

Optimasi Bandwidth Hits Local Up To Dengan Transparent Proxy 2.7 Lusca Menggunakan Metode Network Development Life Cycle

Amat Suroso

STMIK Bani Saleh, ahmad_suroso04@yahoo.com

ABSTRAK

Akses internet sekarang ini sudah menjadi suatu kebutuhan sebagian orang, informasi dengan mudah didapatkan dengan cepat melalui jaringan internet, namun banyak informasi di internet tersebut tidak bermanfaat dan mengarah pada hal yang negatif. Penggunaan jaringan internet juga dimanfaatkan oleh STMIK Bani Saleh, namun kecepatan akses internet di STMIK Bani Saleh tidak stabil akibat penguasaan *bandwidth* secara individu.

Untuk memecahkan masalah ini maka digunakan *proxy server* yang merupakan suatu sistem yang dapat berfungsi untuk menangani *caching*, *filtering web* yang mengarah pada hal negatif. *Proxy server* juga sekaligus sebagai *manajemen bandwidth* yang sangat diperlukan supaya tidak terjadi penguasaan *bandwidth* secara individu oleh *client*. Linux OpenSUSE 12.1 digunakan sebagai sistem operasi *proxy server* dan menggunakan metode NDLC (*Network Development Life Cycle*) untuk pengaturan *bandwidth*.

Beberapa tahapan dilakukan untuk menyelesaikan permasalahan tersebut. Pertama tahap analisa sampai perencanaan dan design perangkat hardware maupun software. Berikutnya dibuatlah simulasi dengan tools jaringan sehingga dapat di amati akses dan proses yang akan di buat. Pada tahap imlementasi yang dilakukan adalah melakukan pengujian ketika *client* mengakses situs yang terlarang atau pemakaian kapasitas *bandwidth* yang berlebihan, maka sistem dapat langsung memblokir akses ke situs tersebut, tahapan berikutnya untuk mempercepat akses internet digunakan sistem *urlcaching*. Sistem *url caching* mampu melakukan penyimpanan *url caching* terhadap situs yang telah dibuka, sehingga akses internet ke situs yang sebelumnya pernah diakses akan lebih cepat untuk yang berikutnya.

Kata kunci: *Proxy, Manajemen Bandwidth, Filtering*

1 .PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan internet harus diiringi efisiensi, kecepatan, dan keunggulan dari sisi teknologi terutama penghematan *bandwidth* yang telah diberikan oleh ISP (*Internet Service Provider*). Kecepatan dalam memproses pencarian suatu alamat pada *browsers* sangat diperlukan terutama digunakan untuk hal-hal yang bersifat positif, seperti : mencari referensi untuk penulisan sebuah karya tulis, mencari informasi berita terkini lewat situs-situs berita online, dll.

Proxy server sebagai sebuah solusi untuk mampu menyelesaikan permasalahan kelambatan akses internet, dibutuhkan sebuah komputer *server* yang mampu menampung banyaknya komputer *client*. Sehingga semakin banyak *client*, maka semakin besar pula kapasitas komputer *server* tersebut. Adapun komponen yang sangat penting yang harus diperhatikan dalam sebuah komputer *server* adalah *memory*, karena pada

dasarnya *manajemen proxy* adalah *sharing memory* antara komputer *server* dengan komputer *client*.

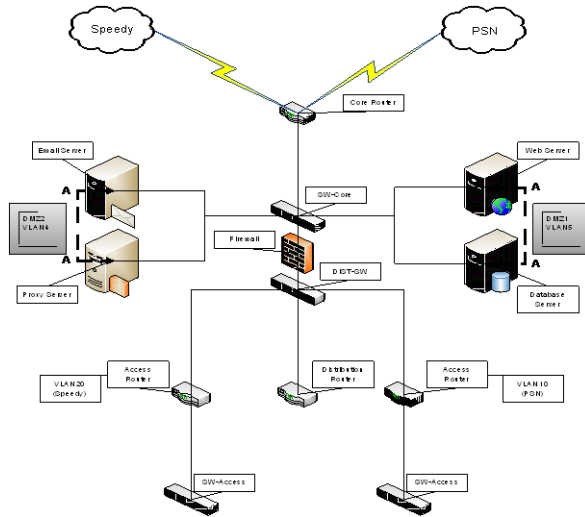
Memory komputer *server* harus disesuaikan dengan banyaknya jumlah komputer *client* yang menggunakan *proxy*. Adapun *memory* minimal komputer *server* adalah 1 Giga Byte untuk mengatasi masalah pada komputer *client* yang sering tersendat-sendat saat menjalankan aplikasi.

