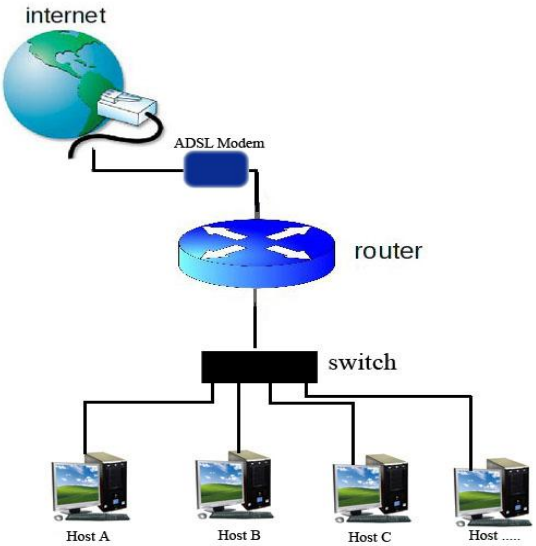
BAB II
LANDASAN TEORI

2.1 PC *(Personal Computer) Router*

PC *(Personal Computer) Router* merupakan sebuah komputer yang dijadikan sebagai *router* yang memiliki fungsi seperti *router* pada umumnya, yang berfungsi untuk mengirimkan paket data melalui sebuah jaringan atau internet menuju suatu tujuan, sebagai penghubung dua buah atau lebih pada jaringan yang berbeda.

2.2 Definisi Jaringan Komputer

Jaringan komputer adalah sekelompok komputer otonom (*stand alone*) yang saling berhubungan antara satu dengan lainnya menggunakan protokol komunikasi melalui media komunikasi sehingga dapat saling berbagi informasi, program-program. Selain itu jaringan komputer bisa diartikan sebagai kumpulan sejumlah terminal komunikasi yang berada diberbagai lokasi yang terdiri dari lebih satu komputer yang saling berhubungan [3].



Gambar 2.1 Topologi PC *Router* Sederhana

2.2.1 Klasifikasi Jaringan Komputer Berdasarkan Fungsinya

Jaringan komputer dapat dibangun dan diolah dengan mudah jika pengguna yang akan membangun jaringan tersebut memahami konsep jaringan. Terutama dalam hal tipe dan arsitektur suaru jaringan komputer. Hal ini penting karena tipe dan struktur sebuah jaringan menentukan perangkat apa yang harus disediakan untuk membangun jaringan tersebut. Berikut ini akan diuraikan mengenai tipe-tipe jaringan berdasarkan bidangnya masing-masing.

Berdasarkan fungsinya pada dasarnya setiap jaringan komputer ada yang berfungsi sebagai *client* dan juga *server*. Tetapi ada jaringan yang memiliki komputer yang khusus didedikasikan sebagai *server* sedangkan yang lain sebagai *client*. Ada juga yang tidak memiliki komputer yang khusus berfungsi sebagai *server* saja, serta berdasarkan konsep hubungan terdiri dari jaringan *client server* dan jaringan *peer to peer*.

2.2.1.1 *Client Server*

Pada jaringan *client server* terdapat sebuah komputer yang berfungsi sebagai *server* sedangkan komputer-komputer yang lain berfungsi sebagai *client*. Sesuai namanya maka komputer *server* berfungsi dan bertugas untuk melayani seluruh komputer yang terdapat dalam jaringan tersebut.

Pada sebuah komputer dimungkinkan untuk digunakannya lebih dari komputer *server*, bahkan dengan kemampuan dan fasilitas yang berbeda. Sedangkan komputer-komputer *client* sesuai dengan namanya menerima pelayanan dari komputer *server*. Komputer-komputer ini disebut juga dengan *workstation*, yaitu komputer dimana pengguna jaringan dapat mengakses dan memanfaatkan pelayanan yang diberikan oleh komputer *server*.

2.2.1.2 *Peer-to-peer*

Pada jaringan *peer to peer* setiap komputer yang terhubung dalam jaringan dapat berkomunikasi dengan komputer-komputer lain secara langsung tanpa melalui komputer perantara. Pada jaringan tipe ini sumber daya komputer terbagi

pada seluruh komputer yang terhubung dalam jaringan tersebut, baik sumber daya yang berupa perangkat keras maupun perangkat lunak dan datanya.

Sebuah komputer yang terhubung dalam sebuah jaringan *peer to peer* pada prinsipnya mampu untuk bekerja sendiri sebagai sebuah komputer *standalone* (berdiri sendiri). Membangun sebuah jaringan seperti ini pengguna bisa menggunakan komputer-komputer yang memiliki kemampuan yang setara karena keamanan dalam jaringan tersebut diatur dan di kontrol oleh masing-masing komputer dalam jaringan tersebut.

Dalam tugas akhir ini topologi jaringan komputer yang digunakan adalah topologi *client-server*.

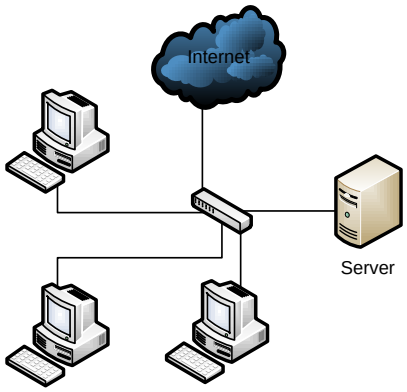
2.2.2 Klasifikasi Jaringan Komputer Berdasarkan Cakupan Geografis

Jaringan komputer dapat dibedakan berdasarkan cakupan geografisnya, ada 3 kategori utama jaringan komputer, diantaranya:

1. LAN (*Lokal Area Network*)
2. MAN (*Metropolitan Area Network*)
3. WAN (*Wide Area Network*)

2.2.2.1 LAN (*Lokal Area Network*)

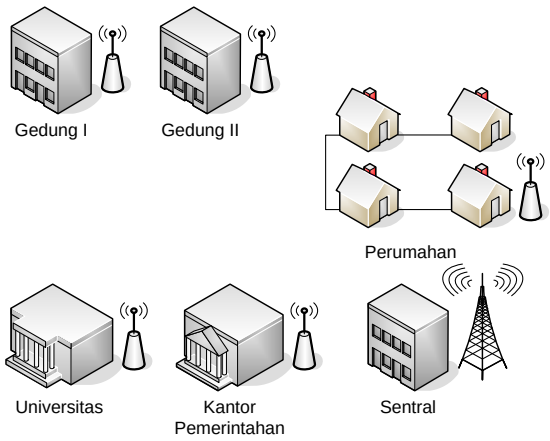
Lokal Area Network adalah jaringan yang dibatasi oleh area yang relatif kecil, umumnya dibatasi oleh area lingkungan seperti sebuah kantor pada sebuah gedung. LAN seringkali menggunakan teknologi transmisi kabel tunggal. LAN tradisional beroperasi pada kecepatan mulai 10 sampai 100 Mbps (Mega bit/detik) dengan delay rendah (puluhan mikro *second*) dan mempunyai faktor kesalahan yang kecil. LAN-LAN modern dapat beroperasi pada kecepatan yang lebih tinggi, sampai ratusan Megabit/detik.



