

# Distributed Proxy Server dengan Squid pada Sistem Operasi Windows 7

Gina Akmalia, Elvyna Tunggawan, Kevin Sungiardi, Alfian Lazuardi

Program Studi Sistem Informasi, Universitas Multimedia Nusantara, Tangerang, Indonesia  
ginaakmal@gmail.com, vrc.vyna@gmail.com, kevsung31@gmail.com,  
alfian.hungyanglim@gmail.com

Diterima 3 Desember 2013

Disetujui 17 Desember 2013

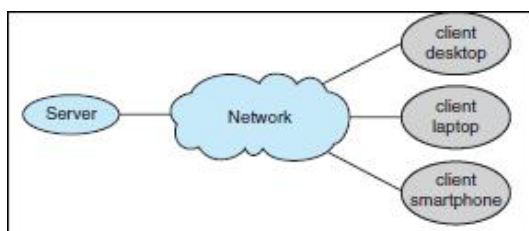
**Abstract**—Proxy server is the intermediary between client and the Internet. The usage of proxy server is one of many means to avoid excessive access to the Internet. Proxy server's cache will save every web page which has been accessed, so clients can save their bandwidth when they access same web pages repeatedly. Distributed proxy is a collection of connected proxy servers which has shared cache. This research will prove that distributed proxy server can shorten data access time and minimize bandwidth. On this research, we use Squid Proxy Server on Windows 7 as the main tool.

**Index Terms**—distributed proxy, shared cache, Squid.

## I. PENDAHULUAN

Berdasarkan data yang ada dan survey dari AC Nielsen, penggunaan internet mengalami peningkatan yang cukup signifikan selama 10 tahun terakhir ini [1]. Peningkatan tersebut juga disebabkan karena internet tidak lagi digunakan hanya untuk berkomunikasi, melainkan juga hiburan, banking, aktivitas bisnis, dsb. [2]

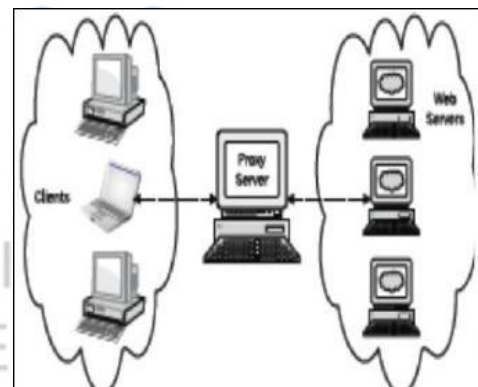
Akibat peningkatan jumlah pengguna internet (selanjutnya akan disebut dengan *client*), dibutuhkan server yang bertugas untuk menanggapi request dari client [3]. Bentuk sistem seperti ini disebut *client-server* (lihat Gambar 1). Dalam model *client-server*, masing-masing *client* mengirimkan *request* langsung ke *server*. *Server* kemudian akan mengelola *request* dan mengirim kembali hasil *request* tersebut ke komputer *client*.



Gambar 1. Model Client-Server [3]

*Proxy server* merupakan komputer yang berada di antara *client* yang melakukan *request* dengan *server* yang dituju (lihat Gambar 2). *Proxy server* memfasilitasi komunikasi antara *client* dan *server* tanpa mengubah *request* maupun *response* yang

dikirim<sup>[4]</sup>. Keuntungan menggunakan *proxy server* yaitu *client* tidak perlu menunggu waktu lama jika ingin merequest data yang sama, karena di dalam *proxy server* sudah tersimpan *cache* yang berisi situs yang sudah diakses *client*. *Cache* merupakan memori kecil yang digunakan untuk menyimpan data yang sering diakses<sup>[5]</sup>. Penerapan *caching* pada *proxy server* merupakan salah satu cara untuk mengurangi *response time* yang dialami *client* <sup>[6]</sup>.



