

# Implementasi Manajemen Pada Mikrotik Menggunakan Metode Hierarchical Token Bucket Dan Queue Tree (Studi Kasus: PT. XYZ)

Mohammad Arief Shofa

Teknik Informatika, Universitas Mercu Buana, Jakarta  
PT. Indojoya Sukses Makmur, Jakarta  
\*ariefshofaa@gmail.com

**Abstrak**— Penelitian ini dilatarbelakangi oleh permasalahan penurunan kinerja jaringan pada head office PT. PT. XYZ yang menggunakan teknologi Active Ethernet, dimana banyak pengguna mengalami lambatnya akses internet akibat tingginya penggunaan bandwidth dan jumlah pengguna yang besar. Permasalahan utama terletak pada belum optimalnya manajemen bandwidth sehingga diperlukan solusi untuk meningkatkan kualitas layanan jaringan. Penelitian ini bertujuan untuk mengimplementasikan Quality of Service (QoS) menggunakan metode Hierarchical Token Bucket (HTB) dan Queue Tree pada perangkat MikroTik sebagai pendekatan dalam mengelola dan mendistribusikan bandwidth secara lebih efisien dan adil. Metode penelitian yang digunakan adalah kombinasi pendekatan kualitatif dan kuantitatif, dengan pengumpulan data melalui observasi, wawancara, serta pengukuran parameter QoS seperti throughput, packet loss, delay, dan jitter menggunakan Wireshark. Hasil penelitian menunjukkan bahwa setelah implementasi HTB dan Queue Tree terjadi peningkatan signifikan pada throughput dari 68,869 kbps menjadi 6021 kbps, serta penurunan packet loss menjadi 0,9%. Selain itu, nilai delay dan jitter tetap stabil dalam kategori sangat baik sesuai standar TIPHON. Kesimpulan dari penelitian ini adalah bahwa penerapan metode HTB dan Queue Tree efektif dalam meningkatkan kinerja jaringan, khususnya dalam optimalisasi penggunaan bandwidth dan peningkatan kualitas layanan. Dengan demikian, metode ini dapat menjadi solusi yang tepat untuk mengatasi permasalahan jaringan pada lingkungan dengan trafik tinggi.

## Article History:

Received: Feb 24, 2024

Revised: Sept 5, 2024

Accepted: Dec 5, 2024

Published: Jan 31, 2025

**Kata Kunci**— Hierarchical Token Bucket; Mikrotik; Management Bandwidth; Quality of Services (QoS); Queue Tree

DOI: 10.22441/jitkom.v9i1.008

## I. PENDAHULUAN

Perkembangan penggunaan internet di lingkungan perusahaan menuntut ketersediaan jaringan yang stabil, cepat, dan mampu melayani banyak pengguna secara bersamaan. Pada head office PT. PT. XYZ, jaringan berbasis Active Ethernet digunakan untuk menghubungkan setiap titik ke pusat data melalui perangkat aktif seperti switch dan router. Namun, tingginya intensitas akses internet dan penggunaan bandwidth yang tidak terkontrol menyebabkan keluhan berupa lambatnya akses web, terutama pada jam kerja sibuk. Kondisi ini sejalan dengan berbagai penelitian yang menunjukkan bahwa pertambahan pengguna dan trafik yang tidak terkelola dapat menurunkan performa jaringan secara signifikan [1], [2], [14]. Oleh karena itu, diperlukan suatu mekanisme pengelolaan bandwidth yang mampu membagi kapasitas jaringan secara adil, terukur, dan sesuai prioritas layanan.

Quality of Service (QoS) merupakan salah satu pendekatan yang banyak digunakan untuk menjaga kualitas layanan jaringan melalui pengaturan bandwidth, delay, jitter, dan packet loss [16], [17]. Dalam evaluasinya, parameter QoS umumnya mengacu

pada standar TIPHON sebagai acuan klasifikasi kualitas jaringan [18]. Penerapan QoS menjadi penting karena kualitas jaringan tidak hanya ditentukan oleh besarnya bandwidth, tetapi juga oleh kemampuan sistem dalam mengelola trafik agar layanan penting tetap berjalan optimal. Hal ini semakin relevan pada infrastruktur yang melayani aktivitas perkantoran, akses web, pertukaran data, dan komunikasi daring secara simultan.

Berbagai penelitian sebelumnya telah membahas implementasi manajemen bandwidth menggunakan beragam metode pada perangkat MikroTik, seperti Queue Tree, Simple Queue, PCQ, HTB, hingga kombinasi beberapa metode tersebut [3], [5], [6], [7], [8], [10], [12], [13], [15]. Hasil penelitian terdahulu menunjukkan bahwa setiap metode memiliki kelebihan dan keterbatasan masing-masing, baik dari sisi kemudahan konfigurasi, fleksibilitas pembagian bandwidth, maupun kemampuan menjaga kestabilan trafik. Penelitian perbandingan juga memperlihatkan bahwa metode yang lebih adaptif dan hierarkis cenderung lebih efektif untuk lingkungan jaringan dengan trafik yang beragam [1], [7], [10], [15].

Selain itu, sejumlah studi lain menegaskan pentingnya optimalisasi jaringan melalui pendekatan pengelolaan akses, kontrol trafik, dan peningkatan efisiensi layanan jaringan di berbagai institusi dan perusahaan [4], [9], [11]. Dalam konteks tersebut, metode Hierarchical Token Bucket (HTB) dinilai mampu memberikan kontrol bandwidth secara hierarkis, sedangkan Queue Tree memungkinkan pengelompokan dan prioritas trafik secara lebih rinci [19], [20]. Berdasarkan uraian tersebut, penelitian ini memfokuskan pada implementasi manajemen bandwidth menggunakan HTB dan Queue Tree pada Mikrotik untuk meningkatkan kinerja jaringan di PT. PT. XYZ, sehingga diharapkan dapat memberikan distribusi bandwidth yang lebih optimal, meningkatkan kualitas layanan, serta mendukung produktivitas pengguna jaringan.

## II. LITERATURE REVIEW

Penelitian mengenai manajemen bandwidth dan Quality of Service (QoS) pada jaringan komputer menunjukkan bahwa pengaturan trafik yang tepat berpengaruh langsung terhadap kestabilan dan kualitas layanan jaringan. Damopolii et al. [1] membandingkan metode Queue Tree dan Simple Queue pada Mikrotik dan menunjukkan bahwa Queue Tree lebih optimal dalam memanfaatkan bandwidth yang tersedia secara dinamis. Temuan serupa juga diperkuat oleh Prihantoro et al. [3] dan Sari et al. [10], yang menyatakan bahwa Queue Tree mampu memberikan pengelolaan trafik yang lebih fleksibel dibandingkan metode sederhana.

Di sisi lain, pendekatan Hierarchical Token Bucket (HTB) dinilai efektif dalam membagi bandwidth secara hierarkis sesuai prioritas layanan. Toresa [7], Qalbi dan Riadi [19], serta Pello dan Efendi [20] menegaskan bahwa HTB mampu meningkatkan efisiensi alokasi bandwidth dan menjaga kualitas trafik penting. Kombinasi HTB dan Queue Tree bahkan dinilai lebih unggul dalam menangani trafik kompleks dibandingkan metode tunggal [15].

Kajian QoS sendiri umumnya diukur melalui parameter throughput, packet loss, delay, dan jitter sebagaimana dijelaskan oleh Apriza et al. [16] dan mengacu pada standar TIPHON [18]. Sejumlah penelitian pada lingkungan laboratorium, kampus, dan jaringan institusi juga menunjukkan bahwa penerapan QoS berbasis Mikrotik dapat meningkatkan performa jaringan secara signifikan [2], [5]. Dengan demikian, HTB dan Queue Tree menjadi pendekatan yang relevan untuk optimalisasi manajemen bandwidth pada jaringan perusahaan.

