

Analisis Algoritma Pergantian *Cache* Pada *Proxy Web Server Internet* Dengan Simulasi

Heru Nurwarsito

Abstrak -- Pertumbuhan jumlah client internet dari waktu ke waktu terus bertambah, maka respon akses internet menjadi semakin lambat. Untuk membantu kecepatan akses tersebut maka diperlukan cache pada Proxy Server. Penelitian ini bertujuan untuk menganalisis performansi Proxy Server pada Jaringan Internet terhadap penggunaan algoritma pergantian cache-nya.

Analisis Algoritma Pergantian Cache Pada Proxy Server didesain dengan metoda pemodelan simulasi jaringan internet yang terdiri dari Web server, Proxy server dan Client. Proxy Server menggunakan model hirarki dan hubungan antar Proxy Server dalam tingkatan yang sama (*sibling*). Akses permintaan dari Client ke Proxy Server dihubungkan pada tingkat paling bawah (*Level 1*). Pada saat Cache penuh, maka diperlukan penghapusan obyek data yang telah tersimpan, penghapusan ini diatur berdasarkan Algoritma LRU, LFU, LFU-Aging dan GDSF.

Dari hasil uji coba menunjukkan bahwa algoritma GDSF mempunyai Hit Ratio dan Byte Hit Ratio yang paling tinggi. Pada model hirarki dua tingkat tanpa *sibling*, tidak banyak berpengaruh terhadap performansinya. Pada model hirarki dengan satu tingkat atau dua tingkat dengan *sibling* dua buah proxy akan meningkatkan performansi berkisar 87-91%. Pada model hirarki satu tingkat dengan *sibling* antar tiga buah dan empat buah proxy yang membentuk jaringan mesh maka akan meningkatkan performansi menjadi 98-99%. Pada model hirarki dengan dua tingkat dengan *sibling* minimal tiga buah proxy yang membentuk mesh, maka akan sangat mengurangi bandwidth yang keluar bisa mencapai 95%.

Kata Kunci-- Proxy Server, hirarki, *sibling*, algoritma pergantian cache, simulasi.

I. PENDAHULUAN

Internet adalah salah satu jawaban dari permasalahan yang ada dalam teknologi informasi. Hal ini terlihat dari banyaknya informasi yang tersedia di internet dan meningkatnya jumlah pengguna atau komputer yang terhubung ke Internet yang bertambah bukan secara linier tetapi secara eksponensial. Sekarang diperkirakan ada lebih dari 30.000 jaringan dengan alamat lebih kurang 30 juta di seluruh dunia. Peningkatan jumlah ini dari waktu ke waktu terus berlipat ganda yang mengikuti pola eksponensial tersebut. [RAD-02].

Dengan pertumbuhan aplikasi internet, jumlah pengguna dan jumlah *client* yang bertambah, maka

respon akses internet yang tersedia menjadi semakin lambat untuk diakses. Maka untuk membantu kecepatan akses digunakan *proxy server* yang berfungsi untuk mempercepat layanan akses internet. [TOP-00]

Oleh karena peranan *proxy server* yang sangat penting maka diperlukan analisis *Proxy Server* tersebut, hal ini dilakukan agar diperoleh performansi yang paling baik. Analisis ini dilakukan dengan tujuan untuk mengetahui bagaimana unjuk kerja *proxy server* dengan algoritma pergantian *cache* yang ada, yaitu algoritma LRU (*Least Recently Used*), LFU (*Least Frequently Used*), LFU-Aging (*Least Frequently Used with Aging*) dan GDSF (*Greedy Dual Size Frequency*) dengan simulasi. Analisis meliputi analisa *Hit Ratio* dan *Byte Hit Ratio* karena akses permintaan *Client*. Akses permintaan *Client* dilakukan secara acak berdasarkan model matematis yang mensimulasikan permintaan alamat internet dari pemakai internet yang sebenarnya.

