

## **Analisis QOS Jaringan Internet ITK dengan Metode Hierarchical Token Bucket**

**Riska Kurniyanto Abdullah<sup>1\*</sup>, Satrio Putra Wibowo<sup>2</sup>, Tegar Palyus Fiqar<sup>3</sup>**

Informatics, Institut Teknologi Kalimantan, Balikpapan, Indonesia.

\*Email Corresponding Author: riska.abdullah@lecturer.itk.ac.id

### **Abstract**

*Kalimantan Institute of Technology (ITK) is a state university in Balikpapan City, East Kalimantan. The quality of the internet network in ITK is not adequate, and this research analyzes Quality of Services (QoS) with the Hierarchical Token Bucket (HTB) method. The test results show that the average value of throughput on download is 853 Kbps (good), there is an increase of 7.8%, packet loss is not very influential, delay is 7.7 ms (very good), there is a very significant decrease in download delay of 31.6%, and jitter is 2,109 ms (bad), not too middle. HTB is good enough to improve QoS, especially on throughput and delay on the download channel on the ITK Network. The HTB method technically helps with the distribution of structured bandwidth management, reducing failures and delays in data delivery. This research shows that HTB can improve the quality of network services at ITK. The results of QoS measurements, such as throughput, packet loss, delay, and jitter, can be a reference for UPT TIK ITK to improve the quality of network services.*

**Keywords:** Quality of Service; Bandwidth Management; Network; Hierarchical Token Bucket

### **Abstrak**

*Institut Teknologi Kalimantan (ITK) adalah perguruan tinggi negeri di Kota Balikpapan, Kalimantan Timur. Kualitas jaringan internet di ITK belum memadai, dan penelitian ini menganalisis Quality of Services (QoS) dengan metode Hierarchical Token Bucket (HTB). Hasil uji coba menunjukkan bahwa nilai rata-rata Throughput pada download adalah 853 Kbps (bagus) terjadi peningkatan 7.8%, Packet Loss tidak terlalu berpengaruh, Delay 7,7 ms (sangat bagus) terjadi penurunan yang sangat signifikan pada delay download yaitu 31.6%, dan Jitter 2.109 ms (jelek) belum terlalu berpengaruh. HTB cukup baik untuk meningkatkan QoS, terutama pada Throughput, dan Delay pada kanal download di Jaringan ITK. Metode HTB secara teknis membantu pembagian manajemen bandwidth yang terstruktur, mengurangi kegagalan dan keterlambatan pengiriman data. Penelitian ini menunjukkan bahwa HTB dapat meningkatkan kualitas layanan jaringan di ITK. Hasil pengukuran QoS, seperti throughput, packet loss, delay, dan jitter ini, dapat menjadi acuan UPT TIK ITK untuk meningkatkan kualitas layanan jaringan.*

**Kata kunci:** Quality of Services; Manajemen Bandwidth; Jaringan; Hierarchical Token Bucket

### **1. Pendahuluan**

Jaringan Komputer adalah sebuah sistem yang terdiri dari komputer, *software* dan perangkat jaringan yang bekerja bersama untuk mencapai suatu tujuan yang sama. Agar dapat mencapai suatu tujuan yang sama, setiap bagian dari jaringan komputer meminta dan memberikan sebuah layanan (*services*). Pihak yang menerima atau meminta sebuah layanan dikatakan sebagai *client* dan pihak yang memberikan sebuah layanan dikatakan sebagai *server*, hal ini disebut dengan sistem *client-server* [1]. Sementara itu, pada Jaringan internet jaringannya bukanlah merupakan sesuatu hal yang baru lagi pada saat ini, banyak individu yang menggunakannya termasuk pada lembaga pendidikan perguruan tinggi. Agar mencukupi kebutuhan civitas tersebut dalam menjalankan kegiatan *browsing*, *streaming*, *upload* dan *download* dibutuhkan manajemen *bandwidth* yang baik sehingga setiap pengguna yang aktif tidak dirugikan [2].

Sering kali muncul permasalahan dalam jaringan internet ketika banyak pengguna yang mengaksesnya secara bersamaan. Dampak dari situasi ini adalah jika tidak ada pengaturan

yang tepat, akan terjadi kemacetan yang pada akhirnya dapat mengakibatkan seluruh pengguna tidak dapat terhubung dengan jaringan internet tersebut. Oleh sebab itu, dalam infrastruktur jaringan penting untuk memiliki manajemen *bandwidth* yang baik, hal ini bertujuan untuk mengatur lalu lintas data agar berjalan lancar dan efisien, sehingga pengiriman data dapat dilakukan dengan cepat dan tepat sasaran [3]. Berdasarkan hasil pengukuran dan pengamatan langsung jaringan internet ITK saat ini masih belum maksimal sehingga perlu untuk dilakukan analisis QoS guna meningkatkan layanan jaringan internal kampus, terutama layanan jaringan internet untuk civitas. QoS sendiri tidak mengenakan batasan, melainkan menjaga kualitas *bandwidth* sehingga semua pengguna yang terhubung dalam jaringan memperoleh kualitas internet yang konsisten dan stabil [4].

Dalam analisis QoS yang dilakukan dalam penelitian ini, digunakan metode *Hierarchical Token Bucket* (HTB). Metode ini membagi *bandwidth* secara bertingkat, dimulai dari tingkat tertinggi sebagai induk hingga paket-paket dialirkan ke tingkatan yang lebih rendah berdasarkan konsep hierarki. Keunggulan utama dari pendekatan ini terletak pada kemampuan untuk mengalokasikan sisa paket yang tidak terpakai pada tingkat paling atas ke tingkatan lebih bawah dengan klasifikasi tertentu. Berbeda dengan metode PCQ dan CBQ, yang tidak memiliki mekanisme serupa dengan HTB [5]. Selain itu penelitian ini menggunakan pendekatan simulasi dan prototipe yang digunakan sebagai lingkungan penelitian dalam menerapkan percobaan [6].

Tujuan dari penelitian ini adalah untuk menyajikan data QoS di ITK yang mana belum pernah dilakukan sebelumnya. Selain itu juga dilakukan pembatasan *bandwidth* dengan metode HTB. Semua kegiatan penelitian dilakukan dalam lingkungan simulasi dan prototipe yang didesain mendekati keadaan riil yang ada di lapangan. Hal ini bertujuan agar pembagian implementasi HTB pada penelitian ini tidak mengganggu sistem yang sudah berjalan. Manfaat dari penelitian ini adalah untuk menyajikan data QoS untuk jaringan yang ada di Institut Teknologi Kalimantan, sebelum dan sesudah metode HTB diimplementasikan. Data ini penting sebagai bahan evaluasi institut dalam pengembangan jaringan selanjutnya untuk mengatasi masalah akses internet yang kurang optimal.

