

# ANALISA *BANDWIDTH* MENGGUNAKAN METODE ANTRIAN *Per Connection Queue*

Sukri<sup>1</sup>, Jumiaty<sup>2</sup>

Jurusan Teknik Informatika, Fakultas Teknik, Universitas Abdurrah

Jl. Riau Ujung no.73 Pekanbaru

Telp. (0761) 38762.

E-mail: ocu.sukri@yahoo.com, jumiatijumy@gmail.com

## ABSTRAK

Jaringan *internet* yang ada saat ini di beberapa tempat seperti warnet (warung *internet*), sering kali terjadi adanya dominasi *bandwidth* antar *client* yang diakibatkan salah satu atau beberapa *client* melakukan *download* sehingga akan mengganggu *client* lain. Untuk mengatasi permasalahan dominasi *bandwidth* antar *client* yang terjadi, maka dilakukanlah pembagian *bandwidth*. Dengan memanfaatkan *router network*, administrator jaringan dapat dengan mudah melakukan manajemen *bandwidth* terhadap komputer *client* yang terhubung dalam jaringan. Teknologi yang digunakan untuk mengimplementasikannya didasarkan pada pendekatan yang disebut QoS (*Quality of Service*) dan disesuaikan dengan standar TIPHON. Cara kerjanya yaitu dengan mengidentifikasi lewat lalu lintas data yang melalui jaringan, kemudian menerapkan kebijakan QoS yang digunakan untuk melindungi dan memprioritaskannya. Tujuan yang hendak dicapai oleh penulis adalah menyeimbangkan alokasi *bandwidth* sesuai untuk kebutuhan *user*, menerapkan aturan antrian pada lalu lintas data agar tidak terjadi antrian. Metode antrian yang digunakan dalam penelitian ini yaitu menggunakan metode *Per Connection Queue* dan *winbox* sebagai *software* untuk *remote router*.

Kata Kunci : *Bandwidth, PCQ, Simple Queue, Queue Tree, Quality Of Service*

## ABSTRAK

*Internet network that exist today in some place like warnet (internet cafes), often happened the dominance of bandwidth between client caused by one or some client to download so that will disturb other client. To overcome the problem of bandwidth dominance between clients that occur, then the bandwidth division is done. By utilizing network routers, network administrators can easily perform bandwidth management of client computers connected in the network. The technology used to implement it is based on an approach called QoS (Quality of Service) and adapted to TIPHON standards. How it works is to identify through data traffic through the network, then apply the QoS policies used to protect and prioritize them. Objectives to be achieved by the author is to balance the bandwidth allocation according to the needs of the user, apply the rules queue in the data traffic in order to avoid the queue. The queuing method used in this research is using Per Connection Queue method and winbox as software for remote router.*

Keywords : *Bandwidth, PCQ, Simple Queue, Queue Tree, Quality Of Service*

## I. PENDAHULUAN

### 1. Latar Belakang

Perkembangan teknologi informasi yang sangat pesat telah membuat banyak perubahan bagi kehidupan manusia saat ini. Hal tersebut ditandai dengan perkembangan teknologi berbagai perangkat keras maupun perangkat lunak yang telah membawa dampak yang cukup besar dalam hal penyajian informasi. Penyajian informasi menjadi lebih cepat, tepat dan akurat tanpa dibatasi oleh ruang dan waktu.

Saat ini jaringan komputer bukanlah sesuatu yang baru, karena hampir setiap instansi menggunakan jaringan komputer untuk memperlancar arus informasi dalam instansi tersebut. Seiring dengan berkembangnya jaringan komputer, *traffic internet* menjadi sangatlah padat. Maka dari itu seorang administrator jaringan harus bisa mengelola *bandwidth*.

Penggunaan *bandwidth* disebuah jaringan bukan hanya dipengaruhi oleh banyaknya *user*, namun juga dipengaruhi oleh jenis serta tingkat kebutuhan pengiriman dan penerimaan (*upload*

dan *download*). Selain itu juga *bandwidth* tersebut seringkali kurang dimanfaatkan secara optimal. Hal ini dapat disebabkan oleh adanya satu atau lebih *client* yang menghabiskan kapasitas *bandwidth* dalam jaringan tersebut untuk mengunduh atau untuk mengakses aplikasi-aplikasi yang dapat menyita kapasitas *bandwidth*.

Jaringan *internet* yang ada saat ini diberberapa tempat seperti warnet (warung *internet*), sering kali terjadi adanya dominasi *bandwidth* antar *client* yang diakibatkan salah satu atau beberapa *client* melakukan *download* sehingga akan mengganggu *client* lain. Salah satu solusi agar *bandwidth* dapat dimanfaatkan lebih optimal adalah dengan mengelola *bandwidth* yang tersedia dalam jaringan tersebut. Dengan demikian, jika ada *client* yang mengakses *internet* yang membutuhkan kapasitas *bandwidth* yang besar, maka *client* lain tidak akan terganggu, karena tiap *client* sudah mempunyai kapasitas *bandwidth* masing-masing yang dapat dipakai untuk mengakses *internet*.

Dengan menggunakan *router mikrotik*, seorang administrator jaringan dapat dengan mudah melakukan manajemen *bandwidth*, namun demikian di dalam *router mikrotik* terdapat beberapa metode antrian yang bisa digunakan untuk melakukan *management bandwidth*, yang diantaranya memiliki kelebihan dan kekurangan dari masing-masing metode antrian.

Untuk mengatasi permasalahan dominasi *bandwidth* antar *client* yang terjadi, maka dilakukanlah pembagian *bandwidth*. Salah satu metode antrian yang digunakan untuk pembagian *bandwidth* yaitu menggunakan metode antrian *Per Connection Queue*, baik dengan menggunakan fitur *Simple Queue* maupun *Queue Tree* yang tersedia dalam *mikrotik*. Metode ini dapat dilakukan untuk kondisi beberapa *client* dan sangat merepotkan jika harus membuat *rule*, sehingga *Per Connection Queue* ini dapat membatasi *bandwidth user* secara merata dalam meningkatkan manajemen jaringan.

Berdasarkan latar belakang yang diuraikan di atas, maka penulis ingin merealisasikan hal tersebut dengan mengajukan judul Tugas Akhir

**“Analisa *Bandwidth* Menggunakan Metode Antrian *Per Connection Queue*”.**

## **2. Perumusan Masalah**

Dalam penulisan Tugas Akhir ini, penulis merumuskan permasalahan yang akan dibahas, yaitu :

1. Bagaimana analisa sistem kerja metode antrian *Per Connection Queue* menggunakan *mikrotik routerboard* ?
2. Bagaimana melakukan konfigurasi serta mengimplementasikan metode antrian *Per Connection Queue* menggunakan *mikrotik routerboard* ?
3. Bagaimana melihat kualitas jaringan (QoS) pada *Simple Queue* dan *Queue Tree* dengan menggunakan metode antrian *Per Connection Queue* ?

