e-ISSN. 2685-4066
DOI 10.30812/bite.v2i2.917

Jurnal Bumigora Information Technology (BITe)
Vol.2 No.2 (September) 2020, Hal 139-148

Analisa Penerapan *Hierarchical Tokken Buket* Untuk Optimalisasi *Management Bandwith* Pada Server Ubuntu

Riska Haerani[1], Lalu Zazuli Azhar[2]

[1,2]Fakultas Teknik, Universitas Bumigora, Mataram
riska.haerani@gmail.com[1], pujutsega@gmail.com[2]

Abstrak

Perkembangan teknologi informasi pada saat ini semakin cepat, berdasarkan pada kebanyakan area *hotspot* belum memiliki system pada server untuk melakukan optimalisasi *management bandwith*. Sehingga karena terbatasnya *bandwith* pada area-area hotspot tersebut menyebabkan koneksi disetiap *user* terganggu karena belum mempunyai system untuk melakukan *management bandwith* dan pada sisi operator melakukan pembatasan jumlah *bandwith* yang diberikan. Metodologi penelitian yang digunakan adalah *Network Development Life Cycle (NDLC)*, terdapat tiga tahapan yang digunakan yaitu *analysis, design, simulation prototyping*. Pada tahap *analysis* memuat tentang pengumpulan data, tahap *design* memuat tentang rancangan jaringan sedangkan tahap *simulation prototyping* memuat tentang instalasi konfigurasi dan ujicoba. Terdapat 3 ujicoba yang dilakukan yani dengan melakukan pengetesan bandwith menggunakan speedtest, melakukan unduh file menggunakan IDM, dan melakukan streaming youtube, untuk skenarionya dengan melakukan ujicoba setelah penerapan dan sebelum HTB diterapkan. Dari permasalahan yang dipaparkan diatas, penulis ingin membuat system untuk pengoptimalisasi penggunaan bandwith dengan metode HTB (*Hierarchical Token Bucket*), sehingga dapat mengoptimalkan penggunaan *bandwith* dan melakukan penyetaraan terhadap pemakaian dari *user*. Pada pengujian *speedtest* sebelum di aktifkan HTB untuk unduh di dapatkan kecepatan 11,32 Mbps sedangkan untuk unggah 10,93 Mbps, untuk streaming Youtube 6,992 Mbps. Setelah diaktifkan HTB maka pengujian speedtest untuk unduh 1,1 Mbps sedangkan untuk unggah 1,63 Mbps, untuk streaming Youtube 1,025 Mbps.

Kata Kunci : *HTB,* Jaringan, *Server, Speedtest, IDM, Youtube*

Abstract

The development of information technology is increasingly rapid, based on most hotspot areas do not yet have a system on the server to optimize bandwidth management. So because the limited bandwidth in these hotspot areas causes the connection in each user to be interrupted because it does not yet have a system to do bandwidth management and on the operator side does limit the amount of bandwidth given. The research methodology used is Network Development Life Cycle (NDLC), there are 3 stages used namely analysis, design, simulation prototyping. In the analysis phase contains about data collection, the design stage contains about network design while the simulation prototyping stage contains about configuration installation and trials. There are three trials conducted by testing bandwidth using speedtest, downloading files using IDM, and streaming youtube, for the scenario by doing a test after implementation and before HTB is applied. From the above problems, the author wants to create a system for optimizing the use of bandwidth with the HTB (Hierarchical Token Bucket) method, so as to optimize bandwidth usage and equalize the usage of the user. In the speedtest test before activating HTB for download, it gets a speed of 11.32 Mbps while for uploading 10.93 Mbps, for streaming Youtube 6.992 Mbps. After activating HTB, the speedtest test is to download 1.1 Mbps while for uploading 1.63 Mbps, for streaming Youtube 1.025 Mbps

Keywords : HTB, Network, *Server, Speedtest, IDM, Youtube*

I. PENDAHULUAN

Manajemen bandwidth merupakan salah satu cara untuk mengelola koneksi internet bagi perusahaan, institusi maupun sekolah agar koneksi internet yang terbatas bisa dinikmati oleh banyak orang walaupun bandwith yang disewa tidak terlalu besar. Institusi yang menggunakan jaringan computer dan internet sebagai pendukung pekerjaan akan memerlukan pengelolaan bandwidth, terutama jika jaringan tersebut mempunyai beberapa divisi, dimana tiap divisi belum tentu memerlukan koneksi internet dengan koneksi cepat. Maka dengan pengelolaan bandwidth akan memberikan keuntungan bagi institusi[1]

Esdinar Manalu, dkk (2017), dalam penelitian ini menerapkan kontrol penggunaan internet,

menerapkan manajemen bandwidth dan menstabilkan koneksi internet. Metode antrian yang digunakan dalam penelitian ini yaitu menggunakan metode Hierarchical Token Bucket (HTB). Tujuan utama dari metode ini yaitu membuat manajemen bandwidth menggunakan router mikrotik pada jaringan wi-fi dengan metode *simple queue* untuk mengatasi masalah jaringan internet sehingga dapat membagi *bandwidth* secara merata kepada setiap user.)[2].

Dian Kurnia (2017), dalam penelitian ini dilakukan menggabungkan beberapa teknik antrian dan parameter QoS (quality of service) dan dilakukan analisis performansi sistem manajemen bandwidth menggunakan metode HTB (Hierarchical Tocken Bucket), PCQ dengan *layer* 7 *protocol* sebagai limit *file* berekstensi, PCQ rate limit dan *Hotspot*[3].

Lukman, dkk (2018) membahas tentang cara penerapan Hierarchical Token Bucket yang diterapkan diperangkat mikrotik untuk melakukan management bandwith terhadap jaringan hotspot RT/RW net pada sebuah layanan hotspot public yang dimana sebelum penerapan HTB banyak keluhan tentang bandwith yang didapat tidak sesuai dengan paket yang dipilih dan menimbulkan ketidaknyamanan pengguna[4].

