

PROPOSAL SKRIPSI S1

Analisis Perbandingan Penghematan *Traffic Data* dengan Menerapkan Metode *Least Recently Used* (LRU), Metode *Least Frequently Used with Dynamic Aging* (LFUDA), dan Metode *Greedy-Dual Size Frequency* (GDSF) pada Proses Pergantian *Cache Squid Proxy Server*

“Studi Kasus: Jaringan Lokal Gedung BAA-BAPSI Universitas Sebelas Maret”



Disusun oleh :

Arief Nur Huda

M0514010

PROGRAM STUDI INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS SEBELAS MARET

SURAKARTA

2018



UNIVERSITAS SEBELAS MARET
PROGRAM STUDI INFORMATIKA

PROPOSAL SKRIPSI S1

Nama : Arief Nur Huda

NIM : M0514010

PERSETUJUAN PEMBIMBING

Proposal Skripsi S1 ini telah disetujui oleh :

Pembimbing 1

Abdul Aziz, S.Kom., M.Cs.

NIP. 198104132005011001

DAFTAR ISI

PROPOSAL SKRIPSI S1	ii
DAFTAR ISI.....	iii
DAFTAR TABEL.....	v
DAFTAR GAMBAR	vi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA	5
2.1 Dasar Teori.....	5
2.1.1 Proxy Server.....	5
2.1.2 Squid	6
2.1.2.1 Hit Ratio	8
2.1.2.2 Byte Hit Ratio.....	8
2.1.2.3 Access.log	8
2.1.3 Algoritma Penggantian <i>Cache</i>	9
2.2 Penelitian Terkait	15
BAB III METODOLOGI PENELITIAN	21
3.1 Studi Literatur	21
3.2 Perencanaan dan Analisis Kebutuhan	22
3.3 Perancangan <i>Proxy Server</i>	22

3.4	Implementasi Perancangan <i>Proxy Server</i>	23
3.5	Pengumpulan Data	24
3.6	Penarikan Hasil	27
3.7	Penarikan Kesimpulan.....	28
	Jadwal Pelaksanaan.....	30
	Daftar Pustaka.....	31

DAFTAR TABEL

Tabel 2.1 Kode <i>output</i> squid (Bramantyo, 2016)	7
Tabel 2.2 Penelitian Terkait	18
Tabel 3. 1 Spesifikasi perangkat	23
Tabel 3. 2 Daftar Website	25
Tabel 3. 3 Kategori <i>Request</i>	27

DAFTAR GAMBAR

Gambar 2.1 Cara kerja <i>proxy server</i> (Bramantyo, 2016).....	6
Gambar 2.2 Flowchart algoritma <i>Least Recently Used</i> (LRU). (Sadewa, Anif, & Hidayat, 2016)	10
Gambar 2. 3 Flowchart algoritma <i>Least Frequently Used with Dynamic Aging</i> (LFUDA). (Khosrowpour, 2005).....	12
Gambar 2.4 Flowchart algoritma <i>Greedy-Dual Size Frequency</i> (GDSF). (Bramantyo, 2016)	14
Gambar 3. 1 Metodologi penelitian	21
Gambar 3.2 Topologi jaringan.....	24
Gambar 3. 3 Alur pengujian dan pengolahan data.....	26
Gambar 3. 4 Desain program berbasis web	29

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada saat melakukan koneksi ke internet dan melakukan aktifitas *browsing*, terkadang koneksi menjadi lambat yang menyebabkan proses memuat halaman web memerlukan waktu yang lebih lama (Syamsu, 2012). Hal tersebut tentunya sangat mengganggu aktifitas *browsing*. Melambatnya koneksi tersebut berhubungan dengan *bandwidth*. Semakin banyak user yang melakukan aktifitas *browsing* pada jaringan yang sama akan berbanding lurus dengan penggunaan *bandwidth* yang tentu saja mengakibatkan melambatnya koneksi internet pada jaringan tersebut (Syamsu, 2012).

Dari penelitian yang dilakukan oleh Yuisar, Liza, dan Yonalanda pada tahun 2015 mengungkapkan bahwa melambatnya koneksi internet pada sebuah jaringan yang disebabkan oleh semakin sempitnya *bandwidth* dapat diantisipasi dengan membuat sebuah *proxy server*. *Proxy server* tersebut menggunakan aplikasi Squid yang terpasang pada sebuah *server* dengan sistem operasi Linux Ubuntu Server 14.04. *Proxy server* tersebut berfungsi untuk menyimpan data-data halaman website baik itu berupa teks, gambar, audio, maupun video. Sehingga saat *user* mengakses internet, halaman website yang berisi konten berupa teks, gambar, audio, maupun video akan tersimpan dalam *proxy server*. Hal tersebut memberi dampak positif berupa akses internet menjadi cepat karena sebagian konten tidak perlu di ambil dari internet tetapi cukup melalui *proxy server* di jaringan lokal.

Dalam penerapan *proxy server* menggunakan aplikasi Squid, terdapat beberapa algoritma pergantian cache dengan default (Ali, Sulaiman, & Ahmad, 2014). Algoritma pergantian *cache* tersebut digunakan untuk mengganti *cache* ketika ruang penyimpanan pada *proxy server* penuh. Penggunaan algoritma tersebut berpengaruh kepada prioritas sebuah konten di dalam ruang penyimpanan *proxy server* (Nanda, Singh, & Saini, 2015). Pada penelitian yang dilakukan oleh Eskandaru, Muhammad Anif, dan Sidiq pada tahun 2016 terhadap *proxy server* Pasar Grosir Pekalongan dengan menggunakan algoritma pergantian *cache Least Frequently Used with*

Dynamic Aging (LFUDA) dan *Greedy-Dual Size Frequency* (GDSF) mendapatkan hasil bahwa algoritma pergantian *cache Greedy-Dual Size Frequency* (GDSF) lebih efisien dibandingkan dengan *Least Frequently Used with Dynamic Aging* (LFUDA) dengan jenis data heterogen, hal ini dibuktikan dengan jumlah persentase *hit ratio* dan *byte hit ratio* dari algoritma *Greedy-Dual Size Frequency* (GDSF) lebih unggul dari *Least Frequently Used with Dynamic Aging* (LFUDA) yaitu sebesar 14,01% : 10,74% untuk *hit ratio* dan 5,78% : 5,30% untuk *byte hit ratio*.

Dikutip dari website Squid (squid), algoritma pergantian *cache* pada aplikasi Squid secara *default* adalah *Least Recently Used* (LRU). Tidak sedikit penelitian yang melakukan modifikasi terhadap algoritma *Least Recently Used* (LRU) seperti penelitian yang dilakukan oleh Tanwir, Gamantyo, dan Achmad pada tahun 2017 yang menerapkan modifikasi algoritma pergantian *cache Least Recently Used* (LRU) dengan metode *First In First Out* (FIFO) dan penelitian yang dilakukan oleh Rassul Ayani, Yong Meng Teo, dan Yean Seen Ng pada tahun 2003 yang menerapkan algoritma LRU-Distance untuk mengurangi polusi *cache* pada sebuah *proxy server*.

Dengan terdapatnya beberapa algoritma pergantian *cache* tentu dapat menyebabkan perbedaan tingkat penghematan *traffic data* yang digunakan pada sebuah jaringan komputer. Oleh karena itu, pada penelitian ini diusulkan analisa perbandingan penghematan *traffic data* dengan menerapkan metode *Least Recently Used* (LRU), metode *Least Frequently Used with Dynamic Aging* (LFUDA), dan metode *Greedy-Dual Size Frequency* (GDSF) pada *proxy server* dengan studi kasus jaringan lokal gedung BAA-BAPSI Universitas Sebelas Maret.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah “Bagaimana melakukan analisa perbandingan penghematan *traffic data* dengan menerapkan metode *Least Recently Used* (LRU), metode *Least Frequently Used with Dynamic Aging* (LFUDA), dan metode *Greedy-Dual Size Frequency* (GDSF) pada *proxy server* jaringan lokal gedung BAA-BAPSI Universitas Sebelas Maret?”.

