



Analisis Penerapan Metode Antrian Hierarchical Token Bucket untuk Management Bandwidth Jaringan Internet

Ira Puspita Sari^a, Sukri^b

^aTeknik Informatika, Teknik, Universitas Abdurrah, ira.puspita.sari@univrab.ac.id

^bTeknik Informatika, Teknik, Universitas Abdurrah, sukri@univrab.ac.id

Abstract

Internet is one of the needs of the community, especially among education and academics such as universities. The high level of internet users is not comparable with the facilities owned by an agency, this makes the internet network manager in Higher Education feel difficult. The cause of the problem that is always a problem is the speed and internet connection is always disjointed, resulting in the slow loading process data while browsing, both at the time of download and at the time of upload. To make internet users in campus area balanced and stable then need bandwidth management based on traffic usage of internet path better. Bandwidth management that can stabilize traffic lanes and equal bandwidth distribution, it can use bandwidth management by Hierarchical Token Bucket (HTB) method. The calculation result in HTB method testing through file download, the average value obtained by TIPHON category standard for throughput value index is 2 with "Medium" category, delay parameter index is 4 with "Very Good" category, jitter parameter index is 3 with the "Good" category and the packet loss parameter index is 4 with the "Very Good" category.

Keywords: Bandwidth, HTB, Management, Internet

Abstrak

Internet merupakan salah satu kebutuhan dari masyarakat, terutama dikalangan dunia pendidikan dan akademisi seperti Perguruan Tinggi. Tingginya tingkat pengguna internet tidak sebanding dengan fasilitas yang dimiliki suatu instansi, hal ini membuat pengelola jaringan internet di Perguruan Tinggi merasa kesulitan. Penyebab dari permasalahan yang selalu menjadi masalah adalah kecepatan dan koneksi internet yang selalu terputus-putus, sehingga berdampak pada lambatnya proses loading data pada saat melakukan browsing, baik pada saat proses download maupun pada saat proses upload. Untuk pengaturan agar pengguna internet di wilayah kampus seimbang dan stabil maka perlu manajemen bandwidth berdasarkan traffic penggunaan jalur internet lebih baik. Manajemen bandwidth yang bisa menstabilkan lajur *traffic* dan pembagian bandwidth yang seimbang, maka dapat menggunakan manajemen bandwidth dengan metode *Hierarchical Token Bucket* (HTB). Hasil perhitungan dalam pengujian metode HTB melalui *download* berkas, nilai rata-rata yang diperoleh berdasarkan standar kategori TIPHON untuk *indeks* parameter *throughput* bernilai 2 dengan kategori "Sedang", *indeks* parameter *delay* bernilai 4 dengan kategori "Sangat Bagus", *indeks* parameter *jitter* bernilai 3 dengan kategori "Bagus" dan *indeks* parameter *packet loss* bernilai 4 dengan kategori "Sangat Bagus".

Kata Kunci: Bandwidth, HTB, Manajemen, Internet

© 2018 Jurnal RESTI

1. Pendahuluan

Penggunaan *internet* pada *campus area network* sudah cukup banyak, baik dari pihak karyawan maupun mahasiswa yang terkoneksi dengan media kabel maupun dengan jaringan *hotspot area*. Karena banyaknya *user* yang menggunakan *internet* pada kampus akan menimbulkan padatnya *traffic* penggunaan jalur *internetyang* tidak teratur dan berlebihan sehingga berdampak pada lambatnya proses loading data pada saat melakukan *browsing*, baik pada saat proses *download* maupun pada saat proses *upload*. Terlebih jika kedua proses ini dilakukan secara

concurrent bisa mengakibatkan *bandwidth internet full* atau *over load* sehingga koneksi *internet* ke *provider* (ISP) bisa menjadi sangat lambat bahkan terputus dan dapat mengakibatkan *crash* atau *down*[1]

Penggunaan koneksi *internet* yang baik dan memadai mutlak diperlukan di Perguruan tinggi supaya para pengguna atau *user* dapat menikmati teknologi secara efisien dan efektif, untuk itu diperlukan pengolahan *management bandwidth*, bukan untuk membatasi tetapi lebih kepada menjaga kualitas *bandwidth*, sehingga ketika ada seorang *client* yang menggunakan data lebih banyak maka *client* yang lain tidak terganggu karena

Diterima Redaksi : 02-06-2018 | Selesai Revisi : 29-06-2018 | Diterbitkan Online : 30-06-2018

setiap *client* sudah memiliki besar *bandwidth* masing-masing. Tanpa adanya *management bandwidth* banyak para *client* akan menggunakan *internet* secara tidak beraturan yang menyebabkan beberapa *client* tidak dapat menggunakan *bandwidth* secara merata.

Manajemen *bandwidth* dapat dikelola dengan menggunakan metode antrian *Per Connection Queue* dapat pembagian *bandwidth* secara adil dan merata untuk masing-masing *client* yang membutuhkan. 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”[2].

Dengan menggunakan *router mikrotik* maka akan dapat dengan mudah melakukan manajemen *bandwidth*, didalam *router mikrotik* terdapat beberapa metode antrian yang digunakan untuk melakukan *managemen bandwidth*. Salah satu metode antrian yang digunakan untuk pembagian *bandwidth* yaitu menggunakan metode antrian *Hierarchical Token Bucket (HTB)*.

