

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Jaringan Komputer merupakan sekumpulan dua atau lebih komputer serta perangkat lain sebagai pendukung terhubungnya antara komputer satu dengan yang lainnya dalam kesatuan yang bertujuan untuk bertukar data (*sharing data*) dan berbagi *resource* yang dimiliki seperti *file*, printer, atau media penyimpanan. Dengan tujuan membawa informasi secara tepat dan tanpa adanya kesalahan dari sisi pengirim (*transmitter*) menuju kesisi penerima (*receiver*) melalui media komunikasi. Jaringan internet dapat juga diartikan sebagai kumpulan dari beberapa komputer, yang bahkan dapat mencapai jutaan komputer di seluruh dunia yang dapat saling berhubungan serta saling terkoneksi satu sama lainnya. Agar komputer dapat saling terkoneksi satu sama lain, maka diperlukan media untuk saling menghubungkan antar komputer. Media yang digunakan itu bisa menggunakan kabel/serat optik, satelit atau melalui sambungan telepon (Harjono, 2009).

2.1.1 Macam- Macam Jaringan Komputer

1. Berdasarkan Jangkauan Geografis

a) *Local Area Network (LAN)*

Local Area Network (LAN), merupakan sebuah jaringan yang terbatas pada ruangan tunggal dalam satu gedung atau menghubungkan beberapa gedung dalam satu area geografis tertutup (Bonnie Suherman, 2012).

b) *Metropolitan Area Network (MAN)*

Metropolitan Area Network (MAN), pada dasarnya merupakan versi *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.

c) *Wide Area Network (WAN)*

Wide Area Network (WAN), 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. Berdasarkan Distribusi Sumber Informasi.

a) Jaringan Terpusat

Yang dimaksud jaringan terpusat adalah jaringan yang terdiri dari komputer *client* dan komputer server dimana komputer *client* bertugas sebagai perantara dalam mengakses sumber informasi / data yang berasal dari komputer server. Dalam jaringan terpusat, terdapat istilah *dumb* terminal (terminal bisu), dimana terminal ini tidak memiliki alat pemroses data.

b) Jaringan Terdistribusi

Jaringan ini merupakan hasil perpaduan dari beberapa jaringan terpusat sehingga memungkinkan beberapa komputer server dan *client* yang saling terhubung membentuk suatu sistem jaringan tertentu.

3. Berdasarkan Media Transmisi Data yang Digunakan

a) Jaringan Berkabel (*Wired Network*)

Media transmisi data yang digunakan dalam jaringan ini berupa kabel. Kabel tersebut digunakan untuk menghubungkan satu komputer dengan komputer lainnya agar bisa saling bertukar informasi/ data atau terhubung dengan internet. Salah satu media transmisi yang digunakan dalam *wired network* adalah kabel *UTP*.

b) Jaringan Nirkabel (*Wireless Network*)

Dalam jaringan ini diperlukan gelombang elektromagnetik sebagai media transmisi datanya. Berbeda dengan jaringan berkabel (*wired network*), jaringan ini tidak menggunakan kabel untuk bertukar informasi/data dengan komputer lain melainkan menggunakan

gelombang elektromagnetik untuk mengirimkan sinyal informasi / data antar komputer satu dengan komputer lainnya. *Wireless adapter*, salah satu media transmisi yang digunakan dalam *wireless network*.

4. Berdasarkan Peranan dan Hubungan Tiap Komputer Dalam Memproses Data.

a) Client-server

Yaitu jaringan komputer dengan komputer yang didedikasikan khusus sebagai server. Sebuah *service*/layanan bisa diberikan oleh sebuah komputer atau lebih. Contohnya adalah sebuah domain seperti *www.detik.com* yang dilayani oleh banyak komputer web server. Atau bisa juga banyak *service*/layanan yang diberikan oleh satu komputer. Contohnya adalah *server* dengan multi *service* yaitu *mail server*, *web server*, *file server*, *database server* dan lainnya.