## **3. Batasan Masalah**

Dalam penelitian ini penulis membatasi permasalahan yang akan dibahas yaitu hanya pada :

1. Analisa *bandwidth* dilakukan menggunakan metode antrian *Per Connection Queue* dengan dua tipe *queue* yaitu *Simple Queue* dan *Queue Tree*.
2. Analisa *bandwidth* dilakukan dengan *server* menggunakan *Mikrotik RB750*, serta aplikasi *winbox* untuk *remote router*.
3. Analisa *bandwidth* dilakukan pada saat proses *upload* dan *download* terhadap paket data dalam jaringan.
4. Analisa kualitas jaringan (QoS) pada *Simple Queue* dan *Queue Tree* dengan metode *Per Connection Queue*.
5. Penelitian hanya pada analisa *bandwidth* dengan tidak membahas aspek *security*.

## **4. Tujuan Penelitian**

Tujuan yang ingin dicapai penulis dari penelitian ini adalah :

1. Membangun jaringan komputer yang terkoneksi dengan *internet* dan memiliki kecepatan akses yang merata.
2. Mengetahui kecepatan akses yang dihasilkan dalam melakukan *upload* dan

*download* pada jaringan yang menggunakan metode antrian *Per Connetion Queue*, baik *Simple Queue* maupun *Queue Tree*.

3. Mengetahui penggunaan *bandwidth* dan intensitas *traffic* pada sebuah jaringan sehingga mengetahui unjuk kinerja jaringan.

## 5. Manfaat Penelitian

Adapun manfaat dari penelitian ini yaitu :

1. Dapat membagi *bandwidth* secara merata untuk masing-masing *client* yang terhubung dalam sebuah jaringan dengan metode *Per Connection Queue*.
2. Memperoleh *bandwidth* yang merata ketika mengakses *internet* untuk melakukan *upload* dan *download* pada jaringan komputer.
3. Dapat mengukur kualitas jaringan yang menggunakan metode antrian *Per Connection Queue*.

## II. TINJAUAN PUSTAKA

### 1. Bandwidth

Menurut Riadi (2010), *Bandwidth* merupakan suatu ukuran dari banyaknya informasi yang dapat mengalir dari suatu tempat ke tempat lain dalam suatu waktu tertentu. *Bandwidth* dapat dipakai untuk mengukur baik aliran data analog maupun aliran data digital. Sekarang *bandwidth* lebih banyak digunakan untuk mengukur aliran data digital. Satuan yang dipakai untuk *bandwidth* adalah *bits per second* atau sering disingkat sebagai bps. *Bit* atau *binery 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 satu tempat ketempat yang lain dalam setiap detik melalui suatu media.

*Bandwidth* (disebut juga *Data Transfer* atau *Trafik*) adalah kapasitas atau daya tampung kabel *Ethernet* agar dapat dilewati *traffic paket* data dalam jumlah tertentu. *Bandwidth* juga dikatakan data yang keluar-masuk (*upload-download*). Di dalam sistem jaringan komputer

dan berbagai jenis digital lainnya, defenisi *bandwidth* sering kali direfensikan sebagai *bitsper sekon*, contohnya jaringan (*network*). *Bandwidth* dapat dipakai untuk mengukur baik aliran data analog maupun data digital (Raharja, 2014).

*Bandwidth* salah satu konsep pengukuran yang sangat penting dalam jaringan, tetapi konsep ini memiliki kekurangan atau batasan, tidak peduli bagaimana cara *user* mengirim informasi maupun media apa yang dipakai dalam penghantaran informasi. Hal ini karena adanya hukum fisika maupun batasan teknologi dan akan menyebabkan batasan terhadap panjang media yang dipakai, kecepatan maksimal yang dapat dipakai, maupun perlakuan khusus terhadap media yang dipakai (Riadi, 2010).

**Tabel 1.** Batasan Panjang Medium dan Kecepatan Maksimum Aliran Data

Jenis Media	Panjang Maksimal	Kecepatan Maksimal
Kabel <i>Coaxial</i> 10 Base 2	200 m	10-100 Mbps
Kabel <i>Coaxial</i> 10 Base 2	500 m	10-100 Mbps
UTP 10 Base T	100 m	10 Mbps
UTP 100 Base TX	100 m	100 Mbps
<i>Multimode</i> 100 Base FX	2 Km	100 Mbps
<i>Single mode</i> 1000 Base LX	3 Km	1 Gbps
<i>Wireless</i>	100 m	2 Mbps
<i>Infra Red</i>	1 m	4 Mbps

Menurut Riadi (2010), *Trafik bandwidth* secara umum dikelompokkan menjadi dua jenis, yaitu :

1. *Up Stream* adalah *bandwidth* yang digunakan untuk mengirim data (misal mengirim *file* melalui *FTP* ke salah satu alamat jaringan).
2. *Down Stream* adalah *bandwidth* yang digunakan untuk menerima data (misal menerima *file* atau data dari satu alamat jaringan).

### 2. Manajemen Bandwidth

Menurut Septiawan (2013), *Bandwidth management* dapat diartikan sebagai proses

mengukur dan mengendalikan pertukaran informasi dalam jaringan komputer, sehingga dapat menghindari hal-hal yang tidak diinginkan yang berakibat pada *network congestion* dan penurunan kemampuan jaringan. Sebuah *manajemen bandwidth* yang baik harus dapat membuat dan menjaga aturan tentang ketersediaan koneksi (dalam hal ini *internet*).

Minimal *bandwidth* diartikan sebagai *bandwidth* yang ditetapkan untuk suatu *class* dalam suatu jaringan. Saat lalu lintas tinggi, *class* yang diberi dengan *bandwidth* minimal ini akan tetap mendapat jatahnya. *Maximal bandwidth* dapat diartikan batasan *bandwidth* yang dapat dipakai oleh suatu *class*. Saat lalu lintas cenderung rendah, sebuah *class* dapat memakai *bandwidth* maksimal. Sebuah *class* juga dapat memprioritaskan *trafik* terhadap jaringan tertentu (Septiawan, 2013).

Menurut Mujahidin (2011), OS Mikrotik Sebagai *Manajemen Bandwidth* dengan Menerapkan Metode *Per Connection Queue* mengatakan manajemen *bandwidth* yaitu “Manajemen berasal dari kata “*to manage*” yang berarti mengatur, mengurus atau mengelola. Berdasarkan definisi tersebut 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*”. Semakin banyak pengguna dan akses komunikasi data, maka akan semakin rumit dan kompleks pula jalur komunikasi tersebut, hal ini akan mempengaruhi kualitas dari pelayanan *Internet Service Provider* (ISP).

Menurut Towidjojo (2014), Untuk proses *manajemen bandwidth* dapat dilakukan dengan beberapa tipe *queue*, yaitu *simple queue* dan *queue tree*.

#### 1. *Simple Queue*

*Simple Queue* merupakan menu pada RouterOS untuk melakukan *manajemen bandwidth* untuk skenario jaringan yang sederhana. Untuk menggunakan *Simple Queue*, pekerjaan *packet classification* dan *marking packet* tidak wajib dilakukan. Meskipun demikian, *Simple Queue* sebenarnya juga bisa

melakukan *manajemen bandwidth* terhadap *packet-packet* yang sudah di *marking*.

#### 2. *Queue Tree*

*Queue Tree* adalah konfigurasi *queue* yang bersifat *one way* (satu arah), ini berarti sebuah konfigurasi *queue* hanya akan mampu melakukan *queue* terhadap satu 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*, kita harus membuat 2 (dua) konfigurasi *queue*.

