

MANAJEMEN BANDWIDTH DENGAN METODA HIERARCHICAL TOKEN BUCKET BERBASIS MIKROTIK DAN MONITORING BANDWIDTH MENGGUNAKAN JAVA

Wahyudi

Program Studi Sistem Informasi, STMIK Indonesia Padang
e-mail : wahyudisy@gmail.com

ABSTRACT

Bandwidth is one thing that is very important in a network. Bandwidth is the amount that shows how much data can be passed to a network connection. Along with the development of technology that will need more and more internet access as well as the available bandwidth, the greater bandwidth available then the better internet access is also available in a network. A lot of internet activity, such as downloading, playing online games or uploading would interfere with rapid access to the other client, it is necessary for a bandwidth management that every client get the same bandwidth. To optimize bandwidth management that is then needed a bandwidth monitoring application that will display the data in and data out of the computer.

Keywords: Bandwidth; Bandwidth Management; Bandwidth Monitoring

1. PENDAHULUAN

Dalam perkembangan teknologi yang sangat pesat sekarang ini, mikrotik menjadi alat yang berperan penting dalam dunia jaringan komputer. Mikrotik dapat mengatur hal-hal yang berhubungan dengan jaringan yang akan dibangun, mulai dari *proxy*, *firewall* hingga manajemen *bandwidth*. Sebuah mikrotik dapat berupa software maupun hardware, mikrotik merupakan sistem operasi *Linux base* yang diperuntukkan sebagai *network router*. Didesain untuk memberikan kemudahan bagi penggunanya. Kebutuhan akan akses internet dewasa ini sangat tinggi. Baik untuk mencari informasi, artikel, maupun pengetahuan terbaru. Pem-bagian *bandwidth* atau dengan kata lain mengontrol *bandwidth* pada setiap *host* adalah satu bagian yang penting untuk dilakukan seorang administrator. Pada banyak *spot/tempat* yang menyediakan fasilitas *wireless*, masalah jaringan yang terjadi sebagian besar dikarenakan banyak user yang banyak melaku-kan aktifitas *download*, sehingga mengganggu aktivitas internet user yang lain. Hal ini disebabkan karena jalur ke internet dipenuhi oleh *traffic* lalu lintas *download* tersebut. Dengan pengontrolan dan monitoring, seorang admin jaringan dapat mengatur *bandwidth* dan melihat *traffic* jaringan yang ditangani.

HTB (*Hierarcichal Token Bucket*) merupakan teknik manajemen *bandwidth* yang memberikan fasilitas pembatasan trafik pada setiap *level* maupun klasifikasi, *bandwidth* yang tidak terpakai bisa digunakan oleh klasifikasi yang lebih rendah. Kita juga dapat melihat HTB seperti suatu struktur organisasi dimana pada setiap bagian memiliki wewenang dan mampu membantu bagian lain yang memerlukan. Disamping itu adanya monitoring yang akan digunakan untuk memonitoring *bandwidth* yang masuk dan keluar dalam sebuah jaringan yang telah dibuat. Mengelola jaringan tanpa memonitor layaknya mengemu-di kendaraan tanpa menggunakan *speedo-menter* atau laju mobil. Bagaimana anda tahu seberapa cepat mobil itu melaju, apakah mobil memakan bahan bakar seefisien yang dijanji-kan penjual. Alternatif aplikasi yang digunakan untuk melakukan pengawasan jaringan terutama untuk penggunaan *bandwidth* yaitu *Bandwidth Monitor*. Aplikasi ini dijalankan di dalam komputer server yang terkoneksi ke internet, aplikasi ini dapat melacak semua lalu lintas di koneksi jaringan dan menampilkan besarnya kecepatan *download* dan *upload* dalam berupa grafik.

PT. VIP Andalas Perkasa adalah perusa-haan yang bergerak dibidang usaha pelayanan penyediaan jasa, media, serta penyelenggaraan seminar, dan *workshop* bisnis. Melalui jaring-an kemitraannya, PT VIP Andalas Perkasa pun mengembangkan pariwisata, jasa, dan menjadi wadah pendukung bagi komunitas-komunitas khususnya di Sumatera Barat. Untuk menun-jang kegiatan di perusahaan tersebut maka dibutuhkan sebuah koneksi yang bagus, tetapi karena perusahaan tersebut masih dalam tahap pengembangan sehingga untuk sistem jaringan masih dibawah standar, dimana para pegawai dapat melakukan aksi-aksi yang dapat membebani jaringan tersebut, sehingga untuk pemecahan masalah tersebut maka dibangun sebuah sistem manajemen *bandwidth* dimana sistem tersebut berfungsi untuk mengatur atau mengalokasi-kan *bandwidth* dengan baik dan bermanfaat.

2. KAJIAN LITERATUR

Jaringan Komputer

Jaringan komputer dapat diartikan sebagai sekelompok komputer yang dihubungkan menggunakan media tertentu sehingga antar komputer dapat saling berhubungan untuk berbagi data, informasi, program aplikasi, dan perangkat keras, seperti *printer*, *scanner*, CD/DVD Drive, ataupun *hardisk* (Madcoms 2013:1). Sedangkan menurut Melwin Syafrizal (2005:2) bahwa “jaringan komputer adalah himpunan “interkoneksi” antara komputer autonomus atau lebih yang terhubung dengan media transmisi kabel atau tanpa kabel (*wireless*)”.

Jaringan komputer adalah dua atau lebih komputer yang saling terhubung dengan menggunakan media penghantar kabel atau tanpa kabel untuk berbagi data dan informasi.

Jenis Jaringan Komputer Berdasarkan Area

Pada dasarnya jaringan komputer dapat dibedakan berdasarkan luas area yang dapat dijangkau oleh jaringan itu sendiri. Hal ini berarti luas area dapat ditentukan berdasarkan jarak/jangkauan dari jaringan itu sendiri. Secara umum jaringan komputer terbagi menjadi 3 jenis, yaitu :

1. Local Area Network (LAN)

Local Area Network (LAN) adalah jaringan yang dibatasi oleh area yang relatif kecil, seperti sebuah perkantoran disebuah gedung, atau sebuah sekolah, dan biasanya sekitar 1 kilometer persegi. “Jaringan LAN adalah jaringan yang menghubungkan beberapa komputer dalam suatu local area (biasanya dalam satu gedung atau antar gedung)”.

Jaringan LAN adalah jaringan komputer yang radiusnya lebih kecil yaitu sekitar 200 m antarnode dengan kecepatan sampai dengan 100 Mbps. Penerapan jaringan ini biasanya pada perkantoran dengan skala kecil, warnet, ataupun kampus.

2. Metropolitan Area Network (MAN)

MAN dapat diartikan sebagai arsitektur jaringan yang di dalamnya terdapat dua atau lebih jaringan LAN yang dihubungkan menjadi satu. “MAN adalah jaringan dengan area operasi lebih besar dari LAN tetapi lebih kecil dari WAN, seperti sebuah kota dengan performa hardware dan kapasitas data yang tinggi”. MAN adalah sebuah jaringan komputer yang menghubungkan beberapa jaringan LAN, dan cakupannya meliputi suatu kota.

3. Wide Area Network (WAN)

Wide Area Network (WAN) adalah sebuah jaringan yang memiliki jarak yang sangat luas, karena radiusnya mencakup sebuah negara dan benua”. “*Wide Area Network (WAN)* adalah jaringan yang biasanya sudah menggunakan media *wireless*, sarana satelit, ataupun kabel serat *optic*, karena jangkauannya yang lebih luas, bukan hanya meliputi satu kota atau antar kota dalam suatu wilayah, tetapi mulai menjangkau area atau wilayah otoritas negara lain”. WAN adalah suatu jaringan yang menghubungkan beberapa jaringan LAN dan MAN sehingga membentuk suatu jaringan yang besar.