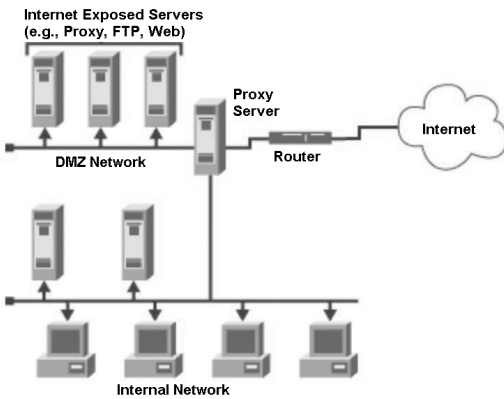
Simulasi *Proxy Server* menggunakan model hirarki dengan satu tingkat dan dua tingkat dengan kombinasi menggunakan hubungan antar *Proxy Server* dalam tingkatan yang sama (*sibling*). Akses permintaan dari *Client* ke *Proxy Server* dihubungkan pada *Proxy Server* tingkat paling bawah (*Level 1*). Apabila obyek data yang diminta tidak ada (*miss*) maka diteruskan ke *Proxy Server* yang lebih tinggi (*Proxy Induk*) atau diteruskan pada *Proxy Server* lain yang satu tingkat (*sibling*) bila ternyata masih *miss* maka diteruskan ke *Web Server*. Bila sudah ada balasan dari *Web Server*, maka obyek data yang baru akan disimpan ke dalam *Cache* pada *Proxy Server* dan diteruskan pada *Client* yang meminta. Apabila *Cache* sudah penuh, maka diperlukan penghapusan obyek data yang telah tersimpan, penghapusan ini diatur berdasarkan Algoritma yang dipergunakan pada *Proxy Server* tersebut.

II. PROXY SERVER

Proxy Server adalah sebuah komputer Server yang berfungsi sebagai gerbang (*gateway*) antara sistem jaringan lokal (*intranet*) dengan sistem jaringan global (*internet*). Pada perkembangannya, teknologi ini kemudian ditengkapi juga dengan sistem *firewall* untuk keamanan dan *cache* untuk mengurangi lalu lintas data di jaringan sekaligus untuk mengurangi *bandwidth* koneksi ke internet. Arsitektur Jaringan *Proxy Server* ditunjukkan pada gambar 1. [TOP-00]

Naskah diterima pada tanggal 22 Mei 2007.

Tri A. Kurniawan bekerja di Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya. Alamat email penulis triak@brawijaya.ac.id.

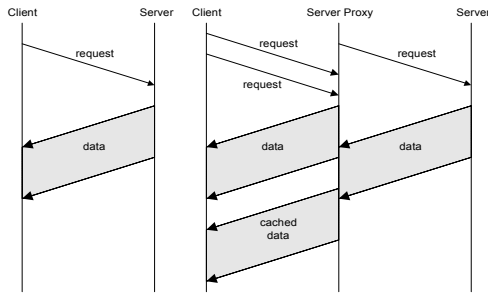


Gambar 1. Arsitektur Jaringan Proxy Server

III. CACHE PROXY SERVER

Proses metoda *caching* ditunjukkan pada gambar 2, misalkan terdapat tiga *host* yang terlibat dalam sebuah koneksi *Client-Server* agar halaman yang diminta *user* dapat tersimpan di *host* yang terdekat. Hubungan *Client-Server* demikian tidak lagi seperti koneksi *Client-Server* sederhana, dimana antara *Client* dan *Server* terkoneksi langsung. Dalam koneksi seperti ini, *host* ketiga yang disebut sebagai *Proxy* bertindak sebagai perantara antara *Client* dengan *Server* yaitu meneruskan permintaan dari *Client* kepada *Server* jika perlu. Jika halaman yang ada di *Proxy* dianggap masih baru maka halaman yang diminta *Client* cukup diambil dari *Proxy*. Pada posisi ini, *Proxy* akan bertindak sebagai *Server* dari sisi *Client* dan sebagai *Client* dari sisi *Server*, karena itu yang sebenarnya terjadi adalah dua buah hubungan *Client-Server* sederhana. Walaupun demikian, *Client* tidak akan merasakan terjadinya dua koneksi tersebut karena pada tingkat aplikasi yang tampak hanyalah *Client* melakukan koneksi langsung dengan *Server* tempat halaman tersebut berada. [TOP-00].

