

BAB II

LANDASAN TEORI

2.1 TCP/IP

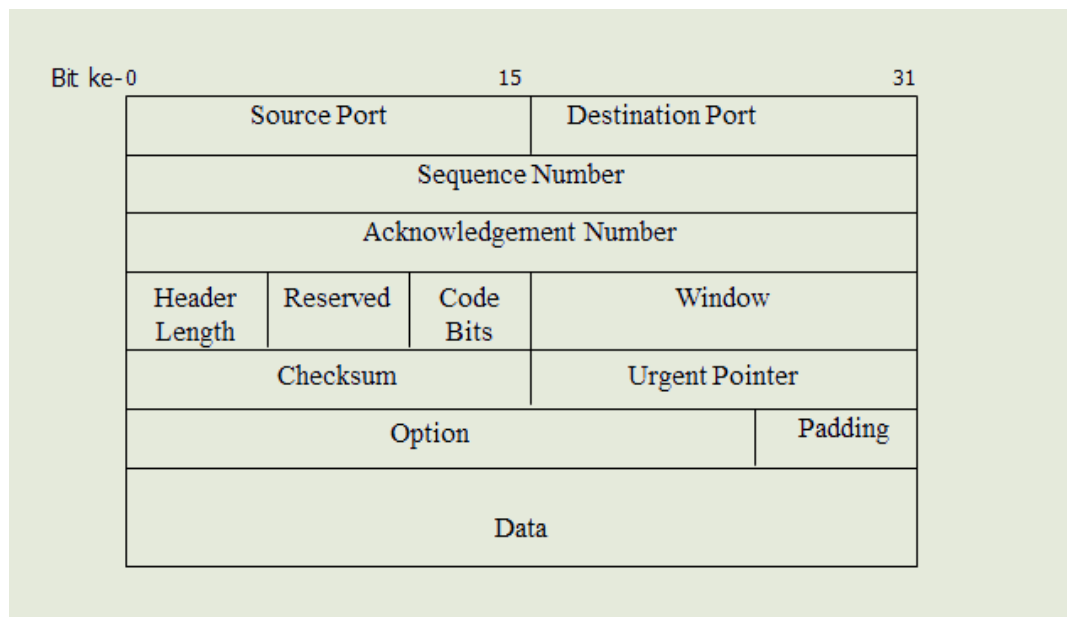
TCP/IP merupakan Komponen penting didalam jaringan. Beberapa bagian *TCP/IP* yang penting dan mencakup di seluruh jaringan yang digunakan sekarang ini adalah :

a. Protokol TCP/IP

Protokol (Harry:2013) merupakan sekumpulan aturan yang mengatur dua atau lebih mesin dalam suatu jaringan dalam melakukan interaksi pertukaran format data. Protokol memiliki suatu fungsi yang spesifik satu sama lain pada sebuah hubungan telekomunikasi. *TCP/IP* merupakan sekumpulan protokol yang dikembangkan untuk mengijinkan komputer-komputer agar dapat saling membagi sumber daya yang dimiliki masing-masing melalui media jaringan. Protokol-protokol *TCP/IP* dikembangkan sebagai bagian dari riset yang dikembangkan oleh *Defense Advanced Research Projects Agency (DARPA)*. Pertama kalinya *TCP/IP* dikembangkan untuk komunikasi antar jaringan yang terdapat pada *DARPA*. Selanjutnya, *TCP/IP* dimasukkan pada distribusi *software UNIX*. Sekarang *TCP/IP* telah digunakan sebagai standar komunikasi internetwork dan telah menjadi protokol *transport* bagi internet, sehingga memungkinkan jutaan komputer berkomunikasi secara global.

Protokol *Transmission Control Protocol*(TCP) adalah protokol yang berada pada model *OSI* dan menyediakan keandalan pengiriman paket secara *stream* dari layanan ke aplikasi dengan menerapkan beberapa mekanisme pengakuan(*acknowledgement*) dan retrans-misi paket pada kasus spesifik(Agus:2012).

Pengiriman aliran ini, kita dapat mengirim data melalui *TCP/IP* dengan format sendiri atau struktur data yang bebas. *TCP/IP* juga merupakan kelompok protokol berorientasi koneksi. Protokol *TCP* juga menyediakan operasi dupleks penuh di mana pengirim dan penerima dapat mengirim dan menerima data pada waktu yang bersamaan. Selain itu, *TCP* juga mempunyai fitur *multiplexing* yang memungkinkan beberapa komunikasi yang terjadi menjadi satu koneksi atau di-*multipleks*-kan.



