

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian ini menggunakan beberapa jurnal terkait, yaitu: Implementasi *Quality of Service* Dengan Metode HTB (*Hierarchical Token Bucket*) pada PT.

Komunika Lima Duabelas yang ditulis oleh Yunus Arifin dari Universitas Udayana, Manajemen *Bandwidth* Dengan Metode HTB (*Hierarchical Token Bucket*) Pada Sekolah Menengah Pertama Negeri 5 Semarang yang ditulis oleh Alfon Indra Wijaya dan L. Budi Handoko, M.Kom dari Universitas Dian Nuswantoro dan Manajemen *Bandwidth* untuk meningkatkan *Quality of Service* yang ditulis oleh Muhd. Iqbal dan Anwar dari Universitas Sumatra Utara Medan.

Menurut Yunus Arifin, pada penelitian Implementasi *Quality of Service* Dengan Metode HTB (*Hierarchical Token Bucket*) pada PT. Komunika Lima Duabelas:

Sering kita menemukan pengguna yang tidak di ketahui karena tidak adanya sistem yang mengatur itu sehingga membuat sembarang orang dapat masuk dan menggunakan *bandwidth* pada suatu tempat dengan seenaknya dan merugikan pihak tertentu. *Quality of Service* (QoS) bukan membatasi tetapi lebih kepada menjaga kualitas *bandwidth*, tanpa adanya *Quality of Service* dalam sebuah Jaringan Intranet mengakibatkan ketidaksinambungan *bandwidth* yang diterima *client*. Berdasarkan hasil penelitian yang telah dilakukan, Setiap paket memperoleh *bandwidth* minimal pada CIR (*Committed Information Rates*) dan lebih dari CIR tetapi tidak melebihi MIR (*Maximum Information Rates*), Selama

traffic pada *parentnya* tidak penuh, serta pemerataan *bandwidth* sesuai prioritasnya saat kondisi *traffic* seluruh paket penuh. *Hierarchical Token Bucket* (HTB) mampu memaksimalkan *bandwidth* yang tidak terpakai, sehingga kualitas pelayanan menjadi lebih meningkat.

Menurut Alfon Wijaya dan L. Budi Handoko, M.Kom, pada penelitian Manajemen *Bandwidth* Dengan Metode HTB (*Hierarchical Token Bucket*) pada Sekolah Menengah Pertama Negeri 5 Semarang:

Kemudahan akses internet itu tidak di iringi dengan meningkatnya jumlah *bandwidth* yang disediakan oleh operator. Mahalnya harga *bandwidth* menyebabkan pembatasan jumlah *bandwidth* yang diberikan oleh operator. Tanpa adanya manajemen *bandwidth*, banyak komputer yang dapat menggunakan internet secara tidak beraturan sehingga menyebabkan komputer yang lain tidak mendapat jatah *bandwidth* yang adil. Tujuan dari penelitian ini adalah menerapkan kontrol penggunaan internet, menerapkan manajemen *bandwidth* dan menstabilkan koneksi internet. Metode yang menjadi dasar dalam penelitian ini adalah HTB (*Hierarchical Token Bucket*) yaitu metode manajemen *bandwidth* yang mempunyai kelebihan dalam pembatasan trafik pada tiap level maupun klasifikasi, sehingga *bandwidth* yang tidak dipakai oleh level yang tinggi dapat digunakan atau dipinjam oleh level yang lebih rendah.

Menurut Muhd. Iqbal dan Anwar, pada penelitian Manajemen *Bandwidth* untuk Meningkatkan *Quality Of Service* (QoS):

Perkembangan konten internet yang begitu pesat membuat kebutuhan *bandwidth* yang besar tetapi semua itu akan berpengaruh kepada *budget* yang akan

dikeluarkan untuk menyediakan *bandwidth*. Untuk itu diperlukan suatu metode untuk optimalisasi *bandwidth* sesuai dengan *budget* yang tersedia. Manajemen *bandwidth* merupakan solusi yang tepat untuk mengatasi hal tersebut. Dalam penerapan manajemen *bandwidth* tidak serta merta dapat diterapkan dengan perlakuan yang sama pada semua instansi tetapi akan dilihat berdasarkan kebutuhan dari instansi tersebut dan berpatokan pada aturan-aturan dalam instansi tersebut. Untuk itu perlu dilakukan analisa-analisa sebelum menetrapkan manajemen *bandwidth*. Tujuan akhirnya yaitu meningkatkan *quality of service* sehingga tiap-tiap *user* didalam jaringan mendapatkan kualitas koneksi yang baik.

2.2 Landasan Teori

2.2.1 Jaringan Komputer

Jaringan Komputer dapat diartikan sebagai dua atau lebih komputer beserta perangkat-perangkat lainnya yang dihubungkan agar dapat saling berkomunikasi dan bertukar informasi, sehingga membantu menciptakan efisiensi dan optimalisasi dalam bekerja (Norton, 1995, p5)

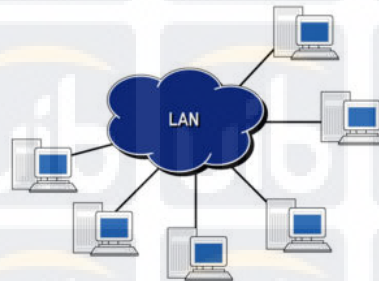
2.2.2 Klasifikasi Jaringan

Terdapat 3 (tiga) tipe jaringan yang sering digunakan, yaitu:

1. *Local Area Network* (LAN)

Menurut Stalling (2004, p16) *Local Area Network* (LAN) adalah suatu jaringan komunikasi yang saling menghubungkan berbagai jenis perangkat dan menyediakan suatu pertukaran data diantara perangkat-perangkat tersebut. LAN biasanya menghubungkan dua atau lebih komputer dan alat-alat yang terhubung dalam sebuah area geografis

yang terbatas (sampai beberapa kilometer), berkecepatan tinggi, dan memiliki error yang rendah dalam suatu perusahaan.



Gambar 2.1 Local Area Network

(sumber: <http://www.hill2dot0.com/wiki/index.php?title=LAN>)

2. Metropolitan Area Network (MAN)

Metropolitan Area Network (MAN) biasanya mencakup area metropolitan yaitu sebuah area yang biasanya lebih besar dari LAN tetapi lebih kecil dari WAN, misalnya antar wilayah dalam satu provinsi (Lammle, 2005, p674). MAN juga dapat menghubungkan beberapa LAN menjadi suatu bagian jaringan yang lebih besar lagi. Cakupan geografis dari MAN itu sendiri tidak menghubungkan area geografis yang berbeda.