Berdasarkan Media Penghantar

Berdasarkan media penghantar, jaringan komputer dapat dibagi menjadi 2 jenis, yaitu: 1) *Wire Network*: adalah jaringan komputer yang menggunakan kabel sebagai media penghantar. 2) *Wireless Network*: adalah jaringan yang menggunakan media penghantar gelombang radio atau cahaya infrared.

Berdasarkan Fungsi

Berdasarkan pola pengoperasian atau fungsi masing-masing komputer maka jaringan komputer dapat dibagi menjadi 2 jenis, yaitu:

1. Peer to Peer

Peer dapat diartikan sebagai rekan kerja. *Peer to peer* adalah jaringan komputer yang terdiri atas beberapa komputer (biasanya tidak lebih dari 10 PC dengan dilengkapi 1 atau 2 buah printer). Model jaringan ini mengutamakan pada aspek penggunaan program, data, dan printer secara bersama-sama. Dalam model *peer to peer*, setiap *host* memberikan layanan ke *peer* lain atau mengambil layanan dari *peer* lain. “*Peer to Peer* adalah jenis jaringan komputer di mana setiap komputer bisa menjadi *server* sekaligus *client*”. *Peer to Peer* adalah jaringan komputer yang terdiri atas beberapa komputer (tidak lebih dari 10 komputer) dimana setiap komputer dapat berperan sebagai *server* dan *client* sekaligus.

2. Client Server

Client Server adalah jaringan komputer yang salah satu (boleh lebih) komputernya difungsikan sebagai *server* untuk melayani komputer lain. Komputer yang dilayani oleh *server* disebut *client*. Layanan yang diberikan bisa berupa akses *Web*, *e-mail*, *file*, atau yang lain. Model hubungan *client*

server memisahkan antara *server* dan *client*. *Server* memberikan layanan jaringan dan *client* menerima layanan”.

Client Server adalah jaringan komputer yang terdiri dari beberapa komputer, dimana salah satu komputer ditunjuk sebagai *server* yang memberikan layanan ke pada *client*.

Model OSI

OSI adalah singkatan dari *Open Systems Interconnection*. OSI dikembangkan oleh ISO (*International Organization for Standardization*). Dalam model struktur protokol OSI, protokol dibagi kedalam tujuh lapis layanan. ISO telah menspesifikasikan suatu model referensi OSI yang membagi-bagi fungsi komunikasi ke dalam tujuh lapis (*layer*)”. Berikut adalah masing-masing tugas dari tiap layer:

1. *Layer Physical*

Lapisan ini bertanggung jawab melakukan translasi secara fisik dari informasi yang terkandung di dalam paket menjadi jalur sinyal secara *actual*, sebagai contoh, bit 0 dan 1 dapat berarti tegangan negatif/ positif atau tegangan rendah/ tinggi. Lapis ini tidak menambahkan informasi apapun kedalam paket yang diperoleh dari lapis di atasnya.

2. *Layer Data-Link*

Lapis *data Link layer* bertanggung jawab dalam 2 hal, yaitu: Memberikan petunjuk kepada paket dalam melewati link dalam jaringan dan memberikan *frame* pada paket yang dikirimkan, yaitu dengan menambahkan alamat fisik tujuan ke dalam paket.

3. *Layer Network*

Lapis *network layer* bertanggung jawab terhadap pengiriman paket-paket (pada lapis yang lebih rendah) dalam dua hal, yaitu: Menambahkan alamat jaringan dan informasi lainnya kedalam paket yang dikirimkan serta membuat keputusan rute yang harus dilalui oleh paket yang ditransmisikan melewati banyak jaringan.

4. *Layer Transport*

Setiap data/ informasi yang dikirim melalui media komunikasi dalam jaringan akan diubah kedalam bentuk unit-unit yang dapat dikelola yang disebut sebagai paket (*packet*). Lapis *transport layer* bertanggung jawab untuk membuat paket-paket tersebut yang memuat data, alamat, urutan, serta mekanisme kontrol kesalahan (*error control*) terhadap data-data/informasi yang dikomunikasikan.

5. *Layer Session*

Session layer bertanggung jawab memberikan 2 macam layanan, yaitu: Mengelola proses komunikasi dua arah, misal *session* komunikasi. Sebagai contoh: ketika seseorang mengambil uang dari mesin ATM, berarti orang tersebut telah berpartisipasi dalam sebuah *session* serta memberikan layanan sinkronisasi

6. *Layer Presentation*

Lapis *presentation layer* bertanggung jawab memberikan 2 macam layanan, yaitu: 1) *Translasi* diperlukan karena sistem pengkodean pada setiap komputer para pemakai bersifat spesifik (berbeda-beda) sehingga perlu translasi menjadi kode dalam standar internasional. 2) Proses *enkripsi* dan kompresi data Lapisan *Presentation layer* juga bertanggung jawab terhadap enkripsi dan kompresi data, meskipun juga akan ditangani oleh lapis lainnya.

7. *Layer Application*

Lapisan *application layer* bertanggung jawab memberikan layanan-layanan aplikasi bagi para pemakai akhir (*end user*), misal aplikasi FTP atau SMTP (e-mail).

Mikrotik

Mikrotik adalah perusahaan kecil berkantor pusat di Latvia, yang dibentuk oleh John Trully dan Arnis Riekstins. Tahun 1996 John dan Arnis memulai dengan system Linux dan MS DOS yang dikombinasikan dengan teknologi *Wireless LAN* (W-LAN) *Aeronet* berkecepatan 2Mbps di Moldova. Barulah kemudian melayani lima pelanggannya di Latvia, karena ambisi mereka adalah membuat satu peranti lunak *router* yang handal dan disebar ke seluruh dunia. Prinsip dasar MikroTik bukan membuat *Wireless ISP* (WISP), tapi membuat program *router* yang handal dan dapat dijalankan di seluruh dunia. Hingga kini, MikroTik telah melayani sekitar empat ratusan pelanggannya.

Linux yang mereka gunakan pertama kali adalah Kernel 2.2 yang dikembangkan secara bersama-sama dengan bantuan 5-15 orang staf R&D MikroTik yang sekarang menguasai dunia *routing* di Negara-negara berkembang. Selain staf di lingkungan *Mikrotik*, menurut arnis, mereka merekrut juga tenaga-tenaga lepas dan pihak ketiga yang dengan intensif mengembangkan *mikrotik* secara maraton.

Berdasarkan bentuk hardware yang di gunakan, *mikrotik* dapat digolongkan dalam dua jenis. Untuk memudahkan bagi pemula dalam memahami *Router Operating Sistem* ini. Dua jenis tersebut adalah :

Mikrotik RouterOS™

Mikrotik RouterOS™ adalah sistem operasi dan perangkat lunak yang dapat digunakan untuk menjadikan komputer biasa menjadi *router network* yang handal, mencakup berbagai fitur yang dibuat untuk ip network dan jaringan *wireless*. Fitur-fitur tersebut diantaranya : *Firewall & Nat, Routing, Hotspot, Point to Point Tunneling Protocol, DNS server, DHCP Server*, dan masih banyak lagi fitur lainnya. Ini adalah versi MikroTik dalam bentuk perangkat lunak yang dapat diinstal pada *Personal Computer (PC)* melalui CD.