## 2. Tinjauan Pustaka

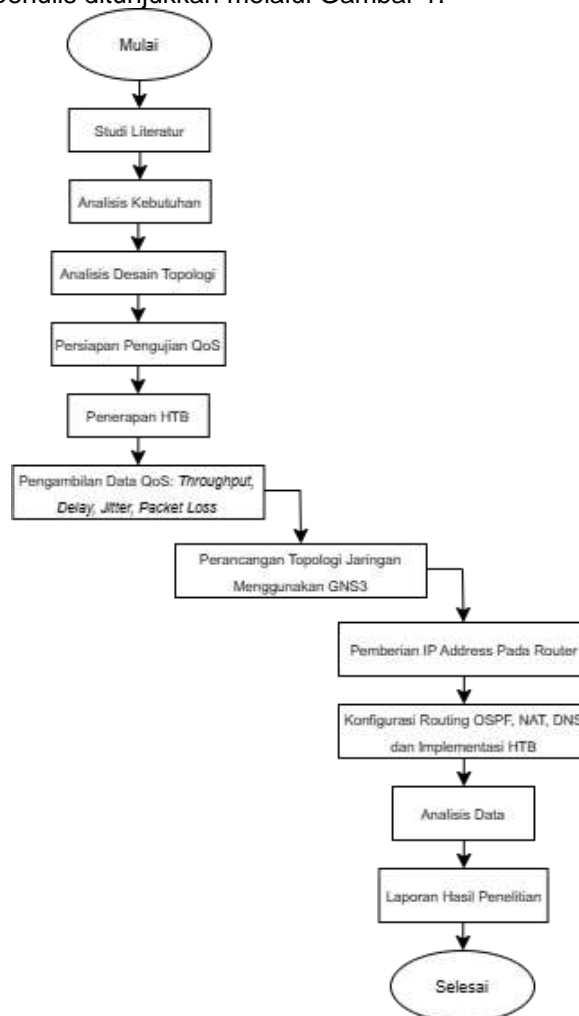
*Hierarchical Token Bucket* (HTB) merupakan sebuah metode dalam melakukan pembagian *bandwidth* yang dilakukan secara hierarki. Dengan menggunakan konsep sebuah turunan yang mana hierarki paling atas akan berperan sebagai induk kemudian *bandwidth* akan didistribusikan keturunannya yang lebih rendah. Kelebihan yang unggul dari metode ini adalah ketika sisa *bandwidth* yang tidak terpakai di hierarki paling atas maka dapat dialokasikan ke hierarki yang paling rendah dengan klasifikasi-klasifikasi tertentu [5]. Selain itu, metode *Hierarchical Token Bucket* ialah merupakan cara untuk membatasi trafik pada setiap klasifikasinya sehingga hal ini dapat membuat *bandwidth* yang tidak terpakai dapat digunakan oleh prioritas lainnya. Lalu terdapat tiga tipe *class* dalam HTB, yaitu: *root*, *inner* dan *leaf*. *Root class* yang berada paling atas memprioritaskan semua trafik harus melewati *class* ini. *Inner Class* memiliki *Parent Class* dan *Child Class*. Sedangkan *Leaf Class* yang berada dalam *class* paling dasar dari *layer* harus lebih tinggi melalui klasifikasi yang harus digunakan melalui *filter*, sehingga dapat membedakan jenis lalu lintas jaringan dan prioritasnya [7]. Selanjutnya diantara kedua metode tersebut di atas terdapat juga metode *Peer Connection Queue* (PCQ). PCQ merupakan sebuah metode pembagian *bandwidth* dengan cara membagi sesuai dengan masing-masing *subclass* atau *subqueue* pada masing-masing *client*. Berdasarkan klasifikasinya metode ini umum digunakan karena dapat diimplementasikan dengan mudah untuk membatasi *bandwidth client* secara dinamis dengan menggunakan konfigurasi yang sederhana [5]. Selain dari hal tersebut terdapat juga metode CBQ, Kelemahan dari metode CBQ ini adalah alokasi *bandwidth* dibagi secara adil jika paket semua antrean memiliki sama (atau sebanding) ukuran [8].

Penelitian yang dilakukan oleh Fahad dkk [9] dengan judul Implementasi Metode *Hierarchical Token Bucket* dan Penjadwalan Otomatis dalam Pembagian *Bandwidth* di Sanggar Kegiatan Belajar. Peneliti menggunakan HTB untuk solusi masalah internet yang tidak merata. Sementara itu, berdasarkan penelitian L. Joko Susanto dkk [10] dengan judul Pengembangan Perancangan Jaringan *Local Area Network* (LAN) Di RSIA Anugrah Medical Centre Metro pada penelitiannya diterapkan manajemen *bandwidth* dengan metode *Simple Queue*. Terlihat dalam implementasinya konfigurasi yang disediakan cukup rumit.

Pada penelitian-penelitian sebelumnya dalam mengatasi pembagian *bandwidth* yang tidak merata dengan cara yang mudah, maka diimplementasikan pembatasan *bandwidth* dengan menggunakan HTB, semua penelitian dilakukan langsung pada jaringan yang berjalan. Untuk penelitian ini metode HTB digunakan sebagai solusi atas ketimpangan *bandwidth* yang membuat akses internet *user* satu sama lain tidak pada kualitas yang sama dan terjamin. Yang menjadi perbedaan dengan penelitian sebelumnya metode HTB tetap diterapkan untuk pembatasan *bandwidth*, namun penelitian ini dilakukan dalam skala simulasi dan prototipe sehingga tidak mengganggu jaringan *native* yang sedang berjalan. Simulasi tersebut dibuat dalam lingkungan GNS3 dan beberapa *virtual machine*. Hal ini tentu sejalan dengan standar *Network Development Life Cycle* (NDLC), yang mana mengharuskan simulasi dan *prototyping* sebelum mengimplementasikan suatu konfigurasi baru pada jaringan komputer yang sudah matang sebelumnya [6].

### 3. Metodologi

Penelitian ini menggunakan beberapa tahap alur penelitian dalam melakukan analisis *Quality of Services* dengan menggunakan metode *Hierarchical Token Bucket*, tahap-tahap penelitian yang dilakukan penulis ditunjukkan melalui Gambar 1.



