

PENERAPAN METODE HIERARCHY TOKEN BUCKET DALAM MANAJEMEN BANDWITH PADA SMK VETERAN JAMBI

Rico

Program Studi Teknik Informatika, STIKOM Dinamika Bangsa, Jambi

Jl. Jendral Sudirman Thehok Jambi Indonesia, 36139

email : reecho86@gmail.com

ABSTRAK

Dengan majunya perkembangan perangkat teknologi informasi dalam era ini membuat hampir semua orang tidak mau ketinggalan informasi dalam menjalankan berbagai aktivitasnya, maka dari itu hampir semua orang bisa mengakses informasi dimana saja dan kapan saja melalui handphone maupun notebook yang telah dilengkapi dengan jaringan internet. Seperti halnya di SMK Veteran Jambi, siswa dan siswi dapat mengakses internet melalui hotspot yang telah disediakan oleh sekolah dengan notebook atau handphone yang mereka miliki untuk mencari informasi pendukung dalam proses kegiatan belajar. Agar pemakaian internet pada SMK Veteran Jambi bisa optimal dalam area hotspot maka diperlukan suatu kontrol untuk manajemen akses pemakaian bandwidth. Dalam penelitian ini, menggunakan metode Hierarchy Token Bucket (HTB) untuk mengatur bandwidth sesuai dengan kebutuhan pemakainya.

Kata kunci: Jaringan Komputer, Bandwith, HTB, QoS

ABSTRACT

With the rapid advancement of the development of information technology in this era makes almost everyone does not want to miss the information in carrying out various activities, and therefore almost everyone can access information anywhere and anytime via mobile phones and notebooks that have been equipped with internet network. As well as on campus, students can access the internet via hotspot that has been provided by the campus with a notebook or mobile phone they have. In order to optimize the use of the Internet in a hotspot, we need a control for managing access bandwidth usage. In this study, using the Hierarchy Token Bucket (HTB) to set the bandwidth according to the needs of the wearer.

Keywords: Networking, Bandwith, HTB, QoS

1. PENDAHULUAN

Dengan pesatnya Perkembangan internet saat ini, Banyak insitusi maupun lembaga organisasi atau pendidikan yang menggunakan layanan internet secara serentak, penggunaan layanan Internet yang beragam sifatnya secara bebas dapat mengakses semua aplikasi yang ada dalam internet seperti email, web, chatting, browsing, dan multimedia . Penyebabnya bandwidth yang ada telah terambil banyak untuk memenuhi user pertama dan kedua karena untuk melihat video secara online atau download yang membutuhkan bandwidth yang cukup besar, sehingga untuk user ketiga mengalami delay.

Quality of Service (QoS) memegang peranan yang sangat penting dimana Linux sebagai salah satu sistem operasi Linux telah menawarkan beberapa teknik QoS untuk memfasilitasi proses manajemen bandwidth pada suatu jaringan Internet. Salah satu metode yang dapat diterapkan yaitu *Hierarchical Token Bucket* (HTB), metode ini akan menjamin pengguna jaringan mendapatkan bandwidth yang sesuai sehingga kinerja jaringan Internet tetap berjalan dengan baik dan lancar.

Rumusan masalah dalam penelitian ini adalah “Bagaimana cara mengoptimalkan pemakaian bandwidth”. Agar penelitian ini sesuai dengan yang direncanakan maka perlu diberikan batasan yang meliputi ; 1) Manajemen bandwidth terhadap user internet pada SMK Veteran Jambi; 2) Menggunakan metode *Hierarchical Token Bucket* (HTB).

Tujuan penelitian ini adalah menerapkan metode *Hierarchical Token Bucket* (HTB) dalam mengoptimalkan penggunaan bandwidth pada SMK Veteran Jambi.

2. LANDASAN TEORI

2.1 Quality of Service (QoS)

Menurut Ferguson & Huston, 1998 : *Quality of Service (QoS)* merupakan metode pengukuran tentang seberapa baik jaringan dan merupakan suatu usaha untuk mendefinisikan karakteristik dan sifat dari satu servis.