b) Jaringan Peer to Peer

Dalam jaringan ini, masing-masing komputer, baik itu komputer *server* maupun komputer *client* mempunyai kedudukan yang sama. Jadi, komputer *server* dapat menjadi komputer *client*, dan sebaliknya komputer *client* juga dapat menjadi komputer server. Contohnya dalam file *sharing* antar komputer di Jaringan *Windows Network Neighbourhood* ada 5 komputer (kita beri nama A,B,C,D dan E) yang memberi hak akses terhadap file yang dimilikinya. Pada satu saat A mengakses file share dari B bernama *data.xls* dan juga memberi akses file *network.doc* kepada C. Saat A mengakses file dari B maka A berfungsi sebagai *client* dan saat A memberi akses file kepada C maka A berfungsi sebagai *server*. Kedua fungsi itu dilakukan oleh A secara bersamaan maka jaringan seperti ini dinamakan *peer to peer*.

2.1.2 Manajemen Jaringan

Manajemen jaringan merupakan pengaturan, pengurusan atau pengelolaan elemen pada jaringan agar jaringan tetap dapat digunakan untuk ke depannya (Subramanian, 2000)

Manajemen jaringan dapat dikategorikan menjadi lima macam, yaitu:

1. *Configuration Management* (Manajemen Konfigurasi)
Meliputi pengaturan alamat (*address*) dan perubahan konfigurasi dari jaringan dan komponen-komponen di dalamnya.
2. *Fault Management* (Manajemen Kesalahan)
Meliputi deteksi masalah, pengisolasian kesalahan, dan perbaikan sehingga jaringan dapat kembali ke operasi normal.
3. *Performance Management* (Manajemen Performa)
Meliputi pengaturan tingkah laku performa dari jaringan yang sedang berjalan dimana performa jaringan ini ditampilkan dalam statistik jaringan seperti *traffic volume*, *network availability*, dan *network delay*.
4. *Security Management* (Manajemen Keamanan)
Mengatur keamanan fisik jaringan, akses ke sumber daya jaringan, dan keamanan komunikasi yang terjadi di dalam jaringan.
5. *Accounting Management* (Manajemen Akunting)
Meliputi manajemen keuangan yang dikeluarkan dalam pengelolaan jaringan.

2.2 Teori yang Berkaitan dengan Tema Penelitian

2.2.1 Bandwidth

Secara umum, *bandwidth* dapat diandaikan sebagai sebuah pipa air yang memiliki diameter tertentu. Semakin besar *bandwidth*, semakin besar pula diameter pipa tersebut sehingga kapasitas volume air (dalam hal ini air merupakan data dalam arti sebenarnya) dapat meningkat. Semakin besar *bandwidth* suatu media, semakin tinggi kecepatan data yang dapat dilaluinya. Pengertian *bandwidth* menurut para ahli adalah sebagai berikut:

1. *Bandwidth* ialah lebar komunikasi di antara saluran yang diukur dalam (Hz. Norton dan Kearns 1999).
2. *Bandwidth* ialah jarak dari frekuensi yang ditransmisikan tanpa menyebabkan sinyal menjadi lemah, (Tanenbaum, 2003).

***Bandwidth* dapat dikategorikan menjadi dua macam :**

1. *Digital bandwidth*

Digital bandwidth merupakan jumlah atau volume data yang dapat dikirimkan melalui sebuah saluran komunikasi dalam satuan *bits per second* tanpa distorsi.

2. *Analog bandwidth*

Analog bandwidth merupakan perbedaan antara frekuensi terendah dengan frekuensi tertinggi dalam sebuah rentang frekuensi yang diukur dalam satuan *Hertz* (Hz) atau siklus per detik, yang menentukan berapa banyak informasi yang bisa ditransmisikan dalam satu saat.

Alokasi atau reservasi *bandwidth* adalah sebuah proses untuk menentukan besar *bandwidth* kepada pemakai dan aplikasi dalam sebuah jaringan. Termasuk di dalamnya menentukan prioritas terhadap berbagai jenis aliran data berdasarkan seberapa penting dan sensitif penundaan terhadap aliran data tersebut. Hal ini memungkinkan penggunaan *bandwidth* yang tersedia secara efisien dan apabila sewaktu-waktu jaringan menjadi lambat, aliran data yang memiliki prioritas yang lebih rendah dapat dihentikan, sehingga aplikasi yang penting dapat tetap berjalan dengan lancar.

