

**ANALISIS MANAJEMEN *BANDWIDTH HIERARCHICAL
TOKEN BUCKET* (HTB) DENGAN MIKROTIK PADA
JARINGAN SMK NEGERI 22 JAKARTA**



MUHAMMAD IQBAL ICHWAN

5235127244

Skripsi Ini Ditulis Untuk Memenuhi Sebagian Persyaratan Dalam Memperoleh
Gelar Sarjana

PENDIDIKAN TEKNIK INFORMATIKA DAN KOMPUTER

FAKULTAS TEKNIK

UNIVERSITAS NEGERI JAKARTA

2017

HALAMAN PENGESAHAN

NAMA DOSEN

TANDA TANGAN

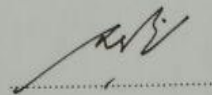
TANGGAL

Lipur Sugiyanta, Ph.D
(Dosen Pembimbing I)



21-8-2017

Prasetyo Wibowo Y, ST., M.Eng
(Dosen Pembimbing II)



21-8-2017

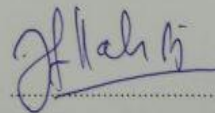
PENGESAHAN PANITIA UJIAN SKRIPSI

NAMA DOSEN

TANDA TANGAN

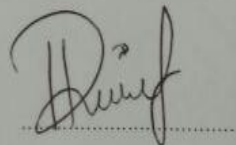
TANGGAL

Hamidillah Ajie, M.T
(Ketua Penguji)



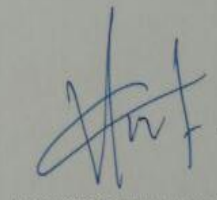
21-8-2017

Diat Nurhidayat., M.Ti
(Sekretaris Penguji)



21-8-2017

Vina Oktaviani
(Dosen Ahli)



18-8-2017

Tanggal Lulus : 21-8-2017

HALAMAN PERNYATAAN

Dengan ini saya menyatakan bahwa :

1. Karya skripsi saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik sarjana, baik di universitas Negeri Jakarta maupun di perguruan tinggi lain
2. Karya tulis ini murni gagasan, rumusan, dan penelitian saya sendiri dengan arahan dosen pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah di tulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dan dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan tidak kebenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademi berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai norma yang berlaku di Universitas Negeri Jakarta.

Jakarta, 16 Agustus 2017

Yang membuat pernyataan



Muhammad Iqbal Ichwan

5235127244

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa yang selalu memberi kelancaran dan kemudahan, sehingga skripsi “Analisis Manajemen Bandwidth Hierarchical Token Bucket (HTB) Dengan Mikrotik Pada Jaringan SMK Negeri 22 Jakarta” dapat terselesaikan.

Dalam menyelesaikan skripsi ini, penulis telah mencurahkan segala kemampuan dan keterbatasan yang dimiliki. Seringkali, penulis juga menemukan kesulitan dalam proses penyusunan skripsi. Namun, ada banyak pihak yang memberikan bimbingan serta dukungan pada peneliti sehingga skripsi ini terselesaikan. Oleh sebab itu, pada kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Lipur Sugiyanta Ph.d, selaku dosen pembimbing I yang selalu memberikan waktu dan bimbingannya dalam menyelesaikan skripsi ini.
2. Bapak Prasetyo Wibowo Yunanto, ST., M.Eng selaku dosen pembimbing II yang selalu memberikan waktu dan bimbingannya dalam menyelesaikan skripsi ini.
3. Ibu Dr. Yuliatris Sastrawidjaya, M.Pd. selaku ketua Program Studi Pendidikan Teknik Informatika dan Komputer, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Jakarta.
4. Seluruh dosen dan staf tata usaha Jurusan Teknik Elektro yang selalu membantu menyediakan informasi dan membantu proses administrasi skripsi.

5. Orang tua yang selalu memberikan semangat dan dukungan serta doa yang ikhlas dalam penyelesaian skripsi.
6. Bapak M.Awaludin S.Kom selaku guru jurusan Teknik Komputer Jaringan di SMK Negeri 22 Jakarta yang sudah banyak memberikan arahan dan masukan mengenai penelitian ini.
7. Teman-teman PTIK yang saling mendukung satu sama lain dalam mengerjakan skripsi.
8. Semua pihak yang secara langsung maupun tidak langsung membantu proses penyelesaian skripsi ini.

Penulis menyadari bahwa skripsi ini masih banyak kekurangan baik isi maupun susunannya. Semoga skripsi ini dapat bermanfaat dan berguna bagi pembaca serta dapat mendukung kemajuan ilmu pengetahuan khususnya bidang pendidikan.

Jakarta, Agustus 2017

Muhammad Iqbal Ichwan

Analisis Manajemen *Bandwidth Hierarchical Token Bucket (HTB)* Dengan Mikrotik Pada Jaringan SMK Negeri 22 Jakarta

Muhammad Iqbal Ichwan

ABSTRAK

Penggunaan Bandwidth di sebuah jaringan seringkali kurang dimanfaatkan secara optimal. Salah satu solusi yang efektif untuk mengatasinya adalah dengan mengelola pemakaian bandwidth dengan melakukan manajemen bandwidth. Saat ini SMK Negeri 22 Jakarta menerapkan manajemen bandwidth dengan mikrotik RB 1100AHX2 menggunakan fitur simple queue. Pada penelitian ini dilakukan manajemen bandwidth dengan mikrotik RB 750Gr3 menggunakan fitur queue tree dengan menggunakan metode hierarchical token bucket (HTB). Setelah dilakukan manajemen bandwidth dengan metode HTB, kemudian dilakukan pengukuran Quality of Service (QoS) dengan parameter bandwidth, throughput, delay, dan packet loss. Hasil yang didapat setelah dilakukan manajemen bandwidth dengan metode HTB yaitu pada parameter throughput mengalami kenaikan 2.7746% pada perangkat pengukuran cnn.com dan 13.3415% pada facebook.com. Pada parameter packet loss mengalami penurunan 13% pada cnn.com, 17% pada facebook.com dan 16% pada mail.yahoo.com. Pada parameter delay mengalami penurunan dari yang sebelumnya 135 ms menjadi 41 ms pada facebook.com dan 108 ms menjadi 90 ms pada mail.yahoo.com. Pada indeks nilai akhir mengalami kenaikan dari 2.33 menjadi 2.89.

Kata kunci : Manajemen bandwidth, Hierarchical Token Bucket (HTB)

Analisis Manajemen *Bandwidth Hierarchical Token Bucket (HTB)* Dengan Mikrotik Pada Jaringan SMK Negeri 22 Jakarta

Muhammad Iqbal Ichwan

ABSTRACT

Bandwidth usage in a network is often underutilized optimally. One effective solutions to overcome it is to manage bandwidth usage by performing bandwidth management. Nowadays SMK Negeri 22 Jakarta applied bandwidth management with simple queue feature by using mikrotik RB 1100AHX2. In this research, bandwidth management is applied using Queue Tree feature with Mikrotik RB 750Gr3 using Hierarchical Token Bucket (HTB) method. After doing bandwidth management with HTB method, then measured Quality of Service (QoS) with bandwidth, throughput, delay, and packet loss as a parameter. After applying bandwidth management with HTB method, the results obtained are throughput increased 2.7746% on cnn.com measurement devices and 13.3415% on facebook.com. Packet loss decreased 13% on cnn.com, 17% on facebook.com and 16% on mail.yahoo.com. Delay decreased from 135 ms to 41 ms on facebook.com and 108 ms to 90 ms on mail.yahoo.com. Final value index increased from 2.33 to 2.89.

Key words : Bandwidth Management, Hierarchical Token Bucket (HTB)

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
HALAMAN PERNYATAAN	iii
KATA PENGANTAR	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN	xv
 BAB I : PENDAHULUAN	
1.1 Latar Belakang Masalah	1
1.2 Identifikasi Masalah	5
1.3 Pembatasan Masalah	6
1.4 Perumusan Masalah	6
1.5 Tujuan Penelitian	6
1.6 Manfaat Penelitian	6
 BAB II : KERANGKA TEORITIK DAN KERANGKA BERPIKIR	
2.1. Kerangka Teoritik	8
2.1.1. Konsep Dasar Jaringan Komputer	8
2.1.1.1. Jaringan Komputer	8

2.1.1.2. Jenis-jenis Jaringan Komputer	9
2.1.2. <i>Quality of Service</i>	10
2.1.3. Parameter Kualitas Layanan	13
2.1.3.1. <i>Bandwidth</i>	13
2.1.3.2. <i>Throughput</i>	14
2.1.3.3. <i>Delay</i>	15
2.1.3.4. <i>Packet loss</i>	16
2.1.4. Mikrotik	16
2.1.4.1. Jenis-jenis Mikrotik	17
2.1.4.2. Fitur-fitur Mikrotik	18
2.1.4.3. Cara Akses Mikrotik	19
2.1.5. <i>Management Bandwidth</i>	20
2.1.5.1. Pengertian <i>Management Bandwidth</i>	20
2.1.5.2. Jenis <i>Management Bandwidth</i>	22
2.1.6. <i>Hierarchical Token Bucket (HTB)</i>	24
2.1.6.1. Parameter <i>Hierarchical Token Bucket (HTB)</i>	28
2.1.6.2. Mekanisme Kerja <i>Hierarchical Token Bucket</i>	28
2.1.6.3. Estimator Pada HTB	29
2.2. Kerangka Berpikir	31

BAB III : METODOLOGI PENELITIAN

3.1. Tempat dan Waktu Penelitian	32
3.2. Alat dan Bahan Penelitian	32
3.2.1. Alat Penelitian	32
3.2.1.1. Perangkat Keras	32

3.2.1.2. Perangkat Lunak	32
3.2.2. Bahan Penelitian	32
3.2.2.1. Alokasi <i>Bandwidth</i>	33
3.2.2.2. Topologi Jaringan	36
3.2.2.3. Spesifikasi <i>Routerboard</i> Mikrotik	37
3.3. Diagram Alir Penelitian	39
3.3.1. Observasi Lapangan	40
3.3.2. Identifikasi Masalah	40
3.3.3. Pengumpulan Data	41
3.3.4. Penerapan Manajemen <i>Bandwidth</i> dengan metode HTB	41
3.3.5. Pengambilan Data	41
3.3.6. Analisis Perbandingan Data Sebelum dan Sesudah Menggunakan Manajemen <i>Bandwidth</i> HTB.....	42
3.4. Teknik dan Prosedur Pengumpulan Data	42
3.5. Teknik Analisis Data	43
3.5.1. Pengukuran <i>Bandwidth</i>	43
3.5.2. Pengukuran <i>Throughput</i>	44
3.5.3. Pengukuran <i>Delay</i>	44
3.5.4. Pengukuran <i>Packet loss</i>	45

BAB IV : HASIL PENELITIAN

4.1. Deskripsi Hasil Penelitian	47
4.1.1. Konfigurasi Router Mikrotik	47
4.1.2. Konfigurasi Manajemen <i>Bandwidth</i> HTB	55
4.1.2.1. Konfigurasi Mangle	55

4.1.2.2. Konfigurasi HTB	59
4.1.3. Hasil Pengukuran <i>Bandwidth</i>	65
4.1.4. Hasil Pengukuran <i>Throughput</i>	66
4.1.5. Hasil Pengukuran <i>Delay</i>	68
4.1.6. Hasil Pengukuran <i>Packet loss</i>	69
4.2. Analisis Data Penelitian	71
4.2.1. Manajemen <i>Bandwidth</i> Hierarchical Token Bucket (HTB)	71
4.2.1.1. Test <i>Download</i> Sebelum Penerapan Manajemen	
<i>Bandwidth</i> HTB	72
4.2.1.2. Test <i>Download</i> Sebelum Penerapan Manajemen	
<i>Bandwidth</i> HTB	73
4.2.2. <i>Bandwidth</i>	75
4.2.3. <i>Throughput</i>	76
4.2.4. <i>Delay</i>	77
4.2.5. <i>Packet loss</i>	79
4.3. Pembahasan	80
4.4. Aplikasi Hasil Penelitian	81
BAB V : KESIMPULAN DAN SARAN	
5.1. Kesimpulan	82
5.2. Saran	82
DAFTAR PUSTAKA	84
LAMPIRAN	86

DAFTAR TABEL

Tabel 1.1 Hasil Akhir Analisis Dari Ruang Guru	4
Tabel 2.1 Indeks Parameter Qos	13
Tabel 2.2 Standarisasi <i>Throughput</i>	15
Tabel 2.3 Standarisasi <i>Delay</i>	16
Tabel 2.4 Standarisasi <i>Packet loss</i>	16
Tabel 3.1 Spesifikasi RB1100AHX2	37
Tabel 3.2 Spesifikasi RB 750Gr3	39
Tabel 4.1 <i>Ip Address</i>	50
Tabel 4.2 Hasil Pengukuran <i>Bandwidth</i>	66
Tabel 4.3 Hasil Pengukuran <i>Throughput</i>	67
Tabel 4.4 Hasil Pengukuran <i>Delay</i>	68
Tabel 4.5 Hasil Pengukuran <i>Packet loss</i>	70
Tabel 4.6 Hasil Akhir Pengukuran <i>Throughput</i>	77
Tabel 4.7 Hasil Akhir Pengukuran <i>Delay</i>	78
Tabel 4.8 Hasil Akhir Pengukuran <i>Packet loss</i>	79
Tabel 4.9 Hasil Analisis Akhir Dari Seluruh Parameter	80

DAFTAR GAMBAR

Gambar 2.1 HTB Sistem	27
Gambar 2.2 Skema Antrian HTB	29
Gambar 2.3 <i>Token Bucket Filter</i>	30
Gambar 2.4 Kerangka Berpikir	31
Gambar 3.1 Topologi Jaringan SMK Negeri 22 Jakarta Sebelum Menerapkan Metode HTB	36
Gambar 3.2 Topologi Jaringan SMK Negeri 22 Jakarta Setelah Menerapkan Metode HTB	36
Gambar 3.3 Diagram Alir Penelitian	40
Gambar 3.4 <i>User Interface</i> Speedtest.net	43
Gambar 3.5 Hasil Pengukuran <i>Throughput</i>	44
Gambar 3.6 Layanan Pada Fitur Netwatch	45
Gambar 3.7 Hasil Pengukuran <i>Delay</i>	45
Gambar 3.8 Hasil Pengukuran <i>Packet loss</i>	46
Gambar 4.1 Tampilan Awal Winbox	47
Gambar 4.2 Tampilan Awal Mikrotik Pada Winbox	48
Gambar 4.3 Merubah <i>Password</i> Mikrotik	49
Gambar 4.4 Konfigurasi <i>IP Address</i>	49
Gambar 4.5 Konfigurasi <i>Gateway</i>	50
Gambar 4.6 Konfigurasi <i>DNS</i>	51
Gambar 4.7 Konfigurasi <i>Masquerade</i>	52
Gambar 4.8 Hasil Konfigurasi <i>Masquerade</i>	53

Gambar 4.9 Konfigurasi DHCP Server	54
Gambar 4.10 Hasil Konfigurasi DHCP Server	55
Gambar 4.11 Konfigurasi <i>Marking Connection</i>	56
Gambar 4.12 Hasil Konfigurasi <i>Marking Connection</i>	57
Gambar 4.13 Konfigurasi <i>Marking Packet</i>	58
Gambar 4.14 Hasil Konfigurasi <i>Marking Packet</i>	58
Gambar 4.15 Konfigurasi <i>Queue Tree</i> Dengan Metode HTB	60
Gambar 4.16 Konfigurasi <i>Parent Queue Traffic Download</i>	61
Gambar 4.17 Konfigurasi <i>Child Queue Traffic Download</i>	62
Gambar 4.18 Konfigurasi <i>Parent Queue Traffic Upload</i>	63
Gambar 4.19 Konfigurasi <i>Child Queue Traffic Upload</i>	64
Gambar 4.20 Hasil Akhir Konfigurasi <i>Queue Tree</i> Dengan Metode HTB	65
Gambar 4.21 Hasil Pengukuran <i>Bandwidth</i> Senin 17 April 2017	66
Gambar 4.22 Hasil Pengukuran <i>Throughput</i> Senin 17 April 2017	67
Gambar 4.23 Grafik <i>Throughput</i> Senin 17 April 2017.....	68
Gambar 4.24 Hasil Pengukuran <i>Delay</i> Senin 17 April 2017.....	69
Gambar 4.25 Grafik <i>Packet loss</i> Senin 17 April 2017	70
Gambar 4.26 Hasil <i>Download</i> Sebelum Penerapan Manajemen <i>Bandwidth</i> HTB Oleh 2 Client Secara Bersamaan	72
Gambar 4.27 Hasil <i>Download</i> Setelah Penerapan Manajemen <i>Bandwidth</i> HTB Oleh 2 Client Secara Bersamaan	73
Gambar 4.28 Uji Coba Download Dengan 1 Client	74
Gambar 4.29 Uji Coba Download Dengan 2 Client	74
Gambar 4.30 Uji Coba Download Dengan 3 Client	75

DAFTAR LAMPIRAN

Lampiran 1 Hasil Pengukuran <i>Bandwidth</i>	86
Lampiran 2 Hasil Pengukuran <i>Throughput</i>	87
Lampiran 3 Hasil Pengukuran <i>Delay</i>	95
Lampiran 4 Hasil Pengukuran <i>Packet loss</i>	96
Lampiran 5 Konfigurasi Marking Connection Untuk 40 User	97
Lampiran 6 Konfigurasi Marking Packet Untuk 40 User	103
Lampiran 7 Konfigurasi Child Queue Traffic Download	107
Lampiran 8 Konfigurasi Child Queue Traffic Upload	111

BAB I

PENDAHULUAN

1.1. Latar Belakang

Jaringan komputer bukanlah sesuatu yang baru saat ini. Hampir setiap tempat pembelajaran seperti sekolah dan perguruan tinggi memiliki jaringan komputer. Dengan perkembangan teknologi jaringan yang semakin pesat, jumlah pengguna jaringan komputer yang bergabung dalam *internet* juga menjadi semakin berlipat ganda.

Hal tersebut dapat dilihat dari banyaknya pengguna jaringan komputer untuk mengakses internet baik itu secara umum maupun pribadi. Dengan semakin banyaknya kebutuhan akan akses internet maka kinerja jaringan harus berada pada kondisi yang baik, maka besarnya *bandwidth* yang dibutuhkan harus dapat memecahkan masalah utama yaitu menyediakan kinerja layanan yang baik untuk dapat memberikan kenyamanan kepada pengguna. Walaupun *bandwidth* yang dimiliki sudah cukup besar tetapi jika tidak dilakukan pengaturan maka akan mengakibatkan pemakaian antar *user* yang tidak seimbang, ada yang cukup cepat dan ada yang lambat bahkan tidak dapat di akses sama sekali.

Salah satu solusi yang efektif untuk mengatasinya adalah dengan melakukan manajemen *bandwidth* agar tidak terjadi kemacetan sebagai akibat dari adanya permintaan akses yang *overload* dalam jaringan.

SMK Negeri 22 Jakarta menggunakan internet sebagai penunjang sarana dan prasarana untuk seluruh masyarakat yang ada di SMK Negeri 22 Jakarta. SMK Negeri 22 Jakarta memiliki 4 program keahlian, antara lain: Akuntansi, Administrasi Perkantoran, Pemasaran, dan Teknik Komputer dan Jaringan. SMK Negeri 22 Jakarta juga memiliki warnet yang berada di lingkungan sekolah yang dapat digunakan oleh siswa maupun masyarakat sekitar.

SMK Negeri 22 Jakarta memanfaatkan internet untuk kegiatan belajar mengajar di LAB Komputer, jaringan pada CCTV, sistem absensi sidik jari, serta untuk menerapkan media pembelajaran online berbasis *web*, karena termasuk dalam salah satu visi sekolah yaitu “Menyelenggarakan pembelajaran berbasis ICT untuk semua kompetensi keahlian”.

Untuk terhubung ke jaringan internet, SMK Negeri 22 Jakarta menggunakan *routerboard* mikrotik tipe RB1100AHX2 yang digunakan untuk membantu kinerja manajemen jaringan tersebut. Saat ini SMK Negeri 22 Jakarta menggunakan manajemen *bandwidth* dengan menggunakan fitur *simple queue*, tetapi manajemen *bandwidth simple queue* dianggap kurang cocok untuk digunakan di SMK Negeri 22 Jakarta karena sering terjadi keluhan dari *user*. *User* mengeluh karena kesulitan untuk mengakses internet yang digunakan untuk menunjang kegiatan belajar mengajar. Selain itu *quality of service* (qos) yang ada di SMK Negeri 22 mendapatkan hasil yang kurang memuaskan jika dibandingkan dengan standarisasi TIPHON (*Telecommunications and Internet Protocol Harmonization Over Network*) berdasarkan skripsi Muhammad Akbar Rizky yang berjudul “ANALISIS KUALITAS LAYANAN (*QUALITY OF*

SERVICE) MIKROTIK ROUTERBOARD PADA JARINGAN INTERNET SMK NEGERI 22 JAKARTA”

Untuk mendapatkan hasil *Quality of service* (QoS), dilakukan perencanaan mengenai pengukuran parameter kualitas layanan jaringan yang terdiri dari *delay*, *throughput*, *bandwidth*, dan *packet loss* dari *node* pengirim ke *node* penerima atau dari ujung ke ujung (End to End) menggunakan alat bantu monitoring speedtest.net dan axence net tools pro 5.0. Speedtest.net digunakan untuk mengukur parameter kualitas layanan *bandwidth* dan axence net tools digunakan untuk mengatur kualitas layanan yang terdiri dari *throughput*, *delay* dan *packet loss*.

Sesudah dilakukan perencanaan, langkah selanjutnya adalah pengukuran data. Pengukuran data dilakukan di ruang guru. Sedangkan sebagai perangkat pengirim di buat menjadi 3 jenis web yaitu facebook.com sebagai perangkat hiburan cnn.com sebagai perangkat portal berita, dan terakhir mail.yahoo.com sebagai perangkat *mailing*.

Tabel 1.1 Hasil akhir analisis dari ruang guru

Web	<i>Average Throughput</i>	Persentase (%)	Predikat TIPHON	Nilai
cnn.com	397516	13.2504	Jelek	1
facebook.com	100606	3.3535	Jelek	1
mail.yahoo.com	296933	9.8976	Jelek	1
Web	<i>Average Packet loss</i>	Persentase (%)	Predikat TIPHON	Nilai
cnn.com	58	18	Sedang	2
facebook.com	58	19	Sedang	2
mail.yahoo.com	58	18	Sedang	2
Web	<i>Average Delay</i>		Predikat TIPHON	Nilai
cnn.com	71		Sangat Bagus	4
facebook.com	135		Sangat Bagus	4
mail.yahoo.com	108		Sangat Bagus	4
Hasil penjumlahan Nilai				21
Dibagi jumlah Parameter yang diukur (9)				$21:9=2.33$

Berdasarkan Tabel 1.1 dapat dilihat bahwa kualitas layanan yang ada pada jaringan SMK Negeri 22 Jakarta yang saat ini masih menggunakan manajemen

bandwidth simple queue mendapatkan predikat kurang memuaskan jika dibandingkan dengan standarisasi TIPHON.

Simple queue memiliki kelemahan untuk menangani *user* yang bersifat dinamis, yaitu *user* yang bisa *online* ataupun *offline* sesuai kemauan mereka. Selain itu berdasarkan paper yang berjudul “*Bandwidth Manajemen Queue Tree VS Simple Queue*” *simple queue* menghasilkan *delay* dan *packet loss* yang jauh lebih besar di bandingkan *queue tree*.

Oleh karena itu penulis ingin menerapkan manajemen *bandwidth queue tree* dengan menggunakan metode *Hierarchical Token Bucket* (HTB). Metode ini dipilih karena metode HTB 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 (Wijaya, 2013). Jadi apabila ada *user* yang tidak aktif, *bandwidth* dapat di pakai atau di pinjam oleh *user* yang sedang aktif. HTB sangat berguna untuk membatasi *rating download* dan *upload client*. Dengan demikian *client* tidak dapat seenaknya menggunakan semua kapasitas *bandwidth*.

Berdasarkan latar belakang diatas penelitian tugas akhir yang akan dibuat berjudul “**Analisis Manajemen *Bandwidth Hierarchical Token Bucket* (HTB) Dengan Mikrotik Pada Jaringan SMK Negeri 22 Jakarta**”.

1.2. Identifikasi Masalah

Dari uraian Latar Belakang Masalah, maka dapat diidentifikasi beberapa masalah dalam penelitian :

1. Adanya keluhan dari *user* karena kesulitan mengakses internet

2. Hasil *Quality of service* (QoS) yang buruk jika dibandingkan dengan standarisasi TIPHON

1.3. Pembatasan Masalah

Batasan masalah dibuat dengan tujuan agar pembahasannya dapat lebih terarah pada sasaran. Adapun batasan-batasan masalah dalam skripsi ini:

1. Manajemen *bandwidth* menggunakan metode HTB pada Mikrotik
2. Manajemen *bandwidth* menggunakan *Routerboard* Mikrotik tipe RB750Gr3
3. Manajemen *bandwidth* menggunakan metode HTB diterapkan untuk *hotspot* ruang guru
4. Pengukuran kualitas layanan dilakukan pada jaringan *hotspot* ruang guru

1.4. Rumusan Masalah

Berdasarkan latar belakang, maka dapat ditentukan rumusan masalah yaitu bagaimana pengaruh manajemen *bandwidth* dengan menggunakan *Routerboard* Mikrotik tipe RB750Gr3 terhadap kualitas layanan?

1.5. Tujuan Penelitian

Berdasarkan rumusan masalah, maka tujuan yang hendak dicapai dalam penelitian ini adalah untuk menerapkan manajemen *bandwidth* menggunakan metode HTB pada *Routerboard* Mikrotik tipe RB750Gr3 agar kualitas layanan QOS di SMK Negeri 22 menjadi lebih baik.

1.6. Manfaat Penelitian

1. Hasil penelitian ini diharapkan dapat membantu administrator jaringan di SMK Negeri 22 Jakarta untuk menerapkan manajemen *bandwidth* HTB.
2. Dapat mengoptimalkan penggunaan *bandwidth*.

3. Penelitian ini diharapkan dapat dijadikan bahan memperluas wawasan dalam bidang jaringan komputer bagi penulis.

BAB II

TINJAUAN PUSTAKA

2.1. Kerangka Teoritik

Kerangka teori diperlukan dalam setiap penelitian untuk memberikan landasan teoritis bagi peneliti dalam menyelesaikan masalah dalam proses penelitian, oleh karena itu peneliti perlu banyak literatur, baik berupa buku teks, majalah, jurnal, dan sebagainya. Kerangka teori juga membantu seorang penulis dalam menentukan tujuan dan arah penelitian, serta sebagai dasar penelitian agar langkah yang ditempuh selanjutnya dapat jelas dan konsisten (Koentjaraningrat,1990).

2.1.1. Konsep Dasar Jaringan Komputer

2.1.1.1. Jaringan Komputer

Jaringan komputer adalah sistem yang terdiri dari komputer-komputer, serta piranti-piranti yang saling terhubung sebagai satu kesatuan. Dengan dihubungkannya piranti-piranti tersebut, alhasil dapat saling berbagi sumber daya antar satu piranti dengan piranti lainnya (Wahana Komputer, 2010). Istilah jaringan komputer sendiri juga dapat diartikan sebagai kumpulan sejumlah terminal komunikasi yang terdiri dari dua komputer atau lebih yang saling terhubung. Jaringan komputer memungkinkan penggunaanya dapat melakukan komunikasi satu sama lain dengan mudah. Tujuan dibangunnya jaringan komputer adalah agar informasi/ data yang dibawa pengirim (transmitter) dapat sampai kepada penerima (receiver) dengan tepat dan akurat.

