Jurnal Sarjana Teknik Informatika e-ISSN: 2338-5197
Volume 1 Nomor 2, Oktober 2013

OPTIMALISASI PENYIMPANAN VIDEO MENGGUNAKAN VIDEOCACHE PADA PROXY SERVER (Studi Kasus pada Warnet Janturan.Net Yogyakarta)

[1]Nunung Budi Listyawan, [2]Imam Riadi (0510088001)

[1]Program Studi Teknik Informatika
[2]Program Studi Sistem Informasi
Universitas Ahmad Dahlan
Prof. Dr. Soepomo, S.H., Janturan, Umbulharjo, Yogyakarta 55164
[1]Email:
[2]Email: imam_riadi@uad.ac.id

ABSTRAK

Peningkatan akses Internet yang tidak selalu bisa dioptimalkan oleh pengguna internet. Masalah-masalah yang timbul sebagai efek dari peningkatan akses Internet antara lain adalah menambah beban web server apalagi kalau sudah menyangkut dengan video streaming pasti berat beban diserver. Padatnya lalu lintas jaringan untuk men-transfer data/berkas, dan lamanya waktu transfer sebagai akibat padatnya lalu lintas jaringan. Salah satu cara untuk mengatasi masalah tersebut adalah menggunakan proxy server videocache.

Subjek dalam penelitian ini adalah optimalisasi penyimpanan video menggunakan videocache pada proxy server squid dan videocache sebagai aplikasinya. Penelitian ini menggunakan metode observasi, metode wawancara. Aplikasi yang dibangun adalah proxy server menggunakan aplikasi squid yang mampu membantu meningkatkan kinerja jaringan pada salah satu warnet yang ada di yogyakarta yaitu Janturan.NET. Penyusunan meliputi dengan prosedur diantaranya indentifikasi masalah, analisis kebutuhan, perancangan jaringan, perancangan sistem, serta implementasi jaringan Proxy Server, sedangkan pengujian system dengan black box dan alpha test.

Hasil penelitian dapat bekerja dengan optimal pada Optimalisasi Penyimpanan video menggunakan videocache Pada Proxy Server. Sehingga para pelanggan warnet merasa tidak dirugikan. Aplikasi telah diuji menggunakan black box test yang berdasarkan hasil uji coba tersebut dapat disimpulkan bahwa aplikasi ini dapat diterapkan dilokasi penelitian.

Kata kunci : *Proxy, Server, Squid, Streaming, Cache.*

1. PENDAHULUAN

Teknologi *web* yang semakin hari semakin berkembang tidak seimbang dengan harga *bandwidth* yang turun sangat perlahan. Ini tentunya akan membuat administrator pusing, disisi lain harus menghemat belanja *bandwidth,* disisi lain pengguna kerap sekali mengakses *video streaming* yang tentunya sangat boros

bandwidth yang paling susah diakses dengan keterbatasan *bandwidth* yang ada adalah *Video Streaming*, dengan teknologi *flash* dengan format *File Flv*. Selain *youtube* banyak juga *website* lainnya yang menyediakan *video streaming* diantaranya *google video, metacafé* dan lain-lain. Ada sebuah ide menarikdari orang-orang yang tergabung di *cachevideos.com* untuk mencache *video-video* yang sering diakses. Mereka menamakan aplikasi dengan nama *videocache*[1].

Penggunaan teknologi jaringan komputer yang lebih sederhana dapat dijumpai pada perusahaan-perusahaan, warung-warung internet, maupun dirumah-rumah yang biasanya merupakan pengguna layanan internet dari ISP tersedia. Layanan internet seperti ini dapat diperoleh melalui kabel maupun nirkabel *(wireless)* yang nantinya sama-sama akan diterima oleh sebuah *modem.* Dari *modem* inilah para pengguna *(user)* dapat menikmati layanan internet yang diberikan. Model internet seperti ini adalah bentuk yang paling ekonomis. Warnet Janturan.NET Yogyakarta mempunyai satu buah server menggunakan *Mikrotik Router* dengan bandwidth 2 Mb yang didapat dari ISP (*Internet Service Provider*) PT. Telekomunikasi Indonesia Tbk (Telkom) yaitu menggunakan paket *Internet* Telkom *Speedy*. Akan tetapi *bandwidth* 2 Mb tersebut setelah di test menggunakan *ping speed test* (www.speedtest.net dan *www.speedtest.telkomspeedy.com*) tidak murni mendapatkan 2 Mb, rata-rata hanya mencapai 1,5 Mbps saja. *Mikrotik Router* menangani 13 *personal computer* (PC) *client* di Janturan.NET Yogyakarta. Sehingga dengan keterbatasan itu (PC) client mengalami *down* atau *lemot*, terlebih untuk pengaksesan *video streaming.* Walaupun dari segi *hardware* sudah cukup memenuhi syarat, tetapi untuk akses jaringannya disana kurang memuaskan kebanyakan *client* mengeluhkan jika mengakses *web* tersebut. Akibatnya banyak *client* yang komplain tentang hal tersebut,dikarenakan akses yang lambat terlebih lagi untuk mengakses situs *Web video streaming.*