Menurut Towidjojo (2014), Pada saat akan menerapkan *queue* pada jaringan, dikenal dua *rate* atau alokasi *bandwidth* yang akan didapat oleh setiap *user*, yaitu :

1. *Committed Information Rate* (CIR), merupakan alokasi *bandwidth* terendah yang bisa didapatkan oleh sebuah *user* jika *traffic* jaringan sangat sibuk. Seburuk apapun keadaan dari jaringan tersebut, komputer *user* tidak akan mendapatkan alokasi *bandwidth* di bawah dari CIR.
2. *Maximum Information Rate* (MIR), merupakan alokasi *bandwidth* maksimum yang bisa didapatkan komputer *user*. MIR biasanya akan didapatkan seorang *user* jika ada alokasi *bandwidth* yang tidak digunakan lagi oleh *user* lain.

#### 3. *Quality of Service* (QoS)

Menurut Septiawan (2013), *Quality of Service* merupakan metode pengukuran tentang seberapa baik jaringan dan merupakan usaha untuk mendefinisikan karakteristik dan sifat suatu layanan. *Quality of Service* digunakan untuk mengukur sekumpulan atribut kinerja yang telah dispesifikasikan dan biasanya diasosiasikan dengan suatu layanan. *Quality of Service* didesain untuk membantu *end user* (*client*) menjadi lebih praktis dengan memastikan bahwa *user* mendapatkan performansi yang handal dari aplikasi-aplikasi berbasis jaringan.

Menurut Helmy (2014), Terdapat beberapa parameter yang harus dipertimbangkan untuk menentukan *Quality of Service* diantaranya *Throughput*, *Delay*, *Jitter* dan *Packet Loss*.

#### 1. *Throughput*

*Throughput* merupakan kecepatan (*rate*) *transfer* data efektif, yaitu diukur dalam bps. *Throughput* merupakan jumlah total kedatangan paket yang sukses yang diamati pada tujuan selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut.

**Tabel 2** Kategori *Throughput*

Kategori <i>Throughput</i>	<i>Throughput</i> (%)	Indeks
Sangat bagus	100 %	4
Bagus	75 %	3
Sedang	50 %	2
Jelek	< 25 %	1

Untuk mengukur nilai *throughput* dapat menggunakan rumus persamaan berikut :

$$Throughput = \frac{\text{Jumlah Data Diterima}}{\text{Lama Pengamatan}}$$

$$\% Throughput = \frac{Throughput}{\text{Alokasi Bandwidth User}} \times 100 \%$$

#### 2. *Delay*

*Delay* merupakan waktu yang dibutuhkan untuk menempuh jarak dari asal ke tujuan. *Delay* dapat dipengaruhi oleh jarak, media fisik, kongesti atau juga waktu proses yang lama. Menurut versi TIPHON, *delay* dapat diklasifikasikan sebagai berikut :

**Tabel 3** Kategori *Delay*

Kategori <i>Delay</i>	Besar <i>Delay</i> (ms)	Indeks
Sangat bagus	< 150 ms	4
Bagus	150 ms – 300 ms	3
Sedang	300 ms – 450 ms	2
Jelek	> 450	1

Untuk mengukur nilai *delay* dapat menggunakan rumus persamaan sebagai berikut :

$$\text{Rata – Rata Delay} = \frac{\text{Total Delay}}{\text{Total Paket Diterima}}$$

#### 3. *Jitter* atau Variasi Kedatangan Paket

Hal ini disebabkan oleh variasi-variasi dalam panjang antrian, dalam waktu pengolahan data, dan juga dalam waktu pemhimpunan ulang paket-paket diakhir perjalanan *jitter*. *Jitter* lazimnya disebut variasi *delay*, berhubungan erat dengan *latency*, yang menunjukkan banyaknya variasi *delay* pada transmisi data di jaringan. *Delay* antrian pada *router* dan *switch* dapat menyebabkan *jitter*. Terdapat empat kategori penurunan performasi jaringan berdasarkan nilai *peak jitter* sesuai dengan versi TIPHON, yaitu :

**Tabel 4** Kategori *Jitter*

Kategori <i>Jitter</i>	<i>Jitter</i> (ms)	Indeks
Sangat bagus	0 ms	4
Bagus	0 ms – 75 ms	3
Sedang	75 ms – 125 ms	2
Jelek	125 ms – 225 ms	1

Untuk mengukur nilai *jitter* dapat menggunakan rumus persamaan sebagai berikut :

$$Jitter = \frac{\text{Total Variasi Delay}}{\text{Total Paket Diterima}}$$

$$= \frac{\text{Delay} - (\text{Rata – Rata Delay})}{\text{Total Paket Diterima}}$$

#### 4. *Packet Loss*

*Packet Loss* merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan total paket yang hilang, dapat terjadi karena *collision* dan *congestion* pada jaringan dan hal ini berpengaruh pada semua aplikasi karena retransmisi akan mengurangi efesiensi jaringan secara keseluruhan meskipun jumlah *bandwidth* cukup tersedia untuk aplikasi-aplikasi tersebut. Jika terjadi kongesti yang cukup lama, *buffer* akan penuh, dan data baru tidak akan diterima. Nilai *packet loss* sesuai dengan versi TIPHON sebagai berikut :

**Tabel 5** Kategori *Packet Loss*

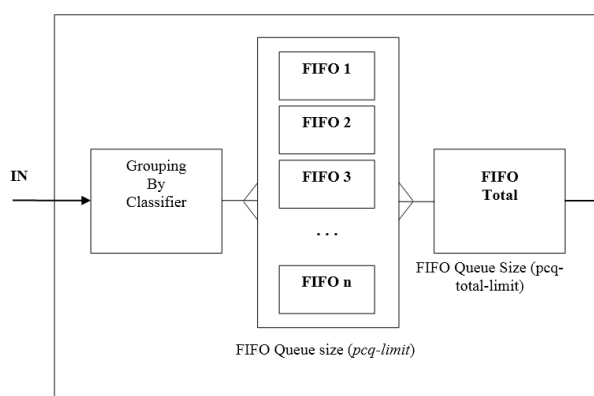
Kategori <i>Packet Loss</i>	Packet Loss (%)	Indeks
Sangat bagus	0 %	4
Bagus	3 %	3
Sedang	15 %	2
Jelek	25 %	1

Untuk mengukur nilai packet loss dapat menggunakan rumus persamaan sebagai berikut :

$$= \frac{\text{Packet Loss}}{\text{Paket Data Dikirim} - \text{Paket Data Diterima}} \times 100\%$$

#### 4. *Per Connection Queue (PCQ)*

Menurut Towidjojo (2014), *Per Connection Queue* merupakan penyempurnaan dari metode *Stochastic Fairness Queuing (SFQ)*. Cara kerja kedua metode ini sama, yaitu berusaha dengan menyeimbangkan *traffic* dengan membuat beberapa *sub stream (sub queue)*. Namun karena merupakan penyempurnaan dari *Stochastic Fairness Queuing*, metode *Per Connection Queue* memiliki beberapa fitur tambahan. Pada *Per Connection Queue*, parameter yang dapat dipilih untuk menjadi *classifier* adalah *src-address*, *dst-address*, *src-port* maupun *dst-port*. Fungsi dari parameter itu adalah sebagai patokan atau standar yang dapat digunakan untuk dijadikan tolak ukur pengujian metode antrian *Per Connection Queue*.



