



Artikel Penelitian

## Analisis Manajemen *Bandwidth* Menggunakan *Hierarchical Token Bucket* Pada Router dengan Standar Deviasi

Pramudhita Ferdiansyah<sup>a\*</sup>, Rini Indrayani<sup>a</sup>, Subektiningsih<sup>a</sup>

<sup>a</sup> Universitas Amikom Yogyakarta, Jln Ringroad Utara, Yogyakarta, 55283, Indonesia

### INFORMASI ARTIKEL

#### Sejarah Artikel:

Diterima Redaksi: 25 Februari 2020

Revisi Akhir: 28 Mei 2020

Diterbitkan Online: 31 Mei 2020

### KATA KUNCI

Bandwidth Manajemen,

Tiphon,

Deviasi

### KORESPONDENSI

E-mail: [ferdian@amikom.ac.id](mailto:ferdian@amikom.ac.id)

### A B S T R A C T

Bandwidth merupakan aspek penting untuk kelancaran dan kenyamanan dalam akses internet. Dalam penggunaan bandwidth internet, setiap pengguna menginginkan kecepatan akses secara maksimal. Kecepatan akses secara maksimal tentu akan berhubungan dengan bandwidth yang tersedia dalam jaringan tersebut. Untuk mendapatkan kecepatan yang maksimal dengan bandwidth yang terbatas, maka diperlukan pengaturan penggunaan bandwidth yang baik untuk menjaga kestabilan trafik lalu-lintas data pada jaringan. Berdasarkan hal tersebut maka dilakukan upaya untuk menyempurnakan teknik manajemen bandwidth.

RouterOS merupakan sistem operasi berbasis linux yang dipergunakan pada router untuk menangani manajemen jaringan dan dapat berjalan di PC maupun *routerboard*. Dengan sistem operasi RouterOS memungkinkan untuk membuat sebuah router sendiri dari komputer langsung. Penelitian ini menekankan bagaimana mengalokasikan bandwidth secara optimal atau merata sesuai dengan kebutuhan pengguna internet dengan model penerapan simpangan baku atau deviasi standar pada *queue* pada router mikrotik

Dengan menerapkan metode *hierarchical token bucket* pada *queue* akan dianalisa kemampuan algoritma tersebut dalam upaya optimalisasi QoS dalam konsep pemerataan *bandwidth*. Analisa yang dilakukan dengan mencari nilai paling optimal dari pemerataan *bandwidth*, yaitu diperolehnya deviasi atau simpangan rata-rata *bandwidth* pada metode *simple queue* dan *hierarchical token bucket*. Pengujian dilakukan dengan membatasi pemakaian *bandwidth* dengan memberikan limitasi untuk batas minimum dan maksimum *bandwidth* secara dinamis. Hasil pengujian menunjukkan model penerapan *hierarchical token bucket* mampu meningkatkan QoS, yaitu dengan diperoleh nilai simpangan lebih sedikit dari nilai rata-rata keseluruhan. Dengan demikian pemakaian *bandwidth* lebih merata sesuai dengan kebutuhan.

## 1. PENDAHULUAN

Penggunaan internet merupakan kebutuhan yang mendasar pada PT. ABC yang merupakan salah satu perusahaan milik negara. Setiap pegawai atau pengguna *bandwidth* memerlukan kecepatan akses internet untuk melakukan aktivitas dan menjalankan aplikasi perkantoran. Kebutuhan akan *bandwidth* internet setiap *user* atau pegawai berbeda-beda sesuai dengan posisi dan penggunaan aplikasi yang diberikan oleh perusahaan. ISP menyediakan *bandwidth* sebesar 30 *Mbps* untuk menangani kebutuhan internet di PT. ABC. Pengaturan QoS atau kualitas layanan penggunaan *bandwidth* internet diperlukan untuk menjaga kestabilan *traffic* data pada jaringan agar tidak terjadi penurunan atau kemacetan internet seperti

terjadinya *bottleneck effect* [1]. Berdasar hal tersebut maka dilakukan upaya untuk memperbaiki atau mengatur ulang setting manajemen penggunaan *bandwidth* pada router.

Implementasi mikrotik routerOS *bandwidth* control jenis *simple queue* dapat mengontrol *throughput* atau *rate* upload dan download dari setiap client yang ada di jaringan. Pada implementasi *bandwidth* control jenis *simple queue* pada skenario berdasarkan IP address, telah diatur *bandwidth* apabila *resource bandwidth* sedang full atau *user* hanya mendapatkan *bandwidth CIR Committed Information Rate* [2].

Penelitian tentang manajemen *bandwidth* dengan melakukan analisis QoS pada pembagian *bandwidth* menggunakan HTB, PCQ dengan layer 7 protocol. Percobaan yang dilakukan oleh Dian Kurnia dengan judul Analisis QoS Pada Pembagian Bandwidth dengan Metode Layer 7 Protocol, PCQ, HTB dan Hotspot di SMK Swasta Al-Washliyah Pasar Senin menghasilkan bahwa metode HTB memiliki nilai *performance* QoS (*throughput*, *jitter*, dan *delay*) yang lebih baik dalam menangani pembagian *bandwidth* [3].

Abdul Malik melakukan penelitian yang berjudul Perbandingan Metode *Simple Queues* dan *Queues Tree* untuk Optimalisasi Manajemen *Bandwidth* Menggunakan Mikrotik. Penelitian tersebut melakukan pengujian pembatasan *bandwidth* yang terhubung ke akses poin terhadap 15 *client* dengan kondisi *download* dan *upload* data. Hasil pengujian tersebut menyimpulkan bahwa metode *queues tree* memiliki nilai *throughput*, *delay*, *packet loss* yang lebih baik sesuai standar TIPHON dibandingkan dengan metode *simple queues* [4].

