



# Melek IT

Program Studi  
Teknik Informatika  
Universitas Wijaya Kusuma Surabaya

Jurnal Teknologi Informasi Dan Komunikasi

Volume 2 No 2 Juli 2013

ISSN 2252-9128



**PERANCANGAN SISTEM BANDWIDTH MANAGEMENT PADA PROXY SERVER MENGGUNAKAN HIERARCHY TOKEN BUCKET.** Indra Kumiawan, F.X. Wisnu Yudo Untoro.

**SISTEM OPTIMALISASI CACHE PROXY MENGGUNAKAN METODE REPLACEMENT PADA SQUID PROXY SERVER.** Erwin Damei Hadi Putra, Tjatusari Widiartin.

**IMPLEMENTASI APLIKASI PENGENDALIAN JARAK JAUH KAMERA DSLR VIA BLUETOOTH BERBASIS ANDROID.** Abednego Eko Harjiyanto, Guendra Kusuma Wardhana.

**IMPLEMENTASI WEB SERVICE BERBASIS SISTEM INFORMASI GEOGRAFIS UNTUK PENCARIAN LOKASI RUMAH MAKAN.** Wira Lamba, Nia Saurina.

**SISTEM OPTIMALISASI BANDWIDTH MENGGUNAKAN METODE LOAD BALANCING PADA SQUID PROXY SERVER.** Ayu Dian Pertiwi, Anang Kukuh A, Beny Y.V. Nasution.

**SISTEM INFORMASI MANAJEMEN RAPAT.** Achmad Nashihuddin, Nonot Wisnu Karyanto.

**HYBRID ARTIFICIAL BEE COLONY- DIVERSITY SEBAGAI ALGORITMA PEMBELAJARAN PADA NEURAL NETWORK.** Shofiya Syidada, Abidatul Izzah, Nuru Aini.

**IMPUTASI MISSING DATA MENGGUNAKAN METODE K-NEAREST NEIGHBOUR DENGAN OPTIMASI ALGORITMA GENETIKA.** Abidatul Izzah, Nur Hayatin.

**PREDIKSI CACAT PERANGKAT LUNAK PADA KODE SUMBER DENGAN SVM DAN SELEKSI ATRIBUT.** Ratih Kartika Dewi, Sarwosri.

**PEMBUATAN MODEL PENILAIAN KELAYAKAN ASPEK MANUSIA PADA PROYEK PERANGKAT LUNAK UNTUK MENINGKATKAN KUALITAS PRODUK PERANGKAT LUNAK.** Parma Hadi R, Sarwosri.

**SELEKSI FITUR TASK-STATE FMRI MENGGUNAKAN NON-OVERLAPPING CROSS-CORRELATION.** Agus Eko Minarno, Fitri Bimantoro, Arrie Kumiawardhani.



## Daftar Isi

1. **PERANCANGAN SISTEM *BANDWIDTH MANAGEMENT* PADA *PROXY SERVER* MENGGUNAKAN *HIERARCHY TOKEN BUCKET*.** Indra Kurniawan, F.X. Wisnu Yudo Untoro (Hal. 1-18)
2. **SISTEM OPTIMALISASI *CACHE PROXY* MENGGUNAKAN METODE *REPLACEMENT* PADA *SQUID PROXY SERVER*** Erwin Darni Hadi Putra, Tjatusari Widiartin. (Hal. 19 - 36)
3. **IMPLEMENTASI APLIKASI PENGENDALIAN JARAK JAUH KAMERA DSLR VIA BLUETOOTH BERBASIS ANDROID.** Abednego Eko Harjiyanto , Guendra Kusuma Wardhana (Hal. 37 - 46)
4. **IMPLEMENTASI WEB SERVICE BERBASIS SISTEM INFORMASI GEOGRAFIS UNTUK PENCARIAN LOKASI RUMAH MAKAN.** Wira Lamba, Nia Saurina (Hal. 47-62)
5. **SISTEM OPTIMALISASI *BANDWIDTH* MENGGUNAKAN METODE *LOAD BALANCING* PADA *SQUID PROXY SERVER*.** Ayu Dian Pertiwi, Anang Kukuh A, Beny Y.V. Nasution (Hal. 63-72)
6. **SISTEM INFORMASI MANAJEMEN RAPAT.** Achmad Nashihuddin, Nonot Wisnu Karyanto (Hal. 73-88)
7. **HYBRID *ARTIFICIAL BEE COLONY- DIVERSITY* SEBAGAI ALGORITMA PEMBELAJARAN PADA *NEURAL NETWORK*** Shofiya Syidada, Abidatul Izzah, Nuru Aini (Hal. 89-94)
8. **IMPUTASI *MISSING DATA* MENGGUNAKAN METODE *K-NEAREST NEIGHBOUR* DENGAN OPTIMASI ALGORITMA GENETIKA.** Abidatul Izzah, Nur Hayatin (Hal. 95-102)
9. **PREDIKSI CACAT PERANGKAT LUNAK PADA KODE SUMBER DENGAN SVM DAN SELEKSI ATRIBUT.** Ratih Kartika Dewi, Sarwosri (Hal. 103-110)
10. **PEMBUATAN MODEL PENILAIAN KELAYAKAN ASPEK MANUSIA PADA PROYEK PERANGKAT LUNAK UNTUK MENINGKATKAN KUALITAS PRODUK PERANGKAT LUNAK.** Parma Hadi Rantellinggi, Sarwosri (Hal. 111-118)
11. **SELEKSI FITUR TASK-STATE FMRI MENGGUNAKAN *NON-OVERLAPPING CROSS-CORRELATION*** Agus Eko Minamo, Fitri Bimantoro, Arrie Kurniawardhani (Hal. 119-130)

(1)

## PERANCANGAN SISTEM *BANDWIDTH MANAGEMENT* PADA *PROXY SERVER* MENGUNAKAN *HIERARCHY TOKEN BUCKET*

Indra Kurniawan<sup>1</sup>, F. X. Wisnu Yudo Untoro<sup>2</sup>

Program Studi Teknik Informatika, Universitas Wijaya Kusuma Surabaya, Jl. Dukuh Kupang XXV/ 54  
Email : indra300990@gmail.com

### ABSTRAK

Kebutuhan akan akses internet dewasa ini sangat tinggi. Dimana dalam suatu jaringan yang terhubung dengan internet, kecepatan *upload* maupun *download* merupakan hal yang sangat penting untuk memperlancar transmisi data.

Dalam sebuah jaringan *hotspot*, sering kali terjadi adanya dominasi *bandwidth* antar pengguna yang diakibatkan salah satu atau beberapa pengguna belum ter-authentikasi melakukan *download* sehingga akan mengganggu pengguna lain. Munculah ide untuk membangun sebuah sistem *bandwidth management* pada *proxy server* menggunakan *hierarchy token bucket*. Dimana di dalam sistem ini berfungsi mengelola pengguna yang belum ter-authentikasi agar dapat di bedakan mana yang dosen mana yang mahasiswa dan mengelola *bandwidth* (*bandwidth management*) yang tersedia dalam jaringan tersebut menggunakan *hierarchy token bucket* (*htb*) dimana salah satunya yaitu dengan membatasi pemakaian kapasitas *bandwidth* tiap pengguna.

Di harapkan setelah menggunakan sistem ini penggunaan *bandwidth* yang ada lebih efektif, lebih optimal dan tidak terjadi lagi perebutan *bandwidth* oleh *user*.

**Kata Kunci :** *bandwidth, bandwidth management, internet, hierarchy token bucket.*

### PENDAHULUAN

#### Latar Belakang

Kebutuhan akan akses internet dewasa ini sangat tinggi. Dimana dalam suatu jaringan yang terhubung dengan internet, kecepatan *upload* maupun *download* merupakan hal yang sangat penting untuk memperlancar transmisi data. Banyak hal yang dapat mempengaruhi kecepatan dua proses tersebut, diantaranya yaitu besarnya *bandwidth* yang digunakan jaringan tersebut dan seberapa efektif *bandwidth* tersebut bisa dimanfaatkan. *Bandwidth* adalah suatu ukuran dari banyaknya informasi yang dapat mengalir dari satu tempat ke tempat lain dalam satu waktu tertentu. *Bandwidth* dapat dipakai untuk mengukur aliran data analog maupun aliran data digital. Sekarang telah menjadi umum jika kata *bandwidth* lebih banyak digunakan untuk mengukur aliran data digital. Satuan *bandwidth* adalah *bits per second* atau sering disingkat *bps*. Jadi semakin besar *bandwidth* yang digunakan oleh suatu jaringan, seharusnya transmisi data pada jaringan tersebut akan menjadi semakin

cepat. Misalnya penggunaan *bandwidth* di jaringan hotspot kampus, Dengan adanya jaringan hotspot kampus bisa di gunakan untuk mencari informasi dengan mudah dan cepat.

