

## BANDWIDTH CONTROL INTERNET ADSL DENGAN METODE HIERARCHICAL TOKEN BUCKET (HTB) DAN QUEUE TREE ROUTER OS PADA JARINGAN CAMPUS AREA NETWORK (CAN)

Juniar Sinaga

Program Studi Teknik Komputer

Akademik Manajemen Informatika dan Komputer Bina Sarana Informatika (AMIK BSI)

Jl. Jatiwaringin Raya No. 18 , Jakarta Timur

<http://synagabonar.wordpress.com>

[juniar\\_sinaga@bsi.ac.id](mailto:juniar_sinaga@bsi.ac.id)

### ABSTRAK

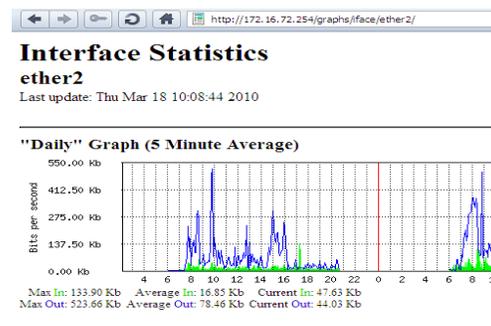
Saat ini pengguna jaringan internet baik melalui transmisi dengan kabel maupun wireless didunia sudah sangat padat dan dengan intensitas yang tinggi. Akibat padatnya akses tersebut dapat mengakibatkan koneksi user antar jaringan menjadi lambat atau over load karena saling berebut bandwidth. Dampak dari penggunaan bandwidth secara bersamaan dan berlebihan bisa juga mengakibatkan jaringan drop hingga membuat modem *down*. Untuk menangani masalah ini diperlukan suatu alat atau device dan metode *controlling* dan *limitation* untuk menghemat penggunaan bandwidth serta mempermudah pengelolaan jaringan internet khususnya yang terhubung dengan Campus Area Network (CAN) maupun Local Area Network (LAN).

Salah satu alat atau device yang sudah terbukti kehandalannya untuk menjalankan metode *controlling* dan *limitation* tersebut adalah Router. Secara umum router didesign dan didistribusikan dalam bentuk *hardware* maupun *software*. Pada penelitian ini router yang digunakan adalah Router OS dengan sistim operasi linux yang disebut dengan Mikrotik Router OS . MikroTik Router OS merupakan sistem operasi yang dirancang khusus untuk network router (Moch.Linto Herlambang dan Azis Catur.L hal 19). Didesain untuk memberikan kemudahan bagi penggunaannya. Administrasinya bisa dilakukan melalui Windows Application (WinBox).. Pada Mikrotik Router OS layanan *bandwith control* ini diimplementasikan dengan metode *Hierarchical Token Bucket* (HTB) dan *Queue Tree* atau antrian. Metode *Queue Tree* atau antrian pada router Mikrotik dapat memberikan jaminan kualitas layanan bandwidth (*Quality Of Service*) yang efisien bagi client atau user yang bisa dikelola oleh pengelola aringan dalam hal ini adalah para administrator jaringan (network administrator) itu sendiri. Kedua metode ini diterapkan pada Jaringan Campus Area Network (CAN) yang terkoneksi dengan internet ADSL (*Asymmetric Digital Subscriber Line*) yang memiliki permasalahan penggunaan bandwidth dimana dalam penelitian ini terbukti dapat diatasi dengan metode *Hierarchical Token Bucket* (HTB) dan *Queue Tree* berbasis PC Router OS.

Kata Kunci: *Bandwidth Control*, *CAN Networks*, *Queue Tree*, HTB, Router OS.

### 12. PENDAHULUAN

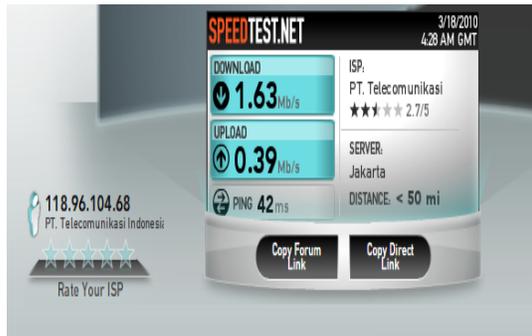
Traffic pengguna akses internet pada saat ini sangat tinggi sekali. Baik untuk mencari informasi , artikel, pengetahuan terbaru atau bahkan hanya untuk *chatting*. Jaringan Campus Area Network dalam penelitian ini menggunakan koneksi *Asymmetric Digital Subscriber Line* atau ADSL upto 2 Mbps dari *Internet Service Provider* (ISP). Akibat tidak stabilnya bandwidth upto 2 Mbps dari provider ini, mengakibatkan internet sering *down*. Traffic penggunaan *bandwidth* di jaringan Campus Area Network (CAN) dapat dilihat pada grafik dibawah ini:



Sumber: <http://172.16.72.aaa/graphs/iface/ether2>

Gambar 1.1. Grafik Penggunaan Pada Jaringan CAN.