Build in Hardware Mikrotik

Merupakan MikroTik dalam bentuk perangkat keras yang khusus dikemas dalam *board router*, atau sering disebut *routerBoard*, yang di dalamnya sudah terinstal sistem operasi *MikroTik RouterOS*. Untuk versi ini, lisensi sudah termasuk dalam board MikroTik. Pada *Router board* ini pengguna langsung dapat memakainya, tanpa harus melakukan instalasi sistem operasi. *Router Board* ini dikemas dalam beberapa bentuk dan kelengkapannya sendiri sendiri. Ada yang difungsikan sebagai *Indoor Router, Outdoor Router* maupun ada yang dilengkapi dengan *wireless router*.

Manajemen Bandwidth

Bandwidth adalah luas atau lebar cakupan frekuensi yang digunakan oleh sinyal dalam medium transmisi. Dalam kerangka ini, *Bandwidth* dapat diartikan sebagai perbedaan antara komponen sinyal frekuensi tinggi dan sinyal frekuensi rendah. Frekuensi sinyal diukur dalam satuan Hertz. Sinyal suara tipikal mempunyai *Bandwidth* sekitar 3 kHz, analog TV broadcast (TV) mempunyai *bandwidth* sekitar 6 MHz.

Bandwidth dalam ilmu komputer adalah suatu penghitungan konsumsi data yang tersedia pada suatu telekomunikasi. Dihitung dalam satuan *bits per seconds* (bit per detik). *Bandwidth* yang tertera komunikasi nirkabel, modem transmisi data, komunikasi digital, elektronik, adalah *bandwidth* yang mengacu pada sinyal analog yang diukur dalam satuan hertz.

Dalam dunia *web hosting*, *bandwidth capacity* (kapasitas lebarpita) sebagai nilai maksimum besaran transfer data (tulisan, gambar, video, suara, dan lainnya) yang terjadi antara sever hosting dengan komputer klien dalam suatu periode tertentu.

Teknik Class Based Queueing

Teknik klasifikasi paket data yang paling terkenal adalah CBQ. Keunggulan dalam penggunaan CBQ adalah mudah dikonfigurasi, memungkinkan *sharing band-width* antar kelas (*class*) dan memiliki fasilitas *user interface*. CBQ mengatur pemakaian *bandwidth* jaringan yang dialokasikan untuk tiap user, dan pemakaian *bandwidth* yang melebihi nilai set akan dipotong (*shaping*). CBQ juga dapat diatur untuk *sharing* dan meminjam *bandwidth* antar *class* jika diperlukan.

Class Based Queueing (CBQ) adalah suatu mekanisme penjadwalan, bertujuan menyediakan *link sharing* antar kelas yang menggunakan jalur fisik yang sama, sebagai acuan untuk membedakan *trafik* yang memiliki prioritas-prioritas yang berlainan. Dengan CBQ, setiap kelas dapat mengalokasikan *bandwidth* miliknya untuk berbagai jenis trafik yang berbeda, sesuai dengan pembagiannya yang tepat untuk masing-masing trafik.

Teknik Hierarchical Token Bucket

Hierarchical Token Bucket (HTB) dikembangkan oleh Martin Devera pada tahun 2001 yang digunakan untuk membatasi akses menuju ke port/IP tertentu tanpa mengganggu trafik *bandwidth* pengguna lain. Aplikasi ini berfungsi sebagai pengganti aplikasi yang masih sering digunakan, yaitu CBQ. HTB diklaim mampu melakukan pembagian trafik yang lebih akurat.

Teknik antrian HTB mirip dengan teknik pada CBQ. Hanya perbedaannya terletak pada opsi, dimana pada HTB opsi yang digunakan jauh lebih sedikit dalam konfigurasinya, serta lebih presisi dalam penggunaannya. Teknik antrian HTB memberikan fasilitas pembatasan trafik pada setiap level ataupun klasifikasinya, sehingga *bandwidth* yang tidak terpakai dapat digunakan oleh klasifikasi lain yang lebih rendah

Karakteristik Java

Pemrograman *java* merupakan pemrograman yang berorientasi objek. Artinya, penulisan program harus dibuat dalam bentuk objek-objek, kemudian memodelkan sifat masing-masing objek dalam program. *Java* mengatur dan menentukan interaksi objek.

Program java sesuai dengan perkembangan teknologi saat ini, yaitu teknologi jaringan lokal seperti jaringan internet. Dalam hal ini, java dapat digunakan untuk menyebarkan data atau informasi dalam *workstation client, e-mail server, database server, dan webserver*.

Program Java yang sudah diterjemahkan (dikompilasi) dengan benar dan berhasil dapat dijalankan pada *platform* yang lain atau pada sistem operasi yang lain. Java menggunakan prinsip WORA (*Write Once and Run Anywhere*). Sekali menuliskan program maka hasil programnya dapat dijalankan dimana saja.

Thread merupakan unit dasar penggunaan *Central Processing Unit (CPU)* yang terdiri dari *Thread_ID, program counter, register set, dan stack*. *Multithread* adalah banyaknya pekerjaan yang dikerjakan dalam satuan waktu yang bersamaan. Sebagai contoh, suatu *web server* mempunyai ratusan *client server*, beberapa *client* yang mengakses *web server* secara bersamaan. Jika *web server* hanya mempunyai *thread*, berarti *client* yang harus menunggu. Oleh karena itu diperlukan *multithread* untuk melayani *client server*.

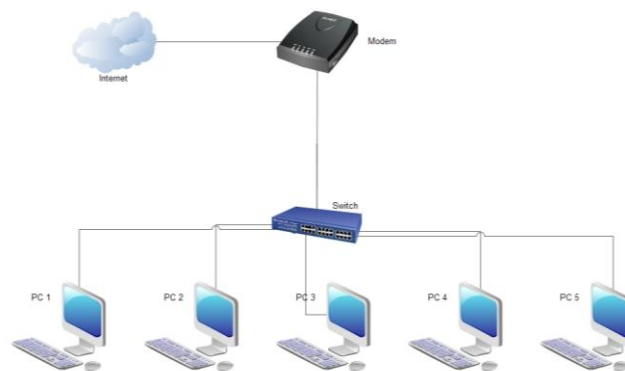
4. METODE PENELITIAN

Dalam penelitian ini dilakukan Penelitian Lapangan (*field research*) dengan melakukan peninjauan langsung ke lapangan. Pengumpulan datanya dengan teknik wawancara dan pemberian daftar pertanyaan (*questioner*) kepada bagian yang terkait dengan penelitian. Penelitian Pustaka (*library research*) dilakukan dengan banyak dibantu buku-buku dan literatur yang berhubungan dengan pemecahan masalah dalam menunjang penyelesaian penelitian. Penelitian Laboratori-um (*laboratory research*) dilakukan dalam memecahkan masalah yang ada maka dalam penelitian ini.