Gambar 2.2 *Lokal Area Network*

2.2.2.2 MAN (*Metropolitan Area Network*)

MAN merupakan suatu jaringan yang cakupannya meliputi suatu kota. MAN menghubungkan LAN-LAN yang yang lokasinya berjauhan. Jangkauan MAN bisa mencapai 10 km sampai beberapa ratusan km, suatu MAN biasanya bekerja pada kecepatan 1,5 sampai 150 Mbps.

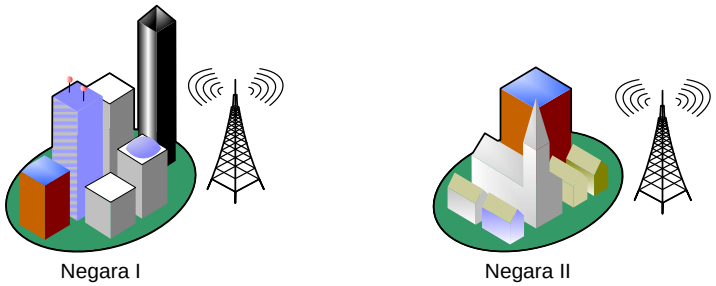


Gambar 2.3 MAN (*Metropolitan Area Network*) pada sebuah kota

2.2.2.3 WAN (*Wide Area Network*)

Wide Area Network (WAN), adalah jaringan yang ruang lingkupnya sudah menggunakan sarana satelit, wireless, ataupun kabel *fiber optic*. WAN memiliki jangkuan yang lebih luas hingga wilayah otoritas negara lain.

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



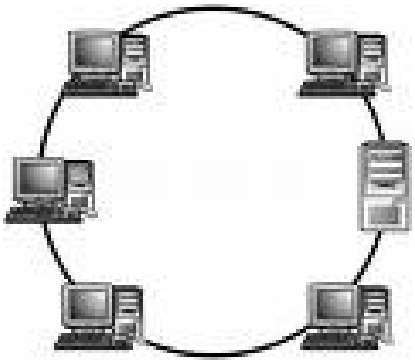
Gambar 2.4 WAN (*Wide Area Network*) antar negara

2.2.3 Klasifikasi Jaringan Komputer Berdasarkan Topologi

Topologi fisik adalah bagian yang menjelaskan hubungan antar komputer yang dibangun berdasarkan kegunaan, keterbatasan, *resource* dan keterbatasan biaya. Dengan demikian, topologi-topologi jaringan yang ada bisa disesuaikan keadaan dilapangan. Topologi ini menjelaskan hubungan perkabelan dan lokasi *node* atau *workstation*.

2.2.3.1 Topologi Cincin *(Ring Topology)*

Topologi jenis cincin ini menghubungkan satu komputer ke dalam satu *loop* tertutup. Pada topologi jenis ini, data atau *message* berjalan mengelilingi jaringan dengan satu arah pengiriman ke komputer selanjutnya terus hingga mencapai komputer yang di tuju. Waktu yang dibutuhkan untuk mencapai terminal tujuan di sebut *walk time* (waktu transmisi).



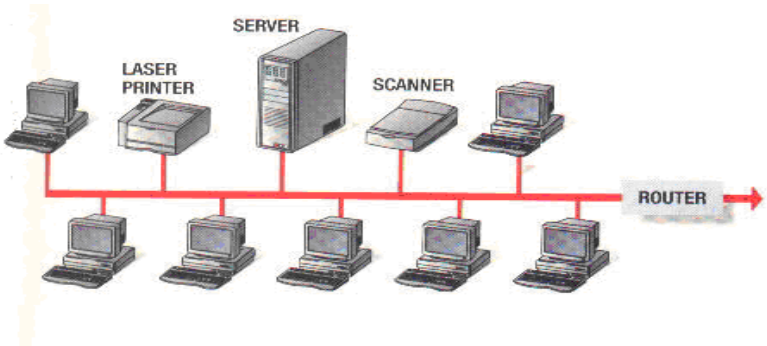
Gambar 2.5 Topologi Cincin

Ada dua hal yang dilakukan oleh suatu terminal ketika menerima data dari komputer sebelumnya, yaitu:

1. Memeriksa alamat yang di tuju dari data tersebut dan menerimanya jika terminal ini merupakan tujuan data tersebut.
2. Terminal akan meneruskan data ke komputer selanjutnya dengan memberikan tanda negatif ke komputer pengirim.

2.2.3.2 Topologi Bus *(Bus Topology)*

Topologi jaringan jenis ini menggunakan sebuah kabel pusat yang merupakan media utama dari jaringan. Topologi bus terdiri dari satu jalur kabel utama dimana pada masing-masing ujungnya diberikan sebuah *terminator*. semua *nodes* pada jaringan (*file server, workstasion,* dan perangkat lainnya) terkoneksi ke sebuah kabel utama *(backbone)*. Jaringan ini biasanya menggunkana kabel *coaxial* sebagai media transmisinya. Jaringan-jaringan *Ethernet* dan *Lokal Talk* menggunakan jaringan ini.



Gambar 2.6 Topologi Bus

Kelebihan dari topologi bus adalah, sebagai berikut:

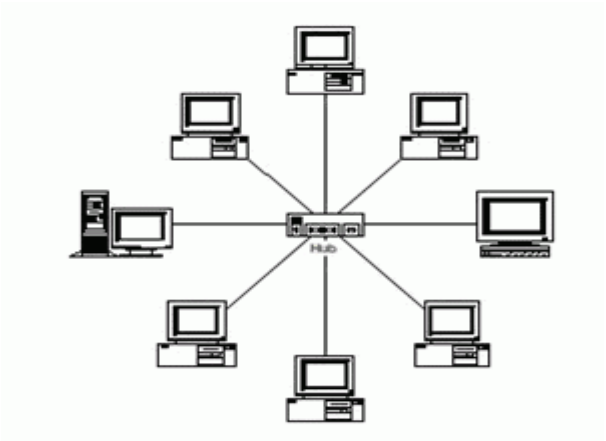
1. Mudah dalam mengkonfigurasi komputer atau perangkat lain ke sebuah kabel utama.
2. Tidak terlalu banyak menggunakan kabel dibandingkan dengan topologi bintang *(star)*.

