

PERANCANGAN TRANSPARENT PROXY SERVER CLEAROS DI JARINGAN HOTSPOT INSTITUT TEKNOLOGI PADANG

Oleh:

IndraWarman¹⁾, Rahmat Hidayat²⁾

¹⁾Dosen Jurusan Teknik Informatika

²⁾Mahasiswa Teknik Informatika

Fakultas Teknologi Industri, Institut Teknologi Padang

Abstract

Internet technology has become an important part in many areas of life. Internet technology allowing users to get the latest information from around the world. For an institution or company that wants to have a internet connection, it must lease bandwidth to ISP's that serve the internet services. But, the price of bandwidth is quite expensive, so not all people can use the internet with convenient. Aim of this research building a proxy server in order to overcome the problem of bandwidth availability. This research was an experimental about proxy server, using ClearOS operating system. The proxy applications that used is lusca squid 14942 version. Squid analyzer is used to see the performance statistics of the proxy server. In this research, the proxy is able to save on bandwidth usage, and increase the speed of internet connections at Padang Institute of Technology hotspot network reach 10 to 30 times faster. In this research, not all objects on the internet can be cached by proxy.

Keywords : Internet, Proxy, ClearOS, Squid, Lusca

1. Pendahuluan.

Teknologi *internet* telah menjadi bagian penting dari berbagai bidang kehidupan. Teknologi *internet* memudahkan pengguna untuk mendapatkan informasi terkini dari seluruh penjuru dunia. Dan banyak lagi kemudahan yang bisa didapatkan dari adanya teknologi *internet*. Agar koneksi *internet* tetap stabil meski jumlah pengguna yang semakin banyak, maka diperlukan penambahan *bandwidth*. Tetapi hal ini juga akan berpengaruh pada biaya yang dikeluarkan semakin besar. Salah satu solusi lain untuk meningkatkan kecepatan akses *internet* adalah dengan *proxy server*.

Proxy akan mengambil file atau objek-objek dari *internet* yang diakses oleh pengguna dan menyimpannya ke *server proxy*. Pengguna yang ingin mengakses situs yang pernah diakses, tidak harus menuju ke *server* utama, dan akan mengambil objek yang telah disimpan di *proxy server*. Hal ini dapat meningkatkan kecepatan dalam mengakses situs atau objek tersebut serta dapat menghemat *bandwidth*. Dari uraian diatas, maka penelitian ini mengambil topik untuk dijadikan sebagai laporan tugas akhir dengan judul “**Perancangan Transparent**

Proxy Server ClearOS di Jaringan Hotspot Institut Teknologi Padang”.

2. Landasan Teori.

Konsep Dasar *Proxy*.

Proxy dapat dipahami sebagai pihak ketiga yang berdiri ditengah-tengah antara kedua pihak yang saling berhubungan dan berfungsi sebagai perantara. Dengan begitu pihak pertama dan pihak kedua tidak secara langsung berhubungan, akan tetapi masing-masing berhubungan dengan perantara, yaitu *proxy*. *Proxyserver* mempunyai 3 fungsi utama yaitu :

1. Connection Sharing

Sebagai perantara antara pengguna dan *server – server* di *internet*.

2. Filtering

Filtering digunakan untuk melindungi jaringan lokal dari serangan atau gangguan yang berasal dari jaringan *Internet* dengan cara melakukan filter atas paket yang lewat dari dan ke jaringan

3. Caching

Caching Proxy menyimpan data yang pernah diminta oleh klien ke suatu tempat penyimpanan di *server*.

TransparentProxy

Salah satu kompleksitas *proxy* pada level aplikasi adalah pada sisi pengguna harus dilakukan konfigurasi yang spesifik untuk bisa menggunakan layanan dari suatu *proxy server*. Tetapi jika pengguna tidak ingin melakukan konfigurasi untuk menggunakan layanan *proxy*, maka *proxy server* harus berjalan secara transparan terhadap pengguna. *Transparent proxy* berguna untuk memaksa user menggunakan *proxy server*, tanpa harus melakukan konfigurasi. Cara kerja *transparent proxy* dengan membelokkan arah dari paket-paket untuk suatu aplikasi tertentu, dengan menggunakan aturan pada *firewall*.

Squid.