Dengan mengabaikan kecepatan transmisi pada jalur jaringan, waktu tanggapan untuk menyampaikan data *web* melalui internet dapat berbeda jauh. Hal ini disebabkan karena banyaknya faktor yang dapat mempengaruhi terhadap waktu tanggapan. Faktor-faktor tersebut diantaranya adalah *bandwidth* yang terpakai untuk mengirimkan data tersebut, trafik yang terjadi ketika pengiriman terjadi juga banyaknya permintaan yang datang pada *Server* yang sama dari pengguna lain yang juga dilayani. Metode *cache* dapat menghilangkan *delay* yang terjadi akibat faktor-faktor tersebut. Jika tidak menggunakan metode *cache*, permintaan data web harus melewati berbagai jenis jaringan untuk dapat sampai ke *Server* yang dituju dan kembali lagi pada pengguna yang meminta data web tersebut. Data web yang diminta dapat berupa data web statis (berupa *file text*, gambar dan hubungan (*link*)) atau dapat pula berupa data web dinamis, yang dibuat oleh search engine, database atau aplikasi berbasis web. *Server* akan menjawab setiap permintaan yang datang dengan mengirimkan data yang diminta satu persatu pada waktu yang sama. [TOP-00].



Gambar 2. Model *Client-Server* sederhana dan Model *Client-Server* menggunakan *Proxy*

IV. ATURAN PERGANTIAN CACHE

Proxy Server memiliki kapasitas media yang tetap untuk menyimpan data. Ketika media penyimpanan telah penuh, *Proxy Server* harus memilih beberapa objek untuk dihapus agar tersedia cukup ruang untuk menyimpan data baru. Aturan pergantian *cache* berarti memilih data yang akan tetap berada pada *cache* dan membuang data yang lain untuk menyediakan tempat bagi data yang baru. Pemilihan algoritma ini akan berpengaruh pada penggunaan *bandwidth* dan *hit rate* dari *cache* itu sendiri. Tujuan dari aturan pergantian *cache* ini adalah untuk memilih objek yang mana yang lebih sering digunakan oleh *client* yang mengakses pada *Proxy Server*.

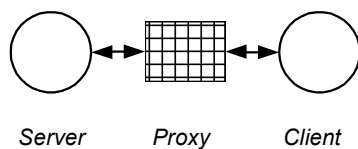
Ada beberapa metoda algoritma yang digunakan dalam aturan pergantian *Cache* pada *Proxy Server*, yaitu LRU (*Least Recently Used*), LFU (*Least Frequently Used*) dan GDSF (*Greedy Dual Size Frequency*). [JDA-99] [PCI-98] [AFJ-98]. LRU adalah salah satu metoda pergantian dasar dan merupakan yang paling umum yang digunakan pada aturan pergantian *cache*. Algoritma LRU merupakan algoritma perbaikan dari algoritma FIFO (*Fist In Fist Out*) yang telah ada sebelumnya. LFU adalah metoda pergantian *cache* untuk menetapkan atau memelihara obyek yang sering diakses atau mempunyai jumlah acuan tinggi maka obyek tetap dipertahankan dalam *Cache* dan akan menggantikan obyek dengan yang mempunyai jumlah acuan paling kecil apabila diperlukan pergantian obyek karena untuk menempatkan obyek yang baru. [LPY-98] [YPZ-98] [PCI-98]. Bila diperlukan untuk menentukan dua obyek dengan referensi yang sama maka digunakan metoda variasi LFU yang merupakan pengembangan dari LFU, yaitu LFU-aging. GDSF pertama kali dikemukakan oleh N. Young pada bulan Juni 1994. Algoritma ini merupakan pengembangan dari algoritma yang sudah ada (LRU). Pada algoritma ini lebih ditekankan ketika ada beberapa obyek yang akan dihapus tetapi mempunyai ukuran yang sama tetapi mempunyai nilai yang berbeda. [LPY-98] [YPZ-98]

V. PEMODELAN SIMULASI

Perangkat lunak Simulasi dibuat dengan bahasa pemrograman java Jbuilder9, dengan menggunakan JDK 1.4. Analisis Algoritma Pergantian *Cache* Pada *Proxy*