Pada STMIK Bani Saleh yang telah menerapkan sistem *sharing* data dengan menggunakan internet, maka diperlukan suatu *manajemen* dalam penggunaan internet sehingga dapat membatasi penggunaan internet yang kadang disalah gunakan oleh pengguna dengan mengakses situs-situs tertentu yang telah dilarang oleh pemerintah atas dasar pertimbangan keamanan jaringan tersebut dan penggunaan internet positif.

1.2. Topologi Jaringan yang dipergunakan.

Gambar di bawah ini menunjukkan topologi global yang akan dibuat dan menjelaskan gambaran topologi yang

akan dibangun dengan *proxy* sebagai gambaran dalam pengerjaan pembangunan infrastruktur di lokasi penelitian. Tujuannya tak lain untuk mempermudah atau sebagai pedoman dasar yang akan diberikan dari perancang infrastruktur ke pengguna.



Gambar 1 : Topologi global jaringan STMIK Bani Saleh

2. TINJAUAN PUSTAKA

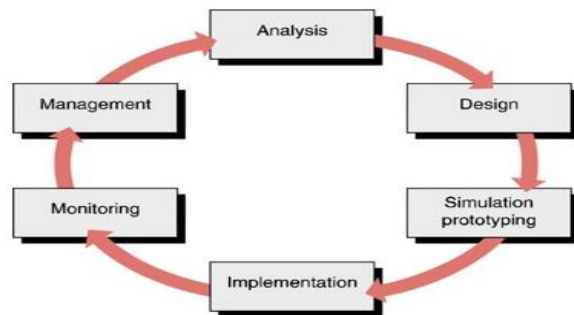
2.1 LANDASAN TEORI

2.1.1. Bandwidth

Bandwidth adalah kapasitas atau daya tampung kabel *ethernet* agar dapat dilewati *traffic* paket data dalam jumlah tertentu. *Bandwidth* juga bisa berarti jumlah konsumsi paket data per satuan waktu dinyatakan dengan satuan *bit per second* [bps]. *Bandwidth internet* disediakan oleh *Internet Service provider* (ISP) dengan jumlah tertentu tergantung sewa pelanggan. (Kustanto, 2008). Dengan *Quality Of Service* (QoS) dapat diatur agar *user* tidak menghabiskan *bandwidth* yang disediakan oleh *provider*. Istilah *bandwidth* muncul dari bidang teknik elektro, dimana *bandwidth* mempresentasikan jarak keseluruhan atau jangkauan diantara sinyal tertinggi dan terendah pada kanal (*band*) komunikasi. Pada dasarnya *bandwidth* mempresentasikan kapasitas dari koneksi, semakin tinggi kapasitas, maka umumnya akan diikuti oleh kinerja yang lebih baik, meskipun kinerja keseluruhan juga tergantung pada faktor-faktor lain, misalnya *latency* yaitu waktu tunda antara masa sebuah perangkat meminta akses ke jaringan dan masa perangkat itu member izin untuk melakukan transmisi.

2.1.2. Network Development Life Cycle (NDLC)

Perancangan infrastruktur yang berupa pengembangan teknologi jaringan komputer merupakan sebuah solusi untuk mengatasi masalah yang dihadapi di lokasi penelitian. Namun tetap harus mempunyai landasan dalam perancangan infrastruktur pengembangan yang akan dilakukan pada penelitian ini. Metode pengembangan yang digunakan adalah NDLC (*Network Development Life Cycle*). Berikut gambar tahapan pada NDLC :



Gambar 2.1 :Metode NDLC (*Network Development Life Cycle*)

(Sumber : Applied Data Communications, A business-Oriented Approach, James E. Goldman, Philips T. Rawles, Third Edition, 2001, John Wiley & Sons : 470)

2.1.5. Proxy Server

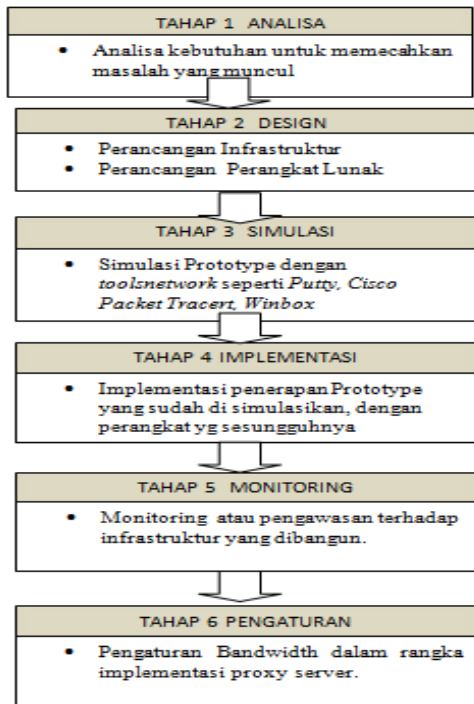
Proxy server adalah sebuah *server* atau program komputer yang berperan sebagai penghubung antara suatu komputer dengan jaringan *internet*. Atau dalam kata lain, *server proxy* adalah suatu jaringan yang menjadi perantara antara jaringan lokal dan jaringan *internet*. *Proxy server* dapat berupa suatu sistem komputer ataupun sebuah aplikasi yang bertugas menjadi *gateway* atau pintu masuk yang menghubungkan komputer kita dengan jaringan luar.

