BAB II
DASAR TEORI

Pada bab ini akan dibahas mengenai teori dan komponen penunjang yang akan digunakan dalam implementasi pengaturan dan prioritas *bandwidth* menggunakan metode HTB berbasiskan GUI. Pembahasan berisi tentang dasar-dasar jaringan, komponen fisik dalam jaringan, Aplikasi, teknik antrian, dan sistem operasi.

2.1 Jaringan Komputer

Jaringan komputer merupakan gabungan antara teknologi komputer dan teknologi komunikasi. Gabungan teknologi ini menghasilkan pengolahan data yang dapat didistribusikan, mencakup pemakaian *database*, aplikasi *software* dan peralatan *hardware* secara bersamaan, sehingga penggunaan komputer yang sebelumnya berdiri sendiri, kini telah diganti dengan sekumpulan komputer yang terpisah akan tetapi saling berhubungan dalam melaksanakan tugasnya, sistem seperti inilah yang disebut jaringan komputer (*computer network*)[1].

Bentuk koneksinya tidak harus melalui kabel melainkan dapat menggunakan, serat optik, gelombang mikro, *wireless* atau satelit komunikasi.Supaya dapat berfungsi, dibutuhkan layanan-layanan yang dapat melakukan pengaturan sumberdaya, dan juga dibutuhkan aturan-aturan (*protocol*) yang mengatur komunikasi dan layanan secara umum untuk seluruh sistem jaringan.

Jaringan komputer dapat dibagi berdasarkan beberapa klasifikasi, di antaranya:

1. Berdasarkan area.
2. Berdasarkan media penghantar.
3. Berdasarkan fungsi.

2.1.1 Berdasarkan Area

Jaringan komputer dapat dibedakan berdasarkan cakupan geografisnya. Ada tiga kategori utama jaringan komputer yaitu[2]:

1. LAN *(Local Area Network)*

   LAN merupakan jaringan kecil yang terdiri dari dua atau lebih komputer dan satu atau lebih server yang terhubung dalam suatu area.Area yang dimaksud biasanya dalam satu gedung atau departemen. LAN dapat dibagi lagi berdasar struktur, perangkat, dan cara kerjanya.
2. MAN *(Metropolitan Area network*)

   pada dasarnya versi lain LAN yang berukuran lebih besar dan biasanya menggunakan teknologi yang sama dengan LAN. MAN dapat mencakup kantor-kantor perusahaan yang letaknya berdekatan atau juga sebuah kota dan dapat dimanfaatkan untuk keperluan pribadi (swasta) atau umum. MAN mampu menunjang data dan suara, bahkan dapat berhubungan dengan jaringan televisi kabel.
3. WAN *(Wide Area Network*)

   WAN merupakan jaringan yang jangkauannya mencakup daerah geografis yang luas, seringkali mencakup sebuah negara bahkan benua. WAN terdiri dari kumpulan mesin-mesin yang bertujuan untuk menjalankan program-program (aplikasi) pemakai.

2.1.2 Berdasarkan Media Penghantar

Berdasarkan media penghantar, jaringan komputer dapat dibagi menjadi dua jenis, yaitu[2] :

1. *Wire Network*

   Wire Network adalah jaringan komputer yang menggunakan kabel sebagai media penghantar,jadi data ditransmisikan pada kabel. Kabel yang umum digunakan pada jaringan komputer biasanya menggunakan bahan dasar tembaga. Ada juga jenis kabel lain yang menggunakan bahan *fiber optics* atau

serat optik. Bahan tembaga banyak digunakan pada LAN. Sedangkan untuk MAN dan WAN menggunakan gabungan kabel tembaga dan serat optik.

2. *Wireless Network*

   Wireless Network adalah jaringan tanpa kabel yang menggunakan media penghantar gelombang radio. Frekuensi yang digunakan pada radio untuk jaringan komputer biasanya menggunakan frekuensi tinggi, yaitu 2.4 GHz dan 5.8 GHz.

2.1.3 Berdasarkan Fungsi

Berdasarkan fungsinya, jaringan komputer dapat dibagi menjadi dua jenis, yaitu [2]:

1. *Client Server*

   Cient server adalah jaringan komputer yang salah satu komputer difungsikan sebagai *server* atau komputer yang memberikan pelayanan kepada komputer lain. *Server* melayani komputer lain yang disebut *client*. Layanan yang diberikan bisa berupa akses *web*, *e-mail*, *file* atau yang lain. *Client server* banyak dipakai pada *internet*. Namun LAN atau jaringan lain pun bisa mengimplementasikan *client server*.

2. *Peer to Peer*

   Peer to peer adalah jaringan komputer dimana setiap komputer bisa menjadi *server* sekaligus *client*. Setiap komputer dapat menerima dan memberikan akses dari atau ke komputer lain.

2.2 Topologi Jaringan Komputer

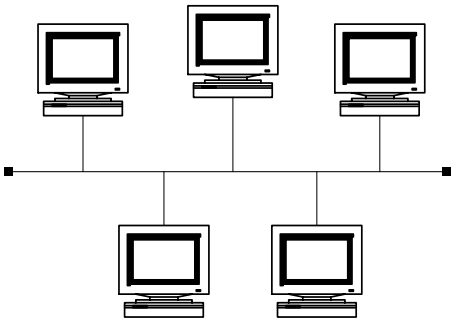
Topologi adalah suatu aturan bagaimana menghubungkan komputer satu sama lain secara fisik dan pola hubungannya antara komponen-komponen yang berkomunikasi melalui media atau peralatan jaringan. Ada dua jenis topologi yaitu topologi fisik (*physical topology*) dan topologi logika (*logical topology*)[2].