Kekurangan topologi bus adalah, sebagai berikut:

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

2.2.3.3 Topologi Bintang *(Star Topology)*

Jenis topologi jaringan ini menggunakan satu terminal sebagai terminal sentral yang menghubungkan ke semua terminal *client*. Terminal sentral ini yang mengarahkan setiap data yang dikirimkan ke komputer yang dituju. Jenis jaringan ini apabila ada salah satu terminal client tidak berfungsi atau media transmisi terputus, maka tidak akan mempengaruhi kerja dari jaringan, karena gangguan tersebut hanya mempengaruhi terminal yang bersangkutan.



Gambar 2.7 Topologi *Star*

Kelebihan dari topologi *star* adalah sebagai berikut:

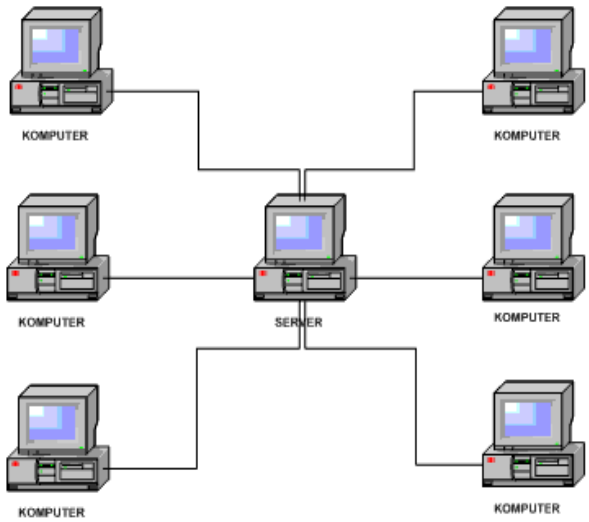
1. Mudah di dalam pemasangan dan pengkabelan
2. Tidak mengakibatkan gangguan pada jaringan ketika akan memasang atau memindahkan perangkat jaringan lainnya
3. Mudah mendeteksi kesalahan dan memindahkan perangkat-perangkat lainnya.

Kekurangan topologi *star* adalah sebagai berikut:

1. Membutuhkan lebih banyak kabel daripada topologi *bus*
2. Membutuhkan *concentrator,* dan bilamana concentrator tersebut rusak, maka semua node yang terkoneksi tidak dapat terdeteksi
3. Lebih mahal daripada topologi bus, karena biaya untuk pengadaan *concentrator*.

2.2.3.4 Topologi *Mesh*

Topologi jaringan ini menerapkan hubungan antar sentral secara penuh. Jumlah saluran harus disediakan untuk membentuk jaringan *Mesh* adalah jumlah sentral dikurangi 1 (n-1, n = jumlah sentral). Tingkat kerumitan jaringan sebanding dengan meningkatnya jumlah sentral yang terpasang. Dengan demikian disamping kurang ekonomis juga relatif mahal dalam pengoperasiannya.

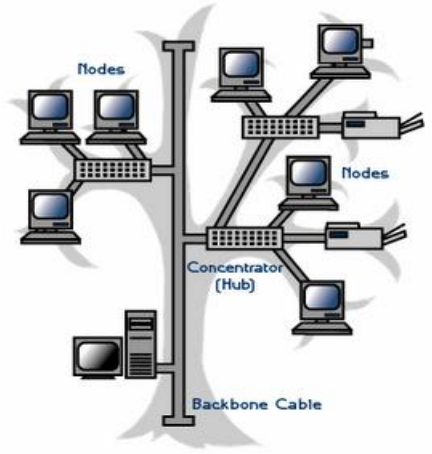


Gambar 2.8 Topologi *Mesh*

2.2.3.5 Topologi *Tree*

Topologi Jaringan Pohon *(Tree)* ini disebut juga sebagai topologi jaringan bertingkat. Topologi ini biasanya digunakan untuk interkoneksi antar sentral dengan hirarki yang berbeda. Untuk hirarki yang lebih rendah digambarkan pada lokasi yang rendah dan semakin keatas mempunyai hirarki semakin tinggi. Topologi jaringan jenis ini cocok digunakan pada sistem jaringan komputer .

Pada jaringan pohon, terdapat beberapa tingkatan simpul *(node)*. Pusat atau simpul yang lebih tinggi tingkatannya, dapat mengatur simpul lain yang lebih rendah tingkatannya. Data yang dikirim perlu melalui simpul pusat terlebih dahulu. Misalnya untuk bergerak dari komputer dengan *node-3* ke komputer *node-7* seperti halnya pada gambar 2.6 di bawah ini, data yang ada harus melewati *node-3, 5* dan *node-6* sebelum berakhir pada *node-7*. Keungguluan jaringan model pohon seperti ini adalah, dapat terbentuknya suatu kelompok yang dibutuhkan pada setiap saat. Sebagai contoh, perusahaan dapat membentuk kelompok yang terdiri atas terminal pembukuan, serta pada kelompok lain dibentuk untuk terminal penjualan. Adapun kelemahannya adalah, apabila simpul yang lebih tinggi kemudian tidak berfungsi, maka kelompok lainnya yang berada dibawahnya akhirnya juga menjadi tidak efektif. Cara kerja jaringan pohon ini relatif menjadi lambat.



Gambar 2.9 Topologi *Tree*

2.3 Definisi *Router*

Router adalah sebuah alat jaringan komputer yang mengirimkan paket data melalui sebuah jaringan atau internet menuju tujuannya, melalui sebuah proses yang dikenal sebagai *routing*. *Router* digunakan sebagai penghubung antar dua atau lebih jaringan untuk meneruskan paket data dari satu jaringan ke jaringan lainnya [10].

2.4 Protokol Jaringan

Protokol adalah sekumpulan aturan yang mendefinisikan beberapa fungsi seperti pembuatan hubungan, mengirim data, pesan, informasi atau *file,* yang harus dipenuhi oleh pengirim dan penerima agar suatu sesi komunikasi data dapat berlangsung dengan baik dan benar. Atau sekumpulan aturan untuk memecahkan masalah-masalah khusus yang terjadi antar alat-alat komunikasi agar transmisi data dapat berjalan dengan baik dan benar.

Elemen-elemen utama protokol adalah:

1. Sintaks, format data atau cara pengkodean yang digunakan untuk mengkodekan sinyal.
2. Semantik, digunakan untuk mengetahui kendali informasi untuk maksud koordinasi dan penanganan kesalahan dan informasi yang telah dikirim.
3. Timing, termasuk penyesuaian kecepatan dan penyusunan paket-paket informasi.

