IMPLEMENTASI *BANDWDITH MANAGEMENT* DAN *FIREWALL SYSTEM* MENGGUNAKAN MIKROTIK OS 2.9.27

Fatsyahrina Fitriastuti[1] , Dodi Prasetyo Utomo[2]

[1, 2] Jurusan Teknik Informatika, Universitas Janabadra
Jl. Tentara Rakyat Mataram No. 57 Yogyakarta 55231
E-Mail : fitri@janabadra.ac.id

ABSTRACT

At present, internet access not only performed using a personal computer (PC) or a laptop, but can use other mobile devices with ease, including tablets, iPhone, and smartphones. More and more devices to access the internet, the greater the bandwidth required. But it turns out that happened on the field at any bandwidth that could have gone only used by multiple devices. This is because there are no restrictions or bandwidth settings for each user. This research seeks to implement bandwidth management using simple queue and queue trees and implementing a firewall system using mangle and filter rules. This study uses the OS Mikrotik 2.9.27 as a router. The results of this study can be proved that by using MikroTik router OS 2.9.27 can be generated that serves as a limiter device and a firewall system, using a firewall filter rules and coupled with the layer 7 protocols can be made a router that serves as a barrier to access some sites desired, and using the limiter firewall mangle and queue trees can be distinguished browsing and downloading speeds.

Keyword : *bandwith management, firewall system, Mikrotik.*

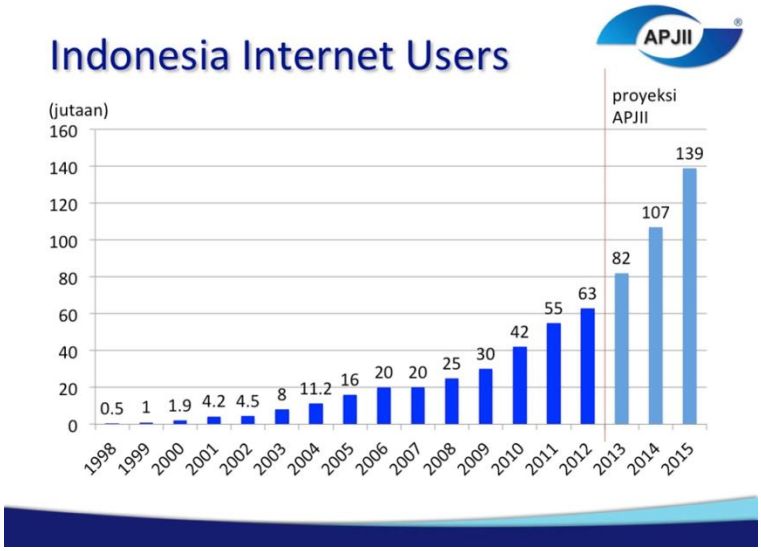
PENDAHULUAN

Internet sudah menjadi salah satu kebutuhan manusia yang penting di jaman modern ini. Saat ini terdapat 2,5 milyar pengguna internet di seluruh dunia (sumber : www.weasocial.sg). Sementara itu, di Indonesia menurut Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) mengungkapkan bahwa jumlah pengguna internet pada tahun 2013 mencapa 71,19 juta, atau sekitar 28% dari jumlah penduduk Indonesia. Ini meningkat 13 persen dibanding tahun 2012 yang mencapai sekitar 63 juta pengguna.

Menurut APJII, meski terjadi pertumbuhan pengguna internet 2013 dalam jumlah signifkan, namun untuk dapat memenuhi tuntutan *International Telecom Union* (ITU) yang menargetkan bahwa 50 persen penduduk Indonesia harus 'melek' internet pada tahun 2015 menjadi sesuatu yang berat. Sesuai dengan MGDs *(Millenium Development Goals)*, seharusnya pengguna internet di Indonesia mencapai 107 juta pada tahun 2014, dan 139 juta pengguna pada 2015 (Gambar 1).

Salah satu hambatan yang harus dihadapi adalah terbatasnya *bandwith* internet di Indonesia. *Bandwith* (lebar pita) merupakan istilah yang digunakan untuk menggambarkan berapa banyak informasi dapat dikirim melalui koneksi jaringan komputer. Ini biasanya dilambangkan sebagai bit per secon (bps), atau dengan beberapa denominasi bit yang lebih besar, seperti Megabits per secon (Mbps) atau Kilobits per secon (Kbps). Saat ini, akses internet tidak hanya dilakukan menggunakan *personal computer* (PC) atau laptop saja, tetapi dapat menggunakan perangkat *mobile* lainnya dengan mudah, diantaranya tablet, iPhone, dan *smartphone.* Bahkan menurut survey yang dilakukan oleh *On Device Research*, perusahaan riset yang berfokus pada sektor *mobile*, di dalam laporannya menjelaskan bahwa 24% penduduk Indonesia adalah pengguna internet. Uniknya, 62 %% dari pengguna internet tersebut mengakses internet melalui perangkat *mobile*, dan

kurang dari 10 persen mengakses internet dari rumah (sumber: id.techinasia.com).