Cara kerja *proxy server* sebenarnya sangat sederhana. Ketika seorang pengguna layanan *proxy* meminta berkas, *files*, sambungan atau sumberdaya dari *public server*, maka *proxy server* meneruskannya ke internet seolah-olah *proxy* tersebut yang meminta. Dan ketika *proxy server* mendapatkan apa yang di minta pengguna, dia memberikan respon kepada pengguna seolah-olah dia adalah *public server* (internet). Dengan menggunakan *proxy*, maka identitas komputer menjadi tersembunyi. *Proxy* ini pada umumnya digunakan untuk kegiatan menyembunyikan identitas atau untuk menghindari pemblokiran akses ke suatu *server*.

3. METODE PENELITIAN

3.1 Tahapan Penelitian

Penelitian ini akan di laksanakan dengan tahapan sebagai berikut :



3.2. Tahapan Dalam Pelaksanaan Penelitian.

A. *Analysis (Analisa)*

Tahap awal ini dilakukan adalah analisa kebutuhan dari infrastruktur yang akan dibangun mulai dari *hardware* dan *software*, analisa permasalahan yang muncul dilokasi penelitian, analisa keinginan *user* sebagai pengguna atau pemilik infrastruktur, dan analisa topologi atau jaringan yang sudah ada saat ini di lokasi penelitian. Metode yang biasa digunakan pada tahap ini diantaranya:

1. Wawancara, dilakukan dengan pihak terkait melibatkan dari struktur manajemen atas sampai ke *level* bawah agar mendapatkan data yang konkrit dan lengkap.
2. *Survey* langsung kelapangan, pada tahap analisis juga dilakukan *survey* langsung kelapangan untuk mendapatkan hasil sesungguhnya dan gambaran seutuhnya sebelum masuk ke tahap desain.

3. Membaca manual atau *blueprint* dokumentasi, pada *analysis* awal ini juga dilakukan dengan mencari informasi dari jurnal penelitian, artikel di internet juga mengikuti perkembangan di masyarakat mengenai penggunaan *proxy* sebagai sistem *manajemen bandwidth*.
4. Menelaah setiap data yang didapat dari data-data sebelumnya, dalam hal ini data diperoleh dengan cara merumuskan hasil wawancara selanjutnya dilakukan analisa data tersebut untuk masuk ke tahap berikutnya.

Adapun yang bisa menjadi pedoman dalam mencari data pada tahap *analysis* ini adalah:

1. *User/people*, siapa yang menjadi admin sistemnya, kegiatan yang sering dilakukan, peta politik yang ada dan *level* teknis *user*;
2. Media H/W dan S/W, peralatan yang ada, status jaringan, ketersediaan perangkat keras dan lunak yang dapat diakses dari peralatan, serta aplikasi yang akan digunakan.
3. Data, jumlah client yang akan dipasang, jumlah inventaris sistem, sistem keamanan yang sudah ada dan area yang akan dipantau.
4. *Network*, konfigurasi jaringan, volume *trafik* jaringan, protokol, *network monitoring* yang ada saat ini, harapan dan rencana pengembangan ke depan;
5. Perencanaan fisik, seperti masalah listrik, tata letak, ruang khusus, sistem keamanan yang ada, dan kemungkinan akan pengembangan ke depan.

B. *Design (Perancangan)*

Dari data-data yang didapatkan dari tahap sebelumnya, tahap *Design* ini akan membuat gambar *design* topologi jaringan yang akan dibangun, diharapkan dengan gambar ini akan memberikan gambaran seutuhnya dari kebutuhan yang ada. *Design* bisa berupa *design* struktur topologi, *design* akses data, *design* tata layout alat, dan sebagainya yang akan memberikan gambaran jelas tentang infrastruktur yang akan dibangun. Biasanya hasil dari *design* berupa: Gambar-gambar topologi (*server farm, firewall, datacenter, storages, lastmiles*, perkabelan, titik akses dan sebagainya).