ANALISIS SISTEM

Analisis Dari Sistem Yang Sedang Berjalan

Jaringan internet yang berjalan di PT. VIP Andalas Perkasa merupakan layanan yang didapatkan dari Telkom. Kemudian akses diteruskan ke modem dan dari modem dihubungkan ke *switch*, barulah dari *switch* dihubungkan ke komputer-komputer dan hub untuk penyambung koneksi dari lantai dua. Berikut gambaran Topologi Jaringan di PT. VIP Andalas Perkasa seperti yang terlihat pada Gambar 1 :



Gambar 1: Topologi Jaringan PT. VIP Andalas Perkasa

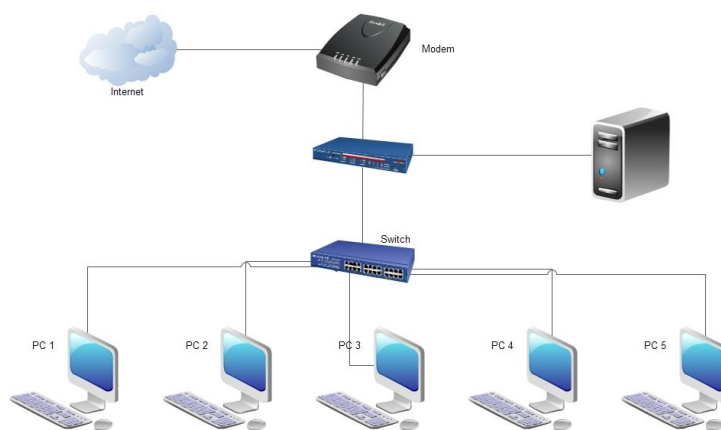
Dari gambar 1 diatas dapat dilihat topologi yang di pakai adalah topologi star, karena dilihat dari fungsi switch yang membagi jaringan ke beberapa komputer sehingga antara komputer dapat saling berhubungan dan komputer juga dapat mengakses internet. Dari topologi di atas juga dapat kita lihat bahwa jaringan tersebut tidak terkontrol dengan baik, sehingga bisa mengakibatkan trafik atau bandwidth menjadi mubazir atau di gunakan tidak tepat. Lalu efeknya akan mengganggu pengguna jaringan yang lain. Contoh : jika satu user dari jaringan tersebut mengakses sebuah video atau mendownload sebuah film, maka semua *bandwidth* yang terdapat pada jaringan tersebut akan langsung beralih ke user yang mendownload tersebut, sehingga user yang lain akan kekurangan *bandwidth* dan membuat pekerjaannya terganggu.

Analisis Dari Sistem Yang Diusulkan

Berdasarkan analisa jaringan yang berjalan pada PT. VIP Andalas Perkasa, maka akan ditambah sebuah *router* untuk mengatur *bandwidth* dari jaringan tersebut, dan juga akan di bangun sebuah server

dimana fungsi server tersebut adalah memantau atau memonitoring pemakaian *bandwidth* dari jaringan tersebut.

Router yang digunakan adalah mikrotik RB750 dimana *router* tersebut sangat efisien dan murah dalam konfigurasi karena sudah memakai GUI. Untuk *server monitoring* sendiri maka di butuhkan sebuah komputer/PC dengan sistem operasi yang digunakan adalah Linux Ubuntu 10.10, dimana *server* tersebut akan dihubungkan ke *mikrotik* RB750 menggunakan kabel UTP dan RJ 45 sebagai konektor. *Server* yang akan dibangun menggunakan sebuah pc dengan sistem operasi yang digunakan Linux Ubuntu 10.10. Server ini akan dihubungkan ke mikrotik RB750 menggunakan kabel UTP dan RJ 45 sebagai konektor. Pengembangan jaringan ini dapat terlihat pada Gambar 2.



Gambar 2: Sistem yang diusulkan

Berdasarkan gambar 2 dapat dilihat bahwa router ditambahkan sesudah modem dari Telkom baru setelah itu di sambungkan ke switch. Port yang tersedia pada router mikrotik ada 5 port, dimana masing-masing port akan digunakan sebagai berikut : *Port 1* akan digunakan untuk menyambungkan modem ke *router*. *Port 2* akan digunakan untuk menyambungkan *router* ke *switch*. *Port 3* digunakan untuk menyambungkan *server monitoring bandwidth* ke *router*.

Berdasarkan keterangan diatas dapat disimpulkan bahwa *port* nomor 3 difungsikan sebagai *server*, pada *server* tersebut akan ditanamkan aplikasi berbasis java yang akan membaca data yang masuk dan data yang keluar melalui jaringan. Data tersebut berupa besarnya *bandwidth* yang digunakan, data yang didapat nantinya akan ditampilkan kepada user berupa grafik, yang akan menampilkan sent data dan receive data.

Kemudian untuk memaksimalkan penggunaan *bandwidth* pada jaringan yang dikelola maka dibutuhkan suatu manajemen *bandwidth* pada mikrotik, manajemen ini digunakan agar setiap *client* yang terhubung ke internet akan mendapatkan besar *bandwidth* yang sama. Teknik yang dilakukan adalah *Hierarchical Token Bucket*, di *HTB bandwidth* yang tersedia untuk suatu kelas ditentukan dan melakukan penandaan untuk penggunaan *bandwidth* dengan jumlah maksimum dan jumlah minimum. Kedua parameter tersebut bekerja pada suatu kelas (*parent*).

Setiap *child* harus memiliki satu buah *parent*, dimana *parent* lah nantinya yang akan meminta *bandwidth*, barulah akan di bagi ke *child-child* yang besarnya di tentukan oleh admin jaringan. Contoh : pada sebuah kantor terdapat jaringan dengan *bandwidth* sebesar 2 Mb, maka untuk dapat membagi *bandwidth* tersebut ke masing-masing *client* dibuatlah *parent* terlebih dahulu, *parent* di setting besaran (*max-limit*) nya sebesar 2 Mb, setelah itu barulah *child-child* di bagi sesuai permintaan *client* atau admin jaringan.

PERANCANGAN SISTEM

Hardware yang digunakan

Dalam pelaksanaan manajemen *bandwidth* dan perancangan aplikasi untuk memonitoring *bandwidth* dibutuhkan sebuah komputer, mikrotik, dan modem sehingga sebuah mikrotik dapat melayani *bandwidth* setiap *client*. Dalam hal ini perangkat keras yang digunakan adalah :

Komputer

Komputer merupakan bagian terpenting dalam hal mengkonfigurasi dan membangun aplikasi manajemen *bandwidth*, karena komputer yang digunakan sebagai server dan sebagai *client*. Berikut spesifikasinya.

Komputer server:Prosesor intel Pentium® Dual Core E5400 @2.70GHz; Motherboard Msi; RAM DDR2 2 GB; VGA on board; Hardisk SATA 250GB; DVD Room; Sistem operasi Linux Ubuntu 10.04; JDK versi 7; Netbeans versi 8

Client:Prosesor Dual Core E5400 @2.70GHz; Hardisk sata 250 GB; RAM 1 GB; Sistem Operasi Win 7

Router mikrotik: Mikrotik routerOS adalah sistem operasi dan perangkat lunak yang dapat digunakan untuk menjadikan komputer biasa menjadi router network yang handal, mencakup berbagai fitur yang dibuat untuk ip network dan jaringan wireless. Spesifikasi mikrotik yang digunakan adalah :Kode produk RB750; Memory 32 GB; Operating System routerOS; 5 buah Port Ethernet

Sistem Operasi yang Digunakan: Sistem operasi yang digunakan adalah Linux Ubuntu 10.10, karena sistem operasi ini memberikan tampilan yang menarik kepada user, mudah digunakan dan selain itu Linux Ubuntu 10.10 adalah sistem operasi *opensource* jadi OS dapat diupgrade kapan saja.

Manajemen Bandwidth

Untuk melakukan pembatasan pada *bandwidth* yang digunakan setiap *client* maka dibutuhkan sebuah manajemen *bandwidth*, dalam melakukan manajemen *bandwidth* digunakan teknik *Hierarchical Token Bucket* berbasis mikrotik yang memungkinkan seorang *admin* jaringan melakukan pengelompokan dalam membuat sebuah *rule queue*.