Gambar 1. Statistik Pengguna Internet di Indonesia dari tahun ke tahun (Sumber: http://www.apjii.or.id)

Semakin banyak perangkat yang melakukan akses internet semakin besar pula *bandwith* yang dibutuhkan. Tetapi yang terjadi di lapangan ternyata berapapun *bandwith* yang di miliki dapat habis hanya digunakan oleh beberapa perangkat saja. Hal ini dikarenakan tidak ada pembatasan atau pengaturan bandwith untuk setiap *user*. Boleh jadi satu perangkat menghabiskan banwith yang cukup besar hanya untuk upload atau download file-file berukuran besar. Mencermati dari informasi diatas maka diperlukan sebuah *tool* yang dapat berfungsi sebagai pembatas atau pembagi kecepatan akses untuk masing-masing *user* internet. *Tool* yang dimaksud adalah *router*. *Router* merupakan sebuah alat yang berfungsi sebagai *gateway* untuk masing-masing *user* agar dapat terhubung dengan internet. Fungsi *router* selain sebagai *gateway* juga berfungsi sebagai *bandwith management.* Dalam penelitian ini akan digunakan Mikrotik OS 2.9.27.

Disamping implementasi *bandwith management*, dalam penelitian ini juga akan diimplementasikan *firewall system* yaitu sistem keamanan jaringan komputer yang digunakan untuk melindungi komputer dari berbagai jenis serangan dari komputer luar. *Firewall system* diperlukan karena dewasa ini internet merupakan jalur bebas hambatan untuk virus dan *malware* (*malicious software*) yang dapat membahayakan perangkat yang digunakan *user* untuk mengakses internet.

Tujuan dari penelitian ini adalah merancang dan mengimplementasikan *bandwith management* dan *firewall system* ke dalam suatu jaringan komputer dengan menggunakan Mikrotik OS 2.9.27 sebagai router.

Priyanto dalam penelitiannya yang berjudul "Analisis *Manajement Bandwidth* Berbasis Sistem Operasi Mikrotik" menerapkan *bandwidth limiter* dengan cara membagi *bandwidth* secara merata ke beberapa client. Menggunakana sistem operasi Mikrotik. Ada penelitian ini, peranan router hanya untuk membagi *bandwidth* dengan batasan tertentu, sehingga apabila hanya ada satu *client* yang sedang running maka *client* tersebut tidak dapat memperoleh *bandwidth* secara keseluruhan tetapi sesuai dengan bandwidth yang telah dibatasai melalui router.

Sementara itu, Tafaul Mujahidin (2011), dalam penelitiannya yang berjudul "OS Mikrotik Sebagai Manajemen Bandwith Dengan Menerapkan Metode *Per Conection Queue*". Dalam penelitian ini *bandwith* juga dibagi secara merata ke seluruh PC *client* dengan memberikan batasan limit sesuai dengan *bandwidth* yang ada. Tetapi dalam penelitian ini, ditambahkan metode PCQ *(Per Connection Queue)* dan juga penerapan

Queue Tree, dimana dengan menerapkan metode PCQ, *bandwidth* bisa dibagi secara otomatis oleh sistem dan apabila *bandwidth* digunakan hanya oleh satu *client*, maka *client* tersebut bisa mendapatkan keseluruhan bandwidth yang ada.

Sementara itu, penelitian yang dilakukan adalah mengimplementasikan *bandwith management* menggunakan metode *simple queue* dan *queue tree* dan mengimplementasikan *firewall system* menggunakan metode *mangle* dan *filter rules.* Impelementasi dilakukan pada CV. Maha Group, yaitu suatu perusahaan IT dengan fokus kegiatan utamanya adalah di bidang *internet application*, dan *internet network infrastructure* yang berlokasi di kota Yogyakarta. Salah satu jenis layanannya adalah ISP *(Internet Service Provider)* yang melayani internet berlangganan sehingga daat digunakan untuk menrapkan bandwit management dan firewall system. Penelitian ini menggunakan Mikrotik OS 2.9.27.

Metode *Bandwith Management*

Bandwidth adalah besaran untuk menunjukkan seberapa banyak data yang dapat dilewatkan dalam koneksi melalui sebuah *network* (Vektanova, 2003). Bandwith disebut juga lebar pita atau kapasitas saluran informasi yaitu kemampuan maksimum dari suatu alat untuk menyalurkan informasi dalam satuan waktu detik. Biasanya dilambangkan sebagai bit per secon (bps), atau dengan beberapa denominasi bit yang lebih besar, seperti Megabits per secon (Mbps) atau Kilobits per secon (Kbps). Bandwith manajement adalah cara pengaturan bandwith agar terjadi pemerataan pemakaian bandwith. Ada beberapa metode yang dapat diterapkan untuk mengimplementasikan bandwith manajement ini diantaranya melalui proxy server, QoS atau traffic shapping, atau pembatasan bandwith atau *limiter.*

Di dalam dunia internet sering di dengar istilah *limiter* atau pembatasan kecepatan untuk melakukan akses ke internet. Ada beberapa jenis *system limiter* yang biasa di aplikasikan ke *router*, mulai dari yang *simple* hingga yang komplek. Dalam penelitian ini digunakan dua metode untuk *bandwith management* yaitu *simple queue* dan *queue tree*.