Terbatasnya *bandwith* pada suatu area hotspot menyebabkan koneksi disetiap *user* terganggu karena belum mempunyai system untuk melakukan *management bandwith* dan pada sisi operator melakukan pembatasan jumlah *bandwith* yang diberikan. Pada saat ini semakin tingginya tingkat kebutuhan akan internet, tentunya hal ini menjadi masalah tersendiri bagi area-area hotspot apabila tidak memiliki system yang memadai. Tanpa adanya management bandwith, banyak *client* yang dapat menggunakan internet secara tidak beraturan, sehingga menyebabkan client yang lain tidak mendapatkan jatah bandwith yang sesuai. Untuk memanagement bandwith rupanya belum mampu dilakukan oleh pihak area-area *hotspot*.

Dari permasalahan yang dipaparkan diatas, penulis ingin membuat system untuk pengoptimalisasi penggunaan bandwith dengan metode HTB (*Hierarchical Token Bucket*) mengunakan server Ubuntu, HTB merupakan salah satu metode antrian yang adil dan bertujuan menerapkan fungsi link sharing untuk setiap client. Pada HTB terdapat TBF (*Token Bucket Filter*) yang berfungsi sebagai alat estimator yang sangat mudah diimplementasikan dikarenakan hanya dengan menggunakan parameter rate HTB dapat mengeset rate bandwidth yang akan diberikan kepada client. Kelebihan HTB yang lainnya yaitu memiliki parameter ceil yang akan mengatur bandwidth pengguna di antara base rate dan nilai ceil rate-nya. Parameter ceil ini juga merupakan alternatif HTB dalam membagi bandwidth ke client dikarenakan HTB akan memberikan bandwidth yang tersisa apabila bandwidth keseluruhan tidak digunakan oleh client dengan syarat bandwidth tersisa yang akan diberikan masih dibawah nilai rate ceil yang diset. HTB merupakan salah satu metode yang mampu memaksimalkan pengoptimalisasian *management bandwith*, sehingga kualitas pelayanan menjadi lebih meningkat dan terjadi pemerataan *bandwidth* sesuai prioritasnya saat kondisi *traffic* seluruh paket penuh. Sehingga dapat mengoptimalkan penggunaan *bandwith* dan melakukan penyetaraan terhadap pemakaian dari *user*.

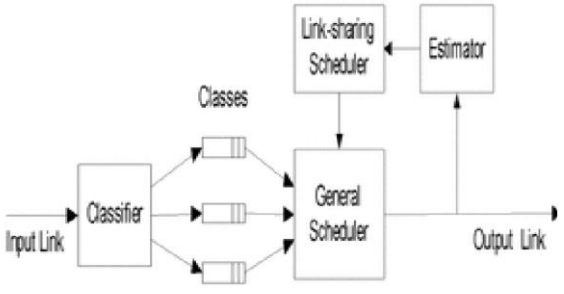
II. METODOLOGI

2.1. *Hierarchical Token Bucket*

HTB merupakan teknik yang menggunakan *Classful Queueing Disciplines* yang memperbolehkan paket diarahkan ke *class* tertentu dan antrian *subqueues*. Sebaik apa *HTB* dalam melakukan pengelolaan *bandwidth* masih perlu diketahui lebih lanjut[5].

2.2. Cara Kerja HTB

Cara Kerja *HTB* tidak ada perbedaan dengan pendahulunya yaitu *CBQ*, hanya saja pada *General Scheduler HTB* menggunakan mekanisme *Deficit Round Robin* (*DRR*) dan pada blok umpan baliknya, *Estimator HTB* tidak menggunakan *Eksponential Weighted Moving Average* (*EWMA*) melainkan *Token Bucket Filter* (*TBF*)[6]



Gambar 1 *Deficit Round Robin*[7]

2.3 Linux

Linux adalah *software system* operasi *open source* yang gratis untuk disebarluaskan di bawah lisensi *GNU*. Jadi pengguna diijinkan untuk menginstal pada computer ataupun menyalin dan menyebarluaskan tanpa harus membayar. Salah satu media dalam pembelajaran *Linux* adalah kamus yang dapat digunakan untuk mencari arti dari kata-kata ataupun

kode yang tidak diketahui. Pemanfaatan *mobile phone* dalam bidang pendidikan dikenal dengan istilah *mobile learning*. *Mobile learning* merupakan sebuah paradigm baru yang telah menciptakan lingkungan pembelajaran baru yaitu pembelajaran yang didukung oleh *mobile phone* dan *transmisinir kabel* berupa teknologi selular.[8]

2.4 Bandwith

Bandwidth, merupakan *rate transfer* data maksimal yang dapat diteruskan antara dua titik.[9]

2.5 Manajemen Bandwith

Manajemen bandwidth merupakan salah satu cara untuk mengelola koneksi internet bagi perusahaan, institusi maupun sekolah agar koneksi internet yang terbatas bisa dinikmati oleh banyak orang walaupun *bandwith* yang disewa tidak terlalu besar. Institusi yang menggunakan jaringan komputer dan internet sebagai pendukung pekerjaan akan memerlukan pengelolaan *bandwidth*, terutama jika jaringan tersebut mempunyai beberapa divisi, dimana tiap divisi belum tentu memerlukan koneksi internet dengan koneksi cepat. Maka dengan pengelolaan *bandwidth* akan memberikan keuntungan bagi institusi.[10]

