

Analisis Penerapan Manajemen Bandwith pada Jaringan Software Defined Network

Kevin Febrinda Diksar Fahmi Putra¹, I Made Suartana²

^{1,2}Jurusan Teknik Informatika, Universitas Negeri Surabaya

¹kevin.17051204067@mhs.unesa.ac.id

³imadesuartana@unesa.ac.id

Abstrak— Peningkatan kebutuhan akses layanan digital menuntut adanya peningkatan sumber daya yang dibutuhkan. Penggunaan internet oleh masyarakat Indonesia saat ini mencapai sekitar 212,35 juta orang. Hal ini merupakan tantangan bagi penyedia layanan digital agar para pengguna dapat mengakses layanan dengan efektif dan efisien. Oleh karena itu, supaya kecepatan koneksi internet yang berkaitan dengan pengaksesan layanan digital dapat dilakukan dengan mudah tanpa masalah, diperlukan manajemen *bandwidth* agar *bandwidth* bisa dikelola dengan efektif sehingga satu user dengan yang lain menggunakan *bandwidth* yang seimbang dan kinerja jaringan menjadi lebih baik. Metode *Hierarchical Token Bucket* atau disingkat HTB ialah metode yang dapat digunakan untuk mengatur manajemen *bandwidth* adalah HTB. Dimana *bandwidth* akan dibagi ke dalam kelas – kelas sehingga mempermudah proses pengaturannya. Dengan menggunakan QoS (*Quality of Service*), metode HTB diterapkan agar menghasilkan kinerja jaringan yang lebih baik dengan parameter pengujian *throughput*, *delay*, *jitter* dan *packet loss*. Selain manajemen *bandwidth*, pengelolaan jaringan dapat dilakukan dengan mengimplementasikan teknologi SDN (*Software Defined Network*) yang mampu mengelola jaringan dengan konsep jaringan terpusat sehingga menghasilkan jaringan yang lebih stabil dan mudah diatur konfigurasinya. Pada penelitian ini dilakukan analisis penerapan manajemen *bandwidth* pada jaringan SDN. Berdasarkan hasil penelitian dan pengukuran QoS, kinerja jaringan yang menggunakan QoS memperoleh hasil yang lebih baik dan lebih unggul dengan nilai rata - rata *throughput* 107828,033 bp/s, *jitter* 10 s, serta *packet loss* memiliki rata - rata 0,65718048% dan parameter *delay* memiliki rata – rata 12 s dengan kategori sangat bagus. Berdasarkan hasil pengujian tersebut didapatkan kesimpulan bahwa manajemen *bandwidth* pada SDN menggunakan metode *Hierarchical Token Bucket* memiliki kinerja yang lebih unggul, efektif, dan efisien.

Kata Kunci— SDN, HTB, Quality of Service.

I. PENDAHULUAN

Meningkatnya kebutuhan akses layanan digital mengharuskan adanya peningkatan sumber daya yang dibutuhkan. Berdasarkan statistika data pada website [internetworldstats](http://internetworldstats.com), jumlah pengguna internet di Indonesia mendekati 212,35 juta jiwa pada Desember 2021 dan merupakan pengguna internet terbanyak ketiga di Asia. Hal ini merupakan tantangan bagi penyedia layanan digital agar para pengguna dapat mengakses layanan dengan mudah tanpa masalah. Namun tidak semua layanan memiliki infrastruktur yang mumpuni, sehingga diperlukan adanya optimalisasi sumber daya yang ada agar mampu memenuhi kebutuhan atau permintaan.

Sumber daya yang cukup krusial dalam layanan digital adalah jaringan komputer. Pengelolaan jaringan merupakan hal yang paling utama untuk memaksimalkan kinerja jaringan. Pengelolaan tersebut dapat dilakukan dengan mengimplementasikan teknologi SDN

(Software Defined Network) yang mampu mengelola jaringan dengan konsep jaringan terpusat sehingga menghasilkan jaringan yang lebih stabil dan mudah diatur konfigurasinya [7]. SDN memisahkan Control Plane dan Forwarding Plane dengan cara memisahkan sistem pengontrol arus data dari perangkat kerasnya. Pada sebagian banyak jaringan biasanya fungsi forward dan control diletakkan pada perangkat seperti router. Sedangkan, pada jaringan control memiliki fungsi forward yang diletakkan pada suatu perangkat kosong berupa switch (forwarding device) dan fungsi control diletakkan pada software terpusat (controller).

