

## **ANALISIS MANAJEMEN BANDWIDTH METODE HIERARCHICAL TOKEN BUCKET (HTB) PENGGUNAAN VIDEO CONFERENCE DAN SOSIAL MEDIA**

Bambang Siswanto<sup>1</sup>, Achmad Fuad<sup>2</sup>, Sabri Ahmad<sup>3</sup>, Rosihan<sup>4</sup>  
*Program Studi Informatika, Fakultas Teknik, Universitas Khairun*

Jl. Jati Metro, Kota Ternate Selatan

Email: [Bhambangsiswanto98@gmail.com](mailto:Bhambangsiswanto98@gmail.com)<sup>1</sup>, [Fuad@unkhair.ac.id](mailto:Fuad@unkhair.ac.id)<sup>2</sup>, [msabri@gmail.com](mailto:msabri@gmail.com)<sup>3</sup>, [rosihan@unkhair.ac.id](mailto:rosihan@unkhair.ac.id)<sup>4</sup>

### **Abstrak**

Sistem online tentunya sudah bukan hal yang baru lagi saat ini, sistem yang membutuhkan *software*, *hardware*, dan *network* yang baik. Belum ditambah lagi tahun 2020 semua negara menghadapi wabah (*Covid-19*) sehingga aktivitas manusia harus beralih ke cara *online*, Kampus III Unkhair Prodi Informatika yang memilih untuk pembelajaran secara dilakukan dengan *online*. Dengan diberlakukannya secara *online* diperlukan manajemen *bandwidth* sehingga akses yang sering digunakan dapat diprioritaskan dan akses yang lain tidak mengganggu trafik akses yang diprioritaskan, namun tetap mendapatkan *bandwidth* yang optimal, dikarenakan walaupun tidak ada aktivitas belajar bertatap muka, tak jarang ada mahasiswa yang tidak memiliki kuota internet datang ke kampus untuk memanfaatkan Wi-Fi. Penelitian ini memakai metode manajemen *bandwidth* HTB untuk pembagian trafik dan pemberian prioritas. Pengambilan data dilakukan dari jam 11:00-12:00, selama 2 hari, hari pertama untuk pengambilan data sistem sebelumnya dan hari ke dua pengambilan data sistem yang menggunakan metode HTB menggunakan aplikasi *wireshark* yang merekam data trafik secara *real time*, data akan dianalisis dan disesuaikan dengan index QOS standar TIPON. Hasil pengujian untuk sistem sebelumnya, kualitas jaringan indexnya untuk *throughput* index 4, *loss* index 4 dan *delay* index 4, sedangkan sistem yang menggunakan HTB, kualitas jaringan indexnya sesuai standar TIPON untuk *throughput* index 3, *loss* index 4 dan *delay* index 4.

**Kata kunci** : Manajemen *Bandwidth*, Jaringan komputer, *Hierarchical token bucket* (HTB).

## **BANDWIDTH MANAGEMENT ANALYSIS HIERARCHICAL TOKEN BUCKET (HTB) USE OF VIDEO CONFERENCE AND SOCIAL MEDIA**

### **Abstract**

*The online system, of course, is nothing new at this time, a system that requires good software, hardware, and network. Not to mention that in 2020 all countries are facing an epidemic (Covid-19) so that human activities must switch to online methods, Campus III Unkhair Informatics Study Program chooses to learn online. With the implementation of online bandwidth management, bandwidth management is needed so that access that is often used can be prioritized and other accesses do not interfere with prioritized access traffic, but still get optimal bandwidth, because even though there are no face-to-face learning activities, it is not uncommon for students who do not have internet quota. come to campus to take advantage of Wi-Fi. This study uses the HTB bandwidth management method for traffic distribution and prioritization. Data retrieval is carried out from 11:00 to 12:00, for 2 days, the first day for retrieval of previous system data and the second day of retrieval of system data using the HTB method using a wireshark application that records traffic data in real time, the data will be analyzed and adjusted to the TIPON standard QOS index. The test results for the previous system, the network quality index for throughput index 4, loss index 4 and delay index 4, while the system using HTB, the network quality index according to TIPON standards for throughput index 3, loss index 4 and delay index 4.*

**Keywords**: *Bandwidth Management*, *Computer network*, *Hierarchical token bucket* (HTB).

## 1. PENDAHULUAN

Sistem *online* tentunya sudah bukan hal baru saat ini, sistem yang membutuhkan *software*, *hardware*, dan koneksi internet yang baik, sehingga dapat digunakan dalam mempermudah urusan atau pekerjaan manusia seperti aktivitas kantor, belanja, berdagang, berpergian, dan belajar-mengajar. Selain itu juga, dapat membantu manusia untuk mencari informasi baik dengan mengakses media sosial atau situs *web* berita.

Belum ditambah lagi tahun 2020 ini, semua negara menghadapi wabah (*Covid-19*) yang sangat berbahaya dan dapat menyebar dengan sangat cepat, sehingga memaksa aktivitas manusia yang biasanya beramai-ramai seperti yang sudah disebutkan sebelumnya harus dikurangi dan mulai beralih ke sistem *online*. Kampus III Unkhair Prodi Informatika contohnya, sala satu dari banyaknya Prodi yang lebih memilih untuk melakukan proses belajar-mengajar *online*, dan pada area Kampus tersebut tentunya disediakan akses internet berupa Wi-Fi untuk mahasiswa, meskipun tidak ada aktivitas belajar bertatap muka, tak jarang ada mahasiswa yang tidak mempunyai kuota internet sendiri atau kuota internetnya sudah habis datang ke kampus untuk memanfaatkan Wi-Fi yang disediakan untuk belajar *online*, menonton seminar tugas akhir oleh mahasiswa teknik informatika yang diselenggarakan oleh pihak prodi, menonton video tutorial untuk projek yang dibuat, atau hanya sekedar mengakses sosial media untuk mencari informasi terbaru dengan tetap mematuhi protokol kesehatan tentunya, tak jarang jaringan Wi-Fi tersebut terasa lambat saat diakses oleh pengguna, dikarenakan pemakaian dari setiap pengguna yang berbeda-beda sehingga *bandwidth* menjadi tidak merata untuk semua pengguna.