Penelitian yang dilakukan oleh Soiful Hadi yang membahas tentang manajemen *bandwidth* dengan router mikrotik dengan menggunakan metode *queue tree* menghubungkan protokol jaringan untuk memperlancar akses dengan server forlap Kemenristekdikti. Penelitian tersebut menghasilkan pengaturan *bandwidth* sehingga pengguna internet mendapatkan layanan yang baik dan merata sesuai dengan pengguna [5].

Dalam melakukan optimalisasi *bandwidth* dapat dengan melakukan pengelompokkan trafik aplikasi internet dan kemudian membandingkan kedua parameter. Perbandingan parameter kualitas layanan sebelum dan sesudah optimalisasi terjadi peningkatan efisiensi *bandwidth* setelah optimalisasi selama 2 minggu hari kerja mencapai 200%. Tingkat kepuasan pengguna internet responden menyatakan cukup puas dengan hasil optimalisasi manajemen *bandwidth* internet [6].

Berdasarkan beberapa literatur diatas terdapat beberapa metode untuk melakukan alokasi dan manajemen *bandwidth*. Diantaranya PCQ (*Per Connection Queue*) dengan menggunakan *simple queue*, penjadwalan otomatis dengan menggunakan sistem Bioma. Pada penelitian ini melakukan pengaturan dan alokasi *bandwidth* menggunakan metode pengembangan dari PCQ yaitu HTB (*Hierarchical Token Bucket*) dan dilakukan penjadwalan secara otomatis pada router mikrotik.

## 2. METODE

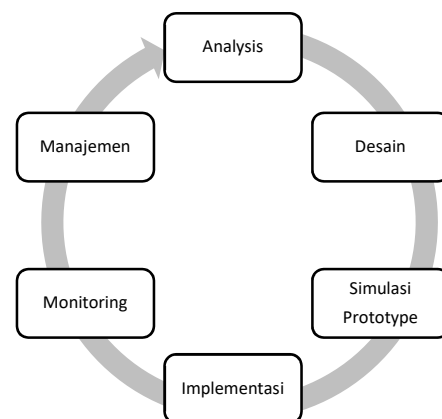
### 2.1. Metode Penelitian

Metode yang digunakan untuk melakukan penelitian ini yaitu dengan menggunakan metode NDLC (*Network Development Life Cycle*) yaitu dengan melakukan *analysis*, *design*, *simulation prototype*, *implementation*, *monitoring* dan *management*. NDLC merupakan metode yang digunakan untuk mengembangkan serta merancang infrastruktur jaringan dan dapat memantau kinerja

jaringan sehingga dapat diketahui statistik kinerjanya [7]. Tahapan yang dilakukan yaitu :

1. Analisis  
Melakukan analisis kebutuhan dan permasalahan yang muncul, menganalisa kebutuhan *client*, dan menganalisa manajemen pembagian *bandwidth*.
2. Desain  
Desain topologi atau pengaturan manajemen *bandwidth* yang akan diimplementasikan berdasar data yang diperoleh dari analisa sebelumnya.
3. Simulasi  
Rancangan pengaturan atau desain sistem disimulasikan untuk menyempurnakan konfigurasi pemerataan *bandwidth*.
4. Implementasi  
Penerapan konfigurasi yang telah diuji pada saat simulasi.
5. Monitoring dan Manajemen.  
Monitoring dilakukan untuk melakukan validasi kesesuaian tujuan dan perbaikan hasil Analisa sebelum dilakukan konfigurasi manajemen *bandwidth*.

Penelitian dilakukan setelah semua kebutuhan sistem didapatkan dari hasil analisis kebutuhan sistem, yaitu manajemen jaringan sesuai dengan kebutuhan *client*. Data analisis diambil melalui *mapping* penggunaan internet yang nantinya akan dikomparasikan dengan data setelah penerapan konfigurasi manajemen *bandwidth*. Metode yang digunakan pada penelitian ini menggunakan NDLC (*Network Development Life Cycle*). Metode tersebut cocok untuk digunakan untuk penelitian model eksperimen karena pada penelitian ini merupakan pengembangan dari arsitektur jaringan pada suatu instansi atau lembaga [8]. Dimana pada tahapan NDLC yaitu melakukan analisis terhadap system yang sudah diterapkan pada tempat penelitian, kemudian melakukan desain ulang sistem dan dilanjutkan dengan mensimulasikan desain atau *prototype*.



Apabila desain protipe yang dirancang sudah sesuai maka desain tersebut diimplementasikan dan selanjutnya dilakukan monitoring.

Gambar 1. Tahapan NDLC

### 2.2. Analisis Data

Analisis data dengan menggunakan router Mikrotik CCR1009 dan routerOS, untuk melakukan akses menggunakan PC dan laptop menggunakan software winbox. Router mikrotik dipilih karena kemudahan penggunaan dan komplekibilitas dari *service* atau

layanan yang sanggup menangani layanan jaringan diatas kemampuan roter rata-rata [9]. Perolehan data penggunaan *bandwidth* diambil dari hasil monitoring dengan menggunakan *wireshark*. Sedangkan untk monitoring trafik data/*bandwidth* dengan menggunakan MRTG yang telah terdapat pada router mikrotik. Sedangkan untuk pengukuran QoS pada manajemen *bandwidth* internet menggunakan parameter standart TIPHON (*Telecommunications and Internet Protocol Harmonization Over Networks*). TIPHON merupakan standart optimasi kinerja QoS agar diketahui seberapa layak kualitas layanan data yang terpenuhi, dimana analisis kualitas yang dikur yaitu *jitter*, *throughput*, *delay*, dan *packet loss* [10].