Koneksi internet *Asymmetric Digital Subscriber Line* atau ADSL upto 2 Mbps jika diukur dengan tool maka akan diperoleh jumlah bandwidth sesungguhnya atau real adalah mendekati 2 Mbps. Setelah dilakukan testing bandwidth dengan tool internet diperoleh *bandwidth* real sebesar:



Sumber: <http://speedtest.net/>

Gambar 1.2. Pengukuran *Bandwidth* ADSL Dengan Tool Internet.

Pengguna internet pada Campus Area Network ini sudah cukup banyak, baik dari pihak karyawan maupun mahasiswa yang terkoneksi dengan media kabel maupun dengan jaringan hotspot area. Karena banyaknya user yang menggunakan internet pada kampus ini menimbulkan padatnya traffic penggunaan akan jalur internet (*bandwidth*) yang tidak teratur dan berlebihan sehingga berdampak pada lambatnya proses loading data pada saat melakukan browsing baik pada saat proses pengambilan data dari internet (*download*) maupun pada saat proses store data keinternet (*upload*).Terlebih-lebih jika kedua proses ini dilakukan secara *concurrent* bisa mengakibatkan bandwith internet full atau *over load* sehingga koneksi internet ke provider (ISP) bisa menjadi sangat lambat bahkan terputus dan dapat mengakibatkan modem *crash* atau *down*. Untuk menjawab permasalahan ini maka penulis menerapkan suatu metode *bandwidth control* atau *bandwidth limitation* menggunakan router, router ini bisanya dari vendor bisa berbentuk software dan hardware. Pada penelitian ini router yang digunakan adalah Mikrotik Router OS yang berbasis PC. Untuk *bandwidth control* atau *bandwidth limitation* Mikrotik Router OS menggunakan metode *Hierarchical Token Bucket* (HTB) yang diimplementasikan dengan teknik *Queue Tree*.

### 13. LANDASAN TEORI

#### 2.1. Pengertian Router

Router adalah perangkat yang akan melewatkan paket IP dari suatu jaringan ke jaringan yang lain, menggunakan metode addressing dan protocol tertentu untuk melewatkan paket data tersebut. Router memiliki kemampuan melewatkan paket IP dari satu jaringan ke jaringan lain yang mungkin memiliki banyak jalur diantara keduanya. Router-router yang saling terhubung dalam jaringan internet turut serta dalam sebuah algoritma routing terdistribusi untuk menentukan jalur terbaik yang dilalui paket IP dari sistem ke sistem lain. Proses routing dilakukan secara *hop by hop*. IP tidak mengetahui jalur keseluruhan menuju tujuan setiap paket. IP routing hanya menyediakan IP address dari router berikutnya yang menurutnya lebih dekat ke host tujuan.Router bekerja pada layer tiga (3) OSI (Open System Interconnection) yang sering digunakan untuk melakukan segmentasi pada jaringan LAN. Fungsi dari Router adalah sebagai berikut:

1. Membaca alamat logika atau ip address *source* dan *destination* untuk menentukan routing dari suatu LAN ke LAN lainnya.
2. Menyimpan routing table untuk menentukan rute terbaik antara LAN ke WAN.
3. Perangkat di layer 3 OSI Layer.
4. Bisa berupa "box" atau software yang menjalankan sebuah daemon routing.

#### 2.2. Defenisi *Bandwidth* Dan *Bandwidth Control Router OS*

Menurut Tanenbaum (2003) *Bandwidth* adalah banyaknya data dalam satuan bits per second yang dapat ditransmisikan lewat sebuah medium jaringan dalam satu satuan waktu.

*Bandwidth control* adalah kumpulan mekanisme yang mengontrol kecepatan akses data ( rate allocation), variabilitas penundaan (delay variability), transfer data secara tepat waktu (timely delivery), dan keandalan transfer data secara real (delivery reliability). *Bandwidth control* pada Mikrotik Router OS sering disebut sebagai *bandwidth limitation* bertujuan untuk membatasi *downlink* dan *uplink* pada saat browsing. Sebagai *bandwidth control* Mikrotik Router OS mendukung penggunaan teknik antrian seperti:

1. PFIFO ( Packet First-In First-Out).
2. BFIFO ( Bytes First-In First-Out).
3. SFQ ( Stochastic Fairness Queueing).
4. RED ( Random Early Detect).
5. PCQ (Per Connection Queue).
6. HTB ( Hierarchical Token Bucket).

### 2.3. Pengertian Sistim Operasi

Sistem operasi adalah sekumpulan rutin perangkat lunak yang berada diantara program aplikasi dan perangkat keras (Bambang Hariyanto,2006 :25). Sistem operasi memiliki tugas yaitu mengelola seluruh sumber daya sistem komputer dan sebagai penyedia layanan. Sistem operasi menyediakan *system call* (berupa fungsi-fungsi atau API (*Application Programming Interface*)). *System Call* ini memberikan abstraksi tingkat tinggi mesin untuk pemrograman. System Call berfungsi menghindari kompleksitas pemrograman dengan memberi sekumpulan instruksi yang lebih mudah dan nyaman, sistem operasi juga sebagai basis untuk program lain dimana program aplikasi dijalankan diatas sistem operasi, program-program itu memanfaatkan sumber daya sistem komputer dengan cara meminta layanan sistem operasi mengendalikan sumber daya untuk aplikasi sehingga penggunaan sumber daya sistem komputer dapat dilakukan secara benar dan efisien.