2.1.1.2. Jenis-jenis Jaringan Komputer

1. Berdasarkan luas area

Berdasarkan luas areanya jaringan komputer dibagi menjadi 3, yaitu (Madcoms, 2010):

a. LAN (*Local Area Network*)

LAN merupakan jaringan yang menghubungkan sejumlah komputer yang ada dalam suatu lokasi dengan area yang terbatas seperti ruang atau gedung. LAN dapat menggunakan media komunikasi seperti kabel dan *wireless*. Biasanya jarak antarnode tidak jauh dari sekitar 200 m.

b. MAN (*Metropolitan Area Network*)

MAN merupakan jaringan yang lebih besar dari jaringan LAN, misalnya antar gedung dalam suatu daerah (wilayah seperti provinsi atau negara bagian). Dalam hal ini menghubungkan beberapa buah jaringan kecil ke dalam ruang lingkup yang lebih besar. Sebagai contoh, jaringan beberapa kantor cabang sebuah bank di dalam kota besar yang dihubungkan antara satu dengan lainnya.

c. WAN (*Wide Area Network*)

WAN merupakan jaringan yang lebih besar dari jaringan LAN dan MAN.

WAN biasanya sudah menggunakan media *wireless*, sarana satelit, ataupun kabel *fiber optic*. Karena jangkauannya yang lebih luas, bukan hanya meliputi satu kota atau antar kota dalam suatu wilayah,

2. Berdasarkan media transmisi data yang digunakan

Berdasarkan media transmisi data yang digunakan jaringan komputer dibagi menjadi 2, yaitu (Wahana Komputer, 2010):

a. Jaringan kabel

Media transmisi data yang digunakan dalam jaringan ini berupa kabel. Kabel tersebut digunakan untuk menghubungkan satu komputer dengan komputer lainnya agar bisa saling bertukar informasi/ data atau terhubung dengan *internet*. Jaringan kabel memiliki kelebihan dalam aspek keamanan dan *transfer* data yang lebih cepat dibandingkan dengan jaringan nirkabel. Salah satu media transmisi yang digunakan dalam jaringan kabel adalah kabel UTP.

b. Jaringan Nirkabel

Berbeda dengan jaringan kabel (*wired network*), jaringan ini tidak menggunakan kabel untuk bertukar informasi/ data dengan komputer lain melainkan menggunakan gelombang elektromagnetik untuk mengirimkan sinyal informasi/ data antar komputer satu dengan komputer lainnya. Jaringan *wireless* lebih mudah dibuat serta lebih fleksibel dibandingkan jaringan kabel.

2.1.2. *Quality of Service (QoS)*

QoS merupakan kependekan dari *Quality of Service*. *Quality of Service* adalah suatu pengukuran tentang seberapa baik jaringan dan merupakan suatu usaha untuk mendefinisikan karakteristik dan sifat dari suatu layanan. Pada jaringan berbasis IP, IP QoS mengacu pada performansi dari paket – paket IP yang lewat melalui satu atau lebih jaringan.

Menurut Gunawan (2008) *Quality Of Service (QoS)* adalah kemampuan suatu elemen jaringan, seperti aplikasi jaringan, *host*, atau *router* untuk memiliki tingkatan jaminan bahwa elemen jaringan tersebut dapat memenuhi kebutuhan suatu layanan. Kualitas layanan jaringan dapat dibedakan menjadi tiga jenis, yaitu:

1. *Intrinsic QoS* merupakan kualitas layanan jaringan yang di dapat melalui :
 - a. Desain teknis jaringan yang menentukan karakteristik koneksi yang melalui jaringan.
 - b. Kondisi akses jaringan, terminasi, *link* antar *switch* yang menentukan suatu jaringan akan memiliki kapasitas yang memadai untuk menangani semua permintaan pengguna.

Dengan kata lain, *intrinsic QoS* tersebut dapat dideskripsikan dengan *parameter-parameter* kinerja suatu jaringan, seperti *latency*, *throughput*, dan lain-lain. *Intrinsic QoS* adalah yang digunakan pada penelitian ini.

2. *Perceived QoS* merupakan kualitas layanan jaringan yang diukur ketika suatu layanan digunakan. *Perceived QoS* sangat tergantung dari kualitas *intrinsic QoS* dan pengalaman pengguna pelayanan yang sejenis, namun *Perceived QoS* ini diukur dengan nilai *mean option score (MOS)* dari pengguna.
3. *Assessed QoS* merujuk kepada seberapa besar keinginan pengguna untuk terus menikmati suatu layanan tertentu. Hal ini berdampak pada keinginan pengguna untuk membayar jasa atas layanan yang dinikmatinya. *Assessed QoS* ini sangat tergantung dari *perceived QoS* masing-masing pengguna.

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. Tujuan dari QoS adalah untuk memuaskan kebutuhan – kebutuhan layanan yang berbeda, dengan infrastruktur yang sama. QoS menawarkan kemampuan untuk mendefinisikan atribut – atribut layanan jaringan yang disediakan, baik secara kualitatif maupun kuantitatif.

Dari segi *networking*, *Quality of Service* (QoS) mengacu pada kemampuan memberikan layanan yang berbeda kepada trafik jaringan dengan kelas – kelas yang berbeda. Tujuan akhir dari *Quality of Service* (QoS) adalah memberikan layanan jaringan yang lebih baik dan terencana dengan *dedicated bandwidth* dan *latency* yang terkontrol dan meningkatkan *loss* karakteristik.

QoS (*Quality of Service*) adalah teknologi yang diterapkan pada jaringan WAN (*Wide Area Network*) yang memungkinkan *administrator* jaringan untuk dapat menangani berbagai efek akibat terjadinya kemacetan (*congestion*) pada lalu lintas aliran paket di dalam jaringan. *Parameter* QoS adalah *delay/latency*, *jitter*, *packet loss*, *throughput*, *MOS*, *echo cancellation* dan *PDD*. QoS dibutuhkan untuk meminimalkan *packet loss*, *delay*, *latency* dan *delay variation* (jitter), menyakinkan *performance*, *mixing* paket data dan suara pada jaringan yang padat, dan dapat mengoptimalkan *queues* untuk memprioritaskan layanan misalnya *traffic voice*, *traffic shaping/buffering* pada jaringan WAN.

Tabel 2.1 Indeks Parameter QoS

Nilai	Persentase (%)	Indeks
3.8 - 4	95 – 100	Sangat Memuaskan
3 – 3.79	75 – 94.75	Memuaskan
2 – 2.99	50 – 74.75	Kurang Memuaskan
1 – 1.99	25 – 49.75	Jelek

(Sumber : TIPHON)

2.1.3. Parameter Kualitas Layanan

2.1.3.1. *Bandwidth*

Bandwidth adalah besaran yang menunjukkan seberapa banyak data yang dapat dilewatkan dalam koneksi melalui sebuah *network* (Anhar, 2016). Satuan yang dipakai untuk *bandwidth* adalah *bit per second (bps)* atau *Byte persecond (Bps)* dimana $1 \text{ Byte} = 8 \text{ bit}$. *Bit* atau *binary digit* adalah basis angka yang terdiri dari angka 0 dan 1. Satuan ini menggambarkan seberapa banyak *bit* (angka 0 dan 1) yang dapat mengalir dari suatu tempat ke tempat lain dalam setiap detiknya melalui suatu media

Dalam sebuah jaringan, *bandwidth* terbagi menjadi 2 yaitu *bandwidth digital* dan *analog* :

- *Bandwidth digital* adalah jumlah atau volume data yang dapat dikirimkan melalui sebuah saluran komunikasi dalam satuan bits per second tanpa distorsi.
- *Bandwidth analog* adalah perbedaan antara frekuensi terendah dengan frekuensi tertinggi dalam sebuah rentang frekuensi yang di ukur dalam

satuan Hertz (Hz) atau siklus per detik, yang menentukan berapa banyak informasi yang bisa ditransmisikan dalam suatu saat.

Pada dasarnya *bandwidth* mempresentasikan kapasitas koneksi, semakin tinggi kapasitas, maka umumnya akan diikuti oleh kinerja yang baik, meskipun kinerja keseluruhan juga tergantung pada faktor-faktor lain, misalnya *latency* yaitu waktu tunda antara masa sebuah perangkat itu memberikan izin untuk melakukan transmisi.

2.1.3.2. Throughput

Ternyata konsep *bandwidth* tidak cukup untuk menjelaskan kecepatan jaringan dan apa yang terjadi di jaringan. Untuk itulah konsep *Throughput* muncul. *Throughput* adalah *bandwidth* aktual yang terukur pada suatu ukuran waktu tertentu dalam satu hari menggunakan rute *internet* yang spesifik ketika sedang mendownload suatu *file*. Selang waktu pengukuran yang terbaik adalah 1 hingga 5 menit untuk menghasilkan grafik *throughput* harian atau mingguan. (Brownlee N, Loosley C, 2001)

Sedangkan *throughput* walau pun memiliki satuan dan rumus yang sama dengan *bandwidth*, tetapi *throughput* lebih pada menggambarkan *bandwidth* yang sebenarnya (aktual) pada suatu waktu tertentu dan pada kondisi dan jaringan *internet* tertentu yang digunakan untuk mendownload suatu *file* dengan ukuran tertentu.

Sayangnya, *throughput* karena banyak alasan, kadang sangat jauh dari *bandwidth* maksimum yang mungkin dari suatu media. Beberapa faktor yang menentukan *throughput* adalah:

a. Piranti jaringan

- b. Topologi jaringan
- c. Banyaknya pengguna jaringan
- d. Spesifikasi komputer *client/user*
- e. Spesifikasi komputer *server*
- f. Induksi listrik dan cuaca

Tabel 2.2 Standarisasi *Throughput*

Kategori <i>Throughput</i>	<i>Throughput</i>	Index
Sangat Bagus	100%	4
Bagus	75%	3
Sedang	50%	2
Jelek	<25%	1

(Sumber : TIPHON)

2.1.3.3. *Delay (latency)*

Delay adalah waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. *Delay* dapat dipengaruhi oleh jarak, media fisik, kongesti atau juga waktu proses yang lama. Menurut khamarullah (2009), *delay* dan *bandwidth* akan menunjukkan kapasitas dan kecepatan suatu jaringan. Terdapat beberapa komponen waktu yang mempengaruhi *delay*, yaitu:

- a. *Transport time / propagation delay* adalah waktu yang dibutuhkan oleh sebuah paket data untuk mengalir melalui *physical link*.
- b. *Queuing / transmission delay* adalah waktu yang dibutuhkan oleh sebuah paket data untuk melewati *router*
- c. *Server response time* adalah waktu yang dibutuhkan oleh *server* untuk memproses sebuah paket data yang datang dan menghasilkan sebuah paket data balasan.

Tabel 2.3 Standarisasi Delay

Kategori Delay	Delay	Index
Sangat Bagus	<150ms	4
Bagus	150 s/d 300ms	3
Sedang	300 s/d 450ms	2
Jelek	>450ms	1

(Sumber : TIPHON)

2.1.3.4. Packet loss

Packet loss merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang, dapat terjadi karena *collision* dan *congestion* pada jaringan dan hal ini berpengaruh pada semua aplikasi karena retransmisi akan mengurangi efisiensi jaringan secara keseluruhan meskipun jumlah *bandwidth* cukup tersedia untuk aplikasi aplikasi tersebut.

Tabel 2.4 Standarisasi Packet loss

Kategori Packet loss	Packet loss	Index
Sangat Bagus	0%	4
Bagus	3%	3
Sedang	15%	2
Jelek	25%	1

(Sumber : TIPHON)

2.1.4. Mikrotik

Router Mikrotik didesain dengan *system modular*, sehingga dimungkinkan untuk menambah *interface wireless* sesuai dengan kebutuhan, hingga sebanyak jumlah slot mini pci yang tersedia. *Processor* dan memori yang tersedia sebanding dengan kemampuan *routerboard* untuk mengalirkan koneksi data, baik sesuai dengan bps (*bit per second*) maupun pps (*packet per second*) nya.

Pada standar perangkat keras berbasis *personal computer* (pc), mikrotik dikenal dengan kestabilan, kualitas kontrol dan fleksibilitas untuk berbagai jenis paket data dan penanganan proses rute atau lebih dikenal dengan istilah *routing*. Sedangkan aplikasi yang dapat diterapkan di mikrotik selain *routing* adalah aplikasi kapasitas akses (*bandwidth*), manajemen, *firewall*, *wireless access point* (WiFi), *backhaul link*, *system hotspot*, *Virtual Private Network* (VPN) *server* dan masih banyak lainnya. (<http://www.mikrotik.com>)

2.1.4.1. Jenis-jenis Mikrotik

1. Mikrotik *RouterOS*

Mikrotik *RouterOS* merupakan sistem operasi yang diperuntukan sebagai *network router*. MikroTik *routerOS* sendiri adalah sistem operasi dan perangkat lunak yang dapat digunakan untuk menjadikan komputer biasa menjadi *router network* yang handal, mencakup berbagai fitur yang dibuat untuk *ip network* dan jaringan *wireless*. Fitur-fitur tersebut diantaranya: *Firewall & Nat*, *Routing*, *Hotspot*, *Point to Point Tunneling Protocol*, *DNS server*, *DHCP server*, *Hotspot*, dan masih banyak lagi fitur lainnya.

2. *Build in hardware* mikrotik

Build in hardware mikrotik merupakan MikroTik dalam bentuk perangkat keras yang khusus dikemas dalam *board router*, atau sering disebut *RouterBoard*, yang di dalamnya sudah terinstal sistem operasi MikroTik *RouterOS*. Didalam *RouterBoard* juga terdapat *Processor*, RAM, ROM dan *memory flash*. Pada *Routerboard* ini pengguna langsung

dapat memakainya, tanpa harus melakukan insatalasi sistem operasi. *RouterBoard* ini dikemas dalam beberapa bentuk dan kelengkapannya sendiri-sendiri. Ada yang difungsikan sebagai *Indoor Router*, *Outdoor Router* maupun ada yang dilengkapi dengan *wireless route*.

2.1.4.2. Fitur-fitur Mikrotik

Mikrotik memiliki banyak fitur yang bisa digunakan, berikut adalah beberapa fitur dari mikrotik (<http://www.mikrotik.com>) :

1. *Address List* : merupakan kumpulan kelompok IP *address* yang berdasarkan nama.
2. *Bridge* : mempunyai fungsi untuk *bridge spinning tree*, *multiple bridge interface* dan *bridging firewalling*.
3. *Asynchronous* : mempunyai dukungan untuk serial PPP *dial-in* atau *dial-out*, memiliki otentikasi CHAP, PAP, MSCHAPv1 dan MSCHAPv2, Radius, *dial on demand*, *modem pool* hingga 128 ports.
4. *Data rate management* : merupakan QoS berbasis HTB yang menggunakan *burst*, PCQ, RED, SFQ, FIFO *Queue*, CIR, MIR, *limit* antar *peer to peer*.
5. *Bonding* : mengkombinasikan beberapa *ethernet* ke dalam satu pipa pada koneksi yang sangat cepat.
6. *Proxy* : *cache* untuk FTP dan HTTP *proxy server*, HTTPS *proxy*; *transparent proxy* untuk DNS dan HTTP; mendukung protokol SOCKS; mendukung *parent proxy*; *static* DNS.
7. *Firewall* dan NAT : mendukung pemfilteran koneksi *peer to peer*, source NAT dan *destination* NAT. Mampu memfilter berdasarkan MAC, IP

address, range port, protokol IP, pemilihan opsi protokol seperti ICMP, *TCP Flags* dan *MSS*.

8. *Hotspot* : *hotspot gateway* dengan otentikasi RADIUS. Mendukung *limit data rate*, *SSL* ,*HTTPS*.
9. DHCP : mendukung DHCP tiap antarmuka; *DHCP Relay*; *DHCP Client*, *multiple network DHCP*; *static* and *dynamic DHCP leases*.
10. *Queue* : mengatur dan membatasi pemakaian *bandwidth internet*.

2.1.4.3. Cara Akses Mikrotik

Untuk mengakses mikrotik ada beberapa cara yang dapat di gunakan yaitu :

1. Putty (SSH)

Remote mikrotik menggunakan Putty biasanya dilakukan ketika akan mengakses *router* dengan *bandwidth* yang pas-pasan, karena kalau kita menggunakan winbox koneksi akan putus-putus, salah satu solusinya yaitu dengan *program SSH Client*.

2. Winbox

Mikrotik bisa juga diakses/remote menggunakan tool winbox (utility kecil di windows yang sangat praktis dan cukup mudah digunakan). Winbox merupakan tool untuk meremote Mikrotik yang paling populer karena selain mudah juga dapat menampilkan menu-menu pada Mikrotik secara GUI.

3. Telnet

Kita dapat me-remote MikroTik menggunakan telnet melalui *program* aplikasi "*command prompt*" (cmd) yang ada pada windows yang fitur

Telnet nya sudah diaktifkan. Namun, penggunaan telnet tidak dianjurkan dalam jaringan karena masalah keamanannya.

4. FTP

Remote menggunakan FTP biasanya dilakukan ketika kita akan melakukan *transfer file* dengan cara *upload* dari komputer lokal ke dalam mikrotik. Hal ini berguna ketika kita akan mengupload *login page hotspot* maupun *upload* paket mikrotik.

5. Web Browser

Mikrotik juga bisa diakses via *web/port 80* pada *browser* dengan cara memasukkan alamat IP mikrotik pada *address bar*.

2.1.5. Management Bandwidth

2.1.5.1. Pengertian Management Bandwidth

Manajemen berasal dari kata "to manage" yang berarti mengatur, mengurus atau mengelola, sedangkan *bandwidth* adalah besaran yang menunjukkan seberapa banyak data yang dapat dilewatkan dalam koneksi melalui sebuah network.

Berdasarkan definisi diatas maka manajemen *bandwidth* dapat diartikan sebagai suatu kegiatan mengatur agar data yang lewat tidak melebihi kapasitas maksimal di dalam sebuah jaringan komputer yang terhubung dengan internet.

Pembagian *bandwidth*:

1. Yang dibagi jalurnya: dengan pembagian kanal (fix/statis)
2. Yang diatur datanya: dengan metode antrian (dinamis-tergantung jumlah data yang lewat) terdiri dari:
 - a. prioritas (biasanya internet di perusahaan-perusahaan)

b. identitas (tidak ada persaingan, diterapkan di ISP)

c. kelas (perpaduan identitas dan prioritas): (biasanya internet di perusahaan)

Saat melewati sebuah jaringan, sebuah data bisa dipastikan akan mengalami penambahan ukuran karena setelah masuk ke suatu jaringan, TCP/IP atau UDP akan menambahkan aneka informasi ke dalam suatu file. Untuk mengetahui berapa ukuran sebenarnya dari suatu file yang lewat di jaringan maka ada yang disebut sebagai *throughput* yakni ukuran data yang sebenarnya (tanpa informasi lain yang ditambahkan oleh TCP/IP atau UDP).

Manajemen *bandwidth* adalah pengalokasian yang tepat dari suatu *bandwidth* untuk mendukung kebutuhan atau keperluan aplikasi atau suatu layanan jaringan. Pengalokasian *bandwidth* yang tepat dapat menjadi salah satu metode dalam memberikan jaminan kualitas suatu layanan jaringan QoS = *Quality Of Services*). (<http://overflow.web.id>, 2009).

Manajemen *bandwidth* adalah sebuah proses penentuan besarnya *bandwidth* kepada tiap pemakai dalam jaringan komputer. Besarnya *bandwidth* akan berdampak kepada kecepatan transmisi, *Bandwidth* internet disediakan oleh provider internet dengan jumlah tertentu tergantung sewa pelanggan. Dengan QoS dapat diatur agar *user* tidak menghabiskan *bandwidth* yang di sediakan oleh provider.

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. Manajemen *bandwidth* memberikan kemampuan

untuk mengatur *bandwidth* jaringan dan memberikan level layanan sesuai dengan kebutuhan dan prioritas sesuai dengan permintaan pelanggan.

Manajemen *bandwidth* dapat digambarkan dengan penggunaan selang atau pipa, yang mengalirkan air ke setiap pengguna, dan setiap pengguna tidak dapat sepuasnya mendapatkan air dari sumbernya (*internet*). Hal ini tergantung dengan *bandwidth* secara global. Pembatasan tersebut adalah nilai maksimal yang didapatkan oleh setiap pengguna dan sewaktu-waktu dapat menurun sesuai dengan keadaan dan kesibukkan pertukaran data. Berikut ini adalah beberapa fitur dari mekanisme kontrol *bandwidth* mikrotik Router OS:

1. Membatasi tingkat data untuk IP *address* tertentu, subnet, protokol, *port*.
2. Memprioritaskan beberapa arus paket.
3. Menggunakan antrian untuk mempercepat *web browsing*.
4. Menerapkan antrian pada interval-interval waktu yang pasti.
5. Berbagi lalu lintas yang tersedia diantara para pengguna secara adil, atau tergantung pada muatan saluran.

2.1.5.2. Jenis *Management Bandwidth*

Secara umum ada 2 jenis manajemen *bandwidth* pada mikrotik, yaitu :

1. *Simple Queue* merupakan jenis *bandwidth management* yang dapat dikonfigurasi secara mudah. Biasanya *simple queue* digunakan hanya untuk pembatasan pemakaian *bandwidth* (*upload* dan *download*) tiap *client*. Jenis *queue* ini sangat cocok di implementasikan pada jaringan skala kecil hingga menengah.

2. *Queue Tree* merupakan jenis *bandwidth management* yang sangat kompleks. Secara garis besar fungsinya sama dengan *simple queue* hanya pada *queue tree bandwidth management* akan di kelompokkan kedalam *group / parent* sehingga akan terlihat seperti hirarki. Untuk menggunakan *queue* ini kita harus mengaktifkan *mangle* di menu *firewall*, sehingga kita dapat melakukan limitasi atau memberikan prioritas pada lalu lintas apapun seperti *email*, *browsing*, *game* dll.

Berikut adalah perbedaan *Simple Queue* dengan *Queue Tree* :

Simple Queue

1. Memiliki aturan urutan yang sangat ketat, antrian diproses mulai dari yang paling atas sampai yang paling bawah.
2. Mengatur aliran paket secara *bidirectional* (dua arah).
3. Mampu membatasi trafik berdasarkan alamat IP.
4. Satu antrian mampu membatasi trafik dua arah sekaligus (*upload/download*).
5. Jika menggunakan *Queue Simple* dan *Queue Tree* secara bersama-sama, *Queue Simple* akan diproses lebih dulu dibandingkan *Queue Tree*.
6. Mendukung penggunaan PCQ sehingga mampu membagi *bandwidth* secara adil dan merata.
7. Bisa menerapkan antrian yang ditandai melalui paket di */firewall mangle*.
8. Mampu membagi *bandwidth* secara fixed.
9. Pengaturannya sangat sederhana dan cenderung statis, sangat cocok untuk admin yang tidak mau ribet dengan traffic control di */firewall mangle*.

Queue Tree

1. Tidak memiliki urutan, setiap antrian akan diproses secara bersama-sama.
2. Mengatur aliran paket secara *directional* (satu arah)
3. Membutuhkan pengaturan */firewall mangle* untuk membatasi trafik per IP.
4. Membutuhkan pengaturan */firewall mangle* terlebih dahulu untuk membedakan trafik *download* dan *upload*.
5. Di nomor duakan setelah *Queue Simple*.
6. Mendukung penggunaan PCQ sehingga mampu membagi *bandwidth* secara adil dan merata.
7. Pengaturan antrian murni melalui paket yang ditandai di */firewall mangle*.
8. Mampu membagi *bandwidth* secara *fixed*.
9. Lebih fleksibel dan butuh pemahaman yang baik di */firewall mangle* khususnya tentang *traffic control*.

2.1.6. Hierarchical Token Bucket (HTB)

Hierarchical Token Bucket (HTB) adalah metode pengelompokan *queue* atau antrian yang berguna untuk menangani berbagai jenis trafik. Implementasi QoS (*Quality of Services*) di Mikrotik banyak bergantung pada sistem antrian *Hierarchical Token Bucket* (HTB). Ada 2 macam *queue* pada Mikrotik yaitu *Simple Queue* dan *Queue Tree*. HTB memungkinkan kita membuat *queue* menjadi lebih terstruktur, dengan melakukan pengelompokan-pengelompokan bertingkat. Yang banyak tidak disadari adalah, jika tidak mengimplementasikan HTB pada *Queue* (baik *Simple Queue* maupun *Queue Tree*), ternyata ada beberapa *parameter* yang tidak bekerja seperti yang di inginkan. Beberapa *parameter* yang tidak bekerja adalah *priority*, dan dual *limitation* (CIR / MIR). CIR (*Comitted*

Information Rate) adalah batas bawah atau minimal trafik (*limit-at*) yang dapat diperoleh antrian. *Limit-at* membatasi minimal trafik suatu antrian, tidak peduli dalam kondisi apapun antrian tidak akan mendapat trafik di bawah batas ini. *MIR* (*Maximal Information Rate*) adalah batas atas atau maksimal trafik (*max-limit*) yang bisa diperoleh antrian. *Max-limit* membatasi maksimal trafik suatu antrian, dan setiap antrian akan mencapai batas ini jika *parent* masih memiliki cadangan *bandwidth*.

Hierarchical Token Bucket (HTB) adalah metode yang berfungsi untuk mengatur pembagian *bandwidth*, pembagian dilakukan secara hirarki yang dibagi-bagi kedalam kelas sehingga mempermudah pengaturan *bandwidth* dengan tepat sehingga penggunaannya menjadi maksimal. HTB diklaim menawarkan kemudahan pemakaian dengan teknik peminjaman dan implementasi pembagian trafik yang lebih akurat. Teknik antrian HTB memberikan fasilitas pembatasan trafik pada setiap level maupun klasifikasi, *bandwidth* yang tidak terpakai dapat digunakan oleh klasifikasi yang lebih rendah. HTB berperan dalam mengontrol penggunaan *bandwidth* terhadap *link* yang diberikan kepada klien. HTB memungkinkan penggunaan fisik *link single* untuk menampilkan *multiple link* dan untuk mengirimkan jenis *traffic* yang berbeda pada tampilan *link* yang berbeda. Dengan kata lain, HTB sangat berguna untuk membatasi rating *download* dan *upload client*. Dengan demikian *client* tidak dapat seenaknya menggunakan semua kapasitas *bandwidth*.

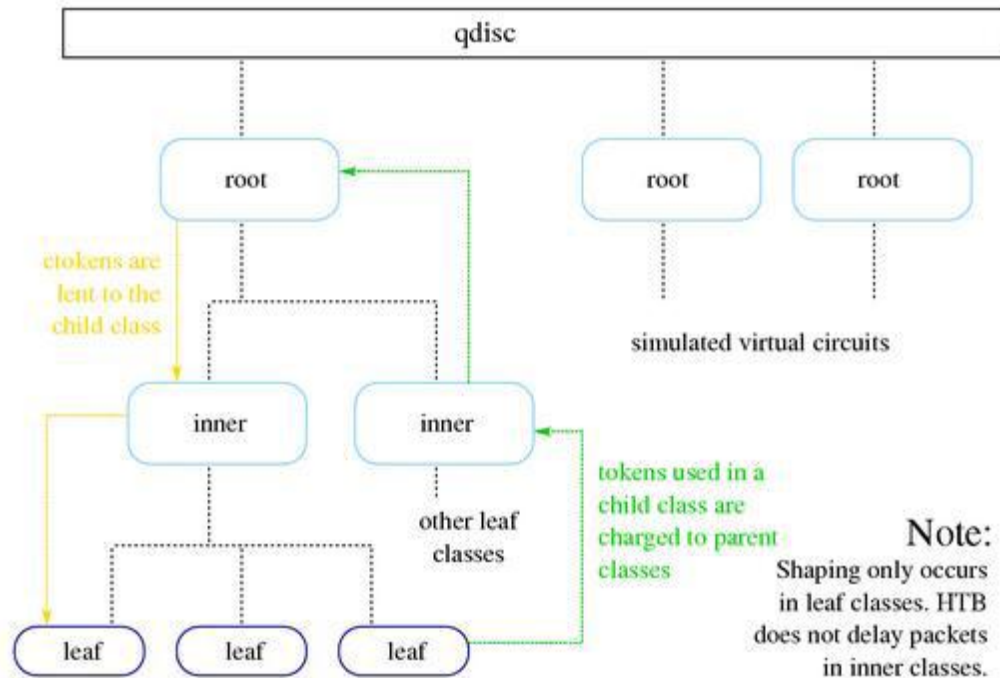
Ada tiga tipe kelas dalam HTB, yaitu : *root*, *inner*, dan *leaf*. *Root class* berada paling atas, dan semua trafik harus melewati kelas ini. *Inner Class* memiliki *parent class* dan *child class*. Sedangkan *leaf class* adalah terminal *class*

yang mempunyai *parent class* tetapi tidak mempunyai *child class*. Pada *leaf class*, trafik dari layer yang lebih tinggi diberikan melalui klasifikasi yang harus digunakan melalui *filter*, sehingga memungkinkan untuk membedakan jenis trafik dan prioritas. Sehingga sebelum trafik memasuki *leaf class* harus diklasifikasikan melalui *filter* dengan berbagai rules yang berbeda. HTB diambil dari algoritma *Token Bucket*. Analogi dari algoritma ini yaitu penetapan kapasitas ember (*Bucket*) pada pengambilan ke sekian kali (*Token*). Ketika pengecekan isi ember hampir penuh maka akan ditentukan kapan mengakhiri pengambilan. Itulah gambaran singkat dari algoritma *Token Bucket* yang selanjutnya digunakan pada *Hierarchical Token Bucket*. Pada konsep HTB disini yang merupakan *bucket* adalah paket data, selanjutnya akan dapat ditentukan berapa *token bucket* yang akan dijalankan. Dengan menjalankan HTB didapat sebuah sistem kontrol *bandwidth* yang akan digunakan oleh sebuah komposisi jaringan.