Permasalahan pada penggunaan jaringan yang kurang stabil dapat diakibatkan kurangnya manajemen bandwidth pada perangkat jaringan. Selain itu, permasalahan yang dapat terjadi yaitu semakin banyak perangkat yang mengakses jaringan maka jaringan pada saat yang bersamaan dapat menyebabkan trafik jaringan meningkat. Saat kebutuhan trafik di suatu perangkat meningkat dapat menyebabkan kinerja perangkat dan jaringan menjadi kurang optimal. Sehingga diperlukan sebuah solusi yaitu salah satunya yang dapat dilakukan ialah manajemen bandwidth untuk memaksimalkan kinerja pada suatu jaringan.

Berdasarkan paparan masalah tersebut, diperlukan manajemen jaringan dalam rangka menjaga kinerja suatu jaringan agar tetap stabil. Beragam cara dapat dilakukan untuk menciptakan manajemen jaringan yang optimal salah satunya yakni manajemen bandwidth. Manajemen bandwidth dilakukan dengan cara membagi bandwidth ke masing – masing jaringan agar tidak terjadi dominasi bandwidth di suatu jaringan yang menyebabkan trafik jaringan meningkat. Manajemen bandwidth dapat diimplementasikan melalui beberapa metode salah satunya yaitu metode HTB atau (Hierarchical Token Bucket).

Penelitian sebelumnya yang telah dilakukan menerapkan metode HTB pada SDN dengan controller OpenDaylight [1]. Penelitian tersebut menunjukkan bahwa kualitas jaringan meningkat saat diterapkan metode tersebut. HTB memiliki keunggulan yaitu dapat membatasi bandwidth dengan tepat serta dapat menjamin bandwidth pada setiap kinerja meskipun dalam kondisi trafik jaringan meningkat. Penelitian ini dilakukan dengan skala jaringan kampus yang terdiri dari

Access Switch, Distribution Switch dan Core Switch yang disimulasikan menggunakan Mininet.

Penelitian lain yang dilakukan untuk menerapkan HTB pada SDN, namun yang digunakan *Floodlight Controller2*. Adapun hasil yang didapat yakni penggunaan metode HTB pada jaringan memiliki hasil lebih baik dibandingkan tanpa metode HTB. Berdasarkan standar TIPHON, dengan HTB nilai jitter tidak ada yang melebihi 14ms dan mendapat kategori bagus, packet loss sebesar 3.3% dengan kategori memuaskan, dan throughput sebesar 95.6% dan mendapatkan kategori bagus.

Pengembangan HTB atau Hierarchical Token Bucket dilakukan oleh Martin Devera menggunakan sekumpulan konfigurasi yang sederhana. HTB yakni metode yang memiliki fungsi pada pelaksanaan manajemen bandwidth yang akan diposes secara hirerarki dengan cara pembagian ke dalam kelas – kelas sehingga memudahkan proses pengaturan bandwidth. Terdapat tiga kelas dalam HTB yakni inner, root, dan leaf. HTB dapat melakukan implementasi pembagian trafik dan teknik peminjaman yang lebih akurat [5].

Kualitas jaringan dapat diukur dengan beberapa indikator, yakni *packet loss*, *throughput*, *delay atau latency*, dan *jitter*. Pengukuran ini mempunyai standar agar kualitas jaringan dapat dikatakan baik. Konfigurasi yang dapat dilakukan untuk meningkatkan kualitas jaringan ada banyak. Salah satu konfigurasi yang dapat dilakukan adalah manajemen bandwidth. Dengan menerapkan konfigurasi manajemen bandwidth pada jaringan berarti memberikan aturan pada router atau controller agar membagi bandwidth yang tersedia untuk pengguna jaringan sesuai dengan prioritas yang diberikan.

Penelitian ini akan mengimplementasikan manajemen bandwidth pada SDN dengan menerapkan metode HTB (Hierarchical Token Bucket) yang bertujuan untuk meningkatkan efektivitas kinerja bandwidth. Manajemen bandwidth perlu dilakukan agar bandwidth bisa dikelola dengan baik sehingga satu user dengan yang lain menggunakan bandwidth yang seimbang dan kinerja jaringan menjadi lebih maksimal. Penerapan manajemen bandwidth dilakukan pada controller *OpenDaylight*. Pada proses simulasi atau pengujian, akan digunakan *mininet* sebagai emulator dan *OpenDaylight* sebagai controller untuk mengetahui kemampuan jaringan SDN [4]. Pengujian menggunakan 4 pengukuran QoS yaitu *delay*, *throughput*, *jitter* dan *packet loss*. Nilai dari 4 pengukuran didapatkan saat penggunaan jaringan sebagai media streaming dan terdapat background traffic. .