Tabel 1. Indeks parameter Quality of Services TIPHON

| Nilai    | Persentase | Indeks           |
|----------|------------|------------------|
| 3,8 – 4  | 95 – 100   | Sangat memuaskan |
| 3 – 3,79 | 75 – 95,75 | Memuaskan        |
| 2 – 2,99 | 50 – 74,75 | Kurang memuaskan |
| 1 – 1,99 | 25 – 49,75 | Buruk            |

(sumber : TIPHON, 2002)

Hasil perolehan pengukuran tersebut kemudian akan dicari nilai rata-ratanya pada tiap parameter untuk dicari nilai terkecil daripada simpangannya dengan algoritma standar deviasi. Standar deviasi yaitu algoritma yang digunakan untuk mencari sebaran atau simpangan rata-rata pada sebuah data [11]. Atau dengan kata lain standar deviasi merupakan rumus ilmu statistik yang dapat digunakan untuk mengukur simpangan data selain pengukuran probabilitas [12].

Dengan deviasi standar dihitung sebaran penggunaan *bandwidth* pada tiap pengguna internet pada PT. ABC. Tidak meratanya penyebaran *bandwidth* akan terlihat dari indikasi penurunan *throughput* yang mengakibatkan besarnya *latency* atau *delay* trafik data [13]. Dengan nilai deviasi yang merata atau kecil maka semakin mendekati nilai dari rata-rata dan memperkecil penggunaan *bandwidth* setiap pengguna internet [2].

Deviasi standar dapat didefinisikan sebagai berikut [14] :

$$s = \sqrt{\frac{\sum(x_i - \bar{x})^2}{n - 1}}$$

Dimana  $x$  adalah nilai rata-rata, dari semua deviasi standar nilai-nilai minimum akan tercapai jika  $x = \bar{x}$  berdasarkan sifat aritmatik yang merupakan pendefinisian deviasi standar atau simpangan baku [15].

Permasalahan yang terjadi pada PT. ABC berdasarkan pengamatan selama observasi adalah sebagai berikut :

1. Akses internet diatur dan dimanajemen dengan cara alokasi statis, yang artinya *bandwidth* internet langsung menuju ke *client* sesuai dengan alokasi dan pada saat *bandwidth* tidak digunakan maka tidak dialokasikan ke pengguna lain.

2. Router menggunakan PC dengan windows server 2003 yang membagi *bandwidth* dengan metode *peer connection queue* (PCQ).
3. Pada saat jam kerja penggunaan *bandwidth* internet sangat tinggi

Penerapan konfigurasi dilakukan setelah rancangan selesai diuji dan dapat berjalan di router mikrotik. Sesuai dengan NDLC maka tahap berikutnya yaitu melakukan monitoring terhadap jaringan internet dan melakukan *capturing* atau pengambilan data. Data tersebut yang akan dikomparasikan dengan data sebelum dilakukan proses manajemen *bandwidth*.

### 3. HASIL

Konfigurasi yang dilakukan yaitu melakukan konfigurasi *queue* pada router mikrotik secara dinamis dengan model *Hierarchical Token Bucket* (HTB). Sekenario ini nantinya akan dibandingkan dengan pengaturan *bandwidth* pada saat menggunakan windows server dengan model pengalokasian secara *simple queue* atau model *Peer Connection Queue* (PCQ), dimana model PCQ merupakan metode pemerataan *bandwidth* secara merata sesuai dengan jumlah pengguna daripada *bandwidth* itu sendiri [16]. Pada konfigurasi HTB *parent* pada *queue tree* tidak perlu diberikan besar *limit-at* karena merupakan level tertinggi dan hanya membutuhkan maksimal *bandwidth* yang dialokasikan [17]. Untuk prioritas diberikan *priority* dengan nilai yang sama, hal tersebut dilakukan agar setiap *client* mendapatkan limitasi *bandwidth* yang sama.

```

Terminal
[admin@ICT-Mikroi_server] > queue tree add name="heripratiwi" parent=DOWNLOAD packet-mark=HERIPRATIWI limit-at=1000k queue=default priority=8 max-limit=2000k burst-limit=4M burst-threshold=1200k burst-time=17s
[admin@ICT-Mikroi_server] > queue tree add name="basuki" parent=DOWNLOAD packet-mark=BASUKI limit-at=1000k queue=default priority=8 max-limit=2000k burst-limit=4M burst-threshold=1200k burst-time=17s
[admin@ICT-Mikroi_server] > queue tree add name="sari" parent=DOWNLOAD packet-mark=SARI limit-at=1000k queue=default priority=8 max-limit=2000k burst-limit=4M burst-threshold=1200k burst-time=17s
[admin@ICT-Mikroi_server] > queue tree add name="rini" parent=DOWNLOAD packet-mark=RINI limit-at=1000k queue=default priority=8 max-limit=2000k burst-limit=4M burst-threshold=1200k burst-time=17s
[admin@ICT-Mikroi_server] > queue tree add name="maryanti" parent=DOWNLOAD packet-mark=MARYANTI limit-at=1000k queue=default priority=8 max-limit=2000k burst-limit=4M burst-threshold=1200k burst-time=17s

```

Gambar 2. Konfigurasi pada router

Dengan menerapkan konfigurasi dengan algoritma HTB apabila semua *client* menggunakan *bandwidth* maka setiap pengguna *bandwidth* hanya akan mempergunakannya sesuai dengan *rule* yang telah dimasukkan yaitu batas terendah. Apabila hanya ada 1 *client* yang melakukan akses, maka *client* tersebut akan mendapatkan *bandwidth* keseluruhan dari alokasi *parent* [18].

Pengambilan data menggunakan *wireshark* dan kemudian data yang diperoleh dikelompokkan berdasarkan keperluan data analisis, yaitu *packet loss*, *delay*, *throughput* dan *jitter*. Setiap data yang diperoleh akan diolah untuk diambil sesuai dengan parameter kualitas layanan jaringan atau *quality of service* [19]. Pengukuran parameter QoS dikonversi dan dikomparasikan datanya menggunakan standart TIPHON.