2.6 VMware

VMware adalah teknologi buatan *Dell* yang menyediakan *platform* perangkat lunak (*software*) untuk melakukan *virtualisasi*. Tidak hanya *software* untuk membuat mesin *virtual* saja, tetapi mereka juga memberikan layanan pendukungnya. Nama *VMware* sendiri diambil dari cabang milik *Dell* yaitu *VMware, In*. Dengan *software VMWare* Kita bias menginstalkan beberapa system operasi termasuk *Linux, Windows*, pada waktu yang sama tanpa kita *merestart* ulang *PC*. Secara mudahnya dapat dikatakan sebagai *PC* di dalam sebuah *PC* .[11]

2.7 Server

Server merupakan sistem komputer yang menyediakan layananlayanan tertentu seperti sistem operasi, program aplikasi maupun data-data informasi kepada komputer lain yang saling terhubung dalam sebuah jaringan komputer. Mengingat fungsi yang dimiliki *server* adalah memberikan layanan kepada *client*, maka *server* dituntut untuk bisa melayani permintaan (*request*) dari semua *client*.[12]

2.8 Client Server

Client server merupakan jaringan yang didesain untuk pengguna, yaitu pengguna tersebut disebut dengan *client*, untuk mengakses sumber daya seperti lagu, koleksi video, atau beberapa layanan lainnya dari komputer pusat yang disebut *server*. Jaringan *client server* merupakan dimana satu komputer terpusat (*server*) adalah pusat dimana banyak komputer pribadi atau *workstation* yang sangat kuat (disebut *client*) terhubung. *Client* menjalankan aplikasi dan mengakses data yang tersimpan dalam *server*.[13]

2.9 IPTables

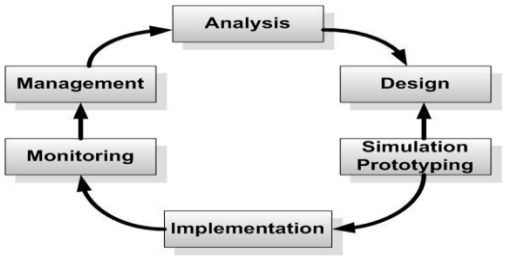
Iptables merupakan *firewall* yang cukup dominan digunakan karena memiliki berbagai macam kemampuan untuk melakukan pengaturan terhadap keluar masuknya paket data[12]

2.10 TCP/IP

TCP/IP protocol suite terdiri dari banyak *protocol* dengan dua *protocol* utama, yaitu *Transport Control Protocol* (*TCP*), dan *Internet Protocol* (*IP*). Dalam *protocol* jaringan *TCP/IP*, sebuah *port* adalah mekanisme yang mengizinkan sebuah computer untuk mendukung beberapa sesi koneksi dengan computer lainnya dan program didalam jaringan. *Port* dapat mengidentifikasi aplikasi dan layanan yang menggunakan koneksi didalam jaringan *TCP/IP*. Sehingga *port* juga mengidentifikasi sebuah proses tertentu dimana sebuah *server* dapat memberikan layanan kepada *client* atau bagaimana sebuah *client* dapat mengakses layanan yang ada pada *server*[13]

2.11 *Network Development Life Cyle*

Metodelogi penelitian yang digunakan adalah *Network Development Life Cycle* (*NDLC*), *NDLC* merupakan suatu model yang melakukan perancangan dan mendefinisikan pengembangan siklus dari jaringan komputer. *NDLC* juga merupakan metode yang bergantung dari pembangunan jaringan komputer sebelumnya. *NDLC* terdiri dari 6 tahapan yaitu *analysis, design, simulation prototyping, implementation, monitoring* dan *management*. Keenam tahapan yang terdapat pada *NDLC*, seperti pada gambar 2.



Gambar 2 *Flowchart* NDLC

Dari 6 tahapan yang ada, penulis hanya menggunakan 3 tahapan yaitu *analysis, design* dan *simulation prototyping*. Sehingga dari 3 tahapan

tersebut penulis dapat melakukan penerapan terhadap *Hierarcichal Token Bucket* dan melakukan analisa pada saat sebelum diterapkannya *Hierarcichal Token Bucket* maupun setelah diterapkan, Berikut penjelasan dari 3 tahapan yang digunakan yaitu:

2.11.1. Tahap Analisis

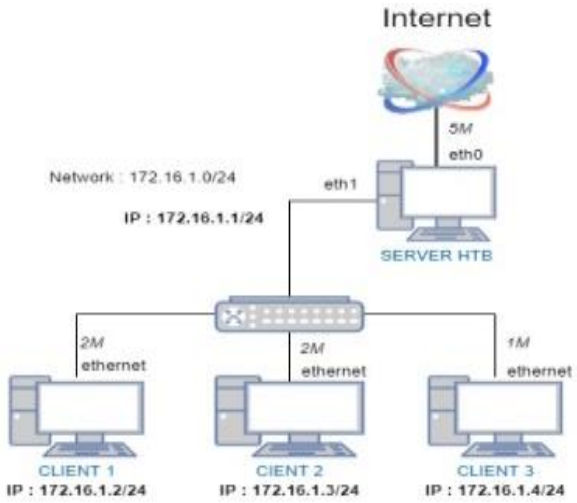
Pada tahapan ini penulis melakukan pengumpulan data baik dari berbagai artikel maupun jurnal ilmiah yang melakukan pembahasan terkait dengan management *bandwith* menggunakan metode *Hierarcichal Token Bucket* yang dipadukan dengan 3 *client* sebagai bahan dalam melakukan ujicoba.

2.11.2. Tahap *Design*

Pada tahapan ini terdiri dari tiga tahapan yaitu rancangan jaringan ujicoba, rancangan pengalamatan ip dan kebutuhan perangkat keras dan perangkat lunak.