2.2.1 Topologi Fisik (*Physical Topology*)

Berikut ini akan dijelaskan mengenai topologi utama yang mejadi dasar bagi topologi fisik lain yang digunakan di dalam jaringan diantaranya[5]:

1. *Bus*

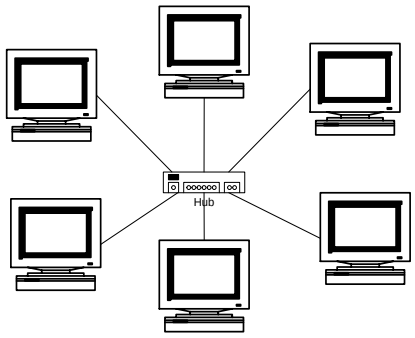
   Topologi *Bus* terdiri dari satu jalur kabel utama dimana pada masing-masing ujungnya diberikan sebuah terminator. Semua nodes (*file server, workstation*, dan perangkat lainya) pada jaringan terkoneksi sebuah kabel utama (*backbone*). Jaringan-jaringan *Ethernet* dan *Local Talk* menggunakan topologi ini.



Gambar 2.1 Topologi Bus

2. *Star* (Bintang)

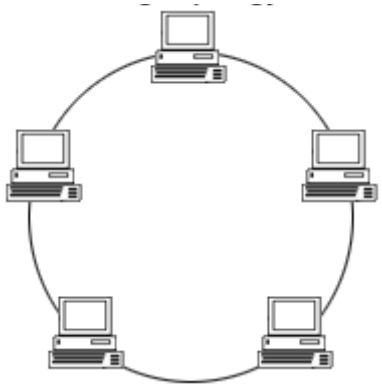
   Topologi model ini dirancang, yang mana setiap *nodes* (*file server, workstation*, dan perangkat lainya) terkoneksi ke jaringan melewati sebuah *concentrator*. Data yang dikirim ke jaringan *local* akan melewati *concentrator* sebelum melanjutkan ke tempat tujuannya. *Concentrator* akan mengatur dan mengendalikan keseluruhan fungsi jaringan, dan juga bertindak sebagai *repeater*.



Gambar 2.2 Topologi Star

3. *Ring*

   Topologi *Ring* menggunakan teknik konfigurasi yang sama dengan topologi *star* tetapi pada topologi ini terlihat bahwa jalur media transmisi menyerupai suatu linkaran tertutup menyerupai cincin.



Gambar 2.3 Topologi Ring

Ketiga topologi fisik tersebut dapat dibuat berdiri sendiri atau dalam jaringan yang besar dapat juga digabungkan sesuai dengan kondisi dan rencana penggunaannya.

2.2.2 Topologi Logika (*Logical Topology*)

Topologi logika berkaitan dengan bagaimana data ditransmisikan pada jaringan, berikut topologi logika dikelompokan berdasarkan metode aksesnya adalah[2]:

1. *Shared Media Topology*

   Pada topologi *shared media*, semua *node* atau *network device* yang terhubung ke jaringan dapat mengakses media jaringan kapan saja ketika diperlukan. Akses ke media dilakukan setiap saat dan tidak dibatasi. Ini merupakan salah satu keuntungan dari topologi ini namun sekaligus juga merupakan kelemahannya. Karena setiap peralatan dapat mengakses media jaringan kapan pun maka kemungkinan terjadi tabrakan data (*collision*) akan cukup besar contoh jaringan yang menggunakan *shared media* adalah semua varian yang menggunakan *Ethernet card* (topologi *bus, star, tree*)[1].

2. *Token Based Topology*

   Topologi *token based* menggunakan sebuah *frame* data bernama *token* yang ditransmisikan mengelilingi jaringan. Data transmisikan pada media jaringan, melewati setiap komputer satu-persatu hanya satu arah saja misalakan searah jarum jam atau sebaliknya. Akses setiap *node* ke media fisik jaringan diatur oleh *token*. Karena pengiriman data dilakukan secara bergantian dan setiap *nod*e harus menunggu giliran, maka tidak akan pernah terjadi *collision*. Namun, waktu tunggu atau *delay* dapat terjadi apabila banyak *node* yang ingin mengirim data. Jaringan yang menggunakan token based di antaranya IBM Token Ring, ATR, FDDI, CDDI, ARCNET[1].

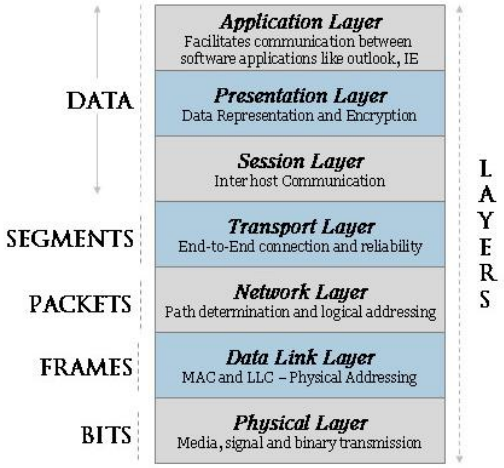
2.3 Protokol Jaringan Komputer

Di dalam jaringan komputer, komunikasi terjadi antara sekumpulan entitas yang berbeda. Entitas adalah sesuatu yang mampu mengirim atau menerima informasi. Namun, dua entitas tidak begitu saja mengirim aliran bit data satu sama lain dan dapat dipahami. Untuk terjadinya komunikasi, entitas harus menyetujui suatu protokol. Protokol adalah seperangkat aturan yang mengatur komunikasi data. Protokol mendefinisikan apa itu komunikasi, bagaimana berkomunikasi dan apa yang terjadi ketika berkomunikasi[4].