QoS digunakan untuk mengukur sekumpulan atribut kinerja yang telah dispesifikasikan dan diasosiasikan dengan suatu servis. QoS memiliki beberapa komponen diantaranya :

1. Delay

Delay merupakan komponen dari QoS yang menunjukkan total waktu yang dilalui suatu paket dari pengirim ke penerima melalui sebuah jaringan.

2. Jitter

Jitter merupakan variasi *delay* antar paket yang terjadi pada suatu jaringan berbasis IP. Variasi beban trafik dan besarnya tumbukkan antar paket (*congestion*) yang ada dalam suatu jaringan bisa mempengaruhi besarnya nilai *jitter*. Semakin besar beban trafik di dalam jaringan akan menyebabkan semakin besar pula peluang terjadinya *congestion*, dengan demikian nilai *jitter* akan semakin besar. Semakin besar nilai *jitter* akan mengakibatkan nilai QoS akan semakin turun. Kategori kinerja jaringan berbasis IP dalam *jitter* versi *Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON)* mengelompokkan menjadi empat kategori penurunan kinerja jaringan berdasarkan nilai *jitter* seperti terlihat pada Tabel 1.

Tabel 1.
Kategori Jitter

Kategori Degradasi	Peak Jitter
Sangat Bagus	0 ms
Bagus	75 ms
Sedang	125 ms
Jelek	225 ms

3. Bandwidth

Bandwidth merupakan kapasitas atau daya tampung kabel Ethernet agar dapat dilewati trafik paket data dalam jumlah tertentu. *Bandwidth* juga biasa berarti jumlah konsumsi paket data per satuan waktu dinyatakan dengan satuan *bit per second (bps)* (Santosa, 2004).

Tabel 2.
Kategori PING

PING	Packet Loss	Keterangan
< 50 ms	0 %	Hasil Memuaskan
± 90 ms	0 %	Baik
± 150 ms	1 %	Cukup
± 300 ms	3 %	Kategori kurang baik, kesulitan dalam menjalankan aplikasi game atau streaming
> 500 ms	20 %	Jelek

4. Latency

Apabila mengirimkan data sebesar 3MB pada saat jaringan sepi waktunya 5 menit tetapi pada saat ramai sampai 15 menit, hal ini disebut *latency*. *Latency* pada saat jaringan sibuk berkisar 50-70 msec (Santosa, 2004).

5. Losses

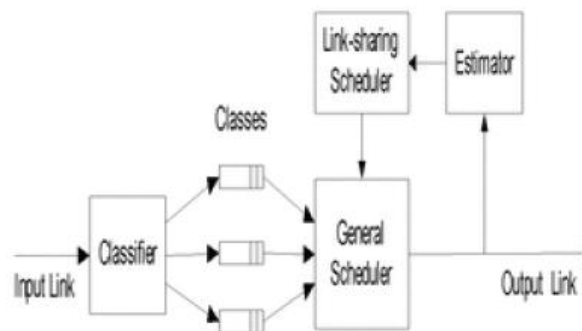
Losses merupakan jumlah paket yang hilang saat pengiriman paket data ke tujuan, kualitas terbaik pada jaringan LAN/WAN jika jumlah losses paling kecil.

6. *Packet Internet Gropher* (PING)

PING merupakan salah satu program yang digunakan untuk menguji komunikasi antar komputer dalam sebuah jaringan melalui protokol TCP/IP. Ping akan mengirimkan *Internet Control Message Protocol* (ICMP) Echo Request messages pada IP Address komputer yang dituju dan meminta respons dari komputer tersebut (Muuss, 1977). PING memiliki kategori ukuran kinerja jaringan berdasarkan ping seperti terlihat pada Tabel 2.

2.2 Hierarchical Token Bucket (HTB)

Hierarchical Token Bucket (HTB) merupakan teknik penjadwalan paket yang digunakan kebanyakan router berbasis Linux, dikembangkan pertama kali oleh Martin Devera (2002). Hierarchical Token Bucket (HTB) merupakan teknik penjadwalan paket yang sering digunakan bagi router-router berbasis Linux, dikembangkan pertama kali oleh Martin Devera (Bunafit, 2005). HTB diklaim menawarkan kemudahan pemakaian dengan teknik peminjaman dan implementasi pembagian trafik yang lebih akurat. Dasar kerja HTB hampir sama dengan disiplin antrian CBQ bahkan diagram blok sistem CBQ dengan HTB tidak ada bedanya, hanya saja pada General Scheduler HTB menggunakan mekanisme Deficit Round Robin (DRR) dan pada blok umpan balik, Estimator, HTB tidak menggunakan Eksponential Weighted Moving Average (EWMA) melainkan Token Bucket Filter (TBF).