## III. METODOLOGI PENELITIAN

Penelitian ini menggunakan pendekatan mixed methods, yaitu gabungan metode kualitatif dan kuantitatif, untuk menganalisis dan mengevaluasi implementasi manajemen bandwidth pada jaringan Mikrotik menggunakan metode Hierarchical Token Bucket (HTB) dan Queue Tree di PT. PT. XYZ. Pendekatan ini dipilih karena penelitian tidak hanya berfokus pada hasil pengukuran teknis jaringan, tetapi juga mempertimbangkan kondisi operasional, kebutuhan pengguna, serta pandangan administrator jaringan terhadap kualitas layanan yang berjalan. Dengan demikian, penelitian dapat memberikan gambaran yang lebih menyeluruh mengenai

efektivitas penerapan Quality of Service (QoS) pada lingkungan jaringan perusahaan.

Objek penelitian ini adalah jaringan komputer pada head office PT. PT. XYZ yang menggunakan perangkat Mikrotik sebagai pengelola lalu lintas jaringan. Permasalahan utama yang diamati adalah lambatnya akses internet akibat tingginya jumlah pengguna dan penggunaan bandwidth yang belum terdistribusi secara optimal. Oleh karena itu, penelitian ini difokuskan pada penerapan metode HTB dan Queue Tree untuk mengatur alokasi bandwidth berdasarkan prioritas trafik agar kualitas layanan jaringan menjadi lebih baik dan lebih stabil.

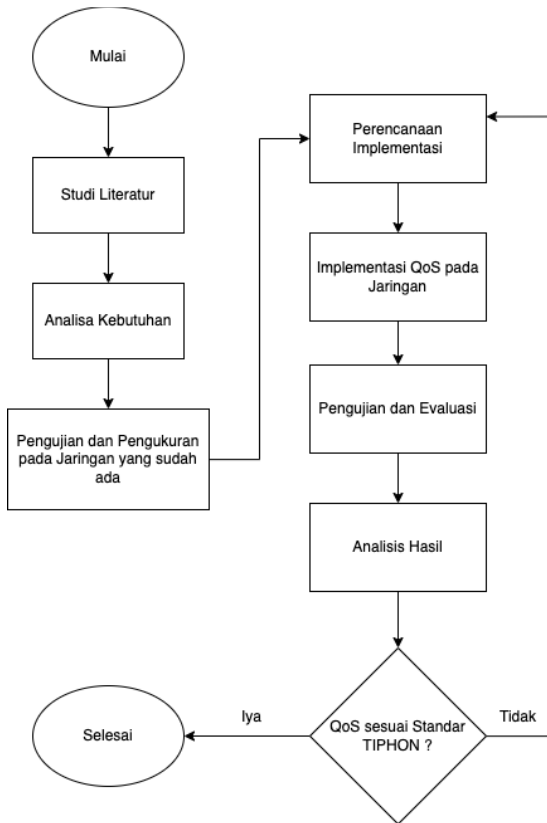
### A. Tahapan Penelitian

Pada tahap pengumpulan data, penelitian dilakukan melalui beberapa teknik. Pertama, observasi langsung terhadap kondisi infrastruktur jaringan yang meliputi modem, router, switch, kabel, serta koneksi client. Observasi ini dilakukan untuk mengetahui kondisi awal jaringan dan mengidentifikasi titik-titik yang berpotensi menimbulkan gangguan. Kedua, wawancara dilakukan kepada staf dan administrator jaringan guna memperoleh informasi mengenai keluhan pengguna, pola penggunaan internet, serta kebutuhan jaringan di lingkungan perusahaan. Ketiga, dilakukan dokumentasi terhadap topologi jaringan, konfigurasi perangkat, dan hasil pengujian sebelum maupun sesudah implementasi. Selain itu, data kuantitatif dikumpulkan menggunakan aplikasi Wireshark untuk menangkap paket data dan mengukur performa jaringan.

Tahapan penelitian diawali dengan analisis kondisi jaringan awal. Pada tahap ini dilakukan identifikasi masalah berdasarkan model lapisan jaringan, mulai dari pemeriksaan media fisik, perangkat jaringan, hingga analisis trafik data. Selanjutnya dilakukan pengukuran parameter Quality of Service yang meliputi throughput, packet loss, delay, dan jitter. Hasil pengukuran awal ini dijadikan sebagai data pembanding untuk mengetahui kondisi jaringan sebelum penerapan manajemen bandwidth. Setelah itu, peneliti merancang konfigurasi QoS pada Mikrotik menggunakan metode HTB untuk membentuk struktur bandwidth secara hierarkis, serta Queue Tree untuk mengelola antrian trafik berdasarkan jenis layanan, alamat IP, dan prioritas penggunaan.

Tahap berikutnya adalah implementasi konfigurasi pada perangkat Mikrotik. Dalam tahap ini dibuat parent queue dan child queue sesuai kebutuhan trafik jaringan perusahaan. Setelah konfigurasi selesai, dilakukan pengujian kembali pada periode operasional sibuk agar diperoleh data yang merepresentasikan kondisi nyata penggunaan jaringan. Data hasil implementasi kemudian dianalisis dengan membandingkan nilai QoS sebelum dan sesudah penerapan HTB dan Queue Tree. Evaluasi dilakukan berdasarkan standar TIPHON untuk menentukan kategori kualitas jaringan pada masing-masing parameter.

Hasil analisis kemudian digunakan untuk menarik kesimpulan mengenai efektivitas metode HTB dan Queue Tree dalam meningkatkan kualitas layanan jaringan. Dengan metode penelitian ini, diharapkan diperoleh gambaran yang jelas mengenai kemampuan konfigurasi QoS dalam mengoptimalkan bandwidth, menekan packet loss, menjaga stabilitas delay dan jitter, serta meningkatkan kinerja jaringan secara keseluruhan pada PT. PT. XYZ.



Gambar 1. Flowchart Penelitian

**B. Quality of Service (QoS)**

Quality of Service (QoS) merupakan kemampuan suatu jaringan dalam menyediakan layanan yang baik melalui pengalokasian bandwidth yang memadai serta pengendalian parameter kinerja jaringan seperti jitter dan delay. QoS digunakan untuk menilai atribut-atribut kinerja layanan jaringan dan menunjukkan kapabilitas jaringan dalam memberikan layanan yang lebih optimal pada trafik tertentu. Dalam pengukurannya, QoS umumnya dievaluasi menggunakan standar TIPHON yang mencakup parameter throughput, packet loss, delay, dan jitter [16]–[18].

**Throughput**

Throughput adalah laju transfer data efektif pada jaringan yang diukur dalam satuan bps, yang menunjukkan jumlah total paket data yang berhasil diterima di tujuan dalam selang waktu tertentu. Parameter ini digunakan untuk menilai seberapa besar kemampuan jaringan dalam menyalurkan data secara aktual. [16]

Tabel 1. Kategori *Throughput*

| Kategori     | <i>Throughput (bps)</i> | Indeks |
|--------------|-------------------------|--------|
| Sangat Bagus | 100                     | 4      |
| Bagus        | 75                      | 3      |
| Sedang       | 50                      | 2      |
| Buruk        | < 25                    | 1      |

$$Throughput = \frac{\text{Paket data diterima}}{\text{Lama pengamatan}} \quad (1)$$

$$\%Throughput = \frac{\text{Paket data diterima}}{\text{Alokasi bandwidth user}} \times 100\% \quad (2)$$

**Packet Loss**

Packet loss merupakan persentase paket data yang hilang selama proses pengiriman dari source menuju destination. Nilai packet loss digunakan sebagai indikator kualitas jaringan, karena semakin besar paket yang hilang, maka semakin menurun kualitas layanan jaringan tersebut [16].