2.4.1 Protokol TCP/IP *(Transmission Control Protocol/Internet Protocol)*

Pada dasarnya jika 2 komputer akan melakukan pertukaran data/informasi, memerlukan sebuah protokol yang bertugas untuk mengatur bagaimana komunikasi antar komputer tersebut. Sekelompok komputer yang terhubung satu sama lain dengan *network interface* (antar muka jaringan) yang kemudian di sebut *computer network* (jaringan komputer) dapat menggunakan banyak macam protokol, agar 2 buah komputer dapat berkomunikasi maka diperlukan protokol

yang sama. Protokol berfungsi mirip bahasa manusia, dimana untuk dapat berbicara dan mengerti satu sama lain diperlukan bahasa yang sama.

TCP/IP merupakan sekelompok protokol yang mengatur komunikasi data komputer dan memungkinkan komputer berbagai jenis dan *vendor* serta berbeda sistem operasi untuk berkomunikasi bersama dengan baik. TCP/IP ini dikembangkan pertama kali oleh lembaga riset Departemen Pertahanan Amerika DARPA *(Defense Advance Research Project Agency).*

Perkembangan TCP/IP yang cepat dan diterima secara luas tidak hanya dikarenakan rekomendasi DARPA, melainkan fitur-fitur penting yang ada pada TCP/IP diantaranya:

1. TCP/IP dikembangkan menggunakan standar protokol yang terbuka. Tersedia secara bebas dan dikembangkan tanpa bergantung pada perangkat keras ataupun sistem operasi tertentu.
2. Tidak tergantung pada spesifik perangkat jaringan tertentu. Hal ini memungkinkan TCP/IP untuk mengintegrasikan berbagai macam jaringan.
3. TCP/IP menggunakan pengalamatan yang unik dalam skala global. Dengan demikian memungkinkan komputer dapat saling berhubungan walaupun jaringannya seluas internet sekarang ini.
4. Standardisasi protokol TCP/IP dilakukan secara konsisten dan tersedia secara luas untuk siapapun tanpa biaya. Hal ini diwujudkan dalam RFC *(Request For Comment).*

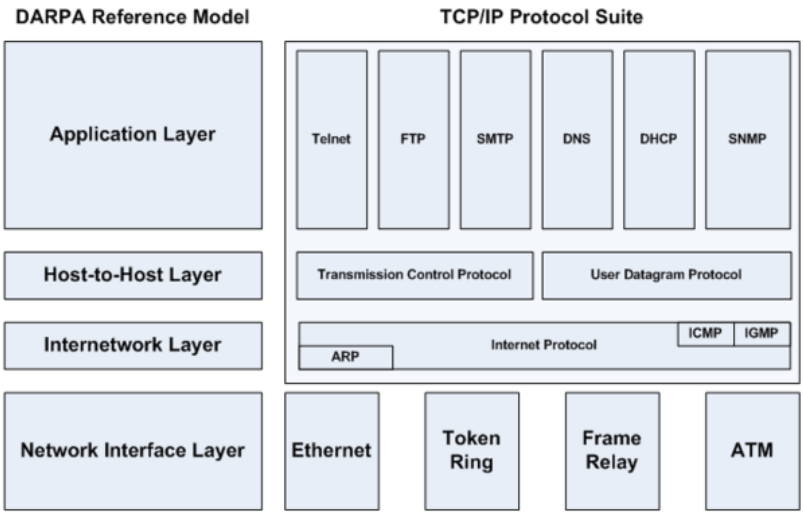
TCP/IP terdiri atas sekumpulan protokol yang masing-masing bertanggung jawab atas bagian-bagian tertentu dalam komunikasi data dan didesain untuk melakukan fungsi-fungsi komunikasi data pada LAN *(Local Area Network)* maupun WAN *(Wide Area Network).* Dengan prinsip pembagian tersebut, TCP/IP menjadi protokol komunikasi data yang fleksibel dan dapat diterapkan dengan mudah di setiap jenis komputer dan antarmuka jaringan, karena sebagian besar isi kumpulan protokol ini tidak spesifik terhadap satu komputer atau peralatan jaringan tertentu. Agar TCP/IP dapat berjalan pada antarmuka jaringan tertentu, hanya diperlukan perubahan pada bagian protokol yang

berhubungan dengan antar-muka jaringan saja. Sekumpulan protokol TCP/IP ini dimodelkan dalam empat lapisan/layer yang bertingkat.

Keempat layer tersebut ialah:

1. *Application Layer,* merupakan *layer* program aplikasi yang menggunakan protokol TCP/IP. Beberapa diantaranya adalah: Telnet, FTP *(File Transfer Protocol),* SMTP *(Simple Mail Transport Protocol),* SNMP *(Simple Network Management Protocol),* HTTP *(Hypertext Transfer Protocol),* DHCP *(Dynamic Host Configuration Protocol)* dan DNS *(Domian Name System)* .
2. *Transport Layer,* berisi protokol yang bertanggung jawab untuk mengadakan komunikasi antar dua komputer. Pada *layer* ini terdiri atas dua protokol, yaitu: TCP *(Transport Control Protocol)* dan UDP *(User Datagram Protocol).*
3. *Internet Layer,* berfungsi untuk menangani pergerakan paket data dalam jaringan dari komputer pengirim ke komputer tujuan. Protokol yang berada dalam fungsi ini antara lain: IP *(Internet Protocol),* ICMP *(Internet Control Message Protocol),* dan IGMP *(Internet Group Management Protocol).*
4. *Network Layer,* merupakan *layer* paling bawah yang bertanggung jawab mengirim dan menerima data dari dan ke media fisik.

Model sekumpulan protokol TCP/IP tersebut dapat digambarkan sebagaimana terlihat pada gambar 2.10



Gambar 2.10 Layer pada TCP/IP

2.4.2 DHCP *(Dynamic Host Configurating System)*

DHCP adalah protokol yang berbasis arsitektur *client-server* yang dipakai untuk memudahkan pengalokasian alamat IP dalam satu jaringan. Sebuah jaringan lokal yang tidak menggunakan DHCP harus memberikan alamat IP kepada semua komputer secara manual. Jika DHCP dipasang di jaringan lokal, maka semua komputer yang tersambung di jaringan akan mendapatkan alamat IP secara otomatis dari *server* DHCP. Selain alamat IP, banyak parameter jaringan yang dapat diberikan oleh DHCP, seperti *default gateway* dan *DNS server*.