Sistem operasi yang dikenal antara lain :

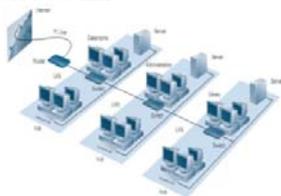
1. Windows (95, 98, ME, 2000, XP, VISTA, SERVER, Windows7)
2. Linux (Red Hat, Slackware, Ubuntu, Fedora, Mikrotik, Debian, OpenSUSE)
3. UNIX
4. FreeBSD (Berkeley Software Distribution)
5. SUN (SOLARIS)
6. DOS (MS-DOS)
7. Machintosh (MAC OS, MAC OSX).

### 2.4. Jaringan Komputer

Jaringan komputer merupakan sekelompok komputer otonom yang saling dihubungkan satu sama lainnya, menggunakan suatu media dan protocol komunikasi tertentu, sehingga dapat saling berbagi data dan informasi (Deris Setiawan,2003:1).Jaringan komputer memungkinkan terjadinya komunikasi yang lebih efisien antar pemakai (*mail dan teleconference*).

Klasifikasi jaringan komputer terdiri dari:

1. LAN (Local Area Network) yaitu Jaringan komputer yang saling terhubung kesuatu komputer server dengan menggunakan topologi tertentu, biasanya digunakan dalam kawasan satu gedung atau kawasan yang jaraknya tidak lebih dari satu km.



Sumber : www.cisco.com

Gambar 2.2. Jaringan LAN atau CAN

2. MAN (Metropolitan Area Network) : Jaringan komputer yang saling terkoneksi dalam satu kawasan kota yang jaraknya bisa lebih dari satu kilo meter. Pilihan untuk membangun jaringan komputer antar kantor dalam suatu kota, kampus dalam satu kota.



Sumber : www.cisco.com

Gambar 2.3. Jaringan MAN

3. WAN (Wide Area Network) : Jaringan komputer yang menghubungkan banyak LAN ke dalam suatu jaringan terpadu, antara satu jaringan dengan jaringan lain dapat berjarak ribuan kilometer atau terpisahkan letak geografi dengan menggunakan metode komunikasi tertentu.



Sumber : www.cisco.com

Gambar 2.4. Jaringan WAN

Topologi jaringan adalah, hal yang menjelaskan hubungan geometris antara unsur-unsur dasar penyusun jaringan, yaitu node, link, dan station.

Secara umum topologi dapat dikategorikan menjadi tiga yaitu:

1. *Signal Topology*

Adalah tata letak pada koneksi actual antar nodes pada sebuah jaringan, yang melalui jalur yang diambil sinyal ketika beinteraksi dengan *nodes* yang lain.

2. *Logical Topology*

*Logical Topology* secara definisi memiliki kemiripan dengan *Signal Topologies*. Definisi dari *Logical Topology* adalah jalur yang diambil oleh data, antar nodes dalam jaringan. Sedangkan pada *signal topologies* didasarkan pada sinyal.

3. *Physical Topology*

Yaitu tata letak nodes dari sebuah jaringan dan hubungan yang terjadi secara actual (fisik) di antaranya. Misalkan layout

dari kabel ,lokasi dari nodes, dan interkoneksi antara nodes dengan kabel pada jaringan.

*Physical Topology* diklasifikasikan menjadi beberapa bentuk topologi, yaitu :

- a. *Point to Point* (Titik ke titik)
- b. *Star Network* (Jaringan Bintang)
- c. *Ring Network* (Jaringan Cincin)
- d. *Mesh Network* (
- e. *Tree Network* (Jaringan Pohon)

### 3. METODE PENELITIAN

#### 3.1. Metode Penelitian Teknik Informatika

Seperti ilmu-ilmu yang lain Teknik Informatika akan berdiri kokoh dan berkembang bila memiliki metode kajiannya sendiri guna menemukan kebenaran. Kebenaran merupakan realitas yang perlu digali dan ditemukan, meskipun sangat tidak mungkin untuk menemukan semua kebenaran yang ada. Kebenaran dalam Teknik Informatika meliputi kebenaran komputasi dan informasi serta penerapannya di dunia nyata. Kebenaran yang dapat dipertanggungjawabkan adalah kebenaran yang diperoleh melalui pengkajian ilmiah (sekali pun terdapat banyak aliran seperti positivistik, rasionalistik, fenomenologik, dan sebagainya dalam merumuskan suatu kebenaran). Pengkajian ilmiah (penelitian) menurut aliran positivistik atau postpositivistik yang banyak dianut peneliti Teknik Informatika merupakan upaya sistematis, investigatif, objektif, logis, hati-hati, dan terencana yang selalu berusaha mencari kebenaran. Kualitas suatu penelitian akan tampak melalui kontribusinya dalam pemecahan masalah yang dihadapi masyarakat dan bagi pengembangan iptek. Oleh sebab itu penentuan rumusan masalah perlu didasari aspek manfaat dari penelitian termasuk dari sudut pandang ekonomi dan kebijakan. Berkaitan dengan strategi pemecahan atau pengatasan masalah William J. Gephart (1972:65), seperti yang dikutip Miarso (2004:36), menyarankan suatu taksonomi strategi empirik dalam mencari kebenaran melalui pengkajian ilmiah. Usaha pencarian kebenaran ini terdiri dari empat lapis, yakni filsafat ilmu, metode umum pemecahan atau pengatasan masalah, strategi operasional, dan sekuen prosedural yang digambarkan sebagai pohon pengkajian ilmiah dengan tiga cabang yang terdiri dari pengembangan, penelitian, dan penilaian. Pengkajian ilmiah dalam bentuk Penelitian Pengembangan, atau lebih tepat disebut Penelitian Rekayasa dapat berupa:

- a. Rencana (Plan).
- b. Rancangan (Design).
- c. Bangunan atau konstruksi (Construct).
- d. Terapan (Applied).

e. Hasil pengembangan (Development) dari suatu model, sistem, atau produk.

. Penelitian rekayasa (termasuk penelitian perangkat lunak) adalah penelitian yang menerapkan ilmu pengetahuan menjadi suatu rancangan guna mendapatkan kinerja sesuai dengan persyaratan yang ditentukan. Rancangan tersebut merupakan sintesis unsur-unsur rancangan yang dipadukan dengan metode ilmiah menjadi suatu model yang memenuhi spesifikasi tertentu. Penelitian diarahkan untuk membuktikan bahwa rancangan tersebut memenuhi spesifikasi yang ditentukan. Penelitian berawal dari menentukan spesifikasi rancangan yang memenuhi spesifikasi yang ditentukan, memilih alternatif yang terbaik, dan membuktikan bahwa rancangan yang dipilih dapat memenuhi persyaratan yang ditentukan secara efisiensi, efektif dan dengan biaya yang murah. Penelitian perangkat lunak komputer dapat digolongkan dalam penelitian rekayasa.

Penelitian Rekayasa dapat berupa:

a. *Forward Engineering* adalah rekayasa yang dilakukan dari perencanaan, perancangan, pembangunan, hingga penerapan, atau pada tahapan-tahapan pendek rekayasa, misal dari perancangan ke pembangunan saja. Rekayasa dilakukan mulai dari abstraksi yang lebih tinggi menuju ke setingkat atau beberapa tingkat lebih rendah.

b. *Reverse Engineering* merupakan rekayasa dari produk, sistem, atau prototipe yang sudah ada menjadi blue print, formula, atau model, atau pada tahapan-tahapan pendek rekayasa, misal dari pembangunan ke perancangan saja. Rekayasa dilakukan mulai dari abstraksi yang lebih rendah menuju ke setingkat atau beberapa tingkat lebih tinggi ('Reverse Engineering' Stefanus: 2005).

c. *Re-engineering* merupakan pengubahan dan pengorganisasian kembali komponen-komponen sistem yang dapat dilakukan terhadap hasil desain atau implementasi saja atau pada keseluruhan tahapan atau abstraksi sistem, tanpa menghilangkan keseluruhan komponen lama agar diperoleh metode, formula, model, prototipe, produk, sistem, atau tools dengan tingkat kesempurnaan dan standar yang lebih tinggi.

#### 3.2. Metode Pengumpulan Data

Dalam penelitian ini langkah-langkah pengumpulan data untuk adalah sebagai berikut:

- a. Melakukan studi literatur mengenai jaringan internet ADSL dan Mikrotik Router OS.
- b. Membangun algoritma HTB dan *queue Tree* Mikrotik Router OS untuk implementasi Bandwidth Control internet pada jaringan Local Area Network (LAN) pada kampus

- Bina Sarana Informatika cabang Jatiwaringin.
- Melakukan pengujian terhadap algoritma HTB dan *Queue Tree* menggunakan Mikrotik Router OS.
  - Menganalisa hasil simulasi untuk melihat algoritma HTB dan *queue tree* apakah lebih baik untuk bandwidth control internet.

Tabel 3.2. Jadwal Penelitian

Jenis	Desembe				Januari				Februari				Maret			
	I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV
Penelitian																
Observasi																
Pengumpulan																
Eksperimen																
Penyusunan																

#### 4. HASIL PENELITIAN DAN PEMBAHASAN

##### 4.1. Paket Algoritma Hierarchical Token Bucket (HTB) Pada Mikrotik Router

Atribut kelas yang terdapat dalam algoritma *Hierarchical Token Bucket* (HTB) adalah:

- Limit-at* yaitu parameter dimana suatu client akan mendapatkan bandwidthnya selama *bandwidth* tersedia, apapun kondisinya.
- Max-limit* merupakan batas kecepatan maksimum suatu *bandwidth*.
- Priority* yaitu prioritas antrian yang bernilai satu sampai dengan delapan (1-8).

Prosedur paket algoritma *Hierarchical Token Bucket* (HTB) adalah:

- Paket yang melalui sebuah router melewati semua empat pohon pada HTB
- Paket yang melalui router, hanya lewat *global-in* dan *global-total* HTB.
- Paket yang melalui sebuah router akan melewati *global-out*, *global-total* dan *interface* HTB.