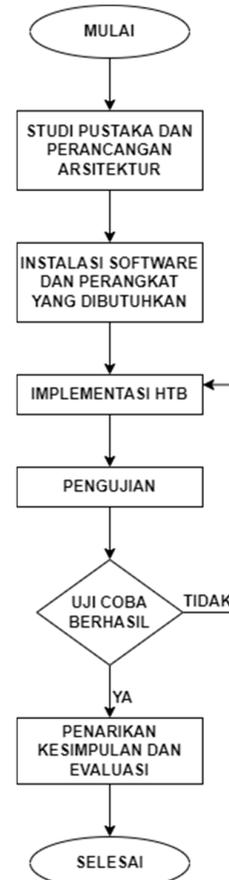
II. METODOLOGI PENELITIAN

Penelitian ini menggunakan *software mininet* untuk emulator dan *OpenDaylight* untuk controller.

A. Alur Penelitian

Beberapa tahapan yang digunakan di penelitian ini dijabarkan pada skema Gambar 1. Penelitian ini diawali dengan pengumpulan studi pustaka yang akan digunakan referensi dalam melakukan penelitian. Setelah dilakukan studi pustaka, peneliti merancang arsitektur jaringan berdasarkan metode yang digunakan. Setelah proses perancangan arsitektur selesai, dilakukan instalasi *software* dan menyiapkan perangkat sesuai

dengan kebutuhan. Sesudah persiapan selesai dimulai proses implementasi HTB diikuti dengan tahap pengujian. Apabila pengujian gagal, sistem harus diperbaiki terlebih dahulu sampai dengan pengujian berhasil. Kemudian yang terakhir yaitu penarikan kesimpulan serta evaluasi terhadap implementasi metode HTB berdasarkan data hasil pengujian.



Gambar 1 Alur Penelitian

B. Manajemen Bandwidth pada SDN

Manajemen *bandwidth* kini sering diimplementasikan pada jaringan dengan tujuan untuk membagi *bandwidth* kepada sesama pengguna di suatu jaringan. Manajemen *bandwidth* merupakan sebuah optimalisasi *traffic* data pada suatu jaringan melalui pengaturan *bandwidth* [6]. Konsep dari *Software Defined Network* (SDN) yakni *software-based* yang terdiri dari *control plane* dan *data plane*, sehingga pengaturan manajemen *bandwidth* tidak lagi menggunakan *hardware-based* [3]. Hal ini menjadikan SDN lebih fleksibel untuk di aplikasikan.

Pada penelitian ini manajemen *bandwidth* yang akan diterapkan pada jaringan SDN menggunakan metode HTB untuk membagi *bandwidth* agar sesama pengguna pada suatu jaringan mendapatkan *bandwidth* yang sama,

sehingga kinerja jaringan sama pada semua pengguna dan tidak saling mengganggu.

Untuk mengukur hasil penerapan metode tersebut digunakan parameter meliputi *throughput*, *delay*, *jiter*, dan *paket loss*. Hasil dari pengujian dan penarikan kesimpulan akan dibahas di hasil dan pembahasan.

Manajemen *bandwidth* yang akan diterapkan jaringan SDN menggunakan metode HTB. Penerapan HTB dilakukan dengan menghubungkan 1 *router* ke 3 PC berbeda. Masing-masing PC menerapkan kebijakan kelas yang berbeda sesuai hirerarki HTB. Hasil kinerja manajemen *bandwidth* pada masing – masing komputer akan dianalisis berdasarkan empat parameter dan disimpulkan performa yang dihasilkan.

Metode HTB mengurutkan jaringan berdasarkan hirarki dan antrian. HTB memiliki beberapa parameter yang akan mem-*filter* dan memprioritaskan paket data yang lewat yakni :

- 1) *Rate*, yaitu parameter yang berfungsi membatasi besar *bandwidth* yang lewat, *rate* akan memotong jumlah data atau menjatuhkan (*drop*) paket data yang terlalu besar.
- 2) *Ceil*, merupakan parameter yang berfungsi untuk menentukan pembagian *bandwidth* antar kelas, pembagian dimulai dari kelas paling bawah urut hingga paling atas. Pembagian *bandwidth* tersebut dapat juga disebut peminjaman *bandwidth* atau *link sharing*.

Adapun tahapan – tahapan penerapan metode HTB pada jaringan SDN yakni:

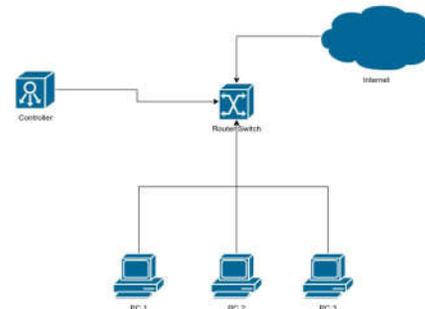
- 1) Setiap perangkat jaringan harus mengaktifkan *OpenFlow* dan *Controller*, kemudian *controller* mendeteksi seluruh perangkat yang tersambung pada jaringan (*Device Provisioning System*). *Controller* bertugas untuk memisahkan *data plane* dan *control plane* pada setiap *switch* serta semua konfigurasi akan diambil alih secara mutlak oleh *controller*. *Switch* berperan sebagai *data plane* yaitu perangkat yang akan menyalurkan aliran data yang berasal dari *controller*. Kemudian *controller* akan menciptakan *Virtual Network* sekaligus mendeteksi interkoneksi di antara *switch*. Topologi yang terdapat pada sistem *GUI controller* akan menampilkan topologi *Virtual Network* yang telah diciptakan oleh *controller*.
- 2) Proses menghubungkan *controller* dengan metode HTB. Tahapan ini memerlukan aplikasi QoS HTB untuk mengatur *rate bandwidth* setiap kelas berupa kode unik (ID) antrian yang biasa disebut *queue id*. Dimana id tersebut akan disebar pada semua *port open vSwitch*. Bahasa *python* digunakan untuk membuat QoS HTB serta diperlukan media interkoneksi yaitu *Java rest API*. Pada saat *controller* sudah bekerja dan tersambung dengan metode HTB selanjutnya dilakukan *traffic classifying* atau klasifikasi pada lalu lintas jaringan. Klasifikasi tersebut mengimplementasikan

protokol pada *layer-3* (IP TCP/UDP) serta *layer-4* (Port TCP/UDP) yang akan menghasilkan data meliputi beberapa paket jaringan. Kemudian data tersebut digunakan untuk membuat bagian – bagian kelas sesuai dengan jenis paket dengan parameter *min-rate* yang memiliki fungsi mengatur banyaknya minimum *bandwidth*, *max-rate* mengatur banyaknya maksimum *bandwidth*, serta *priority* untuk menentukan prioritas pada QoS HTB. Setelah itu sistem akan menyusun *queue tree* atau skema antrian di setiap kelas. Selanjutnya, kelas-kelas tersebut akan dilakukan pengaturan jadwal dengan cara memasukkan masing – masing kelas ke *queuing disciplines (qdisc) scheduling*. Kemudian hasil dikirim pada *controller* berbentuk *flow/policy* menuju tabel *flow*.

- 3) Pada tahap akhir, *controller* pada tabel *flow* akan mendorong *flow/policy configuration* menuju *switch* di setiap jaringan. Kemudian *switch* akan mengatur *flow* untuk melakukan pembatasan *packet queues* dan *bandwidth* pada setiap *host*. Sehingga pada tahap *host* mengakses internet menggunakan *port service* dengan tujuan seperti daftar dalam *flow*, maka *bandwidth* yang terpakai akan dibatasi secara otomatis sesuai dengan ukuran nilai minimal maupun maksimal yang telah ditentukan. Pada sistem berjalan dengan stabil, maka diperlukan pengamatan untuk memantau *traffic bandwidth* dari jaringan. Oleh karena itu, perlu digunakan sistem monitoring *framework sFlow RT* untuk memantau *traffic bandwidth* secara *realtime* yang dapat dibuat menggunakan *Java API* pada *open vSwitch*.

C. Skema Jaringan

Pada penelitian ini akan digunakan topologi sesuai pada gambar 2. Skema jaringan yang digunakan terdiri dari 1 *controller*, 1 *router* dan 3 PC. Penerapan manajemen *bandwidth* dilakukan pada *controller* *OpenDaylight*. Sedangkan pengujian kinerja jaringan menggunakan empat parameter yaitu *delay*, *throughput*, *jiter* dan *packet loss* yang akan diuji menggunakan emulator *Mininet*. Nilai dari parameter tersebut diukur saat penggunaan jaringan sebagai media *streaming* dan terdapat *background traffic*.



Gambar 2 Rancangan Topologi Jaringan

Pada proses simulasi, terdapat dua jenis skenario pengujian yang akan digunakan untuk mengimplementasikan manajemen *bandwidth* menggunakan metode HTB. Skenario pertama masing – masing jaringan menjalankan aplikasi yang berbeda – beda seperti yang digambarkan pada Tabel I sedangkan pada skenario kedua akan dilakukan percobaan menggunakan 2 PC dengan *bandwidth* 1 Mbps kemudian 2 PC tersebut akan melakukan proses *download* bersama. Metode HTB dapat memaksimalkan pembagian *bandwidth* dengan cara mengatur *bandwidth* pada jaringan berdasarkan kelas dengan menerapkan sistem prioritas.

Tabel I
Skenario percobaan 1

	Percobaan 1	Percobaan 2	Percobaan 3
PC 1	Streaming	Download	Browsing
PC 2	Download	Browsing	Streaming
PC 3	Browsing	Streaming	Download

D. Throughput

Throughput yakni besar kecepatan transfer data dengan satuan pengukuran *bit per second (bps)* atau dapat diartikan jumlah seluruh paket yang diterima selama beberapa waktu durasi dibagi dengan durasi interval tersebut. Rumus dari parameter *throughput* dapat dijabarkan dalam persamaan (1).