Teknik antrian HTB mirip dengan CBQ hanya perbedaannya terletak pada opsi, HTB lebih sedikit opsi saat konfigurasi serta lebih presisi. HTB dapat dilihat seperti suatu struktur organisasi dimana pada setiap bagian memiliki wewenang dan mampu membantu bagian lain yang memerlukan, teknik antrian HTB sangat cocok diterapkan pada perusahaan dengan banyak struktur organisasi. Secara konseptual, HTB adalah suatu jumlah yang berubah-ubah dari *token bucket* yang diatur didalam suatu hirarki.

Hierarchical Token Bucket (HTB)

Class structure and Borrowing



Gambar 2.1 HTB Sistem

Root qdisc akan berisi satu kelas (skenario kompleks bisa mempunyai berbagai kelas berkait dengan *root* qdisc). HTB kelas tunggal ini akan diset dengan dua *parameter*, suatu tingkat tarip dan suatu *ceil*. Nilai-Nilai ini harus merupakan yang sama untuk the top-level *class*, dan akan menghadirkan total *bandwidth* yang tersedia di *link* / jaringan. Di HTB, *rate* berarti *bandwidth* yang dijamin tersedia untuk kelas yang ditentukan dan *ceil* (*ceiling*) adalah untuk membatasi berapa banyak yang dapat di pinjam dari kelas lain.

2.1.6.1. *Parameter Hierarchical Token Bucket (HTB)*

HTB mempunyai *parameter* yang menyusunnya dalam antrian yaitu :

1. *Rate*

Parameter rate menentukan *bandwidth* maksimum yang bisa digunakan oleh setiap *class*, jika *bandwidth* melebihi nilai “*rate*”, maka paket data akan dipotong atau dijatuhkan (*drop*).

2. *Ceil*

Parameter ceil di-set untuk menentukan peminjaman *bandwidth* antar *class* (kelas), peminjaman *bandwidth* dilakukan kelas paling bawah ke kelas di atasnya. Teknik ini disebut *link sharing*.

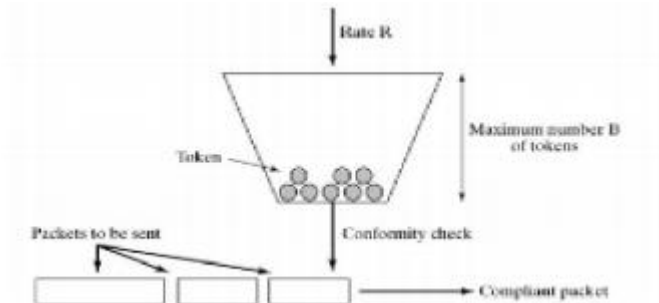
3. *Random Early Detection (RED)*

Random Early Detection atau bisa disebut *Random Early Drop* biasanya digunakan untuk *gateway/router backbone* dengan tingkat trafik yang sangat tinggi. RED mengendalikan trafik jaringan sehingga terhindar dari kemacetan pada saat trafik tinggi berdasarkan pemantauan perubahan nilai antrian minimum dan maksimum.

2.1.6.2. *Mekanisme Kerja Hierarchical Token Bucket (HTB)*

Hierarchical Token Bucket (HTB) merupakan salah satu teknik antrian yang memiliki tujuan untuk menerapkan *link sharing*. Dalam konsep *link sharing*, jika suatu kelas meminta kurang dari jumlah *service* yang telah ditetapkan untuknya, sisa *bandwidth* akan di distribusikan ke kelas – kelas lain yang meminta *service*. HTB menggunakan *Token Bucket Filter (TBF)* sebagai estimator yang sangat mudah diimplementasikan. Estimator ini hanya menggunakan *rate*, sebagai

akibatnya seorang *administrator* hanya perlu mengeset *rate* yang akan diberikan ke suatu kelas.



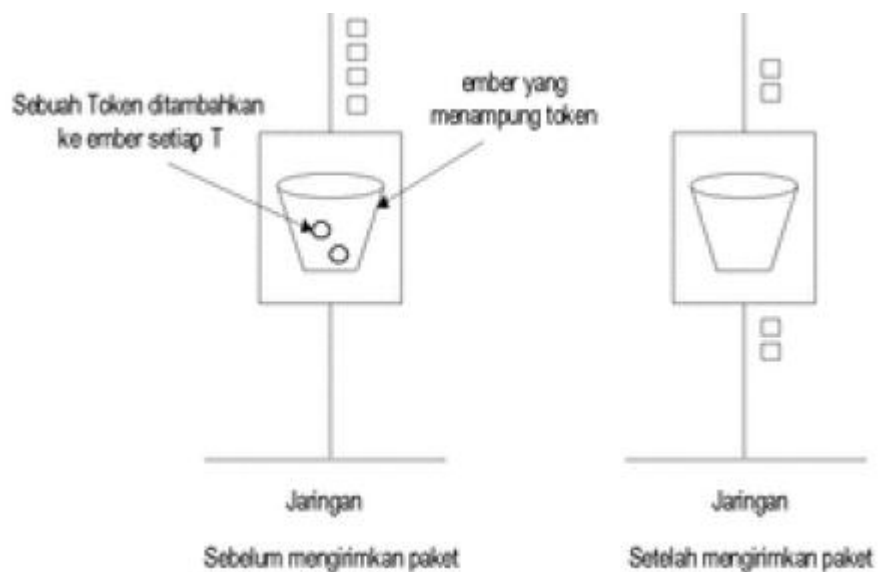
Gambar 2. 2 Skema Antrian HTB

Pada gambar 2.2 HTB memiliki *parameter ceil* sehingga kelas akan selalu mendapatkan *bandwidth* diantara *basrate* dan nilai *ceil ratenya*. *Parameter* ini dianggap sebagai estimator kedua, sehingga setiap kelas dapat meminjam *bandwidth* selama *bandwidth* total yang diperoleh memiliki nilai dibawah nilai *ceil*. Jadi dalam *system* antrian pada HTB jika salah satu *class* membutuhkan sejumlah *bandwidth* yang dibatasi maka sisa *bandwidth* akan didistribusikan ke kelas lainnya. Sebagai contoh, jika *bandwidth* yang ditentukan 512K sedangkan *bandwidth* yang tersedia adalah 1 MB maka sisa *bandwidth* akan didistribusikan ke kelas yang lainnya. Jika *bandwidth* yang didistribusikan masih tersisa dan tidak mendapatkan *class* maka hal yang terjadi adalah sisa *bandwidth* akan tersimpan pada *bandwidth parent*. Karena pada dasarnya mekanisme kerja dari HTB yaitu pembagian secara hirarki dengan *system* antrian.

2.1.6.3. Estimator pada HTB

Hierarchical Token Bucket (HTB) menggunakan *Token Bucket Filter* (TBF) sebagai *Estimator* untuk menentukan apakah suatu kelas/prioritas berada

dalam keadaan *underlimit*, *atlimit* atau *overlimit*. TBF bekerja dengan dasar algoritma ember *token*, setiap paket yang akan dikirimkan harus memiliki *token* yang berada dalam ember *token*, jika *token* tak tersedia didalam ember maka paket-paket yang akan dikirimkan harus menunggu sampai tersedia *token* yang cukup untuk mengirimkan paket yang sedang menunggu.

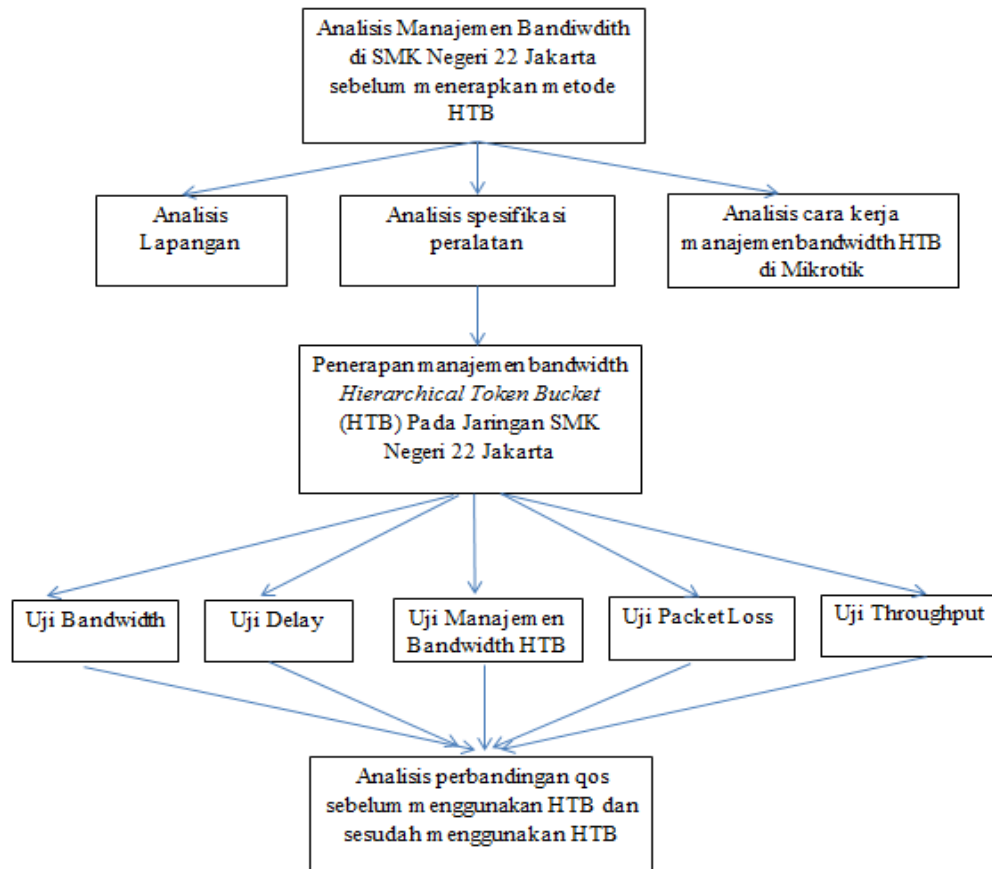


Gambar 2.3 Token Bucket Filter

Implementasi TBF terdiri dari sebuah *buffer* (*bucket*), yang secara konstan diisi oleh beberapa informasi *virtual* yang dinamakan *token*, pada *link* yang spesifik (*token link*). *Parameter* paling penting dari *bucket* adalah ukurannya, yaitu banyaknya *token* yang dapat disimpan. Setiap *token* yang masuk mengumpulkan satu paket yang datang dari antrian data dan kemudian dihapus dari *bucket*.

2.2 Kerangka Berpikir

Dari uraian kerangka teori sebelumnya dapat disusun suatu kerangka berpikir dalam menerapkan metode *Hierarchical Token Bucket* (HTB) pada jaringan SMK Negeri 22 Jakarta. Adapun kerangka berpikir dari penelitian ini dapat divisualisasikan sebagai berikut :



Gambar 2.4 Kerangka Berfikir

BAB III

METODOLOGI PENELITIAN

3.1. Tempat dan Waktu Penelitian

Penelitian dilakukan di SMK Negeri 22 Jakarta, Jalan Raya Condet, Pasar Rebo, Jakarta Timur. Penelitian dilakukan pada bulan Maret 2017 sampai dengan bulan Juni 2017. Skripsi ini ditulis sejak bulan Maret 2017.

3.2. Alat dan Bahan Penelitian

Peralatan serta bahan yang digunakan untuk menganalisa manajemen *bandwidth* HTB di SMK Negeri 22 Jakarta terdiri dari 2 kelompok, yaitu perangkat keras dan perangkat lunak dengan rincian sebagai berikut :

3.2.1. Alat Penelitian

3.2.1.1. Perangkat Keras

- Mikrotik Routerboard RB750Gr3

3.2.1.2. Perangkat Lunak

- Winbox
- Web browser Google Chrome
- *Internet Download Manager*
- Axence net tools pro 5.0

3.2.2. Bahan Penelitian

Bahan penelitian berupa informasi mengenai *bandwidth* yang tersedia di jaringan SMK Negeri 22 Jakarta, pembagian IP *address*, topologi jaringan, dan spesifikasi RB750Gr3.

3.2.2.1. Alokasi *Bandwidth*

Jaringan SMK Negeri 22 Jakarta menggunakan NusaNet sebagai ISP (Internet Service Provider) dan mempunyai *bandwidth* total 100MBPS untuk IIX (Indonesia Internet Exchange) dan 3MBPS untuk IX (Internasional Exchange) yang terhubung melalui jaringan metro fiber optic dari ISP dengan IP *Public* 182.48.191.0 dan subnet /24 (255.255.255.0). Lalu turun ke jaringan lokal SMK Negeri 22 dengan menggunakan mikrotik routerboard dengan tipe RB750Gr3

a. Jaringan LAN

SMK 22 Mempunyai 4 jaringan LAN yaitu;

1. LAN LAB A

Gateway	:	192.168.2.0
Subnet	:	/26 (255.255.255.192)
Konfigurasi IP	:	Statis
<i>Bandwidth</i>	:	12MBps.

2. LAN LAB B

Gateway	:	192.168.4.0
Subnet	:	/24 (255.255.255.0)
Konfigurasi IP	:	Statis
<i>Bandwidth</i>	:	12MBps

3. LAN Bengkel TKJ

Gateway	:	192.168.5.0
Subnet	:	/30 (255.255.255.252)

Konfigurasi IP : Statis
Bandwidth : 20MBps

4. LAN TU (Tata Usaha)

Gateway : 192.168.3.0
Subnet : /24 (255.255.255.0)
Konfigurasi IP : DHCP
Bandwidth : 2MBps

b. Jaringan Wireless

Jaringan wireless di SMKN 22 terbagi menjadi 6 *hotspot* yaitu sebagai berikut :

1. AP GURU (Bridge)

Gateway : 192.168.1.1
Subnet : /24 (255.255.255.0)
Konfigurasi IP : DHCP
Bandwidth : 3Mbps

2. AP Dapodikmen (bridge)

Gateway : 172.16.21.5
Subnet : /24 (255.255.255.0)
Konfigurasi IP : DHCP
Bandwidth : 512kbps

3. AP TU (tata usaha)

Gateway : 192.168.3.4
Subnet : /24 (255.255.255.0)
Konfigurasi IP : DHCP
Bandwidth : 2Mbps

4. SMK22 (bridge)

Gateway : 172.16.21.6
Subnet : /24 (255.255.255.0)
Konfigurasi IP : DHCP
Bandwidth : 512kbps

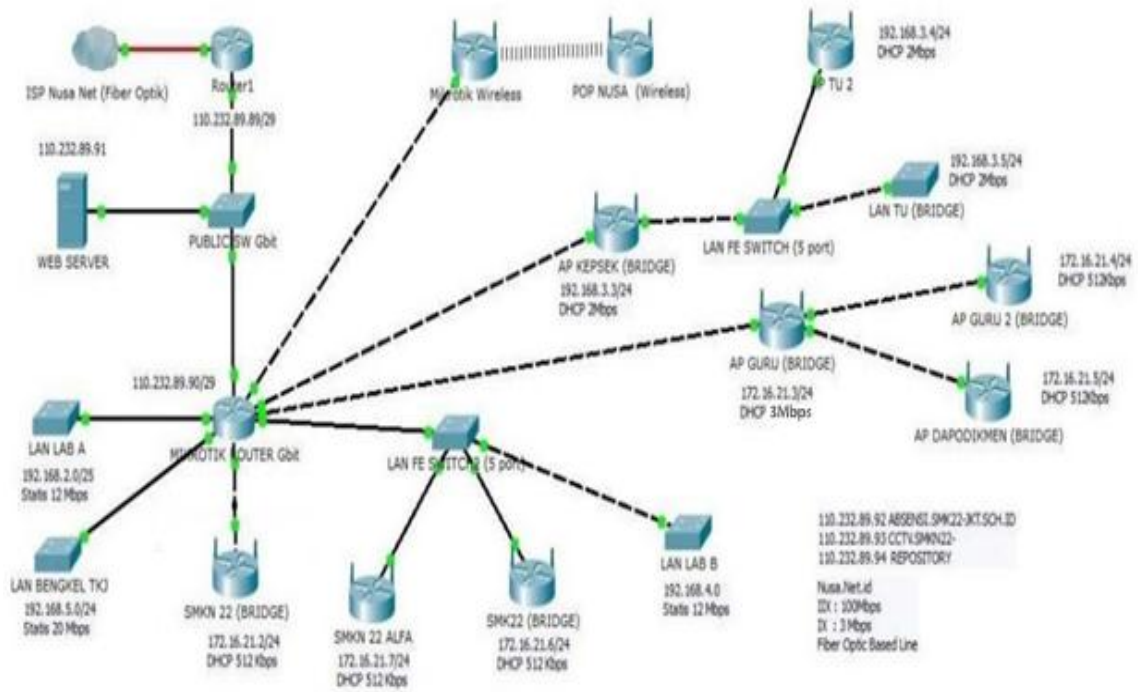
5. SMKN 22 (bridge)

Gateway : 172.16.21.2
Subnet : /24 (255.255.255.0)
Konfigurasi IP : DHCP
Bandwidth : 512kbps

6. SMKN 22 Bravo (bridge)

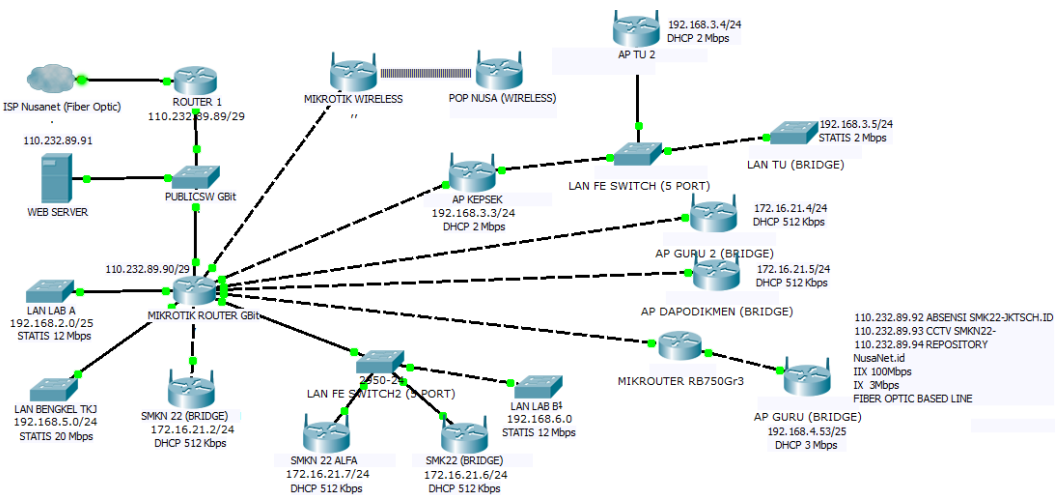
Gateway : 172.16.21.7
Subnet : /24 (255.255.255.0)
Konfigurasi IP : DHCP
Bandwidth : 512kbps

3.2.2.2. Topologi Jaringan



Gambar 3.1. Topologi Jaringan SMK Negeri 22 Jakarta Sebelum

Menerapkan Metode HTB



Gambar 3.2. Topologi Jaringan SMK Negeri 22 Jakarta Setelah

Menerapkan Metode HTB

Infrastruktur yang di gunakan pada topologi

1. Perangkat Keras

- a. Mikrotik Routerboard RB1100AHX2 yang berfungsi sebagai router, proxy *server*, *firewall* dan juga *bandwidth* limiter
- b. TP Link TL-SG1008 sebagai switch dari IP 192.168.x.x
- c. Cisco SF30024 sebagai switch dari IP 172.16.x.x
- d. Access Point Ubiquiti Picostation M2HP sebagai jaringan Hotspot (wireless)

2. Perangkat Lunak

- a. System operasi Windows 7
- b. Router os sebagai system operasi dari Mikrotik Routerboard
- c. Winbox sebagai alat konfigurasi dan remote dari Router Mikrotik
- d. Google Chrome sebagai web browsing
- e. Axence netTools Pro 5 sebagai alat untuk menganalisa paket jaringan

3.2.2.3 Spesifikasi Roterboard Mikrotik

Mikrotik router yang digunakan pada SMK 22 jakarta adalah tipe RB1100AHX2. Berikut adalah spesifikasi dari RB1100AHX2 :

Tabel 3.1 Spesifikasi RB1100AHX2

Product Code	RB1100AHX2
Architecture	PPC
CPU	Freescale P2020 1066MHz Dual Core
Current Monitor	No
Main Storage/NAND	64MB
RAM	1.5GB
SFP Ports	0
LAN Ports	13
Gigabit	Yes
Switch Chip	2
MiniPCI	0
Integrated Wireless	No

MiniPCIe	0
SIM Card Slots	No
USB	No
Memory Cards	1
Memory Card Type	MicroSD
Power Jack	110/220V
802.3af Support	No
POE Input	10-24VDC
POE Output	No
Serial Port	DB9/RS232
Voltage Monitor	Yes
Temperature Sensor	Yes
Dimensions	1U case: 45x75x440mm
Operating System	RouterOS
Temperature Range	-20C .. +45C
RouterOS License	Level6

Routerboard ini ditempatkan di ruang *server* (ICT) Lantai 2 SMK Negeri 22 Jakarta dengan fungsi sebagai gateway untuk sambungan ke jaringan *public*. Untuk selanjutnya mikrotik akan melakukan manajemen dan melakukan proses *routing* dan memberikan IP DHCP kepada *client/user*.

Fitur digunakan adalah manajemen *bandwidth* dengan jenis *simple queue* dengan *bandwidth* yang dibagi seperti yang dijelaskan pada Gambar 3.1. *transparent proxy* digunakan untuk seluruh jaringan di SMK Negeri 22 Jakarta. Karena menggunakan fitur *transparent proxy* maka tidak perlu dilakukan pengaturan ip *proxy* pada *browser*, jadi koneksi dari klien akan dipaksa masuk melalui *proxy server* secara otomatis dengan cara mengarahkan trafik http menuju ke *web proxy*.

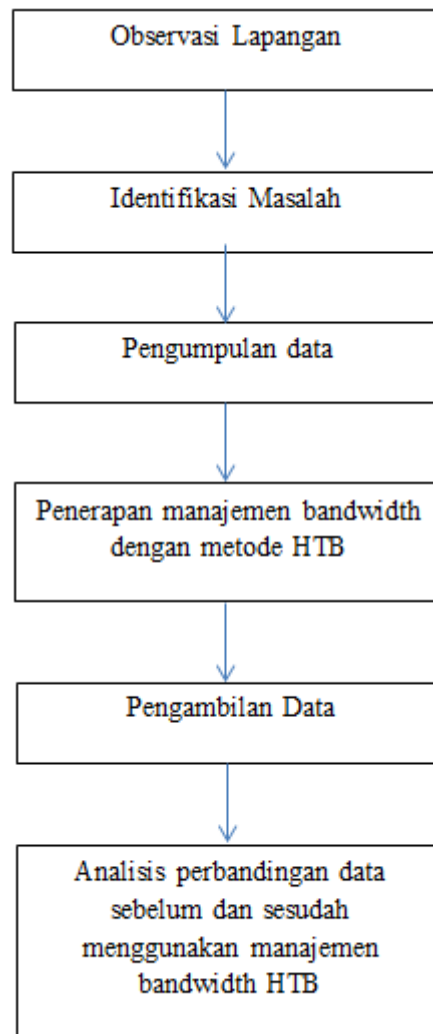
Mikrotik router yang digunakan untuk penelitian ini adalah tipe RB750Gr3. Berikut adalah spesifikasi dari RB750Gr3 :

Tabel 3.2 Spesifikasi RB750Gr3

Product Code	RB750Gr3
Architecture	MMIPS
CPU	MT7621A 2 Core 4 thr 880Mhz
Current Monitor	No
Main Storage/NAND	16MB
RAM	256MB
SFP Ports	0
LAN Ports	5
Gigabit	Yes
Switch Chip	1
MiniPCI	0
Integrated Wireless	No
MiniPCIe	0
SIM Card Slots	No
USB	Yes
Power on USB	Yes
Memory Cards	1
Memory Card Type	MicroSD
Power Jack	8-30V
802.3af Support	No
POE Input	8-30V
POE Output	No
Serial Port	No
Voltage Monitor	Yes
Temperature Sensor	Yes
Dimentions	113x89x28mm
Operating System	RouterOS
Temperature Range	-30C .. +70C
RouterOS License	Level4

3.3. Diagram Alir Penelitian

Peneliti akan menjelaskan langkah-langkah yang akan dilakukan untuk menyelesaikan penelitian ini dalam menganalisis manajemen *Bandwidth* menggunakan metode *hierarchical token bucket* (HTB) pada jaringan SMK Negeri 22 Jakarta dengan menggunakan metode penelitian rekayasa teknik. Ada 6 langkah untuk menyelesaikan penelitian ini, seperti pada gambar berikut :



Gambar 3.3. Diagram Alir Penelitian

3.3.1. Observasi Lapangan

Langkah awal yang akan dilakukan pada penelitian ini adalah melakukan observasi lapangan. Tujuan dari dilakukannya observasi adalah agar dapat memperoleh gambaran yang jelas tentang masalah yang akan dijadikan bahan penelitian.

3.3.2. Identifikasi Masalah

Identifikasi masalah dilakukan untuk mengetahui penyebab masalah dan mencari solusi yang tepat untuk menyelesaikan permasalahan tersebut.

3.3.3. Pengumpulan Data

Langkah selanjutnya yaitu pengumpulan data. Pengumpulan data dilakukan dengan menggunakan teknik wawancara. Data yang digunakan untuk penelitian berupa topologi jaringan di SMK Negeri 22 Jakarta, alokasi Bandwidth, dan spesifikasi peralatan yang digunakan di SMK Negeri 22 Jakarta. Langkah selanjutnya adalah menerapkan manajemen bandwidth dengan metode HTB.

3.3.4. Penerapan manajemen *bandwidth* dengan metode HTB

Setelah melakukan observasi lapangan, identifikasi masalah, dan pengumpulan data langkah selanjutnya adalah menerapkan manajemen bandwidth dengan metode HTB. HTB (Hierarchical Token Bucket) adalah metode pengelompokan queue atau antrian yang berguna untuk menangani berbagai jenis trafik. Hierarchical Token Bucket (HTB) berfungsi untuk mengatur pembagian bandwidth, pembagian dilakukan secara hirarki yang dibagi-bagi kedalam kelas sehingga mempermudah pengaturan bandwidth dengan tepat sehingga penggunaannya menjadi maksimal.

3.3.5. Pengambilan Data

Setelah menerapkan manajemen bandwidth dengan metode HTB (Hierarchical Token Bucket), langkah selanjutnya yaitu melakukan pengambilan data yang dibutuhkan untuk melakukan analisis perbandingan sebelum dan sesudah menggunakan metode HTB (Hierarchical Token Bucket). Data yang diambil adalah data dari uji bandwidth, uji throughput, uji delay dan uji packet loss.

3.3.6. Analisis Perbandingan Data Sebelum dan Sesudah Menggunakan Manajemen *Bandwidth Hierarchical Token Bucket* (HTB)

Setelah mendapatkan hasil yang dibutuhkan yang berupa data dari uji bandwidth, uji throughput, uji delay dan uji packet loss, pada langkah ini adalah membandingkan hasil dari data sebelum menggunakan *hierarchical token bucket* (HTB) dan setelah menggunakan *hierarchical token bucket* (HTB). Setelah membandingkan data sebelum dan sesudah menggunakan *hierarchical token bucket* (HTB).