Gambar 2.1 Format *Protocol TCP*

Keterangan gambar *segment* TCP adalah sebagai berikut :

1. *Source Port*
Angka yang menunjukkan *port* yang memanggil
2. *Destination Port*
Angka yang menunjukkan *port* yang dipanggil
3. *Sequence Number*
Angka yang digunakan untuk memastikan urutan yang benar dari data yang datang
4. *Acknowledgement Number*
Nomor pengakuan(*Acknowledgement*) atau oktet lanjutan
5. *Header length/Data Offset*
Jumlah data pada *header TCP*
6. *Code Bits/Control Bits*
Kontrol (misalnya setup dan terminasi dari suatu sesi)
7. *Window*
Jumlah data(oktet) yang diterima
8. *Checksum*
Checksum yang telah dikalkulasi dari *field header* dan data
9. *Urgent Pointer*
Mengindikasikan akhir dari data yang penting apabila URG dari *code bits/control bits* bernilai 1
10. *Option dan Padding*
Maximum TCP segment size atau data opsi tambahan untuk *header TCP*

11. Data

Data dari protokol lapis atas yang dilewatkan ke protokol *TCP*.

b. Arsitektur *TCP/IP*

Seperti telah disebutkan sebelumnya, *TCP/IP* berisi kumpulan dari protokol-protokol yang melakukan fungsinya masing-masing secara spesifik. Protokol-protokol ini dikumpulkan berdasarkan fungsinya dalam lapisan-lapisan tertentu. *TCP/IP* memiliki 4 lapisan yang antara satu dengan lainnya memiliki protokol dengan fungsi yang saling melengkapi satu sama lain.

c. Datagram IP

Sebuah datagram IP berisi header IP dan data, dan dikelilingi oleh *Media Access Control (MAC) header* dan *MAC trailer*. Satu pesan dapat dikirim dalam urutan datagram-datagram yang disusun kembali menjadi pesan asli pada sisi penerima.

d. *TCP* dan *UDP*

TCP dan *UDP* menggunakan nomor *port* (atau soket) untuk melewati informasi ke lapis yang lebih atas. Nomor *port* digunakan untuk membedakan aplikasi yang berbeda yang melewati jaringan pada saat yang bersamaan. Pengembang software aplikasi telah sepakat untuk menggunakan nomor-nomor *port* yang didefinisikan dalam RFC 1700. Sebagai contoh, semua percakapan yang digunakan untuk aplikasi FTP menggunakan nomor *port* standar 21.

2.2 Perangkat Jaringan

a. *Bridge*

Bridge merupakan suatu alat yang menghubungkan satu jaringan dengan jaringan yang lain yang menggunakan protokol yang sama. *Bridge* menggunakan alamat sumber untuk mempelajari mesin yang mana yang terhubung ke *segment* tertentu, dan *Bridge* menggabungkan informasi yang dipelajari dengan alamat tujuan untuk menghilangkan *forwarding* jika tidak perlu. *Bridge* menghubungkan segmen-segmen LAN di *Data Link layer* pada model OSI. Beberapa *Bridge* mempelajari alamat *Link* setiap devais yang terhubung dengannya pada tingkat *Data Link* dan dapat mengatur alur frame berdasarkan alamat tersebut. Semua LAN yang terhubung dengan *Bridge* dianggap sebagai satu subnetwork dan alamat *Data Link* setiap devais harus unik. LAN yang terhubung dengan menggunakan *Bridge* umum disebut sebagai *Extended LAN*.

b. *Router*

Router merupakan suatu alat ataupun software dalam suatu komputer yang menghubungkan dua buah jaringan atau lebih yang memiliki alamat jaringan yang berbeda. *Router* menentukan akan diarahkan ke titik jaringan yang mana paket yang ditujukan ke suatu alamat tujuan. *Router* biasanya berfungsi sebagai *gateway*, yaitu jalan keluar utama dari suatu jaringan untuk menuju jaringan di luarnya.

c. *Hub*

Hub merupakan suatu alat yang menghubungkan segmen-segmen dalam sebuah LAN. Bila sebuah paket masuk pada suatu *port*, paket tersebut akan dikopikan ke *port-port* yang lain sehingga semua segmen pada LAN dapat melihat paket tersebut.

d. *Switch*

Merupakan kependekan dari *port switching hub*, yaitu hub khusus yang memforward paket ke *port-port* yang bersesuaian berdasarkan alamat paket.

e. *Wireless LAN (Wi-Fi)*

Merupakan teknologi yang ditujukan untuk menggantikan kabel UTP yang digunakan untuk menghubungkan komputer – komputer dalam sebuah tempat.(Sto:2013)

2.3 *Access-Point*

Access-Point merupakan perangkat keras yang memungkinkan perangkat *wireless* lain(seperti laptop dan ponsel) untuk terhubung ke jaringan kabel melalui wi-fi, bluetooth atau perangkat keras lainnya. *Access-Point* dihubungkan ke *router* melalui jaringan kabel dan dapat digunakan untuk saling mengirim data antar perangkat *wireless* dan perangkat kabel pada jaringan.