a. *Simple queue* merupakan salah satu sistem *limiter* yang terdapat pada OS MikroTik dan merupakan cara termudah untuk membatasi laju data dari IP *address* atau *subnet* yang telah di tentukan atau di kenali (Paul, 2011). Keunggulan *simple queue* antara lain *simple queue* dapat melakukan pembatasan *rate* pada koneksi *peer to peer*, dapat melakukan pembatasan trafik pada aplikasi IDM (*internet download manager*) dan dapat melakukan pembatasan secara *fix*(tingkat kebocoran rendah). Tetai, *simple queue* juga memiliki kelemahan, yaitu karena menggunakan sistem *fix* pada limitasinya maka QoS (*quality of service*) sulit untuk diaplikasikan karena tidak bisa mengaplikasikan *parent system*. Pembatasan yang dilakukan *simple queue* adalah membatasi semua trafik paket baik TCP, ICMP, maupun UDP, serta *simple queue* tidak dapat digunakan untuk melakukan *bypass* trafik HIT pada trafik PROXY.

b. *Queue tree*, merupakan salah satu sistem limitasi berikutnya yang sering di aplikasikan pada *router* untuk membatasi data *rate* (Paul, 2011). *Queue tree* memiliki system yang lebih kompleks dibandingkan *simple queue*. *Queue tree* membutuhkan "kerjasama" dari *mangle* untuk menandai paket-paket dari alamat IP atau *subnet* tertentu untuk dijadikan parameter limitasi. Meskipun *queue tree* sedikit lebih sulit untuk diaplikasikan, namun sistem *limiter* ini menjadi idola bagi banyak orang. Yang menjadi keunggulan dari sistem *limiter* ini antara lain, mampu mengaplikasikan sistem *parent child*, mampu melimit berdasarkan paket (terintegrasi dengan *mangle*) sehingga dapat menentukan paket mana yang akan dipilih untuk dibatasi missal TCP, UDP dan ICMP. Selain itu *queue tree* juga memungkinkan apabila ingin melakukan *by pass* pada trafik HIT PROXY. Tetapi *queue tree* juga memiliki beberapa kelemahan antara lain, tidak dapat membatasi trafik yang berasal dari aplikasi IDM, tidak dapat membatasi koneksi *peer to peer*, sering terjadi kebocoran (apabila salah menentukan jumlah *max limit* client pada

sistem *parent*) dan agak sulit untuk pengaplikasiannya karena harus terintegrasi dengan *mangle* sebagai penentu indikator limitasi.

Firewall Mangle

Firewall mangle merupakan salah satu fitur *firewall* yang terdapat dalam MikroTik Router OS yang berfungsi sebagai penanda paket maupun koneksi data yang di *request* oleh *user* (Paul, 2004). Dalam prakteknya *firewall mangle* ini memiliki banyak fitur yang dapat dipergunakan, antara lain :

a. *Layer7 Protocol*
   Fungsi dari *Layer 7 protocol* adalah dipergunakan sebagai alternatif untuk pengisian *content* (Paul, 2004). *Layer 7 Protocol* adalah metode untuk mencari pola dalam *ICMP/TCP/UDP stream*. Atau istilah lainnya *regex pattern*. Cara kerja *Layer7 Protocol* adalah mencocokan (*mathcer*) 10 paket koneksi pertama atau 2 KB koneksi pertama dan mencari pola/*pattern* data yang sesuai dengan yang tersedia. Jika pola ini tidak ditemukan dalam data yang tersedia, *matcher* tidak memeriksa lebih lanjut. Dan akan dianggap *unknown connections*. Semakin banyak koneksi yang mengakses *Layer 7 Protocol* maka akan meningkatkan pula penggunaan memori di *router*. Untuk menghindari hal tersebut perlu menambahkan regular *firewall matchers* (*pattern*) untuk mengurangi jumlah data yang dikirimkan ke *layer 7 filter*.
b. *Connection Byte*
   Fungsi dari *connection byte* adalah sebagai *capture* untuk paket koneksi dengan besaran tertentu (*byte*) (Paul, 2004). Dengan *connection byte*, dapat dibedakan antara paket *browsing* dengan paket *download* berdasarkan besaran paket yang diakses. Apabila paket yang di *request* belum mencapai kriteria yang ditentukan maka paket tersebut tidak akan di *capture* oleh *mangle*, dengan menggunakan cara diatas maka trafik *browsing* dan trafik *download* dapat dibedakan.
c. *Content*
   Dengan menggunakan fitur *content* yang terdapat pada *firewall mangle*, dapat di lakukan *capture* paket berdasarkan alamat *website* tertentu (Paul, 2004). Dengan menggunakan fitur ini sangat memudahkan apabila ingin melakukan *capture* paket pada salah satu alamat *website* dan tidak mempengaruhi *capture* paket ke alamat *website* lainnya. Namun kekurangan dari fitur ini adalah keterbatasan alamat *website* dalam satu *mangle*.
d. *Src. Address List* dan *Dst. Address List*
   Src. Address List dan *Dst. Address List* berfungsi untuk "mengambil" data IP (*internet protocol*) yang terdapat pada address list di dalam fitur *firewall* (Paul, 2004). Dengan menggunakan fitur ini *network administrator* dapat dengan mudah mengelompokan IP (*internet protocol*) yang akan digunakan sebagai indikator penanda paket. Dengan fitur ini *network administrator* tidak direpotkan dengan pembuatan *mangle* yang terlalu banyak karena jumlah IP (*internet protocol*) yang dipergunakan.