Penggunaan HTB diharapkan dapat menjadi solusi alternative untuk *management bandwidth*, terutama pada sebuah universitas yang mempunyai jaringan *wi-fi* dan dilengkapi dengan laboratorium komputer yang terhubung dengan *internet*. Dengan jumlah *client* yang sulit diperkirakan jumlahnya, penerapan *management bandwidth* akan menjadi lebih sulit, maka dari itu metode HTB ini digunakan untuk mempermudah mengontrol penggunaan *bandwidth*. Dengan adanya metode HTB ini dapat membagi secara rata penggunaan *bandwidth* dalam jaringan *internet* dan tidak ada komputer yang mendapat *bandwidth* lebih besar.

2. Tinjauan Pustaka

Secara sederhana jaringan komputer dapat diartikan sebagai sekumpulan beberapa komputer dan peralatan lain yang saling terhubung menggunakan aturan-aturan tertentu. Hubungan ini dapat terjadi menggunakan media fisik berupa kabel, gelombang radio, infra merah, bahkan satelit. Setiap peralatan yang tersambung ke jaringan disebut *node*.

Sebuah jaringan komputer disusun atas beberapa bagian, yang pertama adalah komputer itu sendiri, dan yang lainnya adalah media koneksi (seperti kabel tembaga, *coaxial*, atau fiber optik) dan piranti lain seperti *hub*, *switch*, dan *router*[3].

Hub adalah piranti untuk pembuatan jaringan *star* yang paling lazim digunakan selain *switch*. *Hub* berfungsi sebagai piranti sentral untuk menghubungkan komputer-komputer di LAN. *Hub* tidak memiliki

komponen aktif, fungsinya hanyalah mengatur kabel dan merelay sinyal data ke semua komputer yang ada di jaringan.

Router merupakan piranti jaringan yang lebih canggih dibandingkan dengan *bridge* dan *switch*. Sebuah *router* terdiri dari *hardware* dan *software* untuk mengatur rute data dari asal sumber data ke tujuan. *Router* memiliki sistem operasi yang canggih yang memungkinkan anda untuk mengkonfigurasi *port* koneksinya.

Wifi (wireless fidelity) adalah koneksi tanpa kabel seperti *handphone* dengan mempergunakan teknologi radio sehingga pemakainya dapat mentransfer data dengan cepat dan aman. *Wifi* tidak hanya dapat digunakan untuk mengakses *internet*, *wifi* juga dapat digunakan untuk membuat jaringan tanpa kabel di perusahaan dan universitas. Karena itu banyak orang mengasosiasikan *wifi* dengan “kebebasan” karena teknologi *wifi* memberikan kebebasan kepada pemakainya untuk mengakses *internet* atau mentransfer data dari ruangan yang bertanda “*wifi hot spot*”. Juga salah satu kelebihan dari *wifi* adalah kecepatannya yang beberapakali lebih cepat dari modem kabel sehingga pemakai *wifi* tidak lagi harus berada didalam ruangan tertentu untuk bekerja

Transmission control protocol/internet protocol (TCP/IP) merupakan standar komunikasi data yang digunakan oleh komunitas *internet* dalam proses tukar-menukar data dari satu komputer ke komputer lain di dalam suatu jaringan[4].

Model *transmission control protocol/internet protocol (TCP/IP)* pada awalnya dikembangkan oleh *Advanced Research Project Agency (ARPA)* dari Departemen Pertahanan Amerika Serikat pada tahun 1969. TCP/IP merupakan sebuah standar jaringan terbuka yang bersifat *independent* terhadap mekanisme *transport* jaringan fisik yang digunakan, sehingga dapat digunakan dimana saja[5].

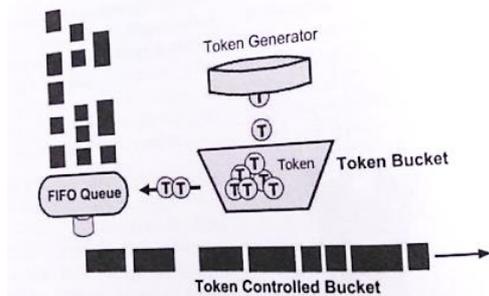
Mikrotik dibuat oleh MikroTikis sebuah perusahaan di kota Riga, Latvia. *Mikrotik* awalnya ditujukan untuk PJI (perusahaan jasa layanan *internet*) atau ISP (*internet service provider*) yang melayani pelanggannya menggunakan teknologi nirkabel atau *wireless*. *Mikrotik* sekarang menyediakan *hardware* dan *software* untuk konektivitas *internet* diseluruh dunia. Produk *hardware* unggulan *mikrotik* berupa *router*, *switch*, antena, dan perangkat pendukung lainnya, sedangkan produk *software* unggulan *mikrotik* adalah *mikrotik routerOS* [6].

Jenis-jenis dari *mikrotik* ada dua yaitu *mikrotik routerOS* dan BUILT-IN *Hardware Mikrotik*[7]. *Mikrotik RouterOS* yang berbentuk perangkat lunak yang dapat di-download di www.mikrotik.com dan Dapat diinstall pada komputer rumahan (*Personal Computer*). BUILT-IN *Hardware Mikrotik* merupakan *mikrotik* dalam bentuk perangkat

keras yang khusus dikemas dalam *board router* yang didalamnya sudah terinstall *Mikrotik RouterOS*.