Gambar 1. Deficit Round Robin

Pada metode HTB terdapat parameter ceil sehingga kelas akan selalu mendapatkan bandwidth diantara base link dan nilai ceil linknya. Parameter ini dapat dianggap sebagai estimator kedua, sehingga setiap kelas dapat meminjam bandwidth selama bandwidth total yang diperoleh memiliki nilai dibawah nilai ceil. Hal ini mudah diimplementasikan dengan cara tidak mengijinkan proses peminjaman bandwidth pada saat kelas telah melampaui link ini (keduanya *leaves* dan interior dapat memiliki ceil). Apabila nilai ceil sama dengan nilai base link, maka akan memiliki fungsi yang sama seperti parameter bounded pada CBQ, dimana kelas-kelas tidak diijinkan untuk meminjam bandwidth. Sedangkan jika nilai ceil diset tak terbatas atau dengan nilai yang lebih tinggi seperti kecepatan link yang dimiliki, maka akan didapat fungsi yang sama seperti kelas non bounded (Yudha, 2007)

Penjadwalan pengiriman paket antrian, maka HTB menggunakan suatu proses penjadwalan yang dapat dijelaskan sebagai berikut (Devara, 2002) :

- Class, merupakan parameter yang diasosiasikan dengan rate yang dijamin (*assured rate*) AR, ceil rate CR, prioritas P, level dan quantum. Class dapat memiliki parent. Selain AR dan CR, didefinisikan juga *actual rate* atau R, yaitu rate dari aliran paket yang meninggalkan class dan diukur pada suatu perioda waktu tertentu.
- Leaf, merupakan class yang tidak memiliki anak. Hanya leaf yang dapat memegang antrian paket.
- Level, dari kelas menentukan posisi dalam suatu hirarki. Leaf-leaf memiliki level 0, root class memiliki level=jumlah level - 1 dan setiap inner class memiliki level kurang dari satu dari parentnya.
- Mode, dari class merupakan nilai-nilai buatan yang diperhitungkan dari R, AR dan CR. Mode-mode yang mungkin adalah: Merah: $R > CR$; Kuning: $R \leq CR$ and $R > AR$; Hijau selain di atas.

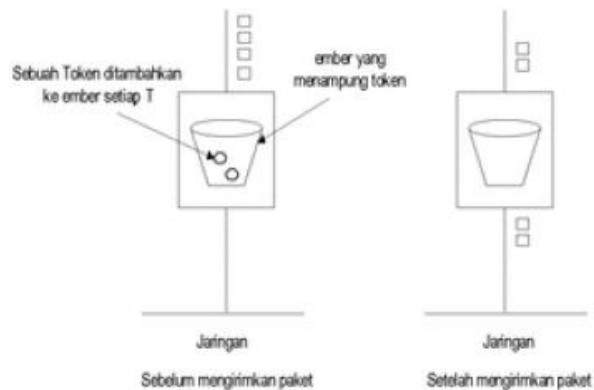
2.2.1 General Scheduler HTB

HTB menganggap hirarki kelas lengkap dan trafik dipisah-pisah menjadi beberapa aliran trafik, algoritma untuk penjadwalan paket adalah sebagai berikut: pertama memilih kelas pada cabang terendah (*leaf class*) yang linknya belum mencapai batas kemudian mulai mengirimkan paket dari kelas yang

memiliki prioritas tertinggi kemudian berlanjut ke yang rendah, apabila link semua kelas melampaui batas link maka dilakukan suatu test melalui suatu putaran lengkap untuk menemukan leaf class yang dapat meminjam bandwidth dari kelas di atasnya (*parent class*) jika tidak ada maka putaran diulangi dengan mencoba meminjam bandwidth dari kelas di atas *parent class* (*grandfather class*).

