

**ANALISIS PERBANDINGAN KINERJA SQUIDGUARD DAN
DANSGUARDIAN SEBAGAI IMPLEMENTASI
WEB FILTERING**

LAPORAN TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika**



Disusun oleh:

Nama : Abdul Gafur

No. Mahasiswa : 05 523 295

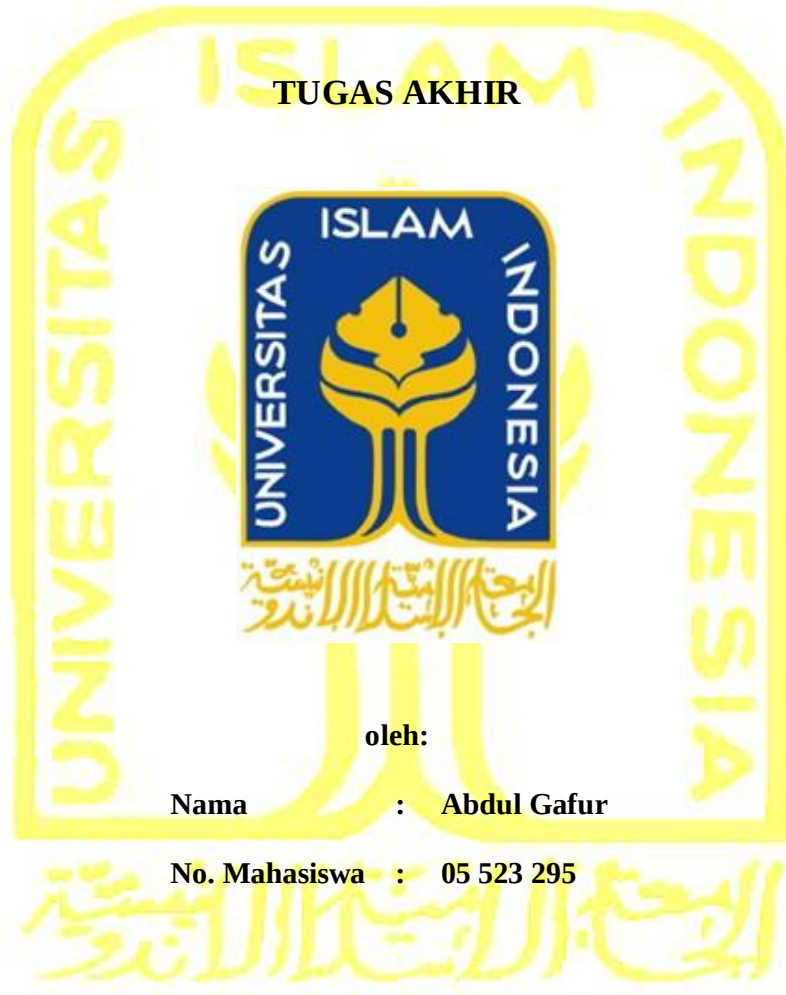
**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2013

LEMBAR PENGESAHAN DOSEN PEMBIMBING

**ANALISIS PERBANDINGAN KINERJA SQUIDGUARD DAN
DANGUARDIAN SEBAGAI IMPLEMENTASI
WEB FILTERING**

TUGAS AKHIR



oleh:

Nama : Abdul Gafur

No. Mahasiswa : 05 523 295

Yogyakarta, Juli 2013

Pembimbing,

Syarif Hidayat, S.Kom., MIT.

LEMBAR PENGESAHAN DOSEN PENGUJI

ANALISIS PERBANDINGAN KINERJA SQUIDGUARD DAN DANSGUARDIAN SEBAGAI IMPLEMENTASI WEB FILTERING

TUGAS AKHIR

oleh:

Nama : Abdul Gafur

No. Mahasiswa : 05 523 295

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana Jurusan Teknik Informatika Fakultas
Teknologi Industri Universitas Islam Indonesia

Yogyakarta, Juli 2013

Tim Penguji,

Syarif Hidayat, S.Kom., M.I.T.
Ketua

Ahmad Luthfi, S.Kom., M.Kom.
Anggota I

Manik Hapsara ST., M.Sc., Ph.D
Anggota II

Mengetahui,

Ketua Jurusan Teknik Informatika
Fakultas Teknologi Industri
Universitas Islam Indonesia



Yudi Prayudi, S.Si., M.Kom.