Squid merupakan salah satu aplikasi *cache* yang berlisensi GPL (*GNU Public License*) dan merupakan implementasi *proxy server* yang dapat menyimpan objek pada *cache*. Setelah dijalankan (*running*), squid akan mendengarkan (*listen*) pada port yang telah ditentukan dan melayani *request* pada port ini. Setelah memperoleh objek yang diminta oleh *client*, squid akan menyimpan objek tersebut ke dalam *harddisk*. Sehingga apabila terdapat permintaan terhadap objek yang sama di lain waktu, objek tersebut akan diambil dari *harddisk*.

Lusca.

Lusca, adalah *proxy* yang sangat populer di kalangan masyarakat Indonesia. Lusca bertujuan untuk memperbaiki kekurangan-kekurangan dalam basis kode sementara squid, serta menjaga kestabilan fungsionalitas dan stabilitas squid. Peningkatan kinerja yang lebih cepat karena memperbaiki kelemahan squid *proxy*. Menangani *local cache content dynamic web proxy / cache platform* dengan kinerja lebih stabil, fitur lebih banyak dan skalabilitas yang dinamis. Lusca mendukung *cpu single core* ataupun *multicore*. Kelebihan lainnya adalah Lusca di buat berbasis squid 2.7 yang secara default sudah memiliki pengaturan ToS (ZPH) dan dapat ditambahkan url rewriter.

Refresh Pattern

Refresh_pattern adalah tag dalam *Squid.conf* yang berfungsi menentukan usia objek didalam *cache* dan dengan melakukan beberapa tuning, *refresh_pattern* dapat meng-*override* (mengganti atau memanipulasi) validitas objek didalam *cache*. Fungsi utama *refresh_pattern* adalah mengatur bagaimana squid akan menangani objek yang berada didalam *cache*. *Refresh_pattern* dapat digunakan untuk menentukan atau memanipulasi validitas object (*fresh* atau *stale*) yang ada didalam *cache* squid sehingga nilai HIT dapat menjadi lebih maksimal.

Squid Analyzer

Squid analyzer merupakan salah satu tools penganalisa log milik Squid yang dapat menyajikan statistik lengkap meliputi waktu, hit, byte, pengguna, jaringan, hingga top url dan top domain yang paling sering diakses. Keunggulan yang ditawarkan dari *squid analyzer* diantaranya adalah tidak perlu adanya database apapun untuk dapat menjalankannya, dan juga dengan tampilan yang cukup *user friendly* dibandingkan SARG ataupun Mysar.

ClearOS

ClearOS adalah sistem operasi yang di kostumasi khusus untuk keperluan *server* yang dibuat oleh *clearfoundation*. Berbasis Linux Red Hat Enterprise 5, menjadikan ClearOS memiliki source base yang kuat dan stabil untuk dijalankan sebagai *server*. Sistem operasi ini dibekali dengan banyak sekali fitur-fitur penting untuk sebuah *server*. Untuk mempermudah konfigurasi, ClearOS juga menyediakan fitur *webconfig*, yaitu konfigurasi berbasis web yang sangat *powerfull*.

3. Metodologi Jenis Penelitian.

Jenis penelitian ini adalah penelitian eksperimen, yaitu merancang sebuah *proxy server* pada jaringan *hotspot* di ITP.

Alat dan Bahan.

Alat dan bahan yang digunakan dalam pembuatan *proxy server* adalah sebagai berikut:

A. Perangkat Keras (*Hardware*)

Perangkat keras (*hardware*) yang digunakan untuk membangun *proxy server* dalam penelitian ini adalah:

1. Komputer Server
2. Memori 1.5 GB (minimal 1 GB)
3. Hard Disk 80 GB (Seagate)
4. LAN Card
5. Kabel UTP, Modem dan peralatan jaringan lainnya.