Tabel 2. Kategori Packet Loss

| Kategori     | <i>Packet Loss (%)</i> | Indeks |
|--------------|------------------------|--------|
| Sangat Bagus | 0-2                    | 4      |
| Bagus        | 3-14                   | 3      |
| Sedang       | 15-24                  | 2      |
| Buruk        | >25                    | 1      |

$$Packet Loss = \frac{\text{Paket data yang hilang}}{\text{Paket data dikirim}} \times 100\% \quad (3)$$

**Delay**

Delay merupakan total waktu tunda yang dialami paket data selama proses transmisi dari sumber ke tujuan. Nilai delay digunakan sebagai indikator kualitas jaringan, karena semakin besar delay maka semakin rendah kualitas layanan yang diterima pengguna. [16], [18]

Tabel 3. Kategori Delay [18]

| Kategori     | <i>Delay</i>      | Indeks |
|--------------|-------------------|--------|
| Sangat Bagus | < 150 ms          | 4      |
| Bagus        | 150 ms s/d 300 ms | 3      |
| Sedang       | 300 ms s/d 450 ms | 2      |
| Buruk        | > 450 ms          | 1      |

$$\text{Rata-rata Delay} = \frac{\text{Total Delay}}{(\text{Paket data diterima}-1)} \quad (4)$$

**Jitter**

Jitter merupakan variasi waktu tunda (delay) antar paket pada jaringan IP. Nilai jitter dipengaruhi oleh fluktuasi beban trafik dan tingkat kongesti jaringan. Semakin besar nilai jitter, maka semakin menurun kualitas layanan jaringan, sehingga jitter perlu dijaga seminimum mungkin untuk memperoleh performa jaringan yang optimal [18], [19].

Tabel 4. Kategori Jitter

| Kategori     | <i>Delay</i>      | Indeks |
|--------------|-------------------|--------|
| Sangat Bagus | 0 ms              | 4      |
| Bagus        | 1 ms s/d 75 ms    | 3      |
| Sedang       | 76 ms s/d 125 ms  | 2      |
| Buruk        | 126 ms s/d 225 ms | 1      |

$$\text{Rata-rata Jitter} = \frac{\text{Total Delay}}{\text{Total paket data}-1}$$

$$\text{Rata-rata Jitter} = \frac{\text{Delay}-\text{rata-rata Delay}}{\text{Total paket data}-1} \quad (5)$$

IV. HASIL DAN ANALISA

A. Penelitian Pendahuluan

Data dari hasil perhitungan dari penelitian pendahuluan dari parameter *throughput*, *packet loss*, *delay* dan *jitter* menggunakan *software wireshark* ditunjukkan pada tabel 5.

Tabel 5. Penelitian Pendahuluan

| No | Parameter   | Hasil perhitungan   | Standar TIPHON |
|----|-------------|---------------------|----------------|
| 1  | Throughput  | 68,869 kbps / 1,38% | Buruk          |
|    | Packet Loss | 2,59%               | Sangat bagus   |
|    | Delay       | 34 ms               | Sangat bagus   |
|    | Jitter      | 34 ms               | Bagus          |
| 2  | Throughput  | 3611 kbps / 72%     | Sedang         |
|    | Packet Loss | 6,17%               | Bagus          |
|    | Delay       | 16 ms               | Sangat bagus   |
|    | Jitter      | 16 ms               | Bagus          |

Tabel 5 menampilkan hasil perhitungan untuk berbagai parameter yang diukur dalam jaringan. Terdapat dua set data yang dianalisis dengan standar TIPHON. Set pertama menunjukkan *throughput* sebesar 68,869 kbps dengan *packet loss* sebesar 2,59%, *delay* 34 ms, dan *jitter* 34 ms. Parameter ini secara umum diklasifikasikan sebagai "Buruk" untuk *throughput* dan "Sangat Bagus" untuk *packet loss*, *delay*, dan *jitter*. Sementara pada set kedua, *throughput* mencapai 3611 kbps dengan *packet loss* sebesar 6,17%, *delay* 16 ms, dan *jitter* 16 ms. Hasil ini dikategorikan sebagai "Sedang" untuk *throughput*, "Bagus" untuk *packet loss* dan *jitter*, serta "Sangat Bagus" untuk *delay*. Data ini memberikan gambaran mengenai performa jaringan yang diukur berdasarkan standar TIPHON.

B. Analisis Permasalahan Pada Jaringan

Layer 1 – Physical Layer

Peneliti melakukan evaluasi dan pengujian terhadap kualitas kabel yang digunakan dalam jaringan, seperti kabel UTP Cat6. Langkah pertama dalam pengujian kualitas kabel adalah menggunakan LAN tester. Alat ini digunakan untuk memeriksa beberapa aspek kritis seperti keutuhan kabel, tingkat isolasi, dan ketahanan terhadap gangguan fisik.

Penggunaan LAN tester memungkinkan peneliti untuk dengan cepat mengidentifikasi potensi masalah, termasuk adanya putus kabel atau kesalahan koneksi. Hasil pengujian dari LAN tester memberikan gambaran awal tentang keandalan kabel dan membantu peneliti dalam menentukan langkah-langkah selanjutnya untuk pemeliharaan atau perbaikan. Dengan demikian, penggunaan alat ini menjadi langkah kritis dalam menjaga dan meningkatkan kualitas fisik kabel pada jaringan.

Semua indikator menyala menunjukkan bahwa kabel jaringan terpasang dengan baik di kedua ujungnya.



(a)



(b)



(c)

Gambar 2. (a) Hasil Test Modem ke Router, (b) Hasil Test Router ke Switch, (c) Hasil Test dari Switch ke PC

Layer 2 – Data Link Layer

Analisis Penggunaan Switch Unmanaged (Linksys) pada Layer 2:

Pada infrastruktur jaringan yang ada, terdapat penggunaan switch Linksys yang termasuk dalam kategori unmanaged. Switch ini, seperti yang telah disebutkan, tidak memiliki kemampuan untuk dikonfigurasi secara lebih mendalam, yang tentu saja membawa konsekuensi dalam pengelolaan layer 2 jaringan.

Layer 3 – Network Layer

Analisis menggunakan perintah ping dan traceroute

Perintah PING

Perintah ini digunakan untuk menguji konektivitas antara perangkat yang melakukan ping dan tujuan yang ditentukan. Hasil ping menunjukkan apakah koneksi berhasil dan berapa waktu yang dibutuhkan untuk melakukan perjalanan pulang-pergi.

```
[arief@Indo Jaya Sukses Makmur] > ping 192.168.1.1
SEQ HOST                SIZE TTL TIME STATUS
0 192.168.1.1          56 64 0ms  S
1 192.168.1.1          56 64 0ms  S
2 192.168.1.1          56 64 0ms  S
3 192.168.1.1          56 64 0ms  S
4 192.168.1.1          56 64 0ms  S
5 192.168.1.1          56 64 0ms  S
6 192.168.1.1          56 64 0ms  S
7 192.168.1.1          56 64 0ms  S
8 192.168.1.1          56 64 0ms  S
9 192.168.1.1          56 64 0ms  S
10 192.168.1.1         56 64 0ms  S
11 192.168.1.1         56 64 0ms  S
12 192.168.1.1         56 64 0ms  S
13 192.168.1.1         56 64 0ms  S
14 192.168.1.1         56 64 0ms  S
```