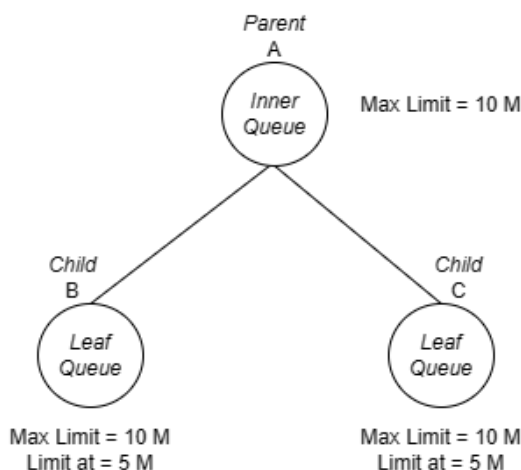
Gambar 1. Diagram Alir Penelitian

Dengan HTB diharapkan *Bandwidth* dapat diatur sesuai dengan kebutuhan. Oleh karena itu perlu diketahui *Bandwidth* adalah suatu satuan kapasitas jalur yang dapat dilalui sebuah *traffic* data dengan batasan tertentu. Definisi lain dari *bandwidth* adalah banyaknya pemakaian pengiriman data dalam satuan *bit per second* yang dapat ditransmisikan dengan medium tertentu dalam suatu waktu. *Bandwidth* dapat di analogikan menjadi sebuah jalur atau jalan raya dan data dapat didefinisikan sebagai mobil yang melintasi jalur tersebut sehingga

dapat diartikan semakin banyaknya data yang melintasi di jalur tersebut maka jalur tersebut menjadi ramai [5]. Sebuah manajemen *bandwidth* yang sempurna harus dapat membuat dan melindungi aturan tentang ketersediaan koneksi dalam perihal ini adalah internet [11]. Selain manajemen *bandwidth* berikut merupakan beberapa metode analisis QoS yaitu *Simple Queue*. *Simple Queue* merupakan sebuah antrian sederhana sebuah fitur pada mikrotik *RouterBoard* untuk melakukan manajemen *bandwidth* dalam skema jaringan sederhana. Antrian tersebut harus selalu diatur pada antarmuka keluar lalu lintas data [12]. Parameter dalam pengaturan *bandwidth* yang optimal dirangkum dalam suatu konsep yaitu QoS (*Quality of Service*). Nilai dari *Quality of Services* menurut standar kualitas jaringan dari *TIPHON (Telecommunications and Internet Protocol Harmonization Over Network)* [2]. Cara mengukur QoS ini sendiri bisa dilakukan dengan Wireshark. Menurut R. Hanipah dan H. Dhika [13] *wireshark* adalah suatu aplikasi yang berfungsi untuk memantau suatu jaringan komputer sekaligus dapat membantu memudahkan seorang *administrator* jaringan untuk melakukan pengawasan terhadap suatu jaringan komputer.

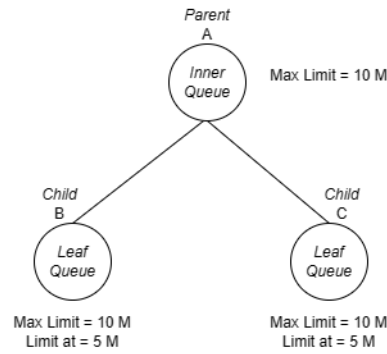
Metode evaluasi yang dilakukan pada penelitian ini adalah metode QoS dengan implementasi *Hierarchical Token Bucket* atau disingkat dengan HTB. HTB adalah salah satu cara disiplin antrean yang mempunyai tujuan untuk menerapkan *link sharing* secara presisi dan adil. Dalam konsep *link sharing* jika suatu kelas meminta kurang dari jumlah *services* yang sudah ditetapkan sebelumnya, maka sisa dari *bandwidth* akan dialokasikan ke kelas-kelas yang lain yang juga meminta *services*. *Hierarchical Token Bucket (HTB)* menggunakan *Token Bucket Filter (TBF)* sebagai estimator untuk menentukan apakah suatu kelas berada dalam keadaan *underlimit*, *atlimit* atau *overlimit*. *Estimator* ini hanya menggunakan parameter *rate*, sehingga seorang *administrator* hanya perlu mengeset *rate* yang akan diberikan ke suatu kelas [14].

*Hierarchical Token Bucket* adalah sebuah sistem untuk mengatur dan mengontrol kapasitas *bandwidth*. Sebuah jaringan internet harus memiliki pengaturan *bandwidth* yang efisien dan efektif. Untuk mendapatkan hasil tersebut diperlukan sebuah pengaturan *bandwidth* yang baik dan metode HTB (*Hierarchical Token Bucket*) merupakan metode yang memiliki efisiensi untuk menghasilkan pengaturan *bandwidth* yang optimum. HTB berperan dalam mengontrol penggunaan *bandwidth* terhadap *link* yang diberikan kepada *client*. Dengan kata lain, HTB sangat berguna untuk menjaga *rating download* dan *upload client*. Sehingga *client* tidak dapat seandainya menggunakan kapasitas *bandwidth* [15].



Gambar 2. Konsep HTB

Pada Gambar 2 menunjukkan penjelasan dari penggunaan *Hierarchical Token Bucket* pada manajemen *bandwidth*. Pada *Inner Queue* akan diberikan *max limit* nya saja, dikarenakan dia berperan sebagai kelas *parent* sehingga tidak membutuhkan *limit at* dan *priority* sedangkan pada *Leaf Queue* dimana akan diberikan *max limit* dan *limit at*, dikarenakan dia berperan sebagai kelas *child*. Jika kedua kelas *child* ini melakukan *download* secara bersamaan, maka mereka akan mendapatkan *bandwidth* yang terbagi rata sesuai dengan *limit at* nya yakni per kelas *child* akan mendapatkan 5 mbps karena total *bandwidth* yang disediakan pada kelas *parent* sebesar 10 mbps.



Gambar 3. Inner Limit HTB

Pada Gambar 3 menunjukkan penjelasan dari penggunaan *Hierarchical Token Bucket* pada manajemen *bandwidth*. Pada *Inner Queue* akan diberikan *max limit* nya saja, dikarenakan dia berperan sebagai kelas *parent* sehingga tidak membutuhkan *limit at* dan *priority* sedangkan pada *Leaf Queue* dimana akan diberikan *max limit* dan *limit at*, dikarenakan dia berperan sebagai kelas *child*. Jika kelas *child B* menggunakan *bandwidth* sebesar 2 mbps maka sisa *bandwidth* dari kelas *child B* akan diberikan ke kelas *child C*, sehingga kelas *child C* mendapatkan total *bandwidth* sebesar 8 mbps. Total dari *bandwidth* yang didapatkan oleh kedua kelas *child* tersebut yakni berjumlah 10 mbps sesuai dengan jumlah *max limit* yang diberikan oleh kelas *parent*.

### 3.1 Analisis Kebutuhan

Pada tahap ini dilakukan analisis kebutuhan untuk mengetahui kebutuhan spesifikasi perangkat keras dan perangkat lunak yang akan diperlukan dalam menunjang penelitian terkait Analisis *Quality of Services* Jaringan Internet Pada Institut Teknologi Kalimantan Menggunakan Metode *Hierarchical Token Bucket*.