Hierarchical token bucket (HTB) adalah metode yang berfungsi untuk mengatur pembagian *bandwidth*, pembagian dilakukan secara hirarki yang dibagi-bagi kedalam kelas sehingga mempermudah pengaturan atau pembagian *bandwidth* dengan tepat sehingga penggunaannya menjadi maksimal. Metode ini merupakan metode utama yang digunakan *router mikrotik* untuk melakukan manajemen *bandwidth*. *Queue* yang dibuat oleh *routerOS* akan disusun berdasarkan prinsip *hierarchical token bucket* (HTB).

Hierarchical token bucket (HTB) adalah metode manajemen *bandwidth* yang digunakan untuk membatasi akses menuju alamat IP tertentu tanpa mengganggu *traffic bandwidth* pengguna lain. *Hierarchical token bucket* (HTB) merupakan teknik penjadwalan paket yang sering digunakan pada *router-router* berbasis *linux*. HTB adalah salah satu teknik penjadwalan yang digunakan pada *simple queue*[8].



Gambar 1. Token Bucket

Berdasarkan pada Gambar 1 dapat di jelaskan bahwa adanya *token generator* yang bertugas menghasilkan *token* dengan durasi atau interval waktu yang tetap. *Token-token* tersebut dikumpulkan oleh ember yang disebut *token bucket* disaat yang bersamaan, paket-paket yang diterima *router* akan dikumpulkan oleh *FIFO queue*, dimana *FIFO queue* ini dapat disamakan dengan *leaky bucket*[9].

Namun sedikit berbeda dengan *leaky bucket*, *FIFO queue* ini memiliki katup (pintu) pada bagian *output*. Katup ini akan terbuka jika ada *token* yang dikeluarkan dari *token bucket* tadi. Sehingga semakin sering *token* diambil dari *bucket* tadi, semakin sering katup tersebut terbuka, dan semakin sering pula paket akan sering dikirimkan.

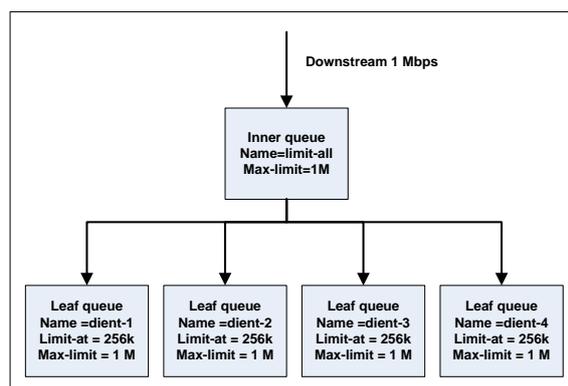
Pada HTB dikenal sebuah metode *simple queue* yang merupakan menu pada *routerOS* untuk melakukan manajemen *bandwidth* untuk skenario jaringan yang sederhana. Untuk menggunakan *simple queue*, pekerjaan *packet classification* dan marketing *packet* tidak wajib untuk dilakukan.

Konfigurasi *queue* pada *simple queue* akan bekerja dengan menggunakan metode HTB, sehingga nantinya

bisa diberikan alokasi *bandwidth* minimum dan maksimum kepada komputer *client*. Dengan menggunakan metode HTB, maka konfigurasi *queue* pada *simple queue* disusun secara hirarki, sehingga nantinya ada *queue* yang berperan sebagai *parent* (induk) dan ada *queue* yang berperan sebagai *child* (anak).

Queue yang berperan sebagai *child* akan selalu meminta jatah *bandwidth* kepada orang tuanya. Yang perlu diingan mengenai HTB, yaitu pertama HTB hanya akan berjalan, apabila *rule queue client* berada dibawah setidaknya 1 *levelparent*, setiap *queue client* memiliki parameter *limit-at* dan *max-limit*, dan *parent queue* harus memiliki besaran *max-limit*. Kedua Jumlah seluruh *limit-at client* tidak boleh melebihi *max-limit parent*. Ketiga *Max-limit* setiap *client* harus lebih kecil atau sama dengan *max-limit parent*. Keempat Untuk *parent* dengan *level* tertinggi, hanya membutuhkan *max-limit* (tidak membutuhkan parameter *limit-at*). Kelima Untuk semua *parent*, maupun sub *parent*, parameter *priority* tidak diperhitungkan. *Priority* hanya diperhitungkan pada *child queue*. Keenam Perhitungan *priority* baru akan dilakukan setelah semua *limit-at* (baik pada *child queue* maupun sub *parent*) telah terpenuhi.

Pada konfigurasi HTB di perlukan adanya CIR (*committed information rate*) dan MIR (*maximum information rate*), untuk memberikan alokasi *bandwidth* maksimum dan minimum. CIR (*committed information rate*) merupakan alokasi *bandwidth* terendah yang bisa didapatkan oleh sebuah komputer *user* jika *traffic* jaringan sangat sibuk[9]. Sedangkan MIR (*maximum information rate*) merupakan alokasi *bandwidth* maksimum yang bisa didapatkan komputer *user*. MIR biasanya akan didapatkan seseorang *user* jika ada alokasi *bandwidth* yang tidak digunakan lagi oleh *user* lain.