Gambar 2. Proxy Server [4]

Banyaknya pengguna internet mempengaruhi lalu lintas data dan memperlambat kecepatan akses internet. Hal ini akan menimbulkan masalah apabila terlalu banyak *request* yang masuk ke *server* dan *server* tidak mampu melayaninya. Untuk mengatasi permasalahan tersebut, dapat diterapkan *distributed proxy server* dalam model *client-server*. *Distributed proxy server* tersusun dari serangkaian *proxy server* yang digabungkan untuk meningkatkan daya tampung *cache* dan

kemampuan *server* dalam memenuhi *request client* [7]. Penelitian kali ini akan membuktikan bahwa penggunaan model *distributed proxy server* dapat mempersingkat waktu akses dengan cara berbagi *cache* antar *proxy server*.

## II. METODE PENELITIAN

Dalam meningkatkan kecepatan akses internet, pengaturan topologi jaringan dan penambahan *proxy server* dapat dilakukan [8]. Dengan adanya *proxy server*, *client* tidak perlu mengirim *request* satu per satu ke internet. Hal ini membuat waktu kecepatan akses *client* lebih cepat dalam mengirimkan *request* terhadap suatu *website*.

Berbeda dengan penelitian sebelumnya yang meneliti cara mempersingkat waktu akses dengan menggunakan Squid pada Linux Ubuntu [9], penelitian ini akan membahas bagaimana 2 *proxy* atau lebih dapat terhubung dan saling berbagi *cache* dengan menggunakan *Squid Proxy Server* pada sistem operasi *Windows 7* sehingga dapat mempersingkat waktu akses jika terdapat banyak *proxy server* dan *client*. Dalam penelitian ini, digunakan dua laptop sebagai *proxy server* dan dua laptop sebagai *client*. Adapun spesifikasinya sebagai berikut.

- A. *Proxy server* A: ASUS A46CM-WX094D
1. Processor : Intel Core i5 3317U-1.7Ghz
  2. RAM : 4GB
  3. IP address : 192.168.1.110 B.
- B. *Client* A : ASUS A46CM-WX091D
1. Processor : Intel Core i3-3217U 1.8 GHz
  2. RAM : 4GB
  3. IP address : 192.168.1.106
- C. *Proxy server* B: ASUS A43SA-VX023D
1. Processor : Intel Core i5-2430M 2.40 GHz
  2. RAM : 2GB
  3. IP address : 192.168.1.105
- D. *Client* B: Acer Aspire One 756-8778cbb
1. Processor : Intel Core i5-2430M 2.40 GHz
  2. RAM : 2GB
  3. IP address : 192.168.1.108

Masing-masing *proxy server* menggunakan *IP address*nya pada LAN Settings (dapat diatur

melalui Control Panel-Network and Internet-Internet Options-tab Connections-LAN Settings, beri tanda centang pada 'Use proxy server for your LAN'), sedangkan masing-masing *client* menggunakan *IP address proxy server* yang dipilih, yaitu *client* A menggunakan *IP address proxy server* A dan *client* B menggunakan *IP address proxy server* B.

Kedua *proxy server* dijadikan *sibling* dengan memanfaatkan fitur *cache\_peer* pada *Squid*, dengan konfigurasi, contoh untuk *proxy server* B:

```
cache_peer IP address type proxy_port
icp_port options
```

```
cache_peer 192.168.1.110 sibling 3128
3130 multicast-siblings
```

Ada beberapa *options* [10] yang dapat digunakan, antara lain

1. *proxy-only*: data yang sudah diberikan ke *client* tidak akan disimpan di *cache server*
2. *weight=n*: pengaturan prioritas dari suatu *proxy*, semakin tinggi nilainya maka *cache proxy* tersebut akan menjadi prioritas utama dalam pengecekan
3. *round-robin*: tidak ada prioritas dalam *cache proxy*, pengecekan *cache proxy* dilakukan secara berurutan (*circular*)
4. *multicast-siblings*: tidak ada yang berperan sebagai *parent*, semua *proxy* berkedudukan sama

Penelitian ini menggunakan *option multicast-siblings*.