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

Tabel II
Kategori Throughput

Kategori Throughput	Throughput (bps)	Indeks
Sangat Bagus	100	4
Bagus	75	3
Sedang	50	2
Jelek	<25	1

Sumber : TIPHON

Tabel II menjelaskan standar minimal *throughput* berdasarkan standar TIPHON.

E. Delay

Delay ialah waktu yang digunakan suatu data untuk mencapai jarak dari asal tempat ke arah tempat yang dituju. Media fisik, jarak, *congestion*, atau waktu yang diperlukan untuk menjalankan proses tersebut merupakan poin yang

dapat mempengaruhi hasil pengujian *delay*. Rumus dari parameter *delay* dapat dijabarkan dalam persamaan (2) berikut.

$$\text{Delay} = \frac{\text{Total delay}}{\text{Total paket yang dapat diterima}} \quad (2)$$

Tabel III
Kategori Delay

Kategori Latensi	Besar Delay(ms)	Indeks
Sangat Bagus	< 150 ms	4
Bagus	150 ms s/d 300 ms	3
Sedang	300 ms s/d 450 ms	2
Jelek	> 450 ms	1

Sumber : TIPHON

Tabel III menjelaskan standar minimal *delay* berdasarkan standar TIPHON.

F. Packet Loss

Packet Loss yaitu parameter yang digunakan menunjukkan sebuah kondisi saat jumlah menunjukkan total paket yang hilang, karena *congestion* dan *collision*. Hal ini mempengaruhi aplikasi yang diakses karena terjadinya transmisi ulang yang akan mempengaruhi kinerja efisiensi seluruh jaringan meskipun jumlah *bandwidth* yang tersedia sudah cukup untuk mengakses aplikasi tersebut. *Packet loss* dapat ditemukan dengan rumus pada persamaan (3).

$$PL = \left(\frac{\text{data dikirim} - \text{data diterima}}{\text{data yang dikirim}} \right) \times 100\% \quad (3)$$

Tabel IV
Kategori Packet Loss

Kategori Degradasi	Packet Loss (%)	Indeks
Sangat Bagus	0	4
Bagus	3	3
Sedang	15	2
Jelek	25	1

Sumber : TIPHON

Tabel IV menjelaskan standar minimal *packet loss* berdasarkan standar TIPHON.

H. Jitter

Jitter ialah variasi panjang antrian pada durasi pemrosesan data serta dalam kurun waktu pengumpulan ulang paket yang terdapat pada durasi akhir perjalanan *jitter*. *Jitter* berhubungan erat dengan *latency* dimana *latency*

terdiri dari banyaknya variasi *delay* saat transmisi data pada jaringan. *Delay* yang terdapat pada *switch* dan *router* dapat menyebabkan *jitter*. *Jitter* dapat ditemukan dengan rumus pada persamaan (4) berikut.

$$Jitter = \frac{\text{Total variasi delay}}{\text{Total paket diterima}} \quad (4)$$

Total variasi *delay* = *Delay*–(rata-rata *delay*)

Tabel V
Kategori *Jitter*

Kategori <i>Jitter</i>	<i>Jitter</i> (ms)	Indeks
Sangat Bagus	0 ms	4
Bagus	0 ms s/d 75 ms	3
Sedang	75 ms s/d 125 ms	2
Jelek	125 ms s/d 225ms	1

Sumber : TIPHON

Tabel V menjelaskan standar minimal *jitter* berdasarkan standar TIPHON.

I. Skenario Pengujian

1) Pengujian Topologi

Pada penelitian ini akan digunakan topologi *star* pada pembuatan skema jaringan yang dibuat menggunakan emulator Mininet serta seluruh perangkat yang telah dirancang dan digunakan pada penelitian akan dilakukan testing.

2) Pengujian Throughput

Pada skenario pertama, emulator Mininet mensimulasikan sebuah *switch*, *server* dan beberapa PC dimana *server* akan menjalankan aplikasi yang berbeda – beda sesuai dengan skenario pada Tabel 1. Sedangkan pada skenario kedua, disimulasikan 2 PC dengan *bandwidth* 1 Mbps kemudian 2 PC tersebut akan melakukan proses *download* bersama. Kemudian hasil pengujian pada parameter *throughput* yang telah disimulasikan akan disimpan. Pengujian dilakukan sebanyak tiga kali sesuai dengan skenario serta terdapat *background traffic*.