2.2.2 Estimator

Hierarchical Token Bucket (HTB) menggunakan *Token Bucket Filter* (TBF) sebagai Estimator untuk menentukan apakah suatu kelas/prioritas berada dalam keadaan underlimit, atlimit atau overlimit. TBF bekerja dengan dasar algoritma ember token, setiap paket yang akan dikirimkan harus memiliki token yang berada dalam ember token, jika token tak tersedia didalam ember maka paket-paket yang akan dikirimkan harus menunggu sampai tersedia token yang cukup untuk mengirimkan paket yang sedang menunggu.



Gambar 2. Token Bucket Filter

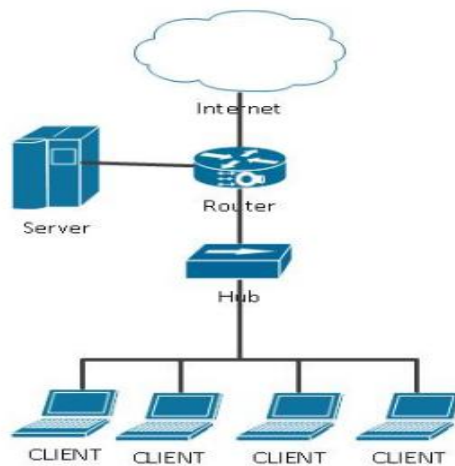
Implementasi TBF terdiri dari sebuah buffer (bucket), yang secara konstan diisi oleh beberapa informasi virtual yang dinamakan token, pada link yang spesifik (token link). Parameter paling penting dari bucket adalah ukurannya, yaitu banyaknya token yang dapat disimpan. Setiap token yang masuk mengumpulkan satu paket yang datang dari antrian data dan kemudian dihapus dari bucket. Dengan menghubungkan algoritma ini dengan dua aliran-token dan data, akan didapati tiga buah kemungkinan skenario:

- Data yang datang pada TBF memiliki link yang sama dengan masuknya token. Dalam hal ini, setiap paket yang masuk memiliki tokennya masing-masing dan akan melewati antrian tanpa adanya delay.
- Data yang datang pada TBF memiliki link yang lebih kecil daripada link token. Hanya sebagian token yang dihapus pada output pada tiap paket data yang dikirim ke antrian, dan token akan menumpuk, memenuhi ukuran bucket. Token yang tidak digunakan kemudian akan dapat digunakan untuk mengirimkan data pada kecepatan yang melampaui link token standar, ini terjadi jika ada ledakan data yang pendek.
- Data yang datang pada TBF memiliki link yang lebih besar daripada link token. Hal ini berarti bucket akan segera kosong dari token, yang menyebabkan TBF akan menutup alirannya untuk sementara. Hal inilah yang dinamakan situasi overlimit. Jika paket-paket tetap datang, maka paket-paket akan segera dibuang.

3. PEMBAHASAN

3.1 Topologi Jaringan

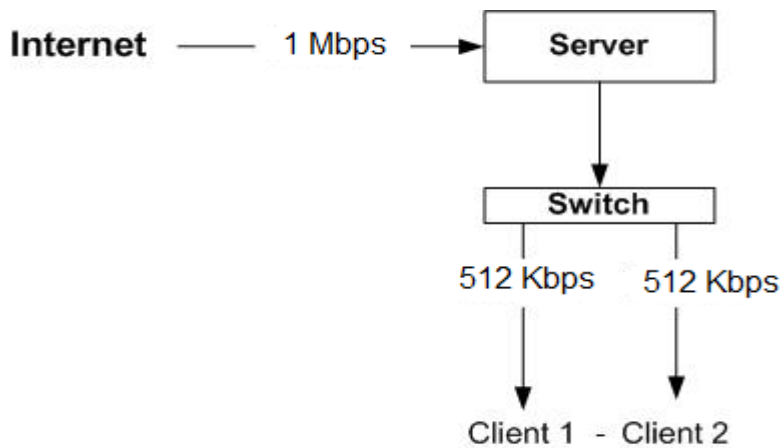
Dalam pembahasan topologi jaringan yang digunakan adalah bentuk topologi star, seperti gambar 3. Masing-masing workstation dihubungkan langsung ke server dengan perantara switch/Hub. Topologi ini memiliki kabel sendiri untuk setiap workstation ke server, maka jalur komunikasi dalam kabel akan semakin lebar sehingga akan meningkatkan kinerja secara keseluruhan. Jika terdapat gangguan di suatu jalur maka gangguan hanya akan terjadi dalam komunikasi antara workstation yang bersangkutan dengan server.