Pembuatan Mark Packet

Sebelum membuat *queue* pada *rule queue* nya, terlebih dahulu harus menandai paket yang akan keluar dan yang akan masuk pada suatu *client*. Penandaan ini dibuat pada *mangle firewall*.

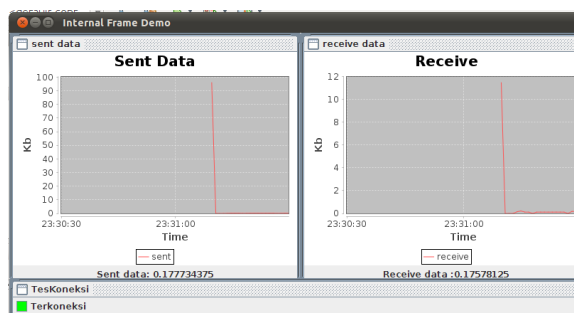
Pembuatan Rule Queue

Queue merupakan antrian yang akan dibuat pada menu *queue* di mikrotik, untuk teknik *Hierarchical Token Bucket*, jenis antrian yang digunakan adalah *queue tree*. Sebelum membuat *queue*, terlebih dahulu yang harus di buat adalah *parent*, karena HTB akan berjalalah jika rule queue berada di bawah setidaknya 1 level *parent*. Untuk *parent max-limit* dapat di isi sebesar *bandwidth* yang tersedia pada jaringan tersebut, dan barulah untuk *rule queue* pembagian *bandwidth* dilakukan.

Aplikasi Monitoring Bandwidth GUI (Graphic User Interface)

Graphical User Interface atau GUI adalah suatu aplikasi java yang berbasis grafik. GUI membuat hasil suatu program tampak menarik dan nyaman ketika digunakan oleh para *user*.

Untuk itu pada perancangan *monitoring bandwidth* ini ada 3 buah *frame* yang akan dibuat yaitu proses masuk (*download*), keluar (*upload*), dan indikator koneksi *server* dengan internet. Terlihat seperti gambar 3



Gambar 3. Form GUI *Monitoring Bandwidth*

Indikator yang Digunakan dalam Pengukuran *Bandwidth*

Pengukuran *Bandwidth*

Untuk melakukan pengukuran *bandwidth* ada dua indikator yang akan dihitung diantaranya, data yang masuk dan data yang keluar dari *client*. Berikut cara menghitung *bandwidth*:

Data Masuk (*Download*). Untuk menghitung data yang masuk ke *client* (*download*) adalah : $Download = \text{data}/\text{sekon}$. Tampilan data yang keluar berupa grafik, data yang masuk akan diperbarui setiap 2 detik, sehingga akan terlihat perubahan data yang masuk dalam bentuk grafik.

Data Keluar (*Upload*). Untuk menghitung data yang keluar adalah: $Upload = \text{data keluar}/\text{sekon}$. Untuk menampilkan dalam bentuk grafik, maka data yang keluar akan diperbarui setiap 2 detik, sehingga akan terlihat data yang keluar dalam bentuk grafik.

Tes Koneksi

Untuk mengetahui tes koneksi ke internet maka digunakan tes koneksi dengan ping. Keluaran akan tampil, bila sistem dapat mengirim paket ke DNS *server*.

Software yang digunakan

Netbeans

NetBeans mengacu pada kedua *platform* kerangka untuk aplikasi *desktop* Java, dan sebuah lingkungan pengembangan terpadu (IDE) untuk pengembangan dengan Java , JavaScript , PHP , Python , Ruby , Groovy , C , C + + , Scala , Clojure , dan lain-lain.

NetBeans IDE ditulis dalam Java dan berjalan di mana JVM diinstal, termasuk Windows, Mac OS, Linux, dan Solaris. Sebuah JDK diperlukan untuk pengembangan fungsionalitas Java, tetapi tidak diperlukan untuk pembangunan di bahasa pemrograman lain.

Platform NetBeans memungkinkan aplikasi untuk dikembangkan dari satu set modular komponen *software* yang disebut modul. Aplikasi berbasis *platform* NetBeans (termasuk IDE NetBeans) dapat diperpanjang oleh pengembang pihak ketiga .

Putty

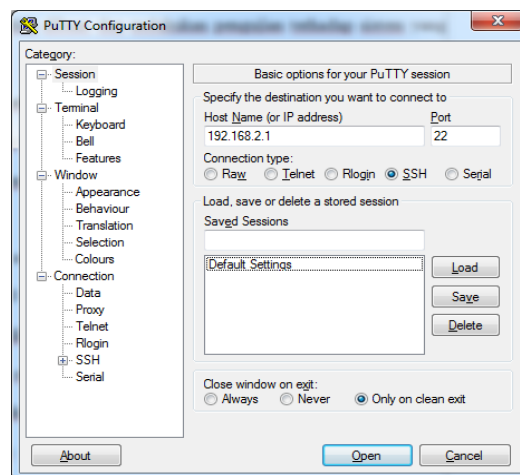
Putty itu *software* kecil untuk melakukan *remote* ke suatu *server* (komputer) yang membuka *port* 21 (telnet), *port* 22 (SSH), dsb. Telnet merupakan *service* yang dijalankan suatu komputer melalui port 21. Telnet bisa dilakukan dan menghubungi baik mesin windows ataupun unix/linux. Intinya harus ada *service* telnet-nya dan dibuka, maka komputer itu (*windows or non windows*) bisa di telnet melalui komputer lain (*windows or non windows*). Bisa melalui perintah 'telnet' di *command windows* atau linux/unix.

4. HASIL DAN PEMBAHASAN

Konfigurasi Mikrotik

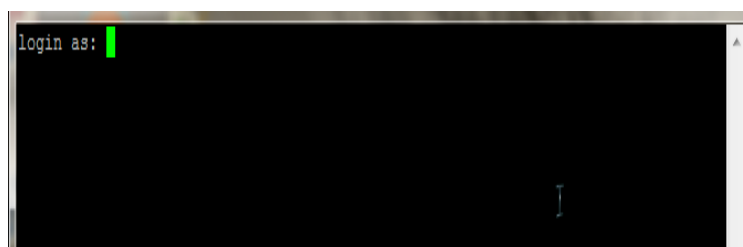
Untuk implementasi manajemen bandwidth, konfigurasi yang dilakukan adalah melalui command line dengan aplikasi putty, Berikut adalah proses konfigurasi yang dilakukan:

Mengetikkan IP address yang sebelumnya sudah disetting pada routerboard mikrotik RB750 192.168.2.1 pada putty seperti pada Gambar 4 dibawah ini, kemudian klik Open.



Gambar 4 Basic Option pada Putty

Kemudian akan masuk pada tampilan seperti Gambar 5 yang meminta *user* untuk melakukan *login*.