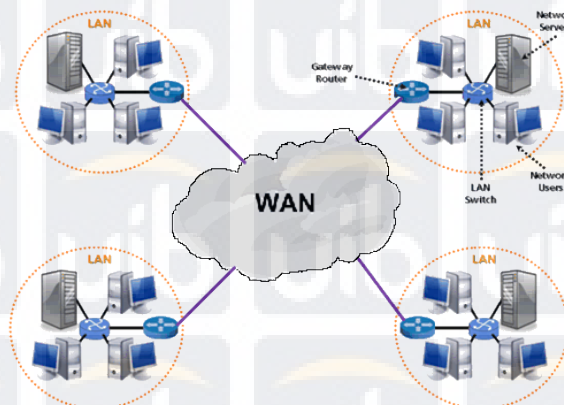
Gambar 2.2 Metropolitan Area Network (MAN)

(sumber: <https://networkschool.files.wordpress.com/2008/03/man.gif>)

3. Wide Area Network (WAN)

Menurut Mann Rubinson (1999, p14) WAN menghubungkan komputer

dan alat-alat lainnya yang terpisah dengan jarak yang sangat besar. WAN dapat melakukan transmisi antar kota, wilayah, dan atau negara. WAN dapat dibuat dari penggabungan fasilitas jalur publik dan pribadi dan komponen-komponen seperti *router*, *bridges*, *terminal*, dan *host*. WAN digunakan untuk menghubungkan jaringan lokal yang satu dengan yang lainnya, sehingga pengguna atau komputer di lokasi yang satu dapat berkomunikasi dengan pengguna atau komputer di lokasi lain.



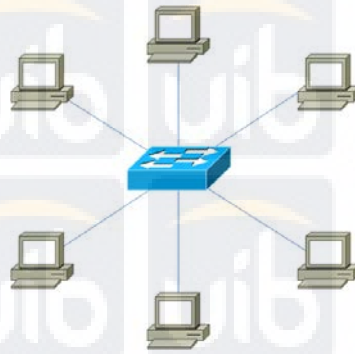
Gambar 2.3 Wide Area Network (WAN)

(sumber: <http://www.netprivateer.com/images/lanwan.gif>)

2.2.3 Topologi Jaringan Komputer

Terdapat beberapa topologi jaringan komputer, yakni:

1. Topologi *Star*



Gambar 2.4 Topologi Star

(sumber:<http://blog.boson.com/Portals/70217/images/startopology.png>)

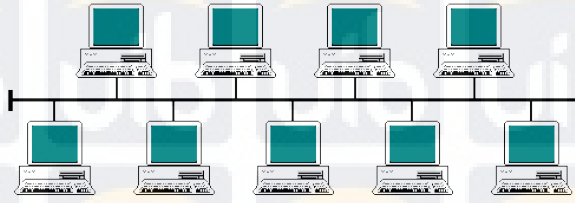
Topologi *star* merupakan salah satu bentuk topologi jaringan yang berupa konvergensi dari *node* tengah ke setiap *node* atau pengguna.

Topologi jaringan *star* termasuk topologi jaringan dengan biaya menengah.

Berikut adalah kelebihan dan kekurangan topologi *star*:

- Tingkat keamanan termasuk tinggi.
- Tahan terhadap lalu lintas jaringan yang sibuk.
- Penambahan dan pengurangan *station* dapat dilakukan dengan mudah.
- Kerusakan pada satu saluran hanya akan mempengaruhi jaringan pada saluran tersebut.
- Jika *node tengah* mengalami gangguan, maka keseluruhan jaringan akan terpengaruh.

2. Topologi Bus



Gambar 2.5 Topologi Bus

(sumber:http://www.technologyuk.net/telecommunications/networks/images/bus_topology.gif)

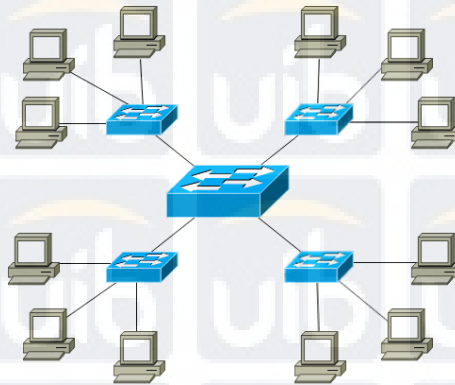
Pada topologi bus, kedua ujung jaringan diakhiri dengan *terminator*. *Barel connector* dapat digunakan untuk memperluasnya. Jaringan ini hanya terdiri dari satu saluran kabel yang menggunakan kabel BNC. Komputer yang ingin terhubung ke jaringan dapat mengkaitkan dirinya dengan men-tap *Ethernet* sepanjang kabel.

Instalasi topologi jaringan bus sangat sederhana, murah dan maksimal. Terdiri atas lima atau lebih komputer. Kesulitan yang sering terjadi ialah kemungkinan terjadinya tabrakan data karena mekanisme jaringan relatif sederhana dan jika salah satu *node* terputus, maka akan mengganggu kinerja dan trafik jaringan secara keseluruhan.

3. Topologi *Extended Star*

Topologi jaringan yang dimana jaringannya didasarkan pada topologi *star* fisik yang memiliki satu atau lebih *repeater* antara titik pusat.

Repeater digunakan untuk memperluas jarak transmisi maksimum dari *point-to-point link* antar pusat dan *node-node* yang tersebar.



Gambar 2.6 Topologi Extended Star

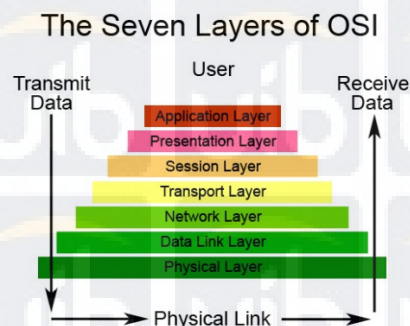
(sumber: <http://blog.boson.com/Portals/70217/images/9%20-%20extended%20star%20topology.png>)

2.2.4 Arsitektur Protokol Jaringan

Terdapat 2 (dua) arsitektur protokol jaringan, yakni:

1. Model referensi OSI

Model referensi OSI adalah sebuah model arsitektural jaringan yang dikembangkan oleh badan *International Organization for Standardization* (ISO) di Eropa pada tahun 1977. OSI sendiri merupakan singkatan dari *Open System Interconnection*. Model ini disebut juga dengan model *OSI seven layer*.



Gambar 2.7 7 layer dalam OSI

(sumber: <http://www.windownetworking.com/img/upl/image0011210155736818.jpg>)

7 layer dalam model OSI, yaitu sebagai berikut.