Fungsi *Access-Point* yaitu :

1. Sebagai pengatur lalu-lintas data, sehingga memungkinkan banyak *client* dapat saling terhubung melalui jaringan.

2. Sebagai *Hub/Switch* yang bertindak untuk menghubungkan jaringan lokal dengan jaringan *wireless/nirkabel*.
3. *Access-Point* dapat memancarkan atau mengirim koneksi data / internet melalui gelombang radio, ukuran kekuatan sinyal juga mempengaruhi area yang akan di jangkau, semakin besar kekuatan sinyal, semakin luas jangkauannya dalam satuan dBm atau mW.

Access-Point yang digunakan pada skripsi ini menggunakan *TP-LINK ADSL* yang di keluarkan oleh telkom speedy melalui jaringan telepon milik telkom. *ADSL(Asymmetric Digital Subscriber Line)* memanfaatkan kabel telepon yang sudah ada sehingga secara otomatis infrastruktur yang tersedia menjadi sangat luas. Ketika pemasangan telkom speedy, sudah disediakan sebuah *WAP(Wireless Access Point)*. *Interface* yang disediakan oleh *WAP TP-Link* ini dibagian belakangnya terdapat sebuah *port RJ-11* yang digunakan sebagai tempat kabel telepon dari telkom. *WAP* ini menyediakan 1 *port RJ 45* yang digunakan untuk menghubungkan *WAP* dengan *Router Mikrotik(S'to:2014)*.

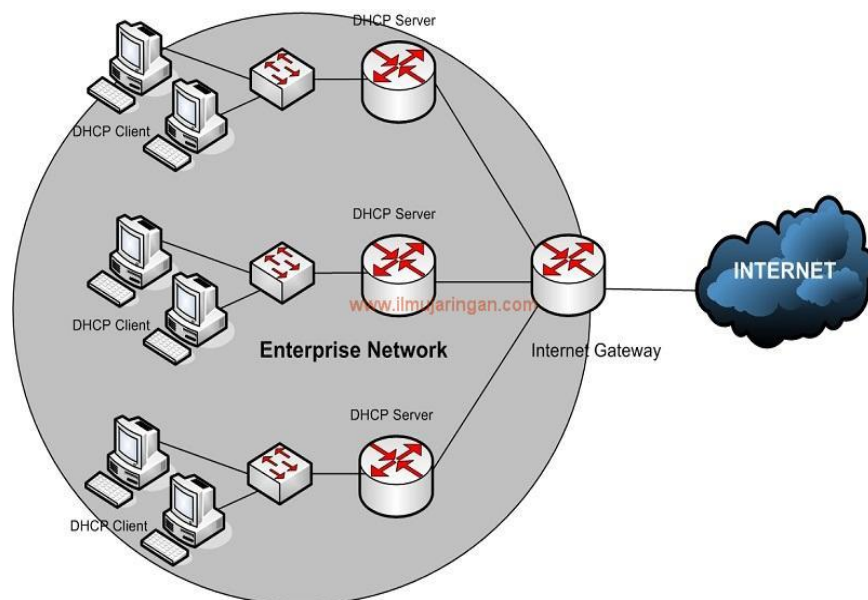
2.4 Pengenalan Router Mikrotik

Router (Rendra:2013) adalah perangkat jaringan yang digunakan untuk menghubungkan beberapa jaringan(network). Dalam jaringan yang lebih kompleks, *Router* digunakan untuk memilihkan jalan bagi paket data untuk mencapai komputer tujuan.

Router Mikrotik merupakan perangkat yang paling cerdas, karena *router* bisa memahami benar bagaimana mengirimkan data dari satu jaringan ke jaringan lain. Mikrotik yang memiliki fitur lengkap ini, mudah di konfigurasi, mencakup jaringan yang cukup luas, lebih menghemat listrik dibandingkan dengan *router* PC, dan harganya yang murah.

Fungsi *Router* Mikrotik yaitu :