Berdasarkan latar belakang diatas, maka dibutuhkan sebuah sistem yang dapat mengoptimalkan kinerja jaringan di Janturan.NET Yogyakarta, lebih tepatnya di Warung Boto sebelah kampus III UAD Yogyakarta.Oleh karena itu akan dilakukan sebuah penelitian, diantaranya bagaimana supaya sebuah *video webportal* yang di akses harus tanpa ada *buffering*. Dengan tujuan untuk memuaskan *client* dan terlebih lagi untuk meningkatkan jumlah pelanggan. Karena salah satu keunggulan warnet ini akan menjadi salah satu yang paling diminati oleh mahasiswa UAD dan warga sekitarnya. Mengutip dari manajemen bahwa ”*Salah satu aset yang paling dominan bagi JanturanNet yaitu mahasiswa UAD Yoyakarta..maka layanilah mereka dengan sepenuh hati”.*

2. KAJIAN PUSTAKA

Alfa Nur Aribawono (2007), [2]. Metoda Implementasi *Hierarchical Caching Proxy*. Pada penelitian ini *hierarchical caching proxy* telah menjadi satu strategi populer untuk meningkatkan kinerja *caching proxy* tunggal, belum ada satu prosedur standar untuk menentukan konfigurasi hirarki terbaik untuk satu set mesin dan beban kerja yang harus dilayaninya. Penelitian ini memberikan satu contoh metoda implementasi hirarki *caching proxy* dalam lingkungan kampus beserta kebijakan yang diterapkan bagi panggunanya. Implementasi ini memudahkan kita mengkaji konfigurasi yang ada, serta membuka kemungkinan untuk melakukan

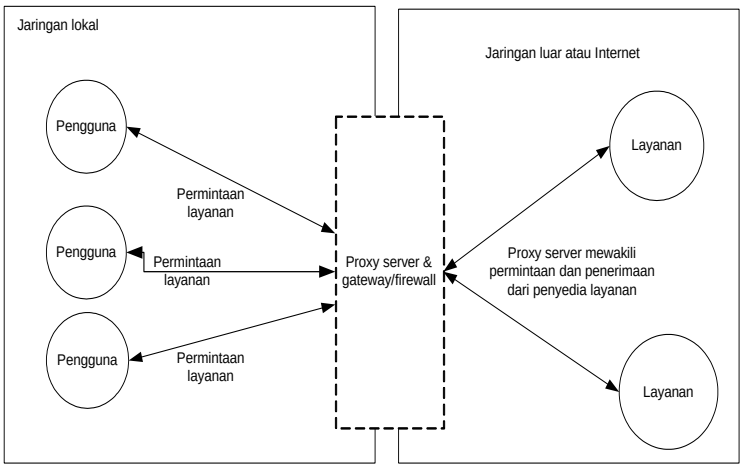
tuning pada hirarki *cache*. *Cache* diimplementasikan menggunakan perangkat lunak *open source Squid.*

Pada penelitian ini menitik beratkan pada perancang *proxy server* pada *videoCache* menggunakan sistem operasi *open source FreeBSD* dan *squid* sebagai aplikasinya. Perbedaan terletak pada metode penyimpanan data yaitu menggunakan *VideoCache* dengan kelebihannya adalah *VideoCache* dapat menyimpan *directory* file *Video streaming* sehingga apabila sebuah *video* tersebut sudah pernah di kunjungi lebih dari 1 kali, *Video* tersebut tidak akan mengalami *buffering* lagi. Sedangkan penelitian terdahulu hanya menyimpan file *directory* halaman *web* saja. Disini peneliti akan menekankan optimasi penyimpanan *cache* pada *proxy server* menggunakan *VideoCache* dengan harapan sebagai solusi untuk meningkatkan kinerja jaringan komputer yang efektif dan efisien.

2.1 Proxy

Proxy adalah aplikasi yang menjadi perantara antara *client* dengan *web server*. Salah satu fungsi *proxy* adalah menyimpan *cache* [2]. Dalam jaringan LAN, apabila client mengakses URL *web* maka *browser* akan mengirim *request* tersebut ke *proxy server*. Apabila tidak tersedia, maka *proxy server* akan menangani langsung permintaan ke *web server*.