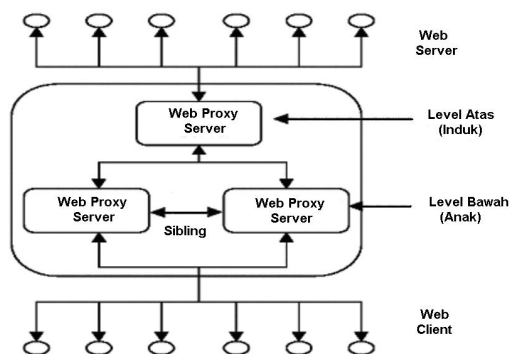
Server didesain dengan metoda pemodelan simulasi jaringan internet. Dalam model ini akan mensimulasikan node jaringan (*network node*) dengan node objek yang tampak secara visual di layar Komputer. Node-node tersebut mewakili Server (*Web Server*), Proxy (*Proxy Server*) Client. Node merupakan kelompok objek tersebut dalam sebuah jaringan.

Model simulasi jaringan internet ditunjukkan pada gambar 3. Pada kenyataannya hubungan pada kedua sisi terdiri banyak sekali titik (*node*), mungkin berupa *router-router*, *gateway* dan hubungan (*link*) fisik lainnya, akan tetapi dalam hal ini digambarkan dengan sebuah hubungan dengan parameter *single round-trip time* (RTT) dan *single loss rate* parameter yang mensimulasikan kelompok *Server* atau *Client*. Pada jaringan yang sebenarnya, *Proxy* ditempatkan menjadi satu pada *Web server* dan *Router*.



Gambar 3. Model Simulasi Jaringan Internet

Proxy Server menggunakan model hirarki bertingkat, dimana hubungan hubungan antar *Proxy Server* dapat dilakukan dalam tingkatan di atasnya yang membentuk hubungan hirarki dan juga dapat dilakukan dengan hubungan pada *Proxy Server* dengan tingkatan yang sama (*sibling*). Akses permintaan dari *Client* ke *Proxy Server* tingkat paling bawah (*Level 1*). Apabila obyek data tidak ada (*miss*) maka diteruskan ke *Proxy Server* yang lebih tinggi (*Proxy Induk*) atau diteruskan pada *Proxy Server* lain yang satu tingkatan (*sibling*) bila ternyata masih *miss* maka diteruskan ke *Web Server*. [MBW-00]. Model Hirarki *Proxy Server* dua tingkat dan hubungan *sibling* ditunjukkan pada gambar 4.



Gambar 4. Model Hirarki Proxy Server dengan sibling

Dalam simulasi yang dibuat, *Server* hanya dapat dipasang satu yang merepresentasikan Kelompok *Server*. *Server* merupakan menggambarkan kelompok *Web Server* http dan ftp. Sedangkan *Proxy* menggambarkan satu *Proxy Server* yang sebenarnya. *Client* menggambarkan kelompok *Client*. *Server*, *Proxy* dan *Client* dapat dipasang lebih dari satu node, dengan

konfigurasi hirarki yang berbeda-beda.

VI. ANALISA DAN PEMBAHASAN

Untuk mengevaluasi hasil uji coba sesuai dengan skenario-skenario. Di bawah ini akan ditunjukkan data hasil pelaksanaan uji coba sesuai dengan skenario yang telah dibuat dan evaluasi berdasarkan data hasil uji coba untuk masing-masing skenario.

• Uji coba 1

Pada Uji coba I, yaitu untuk validasi perangkat lunak, diperoleh hasil seperti pada sebagian pesan yang ditunjukkan pada gambar 5. Pada gambar tersebut tampak potongan data pesan dari hasil simulasi pada Mode *Debug*, dimana mode ini merupakan fasilitas dari perangkat lunak Simulasi. Pada kelompok tersebut merupakan data di tengah-tengah rangkaian pesan secara berurutan. Rangkaian pesan hasil uji coba Validasi ini adalah sangat panjang, beberapa halaman ditunjukkan pada Lampiran I. Data-data tersebut merupakan rangkaian dari proses pergantian data pada *cache Proxy Server* dengan LRU.

Pada data tersebut tampak pesan *Gak cukup, Sisa Cache:78812 Hapus:266 Ukuran:59664*, artinya *cache* tidak cukup untuk menyimpan file obyek data baru, sehingga diperlukan penghapusan file obyek data yang telah tersimpan, yaitu file obyek data URL 266 dengan Ukuran 59664 byte. Pesan berikutnya masih merupakan proses penghapusan file obyek data dan diulang sampai empat kali. Demikian seterusnya sampai *Sisa Cache* mencukupi untuk penyimpanan file obyek data baru.