1. Sebagai *Internet Gateway* bagi LAN.



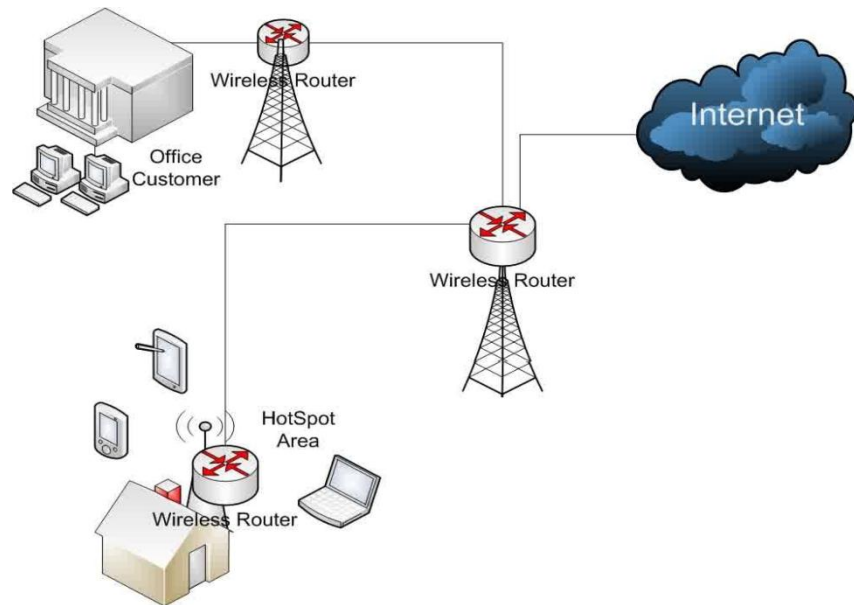
Gambar 2.2 *Router* Sebagai *Internet Gateway* LAN

Inilah peran yang paling banyak dimainkan oleh *Router* Mikrotik. Pada gambar 2.2, *router* ditempatkan di jaringan lokal(LAN) dan jaringan Internet. Jaringan lokal itu terdiri dari beberapa komputer yang ingin mendapatkan akses Internet. *Router* Mikrotik berperan sebagai penghubungnya, sekaligus mengatur lalu lintas Internet, baik yang masuk maupun yang keluar dari jaringan lokal tersebut.

Router Mikrotik akan mampu menyaring apa saja yang boleh diakses di Internet, dan apa saja yang tidak boleh. *Router* Mikrotik akan mampu memblokir *website* tertentu jika diinginkan, bahkan menurut pengaturan waktu. *Router* Mikrotik juga akan mampu membagi *bandwidth* Internet kepada komputer *user*, sehingga tidak ada *user* yang merasa lambat mengakses Internet karena adanya monopoli *user* lain yang sedang mendownload file-file besar.

2. Sebagai *Access-Point* (Indoor/Outdoor)

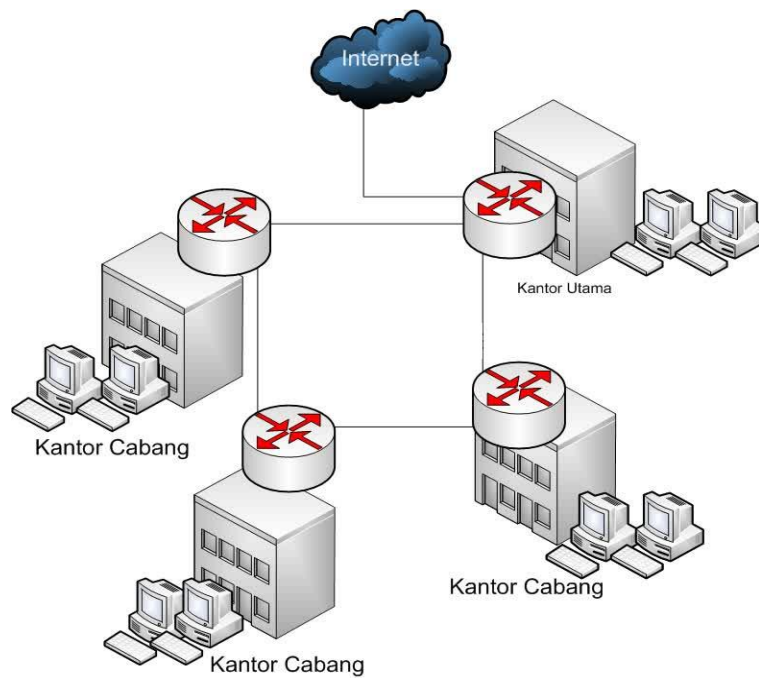
Beberapa *Router* Mikrotik dilengkapi dengan *Interface wireless*, sehingga dapat berfungsi sebagai *Access Point* pada jaringan *wireless (Wi-Fi)*. *Router* Mikrotik dapat digunakan untuk HotSpot berskala kecil, seperti *HotSpot* Cafe sekolah atau Kantor. *Router* Mikrotik juga dapat digunakan untuk jaringan yang lebih besar seperti RT/RW Net atau bahkan jaringan sekelas *Wireless ISP(WISP)*. *Router* Mikrotik dipasarkan dengan berbagai macam tipe dan dukungan terhadap penggunaan Antena untuk memperluas jangkauan jaringan.



Gambar 2.3 Router sebagai Access-Point

3. Sebagai *intermediary device*(Routing)

Fungsi berikutnya yaitu menghubungkan beberapa *network* dan berusaha untuk menentukan jalur terbaik untuk menuju komputer tujuan.