Diagram berikut menggambarkan posisi dan fungsi dari proxy server, diantara pengguna dan penyedia layanan terlihat pada gambar 1 dibawah ini:



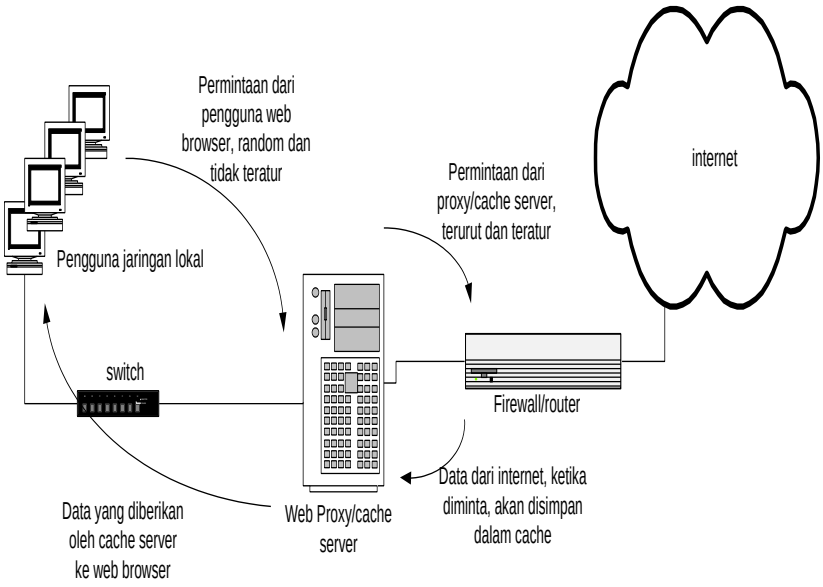
Gambar 1. Diagram alur kerja proxy

Cara kerja Proxy pada dasarnya adalah menyimpan *Cache*. Router berada diantara NOC. Internet dan client. Sewaktu browser membuka hubungan http dengan www.google.com, router segera mengenali bahwa ada paket data yang berasal dari client dengan nomor port 80. Paket tersebut dibelokkan ke *cache server.* Cache server, seperti biasa akan melakukan tugasnya dan memmberikan object yang diminta kepada router dan router memberikanya kepada client tadi. Cache harus mempunyai port selain 80 , misal 8080/3130 (default), karena jika tidak akan terjadi looping. Router memredirect berdasarkan nomor port.

2.2 Caching

Fungsi yang sangat penting dari suatu *proxy server* adalah *caching*. *Proxy server* memiliki mekanisme penyimpanan obyek-obyek yang sudah pernah diminta dari *server-server* di *Internet*, bisaa disebut *caching*. Karena itu, *proxy server* yang juga melakukan proses caching juga bisa disebut cache server, [2]

Diagram berikut menggambarkan proses dan mekanisme caching terlihat pada gambar 2 di bawah ini :



Gambar 2. Mekanisme caching.

Gambar diatas menunjukkan cara kerja *caching*, misal ada pengguna membuka browser dan mengetikkan URL http://www.google.com/. Content yang diminta pada URL tersebut dinamakan '*Internet object'* atau di singkat object. Pertama ia akan bertanya terlebih dahulu ke sebuah DNS (*Domain Name Server)* DNS mencari IP addres dari www.google.com dari databasenya dan memberikan jawabannya kepada browser tadi. Setelah browser mendapatkan IP address, maka ia membuka hubungan http ke wes server tujuan. Web server mendengarkan adanya permintaan dari brpwser menerima content dan hubungan dengan web server bisa di putus. Content lalu ditampilkan dan disimpan dalam hardisk.

Mekanisme caching akan menyimpan obyek-obyek yang merupakan hasil permintaan dari dari para pengguna, yang didapat dari Internet. Karena *proxy server* bertindak sebagai perantara, maka *proxy server* mendapatkan obyek-obyek tersebut lebih dahulu dari sumbernya untuk kemudian diteruskan kepada peminta yang sesungguhnya. Dalam proses tersebut, *proxy server* juga sekaligus menyimpan obyek-obyek tersebut untuk dirinya sendiri dalam ruang disk yang disediakan (*cache*).

Dengan demikian, bila suatu saat ada pengguna yang meminta suatu layanan ke *Internet* yang mengandung obyek-obyek yang sama dengan yang sudah pernah diminta sebelumnya, yaitu yang sudah ada dalam *cache*, maka *proxy server* akan dapat langsung memberikan obyek dari *cache* yang diminta kepada pengguna, tanpa harus meminta ulang ke *server* aslinya di *Internet*. Bila permintaan tersebut tidak

dapat ditemukan dalam cache di *proxy server*, baru kemudian *proxy server* meneruskan atau memintakannya ke *server* aslinya di *Internet* [2].

2.3 Squid

Squid dikenal sebagai aplikasi *cache* yang popular. *Squid* berlisensi GPL (GNU Public License) atau *open source* yang dalam pembuatannya melibatkan banyak orang atau organisasi [3].