HALAMAN PERSEMBAHAN

Kupersembahkan tugas akhir ini untuk :

Yang Tercinta:

Kedua orang tuaku,

*atas kasih sayang dan pengorbanan, kesabaran yang tiada tara,
serta doanya yang selalu menyertai langkahku dalam mengarungi hidup ini.*

Kakak-kakak dan adik-adikku,

atas segala support dan doanya

Dan

nDa-ku, Chanifah Indah Ratnasari,

*yang tak henti-hentinya selalu memberikan semangat,
doa, cinta, dukungan, pengorbanan yang tak mengenal lelah,
serta kesabaran yang luar biasa
selama penulis menyelesaikan studi perkuliahan ini*

Sahabat-sahabatku:

*Teman Seperjuangan selama di Informatika
dan Keluarga Besar laboratorium Terpadu*

Untuk Almamaterku:

Jurusan Teknik Informatika, Universitas Islam Indonesia

HALAMAN MOTTO

“Think Positive!!”

“Do The Right Things and Do The Things Right”

KATA PENGANTAR



Alhamdulillah Rabbil'alamin. Puji dan syukur kehadiran Allah SWT yang telah melimpahkan rahmat, nikmat dan hidayah-Nya. Shalawat serta salam senantiasa tercurah kepada junjungan kita Rasulullah Muhammad SAW beserta para keluarga, sahabat serta para pengikutnya, sehingga terselesaikannya tugas akhir dengan judul “ANALISIS PERBANDINGAN KINERJA SQUIDGUARD DAN DANSGUARDIAN SEBAGAI IMPLEMENTASI WEB FILTERING”

Laporan tugas akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar sarjana pada Jurusan Teknik Informatika Fakultas Teknologi Industri, Universitas Islam Indonesia.

Penulisan dan penyelesaian tugas akhir ini tidak lepas dari saran, bimbingan, dukungan serta bantuan dari berbagai pihak. Untuk itu pada kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Ayahanda dan Ibunda tercinta yang tiada putus-putusnya merangkai untaian do'a, membangkitkan semangat, dan selalu menyejukkan ananda dengan kasih sayang yang luar biasa.
2. Kakak-kakakku dan adik-adikku, terima kasih atas semangat dan dukungan yang selama ini senantiasa kalian curahkan untukku.
3. nDa yang selalu setia mendampingi, mendengarkan keluh kesahku, memberi masukan dan nasehat, membuat api semangat yang kadang redup menjadi berbinar, dan selalu mengiringi dalam setiap langkah. Terima kasih atas semuanya.
4. Bapak Syarif Hidayat, S.Kom., MIT., selaku pembimbing yang telah meluangkan waktu, tenaga, dan pikiran untuk membimbing saya dalam menyelesaikan Tugas Akhir ini.
5. Bapak Yudi Prayudi, S.Si., M.Kom., selaku Ketua Jurusan, Jurusan Teknik Informatika, Fakultas Teknologi Industri.

6. Bapak, Ibu dosen Teknik Informatika dan dosen-dosen serta staff pengajar Universitas Islam Indonesia. Terima kasih atas semua ilmu yang diberikan.
7. Teman-teman Asisten Laboratorium Sistem dan Jaringan Komputer, Ucup, Tiyyok, Yopi, Imel, Lantip, Bram, Doni, Nita, Rohmad serta Bapak Ahmad Rafi'i.
8. Teman-teman asisten di Laboratorium Terpadu Informatika, Sirkel, KSC, PIT, GMM, dan Indok, serta laboran yang menjadi rekan kerja selama penulis menjadi asisten di laboratorium SisJarkom.
9. Teman-teman ALIEN Informatika '05, terima kasih atas kekompakan, kekeluargaan, serta persahabatan yang kalian berikan selama ini, serta seluruh teman-teman Informatika lainnya. Informatika...satu untuk semua, semua untuk satu!!!
10. Semua pihak yang tidak dapat penulis sebutkan satu persatu yang telah membantu terselesaikannya Tugas Akhir ini