## III. HASIL DAN PEMBAHASAN

Penelitian dilakukan dengan menerapkan konfigurasi yang telah disebutkan sebelumnya dan menggunakan 15 *website* sebagai *sample*. Dalam mengakses *website* tersebut, digunakan koneksi internet dengan *bandwidth* 2 Mbps.

Hasil yang diperoleh membuktikan bahwa penerapan *distributed proxy server* dapat mempengaruhi kecepatan akses. Penelitian

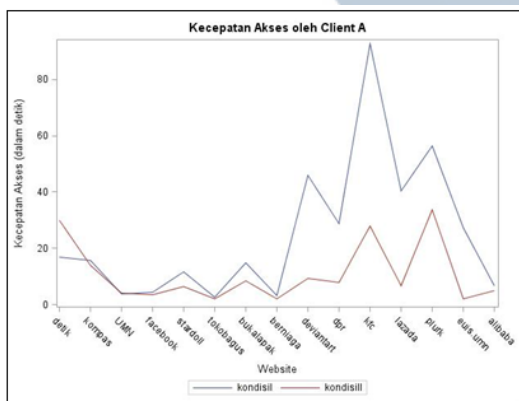
dilakukan dengan cara membandingkan waktu akses pada saat *cache proxy server* belum terdistribusi dan saat sudah terdistribusi. Hasil penelitian dapat dilihat pada Tabel 1 serta Gambar 3 dan 4.

Tabel 1. Hasil *Testing* (dalam detik)

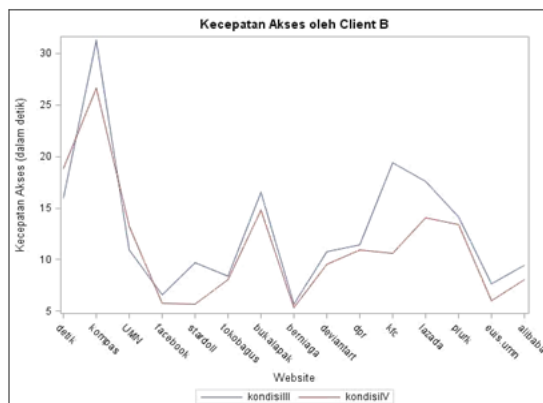
Website	Kondisi I	Kondisi II	Kondisi III	Kondisi IV
tik.com	16,83	29,92	15,91	18,80
kompas.com	15,62	13,98	31,23	26,59
umn.ac.id	3,86	4,13	10,94	13,22
facebook.com	4,2	3,68	6,58	5,76
stardoll.com	11,54	6,59	9,73	5,65
tokobagus.com	2,61	1,98	8,38	8,07
bukalapak.com	14,92	8,35	16,50	14,81
berniaga.com	3,18	2,00	5,71	5,33
deviantart.com	45,94	9,37	10,77	9,52
dpr.go.id	28,69	8,02	11,39	10,97
kfc.com	92,69	28,06	19,40	10,57
lazada.co.id	40,48	6,89	17,55	14,08
plurk.com	56,52	33,79	14,13	13,43
euis.umn.ac.id	27,45	2,16	7,61	6,04
alibaba.com	6,65	5,10	9,46	8,07

Keterangan:

- A. Kondisi I: *cache proxy server* A kosong, *client* A mengirim *request* terhadap *website* di atas. Berarti, *proxy server* A akan meneruskan *request* ke internet
- B. Kondisi II: *cache proxy server* A kosong, sedangkan *cache proxy server* B menyimpan alamat *website* di atas. *Proxy server* A dan B dijadikan *sibling*. *Client* A mengirim *request* terhadap *website* yang sudah ada di *cache*.
- C. Kondisi III: *cache proxy server* B kosong, *client* B mengirim *request* terhadap *website* di atas. Berarti, *proxy server* B akan meneruskan *request* ke internet
- D. Kondisi IV: *cache proxy server* B kosong, sedangkan *cache proxy server* A menyimpan alamat *website* di atas. *Proxy server* A dan B dijadikan *sibling*. *Client* B mengirim *request* terhadap *website* yang sudah ada di *cache*.