1) Rancangan jaringan ujicoba

Rancangan jaringan ujicoba yang digunakan dalam penelitian ini terlihat seprti gambar berikut:



Gambar 3 Rancangan Topologi

Adapun penjelasan dari topologi uji coba *HTB* diatas adalah sebagai berikut:

a) Pada sisi *server* sudah dilakukan konfigurasi untuk setiap *client* yang akan melakukan koneksi terhadap jaringan dengan menggunakan metode *Hierarchical Tocken Bucket* sebagai bahan untuk melakukan pengaturan terhadap *Bandwith* yang akan diatur dan yang akan diberikan pada setiap *client*.
b) Setiap ada *request* koneksi yang masuk ke *server* dari pihak *client* akan di cek oleh *server* menggunakan file konfigurasi dari *tools HTB*, apabila *client* tersebut tidak sesuai dengan aturan pada file konfigurasi *HTB*, maka *client* tersebut tidak dapat dilakukan *Management Bandwith* karena belum terdaftar pada file konfigurasi.
c) Server Ubuntu sebagai jalur penghubung akses internet terhadap jaringan yang ada pada jaringan tersebut yang dimana Sistem Operasi dari *client* menggunakan Sistem Operasi *Windows 7*.
d) Pada *client* tidak bisa melakukan koneksi jaringan apabila pada sisi *server* mati, karena *server* disini difungsikan sebagai gerbang atau jalur terhubungnya *client* ke *internet*.
e) *Server* akan melakukan pengecekan konfigurasi dari file *Hierarchical Tocken Bucket* yang ada pada server dan selanjutnya yang akan diproses untuk *client-client* yang akan melakukan koneksi.
f) Setiap *client* yang akan melakukan koneksi jaringan, *client-client* tersebut akan melewati server sebelum terhubung ke jaringan internet.

2) Rancangan pengalamatan IP

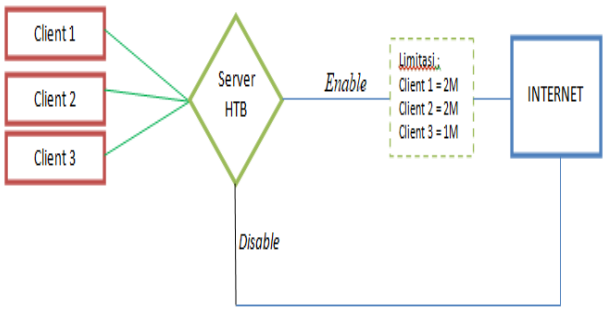
Desain pengalamatan *IP* pada ujicoba analisa penerapan *Hierarchical Tocken Bucket* untuk optimalisasi *Management Bandwith* pada *Server Ubuntu* ini menggunakan kelas B dengan 1 alamat *network* 172.16.1.0/24. Untuk keterangan pengalamatan yang digunakan pada jaringan ujicoba ini dapat dilihat pada tabel 1 berikut:

Tabel 1 Tentang Rancangan Pengalamatan IP

No	Perangkat	IP Address	Netmask	Ket
1	Server	DHCP	DHCP	eth0
2	*Server*	172.16.1.1	255.255.255.0	eth1
3	*Client 1*	172.16.1.2	255.255.255.0	ethernet
4	*Client 2*	172.16.1.3	255.255.255.0	ethernet
5	*Client 3*	172.16.1.4	255.255.255.0	ethernet

3) Rancangan Alur Kerja Sistem

Gambar 4 dibawah ini adalah rancangan *flowchart* proses dari alur kerja *system* yang akan dibuat, seperti terlihat dari gambar berikut:



Gambar 4 Rancangan alur kerja system

Adapun penjelasan dari alur kerja sistem *HTB* diatas adalah sebagai berikut:

a) *Client* 1, 2 dan 3, terhubung ke *server HTB*, yang dimana *server HTB* terhubung langsung ke *Internet*.
b) Pada server *HTB*, konfigurasi limitasi *Bandwidth* untuk masing-masing *client* sudah diatur, yaitu *client* 1 mendapat *bandwidth* 2M, *client* 2 mendapatkan *bandwidth* 2M, dan *client* 3 mendapatkan *Bandwidth* 1M.
c) Jika *service HTB* diaktifkan pada *server*, maka pembagian limitasi *Bandwidth* akan dijalankan dan diberikan limitasi sesuai konfigurasi yang sudah diatur.
d) Jika *service HTB* pada *server* tidak dijalankan (*disable*), maka semua *client* tidak akan diberikan limitasi bandwidth (*client* akan menggunakan seluruh *Bandwidth* yang ada pada *server*).

4) Identifikasi Kebutuhan Perangkat Keras dan Perangkat Lunak

Dalam melakukan analisa *system* yang akan dibangun, terdapat beberapa kebutuhan perangkat yakni dari perangkat keras maupun perangkat lunak yang digunakan untuk melakukan implementasi dan tahapan ujicoba *system*. Adapun kebutuhan yang merupakan dari perangkat keras dan perangkat lunak *sistem* adalah sebagai berikut:

a) Kebutuhan Perangkat Keras

Terdapat beberapa buah kategori kebutuhan perangkat keras pada *system* yang dibangun. Adapun kebutuhan perangkat keras yang digunakan sebagai berikut:

Kebutuhan Perangkat Keras *Server* yang digunakan ialah:

- Laptop Processor AMD A4-9120
- Random Access Memory 1 GB
- Hard Disk 20 GB
- Keyboard
- Mouse
- Network Adapter

Kebutuhan Perangkat Keras c*lient* 1 yang digunakan ialah:

- Laptop Processor AMD A4-9120
- Random Access Memory 1 GB
- Hard Disk 60 GB
- Keyboard
- Mouse
- Network Adapter

Kebutuhan Perangkat Keras *client* 2 yang digunakan ialah:

- Laptop Processor AMD A4-9120
- Random Access Memory 1 GB
- Hard Disk 60 GB
- Keyboard
- Mouse
- Network Adapter

Kebutuhan Perangkat Keras *client* 3 yang digunakan ialah:

- Laptop Processor AMD A4-9120
- Random Access Memory 1 GB
- Hard Disk 60 GB
- Keyboard
- Mouse
- Network Adapter

b) Kebutuhan Perangkat Lunak

Terdapat beberapa buah kategori kebutuhan perangkat lunak pada *system* yang dibangun. Adapun kebutuhan perangkat lunak yang digunakan sebagai berikut:

- *VMWare Workstation Pro 14.1.1* yang digunakan untuk melakukan pemvirtualisasian *server* dan *client*.
- Sistem Operasi *Windows* 7 yang digunakan sebagai sistem operasi dasar pada sisi *client.*
- Sistem Operasi *Windows 10 Home* yang digunakan untuk melakukan tahapan uji coba.
- *Tools* aplikasi *HTB-tools-0.3.0a-i486-1.tgz* yang digunakan untuk optimalisasi *Management Bandwith*.
- Menggunakan *iptables* sebagai pengaturan didalam merutekan koneksi internet dari client yang menuju ke server dan selanjutnya yang akan diteruskan ke *internet*.
- Menggunakan *Speedtest* untuk melakukan percobaan pengetesan koneksi *internet*.
- Aplikasi *Internet Download Manager* yang digunakan untuk melakukan pengecekan *Bandwith* yang diberikan dalam proses download.

- Menggunakan Sosial Media *Youtube* untuk melakukan pengecekan kecepatan download pada saat melakukan pengaksesan video.

2.11.3. Tahap Simulation Prototyping

Pada tahapan ini memuat tentang tahap analisa dari data-data yang didapat, instalasi konfigurasi yang dilakukan pada masing-masing perangkat yang digunakan berdasarkan rancangan ujicoba yang akan diterapkan.

1) Konfigurasi Server

Tahapan pertama adalah melakukan instalasi *server Ubuntu* 16.04 dan selanjutnya melakukan konfigurasi.Adapun proses yang dilakukan untuk melakukan konfigurasi pengalamatan *IP Address* pada *Server* yaitu dengan mengubah *interfaces eth0* dan *eth1*, dengan melakukan perintah sebagai berikut:

```
root@ubuntu:~# nano /etc/network/interfaces
```

2) Konfigurasi IP Tables

Iptables digunakan untuk membuat dan menganalisa tabel-tabel aturan pemfilteran paket *ip* pada sistem operasi *linux* dan juga bisa diartikan sebagai salah satu *tools* yang berfungsi sebagai alat untuk melakukan *filter* (penyaringan) terhadap (*traffic*) lalu lintas data.

Perintah untuk melakukan konfigurasi *iptables* seperti berikut:

root@ubuntu:~# iptables -A FORWARD -i eth0 -n state --state RELATED,ESTABLED -j ACCEPT

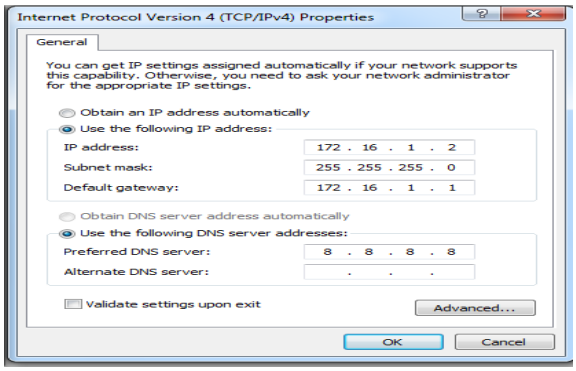
root@ubuntu:~# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

root@ubuntu:~# iptables -A FORWARD -i eth0 -o -j ACCEPT

3) Konfigurasi Client

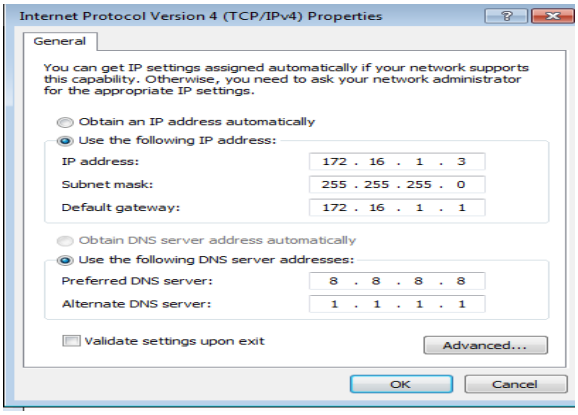
Konfigurasi pengalamatan *IP Address* pada *client* yang menggunakan sistem operasi *windows 7*,yang terhubung dengan *server HTB*.Konfigurasi *ip* dilakukan agar client dapat terhubung dengan *server* dan dapat pula mengakses koneksi *internet*.

Adapun proses yang dilakukan untuk melakukan konfigurasi pengalamatan *IP Address Ethernet* pada client 1 yaitu dengan melakukan konfigurasi *ip* pada *IPv4*. Seperti terlihat pada gambar 5 berikut:



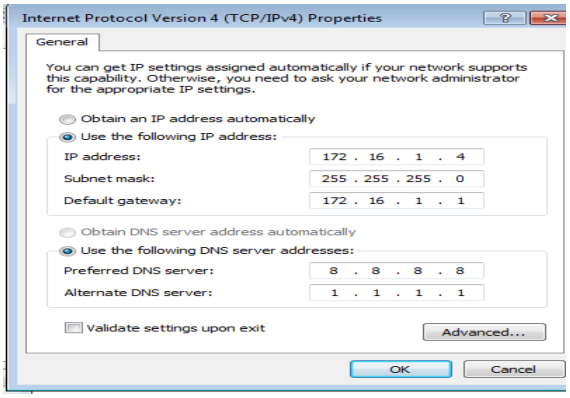
Gambar 5 Hasil Pengalamatan IPv4 Client 1

