BAB 2

LANDASAN TEORI DAN TINJAUAN PUSTAKA

2.1 Konsep Jaringan Komputer

Jaringan komputer (*computer network*) adalah sekumpulan komputer berjumlah banyak yang terpisah-pisah akan tetapi saling berhubungan dalam melaksanakan tugasnya, dengan kata lain jaringan komputer merupakan suatu himpunan interkoneksi (*interconected*) sejumlah komputer *autonomous* (Tanenbaum,2000). Jaringan komputer memungkinkan sejumlah komputer *autonomous* (tidak saling bergantung satu dengan yang lain) dapat saling bertukar data/informasi maupun menggunakan piranti *hardware* serta *software* secara bersama. Tiap komputer, printer atau periferal yang terhubung dengan jaringan disebut *node*. Sebuah jaringan komputer dapat memiliki dua, puluhan, ribuan atau bahkan jutaan *node*.

2.1.1 Komponen Jaringan Komputer

Menurut Agung(2002), secara umum komponen perangkat keras yang dibutuhkan dalam perancangan jaringan komputer adalah DTF (*Data Terminal Equipment*), DCE (*Data Communication Equipment*) dan media transmisi.

a. Terminal (*Data Terminal Equipment/DTF*)

   Terminal merupakan piranti jaringan yang memiliki kemampuan untuk menerima atau mengirimkan data dalam jaringan. Acuan agar piranti jaringan dapat saling berhubungan adalah alamat jaringan. Alamat jaringan akan dijelaskan lebih lanjut. Ada beberapa macam terminal dalam jaringan komputer, yaitu :

1. *Server*

   Suatu komputer (biasanya merupakan komputer *high-end* yang memiliki kemampuan tinggi) yang bertugas melayani permintaan dari *client*. Pada *server*, ter-*instal* aplikasi yang mengatur lalu lintas jaringan.

2. *Client/Workstation*

   Setiap komputer yang terhubung ke *server* dan dapat digunakan *user* untuk melakukan aktifitas tertentu. *Client* selain menjalankan program khusus yang disebut *shell* jaringan, juga bisa berkomunikasi dengan *server*, *client* yang lain maupun perangkat jaringan lainnya.

3. Piranti lain

   Piranti pendukung/tambahan yang dapat berfungsi sebagai *host* dan dapat mengakses sistem jaringan, semisal *printer* maupun *handphone*.

b. Media Komunikasi (*Data Communication Equipment*/DCE)

   Media komunikasi merupakan piranti tambahan yang dipasang di terminal untuk menghubungkan terminal dengan media transmisi. Contoh media komunikasi adalah *network interface card* dan *modem*. Selain itu diperlukan piranti komunikasi pendukung untuk membentuk jaringan komputer, terutama jaringan interkoneksi. Piranti pendukung ini antara lain: *hub*, *switch*, *repeater*, *bridge*, *router* maupun *gateway*. Penggunaan piranti sesuai dengan kebutuhan pengelolaan jaringan.

c. Media transmisi

Media transmisi merupakan piranti yang digunakan untuk melewatkan sinyal data. Media transmisi dapat menggunakan kabel maupun tanpa kabel (*wireless*).

2.1.2 Komunikasi data dan protokol

Prinsip dasar dari komunikasi data adalah melakukan pertukaran data antara 2 pihak melalui media transmisi tertentu. Tujuan dari komunikasi data adalah menyediakan peraturan (*rule*) dan ketentuan (*regulation*) yang memungkinkan komputer-komputer dengan sistem operasi, bahasa, pengkabelan dan lokasi yang berbeda dapat saling berbagi sumber daya(Agung,2002).

Komunikasi data dalam komputer terjadi berdasarkan suatu protokol. Protokol adalah seperangkat aturan-aturan yang mengatur proses komunikasi dalam jaringan komputer(Agung,2002). Walaupun 2 piranti saling terhubung tanpa adanya suatu protokol maka komunikasi tidak dapat dilakukan. Ada banyak protokol berdasar jaringan yang dikembangkan *vendor* tertentu antara lain: NetBios (IBM-Kompatibel), AppleTalk (Machintos), IPX/SPX (Novell Netware), NetBEUI (Microsoft) dan TCP/IP (UNIX dan *Internet*).

2.2 TCP/IP

TCP/IP (*Transmision Control Protocol/Internet Protocol*) merupakan serangkaian protokol di mana setiap protokol melakukan sebagian dari keseluruhan tugas komunikasi jaringan. Pengimplementasian jaringan memilih di antara protokol ini untuk mencapai fungsi

jaringan yang diinginkan (Agung,2002). Berkat prinsip ini, tugas masing-masing protokol menjadi jelas dan sederhana. Protokol yang satu tidak perlu mengetahui prinsip kerja protokol lain, sepanjang ia masih bisa saling mengirim dan menerima data.

TCP/IP merupakan protokol komunikasi jaringan yang paling fleksibel dan populer saat ini karena berbagai kelebihan yang dimiliki. Protokol ini dapat diterapkan dengan mudah di setiap jenis komputer dan *interface* jaringan, karena sebagian besar isi kumpulan protokol tidak spesifik pada satu komputer dan perlatan jaringan tertentu. Agar TCP/IP dapat berjalan di *interface* jaringan tertentu, hanya perlu dilakukan perubahan pada protokol yang berhubungan dengan *interface* jaringan saja (Purbo,1998).

TCP/IP telah menjadi standar komunikasi komputer *internetwork*. Hal ini tidak lepas dari berbagai kelebihan yang dimiliki. Pengalamatan yang unik namun bersifat global, banyak layanan yang disediakan, dan memiliki fasilitas *routing* menjadikan TCP/IP dapat diterapkan pada *internetwork*.

TCP/IP berkembang atas dasar konsensus/tidak tergantung *vendor* serta bersifat independen dan praktis terhadap perangkat keras apapun. Protokol ini menggunakan standar terbuka dalam bentuk *Request For Comments* (RFC) yang dapat diambil siapapun tanpa biaya. Perkembangan *internet* mendukung protokol TCP/IP menjadi standar umum yang dipakai dalam komunikasi data komputer sekarang. *Internet* terbentuk berdasar kesepakatan teknis secara umum melalui standar dari IETF (*Internet Enginering Task Force*).

2.2.1 Sejarah TCP/IP dan Internet

Menurut Purbo(1998), cikal bakal protokol TCP/IP berawal dari pengembangan jaringan ARPANET (1969-1972) oleh DARPA (*Defence Advance Research Project Agency*) yang menyangkut didalamnya pengembangan jaringan komunikasi data antar komputer dan protokol komunikasi yang digunakan. Protokol komunikasi ARPANET semula menggunanken NCP (*Network Communication Protocol*), namun protokol ini tidak dapat menampung *node* komputer dalam jumlah besar.

DARPA mengembangkan protokol komunikasi yang lebih umum, dinamakan TCP/IP. ARPANET menggunakan TCP/IP pada tahun 1983. Perusahaan Bolt Beranek Newman (BBN) membuat protokol TCP/IP berjalan di atas komputer dengan sistem operasi UNIX. Selanjutnya, TCP/IP berkembang sangat pesat.

2.2.2 Arsitektur TCP/IP

Arsitektur protokol merepresentasikan bagaimana struktur dari *hardware* maupun *software* komputer yang mendukung pertukaran data di antara sistem dan mendukung aplikasi distribusi (Stalling,2001). Arsitektur protokol menjadi dasar dari perkembangan protokol jaringan.

Pada tahun 1977, *International Standard Organization* (ISO) mengeluarkan standar arsitektur protokol berupa OSI *Reference Model*. OSI sendiri merupakan singkatan dari *Open System Interconnection*. Alasan pasti untuk pengembangan model OSI dimaksudkan untuk menyediakan suatu kerangka kerja bagi standarisasi pengembangan protokol jaringan.

Untuk arsitektur TCP/IP, standar dikembangkan oleh *Internet Engineering Task Force* (IETF). Arsitektur ini biasa disebut DARPA *Reference Model*(id.wikipedia.org, 2007). Sejak tahun 1990-an, model DARPA *Reference Model* menjadi arsitektur komersial yang dominan serta sebagai *protocol suite* yang terbanyak mewujudkan pengembangan protokol baru.