- *Physical Layer*: Berfungsi untuk mendefinisikan media transmisi jaringan, metode pensinyalan, sinkronisasi bit, arsitektur jaringan, topologi jaringan dan pengkabelan. Adapun perangkat-perangkat yang dapat dihubungkan dengan *Physical layer* adalah NIC (*Network Interface Card*) berikut dengan kabel-kabelnya.

- *Data Link Layer*: Berfungsi untuk menentukan bagaimana bit-bit data dikelompokkan menjadi format yang disebut sebagai *frame*.

Pada *layer* ini terjadi koreksi kesalahan, *flow control*, pengalamatan perangkat keras seperti halnya *MAC Address*, dan menentukan bagaimana perangkat-perangkat jaringan seperti HUB,

Bridge, *Repeater*, dan *Switch layer 2 (Switch un-manage)* beroperasi. Spesifikasi IEEE 802, membagi *layer* ini menjadi dua *layer* anak, yaitu lapisan *Logical Link Control (LLC)* dan lapisan *Media Access Control (MAC)*.

- *Network Layer*: Berfungsi untuk mendefinisikan alamat-alamat IP, membuat *header* untuk paket-paket, dan kemudian melakukan routing melalui *internetworking* dengan menggunakan *Router* dan *Switch layer-3 (Switch Manage)*.

- *Transport Layer*: 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.

Selain itu, pada *layer* ini juga membuat sebuah tanda bahwa paket

diterima dengan sukses (*acknowledgement*), dan mentransmisikan ulang terhadap paket-paket yang hilang di tengah jalan.

- *Session Layer*: Berfungsi untuk mendefinisikan bagaimana koneksi dapat dibuat, dipelihara, atau dihancurkan. Selain itu, di *layer* ini juga dilakukan resolusi nama.
- *Presentation Layer*: Berfungsi untuk mentranslasikan data yang hendak ditransmisikan oleh aplikasi ke dalam format yang dapat ditransmisikan melalui jaringan. Protokol yang berada dalam *layer* ini adalah perangkat lunak redirektor (*redirector software*), seperti layanan *workstation* (dalam Windows NT) dan juga *Network shell* (semacam *Virtual Network Computing* (VNC) atau *Remote Desktop Protocol* (RDP)).
- *Application Layer*: Berfungsi sebagai antarmuka dengan aplikasi dengan fungsionalitas jaringan, mengatur bagaimana aplikasi dapat mengakses jaringan, dan kemudian membuat pesan-pesan kesalahan. Protokol yang berada dalam *layer* ini adalah HTTP, FTP, SMTP, dan NFS.

2. Model Referensi TCP/IP

Model TCP/IP adalah kerangka kerja untuk deskripsi protokol jaringan komputer yang dibuat pada tahun 1970-an oleh DARPA, Departemen Pertahanan Amerika Serikat. Model TCP/IP memiliki 4 layer abstrak, yaitu:

- *Layer Application*

Terletak di atas pada model TCP/IP dan merupakan tempat program software memperoleh akses ke jaringan. Layer ini kira-kira sesuai dengan layer Session, Presentation, dan layer Application pada model OSI.

- *Layer Transport*

Menghadirkan sesi komunikasi diantara komputer-komputer dan menentukan tipe layanan transport baik berorientasi koneksi (TCP) ataupun berorientasi datagram yang tanpa koneksi (UDP).

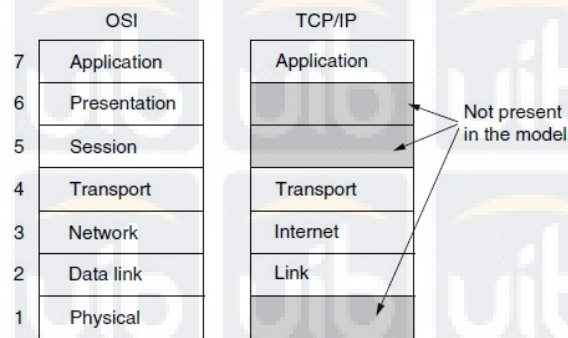
- *Layer Internet*

Mengkapsulasi paket-paket menjadi datagram internet dan mengoperasikan semua algoritma *routing* yang diperlukan. Layer

Internet memiliki persamaan dengan layer Network pada model OSI.

- *Layer Interface Network*

Memiliki persamaan dengan *layer Data Link* dan *layer physical* pada model OSI. *Layer Interface Network* bertanggung jawab untuk mengirimkan dan menerima *frame*. *Layer Interface Network* meletakkan *frame* pada jaringan dan mengeluarkan *frame* dari jaringan.



Gambar 2.8 Model TCP/IP

(sumber: <http://elkaasik.com/wp-content/uploads/2014/07/tcp-ip-model.jpg>)

2.2.5 Peralatan Jaringan

Terdapat beberapa peralatan yang berperan penting dalam jaringan, yakni:

1. *Network Interface Card* (NIC)

Menurut Alexs Wardhana (2011): NIC adalah bagian *hardware* komputer yang dirancang agar komputer dapat berkomunikasi dalam jaringan komputer. Merupakan perangkat *layer 1* dan *layer 2* dalam OSI *layer*. Setiap *network card* memiliki *serial number* unik 48 bit yang disebut *Media Access Control* (MAC). Fungsi MAC adalah memberi tahu NIC apa yang sedang berjalan dalam jaringan MAC diatur oleh IEEE.

Sesuai perkembangan teknologi jaringan, terdapat banyak jenis dan merk kartu jaringan (NIC). Ada tiga hal pokok yang perlu diketahui dari kartu jaringan atau NIC ini, yaitu tipe kartu, jenis protokol, dan juga tipe kabel yang digunakan. Pada komputer lama, kartu jaringan tersambung dengan *motherboard* secara eksternal melalui *slot* atau *port* usb. Namun komputer sekarang ini, NIC sudah terintegrasi langsung

dengan *motherboard*.

Berikut ada beberapa tipe NIC yang biasa digunakan pada skenario-skenario berbeda, yaitu:

- *10/100 Ethernet*: biasanya digunakan di rumah atau komputer kantor yang berskala kecil. Diberi nama 10/100 karena jarak speednya adalah 10 hingga 100 megabit. Terhubung ke *motherboard* menggunakan slot PCI.
- *Gigabit Ethernet*: Kartu jaringan ini menyediakan kecepatan 1 gigabit per detik. Menggunakan slot PCI. Biasanya digunakan untuk menghubungkan kabel fiber optik dari *web server* dan tempat penyimpanan data.
- *Fiber optic NIC*: digunakan untuk perusahaan besar. Menghubungkan kabel fiber optik, dan menyediakan kecepatan sampai dengan 10 gigabit per detik.
- *Wireless NIC*: Karena penggunaan *wireless* yang semakin terkenal, penggunaannya semakin meningkat secara drastis. *Wireless NIC* menyediakan servis yang sama seperti NIC yang menggunakan kabel lainnya, namun tanpa menggunakan kabel apapun ke komputer. *Wireless NIC* menyediakan kecepatan hingga 54 MB per detik.