Penelitian sebelumnya dengan judul juga “Analisis dan perbandingan implementasi metode *simple queue* dengan *hierarchical token bucket* (HTB)” yang dibuat oleh Dulianto Helmy, Heri Priyanto [1].

Berdasarkan hal tersebut penulis mencoba untuk menerapkannya pada ruang lingkup kampus 3 Unkhair prodi informatika, untuk sebagaimana mengetahui (QOS) dari sebelum dan sesudah dilakukannya manajemen *bandwidth*, dengan metode (HTB) untuk *video conference* dan sosial media dengan akses-akses yang diprioritaskan pada jaringan Wi-Fi tersebut..

## 2. TINJAUAN PUSTAKA

### 2.1. Jaringan Komputer

Jaringan komputer adalah bentuk dari hubungan beberapa perangkat atau komputer yang bisa saling berhubungan antar sesama. Perangkat yang artikan pada defenisi ini adalah, perangkat yang menyerupai semua jenis perangkat komputer (*desktop*, *jinjing*, *smartphone*, *PC tablet*) dan perangkat penghubung, Berdasarkan defenisi

mengenai sebuah jaringan komputer [2], maka untuk dapat disebut sebagai sebuah jaringan komputer.

Ada syarat-syarat yang harus dipenuhi. Syarat-syarat tersebut yaitu :

1. Memiliki dua jenis komputer yang saling terkoneksi antar sesama..
2. Terdapat pengguna atau *user* yang saling berhubungan dengan pemakai lain di sebuah sistem, maupun penyedia layanan yang lain.
3. Mempunyai *packet/data* yang saling bertukar didalamnya. Bukan cuman data bisa juga berupa teks, multimedia ataupun informasi hasil proses datanya.
4. Bisa digunakan bersama kepada perangkat keras maupun perangkat lunak.

### 2.2. Manajemen *Bandwidth*

*Bandwidth* manajemen merupakan metode yang dapat dipakai untuk mengoptimalkan beberapa kategori *quality of service* dalam menentukan jenis-jenis trafik jaringan. Dalam artian dimana *bandwidth* akan disetarakan kepada pengguna atau *user* agar jalannya kualitas sebuah jaringan mendapatkan kepuasan tersendiri untuk pengguna tanpa adanya kejadian perebutan *bandwidth* seperti sebelum dilakukannya manajemen [3], ada juga menjelaskan pemakaian *bandwidth* yang lebih baik dapat digunakan administrator jaringan internet dengan mengimplementasikan manajemen *bandwidth* pada trafik datanya. Tanpa bantuan manajemen *bandwidth*, mengakibatkan penggunaan jaringan internet menggunakan *bandwidth* dengan tidak terstruktur [4].

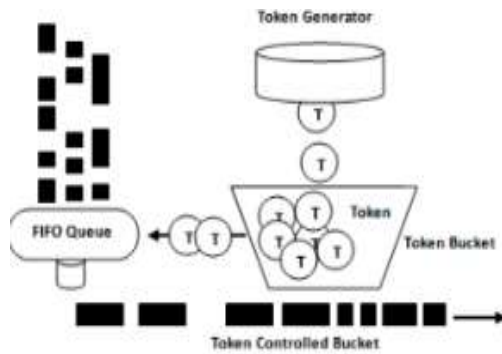
### 2.3. *Hierarchical Token Bucket* (HTB)

HTB ialah metode secara *hierarki* atau antrian yang adil dan bertujuan menerapkan fungsi jalur *sharing* ke masing-masing *client* [5].

HTB merupakan cara *hierarki* di *mikrotik* yang bisa untuk memajemenkan *bandwidth* dengan cara *hirarki* dan *burst* dari *token bucket*. Teknik HTB antrian sendiri menggunakan fasilitas batasan *traffic* untuk setiap level ataupun klasifikasi. *Bandwidth* yang tidak terpakai akan digunakan oleh klasifikasi yang lebih kecil. Fungsi HTB adalah fungsi berbentuk struktur *queue* dengan bentuk *hirarki* yang berperan untuk mengatur hubungan antar sesama kelas *hirarki*. HTB sendiri terdapat 3 jenis kelas antara lain *root*, *inner*, dan *leaf*. *Root class* adalah kelas yang berposisi diatas *hirarki* dan semua trafik yang keluar melewati kelas ini. *Inner class* adalah kelas yang mempunyai kelas *parent* dan kelas *child*. Bagian ini fungsi dalam menyampaikan informasi sehingga *bandwidth* yang lebih untuk kelas *child* yang menyertainya. Terakhir adalah kelas *leaf* adalah kelas penghubung berada di antrian/*hirarki* paling bawah. Kelas ini sendiri akan mengontrol antrian dalam satu lalu lintas yang dilalui. Implementasi dari HTB mempunyai *dual*

limitation yang berarti dua pembagian bandwidth yang berguna untuk pembatas bandwidth yang disetting seimbang bagi setiap parent. Kedua ratelimits tersebut yaitu:

1. *Committed Information Rate (CIR) (limit-at)* adalah bentuk dari kondisi terburuk, ialah proses untuk menentukan kelas batas terbawah atau minimal kecepatan trafiknya (*limit-at*).
2. *Maximal Information Rate (MIR) (max-limit)* pada RouterOS merupakan kondisi terbaik, ialah proses batas maksimal kecepatan (*max-limit*) yang terdapat dalam antrian saat jaringan sedang tidak terpakai. *Average* jalur trafik yang diberikan ke setiap pengguna bisa sampai pada *rate* maksimal, [6].