C. *Simulation Prototype (Simulasi)*

Setelah tahap *design* selesai, selanjutnya adalah mensimulasikan desain dengan cara membuat dalam

bentuk simulasi dengan bantuan *tools* khusus di bidang *network* seperti *Putty*, *Cisco Packet Tracer*, *MS. Office Visio 2010*, *Winbox*, dan sebagainya, hal ini dimaksudkan untuk melihat kinerja awal dari *network* yang akan dibangun dan sebagai bahan presentasi dan *sharing* dengan *team work* lainnya.

D. *Implementation* (Penerapan)

Setelah proses simulasi, masuk tahap selanjutnya yaitu implementasi. Dalam implementasinya, semua yang telah direncanakan dan di *design* sebelumnya akan diterapkan semua. Implementasi merupakan tahapan yang sangat menentukan dari berhasil / gagal nya infrastruktur yang akan dibangun dan ditahap inilah tim akan diuji dilapangan untuk menyelesaikan masalah teknis dan non teknis.

E. *Monitoring* (Pengawasan)

Setelah implementasi, selanjutnya tahapan monitoring yang merupakan tahapan yang penting, agar jaringan komputer dan komunikasi dapat berjalan sesuai dengan keinginan dan tujuan awal dari *user* pada tahap awal analisis, maka perlu dilakukan kegiatan monitoring atau pengawasan terhadap infrastruktur yang dibangun. *Monitoring* bisa berupa melakukan pengamatan pada:

1. Infrastruktur baik *hardware* dan *software* dengan mengamati semua kondisi kehandalan atau *reliability* sistem yang telah dibangun (*Reliability = performance + security*).
2. Memperhatikan jalannya paket data di jaringan (*pewaktuan, latency, peektime, troughput*);
3. Metode yang digunakan untuk mengamati kondisi jaringan dan komunikasi secara umum secara terpusat atau tersebar;
4. Pendekatan yang paling sering dilakukan adalah pendekatan *Network Management*. Dengan pendekatan ini banyak perangkat baik yang lokal dan tersebar dapat dimonitor secara utuh.

F. *Management* (Pengaturan)

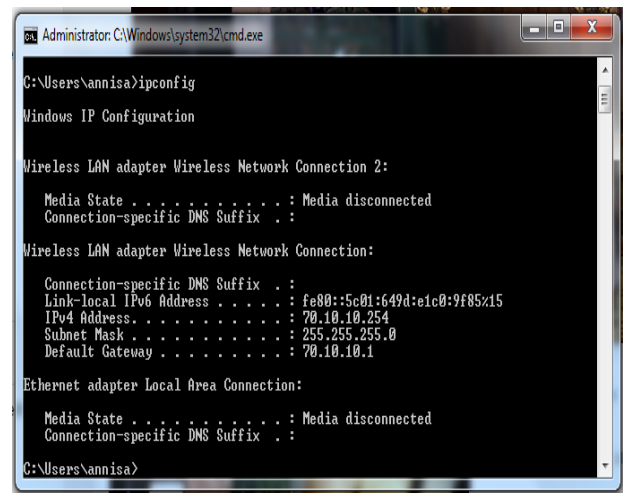
Manajemen atau pengaturan, salah satu yang menjadi perhatian khusus adalah masalah *policy* atau kebijakan, kebijakan perlu dibuat untuk membuat / mengatur agar sistem yang telah dibangun dan berjalan dengan baik dapat berlangsung lama. *Policy* akan sangat tergantung dengan kebijakan tingkat *management* dan strategi bisnis dari lokasi penelitian. Infrastruktur yang

dibangun sebisa mungkin harus dapat mendukung strategi bisnis lokasi penelitian.

4. HASIL PEMBAHASAN PENELITIAN

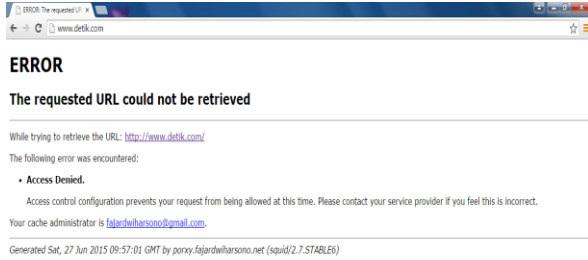
4.1. Hasil Penelitian

Pada tahapan ini akan dijelaskan mengenai hasil dari *implementasi* sistem, didalam penerapan *otentikasi* dengan *proxy server*, hasil yang didapat yaitu situs *www.detik.com* telah di *blocking* oleh *proxy server*. Apabila *user* tetap ingin masuk ke dalam situs *www.detik.com*, maka *user* harus terlebih dahulu melapor / menghubungi *administrator* yaitu *fajardwiharsono@gmail.com* untuk konfirmasi lebih lanjut.