Bandwidth merupakan salah satu faktor penting dalam jaringan. Beberapa hal yang menyebabkan *bandwidth* menjadi bagian penting yang harus diperhatikan adalah :

1. *Bandwidth* berdampak pada kinerja sebuah jaringan

Besarnya saluran atau *bandwidth* akan berdampak pada kecepatan transmisi. Data dalam jumlah besar akan menempuh saluran yang memiliki *bandwidth* kecil lebih lama dibandingkan melewati saluran yang memiliki *bandwidth* yang besar. Kecepatan transmisi tersebut

sangat dibutuhkan untuk aplikasi komputer yang memerlukan jaringan terutama aplikasi *real-time*.

2. *Bandwidth* memiliki keterbatasan

Setiap medium yang digunakan untuk mentransmisikan data memiliki batas maksimal *bandwidth* yang dapat dicapai.

3. *Bandwidth* tidak didapatkan dengan gratis

Penggunaan *bandwidth* untuk *LAN* bergantung pada tipe alat atau medium yang digunakan. Umumnya semakin tinggi *bandwidth* yang ditawarkan oleh sebuah alat atau medium, semakin tinggi pula nilai jualnya. Sedangkan penggunaan *bandwidth* untuk *WAN* bergantung dari kapasitas yang ditawarkan dari pihak *ISP*. Perusahaan harus membeli *bandwidth* dari *ISP* dan semakin tinggi *bandwidth* yang diinginkan, semakin tinggi pula harganya.

4. Kebutuhan akan *bandwidth* akan selalu naik

Setiap sebuah teknologi jaringan baru dikembangkan dan infrastruktur jaringan yang ada diperbaharui, aplikasi yang akan digunakan umumnya juga akan mengalami peningkatan dalam hal konsumsi *bandwidth*.

Satuan dasar dari *bandwidth* adalah *bits per second (bps)*. Walaupun satuan dasar yang dipakai *bps*, unit satuan yang lebih besar lebih umum dipakai. *Network bandwidth* biasanya dihitung dalam satuan *thousands bits per second (Kbps)*, *millions bits per second (Mbps)*, *billions bits per second (Gbps)*, dan *trillions bits per second (Tbps)*. Satuan ini umum digunakan dalam pemakaian sehari-hari, terutama karena semakin meningkatnya kebutuhan *bandwidth* dan perkembangan teknologi informasi.

Besarnya *bandwidth* bervariasi tergantung dari tipe medium yang digunakan serta teknologi *LAN* atau *WAN* yang digunakan. Fisik dan medium yang digunakan juga turut mempengaruhi besarnya *bandwidth*. Sinyal data dapat melalui kabel *twisted-pair*, kabel koaksial, kabel serat optik, dan udara. Perbedaan dari bagaimana sinyal tersebut berjalan secara fisik mengakibatkan batasan mendasar terhadap besarnya

kapasitas ditentukan oleh kombinasi dari medium fisik dan teknologi yang dipilih untuk bisa mendeteksi dan mengirimkan sinyal data dalam sebuah jaringan.

2.2.2 Quality of Service (QoS)

Quality of Services adalah sebuah mekanisme layanan standar mutu dari sebuah aplikasi atau produk, *QOS* didalam jaringan komputer digunakan untuk mengukur tingkat kualitas koneksi jaringan (Athailah, 2013)

Kinerja jaringan komputer dapat bervariasi akibat beberapa masalah, seperti halnya masalah bandwidth, latency, dan jitter, yang dapat membuat efek yang cukup besar bagi banyak aplikasi. *Quality of Services (QoS)* ini dapat menjadikan *bandwidth*, *latency*, dan *jitter* dapat diprediksi dan dicocokkan dengan kebutuhan aplikasi yang digunakan didalam jaringan tersebut.

2.2.3 Mikrotik

MikroTik RouterOS™ adalah sistem operasi yang dirancang khusus untuk *network* router. Dengan menggunakan sistem operasi ini, dapat dibuat router dari sebuah komputer. Untuk negara berkembang, solusi MikroTik sangat membantu *ISP* atau perusahaan-perusahaan kecil yang ingin bergabung dengan internet. Walaupun sudah banyak tersedia perangkat router mini sejenis NAT, MikroTik merupakan solusi terbaik dalam beberapa kondisi penggunaan komputer dan perangkat lunak, (Herlambang dan Catur L, 2008).