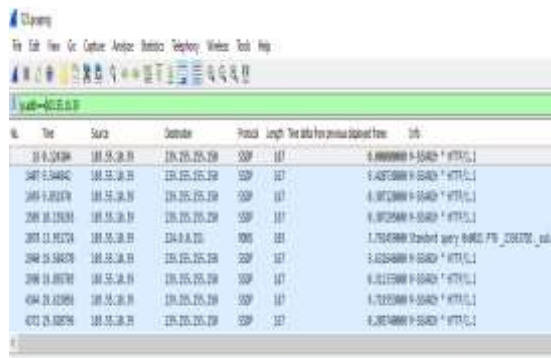


Gambar 1 HTB [5]

**2.4. Wireshark**

Wireshark adalah network protocol analyzer, dan tergolong didalamnya network analysis tool atau packet sniffer.

Selama terjadinya alur dari trafik paket pada satu network, maka packet sniffer bisa merekam PDU, bisa melakukan decoding, bisa juga analisis terkait isi dari paket. Wireshark itu juga sebagai packet sniffer yang dibuat dan direncanakan sedemikian agar bisa mengenali dan menguraikan berbagai macam jenis protokol yang berjalan dalam sebuah jaringan[7].



Gambar 2 Wireshark Apk

**2.5. Quality Of Service**

Menurut Houston dalam QOS adalah cara pengukuran terkait kualitas baik dan buruknya suatu

jaringan dan merupakan suatu usaha untuk mendefinisikan sebuah karakteristik dari suatu layanan, yang diberikan kepada para pengguna atau user untuk akses tersebut. QOS atau quality of service itu sendiri dipakai dalam pengukuran atau menghitung sebuah kelompok data atau paket dari atribut kinerja sebuah jaringan yang sudah diperinci dan diberikan dengan suatu layanan yang akan diberikan dan dipakai pengguna, seperti layanan Hotspot [4]. Parameter QOS yang umumnya digunakan dalam mengukur jaringan dan akan digunakan pada penelitian ini, yaitu [8]:

1. Packet loss diartikan bahwa hilangnya sejumlah paket data pada jaringan komputer selama proses transmisi paket data mencapai tujuannya dengan kata lain pada saat sedang melakukan akses internet.

$$loss = \frac{\text{paket dikirim} - \text{paket diterima}}{\text{paket data yang dikirim}} \times 100\%$$

Tabel 1 Penilaian Loss

Kualitas Loss	Loss(%)	Nilai
Sangat bagus	0% - 2%	4
Bagus	3% - 14%	3
Sedang	15% - 24%	2
Jelek	>25%	1

2. Throughput merupakan nilai sebenarnya dari jaringan untuk pengiriman data. Umumnya throughput disangkut pautkan dengan bandwidth yang asli saat waktu tertentu dan pada kondisi serta internet tersebut.

$$Throughput = \frac{\text{jumlah data yang diterima}}{\text{lama pengamatan}}$$

Tabel 2 Penilaian Throughput

Kualitas Throughput	Throughput	Nilai
Sangat bagus	>2,1 Mbps	4
Bagus	701 – 1200 kbps	3
Sedang	339 – 700 kbps	2
Jelek	0 – 338 kbps	1

3. Delay merupakan waktu tunda yang dimiliki sebuah paket yang terjadi dari proses transmisi dari satu titik ke titik lain yang menjadi tujuannya.

$$delay = \frac{\text{total delay}}{\text{total paket diterima}}$$

Tabel 3 Penilaian Delay

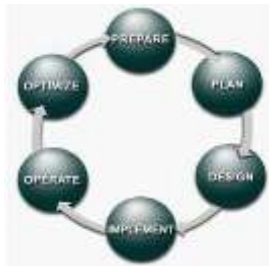
Kualitas Delay	Nilai Delay	Nilai
Sangat bagus	<150 ms	4
Bagus	150 – 300 ms	3
Sedang	301 – 450 ms	2
Jelek	>450 ms	1

**2.6. Metode Pengembangan Sistem**

Pada penelitian ini memakai Metode PPDIIO, Cisco telah berhasil membuat atau

menciptakan sebuah bentuk alur dari sebuah pembangunan sistem jaringan, dimana pada formula itu sendiri terbagi menjadi enam fase.

Enam fase tersebut adalah tahap *Prepare* atau persiapan, *Plan* atau perencanaan, *Design* atau desain, *Implement* atau implementasi, *Operate* atau operasi, dan *Optimize* atau optimasi [9]. Tahapan tersebut dikenal dengan PPDIIO, dapat dilihat pada gambar 3.



Gambar 3 Metode Pengembangan Sistem PPDIIO

1. *Prepare*

*Prepare* adalah ialah dilakukan persiapan kerja dengan baik, dimana peneliti merencanakan hal-hal teknis yang nantinya akan di terapkan oleh penulis.

2. *Plan*

*Plan* adalah ialah dilakukan perencanaan untuk penelitian yang menyangkut keseluruhan, seperti metode penelitian yang akan dipakai, desain topologi jaringan, dan lain-lain.

3. *Design*

*Design* ialah proses rencana terkait pengguna serta *design* daii bentuk topologi yang akan diimplementasikan. .

4. *Implement*

*Implement* ialah tahapan dari proses diterapkannya semua rencana, seperti konfigurasi manajemen dengan metodenya dan didesign sesuai dengan hasil analisa oleh penulis.