Setelah ditelusuri satu persatu pesan tersebut menunjukkan bahwa terjadi proses permintaan dan balasan dari *Client* ke *Proxy*, dari *Proxy* ke *Server* dan sebaliknya dari *Server* ke *Proxy* dan diteruskan ke *Client* dan dilakukan penyimpanan obyek data di *cache Proxy Server*. Pada gambar tersebut menunjukkan proses *Proxy* disaat *cache* sudah penuh sehingga diperlukan penghapusan file obyek data yang telah tersimpan dan pesan tersebut menunjukkan proses penghapusan dan penyimpanan data berhasil dilakukan. Hal ini menunjukkan bahwa perangkat lunak simulasi telah bekerja dengan benar sesuai dengan yang apa yang telah dirancang.

```

Node:1 (Client): Permintaan:153
Node:2 (Proxy): Membalas:153
Node:2 (Proxy): Miss-Diteruskan permintaan ke Induk:153
Node:3 (Server): Permintaan:153
Node:3 (Server): Membalas:153 Ukuran:225392
Node:2 (Proxy): Balasan:153 Ukuran:225392
Node:2 (Proxy): Miss-Membalas:153 Ukuran:225392
Gak cukup, Sisa Cache:78812 Hapus:266 Ukuran:59664
Gak cukup, Sisa Cache:138476 Hapus:513 Ukuran:37952
Gak cukup, Sisa Cache:176428 Hapus:515 Ukuran:38153
Gak cukup, Sisa Cache:214581 Hapus:293 Ukuran:59260
Sisa Cache:273841 Menyimpan:153 Ukuran:225392
Node:1 (Client): Balasan:153 Ukuran:225392
Node:1 (Client): Permintaan:277
Node:2 (Proxy): Membalas:277
Node:2 (Proxy): Miss-Diteruskan permintaan ke Induk:277
Node:3 (Server): Permintaan:277
Node:3 (Server): Membalas:277 Ukuran:46589
Node:2 (Proxy): Balasan:277 Ukuran:46589
Node:2 (Proxy): Miss-Membalas:277 Ukuran:46589
Sisa Cache:48449 Menyimpan:277 Ukuran:46589
Node:1 (Client): Balasan:277 Ukuran:46589

```

Gambar 5. Sebagian pesan (potongan di tengah) data uji coba Validasi

- Uji coba 2

Uji coba 2 adalah untuk proses simulasi *cache* yang dilakukan dengan *Proxy Server* hirarki satu tingkat atau tunggal dengan menerapkan algoritma LRU, LFU, LFU-Aging dan GDSF. Berdasarkan data hasil pengujian maka dapat dikelompokkan *Hit Ratio* dan *Byte Hit Ratio* seperti tampak pada Tabel 1 dan Tabel 2.

Dari parameter *Hit Ratio* dan *Byte Hit Ratio* mempunyai kecenderungan bahwasanya Algoritma GDSF mempunyai *Hit Ratio* dan *Byte Hit Ratio* yang paling tinggi, disamping itu LFU mempunyai *Byte Hit Ratio* yang cukup baik.

- Uji coba 3

Uji coba 3 adalah untuk proses simulasi *cache* yang dilakukan dengan *Proxy Server* hirarki dua tingkat atau ganda dengan menerapkan hirarki tingkatan untuk algoritma LRU, LFU, LFU-Aging dan GDSF.

Hasil uji coba hirarki dua tingkat dengan algoritma lainnya ditunjukkan pada Tabel 3. Pada tabel ini hanya mengambil nilai-nilai total hasil simulasi pada uji coba tersebut.

Dari hasil simulasi pada tabel 3 dapat diamati dengan membandingkan *Hit Ratio* dan *Byte Hit Ratio* pada tabel 1 dan tabel 2 yang merupakan hasil uji coba hirarki satu tingkat, *Hit Ratio* dan *Byte Hit Ratio* nilainya hampir sama. Ini menunjukkan untuk model hirarki dengan dua tingkat tidak banyak berpengaruh terhadap performansi terhadap *Proxy Server*.

- Uji coba 4

Uji coba 4 adalah pengujian untuk proses simulasi simulasi Sibling dua tingkat untuk proses simulasi *cache* yang dilakukan dengan *Proxy Server* hirarki satu tingkat dan dua tingkat dengan menghubungkan *Proxy* yang setingkat.