1.3 Batasan Masalah

Berdasarkan latar belakang dan rumusan masalah yang telah dikemukakan, penelitian ini mempunyai batasan-batasan masalah sebagai berikut:

1. Menggunakan konsep *forward proxy server* dan *non-transparent proxy server*
2. Data diambil dari jaringan lokal gedung BAA-BAPSI Universitas Sebelas Maret.
3. Data yang digunakan untuk pengujian aplikasi berupa *traffic data* pada jaringan lokal gedung BAA-BAPSI Universitas Sebelas Maret dalam bentuk file *access.log*.
4. Pembangunan *proxy server* menggunakan perangkat lunak Squid 3.5.12 dan sistem operasi Linux Ubuntu Server 16.04.
5. Proses *caching* berada di sisi *harddisk* pada *proxy server* yang digunakan.
6. Parameter yang diuji adalah *Hit Ratio*, *Byte Hit Ratio*, dan persentase penghematan *traffic data*.
7. *Proxy server* hanya melakukan proses *caching* terhadap protokol HTTP dan HTTPS.
8. Pengujian dilakukan dengan simulasi 40 *user*.

1.4 Tujuan Penelitian

Dilakukannya penelitian ini bertujuan untuk melakukan analisa perbandingan penghematan *traffic data* dengan menerapkan metode *Least Recently Used* (LRU), metode *Least Frequently Used with Dynamic Aging* (LFUDA), dan metode *Greedy-Dual Size Frequency* (GDSF) pada pergantian *cache* di *proxy server* untuk kemudian didapatkan persentase penghematan *traffic data* dari tiap-tiap metode yaitu *Least Recently Used* (LRU), *Least Frequently Used with Dynamic Aging* (LFUDA), dan *Greedy-Dual Size Frequency* (GDSF) yang kemudian akan diterapkan pada *proxy server* jaringan lokal gedung BAA-BAPSI Universitas Sebelas Maret.

1.5 Manfaat Penelitian

Manfaat dari adanya penelitian ini adalah diharapkan mampu memberikan rekomendasi metode *cache replacement* antara metode *Least Recently Used* (LRU), metode *Least Frequently Used with Dynamic Aging* (LFUDA), atau metode *Greedy-Dual Size Frequency* (GDSF)) untuk diterapkan pada *proxy server* jaringan lokal gedung BAA-BAPSI Universitas Sebelas Maret.

1.6 Sistematika Penulisan

Berikut adalah sistematika penulisan laporan tugas akhir yang dibuat dengan uraian sebagai berikut:

BAB I PENDAHULUAN

Pada Bab I diuraikan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat, metodologi dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada Bab II dijelaskan dasar teori yang mendasari pembahasan dan penelitian terkait yang pernah dilakukan.

BAB III METODOLOGI PENELITIAN

Pada Bab IV dijelaskan pembahasan mengenai metodologi penelitian yang berupa studi literatur, perencanaan dan analisis kebutuhan, perancangan *proxy server*, implementasi perancangan *proxy server*, pengumpulan data, penarikan hasil, dan penarikan kesimpulan.

BAB IV PEMBAHASAN

Pada Bab IV diuraikan penyelesaian rumusan masalah dengan menggunakan metodologi penelitian guna mencapai tujuan penelitian.

BAB V PENUTUP

Pada Bab V berisi kesimpulan dan saran yang berguna untuk pengembangan penelitian kedepannya.

BAB II

TINJAUAN PUSTAKA

2.1 Dasar Teori

2.1.1 Proxy Server

Proxy server adalah sebuah sistem komputer yg berada di antara *client* yang melakukan *request* objek dari *server* tujuan yang melayani *request* objek. *Proxy server* memberikan fasilitas komunikasi antara *client* dan *server* tujuan tanpa melakukan perubahan *request* atau balasan dari *request* tersebut. Saat *client* melakukan *request* objek dari server tujuan, *proxy server* akan mengambil alih koneksi *client* dan mewakili dirinya sebagai *client* ke server tujuan serta meminta sumber daya atas nama *client*. Jika jawaban *request* diterima, proxy server mengembalikan kepada *client* dan kemudian menyimpan hasil balasan tersebut kedalam sebuah *cache* (Witono, 2013). *Proxy server* diletakkan diantara aplikasi *client* dan aplikasi server tersebut, dapat digunakan untuk mengendalikan maupun memonitor lalu lintas paket data yang melewatinya (Wagito, 2007).

Fungsi dari sebuah proxy server adalah sebagai berikut (Wagito, 2007):

1. *Connection sharing*

Proxy server menjadi sebuah *gateway* yang menjadi penghubung antara jaringan lokal dengan jaringan luar bagi *client* dibawahnya.

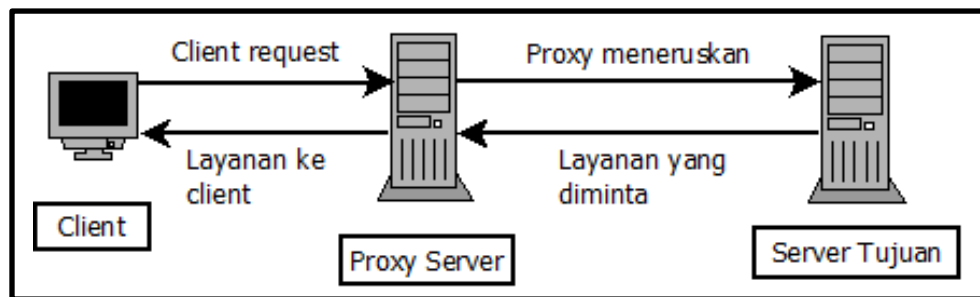
2. *Filtering*

Proxy server sebagai filtering digunakan untuk melindungi jaringan lokal terhadap serangan dari jaringan luar.

3. *Caching*

Proxy server akan menyimpan konten yang diminta oleh *client* ke dalam penyimpanan *proxy server*. Sehingga, ketika *client* melakukan *request* yang mengandung konten yang sama dengan konten yang tersimpan dalam *proxy server*, maka konten tersebut tidak perlu diminta ke *server* asli, cukup diambil dari *proxy server*.

Berikut ini adalah cara kerja dari sebuah *proxy server* sebagai *caching* yang dapat dilihat pada gambar 2.1 berikut:



Gambar 2.1 Cara kerja *proxy server* (Bramantyo, 2016)

Ketika *client* melakukan sebuah *request* maka *proxy server* akan mencari di *local cache* terlebih dahulu, apakah *request* tersebut ada atau tidak, jika ada, maka *proxy server* akan memberikan hasil tersebut ke *client* tersebut, namun jika tidak ada, maka *proxy server* akan meneruskan ke *server* yang dituju seolah-olah *proxy server* tersebut yang memberikan *request* ke *server* asli lalu. Ketika mendapatkan balasan maka akan dilakukan proses *caching* ke *cache* lokal lalu diteruskan ke *client* yang melakukan *request* tersebut. (Wagito, 2007).

2.1.2 Squid

Menurut (Raifudin, 2008), squid merupakan mesin *caching proxy* untuk *client web*, seperti HTTP, HTTPS, FTP, gopher dan layanan sejenis lainnya. Squid mampu menurunkan konsumsi *bandwidth* sekaligus mempercepat waktu respons. Ini terwujud dengan melakukan *caching* halaman web dan menggunakan ulang halaman yang sering dikunjungi, serta squid dapat menyaring situs-situs yang boleh diakses. Squid merupakan *software proxy* yang banyak dipakai dan dapat diperoleh secara gratis, squid memiliki banyak fitur yang ditawarkan, juga mendukung SSL, *extensive access control*, dan *logging request* yang lengkap.