Konfigurasi pada saat menggunakan *simple queue* atau *Per Connection Queue* diperoleh hasil seperti yang ditunjukkan pada tabel dibawah ini.

Tabel 2. Parameter QoS dengan *simple queue*

| No                     | Parameter   | Total pengukuran | Indeks | Standar TIPHON |
|------------------------|-------------|------------------|--------|----------------|
| 1                      | Packet loss | 3,87             | 3      | Bagus          |
| 2                      | Delay       | 19,35            | 4      | Sangat bagus   |
| 3                      | Jitter      | 2,43             | 3      | Bagus          |
| 4                      | Throughput  | 170,89           | 4      | Sangat Bagus   |
| <b>Total rata-rata</b> |             |                  | 3,5    | Memuaskan      |

Setelah dilakukan konfigurasi diperoleh data rata-rata untuk parameter QoS ditunjukkan pada tabel 3 berikut ini.

Tabel 3. Parameter QoS dengan HTB

| No                     | Parameter   | Total pengukuran | Indeks | Standar TIPHON |
|------------------------|-------------|------------------|--------|----------------|
| 1                      | Packet loss | 1.80             | 4      | Sangat bagus   |
| 2                      | Delay       | 1.55             | 4      | Sangat bagus   |
| 3                      | Jitter      | 1.20             | 3      | Bagus          |
| 4                      | Throughput  | 151              | 4      | Sangat Bagus   |
| <b>Total rata-rata</b> |             |                  | 3,75   | Memuaskan      |

Secara garis besar menurut standart TIPHON kedua metode pembagian *bandwidth* termasuk kategori yang memuaskan [20]. Jika dilihat pada tiap parameter akan terlihat perbedaan yang cukup signifikan konfigurasi pemerataan alokasi *bandwidth* yang dihasilkan. Pada parameter *packet loss* diperoleh data yang ditunjukkan oleh tabel 4 dibawah ini. Kemudian dicari nilai dari simpangan pada tiap data atau pengguna untuk mengetahui besar simpangan pada penggunaan *bandwidth*.

Untuk nilai simpangan parameter *packet loss* pada metode *simple queue/PCQ* diperoleh simpangan sebesar :

$$s = \sqrt{\frac{\sum(x_i - \bar{x})^2}{n-1}}$$

$$s = \sqrt{\frac{276,985}{26}} = 3,26$$

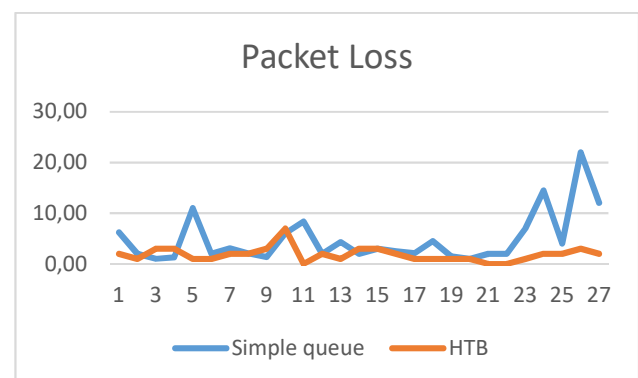
Pada metode pertama (*simple queue/PCQ*) diperoleh simpangan sebesar 3,26. Sedangkan nilai simpangan pada metode *Hierarchical Token Bucket* dihitung dengan rumus deviasi yang sama diperoleh simpangan sebesar :

$$s = \sqrt{\frac{53,61}{26}} = 1,44$$

Tabel 4. Komparasi parameter *packet loss*

| DATA | <i>Simple queue</i> | HTB  |
|------|---------------------|------|
| 1    | 6.21                | 2.00 |
| 2    | 2.00                | 1.00 |
| 3    | 1.00                | 3.00 |
| 4    | 1.30                | 3.00 |
| 5    | 11.02               | 1.00 |
| 6    | 2.00                | 1.00 |
| 7    | 3.03                | 2.00 |
| 8    | 2.07                | 2.00 |
| 9    | 1.38                | 3.00 |
| 10   | 6.02                | 7.00 |
| 11   | 8.35                | 0.00 |
| 12   | 2.00                | 2.00 |
| 13   | 4.30                | 1.00 |
| 14   | 2.00                | 3.00 |
| 15   | 3.02                | 3.00 |
| 16   | 2.50                | 2.00 |
| 17   | 2.11                | 1.00 |
| 18   | 4.50                | 1.00 |
| 19   | 1.50                | 1.00 |
| 20   | 1.00                | 1.00 |
| 21   | 2.00                | 0.00 |
| 22   | 2.00                | 0.00 |
| 23   | 7.00                | 1.00 |
| 24   | 14.50               | 2.00 |
| 25   | 4.00                | 2.00 |
| 26   | 22.00               | 3.00 |
| 27   | 12.00               | 2.00 |

Pada parameter *packet loss* setelah menggunakan metode yang kedua maka diperoleh nilai minimum simpangan sebesar 1,44. Dari perolehan tersebut dapat diperoleh optimalisasi pada *packet loss* sebesar 1,83%. Grafik komparasi parameter *packet loss* ditunjukkan pada gambar 3 dibawah ini.