Gambar 3. Model Topologi Jaringan

3.2 Manajemen Bandwith

Manajemen bandwidth ini akan membatasi penggunaan bandwidth jaringan Internet, manajemen dilakukan untuk membagi rata bandwidth per-client agar tidak terjadi congestion, jika sebuah jaringan Internet belum menerapkan manajemen bandwidth maka salah satu client menggunakan bandwidth secara penuh, client-client setelahnya akan mengalami antrian permintaan paket data dan mendapatkan bandwidth ketika permintaan paket data dari client 1 terpenuhi. Hal ini dapat mengganggu client-client lain dan mengganggu kinerja dari jaringan internet itu sendiri seperti terlihat pada Gambar 3.

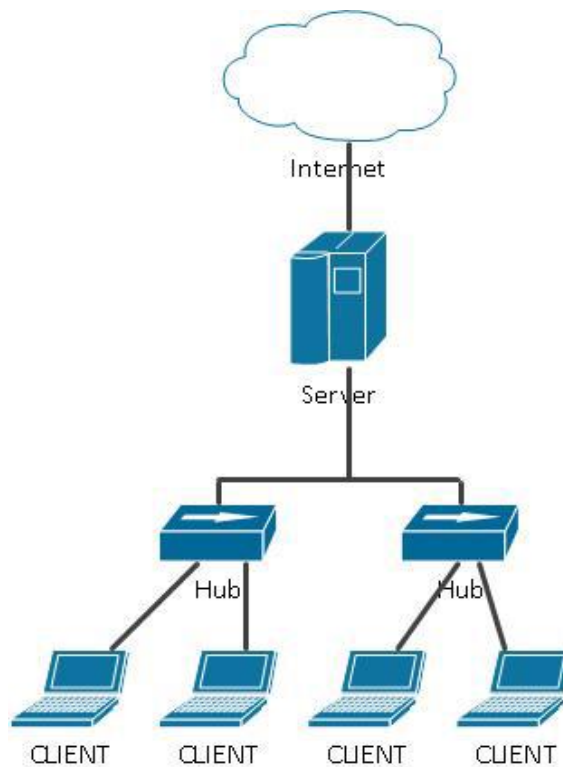


Gambar 4. Proses Manajemen Bandwith

Berdasarkan Gambar 4. maka server akan mendapatkan bandwidth sebesar 128 Kbps, setiap client akan melakukan browsing dan download/upload, client memiliki aturan yang berbeda sehingga tidak terjadi gangguan antara client 1 dan client 2 tetapi dapat meminjam bandwidth jika salah satu client tidak aktif.

3.3 Perancangan Jaringan

Perancangan sistem jaringan merupakan tahap pembuatan rancangan jaringan, dalam penelitian ini menggunakan rancangan seperti Gambar 5, terdapat sebuah server menggunakan sistem operasi Linux Fedora Core 9 kernel 2.6.25-14 dan memiliki 5 client.