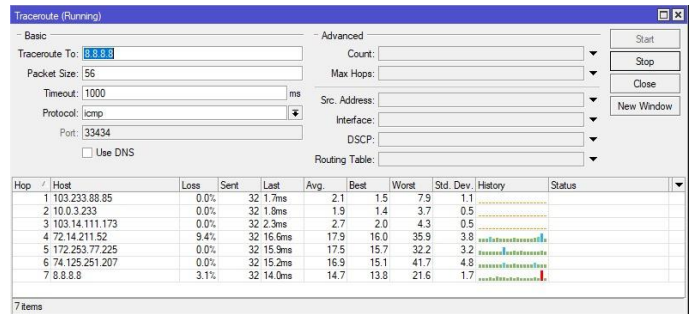
Gambar 3. Hasil Test Ping

Berdasarkan hasil ping yang diberikan ke alamat IP 192.168.1.1, kesimpulan yang dapat diambil adalah:

- Status Koneksi: Semua paket ping menuju 192.168.1.1 mendapatkan respons dengan waktu yang sangat rendah (0ms atau 1ms). Hal ini menunjukkan bahwa host 192.168.1.1 secara aktif merespons permintaan ping.
- Ketidakstabilan Waktu Respons: Waktu respons terlihat sangat konsisten, dengan mayoritas ping mendapatkan respons dalam 0ms. Hanya pada ping ke-5 terdapat peningkatan ke 1ms. Ini menunjukkan konsistensi yang tinggi dalam waktu respons, yang biasanya diinginkan untuk koneksi yang stabil.
- TTL (Time to Live): TTL yang tertera adalah 64, yang merupakan nilai TTL default pada sebagian besar sistem. Hal ini menunjukkan bahwa paket ping telah melewati beberapa hop di jaringan, tetapi tidak memberikan informasi tentang jumlah hop atau topologi jaringan.

Perintah Traceroute

Traceroute digunakan untuk melacak rute yang diambil paket data dari perangkat pengirim ke tujuan. Ini membantu mengidentifikasi setiap hop atau perangkat jaringan yang dilalui oleh paket dalam perjalanan ke tujuan.



Gambar 4. Hasil Traceroute

Berdasarkan hasil trace route yang diberikan, kita dapat menyimpulkan beberapa hal:

- Hop Pertama (103.233.88.85): Waktu respons rendah (6.1ms) menunjukkan koneksi lokal ke router pertama.
- Hop Kedua (10.0.3.233): Respons yang cepat (2.2ms) menunjukkan konektivitas yang baik ke router kedua.
- Hop Ketiga (103.14.111.173): Respons tetap rendah (2.5ms), menunjukkan koneksi yang baik ke router ketiga.
- Hop Keempat (72.14.211.52): Terdapat kehilangan paket sebesar 23%, dan waktu respons meningkat (17.4ms). Hal ini mungkin menunjukkan masalah di jalur ini, seperti kelebihan beban atau gangguan.
- Hop Kelima (172.253.77.225): Respons kembali normal (15.5ms) setelah hop sebelumnya. Koneksi tampak stabil di sini.
- Hop Keenam (74.125.251.207): Respons tetap rendah (15.3ms), menunjukkan koneksi yang baik ke server berikutnya.
- Tujuan (8.8.8.8): Respons stabil (15ms) dan tidak ada kehilangan paket. Koneksi tampak baik ke tujuan akhir, yang dalam hal ini adalah server DNS Google (8.8.8.8).

Kesimpulannya, rute ke tujuan tampaknya stabil dengan sebagian kecil kehilangan paket di hop keempat. Mungkin ada sedikit gangguan pada jalur tersebut, namun secara keseluruhan, koneksi tampaknya baik dan stabil.

Secara keseluruhan, hasil ping menunjukkan koneksi yang stabil dan responsif antara host dan alamat IP 192.168.1.1. Tidak ada tanda-tanda masalah atau kegagalan koneksi dalam hasil ping ini.

Layer 4 – Transport Layer

Tabel 6. Analisis Transport Layer

| No.   | Time      | Source        | Destination   | Protocol | Length |
|-------|-----------|---------------|---------------|----------|--------|
| 646   | 15209146  | 192.168.6.23  | 17253118202   | HTTP     | 366    |
| 650   | 15231663  | 17253118202   | 192.168.6.23  | OCSP     | 348    |
| 719   | 16261560  | 192.168.6.23  | 17.253.61.206 | HTTP     | 366    |
| 723   | 16285826  | 17.253.61.206 | 192.168.6.23  | OCSP     | 348    |
| 44717 | 811997212 | 192.168.6.23  | 52.84.228.88  | HTTP     | 375    |
| 44725 | 812019281 | 52.84.228.88  | 192.168.6.23  | OCSP     | 1473   |
| 45405 | 812563928 | 192.168.6.23  | 74.125.130.94 | HTTP     | 377    |
| 45480 | 812583493 | 74.125.130.94 | 192.168.6.23  | OCSP     | 790    |
| 45606 | 812649204 | 192.168.6.23  | 74.125.68.94  | HTTP     | 377    |
| 45729 | 812675080 | 74.125.68.94  | 192.168.6.23  | OCSP     | 791    |
| 48452 | 814825899 | 192.168.6.23  | 74.125.68.94  | HTTP     | 377    |

## Analisis Transport Layer:

- Data mencakup komunikasi menggunakan protokol HTTP dan OCSP.
- Terdapat pertukaran permintaan (HTTP Request) dan respons (OCSP Response).
- Panjang data bervariasi, menunjukkan adanya variasi dalam jumlah data yang ditransmisikan.
- Waktu transmisi data bervariasi, menunjukkan keragaman dalam waktu respons.
- Sumber (Source) dan tujuan (Destination) data dapat berubah, menunjukkan komunikasi jaringan yang dinamis.

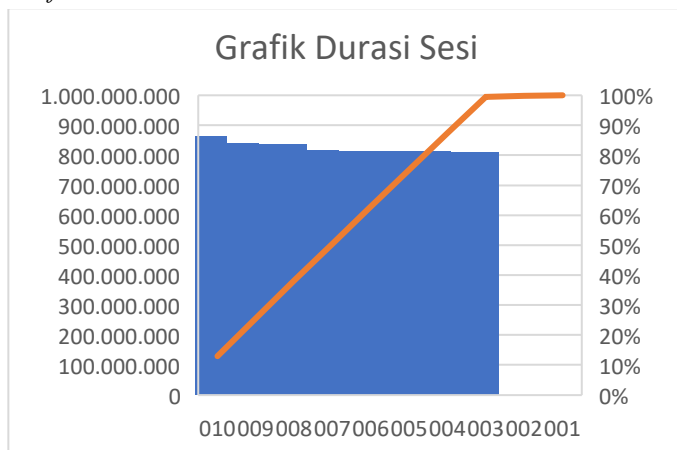
## Layer 5 – Session Layer

### Identifikasi Sesi

Tabel 7. Analisis Session Layer

| ID Sesi | Waktu Pembukaan | Waktu Penutupan | Durasi   |
|---------|-----------------|-----------------|----------|
| 001     | 15.209.146      | 15.231.663      | 0.022517 |
| 002     | 16.261.560      | 16.285.826      | 0.024266 |
| 003     | 811.997.212     | 812.019.281     | 0.024266 |
| 004     | 812.563.928     | 812.583.493     | 0.019565 |
| 005     | 812.649.204     | 812.675.080     | 0.025876 |
| 006     | 814.825.899     | 814.852.834     | 0.026935 |
| 007     | 818.030.888     | 818.273.818     | 0.24293  |
| 008     | 837.681.812     | 837.739.502     | 0.05769  |
| 009     | 841.988.321     | 842.225.415     | 0.237094 |
| 010     | 863.242.934     | 863.507.243     | 0.264309 |