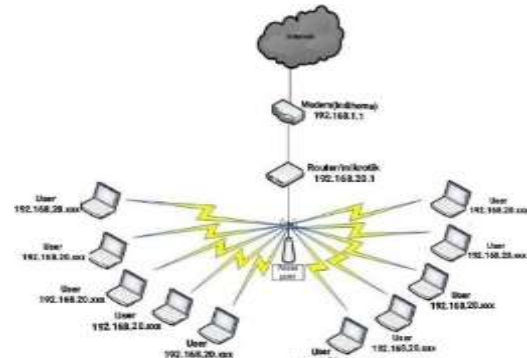
5. *Operate*

*Operate* ialah tahapan dimana akan diuji terkait sistem manajemen *bandwidth* yang dikonfigurasi sebelumnya, akan dilihat sistem yang berjalan secara langsung, dan sekaligus melakukan pengujian dari *hotspot* secara langsung oleh pengguna. Pada tahap ini juga dimanfaatkan untuk mencari jalan keluar dari masalah yang dapati, dan juga akan dilakukan pemantauan pada sistem yang dibuat..

6. *Optimize*

*Optimize* adalah proses terakhir, sesudah melakukan monitoring dan akan diupdate sistem yang dibuat sebelumnya, hal ini dapat berguna agar ketika terjadi kesalahan konfigurasi sebelumnya, akan dilakukan pembaharuan untuk mengatasi kesalahan konfigurasi sebelumnya.

2.7. Desain Topologi Jaringan



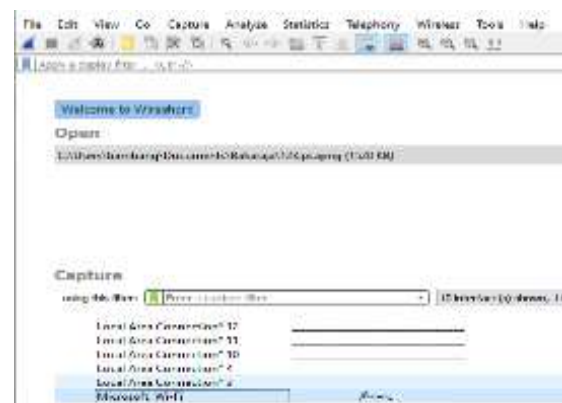
Gambar 4 Desain Topologi Jaringan

Pada gambar 3.2. Dimana peran dari pada *mikrotik* (*router*) akan menjadi pusat setingan untuk melakukan manajemen *bandwidth* dengan konfigurasi HTB, *mikrotik* ini sendiri terhubung langsung ke modem, dan peran untuk membagikan ip dan akses internet yang sudah di konfigurasi sebelumnya, pada pengguna ke akses *point* untuk Wi-Fi .

3. HASIL DAN PEMBAHASAN

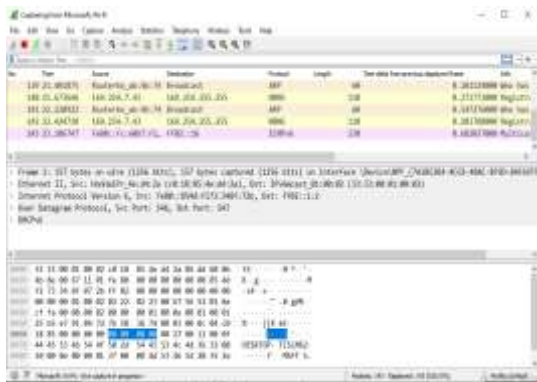
Pada penelitian pemantauan dan pengambilan data menggunakan *wireshark* , dan melakukan filter aktifitas dari pengguna *hotspot*, untuk melihat aliran paket dari pengguna Wi-Fi dengan akses yang dituju.

Cara mencari paket *loss* ketikkan "*tcp.analysis.lost\_segment*" dan dilanjutkan pada menu *statistics* lalu pilih *capture* file *propertis* untuk melihat paket data yang dikirim (*packet*) dan paket yang di terima dilihat pada bagian *displayed* (*packets*), untuk mencari *throughput* ketikkan "*ip.addr== ip yang mau dilihat throughput nya*", dilanjutkan pada menu *statistics* lalu pilih *capture* file *propertis* untuk melihat jumlah data yang diterima (*bytes*) dan lama waktu pengamatan (*time span,s*), sedangkan untuk mencari *delay* ketikkan "TCP" sesudah filter IP sebelumnya, dilanjutkan pada tab menu file dan dilanjutkan export *packet dissections* dan tekan *as Csv*, maka hasil akan tersimpan dengan tipe *axcel* untuk dilihat waktu (*time*) dari aktivitas ip pengguna yang sudah difilter untuk dikelola dalam mencari *delay*.



Gambar 5 Tampilan Depan *Wireshark*  
 Pada gambar diatas, dimana tampilan awal

dari *wireshark* ,merupakan tampilan yang melihat *interface* internet yang ada di laptop.



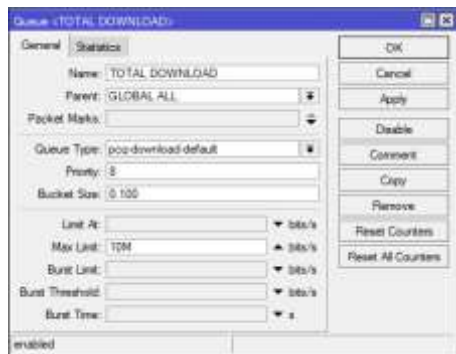
Gambar 6 Monitoring *Wireshark*

Pada gambar diatas, dimana aliran paket data internet dalam satu segmen jaringan Wi-Fi dapat dimonitoring secara detail dimulai dari ip src, ip dst, protokol yang digunakan.

### 3.1. Konfigurasi *QueueTree* Dan Penerapan HTB

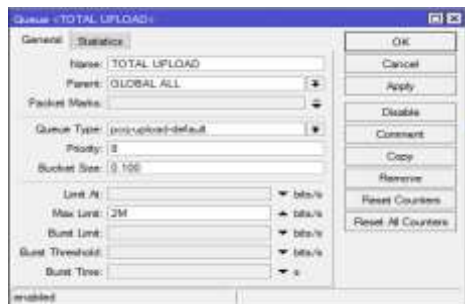
#### 3.1.1 Konfigurasi Untuk Membuat *Parent*

Untuk pembuatan *parent* berfungsi mengatur perhitungan pembagian *bandwidth* dan distribusi *bandwidth* berdasarkan akses yang tertentu.