Gambar 5. Perancangan Sistem Jaringan

3.4 Perancangan Manajemen Bandwith

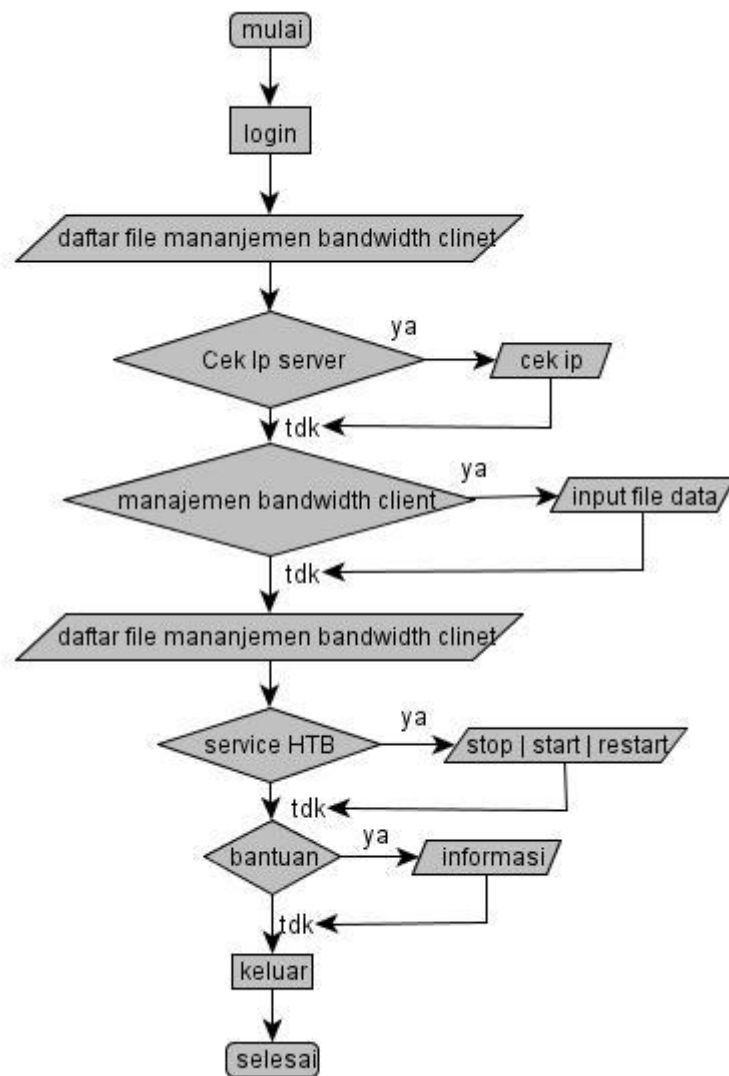
Perancangan manajemen bandwith dilakukan dengan menggunakan bantuan aplikasi yang khusus dikembangkan untuk melakukan hal ini. Program ini merupakan alat bantu bagi administrator untuk mempermudah menjalankan, mengelola atau mengatur bandwidth tiap client yang semula atau awalnya dengan mengetikkan perintah-perintah di konsol untuk mengelola, menjalankan dan mengatur bandwidth. Program ini dibuat dengan menggunakan bahasa pemrograman PHP dan MySQL sebagai database untuk menyimpan file-file konfigurasi bandwidth, untuk selanjutnya program ini akan disebut dengan nama imanbw (Interface Manajemen Bandwidth). Skenario rancangan untuk menentukan pembatasan bandwidth setiap client dapat dilihat seperti pada Tabel 3.

Tabel 3.

Kategori PING

Client	Rate	Ciel
Client 1	256	512
Client 2	256	512
Client 3	256	512
Client 4	256	512
Client 5	256	512

Program aplikasi manajemen bandwith yang diterapkan ini memiliki cara kerja seperti terlihat pada Gambar 6.



Gambar 6. Flowchart Program Aplikasi Manajemen Bandwith

Berdasarkan gambar 6, maka cara penggunaan aplikasi manajemen bandwith dapat dijelaskan sebagai berikut :

- User admin login ke program aplikasi manajemen bandwith.
- Halaman utama akan langsung menuju ke daftar file.
- Halaman utama memiliki beberapa menu diantaranya menu cek IP, jika ya maka akan melihat IP dari server, jika tidak maka dapat dilanjutkan memilih menu selanjutnya.
- Menu manajemen bandwidth, jika ya maka akan melakukan input file data konfigurasi, jika tidak maka akan langsung menuju halaman daftar file.
- Menu service, jika ya maka perintah untuk menjalankan HTB, jika tidak maka dapat menuju menu help untuk mengetahui tata cara pemakaian program aplikasi manajemen bandwith.
- Jika sudah melakukan konfigurasi user admin dapat keluar dari program aplikasi manajemen bandwith atau kembali login lagi untuk melakukan aktifitas yang dibutuhkan.