Kebutuhan perangkat keras yang menunjang penelitian analisis *quality of services* jaringan internet pada Institut Teknologi Kalimantan menggunakan metode *hierarchical token bucket* dapat dilihat pada Tabel 1.

Tabel 1 Kebutuhan Perangkat Keras

No	Perangkat Keras	Keterangan
1.	Laptop/PC	Digunakan sebagai pengerjaan pada penelitian.
2.	Router	Digunakan sebagai alat untuk mengkonfigurasi jaringan dengan media perantaranya adalah winbox.
3.	Switch	Digunakan sebagai alat untuk menyebarluaskan atau menjembatani satu koneksi jaringan ke beberapa <i>client</i> .

Selanjutnya Kebutuhan perangkat lunak yang menunjang penelitian analisis *quality of services* jaringan internet pada Institut Teknologi Kalimantan menggunakan metode *hierarchical token bucket* dapat dilihat pada Tabel 2.

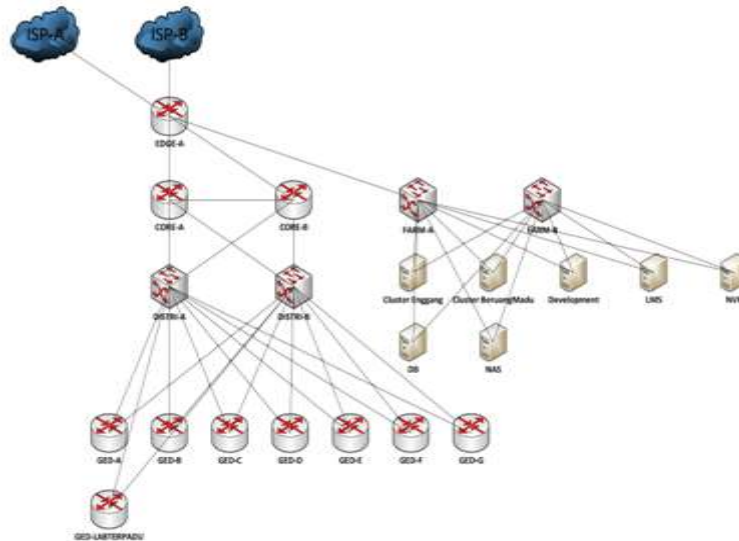
Tabel 2 Kebutuhan Perangkat Lunak

No	Perangkat Lunak	Keterangan
1.	Sistem Operasi	Windows.
2.	Winbox	Melakukan administrasi terhadap Mikrotik RouterOS dengan tampilan GUI.
3.	Wireshark	Aplikasi yang berfungsi untuk memonitoring suatu jaringan komputer secara rinci dalam bentuk GUI.
4.	Oracle VM VirtualBox	Perangkat lunak virtualisasi untuk menginstal sistem operasi dalam bentuk virtual.
5.	GNS3	Aplikasi <i>open source</i> yang digunakan untuk

No	Perangkat Lunak	Keterangan
		menyimulasikan kondisi jaringan asli dengan perangkat asli yang ada di dalamnya secara virtual.

### 3.3 Analisis Desain Topologi

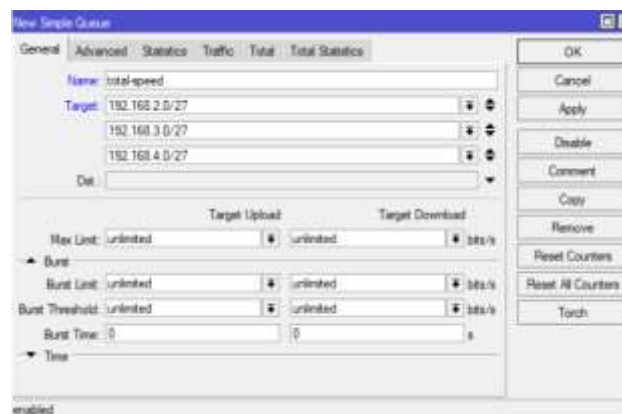
Pada tahap ini merupakan tahapan dalam menganalisis gambaran desain dari topologi jaringan pada Institut Teknologi Kalimantan. Desain dari topologi jaringan ini menggambarkan jalur dari infrastruktur jaringan yang terhubung satu sama lain. Adapun topologi jaringan pada penelitian ini dapat dilihat pada Gambar 4.



Gambar 4. Topologi Jaringan ITK

### 3.4 Implementasi HTB

Pada tahap ini adalah dilakukannya implementasi *Hierarchical Token Bucket* (HTB) pada pengaturan konfigurasi jaringan kelas *parent* dan *child* pada router yang ada di setiap gedung di Institut Teknologi Kalimantan melalui simulasi menggunakan GNS3. Adapun konfigurasi dapat dilihat pada gambar Gambar 5 dan Gambar 6.



Gambar 5. Konfigurasi Kelas Parent

Pada konfigurasi *router* di Gambar 5 untuk kelas *parent* dengan target sesuai dengan IP *Client* yang sudah dikonfigurasi pada jaringan tersebut. Pada metode *Hierarchical Token Bucket* kelas *parent* hanya membutuhkan *max limit* dan *limit at*.



Gambar 6. Konfigurasi Kelas Child

Pada konfigurasi *router* untuk kelas *child*, pada fitur *general* dimana kita memilih sasaran IP atau *client* yang akan dijadikan *child* dari *parent* yang sebelumnya telah dikonfigurasi dan menentukan *max limit* yang tidak melebihi dari kelas *parent*.

#	Name	Target	Upload Max Limit	Download Max Limit	Packet Marks
0	total_speed	192.168.2.0/27, 192.168...	7M	12M	
2	PC2	192.168.4.0/27	2M	3M	
1	PC1	192.168.3.0/27	2M	3M	
3	LAN	192.168.2.0/27	3M	5M	

Gambar 7. Tampilan Queue

Pada tampilan dari fitur *Queue list* yang telah dibuat dengan *total\_speed* sebagai kelas *parent* dan PC1, PC2, dan LAN sebagai kelas *child* dan untuk *total\_speed* sendiri memiliki *max limit* lebih besar dari pada *max limit* yang dimiliki oleh PC1, PC2 dan LAN dikarenakan *total\_speed* bertindak sebagai kelas *parent* di mana kelas tersebut nantinya akan membagi *bandwidth* yang dimilikinya dan dibagikan secara rata ke dalam kelas *child*.