Firewall Filter Rules

Filter filter rules merupakan salah satu fitur yang terdapat di MikroTik Router OS yang berfungsi sebagai pembatas ataupun pemberi akses paket koneksi (Paul, 2004). Di dalam *filter rules* terdapat fungsi DROP (menutup), ACCEPT (mengijinkan) yang digunakan sebagai pemberi akses paket koneksi baik koneksi yang diijinkan maupun paket yang tidak diperbolehkan melewati router. Seperti halnya *firewall mangle* di dalam *filter rules* terdapat fungsi seperti *connection byte*, *content* dan *dst address list* sebagai parameter untuk menandai paket yang akan di eksekusi. Yang menjadi perbedaan antara *firewall mangle* dan *filter rules* adalah apabila *firewall mangle* bertugas sebagai penanda paket yang nantinya akan di eksekusi oleh *limiter* (pada umumnya) sedangkan *firewall filter rules* bertugas sebagai pemberi akses untuk paket-paket yang diinginkan. Sebagai contoh apabila ingin dilakukan pembatasan kecepatan untuk paket tertentu maka yang bertanggung jawab adalah *firewall mangle* dan *limiter*, namun apabila ingin menutup akses ke alamat website atau IP tertentu maka ini menjadi tugas *filter rules* dan bias di kombinasikan dengan *address list*.

Address List

Address list adalah "tempat" untuk mengelompokan IP yang akan dijadikan sebuah *group* (Paul, 2004). Dengan

menggunakan *address list* ini lebih memudahkan apabila ingin menggunakan beberapa IP saja dari dalam satu segmen IP. IP yang akan di daftarkan nantinya dapat secara acak, hanya saja penamaan *group* dari IP tersebut harus sama apabila ingin masuk dalam satu *group*. Fungsinya adalah apabila memerlukan pembatasan kecepatan akses secara *group*, dapat menggunakan fitur ini untuk melakukan pendataan IP.

Mikrotik OS

MikroTik RouterOS™ adalah sistem operasi dan perangkat lunak yang dapat digunakan untuk menjadikan komputer manjadi *router network* yang handal, mencakup berbagai fitur yang dibuat untuk IP *network* dan jaringan *wireless*, cocok digunakan oleh ISP dan *provider hotspot* (Valens, 2004). Mikrotik OS merupakan OS yang sangat familiar dikalangan *network administrator* atau teknisi jaringan internet. Mikrotik OS didukung dengan fitur GUI (*grafic user interface*) yang memudahkan pengguna untuk mengakses ataupun melakukan konfigurasi menggunakan aplikasi WinBox. Dengan fitur GUI, OS tersebut menjadi lebih mudah digunakan karena kebanyakan OS *router* lainnya masih berbasis CLI (*command line interface*).

MikroTik RouterOS™ memiliki tingkatan level. Tiap level memiliki kemampuannya masing-masing, mulai dari level 3, hingga level 6. Secara singkat, level 3 digunakan untuk *router* ber*interface ethernet*, level 4 untuk *wireless client* atau *serial interface*, level 5 untuk *wireless* AP, dan level 6 tidak mempunyai limitasi apapun. Untuk aplikasi *hotspot*, bisa digunakan level 4 (200 *user*), level 5 (500 *user*) dan level 6 (*unlimited user*).

WinBox

WinBox adalah sebuah *utility* yang digunakan untuk melakukan *remote* ke *server* mikrotik dalam mode GUI (Valens, 2004). Jika untuk mengkonfigurasi mikrotik dalam *text mode* melalui PC itu sendiri, maka untuk mode GUI yang menggunakan winbox ini dapat melakukan konfigurasi mikrotik melalui komputer *client*.

Mengkonfigurasi mikrotik melaui WinBox ini lebih banyak digunakan karena selain penggunaannya yang mudah karena tidak harus menghapal perintah-perintah *console*. Kelebihan dari WinBox ini adalah kemudahan dalam melakukan *remote* karena berbasis GUI.

Putty

Selain menggunakan aplikasi WinBox untuk mengakses *router* juga dapat menggunakan aplikasi *putty* (Valens, 2004). Berbeda dengan WinBox yang mengaplikasikan sistem GUI *putty* adalah sebuah *utility* yang digunakan untuk melakukan *remote* ke *server* dalam mode CLI (*command line interface*). Dengan menggunakan *putty*proses *remote* menggunakan *protocol* SSH yang mempunyai *default* port 22. Kelebihan dari *utility putty* ini antara lain adalah lebih ringan dalam proses *remote* karena hanya menggunakan sedikit *bandwidth* untuk proses *load text*. Namun me*remote* menggunakan putty ini juga memiliki kelemahan antara lain harus menghafal *script* perintah karena berbasis CLI.