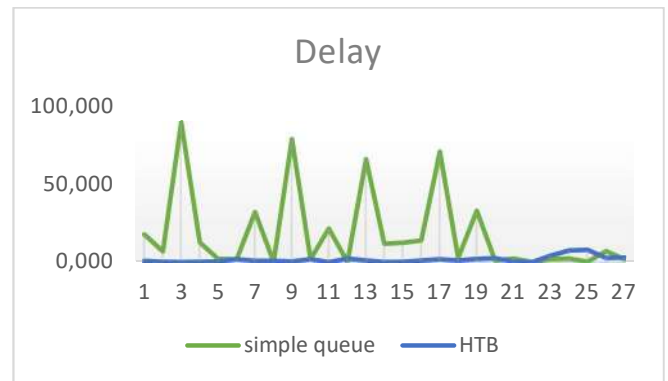
Gambar 3. Komparasi *packet loss*

Parameter berikutnya yaitu *delay*, diperoleh data seperti pada tabel 5 berikut.

Tabel 5. Komparasi parameter *delay*

| DATA | <i>Simple queue</i> | HTB   |
|------|---------------------|-------|
| 1    | 17.788              | 0.717 |
| 2    | 6.803               | 0.232 |
| 3    | 89.816              | 0.057 |
| 4    | 12.699              | 0.109 |
| 5    | 2.115               | 0.420 |
| 6    | 1.829               | 1.735 |
| 7    | 32.142              | 0.787 |
| 8    | 0.084               | 0.758 |
| 9    | 79.018              | 0.300 |
| 10   | 1.802               | 1.915 |
| 11   | 21.493              | 0.030 |
| 12   | 0.020               | 2.335 |
| 13   | 66.176              | 1.022 |
| 14   | 11.858              | 0.027 |
| 15   | 12.502              | 0.157 |
| 16   | 13.782              | 1.050 |
| 17   | 70.864              | 1.854 |
| 18   | 2.709               | 0.948 |
| 19   | 32.932              | 2.059 |
| 20   | 0.930               | 2.495 |
| 21   | 2.322               | 0.323 |
| 22   | 0.026               | 0.029 |
| 23   | 1.829               | 4.200 |
| 24   | 2.142               | 7.353 |
| 25   | 0.084               | 7.875 |
| 26   | 7.018               | 2.581 |
| 27   | 1.802               | 2.966 |

Dari perolehan data seperti yang ditunjukkan oleh tabel 5 diatas dapat dihitung nilai simpangan pada parameter *delay*. Deviasi yang diperoleh pada metode *simple queue* sebesar 26,30 sedangkan setelah menggunakan metode *hierarchical token bucket* diperoleh nilai deviasi sebesar 2,02. Pada parameter *delay* diperoleh optimalisasi yang tinggi, yaitu sebesar 24,28 ms. Grafik komparasi parameter *delay* ditunjukkan oleh gambar 4.



Gambar 4. Komparasi parameter *delay*

Parameter berikutnya yang dianalisis adalah *jitter*. *Jitter* merupakan blok atau bagian-bagian yang berurutan yang nilai dari blok tersebut dipengaruhi oleh beban trafik dan besarnya tabrakan antar paket pada sebuah trafik jaringan [19]. Parameter *jitter* dianalisis dengan kondisi yang sama diperoleh data sebagai berikut.

Tabel 6. Komparasi *jitter*

| DATA | <i>Simple queue</i> | HTB   |
|------|---------------------|-------|
| 1    | 0.000               | 0.000 |
| 2    | 2.410               | 0.000 |
| 3    | 0.131               | 1.382 |
| 4    | 1.451               | 0.000 |
| 5    | 1.227               | 0.000 |
| 6    | 0.000               | 2.302 |
| 7    | 0.000               | 0.000 |
| 8    | 2.417               | 1.023 |
| 9    | 0.442               | 0.000 |
| 10   | 2.522               | 0.000 |
| 11   | 3.245               | 0.000 |
| 12   | 8.114               | 2.678 |
| 13   | 3.226               | 0.000 |
| 14   | 4.025               | 1.060 |
| 15   | 11.321              | 0.000 |
| 16   | 3.053               | 0.000 |
| 17   | 5.246               | 2.050 |
| 18   | 2.938               | 3.000 |
| 19   | 6.128               | 5.309 |
| 20   | 0.482               | 2.000 |
| 21   | 0.000               | 3.214 |
| 22   | 0.024               | 0.485 |
| 23   | 0.000               | 3.263 |
| 24   | 0.000               | 2.302 |
| 25   | 2.417               | 0.000 |
| 26   | 0.442               | 1.023 |
| 27   | 2.522               | 0.000 |

Pada parameter *jitter* dapat diketahui nilai deviasi atau simpangannya yang diambil dari data pada tabel 6 diatas. Maka nilai deviasi dari *jitter* metode pertama (*simple queue*) adalah sebagai berikut :

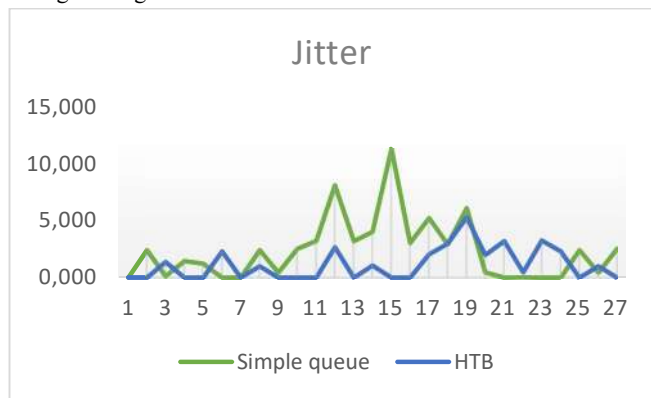
$$s = \sqrt{\frac{\sum(x_i - \bar{x})^2}{n-1}}$$

$$s = \sqrt{\frac{194,15}{26}} = 2,73$$

Sedangkan pada metode kedua yaitu dengan menggunakan metode *hierarchical token bucket* diperoleh nilai deviasi atau simpangannya sebesar :

$$s = \sqrt{\frac{52,74}{26}} = 1,42$$

Dari perolehan deviasi pada kedua metode, maka dapat diketahui bahwa komparasi pada parameter *jitter* diperoleh nilai optimal sebesar 1,308 sec. Komparasi data yang lebih detail ditunjukkan oleh gambar grafik dibawah ini.