### Grafik Durasi Sesi



Gambar 4. Grafik Durasi Sesi

### Stabilitas Koneksi

Tabel 7. Stabilitas Koneksi

| ID Sesi | Paket Hilang (%) | Latensi (ms) | Sinyal Error |
|---------|------------------|--------------|--------------|
| 001     | -                | -            | -            |
| 002     | -                | -            | -            |
| 003     | -                | -            | -            |
| 004     | -                | -            | -            |
| 005     | -                | -            | -            |

| ID Sesi | Paket Hilang (%) | Latensi (ms) | Sinyal Error |
|---------|------------------|--------------|--------------|
| 006     | -                | -            | -            |
| 007     | -                | -            | -            |
| 008     | -                | -            | -            |
| 009     | -                | -            | -            |
| 010     | -                | -            | -            |

### Ringkasan Respon Pembukaan dan Penutupan Sesi.

Sesi 001 - Sesi 010: Pembukaan dan penutupan sesi berlangsung tanpa masalah.

### Pencatatan Aktifitas Sesi

- Sesi 001: GET request ke alamat OCSP dan respons.
- Sesi 002: GET request ke alamat OCSP dan respons.
- Sesi 003: GET request ke alamat OCSP dan respons.
- Sesi 004: GET request ke alamat OCSP dan respons.
- Sesi 005: GET request ke alamat OCSP dan respons.
- Sesi 006: GET request ke alamat OCSP dan respons.
- Sesi 007: GET request ke alamat OCSP dan respons.
- Sesi 008: GET request ke alamat OCSP dan respons.
- Sesi 009: GET request ke alamat OCSP dan respons.
- Sesi 010: GET request ke alamat OCSP dan respons.

### Konflik atau Masalah Teridentifikasi

Tabel 8. Konflik dan Masalah

| ID Sesi | Jenis Masalah | Tindakan Perbaikan |
|---------|---------------|--------------------|
| 001-010 | Tidak Ada     | -                  |

### Analisis Sinyal Error atau Exception

Selama Sesi 001 - Sesi 010, tidak terdapat sinyal error atau masalah yang teridentifikasi.

### Catatan Keamanan Sesi

Selama Sesi 001 - Sesi 010, tidak terdapat aktivitas mencurigakan atau masalah keamanan yang teridentifikasi.

### Log Aktifitas

## Layer 6 – Presentation Layer

### Analisis Protokol HTTP

- Permintaan HTTP ke "17.253.118.202" pada waktu "15.209146" dengan metode GET dan panjang paket 366.
- Respons OCSP dari "17.253.118.202" ke "192.168.6.23" pada waktu "15.231663" dengan panjang paket 348.
- Permintaan HTTP ke "17.253.61.206" pada waktu "16.261560" dengan metode GET dan panjang paket 366.
- Respons OCSP dari "17.253.61.206" ke "192.168.6.23" pada waktu "16.285826" dengan panjang paket 348.
- Permintaan HTTP ke "52.84.228.88" pada waktu "811.997212" dengan metode GET dan panjang paket 375.
- Respons OCSP dari "52.84.228.88" ke "192.168.6.23" pada waktu "812.019281" dengan panjang paket 1473.

- Permintaan HTTP ke "74.125.130.94" pada waktu "812.563928" dengan metode GET dan panjang paket 377.
- Respons OCSP dari "74.125.130.94" ke "192.168.6.23" pada waktu "812.583493" dengan panjang paket 790.
- Permintaan HTTP ke "74.125.68.94" pada waktu "812.649204" dengan metode GET dan panjang paket 377.
- Respons OCSP dari "74.125.68.94" ke "192.168.6.23" pada waktu "812.675080" dengan panjang paket 791.
- Permintaan HTTP ke "74.125.68.94" pada waktu "814.825899" dengan metode GET dan panjang paket 377.
- Respons OCSP dari "74.125.68.94" ke "192.168.6.23" pada waktu "814.852834" dengan panjang paket 790.
- Permintaan HTTP ke "104.22.4.69" pada waktu "818.030888" dengan metode GET dan panjang paket 488.
- Respons dari "104.22.4.69" ke "192.168.6.23" pada waktu "818.273818" dengan status HTTP 200 OK dan panjang paket 335.
- Permintaan HTTP ke "192.124.249.22" pada waktu "837.681812" dengan metode GET dan panjang paket 348.
- Respons OCSP dari "192.124.249.22" ke "192.168.6.23" pada waktu "837.739502" dengan panjang paket 1256.
- Permintaan HTTP ke "192.124.249.36" pada waktu "841.988321" dengan metode GET dan panjang paket 346.
- Respons OCSP dari "192.124.249.36" ke "192.168.6.23" pada waktu "842.225415" dengan panjang paket 1257.
- Permintaan HTTP ke "74.125.68.94" pada waktu "863.242934" dengan metode GET dan panjang paket 377.
- Respons OCSP dari "74.125.68.94" ke "192.168.6.23" pada waktu "863.507243" dengan panjang paket 790.

*Pengecekan Respons OCSP*

Semua respons OCSP memiliki panjang paket yang sesuai dan tidak menunjukkan indikasi kesalahan.

*Detil Permintaan HTTP*

Semua permintaan HTTP menggunakan metode GET dan memiliki panjang paket yang bervariasi

*Status Koneksi HTTP*

Status HTTP 200 OK diberikan sebagai respons dari "104.22.4.69" pada waktu "818.273818".

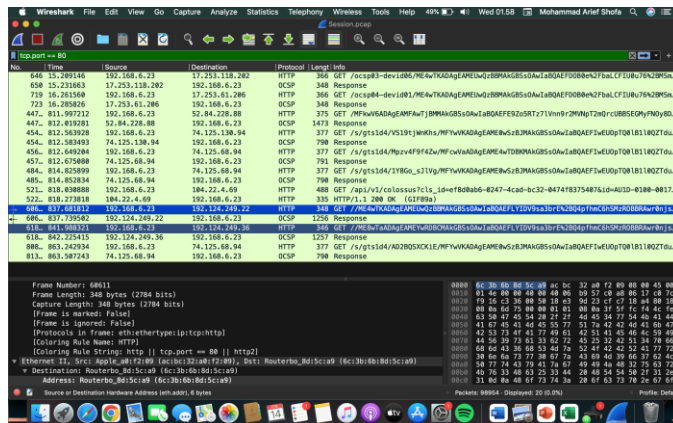
*Identifikasi Tujuan Komunikasi*

Tujuan komunikasi teridentifikasi dengan jelas dari dan ke alamat IP yang berbeda.

*Analisis Keberhasilan Permintaan*

Semua permintaan HTTP dan respons OCSP terlihat berhasil tanpa indikasi kegagalan.

*Log Aktifitas*



Gambar 5. Log Aktifitas Jaringan

Layer 7 – Application Layer

Berdasarkan hasil penelitian pendahuluan, terdapat beberapa temuan penting terkait dengan parameter pada Application Layer. Analisis tersebut memberikan gambaran tentang kinerja sistem pada lapisan aplikasi, khususnya terkait dengan dua set parameter yang diukur. Berikut adalah pengembangan narasi berdasarkan data dan kesimpulan yang diberikan:

Hasil Penelitian Pendahuluan pada Application Layer :

Dalam konteks lapisan aplikasi, terdapat dua set parameter yang dianalisis untuk mengevaluasi performa sistem.