**Gambar 1** Cara Kerja PCQ

Menurut Towidjojo (2014), berdasarkan Gambar 2.2 dapat dijelaskan pada saat sekumpulan paket atau *traffic* masuk ke dalam

konfigurasi *queue* yang menggunakan metode *Per Connection Queue*, maka yang pertama dilakukan *Per Connection Queue* adalah menggunakan *classifier* untuk memisahkan seluruh aliran *packet (stream)* yang masuk menjadi beberapa *sub stream*. *Classifier* yang digunakan bisa saja *src-address*, *dst-address*, *src-port* maupun *dst-port*. Misalnya, jika menggunakan *src-address* sebagai *classifier*, maka *sub stream* dibangun berdasarkan jumlah IP *address* atau jumlah *client*. Setelah melewati bagian *classifier*, terlihat bahwa sekumpulan *packet* akan menjadi beberapa *sub stream*, pada gambar terlihat menjadi 3 (tiga) *sub stream*. Jumlah *sub stream* ini bisa saja dibatasi dengan menggunakan parameter tambahan. Terlihat pula pada gambar bahwa jumlah *sub stream* bisa mencapai jumlah *n sub stream*.

Setelah terpisah-pisah menjadi beberapa *sub stream*, maka pada setiap *sub stream* akan diterapkan metode *Queue First In First Out (FIFO)*. *Per Connection Queue* juga dapat melakukan pembatasan *rate* (kecepatan) pada setiap *sub stream*. Disinilah kerja utama dari *Per Connection Queue*, terlihat bahwa *Per Connection Queue* menyeimbangkan *traffic* dengan membuat *sub stream* untuk setiap *client* (jika memang IP *address* yang digunakan sebagai *classifier*). Sesaat sebelum keluar dari *Per Connection Queue*, keseluruhan *sub stream* tadi akan disatukan kembali, terlihat sebagai kotak *First In First Out total*. Pada saat keseluruhan *sub stream* tersebut disatukan kembali, *Per Connection Queue* akan kembali menerapkan metode *queue First In First Out* pada keseluruhan *sub stream* yang telah digabungkan tadi. Pada posisi ini, *Per Connection Queue* juga masih dapat melakukan pembatasan *rate* terhadap keseluruhan *sub-stream* (Towidjojo, 2014).

### III. METODE PENELITIAN

#### 1. Kerangka Penelitian

Metodologi penelitian yang digunakan dalam penelitian ini adalah sebagai berikut:

### A. Identifikasi Masalah

Identifikasi masalah adalah suatu tahapan awal dan sangat penting dalam suatu penelitian. Permasalahan yang ditemui adalah koneksi *internet* yang lambat dan tidak terbagi rataanya *bandwidth* pada masing-masing komputer *client* sehingga perlu dilakukan *manajemen bandwidth* untuk memecahkan permasalahan.

Adapun cara yang digunakan untuk mengumpulkan data informasi yang digunakan dalam penelitian ini adalah :

#### 1. Penelitian Lapangan

Penulis melakukan pengamatan langsung terhadap kegiatan yang berhubungan dengan masalah yang diambil. Hasil dari pengamatan tersebut langsung dicatat oleh penulis dan dari kegiatan observasi dapat diketahui kesalahan atau proses dari kegiatan tersebut. Dalam observasi ini penulis dapat mengetahui secara langsung permasalahan konektifitas *internet* yang tidak dilakukannya manajemen *bandwidth* di beberapa tempat.

#### 2. Studi Pustaka (Literatur)

Studi pustaka yaitu dengan cara mengumpulkan data dan informasi dari buku-buku, jurnal serta penelitian terdahulu, baik secara internal maupun eksternal membantu dan mempermudah dalam mengimplementasikan sistem yang akan dibuat.

### B. Analisa Masalah

Dari tahap analisis dapat diketahui dengan jelas masalah-masalah apa saja yang sering muncul, bagaimana cara menyelesaikan masalah atau kendala pada pengaturan dan pembagian *bandwidth* sampai solusi yang dapat diajukan untuk memecahkan masalah tersebut. Untuk mengurangi dampak ketidak stabilan koneksi *internet* perlu adanya *router* dalam jaringan, yang bertugas melakukan pengaturan pemakaian *bandwidth* dan pembagian *bandwidth* seefektif mungkin ke seluruh *client*, jadi setiap *user* akan mendapatkan jumlah *bandwidth* yang sama banyak dalam proses *upload* ataupun *download* data dari *internet*, dengan ini di harapkan tidak akan ada lagi *user* yang mengeluh atas lambatnya

koneksi *internet* bila ada *user* lain yang sedang aktif. Atas dasar tersebut, mengaplikasikan *Mikrotik OS* sebagai *router* jaringan yang memiliki *feature* dan *tools* yang cukup lengkap baik untuk jaringan kabel maupun jaringan *wireless*. *Bandwidth* adalah besar *byte* penggunaan pada *transfer* data dalam jaringan. Oleh karena itu diperlukan program yang dapat mengatur alur *Bandwidth* dari masing-masing komputer yang melewati *router* tersebut.

Pada tahap ini dilakukan analisa manajemen *bandwidth* dengan *Simple Queue* dan *Queue Tree* pada *Per Connection Queue*. Pada analisa kedua *type* antrian dalam *Per Connection Queue* tersebut dapat dilihat *Quality of Service* (QoS) yang merupakan hasil dari manajemen yang telah diterapkan pada sebuah jaringan yang terkoneksi dengan *internet*.

### C. Analisa Sistem Manajemen Bandwidth

Pada tahap ini terdapat tiga langkah kerja yang akan dilakukan yaitu implementasi *Simple Queue* pada metode *Per Connection Queue*, implementasi *Queue Tree* pada Metode *Per Connection Queue* dan *Quality of Service* pada Metode *Per Connection Queue*.

#### 1. Implementasi Simple Queue pada metode Per Connection Queue

Pembuatan sistem dengan menggunakan *mikrotik routerboard RB750* serta aplikasi *winbox* yang digunakan untuk *remote router* yang diletakkan pada *server* dan pembagian *bandwidth* dilakukan menggunakan *Simple Queue* yang sudah tersedia di dalamnya.

#### 2. Implementasi Queue Tree pada metode Per Connection Queue

Pembuatan sistem ini sama dengan implementasi pada *Simple Queue* yang telah dijelaskan di atas, perbedaannya hanya pada *type* antriannya menggunakan *Queue Tree* yang sudah tersedia pada *mikrotik routerboard*. Kedua implementasi tersebut dapat dilihat perbedaan pada masing-masing tipe dalam penerapan manajemen *bandwidth*.

#### 3. Quality of Service pada Simple Queue dan Queue Tree dengan Metode PCQ

Setelah selesai dari tahap implementasi *Simple Queue* dan *Queue Tree* pada metode *Per*

*Connection Queue*, maka dari hasil tersebut dapat dilihat *Quality of Service* atau layanan pada jaringan yang telah diterapkan manajemen *bandwidth* dengan mikrotik routerboard RB750.