Mikrotik adalah sebuah perusahaan yang bergerak di bidang produksi perangkat keras (*hardware*) dan perangkat lunak (*Software*) yang berhubungan dengan sistem jaringan komputer yang berkantor pusat di Latvia, bersebelahan dengan Rusia. Mikrotik didirikan pada tahun 1995 untuk mengembangkan router dan sistem *ISP (Internet Service Provider)* nirkabel.

Mikrotik dibuat oleh Mikrotik sebuah perusahaan di kota Riga, Latvia. Latvia adalah sebuah negara yang merupakan “pecahan” dari

negara Uni Soviet dulunya atau Rusia sekarang ini. Mikrotik awalnya ditujukan untuk perusahaan jasa layanan Internet (PJI) atau *Internet Service Provider (ISP)* yang melayani pelanggannya menggunakan teknologi nirkabel atau *wireless*. Saat ini MikroTik memberikan layanan kepada banyak ISP nirkabel untuk layanan akses Internet dibanyak negara di dunia dan juga sangat populer di Indonesia. MikroTik sekarang menyediakan *hardware* dan *software* untuk konektivitas internet di sebagian besar negara di seluruh dunia. Produk *hardware* unggulan Mikrotik berupa Router, Switch, Antena, dan perangkat pendukung lainnya. Sedangkan produk *Software* unggulan Mikrotik adalah MikroTik RouterOS

2.2.3.1 Jenis - Jenis Manajemen *Bandwidth* Pada Mikrotik

Ada lima tipe manajemen *bandwidth* yang dapat dipakai dalam mikrotik, yaitu *queue simple*, *queue tree*, *peer connection queue*, *class based queue*, *hierarchial token bucket*. Lima tipe *management bandwidth* ini memiliki keunggulan masing masing.

1. *Simple Queue*

Simple Queue merupakan cara paling mudah untuk membatasi datarate untuk alamat IP tertentu atau subnet. *Simple Queue* juga dapat digunakan untuk membangun aplikasi *QoS*. *Simple Queue* memiliki fitur yang terintegrasi, yaitu:

- a) Memiliki aturan urutan yang sangat ketat, queue diproses dari mulai yang paling atas sampai yang paling bawah.
- b) Mengatur aliran paket secara *bidirectional* (dua arah).
- c) Mampu membatasi traffic berdasarkan alamat *IP*.
- d) Satu *queue* mampu membatasi *traffic* dua arah sekaligus (*upload/download*).
- e) Bisa menerapkan *queue* yang ditandai melalui paket di *firewall mangle*.
- f) Mampu membagi *bandwidth* secara *fixed*.

g) Sesuai namanya, pengaturannya sangat sederhana dan cenderung statis. Satu item konfigurasi di *Simple Queue* dapat membuat 0-3 *queue* terpisah, satu *queue* di *global-in*, satu *queue* di *global-out*, dan satu *queue* di total global. Jika semua sifat *queue* memiliki nilai default (tidak ada batas yang ditetapkan, jenis *queue* ini adalah *default*), dan *queue* tidak memiliki *child*, maka *queue* tidak benar-benar dibuat. *Simple Queue* memiliki aturan ketat, masing-masing paket harus melalui setiap *queue* sampai memenuhi persyaratan.

2. *Queue Tree*

Queue Tree berfungsi untuk mengimplementasikan fungsi yang lebih kompleks dalam limit *bandwidth* pada mikrotik dimana penggunaan packet mark-nya memiliki fungsi yang lebih baik. Digunakan untuk membatasi satu arah koneksi saja baik itu *download* maupun *upload*. Secara umum *Queue Tree* ini tidak terlihat berbeda dari *Simple Queue*. Perbedaan yang dapat dilihat secara langsung hanya dari sisi cara pakai atau penggunaannya. Dimana *Simple Queue* secara khusus memang dirancang untuk kemudahan konfigurasi sementara *Queue Tree* dirancang untuk melaksanakan tugas antrian yang lebih kompleks.