Gambar 5 Login Admin

Setelah sukses melakukan *login* maka akan keluar tampilan jendela *commandline* mikrotik seperti Gambar 6 dibawah ini.


```

MikroTik RouterOS 5.9 (c) 1999-2011      http://www.mikrotik.com/

jan/02/1970 00:00:36 system,error,critical router was rebooted without proper sh
u
tdown (cause 1)
jan/02/1970 00:00:11 system,error,critical router was rebooted without proper sh
u
tdown (cause 1)
jan/02/1970 00:00:11 system,error,critical router was rebooted without proper sh
u
tdown (cause 1)
[admin@MikroTik] >

```

Gambar 6 Tampilan *CommandLine* Mikrotik

Berikutnya mengubah nama mikrotik dengan mengetikkan perintah `/system identity set name=VIP` seperti Gambar 7 berikut.

```

[admin@MikroTik] > system identity set name=VIP
[admin@VIP] >

```

Gambar 7 Mengubah nama mikrotik

Memberi nama *interface* pada masing-masing *port* yang terhubung pada mikrotik dengan mengetikkan perintah `/interface Ethernet set ether1 name=[nama interface]` dan melihat *interface* yang telah diubah namanya tadi dengan mengetikkan perintah `/interface print`, seperti yang tampak pada Gambar 8 dibawah ini.

```

[admin@VIP] > interface ethernet set ether1 name=ether1-gateway
[admin@VIP] > interface ethernet set ether2 name=ether2-kantor
[admin@VIP] > interface ethernet set ether3 name=ether3-server
[admin@VIP] >

```

Gambar 8 Memberi nama interface

Memberi *IP address* pada setiap *interface* yang telah diset dengan mengetikkan perintah `/ip address add address=[ip address] interface=[nama interface]` dan melihat *ip address* yang telah disetting dengan mengetikkan perintah `/ip address print`, seperti yang terlihat pada Gambar 9 dan 10.

```

[admin@VIP] > ip address add address=192.168.1.254/24 interface=ether1-gateway
[admin@VIP] > ip address add address=192.168.2.1/24 interface=ether2-kantor
[admin@VIP] > ip address add address=192.168.100.1/24 interface=ether3-server
[admin@VIP] >

```

Gambar 9 Set *IP address* mikrotik

```

[admin@VIP] > ip add print
Flags: K - disabled, I - invalid, D - dynamic
#  ADDRESS          NETWORK          INTERFACE
0  192.168.1.254/24  192.168.1.0     ether1-gateway
1  192.168.2.1/24   192.168.2.0     ether2-kantor
2  192.168.100.1/24 192.168.100.0   ether3-server
[admin@VIP] >

```

Gambar 10 List *IP Address* pada port ethernet mikrotik

Menambahkan *IP route* pada mikrotik dengan mengetikkan perintah `/ip route add gateway=192.168.1.255` seperti yang terlihat pada gambar 11 berikut.

```

[admin@VIP] > ip route add gateway=192.168.1.254
[admin@VIP] >

```

Gambar 11 Menambahkan *IP route*

Kemudian setting NAT (*NetworkAddressTranslation*) seperti yang terlihat pada gambar 12.

```
[admin@VIP] > ip firewall nat add chain=srcnat out-interface=ether1-gateway action=masquerade
[admin@VIP] >
```

Gambar 12 Setting IP NAT mikrotik

Kemudian set DNS pada mikrotik dengan mengetikkan perintah `/ip dns set server=202.130.192.75,202.135.0.155 allow-remote request=yes` seperti yang terlihat pada Gambar 13 dibawah ini.

```
[admin@VIP] > ip dns set servers=203.130.192.74,202.134.0.155 allow-remote-requests=yes
[admin@VIP] > ip dns print
servers: 203.130.192.74,202.134.0.155
allow-remote-requests: yes
max-udp-packet-size: 512
cache-size: 2048KiB
cache-max-ttl: 1w
cache-used: 10KiB
[admin@VIP] >
```

Gambar 13 Set IP DNS mikrotik

Kemudian untuk proses manajemen *bandwidth* melalui *Hierarchichal Token Bucket* hal pertama yang dilakukan adalah dengan membuat *markpacket* untuk menandai paket yang keluar dan paket yang akan masuk ke dalam jaringan yang dibuat pada *ip firewall*, maka pada *commandline* ketikkan perintah berikut perintahnya `/ip firewall mangle add chain=[pilihan chain] action=[mark packet] new-packet-mark=[nama penanda ex: client-1]` seperti yang terlihat pada Gambar 14 berikut.

```
[admin@VIP] > ip firewall mangle add action=mark-packet chain=forward comment=PC1 dst-address=192.168.2.2 new-packet-mark=client1 passthrough=no
[admin@VIP] > ip firewall mangle add action=mark-packet chain=forward comment=PC2 dst-address=192.168.2.3 new-packet-mark=client2 passthrough=no
[admin@VIP] > ip firewall mangle add action=mark-packet chain=forward comment=PC3 dst-address=192.168.2.4 new-packet-mark=client3 passthrough=no
[admin@VIP] > ip firewall mangle add action=mark-packet chain=forward comment=PC4 dst-address=192.168.2.5 new-packet-mark=client4 passthrough=no
[admin@VIP] > ip firewall mangle add action=mark-packet chain=forward comment=PC5 dst-address=192.168.2.6 new-packet-mark=client5 passthrough=no
[admin@VIP] >
```

Gambar 14 Setting mangle

Kemudian untuk *listmangle* yang telah dibuat dapat dilihat dengan mengetikkan perintah `/ip firewall mangle print` seperti yang tampak pada gambar 15 dibawah ini.

```
[admin@VIP] > ip firewall mangle print
Flags: X - disabled, I - invalid, D - dynamic
0 ;; PC1
chain=forward action=mark-packet new-packet-mark=client1 passthrough=no
dst-address=192.168.2.2

1 ;; PC2
chain=forward action=mark-packet new-packet-mark=client2 passthrough=no
dst-address=192.168.2.3

2 ;; PC3
chain=forward action=mark-packet new-packet-mark=client3 passthrough=no
dst-address=192.168.2.4

3 ;; PC4
chain=forward action=mark-packet new-packet-mark=client4 passthrough=no
dst-address=192.168.2.5

4 ;; PC5
chain=forward action=mark-packet new-packet-mark=client5 passthrough=no
dst-address=192.168.2.6
[admin@VIP] >
```

Gambar 15 List IP firewall mangle

Langkah berikutnya dengan membuat *queue* untuk manajemen *bandwidth* yang akan dibuat, untuk pembagian *bandwidth* dilakukan batasan untuk *download*, sehingga ketika *client* melakukan aktifitas *download* maka *bandwidth* yang diterima masing-masing *client* akan sama, sehingga tidak mengganggu *user* lain dalam pemakaian *bandwidth*. Untuk perintahnya dapat dilihat pada Gambar 16.

```
[admin@VIP] > queue tree add max-limit=2M name=AllDownload parent=global-out priority=1
[admin@VIP] > queue tree add limit-at=300k max-limit=512k name=PC-Client1 packet-mark=client1 parent=AllDownload priority=8 queue=default
[admin@VIP] > queue tree add limit-at=300k max-limit=512k name=PC-Client2 packet-mark=client2 parent=AllDownload priority=8 queue=default
[admin@VIP] > queue tree add limit-at=300k max-limit=512k name=PC-Client3 packet-mark=client3 parent=AllDownload priority=8 queue=default
[admin@VIP] > queue tree add limit-at=300k max-limit=512k name=PC-Client4 packet-mark=client4 parent=AllDownload priority=8 queue=default
[admin@VIP] > queue tree add limit-at=300k max-limit=512k name=PC-Client5 packet-mark=client5 parent=AllDownload priority=8 queue=default
[admin@VIP] >
```

Gambar 16 Perintah untuk membuat *Queuetree*

Setelah *queue* selesai dibuat, untuk melihat *listqueue* tadi maka pada *commandline* ketikkan perintah */queue tree print* seperti yang tampak pada Gambar 17 berikut ini.