2. Router

Sebuah alat yang mengirimkan paket data melalui sebuah jaringan atau Internet menuju tujuannya, melalui sebuah proses yang dikenal sebagai

penghalaan. Proses penghalaan terjadi pada lapisan 3 (Lapisan jaringan seperti *Internet Protocol*) dari protokol tumpukan (*stack protocol*) tujuh-lapis OSI.



Gambar 2.9 Mikrotik RouterBoard 750

Sumber: www.mikrotik.com

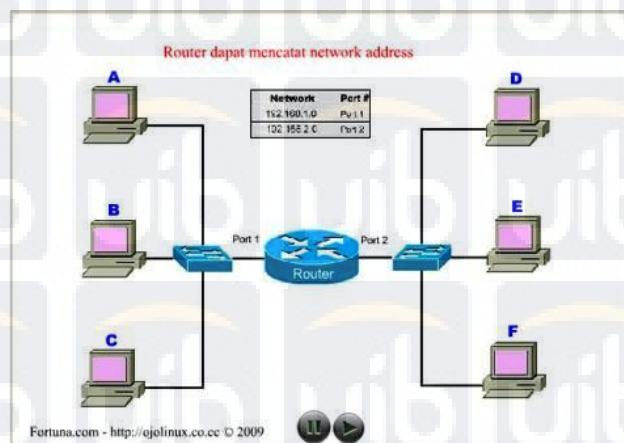
Router berfungsi sebagai penghubung antar dua atau lebih jaringan untuk meneruskan data dari satu jaringan ke jaringan lainnya. *Router* sering digunakan dalam jaringan berbasis teknologi protokol TCP/IP, dan router jenis ini disebut juga dengan *IP router*. *Router* digunakan untuk menghubungkan banyak jaringan kecil ke sebuah jaringan yang lebih besar, yang disebut dengan *internetwork*, ataupun membagi sebuah jaringan besar ke dalam beberapa *subnetwork* untuk meningkatkan kinerja dan juga mempermudah manajemennya.

Router terbagi dalam dua jenis, yakni:

- *Router Statis*: *router* yang memiliki *tabel routing* statis yang di-*setting* secara manual.
- *Router Dinamis*: *router* yang memiliki dan membuat *tabel routing* dinamis dengan cara mendengarkan lalu lintas jaringan dan juga dengan saling berhubungan dengan *router* lainnya.

Prinsip kerja dari *router* mirip dengan *bridge*, yaitu sama-sama dapat

meneruskan paket data jaringan dan juga dapat membagi jaringan menjadi beberapa segmen atau menyatukan segmen-segmen jaringan. Yang membedakan ialah *router* berjalan pada lapisan ketiga OSI layer (lapisan jaringan/*network*) dan menggunakan skema pengalamatan pada lapisan tersebut seperti *IP address*, sedangkan *bridge* berjalan pada lapisan kedua (*data link*) dan menggunakan skema pengalamatan yang digunakan pada lapisan itu, yakni *MAC address*.



Gambar 2.10 Cara Kerja Router

Pada gambar diatas terdapat 2 buah *network* yang terhubung dengan sebuah *router*. *Network* sebelah kiri yang terhubung ke *port 1* *router* mempunyai alamat *network* 192.168.1.0 dan *network* sebelah kanan terhubung ke *port 2* dari *router* dengan *network address* 192.155.2.0. Komputer A mengirim data ke komputer C, maka *router* tidak akan meneruskan data tersebut ke *network* lain.

Begitu juga ketika komputer F mengirim data ke E, *router* tidak akan meneruskan paket data ke *network* lain.

Ketika komputer F mengirimkan data ke komputer B, maka *router*

akan meneruskan paket data tersebut ke komputer B.

3. HUB / Switch

HUB meneruskan paket dan berperan pada *Layer Physical* OSI model.

HUB hanya meneruskan data tanpa memiliki kecerdasan mengenai alamat-alamat yang dituju. HUB hanya memiliki satu *collision domain*, sehingga walaupun komputer dihubungkan ke port yang berlainan, tetapi tetap pada satu *collision domain*.

Switch merupakan pengganti dari Hub. *Switch* menghubungkan alat-alat jaringan dengan membagi jaringan dengan segmen terpisah. Setiap segmen merupakan satu *collision domain*. *Switch* bekerja pada *layer 2* OSI (*Data Link*). Berbeda dengan HUB, *switch* mampu melakukan pemeriksaan kesalahan pada *frame* yang melaluinya.

4. Modem ADSL

Modem adsl adalah sebuah alat untuk menghubungkan sebuah komputer atau *router* ke sebuah saluran telepon DSL (*Digital Subscriber Line*) menggunakan layanan ADSL (*Asymmetric Digital Subscriber Line*).

2.2.6 Internet Protocol (IP)

Internet Protocol atau disingkat IP adalah protokol lapisan jaringan (*Layer Network* pada OSI) atau protokol lapisan *internetwork* yang digunakan oleh protokol TCP/IP untuk melakukan pengalamatan dan *routing* paket data antar *host-host* di jaringan komputer berbasis TCP/IP. Menurut (Rouse, 2008): *Internet Protocol* (IP) adalah metode atau protokol dimana data dikirim melalui satu

komputer ke komputer lain yang ada di internet. Setiap komputer (disebut juga sebagai *host*) di internet memiliki setidaknya satu *IP address* sebagai identitas unik dari semua komputer yang ada di internet.

Menurut (Feriantano, 2013): *Internet Protocol Address* merupakan singkatan dari *IP address*. Pengertian *IP address* adalah suatu identitas numerik yang dilabelkan kepada suatu alat seperti komputer, *router* atau *printer* yang terdapat dalam suatu jaringan komputer yang menggunakan *internet protocol* sebagai sarana komunikasi.

IP merupakan suatu protokol tanpa koneksi, yang berarti bahwa tidak ada koneksi berkelanjutan diantara masing-masing *host* yang saling berkomunikasi. Tiap paket yang ada di *Internet* dianggap sebagai suatu unit independen dari data tanpa hubungan dengan unit data lainnya. Alasan dari ini adalah dari TCP, yang merupakan protokol berorientasikan koneksi yang terus melacak *sequence* dari tiap-tiap paket dalam sebuah pesan. Fungsi IP antara lain adalah sebagai alat identifikasi *host* atau *interface* pada jaringan dan sebagai alamat lokasi jaringan. IP berada di *layer 3* pada OSI model, yaitu *Networking Layer*.