Penulis menyadari sepenuhnya bahwa Tugas Akhir ini masih jauh dari sempurna, karena itu saran dan kritik sangat penulis harapkan untuk perbaikan di masa mendatang. Semoga Tugas Akhir ini dapat berguna dan mendatangkan manfaat bagi semua pihak.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Yogyakarta, Juli 2013

Abdul Gafur

SARI

Web filtering adalah penyaringan informasi dari internet terhadap materi-materi yang tidak pantas. Salah satu metode yang umum digunakan untuk *web filtering* adalah penyaringan berdasarkan alamat domain (*domain filtering*). Squidguard dan Dansguardian adalah dua contoh aplikasi *web filtering* dengan fitur *domain filtering* yang populer digunakan. Squidguard dan Dansguardian tentunya memiliki kinerja yang berbeda dalam melakukan proses *filtering* tersebut. Secara umum kinerja suatu software dapat dilihat berdasarkan *performance*, yang dapat dinilai berdasarkan bagaimana efisiensi waktu dari software tersebut dalam menyelesaikan *task* yang diberikan, dan *cost*, yang dinilai berdasarkan bagaimana penggunaan *resource hardware* terutama penggunaan CPU dan memori. Dalam menentukan software apa yang akan digunakan untuk pengimplementasian *web filtering*, manajemen membutuhkan informasi tentang bagaimana perbandingan kinerja dari software-software tersebut.

Dalam penelitian ini dilakukan pengujian kinerja Squidguard dan Dansguardian berdasarkan parameter waktu eksekusi data, penggunaan memori, serta penggunaan CPU. Pengujian dilakukan dengan cara melakukan simulasi proses penyaringan oleh Squidguard dan Dansguardian dalam berbagai macam kondisi kepadatan lalu lintas data. Data dari hasil pengujian dicatat dan ditampilkan dalam bentuk tabel dan grafik, kemudian dilakukan uraian dan penafsiran, serta ditarik kesimpulan bagaimana perbandingan kinerja Squidguard dan Dansguardian berdasarkan perbandingan nilai rata-rata tiap parameter pada tabel dan grafik tersebut. Berdasarkan hasil penelitian dapat disimpulkan bahwa kinerja Squidguard lebih baik dibandingkan Dansguardian karena *cost* yang digunakan oleh Squidguard dalam membangun *web filtering* lebih rendah daripada Dansguardian dengan *performance* yang tidak jauh berbeda.

Kata Kunci : *Web filtering, domain filtering, Squidguard, Dansguardian.*

TAKARIR

Access Control Lists	Daftar Kontrol Akses
Access Controller	Pengendali Akses
Active Web Content Filter	Penyaring konten web yang bekerja secara aktif
Blacklist	Daftar hitam
Bridge	Jembatan
Command Line	Baris perintah
Connection Sharing	Membagi koneksi
Content Filtering	Penyaringan konten
Cost	Biaya
Database	Basisdata
Domain Filtering	Penyaringan domain
Filter	Saring
Filtering	Penyaringan
Filtering Rules	Aturan penyaringan
Free	Bebas
Free Software Licence	Lisensi Perangkat Lunak Bebas
Gateway	Pintu gerbang
Hardware	Perangkat keras
Host	Tuan Rumah
Load Average	Beban rata-rata
Local	Lokal

Malware	Perangkat Lunak Jahat
Memory Usage	Penggunaan memori
Network Administrator	Administrator jaringan
Page	Halaman
Performance	Performa
Plugin	Alat tambahan
Portable	Mudah dipindahkan
Processing Time	Waktu pemrosesan
Redirect	Mengalihkan
Reliable	Handal
Reply	Balasan
Repository	Tempat penyimpanan
Request	Permintaan
Resource	Sumber daya
Rewrite	Menulis kembali
Router	Alat yang mengatur jalur koneksi data
Searching	Pencarian
Selective Logging	Pencatatan selektif
Server	Pelayan
Software	Perangkat Lunak
Source Code	Kode Sumber
Unregistered	Tidak terdaftar
User	Pengguna