HTB memiliki lebih dari satu interface fisik yaitu:

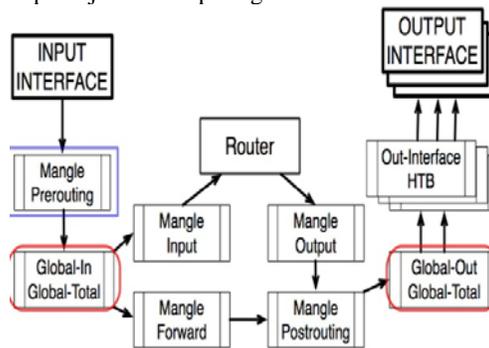
- Global-in* (menggambarkan semua data yang masuk ke dalam interface) dieksekusi setelah *mangle* dan *dst-nat*.
- Global-out* (menggambarkan semua data yang keluar dari interface) queue dikirim sebelum salah satu paket dikirim ke spesifik *interface*.
- Global-Total* (menggambarkan aliran data yang melalui router). Contoh:  $total\ max\ limit = 256000\ kbps$ . Maka besar  $upload + download = 256\ kbps$  (maximum).

Beberapa hal yang perlu diingat supaya HTB dapat diterapkan dengan baik yaitu :

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

##### 4.2. Mangle

Sebelum melakukan pelimitan, harus terlebih dahulu menandai aliran paket menggunakan suatu tanda yang disebut dengan *mangle*. *Mangle* dibuat supaya aliran paket-paket yang akan ditandai dapat dikenali oleh *queue tree* pada Mikrotik Router OS (Moch.Linto Herlambang dan Azis Catur L :101). Selain itu *mangle* ini bertujuan untuk membedakan paket yang *downlink only* atau *uplink only* sehingga limit pada bandwidth dapat bekerja optimal. Dalam diagram *mangle* dengan HTB dan *queue* dapat dijelaskan seperti gambar dibawah ini:



Sumber: <http://nux@ufoakses.co.id>

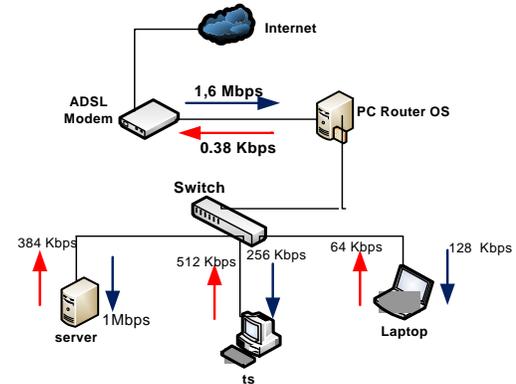
Gambar 4.1. Hubungan *Queue* dan *Mangle* Dengan HTB

##### 4.3. Prinsip Kerja Queue Tree

*Queue tree* hanya satu arah. Harus ada satu antrian untuk *download* dan satu untuk *upload*. Antrian *queue tree* bekerja hanya dengan menandai paket. Penandaan paket ini harus dibuat dalam *firewall mangle*. *Queue tree* memungkinkan untuk membangun hierarki antrian yang kompleks ([www.mikrotik.co.id/artikel\\_lihat.php?id=29](http://www.mikrotik.co.id/artikel_lihat.php?id=29)).

Hubungan *queue tree* dengan *mangle* adalah sebagai berikut:

- Mangle harus dilakukan sebelum *queue*
- Mangle harus dilakukan pada chain prerouting agar dapat digunakan pada *simple queue*
- Jika *queue* dilakukan hanya pada chain tertentu pakai *queue tree*.
- Mangle pada chain prerouting akan berpengaruh pada semua trafik input ataupun *forward*.
- Mangle pada chain postrouting akan berpengaruh pada trafik output dan *forward*.
- Kalau melakukan *mangle* pada chain *forward* maka lakukan *queue* pada *global out* atau *interface queue*.



B. Skema Topologi Setelah Bandwidth Control

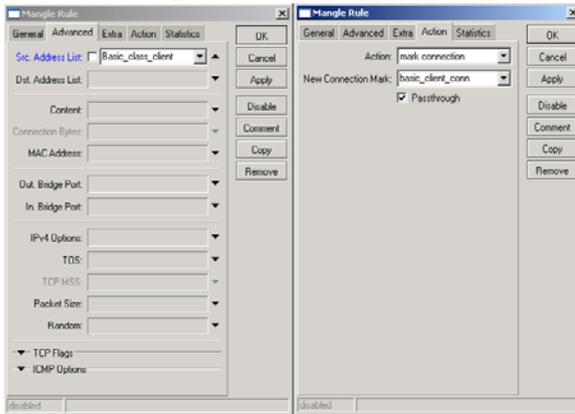
Gambar 4.3. Skema Topologi Network

Pada skema topologi yang terdapat pada gambar B diatas, dapat dilihat jelas pengalokasian *bandwidth* yang terdiri dari *download* dan *upload*. Pada setiap user alokasi *bandwidth* disetting berbeda-beda tergantung sesuai dengan jumlah yang dialokasikan oleh network administrator.