### 3.5 Pengambilan Data QoS

Pada tahap pengambilan data ini dilakukan agar dapat mengetahui nilai dari parameter QoS: *Throughput*, *Packet Loss*, *Delay*, *Jitter* dari hasil analisis yang sudah didapatkan dalam melakukan pengukuran parameter QoS.

$$\begin{aligned}
 \textit{Throughput} &= \text{Total Paket Data yang Dikirim} : \text{Waktu Pengiriman Data} \\
 &= 20069424 \text{ bytes} : 99.336 \text{ s} \\
 &= 202035,757 \text{ bytes/s} \\
 &= 202035,757 \times 8 \\
 &= 1616286,059 \text{ bits/s} \\
 &= 1.578 \text{ Kbps}
 \end{aligned}$$

Interface	Current Queue	Current Size	Link Error	Packet size limit (bytes)
eth1	0 (0.0%)	none	0/0/0/0	32768 bytes

Statistics	Current	Statistical	Packet
Packets/s	20712	46 (0.2%)	---
Bytes/s	136,000	46,336	---
Average size	133.5	0.5	---
Average packet size, B	900	113	---
Bytes	2200000	1011 (0.4%)	0
Average bytes/s	136.5	46	---
Average size/s	130.9	438	---

Gambar 8. Pengambilan Data Throughput

Pada Gambar 8 merupakan tangkapan layar aplikasi *wireshark* pada saat melakukan pengambilan data untuk mengetahui nilai dari parameter *Throughput* yang didapatkan. Rumus perhitungan *Throughput* adalah **Total Paket Data yang Dikirim : Waktu Pengiriman Data**.

$$\begin{aligned}
 \text{Packet Loss} &= (((\text{Paket Dikirim} - \text{Paket Diterima}) : \text{Paket Dikirim}) \times 100) \\
 &= ((20732 - 20731) / 20732) \times 100 \\
 &= (1 / 20732) \times 100 \\
 &= 0.004 (0.0\%)
 \end{aligned}$$



Gambar 9. Pengambilan Data Packet Loss

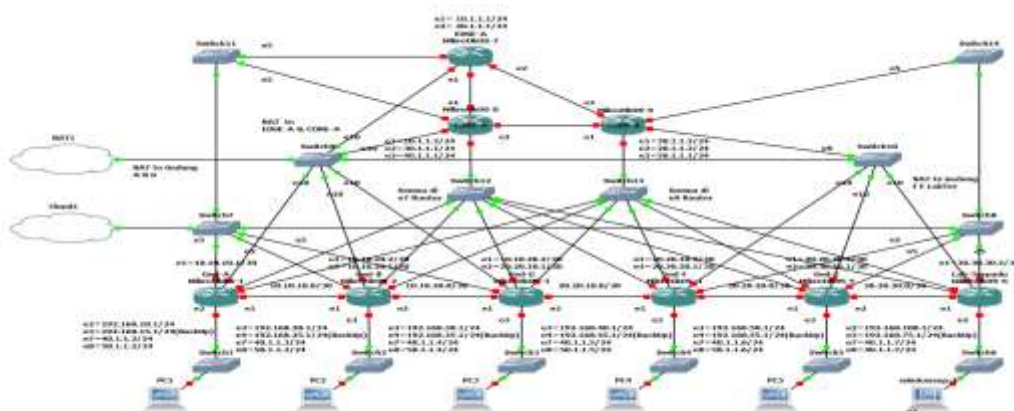
Pada Gambar 9 merupakan tangkapan layar dari aplikasi *wireshark* pada saat melakukan pengambilan data untuk mengetahui nilai dari parameter *Packet Loss* yang didapatkan. Rumus perhitungan *Packet Loss* adalah **(Paket Dikirim – Paket Diterima) dibagi Paket Dikirim dikali 100**.

$$\begin{aligned}
 \text{Delay} &= \text{Total Delay} : \text{Total Paket yang Diterima} \\
 &= 466,420527 : 25323 \\
 &= 0,018418849 \text{ s} \\
 &= 18,41 \text{ ms}
 \end{aligned}$$

$$\begin{aligned}
 \text{Jitter} &= \text{Total Variasi Delay} : (\text{Total Paket yang Diterima} - 1) \\
 &= 150326 : 25323 \\
 &= 5,936342 \text{ s} \\
 &= 5.936 \text{ ms}
 \end{aligned}$$

#### 4. Hasil dan Pembahasan

##### 4.1 Perancangan Topologi Jaringan Menggunakan GNS3



Gambar 10. Topologi Jaringan dengan GNS3



Pada Gambar 10 menunjukkan sebuah rancangan topologi jaringan yang dibuat sedemikian rupa menyesuaikan dengan topologi jaringan yang ada di ITK. Dalam topologi jaringan tersebut terdapat 6 bangunan gedung yang terdiri dari Gedung A, Gedung B, Gedung G, Gedung F, Gedung E, dan Laboratorium Terpadu. Masing-masing gedung memiliki 1 buah *router* dan *switch*. Serta terdapat 3 buah *router* utama yakni EDGE-A, CORE-A, dan CORE-B.

#### 4.2 Pemberian IP Address Pada Router

Pada tahap ini kita memasuki tahap pemberian *IP Address* yang nantinya akan diberikan ke masing-masing *ethernet* pada *router* di setiap masing-masing gedung dan juga tidak lupa untuk memberikan *IP Address* pada PC (*Client*) dari masing-masing *router*. *IP Address* yang diberikan kepada *client* nantinya dapat berupa IP statis maupun IP DHCP. Untuk pembagian *network IP Address* pada masing-masing perangkat dapat dilihat pada Gambar 11.



```

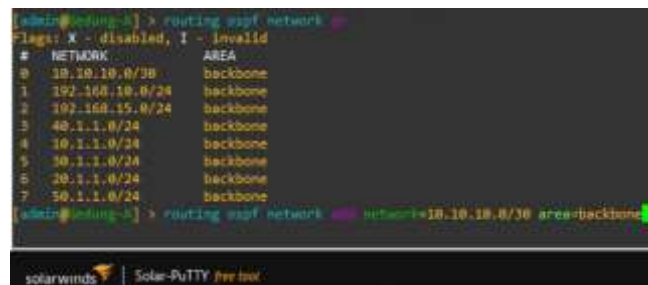
[admin@building-1] > ip address
Flags: X - disabled, I - invalid, D - dynamic
# ADDRESS NETWORK INTERFACE
0 10.10.10.1/30 10.10.10.0 ether1-ke Gedung
1 192.168.10.1/24 192.168.10.0 ether2-windows XP
2 192.168.15.1/24 192.168.15.0 ether3-Client Backup
3 40.1.1.2/24 40.1.1.0 ether7-ke CORE-A
4 192.168.122.192/24 192.168.122.0 ether8-ISP (Cloud)
5 30.1.1.2/24 30.1.1.0 ether9-ke CORE-B
[admin@building-1] > ip address address 10.10.10.1 interface ether1
  
```

Gambar 11. Pemberian IP Address Router

Pada Gambar 11 menunjukkan hasil konfigurasi *IP Address* pada *router* gedung A yang dikonfigurasi melalui terminal dari GNS3. Pada terminal tersebut terdapat sebanyak 6 *port ethernet* pada *router* gedung A dan masing-masing telah diberikan *IP Address*. Ether1 diberikan agar *router* gedung A dapat terhubung ke *router* gedung B (tetangganya). Ether 2 diberikan sebagai *client* virtual yang dijalankan pada sistem operasi windows xp. Ether 3 diberikan sebagai *client* cadangan sehingga jika terjadi *trouble* pada ether 2 maka dapat langsung dialihkan ke ether 3 pada *router*. Ether 7 diberikan agar *router* gedung A dapat terhubung dengan *router* CORE-A. Ether 8 diberikan sama halnya dengan ether 7 agar *router* gedung A dapat terhubung dengan *router* CORE-B. Ether 10 digunakan sebagai *Internet Service Provider* (ISP). *IP Address* tersebut didapatkan secara otomatis dari sumber internet.