Gambar 3. Kecepatan Akses oleh *Client* A



Gambar 4. Kecepatan Akses oleh *Client* B

Berdasarkan data yang diperoleh, dapat dilakukan perhitungan sederhana dengan rumus [11]

$$\bar{x} = \frac{\sum x}{n} = \frac{x_1 + x_2 + x_3 + \dots + x_n}{n} \quad (1)$$

di mana

$\bar{x}$  = rata-rata kecepatan akses n = banyaknya data

$\bar{x}_1$  = rata-rata kecepatan akses sebelum *cache* terdistribusi

$\bar{x}_2$  = rata-rata kecepatan akses setelah *cache* terdistribusi

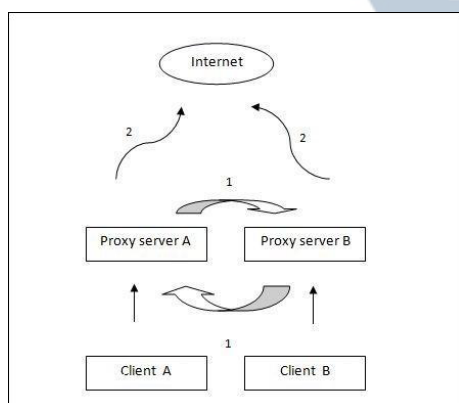
Dengan melakukan perhitungan berdasarkan persamaan (1), dapat diketahui rata-rata waktu akses dari tiap kondisi tersebut, yaitu kondisi I 24,753 detik, kondisi II 10,934 detik, kondisi III 13,019 detik, dan kondisi IV 11,394 detik. Adapun perbedaan selisih rata-rata antara kondisi I & II adalah 13,819 detik dan kondisi III & IV adalah 1,625 detik.

Melalui Gambar 2 dan 3, terlihat jelas adanya pengaruh dari penggunaan *cache* terhadap kecepatan akses. Apabila dihitung, persentase perubahan waktu

akses dari kondisi I ke kondisi II sebesar 55,827% dan kondisi III ke kondisi IV sebesar 12,481%.

Kedua angka tersebut berbeda cukup jauh akibat koneksi internet yang kurang stabil.

Pada dasarnya, konsep penyimpanan *cache* pada *proxy server* mirip dengan penyimpanan *cache* pada *client-server*. Perbedaannya, *cache* pada *server* biasa tidak dapat saling berbagi *cache*, sedangkan *cache proxy server* dapat dibagi dan digunakan bersama-sama dengan *proxy server* lain. Ketika *client* mengakses suatu *website*, alamat *website* tersebut akan tersimpan pada *cache* milik *proxy server*. Apabila *proxy server* saling berhubungan, maka ketika *client* A melakukan *request* terhadap suatu *web page*, maka *proxy server* A akan mencarinya pada *cache*. Jika tidak ditemukan, *request* akan dialihkan ke *cache* milik *proxy server* B. Apabila data yang di-*request* tidak tersedia di *cache*, *proxy server* A akan mengirim *request* ke *web server* melalui internet. Ilustrasi dapat dilihat pada Gambar 4, di mana prioritas pertama ada pada metode *shared cache* dan prioritas kedua mengirim *request* langsung ke internet.



Gambar 4. Ilustrasi

Penggunaan metode *shared cache* dapat membantu pengguna internet dalam mengakses *website* seperti portal berita maupun blog yang sering kali diperbaharui. Meskipun konten *website* cepat berubah dan berbeda dengan yang sudah tersimpan di *cache proxy server* sebelumnya, ada beberapa bagian *website* yang masih sama, misalnya *styling* dan *template/layout* sehingga *proxy server* tidak perlu mengunduh seluruh isi *website* melalui internet. Hal inilah yang menyebabkan akses internet menjadi lebih cepat.

## V. SIMPULAN

Berdasarkan hasil yang diperoleh dari penelitian di atas, terlihat perbedaan antara kondisi I dan kondisi II serta kondisi III dan IV. Pada kondisi II dan IV, waktu kecepatan akses lebih cepat dibandingkan dengan kondisi I dan III. Selain penggunaan *shared cache*, kecepatan koneksi internet juga mempengaruhi hasil penelitian.