Gambar 4.1 : Tampilan *cmd* untuk mengecek *ip* yang digunakan pada *client*

Gambar di atas adalah cara untuk mengecek *ip address* yang digunakan oleh *user* di dalam komputer *client*, yaitu 70.10.10.254 dengan *netmask* 255.255.255.0 dan *gateway* 70.10.10.1. Melihat *ip address* yang digunakan oleh *client*, maka *client* diizinkan mengakses internet oleh *proxy server* dengan ketentuan tertentu.



Gambar 4.2 : Tampilan situs yang diblocking oleh proxy server

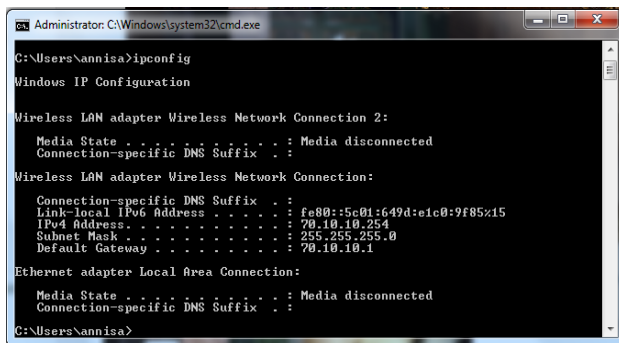
Gambar di atas menampilkan salah satu situs yang telah di-blokir oleh proxy yaitu berupa www.detik.com. Apabila user ingin masuk ke dalam situs tersebut, maka user harus menghubungi administrator terlebih dahulu yaitu fajardwiharsono@gmail.com.

4.2. Ujicoba Hasil Penelitian

4.2.1. Proxy Server

Langkah-langkah untuk melakukan ujicoba sistem proxy server dengan menggunakan squid yang telah diinstall di dalam sistem operasi linux OpenSUSE 12.1 adalah sebagai berikut:

- a. User harus mengubah ip address yang ada di dalam komputer user (komputer client) dengan ip address yang telah didaftarkan sebagai ip address yang memiliki hak akses ke dalam internet.



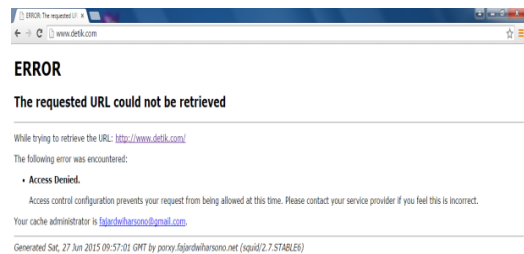
Gambar 4.6 : Tampilan ip address pc client sudah diubah

- b. Setelah itu user yang telah mengubah ip address yang memiliki hak akses ke dalam internet baru dapat menggunakan internet.



Gambar 4.7 : Tampilan browser pc client yang dapat akses internet

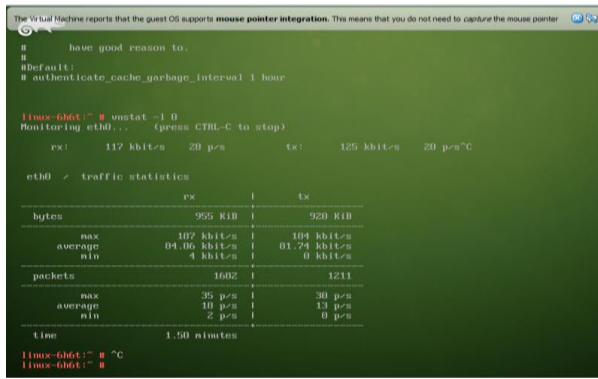
- c. Untuk situs-situs tertentu yang sudah didaftarkan sebagai situs terlarang untuk diakses oleh user, maka akan terdapat sebuah pesan peringatan (notification) berupa tulisan bahwa user telah mengakses situs yang dilarang untuk diakses.



Gambar 4.8 : Tampilan notification pada situs yang telah diblokir oleh proxy server.