Squid secara sederhana dapat dikatakan sebagai software yang diaplikasikan untuk membuat HTTP atau FTP cache. Hal yang bisaanya terjadi waktu kita *surfing* diInternet adalah browser membuat *cache* lokal dalam *hardisk*. Selanjutnya untuk mengakses situs yang sama, browser menampilkan data yang ada di-*cache*-nya, Analogi ini mirip dengan cara kerja *squid*.

Squid adalah *cache* yang dimiliki bersama sebuah jaringan. Semua *host* yang diijinkan dapat meminta data *cache* ke *server cache*. Keuntungan dari penggunaan *cache* ini adalah efisiensi. Misalnya sebuah *host* meminta *squid server* untuk mengambilkan data suatu situs, maka squid akan mengambikan data meletakkannya di *hardisk server*. *Server* ini tidak perlu lagi mendownload dari situs tersebut, tapi cukup memberikan apa yang ada di dalam *cache*. Jelas ini lebih cepat daripada mengambil langsung dari situs yang bersangkutan. Selain lebih cepat, *squid* juga menghemat penggunaan *bandwidth* [3].

2.4 FreeBSD

FreeBSD merupakan suatu sistem operasi (Operating System) turunan *Unix*. *Unix* adalah sistem operasi yang diciptakan sekitar 25 tahun yang lalu yang merupakan sistem operasi yang sangat mahal namun handal dengan stabilitasnya dan di khususkan untuk komputer *mainframe*. Dengan hadirnya membuat rasa penasaran orang akan kehandalan dan stabilitas Unix [3]. Namun sedikit berbeda dengan Linux, karena tidak semua yang ada di *Unix* ada di *Linux*. Tetapi kehandalan dan kestabilan *Unix* tetap diwarisi oleh *Linux*, terutama jika digunakan untuk *server* atau pada *mainframe*.

Menurut situs resminya, FreeBSD merupakan suatu *project* yang bertujuan untuk menyediakan suatu software yang dapat digunakan untuk masyarakat banyak tanpa adanya suatu kompensasi yang harus dibayar oleh masyarakat tersebut. FreeBSD merupakan turunan dari UNIX versi 4.4.BSD-Lite untuk komputer Intel (x86), DEC Alpha, dan Sun Ultra SPARC. *FreeBSD* dibuat oleh Computer Systems Research Group (CSRG) di *University of California at Berkeley*. Menurut *FreeBSD Handbook* (2011) [3].

3. METODE PENELITIAN

Subyek penelitian ini adalah optimalisasi penyimpanan video menggunakan *VideoCache* pada *proxy server* dengan menggunakan sistem operasi *FreeBSD* dan *squid* sebagai aplikasinya di *Janturan.Net* Yogyakarta.

3.1. Metode Pengumpulan Data

Pengumpulan data merupakan suatu usaha yang dilakukan untuk memperoleh data atau dokumentasi yang dibutuhkan dalam penelitian. Data yang diperoleh kemudian diproses sesuai dengan kebutuhan penelitian. Penelitian ini menggunakan metode pengumpulan data antara lain :

3.1.1. Studi Literatur

Studi Literatur merupakan cara pengumpulan data dengan membaca buku referensi atau dokumentasi yang berhubungan dengan penelitian, *browsing* atau *searching* merupakan cara pengumpulan data dengan cara *browsing* di*Internet* untuk mencari data atau dokumentasi yang berhubungan dengan obyek yang diteliti.

3.1.2. Metode Observasi

Metode observasi merupakan metode pengamatan secara langsung terhadap segala lalu lintas koneksi *Internet* yang menggunakan jalur akses *Internet* menuju *router* di *Janturan.Net* komputer menggunakan *tools ping speed test*, serta mengetahui kekurangan dari perangkat pendukung jaringan dan segala aktivitas yang dilaksanakan di *Janturan.Net* komputer untuk mengetahui masalah yang sering terjadi dalam jaringan Janturan.Net Yogyakarta.

3.2. Analisis Kebutuhan *User*

Kebutuhan *User* (Pengguna warnet), serta *admin*, dan *teknisi* jaringan adalah adanya sebuah sistem yang dapat meningkatkan kinerja dari jaringan yaitu dengan memaksimalkan penggunaannya serta meningkatkan kenyamanan dalam mengakses, menggelola serta memonitoring jaringan. Hal ini dapat terpenuhi bila sistem yang dibangun memenuhi unsur-unsur yang mereka perlukan. Dengan melakukan observasi dalam sebuah penelitian maka dapat diambil beberapa hal yang menyangkut kebutuhan *user* sebagai pengguna dan pengelola jaringan adalah:

1. Perlunya sistem yang dapat memaksimalkan penggunaan jaringan.
2. Perlunya sistem yang mampu meningkatkan kenyamanan dalam mengakses jaringan.
3. Perlunya sistem yang mampu memantau atau *memonitoring* dari aktifitas penggunaan jaringan.