Penelitian ini membuktikan bahwa dengan penggunaan metode *distributed proxy server* dapat mempercepat kecepatan akses dan memperkecil *bandwidth* dikarenakan *proxy server* tidak perlu terlalu sering mengirim *request* langsung ke internet, karena waktu yang dibutuhkan *server* untuk mengirim *request* ke internet jauh lebih lama dibandingkan dengan *proxy server* mengirim *request* ke *proxy server* lainnya.

Namun, hambatan yang terjadi pada penelitian ini ialah pengumpulan data pada saat *testing* dilakukan hanya satu kali sehingga dapat terjadi pembiasan data dan hasil yang kurang akurat. Maka dari itu, pada penelitian berikutnya sebaiknya dilakukan minimal tiga kali percobaan akses *website* saat *testing* agar data yang digunakan merupakan kecepatan akses rata-rata. Selain itu, ketersediaan sumber daya lebih banyak, seperti komputer *client* serta koneksi internet yang cepat dan stabil akan membantu dalam mendapatkan data yang lebih akurat.

## UCAPAN TERIMA KASIH

Terima kasih kepada Bapak Yustinus Eko Soelistio serta rekan-rekan mahasiswa Universitas Multimedia Nusantara atas dukungan dan bantuan yang diberikan selama pengerjaan paper ini.

## DAFTAR PUSTAKA

- [1] Musriha, R., Gilang, "Pengaruh Intensitas Pemakaian Internet terhadap Penggunaan Internet untuk Berbelanja Online yang Dimoderasi oleh Consumer Innovativeness di Surabaya," belum terbit.
- [2] Seybert, Heidi, "Internet Use by Household and Individuals," Eurostat, vol. 12, no. 50, hal. 1-7, Desember 2012.
- [3] Silberschatz, A., Galvin, P. B., Gagne, Greg,

- “Introduction,” di dalam Operating System Concepts, edisi 9. United States of America: Wiley, 2013, bab 1, subbab 11, hal. 38.
- [4] Umeshchandra, T. K., Ramjibhai, P. C., “Improve Squid Proxy’s Performance Using New Cache Replacement Architecture,” IJMIE, vol. 2, no. 7, hal. 418-431, Juli 2012.
- [5] Milutinovic, Veljko. (2000, Juli). Caching in Distributed Systems. IEEE Concurrency [Media Online]. *Volume 3*, halaman 1. Alamat situs: <http://home.etf.rs/~vm/papers/p3gei.lo.pdf>
- [6] Shim, J., Scheuermann, P., Vingralek, R., “Proxy Cache Design: Algorithms, Implementation and Performance,” belum terbit.
- [7] Tsui, K.C., Kaiser, M.J., Liu, J., “Distributed Proxy Server Management: A Self-Organized Approach,” Kluwer Academic Publisher, hal. 2, Sept. 2013.
- [8] Wibowo, Fajar A., “Desain Distribusi Linux untuk Optimalisasi Server dan Client Kafe Internet (Studi Kasus: Warung Internet Rocket. Net Kebumen),” belum terbit.
- [9] Arjuni, Sandy, “Perancangan dan Implementasi Proxy Server dan Manajemen Bandwidth Menggunakan Linux Ubuntu Server (Studi Kasus di Kantor Manajemen PT.Wisma Bumiputera Bandung),” belum terbit.
- [10] Hidayat, Risanuri. 2009, 12 Agustus. Konsep Proxy [media online]. Alamat situs: [http://opensource.telkomspeedy.com/wiki/index.php/Konsep\\_Proxy](http://opensource.telkomspeedy.com/wiki/index.php/Konsep_Proxy).
- [11] Black, Ken, “Descriptive Statistics,” di dalam Business Statistics for Contemporary Decision Making, edisi 6. United States of America: Wiley, 2010, bab 3, subbab 1, hal. 49.