Gambar 2.4 Routing antar kantor cabang

2.5 *QoS (Quality of service)*

Quality of service yang ditulis oleh Morgan(Adrian:2008), didefinisikan bahwa *QoS* adalah suatu pengukuran tentang seberapa baik jaringan dan merupakan suatu usaha untuk mendefinisikan karakteristik dan sifat dari suatu *servis*. *QoS* digunakan untuk mengukur sekumpulan atribut performansi yang telah dispesifikasikan dan biasanya diasosiasikan dengan suatu *servis*. *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 cukup besar dalam jaringan berbasis IP dan internet secara keseluruhan.

Komponen *QoS* adalah :

1. *Delay*, didefinisikan sebagai total waktu tunda suatu paket yang diakibatkan oleh proses transmisi dari satu titik ke titik lain yang menjadi tujuannya.
2. *Jitter*, didefinisikan sebagai variasi dari *delay* atau variasi waktu kedatangan paket. Banyak hal yang dapat menyebabkan *jitter*, diantaranya adalah peningkatan trafik secara tiba-tiba sehingga menyebabkan penyempitan *bandwidth* dan menimbulkan antrian. Selain itu, perbedaan kecepatan terima dan kirim paket dari setiap node juga dapat menyebabkan *jitter*.
3. *Packet loss*, didefinisikan sebagai banyaknya paket yang hilang pada suatu jaringan paket yang disebabkan oleh tabrakan (*collision*),

penuhnya kapasitas jaringan, dan penurunan paket yang disebabkan oleh habisnya TTL (*Time To Live*) paket.

4. *Bandwidth*, merupakan *rate* transfer data maksimal yang dapat diteruskan antara dua titik.
5. *Throughput*, *Throughput*, adalah jumlah total kedatangan paket IP sukses yang diamati di tempat pengukuran pada destination selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut (sama dengan, jumlah pengiriman paket IP sukses per service-second). Berikut adalah perhitungan rumus dalam mencari nilai *throughput*:

$$\text{Throughput} = \frac{\text{Paket diterima}}{\text{Interval waktu antara paket pertama dan terakhir}}$$

2.6 Teknik *Queueing*

Router Mikrotik (Rendra:2013) memiliki fitur *Queue* yang dapat melakukan pengaturan(manajemen) alokasi *bandwidth* bagi setiap komputer *user*. Dengan menerapkan manajemen *bandwidth*, maka perbaikan terhadap kualitas layanan di jaringan (*Quality of service*) telah terpenuhi. *Quality of service* atau *QoS* akan memberikan jaminan alokasi *bandwidth* minimum pada setiap komputer di dalam jaringan.

Queues digunakan untuk membatasi dan memprioritaskan *traffic*:

1. Batas kecepatan data untuk alamat IP tertentu, *subnet*, protokol, *port*, dan parameter lainnya
2. Batas *peer-to-peer traffic*

3. memprioritaskan beberapa paket yang mengalir
4. mengkonfigurasi jalan *traffic-bandwidth* untuk browsing yang lebih cepat
5. menerapkan batasan yang berbeda berdasarkan waktu
6. *sharing* yang tersedia di lalu lintas jaringan pada pengguna yang sama, atau tergantung pada beban *channel*

Dalam *RouterOS*, struktur hirarki dapat dilampirkan pada 4 tempat yang berbeda:

- 1 *global-in* : merupakan semua *Interface* yang masuk secara umum (masuknya antrian). *Queues* yang melekat pada *global-in* berlaku untuk lalu lintas yang diterima oleh *router* sebelum paket *filtering*.
- 2 *global-out* : mewakili semua output *Interface* pada umumnya (egress *queue*).
- 3 *Global-total* : mewakili semua *input* dan *output* antarmuka bersamaan (dengan kata lain agregasi *global-in* dan *global-out*). Digunakan dalam kasus ketika *user* yang dilimit(dibatasi *bandwidth*-nya), menggunakan *upload* dan *download*.
- 4 *Interface name* : merupakan salah satu antarmuka keluar tertentu. Hanya lalu lintas yang ditunjuk untuk pergi keluar melalui antarmuka ini akan melewati antrian HTB ini.

Dalam menjalankan *Queue*, *router* Mikrotik memiliki dua cara, yaitu :

1. ***Simple queue***, cara ini merupakan cara termudah untuk melakukan pengaturan *bandwidth*, di terapkan pada jaringan skala kecil sampai menengah untuk mengatur pemakaian *bandwidth upload* dan

download pada setiap *user*, seperti *single-client* batasan *upload* / *download*, pembatasan *traffic* P2P(*peer to peer*), dll.

Cara paling mudah untuk membatasi data *rate* untuk alamat IP dan/atau subnet tertentu, adalah dengan menggunakan *Simple queue*. Teknik ini, dapat digunakan untuk membangun *QoS*(*Quality of service*) yang baik.