Gambar 2. Inner Queue dan Leaf Queue

Berdasarkan pada Gambar 2 terlihat ada beberapa konfigurasi *queue* yang disusun secara hirarki, terdapat satu *queue* yang bertindak sebagai *inner queue* dan empat *queue* yang bertindak sebagai *leaf queue* yang terletak paling bawah dari susunan hirarki HTB.

Adapun *inner queue* adalah *queue* yang bertugas sebagai pembagi alokasi *bandwidth* (distribution) kepada setiap *leaf queue*, selama *inner queue* tadi masih memiliki persediaan *bandwidth*.

Administrasi Mikrotik bisa dilakukan melalui *Windows Application (Winbox)*. Pada saat ini, *winbox* telah di tampilkan secara *graphical*, sehingga *user* dengan mudah dapat mengakses dan konfigurasi *router* sesuai kebutuhan dengan mudah efektif dan efisien. Memperkecil kesalahan pada waktu *setup konfigurasi*, mudah dipahami dan *customable* sesuai yang diinginkan [10].

Bandwidth merupakan konsep pengukuran yang penting dalam jaringan. Banyaknya ukuran suatu data atau informasi yang dapat mengalir dari suatu tempat ke tempat lain dalam sebuah *network* di waktu tertentu. *Bandwidth* juga dapat dipakai untuk mengukur baik aliran data analog maupun digital.

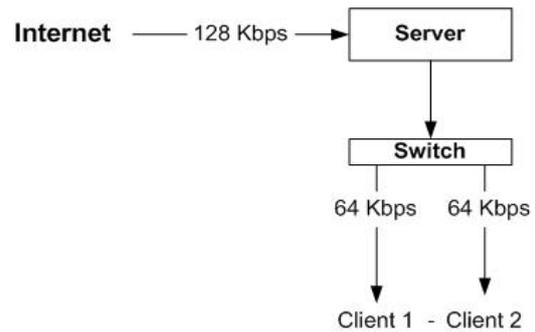
Traffic bandwidth secara umum dikelompokkan menjadi dua jenis yaitu *Up Stream* merupakan *bandwidth* yang digunakan untuk mengirim data dan *Down Stream* merupakan *bandwidth* yang digunakan untuk menerima data [11].

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*). *Bandwidth* dapat dipakai untuk mengukur baik aliran data analog maupun data digital [12].

Bandwidth management adalah suatu cara yang dapat digunakan untuk *management* dan mengoptimalkan berbagai jenis jaringan dengan menerapkan layanan *Quality Of Service (QoS)* untuk menetapkan tipe-tipe lalu lintas jaringan. Sedangkan *QoS* adalah kemampuan untuk menggambarkan suatu tingkatan pencapaian didalam suatu sistem komunikasi data [13].

OS Mikrotik Sebagai *Manajemen Bandwidth* dengan Menerapkan Metode *Hierarchical Token Bucket* mengatakan manajemen *bandwidth* yaitu “Manajemen berasal dari kata “*to manage*” yang berarti mengatur, mengurus atau mengelola [14]. Semakin banyak pengguna dan pengakses komunikasi data, maka akan semakin rumit dan kompleks pula jalur komunikasi tersebut, hal ini akan mempengaruhi kualitas dari pelayanan *Internet Service Provider (ISP)*.

Berdasarkan pada Gambar 3 maka *server* akan mendapatkan *bandwidth* sebesar 128 Kbps, setiap *client* akan melakukan *browsing* dan *download/upload*, *client* memiliki aturan yang berbeda sehingga tidak terjadi gangguan antara *client 1* dan *client 2* tetapi dapat meminjam *bandwidth* jika salah satu *client* tidak aktif [15].



Gambar 3. Proses Manajemen *Bandwidth*

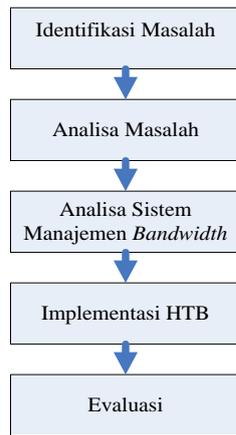
Ketika kita pertama kali mendengar kata *QoS* atau *Quality Of Service* kita pasti akan mengartikannya sebagai kualitas dari suatu pelayanan. Sebenarnya, *QoS* sangat populer dan menyimpan banyak istilah yang sangat sering dilihat dari perspektif yang berbeda yaitu dari segi jaringan, pengembangan aplikasi dan lain sebagainya.

Tujuan dari *Quality of Service* adalah untuk memenuhi kebutuhan-kebutuhan layanan yang berbeda, yang menggunakan infrastruktur yang sama. *Quality of Service* menawarkan kemampuan untuk mendefinisikan atribut-atribut layanan yang disediakan baik secara kualitatif maupun kuantitatif, *Quality of Service* memegang peranan yang sangat penting dalam hal pembagian *bandwidth*. Salah satunya adalah dengan menggunakan metode *hierarchical token bucket (HTB)*, yang menjamin para pengguna jaringan mendapatkan *bandwidth* yang sesuai dengan yang telah didefinisikan, dan juga terdapat fungsi pembagian *bandwidth* yang adil diantara para pengguna jaringan sehingga *performance* jaringan tetap dapat terjaga [16].