Arsitektur DARPA *Reference Model* terdiri dari 5 lapisan, yaitu :

a. *Application Layer* : Menyediakan komunikasi diantara proses dan aplikasi pada *host* terpisah. Pada layer inilah terletak semua aplikasi yang menggunakan protokol TCP/IP ini. Contoh aplikasi yang dimiliki antara lain : DHCP (*Dynamic Host Configuration Protocol*) yaitu protokol untuk distribusi IP pada jaringan dengan jumlah IP yang terbatas, TFTP (*Trivial File Transfer Protocol*) yaitu Protokol untuk transfer *file*, dsb.
b. *Transport Layer* : Menyediakan layanan transfer data ujung-ke-ujung. Lapisan ini meliputi mekanisme-mekanisme keandalan. Protokol inilah yang menyediakan komunikasi antara dua *host*/komputer. Kedua protokol itu adalah TCP dan UDP.
c. *Internet Layer* : Berkaitan dengan *routing* data dari sumber ke *host* tujuan melewati satu jaringan atau lebih yang dihubungkan melalui *router*. Protokol inilah yang bertanggung jawab dalam proses pengiriman paket ke alamat yang tepat. Pada layer ini terdapat 3 protokol yaitu

IP, ARP da ICMP. IP (*Internet Protocol*) berfungsi menyampaikan paket data ke alamat yang tepat. ARP (*Address Resolution Protocol*) ialah protokol yang digunakan menemukan alamat *hardware* dari *host*/komputer yang terletak pada network yang sama. Sedangkan ICMP (*Internet Control Message Protocol*) ialah protokol yang bertanggung jawab mengirimkan pesan dan melaporkan kegagalan pengiriman.

d. *Network Interface Layer* : Berkaitan dengan *logical interface* diantara suatu ujung sistem dan jaringan. Layer yang bertanggaung jawab mengirim dan menerima data ke dan dari media fisik. Media fisiknya dapat berupa kabel, serat optik atau gelombang radio. Protokol ini harus mampu menterjemahkan sinyal listrik menjadi data digital yang dimengerti komputer(Purbo,1998). Contoh protokol ini adalah *Ethernet* (IEEE *Standard* 802.3) yaitu protokol yang menangani komposisi paket data yang menggunakan metode akses CSMA/CD, FDDI (*Fiber Distributed Data Interface*) yaitu protokol yang menghubungkan 2 atau lebih jaringan pada jarak yang jauh dimana metode akses yang digunakan adalah metode *token passing*, dsb.

Pada prinsipnya, model referensi OSI(*OSI Reference Model*)) dan TCP/IP(DARPA *Reference Model*) memiliki banyak persamaan. Keduanya didasarkan pada konsep tumpukkan (*stack*) protokol yang saling bergantung. Tabel 2.1 berikut menunujukkan perbandingan hubungan

arsitektur model referensi OSI dengan model referensi TCP/IP sebagai protokol standar *internet*.

Tabel 2.1 Perbandingan hubungan arsitektur model OSI dengan model TCP/IP (Sumber : Prihanto,2003)

No	Model OSI	Model TCP/IP
7	*Application*	*Application*
6	*Presentation*	
5	*Session*	
4	*Transport*	*Transport*
3	*Network*	*Internet*
2	*Data Link*	*Network Interface*
1	*Physic*	

2.2.3 Protokol Lapisan Transport

Pada lapisan *transport* terdapat dua buah protokol *end-to-end* yaitu TCP dan UDP.

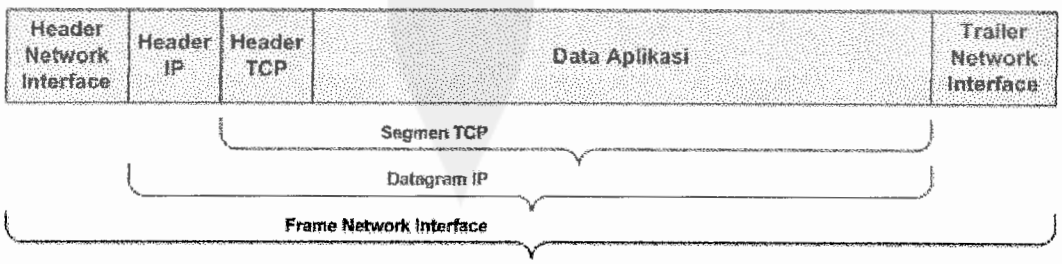
2.2.3.1 TCP

Transmission Control Protocol (TCP) adalah suatu protokol yang berada di lapisan *transport* (baik itu dalam tujuh lapis model referensi OSI atau DARPA *Reference Model*) yang berorientasi sambungan (*connection-oriented*) dan dapat diandalkan (*reliable*). Prinsip pertukaran data dalam TCP diawali dari memastikan koneksi ke *destination* terlebih dahulu. Hasan(2005) menerangkan bahwa TCP mengijinkan sebuah aliran *byte* yang berasal dari satu mesin untuk dikirimkan tanpa kesalahan ke sebuah mesin yang ada di *internet*. TCP memecah aliran memecah alira *byte* data menjadi pesan-pesan diskret dan meneruskan ke lapisan *internet*. Pada mesin tujuan, proses TCP penerima merakit kembali pesan-pesan yang diterima menjadi sebuah output. TCP juga menangani pengendalian aliran

untuk memastikan bahwa pengirim yang cepat tidak akan membanjiri pesan-pesan yang akan diterima oleh penerima yang lambat.

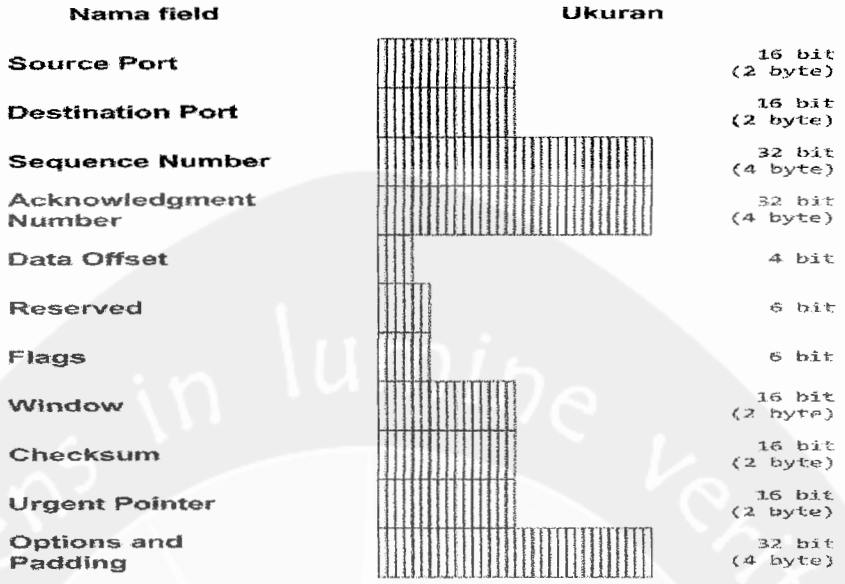
IP diterapkan pada seluruh sistem, bertindak sebagai *relay* untuk memindahkan satu blok data dari *host*, melewati satu *router* atau lebih, menuju *host* lain. TCP hanya diterapkan pada *end-system* dan menjaga *track* satu blok data untuk memastikan bahwa semuanya dikirim menuju aplikasi yang tepat secara *reliabel* (Stalling,2001).

TCP akan mengirimkan data dalam suatu segmen. Sebuah segmen TCP terdiri atas sebuah *header* dan segmen data (*payload*), yang dienkapsulasi dengan menggunakan *header* IP dari protokol IP. Konfigurasi suatu segmen dapat dilihat pada gambar 2.1. Ukuran dari *header* TCP adalah bervariasi, yang terdiri atas beberapa *field* yang ditunjukkan dalam gambar 2.2. Sebuah segmen dapat berukuran hingga 65495 *byte*: 2^{16}-(ukuran *header* IP terkecil (20 *byte*)+ukuran *header* TCP terkecil (20 byte)). Data IP tersebut akan dienkapsulasi lagi dengan menggunakan header protokol *network interface* (lapisan pertama dalam DARPA Reference Model) menjadi *frame* lapisan *Network Interface*(id.wikipedia.org,2007).