Penggunaan *bandwidth* di sebuah jaringan hotspot kampus sering kali kurang dimanfaatkan secara optimal. Hal ini dapat disebabkan oleh adanya satu atau lebih pengguna yang belum ter-authentikasi menghabiskan kapasitas *bandwidth* dalam jaringan tersebut untuk *download* dengan menggunakan aplikasi *download manager* yang dapat menyita kapasitas *bandwidth*. Sehingga akibatnya pengguna lain tidak mendapatkan porsi *bandwidth* yang sesuai atau memadai. Keadaan yang seperti itu bisa mengganggu aktivitas *user* yang memang menggunakan internet sesuai kebutuhannya, seperti mencari informasi terbaru untuk menambah wawasan dan pengetahuan mereka seputar perkuliahan. Atau mengakses aplikasi yang sebenarnya hanya memerlukan *bandwidth* yang kecil, membutuhkan waktu yang lama untuk mengakses aplikasi yang diinginkan.

Dalam sebuah jaringan hotspot, sering kali terjadi adanya dominasi *bandwidth* antar pengguna

yang diakibatkan salah satu atau beberapa pengguna belum ter-authentikasi melakukan *download* sehingga akan mengganggu pengguna lain. Munculah ide untuk merancang sebuah sistem *bandwidth management* pada *proxy server* menggunakan *hierarchy token bucket*. Dimana di dalam sistem ini berfungsi mengelola pengguna yang belum ter-authentikasi agar dapat di bedakan mana yang dosen mana yang mahasiswa dan mengelola *bandwidth (bandwidth management)* yang tersedia dalam jaringan tersebut dimana salah satunya yaitu dengan membatasi pemakaian kapasitas *bandwidth* tiap pengguna. Dengan demikian jika ada pengguna yang mengakses internet yang membutuhkan kapasitas *bandwidth* yang besar, maka pengguna lain tidak akan terganggu, karena tiap-tiap pengguna sudah mempunyai kapasitas *bandwidth* masing masing yang dapat dipakai untuk mengakses internet.

Di harapkan setelah menggunakan sistem ini penggunaan *bandwidth* yang ada lebih efektif, lebih optimal dan tidak terjadi lagi perebutan *bandwidth* oleh *user*. Dan sarana dan prasarana internet di kampus tersebut lebih baik dan nyaman untuk di gunakan oleh *user*.

## TINJAUAN PUSTAKA

### Pengertian Bandwidth Management

Penggunaan Internet bersama pastinya mempengaruhi *bandwidth* dan kecepatan transfer data antar komputer. Oleh karena itu dibutuhkan manajemen *bandwidth* agar tidak terjadi penguasaan *bandwidth* secara tunggal. Dapat dibayangkan bagaimana jika terjadi penguasaan *bandwidth*, hal ini akan mempengaruhi koneksi pengguna lain. Tanpa manajemen *bandwidth*, setiap komputer akan secara otomatis memperluas *bandwidth*-nya tergantung kebutuhan yang terbesar. Sehingga, apabila ada yang *download* video dan *file* berukuran besar maupun *streaming* yang akan berakibat melambatnya koneksi pada komputer lain. Hal ini disebut dengan *fair usage* atau penggunaan yang adil.

### Pengertian Proxy Server

*Proxy* dapat dipahami sebagai pihak ketiga yang berdiri ditengah-tengah antara kedua pihak yang saling berhubungan dan berfungsi sebagai perantara, sedemikian sehingga pihak pertama dan pihak kedua tidak secara langsung berhubungan, akan tetapi masing-masing berhubungan dengan perantara, yaitu *proxy*. Sebuah analogi; bila seorang mahasiswa meminjam buku di perpustakaan, kadang si mahasiswa tidak diperbolehkan langsung mencari

dan mengambil sendiri buku yang kita inginkan dari rak, tetapi kita meminta buku tersebut kepada petugas, tentu saja dengan memberikan nomor atau kode bukunya, dan kemudian petugas tersebut yang akan mencarinya dan mengambilkan bukunya.

Dalam kasus diatas, petugas perpustakaan tersebut telah bertindak sebagai perantara atau *Proxy*. Petugas tersebut juga bisa memastikan dan menjaga misalnya, agar mahasiswa hanya bisa meminjam buku untuk mahasiswa, dosen boleh meminjam buku semua buku, atau masyarakat umum hanya boleh meminjam buku tertentu. Mungkin proses tersebut menjadi lebih lama dibandingkan bila kita langsung mencari dan mengambil sendiri buku yang kita inginkan.

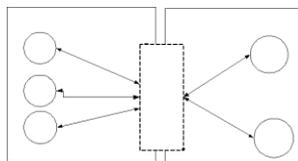
Namun bila saja setiap kali petugas mencari dan mengambil buku untuk seseorang, si petugas juga membuat beberapa salinan dari buku tersebut sebelum memberikan bukunya kepada orang yang meminta, dan menyimpannya di atas meja pelayanan, maka bila ada orang lain yang meminta buku tertentu, sangat besar kemungkinan buku yang diminta sudah tersedia salinannya diatas meja, dan si petugas tinggal memberikannya langsung. Hasilnya adalah layanan yang lebih cepat dan sekaligus keamanan yang baik.

Analogi diatas menjelaskan konsep dan fungsi dasar dari suatu *proxy* dalam komunikasi jaringan komputer dan internet. *Proxy server* mempunyai 3 fungsi utama yaitu *Connection Sharing*, *Filtering* dan *Caching*. Masing masing fungsi akan dijelaskan lebih lanjut dibawah. *Proxy* dalam pengertiannya sebagai perantara, bekerja dalam berbagai jenis protokol komunikasi jaringan dan dapat berada pada level-level yang berbeda pada hirarki *layer* protokol komunikasi jaringan. Suatu perantara dapat saja bekerja pada *layer Data-Link*, *layer Network* dan *Transport*, maupun *layer Aplikasi* dalam hirarki *layer* komunikasi jaringan menurut *OSI*. Namun pengertian *proxy server* sebagian besar adalah untuk menunjuk suatu *server* yang bekerja sebagai *proxy* pada *layer Aplikasi*, meskipun juga akan dibahas mengenai *proxy* pada level sirkuit.

Dalam suatu jaringan lokal yang terhubung ke jaringan lain atau internet, pengguna tidak langsung berhubungan dengan jaringan luar atau internet, tetapi harus melewati suatu *gateway*, yang bertindak sebagai batas antara jaringan lokal dan jaringan luar. *Gateway* ini sangat penting, karena jaringan lokal harus dapat dilindungi dengan baik dari bahaya yang mungkin berasal dari internet, dan hal tersebut

akan sulit dilakukan bila tidak ada garis batas yang jelas jaringan lokal dan internet. *Gateway* juga bertindak sebagai titik dimana sejumlah koneksi dari pengguna lokal akan terhubung kepadanya, dan suatu koneksi ke jaringan luar juga terhubung kepadanya.

Dengan demikian, koneksi dari jaringan lokal ke internet akan menggunakan sambungan yang dimiliki oleh *gateway* secara bersama-sama (*connection sharing*). Dalam hal ini, *gateway* adalah juga sebagai *proxy server*, karena menyediakan layanan sebagai perantara antara jaringan lokal dan jaringan luar atau internet. Diagram berikut menggambarkan posisi dan fungsi dari *proxy server*, diantara pengguna dan penyedia layanan:



Gambar 2.1 diagram posisi dan fungsi *proxy server*.

Pada gambar 2.1 diatas di jelaskan bahwa *proxy server* adalah sebagai *gateway* atau jembatan antara jaringan lokal dan jaringan luar atau internet.

### Squid

*Squid* adalah sebuah *daemon* yang digunakan sebagai *proxy server* dan *web cache*. *Squid* memiliki banyak jenis penggunaan, mulai dari mempercepat *server web* dengan melakukan *caching* permintaan yang berulang-ulang, *caching DNS*, *caching situs web*, dan *caching* pencarian komputer di dalam jaringan untuk sekelompok komputer yang menggunakan sumber daya jaringan yang sama, hingga pada membantu keamanan dengan cara melakukan penyaringan (*filter*) lalu lintas. Meskipun seringnya digunakan untuk protokol *HTTP* dan *FTP*, *Squid* juga menawarkan dukungan terbatas untuk beberapa protokol lainnya termasuk *Transport Layer Security (TLS)*, *Secure Socket Layer (SSL)*, *Internet Gopher*, dan *HTTPS*.