Pada proses selanjutnya dilakukan konfigurasi pengalamatan *IP Address Ethernet* pada *client* 2 yaitu dengan melakukan konfigurasi *ip* pada *IPv4*. Seperti terlihat pada gambar 6 berikut:



Gambar 6 Hasil Pengalamatan IPv4 Client 2

Kemudian setelah proses diatas selanjutnya dilakukan konfigurasi pengalamatan *IP Address Ethernet* pada client 3 yaitu dengan melakukan konfigurasi *ip* pada *IPv4*. Seperti terlihat pada gambar 7 berikut:



Gambar 7 Hasil Pengalamatan IPv4 Client 3

4) Konfigurasi HTB

Konfigurasi *HTB* meliputi melakukan instalasi paket *tools HTB* agar nantinya *tools* tersebut yang akan digunakan untuk melakukan optimalisasi

Management Bandwith pada suatu jaringan yang akan dilakukan untuk melakukan *Limitasi Bandwith* secara manual yang akan diatur pada file konfigurasi dari *HTB* tersebut. *Tools HTB* digunakan untuk melakukan optimalisasi management bandwith terhadap suatu jaringan yang diterapkan pada *server* sehingga dapat melakukan management koneksi untuk setiap client ke jaringan *server HTB*.Adapun perintah untuk mengunduh *file HTB* sebagai berikut:

```
root@ubuntu:~#wgethttps://dl.dropbox.com/s/tswxjarark1uym3/HTB-tools-0.3.0a-i486-1.tgz
```

Setelah *Tools HTB* selesai diunduh,proses selanjutnya melakukan *extrack* dan installasi. Installasi *tools* dilakukan dengan memindahkan folder *htb* di /sbin ke /sbin yang ada di *server* dengan mengetikan perintah:

```
root@ubuntu :~# mv sbin/htbgen /sbin
root@ubuntu :~# mv sbin/q_checkcfg /sbin
root@ubuntu :~# mv sbin/q_parser /sbin
root@ubuntu :~# mv sbin/q_show /sbin
root@ubuntu :~# mv etc/htb /etc
```

Proses selanjutnya melakukan perubahan nama file yang berada di /etc/htb dengan menghilangkan kata new pada file "eth0-qos.cfg.new" dan "eth1-qos.cfg.new", dengan melakukan perintah seagai berikut:

root@ubuntu :~# mv /etc/htb/ eth0-qos.cfg.new /etc/htb/ eth0-qos.cfg

root@ubuntu :~# mv /etc/htb/ eth1-qos.cfg.new /etc/htb/ eth1-qos.cfg

Pada proses selanjutnya pindahkan file /etc/rc.d/rc.htb.new ke /etc/init.d/ dan ubah nama filenya menjadi rc.htb dengan perintah:

root@ubuntu :~#mv etc/rc.d/rc.htb.new /etc/init.d/rc.htb

Kemudian melakukan pengatuan izin akses dari file rc.htb pada folder "/etc/init.d" dengan melakukan perubahan *permission* sehingga dapat dieksekusi dengan menggunakan perintah:

```
root@ubuntu :~# chmod 775 /etc/init.d/rc.htb
```

Setelah proses diatas, selanjutnya melakukan konfigurasi terhadap file "eth1-qos.cfg" sesuai *topology HTB* yang akan diterapkan, jika *interfaces* yang berhubungan langsung dengan client adalah *eth1* maka yang akan dilakukan konfigurasi adalah file "eth1-qos.cfg". Adapun perintahnya sebagai berikut:

```
root@ubuntu :~# nano /etc/htb/eth1-qos.cfg
```

III. HASIL DAN PEMBAHASAN

Berdasarkan hasil uji coba yang telah dilakukan maka dapat diperoleh hasil analisa sebagai berikut:

a. Sebelum *HTB* diterapkan kedalam *server*, maka client akan melalui *server* terlebih dahulu sebelum melakukan koneksi *internet*, karena *server* tersebut menjadi *gateway* dari client.
b. Pada saat sebelum diterapkan *HTB*, client yang melakukan koneksi *internet* mendapatkan bandwith yang diberikan oleh *Internet Service Provider* tanpa dilakukan *limitasi* pada *server*.
c. Apabila *Internet Service Provider* memiliki bandwith lebih dari 5Mbps, maka *server HTB* hanya akan mengambil bandwith sebesar 5Mbps.
d. Setelah *HTB* diterapkan, pada *server* sudah dilakukan konfigurasi untuk setiap client, konfigurasi tersebut meliputi bandwith, *limit*, dan *upload* yang dimana pada client 1 dan 2 dilakukan *limitasi* sebesar 2Mbps dan pada client 3 dilakukan *limitasi* bandwith sebesar 1Mbps.
e. Pada saat setelah penerapan *HTB*, pada client yang akan melakukan koneksi ke *internet* terlebih dahulu akan melewati *server* dan akan dilakukan pengecekan oleh konfigurasi *HTB* yang terdapat pada file "/etc/htb/eth1-qos.cfg" dimana pada file konfigurasi tersebut terdapat beberapa pengaturan untuk melakukan management bandwith terhadap setiap client.
f. Pada ujicoba yang sudah dilakukan dengan pengujian menggunakan parameter yang berbeda dapat disimpulkan bahwa ujicoba yang dilakukan memiliki hasil yang sama pada pengaturan limit maupun upload.

Pembahasan selanjutnya adalah membahas hasil dari uji coba dengan menggunakan *speedtest* pada setiap client.Tanpa mengaktifkan *HTB* pada *server*,maka setiap client yang melakukan pengaksesan ke *internet* mendapatkan bandwith sepenuhnya dari *internet service provider* dan tidak ada pengaturan *limitasi* yang dilakukan pada *server*.Pada uji coba *speesdtest* ini client dapat mengetahui bandwith yang diberikan.