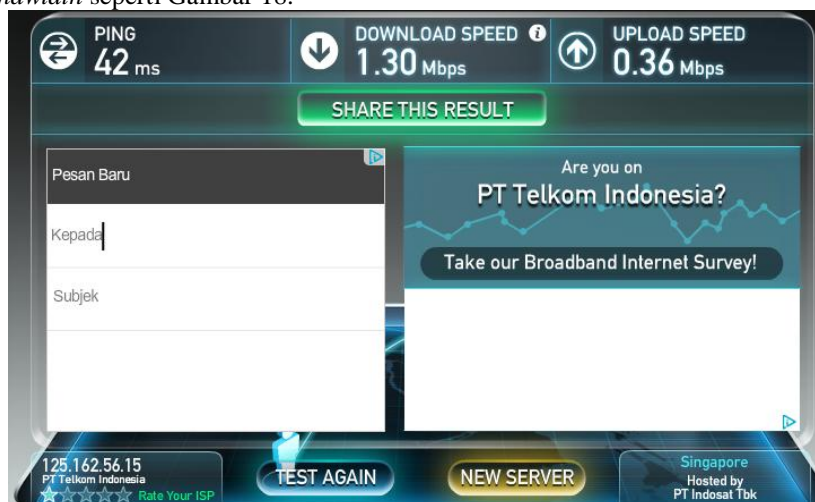
```
192.168.2.1 - PuTTY
Flags: X - disabled, I - invalid
0 name="AllDownload" parent=global-out limit-at=0 priority=1 max-limit=2M burst-limit=0 burst-threshold=0 burst-time=0s
1 name="PC-Client1" parent=AllDownload packet-mark=client1 limit-at=300k queue=default priority=8 max-limit=500k burst-limit=0 burst-threshold=0 burst-time=0s
2 name="PC-Client2" parent=AllDownload packet-mark=client2 limit-at=300k queue=default priority=8 max-limit=512k burst-limit=0 burst-threshold=0 burst-time=0s
3 name="PC-Client3" parent=AllDownload packet-mark=client3 limit-at=300k queue=default priority=8 max-limit=512k burst-limit=0 burst-threshold=0 burst-time=0s
4 name="PC-Client4" parent=AllDownload packet-mark=client4 limit-at=300k queue=default priority=8 max-limit=512k burst-limit=0 burst-threshold=0 burst-time=0s
5 name="PC-Client5" parent=AllDownload packet-mark=client5 limit-at=300k queue=default priority=8 max-limit=512k burst-limit=0 burst-threshold=0 burst-time=0s
[Q quit|D dump|down]
```

Gambar 17 *QueueList*

Pengujian

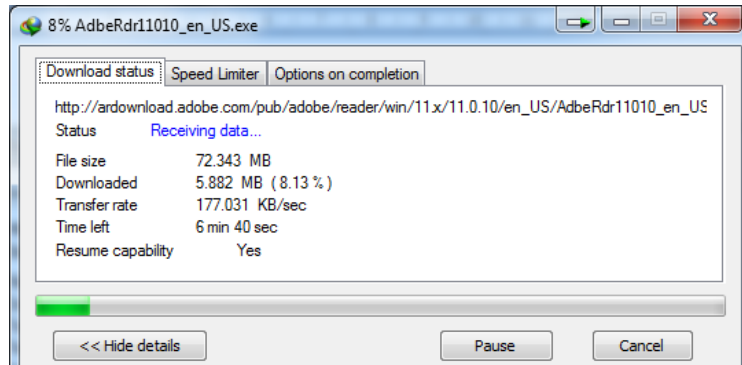
Untuk melakukan tes hasil konfigurasi dari manajemen *bandwidth* yang telah dilakukan maka pada pengujian ini akan dilakukan pengukuran *bandwidth* sebelum dan sesudah dimanajemen.

1. Sebelum dilakukan tes, maka terlebih dahulu melihat jumlah *bandwidth* pada PT. VIP Andalas Perkasa dengan mengakses halaman berikut <http://speedtest.net> maka akan tampak besar *bandwidth* seperti Gambar 18.



Gambar 18 *TestBandwidth* sebelum manajemen

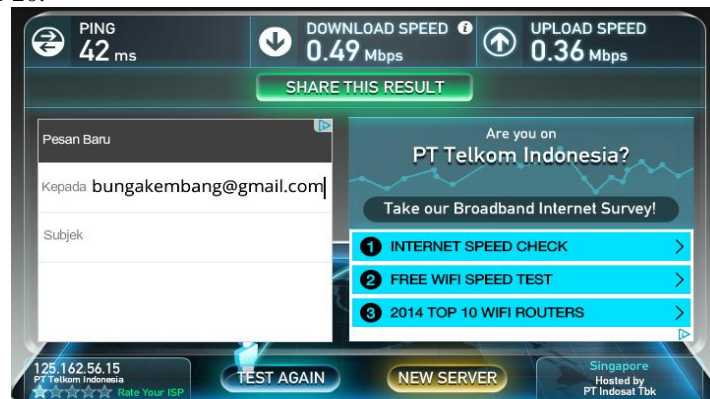
2. Sebelum melakukan manajemen *bandwidth*, maka dilakukan tes *download* sebuah file bertipe *exe*, ukuran 72.353 MB dengan aplikasi *Internet Download Manager*, dengan hasil seperti Gambar 19 berikut ini.



Gambar 19 DownloadFile sebelum manajemen *bandwidth*

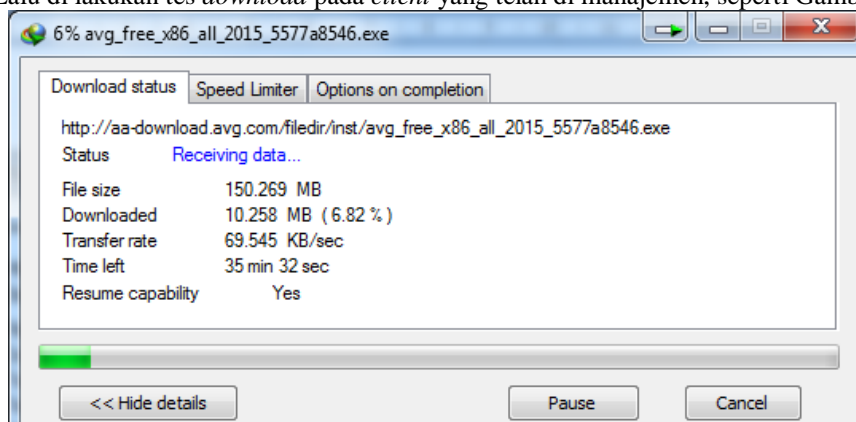
Tampak bahwa seperti yang ditunjukkan Gambar 19, besar file 72.252 MB, dan kecepatan *download* sebesar 177.031 KB/sec. Jumlah *bandwidth* yang tersedia adalah 1.30 Mbps, seperti yang tertera pada Gambar 19 bahwa untuk *download* setiap *client* rata-rata mampu melakukan *download* dengan kecepatan hingga 177.031 KB/sec.

- Setelah di lakukan manajemen *bandwidth* dengan masing-masing *client* diberi *bandwidth* sebesar 512 kb, maka di lakukan tes *bandwidth* dari <http://speedtest.net>, akan terlihat seperti Gambar 20.



Gambar 20 Test *bandwidth* sesudah manajemen

- Lalu di lakukan tes *download* pada *client* yang telah di manajemen, seperti Gambar 21.