Ketika mengirim atau menerima data (contohnya sebuah situs web atau e-mail), pesan akan dibagi menjadi bagian-bagian kecil yang disebut dengan paket (*packet*). Setiap paket ini berisi *internet address* dari pengirim dan juga *internet address* penerima. Paket kemudian akan dikirimkan terlebih dahulu ke komputer *gateway*. Komputer *gateway* membaca alamat tujuan dan melanjutkan paket tersebut ke *gateway* berikutnya sampai ia dikenali oleh *gateway* di internet bahwa paket tersebut merupakan untuk komputer tujuan. Kemudian *gateway* akan

melanjutkan paket langsung ke komputer tujuan sesuai dengan *address* yang diberikan sebelumnya.

Dikarenakan pesan dibagi-bagi menjadi beberapa paket, tiap paket mampu dan bila perlu dikirim melalui jalur yang berbeda di Internet. Paket-paket tadi dapat sampai dengan susunan yang berbeda dengan susunan ketika dikirim.

Internet Protocol (IP) hanya mengirim saja, dan semuanya tergantung oleh protokol lainnya, yaitu *Transmission Control Protocol* (TCP) untuk dapat menyusun kembali dengan susunan yang benar.

Dalam menentukan IP, yang harus diperhatikan antara lain:

1. Pengalamatan IP

Pengalamatan IP (IPv4) terdiri dari 4 byte (32 bit) dan dipisahkan oleh titik dengan masing-masing 8 bit. Setiap bit dalam oktet tersebut mempunyai bobot biner (128, 64, 32, 16, 8, 4, 2, 1). Nilai minimum oktet tersebut adalah 0 dan maksimum adalah 255.

Setiap alamat IP ini terdiri dari bagian *network* dan *host*. Bagian *network* adalah alamat yang menandakan alamat jaringan, sedangkan bagian *host* adalah alamat yang menandakan alamat *workstation* tersebut.

2. IP *Subnet Mask*

Suatu alamat IP dapat dibagi menjadi beberapa sub-jaringan dengan cara meminjam bit bagian *host* untuk dijadikan bagian *network*. *Subnet mask* dari IP tersebut diubah menjadi satu, yang menandakan bahwa bit tersebut adalah bagian *network*.

Pada *IP tables* dikenal istilah *chain* yaitu tempat terjadi proses pemfilteran.

Terdapat beberapa *chain*:

- *PREROUTING* - Titik dimana kita bisa memanipulasi paket *network* sebelum dia memasuki keputusan *routing*, apakah ia akan masuk ke dalam atau hanya sekedar 'lewat'.
- *INPUT* - Titik dimana kita bisa melakukan pemeriksaan terhadap paket *network* yang akan masuk.
- *OUTPUT* - Titik dimana kita melakukan pemeriksaan terhadap paket *network* yang dihasilkan sebelum *routing*.
- *FORWARD* - Titik dimana kita melakukan pemeriksaan terhadap paket *network* yang hanya sekedar 'lewat'.
- *POSTROUTING* Titik dimana kita bisa melakukan manipulasi terhadap paket yang akan keluar.

Masing-masing *chain* berisi daftar *rules*. Ketika sebuah paket dikirim ke suatu *chain*, paket ini dibandingkan dengan masing-masing *rule* di dalam *chain* dalam urutan atas ke bawah.

2.2.7 Network Address Translation (NAT)

Network Address Translation (NAT) adalah sebuah *instrument* algoritma untuk meminimalkan kebutuhan untuk pengalamatan IP yang unik secara global, memungkinkan sebuah organisasi yang memiliki alamat-alamat yang tidak unik secara global untuk terhubung ke internet, dengan cara menerjemahkan alamat-alamat tersebut yang bisa di *route* secara global (Lammle, 2005, p678).

2.2.8 Dynamic Host Configuration Protocol (DHCP)

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

2.2.9 Bandwidth

Bandwidth adalah jumlah trafik informasi yang bisa melewati suatu koneksi jaringan pada periode waktu tertentu. *Bandwidth* adalah faktor yang penting yang digunakan untuk menganalisis *performance* jaringan, mendesain jaringan baru, dan memahami internet. Unit dasar *bandwidth* adalah bits per second (bps).

Berikut adalah beberapa pengertian *bandwidth* menurut ahli:

1. Menurut Tanenbaum (2003, p88), *bandwidth* adalah jarak dari frekuensi yang ditransmisikan tanpa menyebabkan signal menjadi lemah.
2. Menurut Setiawan (2013), *bandwidth* adalah suatu ukuran dari banyaknya informasi yang dapat mengalir dari suatu tempat ke tempat lain dalam suatu waktu tertentu. *Bandwidth* dapat diibaratkan sebagai sebuah pipa air memiliki diameter tertentu. Semakin besar *bandwidth*, maka semakin banyak pula air yang dapat mengalir di dalam pipa tersebut.

2.2.10 *Quality of Service (QoS)*

Quality of Service (QoS) didefinisikan sebagai suatu pengukuran tentang seberapa baik jaringan dan merupakan suatu usaha untuk mendefinisikan karakteristik dan sifat dari suatu layanan. Kinerja jaringan komputer dapat bervariasi akibat beberapa masalah, seperti halnya masalah QoS mengacu pada kemampuan jaringan untuk menyediakan layanan yang lebih baik pada trafik jaringan tertentu melalui teknologi yang berbeda-beda.

QoS merupakan suatu tantangan yang besar dalam jaringan berbasis IP dan internet secara keseluruhan. Tujuan dari QoS adalah untuk memenuhi kebutuhan-kebutuhan layanan yang berbeda, yang menggunakan infrastruktur yang sama. QoS menawarkan kemampuan untuk mendefinisikan atribut-atribut layanan yang disediakan, baik secara kualitatif maupun kuantitatif. *Bandwidth*, *latency* dan *jitter*, yang dapat membuat efek yang cukup besar bagi banyak aplikasi. Sebagai contoh, komunikasi suara (seperti VoIP atau *IP Telephony*) serta *video streaming* dapat membuat pengguna frustrasi ketika paket data aplikasi tersebut dialirkan di atas jaringan dengan *bandwidth* yang tidak cukup, dengan *latency* yang tidak dapat diprediksi, atau *jitter* yang berlebihan. Tujuan akhir dari QoS adalah memberikan *network service* yang lebih baik dan terencana dengan *dedicated bandwidth*, *jitter* dan *latency* yang terkontrol dan meningkatkan *loss characteristic*.

2.2.11 Mikrotik

Mikrotik adalah sebuah perusahaan yang bergerak di bidang produksi perangkat keras (*hardware*) seperti *routerBOARD* dan perangkat lunak (*software*)

seperti Mikrotik RouterOS, yang berhubungan dengan sistem jaringan komputer yang berkantor pusat di Latvia, bersebelahan dengan Rusia.