Gambar 7 Konfigurasi Pembuatan *Parent* total *download*

Pada gambar diatas, setelah membuka menu *queue* di *router* tekan tanda + berwarna biru, pada *tab name* silahkan sesuaikan nama dari *inner* atau *parent* yang mau dibuat, dimana penulis menandai dengan nama *TOTAL DOWNLOAD*.



Gambar 8 Konfigurasi Pembuatan *Parent* Total *Upload*

Pada gambar diatas, konfigurasi dilakukan sama dengan langkah sebelumnya hanya saja disini *inner* yang dibuatkan dikhususkan untuk akses

*upload*.



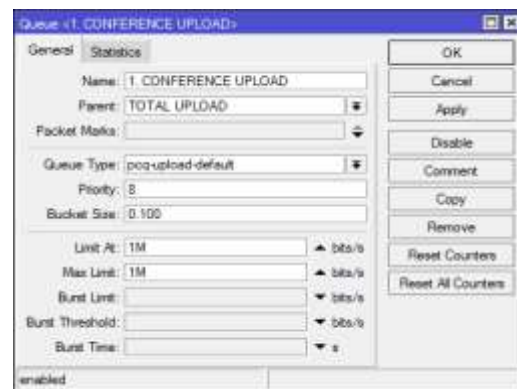
Gambar 9 Pembuatan *Inner Download* Video *Conference*

Pada gambar diatas, untuk nama *inner* yang dibuat, ditandai dengan nama *1.CONFERENCE DOWNLOAD* hal itu bertujuan untuk pembuatan *leaf* atau *child queue* untuk akses *zoom* dan *google meet*, pembagian *bandwidth* sesuai dengan perhitungan distribusi dari *parent* yang dibuatkan.



Gambar 10 Pembuatan *Inner Download* Sosial Media

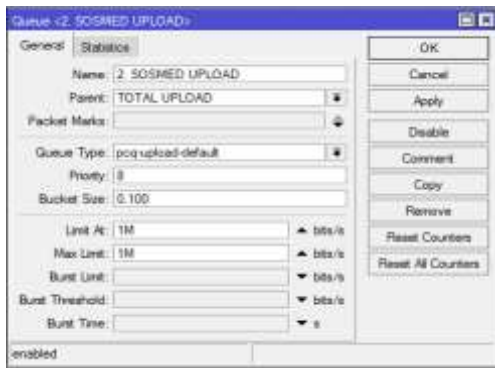
Pada gambar diatas, konfigurasi dilakukan seperti sebelumnya hanya saja pada *name* dari *inner* yang dibuat diganti *2. SOSMED DOWNLOAD*.



Gambar 11 Pembuatan *Inner Upload* Video *Conference*

Pada gambar diatas, dimana untuk nama *inner* yang dibuat, ditandai dengan nama *1.CONFERENCE UPLOAD* bertujuan untuk pembuatan *leaf* atau *child queue* untuk akses *zoom* dan *google meet* pembagian *bandwidth* sesuai dengan perhitungan distribusi dari *parent* yang

dibuatkan

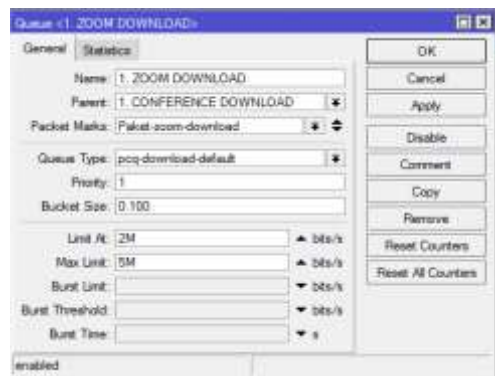


Gambar 12 Pembuatan *Inner* untuk Upload Sosial Media

Pada gambar diatas, konfigurasi yang dilakukan sama dengan sebelumnya, hanya saja yang diubah nama untuk *inner* yaitu 2. SOSMED UPLOAD.

### 3.1.2 Konfigurasi Untuk Membuat *Leaf*

Untuk pembuatan *leaf* berfungsi sebagai penentuan berapa besar *max limit* dan *limit at* dari akses yang dituju berdasarkan pembagian *traffic mangel* dan *parent* yang menjadi parameternya



Gambar 13 Pembuatan *Leaf* untuk Zoom Download

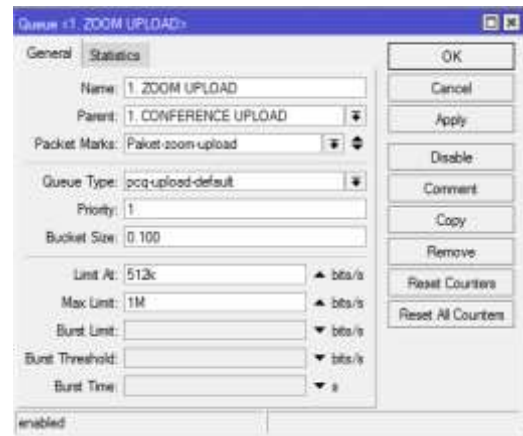
Pada gambar diatas, dimana *leaf* atau *child queue* dibuatkan dengan nama 1.ZOOM DOWNLOAD, *parent* nya pilih 1.CONFERENCE DOWNLOAD, *packet mark* pilih paket-zoom-download.



Gambar 13 Pembuatan *Child* untuk Facebook Download

Pada gambar diatas, sama dengan yang sebelumnya, yang diubah nama *child queue* 1.FACEBOOK DOWNLOAD, *parent* nya diganti

atau diarahkan ke 2. SOSMED DOWNLOAD, *packet mark* diganti Paket-facebook-download.