METODE PENELITIAN

Metode atau tahapan dalam melakukan penelitian ini adalah :
lain :

1. Pengumpulan Data

Data-data yang diperlukan, diantaranya diperoleh dari *interview* atau wawancara secara langsung dengan para teknisi CV. Maha Group.

2. Rancangan Sistem

Melakukan perancangan sistem dengan tahapan merancang alokasi *bandwith*, merancang topologi jaringan, menentukan website yang akan di *bypass* maupun di blok, menentukan besaran kategori paket download serta pembagian *IP address*.

3. Proses Instalasi dan Konfigurasi

Melakukan proses instalasi *router* MikroTik sebagai *bandwith management system*, konfigurasi sistem *limiter* dan *firewall system*.

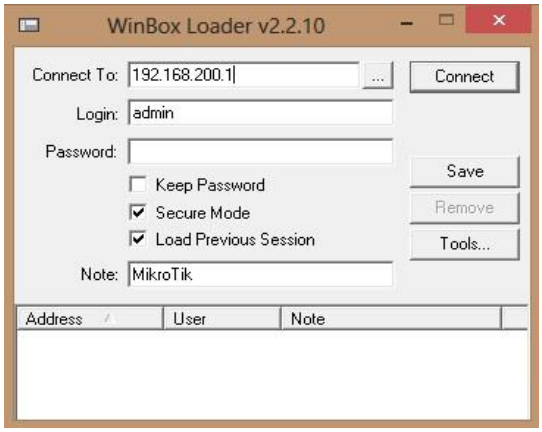
4. Implementasi

Implementasi *router* pada jaringan sebagai *bandwith management system* dan *firewall system*.

HASIL DAN PEMBAHASAN

WinBox

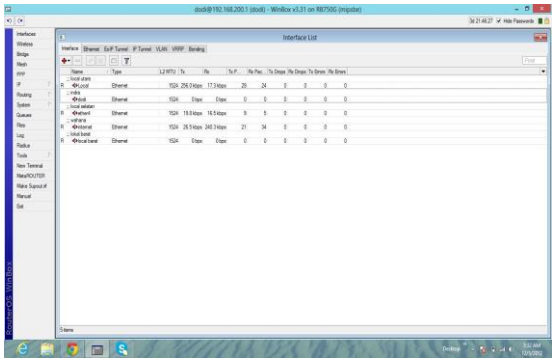
Sebelum masuk dan melakukan konfigurasi ke sistem, terlebih dahulu digunakan aplikasi *WinBox* untuk me-*remote router*. Dalam penggunaan aplikasi *WinBox* dibutuhkan alamat IP *router*, IP user beserta *password* dari *router* yang akan di *remote*.



Gambar 2. Aplikasi WinBox

Tabel *Interface*

Tabel *interface* berfungsi sebagai tabel pemantau jumlah *Lancard* yang terdeteksi oleh perangkat *router*. Didalam *table interface* ini juga dapat melakukan pelabelan terhadap *Lancard* yang telah dikenali sebelumnya supaya tidak terjadi kesalahan peletakan IP *address* ataupun pemasangan kabel jaringan. Didalam table *interface* ini juga dapat terpantau apakah kabel jaringan sudah terkoneksi ataupun belum terkoneksi.

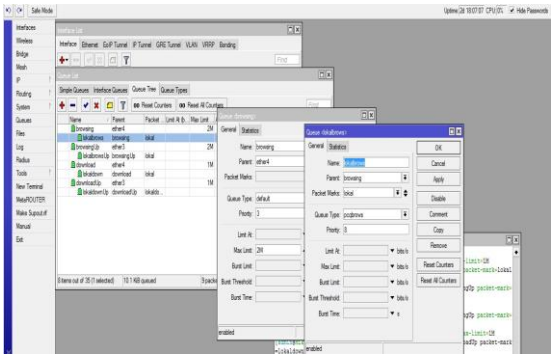


Gambar 3. Tabel Interface

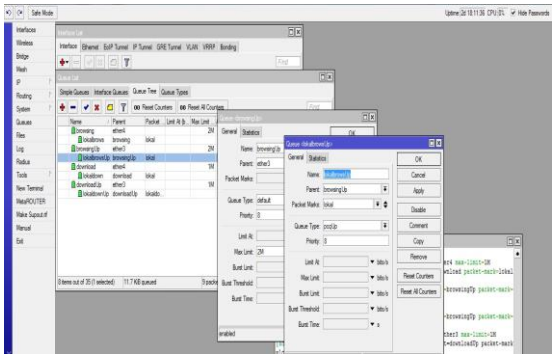
Implementasi *Limiter Queue Tree*

Di dalam tabel *queue* list terdapat beberapa tipe limiter antara lain *simple queue* dan *queue tree*. Di dalam tab *queue tree* terdapat fungsi menambahkan *limiter*, menghapus *limiter*, mematikan *limiter* dan mengaktifkan *limiter*. Ke empat fungsi diatas adalah fungsi-fungsi yang sering digunakan dalam implementasi *system limiter*. Apabila fungsi yang dipilih adalah fungsi *ADD* atau menambahkan yang di wakili dengan tanda (+) berwarna merah, maka akan tampil kotak dialog yang berisi keterangan-keterangan untuk diisi.