2.3.1 OSI (Open System Interconnection) Model

Suatu jaringan komputer dibangun dengan memperhatikan arsitertur standar yang dibuat lembaga standar industri dunia. Standar jaringan yang saat ini diakui dunia adalah *The Open System Connection*atau OSI yang dibuat oleh lembaga ISO (*International Standar Organization*), Amerika Serikat. Seluruh fungsi kerja jaringan komputer dan komunikasi antar entitas diatur dalam standar ini[4].

OSI adalah suatu standar komunikasi antar entitas yang terdiriatas 7 lapisan. Ketujuh lapisan tersebut mempunyai peran dan fungsi yang berbeda satu terhadap yang lain. Setiap *layer* bertanggung jawab secara khusus pada proses komunikasi data. Tujuan utama penggunaan model OSI adalah untuk membantu desainer jaringan memahami fungsi dari tiap-tiap *layer* yang berhubungan dengan aliran komunikasi data, termasuk jenis-jenis protokol jaringan dan metode transmisi. Model OSI dibagi menjadi 7 *layer*, dengan karakteristik dan fungsinya masing-masing. Tiap *layer* harus dapat berkomunikasi dengan layer di atasnya maupun di bawahnya secara langsung melalai beberapa protokol dan *standard*.



Gambar 2.4 OSI Layers Model

1. *Physical layer*

   Lapisan terendah ini mengatur singkronisasi pengiriman dan penerimaan data, spesifikasi mekanik, elektrik dan *interface* antar terminal, seperti besar tegangan, frekuensi, inpedansi koneksi *pin* dan jenis kabel.

2. *Data Link layer*

   Pada lapisan ini data diubah dalam bentuk paket, singkronisasi paket yang dikirim maupun yang diterima menjadi format yang disebut *frame.* Selain itu, pada level ini terjadi koreksi kesalahan, flow control, pengalamatan perangkat keras seperti halnya *Media Acces Control* (MAC *address*) dan menetukan bagaimana perangkat-perangkat jaringan seperti *hub, bridge, repeater* dan *switch layer* 2 beroperasi.

3. *Network layer*

   Lapisan ini menentukan rute pengiriman dan mendefinisikan alamat-alamat IP, membuat *header* untuk paket-paket, dan kemudian melakukan *routing* dan *switch layer* 3 agar data sampai ditempat tujuan dengan benar.

4. *Transport layer*

   Lapisan ini mengatur keutuhan data, menerima data dari lapisan *session* dan meneruskannya ke lapisan *network.* Lapisan ini berfungsi untuk memecah data ke dalam paket-paket data serta memberikan nomor urut ke paket-paket tersebut sehingga dapat disusun kembali pada sisi tujuan setelah diterima.

5. *Sesion layer*

   Lapisan ini menyiapkan saluran komunikasi dan terminal dalam hubungan antar terminal, mengoordinasikan proses pengiriman dan penerimaan serta mengatur pertukaran data.

6. *Presentation layer*

   Pada lapisan ini dilakukan konversi data agar data yang dikirim dapat dimengerti oleh penerima, kompresi teks dan penyendian data.

7. *Application layer*

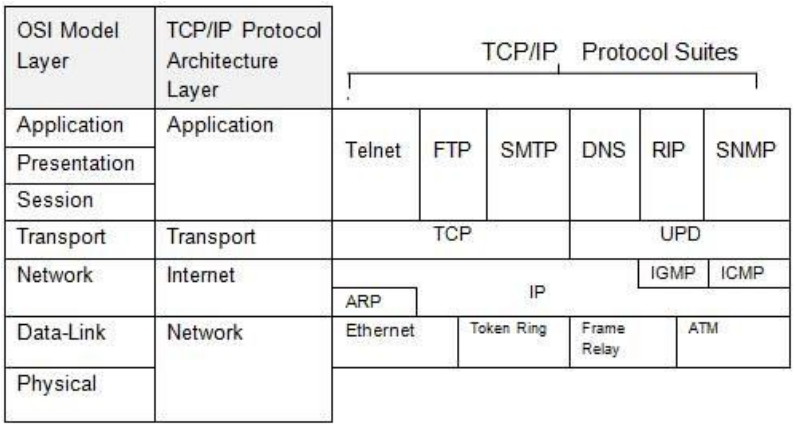
   Lapisan paling atas ini mengatur interaksi pengguna komputer dengan program aplikasi yang dipakai. Lapisan ini juga mengatur bagaimana aplikasi dapat mengakses jaringan, dan kemudian membuat pesan pesan

kesalahan.Protokol yang berada pada lapisan ini adalah HTTP, FTP, SMTP, dan NFS.

Ketika data ditranfer melalui jaringan, maka data sebelumnya harus melewati ketujuh dari satu terminal, mulai dari *layer* aplikasi sampai *physical layer*, kemudian di sisi penerima data tersebut juga melewati *layer physical* sampai aplikasi. Pada saat data melewati satu persatu layer dari sisi pengirim, maka akan ditambahkan satu *header* sedangkan pada sisi penerima *header* dibuka sesuai dengan *layer*nya.

2.3.2 TCP/IP Protocol suite

TCP/IP *protocol suite* dikembangkan sebelum model OSI. Oleh karena itu, lapisan di TCP / IP *protocol suite* tidak sama persis dalam model OSI. TCP/IP *protocol suite* memiliki empat lapisan yaitu *network*, *internet*, *transport*, dan *application*. Namun, ketika TCP / IP dibandingkan dengan OSI dapat dikatakan bahwa lapisan *network* setara dengan gabungan dari layer *data link* dan *physical. Internet layer* setara dengan layer *network,* dan *applicaton layer*mengerjakan pekerjaan yang sama seperti *session*, *presentation*, dan *application layer*[4] .



Gambar 2.5 TCP/IP dan OSI model

1. *Application layer*

   Berfungsi menyediakan akses aplikasi terhadap jaringan TCP/IP. Layer ini menangani *high-level protocol*, masalah representasi data, proses *encoding* dan dialog *control* yang memungkinkan terjadinya komunikasi antar aplikasi jaringan.