Mikrotik RouterOS merupakan sistem operasi berbasis Linux yang diperuntukkan sebagai *network router*. Didesain untuk memberikan kemudahan bagi penggunanya, konfigurasi dapat dilakukan melalui *Windows application* (*WinBox*). Perangkat usungan mikrotik contohnya yakni RouterBoard.

2.2.11.1 Metode Konfigurasi

1. *Console* – Mikrotik *RouterBoard* ataupun *PC Router* dapat diakses langsung melalui *console / shell* ataupun *remote access* menggunakan *putty*.
2. *WinBox* – *WinBox* digunakan untuk mengakses dan konfigurasi Mikrotik *Router* yang menampilkan fitur-fitur dan konfigurasi mikrotik dengan tampilan antarmuka yang baik atau GUI (*Graphical User Interface*). *Winbox* paling sering digunakan karena memiliki tampilan yang intuitif dan mudah dioperasikan.
3. *Web* – *Mikrotik RouterBoard* ataupun *PC Router* dapat diakses via *web* dengan menggunakan *web browser* (*Mozilla Firefox, Google Chrome, dll.*).

2.2.11.2 Bandwidth Management pada Mikrotik

Bandwidth management adalah proses mengukur dan mengontrol komunikasi (lalu lintas paket) pada *network link* untuk menghindari penggunaan melebihi kapasitas pada *network link* yang dapat mengakibatkan kemacetan jaringan dan kinerja yang buruk. Berikut adalah beberapa fitur mekanisme

manajemen *bandwidth* pada mikrotik:

1. Memprioritaskan beberapa arus paket.
2. Membatasi tingkat data untuk *IP address* tertentu, *subnet*, *protocol*, dan *port*.
3. Menggunakan antrian (*queue*) untuk mempercepat web browsing.
4. Menerapkan antrian pada interval-interval waktu yang pasti.
5. Berbagi lalu lintas yang tersedia diantara para pengguna secara adil sesuai muatan *bandwidth*.

2.2.11.3 Menu Manajemen pada Mikrotik

1. *Interface* – Menu *interface* merupakan gerbang trafik keluar atau masuk ke *Mikrotik*. Secara *default Mikrotik* hanya mengenali *interface* yang secara fisik memang ada. Kita dapat merubah nama *interface* tersebut dengan tujuan untuk memudahkan dalam mengidentifikasi fungsi.
2. *IP* – Menu *IP* adalah menu utama dengan berbagai pilihan yang berhubungan dengan konfigurasi *Internet Protocol*. Didalam Menu *IP* terdapat beberapa sub-menu, yakni:
 - a. Sub-menu *Address*: bagian utama yang digunakan untuk membuat *router* bekerja. Mikrotik saat ini hanya mendukung IPv4 dengan *subnet mask*. Mikrotik dapat menggunakan *IP Address* secara *static* ataupun *dynamic*.
 - b. Sub-menu *Routes*: Sub-menu ini menampilkan kondisi tabel *routing* baik aktif maupun yang cadangan. Daftar *routing* ini bisa

bersifat permanen (*read only*), statis, dan dinamis.

- c. Sub-menu *Firewall*: berisi konfigurasi paket *filter* dan fitur untuk mengatur fungsi keamanan yang juga mengatur arus data dari *router* maupun yang ke *router*. Fungsi *Network Address Translation* juga merupakan *tools* yang digunakan untuk pembatasan akses secara langsung dan melindungi trafik yang akan keluar dari *router*.
- d. Sub-menu *DNS*: digunakan untuk mengurangi trafik DNS ke internet dan mempercepat waktu yang *resolve* dapat digunakan fungsi *DNS cache*. Mikrotik *DNS cache* dapat menggunakan *DNS server primary* dan *secondary*.

3. *Queues – Quality of Service (QoS)* berarti *router* harus melakukan prioritas dan mengatur trafik jaringan. QoS tidak hanya sebatas untuk membatasi tetapi lebih bertujuan untuk meningkatkan dan menjaga kualitas. Mikrotik memiliki mekanisme pengaturan *bandwidth*, antara lain:

- a. Kecepatan data berdasarkan *IP address, subnet, protocol, dan port*.
- b. Penggunaan *burst* untuk meningkatkan kecepatan *Web-Access*.
- c. Pembagian trafik secara merata ke setiap pengguna.

Queuing digunakan saat trafik meninggalkan *router* menuju *interface* fisik atau menuju ke *interface virtual (global-in, global-out, dan global-total)*. Masing-masing *virtual interface* tersebut berfungsi sebagai berikut:

- a. *Global-in*: informasi semua trafik yang diterima semua *interface router* sebelum melalui paket *filter*. *Global-in queuing* dieksekusi setelah *mangle* dan *dst-nat*.
- b. *Global-out*: informasi semua trafik yang keluar dari *interface router*. *Queue* yang dipasang disini akan mengatur trafik sebelum meninggalkan *router*.
- c. *Global-total*: informasi semua trafik yang keluar masuk *interface router*. Jika *queuing* dipasang maka akan membatasi total kecepatan pada kedua arah. QoS dapat beroperasi dengan cara *drop* paket, data tidak akan berpengaruh pada paket TCP karena setiap paket yang di-*drop* akan dikirimkan ulang.

2.2.11.4 Metode Queue

Secara konseptual, *QoS (Quality of Service)* diimplementasikan sebagai mekanisme *queue* (manajemen bagaimana paket menunggu untuk disalurkan ke sebuah *interface*). *Queue* bekerja saat meninggalkan *interface (packet flow)* sehingga bagaimana kita dapat membatasi trafik yang masuk ke dalam *router*. Perlu di ketahui bahwa dalam sebuah *interface* hanya satu disiplin atau aturan yang dapat kita terapkan pada mekanisme *queue*.

Control Queing adalah permintaan dan kecepatan paket yang melalui *interface* serta mendefinisikan mana paket yang akan menunggu atau dikirimkan dan mana yang akan di-*drop*.

Baik *Queue Simple* maupun *Queue Tree*, keduanya sama-sama menerapkan sistem antrian yang didasarkan pada konsep HTB (*Hierarchical*

Token Bucket). Jadi sebenarnya sulit bagi kita menemukan perbedaan diantara keduanya secara sekilas. Perlu analisa yang lebih mendalam karena sebagian besar parameter *Queue Simple* yang juga merupakan properti HTB dapat digunakan atau diterapkan pada *Queue Tree* begitu juga sebaliknya.

Dengan adanya pembaharuan OS Routerboard (versi 6) sebagai alasan penyempurnaan, ada yang dihilangkan dan tambahan beberapa fitur pada *Queue Simple* dan *Queue Tree*, sehingga semakin sulit kita menemukan perbedaan yang sangat signifikan diantara keduanya.