4. IMPLEMENTASI

4.1 Implementasi Program Aplikasi Manajemen Bandwith

Implementasi program aplikasi manajemen bandwith dapat dilakukan dengan menjalankan servis httpd dan mysql, setelah semuanya aktif maka selanjutnya dapat diakses menggunakan browser dengan alamat <http://localhost/bwcontrol> dari server, bila dari client diketikkan http://ipaddress_server/bwcontrol seperti terlihat pada Gambar 7.



Gambar 7. Pengujian Program Aplikasi Manajemen Bandwith

4.2 Tabel User dan Network

Tabel user diperlukan untuk login pada program aplikasi manajemen dan tabel network untuk file-file konfigurasi HTB agar mudah di edit oleh administrator. Pembuatan tabel user dan network sebagai berikut:

Kode program 1. Create table users

```
CREATE TABLE `users` (
  `idreg` int(10) NOT NULL auto_increment,
  `usr` varchar(30) NOT NULL,
  `pwd` varchar(30) NOT NULL,
  PRIMARY KEY (`idreg`)
) TYPE=MyISAM AUTO_INCREMENT=1 ;
```

Kode program 2. Create table networks

```
CREATE TABLE `networks` (
  `idreg` int(10) NOT NULL auto_increment,
  `nm` varchar(35) NOT NULL,
  `r2q` varchar(255) NOT NULL,
  `rate` varchar(255) NOT NULL,
  `quantum` varchar(255) NOT NULL,
  `leaf` varchar(255) NOT NULL,
  `ceil` varchar(255) NOT NULL,
  `rule` varchar(255) NOT NULL,
  `mark` varchar(255) NOT NULL,
  PRIMARY KEY (`idreg`)
) TYPE=MyISAM AUTO_INCREMENT=2 ;
```

4.3 Konfigurasi Program Aplikasi Manajemen Bandwith

Program aplikasi manajemen bandwith memerlukan konfigurasi penyesuaian dengan path konfigurasi HTB di atas agar dapat dijalankan sesuai fungsinya. Konfigurasi yang pertama adalah tentang koneksi database dengan program aplikasi manajemen bandwith karena fungsi database di sini sebagai penyimpan file-file konfigurasi HTB agar dapat di edit dan dihapus dengan mudah, seperti dibawah ini:

Kode program 3. Koneksi database dengan program aplikasi manajemen bandwith dengan skrip PHP

```
<?
mysql_connect('localhost','root','administrator');
mysql_select_db('bwcontrol');
//menghapus session pesan
unset($_SESSION['mesg']);
?>
```


Koneksi di atas disesuaikan dengan database yang telah dibuat dan tersimpan dengan nama file database.php. Konfigurasi kedua yaitu menyamakan path untuk meletakkan file-file konfigurasi HTB dan eksekusi file htb.init.

Kode program 4. Path direktori HTB

```
<?
include 'function.php';
define("PATH_NETWORKS","/etc/sysconfig/htb/");
?>
```

Konfigurasi di atas menunjukkan tempat file-file konfigurasi HTB disimpan ketika menggunakan imanbw dan tersimpan dengan nama setting.php. Melakukan eksekusi htb.init dengan program aplikasi manajemen bandwidth juga harus disesuaikan dengan path di lokasi htb.init yaitu di /usr/local/sbin tersimpan dengan nama service.php.

Kode program 5. Path eksekusi htb.init

```
<?
session_start();
if($_POST['action'] == 'Start'){
if($output = shell_exec('/usr/local/sbin/htb.init start'))
$_SESSION['mesg'] = 'Service has been started.<br>';
else
$_SESSION['mesg'] = 'Service started failed.<br>';
}
elseif($_POST['action'] == 'Stop'){
if($output = shell_exec('/usr/local/sbin/htb.init stop'))
$_SESSION['mesg'] = 'Service has been stopped.<br>';
else
$_SESSION['mesg'] = 'Service stopped failed.<br>';
}
elseif($_POST['action'] == 'Restart'){
if($output = shell_exec('/usr/local/sbin/htb.init restart'))
$_SESSION['mesg'] = 'Service has been restarted.<br>';
else
$_SESSION['mesg'] = 'Service restarted failed.<br>';
}
$_SESSION['mesg'] .= "<pre>$output</pre>";
header("location:index.php?menu=service");
?>
```