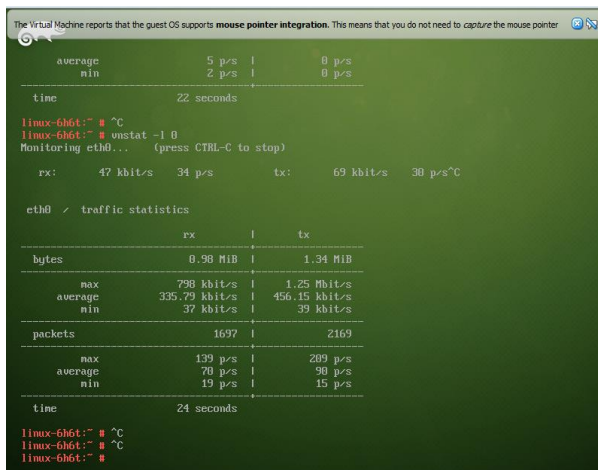
4.4.1. Bandwidth Hits Local Up Menggunakan VNSTAT

Dengan adanya vnstat di dalam linux OpenSUSE 12.1, maka administrator dapat dengan mudah memonitoring/memantau traffic bandwidth yang ada pada jaringan internet dan bisa mengetahui perbedaan kecepatan akses data internet dibandingkan sebelum menggunakan proxy transparent dengan menggunakan perintah di dalam terminal linux OpenSUSE 12.1.



Gambar 4.9 : Tampilan pengujian monitoring kecepatan akses user membuka situs

<https://www.youtube.com> sebelum menggunakan proxy

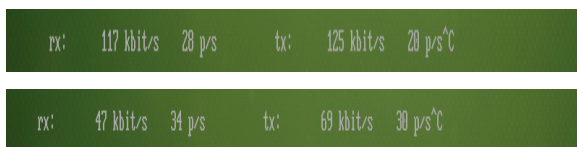


Gambar 4.10 : Tampilan pengujian monitoring kecepatan akses user membuka situs

<https://www.youtube.com> setelah menggunakan proxy

a. Perbandingan nilai rx, tx dan packets

Dari gambar 4.9 dan gambar 4.10, Maka didapatkan data sebagai berikut:



Gambar 4.11 : Perbandingan nilai rx, tx dan packets

Dari gambar perbandingan nilai upload packets atau disebut juga dengan rx, download packets atau disebut

juga dengan tx dan packets pada saat akses pada gambar 4.11, maka dapat dijelaskan sebagai berikut :

Tabel 4.1 : Hasil perbandingan rx, tx dan packets

Response Time	Rx	tx	Packets
Pengujian Sebelum	117 Kbit/s	125 Kbit/s	28 P/s
Pengujian Setelah	47 Kbit/s	69 Kbit/s	32 P/s
Rata – rata	82 Kbit/s	97 Kbit/s	30 P/s

Dari tabel 4.1 yang berada di atas dapat disimpulkan bahwa terjadi perbedaan nilai kecepatan akses baik nilai upload packets atau disebut juga dengan rx, download packets atau disebut juga dengan tx dan packets sebelum menggunakan dan setelah menggunakan proxy. Dari tabel di atas juga kita dapat menghasilkan rata-rata untuk upload packets atau disebut juga dengan rx, download packets atau disebut juga dengan tx dan packets yang diakses oleh user. Rata-rata untuk nilai upload packets atau disebut juga dengan rx = 82 Kbit/s, download packets atau disebut juga dengan tx = 97 Kbit/s dan untuk packets = 30 P/s.

Dari penjelasan di atas, dapat dibuatkan Nilai efektifitas upload packets atau rx (Er_x) yang didapatkan dengan rumus :

$$Er_x = \frac{(rx \text{ Before} - rx \text{ After})}{(rx \text{ Before})} \times 100\%$$

$$Er_x = \frac{117 - 47}{117} \times 100\%$$

$$Er_x = 59.83\%$$

Dari penjelasan di atas, dapat dibuatkan Nilai efektifitas download packets atau tx (Et_x) yang didapatkan dengan rumus :

$$Et_x = \frac{(tx \text{ Before} - tx \text{ After})}{(tx \text{ Before})} \times 100\%$$

$$Et_x = \frac{125 - 69}{125} \times 100\%$$

$$Et_x = 44.80\%$$

b. Perbandingan nilai rx bytes

```

rx
-----
bytes                955 KiB
-----
max                  187 kbit/s
average              84.86 kbit/s
min                   4 kbit/s
    
```

Dari Gambar 4.9 dan Gambar 4.10 juga didapatkan hasil perbandingan nilai bytes untuk upload atau disebut juga dengan rx bytes.

```

rx
-----
bytes                0.98 MiB
-----
max                  798 kbit/s
average              335.79 kbit/s
min                   37 kbit/s
    
```

Gambar 4.13 : Perbandingan nilai rx bytes

Dari gambar perbandingan nilai bytes untuk upload atau disebut juga dengan rx bytes pada saat akses oleh user pada gambar 4.13, maka dapat dijelaskan sebagai berikut :

Tabel 4.2 : Hasil perbandingan nilai rx bytes

Response Time	Minimum rx	Maximum rx	Average rx
Pengujian Sebelum	4 Kbit/s	187 Kbit/s	84.86 Kbit/s
Pengujian Setelah	37 Kbit/s	798 Kbit/s	335.79 Kbit/s
Rata – rata	20.5 Kbit/s	492.5 Kbit/s	210.32 Kbit/s