Gambar 5. Komparasi parameter *jitter*

Dan yang terakhir yaitu komparasi *throughput*. *Throughput* merupakan gambaran aktual *bandwidth* sebenarnya yang terukur pada satuan waktu tertentu [21]. Hasil monitoring dengan dari dua metode dapat diambil data pada tabel 7.

Sedangkan simpangan *throughput* terhadap tiap pengguna *bandwidth* pada metode *simple queue* adalah sebagai berikut :

$$s = \sqrt{\frac{\sum(x_i - \bar{x})^2}{n-1}}$$

$$s = \sqrt{\frac{1583739,48}{26}} = 246,805$$

Pada metode kedua dengan menggunakan *hierarchical token bucket* diperoleh nilai deviasi sebagai berikut :

$$s = \sqrt{\frac{\sum(x_i - \bar{x})^2}{n-1}}$$

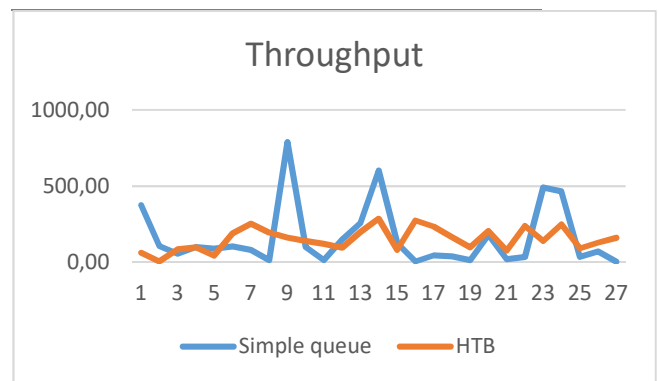
$$s = \sqrt{\frac{390470,73}{26}} = 122,54$$

Pada analisis komparasi pada kedua metode yaitu metode *simple queue* dengan *hierarchical token bucket* diperoleh optimalisasi pada *throughput* sebesar 124,25. Jika dilihat pada grafik akan terlihat jelas simpangan *throughput* pada tiap data.

Grafik komparasi *throughput* ditunjukkan pada grafik di Gambar 6.

Tabel 7. Komparasi *throughput*

| Data | Simple queue | HTB    |
|------|--------------|--------|
| 1    | 375.23       | 62.00  |
| 2    | 106.01       | 5.00   |
| 3    | 55.12        | 85.00  |
| 4    | 98.32        | 97.00  |
| 5    | 89.10        | 42.00  |
| 6    | 103.11       | 190.01 |
| 7    | 79.32        | 252.13 |
| 8    | 12.32        | 193.66 |
| 9    | 789.12       | 160.78 |
| 10   | 98.22        | 138.85 |
| 11   | 12.54        | 120.58 |
| 12   | 147.22       | 95.00  |
| 13   | 257.12       | 198.09 |
| 14   | 601.02       | 285.74 |
| 15   | 124.21       | 80.02  |
| 16   | 5.21         | 272.88 |
| 17   | 44.21        | 233.93 |
| 18   | 38.22        | 164.43 |
| 19   | 13.21        | 96.44  |
| 20   | 179.22       | 204.40 |
| 21   | 19.11        | 76.73  |
| 22   | 34.16        | 237.51 |
| 23   | 491.55       | 138.85 |
| 24   | 465.12       | 248.47 |
| 25   | 34.11        | 91.35  |
| 26   | 70.11        | 127.89 |
| 27   | 2.14         | 160.78 |



Gambar 6. Komparasi parameter *throughput*

#### 4. PEMBAHASAN

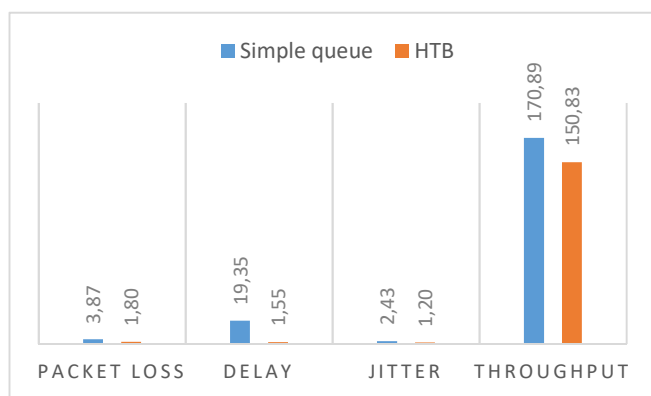
Dari perolehan nilai QoS memiliki sedikit perbedaan, akan tetapi jika dilihat dari pemerataan alokasi *bandwidth* dapat dilihat bahwa setelah implementasi *hierarchical token bucket* pada *queue tree*

lebih sedikit perbedaannya dibandingkan dengan metode *per connection queue* atau *simple queue*. Nilai perbandingan rata-rata perolehan nilai dari parameter QoS dapat dilihat pada tabel 4.

Tabel 4. Komparasi rata-rata tiap parameter QoS

| Parameter   | Simple queue | Hierarchical token bucket |
|-------------|--------------|---------------------------|
| packet loss | 3.87         | 1.80                      |
| delay       | 19.35        | 1.55                      |
| jitter      | 2.43         | 1.20                      |
| throughput  | 170.89       | 150.83                    |

Dari tabel terlihat bahwa parameter *delay* pada metode *simple queue* dengan *hierarchical token bucket* memiliki nilai optimalisasi yang paling signifika. Perbandingan nilai rata-rata penggunaan *bandwidth* tiap parameter dapat juga dilihat pada gambar grafik berikut.