2. *Transport layer*

   Berfungsi membuat komunikasi antar dua host. *Layer Transport* menyediakan layanan pengiriman dari sumber data menuju ke tujuan data dengan cara membuat *logical connection* di antara keduanya.

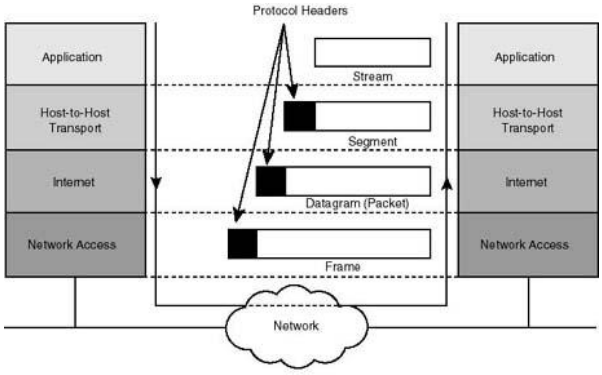
3. *Internet layer*

   Berfungsi untuk melakukan routing dan pembuatan paket IP menggunakan teknik *encapsulation* (*header* dan *data*). *Internet layer* memiliki tugas utama untuk memilih rute dalam sebuah jaringan. Selain itu, layer ini juga bertugas untuk melakukan *packet swictching* untuk mendukung tugas utama tersebut.

4. *Network layer*

   Berfungsi untuk menyimpan frame-frame data yang akan dikirim ke media jaringan. Layer ini bertugas bertugas mengatur semua hal yang diperlukan sebuah paket IP.

Seperti halnya model OSI, pengiriman data yang dilakukan pada model TCP/IP pun menggunakan prinsip *encapsulation*. Ilustrasi pada gambar 2.6 dapat digunakan untuk memahami proses pengiriman dan penerimaan data pada model TCP/IP[2].



Gambar 2.6 Proses *encapsulation* pada model TCP/IP

Pertama datangnya data ditangani oleh *layer Application* menggunakan aplikasi pada TCP/IP seperti *e-mail*, FTP, telnet, dan yang lainya. Ketika data dikirim ke layer ini, maka *header* ditambahkan ke data dan dibentuk paket-paket dengan panjang tertentu yang siap untuk dikirim ke *layer host to host transport*. Setelah data sampai di layer *host to host*, protokol menambahkan *header* pada setiap paket. Contoh *header* adalah informasi mengenai urutan segmentasi untuk menjaga integritas data dan *bit-bit* parity untuk keperluan deteksi dan koreksi kesalahan (*error detection* dan *error correction*).

Selanjutnya data dikirim ke *layer Network* dan diberi *header* yang berisi informasi alamat tujuan, alamat pengirim, dan informasi lain yang dibutuhkan untuk melakukan *routing*. Sehingga dapat ditentukan *network* mana (melalui *interface* tertentu) yang akan dikirim. Pada layer ini juga terjadi segmentasi data, panjang paket yang akan dikirimkan harus sesuai dengan kondisi media komunikasi dari jaringan yang akan dilalui. Protokol dalam lapisan ini menyiapkan data dalam bentuk yang paling sesuai untuk dikirimkan melalui media komunikasi tertentu.Selanjutnya data siap dikirimkan ke *layer network interface*.

Pada *layer Network Interface* ini data diubah menjadi besaran-besaran listrik atau fisik seperti tegangan, arus, gelombang radio maupaun cahaya, sesuai dengan media yang digunakan yang menghasilkan *bit-bit* data yang siap untuk dikirim. Setelah *bit-bit* sampai ke komputer penerima, proses *decapsulatin* atau kebalikan dari *encapsulation* dilakukan.Data dibuka *header*nya satu persatu, mulai dari *layer* terbawah hingga *layer* paling atas. Setelah sampai pada *layer Application* barulah data yang sebenarnya bisa terbaca oleh aplikasi TCP/IP.

2.4 IP Address

IP Address adalah sebuah alamat yang diberikan ke peralatan jaringan untuk mengakses internet atau ke suatu jaringan komputer dengan menggunakan protocol TCP/IP. Setiap komputer yang terhubung ke jaringan harus mempunyai alamat yang unik. IP Address itu sendiri ditentukan oleh *subnetmask. Subnetmask* ini berfungsi untuk membedakan bagian mana yang disebut sebagai *host*[3].

IP Address terdiri dari bilangan biner sepanjang 32 bit yang dibagi atas 4 segmen. Tiap segmen terdiri atas 8 bit yang berarti memiliki nilai decimal dari 0 sampai 255. Jadi jaringan TCP/IP dengan 32 bit address ini mampu membuat sebanyak 2^{32}atau lebih dari 4 milyar *host*. Ada 5 kelas IP Address namun ada 3 kelas yang utama dalam TCP /IP, yakni Kelas A, Kelas B dan Kelas C sedangkan kelas D digunakan sebagai *multicast* dan kelas D sebagai *research* serta tidak digunakan untuk umum. perangkat lunak *internet protocol* menentukan pembagian jenis kelas ini dengan menguji bit pertama dari IP Addres. Penentuan kelas ini dilakukan dengan cara berikut :

1. IP Address kelas A

   Jika bit pertama dari IP Address adalah 0, maka network merupakan kelas A. bit awal dan 7 bit berikutnya (8 bit pertama) merupakan bit network sedangkan 24 bit terakhir merupakan bit host. Dengan demikian hanya ada 128 *network* di kelas A, yakni nomor 0.xxx.xxx.xxx sampai 127.xxx.xxx.xxx, tetapi network dapat mendukung lebih dari 16 juta host (xxx adalah variabel, nilainya dari 0 sampai dengan 255). Ilustrasinya dapat dilihat pada Tabel 2.1 berikut[3]:

Tabel 2.1 Struktur IP Address Kelas A

0-127	0-255	0-255	0-255
0nnnnnnn	hhhhhhhh	hhhhhhhh	hhhhhhhh
Bit-bit network ⟷	Bit-bit host ⟷		

2. IP Address kelas B

   Jika 2 bit pertama dari IP Address adalah 10, maka network merupakan kelas B. dua bitawal dan 14 bit berikutnya (16 bit pertama) merupakan bit network sedangkan 16 bit terakhir merupakan bit host. Dengan demikian terdapat lebih dari 16 ribu network kelas B (64 x 256), yakni dari network 128.0.xxx.xxx – 192.255.xxx.xxx. setiap network kelas B mampu

mendukung lebih dari 65 ribu host. Iliustrasinya dapat dilihat pada tabel 2.2 berikut:

Tabel 2.2 Struktur IP Address Kelas B

128-191	**0-255**	0-255	0-255
10nnnnnn	**nnnnnnnn**	hhhhhhhh	hhhhhhhh
Bit-bit network ⟷		Bit-bit host ⟷	

3. IP Address kelas C

   Jika bit pertama dari IP Address adalah 110,maka network merupakan kelas C. Tiga bit awal dan 21 bit berikutnya (24 bit pertama) merupakan bit network sedangkan 8 bit terakhir merupakan bit host. Dengan demikian terdapat lebih dari 2 juta network kelas C (32 x 256 x 256), yakni dari nomor 192.0.0.xxx sampai 223.255.255.xxx.setiap network kelas C hanya mampu mendukung sekitar 256 host. Ilustrasinya dapat dilihat pada tabel 2.3 berikut

Tabel 2.3 Struktur IP Address Kelas C

192-223	**0-255**	**0-255**	0-255
110nnnnn	**nnnnnnnn**	**nnnnnnnn**	hhhhhhhh
Bit-bit network ⟷			Bit–bit host ⟷

2.5 Router

Router merupakan sebuah *device* atau alat yang dapat menghubungkan dua atau lebih jaringan komputer yang berbeda. Secara umum *router* adalah alat pada suatu jaringan komputer yang bekerja di *network layer* pada lapisan OSI. Dalam *router* ini terdapat *routing table* yaitu tabel yang berisi alamat-alamat jaringan yang dibutuhkan untuk menetukan tujuan dari paket-paket data yang akan dilewatkan pada suatu jaringan tersebut[6].

Untuk membuat *router*, bisa memanfaatkan suatu jenis sistem operasi Windows, Unix, Linux atau jenis sistem operasi lain pada komputer PC dengan hanya menambahkan dua buah *network interface card* (NIC).

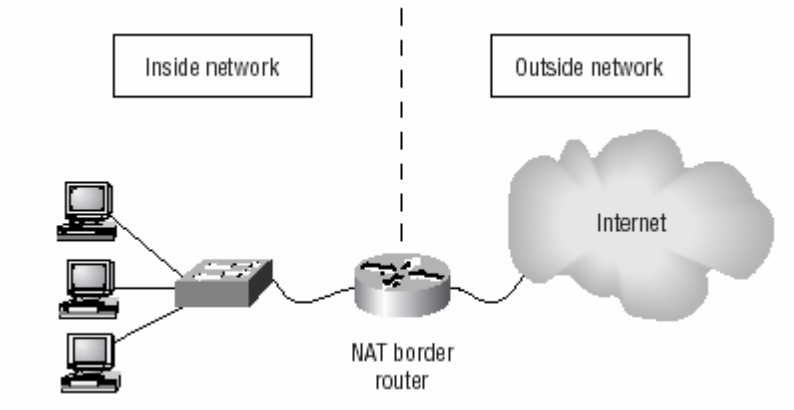
Di dalam PC router dapat di pasang beberapa aplikasi yang dibutuhkan untuk aplikasi webHTB supaya dapat berjalan dengan baik pada sistem.Salah satu diantaranya adalah *Webserver Apache ,MySQL* dan *Network Address Translation* (NAT).

2.5.1 Network Address Translation (NAT)

Keterbatasan alamat IPv.4 merupakan masalah pada jaringan global atau Internet. Untuk memaksimalkan penggunakan alamat IP yang diberikan oleh *Internet Service Provider* (ISP) dapat digunakan *Network Address Translation* atau NAT. NAT membuat jaringan yang menggunakan alamat lokal (*private*), alamat yang tidak boleh ada dalam tabel routing internet dan dikhususkan untuk jaringan lokal/intranet, dapat berkomunikasi ke internet dengan cara memetakan alamat IP internet yang dialokasikan oleh ISP.

2.5.2 Cara Kerja Network Address Translation (NAT)

NAT dikonfigurasikan di *router* atau *router* prosesor diperbatasan antara *inside network* dan *outside network*. *Router* memisahkan antara *inside* dengan *outside network*. NAT menterjemahkan *inside local* IP address menjadi *inside* global IP *address* yang memperkenankan data untuk mengalir ke *outside network* (internet).



Gambar 2.7 Mekanisme *Network Address Translation* (NAT)

NAT memberi keuntungan pada faktanya bahwa relative sedikit pengguna jaringan yang menggunakan IP publik untuk setiap hostnya. NAT melakukan ini dengan menggunakan proses *switching* untuk menukar *source address* di paket data, dan dikirimkan melalui *router*. Hal ini memungkinkan jumlah host pada *inside network* lebih banyak dari jumlah alamat IP *registered*nya. Jadi sangat bermanfaat bagi organisansi yang mempunyai sedikit alamat IP *registered* (IP public).

