PDU :

1. GETBULK REQUEST – iterasi yang lebih cepat untuk mendapatkan informasi
2. INFORM – acknowledge terhadap TRAP

SNMP menggunakan UDP pada port 161 untuk agent dan 162 untuk manager. Manager mengirimkan permintaan terhadap agent pada port 161 dan diterima pada manager pada port 162.

14.1.4. Perkembangan dan penggunaan

Version 1
RFC untuk SNMP, dikenal dengan nama Simple Network Management Protocol version 1, pada tahun 1988 :

- RFC 1065 – Structure and identification of management information for TCP/IP-based internets
- RFC 1066 – Management information base for network management of TCP/IP-based internets
- RFC 1067 – A Simple Network Management Protocol

Kemudian menjadi kadaluwarsa dengan digantikan dengan :

- RFC 1155 – Structure and identification of management information for TCP/IP-based internets
- RFC 1156 – Management information base for network management of TCP/IP-based internets
- RFC 1167 – A Simple Network Management Protocol

Versi 1 memiliki kelemahan pada sistem authentifikasi karena mengirimkan password secara plain text.

Version 2
Versi 2 ini banyak yang tidak menggunakan dikarenakan ketidak cocokan framework.

Simple Network Management Protocol version 2 (RFC 1441 – RFC 1452) dan juga dikenal sebagai SNMP v2. Diperkenalkan GETBULK sebagai alternatif dari GETNEXT. Dikenalkan juga Community-Based Simple Network Management Protocol version 2 atau yang disebut SNMP v2c sebagai pengganti sistem authentikasi
User-Based Simple Network Management Protocol version 2, atau SNMP v2u yang digunakan untuk memperbaiki keamanan dari SNMP v1.

Version 3
Versi ini didefinisikan pada RFC 3411 – RFC 3418 yaitu Simple Network Management Protocol version 3, dikeluarkan pada tahun 2004.

Pada prakteknya SNMP bisa menggunakan versi SNMPv1, SNMPv2c, atau SNMPv3. Dijabarkan pada RFC 3584 – Coexistence between Version 1, Version 2, and Version 3 of the Internet-Standard Network Management Framework.

Contoh Penggunaan

- Memonitoring waktu penggunaan suatu perangkat (sysUpTimeInstance)
- Inventory dari versi sistem operasi (sysDescr)
- Mengkoleksi informasi suatu interface (ifName, ifDescr, ifSpeed, ifType, ifPhysAddr)
- Mengukur throughput interface dari jaringan (ifInOctets, ifOutOctets)
- Menarik informasi cache dari ARP (ipNetToMedia)

14.1.5. Mengimplementasikan SNMP

snmpwalk

```
snmpwalk -c public punch system
SNMPv2-MIB::sysDescr.0 = STRING: Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-IO3-M), Version 12.2(15)T5,  RELEASE SOFTWARE (fc1)
TAC Support: http://www.cisco.com/tac
Copyright (c) 1986-2003 by cisco Systems, Inc.
Compiled Thu 12-Jun-03 15:49 by eaarm
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.9.1.187
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (835747999) 96 days, 17:31:19.99
SNMPv2-MIB::sysContact.0 = STRING: wikiuser
SNMPv2-MIB::sysName.0 = STRING: punch
SNMPv2-MIB::sysLocation.0 = STRING: test
SNMPv2-MIB::sysServices.0 = INTEGER: 78
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (0) 0:00:00.00
```

Gambar 14.1 Keluaran dari snmpwalk

Router Graphing Software
Banyak informasi yang bisa ditampilkan, misal performance, load dan error rate dari suatu jaringan seperti router atau switch. Kemudian dengan fungsi khusus, informasi yang didapat diolah menjadi dalam bentuk grafik.

Contoh aplikasi Multi Router Traffic Grapher dan Cacti

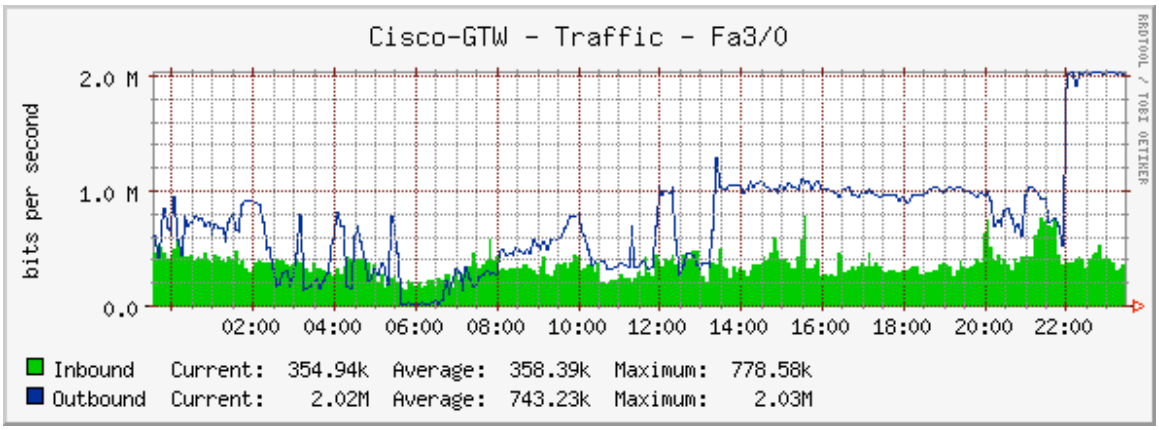
14.2. Multi Router Traffic Grapher

Multi Router Traffic Grapher atau yang disingkat MRTG adalah free software yang digunakan untuk memonitoring traffik load pada link jaringan. Dimana pengguna dapat melihat laporan dalam bentuk grafik.

MRTG ditulis dalam bentuk perl dan berjalan di UNIX/Linux dan juga pada sistem operasi Windows dan juga pada Netware. MRTG menggunakan lisensi Gnu GPL.

Gambar 14.2 Logo MRTG

Dikembangkan pertama kali oleh Tobias Oetiker dan Dave Rand, pertama kali digunakan untuk memonitoring router. Sekarang sudah dikembangkan untuk menjadi report berbagai macam. Informasi lengkap dapat dilihat di http://oss.oetiker.ch/mrtg/



Gambar 14.3 Contoh traffik MRTG

MRTG berkembang menjadi RRDTool, yaitu round-robin database tool. Penggunaan RRDTool dapat dikembangkan menjadi berbagai macam aplikasi contohnya cacti, JFFNms dan masih banyak lainnya.