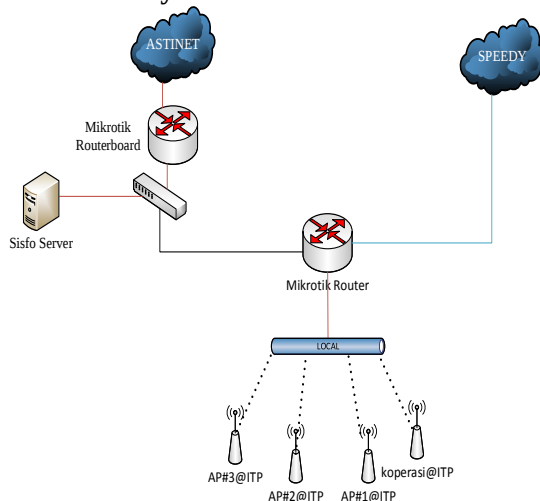
B. Perangkat Lunak (*Software*)

Perangkat lunak (*software*) yang digunakan untuk membangun *proxy server* dalam penelitian ini adalah:

1. Sistem operasi ClearOS Community.
2. Squid sebagai aplikasi *proxy server*.
3. Putty untuk remot server.
4. Mozilla Firefox atau google chrome untuk konfigurasi melalui *webconfig*.

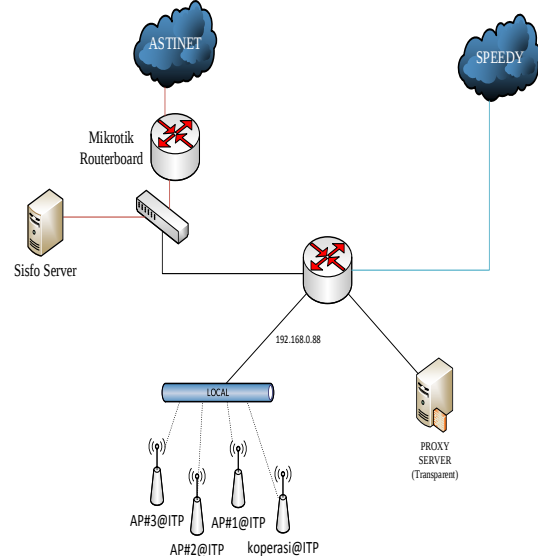
Topologi

1. Topologi Sebelum Digunakan Proxy



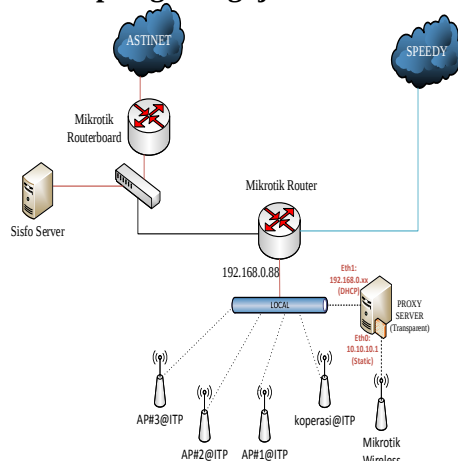
Gambar 3.1 Topologi Sebelum Digunakan Proxy.

2. Topologi Setelah Digunakan Proxy



Gambar 3.2 Topologi Setelah Digunakan Proxy.

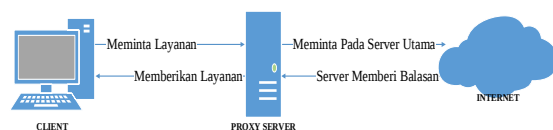
3. Topologi Pengujian



Gambar 3.3 Topologi Pengujian

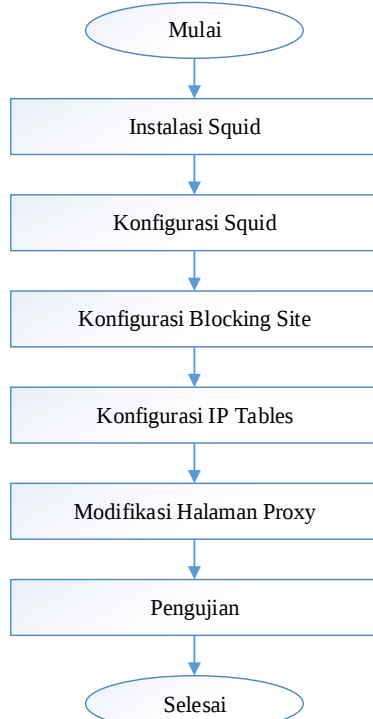
Context Diagram

Context diagram merupakan alat bantu perancangan sistem secara global yang memperlihatkan sistem secara umum dan bagian-bagian dari sub sistem yang terlibat didalam sistem secara keseluruhan, keterkaitan dan interaksi antar subsistem.