Pembahasan selanjutnya adalah membahas hasil dari uji coba dengan menggunakan *internet download manager* pada setiap client.Tanpa mengaktifkan *HTB* pada *server*, maka setiap client yang melakukan *download* file dari *internet* mendapatkan bandwith sepenuhnya dari *internet service provider* untuk melakukan *download* file, sehingga tidak ada pengaturan *limitasi* yang dilakukan pada *server*. Pada uji coba *internet download manager* ini client dapat mengetahui besar kecepatan bandwith yang digunakan pada saat *download* file.

Pembahsan selanjutnya adalah membahas hasil dari ujicoba dengan melakukan streaming youtube pada setiap client tanpa mengaktifkan HTB pada server, maka setiap client yang melakukan streaming youtube mendapatkan bandwith sepenuhnya dari internet service provider pada saat mengakses video pada youtube, sehingga tidak ada pengaturan *limitasi* yang dilakukan pada *server*.Pada uji coba streaming youtube ini client dapat mengetahui besar kecepatan bandwith yang digunakan pada saat mengakses video.

Pembahasan selanjutnya adalah membahas hasil dari uji coba dengan menggunakan *speedtest* pada setiap client.Setelah diterapkan *HTB* pada *server*, maka setiap client yang melakukan pengaksesan ke *internet* mendapatkan bandwith sesuai dari *limitasi* bandwith yang sudah diterapkan pada file konfigurasi *HTB*, yang dimana client 1 dan 2 diberikan bandwith sebesar 2Mbps dan pada client 3 diberikan 1Mbps. Setelah diterapkan *HTB* pada uji coba *speedtest* ini, client dapat mengetahui bandwith yang diberikan oleh *server HTB*.

Pembahasan selanjutnya adalah membahas hasil dari uji coba dengan menggunakan *internet download manager* pada setiap client. Setelah diterapkan *HTB* pada *server*, maka setiap client yang melakukan *download* file dari *internet* mendapatkan bandwith sesuai dari *limitasi* bandwith yang sudah diterapkan pada file konfigurasi *HTB*, yang dimana client 1 dan 2 diberikan bandwith sebesar 2Mbps untuk perhitungan Kilo Byte maka 2Mbps=2048KBps, setelah itu 2048 dibagi dengan 8 bit yang dimana hasilnya sebesar 256KBps, dan pada client 3 diberikan 1Mbps untuk perhitungan Kilo Byte maka 1Mbps=1024KBps, setelah itu 1024 dibagi dengan 8 bit yang dimana hasilnya sebesar 128KBps. Setelah diterapkan HTB pada uji coba *internet download manager* ini, client dapat mengetahui besar kecepatan bandwith yang digunakan pada saat *download* file setelah dilakukan *limitasi* pada *server HTB*.

Pembahasan selanjutnya adalah membahas hasil dari uji coba dengan melakukan streaming youtube pada setiap client pada setiap client.Setelah diterapkan *HTB* pada *server*, maka setiap client yang melakukan pengaksesan ke *internet* mendapatkan bandwith sesuai dari *limitasi* bandwith yang sudah diterapkan pada file konfigurasi *HTB*, yang dimana client 1 dan 2 diberikan bandwith sebesar 2Mbps dan pada client 3 diberikan 1Mbps. Setelah diterapkan *HTB* pada uji coba *streaming youtube* ini, client dapat mengetahui bandwith yang diberikan oleh *server HTB*.

Tabel 2 Tabel Perbandingan Sebelum Penerapan HTB

NO	Client	Rata-rata Sebelum Diaktifkan HTB			
		Speedtest		IDM (KB)	Youtube (Mbps)
		Unduh (Mbps)	Unggah (Mbps)		
1	Client 1	11.3	10.93	410.922	6.478
2	Client 2	11.37	10.18	404.349	7.249
3	Client 3	11.28	11.68	404.359	7.249

Pada tabel 2 diatas terlihat nilai rata-rata dari sebelum *HTB* (*Hierarchical Token Bucket*) diaktifkan,bandwith yang digunakan pada setiap client yang mengakses *internet* diberikan secara penuh oleh *server* tanpa adanya *limitasi* bandwith.Tabel merupakan hasil rata-rata dari sepuluh kali percobaan yang dilakukan pada setiap client. Untuk tabel setelah diterapkan *HTB* dapat dilihat pada tabel 3 berikut

Tabel 3 Tabel Perbandingan Sesudah Penerapan HTB

NO	Client	Rata-rata Sesudah Diaktifkan HTB			
		Speedtest		IDM (KB)	Youtube (Mbps)
		Unduh (Mbps)	Unggah (Mbps)		
1	Client 1	1.34	2.00	134.941	0.979
2	Client 2	1.16	2.0	150.098	1.208
3	Client 3	0.8	0.9	150.098	0.8908

Terlihat pada tabel 3 diatas merupakan tabel nilai rata-rata dari sesudah *HTB* (*Hierarchical Token*

Bucket) diaktifkan, bandwith yang digunakan pada setiap client mendapatkan *bandwith* yang sudah ditentukan oleh *server HTB*.Tabel diatas merupakan hasil rata-rata dari sepuluh kali percobaan yang dilakukan pada setiap client untuk mengetahui besar maksimal bandwidth yang diberikan oleh *server HTB*.