#### 4.3 Konfigurasi Routing OSPF

Pada tahap ini merupakan tahap untuk melakukan sebuah konfigurasi pada jaringan internet yang sebelumnya sudah dibuat terlebih dahulu perancangan dan desain dari topologi yang telah dibuat sedemikian rupa menyesuaikan dari topologi yang ada pada mitra dalam penelitian ini. Dalam tahap ini akan dilakukannya *routing* menggunakan protokol *Open Shortest Path First* (OSPF), tujuan dari *routing* yaitu untuk menentukan lintasan terbaik atau yang paling efisien untuk mengirimkan paket data dari suatu jaringan ke jaringan lainnya.



```

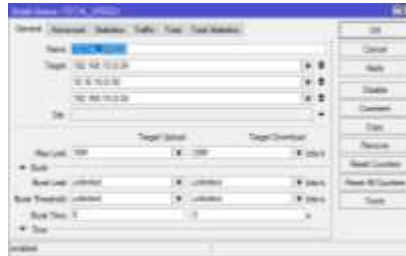
[admin@building-1] > routing ospf network
Flags: X - disabled, I - invalid
# NETWORK AREA
0 10.10.10.0/30 backbone
1 192.168.10.0/24 backbone
2 192.168.15.0/24 backbone
3 40.1.1.0/24 backbone
4 10.1.1.0/24 backbone
5 30.1.1.0/24 backbone
6 20.1.1.0/24 backbone
7 50.1.1.0/24 backbone
[admin@building-1] > routing ospf network network 10.10.10.0/30 area=backbone
  
```

Gambar 12. Routing OSPF

Pada Gambar 12 menunjukkan hasil dari konfigurasi *routing* OSPF pada *router* gedung A yang dikonfigurasi melalui terminal dari GNS3, dilihat dari hasil tersebut, sebanyak 8 *network interface* telah berhasil dikonfigurasi pada perangkat *router* mikrotik yang dimiliki oleh gedung-A.

### 4.2 Implementasi HTB

Pada tahap kali ini adalah melakukan implementasi metode *Hierarchical Token Bucket* (HTB) dalam pembagian manajemen *bandwidth* pada setingan konfigurasi jaringan di setiap perangkat *router* mikrotik yang dimiliki oleh setiap gedung di Institut Teknologi Kalimantan yang dilakukan dengan simulasi menggunakan *Graphical Network Simulator version 3* (GNS3).



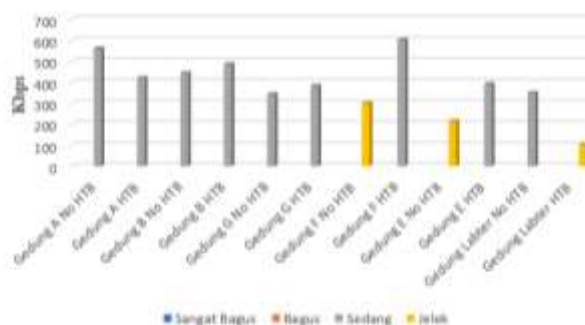
Gambar 13. Konfigurasi Parent pada Router

Pada Gambar 13 menunjukkan konfigurasi pada perangkat *router* mikrotik yang dimiliki oleh salah satu gedung untuk bagian kelas *parent* dengan memasukkan target IP *Network* nya yang disesuaikan dengan *IP Address* yang sudah diberikan di masing-masing *router*. Dalam penerapan metode HTB kelas *parent* hanya membutuhkan *max limit* dan *limit at* saja dan target *download* diberikan *bandwidth* sebesar 20 Mbps serta target *upload* diberikan *bandwidth* sebesar 10 Mbps.



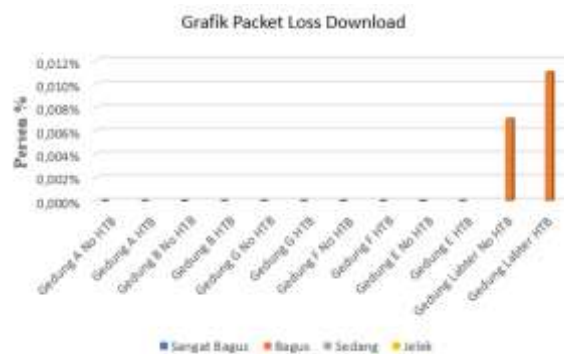
Gambar 14. Konfigurasi Child pada Router

Pada Gambar 14 menunjukkan konfigurasi pada perangkat *router* mikrotik yang dimiliki oleh salah satu gedung untuk bagian kelas *child* dengan memasukkan target IP *Network* pada fitur *general* yang ditujukan ke *client* dari perangkat *router* tersebut yang akan dijadikan *child* dari *parent* yang sebelumnya telah dikonfigurasi dan menentukan *max limit* dan *limit at* yang tidak melebihi dari kelas *parent*. Pada kasus ini kelas *child* dibuat sebanyak 2 *client* dalam penerapan metode HTB masing-masing *client* diberikan target *download* sebesar 10 Mbps dan target *upload* sebesar 5 Mbps dimana total keseluruhan target *download* dan *upload* dari kedua *client* tersebut tidak melebihi dari jumlah *bandwidth* yang sudah dibuat pada kelas *parent* dan kedua *client* pada kelas *child* tersebut memiliki sekala *priority* yang sama yaitu sebesar delapan. Sehingga *bandwidth* yang didapatkan jika ditotalkan yakni sebesar 20 Mbps untuk target *download* dan 10 Mbps untuk target *upload*.