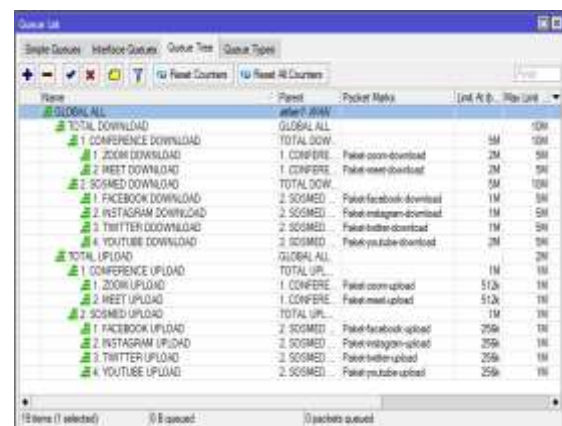
Gambar 14 Pembuatan *Child* untuk Zoom Upload

Pada gambar diatas, dimana *leaf* atau *child queue* dibuatkan dengan nama 1.ZOOM UPLOAD, *parent* nya pilih 1.CONFERENCE UPLOAD, *packet mark* pilih paket-zoom-upload, *queue type* pilih PCQ-upload-default.



Gambar 15 Pembuatan *Child* untuk Facebook Upload

Pada gambar diatas, sama dengan yang sebelumnya, yang diubah nama *child queue* 1.FACEBOOK UPLOAD, *parent* nya diganti atau diarahkan ke 2. SOSMED UPLOAD.



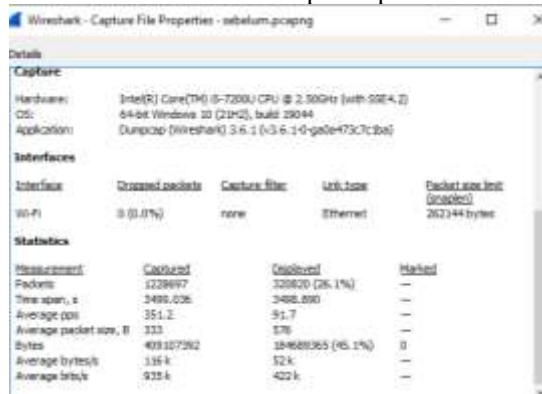
Gambar 16 Tampilan *Queue Tree*

Pada gambar diatas, setelah langkah konfigurasi yang dibuat sebelumnya selesai, maka daftar *queue tree* akan muncul antrian yang tersusun

berdasarkan induk *queue* dan anak *queue*

### 3.2. Perhitungan QOS Sebelum dan Sesudah Penerapan HTB

Pada tahap ini, dimana akan di lakukan perhitungan manual dengan rumus perhitungan QOS untuk masing-masing parameter, untuk sistem sebelum dan sesudah penerapan HTB.



Gambar 17 Summary Throughput Sebelum Penerapan HTB

Pada gambar diatas, hasil tangkapan atau rekaman dari *wireshark*.

#### 1. Perhitungan *throughput* sebelum HTB.

$$\text{Throughput} = \frac{\text{jumlah data yang diterima}}{\text{lama pengamatan}}$$

$$\text{Throughput} = \frac{409107392 \text{ B}}{3499,036 \text{ s}}$$

$$\text{Throughput} = 116.920029402384 \text{ Byte} \times 8$$

(diubah dari *Byte* ke *bit*)

$$\text{Throughput} = 935360235219072 \text{ Bit} \times 1000$$

$$\text{Throughput} = 935 \text{ kbps}$$



Gambar 18 Summary Loss Sebelum Penerapan HTB

Pada gambar diatas, adalah hasil tangkapan atau rekaman dari *wireshark*

#### 2. Perhitungan *loss* sebelum penerapan HTB

$$\text{loss} = \frac{\text{paket dikirim} - \text{paket diterima}}{\text{paket data yang dikirim}} \times 100\%$$

$$\text{loss} = \frac{(1228697 - 1222618)}{(1228697)} \times 100\%$$

$$\text{loss} = 0,5\%$$

#### 3. Perhitungan *delay* sebelum penerapan HTB

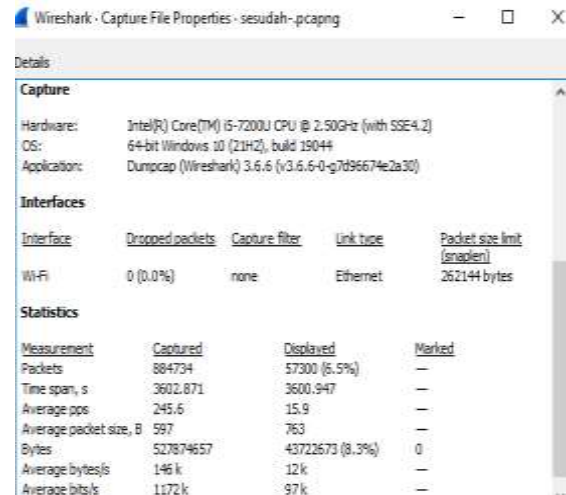
Pada hitungan *delay* seperti yang sudah di jelaskan pada bab 3 untuk pengambilan data pada *wireshark*, dikelola melalui *excel*.