3.4. Teknik dan Prosedur Pengumpulan Data

Teknik pengumpulan data yang digunakan dalam penulisan skripsi ini adalah :

1. Observasi atau pengamatan langsung, Pada tahap ini peneliti melakukan pengajuan berkas-berkas penelitian pada lokasi penelitian, dan melakukan pengamatan langsung atau peninjauan secara cermat dan langsung di lapangan atau lokasi penelitian.
2. Wawancara, dilakukan selama pelaksanaan penelitian dilakukan di SMK Negeri 22 Jakarta. Wawancara ini dilakukan kepada guru sekaligus kajar TKJ di SMK Negeri 22 Jakarta.
3. Studi pustaka, dilakukan dengan mencari berbagai buku referensi yang berkaitan dengan laporan skripsi. Tidak hanya buku, penulis juga mencari dari berbagai media cetak lainnya seperti jurnal serta skripsi dari berbagai mahasiswa yang telah lulus. Pencarian informasi ini dilakukan di beberapa tempat, seperti perpustakaan Universitas Negeri Jakarta dan beberapa toko buku.

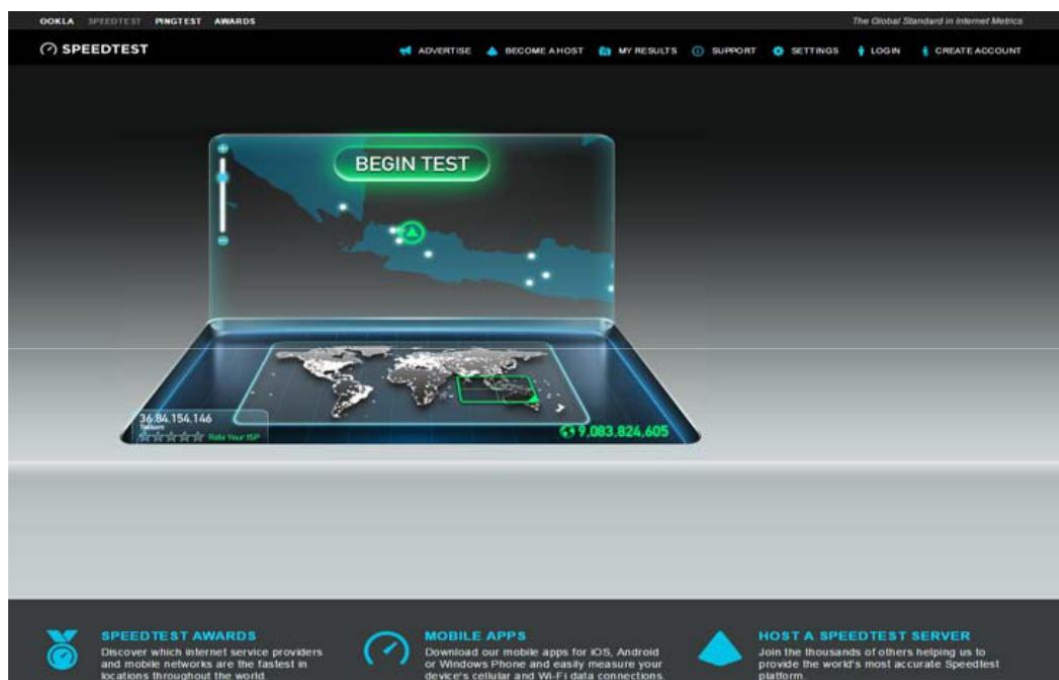
3.5. Teknik Analisis Data

Pada tahap ini dilakukan perbandingan data sebelum dan sesudah menggunakan HTB terhadap parameter kualitas layanan yang terdiri dari *bandwidth*, *throughput*, *delay* dan *packet loss* pada proses transmisi data dari pengirim ke perangkat penerima (*destination*) menggunakan *software monitoring* speedtest.net dan axence net tools pro 5.0.

Pengukuran dilakukan pada jaringan *hotspot* ruang guru sebagai penerima paket data (*destination*). Penelitian dilakukan dengan interval waktu dari jam 08.00 sampai 15.00 (waktu jam sekolah)

3.5.1. Pengukuran *Bandwidth*

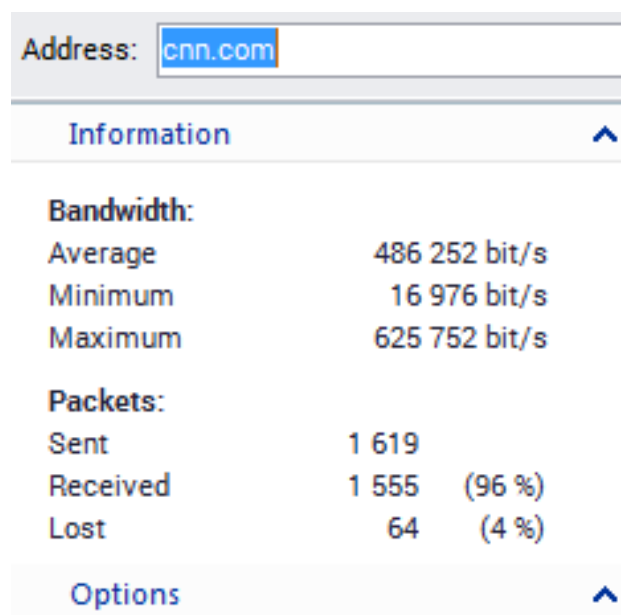
Pengukuran *bandwidth* dilakukan menggunakan *bandwidth monitor* speedtest.net dengan menggunakan *browser*. Satuan dapat diubah sesuai kebutuhan. Namun yang digunakan pada penelitian ini adalah satuan kilobits.



Gambar 3.4. User Interface Speedtest.net

3.5.2. Pengukuran *Throughput*

Pengukuran *throughput* dilakukan dengan mengirim sejumlah packet ke server tujuan melalui perangkat jaringan SMK Negeri 22 Jakarta. *Throughput* pada pengukuran ini diartikan sebagai banyaknya paket yang sampai ke penerima dalam kurun waktu yang telah ditentukan. Langkah yang dilakukan untuk mendapatkan hasil pengukuran *throughput* menggunakan *software axence net tools pro 5.0*.



The screenshot shows the 'Information' section of the Axence Net Tools interface. At the top, the 'Address' field is set to 'cnn.com'. Below this, the 'Information' section is expanded to show the following data:

Bandwidth:		
Average	486 252 bit/s	
Minimum	16 976 bit/s	
Maximum	625 752 bit/s	
Packets:		
Sent	1 619	
Received	1 555	(96 %)
Lost	64	(4 %)

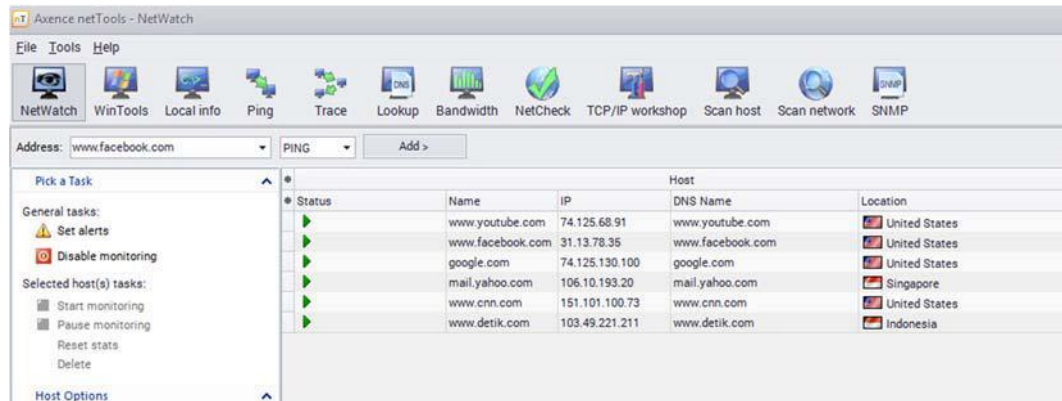
At the bottom of the screenshot, the 'Options' section is also visible and expanded.

Gambar 3.5. Hasil Pengukuran Throughput

3.5.3 Pengukuran *Delay*

Delay pada pengukuran ini adalah waktu tunda pada pengiriman unit data dari node pengirim ke perangkat penelitian. Nilai *delay* diukur menggunakan *software axence net tools* dengan mengaktifkan fitur *netwatch*.

1. Fitur Netwatch



Gambar 3.6. Layanan Pada Fitur Netwatch

2. Statistik Hasil *Delay*

Sent time	Response time (ms)
12:21:33	57
12:21:32	57
12:21:31	88
12:21:30	57
12:21:29	57
12:21:28	55
12:21:27	55
12:21:26	55
12:21:25	64
12:21:24	57
12:21:23	55
12:21:22	55

Gambar 3.7. Hasil Pengukuran Delay

3.5.4. Pengukuran *Packet loss*

Packet loss pada pengukuran ini merupakan persentase banyaknya paket data yang hilang pada proses pengiriman data dari node pengirim ke node penerima. Nilai *packet loss* diukur menggunakan *software* axence net tools dengan mengaktifkan fitur netwatch.

Packets		
Sent	Lost	% Lost
1001	1	0
1001	2	0
1000	1	0
1001	2	0

Gambar 3.8. Hasil Pengukuran Packet loss

BAB IV

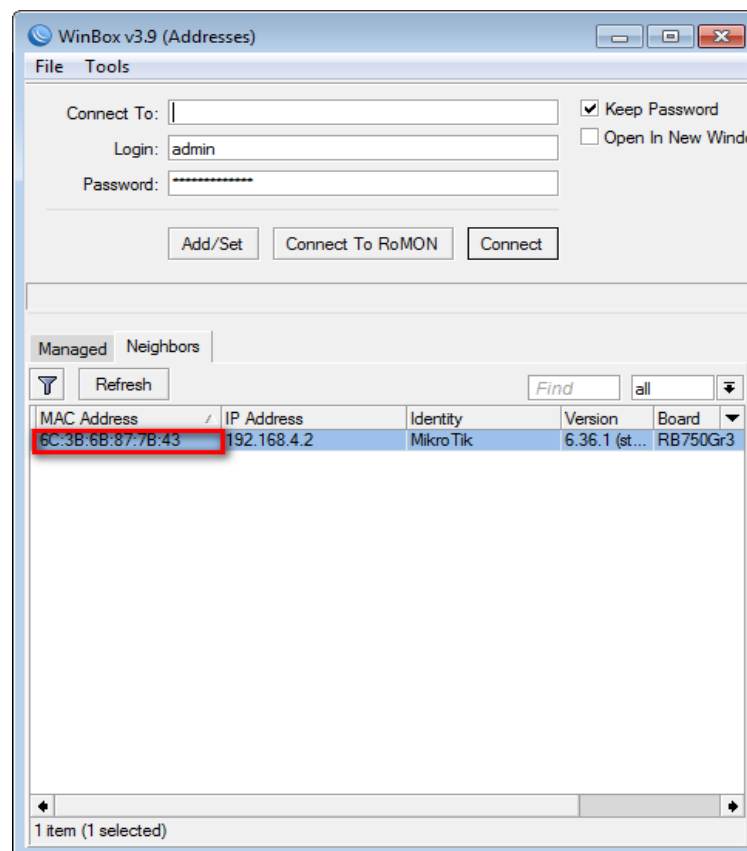
HASIL PENELITIAN

4.1. Deskripsi Hasil Penelitian

4.1.1. Konfigurasi Router Mikrotik

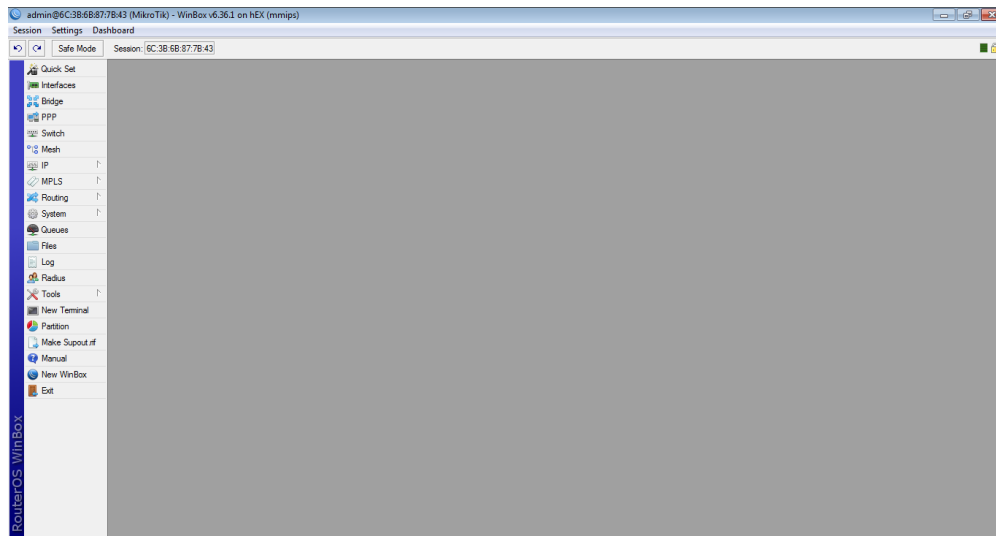
Karena konfigurasi dilakukan menggunakan RB750Gr3 maka tidak perlu melakukan installasi lagi. Dalam sistem ini pengaksesan RB750Gr3 akan menggunakan winbox karena mudah dipahami dan digunakan. Adapun cara pengaksesannya melalui winbox adalah sebagai berikut :

1. Buka aplikasi Winbox, Klik *MAC Address* yang tampil karena mikrotik belum mempunyai *IP address*, kemudian klik *connect* untuk terkoneksi ke mikrotik.



Gambar 4.1. Tampilan Awal Winbox

2. Winbox akan melakukan koneksi ke mikrotik



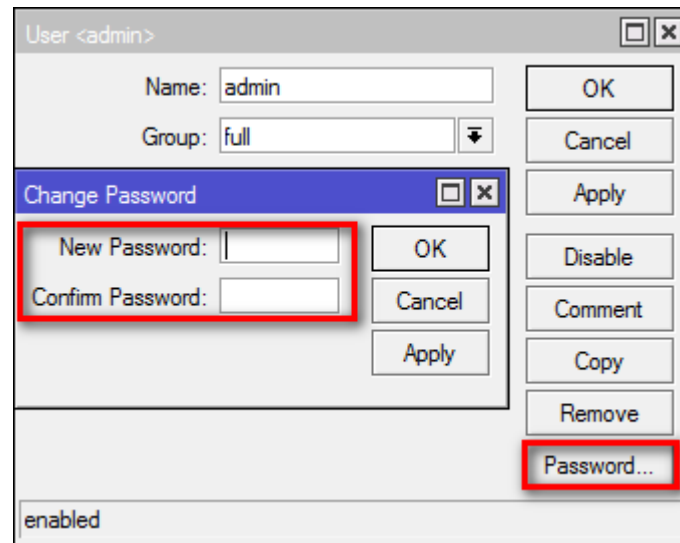
Gambar 4.2. Tampilan Awal Mikrotik Pada Winbox

Setelah terkoneksi ke mikrotik langkah selanjutnya adalah melakukan konfigurasi *router* mikrotik, cara-caranya adalah sebagai berikut :

1. Mengubah *password* admin

Langkah ini sebenarnya tidak wajib dilakukan tetapi merubah *password* admin dilakukan untuk alasan keamanan untuk menghindari perbuatan orang yang tidak bertanggung jawab. Langkah yang dilakukan adalah

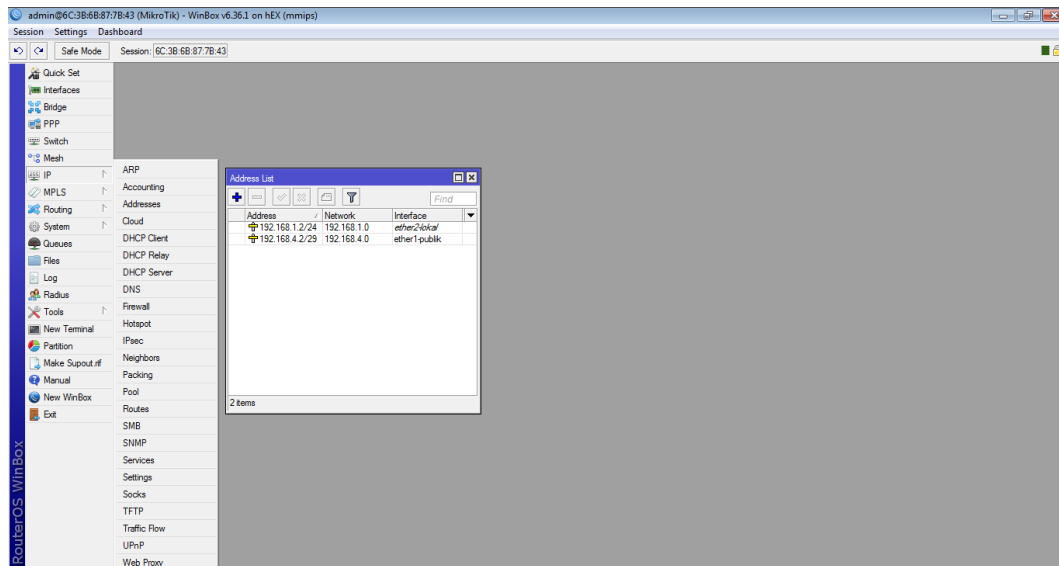
- a. klik system → *user*
- b. klik pada system default *user*
- c. klik *password*, kemudian isi *password*



Gambar 4.3. Merubah *Password* Mikrotik

2. Menambahkan IP Address

Menambahkan IP Address dengan cara klik IP→Addresses →klik tanda (+) isi *address, network* dan pilih interface.



Gambar 4.4. Konfigurasi IP Address

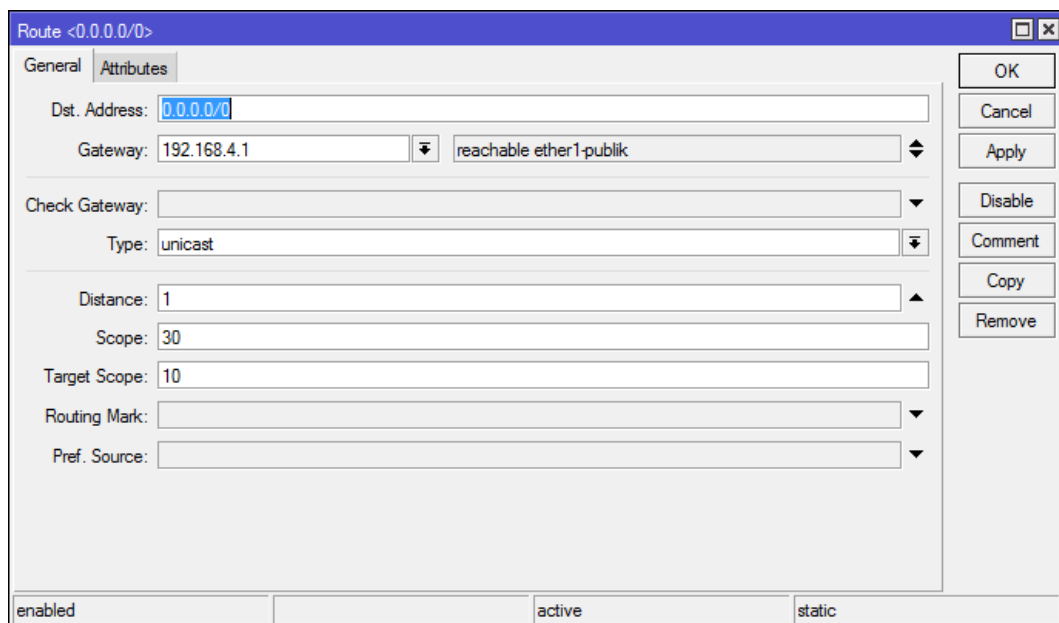
Disini IP yang digunakan untuk interface adalah :

Tabel 4.1. IP Address

Interface	Address	Network
Ether1-publik	192.168.4.2/29	192.168.4.0
Ether2-lokal	192.168.1.1/24	192.168.1.0

3. Memasukkan *Gateway*

Buka menu IP → Routes kemudian klik tanda (+) kemudian isi *gateway* dari modem yaitu 192.168.4.1 . Fungsi *gateway* adalah untuk menjadikan mikrotik sebagai jalan keluar untuk mengakses internet. Jadi setiap komputer yang akan mengakses internet akan melalui *router* mikrotik tersebut.

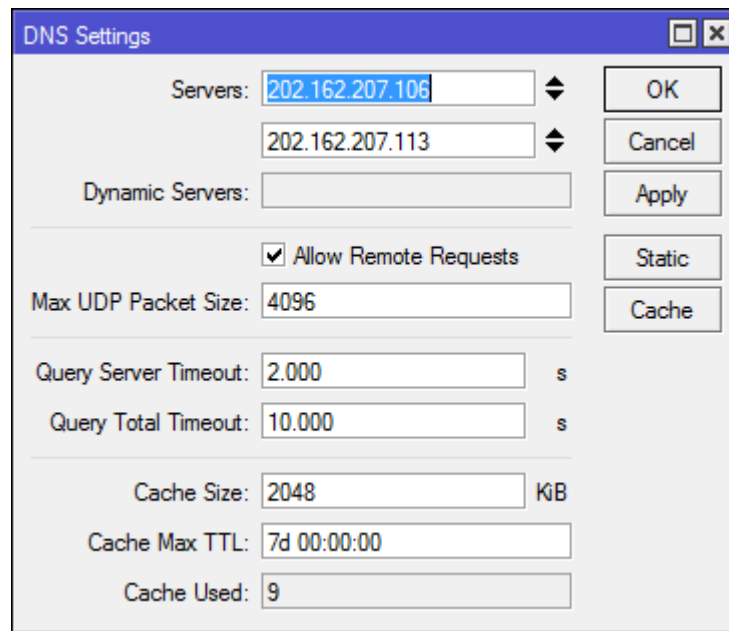


Gambar 4.5 Konfigurasi *Gateway*

4. Konfigurasi DNS

Buka menu IP → DNS isi DNS dengan IP 202.162.207.106 dan 202.162.207.113. Agar semua computer *client* dalam LAN bisa lebih cepat dalam

melakukan *browsing* internet, maka *Router* Mikrotik ini perlu diset agar bisa melakukan penyimpanan cache DNS. Sehingga setiap kali computer *client browsing* tidak perlu mengambil informasi dari *server* ISP, namun langsung mengambil informasi yang ada dalam *server* DNS *Router* Mikrotik ini.



Gambar 4.6. Konfigurasi DNS

Konfigurasi DNS berfungsi untuk mengarahkan paket yang menuju *router* untuk diterjemahkan menuju DNS (Domain Name System) primer dan sekunder yang berada di ruang ICT SMK Negeri 22 Jakarta. Perintah `allow-remote-request = yes` digunakan supaya *client* bisa menggunakan alamat ip gateway local untuk digunakan sebagai DNS yang nantinya akan tetap diarahkan menuju alamat DNS primer atau sekunder.

5. Konfigurasi Masquerade

Buka menu IP → *Firewall* → pilih NAT → klik tanda (+), pada tab general pilih chain = srcnat, out. interface = ether1-publik, kemudian klik tab action pilih action = masquerade

The screenshot shows the Mikrotik WinBox Firewall configuration window. The 'Filter Rules' tab is active, and a rule named 'mas...' is selected. The rule is configured as follows:

#	Action	Chain	Src. Address	Dst. Address	Proto...	Src. Port	Dst. Port	In. Inter...	Out. Int...	Bytes	Packets
0	mas...	srcnat							ether1-...	97.8 KB	697

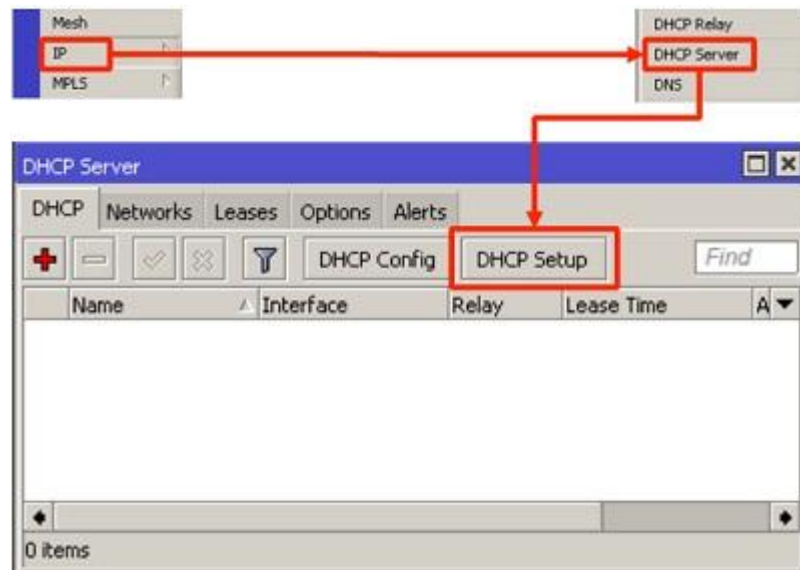
The status bar at the bottom indicates '1 item (1 selected)'.

Gambar 4.8. Hasil Konfigurasi Masquerade

Masquerade adalah salah satu bentuk *Network Address Translation* yang memungkinkan multi-host dalam sebuah jaringan *private* untuk dapat terkoneksi ke internet melalui sebuah ip public tunggal, masquerade berfungsi untuk mentranslasikan ip *address* dan port dalam local area secara *realtime*, sehingga dengan adanya NAT setiap komputer pada jaringan lokal dapat mengakses internet. Sampai tahap ini komputer klien sudah bisa terhubung ke internet.

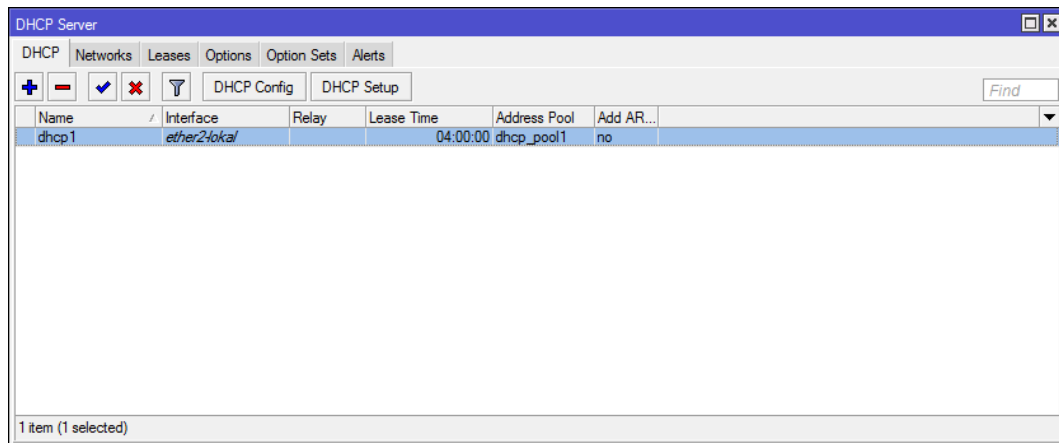
6. Konfigurasi DHCP Server

DHCP *Server* akan sangat tepat diterapkan pada jaringan yang memiliki *user* yang sifatnya dinamis, dengan jumlah *client* yang tidak tetap dan selalu berubah. Pada kasus ini terjadi pada *hotspot* di ruang guru. Konfigurasi DHCP *Server* dapat dilakukan pada menu IP → DHCP *Server* → Klik DHCP Setup



Gambar 4.9. Konfigurasi DHCP Server

Dengan menekan tombol DHCP Setup, wizard DHCP akan menuntun untuk melakukan setting dengan menampilkan kotak-kotak dialog pada setiap langkah nya. Langkah pertama adalah pilih pada interface mana DHCP Server akan aktif. Pada kasus ini DHCP Server diaktifkan pada ether2-lokal. Selanjutnya tentukan DHCP Address Space, karena IP address yang digunakan adalah 192.168.1.1/24 maka secara otomatis Wizard akan menawarkan DHCP Address Space : 192.168.1.0/24. Setelah itu tentukan IP gateway untuk DHCP ini, IP gateway adalah IP Address dari interface yang menjembatani antara jaringan LAN dan Mikrotik, disini kita menggunakan IP 192.168.1.1. Kemudian tentukan DHCP IP Address Range, karena jumlah user di ruang guru ada 40 user maka IP Address Range yang di buat adalah 192.168.1.2 – 192.168.1.41. Selanjutnya tentukan IP Address DNS Server, IP DNS yang digunakan disini adalah 202.162.207.106 dan 202.162.207.113. Langkah terakhir tentukan Lease Time, lease time adalah lama waktu yang diberikan kepada Client untuk menggunakan IP Address otomatis dari DHCP Server Mikrotik.