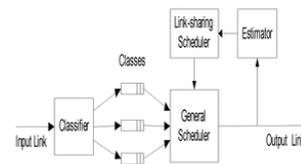
### Hierarchy Token Bucket

*Hierarchical Token Bucket (HTB)* merupakan teknik penjadwalan paket yang baru-baru ini diperkenalkan bagi *router* berbasis *Linux*, dikembangkan pertama kali oleh Martin Devera pada akhir 2001 untuk diproyeksikan sebagai pilihan (atau pengganti) mekanisme penjadwalan yang saat ini masih banyak dipakai yaitu *CBQ*.

Sebagai catatan, apabila nilai *ceil* sama dengan nilai *base link*, maka akan memiliki fungsi yang sama seperti parameter *bounded* pada *CBQ*, di mana 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*.

Istilah *bandwidth management* sering diartikan dengan istilah *traffic control*, yang dapat didefinisikan sebagai pengalokasian yang tepat dari suatu *bandwidth* untuk mendukung kebutuhan atau keperluan aplikasi atau suatu layanan jaringan. Istilah *bandwidth* dapat didefinisikan sebagai kapasitas atau daya tampung suatu *channel* komunikasi (*medium* komunikasi) untuk dapat dilewati sejumlah *traffic* informasi atau data dalam satuan waktu tertentu. Umumnya *bandwidth* dihitung dalam satuan *bit*, *kbit* atau *bps* (*byte per second*). Pengalokasian *bandwidth* yang tepat dapat menjadi salah satu metode dalam memberikan jaminan kualitas suatu layanan jaringan (*QoS = Quality Of Services*).

Teknik antrian *Hierarchical Token Bucket* mirip dengan *CBQ* hanya perbedaannya terletak pada opsi, *Hierarchical Token Bucket* lebih sedikit opsi saat konfigurasi serta lebih presisi. Teknik antrian *Hierarchical Token Bucket* memberikan kita fasilitas pembatasan trafik pada setiap level maupun klasifikasi, *bandwidth* yang tidak terpakai bisa digunakan oleh klasifikasi yang lebih rendah. Kita juga dapat melihat *Hierarchical Token Bucket* seperti suatu struktur organisasi dimana pada setiap bagian memiliki wewenang dan mampu membantu bagian lain yang memerlukan, teknik antrian *Hierarchical Token Bucket* sangat cocok diterapkan pada perusahaan atau institusi pendidikan dengan banyak struktur organisasi.



Gambar 2.2 *hierarchy token bucket*.

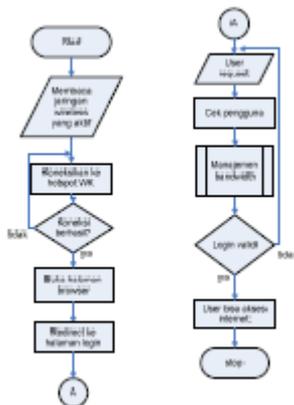
## ANALISA DAN PERANCANGAN

### Perencanaan Konsep Sistem

Sistem *bandwidth management* pada *proxy server* menggunakan *hierarchy token bucket* yang akan dibangun mempunyai fitur yaitu membuat autentikasi pengguna *hotspot* yang di integrasikan dengan manajemen *bandwidth*. salah satu cara yaitu

dengan membuat beberapa tingkatan dalam autentikasi tersebut. Tingkatan tersebut di bagi menjadi dua yaitu dosen dan mahasiswa. Masing-masing tingkatan tersebut sudah mendapatkan kapasitas *bandwidth* yang di tentukan.

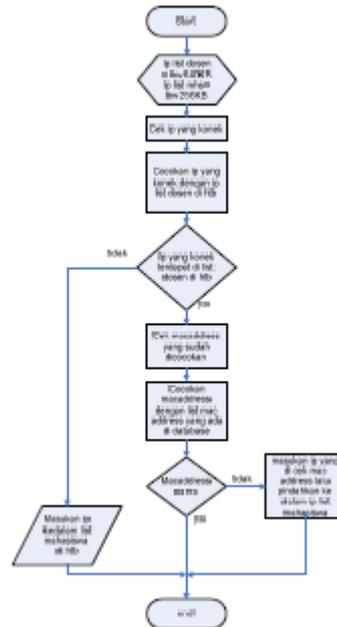
Apabila ada pengguna yang menghabiskan kapasitas *bandwidth* dalam jaringan tersebut dengan cara *men-download* menggunakan aplikasi *download manager* dan akibatnya pengguna lain tidak mendapatkan porsi *bandwidth* yang sesuai atau memadai, bisa di atasi dengan sistem ini. Dengan demikian jika ada pengguna yang mengakses internet yang membutuhkan kapasitas *bandwidth* yang besar, maka pengguna lain tidak akan terganggu, karena tiap-tiap pengguna sudah mempunyai kapasitas *bandwidth* masing-masing yang dapat dipakai untuk mengakses internet. Dan di harapkan setelah menggunakan sistem ini penggunaan *hotspot* lebih optimal dan tidak terjadi perebutan *bandwidth* kembali oleh *user* yang menggunakan fasilitas tersebut. Dan sarana dan prasarana internet kampus lebih baik dan nyaman untuk digunakan. Untuk lebih jelasnya bisa melihat desain *flowchart* sistem manajemen *bandwidth* berikut:



Gambar 3.1 *flowchart* sistem manajemen *bandwidth*

Pada gambar 3.1 diatas merupakan runtutan cara kerja sistem manajemen *bandwidth* sebagai berikut, pertama setiap *user* yang masuk kedalam *hotspot* kita lewat *wireless* dan mencoba untuk *browsing* internet, semuanya akan *redirect* ke *login* yang meminta klien untuk memasukkan *username* dan *password*. Ketika *username* dan *password* telah dimasukkan maka sistem akan mengecek apakah ada *username* dan *password* yang dimasukkan oleh si *user* bersangkutan. Lalu ada proses manajemen *bandwidth* dimana di dalamnya ada proses lagi yaitu proses untuk manajemen *bandwidth* untuk *user*. Sistem akan memproses

apakah *user* tersebut login sebagai dosen atau mahasiswa. Jika sudah, maka sistem akan memberikan izin sehingga si *user* bisa *surfing* di internet, dan jika tidak, maka si akan meminta login ulang dan begitu seterusnya. Disini penjelasan proses manajemen *bandwidth*:



Gambar 3.2 *flowchart* proses manajemen *Bandwidth*

Pada gambar 3.2 proses manajemen *bandwidth* dimana di Perencanaan kebutuhan sistem di tentukan parameter bahwa ip list dosen di *htb* mendapatkan *bandwidth* 640KB, dan ip list mahasiswa di *htb* mendapatkan *bandwidth* 256KB. Pertama sistem akan mengecek ip yang konek. Setelah itu sistem akan mencocokkan ip yang konek dengan ip list dosen. Disini ip list dosen sebagai acuannya karena disini ip list dosen lebih besar kapasitas *bandwidth* yang di dapat. Jika ip yang konek tidak terdapat maka sistem akan langsung memasukan kedalam ip list mahasiswa di dalam *htb*, dan Jika ip yang konek terdapat di ip list dosen maka sistem akan mengecek *mac address* ip tersebut. Setelah di cek, maka *mac address* tersebut akan di cocokkan dengan data *mac address* di dalam *database*. Jika *mac address* tersebut cocok maka biarkan. Dan jika tidak cocok maka sistem akan menghapus ip tersebut yang sudah terdapat di dalam ip list dosen pindahkan ke dalam ip list mahasiswa. Setelah itu selesai.

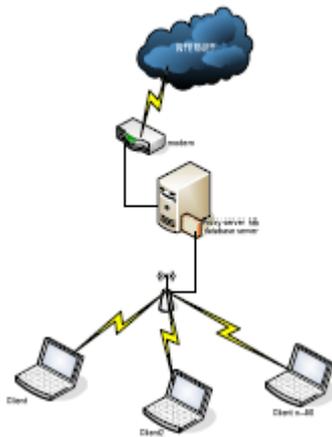
**Perencanaan kebutuhan sistem**

Berisi perancangan (desain) dari perangkat keras maupun lunak yang akan digunakan dalam

melakukan simulasi sistem *hotspot*, penentuan perangkat lunak dan topologi yang akan digunakan, sekaligus pengaturan perangkat keras tersebut agar sesuai dengan topologi yang diinginkan.

### Desain Jaringan yang di gunakan

Topologi jaringan komputer nirkabel yang akan digunakan penulis terhadap studi literatur yang telah dilakukan yaitu topologi dengan konsep *Portal*, dimana konsep dari topologi ini ialah topologi jaringan yang umum digunakan untuk *hotspot*. *Hotspot* menjadi portal untuk akses bagi *pc client*.