Kegunaan teknik antrian *Simple queue* ini yaitu :

- a. *Peer-to-peer traffic queuing* (*Peer-to-Peer* lalu – lintas *queue*),
- b. Menerapkan aturan antrian pada interval waktu yang ditentukan,
- c. Memiliki prioritas IP,
- d. Menggunakan *packet mark* dari *ip mangle* di menu *firewall*,
- e. Penjadwalan lalu – lintas (*traffic*) 2 arah dengan 1 rule yang memiliki 2 perintah *upload* dan *download*.

Sebuah konfigurasi pada *simple queue* dapat membuat 0 sampai 3 antrian terpisah satu antrian di *global-in*, satu antrian di *global-out* dan satu antrian di *global-total*. Jika semua sifat antrian memiliki nilai *default* (tidak ada batas yang ditetapkan, jenis antrian *default*), dan antrian tidak memiliki anak *queue*, maka tidak bisa dibuat. Dengan cara ini, untuk membuat antrian *global-total* dapat dihindari jika hanya *upload/download* dengan batasan tertentu yang digunakan.

Simple queue memiliki urutan perintah yang ketat. Setiap paket harus melalui setiap antrian sampai akan memenuhi

persyaratan. (Dalam kasus 1000 antrian, paket antrian terakhir harus berjalan melalui 999 antrian sebelum mencapai tujuannya(*mikrotik.com*)).

2. *Queue tree*, cara ini relatif lebih rumit, namun dapat melakukan pembatasan *bandwidth* berdasarkan grup dengan cara hirarki, yaitu melaksanakan perintah antrian yang maju. Penggunaan *Queue tree* membutuhkan paket yang mengalir menggunakan fitur *Mangle* pada *Firewall* untuk memanggil ke *Queue tree*.

Queue tree hanya membuat satu arah antrian pada *HTB*(*Hierarchical Token Bucket*). Ini satu-satunya cara bagaimana menambahkan antrian pada *Interface* yang terpisah. Cara ini digunakan untuk memudahkan konfigurasi pada *mangle*. *Download* dan *upload* tidak perlu secara terpisah, namun hanya *upload* yang mendapatkan *Public Interface* dan *download* akan menjadi *Private Interface*. *Queue tree* memungkinkan untuk memiliki antrian ganda (contoh: prioritas lalu-lintas pada *global-in* dan *global-out*. Pembatasan per client pada *Interface* yang keluar) jika terdapat konfigurasi pada *simple queue* dan *queue tree* pada metode *HTB* yang berjalan bersamaan. Namun *Simple queue* akan didahulukan sebelum *queue tree* berlangsung. *Queue tree* tidak berjalan bersamaan dengan *simple queue*(*mikrotik.com*).

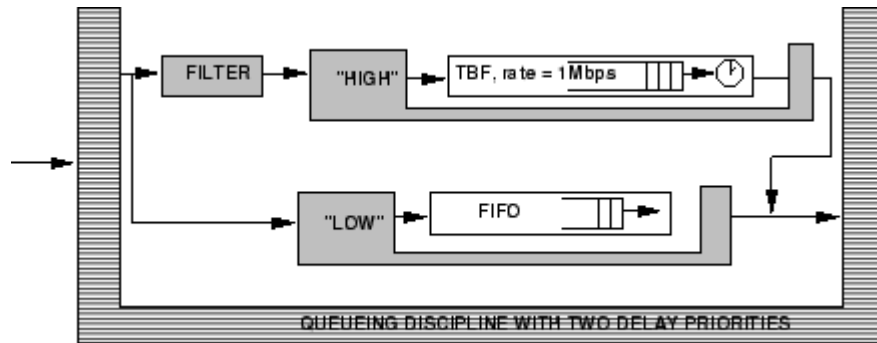
Implementasi antrian di MikroTik *RouterOS* didasarkan pada *Hierarchical Token Bucket (HTB)*. *HTB* memungkinkan untuk membuat struktur hirarkis antrian dan menentukan hubungan antara

antrian. Selain *HTB*, metode *PCQ* (*Per Connection Queue*) juga digunakan untuk membuat struktur dan antrian, namun kedua metode tersebut memiliki fungsi yang sama, namun perbedaannya terletak pada pembagian *bandwidth*-nya.

1. Hierarchical Token Bucket (HTB)

Setiap perangkat jaringan dapat diasosiasikan dengan suatu disiplin antrian. Dengan antrian, kita dapat menentukan bagaimana data akan dikirimkan. Harus dipahami bahwa kita hanya dapat menshaping data yang akan kita kirimkan. Antrian diidentifikasi dengan menggunakan bantuan sebuah handle yang memiliki format X:Y, di mana X merepresentasikan nomor mayor, dan Y merepresentasikan nomor minor. Handle-handle inilah yang akan mengasosiasikan kelas-kelas ke disiplin antrian masing-masing.