Hasil uji coba sibling hirarki satu tingkat dengan 2 buah *Proxy* dengan algoritma yang sama ditunjukkan pada Tabel 4.

Sedangkan hasil uji coba sibling hirarki satu tingkat dengan 4 buah *Proxy* dengan algoritma yang berbeda dengan ditunjukkan pada Tabel 5.

Untuk uji coba Sibling hirarki dua tingkat dengan kombinasi *Proxy* dengan algoritma yang bervariasi ditunjukkan pada tabel 6.

Dari hasil uji coba simulasi Sibling pada satu tingkat dan dua tingkat, seperti yang ditunjukkan pada tabel 4 menunjukkan bahwa dengan adanya hubungan sibling antar dua buah *proxy* yang berdekatan pada hirarki akan meningkatkan performansi berkisar 86-91%.

Sedangkan berdasarkan tabel 5 dan tabel 6. menunjukkan bahwa dengan adanya hubungan sibling antar tiga buah dan empat buah *proxy* yang berdekatan dengan membentuk jaringan mesh, dimana setiap *proxy* terhubung sibling dengan *proxy* lainnya, maka akan meningkatkan performansi menjadi 98-99%.

Berdasarkan tabel 6 menunjukkan untuk hirarki dua tingkat dengan hubungan sibling dengan minimal tiga

buah *proxy* yang membentuk mesh, maka akan sangat mengurangi bandwidth yang keluar, terutama untuk penempatan *proxy* pada tingkat atas dengan algoritma yang berbasis frekwensi yaitu LFU, LFUA dan GDSF, maka pengurangan bandwidth yang keluar bisa mencapai 95%. Hal ini sangat menguntungkan untuk menghemat bandwidth.

VII. SIMPULAN

Berdasarkan aplikasi yang telah dibuat, hasil uji coba dan evaluasi yang dilakukan terhadap perangkat lunak simulasi yang telah dilakukan dan dibahas pada Bab VI dapat diperoleh kesimpulan sebagai berikut:

- Dari hasil simulasi pada Mode Debug dan dengan mengamati satu persatu pesan proses permintaan dan balasan menunjukkan bahwa perangkat lunak simulasi telah bekerja dengan benar sesuai dengan yang apa yang telah dirancang.
- Dari parameter *Hit Ratio* dan *Byte Hit Ratio* mempunyai kecenderungan bahwasanya Algoritma GDSF mempunyai *Hit Ratio* dan *Byte Hit Ratio* yang paling baik, ini menunjukkan bahwasanya *Proxy Server* dengan Algoritma GDSF memiliki performansi yang paling baik.
- Pada model hirarki dengan dua tingkat tanpa hubungan sibling tidak banyak berpengaruh terhadap performansinya.
- Pada model hirarki dengan satu tingkat dan dua tingkat dengan hubungan sibling antar dua buah *proxy* yang berdekatan akan meningkatkan performansi berkisar 87-91%.
- Pada model hirarki dengan satu tingkat dengan adanya hubungan sibling antar tiga buah dan empat buah *proxy* yang berdekatan dengan membentuk jaringan mesh dimana setiap *proxy* terhubung sibling dengan *proxy* lainnya, maka akan meningkatkan performansi menjadi 98-99%.
- Pada model hirarki dengan dua tingkat dengan hubungan sibling dengan minimal tiga buah *proxy* yang membentuk mesh, maka akan sangat mengurangi bandwidth yang keluar, dalam simulasi bisa mencapai 95%.

DAFTAR PUSTAKA

- [AFJ-98] M. Arlitt, R. Friedrich, and T. Jin. 1998. *Performance evaluation of Web proxy cache replacement policies*. Proceedings of Performance Tools '98. Lecture Notes in Computer Science, 1469:193--206, www.hpl.hp.com/techreports/98/HPL-98-97.pdf.
- [MBW-00] Mudashiru Busari, Carey Williamson, 2000. *Simulation Evaluation of a Heterogenous Web Proxy Caching Hierarchy*, Department of Computer Science University of Saskatchewan, Canada
- [JDA-99] John Dilley, Martin Arlitt, October, 1999, *Improving Proxy Cache Performance-Analyzing Three Cache Replacement Policies*, http://www.hpl.hp.com/techlib/1999/HPL-1999-142.pdf, Technical Report, HP Laboratories Palo Alto.