Gambar 3.3 desain jaringan *proxy server*

Pada gambar 3.3 diatas terdapat *server proxy* yang terkoneksi dengan modem internet. Dimana fungsi *proxy server* yaitu sebagai *gateway* atau batas jaringan local dengan jaringan luar. Didalam *server proxy* juga ada *htb* (*hierarchy token bucket*) yang berfungsi sebagai aplikasi untuk me-manajemen *bandwidth* pengguna. Didalam *server proxy* tersebut juga terdapat *Mysql* sebagai *database servernya*. Internet tersebut disebar oleh *server proxy* menggunakan *access point*, sehingga internet yang yang disebar tersebut bisa di akses oleh *client* menggunakan *wifi*.

### Hardware yang digunakan

Ada beberapa spesifikasi *hardware* dan peralatan jaringan yang di gunakan pada pengujian tugas akhir ini yaitu:

1. Router TPLINK MR3420



Gambar 3.4 Router *Wireless* TP-Link MR3420  
Tampak Depan & Belakang.

Spesifikasi : *USB 2.0 Port for LTE/ HSPA/ HSDPA/ UMTS/ EVDO USB Modem 1 10/100Mbps WAN Port, 4 10/100Mbps LAN Ports, support the auto-Negotiation and auto-MDI/MDIX.*

2. Modem smartfren CE682 REV A
3. *Server* laptop HP COMPAQ PRESARIO CQ40

### Software Yang Digunakan

Selain kebutuhan *hardware* yang telah di jelaskan diatas, ada beberapa *software* yang dibutuhkan untuk dapat dilakukannya pengujian sistem *bandwidth management* menggunakan *hierarchy token bucket* tanpa ada masalah. *Software-software* yang digunakan adalah sebagai berikut:

1. *Laptop server*
  - a. sistem operasi : *linux ubuntu server 10.04*
  - b. aplikasi : *squid , MySQL, HTB*
2. *Laptop klien*
  - a. sistem operasi : *Windows 7 ultimate SP1 x32*

### Perencanaan pengujian

Pada penelitian yang dilakukan di jaringan *wireless* yang ada. Penulis menemukan kriteria untuk di lakukan pengujian. Kriteria tersebut di bagi menjadi 3 bagian:

1. Pengujian *login user* pada *proxy server*.

Pada pengujian ini adalah memeriksa apakah *user* yang telah terautentikasi dengan benar sehingga dapat melakukan akses internet keluar cara kerjanya saat seorang pengguna berusaha untuk melakukan *browsing* ke Internet, *proxy* akan memaksa pengguna yang belum terautentikasi untuk menuju ke *Authentication web* dan akan di beri *prompt login*. Jika *username* dan *password* yang di masukkan benar maka *user* bisa mengakses internet luar.

2. Pengujian manajemen user

Pengujian ini dilakukan untuk mengetahui apakah bisa jika ada klien yang ingin masuk jika jumlah klien yang di tentukan itu sudah penuh.

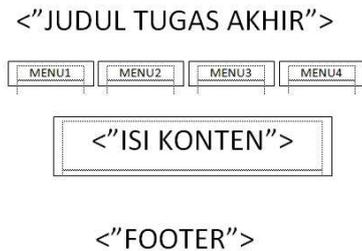
3. Pengujian *download* klien

Pengujian kali ini bertujuan agar bisa

menunjukkan seberapa besar komputer klien jika mendownload menggunakan IDM (*Internet Download Manager*).

### Perencanaan antarmuka

Pada perencanaan antarmuka, digunakan sebagai *format* desain yang digunakan untuk setiap tampilan menu yang nantinya dijadikan sebagai *template* utama. Bisa dilihat pada gambar 3.5:



Gambar 3.5 Desain Perencanaan Antarmuka.

Pada gambar 3.5, desain perencanaan antarmuka memiliki 4 bagian yang diantaranya adalah sebagai berikut:

1. Judul tugas akhir (berisi judul dari tugas akhir).
2. MENU1-4 (berisi menu-menu dari modul yang tersedia).
3. ISI KONTEN (berisi konten dari masing-masing modul).
4. FOOTER (berisi identitas pembuat).

## IMPLEMENTASI SISTEM

### Konfigurasi Pada komputer Server

Komputer server yang akan dibuat ini supaya *client* bisa terhubung dengan server atau untuk menerima permintaan dari *client*, sehingga di lakukan persiapan supaya komputer server dalam pengujian bisa sesuai dengan yang diinginkan. Adapun persiapan tersebut antara lain.

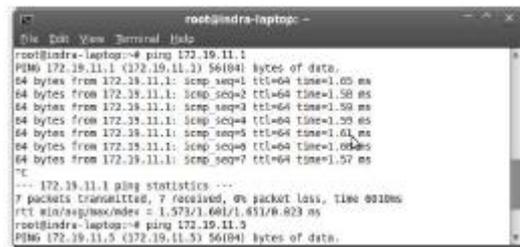
### Pemasangan IP Address Server

Pada komputer server ini akan diisikan *IP Address* sebagai *IP* fisik dari komputer server tersebut. Adapun isi dari *file/etc/network/interfaces* yang terletak di *Terminal Console* sebagai berikut.

```
1. Auto lo
2. iface lo inet loopback
3. auto eth0
4. iface eth0 inet static
5.     address 103.29.231.26
6.     netmask 255.255.255.0
7.     gateway 103.29.231.17
8. auto eth1
9. iface eth1 inet static
10.    address 172.19.11.2
11.    netmask 255.255.255.0
12.    gateway 172.19.11.1
```

Gambar 4.1 IP Address eth0

Penjelasan 4.1 adalah, *eth0* dan *eth1* merupakan *interfaces card* yang sesungguhnya pada komputer server. Untuk menguji berhasil apa tidaknya pemasangan *IP address* pada komputer server maka akan dilakukan perintah sebagai berikut :



Gambar 4.2 Ping 172.19.11.1

### Konfigurasi Router

Pada komputer server ini membutuhkan sebuah router untuk menyebarkan internet yang nantinya akan di akses oleh user. Disini router di konfigurasi DHCP (*Dynamic Host Configuration Protokol*). Dimana nanti user yang mengakses router tersebut akan mendapatkan *ipaddress* secara otomatis. supaya penulis bisa menggunakan komputer server sesuai dengan konsep yang sudah dibuat. Adapun langkah-langkah untuk mengkonfigurasi router tersebut adalah sebagai berikut :

1. buka browser dan akses *ipaddress* router yaitu 172.19.11.1, maka akan muncul tampilan *tplink router* . Pilih DHCP – DHCP Setting. Untuk lebih jelas bisa di gambarkan dengan gambar sebagai berikut:

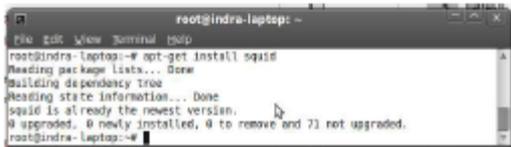


Gambar 4.3 Setting DHCP

### Pemasangan Proxy Server

Penggunaan *Proxy Server* atau yang biasa disebut dengan *application firewall*, yang bertujuan perantara antara jaringan luar dan jaringan lokal, sehingga dengan menggunakan *proxy* memudahkan kita mengatur / mengontrol akses internet dan me-manajemen *bandwidth* bagi *user*. Adapun langkah-langkah yang dilakukan untuk membuat *Proxy Server*, sebagai berikut :

1. Instalasi *Proxy Server* :
  - `root@indra-laptop:~# apt-get install squid` untuk lebih jelasnya bisa lihat gambar berikut:



Gambar 4.4 Proses Install Squid

- a. Konfigurasi *squid* agar menjadi mode *ncsa authentication* dengan cara sebagai berikut:

1. Buat *username* dan *password* `root@indra-laptop:~# htpasswd /etc/squid/passwd namauser`  
**Output:** `New password: Re-type new password:`  
 Adding password for user `namauser`

2. Setelah itu pastikan bahwa *squid* bisa membaca *file passwd* tersebut dengan cara :

```
root@indra-laptop:~# chmod o+r /etc/squid/passwd
root@indra-laptop:~# dpkg -L squid | grep ncsa_auth
```

**Output:** `/usr/lib/squid/ncsa_auth`

- b. Ubah dan menambahkan perintah di bawah ini pada *file gedit /etc/squid/squid.conf* yang terletak di *Terminal Console* :

1. `auth_param basic program`
2. `/usr/lib/squid/ncsa_auth`
3. `/etc/squid/passwd`
4. `auth_param basic children 5`
5. `auth_param basic realm Squid proxy-caching web server`
6. `auth_param basic credentialsttl 2 hours`
7. `auth_param basic casesensitive off`
8. `acl ncsa_users proxy_auth`
9. `REQUIRED`
10. `http_access allow ncsa_users`

Gambar 4.5 Squid.conf

Gambar 4.5 merupakan perintah untuk merubah konfigurasi *squid* menjadi mode *ncsa authentication* pada komputer *server* tersebut.