3) Pengujian Delay

Pengujian ini dapat disimulasikan oleh emulator Mininet dengan cara *switch*, *server*, dan beberapa PC akan menjalankan aplikasi yang berbeda – beda sesuai dengan skenario pertama pada Tabel 1. Selanjutnya, dilakukan skenario kedua dengan 2 PC yang akan diberikan *bandwidth* 1 Mbps kemudian 2 PC tersebut akan melakukan proses *download* bersama. Hasil pengujian pada parameter *delay* yang telah disimulasikan akan disimpan. Pengujian dilakukan sebanyak tiga kali sesuai skenario serta terdapat *background traffic*.

4) Pengujian Packet Loss

Pengujian *packet loss* akan disimulasikan oleh emulator Mininet pada skenario pertama dengan cara *switch*, *server*, dan beberapa PC akan menjalankan

aplikasi yang berbeda – beda sesuai dengan skenario pada Tabel 1. Selanjutnya pada skenario kedua akan disimulasikan 2 buah PC dengan *bandwidth* 1 Mbps kemudian 2 PC tersebut akan melakukan proses *download* bersama. Kemudian hasil pengujian pada parameter *packet loss* yang telah disimulasikan akan disimpan. Pengujian diulang sebanyak tiga kali sesuai dengan skenario serta terdapat *background traffic*.

5) Pengujian Jitter

Emulator Mininet pada skenario pertama mensimulasikan sebuah *switch*, *server* dan beberapa PC dimana *server* akan menjalankan aplikasi yang berbeda – beda sesuai dengan skenario pada Tabel 1. Skenario kedua akan dilakukan dengan mensimulasikan 2 PC dengan *bandwidth* 1 Mbps kemudian 2 PC tersebut akan melakukan proses *download* bersama. Kemudian hasil pengujian pada parameter *jitter* yang telah disimulasikan akan disimpan. Pengujian dilakukan sebanyak tiga kali sesuai dengan skenario serta terdapat *background traffic*.

III. HASIL DAN PEMBAHASAN

Pada tahap ini pengujian yang telah dilaksanakan pada jaringan akan disajikan dan dijabarkan beserta data pengujianya. Hal yang diuji pada penelitian ini adalah penerapan manajemen bandwidth menggunakan metode HTB dan tanpa menggunakan HTB. Berikut ini hasil data dari pengujian yang telah dilakukan sesuai dengan skenario.

A. Implementasi Metode HTB (Hierarchical Token Bucket)

Untuk mengetahui perbedaan kinerja jaringan yang menggunakan metode HTB dan tidak menggunakan HTB, implementasi dan pengujian dilakukan dengan dua tahap yaitu manajemen *bandwith* menggunakan metode HTB dan tanpa menggunakan HTB. Pada pengujian implementasi HTB dilakukan konfigurasi HTB pada sesi *streaming*, *download*, dan *browsing*. Kode konfigurasi implementasi HTB dijelaskan pada Gambar 3.

```

ovs-vsctl -- \
    add-br br0 -- \
    add-port br0 eth0 -- \
    add-port br0 vif1.0 -- set
interface vif1.0 ofport request=5 -- \
    add-port br0 vif2.0 -- set
interface vif2.0 ofport request=6 -- \
    set port eth0 qos=@newqos -- \
    --id=@newqos create qos
type=linux-htb \
    other-config:max-
rate=1000000000 \
    queues:123=@vif10queue \
    queues:234=@vif20queue -- \
    --id=@vif10queue create queue
other-config:max-rate=10000000 -- \
    --id=@vif20queue create queue
other-config:max-rate=20000000
    
```

Gambar 3 Source Code implementasi HTB

B. Analisis Hasil Pengujian Skenario Pertama

1) Analisis Data Hasil Pengujian Throughput

Tabel VI
Perbandingan pengujian Throughput

PC	Dengan QoS (bp/s)	Tanpa QoS (bp/s)
1	107813,1	39611,26
2	107794,7	33720
3	107876,3	37545,32
Rata- rata	107828,033	36958,86

Throughput merupakan jumlah data yang diterima dalam setiap sekor. Semakin tinggi nilai throughput semakin baik kinerja suatu jaringan. Hasil dari pengujian *throughput* yang telah dijabarkan pada Tabel VI menunjukkan adanya perbedaan pada jaringan SDN dengan menerapkan HTB dibandingkan SDN tanpa menerapkan HTB saat menjalankan skenario pengujian. Hasil pengujian kinerja dari jaringan SDN yang memakai HTB memiliki rata – rata 107828,033 bp/s dan jaringan yang tanpa memakai HTB memiliki rata-rata *throughput* 36958,86 bp/s. Dalam hal ini rata-rata nilai *throughput* pada jaringan SDN yang memakai HTB lebih tinggi sehingga dapat dikatakan jaringan ini mempunyai kinerja yang lebih baik.