Pada *Quality of Service* ini yang akan dianalisa yaitu *throughput*, *delay*, *jitter* dan *packet loss* yang terjadi dalam sistem manajemen *bandwidth* yang telah dibuat.

#### D. Implementasi

Pada tahap implementasi ini akan dilakukan penerapan rancangan yang dianalisa guna untuk pembagian *bandwidth* dengan hasil performasi jaringan sama rata. Implementasi manajemen *bandwidth* dengan tipe antrian *Simple Queue* pada metode *Per Connection Queue* dilakukan apabila jaringan tersebut merupakan jaringan yang sederhana dan menengah seperti *Local Area Network* (LAN). Sedangkan untuk tipe antrian *Queue Tree* digunakan untuk skala jaringan yang rumit dan terdapat berbagai macam jaringan seperti sebuah kantor yang memiliki jaringan *local*.

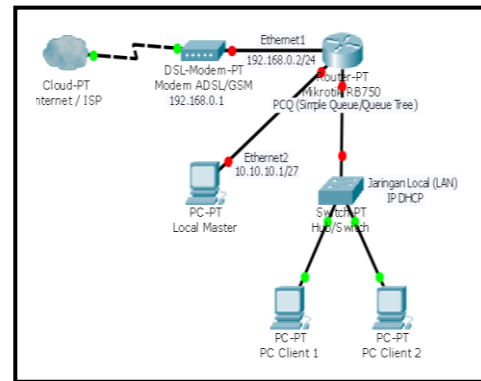
#### E. Evaluasi

Evaluasi merupakan kegiatan yang membandingkan antara hasil implementasi dengan kriteria dan standar yang telah ditetapkan untuk melihat keberhasilannya. Dari evaluasi kemudian akan tersedia informasi mengenai sejauh mana suatu kegiatan tertentu telah dicapai sehingga bisa diketahui bila terdapat selisih antara standar yang telah ditetapkan dengan hasil yang bisa dicapai.

### IV. HASIL DAN PEMBAHASAN

#### 1. Desain Jaringan

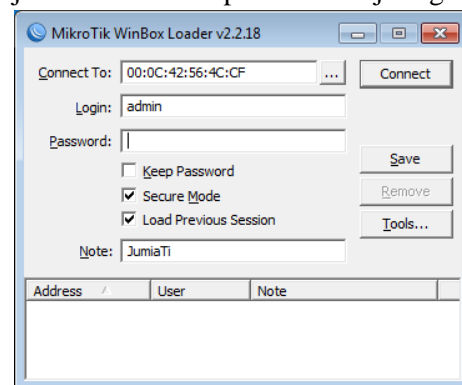
Topologi yang digunakan dalam penelitian tugas akhir ini adalah sebagai berikut :



Gambar 2 Desain Jaringan

#### 2. Instalasi dan Konfigurasi Mikrotik

Pada proses instalasi dan konfigurasi ini, penulis menggunakan aplikasi *winbox* untuk *remote router*. Dalam penggunaan aplikasi *winbox* terdapat beberapa konfigurasi dasar yang perlu dilakukan sebelum menerapkan sistem manajemen *bandwidth* pada sebuah jaringan.



Gambar 3 Aplikasi Winbox

Pada gambar 4.2 di atas, penulis menggunakan MAC Address untuk *login* ke *mikrotik*. Setelah *login* dengan aplikasi *winbox*, ada beberapa langkah kerja yang perlu dijelaskan sebelum masuk pada konfigurasi sistem manajemen *bandwidth* yang akan dibuat, diantaranya adalah :

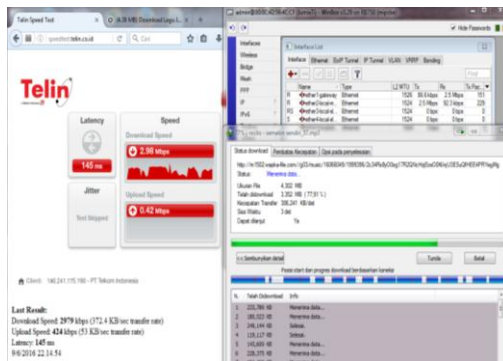
1. Pengujian metode *Simple Queue* dengan *Per Connection Queue*

Data pengujian dilakukan dengan cara melakukan *download* tanpa menggunakan *Simple Queue* PCQ dan dengan menggunakan *Simple Queue* dengan PCQ seperti pada gambar 4.25.

Tampilan dari *download transfer rate* menggunakan *internet download manager* sebelum menggunakan *Simple Queue*. Dengan

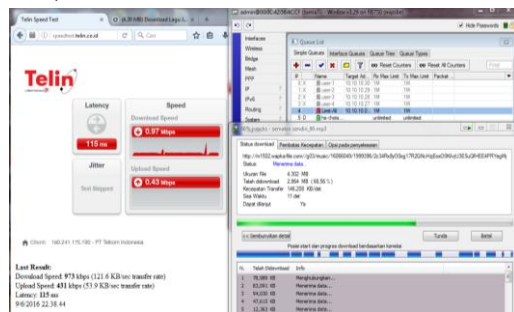


ukuran *file* = 4.30 MB, kecepatan *transfer* = 386.241 KB/det.



**Gambar 4** Transfer Rate Sebelum Menggunakan *Simple Queue* dengan PCQ

Tampilan dari *download transfer rate* menggunakan *Internet Download Manager* setelah menggunakan *Simple Queue*. Dengan ukuran *file* = 4.30 MB, kecepatan *transfer* = 146,208 KB/det.

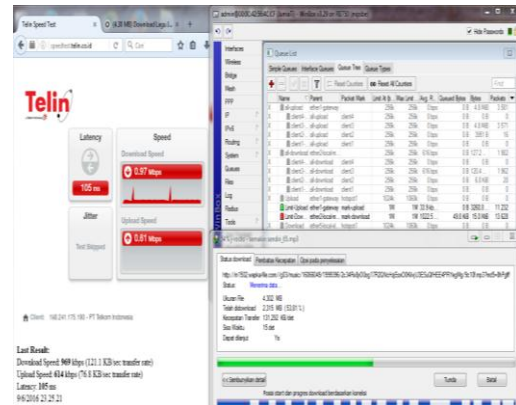


**Gambar 5** Transfer Rate Setelah Menggunakan *Simple Queue* dengan PCQ

## 2. Pengujian metode *Queue Tree* dengan PCQ

Data pengujian diambil dengan cara melakukan *download* menggunakan *queues tree*. Dalam metode ini kita harus mengaktifkan *fitur mangle* pada *firewall*. Seperti yang sudah dilakukan sebelumnya.

Tampilan dari *download transfer rate* menggunakan *Internet Download Manager* setelah menggunakan *Queue Tree*. Dengan ukuran *file* = 4,30 MB, kecepatan *transfer* = 131,292 kb/det

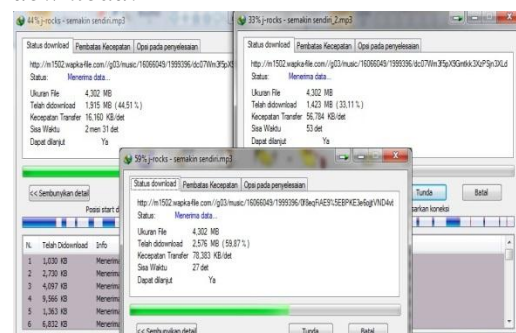


**Gambar 6** Form Transfer Rate dengan Menggunakan *Queue tree* dengan PCQ

## 3. Analisa dan Pengujian Kualitas Layanan (*Quality of Service*)

Pada bagian ini penulis menguji kinerja dari *mikrotik RB750* yang telah dikonfigurasi sebagai manajemen *bandwidth* dengan metode antrian *Per Connection Queue* dengan *type* manajemen *bandwidth Simple Queue* dan *Queue Tree*.