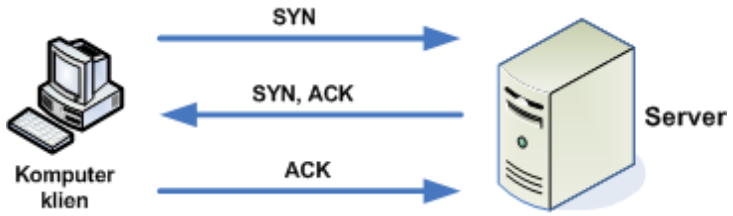
2.4.3 TCP *(Transmision Control Protocol)*

TCP merupakan protokol yang berada pada *layer transport* dari *layer* TCP/IP. TCP adalah protokol yang bersifat *byte stream*, *connection-oriented* dan *reliable* dalam pengiriman data. TCP menggunakan komunikasi *byte-stream*, yang berarti bahwa data dinyatakan sebagai suatu urutan-urutan *byte*. *Connecton-oriented* berarti sebelum terjadi proses pertukaran data antar komputer terlebih dahulu harus di bentuk suatu hubungan. Hal ini dapat doanalogikan dengan proses pendialan nomor telepon dan akhirnya terbentuk hubungan.

Kehandalan TCP dalam mengirimkan data di dukung oleh mekanisme yang disebut *Positive Acknowledgement with Re-transmission* (PAR). Data yang

di kirim dari *layer* aplikasi akan di pecah-pecah dalam bagian-bagian yang lebih kecil dan di beri nomor urut sebelum di kirim ke *layer* berikutnya. Unit data yang sudah di pecah-pecah tadi di sebut *segment*. TCP selalu meminta konfirmasi setiap kali selesai mengirimkan data, apakah data tersebut sampai pada komputer tujuan dan tidak rusak. Jika data berhasil sampai tujuan, TCP akan mengirimkan data urutan berikutnya. Jika tidak berhasil, maka TCP akan melakukan pengiriman ulang urutan data yang hilang atau rusak tersebut. Dalam kenyataannya TCP menggunakan sebuah *acknowledgement* (ACK) sebagai suatu pemberitahuan antara komputer pengirim dan penerima.

Proses pembuatan koneksi TCP disebut juga dengan *Three-way Handshake*. Tujuan metode ini adalah agar dapat melakukan sinkronisasi terhadap nomor urut dan nomor *acknowledgement* yang dikirimkan oleh kedua pihak dan saling bertukar ukuran TCP *Window*. Prosesnya dapat digambarkan sebagai berikut:



Gambar 2.11 Proses Pembuatan koneksi TCP
(Three-way Handshake)

Keterangan dari gambar 2.3 adalah sebagai berikut:

1. *Host* pertama (yang ingin membuat koneksi) akan mengirimkan sebuah segmen TCP dengan flag SYN diaktifkan kepada *host* kedua (yang hendak diajak untuk berkomunikasi).
2. *Host* kedua akan meresponsnya dengan mengirimkan segmen dengan *acknowledgment* dan juga SYN kepada *host* pertama.
3. *Host* pertama selanjutnya akan mulai saling bertukar data dengan *host* kedua.

TCP menggunakan proses jabat tangan yang sama untuk mengakhiri koneksi yang dibuat. Hal ini menjamin dua *host* yang sedang terkoneksi tersebut telah menyelesaikan proses transmisi data dan semua data yang ditransmisikan

telah diterima dengan baik. Itulah sebabnya, mengapa TCP disebut dengan koneksi yang *reliable*.

2.4.4 UDP *(User Datagram Protocol)*

UDP merupakan protokol yang juga berada pada *layer transport* selain TCP. Protokol ini bersifat *connectionless* dan *unreliable* dalam pengiriman data. *Connectionless* berarti tidak diperlukannya suatu bentuk hubungan terlebih dahulu untuk mengirimkan data. *Unreliable* berarti pada protokol ini tidak di jamin akan sampai pada tujuan yang benar dan dalam kondisi yang benar pula. Kehandalan pengiriman data pada protokol ini menjadi tanggung jawab dari program aplikasi pada *layer* atasnya. Jika dibandingkan dengan TCP, UDP adalah protokol yang lebih sederhana dikarenakan proses yang ada didalamnya lebih sedikit. Dengan demikian aplikasi yang memanfaatkan UDP sebagai protokol *transport* dapat mengirimkan data tanpa melalui proses pembentukan koneksi terlebih dahulu. Hal ini pun terjadi pada saat mengakhiri suatu koneksi, sehingga dalam banyak hal proses yang terjadi sangatlah sederhana dibanding jika mengirimkan data melalui protokol TCP.

Protokol UDP akan melakukan fungsi *multiplexing/demultiplexing* seperti yang dilakukan protokol TCP, bila suatu program aplikasi akan memanfaatkan protokol UDP untuk mengirimkan informasi dengan menentukan nomor *port* pengirim (*source port)* dan nomor *port* penerima *(destination port),* kemudian menambahkan sedikit fungsi koreksi kesalahan lalu meneruskan segmen yang terbentuk ke protokol *layer internet*. Pada *layer internet* segmen tersebut ditambahi informasi dalam bentuk datagram IP dan keudian ditentukan cara terbaik untuk mengantarkan segmen tersebut ke sisi penerima. Jika segmen tersebut tiba pada sisi penerima, protokol UDP menggunakan nomor *port* informasi IP pengirim dan penerima untuk mengantarkan data dalam segmen ke proses program aplikasi yang sesuai.

Beberapa hal yang harus diperhatikan jika suatu program aplikasi akan menggunakan protokol UDP sebagai protokol *transport:*

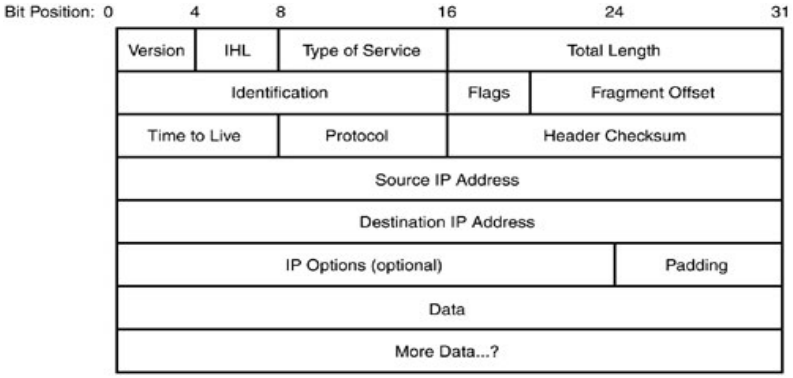
1. Tidak ada pembentukan koneksi. Protokol UDP hanya mengirim informasi begitu saja tanpa melakukan proses awal sebelumnya.
2. Tidak ada pengkondisian koneksi. Protokol UDP tidak melakukan penentuan kondisi koneksi yang berupa parameter-parameter seperti *buffer* kirim dan terima, kontrol kemacetan, nomor urutan segmen, dan *acknowledgement*.
3. Memiliki *header kecil*. Protokol UDP meiliki 8 *byte header* di banding 20 *header byte* pada TCP.
4. Tidak ada pengaturan laju pengiriman. Protokol UDP hanya menekankan kecepatan kirim pada laju program aplikasi dalam menghasilkan data, kemampuan sumber kirim (berdasarkan CPU, laju pewaktuan, dan lain-lain) dan bandwidth akses menuju internet. Jika terjadi kemacetan jaringan, sisi penerima tidak perlu menerima seluruh data yang di kirim. Dengan demikian laju penerimaan data dibatasi oleh faktor kemacetan jaringan yang terjadi, walaupun pada sisi kirim tidak memperhatikannya.