• Parameter 1:

Parameter pertama menunjukkan bahwa terdapat kinerja rendah pada aspek Throughput, sedangkan parameter lainnya memenuhi atau bahkan melebihi standar yang ditetapkan. Meskipun beberapa indikator performa seperti Packet Loss, Delay, dan Jitter memperlihatkan hasil yang memuaskan, Throughput menjadi fokus utama perhatian.

• Parameter 2:

Sebaliknya, parameter kedua secara umum memenuhi standar TIPHON, kecuali pada kategori Throughput yang tergolong pada kategori Sedang. Meskipun demikian, indikator lainnya seperti Packet Loss, Delay, dan Jitter masih berada dalam batas standar yang diinginkan.

• Kesimpulan:

Dari hasil analisis tersebut, terlihat bahwa terdapat kebutuhan perbaikan pada Parameter 1, terutama dalam meningkatkan Throughput guna memenuhi standar TIPHON secara keseluruhan.

Secara keseluruhan, jaringan menunjukkan stabilitas dan kinerja yang baik, dengan beberapa area yang memerlukan perhatian, terutama terkait dengan kualitas kabel dan parameter Throughput pada Application Layer.

Disarankan untuk mempertimbangkan upgrade pada switch yang dapat dikelola dan melakukan perbaikan pada Parameter 1 untuk meningkatkan kinerja jaringan secara menyeluruh.

C. Implementasi Metode HTB dan Queue Tree

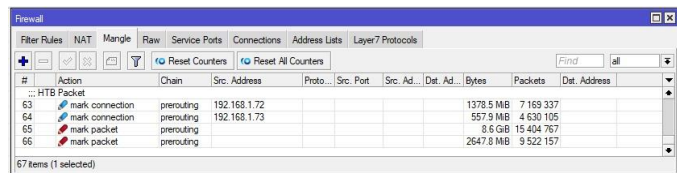
Metode Implementasi Konfigurasi QoS menggunakan HTB dan Queue Tree memberikan keuntungan yang signifikan dibandingkan dengan metode QoS MikroTik lainnya seperti Simple Queue atau Queue Tree tanpa HTB. Dengan HTB,

aliran data dapat diatur secara lebih efisien dengan pengaturan hierarkis yang memungkinkan penentuan prioritas yang lebih tepat dan distribusi bandwidth yang lebih optimal. Sementara itu, Queue Tree memberikan fleksibilitas dalam mengelola antrian dengan lebih detail, memungkinkan penyesuaian aturan sesuai kebutuhan dan prioritas aplikasi.

Dibandingkan dengan Simple Queue, yang sering digunakan untuk mengatur bandwidth pada MikroTik, metode HTB dan Queue Tree menawarkan kontrol yang lebih tinggi dan kemampuan pengaturan yang lebih rumit. Meskipun lebih kompleks dalam konfigurasi, HTB dan Queue Tree memberikan hasil yang lebih baik dalam mengelola trafik jaringan yang kompleks dan memprioritaskan aplikasi yang kritis.

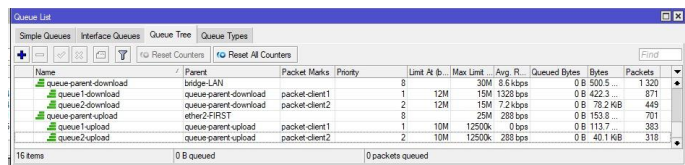
Secara keseluruhan, meskipun metode HTB dan Queue Tree memerlukan pemahaman yang lebih dalam tentang konfigurasi, mereka lebih powerful dan fleksibel dalam mengelola trafik jaringan dibandingkan dengan metode QoS MikroTik lainnya. Oleh karena itu, untuk kebutuhan jaringan PT. PT. XYZ, metode HTB dan Queue Tree menjadi pilihan yang lebih unggul dan sesuai dengan kebutuhan.

- Konfigurasi HTB digunakan untuk mengatur hierarki alokasi bandwidth dengan pembuatan parent queue dan child queue berdasarkan prioritas trafik.



Gambar 6. Konfigurasi HTB

- Sementara itu, metode Queue Tree dimanfaatkan untuk manajemen trafik jaringan dengan pembagian bandwidth untuk alamat IP tertentu serta pengaturan prioritas trafik. Melalui konfigurasi ini, upaya dilakukan untuk meningkatkan efisiensi penggunaan bandwidth dan kualitas layanan pada jaringan MikroTik dengan penekanan pada pengaturan prioritas trafik dan alokasi bandwidth yang terukur.



Gambar 7. Konfigurasi Queue Tree

#### D. Implementasi Metode HTB dan Queue Tree

Analisis metode Hierarchical Token Bucket (HTB) dan Queue Tree dilakukan dengan menggunakan data capture dari aplikasi Wireshark pada tanggal 20 Januari 2024, dalam rentang waktu jam 13.00 - 16.00, yang merupakan periode sibuk kerja.

#### Perhitungan Throughput

Berdasarkan *capture file properties* setelah memfilter ke TCP, dapat dilihat paket-paket data yang didapat dari hasil pengukuran menggunakan *software wireshark*, maka dapat dilakukan perhitungan untuk mencari nilai throughput.

| Measurement            | Captured  | Displayed          | Marked |
|------------------------|-----------|--------------------|--------|
| Packets                | 460732    | 460732 (100.0%)    | —      |
| Time span, s           | 1980.441  | 1980.441           | —      |
| Average pps            | 232.6     | 232.6              | —      |
| Average packet size, B | 589       | 589                | —      |
| Bytes                  | 271424053 | 271424053 (100.0%) | 0      |
| Average bytes/s        | 137 k     | 137 k              | —      |
| Average bits/s         | 1096 k    | 1096 k             | —      |

Gambar 8. Perhitungan Throughput

$$\text{Throughput} = \frac{\text{Paket data diterima}}{\text{Lama pengamatan}}$$

$$\text{Throughput} = \frac{460732}{1980,441}$$

$$\text{Throughput} = 232,641 \text{ bytes/s}$$

Diubah ke satuan bit :

$$\text{Throughput} = 232,641 \frac{\text{bytes}}{\text{s}} \times 8 = 1861 \text{ kbps}$$

$$\% \text{Throughput} = \frac{\text{Paket data diterima}}{\text{Alokasi bandwidth user}} \times 100\%$$

$$\% \text{Throughput} = \frac{1861}{10000 \text{ kbps}} \times 100\% = 18,5\%$$

#### Perhitungan Packet Loss

Berdasarkan hasil *capture* setelah memfilter *tcp.analysis.lost\_segment* dapat dilihat paket-paket data yang didapat dari hasil pengukuran menggunakan *software wireshark*, maka dapat dilakukan perhitungan untuk mencari nilai *packet loss*.

| Measurement            | Captured  | Displayed     | Marked |
|------------------------|-----------|---------------|--------|
| Packets                | 460732    | 1925 (0.4%)   | —      |
| Time span, s           | 1980.441  | 1977.489      | —      |
| Average pps            | 232.6     | 1.0           | —      |
| Average packet size, B | 589       | 265           | —      |
| Bytes                  | 271424053 | 509440 (0.2%) | 0      |
| Average bytes/s        | 137 k     | 257           | —      |
| Average bits/s         | 1096 k    | 2060          | —      |