Internet adalah suatu sistem jaringan komunikasi beberapa komputer yang terhubung tanpa batas waktu maupun tempat sehingga dapat dikatakan sebagai suatu komunitas jaringan global. berasal dari istilah *interconnection network*, yaitu hubungan antar jaringan komputer (*network*). Untuk mengakses *internet*, kita dapat menggunakan jasa penyedia layanan *internet service provider (ISP)*. Dengan jasa perusahaan *ISP* ini, kita bisa mendapatkan jalur *internet* setelah komputer kita terhubung dengan komputer *server*. Lewat komputer *server* yang dimiliki oleh *ISP* itulah kita terhubung dengan jaringan komputer diseluruh dunia [17].

3. Metodologi Penelitian

Kerangka penelitian adalah suatu cara untuk mendapatkan suatu tujuan tertentu. Dan untuk mencapai tujuan tersebut penelitian ini akan dilakukan dengan cara mengikuti kerangka kerja penelitian seperti pada gambar 4 dibawah ini.



Gambar 4. Kerangka Kerja Penelitian

Masalah yang akan diidentifikasi adalah untuk mengetahui permasalahan yang terjadi dalam konektivitas *internet* yang akan diimplementasikan ke dalam sistem manajemen dengan *mikrotik*.

Adapun cara yang digunakan untuk mengumpulkan data informasi dalam penelitian dengan pengamatan (*Observation*), penulis melakukan pengamatan langsung terhadap kegiatan yang berhubungan dengan masalah yang diambil. Hasil dari pengamatan tersebut akan langsung dicatat oleh penulis dan dari kegiatan observatif dapat diketahui kesalahan atau proses dari kegiatan tersebut. Studi Pustaka (*Literatur*) dengan cara mengumpulkan data dan informasi dari buku-buku, jurnal, dan pihak universitas Abdurrah, baik secara internal maupun eksternal untuk membantu dan mempermudah sistem yang akan dibuat.

Pada tahap analisa masalah dilakukan analisa manajemen *bandwidth* dengan *Simple queue* pada *Hierarchical Token Bucket*. Pada analisa kedua *type* antrian dalam *Hierarchical Token Bucket* tersebut dapat dilihat *Quality of Service (QoS)* yang merupakan hasil dari manajemen yang telah diterapkan pada sebuah jaringan yang terkoneksi dengan *internet*.

Pada tahap analisa sistem manajemen *bandwidth* terdapat dua langkah kerja yang akan dilakukan yaitu implementasi *Simple Queue* pada metode *Hierarchical Token Bucket* dan *Quality of Service* pada metode *Hierarchical Token Bucket*.

Implementasi *Simple Queue* pada metode *Hierarchical Token Bucket* dengan 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 didalamnya.

Quality of Service pada *Simple Queue* dengan Metode HTB diterapkan setelah selesai dari tahap implementasi *Simple Queue* pada metode *Hierarchical Token Bucket*, maka dari hasil tersebut dapat dilihat *Quality of Service*

atau layanan pada jaringan telah diterapkan manajemen *bandwidth* dengan *mikrotikrouterboardRB750*.

Terdapat beberapa parameter yang harus dipertimbangkan untuk menentukan *Quality Of Service*[8], yaitu *Throughput* untuk kecepatan (*rate*) *transfer* data efektif yang 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. Lihat Tabel 1.

Tabel 1. 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 :

$$Throughput = \frac{\text{Jumlah Data Diterima}}{\text{Lama Pengamatan}} \quad (1)$$

$$\% Throughput = \frac{Throughput}{\text{Alokasi Bandwidth User}} \times 100 \% \quad (2)$$

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, lihat Tabel 2.

Tabel 2. 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:

$$\text{Rata – Rata Delay} = \frac{\text{Total Delay}}{\text{Total Paket Diterima}} \quad (3)$$

Jitter atau Variasi Kedatangan Paket, Hal ini disebabkan oleh variasi-variasi dalam panjang antrian, dalam waktu pengolahan data, dan juga dalam waktu penghimpunan 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 performansi jaringan berdasarkan nilai *peakjitter* sesuai dengan versi TIPHON. Lihat Tabel 3.

Tabel 3. 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 :

$$Jitter = \frac{\text{Total Variasi Delay}}{\text{Total Paket Diterima}} \quad (4)$$

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

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 efisiensi 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 *packetloss* sesuai dengan versi TIPHON sebagai berikut, lihat Tabel 4.

Tabel 4. Kategori *Packet Loss*

Kategori <i>Packet Loss</i>	<i>Packet Loss</i> (%)	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 :

$$Packet Loss = \frac{\text{Paket Data Dikirim} - \text{Paket Data Diterima}}{\text{Paket Data Dikirim}} \times 100 \% \quad (5)$$

Pada tahap implementasi HTB akan dilakukan penerapan rancangan yang dianalisis untuk pembagian *bandwidth* dengan hasil jaringan sama rata. Implementasi manajemen *bandwidth* dengan tipe antrian *Simple Queue* pada metode *Hierarchical Token Bucket* dilakukan pada jaringan *wi-fi*.

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.

4. Hasil dan Pembahasan

Parameter kualitas jaringan dalam penelitian ini meliputi *throughput*, *delay*, *jitter* dan *packet loss*. Sistem akan dianalisis mengenai tingkat pencapaian kualitas jaringan sistem penggunaan teknik antrian dengan tipe *simple queue* terhadap kinerja sistem manajemen *bandwidth* menggunakan *Software Network Analyzer Wireshark*. Dalam pengujian ini dilakukan dengan komputer yang berlaku sebagai *client* yang melakukan aktifitas *download*.