2.4.5 IP *(Internet Protocol) Address Versi 4* (IPv4)

Protokol yang paling penting yang berada pada *layer internet* TCP/IP. Semua protokol TCP/IP yang berasal dari *layer* atasnya mengirimkan data melalui protokol IP ini. Seluruh data harus dilewatkan, di olah oleh protokol IP dan dikirimkan sebagai datagram IP untuk sampai ke sisi penerima. Dalam melakukan pengiriman data, protokol IP in bersifat *unreliable, connectionless* dan datagram *delivery service*.

Unreliable berarti protokol IP tidak menjamin datagram yang dikirim pasti sampai ke tujuan. Protokol IP hanya melakukan cara terbaik untuk menyampaikan datagram yang di kirim ke tujuan. Jika pada perjalanan datagram tersebut terjadi hal-hal yang tidak diinginkan (putusnya jalur, kemacetan, atau sisi penerima yang di tuju sedang mati), protokol IP hanya memberikan pemberitahuan pada sisi kirim kalau telah terjadi permasalahan pengiriman data ke tujuan melalui protokol ICMP. *Connectionless* berarti tidak melakukan pertukaran kontrol informasi *(handshake)* untuk membentuk koneksi sebelum mengirimkan data.

Datagram *delivery service* berarti setiap datagram yang di kirim tidak tergantung pada datagram yang lainnya. Dengan demikian kedatangan datagram pun bisa jadi tidak berurutan. Metode ini di pakai untuk menjamin sampainya datagram ketujuannya, walaupun salah satu jalur menuju tujuan mengalami masalah.



Gambar 2.12 Format datagram IP

Pada bagian *header* dari protokol IP seperti yang terlihat pada gambar 2.4 terdapat bagian pengalamatan sumber kirim dan tujuan masing-masing sebesar 32-bit. Pengalamatan *(IP Addressing)* adalah bagian yang terpenting dalam jaringan TCP/IP. Alamat inilah yang sering dinamakan sebagai alamat internet yang harus dimiliki setiap node yang terhubung dalam jaringan internet. Format IP Address yang dinyatakan dalam bilangan 32-bit dimana tiap 8 bitnya dipisahkan oleh tanda titik. Untuk memudahkan distribusinya, IP Address di bagi dalam beberapa kelas. Pembagian IP Address bisa di lihat sebagai berikut ini:

1) **Kelas A**

   Jika bit pertama dari IP address adalah 0, maka IP tersebut digolongkan dalam kelas A seperti berikut :

0 – 127	0 - 255	0 – 255	0 -255
0XXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX

   IP address kelas A biasanya digunakan untuk *IP backbone*.

2) **Kelas B**

Jika 2 bit pertama dari IP address adalah 10, maka IP tersebut digolongkan dalam kelas B sebagai berikut :

128 – 191	0 - 255	0 – 255	0 -255
10XXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX

IP address kelas B biasanya digunakan untuk IP publik dan ISP.

3) **Kelas C**

Jika 3 bit pertama dari IP address adalah 110, maka IP tersebut digolongkan dalam kelas C sebagai berikut :

192– 223	0 - 255	0 – 255	0 -255
10XXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX

IP address kelas C biasanya digunakan untuk Internet, IP kelas ini yang biasanya kita pakai.

4) **Kelas D**

Jika 4 bit pertama dari IP address adalah 1110, maka IP tersebut digolongkan dalam kelas D sebagai berikut :

224– 239	0 - 255	0 - 255	0 -255
1110XXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX

IP address kelas D biasanya digunakan untuk Broadcast (Tv, Radio)

5) **Kelas E**

Jika 4 bit pertama dari IP address adalah 1111, maka IP tersebut digolongkan dalam kelas E sebagai berikut :

240– 255	0 - 255	0 - 255	0 -255
1111XXX	XXXXXXXX	XXXXXXXX	XXXXXXXX

Sedangkan IP address kelas E dicadangkan untuk kegiatan ekperimental.

2.4.6 *Ethernet*

Protokol *Ethernet* diciptakan oleh perusahaan *Xerox* sekitar tahun 1970. Pada tahun 1980, perusahaan *Xerox* bersama dengan perusahaan *Digital Equipment Corporation* (DEC) dan *Intel* menciptakan spesifikasi *Ethernet* versi-2 yang kompatibel dengan spesifikasi *IEEE 802.3*.

Protokol *Ethernet* sejauh ini adalah yang paling banyak digunakan, *Ethernet* menggunakan metode akses yang disebut CSMA/CD (*Carrier Sense Multiple Access/Collision Detection).* Sistem ini menjelaskan bahwa setiap komputer memperhatikan ke dalam kabel dari *network* sebelum mengirimkan sesuatu kedalamnya. Jika dalam jaringan tidak ada aktifitas atau bersih, komputer akan mentransmisikan data. Jika ada transmisi lain di dalam kabel, komputer akan menunggu dan akan mencoba kembali mentransmisikan jika jaringan telah bersih. Terkadang dua buah komputer melakukan transmisi pada saat yang sama, ketika hal ini terjadi masing-masing komputer akan mundur dan akan menunggu kesempatan secara acak untuk mentransmisikan data kembali. Metode ini dikenal dengan koalisi, dan tidak akan berpengaruh pada kecepatan transmisi dari *network*.

Protokol *Ethernet* dapat digunakan pada model jaringan Garis lurus , Bintang, atau Pohon . Data dapat ditransmisikan melewati kabel *twisted pair*, c*oaxial,* ataupun kabel *fiber optic* pada kecepatan 10 Mbps.

Jenis-jenis Ethernet diantaranya yaitu:

1. *Ethernet*

   Memiliki kecepatan akses data 10 Mbit/detik. Standar yang digunakan adalah: *10BaseT, 10BaseF, 10Base2 dan 10Base5*.

2. *Fast Ethernet*

   Memiliki kecepatan akses data 100 Mbit/detik. Standar yang digunakan adalah: *100BaseFX, 100BaseT, 100BaseT4* dan *100BaseTX*. Protokol ini cepat menjadi populer, karena memberikan kecepatan 10 kali lebih tinggi dibandingkan *10BaseT* dengan harga yang relatif murah.