Gambar 2.1 Data TCP

(Sumber : id.wikipedia.org,2007)



Gambar 2.2 *Header* TCP

(Sumber : id.wikipedia.org,2007)

Suatu *header* IP dari sebuah segmen TCP, *field Source IP Address* diatur menjadi alamat *unicast* dari sebuah antarmuka host yang mengirimkan segmen TCP yang bersangkutan. Sementara itu, *field Destination* IP *Address* juga akan diatur menjadi alamat *unicast* dari sebuah antarmuka *host* tertentu yang dituju. Hal ini dikarenakan, protokol TCP hanya mendukung transmisi *one-to-one*.

Komunikasi data TCP memiliki karakteristik-karakteristik berikut:

a. Berorientasi sambungan (*connection-oriented*): Sebelum data dapat ditransmisikan antara dua *host*, dua proses yang berjalan pada lapisan aplikasi harus melakukan negosiasi untuk membuat sesi koneksi terlebih dahulu. Koneksi TCP ditutup dengan menggunakan proses terminasi koneksi TCP (TCP *connection termination*).

b. *Full-duplex*: Untuk setiap *host* TCP, koneksi yang terjadi antara dua *host* terdiri atas dua buah jalur, yakni jalur keluar dan jalur masuk. Dengan menggunakan teknologi lapisan yang lebih rendah yang mendukung *full-duplex*, maka data pun dapat secara simultan diterima dan dikirim. *Header* TCP berisi nomor urut (TCP *sequence number*) dari data yang ditransmisikan dan sebuah *acknowledgment* dari data yang masuk.

c. Dapat diandalkan (*reliable*): Data yang dikirimkan ke sebuah koneksi TCP akan diurutkan dengan sebuah nomor urut paket dan akan mengharapkan paket *positive acknowledgment* dari penerima. Jika tidak ada paket *Acknowledgment* dari penerima, maka segmen TCP (protocol data unit dalam protokol TCP) akan ditransmisikan ulang. Pada pihak penerima, segmen-segmen duplikat akan diabaikan dan segmen-segmen yang datang tidak sesuai dengan urutannya akan diletakkan di belakang untuk mengurutkan segmen-segmen TCP. Untuk menjamin integritas setiap segmen TCP, TCP mengimplementasikan penghitungan TCP *Checksum*.

d. *Byte stream*: TCP melihat data yang dikirimkan dan diterima melalui dua jalur masuk dan jalur keluar TCP sebagai sebuah *byte stream* yang berdekatan (kontigu). Nomor urut TCP dan nomor *acknowlegment* dalam setiap *header* TCP didefinisikan juga dalam bentuk *byte*. Meski demikian, TCP tidak mengetahui batasan pesan-pesan di dalam *byte stream* TCP tersebut. Untuk melakukannya, hal ini diserahkan kepada protokol lapisan aplikasi (dalam DARPA

Reference Model), yang harus menerjemahkan *byte stream* TCP ke dalam "bahasa" yang dipahami.

e. Memiliki layanan *flow control*: Untuk mencegah data terlalu banyak dikirimkan pada satu waktu, yang akhirnya membuat "macet" jaringan *internetwork* IP, TCP mengimplementasikan layanan *flow control* yang dimiliki oleh pihak pengirim yang secara terus menerus memantau dan membatasi jumlah data yang dikirimkan pada satu waktu. Untuk mencegah pihak penerima untuk memperoleh data yang tidak dapat disangganya (*buffer*), TCP juga mengimplementasikan *flow control* dalam pihak penerima, yang mengindikasikan jumlah *buffer* yang masih tersedia dalam pihak penerima.

f. Melakukan segmentasi terhadap data yang datang dari lapisan aplikasi (dalam *DARPA Reference Model*)

g. Mengirimkan paket secara "*one-to-one*": hal ini karena memang TCP harus membuat sebuah sirkuit logis antara dua buah protokol lapisan aplikasi agar saling dapat berkomunikasi. TCP tidak menyediakan layanan pengiriman data secara *one-to-many*(id.wikipedia.org,2007).

TCP umumnya digunakan ketika protokol lapisan aplikasi membutuhkan layanan transfer data yang bersifat andal, yang layanan tersebut tidak dimiliki oleh protokol lapisan aplikasi tersebut.

TCP mampu mengindikasikan sebuah lokasi tertentu untuk menyampaikan segmen-segmen TCP yang dikirimkan yang diidentifikasi dengan menggunakan TCP *Port Number*. Nomor-nomor di bawah angka 1024 merupakan port yang

umum digunakan dan ditetapkan oleh *IANA (Internet Assigned Number Authority)*. Contoh *port* TCP antara lain dapat dilihat pada tabel 2.2.

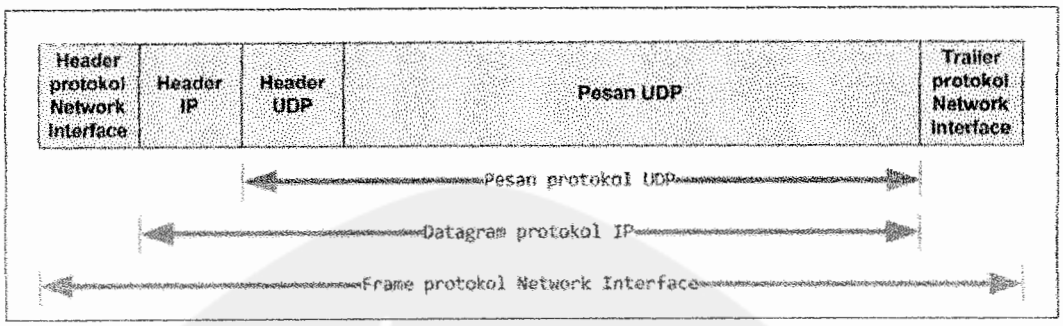
Tabel 2.2 *Port* TCP(Sumber : id.wikipedia.org,2007)

Nomor port TCP	Keterangan
20	*File Transfer Protocol*/FTP (digunakan untuk saluran data)
21	*File Transfer Protocol*/FTP (digunakan untuk saluran kontrol)
25	*Simple Mail Transfer Protocol*/SMTP yang digunakan untuk mengirim e-mail
23	Telnet
80	*Hypertext Transfer Protocol*/HTTP yang digunakan untuk World Wide Web.
110	*Post Office Protocol* 3/POP3 yang digunakan untuk menerima e-mail.
139	NetBIOS over TCP *session service*

2.2.3.2 UDP

User Datagram Protocol (UDP) adalah salah satu protokol lapisan *transport* TCP/IP yang mendukung komunikasi yang tidak andal (*unreliable*), tanpa koneksi (*connectionless*) antara host-host dalam jaringan yang menggunakan TCP/IP. Protokol yang digunakan bagi aplikasi-aplikasi yang tidak memerlukan pengurutan seperti pada TCP. UDP juga digunakan secara meluas pada *query* dan aplikasi *client/server* jenis *request-reply*, dimana pengiriman yang cepat lebih diutamakan dibanding pengiriman yang akurat.

UDP, berbeda dengan TCP yang memiliki satuan paket data yang disebut dengan segmen, UDP melakukan pengepakan terhadap data ke dalam pesan-pesan UDP (*UDP Messages*). Konfigurasi data UDP dapat dilihat pada gambar 2.3.

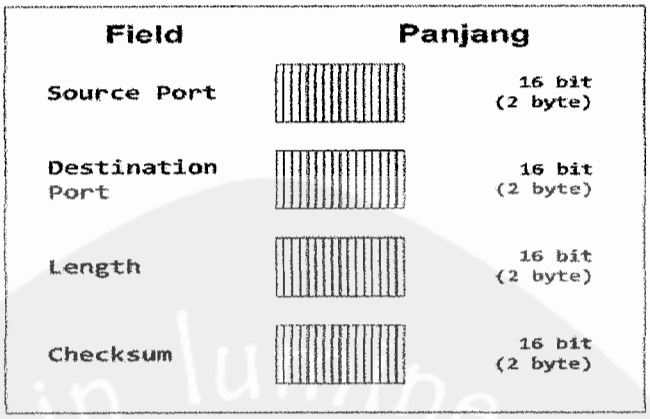


Gambar 2.3 Data UDP

(Sumber : id.wikipedia.org,2007)

Sebuah pesan UDP akan dikirimkan ke protokol lapisan selanjutnya (lapisan *internetwork*) setelah mengepaknya menjadi *datagram* IP. Enkapsulasi terhadap pesan-pesan UDP oleh protokol IP dilakukan dengan menambahkan *header* IP dengan protokol IP nomor 17 (0x11). Pesan UDP dapat memiliki besar maksimum 65507 *byte*: 65535 (2^{16})-20 (ukuran terkecil dari header IP)-8 (ukuran dari *header* UDP) *byte*. *Datagram IP* yang dihasilkan dari proses enkapsulasi tersebut, akan dienkapsulasi kembali dengan menggunakan *header* dan *trailer* protokol lapisan *Network Interface* yang digunakan oleh host tersebut.