Untuk melakukan monitoring *traffic data client* yang melewati squid, maka akan terdapat kode *output* / keluaran sebagai berikut yang dapat dilihat pada table 2.1 berikut:

Tabel 2.1 Kode *output* squid (Bramantyo, 2016)

Kode	Keterangan
TCP_HIT	Objek yang diminta ada dalam <i>cache</i> dan berhasil diantar ke <i>client</i>
TCP_MISS	Objek yang diminta tidak ada dalam <i>cache</i>
TCP_REFRESH_UNMODIFIED	Objek yang diminta telah kadaluarsa. Menghasilkan kode 304
TCP_REFRESH_MODIFIED_HIT	Objek yang diminta ada dalam <i>cache</i> namun telah kadaluarsa. <i>Query</i> IMS tidak dapat dijalankan
TCP_REFRESH_HIT	Squid menemukan <i>cache</i> yang diminta oleh <i>client</i> dan melakukan validasi ke <i>server</i> tujuan, tetapi <i>server</i> tujuan mengirimkan kode <i>not modified</i> . Sehingga squid menganggap <i>cache</i> tersebut masih valid
TCP_REFRESH_MISS	Objek yang diminta sudah kadaluarsa
TCP_MEM_HIT	Objek yang diminta ada dalam <i>cache memory</i> , bukan dalam <i>disk</i>
NONE	Terjadi error

2.1.2.1 *Hit Ratio*

Hit ratio mendeskripsikan perbandingan jumlah *request* yang *hit* (dapat dilayani oleh *proxy*) dibanding dengan total *request* dari *client*. Semakin tinggi *Hit Ratio* maka semakin besar penghematan *traffic data* yang terjadi. (Sadewa, Anif, & Hidayat, 2016)

2.1.2.2 *Byte Hit Ratio*

Byte hit ratio adalah perbandingan *byte* yang diterima dari *cache* dengan *byte* keseluruhan yang diterima *client*. Perhitungan *Byte Hit Ratio* berdasarkan jumlah *byte request* dari *client* yang *hit*. Dengan demikian semakin besar *byte hit ratio*, maka semakin hemat *traffic data* yang digunakan. (Hardianto, 2015)

2.1.2.3 *Access.log*

Menurut website squid (www.squid-cache.org), access log adalah sebuah file yang dibentuk dari konfigurasi Squid. Access log berfungsi sebagai *reporting akses traffic data* dari *client* yang melalui *proxy server*. Berikut ini adalah contoh isi dari access log:

<u>1726351927.011</u>	<u>19131</u>	<u>10.10.10.2</u>	<u>TCP_MISS/200</u>	<u>360</u>
I	II	III	IV V	VI
<u>GET</u>	<u>http://www.xyz.com</u>	<u>DIRECT/63.211.210.20</u>		
VII	VIII	IX	X	
<u>text/html</u>				
XII				

Keterangan:

- I = *Timestamp*
- II = *Response time*
- III = *IP address client*
- IV = *Code status*
- V = *HTTP status*

VI	=	<i>Transfer size</i>
VII	=	<i>HTTP method</i>
VIII	=	<i>Url</i>
IX	=	<i>Hierarchy</i>
X	=	<i>IP address server</i>
XI	=	<i>Content type</i>

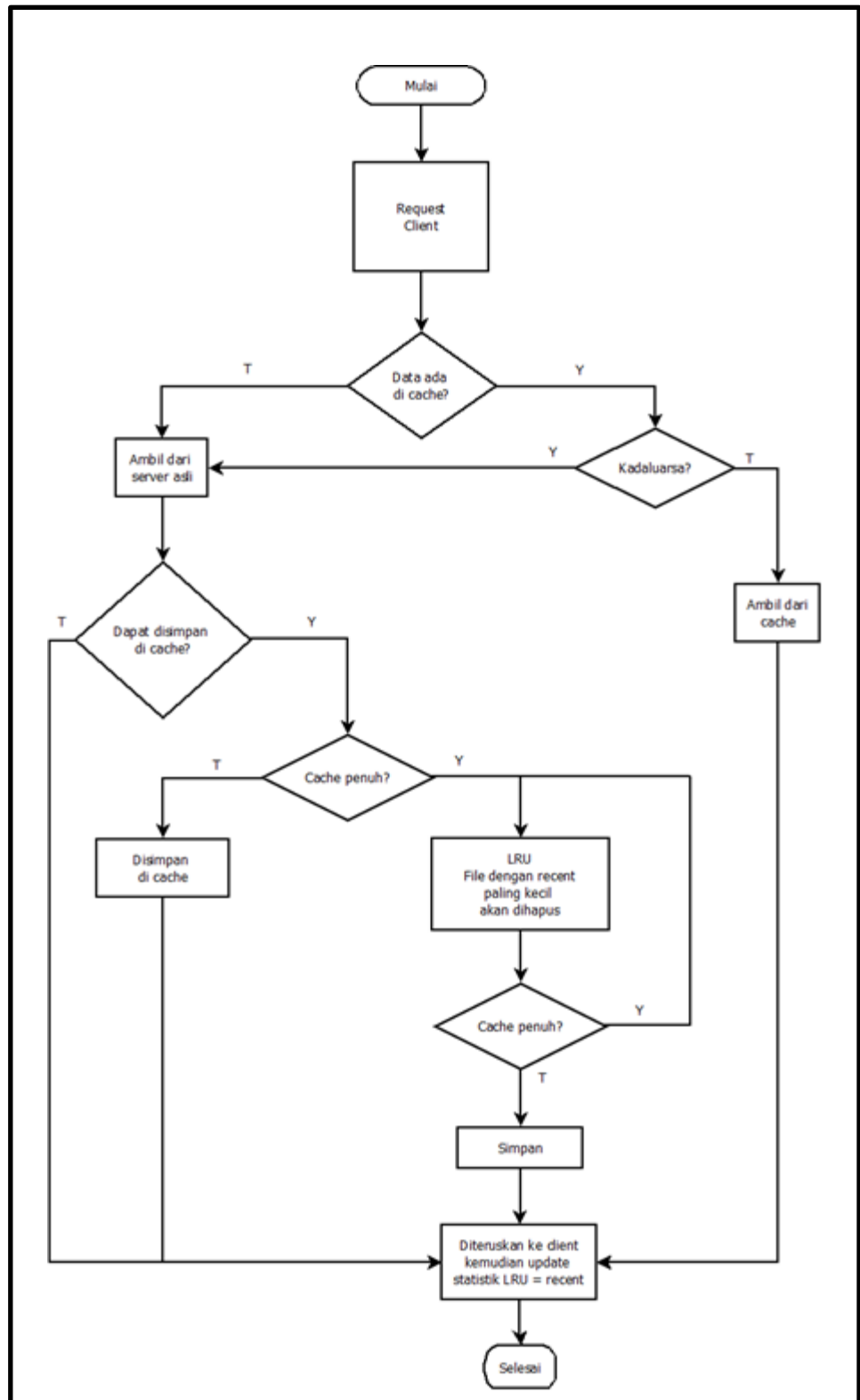
2.1.3 Algoritma Penggantian *Cache*

Tata aturan yang mengatur bagaimana pergantian suatu *cache* objek dalam sebuah penyimpanan disebut dengan algoritma penggantian *cache*. Dalam squid, objek disimpan pada *memory* dan *disk*. Objek berisikan isi URL yang diminta *client* dan disimpan dalam bentuk file binary, masing-masing objek mempunyai metadata yang sebagian dari isinya disimpan di dalam memori untuk memudahkan melacak dimana letak objek dan apa isi dari objek tersebut. Adapun hal yang harus diamati untuk optimasi Squid adalah kapasitas harddisk untuk *cache*. Semakin besar kapasitas *cache*, berarti semakin lama umur objek tersebut bisa disimpan, jika pemakaian harddisk sudah mendekati batas atas (*cache_swap_high*) maka penggantian objek akan semakin sering dilakukan. Squid memiliki beberapa algoritma penggantian objek *cache*, antara lain *Least Recently Used* (LRU), *Least Frequently Used with Dynamic Aging* (LFUDA), dan *Greedy-Dual Size Frequency* (GDSF). (Witono, 2013).