Gambar 4.10. Hasil Konfigurasi DHCP Server

4.1.2. Konfigurasi Manajemen *Bandwidth* HTB

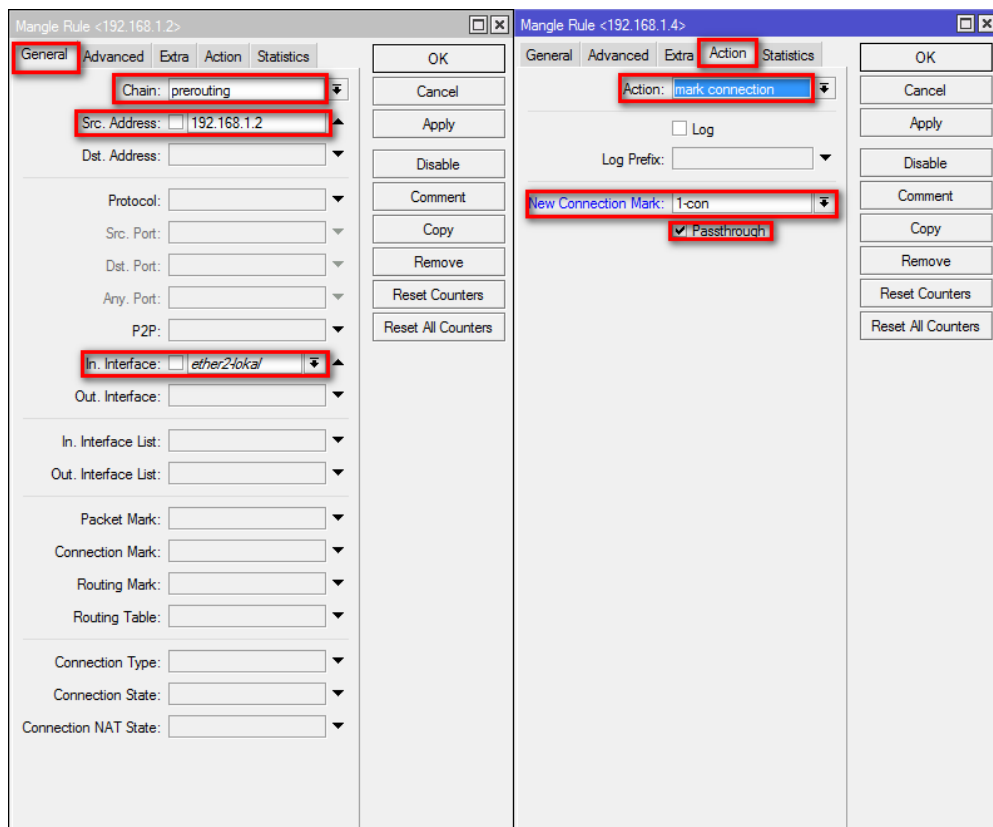
4.1.2.1. Konfigurasi *Mangle*

Setelah mengkonfigurasi *router* mikrotik, langkah selanjutnya adalah mengkonfigurasi manajemen *bandwidth queue tree* dengan metode HTB. Sebelum melakukan pembagian *bandwidth* pada *router* Mikrotik, maka harus menandai aliran paket menggunakan *mangle* (istilah pada Mikrotik) agar paket tersebut dapat dikenal oleh *queue tree*. *Mangle* berfungsi sebagai pembelah IP *traffic* dan memberi tanda (*Mark*) pada suatu IP *traffic* yang nanti akan di proses selanjutnya sesuai kebutuhan jaringan.

Langkah pertama yang harus di lakukan adalah melakukan konfigurasi *marking packet* pada *firewall mangle*. Konfigurasi ini akan diawali dengan melakukan *marking connection* pada chain prerouting yang berguna untuk menangkap koneksi yang dibuat oleh *client*. Setelah konfigurasi *marking connection* dilakukan, maka dilanjutkan dengan melakukan konfigurasi *marking packet*. Konfigurasi *marking packet* ini juga dilakukan pada chain prerouting.

1. Marking Connection

Langkah yang dilakukan untuk *marking connection* adalah buka menu IP → Firewall → Mangle → klik tanda (+) pada tab general, pilih chain = prerouting, src. address = 192.168.1.2 , in. Interface = ether2-lokal. Pada tab action pilih action = mark connection, new connection mark = 1-con dan ceklis passthrough. Passthrough = yes berfungsi agar *packet* yang sudah diproses pada konfigurasi *marking connection*, masih dapat di *marking* oleh konfigurasi *marking packet* selanjutnya.



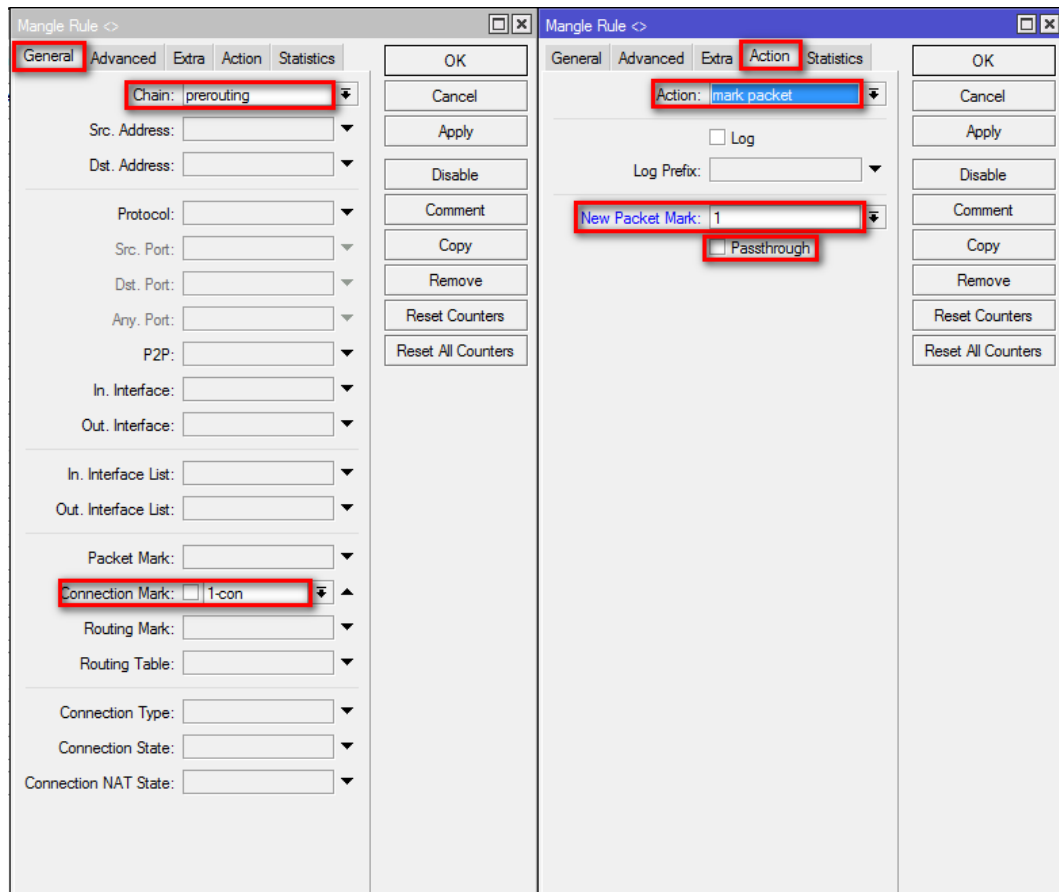
Gambar 4.11. Konfigurasi Marking Connection

#	Action	Chain	Src. Address	Dst. Address	Proto.	Src. Port	Dst. Port	In. Inter.	Out. Int.	Bytes	Packets
0	mark	prerouting	192.168.1.2					ether21...		0 B	0
1	mark	prerouting	192.168.1.3					ether21...		0 B	0
2	mark	prerouting	192.168.1.4					ether21...		0 B	0
3	mark	prerouting	192.168.1.5					ether21...		0 B	0
4	mark	prerouting	192.168.1.6					ether21...		0 B	0
5	mark	prerouting	192.168.1.7					ether21...		0 B	0
6	mark	prerouting	192.168.1.8					ether21...		0 B	0
7	mark	prerouting	192.168.1.9					ether21...		0 B	0
8	mark	prerouting	192.168.1.10					ether21...		0 B	0
9	mark	prerouting	192.168.1.11					ether21...		0 B	0
10	mark	prerouting	192.168.1.12					ether21...		0 B	0
11	mark	prerouting	192.168.1.13					ether21...		0 B	0
12	mark	prerouting	192.168.1.14					ether21...		0 B	0
13	mark	prerouting	192.168.1.15					ether21...		0 B	0
14	mark	prerouting	192.168.1.16					ether21...		0 B	0
15	mark	prerouting	192.168.1.17					ether21...		0 B	0
16	mark	prerouting	192.168.1.18					ether21...		0 B	0
17	mark	prerouting	192.168.1.19					ether21...		0 B	0
18	mark	prerouting	192.168.1.20					ether21...		0 B	0
19	mark	prerouting	192.168.1.21					ether21...		0 B	0
20	mark	prerouting	192.168.1.22					ether21...		0 B	0
21	mark	prerouting	192.168.1.23					ether21...		0 B	0
22	mark	prerouting	192.168.1.24					ether21...		0 B	0
23	mark	prerouting	192.168.1.25					ether21...		0 B	0
24	mark	prerouting	192.168.1.26					ether21...		0 B	0
25	mark	prerouting	192.168.1.27					ether21...		0 B	0
26	mark	prerouting	192.168.1.28					ether21...		0 B	0
27	mark	prerouting	192.168.1.29					ether21...		0 B	0
28	mark	prerouting	192.168.1.30					ether21...		0 B	0
29	mark	prerouting	192.168.1.31					ether21...		0 B	0
30	mark	prerouting	192.168.1.32					ether21...		0 B	0
31	mark	prerouting	192.168.1.33					ether21...		0 B	0
32	mark	prerouting	192.168.1.34					ether21...		0 B	0
33	mark	prerouting	192.168.1.35					ether21...		0 B	0
34	mark	prerouting	192.168.1.36					ether21...		0 B	0
35	mark	prerouting	192.168.1.37					ether21...		0 B	0

Gambar 4.12. Hasil Konfigurasi *Marking Connection*

2. *Marking Packet*

Setelah konfigurasi *marking connection* kemudian dilanjutkan dengan membuat konfigurasi *marking packet*. Konfigurasi ini harus berdasarkan konfigurasi *marking connection* sebelumnya, sehingga harus menggunakan parameter *connection-mark = 1-con*. Parameter *connection-mark = 1-con* merupakan hasil dari konfigurasi *marking connection* sebelumnya. Langkah yang dilakukan untuk *marking* paket adalah buka menu IP → Firewall → Mangle → klik tanda (+) pada tab general, pilih chain = prerouting *connection-mark = 1-con*. Pada tab action pilih action = *mark packet*, *new packet mark = 1*, *passthrough = no* agar *packet* yang sudah di *marking* oleh konfigurasi *marking packet* tidak lagi diproses oleh konfigurasi yang lain.



Gambar 4.13. Konfigurasi *Marking Packet*

The screenshot shows the Mikrotik WinBox Firewall rule list. The table has columns for #, Action, Chain, Src. Address, Dst. Address, Proto., Src. Port, Dst. Port, In. Inter., Out. Int., Bytes, and Packets. The first 100 rows are selected, showing a list of rules with Action 'mar...', Chain 'prerouting', and 0 Bytes and 0 Packets. The status bar at the bottom indicates '100 items (1 selected)'.

#	Action	Chain	Src. Address	Dst. Address	Proto.	Src. Port	Dst. Port	In. Inter.	Out. Int.	Bytes	Packets
50	mar...	prerouting								0 B	0
51	mar...	prerouting								0 B	0
52	mar...	prerouting								0 B	0
53	mar...	prerouting								0 B	0
54	mar...	prerouting								0 B	0
55	mar...	prerouting								0 B	0
56	mar...	prerouting								0 B	0
57	mar...	prerouting								0 B	0
58	mar...	prerouting								0 B	0
59	mar...	prerouting								0 B	0
60	mar...	prerouting								0 B	0
61	mar...	prerouting								0 B	0
62	mar...	prerouting								0 B	0
63	mar...	prerouting								0 B	0
64	mar...	prerouting								0 B	0
65	mar...	prerouting								0 B	0
66	mar...	prerouting								0 B	0
67	mar...	prerouting								0 B	0
68	mar...	prerouting								0 B	0
69	mar...	prerouting								0 B	0
70	mar...	prerouting								0 B	0
71	mar...	prerouting								0 B	0
72	mar...	prerouting								0 B	0
73	mar...	prerouting								0 B	0
74	mar...	prerouting								0 B	0
75	mar...	prerouting								0 B	0
76	mar...	prerouting								0 B	0
77	mar...	prerouting								0 B	0
78	mar...	prerouting								0 B	0
79	mar...	prerouting								0 B	0
80	mar...	prerouting								0 B	0
81	mar...	prerouting								0 B	0
82	mar...	prerouting								0 B	0
83	mar...	prerouting								0 B	0
84	mar...	prerouting								0 B	0
85	mar...	prerouting								0 B	0

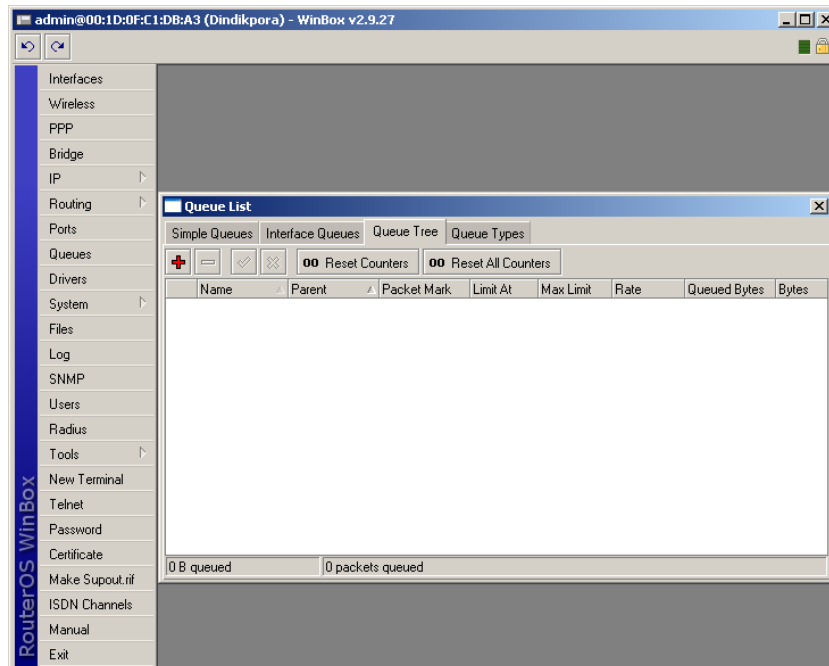
Gambar 4.14. Hasil Konfigurasi *Marking Packet*

4.1.2.2. Konfigurasi HTB

Konfigurasi HTB disini menggunakan *queue* jenis *queue tree*. Konfigurasi *queue* pada *queue tree* memiliki kesamaan dalam penggunaan parameter *priority* maupun parameter *burst* dengan *simple queue*. *Queue tree* adalah konfigurasi yang bersifat *one way* (satu arah), ini berarti sebuah konfigurasi sebuah konfigurasi *queue* hanya mampu melakukan *queue* terhadap 1 arah jenis *traffic*. Jika sebuah konfigurasi *queue* pada *queue tree* ditujukan untuk melakukan *queue* terhadap *bandwidth download*, maka konfigurasi tersebut tidak akan melakukan *queue* untuk *bandwidth upload*, demikian pula sebaliknya. Sehingga untuk melakukan *queue* terhadap *traffic upload* dan *download* dari sebuah komputer *client*, harus di buat 2 konfigurasi *queue*. Langkah terpenting yang harus dilakukan untuk mengkonfigurasi HTB adalah membuat *inner queue* atau disebut *parent*. Karena *queue tree* bersifat *one way* (satu arah) maka harus di buat 2 konfigurasi untuk *traffic download* dan *upload*.

1. Konfigurasi *Traffic Download*

Langkah untuk membuat *Inner queue traffic download* adalah buka menu *Queue*, pilih *queue tree*, kemudian klik tanda (+).



Gambar 4.15. Konfigurasi *Queue Tree* Dengan Metode HTB

Buat rule dengan parameter sebagai berikut :

Pada tab general :

Name = *all-download* ruang guru

Parent = ether2-lokal

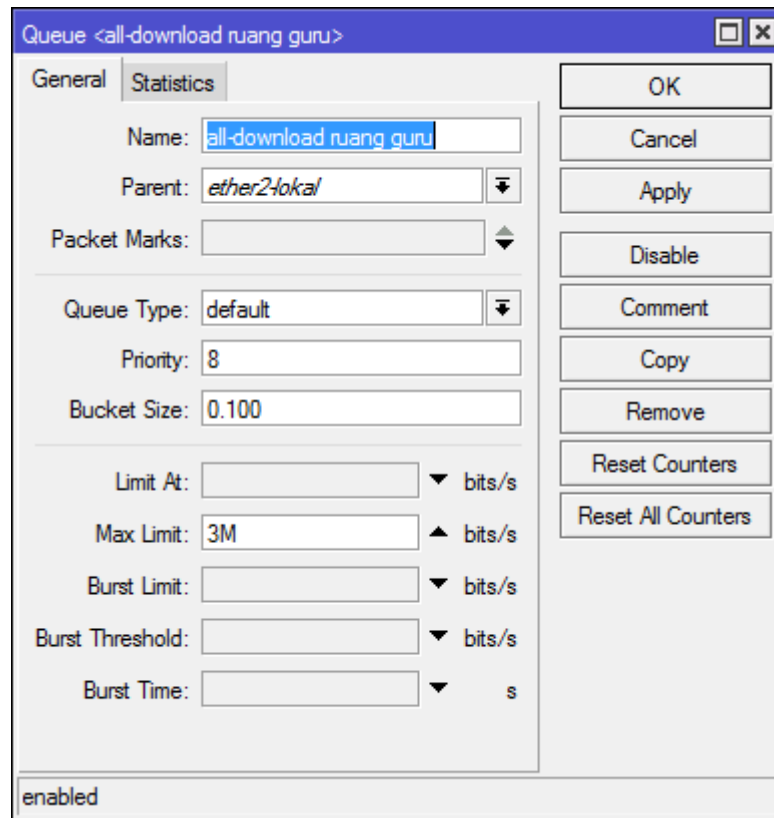
Queue type = default

Priority = 8

Bucket size = 0.100

Max limit = 3M

Kemudian klik apply dan ok



Gambar 4.16. Konfigurasi Parent Queue Traffic Download

Setelah membuat *inner queue* atau *parent* untuk *traffic download*, selanjutnya adalah membuat *child queue* untuk *traffic download*. Langkah untuk membuat *child queue* adalah buka menu *Queue*, pilih *queue tree*, kemudian klik tanda (+). Buat rule dengan parameter sebagai berikut :

Pada tab general :

Name = 1-download

Parent = all-download ruang guru

Packet Marks = 1

Queue type = default

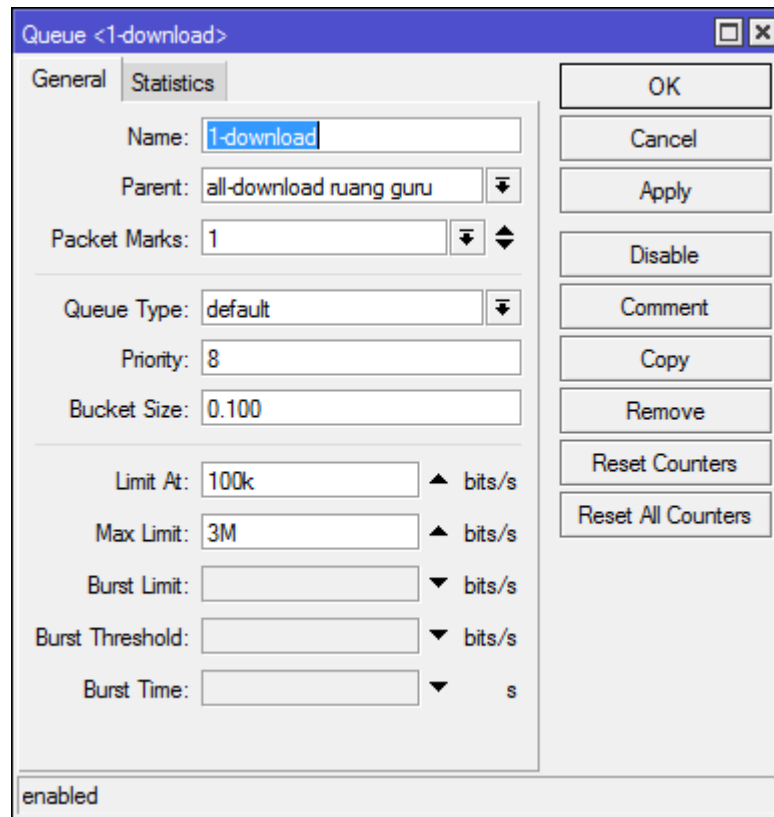
Priority = 8

Bucket size = 0.100

Limit at = 100k

Max limit = 3M

Kemudian klik apply dan ok



Gambar 4.17. Konfigurasi Child Queue Traffic Download

2. Konfigurasi Traffic Upload

Langkah untuk membuat *Inner queue traffic upload* adalah buka menu *Queue*, pilih *queue tree*, kemudian klik tanda (+). Buat rule dengan parameter sebagai berikut :

Pada tab general :

Name = all-upload ruang guru

Parent = ether1-publik

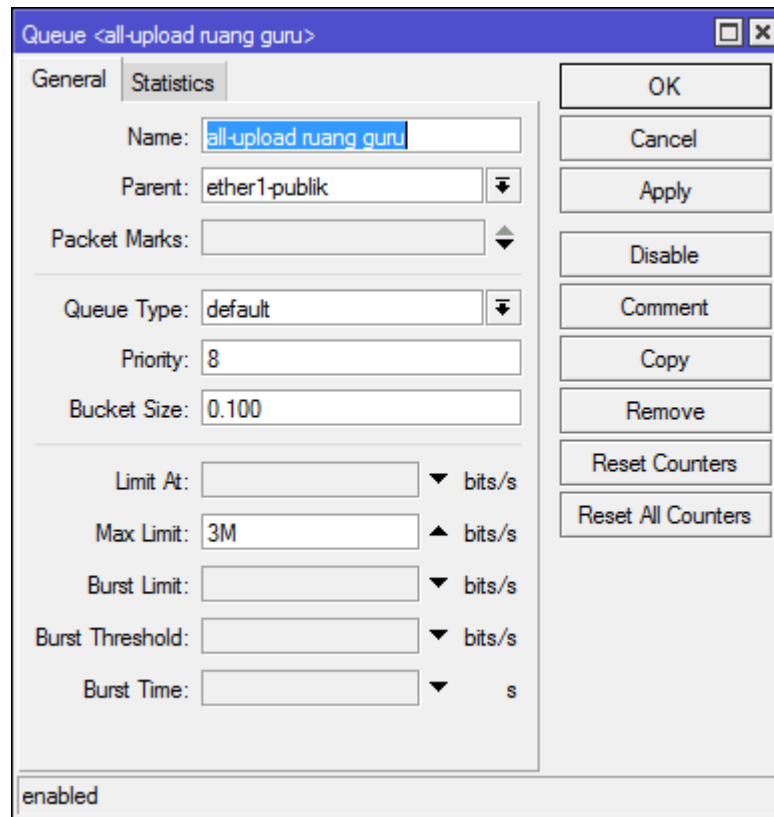
Queue type = default

Priority = 8

Bucket size = 0.100

Max limit = 3M

Kemudian klik apply dan ok



Gambar 4.18. Konfigurasi Parent Queue Traffic Upload

Setelah membuat *inner queue* atau *parent* untuk *traffic upload*, selanjutnya adalah membuat *child queue* untuk *traffic upload*. Langkah untuk membuat *child queue* adalah buka menu *Queue*, pilih *queue tree*, kemudian klik tanda (+). Buat rule dengan parameter sebagai berikut :

Pada tab general :