Header IP dari sebuah pesan UDP, *field Source* IP *Address* akan diset ke antarmuka *host* yang mengirimkan pesan UDP yang bersangkutan; sementara *field Destination* IP *Address* akan diset ke alamat IP *unicast* dari sebuah *host* tertentu, alamat IP *broadcast*, atau alamat IP *multicast*. *Header* UDP diwujudkan sebagai sebuah *header* dengan 4 buah *field* memiliki ukuran yang tetap, seperti tersebutkan dalam gambar 2.4.



Gambar 2.4 *Header* UDP
(Sumber : id.wikipedia.org,2007)

Komunikasi data UDP memiliki karakteristik-karakteristik berikut:

a. *Connectionless* (tanpa koneksi): Pesan-pesan UDP akan dikirimkan tanpa harus dilakukan proses negosiasi koneksi antara dua host yang hendak berukar informasi.
b. *Unreliable* (tidak andal): Pesan-pesan UDP akan dikirimkan sebagai *datagram* tanpa adanya nomor urut atau pesan *acknowledgment*. Protokol lapisan aplikasi yang berjalan di atas UDP harus melakukan pemulihan terhadap pesan-pesan yang hilang selama transmisi. Umumnya, protokol lapisan aplikasi yang berjalan di atas UDP mengimplementasikan layanan keandalan mereka masing-masing, atau mengirim pesan secara periodik atau dengan menggunakan waktu yang telah didefinisikan.
c. UDP menyediakan mekanisme untuk mengirim pesan-pesan ke sebuah protokol lapisan aplikasi atau proses tertentu di dalam sebuah *host* dalam jaringan yang menggunakan TCP/IP. *Header* UDP

berisi *field Source Process Identification* dan *Destination Process Identification.*

d. UDP menyediakan penghitungan *checksum* berukuran 16-*bit* terhadap keseluruhan pesan UDP.

Komunikasi data UDP tidak menyediakan layanan-layanan antar-*host* berikut:

a. UDP tidak menyediakan mekanisme penyanggaan (*buffering*) dari data yang masuk ataupun data yang keluar. Tugas *buffering* merupakan tugas yang harus diimplementasikan oleh protokol lapisan aplikasi yang berjalan di atas UDP.

b. UDP tidak menyediakan mekanisme segmentasi data yang besar ke dalam segmen-segmen data, seperti yang terjadi dalam protokol TCP. Karena itulah, protokol lapisan aplikasi yang berjalan di atas UDP harus mengirimkan data yang berukuran kecil (tidak lebih besar dari nilai *Maximum Transfer Unit/MTU*) yang dimiliki oleh sebuah antarmuka di mana data tersebut dikirim. Karena, jika ukuran paket data yang dikirim lebih besar dibandingkan nilai MTU, paket data yang dikirimkan bisa saja terpecah menjadi beberapa fragmen yang akhirnya tidak jadi terkirim dengan benar.

c. UDP tidak menyediakan mekanisme *flow-control*, seperti yang dimiliki oleh TCP.

Seperti halnya TCP, UDP juga memiliki saluran untuk mengirimkan informasi antar *host*, yang disebut dengan UDP *Port*. Untuk menggunakan protokol UDP, sebuah aplikasi harus menyediakan alamat IP dan nomor UDP *Port* dari *host* yang dituju. Sebuah UDP *port* berfungsi sebagai sebuah *multiplexed message queue*, yang berarti

bahwa UDP *port* tersebut dapat menerima beberapa pesan secara sekaligus. Setiap *port* diidentifikasi dengan nomor yang unik, seperti halnya TCP, tetapi meskipun begitu, UDP *Port* berbeda dengan TCP *Port* meskipun memiliki nomor *port* yang sama. Tabel 2.3 menampilkan beberapa UDP *port* yang telah dikenal secara luas.

Tabel 2.3 *Port* UDP(Sumber : id.wikipedia.org,2007)

Nomor Port UDP	Keterangan
53	*Domain Name System* (DNS) *Name Query*
67	BOOTP *client* (*Dynamic Host Configuration Protocol* /DHCP)
68	BOOTP *server* (DHCP)
69	*Trivial File Transfer Protocol* (TFTP)
137	NetBIOS *Name Service*
138	*NetBIOS Datagram Service*
161	*Simple Network Management Protocol* (SNMP)
445	*Server Message Block* (SMB)
520	*Routing Information Protocol* (RIP)
1812/1813	*Remote Authentication Dial-In User Service* (RADIUS)

Protokol UDP sering digunakan dalam beberapa tugas berikut:

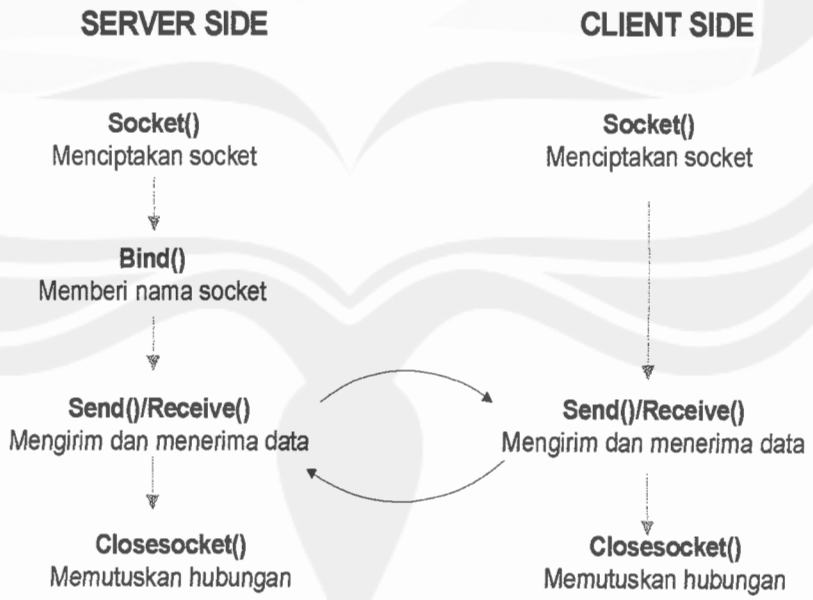
a. Protokol yang "ringan" (*lightweight*): Untuk menghemat sumber daya memori dan prosesor, beberapa protokol lapisan aplikasi membutuhkan penggunaan protokol yang ringan yang dapat melakukan fungsi-fungsi spesifik dengan saling bertukar pesan. Contoh dari protokol yang ringan adalah fungsi *query* nama dalam protokol lapisan aplikasi *Domain Name System*.

b. Protokol yang tidak membutuhkan keandalan. Contoh protokol ini adalah protokol *Routing Information Protocol* (RIP).