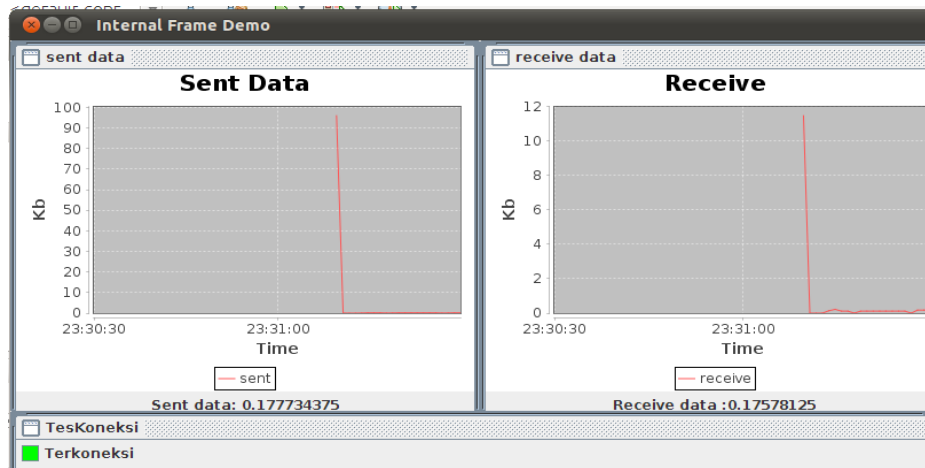


Gambar 21 Download file sesudah manajemen *bandwidth*

Tampak bahwa seperti yang ditunjukkan Gambar 21, besar file 150.269 MB, dan kecepatan *download* sebesar 69.555 KB/sec. Jumlah *bandwidth* yang diberikan adalah 512 kbps, seperti yang tertera pada Gambar 21 bahwa untuk *download* setiap *client* hanya mampu melakukan *download* dengan kecepatan hingga 69.555 KB/sec. Dengan demikian setiap *client*

tidak akan mengganggu *client* yang lain jika sama-sama melakukan *download* atau pun *browsing*.

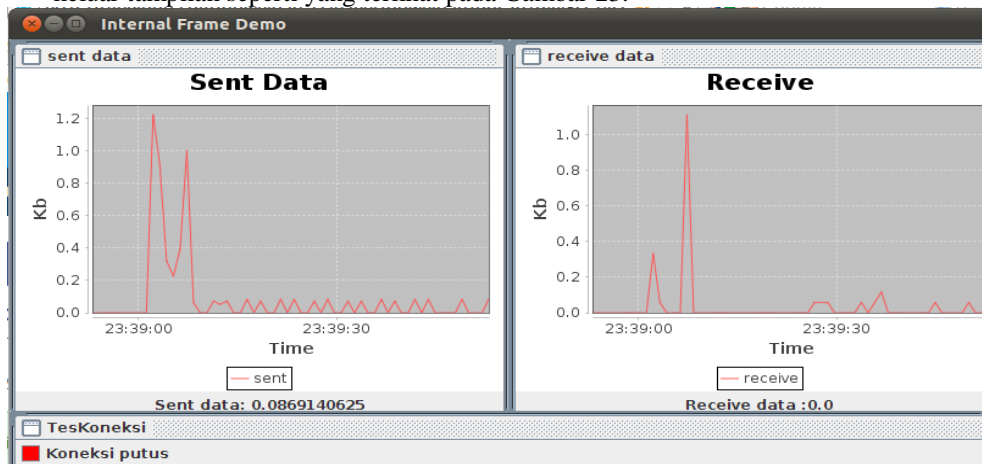
5. Pengujian *Monitoringbandwidth* diuji pada *server* yang berbasis linux, ketika aplikasi tersebut dijalankan maka akan tampak seperti Gambar 22.



Gambar 22 Tampilan *MonitoringBandwidth* Terkoneksi

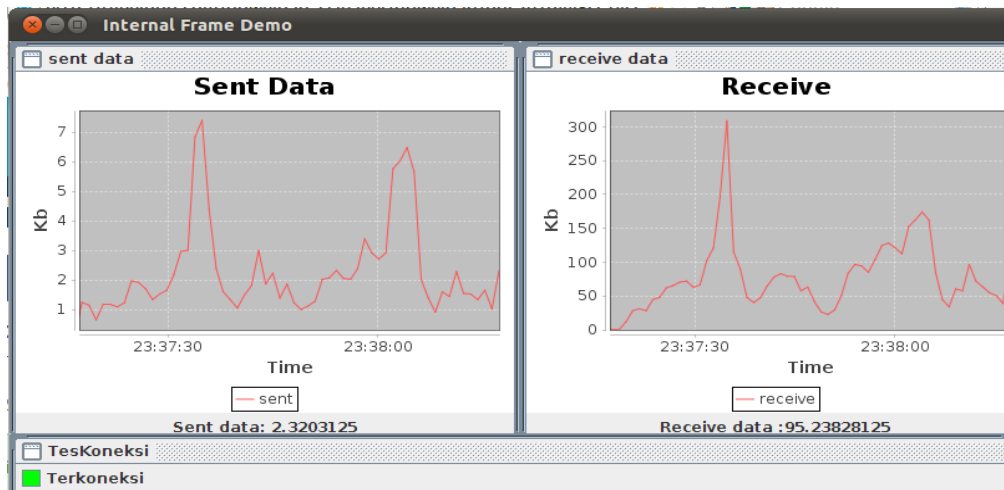
Dapat dilihat ada pergerakan grafik yang berarti menunjukkan bahwa ada *client* yang mengakses internet. Pada indikator “Terkoneksi” menunjukkan bahwa *server* sudah terkoneksi ke internet.

6. Apabila aplikasi yang dijalankan namun ketika *server* tidak terkoneksi ke internet maka akan keluar tampilan seperti yang terlihat pada Gambar 23.



Gambar 23 Tampilan *MonitoringBandwidth* Koneksi Putus

7. Tampilan koneksi ketika 5 PC terkoneksi ke internet ketika membuka *webbrowser* secara bersamaan. Seperti Gambar 24.



Gambar 24 Koneksi 5 PC terkoneksi internet

5. PENUTUP

Implementasi Manajemen *Bandwidth* dengan Teknik *Hierarchiecal Token Bucket* berbasis Mikrotik dan *Monitoring Bandwidth* dengan Java pada PT VIP Andalas Perkasa, dapat disimpulkan: Manajemen *bandwidth* menyebabkan pemakaian *bandwidth* menjadi teratur dan tersalurkan dengan baik ke setiap karyawan tanpa ada kesenjangan. Untuk melakukan teknik HTB sesuai *standart* yang telah di tentukan, untuk memberikan manajemen *bandwidth* yang optimal. Pengontrolan *bandwidth*, mengatasi masalah pemakaian *bandwidth* yang bebas oleh karyawan, sehingga tidak mengganggu jatah *bandwidth* karyawan lainnya. *Monitoring Bandwidth* yang dibuat dengan pemograman java dan di dalam sistem linux bekerja dengan cara mengambil jumlah data yang masuk dan jumlah data keluar, kemudian data tersebut di konversikan dalam bentuk grafik.

DAFTAR PUSTAKA

- _____. 2012. *Cisco CCNA & Jaringan Komputer*. Bandung : Informatika
- Antonius, Melvin dkk. 2004. *Membuat Animasi Dengan Java*. Jakarta : Elex Media Komputindo.
- MADCOMS. 2013. *Cepat & Mudah Membangun Sistem Jaringan Komputer*. Yogyakarta : C.V Andi Offset
- Siallagan, Sariadin. 2009. *Pemograman Java Dasar - dasar Pengenalan & Pemahaman*. Yogyakarta : C.V Andi Offset
- Sofana, Iwan. 2008. *Membangun Jaringan Komputer*. Bandung: Informatika
- Supriyanto, Aji. 2007. *Pengantar Teknologi Informasi*. Jakarta : Salemba Infotek
- Sutono. 2010. *Perangkat Komputer dan Tools Pendukungnya*. Bandung : Modula