Hasil *speedtest* dan *capture* data oleh *wireshark* menggunakan *simple queue* dengan HTB dengan *bandwidth* 1 Mbps.

Measurement	Captured	Displayed	Marked
Packets	31821	31821 (100.0%)	—
Time span, s	426,758	426,758	—
Average pps	74,6	74,6	—
Average packet size, B	462,5	462,5	—
Bytes	14731608	14731608 (100.0%)	0
Average bytes/s	34 k	34 k	—
Average bits/s	276 k	276 k	—

Gambar 5. Summary *Wireshark* Pada *Simple Queue* Dengan HTB

Berdasarkan data yang telah dilakukan dengan *wireshark* pada *simple queue* dengan HTB *bandwidth* 1 Mbps, didapatkan nilai dengan perhitungan pengujian *Throughput* pada *Simple Queue* dengan HTB

$$Throughput = \text{Paket Data Diterima} / \text{Lama Pengamatan}$$

$$= 14731608 \text{ bytes} / 426,758 \text{ sec}$$

$$= 34519,81685 \text{ bytes/sec}$$

$$= 276158,5348 \text{ kbps}$$

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

$$= 276158,5348 \text{ kbps} / 1024 \text{ kbps} \times 100 \%$$

$$= 26,96 \%$$

Pengujian *Delay* pada *Simple Queue* dengan HTB

$$\text{Rata-rata Delay} = \text{Total Delay} / \text{Total Paket Diterima}$$

$$= 426,758 \text{ sec} / 31821$$

$$= 0,013411206 \text{ sec}$$

$$= 13,41 \text{ ms}$$

Pengujian *Jitter* pada *Simple Queue* dengan HTB

$$Jitter = \text{Total Variansi Delay} / \text{Total Paket Diterima}$$

$$= (426,758 \text{ sec} - 0,013411206 \text{ sec}) / 31821$$

$$= 0,013410787 \text{ sec}$$

$$= 13,41 \text{ ms}$$

Pengujian *Packet Loss* pada *Simple Queue* dengan HTB *Packet Loss*=

$$(\text{Data Dikirim} - \text{Paket Data Diterima}) \times 100 \%$$

$$\text{Paket Data Dikirim}$$

$$= 31821 - 31821 / 31821 \times 100 \%$$

$$= 0 \%$$

Hasil *Speedtest* data Oleh *Wireshark* Tanpa HTB *simple queue* tanpa HTB dengan *bandwidth* 1 Mbps.

Measurement	Captured	Displayed	Marked
Packets	4650	4650 (100.0%)	—
Time span, s	28,985	28,985	—
Average pps	160,4	160,4	—
Average packet size, B	826,5	826,5	—
Bytes	3843979	3843979 (100.0%)	0
Average bytes/s	132 k	132 k	—
Average bits/s	1060 k	1060 k	—

Gambar 6. Summary *Wireshark* Pada *Simple Queue* Tanpa HTB

Dari *capture* data yang telah dilakukan dengan *wireshark* pada *simple queue* tanpa HTB *bandwidth* 1 Mbps, maka didapatkan nilai dengan cara perhitungan pengujian *Throughput* pada *Simple Queue* tanpa HTB

$$Throughput = \text{Paket Data Diterima} / \text{Lama Pengamatan}$$

$$= 3843979 \text{ bytes} / 28,985 \text{ sec}$$

$$= 132619,5963 \text{ bytes/sec}$$

$$= 1060956,771 \text{ kbps}$$

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

$$= \frac{1060956,771 \text{ kbps}}{1024 \text{ kbps}} \times 100 \%$$

$$= 10,36\%$$

Pengujian *Delay* pada *Simple Queue* tanpa HTB Rata-rata *Delay* = Total *Delay* / Total Paket Diterima

$$= \frac{28,985 \text{ sec}}{4650}$$

$$= 0,006233333 \text{ sec}$$

$$= 6,2 \text{ ms}$$

Pengujian *Jitter* pada *Simple Queue* tanpa HTB *Jitter* = Total Variansi *Delay* / Total Paket Diterima

$$= \frac{(28,985 \text{ sec} - 0,006233333 \text{ sec})}{4650}$$

$$= 0,006231993 \text{ sec}$$

$$= 6,2 \text{ ms}$$

Pengujian *Packet Loss* pada *Simple Queue* tanpa HTB *Packet Loss*

$$= \frac{(\text{Data Dikirim} - \text{Paket Data Diterima}) \times 100 \%}{\text{Paket Data Dikirim}}$$

$$= \frac{4650 - 4650}{4650} \times 100 \%$$

$$= 0 \%$$

Setelah skenario percobaan dilakukan, didapatkan perbandingan nilai akhir QoS *simple queue* dengan HTB dan *simple queue* tanpa HTB berdasarkan standar TIPHON, lihat tabel 5.