Name = 1

Parent = all-upload ruang guru

Packet Marks = 1

Queue type = default

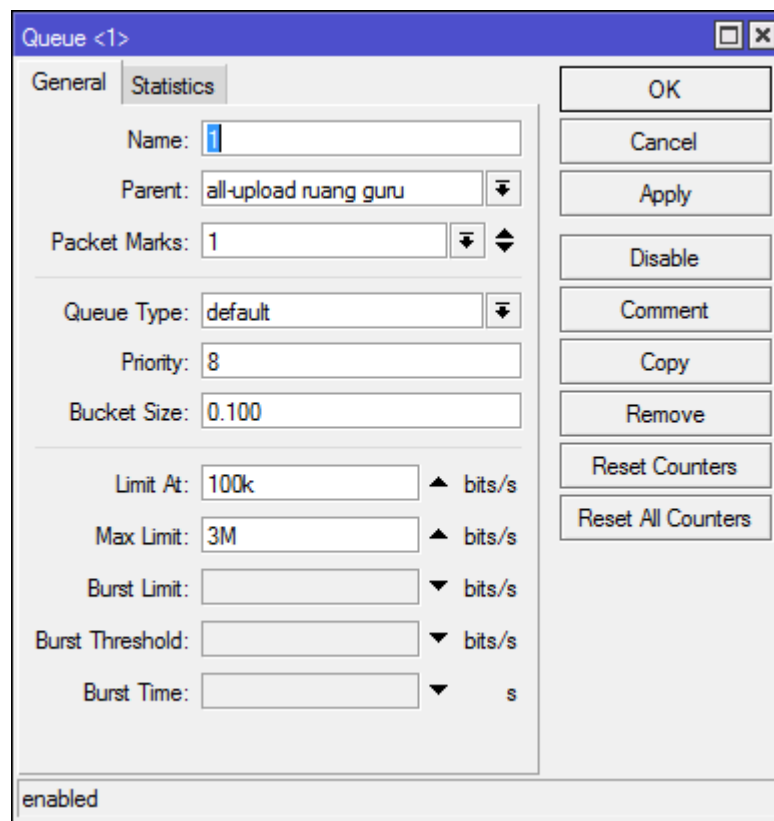
Priority = 8

Bucket size = 0.100

Limit at = 100k

Max limit = 3M

Kemudian klik apply dan ok



Gambar 4.19. Konfigurasi *Child Queue Traffic Upload*

Name	Parent	Packet...	Limit At (b...	Max Limit	Avg. R...	Queued Bytes	Bytes	Packets
all-downlo...	ether2/lokal			3M	0 bps	0 B	0 B	0
1-dow...	all-download r...	1	100k	3M	0 bps	0 B	0 B	0
2-dow...	all-download r...	2	100k	3M	0 bps	0 B	0 B	0
3-dow...	all-download r...	3	100k	3M	0 bps	0 B	0 B	0
4-dow...	all-download r...	4	100k	3M	0 bps	0 B	0 B	0
5-dow...	all-download r...	5	100k	3M	0 bps	0 B	0 B	0
6-dow...	all-download r...	6	100k	3M	0 bps	0 B	0 B	0
7-dow...	all-download r...	7	100k	3M	0 bps	0 B	0 B	0
8-dow...	all-download r...	8	100k	3M	0 bps	0 B	0 B	0
9-dow...	all-download r...	9	100k	3M	0 bps	0 B	0 B	0
10-do...	all-download r...	10	100k	3M	0 bps	0 B	0 B	0
11-do...	all-download r...	11	100k	3M	0 bps	0 B	0 B	0
12-do...	all-download r...	12	100k	3M	0 bps	0 B	0 B	0
13-do...	all-download r...	13	100k	3M	0 bps	0 B	0 B	0
14-do...	all-download r...	14	100k	3M	0 bps	0 B	0 B	0
15-do...	all-download r...	15	100k	3M	0 bps	0 B	0 B	0
16-do...	all-download r...	16	100k	3M	0 bps	0 B	0 B	0
17-do...	all-download r...	17	100k	3M	0 bps	0 B	0 B	0
18-do...	all-download r...	18	100k	3M	0 bps	0 B	0 B	0
19-do...	all-download r...	19	100k	3M	0 bps	0 B	0 B	0
20-do...	all-download r...	20	100k	3M	0 bps	0 B	0 B	0
21-do...	all-download r...	21	100k	3M	0 bps	0 B	0 B	0
22-do...	all-download r...	22	100k	3M	0 bps	0 B	0 B	0
23-do...	all-download r...	23	100k	3M	0 bps	0 B	0 B	0
24-do...	all-download r...	24	100k	3M	0 bps	0 B	0 B	0
25-do...	all-download r...	25	100k	3M	0 bps	0 B	0 B	0
26-do...	all-download r...	26	100k	3M	0 bps	0 B	0 B	0
27-do...	all-download r...	27	100k	3M	0 bps	0 B	0 B	0
28-do...	all-download r...	28	100k	3M	0 bps	0 B	0 B	0
29-do...	all-download r...	29	100k	3M	0 bps	0 B	0 B	0
30-do...	all-download r...	30	100k	3M	0 bps	0 B	0 B	0
31-do...	all-download r...	31	100k	3M	0 bps	0 B	0 B	0
32-do...	all-download r...	32	100k	3M	0 bps	0 B	0 B	0
33-do...	all-download r...	33	100k	3M	0 bps	0 B	0 B	0
34-do...	all-download r...	34	100k	3M	0 bps	0 B	0 B	0
35-do...	all-download r...	35	100k	3M	0 bps	0 B	0 B	0
36-do...	all-download r...	36	100k	3M	0 bps	0 B	0 B	0
37-do...	all-download r...	37	100k	3M	0 bps	0 B	0 B	0
38-do...	all-download r...	38	100k	3M	0 bps	0 B	0 B	0
39-do...	all-download r...	39	100k	3M	0 bps	0 B	0 B	0
40-do...	all-download r...	40	100k	3M	0 bps	0 B	0 B	0
all-upload...	ether1/publik			3M	0 bps	0 B	0 B	0
1-upload...	all-upload rus...	1	100k	3M	0 bps	0 B	0 B	0
2-upload...	all-upload rus...	2	100k	3M	0 bps	0 B	0 B	0
3-upload...	all-upload rus...	3	100k	3M	0 bps	0 B	0 B	0
4-upload...	all-upload rus...	4	100k	3M	0 bps	0 B	0 B	0
5-upload...	all-upload rus...	5	100k	3M	0 bps	0 B	0 B	0
6-upload...	all-upload rus...	6	100k	3M	0 bps	0 B	0 B	0
7-upload...	all-upload rus...	7	100k	3M	0 bps	0 B	0 B	0
8-upload...	all-upload rus...	8	100k	3M	0 bps	0 B	0 B	0
9-upload...	all-upload rus...	9	100k	3M	0 bps	0 B	0 B	0
10-upl...	all-upload rus...	10	100k	3M	0 bps	0 B	0 B	0
11-upl...	all-upload rus...	11	100k	3M	0 bps	0 B	0 B	0
12-upl...	all-upload rus...	12	100k	3M	0 bps	0 B	0 B	0
13-upl...	all-upload rus...	13	100k	3M	0 bps	0 B	0 B	0
14-upl...	all-upload rus...	14	100k	3M	0 bps	0 B	0 B	0
15-upl...	all-upload rus...	15	100k	3M	0 bps	0 B	0 B	0
16-upl...	all-upload rus...	16	100k	3M	0 bps	0 B	0 B	0
17-upl...	all-upload rus...	17	100k	3M	0 bps	0 B	0 B	0
18-upl...	all-upload rus...	18	100k	3M	0 bps	0 B	0 B	0
19-upl...	all-upload rus...	19	100k	3M	0 bps	0 B	0 B	0
20-upl...	all-upload rus...	20	100k	3M	0 bps	0 B	0 B	0
21-upl...	all-upload rus...	21	100k	3M	0 bps	0 B	0 B	0
22-upl...	all-upload rus...	22	100k	3M	0 bps	0 B	0 B	0
23-upl...	all-upload rus...	23	100k	3M	0 bps	0 B	0 B	0
24-upl...	all-upload rus...	24	100k	3M	0 bps	0 B	0 B	0
25-upl...	all-upload rus...	25	100k	3M	0 bps	0 B	0 B	0
26-upl...	all-upload rus...	26	100k	3M	0 bps	0 B	0 B	0
27-upl...	all-upload rus...	27	100k	3M	0 bps	0 B	0 B	0
28-upl...	all-upload rus...	28	100k	3M	0 bps	0 B	0 B	0
29-upl...	all-upload rus...	29	100k	3M	0 bps	0 B	0 B	0
30-upl...	all-upload rus...	30	100k	3M	0 bps	0 B	0 B	0
31-upl...	all-upload rus...	31	100k	3M	0 bps	0 B	0 B	0
32-upl...	all-upload rus...	32	100k	3M	0 bps	0 B	0 B	0
33-upl...	all-upload rus...	33	100k	3M	0 bps	0 B	0 B	0
34-upl...	all-upload rus...	34	100k	3M	0 bps	0 B	0 B	0
35-upl...	all-upload rus...	35	100k	3M	0 bps	0 B	0 B	0
36-upl...	all-upload rus...	36	100k	3M	0 bps	0 B	0 B	0
37-upl...	all-upload rus...	37	100k	3M	0 bps	0 B	0 B	0
38-upl...	all-upload rus...	38	100k	3M	0 bps	0 B	0 B	0
39-upl...	all-upload rus...	39	100k	3M	0 bps	0 B	0 B	0
40-upl...	all-upload rus...	40	100k	3M	0 bps	0 B	0 B	0

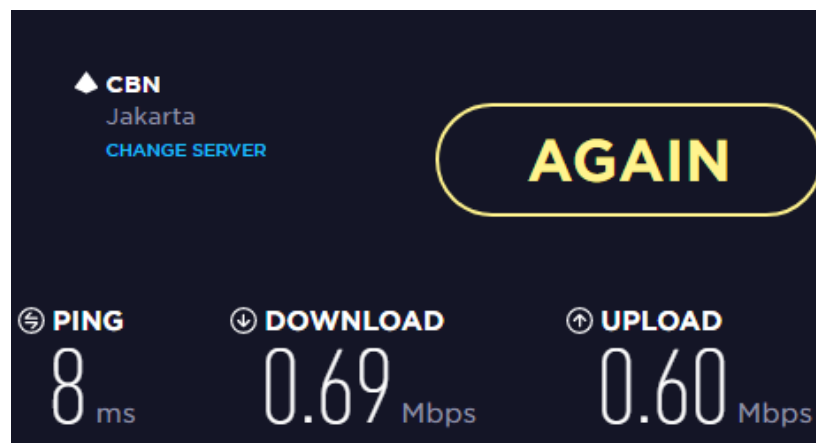
Gambar 4.20. Hasil Akhir Konfigurasi *Queue Tree* Dengan Metode HTB

4.1.3. Hasil Pengukuran *Bandwidth*

Berdasarkan hasil pengukuran yang dilakukan pada jaringan *hotspot* ruang guru, didapatkan hasil bandwidth dalam kilobit per second (Kbps) seperti tabel dibawah ini.

Tabel 4.2. Hasil Pengukuran *Bandwidth*

Hari/Tanggal	<i>Bandwidth</i> (Kbps)	Kapasitas <i>Bandwidth</i> (Kbps)	Minimal <i>Bandwidth/User</i> (Kbps)
Senin, 17 April 2017	690	3000	100
Selasa, 18 April 2017	700	3000	100
Rabu, 19 April 2017	970	3000	100
Kamis, 20 April 2017	510	3000	100
Jumat, 21 April 2017	620	3000	100
Total Rata-rata <i>Bandwidth</i>	698	3000	100



Gambar 4.21 Hasil Pengukuran *Bandwidth* Senin 17 April 2017

4.1.4 Hasil Pengukuran *Throughput*

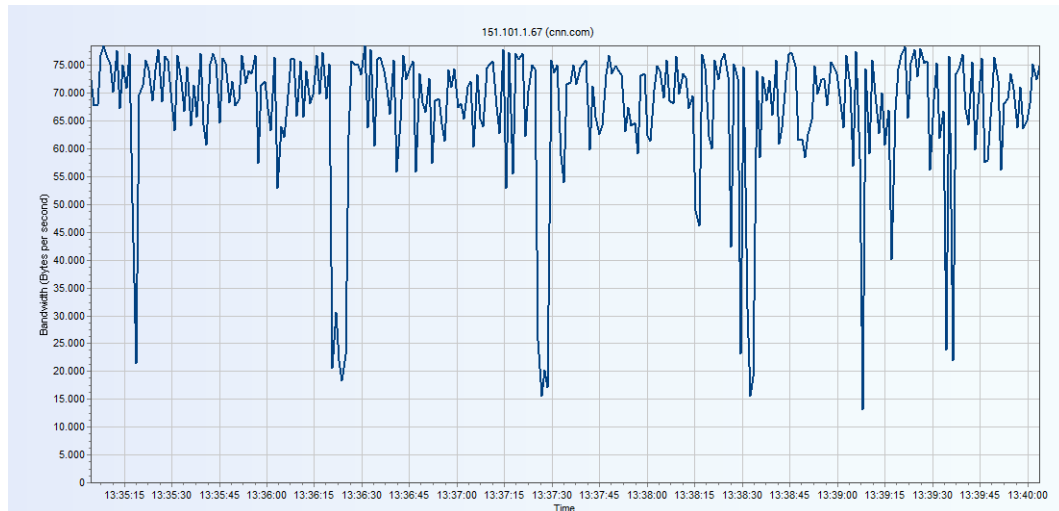
Berdasarkan pengukuran *throughput* yang dilakukan menggunakan software axence net tools pada jaringan *hotspot* ruang guru, didapat hasil *throughput* dalam bit per second (bps).

Tabel 4.3. Hasil Pengukuran *Throughput*

Hari / Tanggal	Perangkat Pengukuran	<i>Throughput</i> (b/s)		
		Min	Max	Avg
Senin, 17 April 2017	www.cnn.com	104 560	628 112	527 346
	www.facebook.com	120 048	625 024	536 148
	www.mail.yahoo.com	18 200	260 840	233 046
Selasa, 18 April 2017	www.cnn.com	16 976	625 752	486 252
	www.facebook.com	16 520	621 688	478 719
	www.mail.yahoo.com	24 512	260 336	230 513
Rabu, 19 April 2017	www.cnn.com	16 864	691 136	403 313
	www.facebook.com	16 584	612 288	465 168
	www.mail.yahoo.com	24 040	274 176	235 296
Kamis, 20 April 2017	www.cnn.com	18 136	705 576	395 170
	www.facebook.com	17 752	717 200	529 243
	www.mail.yahoo.com	52 664	260 920	235 291
Jumat, 21 April 2017	www.cnn.com	17 048	734 416	591 685
	www.facebook.com	16 528	625 736	494 996
	www.mail.yahoo.com	16 560	261 880	215 133

Address: cnn.com	
Information	
Bandwidth:	
Average	527 346 bit/s
Minimum	104 560 bit/s
Maximum	628 112 bit/s
Packets:	
Sent	1 271
Received	1 271 (100 %)
Lost	0 (0 %)
Options	

Gambar 4.22. Hasil Pengukuran *Throughput* Senin 17 April 2017



Gambar 4.23. Grafik *Throughput* Senin 17 April 2017

4.1.5. Hasil Pengukuran *Delay*

Berdasarkan pengukuran *delay* menggunakan software axence net tools yang dilakukan pada jaringan *hotspot* ruang guru, didapat hasil *delay* dalam mili second (ms).

Tabel 4.4. Hasil Pengukuran *Delay*

Hari / Tanggal	Perangkat Pengukuran	<i>Response Time</i> (ms)				
		Last	Avg	Min	Max	Sent
Senin, 17						
April 2017	www.cnn.com	32	27	21	156	1000
	www.facebook.com	25	27	21	229	1000
	mail.yahoo.com	57	110	54	424	1000
Selasa, 18						
April 2017	www.cnn.com	24	52	22	996	1103
	www.facebook.com	23	48	21	971	1103
	mail.yahoo.com	58	122	55	973	1103
Rabu, 19 April 2017						
		24	26	22	566	1120

	www.cnn.com					
	www.facebook.com	27	26	21	585	1120
	mail.yahoo.com	220	136	54	527	1120
Kamis, 20						
April 2017	www.cnn.com	213	216	211	353	1000
	www.facebook.com	24	39	24	209	1000
	mail.yahoo.com	25	26	22	124	1000
Jumat, 21						
April 2017	www.cnn.com	30	184	22	997	1000
	www.facebook.com	25	66	22	998	1000
	mail.yahoo.com	63	109	62	999	1000

Sent time	Response time (ms)
12:21:33	57
12:21:32	57
12:21:31	88
12:21:30	57
12:21:29	57
12:21:28	55
12:21:27	55
12:21:26	55
12:21:25	64
12:21:24	57
12:21:23	55
12:21:22	55

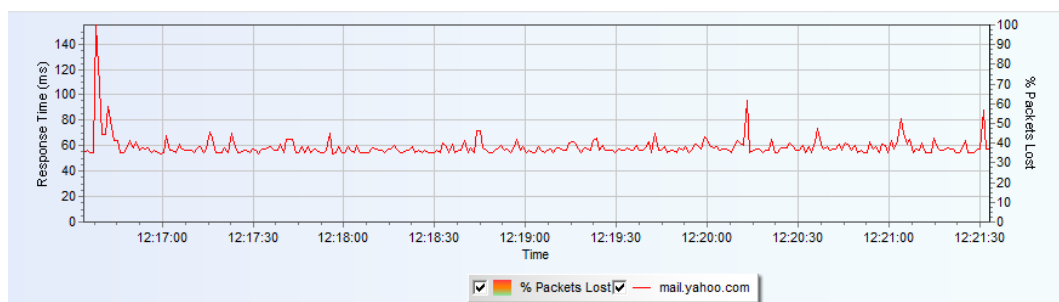
Gambar 4.24 Hasil Pengukuran *Delay* Senin 17 April 2017

4.1.6. Hasil Pengukuran *Packet loss*

Berdasarkan pengukuran *packet loss* menggunakan software axence net tools yang dilakukan pada jaringan *hotspot* ruang guru, didapat hasil *packet loss* sebagai berikut:

Tabel 4.5. Hasil Pengukuran *Packet loss*

Hari / Tanggal	Perangkat Pengukuran	Packets		
		Sent	Loss	Loss%
Senin, 17 April 2017	www.cnn.com	1000	2	0
	www.facebook.com	1000	1	0
	mail.yahoo.com	1000	1	0
Selasa, 18 April 2017	www.cnn.com	1104	66	6
	www.facebook.com	1104	67	6
	mail.yahoo.com	1104	74	7
Rabu, 19 April 2017	www.cnn.com	1120	1	0
	www.facebook.com	1120	0	0
	mail.yahoo.com	1120	1	0
Kamis, 20 April 2017	www.cnn.com	1000	138	14
	www.facebook.com	1000	32	3
	mail.yahoo.com	1000	27	3
Jumat, 21 April 2017	www.cnn.com	1000	27	3
	www.facebook.com	1000	17	2
	mail.yahoo.com	1000	18	2

Gambar 4.25. Grafik *Packet loss* Senin 17 April 2017

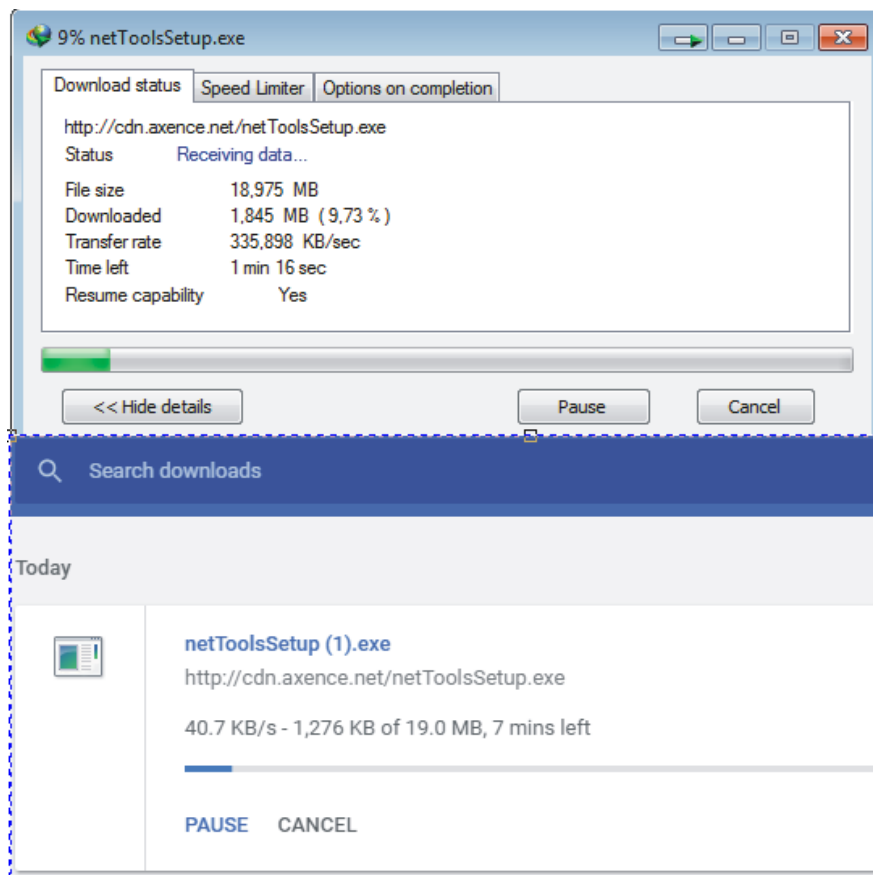
4.2. Analisis Data Penelitian

Setelah mengkonfigurasi manajemen *bandwidth queue tree* dengan metode HTB dan mendapatkan hasil dari pengukuran pada jaringan *hotspot* ruang guru terhadap parameter kualitas layanan yang terdiri dari *bandwidth*, *throughput*, *packet loss* dan *delay*. Data hasil penelitian ini dianalisis untuk mengetahui perbedaan manajemen *bandwidth simple queue* yang digunakan SMK Negeri 22 saat ini dengan manajemen *bandwidth queue tree* menggunakan metode HTB terhadap parameter kualitas layanan yang terdiri dari *bandwidth*, *throughput*, *delay* dan *packet loss*.

4.2.1. Manajemen *Bandwidth Hierarchical Token Bucket* (HTB)

Pada bagian ini penulis menguji manajemen *bandwidth* dengan metode Hierarchical Token Bucket (HTB) dan menggunakan system antrian *queue tree*. Pengujian dilakukan menggunakan dua laptop yang berlaku sebagai *client*.

4.2.1.1. Test *Download* Sebelum Penerapan Manajemen *Bandwidth* HTB

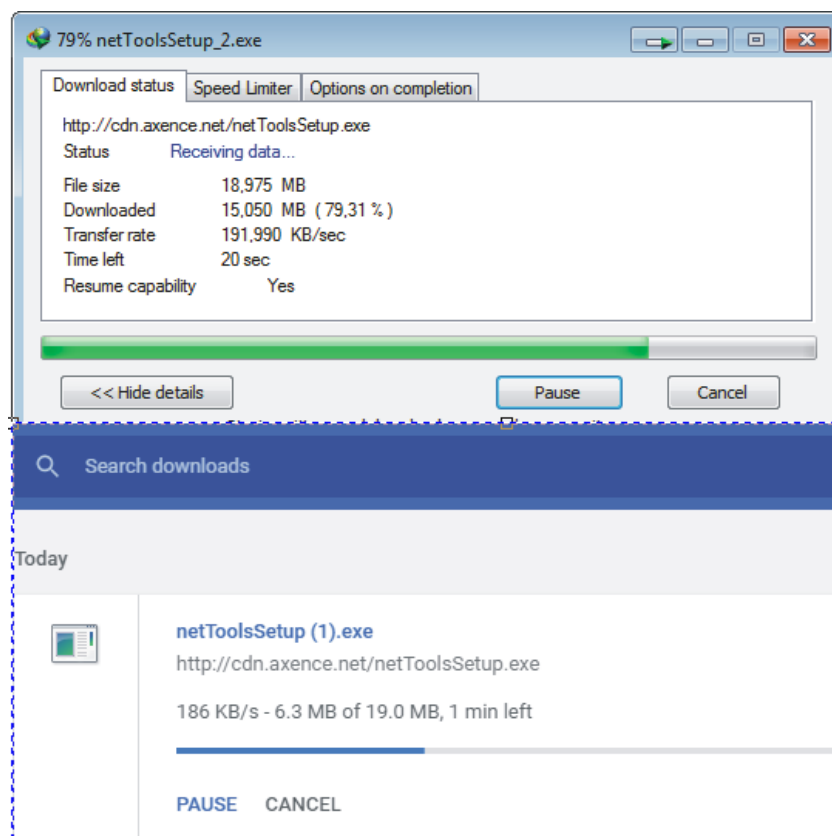


Gambar 4.26. Hasil *download* sebelum penerapan manajemen *bandwidth* HTB oleh 2 *client* secara bersamaan

Berdasarkan Gambar 4.26. sebelum menggunakan manajemen *bandwidth* dengan metode HTB distribusi *bandwidth* ke setiap *client* tidak merata dan kedua *client* saling berebut *bandwidth*. Alokasi *bandwidth* yang didistribusikan ke *router* mikrotik adalah sebesar 3000 Kbps (375KB/s), dimana ada 2 *client* melakukan aktivitas *download* secara bersamaan ke url yang sama yaitu <http://http://axence.net/en/axence-nettools>. Berdasarkan gambar diatas bisa dilihat bahwa distribusi *bandwidth* ke kedua pc *client* tidak terdistribusi secara merata dikarenakan salah satu pc *client* menggunakan software *download manager*. Hasil *download* yang diperoleh oleh *client* yang menggunakan *download manager*

adalah sebesar 335,898 KB/s sedangkan hasil *download* oleh *client* yang tidak menggunakan *download manager* adalah sebesar 40,7 KB/s. Hampir seluruh distribusi *bandwidth* ke *router* diambil oleh *pc client* yang melakukan *download* dengan menggunakan *download manager* yaitu sebesar $8 \times 335,898 \text{ KB/s} = 2687 \text{ Kbps}$.

4.2.1.2. Test *Download* Setelah Penerapan Manajemen *Bandwidth* HTB



Gambar 4.27. Hasil *download* setelah penerapan manajemen *bandwidth* HTB oleh 2 *client* secara bersamaan

Bandwidth yang didistribusikan ke *router* mikrotik sebesar 3Mbps. *Download* maksimum yang dialokasikan adalah sebesar 3Mbps dan *download* minimum yang diberikan sebesar 100 Kbps, berdasarkan Gambar 4.27 kedua *client* melakukan *download* secara bersamaan ke url yang sama yaitu <http://axence.net/en/axence-nettools>. dari gambar tersebut bisa disimpulkan bahwa

distribusi *bandwidth* ke kedua *pc client* bisa terdistribusi secara merata meskipun salah satu *client* menggunakan *download* manager, *bandwidth* yang diterima oleh kedua *client* tetap sama rata yaitu *client1* yang melakukan *download* dengan menggunakan *download* manager memperoleh *download* sebesar 191 KB/s sedangkan *client 2* yang melakukan *download* tanpa menggunakan *download* manager memperoleh 186 KB/s.

Untuk memastikan apakah metode HTB yang di buat benar-benar berhasil, dilakukan uji coba lagi dengan melakukan *download* ke alamat url <http://repo.ugm.ac.id/iso/ubuntu/releases/zesty/ubuntu-17.04-server-amd64.iso> . Uji coba *download* dilakukan 3 kali, yaitu ketika hanya 1 *client* yang mendownload, 2 *client* yang mendownload dan 3 *client* yang mendownload.

The screenshot shows the Queue List interface with the following data:

Name	Parent	Packet ...	Limit At (b...	Max Limit ...	Avg. Rate	Queued Bytes	Bytes	Packets
all-downlo...	ether2-lokal			3M	3.0 Mbps	0 B	186.3 ...	134 146
0	all-download r...	46	100k	3M	3.0 Mbps	39.8 KiB	80.0 MiB	56 101
1	all-download r...	2	100k	3M	4.6 kbps	0 B	71.4 MiB	50 672
2	all-download r...	1	100k	3M	0 bps	0 B	9.7 MiB	7 012

Gambar 4.28. Uji Coba Download Dengan 1 Client

The screenshot shows the Queue List interface with the following data:

Name	Parent	Packet ...	Limit At (b...	Max Limit ...	Avg. Rate	Queued Bytes	Bytes	Packets
all-downlo...	ether2-lokal			3M	3.0 Mbps	0 B	297.6 ...	211 870
0	all-download r...	46	100k	3M	1528.1 kbps	72.3 KiB	152.9 ...	106 910
1	all-download r...	2	100k	3M	1506.3 kbps	66.4 KiB	109.9 ...	77 654
2	all-download r...	1	100k	3M	0 bps	0 B	9.7 MiB	7 012

Gambar 4.29. Uji Coba Download Dengan 2 Client

Name	Parent	Packet ...	Limit At (b...	Max Limit ...	Avg. Rate	Queued Bytes	Bytes	Packets
all-downlo...	ether2-lokal			3M	3.0 Mbps	0 B	894.8 ...	631 291
0	all-download r...	46	100k	3M	1031.9 kbps	67.8 KiB	447.9 ...	312 081
1	all-download r...	2	100k	3M	1014.7 kbps	63.4 KiB	398.0 ...	281 612
2	all-download r...	43	100k	3M	1061.2 kbps	63.4 KiB	14.1 MiB	10 249

Gambar 4.30. Uji Coba Download Dengan 3 Client

Berdasarkan Gambar 4.28 ketika hanya satu *client* yang melakukan *download*, *client* tersebut mendapat seluruh *bandwidth* yang ada. Berdasarkan Gambar 4.29 ketika ada 2 *client* yang melakukan *download*, maka *bandwidth* akan di bagi rata untuk kedua *client*. Berdasarkan Gambar 4.30 ketika ada 3 *client* yang melakukan *download*, maka *bandwidth* akan dibagi rata sehingga semua *client* akan mendapatkan *bandwidth* yang sama.

4.2.2. Bandwidth

Saat ini SMK Negeri 22 melakukan manajemen *bandwidth* jenis *simple queue*. Dengan manajemen *bandwidth simple queue* ini administrator memberi *bandwidth* 3 Mbps ke IP jaringan, jadi setiap *user* di dalam jaringan tersebut saling berebut *bandwidth* dan bahkan terkadang ada salah satu *user* mendominasi pemakaian *bandwidth* tersebut. Dengan manajemen *bandwidth queue tree* dengan metode HTB ini administrator dapat membatasi *bandwidth* minimum dan maximum yang dapat digunakan oleh tiap klien. Dengan metode HTB ini *bandwidth* yang tidak dipakai dapat digunakan atau dipinjam oleh *user* yang aktif. Jadi apabila ada *user* yang tidak aktif, *bandwidth* dapat di pakai atau di pinjam oleh *user* yang sedang aktif. Manajemen *bandwidth* dengan metode HTB ini membagi *bandwidth* 3 Mbps ke 40 *client* dengan minimal *bandwidth* 100 kbps per

client. Jadi setiap *client* dipastikan mendapat *bandwidth* minimalnya yaitu 100 kbps.