Gambar 3.4 Context Diagram

Tahapan Perancangan Proxy



Gambar 3.5 Tahapan Perancangan

4. Pembahasan

a. Pengujian Sampel Pertama

Untuk Sampel pertama, yaitu *download* sebuah file rar dari *server* indowebster.com yang berukuran kurang lebih 44 MB.

Table showing network traffic data for a download before caching:

Et	Port	Src	Dest	VLAN Id	Tx Rate	Rx Rate	Tx Pack	Rx Pack
800 (ip)		192.168.88.254	192.168.88.1		0.0 Kbps	1132 Kbps	1	2
800 (ip)		192.168.88.254	192.168.88.1		6.6 Mbps	192.4 Kbps	885	345
800 (ip)		192.168.88.254	192.168.88.1		228 Kbps	628 Kbps	8	1

Gambar 4.1 *Download* 1 sebelum objek ter-cache.

Gambar 4.1 diatas merupakan hasil *download* objek pertama sebelum objek tersebut berada di-cache. Kecepatan yang diperoleh adalah 6,6 mbps.

Table showing network traffic data for a download after caching:

Et	Port	Src	Dest	VLAN Id	Tx Rate	Rx Rate	Tx Pack	Rx Pack
800 (ip)		192.168.88.254	192.168.88.1		4.2 Kbps	1432 Kbps	1	2
800 (ip)		192.168.88.253	192.168.88.1		47.2 Mbps	346 Kbps	3905	807

Gambar 4.2 *Download* 1 setelah objek ter-cache.

Gambar 4.2 diatas adalah hasil *download* dari objek pertama setelah objek tersebut ter-cache oleh *proxy server*. Kecepatan yang didapat adalah 47 mbps.

Sampel Kedua

Sampel kedua adalah *download* file berupa pdf dari 4shared.com yang berukuran 32 MB lebih.

Table showing network traffic data for a download before caching (sample 2):

Et	Port	Src	Dest	VLAN Id	Tx Rate	Rx Rate	Tx Pack	Rx Pack
800 (ip)		192.168.88.254	192.168.88.1		2.5 Kbps	1458 Kbps	1	2
800 (ip)		192.168.88.252	192.168.133.150		875.3 Kbps	41.6 Kbps	185	94

Gambar 4.3 *Download* 2 sebelum objek ter-cache.

Pada gambar 4.3 terlihat bahwa kecepatan yang didapat saat men-*download* file pdf dari 4shared adalah 879 kbps. Objek tersebut adalah sebelum ter-cache oleh *proxy*.

Table showing network traffic data for a download after caching (sample 2):

Et	Port	Src	Dest	VLAN Id	Tx Rate	Rx Rate	Tx Pack	Rx Pack
800 (ip)		192.168.88.247	192.168.133.150		31.5 Kbps	518 Kbps	3610	1175
800 (ip)		192.168.88.252	192.168.133.150		2.8 Kbps	44.7 Kbps	207	132
800 (ip)		192.168.88.254	192.168.133.150		18.5 Kbps	330 Kbps	6336	768
800 (ip)		192.168.88.248	192.168.133.150		788 Kbps	525 Kbps	1	1

Gambar 4.4 *Download* 2 setelah objek ter-cache.

Gambar 4.4 merupakan kecepatan download objek dari 4shared setelah objek ter-*cache* oleh *proxy*. *Download* dilakukan oleh beberapa komputer secara bersamaan. Kecepatan yang diperoleh oleh masing-masing komputer berbeda-beda.

Sampel Ketiga

Sampel ketiga adalah *download* aplikasi KMPlayer dari situs resminya yaitu kmplayer.com yang berukuran 31 MB.

Et.	Port	Src.	Dest.	VLAN Id	Tx Rate	Rx Rate	Tx Pack.	Rx Pack.
800 (ip)		192.168.88.254	192.168.88.1		8.4 kbps	1456 bps	1	2
800 (ip)		192.168.88.254	125.160.18.33		400.9 kbps	9.7 kbps	43	22

Gambar 4.5 *Download* 3 sebelum objek ter-*cache*.