Dua tipe NAT adalah statikdan dinamik yang keduanya dapat digunakan secara terpisah maupun bersamaan yang diantaranya yaitu:

1. Statik

   Static translation terjadi jika mengkonfigurasikan tabel alamat dengan alamat IP secara manual. Satu alamat IP didalam inside network bekerja menggunakan satu alamat IP *registered*, yang secara manual dikonfigurasiakan oleh network administrator untuk mengakses network luar.
2. Dinamik

   Dynamic mapping memungkinkan administrator untuk mengkonfigurasikan satu atau sekelompok alamat IP *registered* di NAT router. Hal ini memungkinkan banyak host internal untuk memanfaatkan satu alamat IP.

2.5.3 Aplikasi NAT di Linux (*IPTables*)

Iptables merupakan salah satu aplikasi NAT yang berada di lingkungan sistem operasi Linux. *Iptables*digunakan untuk administrasi *filtering* paket dan

Network Address Trasnlation (NAT) pada IPv4, gambaran umum NAT pada *iptables* digunakan untuk menerjemahkan *localnon-unique* (ilegal untuk digunakan diinternet) alamat IP menjadi legal *registered internet* IP *address*. Sebelum menempatkan paket dari jaringan lokal ke internet atau jaringan lainnya.

2.6 Kriteria Jaringan

Sebuah jaringan harus mampu memenuhi sejumlah kriteria. Yang paling penting dari ini adalah performansi, kehandalan, dan *security*[4].

Performansi dapat diukur dengan banyak cara, termasuk waktu transit dan waktu respon. Waktu transit adalah jumlah waktu yang dibutuhkan pesan untuk melakukan perjalanan dari satu perangkat ke perangkat lainnya. Sedangkan waktu respon yaitu waktu tanggapan saat paket dikirimkan dengan benar. Kinerja jaringan tergantung pada sejumlah faktor, termasuk jumlah pengguna, jenis media transmisi, kemampuan dari perangkat keras yang terhubung, dan efisiensi perangkat [4].

Selain ketepatan pengiriman, kehandalan jaringan diukur dengan seberapa banyak frekuensi kegagalan, waktu yang dibutuhkan untuk dapat mengembalikan dari kegagalan sistem, dan ketahanan jaringan[4].

Masalah keamanan jaringan termasuk melindungi data dari akses yang tidak diizinkan, melindungi data dari kerusakan dan melaksanakan prosedur untuk pemulihan dari akses yang tidak diizinkan dan kerusakan data[4].

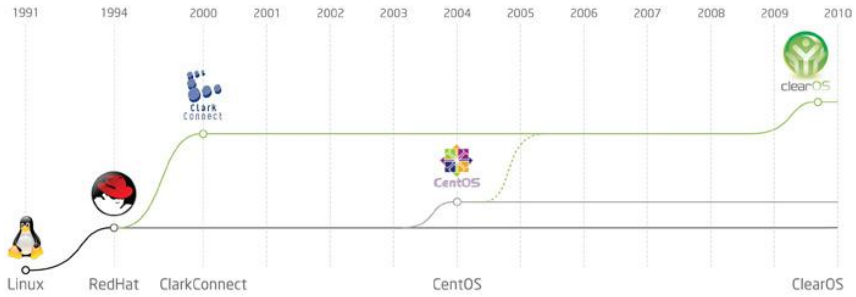
2.7 Bandwidth

Definisi dari *Bandwidth* adalah banyaknya ukuran suatu data atau informasi yang dapat mengalir dari suatu tempat ke tempat lain dalam sebuah network di waktu tertentu. *Bandwidth* dapat dipakai untuk mengukur baik aliran data analog maupun data digital. Sekarang sudah menjadi umum jika kata bandwith lebih banyak dipakai untuk mengukur aliran data digital.

Satuan yang dipakai untuk *bandwidth* adalah *Bits Per Second* atau sering disingkat *Bps*. Seperti diketahui bahwa bit atau binary digit adalah basis angka yang terdiri dari 0 dan 1. Satuan ini menggambarkan berapa banyak bit (angka 0 dan 1) yang dapat mengalir dari satu tempat ke tempat yang lainnya dalam setiap detiknya melalui suatu media[8].

2.8 Sistem Operasi ClearOS

Salah satu komponen utama LAN adalah sistem operasi komputer. Sistem operasi ini berfungsi untuk mengatur komunikasi jaringan berupa dokumen, printer, scanner, dan perangkat-perangkat lainnya. Sistem operasi dapat membedakan arsitektur dalam pemanfaatan fasilitas-fasilitas yang ada di jaringan. Misalkan membedakan perangat jaringan seperti *Ethernet* dan *Token Ring* atau arsitektur lainnya[4].



Gambar 2.8 Timeline Pendistribusian ClearOS

Banyak sekali sistem operasi yang ada pada saat ini. Salah satunya adalah Linux dengan berbagai distro atau distribusinya, seperti RedHat, Mandrake, Debian, SuSe, Slakware, CenOS, ClearOS dan lain sebagainya. Masing-masing distro Linux mempunyai keuntungan tersendiri, tergantung digunakan untuk apa. Dalam perancangan digunakan sistem operasi ClearOS yang di khususkan untuk keperluan *server*.