1. *Name* : dalam kolom *name* dapat diisikan dengan nama perangkat yang akan di *limit*.
2. *Parent* : dalam kolom *parent* berfungsi untuk menentukan *limiter* tersebut menginduk ke *limiter* lain atau ke salah satu *interface*
3. *Packet Marks* : dalam kolom *packet marks* di tentukan paket koneksi yang akan di *limit*
4. *Queue type* : dalam kolom *queue type* dapat dipilih jenis *limiter* yang telah dibuat sebelumnya didalam tab *Queue Types*
5. *Priority* : dalam kolom *priority* berfungsi untuk menentukan prioritas urutan *limiter* yang akan dijalankan terlebih dahulu.
6. *Limit at* : di dalam kolom max limit terdapat besaran bandwith yang berfungsi sebagai pembatas minimal.
7. *Max Limit* : di dalam kolom max limit terdapat besaran bandwith yang berfungsi sebagai pembatas maksimal.



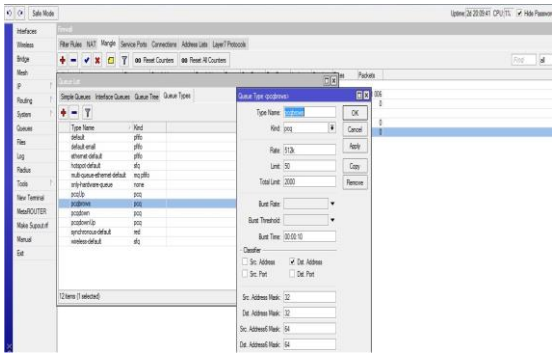
Gambar 4. Kotak dialog *queue tree limit download*



Gambar 5. Kotak dialog *queue tree limit upstream*

Implementasi Queue Types

Queue type berfungsi untuk membantu mengatur kerja *limiter* baik pada *simple queue* maupun *queue tree*. Dengan menggunakan *queue type* dapat memudahkan dalam pembatasan kecepatan tanpa harus memasukkan besaran *limiter* pada masing-masing IP yang akan dibatasi kecepatannya.

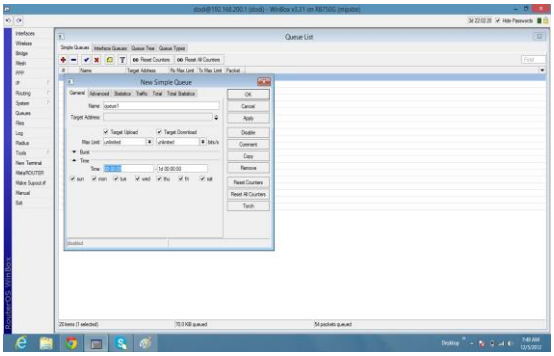


Gambar 6. Kotak dialog *queue type*

Pada gambar diatas terdapat beberapa kolom yang memiliki *value*. Pada kolom-kolom diatas yang harus diubah *value* adalah kolom *Type Name* yang akan menjadi nama untuk *queue type* yang dibuat dan pada kolom *Rate* yang akan berfungsi sebagai pembatasan kecepatan maksimal pada masing-masing IP.

Implementasi *Timer* pada *Simple Queue*

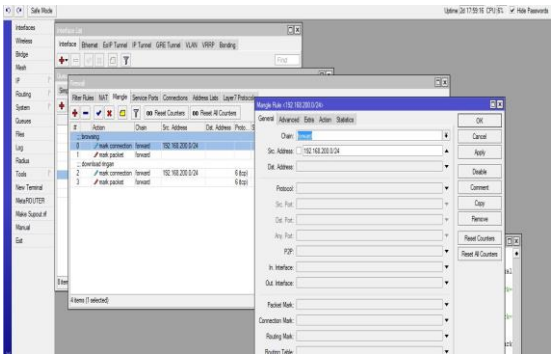
Di dalam *limiter simple queue* terdapat juga fitur *time*. Fitur *time* biasanya difungsikan apabila terdapat dua sistem atau lebih besaran konfigurasi *limiter* yang akan digunakan. Misal dalam implementasi terdapat dua konfigurasi besaran limiter antara waktu siang dan malam, fitur *time* dapat diaktifkan supaya tidak perlu mematikan dan menghidupkan limiter secara manual cukup mengaktifkan fitur *time*, maka limiter akan berganti secara otomatis.