Parameter kualitas jaringan dalam penelitian ini meliputi *throughput*, *delay*, *jitter* dan *packet loss*. Sistem akan dianalisis mengenai tingkat pencapaian kualitas jaringan sistem manajemen *bandwidth* dengan metode antrian *Per Connection Queue* menggunakan *Software Network Analyzer Wireshark*. Dalam pengujian ini dilakukan dengan komputer yang berlaku sebagai *client* yang melakukan aktifitas *download*.



**Gambar 7** Hasil *Download* Sesuai dengan *Limit Bandwidth*

Pada kondisi seperti gambar 4.31 di atas, alokasi *bandwidth* yang didistribusikan pada

jaringan tersebut adalah sebesar 1Mbps, dimana *client* melakukan aktifitas *download* ke url yang sama.

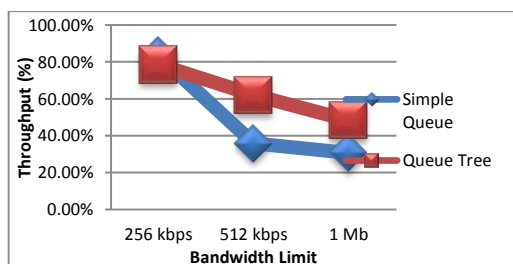
Adapun metode pengambilan data *sample* dalam pengujian ini dilakukan dengan cara berikut :

1. Waktu pengambilan data dibatasi kurang dari 1 menit.
2. Perangkat lunak yang digunakan adalah *Software Network Analyzer Wireshark*.
3. *Bandwidth* yang dibatasi dalam pengujian adalah 256 kbps, 512 kbps dan 1 Mb.
4. Metode antrian yang digunakan dalam pengujian ini yaitu *Per Connection Queue* dengan manajemen *bandwidth Simple Queue* dan *Queue Tree*.
5. Pengujian dilakukan dengan mengunduh *file* dari *internet* dengan ukuran 4.30 MB.
6. Pengukuran dilakukan disisi *client*.

Setelah skenario percobaan dilakukan, didapatkan data *throughput* seperti pada tabel dan grafik berikut :

**Tabel Perbandingan Throughput**

<i>Bandwidth Managemement</i>	Ukur an Berk as	<i>Bandwidth Limit</i>	<i>Throughput</i>
<i>Simple Queue</i>	4.30 MB	256 kbps	83,38 %
		512 kbps	35,81 %
		1 Mb	30,65 %
<i>Queue Tree</i>		256 kbps	78,57 %
		512 kbps	61,98 %
		1 Mb	48.32 %



**Gambar 8** Grafik Perbandingan *Throughput*

Dari tabel 4.1 dan gambar 4.50 di atas dapat dilihat, dalam mengunduh berkas dengan ukuran 4,30 MB, manajemen *bandwidth* dengan

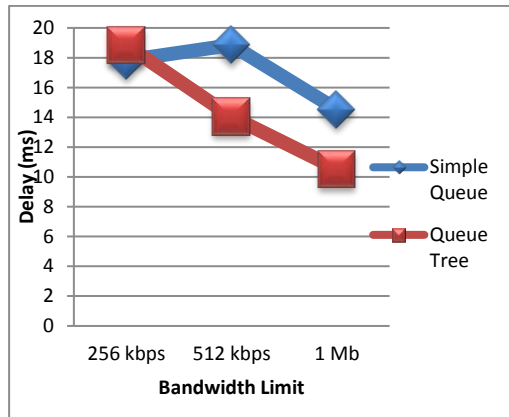
*limit bandwidth* 256 kbps pada *Simple Queue* dan *Queue Tree* memperoleh nilai *throughput* yang hampir sama, yaitu dengan nilai 83,38% untuk *Simple Queue* dan 78,57 % untuk *Queue Tree*. Sedangkan pada *limit bandwidth* 512 kbps dan 1 Mb memperoleh nilai yang jauh berbeda. Parameter *throughput* pada *Simple Queue limit bandwidth* 512 kbps memperoleh nilai 35,81 % dan *limit bandwidth* 1 Mb memperoleh nilai 30,65 %. Sedangkan *Queue Tree limit bandwidth* 512 kbps memperoleh nilai *throughput* 61,98 % dan *limit bandwidth* 1 Mb memperoleh nilai 48,32 %. Nilai parameter *throughput* yang dihasilkan dalam penelitian ini termasuk dalam kategori “Sedang” dalam standar kategori *throughput* menurut standar TIPHON.

Dalam pengujian ini, secara umum nilai *throughput* pada *Queue Tree* lebih besar dibandingkan dengan nilai *throughput* pada *Simple Queue*. Hal ini berkaitan dimana pada dasarnya dalam konfigurasi *Simple Queue* untuk satu antrian dapat membatasi *traffic* 2 arah sekaligus (*upload* dan *download*), sementara pada *Queue Tree traffic upload* dan *download* dibedakan dengan masing-masing konfigurasi. Selain itu, penurunan nilai *throughput* dapat disebabkan oleh beberapa faktor diantaranya perangkat jaringan, topologi jaringan atau induksi listrik dan cuaca.

Selanjutnya untuk data *delay* penggunaan fitur *Simple Queue* dan *Queue Tree* dapat dilihat pada tabel berikut :

**Tabel Perbandingan Delay**

<i>Bandwidth Management</i>	Ukuran Berkas	<i>Bandwidth Limit</i>	<i>Delay / Response Time</i>
<i>Simple Queue</i>	4.30 MB	256 kbps	17,82 ms
		512 kbps	18,85 ms
		1 Mb	14,52 ms
<i>Queue Tree</i>		256 kbps	18,83 ms
		512 kbps	14,04 ms
		1 Mb	10,54 ms



**Gambar 9** Grafik Perbandingan Delay

Dari tabel 4.2 dan gambar 4.51 di atas, secara umum hasil analisis perbandingan parameter *delay* yang dihasilkan pada manajemen *bandwidth Simple Queue* lebih lama dibandingkan manajemen *bandwidth* dengan *Queue Tree*. Nilai parameter *delay* pada limit *bandwidth* 256 kbps *Simple Queue* dan *Queue Tree* memperoleh nilai hampir sama. Pada limit *bandwidth* 256 kbps *Simple Queue* memperoleh nilai 17,82 ms dan limit *bandwidth* 256 kbps pada *Queue Tree* memperoleh nilai 18,83 ms. Nilai parameter *delay* pada *Simple Queue* limit *bandwidth* 512 kbps memperoleh nilai 18,85 ms dan limit *bandwidth* 1 Mb memperoleh nilai 14,52 ms. Sedangkan *Queue Tree* nilai *delay* lebih rendah yaitu pada limit *bandwidth* 512 kbps memperoleh nilai 14,04 ms dan limit *bandwidth* 1 Mb memperoleh nilai 10,54 ms.