Berdasarkan hasil pengukuran yang telah dilakukan, dapat dilihat perbandingan minimal *bandwidth* yang dimiliki oleh tiap *client* dengan *bandwidth* tersedia yang didapat oleh *client* seperti pada Tabel 4.2. *Bandwidth* yang didapat berada diatas nilai *bandwidth* minimal yang diberikan ke tiap *client* yaitu 100 kbps. Hal ini terjadi karena tidak semua user sedang aktif atau menggunakan jatah *bandwidth*nya, jadi *bandwidth* yang di dapat bisa melebihi *bandwidth* minimal yang di berikan kepada tiap user.

4.2.3. Throughput

Throughput merupakan kemampuan sebenarnya suatu jaringan dalam melakukan pengiriman data. Dari hasil perhitungan *throughput* untuk perangkat monitor *server* cnn.com seperti pada Tabel 4.6. didapati rata-rata *throughput* sebesar 480.753 bps, jika dibandingkan dengan *bandwidth* jaringan maka didapat persentase sebesar 16.025 %. Sedangkan untuk perangkat monitor *server* facebook.com rata-rata *throughput* sebesar 500.854 bps, jika dibandingkan dengan *bandwidth* jaringan persentasenya sebesar 16.695 %. Untuk perangkat monitor *server* mail.yahoo.com rata-rata *throughput* sebesar 229855 bps, jika dibandingkan dengan *bandwidth* jaringan persentasenya sebesar 7.662 %.

Tabel 4.6. Hasil Akhir Pengukuran *Throughput*

No	Nama Perangkat	Server Tujuan	Bandwidth Tersedia	Rata-rata <i>Throughput</i>	Persentase (%)
1	Ruang Guru	www.cnn.com	3000000	480753	16.025
		www.facebook.com	3000000	500854	16.695
		www.mail.yahoo.com	3000000	229855	7.662

Berdasarkan Tabel 4.6 hasil rata-rata *throughput* pada cnn.com, facebook.com dan mail.yahoo.com lebih tinggi dari *bandwidth* minimum yang diberikan untuk setiap *client* yaitu 100 kbps dengan total 40 klien. Bila dibandingkan dengan *bandwidth* yang tersedia di jaringan, memang hanya mendapatkan persentase maksimal 16.695 % yaitu pada www.facebook.com. Hal ini disebabkan karena proses pengukuran dilakukan pada trafik yang cukup padat. Tetapi jika dibandingkan dengan *bandwidth* minimum yang diberikan ke setiap *client* yaitu 100 kbps, nilai tersebut jauh lebih tinggi dari *bandwidth* minimum yang diberikan ke setiap *client*. Hal yang menyebabkan nilai *throughput* tiap klien berbeda dengan *bandwidth* yang tersedia adalah jarak media transmisi antar klien dengan *server* dan media-media yang dilalui, serta spesifikasi komputer yang digunakan oleh *client*.

4.2.4. Delay

Berdasarkan hasil pengukuran seperti Tabel 4.7. untuk perangkat monitor *server* cnn.com didapati rata-rata *delay* sebesar 101 ms, *delay* ini menurut versi TIPHON termasuk kategori *delay* sangat bagus karena nilai *delay* <150 ms. Selanjutnya untuk pengukuran pada perangkat monitor *server* facebook.com

didapati rata-rata *delay* sebesar 41 ms dan mendapatkan predikat sangat bagus karena *delay* <150 ms. Sedangkan untuk perangkat monitor *server* mail.yahoo.com didapatkan rata-rata *delay* sebesar 90 ms, *delay* ini dikategorikan sangat bagus menurut TIPHON karena *delay* <150 ms.

Tabel 4.7. Hasil Akhir Pengukuran *Delay*

No	Perangkat Pengukuran	Rata-rata <i>Delay</i> (ms)	TIPHON
1	Ruang Guru		
	www.cnn.com	101	Sangat bagus
	www.facebook.com	41	Sangat Bagus
	www.mail.yahoo.com	90	Sangat Bagus

Faktor yang mempengaruhi perbedaan nilai *delay* untuk tiap-tiap perangkat pengukuran yaitu adanya perbedaan jarak *server* dari 3 perangkat pengukuran yaitu cnn.com, facebook.com dan mail.yahoo.com. www.cnn.com mendapat rata-rata *delay* tertinggi karena lokasi *server* nya berada jauh dari Indonesia yaitu di United States sedangkan lokasi *server* facebook.com dan mail.yahoo.com berada lebih dekat dari Indonesia yaitu di singapore. Selain itu jarak media transmisi untuk tiap-tiap perangkat pengukuran pada SMK Negeri 22 Jakarta ini ke *server* mikrotik juga mempengaruhi besarnya nilai *delay*, seperti halnya pada *workstation* ruang guru terdapat tambahan media fisik yaitu switch yang menyebabkan semakin panjangnya perjalanan paket data.

4.2.5. Packet loss

Menurut versi TIPHON standarisasi pengukuran nilai *packet loss* dapat dikategorikan sangat bagus apabila *packet loss* 0%, bagus jika *packet loss* 3%, sedang jika *packet loss* 15% dan jelek apabila *packet loss* 25%.

Tabel 4.8. Hasil Akhir Pengukuran *Packet loss*

No	Perangkat Pengukuran	<i>Packet loss</i>			TIPHON
		Sent	Loss	%Loss	
1	Ruang Guru				
	www.cnn.com	1044	47	5	Bagus
	www.facebook.com	1044	23	2	Sangat Bagus
	www.mail.yahoo.com	1044	24	2	Sangat Bagus

Berdasarkan hasil pengukuran seperti Tabel 4.8 untuk perangkat monitor *server* cnn.com didapati rata-rata *packet loss* sebesar 5%, *packet loss* ini menurut versi TIPHON termasuk kategori *packet loss* bagus karena *packet loss* ada diantara 3% s.d 15%. Selanjutnya untuk pengukuran pada perangkat monitor *server* www.facebook.com didapati rata-rata *packet loss* sebesar 2% dan mendapatkan predikat sangat bagus karena *packet loss* berkisar 0% s.d 3%. Sedangkan untuk perangkat monitor *server* www.mail.yahoo.com didapatkan rata-rata *packet loss* sebesar 2% dan mendapat kategori bagus menurut TIPHON karena *packet loss* berkisar 0% s.d 3%. Faktor yang mempengaruhi *packet loss* ini disebabkan karena proses pengukuran yang dilakukan pada saat trafik sedang cukup padat sehingga menyebabkan kemungkinan terjadinya tabrakan antar paket data atau *collision*, serta penumpukan paket data hingga melampaui kapasitas.

4.3. Pembahasan

Berdasarkan hasil analisis dari parameter *throughput*, *delay*, *packet loss* dan *bandwidth* didapatkan hasil akhir dari penelitian ini.

Tabel 4.9. Hasil Analisis Akhir Dari Seluruh Parameter

Web	<i>Average Throughput</i>	Persentase (%)	Predikat Tiphon	Nilai
cnn.com	480753	16.025	Jelek	1
facebook.com	500854	16.695	Jelek	1
mail.yahoo.com	229855	7.662	Jelek	1
Web	<i>Average Packet loss</i>	Persentase (%)	Predikat Tiphon	Nilai
cnn.com	47	5	Bagus	3
facebook.com	23	2	Sangat Bagus	4
mail.yahoo.com	24	2	Sangat Bagus	4
Web	<i>Average Delay</i>		Predikat Tiphon	Nilai
cnn.com	101		Sangat Bagus	4
facebook.com	41		Sangat Bagus	4
mail.yahoo.com	90		Sangat Bagus	4
			Hasil penjumlahan Nilai	26
			Dibagi jumlah Parameter yang diukur (9)	$26:9=2.89$

Berdasarkan hasil akhir dari seluruh parameter didapatkan indeks nilai akhir sebesar 2.89. Nilai ini ada dikisaran sedang pada predikat tiphon. Jika dibandingkan dengan Tabel 1.1 yaitu uji kualitas layanan sebelum menggunakan metode HTB, hasil setelah menggunakan metode HTB mengalami peningkatan pada parameter *throughput* pada perangkat pengujian www.cnn.com mengalami peningkatan dari yang sebelumnya sebesar 13.2504% menjadi 16.025% dan pada perangkat pengujian www.facebook.com juga mengalami kenaikan dari yang sebelumnya 3.3535% menjadi 16.695%. Pada parameter *packet loss*, setelah menggunakan metode HTB mengalami penurunan *packet loss* dari yang sebelumnya 18% menjadi 5% pada www.cnn.com, 19% menjadi 2% pada www.facebook.com dan 18% menjadi 2% pada www.mail.yahoo.com. Pada parameter *delay* juga mengalami penurunan nilai *delay* setelah menggunakan metode HTB dari yang sebelumnya 135 ms menjadi 41 ms pada www.facebook.com dan 108 ms menjadi 90 ms pada www.mail.yahoo.com. Dan pada indeks nilai akhir mengalami kenaikan dari yang sebelumnya 2.33 menjadi 2.89.

4.4. Aplikasi Hasil Penelitian

Setelah penelitian dilakukan Meskipun pada indeks nilai akhir ini belum mendapatkan hasil bagus tapi hasil ini sudah dapat membuktikan bahwa manajemen *bandwidth queue tree* dengan metode HTB dapat menurunkan nilai *delay* dan *packet loss* dan membuat kualitas layanan menjadi lebih baik. Peneliti berharap manajemen *bandwidth* dengan metode HTB dapat diterapkan di SMK Negeri 22 Jakarta karena dapat menurunkan nilai *delay* dan *packet loss* serta membuat kualitas layanan menjadi lebih baik.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

1. Manajemen *bandwidth* menggunakan metode *hierarchical token bucket* (HTB) berhasil di terapkan di *hotspot* ruang guru SMK Negeri 22 Jakarta menggunakan mikrotik RB750Gr3 dengan menggunakan fitur *queue tree*
2. Kualitas layanan di SMK Negeri 22 Jakarta saat menggunakan manajemen *bandwidth simple queue* mendapatkan indeks nilai akhir sebesar 2.33. Setelah menggunakan manajemen *bandwidth queue tree* dengan metode *hierarchical token bucket* (HTB), kualitas layanan di SMK Negeri 22 Jakarta menjadi lebih baik. Pada parameter *throughput* mengalami kenaikan 2.7746% pada perangkat pengukuran *cnn.com* dan 13.3415% pada *facebook.com*. Pada parameter *packet loss* mengalami penurunan 13% pada *cnn.com*, 17% pada *facebook.com* dan 16% pada *mail.yahoo.com*. Pada parameter *delay* mengalami penurunan dari yang sebelumnya 135 ms menjadi 41 ms pada *facebook.com* dan 108 ms menjadi 90 ms pada *mail.yahoo.com*. Pada indeks nilai akhir mengalami kenaikan dari 2.33 menjadi 2.89.

5.2. Saran

1. Penulis berharap untuk penelitian selanjutnya dapat menambahkan *MAC Address* sebagai tambahan untuk melakukan manajemen *bandwidth*. Jadi setiap *client* hanya bisa mengakses *internet* dengan satu *device*

2. Jika ingin dilakukan penambahan *bandwidth* maka dapat dilakukan dengan 2 cara. Cara yang pertama yaitu dengan menambah jumlah *bandwidth* yang disewa. Cara ini memerlukan biaya yang cukup besar. Cara yang kedua adalah dengan menggunakan metode *load balance*. *Load balance* adalah teknik untuk mendistribusikan beban trafik pada dua atau lebih jalur koneksi secara seimbang. Metode *load balance* dapat memaksimalkan *throughput*, memperkecil waktu tanggap dan menghindari *overload* pada salah satu jalur koneksi. Cara ini membutuhkan dana yang tidak terlalu besar.

DAFTAR PUSTAKA

- Akbar Rizki, Muhammad. 2017. Analisis Kualitas Layanan (Quality of Service) Mikrotik Routerboard Pada Jaringan Internet SMK Negeri 22 Jakarta[skripsi]. Jakarta: Fakultas Teknik, Universitas Negeri Jakarta.
- Akmal, Adrian, Fitri Susanti dan M Idham Iskandar. Konfigurasi dan Analisis Manajemen Bandwidth pada PC Router Menggunakan Metode HTB (Hierarchy Token Bucket) dan CBQ (Class Based Queue) Studi Kasus Kantor Badan Pertanahan Nasional Bukit Tinggi. Politeknik Telkom Bandung.
- Arifin, Yunus. (2012). Implementasi Quality Of Service Dengan Metode HTB (Hierarchical Token Bucket) Pada PT. Komunika Lima Duabelas. Universitas Udayana.
- Darmawan, E., Purnama, I., Mahardika, T. I. R., & Wicaksana, I. W. S. (2012). Bandwidth Manajemen Queue Tree VS Simple Queue. Fakultas Ilmu Komputer, Universitas Gunadarma.
- Daniel, Kustanto T dan Saputro. 2009. *Membangun Server Internet dengan Mikrotik OS*. Jakarta. Gaya Media.
- Gunawan, Arif Hamdani 2008, Quality of Service dalam Data Komunikasi
- Irfan, Mochamad dan Periyadi, S.T. (2010). Penerapan Bandwidth Management Menggunakan Metode HTB (Hierarchical Token Bucket) Di PT. Neuronworks. Politeknik Telkom Bandung.
- Kencana, Surya. 2012. Implementasi Algoritma Per Connection Queue (PCQ) Dalam Algoritma Hierarchical Token Bucket (HTB) Untuk Pembagian Bandwidth Pada Warnet Khelambiqunet. Bandung. Jurnal Politeknik Telkom Bandung.
- Nababan, Sabar Saut Martua 2013. Implementasi Bandwidth Management Dan Pengaturan Akses Menggunakan Mikrotik Router OS[skripsi]. Bandung: Fakultas Teknik, Universitas Widyatama.
- Riyadi, Valens. (2012). Implementasi QoS pada Mikrotik Diakses pada tanggal 17 mei 2017, dari (http://www.mikrotik.co.id/artikel_lihat.php?id=29).
- Santosa, B. 2007. Manajemen Bandwidth Internet dan Intranet.
- Sofana, Iwan. 2009. *CISCO CCNA dan jaringan komputer*. Bandung: Informatika.
- Syafrizal, Melwin. 2000. *Pengantar Jaringan Komputer*. Yogyakarta: ANDI Yogyakarta.

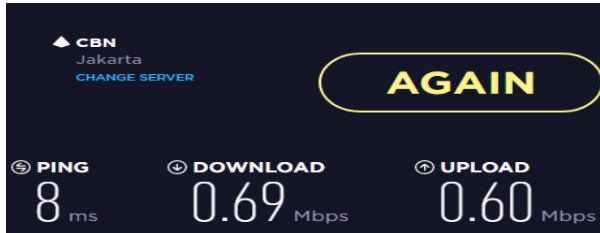
Tiphon. "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) General aspects of Quality of Service (QoS)", DTR/TIPHON-05006 (cb0010cs.PDF).1999.

Wijaya, A. I. & Handoko, L. B. (2013). Manajemen Bandwidth Dengan Metode HTB (Hierarchical Token Bucket) Pada Sekolah Menengah Pertama Negeri 5 Semarang. Jurnal Teknik Informatika Universitas Dian Nuswantoro.

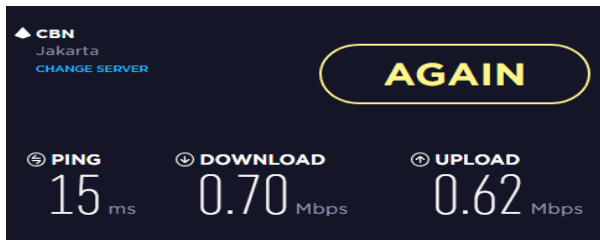
Yanto. Analisis QoS (*Quality of Service*) Pada Jaringan Internet (Studi Kasus: Fakultas Teknik Universitas Tanjungpura)[skripsi]. Tanjungpura : Jurusan Teknik Elektro, Fakultas Teknik, Universitas Tanjungpura.