Pada penelitian ini contoh *bandwidth* yang akan dialokasikan adalah pada komputer user *ts* dan *server*. *Bandwidth* yang dimaksud adalah besarnya *download* dan *upload*. Pada skema jaringan dengan *bandwidth control* diatas digambarkan besar *download* dan *upload* seperti pada tabel dibawah ini:

Tabel 4.1. Bandwidth Yang Akan Dialokasikan Untuk User

Nama User	IP Address	Download	Upload
ts	172.16.72.xxx	512 Kbps	256 Kbps
server	172.16.72.xxx	1 Mbps	384 Kbps



Sumber: <http://www.mikrotik.com/manual 3.0.pdf>

Gambar 4.2. Contoh Rule connection-mark

#### 4.4. Skema Topologi Network

Untuk mempermudah analisa terhadap penggunaan *bandwidth* dengan metode *Hierarchical Token Bucket* (HTB) dan *queue tree* pada kampus pada jaringan CAN ini, perlu mengetahui skema topologi network . Skema topologi network yang dimaksud adalah skema topologi sebelum *bandwidth control* diterapkan dan skema topologi setelah *bandwidth control* diterapkan. Skema topologi ini menjelaskan hubungan antara node atau host dengan switch, antara host dengan modem, antara switch dengan modem, dan antara host dengan server pada jaringan Campus Area Network yang terintegrasi. Sebelum metode ini diimplementasikan masing-masing user dan server masih berebutan dalam menggunakan *bandwidth*, dan belum adanya pengalokasian *bandwidth* seperti yang tampak pada gambar A.

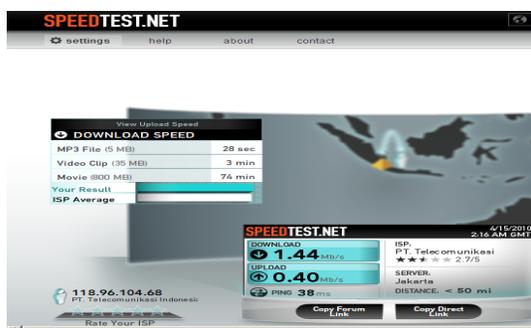
#### 4.5. Testing Bandwidth Sebelum Metode Queue Tree

Testing *bandwidth* dilakukan pada user untuk mengetahui jumlah *bandwidth* pada user sebelum metode diterapkan. Testing menggunakan *tool internet (browser)* dari user yang akan diukur jumlah *bandwidth*nya. Pada simulasi ini pengukuran *bandwidth* dilakukan pada user *ts* dan *server*. Hasilnya diperoleh seperti pada gambar dibawah ini:



Sumber: <http://www.speedtest.net/172.16.72.xxx>

Gambar 4.4. Bandwidth Sebelum Queue Tree Pada User ts



Sumber: <http://www.speedtest.net/172.16.72.xxx>

Gambar 4.5. Bandwidth Sebelum Queue Tree Pada Server

#### 4.6. Simulasi Queue Tree Dengan Mikrotik Router OS

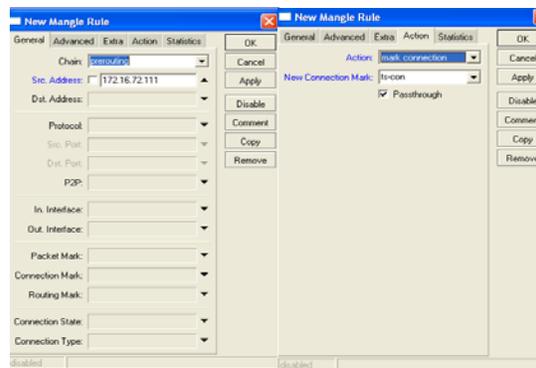
Simulasi dan konfigurasi dilakukan pada dua user, untuk user yang lain sama dengan konfigurasi pada user sebelumnya. Langkah-langkah untuk simulasi adalah sebagai berikut:

1. Konfigurasi *mangle* dengan *winbox* untuk membuat *rule* dan menandai paket.
  - a. Klik **ip > firewall** lalu klik **mangle**, klik tanda tambah (+) merah untuk membuat **mark connection** bagi *client* yang akan dilimit dan isi parameternya seperti dibawah ini:

Chain : prerouting  
 Src. Address : 172.16.72.yyy  
 adalah IP yang akan dilimit, klik tab Action dan isinya :  
 Action : mark

connection

New Con. Mark : ts-con →  
 nama connection mark untuk user ts  
 \



Gambar 4.6. Setting Mark-Connection

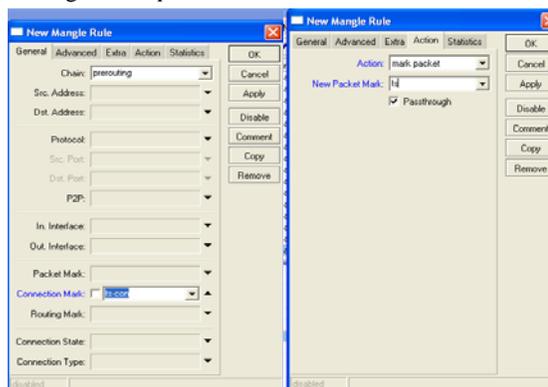
- b. Membuat rule **mark packet**, dengan mengklik tanda tambah merah lagi (+) dan isinya adalah sebagai berikut:

Chain : prerouting  
 Connection Mark : ts-con

adalah nama **mark connection** yang dibuat sebelumnya untuk user **ts**, lalu klik tab Action dan isinya :

Action : mark packet  
 New Packet Mark : ts,nama packet  
**mark** untuk user **ts**

Klik Apply dan OK untuk mengakhiri setting **mark packet**.