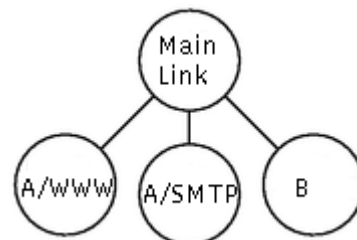
Setiap perangkat jaringan dapat diasosiasikan dengan suatu disiplin antrian. Dengan antrian, kita dapat menentukan bagaimana data akan dikirimkan. Harus dipahami bahwa kita hanya dapat menshaping data yang akan kita kirimkan. Antrian diidentifikasi dengan menggunakan bantuan sebuah handle yang memiliki format X:Y, di mana X merepresentasikan nomor mayor, dan Y merepresentasikan nomor minor. Handle-handle inilah yang akan mengasosiasikan kelas-kelas ke disiplin antrian masing-masing.



Gambar 2.5 Traffic Control pada *Interface Output*

Setiap *switch* dan *router* akan melakukan proses antrian, di mana paket akan disimpan sampai kapasitas tersedia untuk mengirimkannya ke suatu *link*. Waktu yang diperlukan untuk proses antrian akan menimbulkan *delay*, yang akan terakumulasi di setiap peralatan jaringan yang dilaluinya.

HTB merupakan salah satu algoritma yang mengatur disiplin antrian dengan tujuan untuk menerapkan *link sharing* secara presisi dan adil. Dalam konsep *link sharing*, jika suatu kelas meminta kurang dari jumlah service yang telah ditetapkan untuknya, sisa *bandwidth* akan didistribusikan ke kelas-kelas yang lain yang meminta service (Harry:2013).

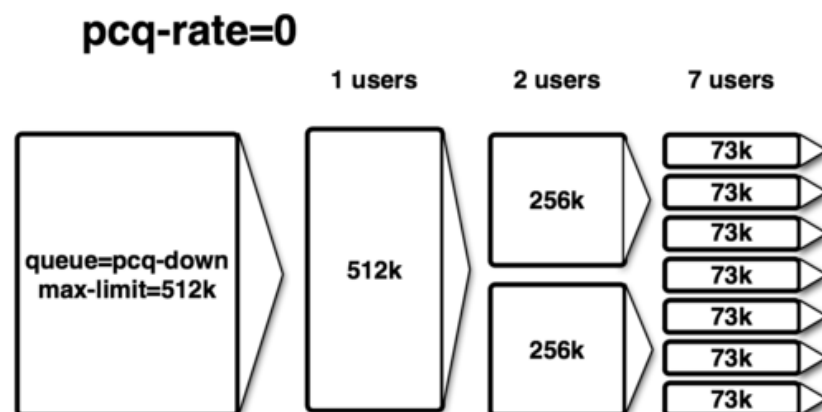


Gambar 2.6 Konsep dasar *link sharing*

2. Per Connection Queue (PCQ)

PCQ pada *queue* type (Rendra:2013) adalah salah satu fitur dari MikroTik untuk membantu memmanage *Traffic rate* dan *Traffic packet* pada banyak *user*, dalam skripsi ini, algoritma ini di gunakan untuk *wireless* connection. PCQ digunakan untuk mengenali arah arus dan digunakan karena dapat membagai *bandwidth* secara adil, merata dan masif. PCQ pada mikrotik digunakan bersamaan dengan fitur *Queue*, dalam *Queue tree*.

PCQ bekerja dengan membuat sub-stream berdasarkan parameter *pcq-classifier* yang dapat berupa IP adres pengirim(*src-Address*) dan IP Address tujuan(*dst-Address*). Manajemen *bandwidth* untuk aktifitas *download*, menggunakan parameter *dst-Address*, karena *Traffic download* datangnya dari Internet dan menuju jaringan lokal, maka yang di jadikan pedoman PCQ adalah IP adres tujuan. Sedangkan untuk *Traffic upload*, parameter *classifier* yang harus digunakan adalah *src-Address*.



Gambar 2.7 Konsep dasar *link sharing*

a. *PCQ Classifier*

PCQ Classifier digunakan untuk membuat *sub-stream*, *PCQ* dapat menggunakan parameter *src-Address*, *dst-Address*, *src-port*, maupun *dst-port*. Parameter yang paling banyak digunakan dan diimplementasikan adalah parameter *src-Address* dan *dst-Address* yang digunakan untuk melakukan manajemen *bandwidth* baik untuk *Traffic upload* maupun *download*.