Gambar 4.5 diatas merupakan hasil *download* objek ketiga yaitu *kmplayer* sebelum objek tersebut berada di-*cache*. Kecepatan yang diperoleh adalah 400 kbps.

Et.	Port	Src.	Dest.	VLAN Id	Tx Rate	Rx Rate	Tx Pack.	Rx Pack.
800 (ip)		192.168.88.253	125.160.18.11		15.9 Mbps	89.2 kbps	1314	206
800 (ip)		192.168.88.254	125.160.18.33		8.8 Mbps	162.7 kbps	728	372
800 (ip)		192.168.88.247	125.160.18.11		4.7 Mbps	88.1 kbps	394	195

Gambar 4.6 *Download* 3 setelah objek ter-*cache*.

Gambar 4.6 merupakan kecepatan *download* objek dari *kmplayer.com* setelah objek ter-*cache* oleh *proxy*. *Download* dilakukan oleh beberapa komputer secara bersamaan. Kecepatan yang diperoleh oleh masing-masing komputer berbeda-beda melebihi kecepatan pertama.

Sampel Keempat

Sampel keempat adalah *download* sebuah *ebook* photoshop dari situs resminya. Kecepatan yang diperoleh saat pertama kali dilakukan *download* adalah 457 kbps seperti terlihat pada gambar 4.7 dibawah ini.

Et.	Port	Src.	Dest.	VLAN Id	Tx Rate	Rx Rate	Tx Pack.	Rx Pack.
800 (ip)		192.168.88.254	192.168.88.1		2.9 kbps	1456 bps	1	2
800 (ip)		192.168.88.252	118.96.30.44		3.6 kbps	4.4 kbps	2	2
800 (ip)		192.168.88.252	125.160.18.33		457.9 kbps	9.7 kbps	43	22

Gambar 4.7 *Download* 4 sebelum objek ter-*cache*.

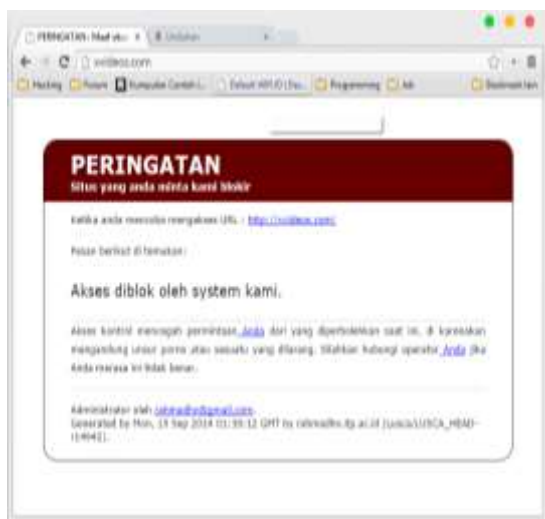
Setelah objek dari *ebook* tersebut ter-*cache*, kecepatan *download* menjadi lebih cepat seperti ditunjukkan pada gambar 4.8 dibawah.

Et.	Port	Src.	Dest.	VLAN Id	Tx Rate	Rx Rate	Tx Pack.	Rx Pack.
800 (ip)		192.168.88.254	192.168.88.1		7.1 kbps	1456 bps	1	2
800 (ip)		192.168.88.247	84.263.187.173		87.4 kbps	2.4 kbps	8	9
800 (ip)		192.168.88.253	125.160.18.113		8.7 Mbps	71.4 kbps	732	134
800 (ip)		192.168.88.254	125.160.18.33		15.4 Mbps	250.5 kbps	1274	681
800 (ip)		192.168.88.247	125.160.18.33		18.9 Mbps	334.2 kbps	1888	780

Gambar 4.8 *Download* 4 setelah objek ter-*cache*.

Sampel Kelima

Sampel kelima merupakan pengujian dari *blocking site*. Terlihat pada gambar 4.9 bahwa situs *xvideos.com* telah berhasil diblok oleh *proxy*.