2) Analisis Data Hasil Pengujian Delay

Tabel VII
Perbandingan Pengujian Delay

Pengujian	Dengan QoS (s)	Tanpa QoS (s)
1	12	30
2	12	40
3	12	36
Rata- rata	12	35,3

Berdasarkan hasil dari pengujian *delay* yang terdapat pada Tabel VII dalam tiga skenario dapat disimpulkan bahwa perbandingan hasil pada pengujian terdapat perbedaan yang sedikit di antara jaringan SDN yang memakai HTB dan jaringan SDN tanpa HTB saat pengujian *delay*. Berdasarkan hasil rata-rata ketiga pengujian tersebut, dapat ditarik kesimpulan bahwasannya jaringan SDN dengan memakai HTB mempunyai kinerja yang lebih unggul dengan nilai rata – rata 12 s daripada jaringan SDN tanpa HTB yang memiliki hasil pengujian rata-rata *delay* 35,3 s. Dalam hal ini rata-rata nilai *delay* pada jaringan SDN yang memakai HTB lebih kecil dan lebih sedikit sehingga dapat dikatakan jaringan ini mempunyai kinerja yang lebih unggul.

3) Analisis Data Hasil Pengujian Packet Loss

Tabel VIII
Perbandingan Pengujian Packet Loss

Pengujian	Dengan QoS (%)	Tanpa QoS (%)
1	1,283356198	11,5487943
2	0,051879451	13,65856413
3	0,63630579	11,82046079
Rata- rata	0,65718048	12,34260641

Pengujian *packet loss* memiliki hasil yang dijabarkan pada Tabel VIII dan dapat ditarik kesimpulan bahwa komparasi hasil pengujian *packet loss* terdapat signifikansi perbedaan rata - rata pada jaringan SDN yang memakai HTB dan jaringan SDN tanpa HTB. Dari hasil data pengujian yang diperoleh, dapat disimpulkan bahwa jaringan SDN dengan memakai HTB memiliki kinerja yang lebih unggul dengan rerata pengujian *packet loss* 0,65718048% daripada jaringan SDN tanpa menggunakan HTB dengan hasil rerata 12,34260641%. Dalam hal ini rata – rata nilai *packet loss* yang dimiliki jaringan SDN dengan HTB lebih sedikit sehingga dapat dikatakan jaringan ini mempunyai kinerja yang lebih baik.

4) Analisis Data Pengujian Jitter

Tabel IX
Perbandingan Pengujian Jitter

Pengujian	Dengan QoS (ms)	Tanpa QoS (ms)
1	10	30
2	10	34
3	10	31
Rata- rata	10	31,6

Berdasarkan hasil pengujian pada Tabel IX sesuai dengan skenario, disimpulkan bahwa kinerja jaringan SDN dengan memakai HTB mempunyai kinerja lebih unggul dengan nilai rata-rata *jitter* 10 s daripada jaringan SDN tanpa memakai HTB dengan rata-rata *jitter* 31,6 s. Sehingga, semakin kecil atau semakin cepat nilai *jitter*, maka semakin baik dan unggul pula jaringan tersebut.

C. Analisis Hasil Pengujian Skenario Kedua

Pada skenario kedua, simulasi dilakukan dengan cara membagi *bandwidth* 1Mbps pada 2 PC yang akan melakukan proses *download* secara bersamaan. Berikut ini merupakan rata – rata hasil pengujian *packet loss*, *delay*, dan *jitter* pada skenario kedua.

Tabel X
Hasil Pengujian Skenario Kedua

Pengujian	Dengan QoS (bp/s)	Tanpa QoS (bp/s)
Packet Loss	590k	598k
Delay	590k	598k
Jitter	590k	598k

Berdasarkan data pada Tabel X, hasil pengujian pada skenario kedua menunjukkan bahwa rerata *packet loss*, *delay*, dan *jitter* menunjukkan nilai yang lebih kecil pada saat diimplementasikan pembagian *bandwidth* menggunakan metode HTB dibandingkan tanpa penggunaan metode HTB. Hal ini menunjukkan bahwa *bandwidth* yang terbagi dapat diterima oleh masing – masing PC secara seimbang.

IV. KESIMPULAN

Berikut ini merupakan hasil dan kesimpulan dari seluruh skenario dan proses pengujian yang telah dilaksanakan pada penelitian ini:

1. Proses manajemen bandwidth menggunakan metode HTB atau *Hierarchical Token Bucket* berhasil dilakukan dengan hasil pengujian yang bervariasi. Hasil pengujian menunjukkan bahwa kinerja jaringan yang memakai HTB lebih unggul pada semua parameter yakni *throughput*, *jitter*, *packet loss* dan *delay*. Dalam hal ini dapat disimpulkan bahwa penerapan SDN menggunakan manajemen *bandwidth* dengan metode HTB (*Hierarchical Token Bucket*) berhasil dilakukan dengan hasil kinerja jaringan yang lebih unggul karena data telah di urutkan dan difilter sesuai prioritasnya sehingga bandwidth tidak terbebani dan paket data tidak mudah rusak.
2. Hasil pengujian skenario pertama dengan parameter *throughput*, *jitter*, *packet loss* dan *delay* pada jaringan SDN yang memakai HTB memiliki performa yang lebih unggul dengan rata-rata *throughput* 107828,033 bp/s berindeks 4 memiliki kategori sangat bagus, *jitter* 10 s berindeks 3 mendapatkan kategori bagus serta *packet loss* memiliki rata - rata 0,65718048% dengan kategori sangat bagus dan parameter *delay* memiliki rata – rata 12 s dengan kategori sangat bagus. Hasil pengujian tersebut dapat ditarik kesimpulan bahwa implementasi manajemen *bandwidth* dengan metode HTB pada SDN memiliki kinerja yang lebih unggul, efektif, dan efisien.
3. Hasil pengujian skenario kedua menunjukkan bahwa pembagian *bandwidth* dapat dilakukan secara seimbang dengan simulasi 2 PC yang menggunakan *bandwidth* 1 Mbps secara bersamaan. Sehingga implementasi

metode HTB dapat dibuktikan bekerja dengan baik dan dapat membagi *bandwidth* secara seimbang di masing – masing PC yang menggunakan jaringan yang sama.

UCAPAN TERIMA KASIH

Ucapan syukur ditujukan kepada Allah SWT, berkat rahmat dan Hidayah-Nya, penelitian ini dapat dilakukan dan berjalan tanpa ada hambatan yang apapun. Serta ucapan terimakasih kepada semua pihak yang telah membimbing, mendukung, dan membantu serta memberikan dukungan penuh kepada penulis sehingga penelitian ini berjalan dan selesai dengan baik dan tepat.

REFERENSI

- [1] Fadli, Ardian. 2018. Implementasi Quality Of Service Pada Campus Network Menggunakan Teknologi Software-Defined Networking Dan OpenDaylight Controller Dengan Metode Hierarchical Token Bucket. Skripsi. Universitas Sumatera Utara. Diunduh dari <http://repositori.usu.ac.id/handle/123456789/6593> diakses tanggal 15 September 2021.
- [2] Iqbal, Muhd. 2020. Analisa Quality of Service pada Jaringan Wireless Berbasis Software-Defined Network dengan Protokol Openflow Menggunakan Floodlight Controller. Artikel Ilmiah. Universitas Syiah Kuala. diunduh dari <http://jurnal.unsyiah.ac.id/JDA/article/view/14771/pdf> diakses tanggal 15 September 2021
- [3] Nugroho, Agung. 2017. Analisis Perbandingan Performa Algoritma Round Robin dan Least Connection untuk Load Balancing pada Software Defined Network. Universitas Brawijaya. diunduh dari <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/download/542/226/> diakses tanggal 15 September 2021
- [4] Eligar, Wijaya. 2020. Load balancing using openday light SDN controller: Case study. Diunduh dari https://www.researchgate.net/publication/346639072_Load_balancing_using_opendaylight_SDN_controller_Case_study diakses tanggal 15 September 2021
- [5] Subhiyanto. 2021. Implementasi Manajemen Bandwidth dengan Metode Hierarchical Token Bucket (HTB) dan Per Connection Queue (PCQ) pada STMIK Antar Bangsa. Diunduh dari <https://ejournal.antarbangsa.ac.id/jti/article/view/436> diakses tanggal 26 Januari 2022
- [6] Prasetya & Ubaidi. 2021. Sistem Manajemen Bandwidth dengan Hierarchical token bucket pada Linux Server Centos (Studi Kasus : Jaringan Kantor Universitas Madura). Diunduh dari <http://jurnal.untagsby.ac.id/index.php/KONVERGENSI/article/view/4672> diakses tanggal 31 Januari 2022
- [7] A.G Respati, Kholis. 2018. Implementasi dan Analisis Bandwidth Management berbasis Website pada Software Defined Network. Diunduh dari <https://repository.telkomuniversity.ac.id/pustaka/142301/implementasi-dan-analisis-bandwidth-management-berbasis-web-pada-software-defined-network.html>. Diakses tanggal 25 Maret 2018
- [8] Fadli, Ardian. 2018. "Implementasi Quality of Service pada Campus Network menggunakan Teknologi Software Defined Networking dan OpenDaylight Controller dengan Metode Hierarchical Token Bucket. Ilmu Komputer dan Teknologi Informasi". Skripsi. Sumatera Utara: Universitas Sumatera Utara.
- [9] Internet World Stats. 2021. ASIA. <https://www.internetworldstats.com/asia.htm>. Diakses tanggal 11 Mei 2022