- [PCI-98] Pei, Cao; Sandy Irani, 1998, *Cost-Aware WWW Proxy Caching Algorithms*, http://www.ics.uci.edu/~irani/gd-size.pdf.
- [RAD-02] Radnet.id, Jakarta, 2002. *Konsep Internet*, http://www.rad.net.id/homes/edward/intbasic/basic.htm.
- [TOP-00] Achmad Husni Thamrin, Onno W. Purbo, 2000. *Proxy Cache Di Internet*, Computer Network Research Group, Institut Teknologi Bandung, Bandung.
- [LPY-98] Linping Yu, 1998, *Internet Web Proxies: Workload Characterization*, Department of Computer Science University of Saskatchewan, Canada.
- [YPZ-98] Yanping Zao, 1998, *Trace-Driven Simulation of caching Strategis for internet Web Proxy*, Department of Computer Science University of Saskatchewan, Canada.

TABEL 1. HIT RATIO (%) HIRARKI TUNGGAL

| Alg \ Cache | 1MB | 5MB | 10MB | 15MB | 20MB | 25MB | 30MB | 40MB | 50MB |
|-------------|-----|-----|------|------|------|------|------|------|------|
| LRU | 2 | 13 | 28 | 41 | 56 | 65 | 71 | 90 | 95 |
| LFU | 2 | 12 | 27 | 40 | 54 | 67 | 77 | 87 | 97 |
| LFUA | 2 | 12 | 26 | 40 | 53 | 59 | 75 | 89 | 93 |
| GDSF | 2 | 13 | 28 | 40 | 57 | 67 | 75 | 92 | 97 |

TABEL 2. BYTE HIT RATIO HIRARKI TUNGGAL

| Alg \ Cache | 1MB | 5MB | 10MB | 15MB | 20MB | 25MB | 30MB | 40MB | 50MB |
|-------------|-----|-----|------|------|------|------|------|------|------|
| LRU | 2 | 13 | 28 | 40 | 54 | 65 | 71 | 90 | 96 |
| LFU | 2 | 12 | 28 | 41 | 53 | 66 | 75 | 88 | 96 |
| LFUA | 2 | 14 | 23 | 39 | 51 | 60 | 74 | 88 | 94 |
| GDSF | 2 | 14 | 28 | 41 | 55 | 66 | 75 | 91 | 97 |

TABEL 3. HASIL UJI COBA HIRARKI DENGAN ALGORITMA CACHE SAMA

| Algrt | Node | Perm | Balasan | Balasan(byte) | Hit | Hit (byte) | HitRatio | ByteHitRatio |
|-------|-------|-------|---------|---------------|-------|------------|----------|--------------|
| LRU | kiri | 33322 | 33321 | 2840480410 | 17324 | 1478858605 | 51 | 52 |
| | kanan | 33329 | 33328 | 2809388201 | 17454 | 1464568881 | 52 | 52 |
| | atas | 31881 | 31881 | 2707089442 | 8037 | 676894209 | 25 | 25 |
| LFU | kiri | 33752 | 33752 | 3943609289 | 16772 | 1411444441 | 49 | 47 |
| | kanan | 33756 | 33755 | 2954041364 | 16035 | 1431152982 | 47 | 48 |
| | atas | 34703 | 34703 | 3055502059 | 13063 | 1078741597 | 37 | 35 |
| LFUA | kiri | 33877 | 33877 | 2940294013 | 16875 | 1432366028 | 49 | 48 |
| | kanan | 33880 | 33880 | 2925480364 | 16507 | 1421637392 | 48 | 48 |
| | atas | 34375 | 34375 | 3011802612 | 12984 | 1133248888 | 37 | 37 |
| GDSF | kiri | 33832 | 33832 | 2934337102 | 17303 | 1488604629 | 51 | 50 |
| | kanan | 33835 | 33835 | 2907199976 | 17561 | 1556254126 | 51 | 53 |
| | atas | 32807 | 32806 | 2797041185 | 11769 | 1044643434 | 35 | 37 |