Dari tabel 4.2 yang berada di atas dapat disimpulkan bahwa terjadi perbedaan nilai bytes untuk upload atau disebut juga dengan rx bytes yang diakses oleh user baik nilai *minimum*, *maximum* dan *average* sebelum menggunakan dan setelah menggunakan proxy. Dari tabel di atas juga kita dapat menghasilkan rata-rata untuk *minimum* = 20.5 Kbit/s, *maximum* = 492.5 Kbit/s dan *average* = 210.32 Kbit/s.

c. Perbandingan nilai tx bytes

```

tx
-----
920 KiB
-----
184 kbit/s
81.74 kbit/s
0 kbit/s
-----
bytes
-----
max
average
min
-----
bytes
-----
max
average
min
    
```

Gambar 4.15 : Perbandingan nilai tx bytes

Dari gambar perbandingan nilai bytes untuk download atau disebut juga dengan tx bytes pada saat akses oleh user pada gambar 4.15, maka dapat dijelaskan sebagai berikut :

Tabel 4.3 : Hasil perbandingan nilai tx bytes

Response Time	Minimum tx	Maximum tx	Average tx
Pengujian Sebelum	0 Kbit/s	184 Kbit/s	81.74 Kbit/s
Pengujian Setelah	39 Kbit/s	1250 Kbit/s	456.15 Kbit/s
Rata – rata	19.5 Kbit/s	717 Kbit/s	268.94 Kbit/s

Dari tabel 4.3 yang berada di atas dapat disimpulkan bahwa terjadi perbedaan nilai nilai bytes untuk download atau disebut juga dengan tx bytes yang diakses oleh user baik nilai *minimum*, *maximum* dan *average* sebelum menggunakan dan setelah menggunakan proxy. Dari tabel di atas juga kita dapat menghasilkan rata-rata untuk *minimum* = 19.5 Kbit/s, *maximum* = 717 Kbit/s dan *average* = 268.94 Kbit/s.

d. Perbandingan nilai rx packets

Dari Gambar 4.9 dan Gambar 4.10 juga didapatkan hasil perbandingan nilai kecepatan untuk upload packets atau disebut juga dengan rx packets.

```

packets                1682
-----
max                    35 p/s
average                18 p/s
min                     2 p/s
-----
packets                1697
-----
max                    139 p/s
average                70 p/s
min                     19 p/s
    
```

Gambar 4.17 : Perbandingan nilai rx packets

Dari gambar perbandingan nilai kecepatan untuk *upload packets* atau disebut juga dengan *rx packets* pada saat akses oleh *user* pada gambar 4.17, maka dapat dijelaskan sebagai berikut :

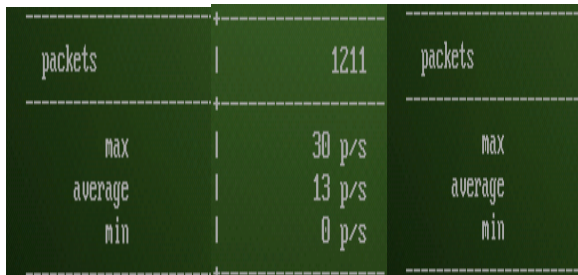
Tabel 4.4 : Hasil perbandingan nilai *rx packets*

Response Time	Minimum rx	Maximum rx	Average rx
Pengujian Sebelum	2 P/s	35 P/s	18 P/s
Pengujian Setelah	19 P/s	139 P/s	78 P/s
Rata – rata	10.5 P/s	87 P/s	48 P/s

Dari tabel 4.4 yang berada di atas dapat disimpulkan bahwa terjadi perbedaan nilai kecepatan untuk *upload packets* atau disebut juga dengan *rx packets* yang diakses oleh *user* baik nilai *minimum*, *maximum* dan *average* sebelum menggunakan dan setelah menggunakan *proxy*. Dari tabel di atas juga kita dapat menghasilkan rata-rata untuk *minimum* = 10.5 P/s, *maximum* = 87 P/s dan *average* = 48 P/s.

e. Perbandingan nilai *tx packets*

Dari Gambar 4.9 dan Gambar 4.10 juga didapatkan hasil perbandingan nilai kecepatan untuk *download packets* atau disebut juga dengan *tx packets*.