- c. *Restart Squid*

```
root@indra-laptop:~# /etc/init.d/squid restart
```

### Pemasangan Hierarchy Token Bucket

Penggunaan *HTB* atau *Hierarchy Token Bucket* bertujuan untuk mengatur pembagian *bandwidth*, pembagian dilakukan secara hirarki yang dibagi-bagi kedalam kelas sehingga mempermudah pengaturan *bandwidth*. Adapun langkah-langkah untuk install *Hierarchy Token Bucket*, sebagai berikut :

- a. Instalasi *Hierarchy Token Bucket* :

di karenakan *package htb* ini tidak tersedia di repositori maka harus *download* secara manual. *Copy* kan file *htb* tersebut ke dalam `/usr/local/src`.

1. Setelah itu ekstrak file tersebut dan akan muncul *etc, install, folder, sbin*.
  - `root@indra-laptop:~# tar -zxvf HTB-tools-0.3.0a-i486-1.tgz`
2. pindahkan isi semua folder *sbin* nya *htb tools* di `/sbin server`
  - `root@indra-laptop:~# cd /usr/local/src/sbin`
  - `root@indra-laptop:~# mv htb /sbin`
  - `root@indra-laptop:~# mv htbgen /sbin`
  - `root@indra-laptop:~# mv q_checkcfg/sbin`
  - `root@indra-laptop:~# mv q_parser /sbin`
  - `root@indra-laptop:~# mv q_show /sbin`
3. setelah itu masuk ke folder *etc* nya *htb tools*, pindahkan *folder htb* ke `/etc` nya *server* dengan cara sebagai berikut:
  - `root@indra-laptop:~# cd /usr/local/src/etc`
  - `root@indra-laptop:~# mv htb /etc`
4. setelah itu hilangkan tulisan *new* yang ada di dalam *folder htb* yang susah kita pindahkan tadi dengan cara sebagai berikut:

- `root@indra-laptop:~# mv /etc/htb/eth0-qos.cfg.new /etc/htb/eth0-qos.cfg`
  - `root@indra-laptop:~# mv /etc/htb/eth1-qos.cfg.new /etc/htb/eth1-qos.cfg`
5. setelah itu masuk ke folder `rc.d` di dalam `etc/htb/tools` dengan cara sebagai berikut:
- `root@indra-laptop:~# cd /usr/local/src/etc/rc.d`
6. masukan file `rc.htb.new` ke dalam folder `init.d` di server kita dan di ganti jadi `rc.htb` dengan cara sebagai berikut:
- `root@indra-laptop:~# mv rc.htb.new /etc/init.d/rc.htb`
7. `chmod` file `rc.htb` dengan cara sebagai berikut:
- `root@indra-laptop:~# chmod 755 /etc/init.d/rc.htb`
- b. Ubah dan menambahkan perintah di bawah ini pada file `/etc/htb/eth1-qos.cfg` yang terletak di `Terminal Console` :

```
1. class wijaya_kusuma {
2.   bandwidth 640;
3.   limit 1024;
4.   burst 2;
5.   priority 1;
6.   client dosen {
7.     bandwidth 256;
8.     limit 640;
9.     burst 2;
10.    priority 1;
11.    src {
12.      172.19.11.4/24; 172.19.11.5/24;
13.    };
14.  };
15.  client mahasiswa {
16.    bandwidth 128;
17.    limit 256;
18.    burst 2;
19.    priority 1;
20.    src {
21.      172.19.11.3/24;172.19.11.6/24;
22.      172.19.11.7/24;
23.    };
24.  };
25.  };
26.  class default { bandwidth 8; };
```

Gambar 4.6 `eth0-qos.cfg`

Perintah pada gambar 4.6 adalah untuk pengaturan `bandwidth` yang di bagi menjadi kelas wijaya kusuma dan di dalam kelas tersebut ada kategori dosen dan kategori mahasiswa.

- c. Jika sudah di konfigurasi maka tinggal menjalankan `htb` toolnya dengan cara sebagai berikut:

```
root@indra-laptop:~# etc/init.d/rc.htb
start_eth0
```

- d. jika ingin melihat `traffic` nya bisa di lihat dengan cara sebagai berikut:

```
root@indra-laptop:~# /etc/init.d/rc.htb
show_eth0
```

#### **Penggabungan Squid Dan Hierarchy token bucket**

di dalam server ini perlu adanya penggabungan antara `squid` dan `htb tools`. Adapun proses `penggabungan` dan langkah-langkah konfigurasinya pada komputer server adalah sebagai berikut:

1. Ubah dan tambahkan perintah di bawah ini pada file `/etc/squid/squid.conf` yang terletak di `Terminal Console` :

```
1. zph_mode tos
2. zph_local 0x30
3. zph_parent 0
4. zph_option 136
```

Gambar 4.7 Konfigurasi `squid.conf`

Perintah pada gambar 4.7 adalah untuk menggabungkan `squid zph` dengan `htb tools`.

2. Setelah itu `restart squid` dengan cara sebagai berikut:

- `root@indra-laptop:~# squid -k reconfigure`

3. konfigurasi `htb-tools`

- `root@indra-laptop:~# q_parser eth0 10000 10000 /etc/htb/eth0-qos.cfg > /home/admin/eth0-qos.sh`

4. Ubah dan tambahkan perintah di bawah ini pada file `/home/admin/eth0-qos.sh` yang terletak di `Terminal Console` :

```
1. $TC qdisc add dev $DEV parent
2. 1:0x10 handle 0x10: pfifo limit 5
3. $TC class add dev $DEV parent
4. 1:1 classid 1:0x15 htb rate 10Mbit
5. $TC filter add dev $DEV parent
6. 1:0 protocol ip prio 1 u32 match
7. ip protocol 0x6 0xff match ip tos
   0x30 0xff flowid 1:15
```

Gambar 4.8 `eth0-qos.sh`

Perintah pada gambar 4.8 adalah untuk mengkonfigurasi `htb-tools`.

### Konfigurasi *rc.local*

Didalam *rc.local* perlu adanya konfigurasi agar internet yang ada bisa di teruskan ke user atau pengguna. Konfigurasi tersebut bisa di lihat pada gambar 4.9 berikut ini:



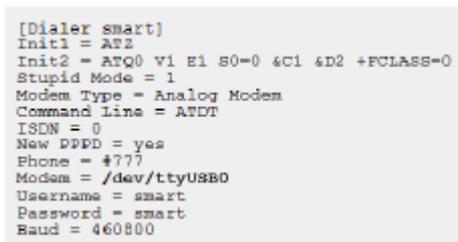
Gambar 4.9 Konfigurasi *rc.local*

Pada gambar 4.9 dapat di jelaskan bahwa *script* tersebut menjalankan perintah internet yang di hasilkan oleh modem akan di teruskan ke koneksi LAN yang mempunyai ip 172.19.11.2 .

### Koneksi Modem

Pada komputer *server* perlu internet jadi konfigurasi koneksi modem sebagai berikut:

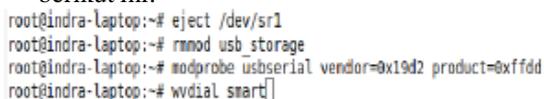
1. Install package *wvdial*  
*apt-get install wvdial*
2. konfigurasi *wvdial* dengan mengetikan perintah sebagai berikut:  
*gedit /etc/wvdial.conf*
3. kemudian ganti isi dari *default* konfigurasinya dengan konfigurasi *dialer smart* pada gambar 4.10 berikut ini:



Gambar 4.10 konfigurasi *Dialer Smart*.

Pada gambar 4.10 adalah konfigurasi *dialer smart* yang nantinya akan di gunakan untuk mengkoneksikan *modem smart*.

4. Lakukan perintah untuk mengkoneksikan modem *smart* bisa dilihat pada gambar 4.11 berikut ini:



Gambar 4.11 perintah untuk mengkoneksikan modem *smart*.

Pada gambar 4.11 adalah perintah untuk mengkoneksikan modem *smart*. Langkah pertama adalah *eject /dev/sr1* dan *rmmod usb\_storage* dan *modprobe usbserial vendor=0x19d2 product=0xffdd* agar modem bisa terbaca sebagai fungsi modem. Langkah kedua adalah *wvdial smart* untuk mengkoneksikan modem ke internet.

### Konfigurasi Pada Komputer Klien

Pada komputer klien harus dilakukan konfigurasi agar komputer klien bisa terkoneksi kedalam *proxy server*. adapun langkah-langkahnya sebagai berikut:

1. Buka *browser*, disini diambil contoh *firefox* bisa dilihat pada gambar 4.12 berikut ini:



Gambar 4.12 *Firefox Preference*

Pada gambar 4.12 adalah untuk menkonfigurasi *proxy* klien. Pilih *tab advanced* setelah itu pilih lagi *tab network* dan terakhir pilih *tab connection setting*. Setelah itu akan muncul *popup menu* kembali seperti gambar 4.13 berikut ini:



Gambar 4.13 *Connection Setting*

Pada gambar 4.13 adalah tampilan konfigurasi *proxy* untuk mengakses internet. Konfigurasi default adalah *auto detect proxy settings for this network* untuk *firefox*. Tapi disini ganti dengan *manual proxy configuration*. Dan isi *http proxy* nya 172.19.11.2 ini adalah ip dari komputer *server*. lalu isi *port* yang akan digunakan yaitu 3128. Setelah itu klik *ok*.