Gambar 4.9 Blocking sites

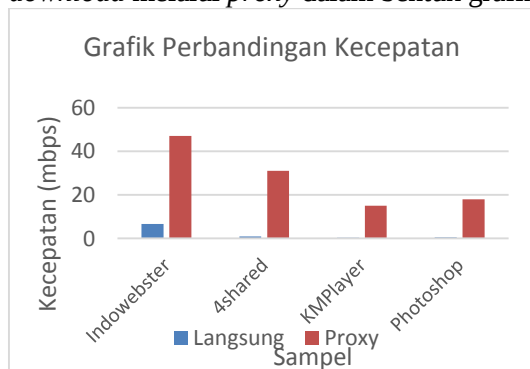
Tabel Perbandingan

Setelah pengujian yang telah dilakukan dari beberapa sampel, maka didapatkan hasil perbandingan antara kecepatan *download* sebelum dan sesudah menggunakan *proxy*. Hasil perbandingan tersebut bisa dilihat pada tabel 4.1.

Tabel 4.1 Perbandingan Kecepatan

Sampel	KecepatanDownload	
	Langsung	Proxy
Indowebster (rar)	6,6 mbps	47 mbps
4Shared (pdf)	879 kbps	31 mbps
KMPlayer (exe)	400 kbps	15 mbps
Photoshop (pdf)	457 kbps	18 mbps

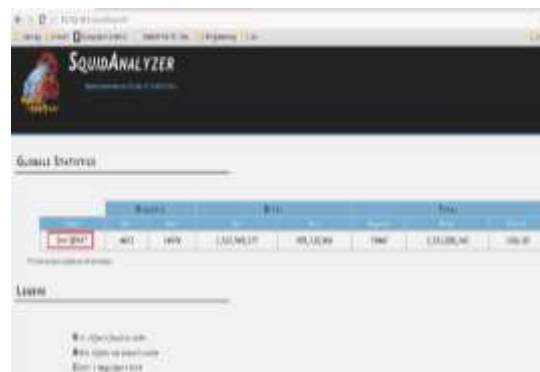
Gambar 4.10 merupakan perbandingan kecepatan *download* langsung dengan *download* melalui *proxy* dalam bentuk grafik.



Gambar 4.10 Grafik perbandingan kecepatan

Pengujian Squidanalyzer

Pengujian *squidanalyzer* digunakan untuk melihat statistik akses pengguna yang melewati *proxy*. Dari *squidanalyzer* terlihat berapa banyak objek yang berhasil ter-cache oleh *proxy*, situs apa yang paling sering dikunjungi dan lain-lain.



Gambar 4.11 Tampilan awal squidanalyzer



Gambar 4.12 Tampilan statistik cache

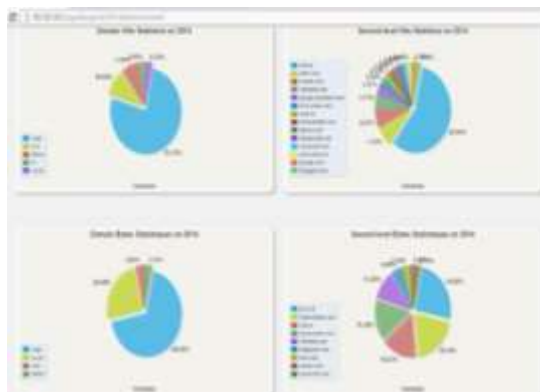


Gambar 4.13 Statistik Pengguna



Domain	IP Address	Count	First Seen	Last Seen
www.google.com	74.125.233.100	100	2014-01-01 00:00:00	2014-01-01 00:00:00
www.facebook.com	31.13.249.100	50	2014-01-01 00:00:00	2014-01-01 00:00:00
www.youtube.com	74.125.233.100	30	2014-01-01 00:00:00	2014-01-01 00:00:00
www.twitter.com	31.13.249.100	20	2014-01-01 00:00:00	2014-01-01 00:00:00
www.linkedin.com	74.125.233.100	10	2014-01-01 00:00:00	2014-01-01 00:00:00
www.instagram.com	31.13.249.100	5	2014-01-01 00:00:00	2014-01-01 00:00:00
www.pinterest.com	74.125.233.100	5	2014-01-01 00:00:00	2014-01-01 00:00:00
www.tumblr.com	31.13.249.100	5	2014-01-01 00:00:00	2014-01-01 00:00:00
www.flickr.com	74.125.233.100	5	2014-01-01 00:00:00	2014-01-01 00:00:00
www.dribbble.com	31.13.249.100	5	2014-01-01 00:00:00	2014-01-01 00:00:00
www.behance.net	74.125.233.100	5	2014-01-01 00:00:00	2014-01-01 00:00:00
www.500px.com	31.13.249.100	5	2014-01-01 00:00:00	2014-01-01 00:00:00
www.deviantart.com	74.125.233.100	5	2014-01-01 00:00:00	2014-01-01 00:00:00
www.artstation.com	31.13.249.100	5	2014-01-01 00:00:00	2014-01-01 00:00:00
www.dailymotion.com	74.125.233.100	5	2014-01-01 00:00:00	2014-01-01 00:00:00
www.vimeo.com	31.13.249.100	5	2014-01-01 00:00:00	2014-01-01 00:00:00
www.youtube.com	74.125.233.100	5	2014-01-01 00:00:00	2014-01-01 00:00:00