Gambar 8. Perhitungan Packet Loss

Berikut ini cara perhitungan untuk mendapatkan nilai *packet loss*:

$$\text{Packet Loss} = \frac{\text{Paket data yang hilang}}{\text{Paket data dikirim}} \times 100\%$$

$$\text{Packet Loss} = \frac{1925}{460732} \times 100\%$$

$$\text{Packet Loss} = 13\%$$

*Perhitungan Delay*

Berdasarkan hasil *Export Packet Dissections* yaitu hasil paket-paket yang diekspor dari *software wireshark ke Microsoft Excel*, maka dapat dilakukan perhitungan untuk mencari *delay*. Berikut ini cara perhitungan untuk mendapatkan nilai *delay*:

$$\text{Rata - rata Delay} = \frac{\text{Total Delay}}{(\text{Paket data diterima} - 1)}$$

$$\text{Rata - rata Delay} = \frac{1980441183}{(460732 - 1)}$$

$$\text{Rata - rata Delay} = 4298,466751 \text{ second} = 42 \text{ ms}$$

*Perhitungan Jitter*

Berdasarkan hasil *Export Packet Dissections* yaitu hasil paket-paket yang diekspor dari *software wireshark ke Microsoft Excel*, maka dapat dilakukan perhitungan untuk mencari *jitter*. Berikut ini cara perhitungan untuk mendapatkan nilai *jitter*:

$$\text{Rata - rata Jitter} = \frac{\text{Total Delay}}{(\text{Total paket data} - 1)}$$

$$\text{Rata - rata Jitter} = \frac{\text{Delay} - (\text{rata - rata delay})}{(\text{Total paket data} - 1)}$$

$$\text{Rata - rata Jitter} = \frac{1980476271 - 4298,542907}{(460732 - 1)}$$

$$\text{Rata - rata Jitter} = 4298,542907 \text{ second} = 42 \text{ ms}$$

*E. Analisis Perbandingan*

Pada tabel 8, dapat disimpulkan bahwa implementasi Quality of Service (QoS) menggunakan metode Hierarchical Token Bucket (HTB) dan Queue Tree pada jaringan MikroTik telah memberikan perubahan yang signifikan dalam kinerja jaringan.

Tabel 8. Analisis Perbandingan

| No | Parameter   | Sebelum Impelentasi |                | Sesudah Implementasi |                |
|----|-------------|---------------------|----------------|----------------------|----------------|
|    |             | Hasil perhitun-gan  | Standar TIPHON | Hasil perhitun-gan   | Standar TIPHON |
| 1  | Throughput  | 68,869 kbps / 1,38% | Buruk          | 1861 kbps / 18,5%    | Sedang         |
|    | Packet Loss | 2,59%               | Sangat bagus   | 0,9 %                | Sangat bagus   |
|    | Delay       | 34 ms               | Sangat bagus   | 42 ms                | Sangat bagus   |
|    | Jitter      | 34 ms               | Bagus          | 42 ms                | Sangat bagus   |
| 2  | Throughput  | 3611 kbps / 72%     | Sedang         | 6021 kbps / 41,8%    | Sedang         |
|    | Packet Loss | 6,17%               | Bagus          | 0,9 %                | Sangat Bagus   |
|    | Delay       | 16 ms               | Sangat bagus   | 42 ms                | Sangat bagus   |
|    | Jitter      | 16 ms               | Bagus          | 42 ms                | Sangat bagus   |

Berikut adalah beberapa kesimpulan dari hasil implementasi:

1. Peningkatan Throughput:

Sebelum implementasi, throughput jaringan tergolong rendah, hanya sekitar 68,869 kbps dengan tingkat keburukan yang signifikan. Setelah implementasi, throughput meningkat drastis menjadi 6021 kbps dengan tingkat yang tergolong sedang. Ini menunjukkan peningkatan yang substansial dalam penggunaan bandwidth.

2. Penurunan Packet Loss:

Packet loss sebelum implementasi mencapai 2,59%, yang tergolong dalam kategori yang cukup baik. Setelah implementasi, tingkat packet loss menurun drastis menjadi hanya 0,9%, menunjukkan peningkatan signifikan dalam keandalan jaringan.

3. Stabilitas Delay dan Jitter:

Delay dan jitter jaringan sebelum dan sesudah implementasi QoS tetap terjaga dengan baik, menunjukkan stabilitas yang tinggi dalam pengiriman data.

4. Peningkatan Kinerja secara Keseluruhan:

Secara keseluruhan, implementasi QoS berhasil meningkatkan kinerja jaringan, terutama dalam hal throughput dan packet loss, sementara menjaga stabilitas dan konsistensi dalam delay dan jitter.

5. Pemenuhan Standar TIPHON:

Hasil perhitungan setelah implementasi QoS memenuhi standar TIPHON dengan baik, terutama dalam hal packet loss, delay, dan jitter yang tergolong dalam kategori sangat bagus atau bagus.

Dengan demikian, dapat disimpulkan bahwa implementasi QoS menggunakan metode HTB dan Queue Tree efektif dalam meningkatkan kinerja jaringan MikroTik, yang dapat memberikan manfaat yang signifikan bagi PT. PT. XYZ.

*F. Identifikasi Kelebihan dan Keterbatasan Metode HTB dan Queue Tree*

*Kelebihan Metode HTB*

- Struktur Hirarkis yang Terorganisir: Dengan HTB, kita dapat membuat struktur hirarkis untuk menetapkan prioritas yang jelas pada setiap bagian jaringan.
- Kontrol Granular atas Bandwidth: HTB memungkinkan untuk memiliki kontrol yang lebih terperinci terhadap alokasi bandwidth. Ini memungkinkan kami untuk menentukan batasan bandwidth untuk setiap kelas layanan dengan lebih fleksibel.

*Kelebihan Metode Queue Tree*

- Pengelompokan Trafik yang Mudah: Queue Tree memungkinkan kami untuk mengelompokkan lalu lintas berdasarkan jenis atau tujuan tertentu. Misalnya, kami dapat mengatur prioritas berbeda untuk lalu lintas media sosial dan lalu lintas streaming video.
- Pemantauan Lebih Sederhana: Dalam kasus Queue Tree, monitoring dan manajemen pemantauan lalu lintas tertentu bisa lebih sederhana karena pengelompokan yang jelas.

*Keterbatasan Metode HTB*

- Kompleksitas Konfigurasi: Konfigurasi HTB memerlukan pemahaman mendalam tentang teknologi dan membutuhkan

waktu yang lebih lama untuk setup yang tepat, terutama pada jaringan yang kompleks.

- Responsif terhadap Trafik Bursty: HTB mungkin tidak responsif secara cepat terhadap perubahan-perubahan yang cepat dalam lalu lintas bursty yang tidak terduga.

#### Keterbatasan Metode *Queue Tree*

- Keterbatasan Pada Pengaturan Detail: *Queue Tree* mungkin memiliki keterbatasan dalam memberikan kontrol detail terhadap alokasi bandwidth. Ini bisa menjadi masalah ketika perlu pengaturan yang sangat spesifik.
- Kesulitan Pemantauan pada Jaringan yang Lebih Besar: Dalam jaringan yang lebih besar, pengelompokan dan pemantauan lalu lintas dapat menjadi lebih kompleks, menyebabkan kesulitan dalam pengaturan dan pemantauan.

Analisis ini membantu dalam memahami kekuatan dan kelemahan masing-masing metode dalam implementasi QoS, memungkinkan penggunaan yang lebih efektif dan penyesuaian yang diperlukan sesuai dengan lingkungan jaringan yang spesifik.