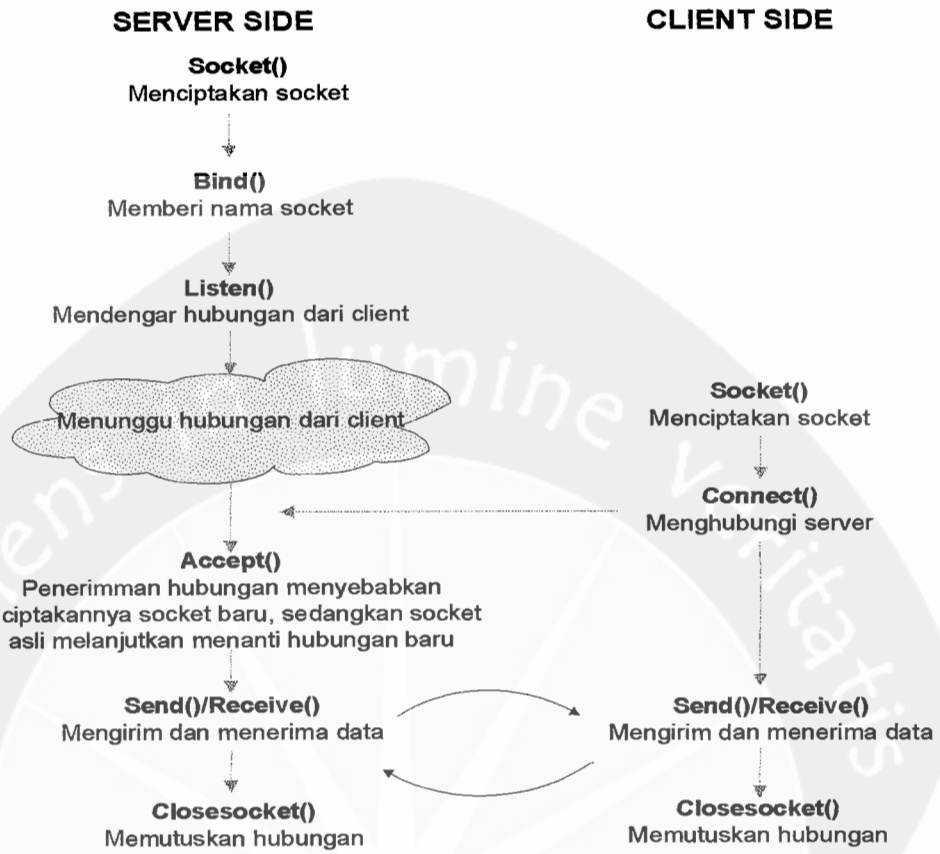
c. Transmisi *broadcast*: Karena UDP merupakan protokol yang tidak perlu membuat koneksi terlebih dahulu dengan sebuah *host* tertentu, maka transmisi broadcast pun dimungkinkan. Sebuah protokol lapisan aplikasi dapat mengirimkan paket data ke beberapa tujuan dengan menggunakan alamat *multicast* atau *broadcast*. Hal ini kontras dengan protokol TCP yang hanya dapat mengirimkan transmisi *one-to-one*. Contoh: *query* nama dalam protokol NetBIOS *Name Service*.

2.2.3.3 Perbandingan TCP dan UDP

Proses pembuatan *socket* antara TCP dan UDP sedikit berbeda satu sama lain. Gambar 2.5 dan gambar 2.6 menunjukkan perbedaan proses pembuatan *socket* keduanya.



Gambar 2.5 Diagram alir pembuatan *socket* UDP



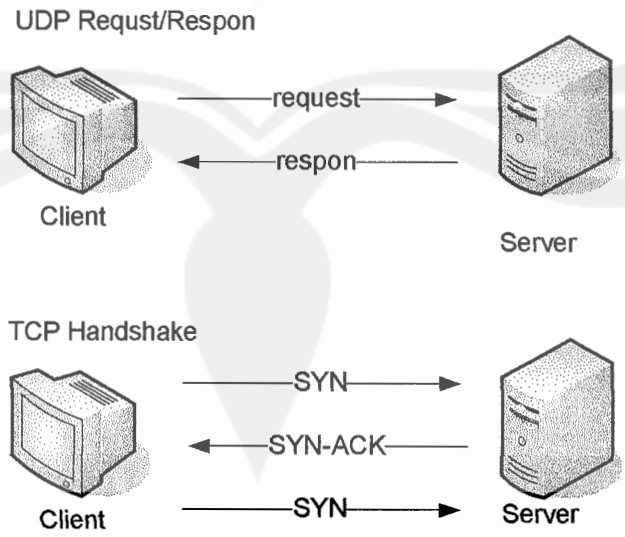
Gambar 2.6 Diagram alir pembuatan *socket* TCP

TCP lahir sebagai sebuah teknologi yang berdasar pada koneksi(*connection oriented*), menjaga sesi aplikasi, dan menjamin tingkat komunikasi yang handal, pengiriman paket yang hilang dan memiliki standarisasi pada tingkat *flow control*. UDP tidak menjaga sesi koneksi (*connectionless*) dan paket yang dikirimkan bersifat independen dari satu paket ke paket yang lain sehingga urutan paket yang datang bukanlah sebuah jaminan dari protokol ini. Hal yang menarik lagi dari dukungan sekuritas, UDP tidak menawarkan mekanisme sekuritas secara *built-in*. UDP unggul dalam suatu keadaan jaringan yang membutuhkan performa tetapi

dengan ketahanan yang tidak terlalu tinggi (Adnan, 2007).

UDP memang memberi keuntungan dari sisi efisiensi *bandwith*, dukungan klien yang lebih banyak, *latency* yang lebih rendah, serta performa yang tinggi adalah ciri-ciri UDP. Walaupun pengembangan aplikasi UDP dapat dikatakan sederhana, implementasi untuk mencapai keuntungan tersebut memerlukan pemikiran dan desain yang baik. Pengiriman protokol UDP dilakukan tanpa *blocking* dengan kata lain protokol UDP tidak memiliki *flow control* dan kewajiban *respons* setiap paket yang dikirimkan.

Hal lain yang mungkin menjadi pertimbangan pemilihan antara UDP dan TCP adalah kecepatan. Pada TCP dibutuhkan sekurang kurangnya tiga pertukaran paket(*Three Way Handshake*). Sedangkan pada UDP, pertukaran data bersifat *request-respon*. Perbandingan pertukaran paket UDP dan TCP terlihat seperti pada gambar 2.7.



Gambar 2.7 Perbandingan pertukaran paket

Proses pertukaran paket TCP dapat digambarkan sebagai berikut:

a. *Client* (yang ingin membuat koneksi) akan mengirimkan sebuah segmen TCP dengan *flag* SYN diaktifkan kepada *server* (yang hendak diajak untuk berkomunikasi). SYN singkatan dari *synchronization*, yang digunakan untuk memberitahukan komputer tujuan suatu permintaan koneksi.
b. *Server* akan mengirimkan suatu paket SYN-ACK sebagai jawaban bisa tidaknya koneksi dilakukan. ACK merupakan singkatan *acknowledgment*.
c. Bila koneksi diterima, *client* selanjutnya akan mulai saling bertukar data dengan *server*.

Hal ini tentunya menunjukkan bahwa pada suatu waktu (terutama bila paket TCP yang dikirimkan kecil) timbul *overhead* yang tidak perlu. Hal ini dapat dihindarkan dengan menggunakan UDP. Hal lain tentang perbedaan TCP dan UDP adalah urutan paket. Pada aplikasi tertentu urutan paket menjadi penting sebagai contoh aplikasi transfer berkas tetapi pada kasus tertentu paket yang hilang atau tidak urut tidak terlalu berpengaruh pada aplikasi *video streaming* yang mendukung *frame skipping*.

Salah satu fitur utama dari UDP dibanding TCP adalah kemampuan untuk mengirim sebuah pesan ke banyak penerima. Pengiriman pesan dapat berupa *broadcasting* yaitu pengiriman pesan kesemua *host* dalam sebuah *subnet*.

Dari penjelasan diatas, TCP dan UDP mengalami revolusi yang berbeda (atau bahkan berlawanan). Perbandingan karakteristik pertukaran data antara TCP dan UDP dapat dilihat pada tabel 2.4.

Tabel 2.4 Perbandingan TCP dan UDP

TCP	UDP
connection oriented	*connectionless*
melakukan *flow control*	tanpa *flow control*
mengutamakan kehandalan (*reliable*), keamanan, fleksibilitas, dan juga kualitas	mengutamakan kecepatan (berorientasi waktu) atau pengiriman periodik, tidak aman
pengurutan paket	tanpa pengurutan
pengiriman paket yang hilang	tanpa pengiriman paket yang hilang
tanpa duplikasi paket	mungkin terjadi duplikasi
tak mendukung *broadcast* dan *multicast*	mendukung *broadcast* dan *multicast*
mendukung *multiserver* dan *multiclient*	tak mendukung *multiserver* dan *multiclient*
overhead tinggi	*overhead* rendah, efisien *bandwith*

2.2.4 Pengalamatan Jaringan

IP merupakan alamat unik dari suatu jaringan (*network*) dan digunakan sebagai identitas baik pengirim maupun penerima paket data. Salah satu komponen penting dalam pengiriman data adalah IP. Jadi agar komunikasi dapat berhasil, setiap entitas dalam sistem harus memiliki alamat khusus.