3. *Gigabit Ethernet*

   Memiliki kecepatan akses data 1000 Mbit/detik atau 1 Gbit/detik. Standar yang digunakan adalah: *1000BaseCX, 1000BaseLX, 1000BaseSX* dan *1000BaseT*. *Gigabit Ethernet* merupakan protokol jenis Ethernet terbaru yang mendukung kecepatan 1000 Mbps.

2.4.7 *Socket*

Socket adalah piranti lunak yang digunakan untuk mengadakan hubungan secara lengkap. Socket merupakan kombinasi alamat IP dan nomor port. Sebuah socket bersifat unik pada suatu jaringan dan mewakili sebuah hubungan khusus antara dua komputer pada jaringan menggunakan sebuah layanan (*service*) khusus.

Socket terdiri dari elemen-elemen utama sebagai berikut:

1. Protokol
2. Lokal IP
3. Lokal Port
4. *Remote* IP
5. *Remote* Port

Berikut penjelasan dari elemen-elemen utama *socket*:

1. Protokol : suatu set aturan yang mengatur bagaimana dua atau lebih entitas dalam sebuah layer berinteraksi.
2. Lokal IP : Nomor lokal IP komputer
3. Lokal Port : Nomor port komputer lokal yang dibuka untuk koneksi.

4. Remote IP : Nomor IP remote komputer
5. Remote Port : Nomor port remote host yang dibuka untuk koneksi

2.4.8 Port

Salah satu elemen penting yang digunakan dalam aplikasi *socket* adalah port. Port merupakan sebuah koneksi data virtual yang digunakan aplikasi untuk bertukar data secara langsung. Terdapat banyak port didalam sebuah sistem komputer dengan fungsinya masing-masing. Sebagai contoh, dalam memonitoring jaringan digunakan service SNMP yang umumnya menggunakan port UDP 161 digunakan untuk mengirimkan dan menerima permintaan query dan port UDP 162 yang digunakan untuk menerima trap dari simpul jaringan. Port 80 digunakan untuk HTTP, port 443 digunakan untuk HTTPS, dan seterusnya.

Nomor-nomor port dikategorikan dalam tiga jenis sebagai berikut:

1. *Well-known ports*

   Merupakan port yang telah digunakan secara internal oleh *operating* sistem, misalnya port untuk koneksi internet, *service* FTP, dan seterusnya. Port yang telah digunakan ini adalah port 0 sampai 1023.
2. *Registered Port*

   Port ini digunakan dalam aplikasi anda, *range*-nya adalah port 1024 hingga port 49151, cukup banyak port yang tersedia yang bebas dipilih sehingga tidak perlu kuatir kekurangan port.
3. *Dynamic/Private Port*. Dari port 29152 sampai dengan port 6.

2.5 Pengenalan Sistem Operasi

Tugas dari sistem operasi yaitu mengurus jalannya program diatasnya, koordinasi input, output, pemrosesan, serta pemasangan dan pembuangan *software*. Bagian kode yang melakukan tugas-tugas inti dan umum dinamakan dengan kernel suatu sistem operasi. Sistem operasi dapat dikatakan sebagai penghubung antara pengguna dengan *hardware* komputer. Sistem operasi merupakan software pada lapisan pertama yang ditaruh pada memori komputer pada saat komputer dinyalakan.

2.5.1 Sistem Operasi *Linux*

Linux merupakan salah satu sistem operasi yang banyak digunakan sebagai *server* dalam jaringan komputer. *Linux* termasuk sistem operasi yang handal dan tahan terhadap virus. Disamping itu spesifikasi *hardware* yang dibutuhkan juga relatif lebih ringan bila dibandingkan dengan sistem operasi lain seperti *Windows* untuk menjalankan fungsi yang sama. Dan kelebihan yang utama adalah sistem operasi *Linux* bersifat *free licence*. Sebagai salah satu contoh sistem operasi *Linux* adalah *Linux OpenSUSE*.

OpenSUSE adalah distro *Linux* versi komunitas yang didukung dan disponsori oleh *Novell*. *OpenSUSE* merupakan distro *Linux opensource* dan gratis yang menjadi dasar pengembangan bagi distro *Linux* komersil yang disediakan oleh *Novell, SUSE Linux Enterprise Server (SLES)* dan *SUSE Linux Enterprise Desktop (SLED)*.

Salah satu keunggulan utama dari *OpenSUSE* dibandingkan distro *Linux* lainnya adalah kelengkapan pustaka dan berlimpahnya *software* yang disertakan. Bersama *Red Hat, SUSE* adalah distro *Linux* versi awal yang terus bertahan dan berkembang hingga sekarang.

Banyak orang yang takut menggunakan *OpenSUSE* karena bias pada lisensi yang digunakan. *OpenSUSE* adalah distro *Linux* yang *free* dan *open source*. *OpenSUSE* dapat digunakan secara bebas dan tanpa biaya. Jika suatu perusahaan atau lembaga menginginkan varian distro berbasis *SUSE* yang disertai dukungan support, tersedia *SLES* dan *SLED*. *Feature* yang sudah stabil dan sudah teruji pada *OpenSUSE* merupakan dasar dari *software* yang disertakan pada *SLES* dan *SLED*.

Distro *Linux OpenSUSE* ini stabil, mudah dalam melakukan deteksi perangkat keras, mudah dikelola dan didukung penuh oleh komunitas pengembang di seluruh dunia serta memiliki dukungan sponsor dari perusahaan besar [4].

2.6 Bahasa Pemograman Java

Java merupakan sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada lingkungan jaringan. *Platform* java terdiri dari kumpulan *library* kelas - kelas *loader* yang dipaket dalam sebuah lingkungan rutin java, sebuah *compiler*, *debugger* serta beberapa komponen lain yang dipaket dalam *Java Development Kit* (JDK).

MIDP (*Mobile Information Device Profile*) adalah spesifikasi untuk sebuah profil J2ME. MIDP memiliki lapisan di atas CLDC API, tambahan untuk mendaur ulang aplikasi, *interface*, jaringan, dan penyimpanan persisten. Pada saat ini terdapat MIDP 1.0 dan MIDP 2.0. Fitur tambahan yang terdapat pada MIDP 2.0 adalah API untuk multimedia. MIDP *User Interface* API yang memiliki API level tinggi dan level rendah. API level rendah berbasiskan penggunaan dari kelas abstrak kanvas, sedangkan pada API level tinggi antara lain *alert, form, list dan textbox* yang merupakan ekstensi dari kelas *abstrak screen*. API level rendah memberikan kemudahan kepada pengembang untuk memodifikasi sesuai dengan kehendaknya, sedangkan API level tinggi biasanya hanya memberikan pengaksesan yang terbatas [1].