Tabel 5. Perbandingan Nilai Akhir QoS

Parameter QoS	Simple Queue Dengan HTB		Simple Queue Tanpa HTB	
	Indeks	Kategori	Indeks	Kategori
Throughput	2	Sedang	1	Jelek
Delay	4	Sangat Bagus	4	Sangat Bagus
Jitter	3	Bagus	3	Bagus
Packet Loss	4	Sangat Bagus	4	Sangat Bagus
Total Rata-Rata	3,2	Bagus	3	Bagus

Berdasarkan pada Tabel 5, nilai akhir QoS *simple queue* dengan HTB dalam standar TIPHON mempunyai nilai indeks 3,2 dengan kategori Bagus, sedangkan nilai akhir QoS *simple queue* tanpa HTB dalam standar TIPHON mempunyai nilai indeks 3 dengan kategori Bagus.

Secara umum perbandingan nilai akhir QoS *simple queue* dengan HTB dan *simple queue* tanpa HTB tidak jauh berbeda. Namun apabila dibandingkan kembali dengan menggunakan nilai tiap-tiap parameter QoS berdasarkan besar *bandwidth* yang ditentukan akan sangat tampak perubahan dari tiap-tiap nilai parameter tersebut.

Berdasarkan tabel diatas *Throughput* digunakan untuk mengetahui jumlah paket yang diterima dalam keadaan baik terhadap waktu total transmisi yang dibutuhkan dari *server* hingga ke *user*. Berdasarkan data yang telah diambil dengan menggunakan aplikasi *Wireshark*, yang

kemudian disajikan dalam bentuk tabel pada tabel 5, dapat dilihat nilai total rata-rata parameter *throughput* pada *simple queue* dengan HTB sebesar 26,96 % dengan kategori “Sedang”. Sedangkan nilai total rata-rata parameter *throughput* pada *simple queue* tanpa HTB sebesar 10,36 % dengan kategori “Jelek”. Nilai *throughput* pada *simple queue* dengan HTB lebih besar dibandingkan dengan nilai *throughput* pada *simple queue* tanpa HTB.

Hasil analisis perbandingan nilai *delay* pada *simple queue* dengan HTB dan *simple queue* tanpa HTB ditunjukkan pada tabel 5, dimana nilai parameter *delay* pada *simple queue* dengan HTB sebesar 13,41 ms dengan kategori “Sangat Bagus”, sedangkan nilai parameter *delay* pada *simple queue* tanpa HTB sebesar 6,2 ms dengan kategori “Sangat Bagus”. Nilai *delay* sangat berpengaruh terhadap seberapa besar *bandwidth* yang disediakan. Semakin besar *bandwidth* yang diberikan, maka akan semakin kecil nilai *delay* yang dihasilkan.

Hasil analisis perbandingan nilai *jitter* pada *simple queue* dengan HTB dan *simple queue* tanpa HTB ditunjukkan pada tabel 5, dimana nilai parameter *jitter* pada *simple queue* dengan HTB sebesar 13,41 ms dengan kategori “Bagus”, sedangkan nilai parameter *jitter* pada *simple queue* tanpa HTB sebesar 6,12 ms dengan kategori “Bagus”. 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.

Packet loss digunakan untuk mengetahui banyak jumlah paket yang hilang atau tidak sampai ke tujuan ketika melakukan pengiriman data dari sumber ke tujuan. Kemudian dilakukan perhitungan dengan persamaan untuk mengetahui pengaruh penggunaan *simple queue* tanpa HTB dan *simple queue* dengan HTB.

Hasil analisis perbandingan *packet loss* pada *simple queue* dengan HTB maupun *simple queue* tanpa HTB ditunjukkan pada tabel 5, dimana parameter *packet loss* pada *simple queue* dengan HTB sebesar 13,41 ms dengan kategori “Sedang”, sedangkan nilai parameter *packet loss* pada *simple queue* tanpa HTB sebesar 6,12 ms dengan kategori “Bagus”.

Berdasarkan hasil pengujian analisa perbandingan yang diperoleh dari perhitungan dengan penangkapan data menggunakan *Software Network Analyzer Wireshark* dengan cara mengunduh berkas dari *internet* menggunakan mikrotik RB941 mampu melakukan manajemen *bandwidth* pada *Simple Queue* menggunakan metode antrian *Hierarchical Token Bucket*.

Dalam pengujian ini tiap-tiap nilai parameter QoS yang dihasilkan pada manajemen *bandwidth simple queue* dengan HTB lebih stabil dibandingkan dengan *simple*

queue tanpa HTB. 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* Wanxp untuk melakukan pengujian dari analisa *bandwidth* dengan menggunakan metode HTB.

5. Kesimpulan

Berdasarkan hasil dari penelitian yang dilakukan maka dapat disimpulkan dan saran, yaitu :

5.1 Simpulan

Berdasarkan hasil analisa dan pengujian yang telah dilakukan penulis, maka dapat disimpulkan bahwa :

1. Metode antrian *Hierarchical Token Bucket* dinilai lebih efektif membagi *bandwidth* secara adil dan merata kepada masing-masing *client* yang membutuhkan *bandwidth*, terlihat dari grafik perhitungan nilai QoS yang telah dilakukan.
2. Dalam pengujian dan perhitungan QoS dinilai lebih sederhana menggunakan fitur manajemen *bandwidth simple queue* untuk mendapatkan hasil yang lebih maksimal dengan memanfaatkan *Software Network Analyzer Wireshark*.
3. Dari hasil perhitungan dalam pengujian metode HTB melalui *download* berkas, nilai rata-rata yang diperoleh berdasarkan standar kategori TIPHON untuk *indeks* parameter *throughput* bernilai 2 dengan kategori “Sedang”, *indeks* parameter *delay* bernilai 4 dengan kategori “Sangat Bagus”, *indeks* parameter *jitter* bernilai 3 dengan kategori “Bagus” dan *indeks* parameter *packet loss* bernilai 4 dengan kategori “Sangat Bagus”.