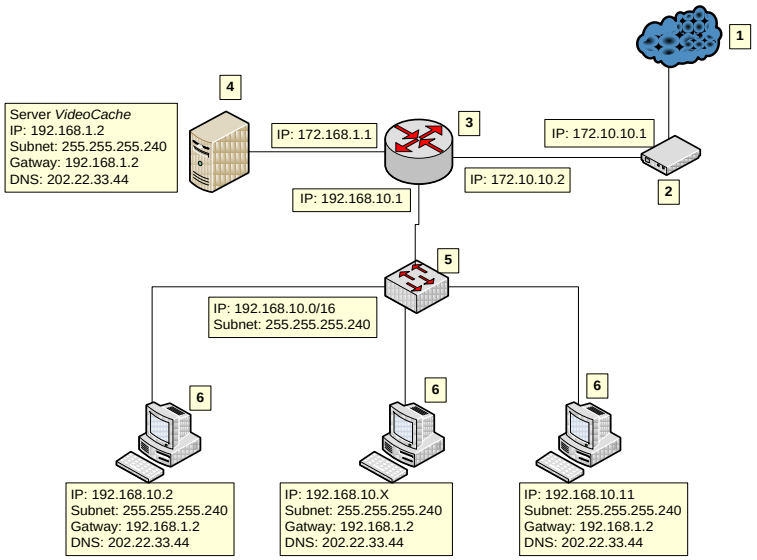
3.3. Analisis Kebutuhan Sistem

Analisis sistem yang dilakukan meliputi pengamatan dengan cara pengaksesan terhadap *web server* khususnya *web streaming* dalam waktu bersamaan, kapan saja *router* mengalami *down*, dikarenakan *router* tidak mampu menangani permintaan *client* yang disebabkan oleh jalur lalu-lintas jaringan yang padat serta kebutuhan pembatasan akses pada saat *client* melakukan *surfing* berlangsung. Sehingga dalam penelitian ini, sistem yang dihasilkan nantinya diharapkan dapat mengatasi segala permasalahan yang disebabkan oleh padatnya jalur lalu-lintas jaringan yang mengakibatkan *router* mengalami *down* sehingga *client* mengalami jaringan yang lambat dengan cara mengoptimalkan penyimpanan *VideoCache* pada *proxy server* yang menggunakan *squid* sehingga meningkatkan kinerja jaringan.

3.4. Perancangan Jaringan

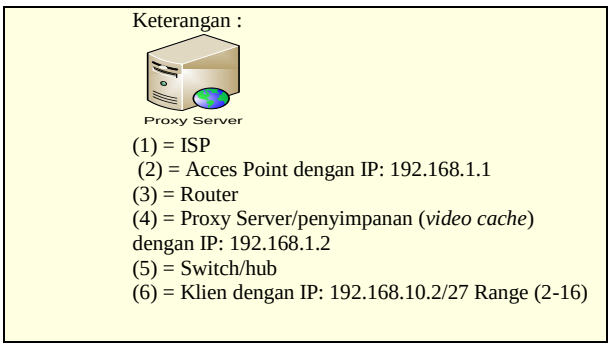
Pada tahap ini dilakukan analisis kebutuhan dan perancangan sistem untuk merumuskan solusi yang tepat dalam pembuatan sistem serta kemungkinan yang dapat dilakukan untuk mengimplementasikan rancangan tersebut. Berikut rencana desain *proxy server* yang akan dibuat seperti pada gambar 3 dibawah ini :

 e-ISSN: 2338-5197



Gambar 3. Rancangan *proxy server*

Keterangan *proxy server* seperti tampak pada gambar 4 berikut :



Gambar 4. Keterangan *proxy server*

Rancangan *proxy server* dapat diuraikan bahwa hal utama yaitu komunikasi *client* terhadap *server* menyangkut komunikasi data, jaringan, koneksi *Internet* dan lain sebagainya. *Mikrotik Router* bertindak sebagai *gateway* yang telah terkoneksi *modem* (adsl speedy) *provider Internet* dan *access point* selanjutnya diteruskan ke *client* melalui *switch*. Dalam hal ini *router* berperan sebagai *controler* dari jaringan ini.

4. HASIL DAN PEMBAHASAN

Tahap implementasi menitik beratkan pada peracangan *proxy server* dengan mengoptimasi penyimpanan menggunakan *VideoCache*. Mempersiapkan komputer yang akan digunakan dalam penelitian, membuat kabel *LAN* atau *crimping*, instalasi sistem operasi yang digunakan yaitu sistem operasi *FreeBSD* dan konfigurasi beberapa *service* setelah proses instalasi selesai. Tahapan dalam implementasi adalah sebagai berikut:

4.1. Konfigurasi *FreeBSD Proxy Server*

Proses instalasi Sistem Operasi *FreeBSD* versi 8.2 i386 pada *proxy server* dimulai dari pemilihan *regional coutry, keyboard layout*, pemilihan partisi *hardisk* yang akan digunakan sebagai *mount point* sampai semua proses instalasi selesai.