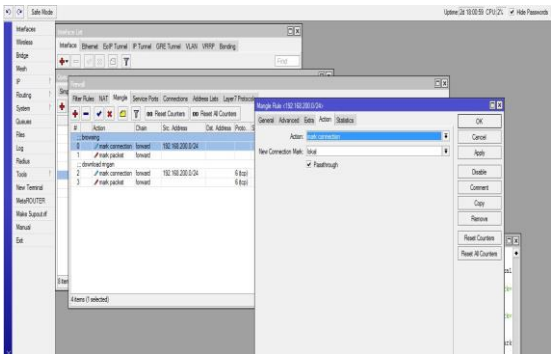
Gambar 7. Fitur *time* pada *simple queue*

Implementasi Mangle

Untuk dapat mengaktifkan pembatasan *bandwith* di *queue tree*, sebelumnya perlu menambahkan *mangle* pada *firewall mangle* yang berfungsi untuk menandai paket koneksi yang berasal dari klien. Dalam konfigurasi *queue tree* diatas, pembatasan *bandwith* di bagi menjadi dua yaitu *limiter download* dan *limiter browsing*. Dengan pemisahan *limiter* seperti diatas maka pada konfigurasi *mangle* memerlukan dua konfigurasi untuk dapat menjalankan konfigurasi *limiter* tersebut.

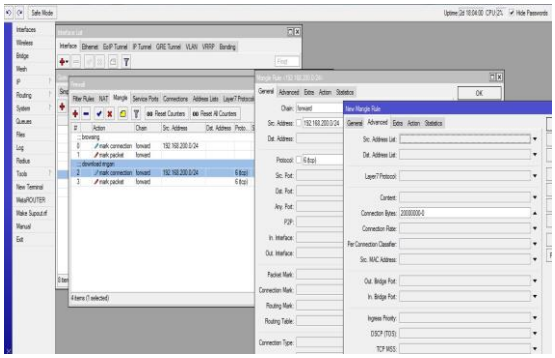


Gambar 7. Konfigurasi *mangle browsing*



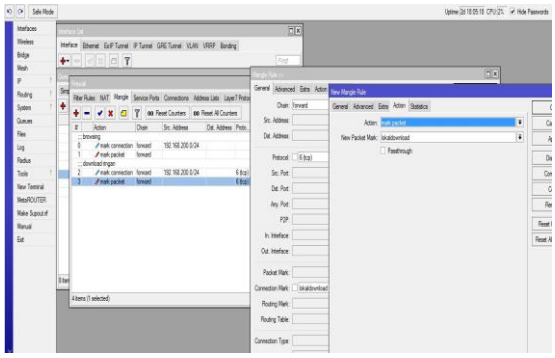
Gambar 8. Konfigurasi mangle browsing 2

Pada gambar pertama terlihat pada kolom Src. Address terdapat IP 192.168.200.0/24 yang merupakan IP dari *network* lokal, kemudian di gambar kedua *network* IP tersebut dilabeli dengan nama "lokal". Paket dengan nama "lokal" tersebut yang akan di jadikan parameter *limiter* di *queue tree* paket browsing.



Gambar 9. Konfigurasi mangle download

Dalam gambar diatas terlihat Src. Address yang sama dengan *mangle browsing*, hal ini dikarenakan IP tersebut merupakan IP dari *network* lokal pada jaringan diatas. Yang menjadi perbedaan adalah pada bagian Connection Bytes pada gambar diatas terdapat parameter angka 20000000-0, yang berarti konfigurasi *mangle* diatas akan menangkap paket apabila klien telah melakukan *request* paket hingga 20MB.

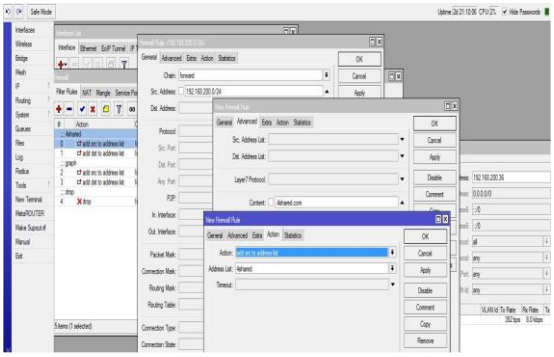


Gambar 10. Konfigurasi *Mangle Download* 2

Pada gambar diatas paket yang telah ditandai dengan parameter Connection Bytes kemudian dilabeli dengan nama "lokaldownload" yang nantinya akan digunakan sebagai parameter *limiter download* pada konfigurasi *queue tree*.

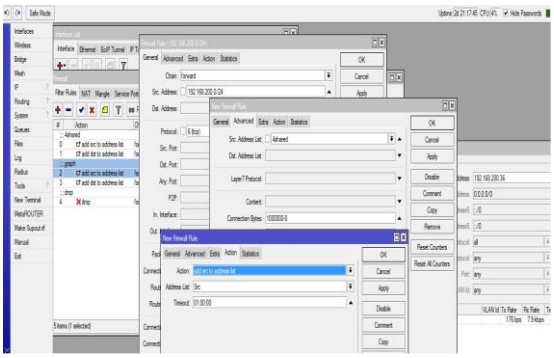
4. 9 Implementasi Firewall Filter Rules

Selain proses pembatasan kecepatan *browsing* maupun *download* yang di *handle* oleh *queue tree* dan *mangle*, dapat juga dilakukan penutupan akses *download* atau *browsing* ke alamat *website* tertentu. Untuk melakukan hal tersebut dapat menggunakan *filter rules* yang dikombinasikan dengan *address list* sebagai penyimpanan IPnya.