## Desain Antarmuka

Pada sub-bab desain antarmuka, dijelaskan setiap desain antarmuka yang telah direncanakan, sehingga pembaca bisa lebih mudah dalam pemahaman menu dan fungsi masing-masing modul yang telah disediakan.

### Form tambah user wireless

Gambar 4.14 form tambah user wireless.

Pada gambar 4.14. merupakan form tambah user dimana di tengah ada form inputan untuk memasukan username, password dan macaddress. Dimana form ini adalah form untuk menambah user wireless dimana nanti yang akan dipakai klien untuk login proxy server.

### Form view data user

Gambar 4.15 Form View Datauser

Pada gambar 4.15 merupakan form untuk melihat siapa saja user yang sudah tersimpan di dalam database. Di form view datauser ada tabel Username, password, dan macaddress disitu nanti akan muncul user siapa saja yang sudah tersimpan. Di form view data user juga ada tab aksi dimana disitu fungsinya juga bisa menghapus user yang ada.

### Form menjalankan squid

Gambar 4.16 Form menjalankan squid.

Pada gambar 4.16 merupakan form untuk menjalankan squid. Disini lebih memudahkan pengguna sistem ini karena mereka tidak perlu menjalankan squid proxy server menggunakan perintah yang di jalankan melalui terminal linux. Disitu juga ada button untuk menghentikan squid.

### Form menjalankan HTB

Gambar 4.17 Form menjalankan HTB.

Pada gambar 4.17 merupakan form untuk menjalankan HTB. Disini pengguna sistem bisa menjalankan HTB dengan mengklik button start htb dan menghentikan htb dengan mengklik stop htb.

## HASIL DAN PEMBAHASAN

### Pengujian Sistem

Pengujian sistem ini dilakukan sesuai dengan skenario yang sudah dibuat pada bab 3, dimana nantinya di bab ujicoba ini mempunyai beberapa bentuk dari pengujian antara lain Tujuan Pengujian Skenario dan Tahapan Pengujian.

### Pengujian login user pada proxy server

Pada pengujian ini adalah Pengujian login user pada proxy server. dimana user memasukan username dan password.

#### a. Tujuan pengujian

Tujuan dari scenario pengujian 1 ini dibuat adalah memeriksa apakah user yang telah terautentikasi dengan benar sehingga dapat melakukan akses internet keluar.

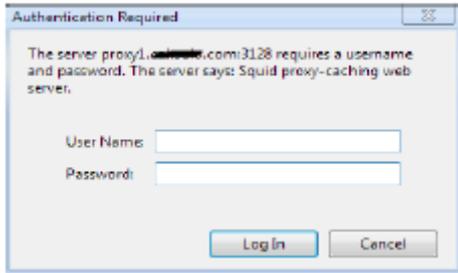
#### b. Tahapan pengujian

Komputer klien mengkoneksikan ke wireless wijaya kusuma. Setelah itu computer klien mengkonfigurasi pada browser ke konfigurasi proxy agar bisa mendapatkan akses internet.

Dari langkah-langkah pengujian yang telah dilakukan tersebut, terdapat 2 cara penilaian pengujian dimana cara tersebut nantinya bisa dijadikan acuan untuk mengetahui apakah pengujian ini dapat dinyatakan berhasil atau gagal. Berikut adalah 2 cara penilain tersebut :

#### 1. Pengujian berhasil

Pengujian dikatakan berhasil jika klien ketika ingin mengakses suatu website muncul popup login proxy yang akan terdapat form inputan username dan password. Bisa dilihat pada gambar 5.1 berikut ini:



Gambar 5.1 login proxy server.

Pada gambar 5.1 merupakan tampilan *form login proxy*. Dimana klien harus memasukan *username* dan *password* untuk bisa mendapatkan akses internet. Jika *username* dan *password* yang di masukan benar maka *browser* akan otomatis *me-redirect* ke halaman *website* yang akan di akses oleh klien.

## 2. Pengujian gagal

Pengujian gagal apabila klien memasukan *username* dan *password* yang salah. Akan muncul *form login* kembali, jika di *cancel* atau di *close* akan muncul *error cache access denied*. Bisa dilihat pada gambar berikut ini:



Gambar 5.2 error cache acces denied.

Pada gambar 5.2 diatas merupakan tampilan jika *login* gagal dan pengguna mengklik *close* atau *cancel form login*. Akan muncul tampilan *error cache access denied*.

## Pengujian manajemen user

Pada pengujian kali ini *computer server* melakukan uji coba manajemen *user*. Ketika jumlah klien yang sudah di tentukan penuh dan ada dosen yang ingin masuk menggunakan fasilitas ini.

### a. Tujuan pengujian

Pengujian ini dilakukan untuk mengetahui apakah bisa jika ada klien yang ingin masuk jika jumlah klien yang di tentukan itu sudah penuh.

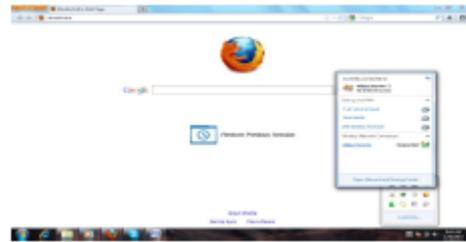
### b. Tahapan pengujian

Jika jumlah klien yang sudah di tentukan, misal *ip range wifi router* yang sudah di konfigurasi *dhcp* 172.19.11.3 sampai 172.19.11.7.

Dari langkah-langkah pengujian yang telah dilakukan tersebut, terdapat 2 cara penilaian pengujian dimana cara tersebut nantinya bisa dijadikan acuan untuk mengetahui apakah pengujian ini dapat dinyatakan berhasil atau gagal. Berikut adalah 2 cara penilain tersebut :

## 1. Pengujian berhasil

Pengujian di katakan berhasil jika pengguna dosen atau mahasiswa yang ingin masuk menggunakan fasilitas ini bisa terkoneksi dengan *wifi wijaya kusuma*. Bisa dilihat pada gambar berikut ini:

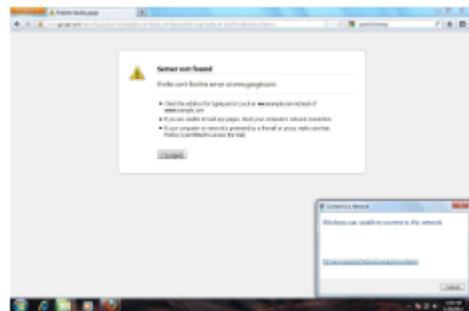


Gambar 5.3 terkoneksi dengan hotspot wijaya kusuma.

Pada gambar 5.3 diatas merupakan tampilan jika *user* atau pengguna belum penuh dan bisa terkoneksi dengan jaringan *hotspot wijaya kusuma*.

## 2. Pengujian gagal

Pengujian dikatakan gagal apabila pengguna dosen yang ingin masuk menggunakan fasilitas ini tidak bisa terkoneksi dan mendapatkan *ip*. Bisa dilihat pada gambar berikut ini:



Gambar 5.4 unable to connect to this network.

Pada gambar 5.4 diatas merupakan tampilan jika ada *user* atau pengguna yang ingin terkoneksi dengan *hotspot wijaya kusuma* dan di *hotspot* tersebut *user* atau pengguna sudah penuh 5 pengguna, maka *user* tersebut tidak akan bisa terkoneksi dengan jaringan *hotspot* tersebut.

## Pengujian download klien

Pada pengujian kali ini komputer klien akan melakukan uji coba *download* menggunakan *IDM (Internet Download Manager)*.

### a. Tujuan pengujian

Pengujian kali ini bertujuan agar bisa menunjukkan seberapa besar komputer klien jika mendownload menggunakan *IDM (Internet Download Manager)*.