4.2. *Compile Kernel FreeBSD*

Kernel adalah program yang berfungsi sebagai *interface* antara *user-level program* dengan *hardware*, *kernel* bertanggung jawab untuk mengelola memori, kontrol keamanan jaringan dan penyimpanan data pada *hardisk*. *Compile kernel* bertujuan untuk memperoleh *kernel* yang ramping dan menambahkan *support* untuk fitur-fitur yang dibutuhkan oleh *server*.

4.3. Konfigurasi *Apache Web Server mod Php5*

Proses konfigurasi *apache* dilakukan dengan cara *manual compile*, untuk mencari sebuah paket yang akan dikonfigurasi, *Apache* melingkupi berbagai *library* sebelum extraksi file yang akan dikonfigurasi.

4.4. Konfigurasi *Squid Proxy*

Proses konfigurasi *squid* dilakukan melalui *ports collection*, untuk mencari sebuah paket yang akan dikonfigurasi, hasil tersebut menandakan bahwa paket *squid* terdapat pada direktori */usr/ports/www/squid*.

4.5. *Squid Report Generator*

Webalizer (*server log file analysis program*) merupakan sebuah aplikasi yang dibuat oleh Pedro Lineu Orso yang bertujuan untuk melihat dan merekam aktivitas user selama berada di-*Internet*. Aplikasi ini sangat lengkap sehingga kita bisa melihat situs-situs apa yang dikunjungi oleh *user*, berapa *bandwidth* yang terpakai, dan sebagainya.

4.6. *Bandwidth Monitor*

Simple Network Management Protocol (SNMP) adalah sebuah protokol yang dirancang untuk memberikan kemampuan kepada pengguna untuk memantau dan mengatur jaringan komputernya secara sistematis dari jarak jauh atau dalam satu pusat kontrol jaringan lokal.

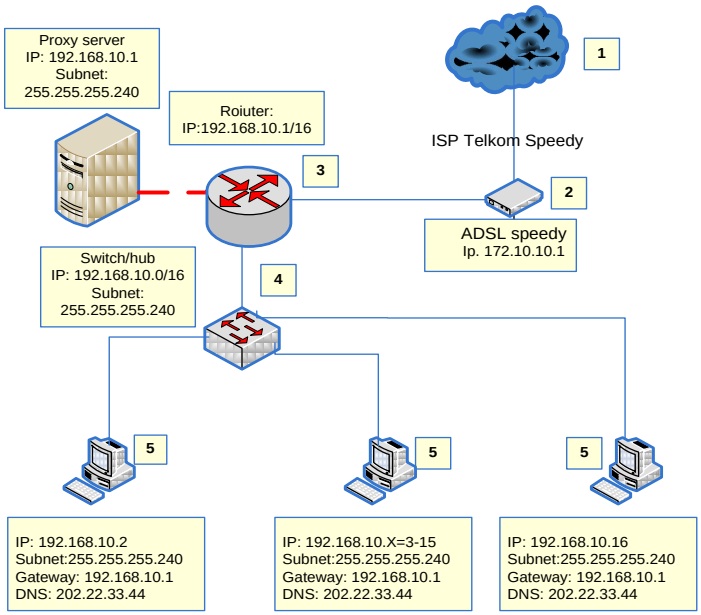
4.7. Pengujian sistem

Pada pengujian ini menitik beratkan apakah sistem *proxy server* telah bekerja sesuai dengan yang diharapkan. Pengujian ini dilakukan dengan beberapa skenario untuk benar-benar menguji sistem sehingga tujuan akhir yang diharapkan akan tercapai.

4.7.1. Pengujian sistem dengan menonaktifkan *proxy server*.

Hal ini dimaksudkan untuk mengetahui perbandingan antara sistem menggunakan *proxy server* dengan tanpa *proxy server* seperti tampak pada gambar 5 berikut:

 e-ISSN: 2338-5197



Gambar 5. Skenario pengujian *proxy server* non aktif (tanda garis merah)

Aliran data aktif terlihat pada jalur *link* warna biru dan nonaktif pada jalur *link* warna merah.