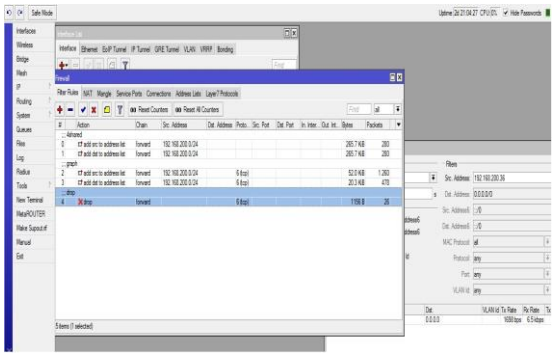


Gambar 11. Konfigurasi filter rules tandai IP website

Pada gambar diatas konfigurasi filter rules berfungsi untuk menangkap IP dari website yang akan difungsikan sebagai parameter DROP. Setelah IP dari website yang akan dijadikan paremeter tersimpan di address list scrip koonfigurasi selanjutnya bertugas sebagai parameter besaran request paket yang di ambil oleh klien menggunakan fitur connection byte kemudian IP tersebut akan disimpan kembali di address list dengan nama Src dan Dst. Dan setelah IP klien selesai ditandai dan tersimpan di dalam address list, maka konfigurasi terakhir berfungsi sebagai penutup jalur koneksi yang menuju alamat website tersebut dengan besaran request paket yang telah tercapai sesuai pada parameter dari script konfigurasi kedua.



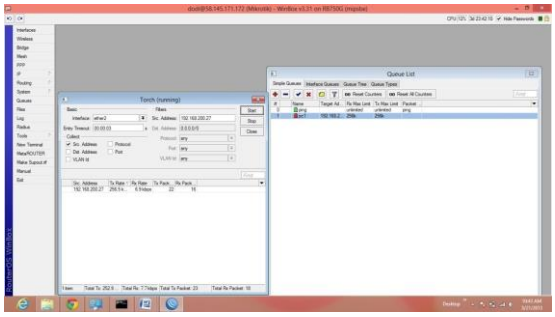
Gambar 12. Konfigurasi filter rules tandai IP klien



Gambar 13. Konfigurasi filter rules DROP koneksi

4.10 Pengecekan Limiter Melalui Torch

Setelah proses konfigurasi *limiter* selesai maka proses selanjutnya adalah pengecekan berfungsi atau tidaknya *limiter* tersebut. Sebagai contoh pc1 dengan Ip *address* 192.168.200.27 dilimit dengan *max-limit* sebesar 256kbps. Pada gambar terlihat bahwa Pc1 telah mencapai kecepatan maksimal yang di berikan (terlihat dari warna merah pada symbol Pc1). Kemudian di tabel torch di samping kiri terlihat bahwa ip 192.168.200.27 telah mencapai kecepatan 256kbps. Dengan hasil tersebut maka limiter telah berhasil di implementasikan pada router ini.



Gambar 14. Pengecekan limiter

KESIMPULAN

Kesimpulan

Kesimpulan yang dapat diambil dari penelitian ini adalah:

1. Dengan menggunakan MikroTik OS 2.9.27 dapat dihasilkan *router* yang berfungsi sebagai perangkat *limiter* dan *firewall system*.
2. Dengan menggunakan *firewall filter rules* dan dipadukan dengan *layer 7 protocols* dapat di buat sebuah *router* yang berfungsi sebagai pembatas akses ke beberapa situs yang diinginkan.
3. Dengan menggunakan *limiter queue tree* dan *firewall mangle* dapat dibedakan kecepatan *browsing* maupun *download*.
4. Implementasi trafik HIT yang digabungkan dengan *proxy server* dapat di jalankan pada router dengan MikroTik Router OS.

Saran

Untuk meningkatkan kualitas bandwith management penulis menyarankan agar kedepannya diterapkan system *limiter* menggunakan *Queue Tree*. Karena dengan menggunakan *system limiter queue tree* trafik HIT (trafik paket yang pernah diakses) dari proxy dapat di optimalkan, sehingga apabila di dalam penerapannya nanti terdapat proxy dapat lebih maksimal dalam *bypass* trafik HIT (trafik paket yang pernah diakses) yang berasal dari proxy.

DAFTAR PUSTAKA

Paul, G, 2011, Manual CD Install, http://wiki.mikrotik.com/wiki/Manual:CD_Install, diakses pada tanggal 13 Maret 2013.

Paul, G, 2011, Manual First Time startup, http://wiki.mikrotik.com/wiki/Manual:First_time_startup, diakses pada tanggal 14 Maret 2013.

Syafizal, M, 2005, **Pengantar Jaringan Komputer**, Andi Publisher, Yogyakarta.

Valens, R, 2005, *Simple Queue, Memisah Bandwith Lokal dan Internasional*, http://mikrotik.co.id/artikel_lihat.php?id=23, diakses pada tanggal 20 Maret 2013.

Valens, R, 2005, *Membuat Jaringan*, http://mikrotik.co.id/artikel_lihat.php?id=5, diakses pada tanggal 21 Maret 2013.