Berikut ini penjelasan mengenai ketiga algoritma penggantian *cache* *Least Recently Used* (LRU), *Least Frequently Used with Dynamic Aging* (LFUDA), dan *Greedy-Dual Size Frequency* (GDSF). (Witono, 2013):

a. *Least Recently Used* (LRU)

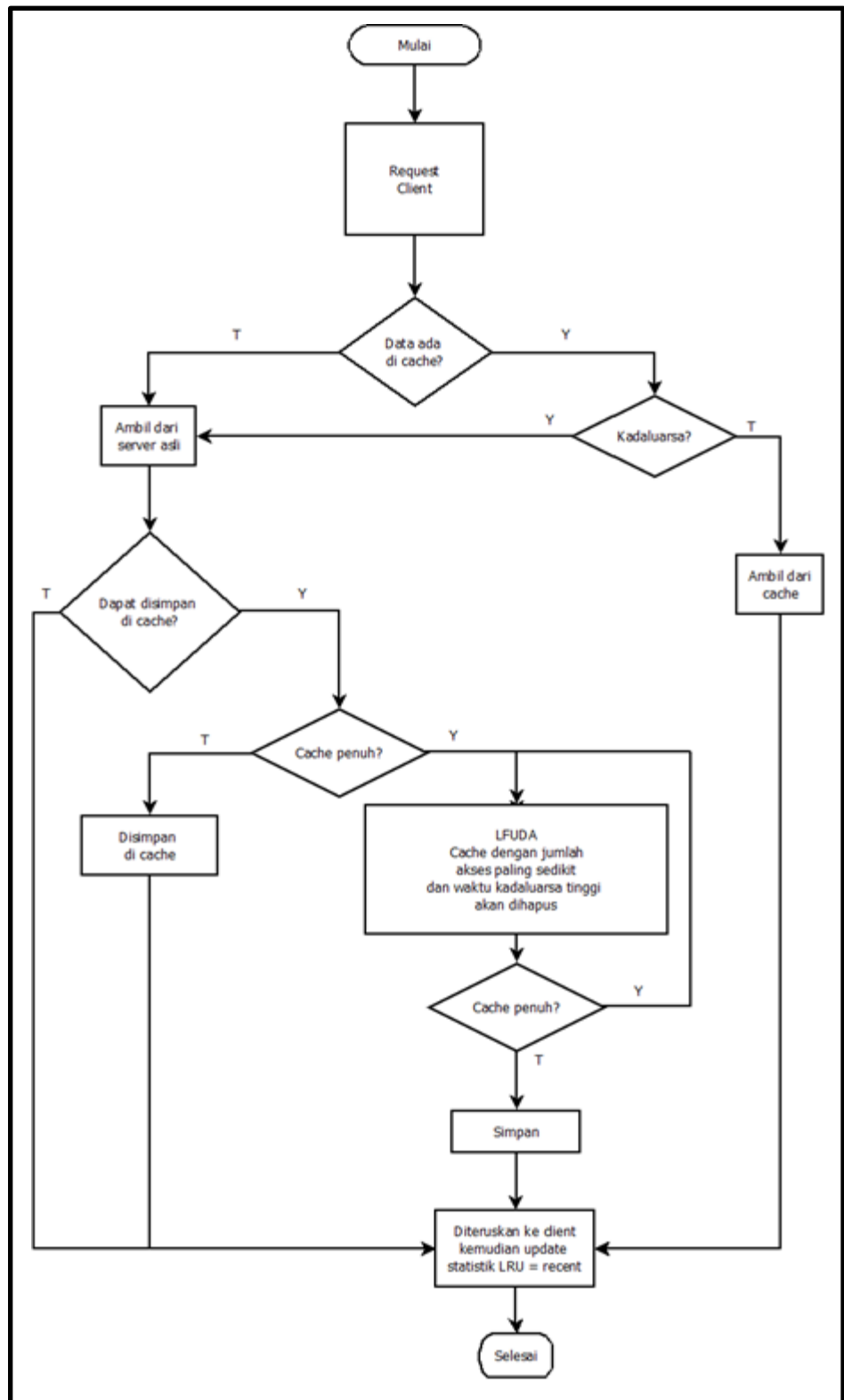
Cara kerja algoritma *Least Recently Used* (LRU) adalah menggantikan halaman yang sudah tidak digunakan dalam jangka waktu yang paling lama (Sadewa, Anif, & Hidayat, 2016). Untuk alur algoritma *Least Recently Used* (LRU) adalah sebagai berikut:



Gambar 2.2 Flowchart algoritma *Least Recently Used* (LRU). (Sadewa, Anif, & Hidayat, 2016)

b. *Least Frequently Used with Dynamic Aging* (LFUDA)

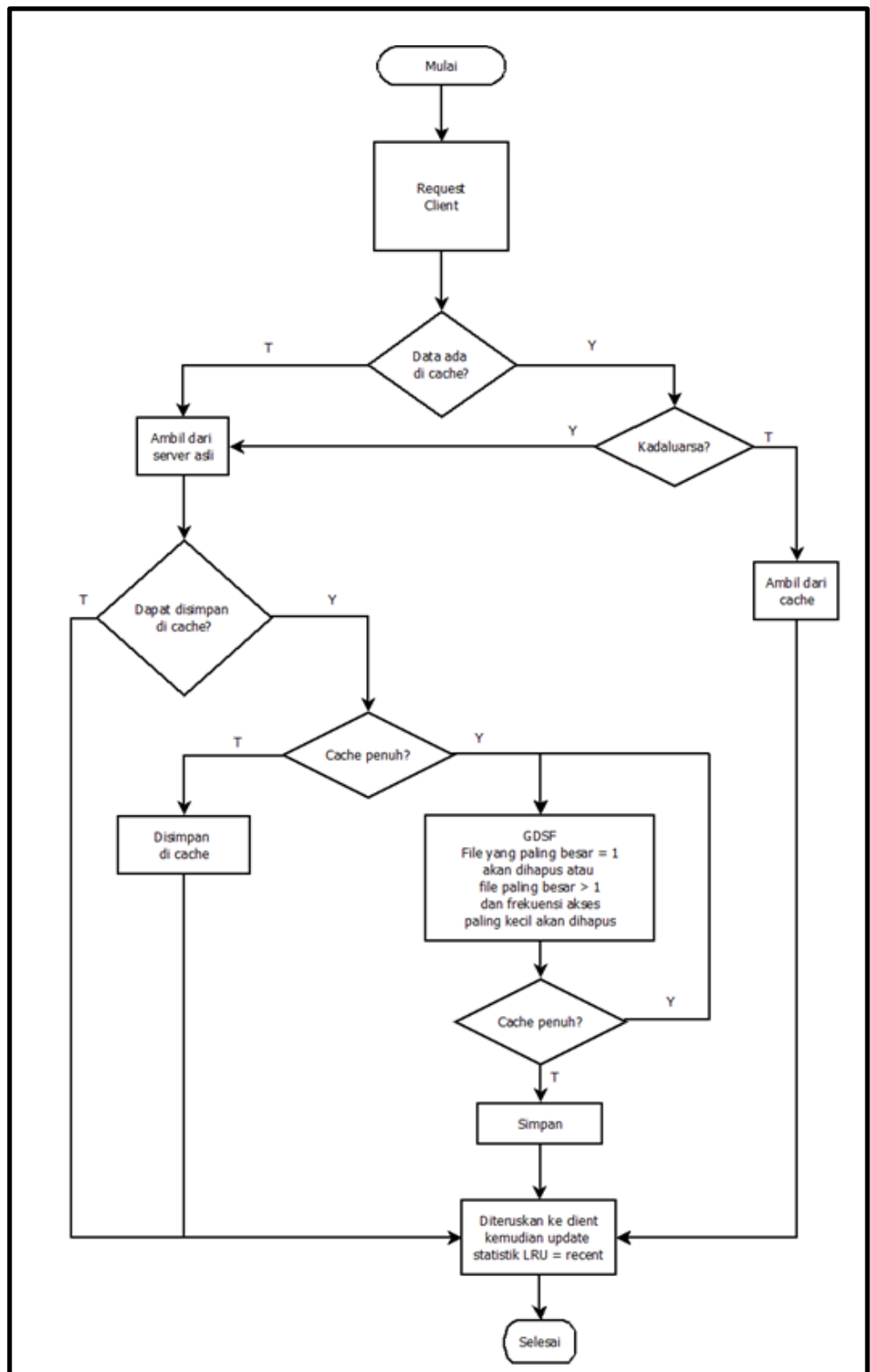
Least Frequently Used with Dynamic Aging (LFUDA) adalah sebuah aturan penggantian objek yang menyimpan objek populer di dalam *cache* dan mencegah objek lain yang kurang populer untuk dapat masuk ke *cache*. *Least Frequently Used with Dynamic Aging* (LFUDA) mengganti objek yang telah diakses dalam jumlah yang paling sedikit. Strategi ini menjaga objek supaya tetap populer dan mengganti objek yang jarang digunakan. *Least Frequently Used with Dynamic Aging* (LFUDA) dikembangkan dalam rangka mengakomodasi kekurangan yang terjadi pada algoritma *Least Frequently Used* (LFU), dimana dalam variasi *dynamic aging* ini memperhitungkan usia objek (Khosrowpour, 2005). Untuk alur algoritma *Least Frequently Used with Dynamic Aging* (LFUDA) adalah sebagai berikut:



Gambar 2. 3 Flowchart algoritma *Least Frequently Used with Dynamic Aging* (LFUDA). (Khosrowpour, 2005)

c. *Greedy-Dual Size Frequency* (GDSF)

Cara kerja algoritma *Greedy-Dual Size Frequency* (GDSF) adalah objek yang memiliki ukuran lebih besar akan mendapatkan prioritas lebih tinggi untuk dihapus. *Greedy-Dual Size Frequency* (GDSF) cenderung untuk mengganti objek yang berukuran lebih besar dahulu, hal ini berfungsi untuk meminimalisir jumlah objek yang telah dikeluarkan. *Greedy-Dual Size Frequency* (GDSF) menggabungkan perhitungan frekuensi dalam pembuatan keputusan, sehingga objek besar yang populer akan memiliki kesempatan bertahan yang lebih baik tanpa harus dikeluarkan secara terus menerus (Bramantyo, 2016). Untuk alur algoritma *Greedy-Dual Size Frequency* (GDSF) adalah sebagai berikut:



Gambar 2.4 Flowchart algoritma *Greedy-Dual Size Frequency* (GDSF). (Bramantyo, 2016)

2.2 Penelitian Terkait

Berikut adalah beberapa penelitian yang berkaitan dengan penelitian yang akan diajukan:

1. **Bramantyo, F. W. (2016). *Perbandingan Hit Ratio dan Byte Ratio Squid untuk Proxy Server dalam Penghapusan File Caching Berdasarkan Metode LRU, LFUDA, dan GDSF*. Yogyakarta.**

Penelitian ini membahas tentang Perbandingan *Hit Ratio* dan *Byte Ratio* Squid untuk *Proxy Server* dalam Penghapusan *File Caching* Berdasarkan Metode *Least Recently Used* (LRU), *Least Frequently Used with Dynamic Aging* (LFUDA), dan *Greedy-Dual Size Frequency* (GDSF). Penelitian tersebut bertujuan untuk mengetahui perbandingan *Hit Ratio* dan *Byte Ratio* antara ketiga metode tersebut. Pada penelitian ini didapatkan hasil bahwa metode penghapusan data *Greedy-Dual Size Frequency* (GDSF) pada squid proxy server akan lebih baik digunakan untuk request website dengan file-file yang kecil. Pada metode penghapusan data *Least Frequently Used with Dynamic Aging* (LFUDA) pada squid proxy server persentasenya relative sama, hal tersebut dapat dikarenakan skenario pada pengujian kali ini menggunakan alamat website yang telah ditentukan dan request website dilakukan secara teratur disetiap komputer. Sedangkan pada kategori *Byte Ratio*, metode *Least Frequently Used with Dynamic Aging* (LFUDA) akan mendapatkan byte hit ratio yang lebih besar ketika mengakses website yang dilakukan lebih dominan pada website besar. Untuk metode *Least Recently Used* (LRU) kurang tepat ditarik kesimpulan.

2. **Zotano, M. G., Sanz, J. G., & Pavón, J. (2015). *Impact of traffic distribution on web cache performance*. *International Journal of Web Engineering and Technology*.**

Penelitian ini membahas tentang dampak dari distribusi trafik pada performa *web cache*. Penelitian ini menggunakan 6 metode pengujian, yaitu LRU (*Last Recently Used*), *Perfect LFU*, *LFU Dynamic Aging*, *Greedy Dual Size* (GDS), GDSF (*Greedy Dual Size Frequency*), dan *Perfect GDSF*. Pada penelitian ini didapatkan hasil bahwa variasi dari GDSF yaitu perfect GDSF mempertimbangkan frekuensi

permintaan objek dari semua objek yang diterima oleh *cache* yang tentunya berdampak pada semakin tinggi nya *Hit Ratio* pada metode *perfect* GDSF dibanding dengan metode GDSF, GDS, LFU, LFU-DA dan LRU. Penulis juga menyatakan kekurangan dari penelitian ini, yaitu pada *caching policies* yang digunakan tidak mempertimbangkan waktu kadaluwarsa (*expired time*) sebuah objek. Dengan demikian, hasilnya bagus untuk distribusi konten statis. Namun, untuk konten dinamis memerlukan strategi yang berbeda untuk melakukan penyegaran objek dalam *cache*.

3. Sadewa, E. E., Anif, M., & Hidayat, S. S. (2016, Oktober). Analisis Komparasi Cache Replacement Algorithm untuk Meningkatkan Kinerja Proxy Server Pasar Grosir Pekalongan. Seminar Nasional Terapan Riset Inovatif.

Penelitian ini membahas mengenai analisis komparasi *cache replacement algorithm* untuk meningkatkan kinerja *proxy server* pasar grosir pekalongan. Pada penelitian ini, peneliti menggunakan metode penghapusan *cache Least Frequently Used with Dynamic Aging* (LFUDA) dan *Greedy-Dual Size Frequency* (GDSF) untuk mengelola *cache* yang ada dalam *proxy server*. Penelitian ini mendapatkan hasil:

- Rata-rata Hit ratio algoritma LFUDA adalah 10,74 % dan rata-rata hit ratio *Greedy-Dual Size Frequency* (GDSF) adalah 14,01%.
- Rata-rata byte hit ratio algoritma LFUDA adalah 5,30 % dan rata-rata byte hit ratio GDSF adalah 5,78%,
- Karakteristik akses dari client pasar grosir Pekalongan menunjukkan bahwa lebih dari 93% akses Internet adalah browsing dengan object kurang dari 100KB.
- Berdasarkan analisis perbandingan serta menimbang karakteristik akses client jaringan pasar grosir Pekalongan, dipilih algoritma *Greedy-Dual Size Frequency* (GDSF) sebagai algoritma penghapusan *cache* terbaik

4. **Tanwir, Hendrantoro, G., & Affandi, A. (2017). Combination of FIFO-LRU Cache Replacement Algorithms on Proxy Server to Improve Speed of Response to Object Request from Clients. *ARNP Journal of Engineering and Applied Sciences*, 710-715.**

Penelitian ini membahas mengenai kombinasi metode FIFO-LRU pada algoritma pergantian *cache* pada sebuah *proxy server* untuk meningkatkan waktu respon objek yang diminta oleh *client*. Peneliti menggunakan metode LRU yang dikombinasikan dengan metode FIFO pada metode pergantian *cache* pada *proxy server*, kemudian dilakukan perbandingan respon time dengan menerapkan metode LRU tanpa modifikasi dan metode FIFO tanpa modifikasi. Penelitian ini mendapatkan hasil:

- Penerapan algoritma kombinasi FIFO-LRU membawa kinerja perbaikan yang relatif besar, dibandingkan dengan algoritma LRU. Persentase tingkat hitung algoritma FIFO-LRU meningkat menjadi 73% untuk ukuran cache 200 MB dan meningkat menjadi 85% pada ukuran cache 100 MB.
- Persentase hit ratio meningkat, dan waktu respon menurun. Akibatnya, waktu respon lebih pendek. Berdasarkan temuan ini, penerapan kombinasi FIFO-LRU sangat dianjurkan.