Mungkin satu-satunya perbedaan yang bisa kita lihat langsung yaitu hanya dari sisi cara pakai atau penggunaannya saja. Dimana *Queue Simple* secara khusus memang dirancang untuk kemudahan konfigurasi sementara *Queue Tree* dirancang untuk melaksanakan tugas antrian yang lebih kompleks dan butuh pemahaman yang baik tentang aliran trafik.

Berikut adalah perbandingan antara *simple queue*, dan *queue tree*:

1. Queue Simple

- Memiliki aturan urutan yang sangat ketat, antrian diproses mulai dari yang paling atas sampai yang paling bawah.
- Mengatur aliran paket secara *bidirectional* (dua arah).
- Mampu membatasi trafik berdasarkan alamat IP.
- Satu antrian mampu membatasi trafik dua arah sekaligus (*upload* dan *download*).
- Jika menggunakan *simple queue* dan *queue tree* bersamaan, maka *queue simple* yang akan diproses terlebih dahulu dibandingkan

queue tree.

- Mendukung penggunaan PCQ sehingga mampu membagi *bandwidth* secara adil dan merata.
- Bisa menerapkan antrian yang ditandai melalui paket di *firewall mangle*.
- Mampu membagi *bandwidth* secara fixed.
- Sesuai namanya, pengaturannya sangat sederhana dan cenderung statis, sangat cocok untuk admin yang tidak mau ribet dengan *traffic control* di *firewall mangle*.

2. Queue Tree

- Tidak memiliki urutan. Setiap antrian akan diproses secara bersama-sama.
- Mengatur aliran paket secara *directional* (satu arah).
- Membutuhkan pengaturan *firewall mangle* untuk membatasi trafik per IP.
- Membutuhkan pengaturan *firewall mangle* terlebih dahulu untuk membedakan trafik *download* dan *upload*.
- Dinomorduakan setelah *simple queue*.
- Mendukung penggunaan PCQ sehingga mampu membagi *bandwidth* secara adil dan merata.
- Pengaturan antrian melalui paket yang ditandai di *firewall mangle*.
- Mampu membagi *bandwidth* secara fixed.
- Lebih fleksibel dan butuh pemahaman yang baik di *firewall mangle*

khususnya tentang *traffic control*.

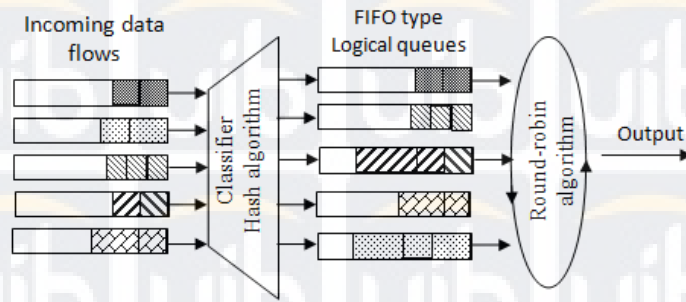
2.2.11.4.1 Scheduler

Scheduler merupakan mekanisme yang digunakan untuk membatasi berdasarkan waktu tunggu paket. Ada beberapa mekanisme *scheduler* yang sering digunakan antara lain:

1. PFIFO – *Packets First-In First-Out*
2. BFIFO - *Bytes First-In First-Out*
3. SFQ - *Stochastic Fairness Queuing*
4. RED – *Random Early Detect*

PFIFO (*Packet-First-In-First Out*) dan BFIFO (*Bytes-First-In-First-Out*) merupakan suatu bentuk *discipline* yang berbasis algoritma FIFO (*First-in-first-out*). Perbedaannya adalah kalau PFIFO perhitungannya diukur menggunakan *packets* sedangkan BFIFO diukur menggunakan *bytes*.

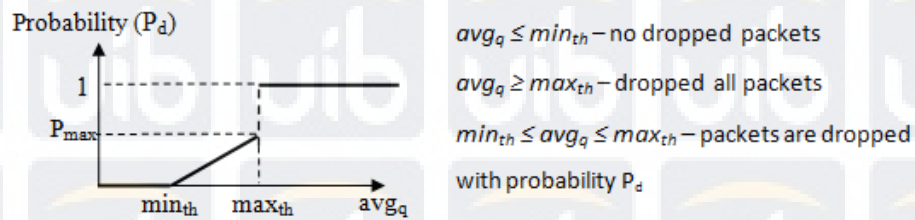
SFQ (*Stochastic Fairness Queuing*) menggunakan *hashing* dan algoritma *Round-Robin*. Nama *stochastic* karena SFQ tidak mengalokasikan antrian (*queue*) pada tiap arus (*flow*), ia memiliki algoritma yang membagi trafik melalui antrian (1024) menggunakan algoritma *hashing*. Merupakan arus lalu lintas yang dapat diidentifikasi secara unik dengan 4 pilihan (*src-adress*, *dst-adress*, *src-port*, dan *dst-port*), sehingga parameter ini digunakan oleh algoritma *hashing* SFQ untuk mengkarifikasi paket ke salah satu dari 1.024 kemungkinan *sub-stream*.



Gambar 2.11 SFQ Operation

Algoritma *Round-Robin* akan mulai mendistribusikan *bandwidth* yang tersedia untuk semua *sub-stream*, pada setiap putaran memberikan *traffic byte sfq-allot*. Kesemua *queue* SFQ berisi 128 *packet* dan terdapat 1024 *sub-stream*.

RED (*Random Early Drop*) merupakan mekanisme antrian (*queue*) yang bertugas untuk menghindari kemacetan jaringan dengan mengontrol ukuran antrian rata-rata. Rata-rata ukuran antrian dibandingkan dengan dua *threshold*: *threshold* minimum (min_{th}) dan *threshold* maximum (max_{th}). Jika rata-rata antrian (avg_q) lebih kecil daripada *threshold* minimum, maka tidak ada paket yang di *drop*. Ketika rata-rata antrian lebih besar daripada *threshold* maksimum, semua paket yang masuk akan di *drop*. Namun apabila rata-rata antrian berada diantara *threshold* minimum dan maksimum, paket akan di *drop* secara acak dengan rumus $P_d = P_{max}(avg_q - min_{th}) / (max_{th} - min_{th})$. Apabila rata-rata antrian berkembang, kemungkinan untuk *drop* paket yang masuk akan semakin besar pula. P_{max} - rasio, yang bisa menyesuaikan paket dengan mengabaikan probabilitas *abruptness*, yang paling sederhana kasus P_{max} bisa sama dengan satu. Diagram pada gambar dibawah ini menunjukkan probabilitas *packet-dropping* menggunakan algoritma RED.