Gambar 7. Komparasi rata-rata tiap parameter.

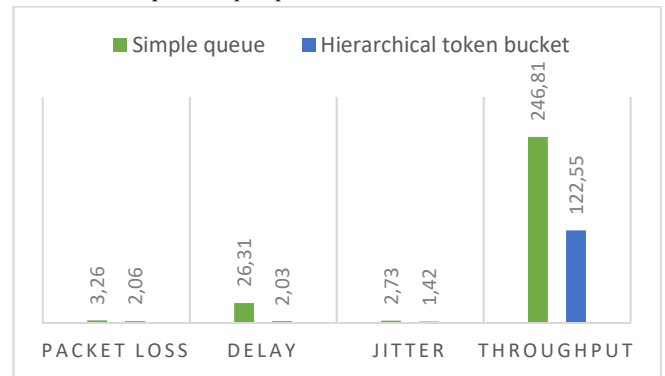
Pada tabel dan grafik terlihat bahwa rata-rata perolehan *qualiti of servie* dari pemakaian *bandwidth* pada metode *simple queue* diperoleh data untuk *packet loss* sebesar 3,87 dan pada metode *hierarchical token bucket* diperoleh rata-rata sebesar 1,80. Dengan demikian pada metode *hierarchical token bucket* lebih optimal. Untuk *delay* diperoleh rata-rata sebesar 19,35 pada metode *simple queue*, dan metode *hierarchical token bucket* diperoleh rata-rata sebesar 1,55. Sedangkan *jitter* diperoleh rata-rata sebesar 2,43 untuk metode *simple queue* dan 1,20 untuk *hierarchical token bucket*. Dan yang terakhir yaitu *throughput* diperoleh rata-rata sebesar 170,89 untuk metode *simple queue* dan 150,83 untuk metode *hierarchical token bucket*.

Tabel 5. Perbandingan deviasi tiap parameter

| Parameter   | Simple queue | Hierarchical token bucket |
|-------------|--------------|---------------------------|
| packet loss | 3.26         | 2.06                      |
| delay       | 26.31        | 2.03                      |
| jitter      | 2.73         | 1.42                      |
| throughput  | 246.81       | 122.55                    |

Sedangkan untuk nilai simpangan (deviasi) atau perbedaan penggunaan *bandwidth* pada tiap pengguna dapat dilihat pada tabel 5.

Konfigurasi alokasi *bandwidth* sebelumnya yang menggunakan metode *simple queue* atau PCQ belum mampu menangani pembagian *bandwidth* secara merata. Skor *Quality of Services* yang yang diperoleh menunjukkan bahwa implementasi HTB pada *queue* lebih baik daripada *simple queue*.



Gambar 8. Komparasi deviasi *bandwidth* tiap parameter

Pada simpangan penggunaan *bandwidth* diperoleh nilai *packet loss* sebesar 3,26 pada metode *simple queue* dan sebesar 2,06 untuk metode *hierarchical token bucket*. Kemudian parameter *delay* diperoleh nilai deviasi pada metode *simple queue* sebesar 26,31 dan 2,03 untuk metode *hierarchical token bucket*. Pada parameter *jitter*, untuk metode *simple queue* diperoleh deviasi sebesar 2,73 dan metode *hierarchical token bucket* diperoleh nilai deviasi sebesar 1,42. Dan parameter *throughput* diperoleh nilai deviasi untuk metode *simple queue* sebesar 246,81 dan metode *hierarchical token bucket* sebesar 122,55.

## 5. KESIMPULAN

Berdasarkan hasil analisa dan pembahasan yang telah dilakukan dapat ditarik kesimpulan bahwa dengan model *hierarchical token bucket* pada *queue tree* memiliki nilai rata-rata lebih sedikit atau kecil dibandingkan dengan metode *simple queue/ per connection queue*. Dengan analisis menggunakan standar deviasi juga terlihat bahwa simpangan penggunaan *bandwidth* rata-rata paling kecil yaitu dengan menggunakan metode *hierarchical token bucket*. Dapat disimpulkan bahwa pemerataan *bandwidth* dengan menggunakan metode *hierarchical token bucket* mampu pemerataan *bandwidth* secara dinamis sesuai kebutuhan pengguna dibandingkan dengan *simple queue*. Namun kedua metode pembagian *bandwidth* tersebut sama-sama memperoleh hasil kualitas layanan yang memuaskan menurut standar TIPHON.