5.2 Saran

Adapun hal-hal yang menjadi saran sebagai pertimbangan untuk pengembangan jaringan kampus agar menjadi lebih baik lagi adalah sebagai berikut:

1. Pada saat pengujian sistem manajemen *bandwidth*, hendaknya menggunakan koneksi *internet* yang stabil.
2. Penggunaan manajemen *bandwidth* dengan *Hierarchical Token Bucket* dapat dikombinasikan dengan metode *Per Connection Queue* dan juga

dapat dilakukan percobaan dengan menggunakan *Queue Tree* untuk mencapai *Quality of Service* yang lebih baik.

3. Menggunakan 9 parameter QoS untuk mendapatkan hasil penelitian yang lebih akurat, yaitu *Packet Loss, Delay, Jitter, Throughput, MOS, Echo Cancellation, Error, Out of Delivery, dan PDD*.

Daftar Rujukan

- [1] Sinaga, J., 2013. Practice Managing Internet Connection Campus Area Network (Can) With Firewall and Address Listmikrotik Router Os. Techno Nusa Mandiri, IX (1), pp. 132–142.
- [2] Sukri., 2017. Analisa Bandwidth Menggunakan Metode Antrian Per Connection Queue. RABIT(Jurnal Teknologi dan Sistem Informasi), 2 (2), pp.244-257.
- [3] Winarno, B., 2012. *Kebijakan publik: teori, proses, dan studi kasus: edisi dan revisi terbaru*. Center for Academic Publishing Service.
- [4] Wardoyo, S, Ryadi, T, and Fahrizal, R., 2014. Analisis Performa File Transport Protocol Pada Perbandingan Metode IPv4 Murni, IPv6 Murni Dan Tunneling 6to4 Berbasis Router Mikrotik. Jurnal Nasional. Teknik Elektro, 3 (2), pp. 106–117.
- [5] Putro, W.O., 2103. Analisis Penerapan Diffserv Pada Teknologi Tcp / Ip Tradisional Untuk Jaringan Perangkat Telekomunikasi 3G. Jurnal Teknologi Informasi, 4(2), pp. 30–44.
- [6] Madcoms. 2015. *Membangun Sistem Jaringan Komputer untuk Pemula*. Yogyakarta : Andi Offset.
- [7] Herlambang, M. L and Azis, C. L., 2008. *Panduan Lengkap Menguasai Router Masa Depan Menggunakan Mikrotik RouterOS*. Yogyakarta : Andi Offset.
- [8] Helmy, D and Priyanto, H., 2015. Analisis Dan Perbandingan Implementasi Metode Simple Queue Dengan Hierarchical Token Bucket (HTB) (Studi Kasus Makosat Brimob Polda Kalbar). Sistem dan Teknologi Informasi, 3(3), pp.228-233.
- [9] Towidjojo, R., 2013. *Mikrotik Kungfu Kitab 2*. Jakarta: Jasakom, pp. 76–78.
- [10] Silitonga, P and Morina, S.I., 2014. Analisis Qos (Quality Of Service) Jaringan Kampus Dengan Menggunakan Microtic Routerboard (Studi Kasus : Fakultas Ilmu Komputer Unika Santo Thomas S.U). Jurnal TIMES, III (2), pp. 19–24.
- [11] Riadi, I., 2010. Optimasi Bandwidth Menggunakan Traffic Shapping. Jurnal Informatika, 4(1), pp. 374–382.
- [12] Pambudi, R and Muslim, A.M., 2017. Implementasi Policy Base Routing dan Failover Menggunakan Router Mikrotik untuk Membagi Jalur Akses Internet di FMIPA Unnes. Jurnal Teknologologi dan Sistem Komputer, 5(2), pp. 57-61.
- [13] Antodi, P.C, Prasetijo, B.A and Widiyanto, D.E., 2017. Penerapan Quality of Service Pada Jaringan Internet Menggunakan Metode Hierarchical Token Bucket. Jurnal Teknologologi dan Sistem Komputer, 5(1), pp. 23-28.
- [14] Rifai, B., 2017. Management Bandwidth Pada Dynamic Queue Menggunakan Metode Per Connection Queuing. Jurnal Ilmu Pengetahuan dan Teknologi Komputer, 2(2), pp. 73–79.
- [15] Riadi, P.W.I, Wicaksono, P., 2011. Implementasi Quality of Service Menggunakan Metode Hierarchical Token Bucket Landasan Teori. JUSI, 1 (2), pp. 93–104.
- [16] Iqbal, M., 2017. Perancangan Sistem Optimalisasi Manajemen Bandwidth Dengan HTB (Hierarchical Token Bucket) Menggunakan Linux. Jaringan Sistem Informasi Robotik, 1(1), pp. 25–29.
- [17] Martanto, A., 2008. Cara Mudah & Cepat Bermain Internet Untuk Pemula. Jakarta : Media Kita.