Nilai *delay* sangat berpengaruh terhadap seberapa besar *bandwidth* yang disediakan. Semakin besar *bandwidth* yang diberikan, maka semakin kecil nilai *delay* yang dihasilkan.

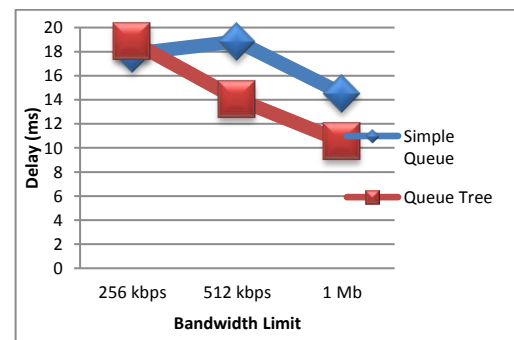
Dalam pengujian ini nilai *delay* pada *Queue Tree* lebih terkontrol dibandingkan dengan *Simple Queue*. Namun nilai parameter *delay* dari kedua *type* antrian tersebut termasuk dalam kategori “**Sangat Bagus**” dalam standar kategori *delay* menurut standar TIPHON.

Untuk data *jitter* dari hasil percobaan dapat dilihat pada tabel dan grafik berikut :

**Tabel 4.3** Perbandingan Jitter

Bandwidth Management	Ukuran Berkas	Bandwidth Limit	Jitter
Simple	4.30	256 kbps	17,82

Queue	MB	ms	
		512 kbps	18,85 ms
		1 MB	14,51 ms
Queue Tree	MB	256 kbps	18,83 ms
		512 kbps	14,04 ms
		1 MB	10,54 ms



**Gambar 10** Grafik Perbandingan Jitter

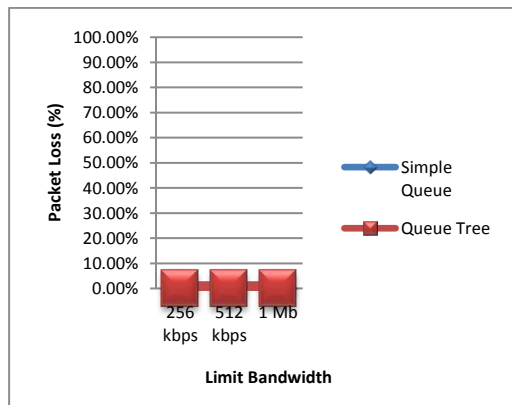
Dari tabel 4.3 dan gambar 4.52 di atas, nilai parameter *jitter* dalam pengujian ini dengan manajemen *bandwidth Simple Queue* dan *Queue Tree* memperoleh nilai yang sama dengan parameter *delay*. Namun demikian nilai *jitter* pada pengujian ini termasuk dalam kategori “**Bagus**” dalam standar kategori *jitter* menurut TIPHON.

Nilai *jitter* sangat berpengaruh terhadap nilai *delay* dan seberapa besar *bandwidth* yang disediakan. Semakin besar *bandwidth* yang diberikan, maka akan semakin kecil nilai *jitter* yang dihasilkan.

Data *packet loss* dari hasil percobaan dapat dilihat pada tabel dan grafik berikut ini.

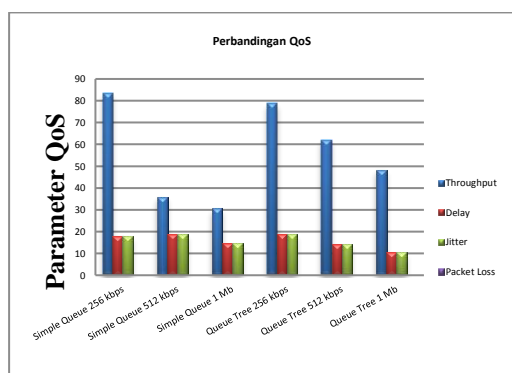
**Tabel 4.4** Perbandingan Packet Loss

<i>Bandwidth Management</i>	Ukuran Berkas	<i>Bandwidth Limit</i>	<i>Packet Loss</i>
<i>Simple Queue</i>	4.30 MB	256 kbps	0%
		512 kbps	0%
		1 Mb	0%
<i>Queue Tree</i>		256 kbps	0%
		512 kbps	0%
		1 Mb	0%



**Gambar 11** Grafik Perbandingan *Packet Loss*

Dari tabel 4.4 dan grafik 4.53 di atas, dalam pengujian dengan mengunduh *file* sebesar 4,30MB, manajemen *bandwidth Simple Queue* maupun *Queue Tree* memperoleh nilai *packet loss* 0%, yang artinya paket yang diterima dari *server* tidak ada mengalami kerusakan ataupun hilang (*loss*) pada saat penerimaan data. Biasanya hal yang menjadi penyebab adanya *packet loss* pada saat penerimaan maupun pengiriman data dari *server* adalah kegagalan pada jaringan, kepadatan *traffic* pada jaringan, kesalahan *hardware* dan keterbatasan *bandwidth* pada jaringan *internet* pada saat penerimaan ataupun pengiriman data dari *server*. Nilai *packet loss* pada pengujian ini termasuk dalam kategori “Sangat Bagus” dalam standar kategori *packet loss* menurut TIPHON.



**Gambar 12** Grafik Perbandingan Parameter *QoS Simple Queue* dan *Queue Tree*

Berdasarkan hasil pengujian analisa perbandingan yang diperoleh dari perhitungan dengan penangkapan data menggunakan *Software Network Analyzer Wireshark* dengan cara mengunduh berkas berukuran 4.30 MB dari

*server*, mikrotik RB750 mampu melakukan manajemen *bandwidth* baik pada *Simple Queue* maupun *Queue Tree* yang menggunakan metode antrian *Per Connection Queue*.

Dalam pengujian ini tiap-tiap nilai parameter QoS yang dihasilkan pada manajemen *bandwidth Queue Tree* lebih stabil dibandingkan *Simple Queue*. Namun hasil pengujian sewaktu-waktu bisa berubah sesuai dengan jaringan *internet* dari ISP (*Internet Service Provider*) mana yang digunakan masing-masing jaringan untuk melakukan pengujian. Selain itu perubahan juga bisa terjadi karena beberapa faktor diantaranya redaman yaitu jatuhnya kuat sinyal karena penambahan jarak pada media *transmisi*, distorsi atau fenomena yang disebabkan bervariasinya kecepatan karena perbedaan *bandwidth* dan masih banyak hal lain yang bisa menyebabkan nilai parameter QoS berubah.

Dalam penelitian ini penulis menggunakan jaringan *internet speedy* untuk melakukan pengujian dari analisa *bandwidth* dengan menggunakan metode *Per Connection Queue*.