$$\text{delay} = \frac{\text{total delay}}{\text{total paket diterima}}$$

$$\text{delay} = \frac{3498,890099\text{s}}{1228697}$$

$$\text{delay} = 0,002847664 \times 1000 \text{ (untuk mengubah S ke Ms)}$$

$$\text{delay} = 2,847663605 \text{ Ms}$$



Gambar 19 Summary Throughput Sesudah Penerapan HTB

#### 4 Perhitungan *throughput* penerapan HTB

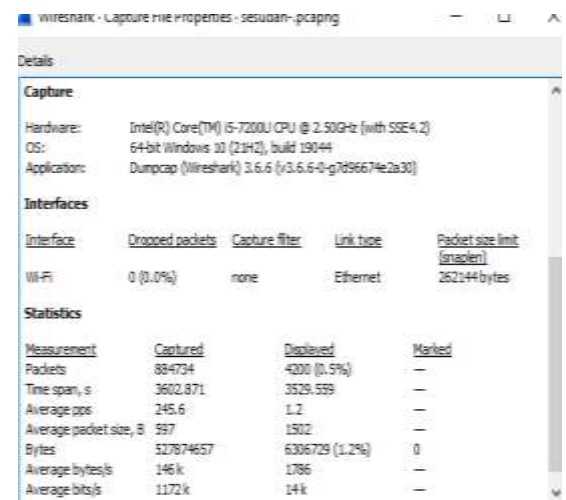
$$\text{Throughput} = \frac{\text{jumlah data yang diterima}}{\text{lama pengamatan}}$$

$$\text{Throughput} = \frac{527874657 \text{ B}}{3602.871 \text{ s}}$$

$$\text{Throughput} = 146,5150034514141 \times 8 \text{ (ubah Byte ke bit)}$$

$$\text{Throughput} = 1.1721200276113128 \text{ bps} \times 1000$$

$$\text{Throughput} = 1172 \text{ kbps}$$



Gambar 20 Summary Loss Sesudah Penerapan HTB

#### 5 Perhitungan *loss* sesudah penerapan HTB

$$\text{loss} = \frac{\text{paket dikirim} - \text{paket diterima}}{\text{paket data yang dikirim}} \times 100\%$$

$$\text{loss} = \frac{(884734 - 880.534)}{(884734)} \times 100\%$$

$$\text{loss} = 0,5\%$$

6 Perhitungan *delay* sesudah penerapan HTB

$$\text{delay} = \frac{\text{total delay}}{\text{total paket diterima}}$$

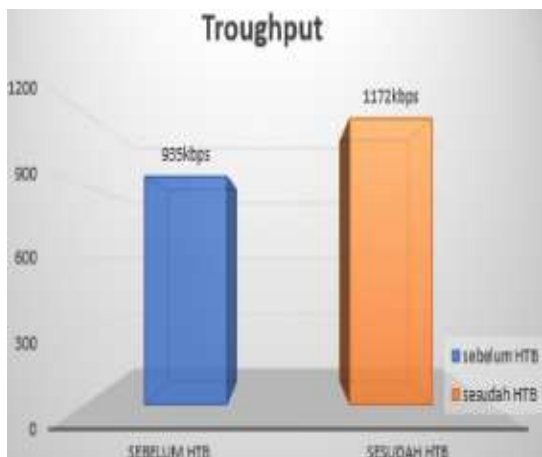
$$\text{delay} = \frac{884501}{3600,947111s}$$

$$\text{delay} = 0,004071162s \times 1000 \text{ (untuk mengubah S ke Ms)}$$

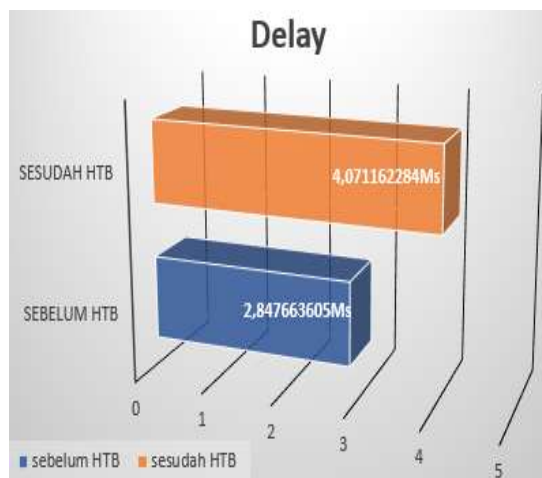
$$\text{delay} = 4,071184Ms$$

### 3.3. Hasil QOS Sebelum dan Sesudah Penerapan HTB

Penulis mengambil data dari sistem hotspot jaringan yang sudah ada pada Lab Rekayasa Perangkat Lunak dan sistem yang sudah dikonfigurasi menggunakan HTB, hasil pengambilan data didapatkan hasil QOS yang digunakan untuk dianalisis datanya. Berikut data yang didapati selama 2 hari (1jam) pengambilan data sistem sebelumnya dan sesudah penerapan HTB.



Gambar 21 Diagram *Troughput* Sebelum Sesudah HTB



Gambar 22 Diagram *Delay* Sebelum dan Sesudah HTB



Gambar 23 Diagram Loss Sebelum dan Sesudah HTB

Pada gambar 21, 22, dan 23 adalah trafik dalam bentuk diagram hasil perhitungan nilai QOS (*throughput*, *delay*, dan *loss*) dari sistem sebelum dan sesudah penerapan HTB.

### 3.4. Analisis

Dari hasil pengambilan data keduanya (sistem sebelum dan sesudah penerapan HTB) masih dikatakan sama-sama efektif dalam membagi *bandwidth* ke pengguna, akan tetapi untuk mengatur *priority/prioritas* akses tertentu, yang dimana ketika *user* aktif secara bersamaan dengan mengakses akses yang berbeda-beda tidak saling mengganggu maka dengan menggunakan HTB lebih efektif. Dengan artian bahwa untuk melakukan manajemen *bandwidth* yang lebih spesifik dan mengarah ke akses tertentu, maka menggunakan HTB dirasa lebih efektif dikarenakan *bandwidth* dapat berjalan berdasarkan akses yang diprioritaskan, dan untuk sisa *bandwidth* yang ada maka *bandwidth* tambahan akan diberikan ke akses dengan prioritas tertinggi, dan akan dibagikan lagi bila akses tersebut sudah mencapai *max-limit* nya, maka akan diberikan lagi ke akses yang memiliki proritas tertinggi ke 2 dan berlaku untuk selanjutnya.