**b. Tahapan pengujian**

Pada pengujian kali ini akan di ambil *sample* atau contoh 5 kali pengujian. Bisa di lihat pada tabel di bawah ini:

No	Skema	Batas <i>maximum download</i>	<i>Bandwidth</i> Tanpa dibatasi <i>HTB</i>	Berhasil / gagal
1	2 dosen 1 mahasiswa	Dosen = 80KB/sec Mahasiswa = 30KB/sec	544KB/sec	berhasil
2	2 dosen 2 mahasiswa	Dosen = 80KB/sec Mahasiswa = 30KB/sec	544KB/sec	berhasil
3	2 dosen 3 mahasiswa	Dosen = 80KB/sec Mahasiswa = 30KB/sec	544KB/sec	berhasil
4	1 dosen 3 mahasiswa	Dosen = 80KB/sec Mahasiswa = 30KB/sec	544KB/sec	berhasil
5	4 dosen 1 mahasiswa	Dosen = 80KB/sec Mahasiswa = 30KB/sec	544KB/sec	berhasil

Tabel 5.1 ruang lingkup uji coba.

Pada tabel 5.1 diatas merupakan ruang lingkup uji coba dimana ada skema pengujian, batas *maximum download*, *bandwidth* tanpa dibatasi oleh *HTB* dan ada berhasil atau gagalnya pengujian tersebut. Skema pengujian diatas telah ditentukan parameter pengujiannya dengan parameter 2 dosen dan 3 mahasiswa. Dan bisa di lihat pada gambar berikut ini:



Gambar 5.5 konfigurasi *htb*

Pada gambar 5.5 adalah konfigurasi umum *htb* untuk di ambil contoh pengujian. Disitu bisa dilihat ada class wijaya kusuma yang mempunyai *bandwidth minimum* 640KB dan mempunyai *bandwidth maximum* 1024KB. Dimana didalam class tersebut ada 2 klien yaitu klien dosen dan klien mahasiswa. Klien dosen mempunyai *bandwidth minimum* 256KB dan mempunyai *bandwidth maximum* 640KB. Di klien mahasiswa mempunyai *bandwidth minimum* 128KB dan mempunyai *bandwidth maximum* 256KB. Dan di



dalam klien klien tersebut sudah di tentukan *ip range* yang nantinya akan di pakai untuk dosen dan mahasiswa. Dan pengujiannya sebagai berikut:

1. Pengujian pertama di ambil contoh 2 dosen dan 1 mahasiswa

Pada kali ini yang *login* bersamaan yaitu 2 dosen dan 1 mahasiswa dimana 2 dosen tersebut menggunakan *ip address* 172.19.11.4 dan 172.19.11.5 dan 1 mahasiswa tersebut 172.19.11.3. ini adalah gambar dosen pertama ketika melakukan *download* menggunakan *IDM* bisa dilihat pada gambar berikut ini:



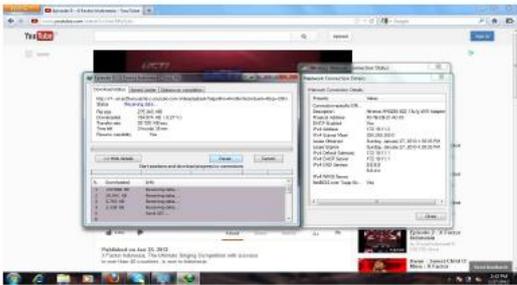
Gambar 5.6 *download idm* dosen 1

Pada gambar 5.6 diatas merupakan proses *download* dosen 1 dengan menggunakan *ip address* 172.19.11.4. dia mendapatkan batas *maximum download* dengan kecepatan 79KB/sec. sama halnya dengan dosen 2 bisa dilihat pada gambar berikut:



Gambar 5.7 *download idm dosen 2*

Pada gambar 5.7 diatas merupakan proses *download* dosen 2 dengan menggunakan *ip address* 172.19.11.5. dia juga mendapatkan batas *maximum download* dengan kecepatan 80KB/sec. berbeda dengan mahasiswa 1 bisa dilihat pada gambar berikut ini;



Gambar 5.8 *download idm mahasiswa 1*

Pada gambar 5.8 diatas merupakan proses *download* mahasiswa 1 dengan menggunakan *ip address* 172.19.11.3. yang mendapatkan batas *maximum download* dengan kecepatan 30KB/sec.

2. Pengujian kedua di ambil contoh 2 dosen dan 2 mahasiswa

Pada kali ini yang *login* bersamaan yaitu 2 dosen dan 2 mahasiswa dimana 2 dosen tersebut menggunakan *ip address* 172.19.11.4 dan 172.19.11.5 dan 2 mahasiswa tersebut 172.19.11.3 dan 172.19.11.6. ini adalah gambar dosen 1 ketika melakukan *download* menggunakan *IDM* bisa dilihat pada gambar berikut ini:



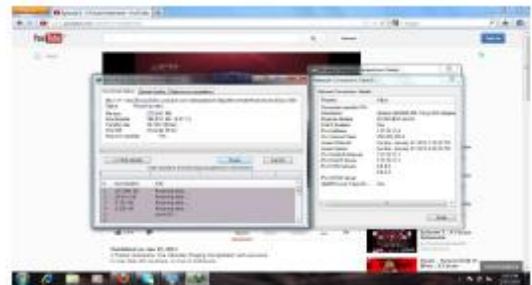
Gambar 5.9 *download idm dosen 1*

Pada gambar 5.9 diatas merupakan proses *download* dosen 1 dengan menggunakan *ip address* 172.19.11.4. dia mendapatkan batas *maximum download* dengan kecepatan 79KB/sec. sama halnya dengan dosen 2 bisa dilihat pada gambar berikut:



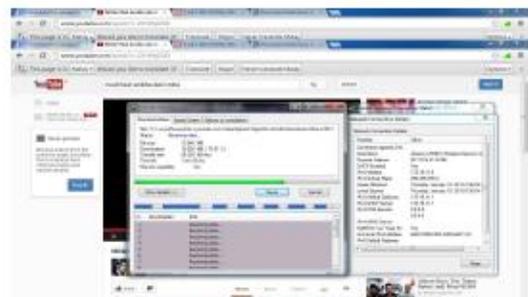
Gambar 5.10 *download idm dosen 2*

Pada gambar 5.10 diatas merupakan proses *download* dosen 2 dengan menggunakan *ip address* 172.19.11.5. dia juga mendapatkan batas *maximum download* dengan kecepatan 80KB/sec. berbeda dengan mahasiswa 1 bisa dilihat pada gambar berikut ini;



Gambar 5.11 *download idm mahasiswa 1*

Pada gambar 5.11 diatas merupakan proses *download* mahasiswa 1 dengan menggunakan *ip address* 172.19.11.3. yang mendapatkan batas *maximum download* dengan kecepatan 30KB/sec. dan mahasiswa 2 bisa di lihat pada gambar berikut ini:



Gambar 5.12 *download idm mahasiswa 2*

Pada gambar 5.12 diatas merupakan proses *download idm* mahasiswa 2 yang menggunakan

*ip address* 172.19.11.6. dan mendapatkan batas *maximum download* dengan kecepatan 28KB/sec.

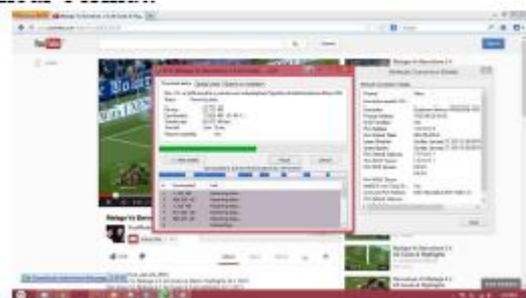
3. Pengujian ketiga di ambil contoh 2 dosen 3 mahasiswa

Pada kali ini yang *login* bersamaan yaitu 2 dosen dan 3 mahasiswa dimana 2 dosen tersebut menggunakan *ip address* 172.19.11.4 dan 172.19.11.5 dan 2 mahasiswa tersebut 172.19.11.3 dan 172.19.11.6 dan 172.19.11.7. ini adalah gambar dosen 1 ketika melakukan *download* menggunakan *IDM* bisa dilihat pada gambar berikut ini:



Gambar 5.13 *download idm* dosen 1

Pada gambar 5.13 diatas merupakan proses *download* dosen 1 dengan menggunakan *ip address* 172.19.11.4. dia mendapatkan batas *maximum download* dengan kecepatan 79KB/sec. sama halnya dengan dosen 2 bisa dilihat pada gambar berikut:



Gambar 5.14 *download idm* dosen 2

Pada gambar 5.14 diatas merupakan proses *download* dosen 2 dengan menggunakan *ip address* 172.19.11.5. dia juga mendapatkan batas *maximum download* dengan kecepatan 80KB/sec. berbeda dengan mahasiswa 1 bisa dilihat pada gambar berikut ini;



Gambar 5.15 *download idm* mahasiswa 1

Pada gambar 5.15 diatas merupakan proses *download* mahasiswa 1 dengan menggunakan *ip address* 172.19.11.3. yang mendapatkan batas *maximum download* dengan kecepatan 30KB/sec. dan mahasiswa 2 bisa di lihat pada gambar berikut ini:



Gambar 5.16 *download idm* mahasiswa 2

Pada gambar 5.16 diatas merupakan proses *download idm* mahasiswa 2 yang menggunakan *ip address* 172.19.11.6. dan mendapatkan batas *maximum download* dengan kecepatan 28KB/sec. dan mahasiswa 3 bisa dilihat pada gambar sebagai berikut:



Gambar 5.17 *download idm* mahasiwa 3

Pada gambar 5.17 diatas merupakan proses *download idm* mahasiswa 3 yang menggunakan *ip address* 172.19.11.7. dan mendapatkan batas *maximum download* dengan kecepatan 30KB/sec.