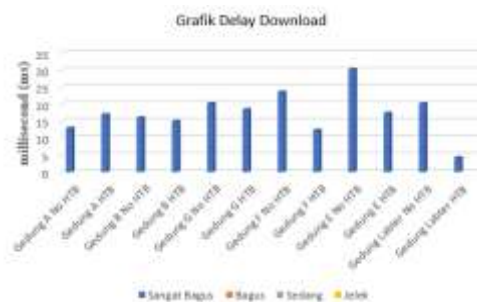
Gambar 15. Grafik Throughput

Pada Gambar 15 menunjukkan grafik hasil dari perhitungan parameter QoS *Throughput* yang didapatkan tanpa menggunakan metode HTB dan dengan menggunakan metode HTB pada setiap *router*. Hasil perhitungan yang didapatkan oleh masing-masing *router* pun berbeda, untuk *router* gedung A tanpa metode HTB mendapatkan nilai 562 Kbps dan 422 Kbps dengan menggunakan metode HTB, untuk *router* gedung B tanpa metode HTB mendapatkan nilai 445 Kbps dan 488 Kbps dengan menggunakan metode HTB, untuk *router* gedung G tanpa metode HTB mendapatkan nilai 345 Kbps dan 384 Kbps dengan menggunakan metode HTB, untuk *router* gedung F tanpa metode HTB mendapatkan nilai 301 Kbps dan 602 Kbps dengan menggunakan metode HTB, untuk *router* gedung E tanpa metode HTB mendapatkan nilai 216 Kbps dan 393 Kbps dengan menggunakan metode HTB, untuk *router* gedung Lab Terpadu tanpa metode HTB mendapatkan nilai 350 Kbps dan 104 Kbps dengan menggunakan metode HTB.



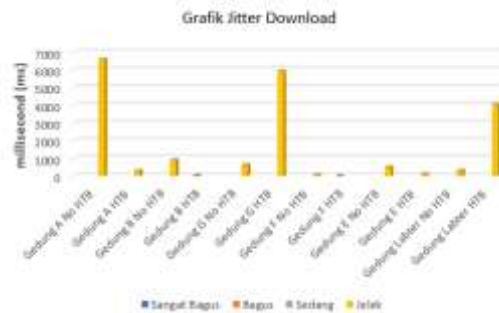
Gambar 16. Grafik Packet Loss

Pada Gambar 16 menunjukkan hasil dari perhitungan parameter QoS *Packet Loss* tanpa menggunakan metode HTB dan dengan menggunakan metode HTB pada setiap *router*. Untuk keseluruhan nilai yang didapatkan pada pengukuran parameter QoS oleh masing-masing *router* yakni mendapatkan nilai rata-ratanya di bawah 3%.



Gambar 17. Grafik Delay

Pada Gambar 17 menunjukkan grafik hasil dari perhitungan parameter QoS *Delay* untuk data *download* dengan menggunakan metode HTB dan tanpa menggunakan metode HTB pada setiap *router*. Hasil perhitungan yang didapatkan oleh masing-masing *router* pun berbeda, untuk *router* gedung A tanpa metode HTB mendapatkan nilai 12,93 ms dan 16,97 ms dengan menggunakan metode HTB, untuk *router* gedung B tanpa metode HTB mendapatkan nilai 16,01 ms dan 14,97 ms dengan menggunakan metode HTB, untuk *router* gedung G tanpa metode HTB mendapatkan nilai 20,22 ms dan 18,41 ms dengan menggunakan metode HTB, untuk *router* gedung F tanpa metode HTB mendapatkan nilai 23,60 ms dan 12,33 ms dengan menggunakan metode HTB, untuk *router* gedung E tanpa metode HTB mendapatkan nilai 30,28 ms dan 17,36 ms dengan menggunakan metode HTB, untuk *router* gedung Lab Terpadu tanpa metode HTB mendapatkan nilai 20,17 ms dan 4,28 ms dengan menggunakan metode HTB.



Gambar 18. Grafik Jitter

Pada Gambar 18 menunjukkan grafik hasil dari perhitungan parameter QoS *Jitter* untuk data *download* dengan menggunakan metode HTB dan tanpa menggunakan metode HTB pada setiap *router*. Hasil perhitungan yang didapatkan oleh masing-masing *router* pun berbeda, untuk *router* gedung A tanpa metode HTB mendapatkan nilai 6.604 ms dan 309 ms dengan menggunakan metode HTB, untuk *router* gedung B tanpa metode HTB mendapatkan nilai 910 ms dan 28,99 ms dengan menggunakan metode HTB, untuk *router* gedung G tanpa metode HTB mendapatkan nilai 650 ms dan 5.936 ms dengan menggunakan metode HTB, untuk *router* gedung F tanpa metode HTB mendapatkan nilai 86,48 ms dan 12,57 ms dengan menggunakan metode HTB, untuk *router* gedung E tanpa metode HTB mendapatkan nilai 525 ms dan 130 ms dengan menggunakan metode HTB, untuk *router* gedung Lab Terpadu tanpa metode HTB mendapatkan nilai 320 ms dan 4.070 ms dengan menggunakan metode HTB.

#### 4.7 Pembahasan

Pada *Tabel 3* dan *Tabel 4* merupakan hasil keseluruhan pengukuran yang dilakukan. Terlihat Konfigurasi jaringan yang ada di ITK dapat di optimalkan terutama pada bagian *delay*. *Delay* yang terjadi bisa dikurangi hingga 31,56 % persen. Hal ini menjadi konfirmasi bahwa penurunan *delay* dapat terjadi dengan diterapkannya HTB. Sesuai dengan penelitian yang dilakukan Ahdan dkk [16] yang mana pada kasus penelitiannya terjadi penurunan signifikan pada *delay* sejumlah 72,9%.

Tabel 3. Quality of Services Tanpa HTB

Parameter	Gedung A	Gedung B	Gedung G	Gedung F	Gedung E	Labter
<i>Throughput</i>	562 Kbps	445 Kbps	345 Kbps	301 Kbps	216 Kbps	350 Kbps
<i>Packet Loss</i>	0 %	0 %	0 %	0 %	0 %	0,007 %
<i>Delay</i>	12,93 ms	16,01 ms	20,22 ms	23,60 ms	30,28 ms	20,17 ms
<i>Jitter</i>	6.604 ms	910 ms	650 ms	86,48 ms	525 ms	320 ms

Tabel 4. Quality of Services dengan HTB

Parameter	Gedung A	Gedung B	Gedung G	Gedung F	Gedung E	Labter
<i>Throughput</i>	422 Kbps	488 Kbps	384 Kbps	602 Kbps	393 Kbps	104 Kbps
<i>Packet Loss</i>	0 %	0 %	0 %	0 %	0 %	0,011 %
<i>Delay</i>	16,97 ms	14,97 ms	18,41 ms	12,33 ms	17,36 ms	4,28 ms
<i>Jitter</i>	309 ms	28,99 ms	5.936 ms	12,57 ms	130 ms	4.070 ms

#### 5. Simpulan

Berdasarkan penelitian yang telah dilakukan selama perancangan sampai analisa perbandingan QoS pada jaringan internet Institut Teknologi menggunakan metode *Hierarchical Token Bucket* maka dapat disimpulkan sebagai berikut. Nilai *Throughput* pada manajemen *bandwidth* dengan menggunakan metode antrean HTB didapatkan sebesar 399 Kbps dibandingkan tanpa menggunakan HTB didapatkan sebesar 370 Kbps. Selanjutnya Nilai *Packet*

Loss pada manajemen *bandwidth* dengan menggunakan metode antrean HTB didapatkan rata-rata di bawah dari 3%. Kemudian, Nilai *Delay* pada manajemen *bandwidth* dengan menggunakan metode antrean HTB didapatkan lebih kecil yakni 14 ms jauh lebih baik dibandingkan tanpa menggunakan HTB didapatkan sebesar 21 ms.