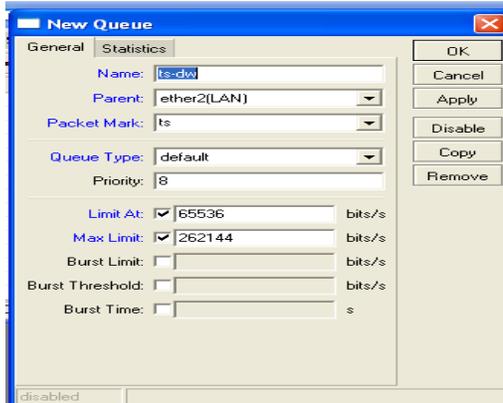
Gambar 4.7. Setting Mark-Packet

2. Konfigurasi *Queue Tree*

Pada simulasi ini jumlah *bandwidth download* yang akan dialokasikan pada user **ts** untuk *limit-at* =64kbps dan *max-limit*=512kbps . Sedangkan *bandwidth upload* yang dialokasikan untuk *limt-at*=64kbps dan *max-limit*=128kbps.

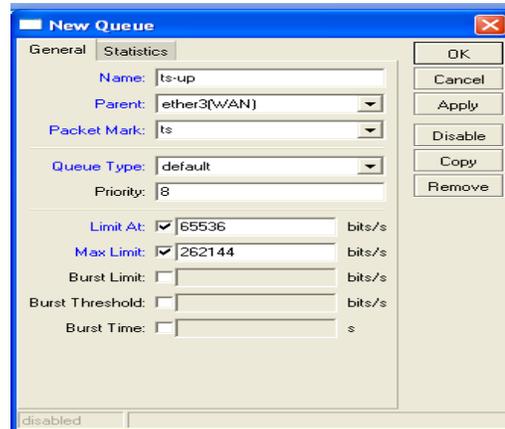
1. Klik **Queues** lalu klik tab **Queues Tree**
  - a. Membuat rule untuk membatasi **max. bandwidth download**, klik tanda tambah (+) dan isi parameternya sebagai berikut :  
 Name : ts-dw parameter untuk menamai **max. limit download ts**  
 Parent : ether2(LAN) adalah nama *interface (network card)* yang terhubung ke LAN.

*Packet Mark* : **ts** adalah nama packet mark yang tadi telah dibuat diatas  
*Limit At* : **65536 (64k)** yaitu untuk client **ts**  
*Max Limit* : **524288 (512k)** adalah **maksimal bandwidth download untuk user ts**  
 Terakhir klik *Apply* dan *OK* untuk mengakhiri *setting bandwidth download*.



Gambar 4.8. Setting *Bandwidth Download*

b. Membuat *rule* untuk membatasi **max. bandwidth upload**, klik tanda tambah (+) dan isi parameternya sebagai berikut :  
*Name* : **ts-up** adalah parameter untuk menamai **max. limit upload user ts**.  
*Parent* : **ether3(WAN)** adalah nama *interface* yang terhubung ke internet.  
*Packet Mark* : **ts** adalah nama packet mark yang telah dibuat sebelumnya.  
*Limit At* : **65536 (64k)** adalah *bandwidth* awal yang akan dilimit untuk user **ts**.  
*Max Limit* : **131072 (128k)** adalah **maksimal bandwidth upload untuk ts**. Terakhir klik *Apply* dan *OK* untuk mengakhiri *setting bandwidth upload*.



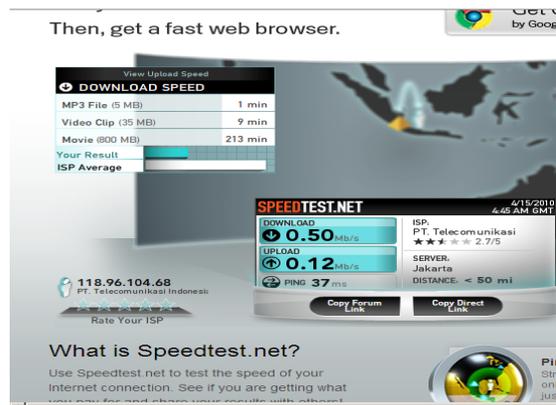
Gambar 4.9. Setting *Bandwidth Upload*

Sesudah metode *Hierarchical Token Bucket* (HTB) dan *Queue Tree* ini diterapkan dan dilakukan pengukuran *bandwidth* (*download* dan *upload*) pada user maka hasil yang diperoleh sesuai dengan yang dilokasikan.