Untuk mengetahui perbandingan penelitian terkait diatas, dapat dilihat pada table 2.2 berikut:

Tabel 2.2 Penelitian Terkait

No	Referensi	Judul	Metode	Hasil / Temuan
1	(Bramantyo, 2016)	Perbandingan <i>hit ratio</i> dan <i>byte hit ratio squid</i> untuk <i>proxy server</i> dalam penghapusan file <i>caching</i> berdasarkan metode LRU, LFUDA, dan GDSF	<ul style="list-style-type: none"> • LRU (<i>Least Recently Used</i>) • LFUDA (<i>Least Frequently Used with Dynamic Aging</i>) • GDSF (<i>Greedy-Dual Size Frequency</i>) 	<ol style="list-style-type: none"> 1. Metode penghapusan data GDSF pada <i>squid proxy server</i> akan lebih baik digunakan untuk <i>request</i> website dengan file-file yang kecil. 2. Pada kategori Byte Ratio, metode LFUDA akan mendapatkan byte hit ratio yang lebih besar ketika mengakses website yang dilakukan lebih dominan pada website besar. 3. Untuk metode LRU kurang tepat ditarik kesimpulan
2	(Zotano, Sanz, & Pavón, 2015)	<i>Impact of Traffict Distribution on Web Cache Performance</i>	<ul style="list-style-type: none"> • LRU (<i>Last Recently Used</i>) • <i>Perfect</i> LFU • LFU <i>Dynamic Aging</i> • <i>Greedy Dual Size</i> (GDS) • GDSF (<i>Greedy Dual Size Frequency</i>) • <i>Perfect</i> GDSF 	Sebuah variasi dari GDSF yaitu <i>perfect</i> GDSF mempertimbangkan frekuensi permintaan objek dari semua objek yang diterima oleh cache yang tentunya berdampak pada semakin tinggi nya Hit Ratio pada metode perfect GDSF dibanding dengan metode GDSF, GDS, LFU, LFU-DA dan LRU.

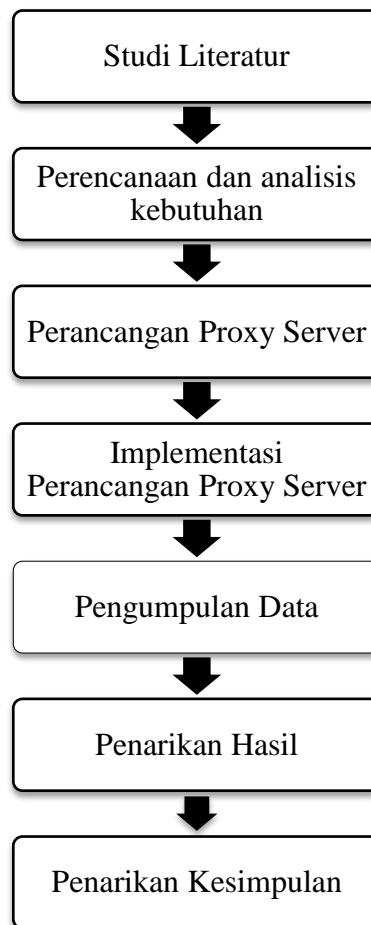
3	(Sadewa, Anif, & Hidayat, 2016)	Analisis Komparasi <i>Cache Replacement Algorithm</i> untuk Meningkatkan Kinerja Proxy Server Pasar Grosir Pekalongan	<ul style="list-style-type: none"> • LFUDA • GDSF 	<ul style="list-style-type: none"> • Rata-rata <i>Hit ratio</i> algoritma LFUDA adalah 10,74 % dan rata-rata <i>hit ratio</i> GDSF adalah 14,01% • Rata-rata <i>byte hit ratio</i> algoritma LFUDA adalah 5,30 % dan rata-rata <i>byte hit ratio</i> GDSF adalah 5,78% • Karakteristik akses dari <i>client</i> pasar grosir Pekalongan menunjukkan bahwa lebih dari 93% akses Internet adalah <i>browsing</i> dengan <i>object</i> kurang dari 100KB. • Berdasarkan analisis perbandingan serta menimbang karakteristik akses <i>client</i> jaringan pasar grosir Pekalongan, dipilih algoritma GDSF sebagai algoritma penghapusan <i>cache</i> terbaik
---	---------------------------------	---	---	--

4	(Tanwir, Hendrantoro, & Affandi, 2017)	<i>Combination of FIFO-LRU Cache Replacement Algoritms on Proxy Server to Improve Speed of Response to Object Request from Clients</i>	<ul style="list-style-type: none"> • FIFO • LRU • FIFO-LRU 	<p>1. Hasil dari penelitian ini secara umum menunjukkan bahwa penerapan algoritma kombinasi FIFO-LRU membawa kinerja perbaikan yang relatif besar, dibandingkan dengan algoritma LRU. Persentase tingkat hitung algoritma FIFO-LRU meningkat menjadi 73% untuk ukuran cache 200 MB dan meningkat menjadi 85% pada ukuran cache 100 MB.</p> <p>2. Persentase <i>hit ratio</i> meningkat, dan waktu respon menurun. Akibatnya, waktu respon lebih pendek. Berdasarkan temuan ini, penerapan kombinasi FIFO-LRU sangat dianjurkan.</p>
---	--	--	---	---

BAB III

METODOLOGI PENELITIAN

Metode yang digunakan dalam penelitian ini terdiri dari:



Gambar 3. 1 Metodologi penelitian

3.1 Studi Literatur

Pada tahapan studi literatur dilakukan pencarian bahan materi yang berkaitan dengan penelitian ini. Studi literatur dapat bersumber pada jurnal internasional maupun nasional yang sudah diseminarkan ke publik, hasil skripsi mahasiswa, maupun

kunjungan ke instansi yang menerapkan *proxy server* dalam jaringan internet yang digunakan.

3.2 Perencanaan dan Analisis Kebutuhan

Pada tahapan perencanaan dan analisis kebutuhan dilakukan analisis terhadap perangkat, topologi jaringan, parameter pengujian, dan lingkungan implementasi *proxy server* yang akan dibangun yaitu di jaringan lokal gedung BAA-BAPSI Universitas Sebelas Maret yang diterapkan untuk jaringan lokal gedung BAA-BAPSI Universitas Sebelas Maret.

Untuk perangkat, dilakukan analisis kebutuhan perangkat keras maupun perangkat lunak yang akan digunakan sebagai *proxy server*. Sedangkan untuk analisa topologi dilakukan analisa penempatan posisi *proxy server* dalam topologi jaringan yang digunakan (dalam penelitian ini adalah topologi jaringan lokal gedung BAA-BAPSI Universitas Sebelas Maret).

Untuk Analisis parameter pengujian dilakukan analisis parameter yang akan diujikan pada *proxy server* pada penelitian ini. Parameter tersebut adalah jumlah total *traffic data* dari *proxy server* (total *request* dari *client*) dan total *traffic data* (*request* dari *client* yang terdapat di dalam *cache proxy*) yang keluar masuk pada *proxy server* ke *client* yang kemudian akan diolah untuk diambil persentase penghematan *traffic data*.

3.3 Perancangan Proxy Server

Pada tahapan perancangan dilakukan perancangan topologi jaringan, parameter konfigurasi *proxy server*, dan skema pengujian *proxy server*.

Perancangan topologi dilakukan dengan memposisikan *proxy server* kedalam topologi jaringan lokal gedung BAA-BAPSI Universitas Sebelas Maret. Sedangkan untuk parameter konfigurasi dilakukan perancangan beberapa konfigurasi yang akan diterapkan pada aplikasi *squid*. Konfigurasi tersebut meliputi konfigurasi pergantian *cache*, metode pergantian *cache* yang digunakan, kapasitas penyimpanan *cache*, dan konfigurasi untuk memonitoring *traffict data proxy server*. Berikutnya adalah perancangan skema pengujian. Skema pengujian yang dilakukan adalah menguji berapa persen *traffic data* yang mampu dihemat oleh jaringan lokal gedung BAA-BAPSI Universitas Sebelas Maret dengan menggunakan *proxy server*. Pada tahapan skema

pengujian ini, akan diterapkan metode penggantian *cache Least Recently Used* (LRU), metode *Least Frequently Used with Dynamic Aging* (LFUDA), dan metode *Greedy-Dual Size Frequency* (GDSF) pada *proxy server*.