Gambar 2.12 RED Operation

2.2.11.4.2 Shaper

Shaper merupakan bentuk atau teknik mengontrol kecepatan aliran data atau dapat juga sebagai *scheduling job*. *Shaper* yang sering digunakan antara lain:

1. PCQ – *Per Connection Queue - Per Connection Queuing* (PCQ) mirip dengan SFQ, tetapi memiliki fitur tambahan, yakni:

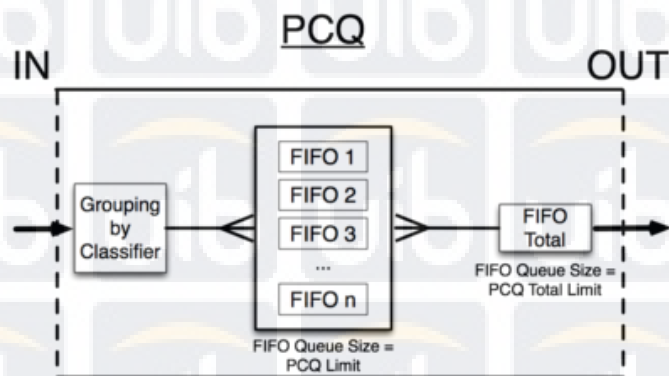
- a. Memungkinkan untuk memilih pengidentifikasi aliran (*flow*) (dari *dst-address | dst-port | src-address | src-port*). Sebagai contoh jika mengklasifikasikan arus dengan *src-address* pada *interface* lokal (antarmuka dengan klien), masing-masing *sub-stream* PCQ akan meng-*upload* satu klien tertentu.
- b. Hal ini dimungkinkan untuk menetapkan batasan kecepatan untuk *sub-stream* dengan opsi *pcq-rate*. Jika *pcq-rate* = 0, maka *sub-stream* akan membagi lalu lintas yang tersedia sama rata.

Adanya PCQ ditujukan untuk mengoptimalkan sistem QoS dalam skala yang besar, dimana semua antrian (*queue*) adalah sama pada semua *sub-stream*. Contohnya sebuah *sub-stream* dapat men-*download* atau *upload* untuk salah satu *client* (IP) atau koneksi ke server. Algoritma PCQ sangat mudah, ia menggunakan salah satu

classifier untuk menentukan satu *sub-stream* dengan yang lainnya, kemudian mengaplikasikannya ke masing-masing besaran individual antrian FIFO dan batasan pada masing-masing *sub-stream*, kemudian mengelompokkan semua *sub-stream* dan mengaktifkan Global FIFO *queue size* dan batasan tersebut.

Parameter dari PCQ:

- a. *Pcq-classifier* (*dst-address* | *dst-port* | *src-address* | *src-port*; *default*: “”) - merupakan pilihan dari *sub-stream identifiers*.
- b. *Pcq-rate* (*number*) - merupakan jumlah maksimal *data rate* yang ada pada masing-masing *sub-stream*.
- c. *Pcq-limit* (*number*) – merupakan jumlah antrian (*queue size*) dari satu *sub-stream* (dalam KiloByte)
- d. *Pcq-total-limit* (*number*) – merupakan jumlah antrian (*queue size*) dari antrian global FIFO. (dalam KiloByte)



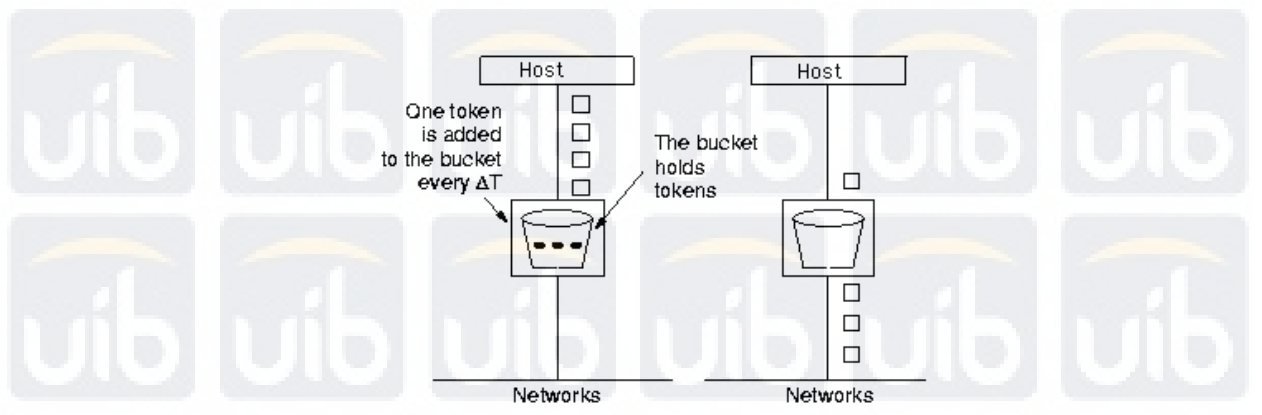
Gambar 2.13 Skema PCQ

2. HTB – *Hierarchical Token Bucket* adalah metode antrian (*queuing*) yang berguna untuk mengendalikan berbagai macam jenis *traffic*. Diperlukan

untuk mengikuti 3 langkah dasar untuk membuat HTB:

- a. Cocokkan dan tandai *traffic*. Klasifikasikan *traffic* untuk penggunaan kedepan. Cocokkan antara satu atau parameter yang cocok untuk memilih paket dari kelas yang spesifik.
- b. Buat aturan (*rule*) untuk menandai *traffic*. Letakkan kelas *traffic* menjadi antrian (*queue*) yang spesifik agar dapat menjalankan *action* untuk masing-masing kelasnya.
- c. Berikan aturan (*policy*) pada setiap *interface*. Tambahkan *policy* pada semua *interface* (*global-in*, *global-out*, atau *global-total*), pada *interface* tertentu ataupun pada *parent queue* tertentu.

HTB memberikan akses untuk membuat struktur *hierarchical queue* dan menentukan relasi antar *queue*, seperti “*parent-child*” ataupun “*child-child*”. Begitu antrian memiliki setidaknya satu *child*, ia menjadi sebuah *inner queue* (antrian dalam), dan antrian yang tidak memiliki *child*, disebut dengan *leaf queue*. *Leaf queue* mengkonsumsi trafik, *inner queue* bertanggung jawab untuk distribusi *traffic*. Semua *leaf queue* diperlakukan secara sama. Dalam mikrotik, penting untuk menentukan opsi *parent* untuk menetapkan apakah *queue* tertentu menjadi *child queue* bagi *queue* lainnya.



Gambar 2.14 Skema Hierarchical Token Bucket