IV. KESIMPULAN

Berdasarkan hasil uji coba yang dilakukan dapat disimpulkan, bahwa dari penelitian terkait Analisa Penerapan *Hierarchical Token Bucket Untuk Optimalisasi Management Bantdwith Pada Ubuntu Server* ini, server HTB mampu menerapkan management bandwith pada suatu jaringan dan berhasil melakukan optimalisasi management bandwith pada setiap client

Pada tahap ujicoba setiap client melakukan ujicoba denan menggunakan *speedtest*, *IDM* dan *streaming youtube* yang dimana ketiga ujicoba tersebut digunakan untuk mengetahui bandwith yang diberikan oleh server sebelum diterapkan maupun setelah diterapkan *HTB* dimana pada client 1 mendapatkan bandwith 2Mbps yang sebelum diaktifkan HTB rata-rata yang didapatkan pada speedtest untuk unduh 11.3Mbps dan unggah 10.93Mbps, internet download manager 410.922Mbps dan youtube 6.478Mbps. Sedangkan setelah diaktifkan HTB rata-rata yang didapatkan pada speedtest untuk unduh 1.34Mbps dan unggah 2.00Mbps, internet download manager 134.941Mbps dan youtube 979Mbps. Pada client 2 mendapatkan bandwith 2Mbps yang sebelum diaktifkan HTB rata-rata yang didapatkan pada speedtest untuk unduh 11.37Mbps dan unggah 10.18Mbps, internet download manager 404.349Mbps dan youtube 7.249Mbps. Sedangkan setelah diaktifkan HTB rata-rata yang didapatkan pada speedtest untuk unduh 1.16Mbps dan unggah 2.00Mbps, internet download manager 150.098Mbps dan youtube 1.208Mbps. Pada client 3 mendapatkan bandwith 1Mbps yang sebelum diaktifkan HTB rata-rata yang didapatkan pada speedtest untuk unduh 11.28Mbps dan unggah 11.68Mbps, internet download manager 404.359Mbps dan youtube 7.249Mbps. Sedangkan setelah diaktifkan HTB rata-rata yang didapatkan pada speedtest untuk unduh 0.8Mbps dan unggah 0.9Mbps, internet download manager 150.098Mbps dan youtube 890.8Mbps.

Setiap client akan mendapat bandwith sepenuhnya dari *internet service provider* apabila belum diterapkan *Hierarchical Token Bucket* pada *server*, sedangkan setiap client akan mendapatkan limitasi bandwith apabila sudah diterapkan *Hierarchical Token Bucket* pada server.

Pada ujicoba yang sudah dilakukan dengan pengujian menggunakan parameter yang berbeda dapat disimpulkan bahwa ujicoba yang dilakukan memiliki hasil yang sama pada pengaturan limit maupun upload. Dampak dari penyesuaian parameter ini seperti limit berpengaruh pada maksimum bandwith yang dapat diperoleh setiap client dan pada priority berpengaruh pada bandwith yang tersisa, maka sisa bandwith tersebut dialokasikan kepada client yang menjadi prioritas utama yang dimana prioritas dengan angka yang lebih kecil menjadi prioritas pertama.

REFERENSI

[1] L. Lukman, A. M. Saputro, A. S. Wicaksono, F. H. T. Hartomo, and M. N. Jatun, "Manajemen Bandwidth Menggunakan Metode Hierarchical Token Bucket (HTB) di Farid.net," *Creative Information Technology Journal*, vol. 5, no. 3, p. 209, 2019

[2] E. Manalu and D. Arisandi, "Analisa Management Bandwidth Dengan Metode Antrian Hirarchical Token Bucket," *PROSIDING 2th Celscitech-UMRI 2017*, vol. 2, no. 1, pp. 10–17, 2017.

[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 of Computer Engineering System and Science)*, vol. 2, no. 2, pp. 102–111, 2017.

[4] A. M. Lukman and Y. Bachtia, "Analisis sistem keamanan jaringan dengan," *Computer Engineering, System And Science*, vol. 1, no. 1, pp. 9–14, 2016.

[5] M. Iqbal, P. Studi, and T. Komputer, "Perancangan Sistem Optimalisasi Manajemen Bandwidth dengan HTTB (Hierarchical Token Bucket)," *Jaringan Sistem Informasi Robotik*, vol. 1, no. 01, pp. 25–29, 2017.

[6] R. Tulloh, R. M. Negara, and A. N. Hidayat, "Simulasi Virtual Local Area Network (VLAN) Berbasis Software Defined Network (SDN) Menggunakan POX Controller," *JURNAL INFOTEL - Informatika Telekomunikasi Elektronika*, vol. 7, no. 2, p. 129, 2015

[7] Lisnawita, "Manajemen Bandwidth Menggunakan Metode Hierarchical Token Bucket (HTB)," *Jurnal Teknologi & Komunikasi Digital Zone*, 2016.

[8] R. F. Sri Supatmi, Taufiq Nuzwir Nizar, "Perangkat Pendukung Forensik Lalu Lintas Jaringan," *Jurnal Teknik Komputer Unikom – Komputika*, vol. 3, no. 2, pp. 32–33, 2016.
[9] F. M. Hanif, "Meningkatkan Hasil Belajar Jaringan Dasar Model Pembelajaran Berbasis Proyek (Project Based Learning) Siswa Kelas X TKJ 1 SMKN 1 Bangkinang," *Jurnal Pendidikan Tambusai*, 2018
[10] A. Rahmatulloh and F. MSN, "Implementasi Load Balancing Web Server menggunakan Haproxy dan Sinkronisasi File pada Sistem Informasi Akademik Universitas Siliwangi," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 3, no. 2, pp. 241–248, 2017,
[11] Andrewa and S. Hendi, "Pengolahan Stok Barang Toko Dual," vol. VIII, no. 02, pp. 22–28, 2019.
[12] E. Sularno, "Analisa Dan Implementasi Iptables Dengan Debian Server Sebagai Filtering Firewall Web," vol. 3, no. 1, pp. 106–121, 2016.
[13] A. M. Syafar, "Kendali Perangkat Listrik Dan Monitoring Daya Pada MCB Berbasis TCP/IP," *Jurnal Instek*, vol. 1, no. 1, 2016.