Untuk spesifikasi perangkat yang digunakan sebagai *proxy server* adalah sebagai berikut:

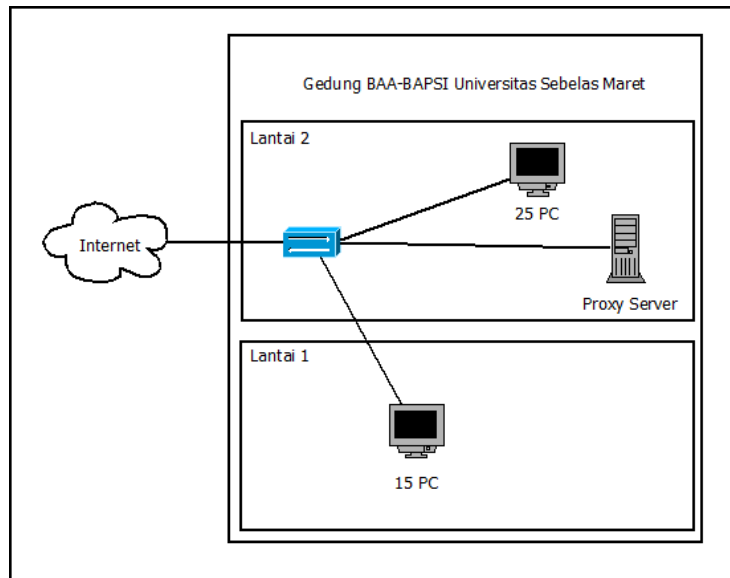
Tabel 3. 1 Spesifikasi perangkat

Spesifikasi	
Motherboard	Biostar G41D3+
Prosesor	Intel Dual Core
RAM	1 GB
Harddisk	250 GB
Alokasi ukuran penyimpanan cache	10 GB

3.4 Implementasi Perancangan *Proxy Server*

Implementasi perancangan ini meliputi implementasi topologi jaringan ke dalam lingkungan uji, implementasi perancangan konfigurasi *proxy server*, dan implementasi skema pengujian *proxy server*.

Implementasi selanjutnya adalah memposisikan *proxy server* di topologi jaringan lokal gedung BAA-BAPSI Universitas Sebelas Maret berada sejajar dengan *client*, hal tersebut karena topologi yang digunakan adalah topologi *star*.



Gambar 3.2 Topologi jaringan

Pada topologi tersebut, client akan dikonfigurasi manual untuk menggunakan *proxy server* yang ada di jaringan lokal gedung BAA-BAPSI Universitas Sebelas Maret.

3.5 Pengumpulan Data

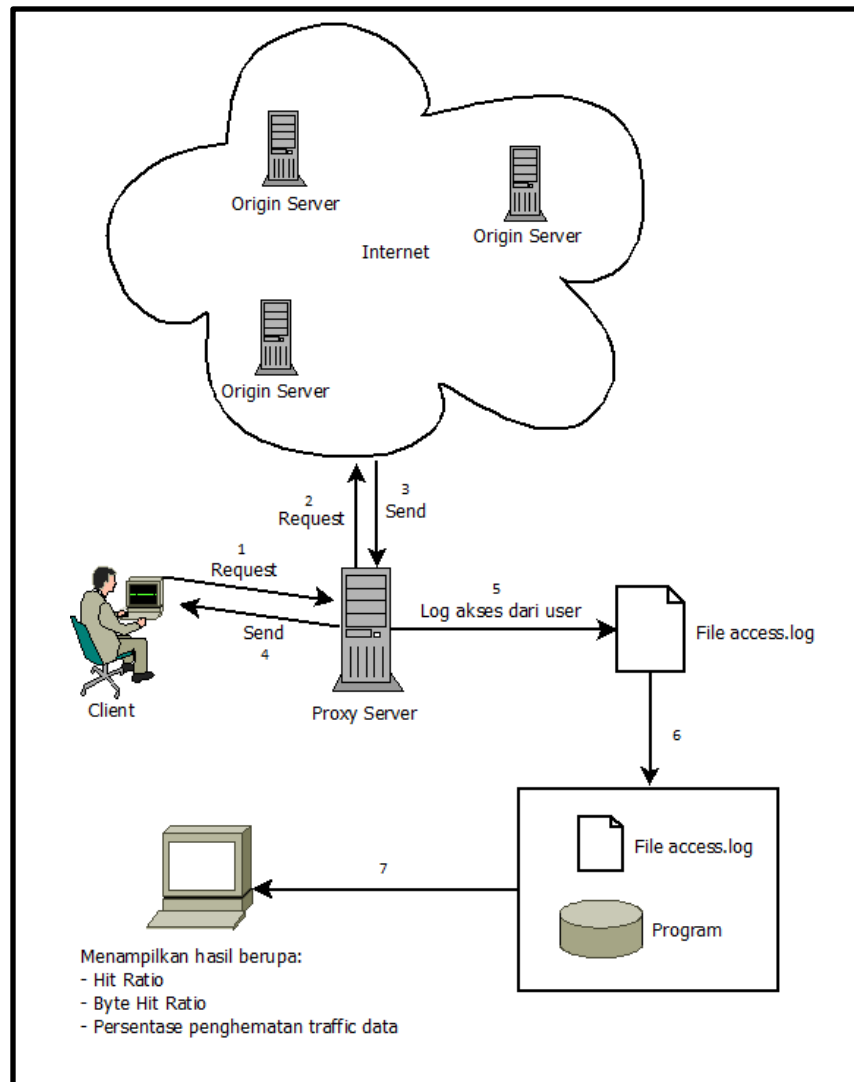
Pada tahapan pengumpulan data, hasil dari implementasi perancangan tersebut dilakukan simulasi sesuai dengan parameter pengujian yaitu *Hit Ratio* dan *Byte Hit Ratio* untuk kemudian didapatkan persentase penghematan *traffic data* pada jaringan lokal gedung BAA-BAPSI Universitas Sebelas Maret. Pengujian tersebut dilakukan di lingkungan implementasi *proxy server* yaitu di jaringan lokal gedung BAA-BAPSI Universitas Sebelas Maret dengan menggunakan 40 *user*.

Pengujian dilakukan dengan cara simulasi yaitu disimulasikan 40 user melakukan request ke 10 laman website yang terdapat pada tabel 3.2 berikut:

Tabel 3. 2 Daftar Website

Url	Protokol
https://www.kompas.com	https
https://www.okezone.com	https
https://www.viva.co.id	https
https://www.sindonews.com	https
https://www.kompasiana.com	https
http://beritasatu.com	http
http://www.republika.co.id	http
http://surakarta.go.id	http
http://www.thejakartapost.com	http
http://www.metrotvnews.com	http

Total waktu pengujian adalah selama 6 hari, yaitu 2 hari dengan menggunakan metode *Least Recently Used* (LRU), 2 hari menggunakan metode *Least Frequently Used with Dynamic Aging* (LFUDA), dan 2 hari dengan menggunakan metode *Greedy-Dual Size Frequency* (GDSF). Setiap *client* melakukan request secara otomatis dengan bantuan *program batch* ke website pada tabel 3.2 dengan persentase 50% akses website dengan protokol http dan 50% akses website dengan protokol https. Hasil dari pengujian dengan simulasi tersebut akan diperoleh 3 buah file access.log yaitu file access.log dari *proxy server* dengan menggunakan algoritma *Least Recently Used* (LRU), file access.log dari *proxy server* dengan menggunakan algoritma *Least Frequently Used with Dynamic Aging* (LFUDA), dan file access.log dari *proxy server* dengan menggunakan algoritma *Greedy-Dual Size Frequency* (GDSF) yang kemudian akan diolah dengan program berbasis web dengan kombinasi *bash programming* untuk dapat memberikan hasil berupa *hit ratio*, *byte hit ratio*, dan persentase penghematan *traffic data*. Untuk alur pengujiannya dapat dilihat pada gambar berikut:



Gambar 3. 3 Alur pengujian dan pengolahan data

Keterangan:

1. Proses *client* melakukan request data ke *proxy server*
2. *Proxy server* melakukan proses *checking* ke *cache* apakah data yang di-*request client* ada dalam *cache*. Jika ya, maka *proxy server* akan melakukan langkah ke 4. Jika tidak, maka *proxy server* akan meneruskan *request* tersebut ke *server* tujuan (asli).
3. *Server* asli mengirimkan data yang di-*request* ke *proxy server*
4. *Proxy server* mengirimkan data yang di-*request client*

5. Traffic *request* data dari *client* tercatat dalam sebuah file yang bernama *access.log*
6. File *access.log* tersebut kemudian diolah menggunakan aplikasi berbasis website dengan kombinasi *bash programming*
7. Hasil dari pengolahan data tersebut berupa *Hit Ratio*, *Byte Hit Ratio*, dan Persentasi penghematan *traffic data*

3.6 Penarikan Hasil

Pada tahapan penarikan hasil dilakukan pengolahan data yang sudah didapatkan dari proses pengumpulan data dengan menerapkan *filtering* parameter pada file *access.log* dari tiap-tiap algoritma pergantian *cache* dengan menggunakan bahasa pemrograman *bash*. Pada tahapan *filtering*, parameter yang di *filter* adalah *code status* dan *transfer size*. Tujuan dilakukan proses *filtering* pada parameter *code status* adalah untuk mendapatkan *hit ratio* dan parameter *transfer size* untuk mendapatkan *byte hit ratio*.

Syarat suatu request dikatakan “*hit*” (konten yang di-*request* terdapat pada *proxy server*) dapat dilihat pada tabel 3.3:

Tabel 3. 3 Kategori *Request*

<i>Code Status</i>	Keterangan
TCP_HIT	Objek yang diminta ada dalam <i>cache</i> dan berhasil diantarkan ke klien
TCP_REFRESH_HIT	Squid menemukan <i>cache</i> yang diminta oleh klien dan melakukan validasi ke server tujuan, tetapi komputer asal mengirimkan kode <i>not modified</i> . Sehingga squid menganggap <i>cache</i> tersebut masih valid
TCP_REF_FAIL_HIT	Squid menemukan file <i>cache</i> yang diminta klien dan melakukan validasi ke <i>server</i> tujuan, tetapi <i>server</i> asli tidak merespon. Squid menganggap salinan tersebut valid dan memberikannya ke klien

TCP_IMS_HIT	Klien melakukan akses IMS untuk sebuah objek yang berada dalam <i>cache</i> baru
TCP_REFRESH_UNMODIFIED	Objek yang di-request telah kadaluarsa. Menghasilkan kode 304
TCP_REFRESH_UNMODIFIED_HIT	Objek yang diminta ada dalam <i>cache</i> namun telah kadaluarsa. <i>Query</i> IMS tidak dapat dijalankan
TCP_NEGATIVE_HIT	<i>Request</i> untuk objek yang di-cache <i>negative</i> , misalnya 404 not found
TCP_MEM_HIT	Data valid dari objek yang di-request ada dalam <i>cache</i> memori, bukan dalam disk
TCP_OFFLINE_HIT	Objek yang di-request ada dalam <i>cache</i> saat mode <i>offline</i>

Untuk mendapatkan hasil *Hit Ratio*, *Byte Hit Ratio*, dan persentase penghematan *traffict data* dari ketiga algoritma tersebut maka dibuatlah sebuah program berbasis web dengan kombinasi *bash programming* untuk mengolah data (file access.log) sehingga menampilkan informasi mengenai *Hit Ratio*, *Byte Hit Ratio*, dan presentase penghematan *traffic data*.

3.7 Penarikan Kesimpulan

Setelah melakukan penarikan hasil berdasarkan parameter pengujian, langkah berikutnya adalah penarikan kesimpulan. Penarikan kesimpulan memberikan hasil berupa algoritma pergantian *cache* yang lebih bagus (memiliki hit ratio, byte hit ratio, dan persentase penghematan *traffic data* paling besar dari perbandingan ketiga algoritma yaitu *Least Recently Used* (LRU), *Least Frequently Used with Dynamic Aging* (LFUDA), dan *Greedy-Dual Size Frequency* (GDSF) dalam melakukan proses *caching* pada protokol http dan https dengan metode simulasi.

ANALISIS PERBANDINGAN PENGHEMATAN *TRAFFIC DATA* DENGAN MENERAPKAN METODE LRU, METODE LFUDA, DAN METODE GDSF



Gambar 3. 4 Desain program berbasis web

Jadwal Pelaksanaan

No	Kegiatan	Bulan															
		Maret				April				Mei				Juni			
		Minggu ke-															
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	Studi Literatur																
2.	Perencanaan dan Analisis Kebutuhan																
3.	Perancangan Proxy Server																
4.	Implementasi Perancangan Proxy Server																
5.	Pengumpulan Data																
6.	Penarikan Hasil																
7.	Penarikan Kesimpulan																

Daftar Pustaka

- Ali, W., Sulaiman, S., & Ahmad, N. (2014). Performance Improvement of Least-Recently-Used Policy in Web Proxy Cache Replacement Using Supervised Machine Learning. *Int. J. Advance. Soft Comput. Appl.*
- Ayani, R., Teo, Y. M., & Ng, Y. S. (2003). Cache Pollution in Web Proxy Server.
- Bramantyo, F. W. (2016). *Perbandingan Hit Ratio dan Byte Ratio Squid untuk Proxy Server dalam Penghapusan File Caching Berdasarkan Metode LRU, LFUDA, dan GDSF*. Yogyakarta.
- Hardianto, T. (2015). *Analisis Unjuk Kerja Pengaruh Hit Ratio Pada Squid Proxy Terhadap Sumber Daya Komputer Server*. Yogyakarta.
- Khosrowpour, M. (2005). *Encyclopedia of Information Science and Technology* (Vols. 1-5). Idea Group Inc (IGI).
- Nanda, P., Singh, S., & Saini, G. L. (2015). A Review of Web Caching Techniques and Caching Algorithms for Effective and Improved Caching. *International Journal of Computer Applications*, 41-45.
- Raifudin, R. (2008). *Squid Koneksi Anti Mogok*. Yogyakarta: Andi Offset.
- Sadewa, E. E., Anif, M., & Hidayat, S. S. (2016, Oktober). Analisis Komparasi Cache Replacement Algorithm untuk Meningkatkan Kinerja Proxy Server Pasar Grosir Pekalongan. *Seminar Nasional Terapan Riset Inovatif*.
- Squidadm. (2013). Retrieved November 25, 2017, from squid-cache.org: http://www.squid-cache.org/Doc/config/cache_replacement_policy/ diakses pada 20 Maret 2018
- Syamsu, S. (2012). Peningkatan Performa Proxy Server Berbasis Squid dengan Tuning Parameter Konfigurasi. *Jurnal Inspiration*, 41-47.

- Tanwir, Hendrantoro, G., & Affandi, A. (2017). Combination of FIFO-LRU Cache Replacement Algorithms on Proxy Server to Improve Speed of Response to Object Request from Clients. *ARPN Journal of Engineering and Applied Sciences*, 710-715.
- Wagito. (2007). *Panduan Praktikum Jaringan Komputer*. Yogyakarta: STMIK Akakom Yogyakarta.
- Witono, T. (2013). Algoritma Penggantian Cache Sebagai Optimalisasi Kinerja pada Proxy Server.
- Yuisar, Yulianti, L., & H, Y. S. (2015). Analisa Pemanfaatan Proxy Server Sebagai Media Filtering dan Caching Pada Jaringan Komputer. *Jurnal Media Infotama*, 81-90.
- Zotano, M. G., Sanz, J. G., & Pavón, J. (2015). Impact of traffic distribution on web cache performance. *International Journal of Web Engineering and Technology*.