4. IMPLEMENTASI

4.1 Implementasi Program Aplikasi Manajemen Bandwith

Pengujian dilakukan terhadap bandwidth ketika sebelum menerapkan HTB dan sesudah menerapkan HTB. Pengujian ini dilakukan untuk mengetahui apakah metode HTB dalam manajemen bandwidth dapat berjalan dengan baik dan lancar. Dalam pengujian ini dilakukan dengan metode Comparison Test. Comparison test merupakan pengujian yang membandingkan hasil dari aturan yang telah ditentukan untuk mendapatkan data yang identik dengan aturan - aturan yang telah diberikan, sehingga dapat dilihat perbedaannya. Pengujian dilakukan menggunakan iperf dan ping, dengan sintaks perintah: Seluruh client melakukan aktifitasnya, server memberikan perintah iperf -c ip_client -t 10 -i 1 ke setiap client-nya maka akan mendapatkan nilai intervals, transfer, rate.

Tabel 3.

Upload 128K dan Download =64K

IP Address	Download			Upload		
	Ukuran File	Max	Min	Ukuran File	Max	Min
192.168.1.10	700 Mb	6.0 Mbps	4.5 Mbps	700 Mb	8.2 Mbps	5.4 Mbps
	200 Mb	6.0 Mbps	4.5 Mbps	200 Mb	7.5 Mbps	6.8 Mbps
	1.5 MB	100 Kbps	10 Kbps	300 K	80 Kbps	10 Kbps

Tabel 4.
Upload 64K dan Download =128K

IP Address	Download			Upload		
	Ukuran File	Max	Min	Ukuran File	Max	Min
192.168.1.10	7.50 MB	90.5 Kbps	10.2 Mbps	7.50 MB	2.2 Mbps	1.4 Mbps
	30 KB	83.4 Kbps	10.5 Mbps	30 KB	5.5 Mbps	3.8 Mbps
	2.5 MB	76.5 Kbps	10.7 Kbps	250 K	65 Kbps	12 Kbps

5. KESIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan maka dapat diambil kesimpulan bahwa, HTB dapat melakukan pembatasan bandwidth dengan baik dari seluruh client yang ada dan program imanbw yang dikembangkan dapat membantu memudahkan administrator dalam melakukan manajemen bandwidth di lingkungan jaringan yang dikelolanya.

DAFTAR PUSTAKA

- [1] Bunafit Nugroho. 2005. "Instalasi & Konfigurasi Jaringan Windows & Linux ". Yogyakarta. Andi Yogyakarta.
- [2] Balan, Doru G, Alin P(2009). Extended Linux HTB Queuing Discipline Implementations. International Journal Of Information Studies, University Of Suceava, Romania.
- [3] Devara, M., 2002, *Hierarchical Token Bucket Theory*, <http://luxik.cdi.cz/~devik/qos/htb/manual/theory.htm>. Diakses 25 Maret 2015.
- [4] Ferguson, P. & Huston, G., 1998, *Quality of Service*, John Wiley & Sons Inc.
- [5] Muuss, M., 1977, *The Story of PING program*, <http://ftp.arl.mil/~mike/ping.html>. Diakses 18 Maret 2015.
- [6] Riadi, Imam & Prio, Wahyu. 2011. *Implementasi Quality of Service Menggunakan Metode Hierarchical Token Bucket*. Jurnal Sistem Informasi, Vol. 1, No. 2 ISSN 2087-8737. Yogyakarta : Universitas Ahmad Dahlan.
- [7] Santosa, B., 2004, *Manajemen bandwidth internet dan intranet*, http://stream.plasa.com/onno/gfe/view.php?file=referensi_bahasa_indonesia_2/network/bwmanagement.pdf. Diakses 22 Maret 2015.
- [8] Stallings, william.2001 *Data & Computer Communications* (terjemahan). Jakarta: Salemba Teknika
- [9] Yudha, 2007, *Disiplin Antrian (Queueing Discipline/Qdisc)*, <http://omyudha.multiply.com/journal/item/9>. Diakses 22 Maret 2015.