Pengalamatan IP sangat tergantung dari versi protokol IP yang digunakan dalam jaringan. Alamat IP versi 4 (sering disebut dengan Alamat IPv4) adalah sebuah jenis pengalamatan jaringan yang digunakan di dalam protokol jaringan TCP/IP yang menggunakan

protokol IP versi 4. Panjang totalnya adalah 32-bit, dan secara teoritis dapat mengalamati hingga 4 miliar host komputer di seluruh dunia. Contoh alamat IP versi 4 adalah 192.168.0.3 (id.wikipedia.org, 2007).

Secara umum alamat-alamat IPv4 dibagi menjadi 2 bagian :

a. Bagian *net id* yang menunjukkan jaringan kemana *host* dihubungkan
b. Bagian *host id* sebagai pengenal unik pada setiap *host* dalam jaringan.

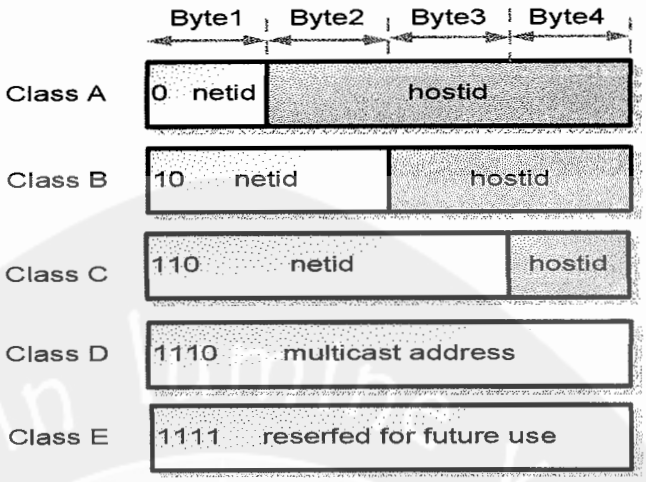
Untuk memudahkan identifikasi alamat IP dituliskan sebagai 4 nilai numerik 8 bit(nilai maksimum 255).

Dalam RFC 791, alamat IP versi 4 dibagi ke dalam beberapa kelas, dilihat dari oktet pertamanya, seperti terlihat pada tabel 2.5, gambar 2.8 dan gambar 2.9. Sebenarnya yang menjadi pembeda kelas IP versi 4 adalah pola biner yang terdapat dalam oktet pertama (utamanya adalah bit-bit awal/*high-order bit*), tapi untuk lebih mudah mengingatnya, akan lebih cepat diingat dengan menggunakan representasi desimal.

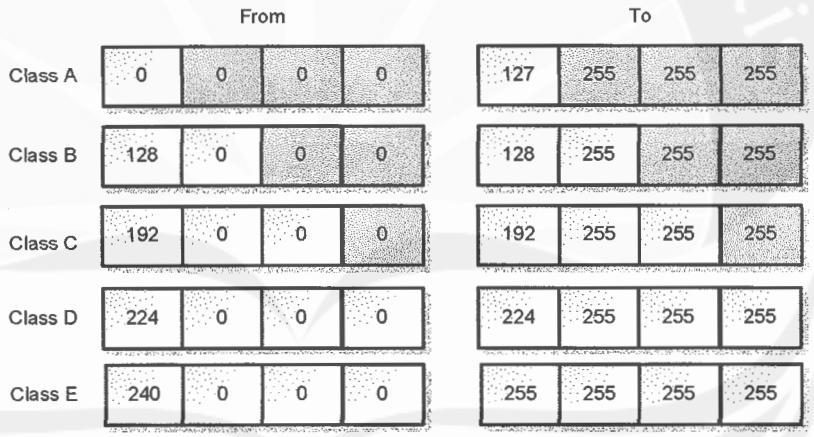
Tabel 2.5 Kelas-Kelas Alamat IP

(Sumber : id.wikipedia.org,2007)

Kelas Alamat IP	Oktet pertama (*decimal*)	Oktet pertama (*biner*)	Digunakan oleh
Kelas A	1–126	0xxx xxxx	Alamat *unicast* untuk jaringan skala besar
Kelas B	128–191	1xxx xxxx	Alamat *unicast* untuk jaringan skala menengah hingga skala besar
Kelas C	192–223	110x xxxx	Alamat *unicast* untuk jaringan skala kecil
Kelas D	224–239	1110 xxxx	Alamat *multicast* (bukan alamat *unicast*)
Kelas E	240–255	1111 xxxx	Direservasikan;umumnya digunakan sebagai alamat percobaan (eksperimen); (bukan alamat unicast)



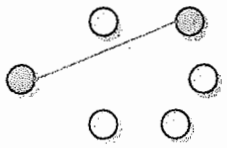
Gambar 2.8 Kelas-Kelas Alamat IP
(Sumber : Mata Kuliah Jaringan Komputer Universitas Atma Jaya Yogyakarta)



Gambar 2.9 Distribusi alamat IP
(Sumber : Mata Kuliah Jaringan Komputer Universitas Atma Jaya Yogyakarta)

Alamat IPv4 terbagi menjadi beberapa jenis, yakni sebagai berikut:

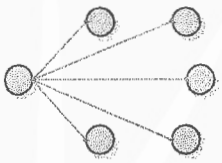
1. Alamat *Unicast*, merupakan alamat IPv4 yang didesain agar diproses oleh satu *node* IP dalam segmen jaringan. Alamat *unicast* digunakan dalam komunikasi *one-to-one* seperti terlihat pada gambar 2.10.



Gambar 2.10 *Unicast*

(Sumber : en.wikipedia.com,2007)

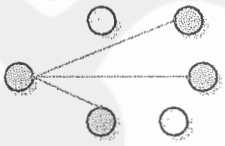
2. Alamat *Broadcast*, merupakan alamat IPv4 yang didesain agar diproses oleh setiap *node* IP dalam segmen jaringan yang sama. Alamat broadcast digunakan dalam komunikasi *one-to-everyone* seperti terlihat pada gambar 2.11.



Gambar 2.11 *Broadcast*

(Sumber : en.wikipedia.com,2007)

3. Alamat *Multicast*, merupakan alamat IPv4 yang didesain agar diproses oleh satu atau beberapa node dalam segmen jaringan yang sama atau berbeda. Alamat multicast digunakan dalam komunikasi *one-to-many* seperti terlihat pada gambar 2.12.



Gambar 2.12 *Multicast*

(Sumber : en.wikipedia.com,2007)

2.2.5 *Broadcasting*

Broadcasting adalah mengirimkan suatu paket ke seluruh host yang ada dalam jaringan lokal (LAN) khususnya *ethernet* dan *token ring*. Tidak semua jaringan

mendukung *broadcasting*, contohnya adalah jaringan X.25(en.wikipedia.org,2007). Pada jaringan *ethernet*, pengiriman paket secara *broadcast* dimungkinkan karena alamat tujuan pengiriman paket menggunakan alamat *broadcast*. Protokol lapisan *transport* yang mendukung *broadcasting* adalah UDP.

Alamat *broadcast* IPv4 adalah alamat yang dibentuk dengan cara mengeset semua bit *host* menjadi 1 dalam sebuah alamat yang menggunakan kelas (*classful*). Contohnya adalah, dalam NetID 131.107.0.0/16, alamat broadcast-nya adalah 131.107.255.255. Alamat *network broadcast* digunakan untuk mengirimkan sebuah paket untuk semua *host* yang terdapat di dalam sebuah jaringan yang berbasis kelas. *Router* tidak dapat meneruskan paket-paket yang ditujukan dengan alamat *network broadcast*(id.wikipedia.org,2007).

Network broadcasting memiliki pengalamatan khusus yang dinamakan alamat *limited broadcast*. Alamat ini adalah alamat yang dibentuk dengan mengeset semua 32 bit alamat IP versi 4 menjadi 1 (11111111111111111111111111111111 atau 255.255.255.255). Alamat ini biasa digunakan ketika sebuah *node* IP harus melakukan penyampaian data secara *one-to-everyone* di dalam sebuah jaringan lokal tetapi ia belum mengetahui *network identifier*-nya. Contoh penggunaanya adalah ketika proses konfigurasi alamat secara otomatis dengan menggunakan *Boot Protocol* (BOOTP) atau *Dynamic Host Configuration Protocol* (DHCP). Sebagai contoh, dengan DHCP, sebuah klien DHCP harus menggunakan alamat ini untuk semua lalu lintas

yang dikirimkan hingga *server* DHCP memberikan alamat IP kepadanya(id.wikipedia.org,2007).