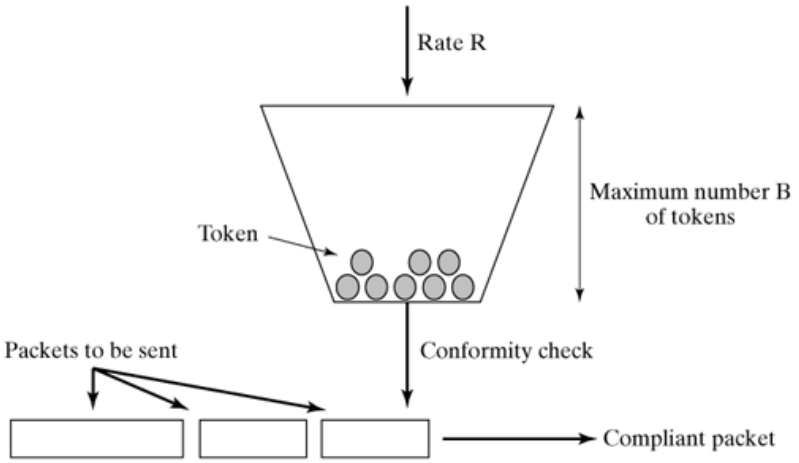
ClearOS Enterprise adalah linux khusus *server*, yang bersifat bebas dan *open source*. Diturunkan langsung dari CentOS yang merupakan hasil kloning dari RHEL (*RedHat Enterprise Linux*) membuat ClearOS mempunyai *Base system* yang kuat.

ClearOS diciptakan sebagai Sistem Operasi untuk Router. Linux ClearOS bersifat bebas untuk versi *Enterprise Edition*, stabil dan konfigurasi dapat dilakukan dengan mudah (*web-based*). ClearOS dikonfigurasi dengan *Linux Command*, ClearOS juga dapat dikonfigurasikan lewat *console* (*text based*). ClearOS tersedia juga dalam versi berbayar dengan beberapa aplikasi tambahan serta *tehnical support online* dari pengembang. fitur yang dapat dikonfigurasikan pada ClearOS adalah *Gateway Service*, *DNS Service*, *Content Filtering*, *Monitoring System,*dan *Web Server***.**

2.9 Teknik Antrian Bandwidth HTB

Hierarchical Token Bucket (HTB) merupakan teknik penjadwalan paket yang diperkenalkan bagi *router* berbasis Linux, dikembangkan pertama kali oleh Martin Devera pada akhir 2001 untuk diproyeksikan sebagai pilihan atau pengganti mekanisme penjadwalan yang saat ini masih banyak dipakai *yaitu Class Based Queueing* (CBQ) memiliki parameter yang banyak daripada HTB. HTB menawarkan kemudahan pemakaian dengan teknik peminjaman dan implementasi pembagian trafik yang lebih akurat[7].

HTB (*Hirarchy Token Bucket*) merupakan salah satu teknik antrian yang memiliki tujuan untuk menerapkan *link sharing* secara akurat dan presisi. Dalam konsep *link sharing*, jika suatu kelas meminta kurang dari jumlah *service* yang telah ditetapkan untuknya, sisa *bandwidth* akan didistribusikan ke kelas-kelas lain yang meminta *service*. HTB menggunakan TBF sebagai estimator yang sangat mudah diimplementasikan. TBF sangat mudah disetting karena banyak dari administrator yang memiliki ilmu tentangnya. Estimator ini hanya menggunakan rate, sebagai akibatnya seorang adminitrastor hanya perlu mereset *rate* yang akan diberikan ke suatu kelas.



Gambar 2.9 Konsep *Token Bucket Filter* (TBF)

Pada HTB ini memiliki *ceil* yaitu parameter yang disetting untuk menetukan peminjaman *bandwidth* antar sehingga kelas akan selalu mendapatkan *bandwidth* di antara *baserate* dan nilai *ceilrate*nya. Parameter ini dianggap sebagai estimator kedua, sehingga setiap kelas dapat meminjam *bandwidth* selama *banwidth* total yang diperoleh memiliki nilai di bawah nilai *ceil*. Hal ini mudah diimplementasikan dengan cara tidak mengijinkan proses peminjaman *bandwidth* pada saat kelas telah melampaui *rate* ini. Sebagai catatan, apabila nilai *ceil* sama dengan nilai *base rate,* maka akan memiliki fungsi yang sama seperti parameter *bounded* pada CBQ, di mana kelas – kelas tidak diijinkan untuk meminjam *bandwidth*. Sedangkan jika nilai *ceil* disetting tidak terbatas atau dengan nilai yang lebih tinggi dari kecepatan link yang dimiliki, maka akan didapat fungsi yang sama seperti kelas *non-bounded*. Sebagai contoh, seorang administrator menjamin *bandwidth* 1Mbit untuk suatu kelas dan mengijinkan *bandwidth* sampai 2 Mbit pada kelas tersebut apabila link dalam keadaan *idle*. Parameter ceil ini sangatlah berguna untuk ISP karena pada ISP kemungkinan besar akan memakainya untuk membatasi jumlah *service* yang diterima oleh suatu pelanggan walaupun pelanggan lain tidak melakukan permintaan *service*.

Untuk menjadwalkan pengiriman paket dari antrian, maka HTB menggunakan suatu proses penjadwalan yang dapat dijelaskan sebagai berikut :

1. **Class** merupakan parameter yang di asosiasikan dengan rate yang menjamin (*assured rate*) AR, *ceil rate* CR, prioritas P, level dan quantum. *Class* dapat

memilih *parent*. Selain AR dan CR, didefinikan juga *actual rate* atau R, yaitu dari aliran paket yang meninggalkan *class* dan diukur pada suatu perioda waktu tertentu.

2. **Leaf** merupakan *class* yang tidak memiliki anak. Hanya *leaf* yang dapat memegang antrian paket.
3. **Level** dari kelas menentukan posisi dimana suatu hirarki. *Leaf-leaf* memiliki level 0, *root class* memiliki level = jumlah level-1 setiap *inner class* memiliki level kurang satu parentnya.