2.7 PHP *(Personal Home Page/PHP Hypertext Processor)*

PHP adalah bahasa *scripting* yang menyatu dengan HTML dan dijalankan pada *server side*. Artinya semua sintaks yang diberikan akan sepenuhnya dijalankan pada *server* sedangkan yang dikirimkan ke *browser* hanya hasilnya saja. PHP bersifat *open source* dan telah digunakan oleh hampir seluruh *web developer* di seluruh dunia. Karena sifatnya yang *open source* dan semakin banyaknya *user* yang menggunakannya, membuat bahasa pemrograman ini mengalami perkembangan yang sangat cepat. PHP hampir dapat berjalan di semua sistem operasi seperti *Windows, Unix, Linux* dan varinnya, *Mac OS X, RIC OS* dan lain-lain. PHP juga bisa berjalan hampir di semua *web server* yang ada sekarang ini, seperti *Apache Web Server, IIS, Personal Web Server, Caudium, Xitami, Omni* dan masih banyak lagi. Dengan begitu sistem operasi dan *web server* apapun yang digunakan, PHP dapat berjalan dengan baik [2].

2.8 *Apache Web Server*

Apache web server adalah *web server* yang dapat dijalankan di banyak sistem operasi *(Unix, BSD, Linux, Microsoft Windows* dan *Novell Netware* serta *platform* lainnya) yang berguna untuk melayani dan memfungsikan situs *web*. Protokol yang digunakan untuk melayani fasilitas web ini menggunakan *Hypertext Transfer Protokol* (HTTP). *Apache* memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigur, autentikasi berbasis basis data dan lain-lain. *Apache* juga didukung oleh sejumlah antarmuka pengguna berbasis grafik *(GUI)* yang memungkinkan penanganan *server* menjadi mudah. *Apache* merupakan perangkat lunak sumber terbuka dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan *Apache Software Foundation.(ASF)*

Apache Web Server mempunyai kelebihan sebagai berikut:

1. *Apache* termasuk dalam kategori *freeware*.
2. *Apache* mudah sekali proses instalasinya jika dibanding *web server* lainnya seperti NCSA *(National Center for Supercomputing Applications)*, IIS *(Internet Information Server)*, dan lain-lain.
3. Mampu beroperasi pada berbagai *platform* sistem operasi.
4. Mudah mengatur konfigurasinya. *Apache* mempunyai hanya empat file konfigurasi.
5. Mudah dalam menambahkan *peripheral* lainnya ke dalam *platform web* servernya [8].

2.9 XML (*Extensible Markup Language*)

XML adalah suatu metode dalam membuat penanda/markup pada sebuah dokumen seperti pendahulunya SGML (*Standard Generalized Markup Language*) maupun HTML (*HyperText Markup Language*). *Markup Language* adalah suatu metode untuk mendeskripsikan tentang kumpulan data. Jadi suatu *Markup Language* lebih merupakan suatu informasi tentang suatu data dan bukan data itu sendiri. Pemakaian XML biasanya dijadikan sebagai data yang dapat diterima pada sistem operasi apapun (bebas platform) dan aplikasi manapun [7].

2.10 SNMP (*Simple Network Management Protokol*)

Protokol yang dapat digunakan untuk melakukan menagemen jaringan. Dengan menggunakan protokol ini kita bisa mendapatkan informasi tentang status dan keadaan dari suatu jaringan.

SNMP menggunakan UDP (*User Datagram Protokol*) sebagai protokol *transport* untuk mengirimkan pertanyaan dan menerima jawaban dari agen SNMP.

SNMP terdiri dari dua jenis yakni:

1. *Network Management Station* yang berfungsi sebagai pusat penyimpanan untuk pengumpulan dan analisa dari data manajemen jaringan.
2. Peralatan yang dimanage menjalakan SNMP agent yaitu proses *background* yang memonitor peralatan tersebut dan mengkomunikasikannya *ke network management station* [9].

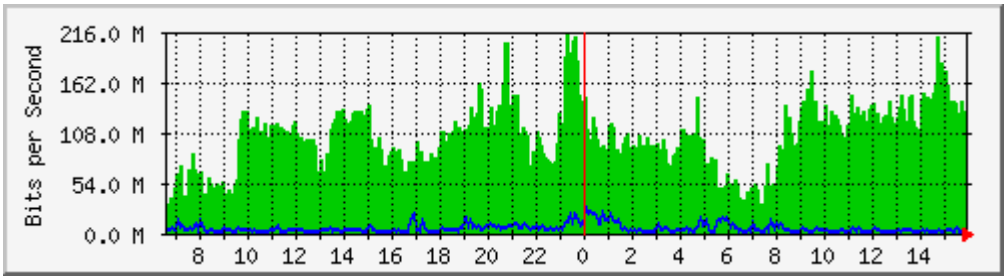
2.11 MRTG (*Multi Router Traffic Grapher*)

Multi Router Traffic Grapher atau yang biasa disingkat dengan MRTG, merupakan aplikasi yang dapat digunakan untuk memonitoing *traffic load* dalam suatu jaringan. MRTG dapat digunakan oleh seorang user untuk melihat *traffic load* yang terdapat pada jaringan pada kurun waktu tertentu dalam bentuk tampilan graphic. MRTG dibuat dengan menggunakan bahasa *Perl* dan dapat berjalan pada beberapa sistem operasi seperti *Unix/Linux* dan *Windows*. Pada awalnya, MRTG dibuat oleh Tobias Oetiker dan Dave Rand untuk memonitor *traffic router*. Selanjutnya, aplikasi ini dikembangkan sebagian yang dapat menghasilkan *graph* dan statistik dari *traffic* jaringan.

MRTG menggunakan *Simple Network Managemen Protokol*(SNMP) untuk mengirimkan dua buah *object identifers* (OIDs) ke sebuah perangkat. Perangkat ini harus dapat mendukung SNMP dan kemudian *Management Information Base* (MIB) akan mencari OID's yang telah dispesifikasikan. Setelah mengoleksi informasi, MIB akan mengirim baik raw data yang telah di enkapsulasi dalam sebuah protokol SNMP. MRTG menyimpan data ini dalam sebuah *log client* dari *record* data yang telah dihasilkan oleh suatu perangkat.

Selanjutnya MRTG akan menghasilkan dokumen HTML dari *log* yang dihasilkan yang didalamnya berisi daftar *traffic* dalam bentuk detail *graph* untuk perangkat yang dimaksud.

MRTG digunakan untuk memonitoring trafik *router* jaringan. Hasil *monitoring* dapat berupa grafik per jam, per hari, per bulan maupun per tahun. Ada pun contoh grafik hasil monitoring [6]:



Gambar 2.13 Grafik MRTG