Semua *host*, yang berbasis kelas atau tanpa kelas akan mendengarkan dan memproses paket jaringan yang dialamatkan ke alamat ini. Meskipun kelihatannya dengan menggunakan alamat ini, paket jaringan akan dikirimkan ke semua *node* di dalam semua jaringan, ternyata hal ini hanya terjadi di dalam jaringan lokal saja, dan tidak akan pernah diteruskan oleh *router* IP, mengingat paket data dibatasi saja hanya dalam segmen jaringan lokal saja. Karenanya, alamat ini disebut sebagai *limited broadcast*(id.wikipedia.org,2007).

2.3 Konsep *Remote Computer*

Administrator atau pengguna dalam kondisi tertentu harus mengoperasikan atau memperoleh informasi suatu komputer yang lain, bahkan mungkin berada di lokasi berbeda. Solusi yang dikembangkan untuk mengatasi permasalahan ini adalah dengan mekanisme *remote computer*, dimana terdapat suatu *remote network management software* (*remote computer software*) yang dapat melakukan *remote access* maupun *remote control* terhadap komputer lain bahkan seluruh komputer dalam jaringan. *Software* dalam kategori ini sudah banyak dikembangkan antara lain :*Remote Execute*, *Virtual Network Computing (VNC)*, *Windows Remote Dekstop*. Layanan dan keunggulan yang dimiliki masing-masing *software* berbeda satu sama lain.

Remote computer telah dikembangkan sedemikian rupa. Namun secara umum, *remote computer* memiliki 2 prinsip kerja utama, yaitu :

a. *Remote Access*

Mekanisme ini memungkinkan pengguna dapat bekerja (memperoleh akses) di lokasi yang berbeda dengan sumber daya komputer yang digunakan.

b. *Remote Control*

Mekanisme ini memungkinkan pengguna dapat melakukan kontrol atau memberikan perintah tertentu kepada suatu komputer meskipun berada di lokasi yang berbeda dengan sumber daya komputer yang digunakan.

Faktor krusial yang dimiliki sistem *remote computer* adalah masalah keamanan. Keamanan diperlukan untuk menghindarkan akses dan kontrol dari orang yang tidak berwewenang.

2.4 *Windows Management Instrumentation* (WMI)

Microsoft *Windows Management Instrumentation* (WMI) suatu teknologi yang disediakan *Microsoft* untuk melakukan manajemen di lingkungan *Microsoft Windows® platform*. Manajemen yang dimaksudkan menyangkut *configure, control, manage, monitor status,* dan *collect performance information* baik secara *local* maupun *remote*. WMI adalah implementasi dari *Desktop Management Task Force's* (DMTF) *Web-Based Enterprise Management* (WBEM) dan DMTF *Common Information Model* (CIM). WMI memiliki *environtment* yang sangat kompleks baik itu *software component* maupun *service* yang direpresentasikan kedalam suatu *class* manajemen objek. *Class* ini memiliki *properties* yang mendeskripsikan *data* dan *methods*.

WMI terdiri dari 3 komponen penting, komponen tersebut adalah :

a. *Management Infrastructure*

   Komponen ini menyangkut CIM *Object Manager* (CIMOM) dan CIM *Object Manager Repository*. Penggunaan utamanya adalah untuk menyimpan definisi bagan dan informasi *provider-binding*. Biasanya, data didapat kembali secara dinamis dari *provider* atas permintaan.

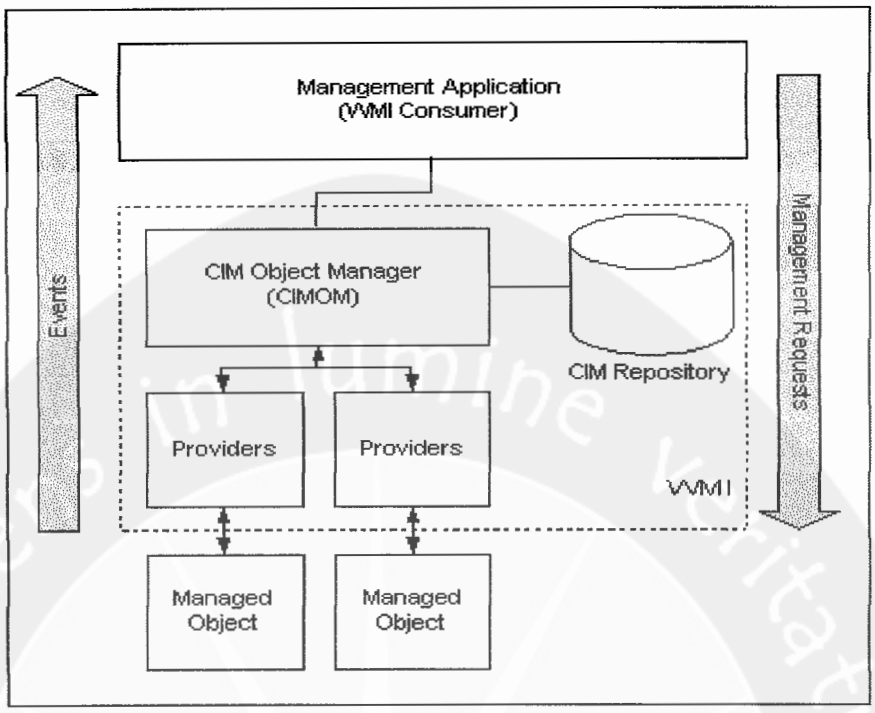
b. *WMI consumers*

   Komponen ini melakukan monitor terhadap WMI *event* yang dilakukan CIMOM dan memungkinkan adanya pelaksanaan perintah/*event* tertentu.

c. *WMI providers*

   Komponen ini adalah perantara antara CIMOM dan *object* yang akan diatur (*application* dan *component*). *Provider* melakukan interaksi dengan WMI *object model* dan memberikan data kepada CIMOM dari *managed object*, menangani permintaan WMI *Consumer* serta melakukan *generete event(Srivastava,2001)*.

Antar komponen dalam teknologi WMI memiliki keterkaitan dan merupakan satu kesatuan. Gambar 2.12 menggambarkan komponen teknologi WMI dan hubungan antar yang terjadi komponen didalammya.



Gambar 2.13 *WMI technology components*

WMI sangat membantu dalam manajemen komputer baik *desktop* maupun jaringan. Contoh manajemen data yang mungkin dilakukan dalam WMI antara lain:

a. *Component support information*

   Termasuk didalammya akses terhadap nama manufaktur dan *version number.*

b. *Component status*

   Indikator, sebagai contoh suatu komponen apakah dalam *status processing, idle* atau *disabled.*