## V. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Berdasarkan pembahasan pada bab sebelumnya, penulis mengambil kesimpulan diantaranya :

1. Metode antrian *Per Connection Queue* dapat pembagian *bandwidth* secara adil dan merata untuk masing-masing *client* yang membutuhkan.
2. Fitur manajemen *bandwidth Simple Queue* dinilai lebih sederhana dalam konfigurasi, namun akan menjadi sulit jika jumlah *client* banyak. Sedangkan *Queue Tree* sedikit lebih rumit dalam konfigurasi, tetapi bisa mengontrol *client* dengan jumlah yang banyak.
3. Dalam pengujian ini yang memanfaatkan *Software Network Analyzer Wireshark*, hasil perhitungan manajemen *bandwidth Queue Tree* dengan antrian *Per*

*Connection Queue* lebih stabil dibandingkan *Simple Queue*.

4. Dari hasil perhitungan dalam pengujian *download* berkas, nilai rata-rata yang diperoleh berdasarkan standar kategori TIPHON untuk *indeks* parameter *throughput* adalah 2,3 dengan kategori **“Sedang”**, *indeks* parameter *delay* adalah 4 dengan kategori **“Sangat Bagus”**, *indeks* parameter *jitter* adalah 3 dengan kategori **“Bagus”** dan *indeks* parameter *packet loss* adalah 4 dengan kategori **“Sangat Bagus”**.
5. Analisa dan pengujian manajemen *bandwidth* dengan menggunakan metode antrian *Per Connection Queue* dalam penelitian ini dilakukan menggunakan jaringan *speedy*, sehingga pengujian pada jaringan *internet* yang berbeda ISP bisa saja memperoleh nilai parameter QoS yang berubah sesuai dengan kondisi jaringan saat melakukan pengujian.

## 5.2 Saran

Berdasarkan analisis hasil dan pembahasan penelitian yang telah dilakukan, maka saran yang dapat penulis sampaikan adalah :

1. Lakukan *backup* konfigurasi *router*, apabila ada kesalahan atau hal lain yang menyebabkan koneksi jaringan terganggu, konfigurasi dapat di *restor* kembali.
2. Pada saat pengujian sistem manajemen *bandwidth* diharapkan menggunakan koneksi *internet* yang stabil.
6. Pengerjaan tugas akhir tentang analisan *bandwidth* ini disertakan pengujian kualitas layanan pada jaringan, namun perhitungan untuk masing-masing parameter masih dilakukan secara manual. Diharapkan pada pengembangan selanjutnya mampu membahas lebih detail tentang parameter QoS dan membuat aplikasi QoS untuk kemudahan administrator jaringan dalam menghitung dan melihat kualitas sebuah jaringan

## DAFTAR PUSTAKA

- Aziz, Catur. 2008. *“Panduan Lengkap Menguasai Router Masa Depan Menggunakan Mikrotik Routers”*. Andi; Yogyakarta
- Fitriastuti, Fatsyahrina, dkk. 2013. *“Implementasi Bandwidth Management dan Firewall System Menggunakan Mikrotik OS 2.9.27”*. Jurnal. Universitas Janabadra. Yogyakarta
- Helmy, Dulianto, dkk. 2014. *“Analisa Dan Perbandingan Implementasi Metode Simple Queue dengan Hierarchical Token Bucket (HTB) (Studi Kasus Makosat Brimob Polda Kalbar)”*. Jurnal. Universitas Tanjungpura
- Herlambang, Moch Linto, dkk. 2008. *“Panduan Lengkap Menguasai Router Masa Depan Menggunakan Mikrotik RouterOS<sup>TM</sup>”*. Andi; Yogyakarta
- Imansyah, Surya. 2010. *“Bandwidth Management Dengan Menggunakan Mikrotik Router OS pada RTRW-Net”*. Skripsi Teknik Informatika Universitas Islam Negeri Syarif Hidayatullah Jakarta
- Irawan, Budhi. 2005. *“Jaringan Komputer”*. Yogyakarta; Graha Ilmu
- Mujahidin, Tafaul. 2011. *“OS Mikrotik Sebagai Manajemen Bandwidth dengan Menerapkan Metode Per Connection Queue”*. Naskah Publikasi. Yogyakarta; Amikom
- Muryanto, Prasetyo Uji. 2011. *“Implementasi Sistem Wireless Security Dan Manajemen Bandwidth Berbasis RADIUS (Remote Authentication Dial In User Service) Server dengan Mikrotik”*. Skripsi Teknik Informatika Universitas Islam Negeri Syarif Hidayatullah Jakarta
- Nugroho, Bunafit. 2005. *“Instalasi dan Konfigurasi Jaringan Windows & Linux”*. Yogyakarta; Andi
- Nuh, Zainuddin. 2013. *“Rancang Bangun Proxy Server Berbasis Debian Squeeze 6.0*



- Proxy Squid pada SMK Negeri 5 Pekanbaru*". Skripsi Teknik Informatika Universitas Abdurrah Pekanbaru
- Putro, Okma Winarko. 2013. "*Analisis Penerapan Diffserv Pada Teknologi TCP/IP Tradisional Untuk Jaringan Perangkat Telekomunikasi 3G Berbasis IP Di PT Indosat, Tbk. Cabang Malan*". Jurnal Teknologi Informasi; STMIK PPKIA Pradnya Paramita Malang
- Raharja, Shofiyan Rahman. 2014. "*IP Policy Routing Simple Load Balancing Methos with Failover PCC Queue Tree PCQ Di Mikrotik pada badan meteorologi klimatologi dan geofisika (BMKG)*". Universitas Dian Nuswantoro
- Riadi, Imam. 2010. "*Optimasi Bandwidth Menggunakan Traffic Shapping*". Jurnal Informatika; Universitas Ahmad Dahlan Yogyakarta
- Riadi, Imam, dkk. 2011. "*Implementasi Quality of Service menggunakan Metode Hirarchical Toket Busket*". JUSI Vol 1, No. 2, Yogyakarta
- Septiawan, Didit Aji. 2013. "*Membangun Prioritasisasi Lalu Lintas Data (Internet) Menggunakan HTB Queueing Disciplines Pada Jaringan Lokal SMK N 1 Nanggulan*". Naskah Publikasi; Amikom Yogyakarta
- Silitonga, Parasian, dkk. 2014. "*Analisis QoS (Quality of Service) Jaringan Kampus dengan Menggunakan Microtic Routerboard*". Jurnal TIMES; RSUP Haji Adam Malik Medan
- Sudarmaji. 2014. "*Bandwidth Management Network Design Of Wireless Local Area Network (Wlan) Diploma 3 Program Information Management Universitas Muhammadiyah Metro*". Jurnal Mikrotik; Program Diploma 3 Manajemen Informatika UM Metro
- Syafrizal, Melwin. 2005. "*Pengantar Jaringan Komputer*". Yogyakarta: perpustakaan nasional
- Towidjojo, Rendra. 2013. "*Mikrotik Kung Fu : Kitab I*". Jasakom
- Towidjojo, Rendra. 2014. "*Mikrotik Kung Fu : Kitab 3 Manajemen Bandwidth*". Jasakom
- Wardoyo, Siswo, dkk. 2014. "*Analisis Performa File Transport Protocol Pada Perbandingan Metode Ipv4 Murni, Ipv6 Murni Dan Tunneling 6to4 Berbasis Router Mikrotik*". Jurnal; Universitas Sultan Ageng Tirtayasa Cilegon, Indonesia
- Yugianto, Gin-Gin. 2012. "*Router Teknologi, Konsep, Konfigurasi Dan Troubleshooting*". Bandung: Informatika Bandung