## DAFTAR PUSTAKA

- [1] R. McLeod and G. P. Schell, *MANAGEMENT INFORMATION SYSTEMS*, 10th ed. New Jersey: Pearson education, 2007.
- [2] A. Kumar *et al.*, “BwE: Flexible, hierarchical bandwidth allocation for WAN distributed computing,” *SIGCOMM 2015 - Proc. 2015 ACM Conf. Spec. Interes. Gr. Data Commun.*, pp. 1–14, 2015, doi: 10.1145/2785956.2787478.
- [3] 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, pp. 102–111, 2017, doi: 10.24114/cess.v2i2.6541.
- [4] A. Malik, L. M. F. Aksara, and M. Yamin, “PERBANDINGAN METODE SIMPLE QUEUES DAN QUEUES TREE UNTUK OPTIMASI MANAJEMEN BANDWIDTH MENGGUNAKAN MIKROTIK (STUDI KASUS: PENGADILAN TINGGI AGAMA KENDARI),” vol. 3, no. 2, pp. 1–8, 2017.
- [5] S. Hadi and W. Riska, “IMPLEMENTASI MANAJEMEN BANDWIDTH MENGGUNAKAN QUEUE TREE PADA UNIVERSITAS SEMARANG,” *Pengemb. Rekayasa dan Teknol.*, vol. 15, pp. 112–117, 2019.
- [6] F. Kusuma, “MANAJEMEN BANDWIDTH DAN OPTIMASI JARINGAN MENGGUNAKAN METODE PCQ,” Universitas Muhammadiyah Gresik, 2018.
- [7] O. Puspita, D. Anggorowati, M. T. Kurniawan, and U. Y. K. S. H., “Desain Dan Analisa Infrastruktur Jaringan Wireless Di Pdi-Lipi Jakarta Dengan Menggunakan Metode Network Development Life Cycle ( Ndlc ) Design and Analysis of Infrastructure Wireless Network in Pdi-Lipi Jakarta Using Network Development Life Cycle Nd,” *Telkom Univ.*, vol. 2, no. 2, pp. 5811–5819, 2015.
- [8] K. Rianafirin and M. T. Kurniawan, “Design network security infrastructure cabling using network development life cycle methodology and ISO/IEC 27000 series in Yayasan Kesehatan (Yakes) Telkom Bandung,” *Proc. 2017 4th Int. Conf. Comput. Appl. Inf. Process. Technol. CAIPT 2017*, vol. 2018-Janua, pp. 1–6, 2018, doi: 10.1109/CAIPT.2017.8320681.
- [9] Athailah, *PANDUAN SINGKAT MENGUASAI ROUTER MIKROTIK UNTUK PEMULA*, 1st ed. Jakarta: Mediakita, 2013.
- [10] Y. A. Pranata, I. Fibriani, and S. B. Utomo, “Analisis Optimasi Kinerja Quality of Service Pada Layanan Komunikasi Data Menggunakan Ns-2 Di Pt. Pln (Persero) Jember,” *Sinergi*, vol. 20, no. 2, p. 149, 2016, doi: 10.22441/sinergi.2016.2.009.
- [11] N. Nafi'iyah, “Perbandingan Modus , Median , K\_ Standar Deviasi , Iterative , Mean Dan Otsu Dalam Thresholding,” vol. 8, no. 2, pp. 31–36, 2016.
- [12] M. M. Hanafi, “Risiko, Proses Manajemen Risiko, dan Enterprise Risk Management,” *Manag. Res. Rev.*, pp. 1–40, 2014.
- [13] A. Mukhtarom, A. Basuki, and M. Aswin, “Pemerataan Utilisasi Jaringan Multipath dengan Mode Controller Proactive-Reactive pada Software Defined Networking,” vol. 11, no. 2, pp. 60–64, 2017.
- [14] W. Budiaji, Y. L. A. Salampessy, A. Fakultas, P. Universitas, and S. Ageng, “Dalam Penelitian Pertanian,” *J. Ilmu Penelit. Dan Perikan.*, vol. 1(1), no. 1, pp. 37–42, 2012.
- [15] D. I. Rinawati, D. P. Sari, and F. Muljadi, “Penentuan Waktu Standar Dan Jumlah Tenaga Kerja Optimal Pada Produksi Batik Cap (Studi Kasus: Ikm Batik Saud Effendy, Laweyan),” *J@Ti Undip J. Tek. Ind.*, vol. 7, no. 3, pp. 143–150, 2013, doi: 10.12777/jati.7.3.143-150.
- [16] B. Sugiantoro and Y. B. Mahardhika, “Analisis Quality Of Service Jaringan Wireless Sukanet Wifi Di Fakultas Sains Dan Teknologi Uin Sunan Kalijaga,” *J. Tek. Inform.*, vol. 10, no. 2, pp. 191–201, 2018, doi: 10.15408/jti.v10i2.7027.
- [17] A. S. D. Terto and E. P. Laksana, “ANALYSIS OF BANDWIDTH MANAGEMENT USING QUEUE TREE AS TRAFFIC CONTROL WITH PFIFO METHOD,” *J. Maest. Arsit. dan Tek. Elektro*, vol. 2, no. Oktober, pp. 398–407, 2019.
- [18] A. Syukur, “Analisis Management Bandwidth Menggunakan Metode Per Connection Queue (PCQ) dengan Authentikasi RADIUS,” *It J. Res. Dev.*, vol. 2, no. 2, p. 78, 2018, doi: 10.25299/itjrd.2018.vol2(2).1260.
- [19] M. Unik and R. A. Pratama, “PENERAPAN METODE HTB DAN DIFFSERV GUNA PENINGKATAN QOS,” vol. 9, no. 3, pp. 35–40, 2019.
- [20] Tiphon, *Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 3; End-to-end Quality of Service in TIPHON systems; Part 7: Design guide for elements of a TIPHON connection from an end-to-end speech transmission performance point of*, vol. 1, 2002.
- [21] W. Y. Pusvita and Y. Huda, “ANALISIS KUALITAS LAYANAN JARINGAN INTERNET WIFIID MENGGUNAKAN PARAMETER QOS ( Quality Of Service ) Westi Yulia Pusvita 1), Yasdinul Huda 2) 2,” vol. 7, no. 1, 2019.

## BIODATA PENULIS

Pramudhita Ferdiansyah merupakan civitas akademik di lingkungan Universitas Amikom Yogyakarta yang tertarik pada bidang jaringan computer, cloud computing dan virtualisasi serta memiliki hobi pada bidang fotografi dan bersepeda.

Rini Indrayani. Penulis merupakan civitas akademik di lingkungan Universitas Amikom Yogyakarta yang tertarik pada bidang keamanan data seperti enkripsi dan perundang-undangan IT.

Subektiningsih merupakan civitas akademik di lingkungan Universitas Amikom Yogyakarta yang tertarik pada bidang forensik digital seperti penelusuran pencurian data, penelusuran manipulasi data, dan hukum perdata.