Berdasarkan hasil di atas dapat dilihat bahwa kualitas jaringan dengan menggunakan metode antrean HTB lebih optimal, hal ini dikarenakan *bandwidth* akan terbagi sesuai dengan *rule* yang diterapkan. Dengan menggunakan metode antrean HTB dapat dijadikan sebagai rekomendasi dan evaluasi kepada UPT TIK Institut Teknologi Kalimantan dalam meningkatkan kualitas layanan jaringan.

#### Daftar Referensi

- [1] A. Z. Fadilah dan R. R. Saedudin, "Analisis Simulasi Manajemen Bandwidth Menggunakan Metode Hierarchical Token Bucket (htb) Untuk Meningkatkan Quality Of Service (qos)," *e-Proceeding Eng.*, vol. 8, no. 5, hal. 9072–9078, 2021.
- [2] D. Kurnia, "Analisis Qos Pada Pembagian Bandwidth Dengan Metode Layer 7 Protocol, Pcq, Htb Dan Hotspot Di Smk Swasta Al-Washliyah Pasar Senen," *CESS (Journal Comput. Eng. Syst. Sci.)*, vol. 2, no. 2, hal. 102–111, 2017.
- [3] A. F. Arman, E. Budiman, dan M. Taruk, "Implementasi Metode PCQ pada QoS Jaringan Komputer Fakultas Farmasi Universitas Mulawarman," *J. Rekayasa Teknol. Inf.*, vol. 4, no. 2, hal. 100–107, Des 2020, doi: 10.30872/jurti.v4i2.5111.
- [4] H. Fahmi, "Analisis Qos (Quality of Service) Pengukuran Delay, Jitter, Packet Lost Dan Throughput Untuk Mendapatkan Kualitas Kerja Radio Streaming Yang Baik," *J. Teknol. Inf. dan Komun.*, vol. 7, no. 2, hal. 98–105, 2018.
- [5] A. Nurdianto dan Deli, "Studi Komparsi Managemen Bandwidth Antara Metode Hirarchical Token Bucket (Htb) Dan Peer Connection Queue (Pcq)," *Conf. Business, Soc. Sci. Innov. Technol.*, vol. 1, no. 1, hal. 487–497, 2020.
- [6] A. Restu Mukti dan R. Novrianda Dasmen, "Prototipe Manajemen Bandwidth pada Jaringan Internet Hotel Harvani dengan Mikrotik RB 750r2," *J. Inform. J. Pengemb. IT*, vol. 4, no. 2, hal. 87–92, 2019, doi: 10.30591/jpit.v4i2.1322.
- [7] Y. B. Pello dan R. Efendi, "Analisis Quality of Service Menggunakan Metode Hierarchical Token Bucket (Studi Kasus: Fti Uksw) Quality of Service Analysis Using the Hierarchical Token Bucket Method (Case Study: Swcu Fti)," *J. Inform. dan Komputer) Akreditasi KEMENRISTEKDIKTI*, vol. 4, no. 3, hal. 193–198, 2021, doi: 10.33387/jiko.
- [8] I. N. Bernadus, N. Gunantara, dan K. O. Saputra, "Analisis Kinerja Jaringan Internet dengan Metode Class Based Queueing di Universitas Dhyana Pura," *Maj. Ilm. Teknol. Elektro*, vol. 18, no. 1, hal. 133–140, Mei 2019, doi: 10.24843/MITE.2019.v18i01.P20.
- [9] M. Fahad dan W. Sulisty, "Implementasi Metode Hierarchical Token Bucket dan Penjadwalan Otomatis dalam Pembagian Bandwidth di Sanggar Kegiatan Belajar," *J. Ilm. Komput.*, vol. 19, no. 2, hal. 599–612, 2023.
- [10] L. Joko Susanto *et al.*, "Pengembangan Perancangan Jaringan Local Area Network (LAN) Di RSIA Anugrah Medical Centre Metro," *MIKROTIK J. Manaj. Inform.*, vol. 7, no. 2, hal. 25–36, 2017.
- [11] O. L. Daulay, "Analisis Quality of Services(QoS) Pada Manajemen Bandwidth Menggunakan Metode Hirarchical Token Bucket (HTB) Pada Sistem Jaringan," *JISTech (Journal Islam. Sci. Technol. JISTech)*, vol. 5, no. 2, hal. 18–35, 2020.
- [12] J. Konsera, T. Hartati, dan Y. A. Wijaya, "Analisa Perbandingan Manajemen Bandwidth Simple Queue Dan Queue Tree Pada Aplikasi Video Conference Zoom," *J. Ilmia NERO*, vol. 7, no. 2, hal. 85–94, 2022.
- [13] R. Hanipah dan H. Dhika, "Analisa Pencegahan Aktivitas Ilegal Didalam Jaringan Dengan Wireshark," *DoubleClick J. Comput. Inf. Technol.*, vol. 4, no. 1, hal. 11–23, Agu 2020, doi: 10.25273/doubleclick.v4i1.5668.
- [14] M. Hidayat, E. Budiman, dan Rudiman, "Implementasi Manajemen Bandwidth pada Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Mulawarman Samarinda Menggunakan Metode Hierarchical Token Bucket (HTB)," *Pros. Semin. Ilmu Komput. dan Teknol. Inf.*, vol. 2, no. 1, hal. 99–109, 2017.
- [15] M. Iqbal Ichwan, L. Sugiyanta, dan P. Wibowo Yunanto, "Analisis Manajemen Bandwidth Hierarchical Token Bucket (HTB) dengan Mikrotik pada Jaringan SMK Negeri 22," *PINTER J. Pendidik. Tek. Inform. dan Komput.*, vol. 3, no. 2, hal. 122–126, 2019, doi:

- 10.21009/pinter.3.2.6.
- [16] S. Ahdan, O. Firmanto, dan S. Ramadona, "Rancang Bangun dan Analisis QoS (Quality of Service) Menggunakan Metode HTB (Hierarchical Token Bucket) pada RT/RW Net Perumahan Prasanti 2," *J. Teknoinfo*, vol. 12, no. 2, hal. 49–54, Jul 2018, doi: 10.33365/jti.v12i2.89.