3. *Peer Connection Queue*

PCQ (*Per Connection Queue*) diperkenalkan untuk mengoptimalkan sistem antrian yang sangat besar. Algoritma *PCQ* akan mengelompokkan aliran data masuk dan membedakan setiap aliran berdasarkan parameter *dst-address*, *src-address*, *dst-port* atau *src-port*. Kemudian algoritma FIFO akan menentukan berapa ukuran antrian yang diijinkan dan melakukan pembatasan pada setiap sub-aliran (individual). Lalu melakukan hal yang sama untuk seluruh sub-aliran (global).

4. *Class Based Queue*

CBQ adalah Teknik klasifikasi paket data yang memungkinkan *sharing bandwidth* antar kelas (*class*) dan memiliki fasilitas user

interface. *CBQ* mengatur pemakaian *bandwidth* jaringan yang dialokasikan untuk tiap *user*, pemakaian *bandwidth* yang melebihi nilai set akan *dipotong (shaping)*, *CBQ* juga dapat diatur untuk *sharing* dan meminjam *bandwidth* antar *class*. *CBQ* adalah algoritma pengaturan lalu lintas jaringan yang dikembangkan oleh *Network Research Group at Lawrence Berkeley National Laboratory* sebagai salah satu alternatif teknologi router based yang masih tradisional.

Konsep kerja *CBQ* dimulai saat *classifier* menentukan paket yang datang dan menempatkan ke kelas yang tepat. Kemudian *general scheduler* menentukan *bandwidth* yang diperuntukkan untuk suatu kelas, *estimator* memeriksa apakah kelas-kelas mendapatkan *bandwidth* sesuai dengan yang dialokasikan. Jika suatu kelas kekurangan maka dengan bantuan link-sharing scheduler kelas yang memiliki *bandwidth* yang tidak terpakai bisa dipinjamkan ke kelas yang membutuhkan tambahan *bandwidth*. *CBQ* membagi *user traffic* ke dalam hirarki *class* berdasarkan *ip address*, *protocol* dan tipe aplikasi. Sebagai contoh hirarki *class* berdasarkan tipe aplikasi, pada perusahaan departemen keuangan tentunya tidak membutuhkan akses internet seperti pada departemen teknisi. Karena setiap perusahaan mempunyai peraturan, kebutuhan bisnis dan kebutuhan vital lain yang berbeda. Hal itulah yang akan mendasari pengelompokan hirarki *class* pada *CBQ*.

5. Hierarcial Token Bucket

Hierarchial Token Bucket (HTB) merupakan teknik penjadwalan paket yang baru-baru ini 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 *CBQ*. *HTB* diklaim menawarkan kemudahan pemakaian dengan teknik peminjaman dan implementasi pembagian trafik yang lebih akurat.

Teknik antrian *HTB* (*Hirarchial Token Bucket*) mirip dengan *CBQ*, perbedaannya terletak pada jenis pilihan yang disediakan. *HTB* memiliki lebih sedikit pilihan saat konfigurasi dan lebih presisi. Teknik antrian *HTB* memberikan fasilitas pembatasan trafik pada setiap level maupun klasifikasi, *bandwidth* yang tidak terpakai bisa digunakan oleh klasifikasi yang lebih rendah (Santoso, 2007). Teknik antrian *HTB* cocok diterapkan perusahaan dengan banyak struktur organisasi.

Rumus alokasi *bandwidth* ,maksimal untuk tiap-tiap *client* pada teknik *HTB* adalah sebagai berikut:

- Apabila nilai **max bandwidth client * jumlah client aktif > bandwidth parent**, maka *bandwidth* ,maksimal untuk tiap-tiap *client* adalah:

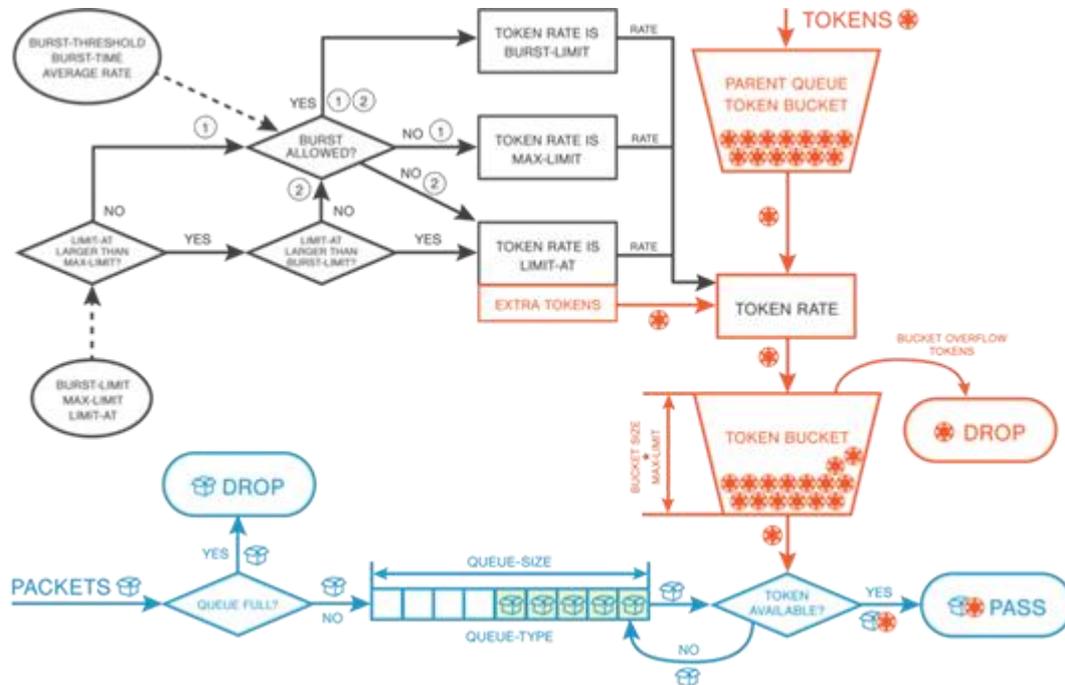
$$MBC - ((MBC * n) - BWP) / n$$

Keterangan:

MBC = max bandwidth client
n = jumlah client aktif
BWP = bandwidth parent

- Apabila nilai **max bandwidth client * jumlah client aktif ≤ bandwidth parent**, maka *bandwidth* ,maksimal untuk tiap-tiap *client* adalah seperti yang diatur pada pengaturan *bandwidth* maksimal untuk tiap-tiap *client*.

Berikut ini algoritma *HTB* :



Gambar 2.1 *Token Bucket Algoritma*

- Token Bucket memiliki algoritma tersendiri untuk mengolah *bandwidth* di tiap satu *rule queue*.
 - Algoritma *Token Bucket* menggunakan analogi ticket/token yang bisa ditampung pada sebuah *buffer* (Bucket)
 - Token dihitung dalam satuan *bytes persecond*
 - Sedangkan *Bucket* adalah *buffer* yang dikalkulasi dari *max-limit*
1. Token ditambahkan setiap $1/r$ detik
 2. Keranjang dapat menampung paling banyak sejumlah b token. Jika token datang ketika keranjang penuh, maka token tersebut akan ditolak
 3. Ketika sebuah paket dari n byte datang, n token dihapus dari keranjang, dan paket akan dikirimkan ke jaringan
 4. Jika lebih sedikit token yang tersedia, tidak ada token yang akan dihilangkan, dan paket data tidak akan diteruskan. Algoritma mengizinkan hingga b byte data, tetapi selama perjalanan paket *output* yangizinkan

dibatasi oleh kecepatan konstan sebesar r . Jika kecepatan *traffic* yang datang melebihi kecepatan yang telah dikonfigurasi, dan tidak terdapat token yang mencukupi dalam keranjang token untuk mengakomodasi *traffic* yang masuk, kelebihan data akan menyebabkan kerusakan dan akan dapat diatasi paket yang tidak diizinkan akan dapat diperlakukan sebagai berikut