Skenario selanjutnya pengujian *bandwidth* dengan cara menggunakan tool *ping speed test*, untuk mengakses sebuah web portal yaitu *youtube* seperti tampak pada gambar 6 berikut:

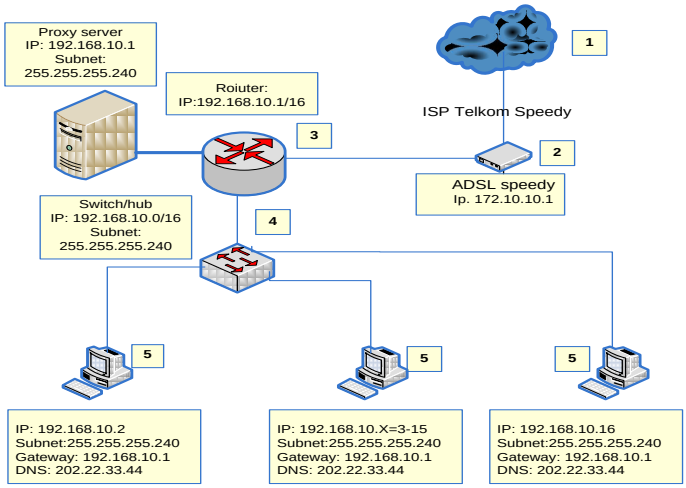
Gambar 6. Hasil *bandwidth test* tanpa mengunakan *proxy server*.

Bandwidth tanpa menggunakan *proxy server* dengan kecepatan *download* sebesar 1,46 Mbps, *upload* 0,38 Mbps dan *response time* 297 ms.

4.7.2 Pengujian sistem dengan mengaktifkan *proxy server*.

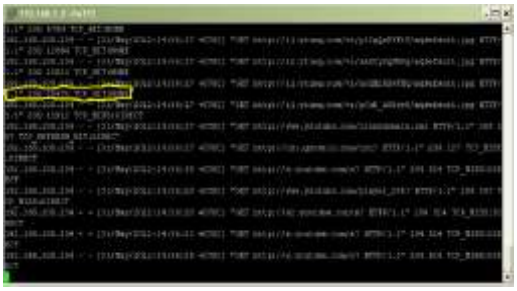
Hal ini dimaksudkan untuk mengetahui apakah kinerja *proxy server* berjalan dengan baik, seperti tampak pada gambar 7 berikut:

 e-ISSN: 2338-5197



Gambar 7. Skenario pengujian *proxy server* aktif

Aliran data aktif terlihat pada jalur *link* warna biru, maka akan terlihat laporan *access* seperti tampak pada gambar 8 berikut ini:



Gambar 8. Hasil *Access log squid*

Keterangan dari *access log squid* di atas adalah:

– *TCP_MISS* : objek yang diminta oleh *client* tidak ada di *proxy server*.
– *TCP_HIT* : objek yang diminta oleh *client* sepenuhnya sudah ter-*cache* di *proxy server*.
– *TCP_NEGATIVE_HIT* : sebagian objek yang diminta oleh *client* ter-*cache* di *proxy server*.
– *TCP_DENIED* : menolak permintaan *client*.

Mekanismenya adalah *Caching* akan menyimpan obyek-obyek yang sudah pernah diminta dari client saat mengunjungi alamat *server-server* di internet. Dengan demikian, bila suatu saat ada pengguna yang meminta suatu layanan ke internet yang mengandung obyek-obyek yang sama dengan yang sudah pernah diminta sebelumnya, yaitu yang sudah ada dalam *Cache*, maka *Proxy Server* akan dapat langsung memberikan obyek dari *Cache* yang diminta kepada pengguna tanpa harus meminta ulang ke server aslinya di internet. Bila permintaan tersebut tidak dapat ditemukan dalam *cache di proxy server*, maka *Proxy server* meneruskan atau memintakan ke server aslinya di internet. Proses *Cache Squid* berjalan dengan ditandai akses *log* dengan alamat situs yang dituju dengan keterangan :

- *Get* : sedang menyimpan *caches*
- *Hit* : bahwa alamat situs telah tersimpan

Skenario selanjutnya pengujian *bandwidth* dengan cara menggunakan tool *ping speed test*, seperti tampak pada gambar 9 berikut:

Gambar 9. Hasil *bandwidth test* dengan mengunakan *proxy server*

Gambar 10. *Traffic graph*

gambar 10 menunjukkan bahwa *bandwidth* dengan menggunakan *proxy server* dengan kecepatan *download* sebesar 1,77 Mbps, *upload* sebesar 0,39 Mbps dan *response time* sebesar 167 ms.