Gambar 4.19 : Perbandingan nilai *tx packets*

Dari gambar perbandingan nilai kecepatan untuk *download packets* atau disebut juga dengan *tx packets* pada saat akses oleh *user* pada gambar 4.19, maka dapat dijelaskan sebagai berikut :

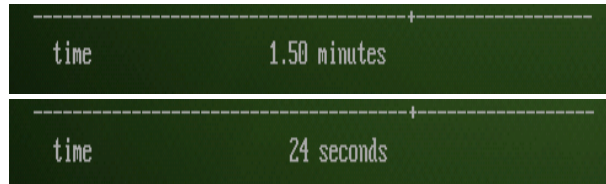
Tabel 4.5 : Hasil perbandingan nilai *tx packets*

Response Time	Minimum tx	Maximum tx	Average tx
Pengujian Sebelum	0 P/s	30 P/s	13 P/s
Pengujian Setelah	15 P/s	289 P/s	90 P/s
Rata – rata	17.5 P/s	159.5 P/s	51.5 P/s

Dari tabel 4.5 yang berada di atas dapat disimpulkan bahwa terjadi perbedaan nilai *tx packets* yang diakses oleh *user* baik nilai *minimum*, *maximum* dan *average* sebelum menggunakan dan setelah menggunakan *proxy*. Dari tabel di atas juga kita dapat menghasilkan rata-rata untuk *minimum* = 17.5 P/s, *maximum* = 159.5 P/s dan *average* = 51.5 P/s.

f. Perbandingan waktu kecepatan akses

Dari Gambar 4.9 dan Gambar 4.10 juga didapatkan hasil perbandingan waktu kecepatan untuk akses situs yang dituju oleh *user*.



Gambar 4.21 : Perbandingan waktu akses situs yang dituju

Dari gambar perbandingan waktu kecepatan untuk akses situs yang dituju oleh *user* seperti pada gambar 4.21, maka dapat dijelaskan sebagai berikut :

Tabel 4.6 : Hasil perbandingan waktu akses situs yang dituju

Response Time	Waktu Akses Situs
Pengujian Sebelum	90 seconds
Pengujian Setelah	24 seconds
Selisih waktu	57 seconds

Dari table 4.6 yang berada di atas dapat disimpulkan bahwa *response time* pada saat sebelum menggunakan *proxy transparent* adalah **90 seconds** sedangkan untuk *response time* pada saat setelah menggunakan *proxy transparent* adalah **24 seconds**. Sehingga didapatkan hasil berupa nilai *response time* untuk selisih waktu yang dibutuhkan untuk mengakses situs <http://www.youtube.com> adalah **57 seconds**.

Dari penjelasan di atas, dapat dibuatkan Nilai efektifitas *time* (ET) yang didapatkan dengan rumus :

$$Et = \frac{(Access\ Before - Access\ After)}{(Access\ Before)} \times 100\%$$

$$Et = \frac{90 - 24}{90} \times 100\%$$

$E_t = 73.33\%$

Response time ini menunjukkan bahwa waktu yang dibutuhkan oleh program aplikasi squid bertambah, dan konfigurasi *firewall filter rule* berjalan sesuai yang diharapkan.

DAFTAR PUSTAKA

- Bennett, Elisabeth E. 2014 "How an Intranet Provides Opportunities for Learning Organizational Culture Implications for Virtual HRD." *Advances in Developing Human Resources* : 1523422314532093.
- James E. Goldman, Philips T. Rawles, Third Edition, 2001, *Applied Data Communications, A business-Oriented Approach*, , John Wiley & Sons
- Towidjojo, Rendra and Muhammad Eno Farhan. "*Router Mikrotik Implementasi Wireless LAN Indoor*". *Jasakom : Bandung, 2015*.
- Pratama, I Putu Agus Eka. "Handbook Jaringan Komputer Teori dan Praktik Berbasis Open Source". *Informatika : Bandung, 2014*.
- Lukas, Jonathan. "Jaringan Komputer". *Graha Ilmu : Yogyakarta, 2006*.
- Wagito. "Jaringan Komputer Teori dan Implementasi Berbasis Linux". *Gava Media : Yogyakarta, 2015*.
- Sofana, Iwan, "Membangun Jaringan Komputer, Mudah Membuat Jaringan Komputer (Wire&Wireless) Untuk Pengguna Windows dan Linux". *Informatika : Bandung, 2013*.
- Towidjojo, Rendra. "Mikrotik Kung Fu Kitab 3 : Kitab Manajemen Bandwidth". *Jasakom : Bandung, 2014*.
- Hardana and Ino Irvantino. "*Konfigurasi Routerboard Mikrotik RB-750*". *Andi : Yogyakarta, 2015*.
- Kustanto & Daniel T Saputro. "*Belajar Jaringan Komputer Berbasis Mikrotik OS*". *Gava Media : Yogyakarta, 2015*.