LAMPIRAN

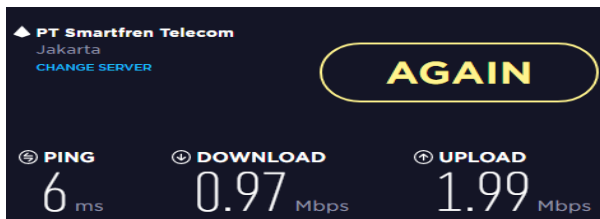
1. Hasil Pengukuran Bandwidth



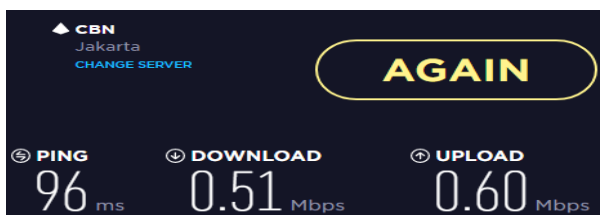
Senin, 17 April 2017



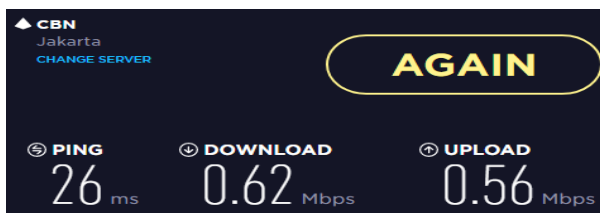
Selasa, 18 April 2017



Rabu, 19 April 2017

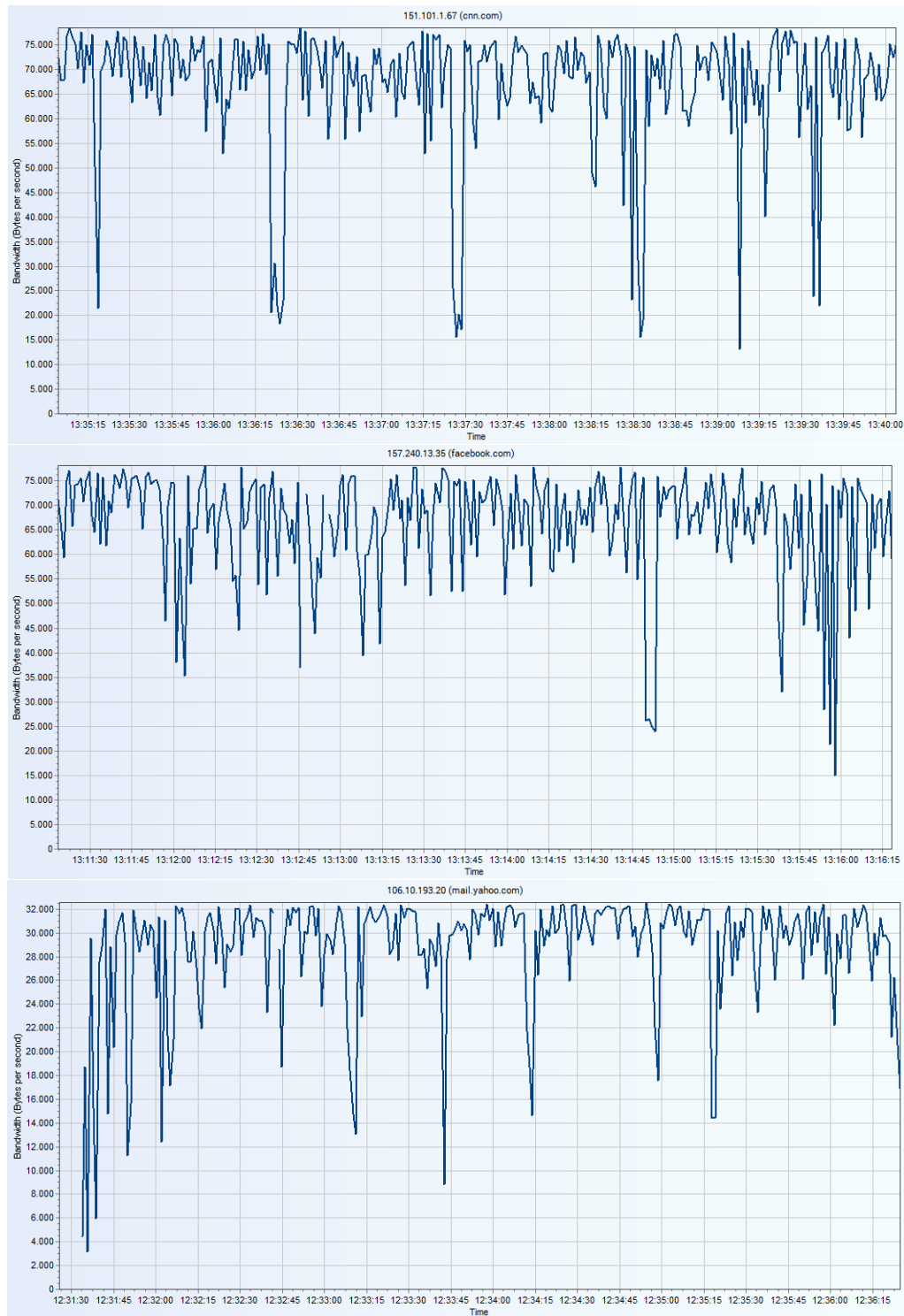


Kamis, 20 April 2017

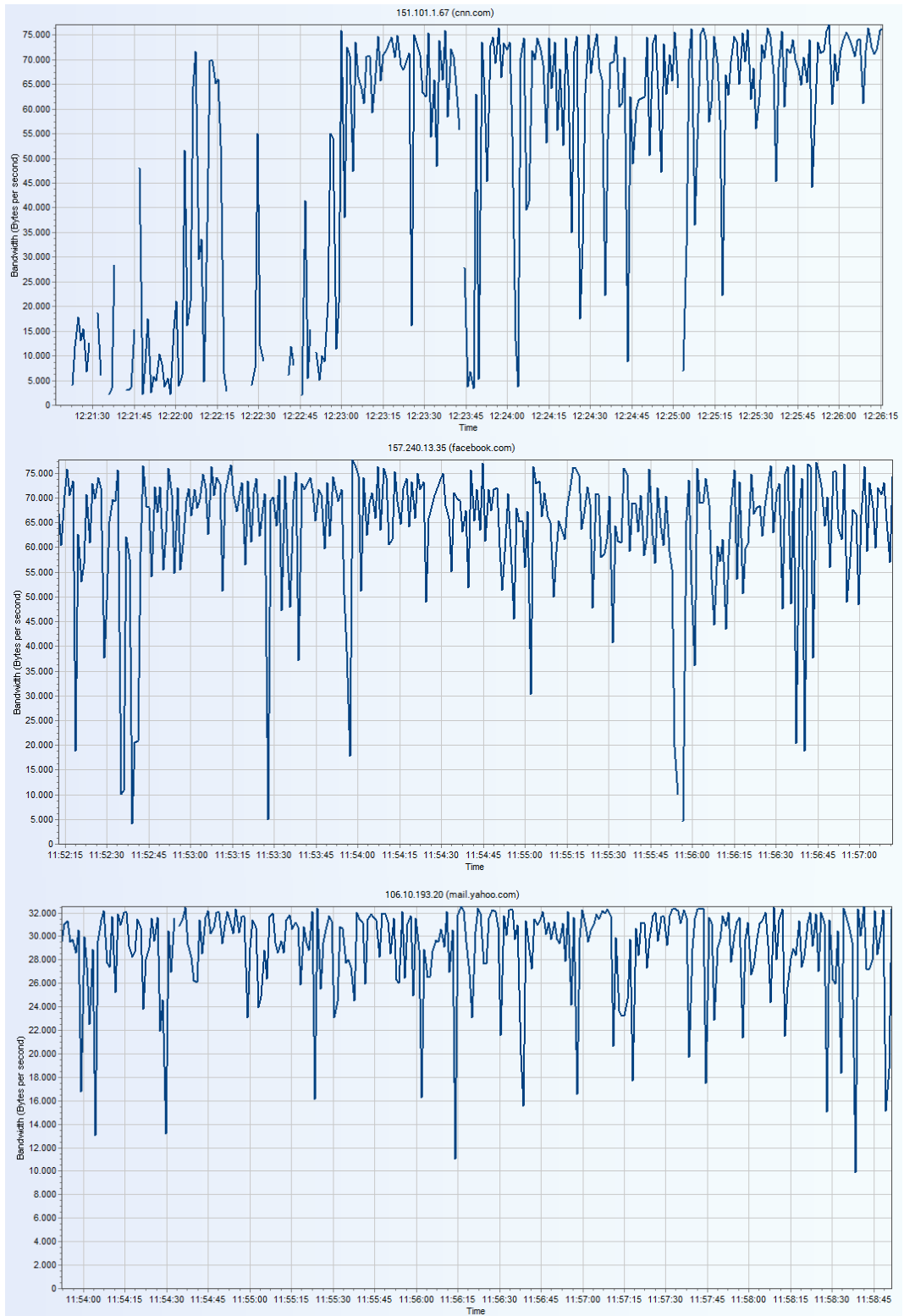


Jumat, 21 April 2017

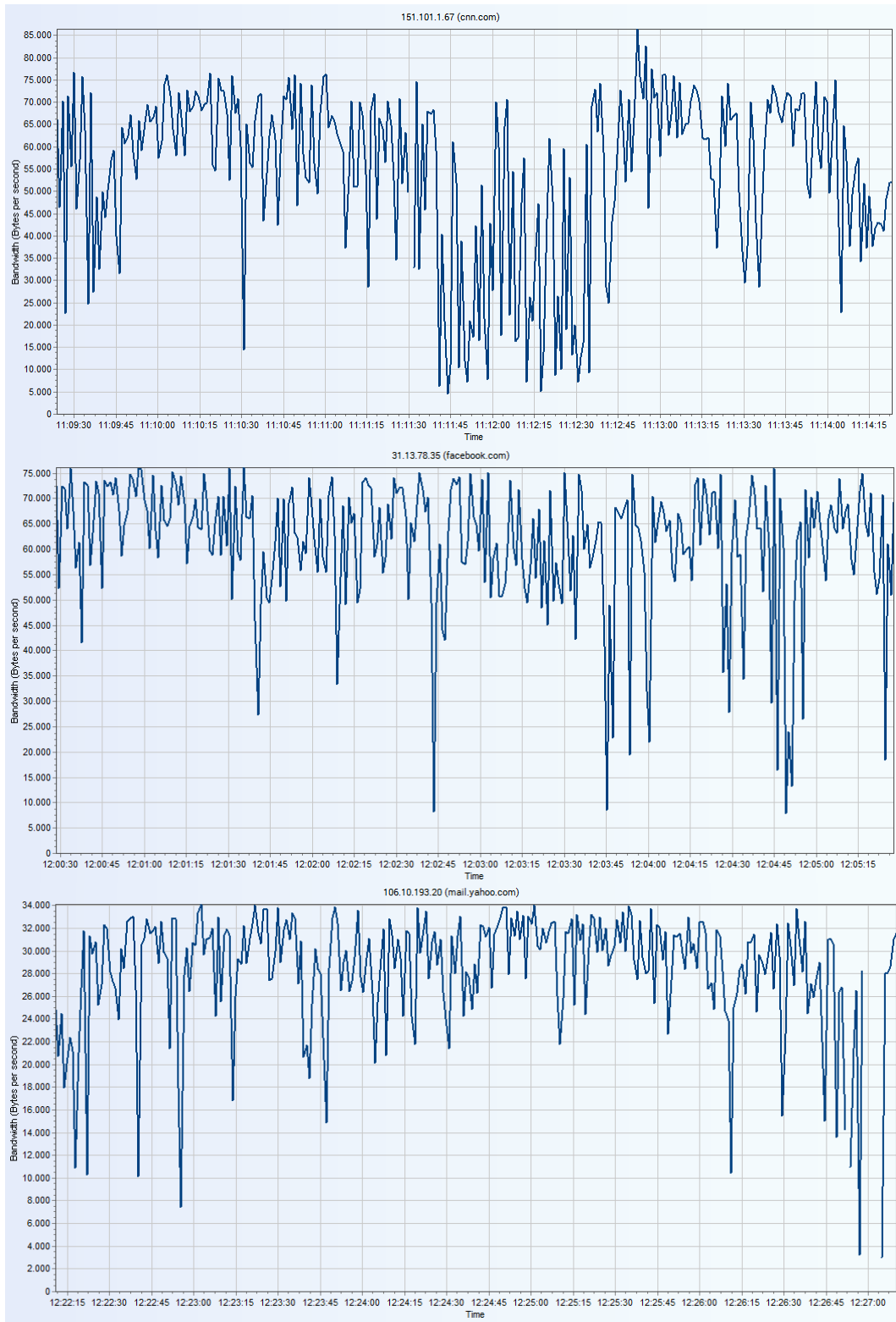
2. Hasil Pengukuran Throughput



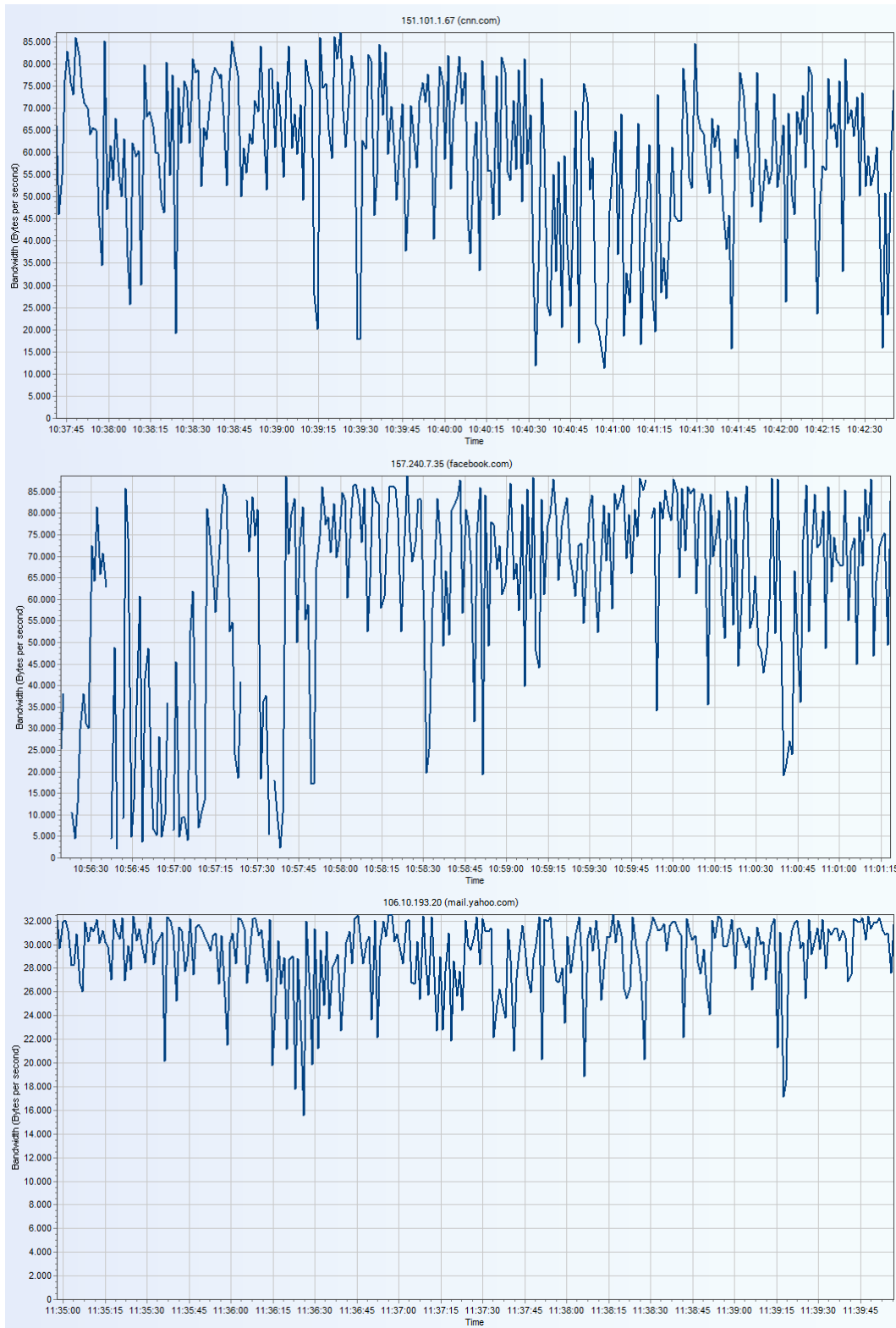
Senin, 17 April 2017



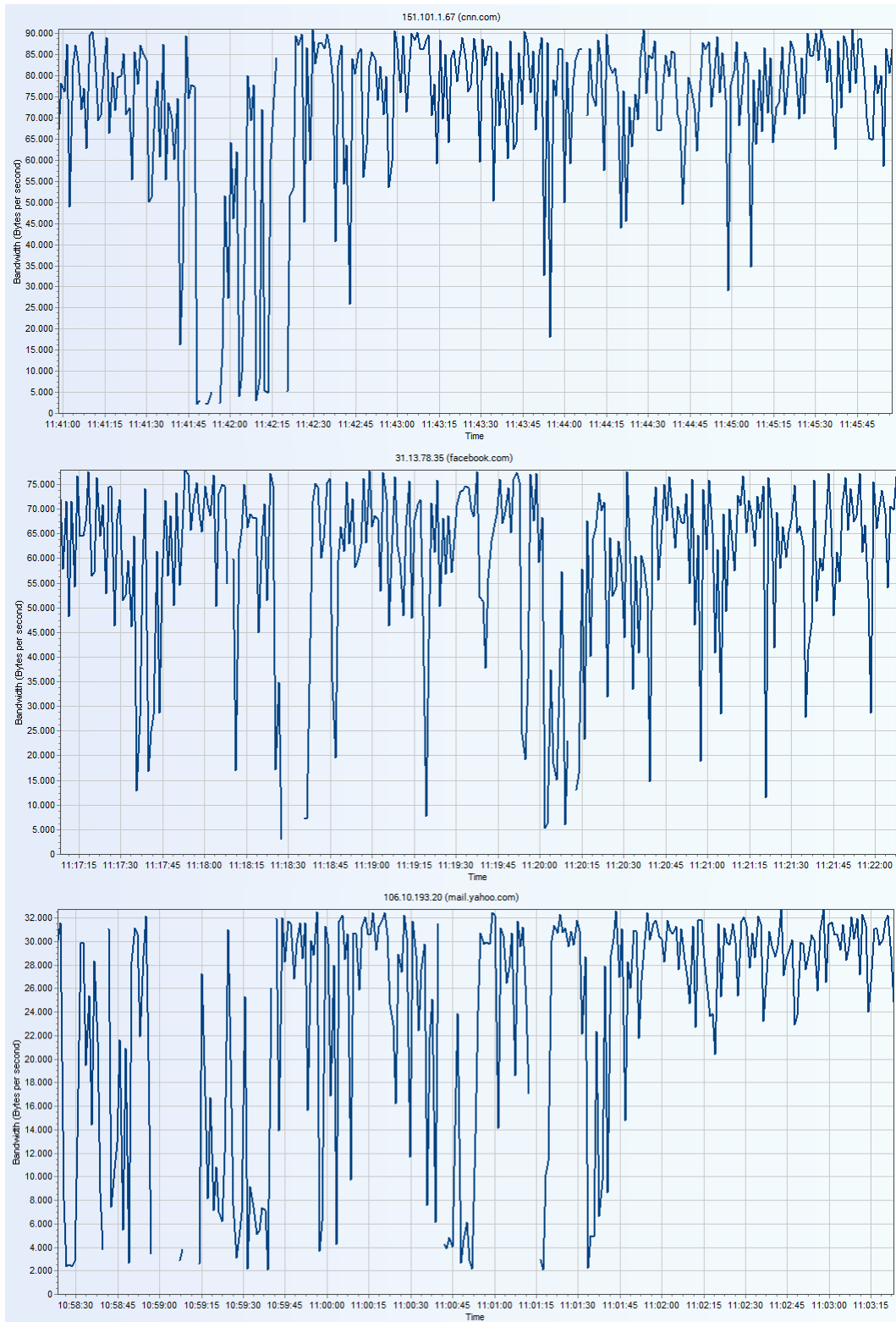
Selasa, 18 April 2017



Rabu, 19 April 2017



Kamis, 20 April 2017



Jumat, 21 April 2017

Address: cnn.com		Address: facebook.com	
Information ^		Information ^	
Bandwidth:		Bandwidth:	
Average	527 346 bit/s	Average	536 148 bit/s
Minimum	104 560 bit/s	Minimum	120 048 bit/s
Maximum	628 112 bit/s	Maximum	625 024 bit/s
Packets:		Packets:	
Sent	1 271	Sent	1 209
Received	1 271 (100 %)	Received	1 200 (99 %)
Lost	0 (0 %)	Lost	9 (1 %)
Options ^		Options ^	

Address: mail.yahoo.com	
Information ^	
Bandwidth:	
Average	233 046 bit/s
Minimum	18 200 bit/s
Maximum	260 840 bit/s
Packets:	
Sent	1 088
Received	1 050 (97 %)
Lost	38 (3 %)
Options ^	

Senin, 17 April 2017

Address: cnn.com		Address: facebook.com	
Information ^		Information ^	
Bandwidth:		Bandwidth:	
Average	486 252 bit/s	Average	478 719 bit/s
Minimum	16 976 bit/s	Minimum	16 520 bit/s
Maximum	625 752 bit/s	Maximum	621 688 bit/s
Packets:		Packets:	
Sent	1 619	Sent	1 474
Received	1 555 (96 %)	Received	1 432 (97 %)
Lost	64 (4 %)	Lost	42 (3 %)
Options ^		Options ^	

Address: mail.yahoo.com	
Information ^	
Bandwidth:	
Average	230 513 bit/s
Minimum	24 512 bit/s
Maximum	260 336 bit/s
Packets:	
Sent	1 080
Received	1 077 (100 %)
Lost	3 (0 %)
Options ^	

Selasa, 18 April 2017

Address: <input type="text" value="cnn.com"/>		Address: <input type="text" value="facebook.com"/>	
Information ^		Information	
Bandwidth:		Bandwidth:	
Average	403 313 bit/s	Average	465 168 bit/s
Minimum	16 864 bit/s	Minimum	16 584 bit/s
Maximum	691 136 bit/s	Maximum	612 288 bit/s
Packets:		Packets:	
Sent	1 453	Sent	1 317
Received	1 358 (93 %)	Received	1 300 (99 %)
Lost	95 (7 %)	Lost	17 (1 %)
Options ^		Options	
Address: <input type="text" value="mail.yahoo.com"/>			
Information ^			
Bandwidth:			
Average	235 296 bit/s		
Minimum	24 040 bit/s		
Maximum	274 176 bit/s		
Packets:			
Sent	1 251		
Received	1 245 (100 %)		
Lost	6 (0 %)		
Options ^			

Rabu, 19 April 2017

Address: <input type="text" value="cnn.com"/>		Address: <input type="text" value="facebook.com"/>	
Information ^		Information ^	
Bandwidth:		Bandwidth:	
Average	395 170 bit/s	Average	529 243 bit/s
Minimum	18 136 bit/s	Minimum	17 752 bit/s
Maximum	705 576 bit/s	Maximum	717 200 bit/s
Packets:		Packets:	
Sent	1 080	Sent	1 080
Received	1 041 (96 %)	Received	1 068 (99 %)
Lost	39 (4 %)	Lost	12 (1 %)
Options ^		Options ^	
Address: <input type="text" value="mail.yahoo.com"/>			
Information ^			
Bandwidth:			
Average	235 291 bit/s		
Minimum	52 664 bit/s		
Maximum	260 920 bit/s		
Packets:			
Sent	1 080		
Received	1 080 (100 %)		
Lost	0 (0 %)		
Options ^			

Kamis, 20 April 2017

Address: cnn.com		Address: facebook.com	
Information ^		Information ^	
Bandwidth:		Bandwidth:	
Average	591 685 bit/s	Average	494 996 bit/s
Minimum	17 048 bit/s	Minimum	16 528 bit/s
Maximum	734 416 bit/s	Maximum	625 736 bit/s
Packets:		Packets:	
Sent	1 264	Sent	1 080
Received	1 255 (99 %)	Received	1 040 (96 %)
Lost	9 (1 %)	Lost	40 (4 %)
Options ^		Options ^	
Address: mail.yahoo.com			
Information ^			
Bandwidth:			
Average		215 133 bit/s	
Minimum		16 560 bit/s	
Maximum		261 880 bit/s	
Packets:			
Sent		1 080	
Received		1 054 (98 %)	
Lost		26 (2 %)	
Options ^			

Jumat, 21 April 2017

3. Hasil Pengukuran Delay

Host						Response Time (ms)				
* Status	Name	IP	▲ DNS Name	Location	Service	Last	Avg	Min	Max	
▶	www.faceb...	31.13.78.35	www.facebook.com	United States	PING	25	27	21	229	
▶	www.yaho...	106.10.178...	www.yahoo.com	Singapore	PING	24	89	22	273	
▶	mail.yahoo...	106.10.193...	mail.yahoo.com	Singapore	PING	57	110	54	424	
▶	www.cnn.c...	151.101.1.67	www.cnn.com	United States	PING	32	27	21	156	

Senin, 17 April 2017

Host						Response Time (ms)				
* Status	Name	IP	▲ DNS Name	Location	Service	Last	Avg	Min	Max	
▶	www.faceb...	31.13.78.35	www.facebook.com	United States	PING	23	48	21	971	
▶	www.yaho...	106.10.178...	www.yahoo.com	Singapore	PING	23	101	22	931	
▶	mail.yahoo...	106.10.193...	mail.yahoo.com	Singapore	PING	58	122	55	973	
▶	www.cnn.c...	151.101.9.67	www.cnn.com	United States	PING	24	52	22	996	

Selasa, 18 April 2017

Host						Response Time (ms)				
* Status	Name	IP	▲ DNS Name	Location	Service	Last	Avg	Min	Max	
▶	mail.yahoo...	98.136.189...	mail.yahoo.com	United States	PING	220	136	54	527	
▶	www.yaho...	106.10.178...	www.yahoo.com	Singapore	PING	23	82	22	553	
▶	www.cnn.c...	151.101.1.67	www.cnn.com	United States	PING	24	26	22	566	
▶	www.faceb...	157.240.7.35	www.facebook.com	Singapore	PING	27	26	21	585	

Rabu, 19 April 2017

Host						Response Time (ms)				
* Status	Name	IP	▲ DNS Name	Location	Service	Last	Avg	Min	Max	
▶	www.faceb...	31.13.78.35	www.facebook.com	United States	PING	24	39	24	209	
▶	www.yaho...	106.10.138...	www.yahoo.com	Singapore	PING	27	27	22	239	
▶	mail.yahoo...	106.10.193...	mail.yahoo.com	Singapore	PING	25	26	22	124	
▶	www.cnn.c...	151.101.53...	www.cnn.com	United States	PING	213	216	211	353	

Kamis, 20 April 2017

Host						Response Time (ms)				
* Status	Name	IP	▲ DNS Name	Location	Service	Last	Avg	Min	Max	
▶	www.yaho...	106.10.178...	www.yahoo.com	Singapore	PING	31	186	26	991	
▶	mail.yahoo...	106.10.193...	mail.yahoo.com	Singapore	PING	63	109	62	999	
▶	www.cnn.c...	151.101.9.67	www.cnn.com	United States	PING	30	184	22	997	
▶	www.faceb...	157.240.13...	www.facebook.com	Singapore	PING	25	66	22	998	

Jumat, 21 April 2017

4. Hasil Pengukuran Packet Loss

Packets		
Sent	Lost	% Lost
1001	1	0
1001	2	0
1000	1	0
1001	2	0

Senin, 17 April 2017

Packets		
Sent	Lost	% Lost
1104	67	6
1104	71	6
1103	74	7
1105	66	6

Selasa, 18 April 2017

Packets		
Sent	Lost	% Lost
1119	1	0
1119	1	0
1120	1	0
1120	0	0

Rabu, 19 April 2017

Packets		
Sent	Lost	% Lost
1000	28	3
966	27	3
1000	138	14
1000	32	3

Kamis, 20 April 2017

Packets		
Sent	Lost	% Lost
999	27	3
1000	18	2
998	27	3
1000	17	2

Jumat, 21 April 2017

5. Konfigurasi Marking Connection Untuk 40 User

Perintah untuk konfigurasi marking connection dengan new terminal adalah :

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-address=192.168.1.2 in-interface=ether2-lokal action=mark-connection new-connection-mark=1-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-address=192.168.1.3 in-interface=ether2-lokal action=mark-connection new-connection-mark=2-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-address=192.168.1.4 in-interface=ether2-lokal action=mark-connection new-connection-mark=3-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-address=192.168.1.5 in-interface=ether2-lokal action=mark-connection new-connection-mark=4-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-address=192.168.1.6 in-interface=ether2-lokal action=mark-connection new-connection-mark=5-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-address=192.168.1.7 in-interface=ether2-lokal action=mark-connection new-connection-mark=6-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-address=192.168.1.8 in-interface=ether2-lokal action=mark-connection new-connection-mark=7-con passthrough=yes
```



```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.9 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=8-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.10 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=9-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.11 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=10-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.12 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=11-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.13 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=12-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.14 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=13-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.15 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=14-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.16 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=15-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.17 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=16-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.18 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=17-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.19 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=17-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.20 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=19-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.21 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=20-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.22 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=21-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.23 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=22-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.24 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=23-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.25 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=24-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.26 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=25-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.27 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=26-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.28 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=27-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.29 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=28-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.30 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=29-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.31 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=30-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.32 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=31-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.33 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=32-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.34 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=33-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.35 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=34-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.36 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=35-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.37 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=36-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.38 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=37-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.39 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=38-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.40 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=39-con passthrough=yes
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting src-  
address=192.168.1.41 in-interface=ether2-lokal action=mark-connection new-  
connection-mark=40-con passthrough=yes
```

6. Konfigurasi Marking Packet Untuk 40 User

Perintah untuk konfigurasi marking connection dengan new terminal adalah :

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-mark=1-con action=mark-packet new-packet-mark=1 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-mark=2-con action=mark-packet new-packet-mark=2 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-mark=3-con action=mark-packet new-packet-mark=3 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-mark=4-con action=mark-packet new-packet-mark=4 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-mark=5-con action=mark-packet new-packet-mark=5 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-mark=6-con action=mark-packet new-packet-mark=6 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-mark=7-con action=mark-packet new-packet-mark=7 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-mark=8-con action=mark-packet new-packet-mark=8 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-mark=9-con action=mark-packet new-packet-mark=9 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-mark=10-con action=mark-packet new-packet-mark=10 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=11-con action=mark-packet new-packet-mark=11 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=12-con action=mark-packet new-packet-mark=12 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=13-con action=mark-packet new-packet-mark=13 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=14-con action=mark-packet new-packet-mark=14 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=15-con action=mark-packet new-packet-mark=15 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=16-con action=mark-packet new-packet-mark=16 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=17-con action=mark-packet new-packet-mark=17 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=18-con action=mark-packet new-packet-mark=18 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=19-con action=mark-packet new-packet-mark=19 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=20-con action=mark-packet new-packet-mark=20 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=21-con action=mark-packet new-packet-mark=21 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=22-con action=mark-packet new-packet-mark=22 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=23-con action=mark-packet new-packet-mark=23 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=24-con action=mark-packet new-packet-mark=24 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=25-con action=mark-packet new-packet-mark=25 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=26-con action=mark-packet new-packet-mark=26 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=27-con action=mark-packet new-packet-mark=27 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=28-con action=mark-packet new-packet-mark=28 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=29-con action=mark-packet new-packet-mark=29 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=30-con action=mark-packet new-packet-mark=30 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=31-con action=mark-packet new-packet-mark=31 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=32-con action=mark-packet new-packet-mark=32 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=33-con action=mark-packet new-packet-mark=33 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=34-con action=mark-packet new-packet-mark=34 passthrough=no
```



```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=35-con action=mark-packet new-packet-mark=35 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=36-con action=mark-packet new-packet-mark=36 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=37-con action=mark-packet new-packet-mark=37 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=38-con action=mark-packet new-packet-mark=38 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=39-con action=mark-packet new-packet-mark=39 passthrough=no
```

```
[admin@MikroTik] >ip firewall mangle add chain=prerouting connection-  
mark=40-con action=mark-packet new-packet-mark=40 passthrough=no
```

7. Konfigurasi Child Queue Traffic Download

Perintah untuk konfigurasi child queue traffic download dengan new terminal adalah :

```
[admin@MikroTik] >queue tree add name=1-download parent=all-download  
ruang guru packet-mark=1 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=2-download parent=all-download  
ruang guru packet-mark=2 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=3-download parent=all-download  
ruang guru packet-mark=3 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=4-download parent=all-download  
ruang guru packet-mark=4 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=5-download parent=all-download  
ruang guru packet-mark=5 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=6-download parent=all-download  
ruang guru packet-mark=6 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=7-download parent=all-download  
ruang guru packet-mark=7 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=8-download parent=all-download  
ruang guru packet-mark=8 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=9-download parent=all-download  
ruang guru packet-mark=9 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=10-download parent=all-download  
ruang guru packet-mark=10 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=11-download parent=all-download  
ruang guru packet-mark=11 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=12-download parent=all-download  
ruang guru packet-mark=12 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=13-download parent=all-download  
ruang guru packet-mark=13 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=14-download parent=all-download  
ruang guru packet-mark=14 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=15-download parent=all-download  
ruang guru packet-mark=15 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=16-download parent=all-download  
ruang guru packet-mark=16 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=17-download parent=all-download  
ruang guru packet-mark=17 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=18-download parent=all-download  
ruang guru packet-mark=18 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=19-download parent=all-download  
ruang guru packet-mark=19 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=20-download parent=all-download  
ruang guru packet-mark=20 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=21-download parent=all-download  
ruang guru packet-mark=21 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=22-download parent=all-download  
ruang guru packet-mark=22 limit-at=100k max-limit=3M
```

[admin@MikroTik] >queue tree add name=23-download parent=all-download
ruang guru packet-mark=23 limit-at=100k max-limit=3M

[admin@MikroTik] >queue tree add name=24-download parent=all-download
ruang guru packet-mark=24 limit-at=100k max-limit=3M

[admin@MikroTik] >queue tree add name=25-download parent=all-download
ruang guru packet-mark=25 limit-at=100k max-limit=3M

[admin@MikroTik] >queue tree add name=26-download parent=all-download
ruang guru packet-mark=26 limit-at=100k max-limit=3M

[admin@MikroTik] >queue tree add name=27-download parent=all-download
ruang guru packet-mark=27 limit-at=100k max-limit=3M

[admin@MikroTik] >queue tree add name=28-download parent=all-download
ruang guru packet-mark=28 limit-at=100k max-limit=3M

[admin@MikroTik] >queue tree add name=29-download parent=all-download
ruang guru packet-mark=29 limit-at=100k max-limit=3M

[admin@MikroTik] >queue tree add name=30-download parent=all-download
ruang guru packet-mark=30 limit-at=100k max-limit=3M

[admin@MikroTik] >queue tree add name=31-download parent=all-download
ruang guru packet-mark=31 limit-at=100k max-limit=3M

[admin@MikroTik] >queue tree add name=32-download parent=all-download
ruang guru packet-mark=32 limit-at=100k max-limit=3M

[admin@MikroTik] >queue tree add name=33-download parent=all-download
ruang guru packet-mark=33 limit-at=100k max-limit=3M

[admin@MikroTik] >queue tree add name=34-download parent=all-download
ruang guru packet-mark=34 limit-at=100k max-limit=3M

```
[admin@MikroTik] >queue tree add name=35-download parent=all-download  
ruang guru packet-mark=35 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=36-download parent=all-download  
ruang guru packet-mark=36 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=37-download parent=all-download  
ruang guru packet-mark=37 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=38-download parent=all-download  
ruang guru packet-mark=38 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=39-download parent=all-download  
ruang guru packet-mark=39 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=40-download parent=all-download  
ruang guru packet-mark=40 limit-at=100k max-limit=3M
```

8. Konfigurasi Child Queue Traffic Upload

Perintah untuk konfigurasi child queue traffic upload dengan new terminal adalah :

```
[admin@MikroTik] >queue tree add name=1-upload parent=all-upload ruang  
guru packet-mark=1 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=2-upload parent=all-upload ruang  
guru packet-mark=2 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=3-upload parent=all-upload ruang  
guru packet-mark=3 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=4-upload parent=all-upload ruang  
guru packet-mark=4 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=5-upload parent=all-upload ruang  
guru packet-mark=5 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=6-upload parent=all-upload ruang  
guru packet-mark=6 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=7-upload parent=all-upload ruang  
guru packet-mark=7 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=8-upload parent=all-upload ruang  
guru packet-mark=8 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=9-upload parent=all-upload ruang  
guru packet-mark=9 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=10-upload parent=all-upload ruang  
guru packet-mark=10 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=11-upload parent=all-upload ruang
guru packet-mark=11 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=12-upload parent=all-upload ruang
guru packet-mark=12 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=13-upload parent=all-upload ruang
guru packet-mark=13 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=14-upload parent=all-upload ruang
guru packet-mark=14 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=15-upload parent=all-upload ruang
guru packet-mark=15 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=16-upload parent=all-upload ruang
guru packet-mark=16 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=17-upload parent=all-upload ruang
guru packet-mark=17 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=18-upload parent=all-upload ruang
guru packet-mark=18 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=19-upload parent=all-upload ruang
guru packet-mark=19 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=20-upload parent=all-upload ruang
guru packet-mark=20 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=21-upload parent=all-upload ruang
guru packet-mark=21 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=22-upload parent=all-upload ruang
guru packet-mark=22 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=23-upload parent=all-upload ruang  
guru packet-mark=23 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=24-upload parent=all-upload ruang  
guru packet-mark=24 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=25-upload parent=all-upload ruang  
guru packet-mark=25 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=26-upload parent=all-upload ruang  
guru packet-mark=26 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=27-upload parent=all-upload ruang  
guru packet-mark=27 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=28-upload parent=all-upload ruang  
guru packet-mark=28 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=29-upload parent=all-upload ruang  
guru packet-mark=29 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=30-upload parent=all-upload ruang  
guru packet-mark=30 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=31-upload parent=all-upload ruang  
guru packet-mark=31 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=32-upload parent=all-upload ruang  
guru packet-mark=32 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=33-upload parent=all-upload ruang  
guru packet-mark=33 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=34-upload parent=all-upload ruang  
guru packet-mark=34 limit-at=100k max-limit=3M
```



```
[admin@MikroTik] >queue tree add name=35-upload parent=all-upload ruang  
guru packet-mark=35 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=36-upload parent=all-upload ruang  
guru packet-mark=36 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=37-upload parent=all-upload ruang  
guru packet-mark=37 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=38-upload parent=all-upload ruang  
guru packet-mark=38 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=39-upload parent=all-upload ruang  
guru packet-mark=39 limit-at=100k max-limit=3M
```

```
[admin@MikroTik] >queue tree add name=40-upload parent=all-upload ruang  
guru packet-mark=40 limit-at=100k max-limit=3M
```

DAFTAR RIWAYAT HIDUP



Muhammad Iqbal Ichwan dilahirkan di Bekasi 25 Oktober 1994, merupakan anak pertama dari pasangan bapak Iswan dan ibu Setiawati. Penulis memulai pendidikan dasarnya pada tahun 2000 di SDN Pejuang 3 Bekasi dan mengakhiri pendidikan dasar pada tahun 2006. Kemudian penulis melanjutkan pendidikan ke tingkat menengah pertama di SMPN 256 Jakarta dan menyelesaikannya pada tahun 2009. Setelah itu, penulis meneruskan pendidikan jenjang SMA di SMAN 89 Jakarta dan lulus pada tahun 2012. Pada tahun yang sama, penulis melanjutkan pendidikan S1 di Universitas Negeri Jakarta Fakultas Teknik pada Program Studi Pendidikan Teknik Informatika dan Komputer dan pada semester 5 mengambil peminatan Teknik Komputer Jaringan. Penulis telah mengikuti PKL (Praktik Kerja Lapangan) di Lembaga Penerbangan dan Antariksa Nasional (LAPAN) dan juga telah mengikuti PKM (Praktik Keterampilan Mengajar) di SMKT KAPIN Jakarta Timur. Selama masa kuliah penulis berdomisili di Jl. Perintis 6 RT 05/08 No 25 X, Pejuang Jaya, Bekasi. Pada akhir perkuliahan, penulis melakukan penelitian skripsi dengan judul “Analisis Penerapan Manajemen Bandwidth Hierarchical Token Bucket (HTB) Dengan Mikrotik Pada Jaringan SMK Negeri 22 Jakarta”.