1. Paketnya di-drop atau dihilangkan
2. Paket akan mengantri sampai waktu tertentu ketika tersedia token yang cukup untuk ditransmisikan.
3. Ditransmisikan namun diberi tanda sebagai paket yang tidak diizinkan sehingga ketika jaringan penuh atau overload, paket tersebut akan di-drop

yang perlu diingat pada *HTB* adalah :

1. *HTB* hanya bisa berjalan, apabila *rule queue client* berada di bawah setidaknya 1 *level parent*, setiap *queue client* memiliki parameter *limit-at* dan *max-limit*, dan *parent queue* harus memiliki besaran *max-limit*.
2. Jumlah seluruh *limit-at client* tidak boleh melebihi *max-limit parent*.
3. *Max-limit* setiap *client* harus lebih kecil atau sama dengan *max-limit parent*.
4. Untuk *parent* dengan level tertinggi, hanya membutuhkan *max-limit* (tidak membutuhkan parameter *limit-at*).
5. Untuk semua *parent*, maupun *sub parent*, parameter *priority* tidak diperhitungkan. *Priority* hanya diperhitungkan pada *child queue*.
6. Perhitungan *priority* baru akan dilakukan setelah semua *limit-at* (baik pada *child queue* maupun *sub parent*) telah terpenuhi.

2.3 .Penelitian Sebelumnya

Dalam penelitian yang dilakukan oleh Rico Renata Putra (2014), dengan judul “Analisis Dan Perancangan Mikrotik Pada Warnet ExNet II” peneliti menggunakan *Router* Mikrotik RB750 untuk mengoptimasi jaringan internet di warnet dengan membangun sistem manajemen *bandwidth* dan *blocking* situs. Metode yang digunakan pada penelitian ini adalah dengan melakukan analisis, observasi, studi literatur, perancangan topologi baru dan evaluasi. Untuk membangun sistem manajemen *bandwidth*, peneliti

menggunakan metode *simple queue*. Peneliti membagi *bandwidth per-client* sebesar *1Mbps*. Pembagian *bandwidth* dilakukan dengan menggunakan *winbox*. Untuk membuktikan penulis melakukan uji coba pertama dengan men *download* sebuah *file* berukuran *105 MB* dari sebuah website dan melihat hasil limitasi *bandwidth* yang telah dibuat apakah sesuai dengan yang diharapkan. Pengujian kedua dilakukan dengan membuka situs yang telah di *block* pada mikrotik dan melihat hasil apakah situs yang sudah di *block* pada mikrotik sesuai dengan konfigurasi. Dari pengujian yang dilakukan, didapatkan hasil kecepatan rata-rata data transfer menunjukkan hasil pengukuran sesuai dengan *limit bandwidth* yang ditetapkan dan situs yang sudah di *block* telah sesuai dengan yang ditetapkan .

Dalam penelitian yang dilakukan oleh Tb. A. Hizbullah A (2012), dengan judul “Optimalisasi *Bandwidth* dan Keamanan Jaringan Dengan Filterisasi Pada Warung Internet Menggunakan Mikrotik Routerboard”, peneliti menggunakan Mikrotik Routerboard untuk mengoptimalkan jaringan warnet dan membangun sistem *bandwidth* management pada warnet tersebut dengan menggunakan metode *queue tree* agar *bandwidth* yang digunakan menjadi lebih efisien. Konfigurasi pada metode *queue tree* dilakukan dengan mengatur *mangle* terlebih dahulu. Pada *mangle* akan dibuat dua buah *mark*, yaitu *mark packet* dan *mark connection*. Baik konfigurasi pada *mangle* maupun pada *queue tree* akan dibuat sesuai jumlah komputer yang akan digunakan. Pengujian dilakukan dengan menggunakan 3 buah komputer sebagai parameter keberhasilan. Masing-masing komputer akan dicoba untuk melakukan proses mengunduh sejumlah data dari internet. Pengujian dilakukan sebanyak 2 kali. Pada pengujian pertama, nilai level prioritas akan disamakan, sedangkan pada pengujian kedua nilai level prioritas masing-masing komputer akan dibedakan. Pengujian ini ditujukan untuk membuktikan apakah *bandwidth* terbagi secara merata dan apakah level prioritas berjalan dengan baik.