Gambar 4.14 Daftar URL yang ter-HIT



Gambar 4.15 Domain yang paling sering dikunjungi.

4. Penutup

a. Kesimpulan

Berdasarkan pada penjelasan dan pembahasan pada bab-bab sebelumnya, dapat diambil beberapa kesimpulan. Dari hasil yang diperoleh, *proxy squid* lusca 14942 pada sistem operasi ClearOS, mampu menghemat pemakaian *bandwidth* karena pengguna tidak mengakses objek ke internet dan hanya mengambilnya di *proxy*. Dengan pengaturan *transparent proxy*, pengguna akan dipaksa melewati *proxy* untuk terkoneksi ke internet meskipun pengguna memasukkan konfigurasi pada browsernya.

b. Saran

Berdasarkan hasil yang didapat dari penelitian yang telah dilakukan, maka dapat disarankan beberapa hal. Akan lebih baik jika menggunakan pengaturan tambahan seperti *refresh_pattern* dan *storeurl* terbaru sebagai *helper*-nya, karena dapat meningkatkan kinerja dari *squid*. Untuk pengaturan *storeurl* dan *regex pattern* kadang berubah-ubah (misalnya youtube), sehingga perlu dilakukan

perubahan pada konfigurasi *storeurl* agar *squid* lusca bisa meng-cache-nya. Untuk *monitoring squid*, bisa menggunakan aplikasi lain yang lebih baik selain aplikasi *squid analyzer* seperti *Manage Engine Firewall Analyzer*. Tetapi *Manage Engine Firewall Analyzer* merupakan produk berbayar.

Pada penelitian ini, belum ada pengujian dengan *bypass proxy*, untuk itu disarankan juga menguji dengan pengaturan *bypass proxy* dan melihat pengaruhnya terhadap *proxy server*.

Daftar Pustaka.

- Hafizh, Muhammad. 2011. *Load Balancing Dengan Metode Per Connection Classifier (PCC) Menggunakan Proxy Server Sebagai Caching Studi Kasus di Madrasah Tsanawiyah An-Nizhamiyyah Cileungsi*. Teknik Informatika. Universitas Islam Negeri Syarif Hidayatullah. Jakarta.
- Micro, Andi. 2012. *Membangun Server Dengan ClearOS 6*. ClearOS Indonesia.
- Rafiudin, Rahmat. 2008. *Squid Koneksi Anti Mogok*. Yogyakarta: Andi.
- Ramadhani, Aprista Noor. *Membangun Proxy Server Dengan Squid Di Smp Negeri 5 Kecamatan Ponorogo Menggunakan Sistem Operasi Freebsd*. Teknik Informatika. Universitas Muhammadiyah. Ponorogo.
- Saini, Kulbir. 2011. *Squid Proxy Server 3.1 Beginner's Guide*. Birmingham: Packt Publishing Ltd.
- Tjahjana, Chrisdyansyah. Diposting pada tanggal 6 Juni 2012. "Pengertian Lusa". <https://sites.google.com/site/chrisdyansyah/tutorial/squid/pengertianlusca>. Diakses pada 23 Mei 2014 pukul 22.50 WIB.
- Winarno, Sugeng. 2006. *Jaringan Komputer Dengan TCP/IP*. Bandung: Penerbit Informatika.