Manajemen *bandwidth* untuk aktifitas *download* menggunakan parameter *dst-Address* (*IP Address* tujuan), karena *Traffic download* datangnya dari Internet dan menuju jaringan lokal, maka yang di jadikan pedoman bagi *PCQ* adalah *IP Address* tujuan. *IP Address* tujuan ini bukan merupakan *IP Address* di Internet, namun merupakan *IP Address* dari komputer client di jaringan lokal. Sedangkan untuk *Traffic upload*, parameter *classifier* yang harus digunakan adalah *src-addres*.

b. *PCQ Rate*

Parameter *pcq-rate* digunakan untuk membatasi *bandwidth* maksimum yang bisa didapatkan oleh setiap *sub-stream*. Jika parameter yang digunakan adalah *pcq-rate=0* maka *sub-stream* bisa saja mendapatkan *bandwidth* maksimum yang nantinya diberikan oleh simple *queue*

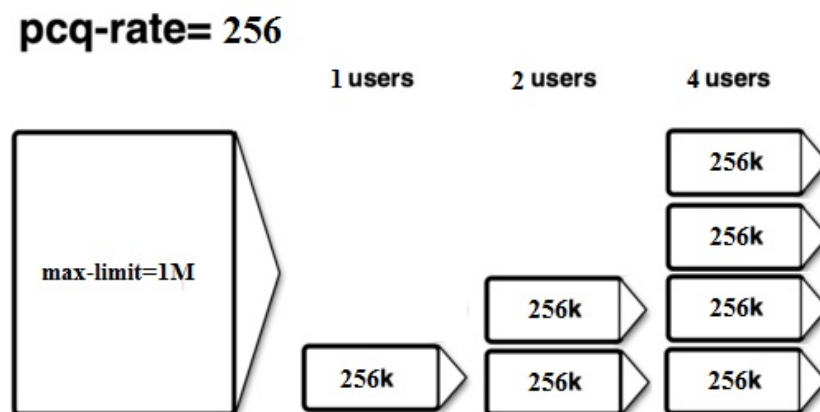
ataupun *queue tree*. Pada gambar 2.7, memperlihatkan pembagian *bandwidth* oleh PCQ, dimana parameter *pcq-rate* yang digunakan adalah 0.

Pada gambar 2.7, sebagai contoh, *simple queue* memberikan alokasi *bandwidth* maksimal (*max-limit*) sebesar 512kbps untuk *Traffic download* pada PCQ yang dibuat. Jika ternyata hanya ada satu komputer client yang menggunakan *Traffic download*, maka komputer tersebut akan mendapatkan *bandwidth* 512kbps, sesuai parameter *max-limit* yang diberikan oleh *Simple queue* atau *Queue tree*.

Jika ternyata *Simple queue/Queue tree* menggunakan parameter *max-limit=2Mbps*, maka komputer tersebut juga akan mendapatkan 2 Mbps. Kesimpulan yang bisa diambil adalah jika parameter *pcq-rate=0* yang digunakan, maka maksimal *bandwidth* yang bisa didapat oleh sebuah *sub-stream* akan sangat relatif dan bergantung dengan parameter *max-limit* yang ada di *Simple queue/Queue tree*.

Batasan maksimal *bandwidth* yang bisa didapat oleh sebuah *sub-stream*, maka dapat menggunakan parameter *pcq-rate*. Misalnya jika menggunakan parameter *pcq-rate=256kbps*, maka walaupun *max-limit* yang diberikan oleh *Queuesimple/Queue tree* adalah 1 Mbps, dan hanya ada satu komputer client yang menggunakan alokasi *bandwidth*, komputer tersebut tetap hanya akan mendapatkan alokasi

bandwidth sebesar 256kbps. Walaupun sebenarnya masih ada sisa *bandwidth* 768kbps, jika ada komputer yang menggunakan *bandwidth*, maka masing-masing akan tetap mendapatkan 256kbps. Ilustrasinya seperti gambar 2.8



Gambar 2.8 PCQ dengan *pcq-rate=256kbps*

c. PCQ Limit dan PCQ Total Limit

Parameter *pcq-limit* dan *pcq-total-limit* dapat digunakan untuk membatasi jumlah *sub-stream*. Kedua parameter ini dinyatakan dalam kiloByte (kB). Parameter *pcq-limit* menyatakan ukuran maksimal dari sebuah *sub-stream*, sedangkan parameter *pcq-total-limit* menyatakan ukuran maksimal dari sebuah *sub-stream*, sedangkan parameter *pcq-total-limit* menyatakan ukuran maksimal dari keseluruhan *sub-stream*. Sehingga jumlah maksimal *sub-stream* yang bisa tercipta adalah hasil pembagian antara *pcq-total-limit* dengan *pcq-limit*.

Secara default *pcq-limit* berukuran 50kB, sedangkan parameter *pcq-total-limit* berukuran 2000kB, sehingga secara default jumlah *sub-stream* yang bisa digunakan adalah $2000/50=40$ *sub-stream*. Perubahan konfigurasi dari kedua parameter ini nantinya harus memperhitungkan ukuran dari RAM yang ada pada *Router* Mikrotik yang digunakan.