Akan tetapi pada kondisi tertentu dimana seluruh akses yang dikonfigurasi diakses oleh semua *user* maka dengan sendirinya *router* akan membagi sesuai dengan *limit-at* yang diberikan, sehingga untuk akses yang diprioritaskan akan memakai *bandwidth* sesuai dengan yang diberikan *limit-at* nya, belum ada tambahan *bandwidth* dari *router* lagi karena *bandwidth* sudah terpakai semua dan *token* yang ada pada *bucket* sudah terpakai semua, *token* atau *bandwidth* tambahan akan di berikan lagi ke *bucket* yang diprioritaskan apabila tidak ada akses lagi dari pengguna ke akses yang berbeda.

## 4. PENUTUP

### 4.1. Kesimpulan

Dari hasil penelitian ini, dilakukan selama 2 hari ( 1 hari sistem sebelum dan 1 hari sesudah penerapan metode HTB ) didapati kesimpulan



sebagai berikut:

1. Sistem HTB berjalan berdasarkan pembagian yang dilakukan menggunakan *port* dan *content* dari akses dirasa cukup efektif dan Cara kerja dari HTB pada *queue tree* sendiri dimana terbagi dari dari *inner queue* yang dikenal dengan *parent queue* atau *queue* induk, yang bertugas untuk mendistribusikan *bandwidth* sesuai dengan jalur yang sudah dikonfigurasi ke *leaf queue*, dan *leaf queue* yang biasa dikenal dengan anak *queue*, yang bertugas untuk memberikan *bandwidth* ke pengguna.
  2. Dari hasil perhitungan QOS jaringan sebelumnya yang belum menggunakan HTB pada *queue tree* didapatkan hasil dari 1 hari penelitian, untuk *throughput* index 3 “BAGUS”, *delay* 4 “SANGAT BAGUS” dan *loss* 4 “SANGAT BAGUS” dan hasil perhitungan QOS jaringan yang menggunakan HTB pada *queue tree* didapatkan hasil untuk *throughput* index 3 “BAGUS”, *delay* index 4 “SANGAT BAGUS” dan untuk *loss* index 4 “SANGAT BAGUS”.
  3. Penggunaan *router/mikrotik* RB951Ui-2HND dalam memajemen *bandwidth* dalam mengoptimalkan *bandwidth* dengan metode HTB secara efektif. Untuk pembagian *bandwidth* *router* akan menjadikan *leaf queue* sebagai acuannya, sehingga untuk *priority* pada *inner queue* tidak berpengaruh, karena yang dijadikan acuan pertama kali adalah *priority*, *maxlimit* dan *limit-at* adalah dari *leaf queue*.
  4. Peran *limit-at* untuk memastikan bahwa dalam keadaan yang buruk *user* akan tetap mendapatkan minimal *bandwidth* yang diberikan sedangkan *maxlimit* sebaliknya, *user* tidak akan diberikan alokasi *bandwidth* yang melewati batas yang diberikan.
  5. Dalam memantau aktifitas paket dan pengambilan data menggunakan *wireshark* lebih perhatikan *script* cara filter data yang ingin digunakan, seperti *throughput* ”TCP”, *loss* “TCP.ANALYSIS.LOSS\_SEGMEN” dan *delay* ”TCP”.
  6. Hasil yang didapati untuk sistem sebelum penerapan HTB dan sesudah, dirasa masih sama-sama efektif berdasarkan hasil pengambilan data yang dilakukan, akan tetapi untuk pembagian akses atau trafik yang diprioritaskan lebih baik ketika menggunakan HTB pada *queue tree*.
- [2] Usman, S. (2018). Skripsi Implementasi Teknologi *Cloud Private Network* Berbasis Teknologi Virtualisasi.
  - [3] Toresa. (2020). Analisa QOS dengan *Simple Queue* , *Queue Tree* , dan *hierarchial Token Bucket* ( Studi Kasus Pro Net Bangkinang ). 1–15. <https://doi.org/10.35314/isi.v5i1.1289>.
  - [4] Budin, S., & Riadi, I. (2019). *Traffic Shaping* Menggunakan Metode HTB ( *Hierarchial Token Bucket* ) Pada Jaringan *Nirkabel*. 1(3), 144–152. <https://doi.org/10.12928/biste.v1i3.1100>.
  - [5] Haerani, R. (2020). Analisa Penerapan *Hierarchial Tokken Buket* Untuk Optimalisasi *Management Bandwith* Pada Server *Ubuntu*. *Jurnal Bumigora Information Technology* (BITE), 2(2), 139–148. <https://doi.org/10.30812/bite.v2i2.917>.
  - [6] Ketut Gede, D. (2020). Penerapan Manajemen *Bandwidth* Menggunakan Metode *Hierarchial Token Bucket* Pada Layanan *Hotspot Mikrotik* Undiksha. 5(1), 146–154. <https://doi.org/10.24114/cess.v5i1.14959>.
  - [7] Subekti, Z. mutaqin. (2018). Masalah Penelitian 1.2.1 Identifikasi Masalah. 8(1), 33–42. <https://doi.org/10.30741/wiga.v7i1.331>.
  - [8] Mahmud, P. T. (2020). *Sniffing* Jaringan Menggunakan *Wireshark*. 5–8. <https://doi.org/10.31219/osf.io/h5wu7>.
  - [9] Tiphon, 1999, “*Telecommunications and Internet Protocol Harmonization Over Networks* (TIPHON) *General aspects of Quality of Service* (QoS)”, DTR/TIPHON-05006 .(cb0010cs.PDF).1999., 3(1), 1. <https://doi.org/10.33372/stn.v3i1.208>.
  - [10] S.Imam. (2017). Penerapan Metode PPDIIO Dalam Pengembangan LAN dan WLAN. 07(01), 65–73.

#### DAFTAR PUSTAKA

- [1] Helmy. (2017). Analisis dan Perbandingan Implementasi Metode *Simple Queue* Dengan *Hierarchial Token Bucket* ( HTB ) ( Studi Kasus Makosat Brimob Polda Kalbar ).