4. Pengujian keempat di ambil contoh 1 dosen 3 mahasiswa

Pada kali ini yang *login* bersamaan yaitu 1 dosen dan 3 mahasiswa dimana 1 dosen tersebut menggunakan *ip address* 172.19.11.5 dan 2 mahasiswa tersebut 172.19.11.3 dan 172.19.11.7. ini adalah gambar dosen 1 ketika melakukan *download* menggunakan *IDM* bisa dilihat pada gambar berikut ini:



Gambar 5.18 *download idm* dosen 1

Pada gambar 5.18 diatas merupakan proses *download* dosen 1 dengan menggunakan *ip address* 172.19.11.5. dia mendapatkan batas *maximum download* dengan kecepatan 80KB/sec. berbeda dengan mahasiswa 1 bisa dilihat pada gambar berikut ini;



Gambar 5.19 *download idm* mahasiswa 1

Pada gambar 5.19 diatas merupakan proses *download* mahasiswa 1 dengan menggunakan *ip address* 172.19.11.3. yang mendapatkan batas *maximum download* dengan kecepatan 30KB/sec. dan mahasiswa 2 bisa di lihat pada gambar berikut ini:



Gambar 5.20 *download idm* mahasiswa 2

Pada gambar 5.20 diatas merupakan proses *download idm* mahasiswa 2 yang menggunakan *ip address* 172.19.11.6. dan mendapatkan batas *maximum download* dengan kecepatan 28KB/sec. dan mahasiswa 3 bisa dilihat pada gambar sebagai berikut:

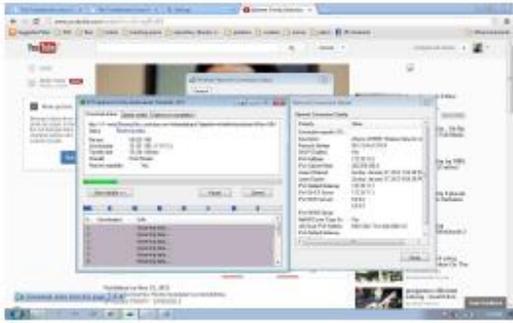


Gambar 5.21 *download idm* mahasiswa 3

Pada gambar 5.21 diatas merupakan proses *download idm* mahasiswa 3 yang menggunakan *ip address* 172.19.11.7. dan mendapatkan batas *maximum download* dengan kecepatan 30KB/sec.

5. Pengujian kelima di ambil contoh 4 dosen 1 mahasiswa

Pada kali ini yang *login* bersamaan yaitu 4 dosen dan 1 mahasiswa dimana 4 dosen tersebut menggunakan *ip address* 172.19.11.4 dan 172.19.11.5 dan 172.19.11.3 dan 172.19.11.6. dan 1 mahasiswa tersebut menggunakan *ip address* 172.19.11.7. ini adalah gambar dosen 1 ketika melakukan *download* menggunakan *IDM* bisa dilihat pada gambar berikut ini :



Gambar 5.22 download idm dosen 1

Pada gambar 5.22 diatas merupakan proses download dosen 1 dengan menggunakan ip address 172.19.11.4. dia mendapatkan batas maximum download dengan kecepatan 79KB/sec. sama halnya dengan dosen 2 bisa dilihat pada gambar berikut:



Gambar 5.23 download idm dosen 2

Pada gambar 5.23 diatas merupakan proses download dosen 2 dengan menggunakan ip address 172.19.11.5. dia juga mendapatkan batas maximum download dengan kecepatan 80KB/sec. sama halnya dengan dosen 3 bisa dilihat pada gambar berikut ini;



Gambar 5.24 download idm dosen 3

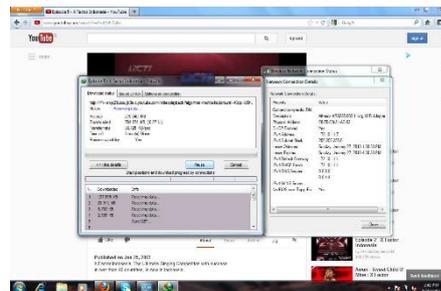
Pada gambar 5.24 diatas merupakan proses download dosen 3 dengan menggunakan ip address 172.19.11.3. yang mendapatkan batas maximum download dengan kecepatan 79KB/sec.

dan sama juga dengan dosen 4 bisa di lihat pada gambar berikut ini:



Gambar 5.25 download idm dosen 4

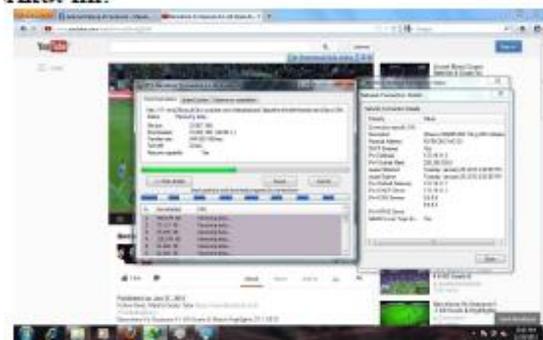
Pada gambar 5.25 diatas merupakan proses download idm dosen 4 yang menggunakan ip address 172.19.11.6. dan mendapatkan batas maximum download dengan kecepatan 79KB/sec. dan berbeda dengan mahasiswa 1 bisa dilihat pada gambar sebagai berikut:



Gambar 5.26 download idm mahasiswa 1

Pada gambar 5.26 diatas merupakan proses download idm mahasiswa 1 yang menggunakan ip address 172.19.11.7. dan mendapatkan batas maximum download dengan kecepatan 30KB/sec.

Dari pengujian diatas, dilakukan juga pengujian pengguna melakukan download menggunakan IDM dan pengguna tersebut tidak dibatasi oleh HTB. Bisa dilihat pada gambar berikut ini:



Gambar 5.27 download pengguna tanpa htb.

Pada gambar 5.27 diatas merupakan hasil *download* pengguna yang tidak dibatasi *bandwidth*nya oleh *HTB*. Sehingga pengguna bisa mendapatkan kecepatan *download* sebesar 544KB/sec.

## PENUTUP

Secara umum Perancangan sistem manajemen *bandwidth* pada *proxy server* menggunakan *hierarchy token bucket* ini dapat berfungsi sebagaimana yang diharapkan. Namun tentu saja masih diperlukan beberapa penyempurnaan yang akan dipaparkan dalam pembahasan bab ini.

## Kesimpulan

Berdasarkan uraian pembahasan perancangan dan pengujian yang telah dilakukan, maka dapat diambil kesimpulan terhadap sistem yaitu rancangan sistem manajemen *bandwidth* pada *proxy server* menggunakan *hierarchy token bucket* dapat digunakan sebagai alternatif yang baik untuk memamanajemen *bandwidth* di kampus.

## Saran

Saran untuk perkembangan lebih lanjut dalam masa mendatang pada Perancangan sistem manajemen *bandwidth* pada *proxy server* menggunakan *hierarchy token bucket* ini yaitu dalam pengujiannya bisa lebih banyak pengguna lagi yang di ujikan. Dan agar username yang di daftarkan bisa digunakan di laptop pengguna lain.

## DAFTAR PUSTAKA

- [1] yHuS\_dCc\_X, 2011. *Definisi Jaringan Komputer*. <http://sengkang-flash.blogspot.com/2011/01/defenisi-jaringan-komputer.html> diakses 3 juli 2012.
- [2] adi4adi. 2010. *HTB (Hierarchical Token Bucket)*.<http://www.scribd.com/doc/24030951/HTB> diakses 2 juli 2012.
- [3] acieeee, 2010. *Pengertian Proxy Server*. <http://acieeee.wordpress.com/2010/04/01/pengertian-proxy-server/> diakses 1 juli 2012
- [4] Croll, Alistair, 2000, *Managing Bandwidth Deploying QoS in Enterprise Networks*, Prentice Hall, New Jersey.
- [5] Devera, Martin, *HTB Manual User Guide* <http://luxik.cdi.cz/~devik/qos/>. Diakses 4 juli 2012.
- [6] *Howto Squid proxy authentication using ncsa\_auth helper*, <http://www.cyberciti.biz/tips/linux-unix-squid-proxy-server-authentication.html>. Diakses 5 september 2012.