TABEL 4. HASIL UJI COBA SIBLING SATU TINGKAT DENGAN ALGORITMA CACHE SAMA

| Algrt | Posisi | Perm | Balasan | Balasan(byte) | Hit | Hit (byte) | HitRatio | ByteHitRatio |
|-------|--------|-------|---------|---------------|-------|------------|----------|--------------|
| LRU | kiri | 33670 | 33670 | 2774510669 | 30815 | 2546873882 | 91 | 91 |
| | kanan | 33673 | 33673 | 2777363826 | 30834 | 2543806561 | 91 | 91 |
| LFU | kiri | 33708 | 33708 | 2942733795 | 29471 | 2587509910 | 87 | 87 |
| | kanan | 33711 | 33711 | 2940339241 | 29306 | 2570353616 | 86 | 87 |
| LFUA | kiri | 33740 | 33740 | 2961762277 | 29267 | 2597510083 | 86 | 87 |
| | kanan | 33747 | 33747 | 2944022386 | 29275 | 2579943850 | 86 | 87 |
| GDSF | kiri | 33875 | 33875 | 2635893811 | 30993 | 2400695774 | 91 | 91 |
| | kanan | 33873 | 33873 | 2640568401 | 30988 | 2411415346 | 91 | 91 |

TABEL 5. HASIL UJI COBA SIBLING SATU TINGKAT DENGAN 4 BUAH PROXY

| Posisi | Perm | Balasan | Balasan(byte) | Hit | Hit (byte) | HitRatio | ByteHitRatio |
|--------|-------|---------|---------------|-------|------------|----------|--------------|
| LRU | 31749 | 31749 | 2622858489 | 31524 | 2603598166 | 99 | 99 |
| LFU | 31750 | 31750 | 2615611043 | 31512 | 2596945917 | 99 | 99 |
| LFUA | 31754 | 31754 | 2631544192 | 31540 | 2613431147 | 99 | 99 |
| GDSF | 31759 | 31759 | 2635538702 | 31527 | 2616731075 | 99 | 99 |

TABEL 6. HASIL UJI COBA SIBLING DUA TINGKAT DENGAN ALGORITMA CACHE BERVARIASI

| Posisi | Algrt | Perm | Balasan | Balasan(byte) | Hit | Hit (byte) | HitRatio | ByteHitRatio |
|--------|-------|-------|---------|---------------|-------|------------|----------|--------------|
| Atas | LRU | 1637 | 1637 | 139333606 | 721 | 62391099 | 44 | 44 |
| Bawah | LFU | 35187 | 35187 | 3040431152 | 34628 | 2992788596 | 98 | 98 |
| Bawah | LFUA | 35199 | 35199 | 3028832571 | 34674 | 2984242543 | 98 | 98 |
| Bawah | GDSF | 35200 | 35200 | 3029724792 | 34647 | 2982623771 | 98 | 98 |
| Atas | LFU | 1195 | 1195 | 96943801 | 36 | 2311518 | 3 | 2 |
| Bawah | LRU | 33663 | 33663 | 2857363551 | 33253 | 2825822611 | 98 | 98 |
| Bawah | LFUA | 33680 | 33680 | 2834635940 | 33302 | 2803799645 | 98 | 98 |
| Bawah | GDSF | 33682 | 33682 | 2858733517 | 33276 | 2824259872 | 98 | 98 |
| Atas | LFUA | 1124 | 1124 | 90887919 | 43 | 3947140 | 3 | 4 |
| Bawah | LRU | 33702 | 33702 | 2750243718 | 33328 | 2719526588 | 98 | 98 |
| Bawah | LFU | 33706 | 33706 | 2752670525 | 33346 | 2722906807 | 98 | 98 |
| Bawah | GDSF | 33712 | 33712 | 2759459642 | 33322 | 2729052571 | 98 | 98 |
| Atas | GDSF | 1254 | 1253 | 100819050 | 69 | 4457550 | 5 | 4 |
| Bawah | LRU | 33755 | 33754 | 2891574364 | 33329 | 2857076030 | 98 | 98 |
| Bawah | LFU | 33763 | 33762 | 2896531415 | 33350 | 2864929937 | 98 | 98 |
| Bawah | LFUA | 33765 | 33765 | 2902421382 | 33350 | 2867821773 | 98 | 98 |