#### 4.7. Hasil Simulasi *Queue Tree* Dengan Mikrotik Router OS

Sesudah metode *Hierarchical Token Bucket* (HTB) dan *Queue Tree* ini diterapkan dan dilakukan pengukuran *bandwidth* (*download* dan *upload*) pada user maka hasil yang diperoleh sesuai dengan yang dilokasikan.

Setelah melakukan simulasi maka dilakukan pengukuran *bandwidth* dengan menggunakan tool internet (<http://www.speedtest.net>). Hasil dari pengukuran *bandwidth* pada user **ts** dan **server** setelah metode *queue tree* diterapkan adalah sebagai berikut:



Sumber: <http://www.speedtest.net/172.16.72.xxx>

Gambar 4.10. Hasil Pengukuran *Bandwidth* Setelah *Queue Tree*

Tabel 4.2. Hasil Alokasi *Bandwidth* Pada User

Nama User	IP Address	Download	Upload
ts	172.16.72.xxx	0.50 Mbps	0.12 Mbps
server	172.16.72.xxx	1.03 Mbps	0.41 Mbps

Total jumlah *bandwidth* yang dialokasikan setelah metode *Hierarchical Token Bucket* (HTB) dan *Queue Tree* ini diterapkan dan

## 5. KESIMPULAN

Berdasarkan simulasi dan analisa diperoleh kesimpulan sebagai berikut:

- Penggunaan metode *Hierarchical Token Bucket* (HTB) dan *Queue Tree* terbukti bisa mengontrol penggunaan *bandwidth* internet ADSL.
- HTB memungkinkan membuat *queue* menjadi lebih terstruktur, dengan melakukan pengelompokan-pengelompokan bertingkat.
- Jika tidak mengimplementasikan HTB pada *Queue* (baik *Simple Queue* maupun *Queue Tree*), ada beberapa parameter yang tidak bekerja seperti: *priority* dan *dual limitation* (CIR/MIR).
- Cara paling mudah untuk melakukan *queue* dengan *queue tree*, adalah dengan menentukan parameter :
  - parent* (yang harus diisi dengan outgoing-interface).
  - packet-mark* (harus dibuat terlebih dahulu di ip-firewall-mangle).
  - max-limit* (yang merupakan batas kecepatan maksimum), atau dikenal juga dengan MIR (*Maximum Information Rate*).
- Sebelum melakukan *queue tree* maka terlebih dahulu harus mengkonfigurasi *mangle* untuk menandai paket.

Saran-saran untuk pengembangan lebih lanjut adalah sebagai berikut:

- Untuk simulasi diharapkan menggunakan perangkat keras yang memadai untuk hasil pengukuran yang lebih baik . Perangkat keras (*hardware*) yang dimaksud adalah dengan menggunakan Mikrotik Routerboard.



Sumber: <http://www.mikrotik.co.id/produk.php>

Gambar 5.1.Mikrotik Routerboard.

dilakukan diperoleh besar *bandwidth* (*download* dan *upload*) pada user adalah seperti yang ditampilkan dalam tabel diatas.Berdasarkan data pada tabel tersebut terbukti bahwa besar *bandwidth* yang di alokasikan sesuai dengan pengukuran berdasarkan tool internet (browser).

- Pelimitan (*limitation*) berdasarkan protokol icmp mungkin dapat diterapkan untuk memperbaiki nilai waktu respon pada manajemen *bandwidth* yang berdasarkan alamat IP.
- Untuk hasil pembagian *bandwidth* yang lebih akurat diharapkan supaya menggunakan *software* Mikrotik Router OS versi terbaru.
- Untuk implementasi *bandwidth control* yang lebih akurat diperlukan *bandwidth fixed* atau menggunakan koneksi *leased line* yang menggunakan media transmisi *fiber optic* (FO).
- Pengukuran atau testing *bandwidth* dapat dilakukan dengan *tool* lain misalnya: <http://speedtest.indosatm2.com>, untuk membuktikan hasil alokasi *bandwidth*.

## 5. DAFTAR PUSTAKA

- Abiona, Olatunde. 2006. *Bandwidth Monitoring & Measurement*. Obafemi Awolowo University Cone.
- Kustanto dan Daniel T.Saputro. 2008. *Membangun Server Internet Dengan Mikrotik OS*.Gava Media.Jakarta.
- Moch. Linto Herlambang dan Aziz Catur L. 2008. *Panduan Lengkap Menguasai Router Masa Depan Menggunakan Mikrotik Router OS*. Andi Offset.Yogyakarta.
- Nazir, M. 1988. *Metodologi Penelitian*. Ghalia Indonesia. Jakarta.
- Stefanus, S. 2005. *Reverse Engineering, Teori dan Aplikasi*. Universitas Diponegoro. Semarang
- Tanenbaum, Andrew S. 2003. *Computer Networks 4th Edition*. Prentice Hall.
- Unknown. 2008. *Divisi Training PT UFOAKSES SUKSES LUARBIASA*. Jakarta.
- <https://www.mikrotik.com/testdocs/ros/2.9/root/queue.pdf>.
- <http://www.mikrotik.com/testdocs/ros/2.9/refman2.9.pdf>.
- <http://www.mikrotik.com/testdocs/ros/3.0/refman3.0.pdf>.