4.7.3 Pengujian sistem dengan pengaksesan beberapa *client* dengan menggunakan *cache proxy server*.

Semua hal ini dimaksudkan untuk mengetahui apakah kinerja *cache proxy server* dalam menangani permintaan *client* berjalan dengan baik, seperti tampak pada tabel [1] berikut:

Menggunakan *VideoCache* dan tanpa *menggunakan proxy serverVideoCache.*

Table [1] daftar pengaksesan client

No	**Client**	**Diakses pada rabu-12-2012 jam 08.24 am**			
		Tanpa videocache	Dengan videocache	Lama buffer/detik/menit/jam	
		Video 1	Video 1	Video 1	Video 1
1.	Client 1	Buffering	Tanpa buffering	35 detik untuk video durasi 15 detik	15 detik untuk durasi video 15 detik
		Video 2	Video 2	Video 2	Video 2
2.	Client 3	Buffering	Tanpa buffering	15 menit untuk video durasi 5 menit	5 menit untuk video durasi 5 menit
		Video 3	Video 3	Video 3	Video 3
3.	Client 5	Buffering	Tanpa buffering	1 jam untuk video durasi 40 menit	40 menit untuk video durasi 40

					menit

Catatan: *untuk video yang di test sudah tersimpan di proxy server videocache.*

4.7.4 Hasil perbandingan sebelum dan sesudah di optimalisasi

Setelah dilakukan penelitian baik sebelum dan sesudah dioptimalisasi suatu aplikasi *proxy server videocache* dapat diketahui perbandingan serta kelemahan-kelamahan masing-masing arsitektur jaringan. Perbandingan arsitektur jaringan dapat dilihat seperti tampak pada tabel 2 Halaman 12

Tabel 2 Perbandingan Arsitektur Jaringan

No	Aspek	Sebelum dioptimalisasi	Setelah dioptimalisasi
1.	Pengaksesan internet yang dinilai lambat terlebih untuk megakses *Video Streaming*.	Membutuhkan buffering	Tidak membutuhkan *buffering* jika sudah dikunjungi lebih dari 1 kali kunjungan.
2.	Kenyamanan client sebagai pengguna atau pemakai warnet.	Tidak memuaskan pengguna atau pemakai akses internet dikarenakan adanya *buffering.*	Lebih puas dan tentunya lebih nyaman dalam menjelajah dunia *video streaming.*
3.	Sistem *proxy server VideoCache* dapat berjalan dengan baik, serta siap untuk diterapkan.	Belum terdapat akses internet yang maksimal.	Lebih luas dan maksimal dalam menjelajah akses internet.

Dari perbandingan yang melingkupi ke tiga aspek diatas maka dapat diperolah perbandingan antara arsitektur *proxy server VideoCache* dengan model jaringan *peer to peer* berupa nilai lebih dari adanya sebuah system sebelum dioptimalisasi dan sesudah dioptimalisasi. Hasil perbandingan dapat dilihat seperti tampak pada table 3 Dibawah ini :

Tabel 3 Perbandingan Arsitektur Proxy server *VideoCache*

No	Proxy server VideoCache	Peer to peer
1.	Penggunaa internet barlangganan lebih maksimal dengan adanya *proxy server videocache* tersebut.	Kurang dimaksimalkannya penggunaan internet berlangganan.
2.	Menghemat *bandwidth* karena adanya *proxy server* tersebut.	Aliran data yang langsung tanpa adanya penyaringan atau cache memerlukan banyak bandwitdth.
3.	Menyediakan kenyamanan mengakses	Bebas mengakses tanpa adanya

	jaringan internet yang lebih baik.	pembatasan akses jaringan.
4.	Akses jaringan dapat dipantau sewaktu-waktu.	tidak terdapat media untuk memantau penggunaan jaringan dan internet.

Dari hasil tabel diatas dapat disimpulkan bahwa arsitektur jaringan *proxy server videocache* dinilai lebih efektif dibanding model jaringan sebelumnya, sedangkan permasalahan yang timbul dalam membangun arsitektur jaringan *proxy server videocache* telah terselesaikan dengan berbagai solusi seperti tampak pada tabel 3 diatas.

5. PENUTUP

Berdasarkan hasil penelitian dan pembahasan yang telah diuraikan sebelumnya maka dapat diambil kesimpulan sebagai berikut :

1. *Server* telah mampu menangani permintaan *client* pada saat mengkses 1 *web VideoStreaming* dalam waktu bersamaan dengan tidak mengalami kelambatan lagi.
2. Kegiatan *surfing* telah mencapai tingkat efektif dan efisien, dengan peran *proxy server VideoCache* sehingga *video* tidak mengalami *buffering* lagi setelah dikunjungi lebih dari satu kali kunjungan.
3. *Proxy server* telah bekerja dan mampu menangani semua permintaan *client*, serta semua situs *web streaming* dapat diakses dengan baik dan dapat dimaksimal penggunaanya.

6. DAFTAR PUSTAKA

[1], Nurwasito, H, 2006, "*Analisa Algoritma Pergantian Cache Pada Proxy Server Dengan Simulasi*", Bandung, Skripsi.

[2], Alfa Nur Aribawono, 2007, "*Metode Implementasi Hierarcical Caching Proxy".* Bandung, Skripsi.

[3], Dody Maryanto, "*Optimalisasi Akses Internet dengan SQUID*". Buku pintar internet Elex Media Komputindo.