#### V. KESIMPULAN

Berdasarkan hasil penelitian, implementasi Quality of Service (QoS) menggunakan metode Hierarchical Token Bucket (HTB) dan *Queue Tree* pada jaringan MikroTik di PT. PT. XYZ terbukti meningkatkan kinerja jaringan secara kuantitatif. Pada kondisi awal, nilai throughput tercatat sebesar 68,869 kbps (1,38%) dengan kategori buruk, sedangkan pada pengukuran kedua sebesar 3611 kbps (72%) dengan kategori sedang. Setelah implementasi, throughput meningkat menjadi 1861 kbps (18,5%) dan 6021 kbps (41,8%), yang menunjukkan pemanfaatan bandwidth menjadi lebih optimal. Nilai packet loss juga mengalami penurunan dari 2,59% dan 6,17% menjadi 0,9%, sehingga kualitas jaringan menjadi lebih baik. Sementara itu, delay berubah dari 34 ms dan 16 ms menjadi 42 ms, namun tetap berada pada kategori sangat bagus. Nilai jitter juga berada pada 42 ms setelah implementasi dan termasuk kategori sangat bagus. Dengan demikian, metode HTB dan *Queue Tree* efektif meningkatkan kualitas layanan jaringan secara menyeluruh.

#### DAFTAR PUSTAKA

- [1] Moh, Suwanto Raharjo, and Joko Triyono, "Analisa Perbandingan Optimalisasi Manajemen Bandwidth Mikrotik Menggunakan Metode *Queue Tree* Dan Simple *Queue* (Studi Kasus Asrama Bogani Yogyakarta Ratmakan GM 1/693)," *Jurnal Jarkom*, vol. 9, no. 1, pp. 21–29, 2021. Available: <https://ejournal.akprind.ac.id/index.php/jarkom/article/view/3671>
- [2] R. W. Lesmana, H. Hannie, and N. Sulistiyowati, "Analisis Quality Of Service (QoS) Pada Laboratorium Komputer MAN 3 Karawang," *CSRID (Computer Sci. Res. Its Dev. Journal)*, vol. 12, no. 3, p. 179, 2021, doi: 10.22303/csrid.12.3.2020.179-190.
- [3] C. Prihantoro, A. K. Hidayah, and S. Fernandez, "Analisis Manajemen Bandwidth Menggunakan Metode *Queue Tree* pada Jaringan Internet Universitas Muhammadiyah Bengkulu," *Just TI (Jurnal Sains Terap. Teknol. Informasi)*, vol. 13, no. 2, p. 81, 2021, doi: 10.46964/justti.v13i2.750.
- [4] A. K. Rahmat Novrianda Dasmien, "Optimasi Jaringan Wireless PT. TASPEN dengan RADIUS," *Techno.COM*, vol. 20, no. No.1, p. 282, 2021.
- [5] M. S. Anwar, "Analisis QoS (Quality of Service) Manajemen Bandwidth menggunakan Metode Kombinasi Simple *Queue* dan PCQ (Per Connection *Queue*) pada Fakultas Teknik Universitas Islam Sumatera Utara," *sudo J. Tek. Inform.*, vol. 1, no. 2, pp. 82–97, 2022, doi: 10.56211/sudo.v1i2.24.
- [6] A. Nurhidayat and W. Sulistyo, "Analisis Dan Implementasi Manajemen Bandwidth Untuk Optimalisasi Layanan Jaringan Internet BUMDes Di Jlumpang," *Progresif: Jurnal Ilmiah Komputer*, vol. 19, no. 2, p. 647, Aug. 2023, doi: 10.35889/progresif.v19i2.1271.
- [7] D. Toresa, "Analisis Qos ( Quality of Service ) Dengan Metode Simple *Queue* Dan Metode Htb ( Hierarchical Token Bucket )," vol. 1, no. 1, pp. 13–24, 2023.
- [8] P. Yanuar, "Analisis Paket Manajemen Bandwidth di Perusahaan Dengan Metode Simple *Queue* dan Quality Of Service," *J. Pendidik. Teknol. Inf.*, vol. 6, no. 1, pp. 11–18, 2023, doi: 10.37792/jukanti.v6i1.840.
- [9] M. Rafi, M. T. Kurniawan, and R. R. Saedudin, "SEIKO : Journal of Management & Business Analisis Perbandingan Manajemen Bandwidth Dengan Metode Random Early Detection ( Red ) Dan Class-Based *Queue* ( Cbq ) Pada Telkom University Landmark Tower ( Tult )," vol. 6, no. 2, pp. 236–246, 2023.
- [10] N. Sari, F. Panjaitan, D. Palembang, J. Jendral Ahmad Yani No, and P. Kota Palembang, "Penerapan Quality of Service Dalam Menganalisis Kualitas Kinerja Metode Simple *Queue* Dan *Queue Tree*," *J. Jupiter*, vol. 15, no. 1, pp. 570–576, 2023.
- [11] S. Suryanto and F. A. Permadi, "Optimalisasi Jaringan Internet Hotspot Menggunakan User Manajemen Pada Pusat Pengembangan SDM Asuransi Indonesia," *J. Infortech*, vol. 1, no. 2, pp. 59–67, 2020, doi: 10.31294/infortech.v1i2.7083.
- [12] F. Leriana, "Analisis QOS (Quality Of Service) Jaringan Komputer Pada Laboratorium Program Studi Pendidikan Teknologi Informasi Fakultas Tarbiyah Dan Keguruan UIN AR-RANIRY Banda Aceh Dan Implementasi Mikrotik Hotspot Bandwith Management Simple *Queue*," vol. 224, no. 11, pp. 122–130, 2019.
- [13] H. P. Situmorang and J. C. Chandra, "Implementasi Manajemen Bandwidth Menggunakan Metode Peer Connection *Queue* Pada SMK Budi Mulia Tangerang," *Idealis*, vol. 2, no. 3, pp. 202–208, 2019.
- [14] R. Faishal Bari, A. Solehudin, and N. Heryana, "Analisis Quality of Service (QoS) Jaringan Internet Berbasis Wireless Local Area Network pada Layanan Indihome," *J. Ilm. Wahana Pendidik.*, vol. 8, no. July, pp. 320–335, 2022.
- [15] M. Muprot and I. Agus Sobari, "Optimalisasi Quality Of Service Peer Connection *Queue* Dengan *Queue Tree* Rw Net Pada Kelurahan Pulau Kelapa Jakarta," *J. Tek. Inform.*, vol. 15, no. 1, pp. 26–32, 2023.
- [16] U. D. Apriza, N. Tjahjamoonsih, F. Imansyah, F. Trias, and E. Kusumawardhani, "Analisis QoS (Quality Of Service) Pada Layanan Internet Jaringan Biznet Home Kota Pontianak," *Journal of Electrical Engineering, Energy, and Information Technology (J3EIT)*, vol. 10, no. 2, 2022, doi: 10.26418/j3eit.v10i2.57249.
- [17] S. Turangga, Martanto, and Y. A. Wijaya, "Analisis Internet Menggunakan Paramater Quality Of Service Pada Alfamart Tuparev 70," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 6, no. 1, pp. 392–398, Apr. 2022, doi: 10.36040/jati.v6i1.4693.
- [18] ETSI, "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); End to End Quality of Service in TIPHON Systems; Part 2: Definition of Quality of Service (QoS) Classes," 2000.
- [19] M. A. K. Qalbi and I. Riadi, "Optimalisasi Jaringan Wireless Menggunakan Quality of Serfice (QoS) dan Algoritma Hierarchical Token Bucket (HTB)," *JSTIE (Jurnal Sarj. Tek. Inform.)*, vol. 7, no. 2, p. 113, 2019, doi: 10.12928/jstie.v7i2.15812.
- [20] Y. B. Pello and 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 )," *JIKO (Jurnal Inform. dan Komputer)* vol. 4, no. 3, pp. 193–198, 2021, doi: 10.33387/jiko.