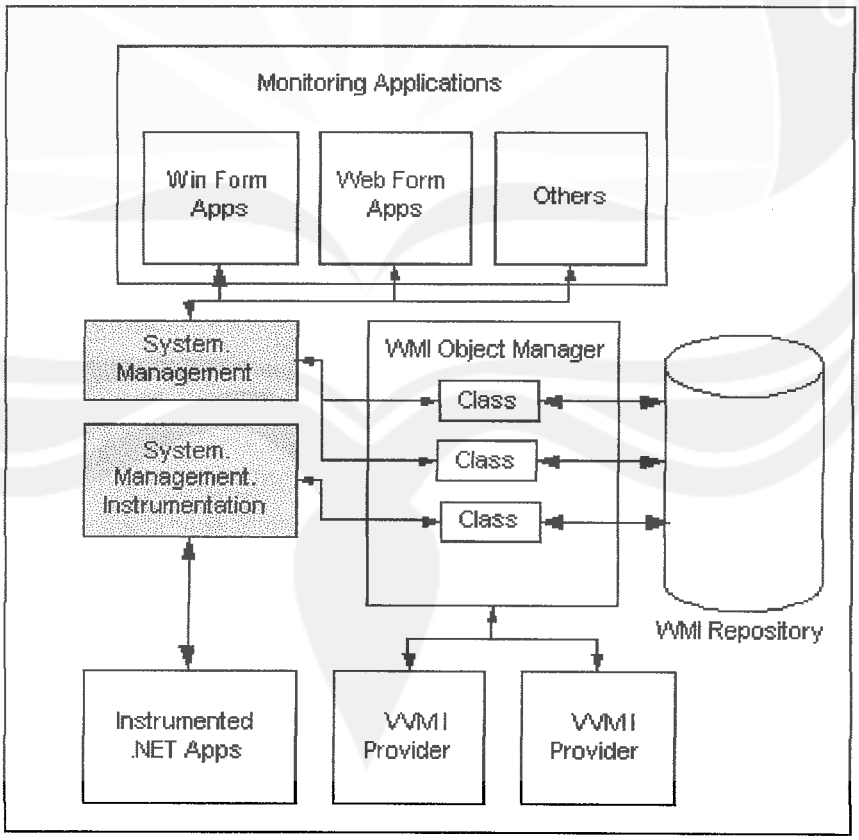
c. *Configuration information*

   Termasuk didalamnya informasi yang biasa disimpan dalam *Windows Registry* atau *runtime setting* yang dapat diubah secara dinamis.

d. *Performance information*

   Menyangkut didalamya *performance* dan data statistik lain yang dapat berubah dengan cepat.

Microsoft menyediakan *.Net Framework* yang merupakan suatu rangkaian kelas yang membantu mengimplementasikan WMI. *Framework Class Library (FCL) System.Management*, dapat digunakan untuk mengatur *event*, mengetahui informasi tentang sistem, *hardware*, dan aplikasi. Informasi diperoleh dalam suatu *query* sesuai parameter yang diberikan. FCL lainnya, *System.Diagnostics* menyediakan akses terhadap *performance* sistem. Instrumen aplikasi .NET *Framework* dapat diatur dalam *System.Management.Instrumentation*. Gambar 13 berikut menggambarkan aliran kerja suatu aplikasi WMI *consumer* yang akan memanfaatkan *.NET Framework* dalam mengakses WMI *provider*.



Gambar 2.14 Arsitektur WMI dalam .NET *Framework*.

2.5 Konsep *Wake On-LAN* (WOL)

Wake On-LAN (WOL) adalah suatu teknologi *hardware* dan *software* yang memungkinkan administrator menghidupkan (*wakeup*) komputer dalam kondisi *sleep mode* pada suatu jaringan secara *remote*. WOL akan membantu administrator dalam proses *maintenance* jaringan. Tanpa menggunakan sistem ini, administrator harus secara fisik menghidupkan komputer satu per satu, sangat tidak efisien pada jaringan yang besar.

Pengembangan teknologi WOL tidak lepas dari kerja sama Intel-IBM dalam *Wired for Management (WfM) Technology*. Teknologi ini di desain untuk membantu profesional jaringan untuk meningkatkan efektifitas dan efisiensi melalui sekumpulan perintah otomatis seperti instalasi *software* dan *upgrade*, *backup* dan penjadwalan *scaning virus*. WOL digunakan pada jaringan *Ethernet* dan *Token Ring*. Intel-IBM memperkenalkan teknologi WOL pada April 1997.

Untuk dapat menggunakan WOL maka yang diperlukan adalah suatu *Wake on LAN network adapter*, *Wake on LAN enabled motherboard* dan *remote management software*. Selain itu BIOS komputer (*Power Management*) harus diset *wake on LAN enable*. Pada saat ini telah banyak produk *network adapter* maupun *motherboard* yang mendukung WOL, *software* pendukung WOL pun sudah banyak dikembangkan baik menggunakan bahasa pemrograman Perl, PHP dsb.

WOL bekerja pada suatu komputer dalam kondisi *shutdown* tetapi NIC memiliki kemampuan untuk memicu akses *power* (tetap terkoneksi ke sumber listrik). NIC akan menangkap paket tetentu yang dikirimkan kepadanya berdasar alamat *Media Access Control* (MAC *address*) yang

dimiliki. Paket adalah suatu *Magic Packet* yang dikirimkan dalam *broadcast frame* melalui protokol *connectionless* (UDP/IPX) pada *port* 7 atau 9. *Frame* ini didefinisikan secara konstan dan direpresentasikan dalam bilangan hexadesimal: FF FF FF FF FF FF. Paket dikirimkan dengan perulangan sebanyak 16 kali tanpa henti atau interupsi. Paket inilah yang memicu NIC untuk menghidupkan komputer.

2.6 Konsep *Remote Installation Services*

Remote Installation Services (RIS) adalah suatu teknologi yang terdapat dalam *Microsoft Windows Server* untuk meng-*install* suatu sistem operasi lewat jaringan. Hal ini dimungkinkan dengan memanfaatkan *Preboot eXecution Environment* (PXE). Secara umum *workstation* dapat melakukan *boot,* dari beberapa piranti : *floppy disk, local physical disk* maupun melalui *network interface.*

PXE adalah suatu *environment* yang memungkinkan *boot* komputer dari *network interface* secara mandiri (bukan dari data *storage*). PXE diperkenalkan oleh Intel dan Systemsoft pada 20 September 1999 (en.wikipedia.org,2007). Proses kerja protokol PXE seperti kombinasi antara DHCP dan TFTP. DHCP digunakan untuk melakukan koneksi dengan *server* sedangkan TFTP digunakan untuk *download bootstrap program* dan *file* lainnya.

Pada sistem RIS, *workstation* akan menjadikan *boot* lewat PXE sebagai *boot* utama, dimana PXE akan meminta layanan BOOTP lewat jaringan. BOOTP dan DHCP sangat berhubungan erat. Workstation akan meminta *IP address*

dengan parameter MAC *address*. MAC *address* adalah kombinasi unik antara *manufacturer code* dan *unique number*. Workstation juga akan memberitahukan GUID (*Globally Unique Identifier*) atau UUID (*Universally Unique Identifier*).

Disisi BOOTP *server*, ketika menerima *request* BOOTP tidak hanya memberikan IP *address* tetapi melakukan *setting* yang memungkinkan *boot* secra *remote*. Dalam kasus ini, RIS akan menjalankan *floppy disk boot image* yang akan dipakai untuk mengambil *system image file*. Selanjutnya, sistem akan membuat konfigurasi yang memungkinkan *workstation* masuk ke dalam domain Windows dan melakukan instalasi sistem operasi lewat jaringan.

Pada windows Server 2003, dua hal penting yang digunakan dalam *remote install* yaitu DHCP dan RIS itu sendiri. *Proxy DHCP* akan menyediakan *Boot Server* dan *File Instruction* ke *client*. RIS akan menggunakan UDP *port* 4011.

2.7 Tinjauan Pustaka

Penelitian dalam bidang *remote computer* telah banyak dilakukan, antara lain yang pernah dilakukan oleh Hardi Hasan(2005) dengan judul SISTEM PENGENDALIAN DAN PEMANTAUAN KOMPUTER JARAK JAUH DENGAN MENGGUNAKAN *WINSOCK CONTROL*. Penelitian ini menghasilkan suatu sistem yang memungkinkan *admin* mengetahui informasi komputer lain, mematikan komputer lain, mengubah pengaturan komputer lain melalui *registry*, melihat aktifitas layar monitor komputer lain, memindahkan *file*, bertukar pesan (*chat*) dan fungsi lainnya. Aplikasi ini dibangun memanfaatkan komponen *ActiveX*

Control yang bernama *Winsock Control* dengan *tools* pemrograman Visual Basic 6.0 untuk mengakses Windows API (*Application Programming Interface*) yang merupakan sekumpulan fungsi-fungsi eksternal yang terdapat dalam *file* kepustakaan Windows (*library*) atau *file library* lainnya. Komponen *Winsock Control* memungkinkan pembuatan aplikasi jaringan dengan hanya mengisi nilai properti, memanfaatkan *event* serta mengeksekusi metode yang telah disediakan.

Pada prinsipnya sistem yang akan dikembangkan dalam penelitian ini akan diimplementasikan di Laboratorium Jaringan Komputer Teknik Informatika Universitas Atma Jaya Yogyakarta. Fungsionalitas sistem disesuaikan dengan kebutuhan di laboratorium tersebut.