Beberapa hal yang penting berkaitan dengan metode HTB, antara lain :

1. Prioritas *Bandwidth*, merupakan penyampaian tingkat layanan berdasarkan pada arti penting koneksi dan permintaan untuk lalulintas jaringan sehubungan dengan koneksi yang lain. Selama periode lalulintas penuh, akan memperlambat prioritas lebih rendah dari aplikasinnya dengan menurunkan prioritas aplikasinya sehingga meningktkan bandwidth ke aplikasi prioritas yang lebih tinggi.
2. Jaminan *Bandwidth*, merupakan penetapan batas minimum dan jumlah maksimum bandwidth ke saluran spesifik dan koneksi saluran sebetulnya. Dengan meminjam *bandwidth* kelebihan yang tersedia saat itu, sehingga koneksi dapat melebihi dari batas minimum yang dijamin. Rasio garansi juga menjamin prediksi kualitas layanan yang ada dengan membandingkan waktu kritis aplikasi terhadap level penerimaan kostanta layanan selama waktu puncak dan waktu saat tidak terjadi puncak lalulintas data.
3. Request *Bandwidth*, merupakan permintaan *bandwidth* yang minimum *byte* pertama suatu koneksi sampai koneksi diakhiri. Ini akan bermanfaat ketika *bottleneck* yaitu peristiwa terhambatnya proses aliran data.

2.10 Web Server Apache

Web Server adalah software *server* yang menjadi inti dari *World Wide Web* (WWW). *Web server* menunggu permintaan dari *client* yang menggunakan *browser* seperti Netscape Navigator, Internet Explorer, Mozilla, dan program

browser lainnya. Jika ada permintaan dari *browser*, maka *web server* akan memproses permintaan itu dan kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke *browser*. Data ini mempunyai format yang standar disebut dengan format SGML (*Standard General Markup Language*). Data yang berupa format ini kemudian akan ditampilkan oleh *browser* sesuai dengan kemampuan *browser* itu[3].

Web Server untuk berkomunikasi dengan clientnya (web browser) mempunyai protokol sendiri yaitu HTTP (*HyperTextTransfer Protocol*). Dengan protokol ini, komunikasi antar *webserver* dengan clientnya (*browser*) dapat saling dimengerti dan lebih mudah. Kata *Hyper Text* mempunyai arti bahwa seorang pengguna internet dengan web browsernya dapat membuka dan mambaca dokumen-dokumen yang ada dalam komputernya atau bahkan komputer yang jauh tempatnya sekalipun. Hal ini memberikan cita rasa dari suatu proses yang tridimensional, artinya pengguna internet dapat membaca dari satu dokumen ke dokumen yang lain hanya dengan mengklik beberapa bagian dari halaman-halaman dokumen (web) itu.

Apache adalah *Web Server* yang dapat dijalankan pada banyak sistem operasi (Unix, BSD, Linux, dan Microsoft Windows) yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas ini adalah HTTP.

Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigurasi, 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*.

2.11 Pengelolaan Database MySQL

Sebuah *database* adalah sebuah data yang terstruktur. Didalamnya dapat berupa apa saja, dari hal-hal yang sederhana sampai hal-hal yang kompleks. Untuk menambah, mengakses, dan melakukan sebuah proses pada sebuah

komputer *database*, diperlukan sebuah *database* yang dapat melakukan sistem manajemen data seperti *MySQL*. Sejak komputer dapat menangani berbagai sistem data yang cukup besar, manajemen *database* memainkan peranan penting pada komputer tersebut, baik sebagai utilitas yang berdiri sendiri maupun bagian dari aplikasi-aplikasi lainya[3].

Disamping itu, penggunaan *database* mempunyai keuntungan yang lebih, karena data data yang disimpan akan ditempatkan berdasarkan tabel-tabel yang terpisah. Ini akan mempercepat waktu dan lebih fleksibel. Tabel-tabel tersebut kemudian bisa dikaitkan berdasarkan hubungan definisinya sehingga memudahkan pada saat data tersebut diperlukan. Aplikasi *database* yang akan digunakan hanya aplikasi yang cukup popular di pergunakan pada *server-server* yaitu *MySQL*.

MySQL merupakan sebuah sistem manajemen *database open source* yang popular dan gratis untuk platform UNIX. Sistem manajemen *databaseMySQL* menggunakan kumpulan perintah sederhana untuk memasukan, memanggil, menghapus, dan memperbarui data, dengan ini pengguna dapat menggembangkan *database* yang kompleks[3].

Beberapa kemampuan *MySQL* adalah sebagai berikut :

1. *MySQL* bisa di akses dan dimanipulasi dari sejumlah bahasa pemograman terkenal, diantaranya adalah *C*, *C++*, *Java*, *Perl*, *Phyton*, dan *PHP*
2. *MySQL* mendukung tipe data yang umum digunakan, termasuk *FLOAT, DOUBLE, CHART, TEXT, BLOB, DATE SET,* dan *ENUM*
3. *MySQL* memungkinkan alokasi *password* tiap *server*. *Password* yang melalui *MySQL* auntetifikasi terenkripsi.
4. *MySQL* mendukung berbagai macam koneksi, seperti TCP/IP, *socket* UNIX, dan koneksi untuk Windows NT/2000.
5. *MySQL* bisa diperoleh secara gratis termasuk aplikasi-aplikasi lain yang diperlukan dalam memakai *MySQL*.

MySQL juga merupakan salah satu sistem manajemen *database* yang stabil di pasaran. *MySQL* diluncurkan pertama kali pada pertengahan 1996, beberapa kekurangan desaindengan cepat dapat diketahui dan diperbaiki. Sekarang *MySQL*

sudah menjadi sangat stabil dan mampu melakukan penyimpanan data yang besar hingga sanggup menampung data lebih dari 4 Gigabyte. Database *MySQL* mendukung hampir seluruh bahasa pemograman, temasuk PHP dan C.