Virtual Machine

Mesin Virtual

Web

Jaringan

Web Base

Berbasis web

Website

Situs web

DAFTAR ISI

LEMBAR PENGESAHAN DOSEN PEMBIMBING	ii
LEMBAR PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERSEMBAHAN	iv
HALAMAN MOTTO	v
KATA PENGANTAR.....	vi
SARI.....	viii
TAKARIR.....	ix
DAFTAR ISI.....	xii
DAFTAR TABEL	xiv
DAFTAR GAMBAR.....	xv
BAB IPENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian.....	3
1.6 Metodologi Penelitian	3
1.7 Sistematika Penulisan.....	5
BAB IILANDASAN TEORI.....	6
2.1 Konsep Dasar Proxy.....	6
2.2 Web Filtering	7
2.3 Squid.....	8
2.4 SquidGuard	9
2.5 Dansguardian	11

2.6 Webserver Stress Tool.....	12
BAB IIIMETODOLOGI	15
3.1 Skenario	15
3.2 Topologi.....	16
3.3 Analisis Kebutuhan	16
3.3.1 Perangkat Keras.....	17
3.3.2 Perangkat Lunak.....	18
3.4 Instalasi dan Konfigurasi pada Server.....	19
3.4.1 Instalasi	19
3.4.2 Konfigurasi.....	20
3.4.3 Validasi Pengujian.....	23
3.5 Pengujian	23
3.5.1 Metode	23
3.5.2 Parameter Pengujian	24
3.5.3 Alur Pengujian dan Pengambilan Data.....	26
3.6 Analisis.....	27
BAB IVHASIL DAN PEMBAHASAN.....	28
4.1 Tabel Hasil Pengujian.....	28
4.1.1 Menggunakan parameter load average	28
4.1.2 Menggunakan parameter memory usage	29
4.1.3 Menggunakan parameter rata-rata waktu eksekusi data.....	31
4.2 Analisis Hasil Pengujian.....	32
BAB VKESIMPULAN DAN SARAN	36
5.1 Kesimpulan	36
5.2 Saran.....	37
DAFTAR PUSTAKA.....	38
LAMPIRAN	

DAFTAR TABEL

Tabel 3.1 Tabel skenario pengujian.....	15
Tabel 3.2 Tabel perangkat lunak yang digunakan pada server	18
Tabel 4.1 Nilai rata-rata hasil pengujian berdasarkan parameter load average.....	28
Tabel 4.2 Nilai rata-rata hasil pengujian berdasarkan parameter memory usage..	29
Tabel 4.3 Nilai rata-rata hasil pengujian berdasarkan parameter rata-rata waktu eksekusi data.....	31

DAFTAR GAMBAR

Gambar 2.1 Diagram cara kerja proxy.....	7
Gambar 2.2 Diagram web filtering dengan proxy	8
Gambar 2.3 Diagram web filtering dengan Squidguard/Dansguardian.....	12
Gambar 3.1 Desain skema jaringan pengujian pada network 10.0.0.0/24.....	16
Gambar 3.2 Contoh pengujian <i>memory usage</i> dan <i>load average</i>	25
Gambar 3.3 Contoh pengujian rata-rata waktu eksekusi data	26
Gambar 3.4 Alur metode pengambilan data.....	27
Gambar 4.1 Grafik perbandingan <i>load average</i>	28
Gambar 4.2 Grafik perbandingan <i>memory usage</i>	30
Gambar 4.3 Grafik perbandingan rata-rata waktu eksekusi data	31

BAB I

PENDAHULUAN

1.1 Latar Belakang

Internet telah dikenal sebagai salah satu sarana komunikasi dan penyedia informasi yang mudah untuk diakses oleh masyarakat luas. Pada dasarnya, berbagai jenis informasi dapat secara bebas diperoleh dari Internet oleh siapa saja tanpa ada batasan apapun. Dengan beragamnya jenis informasi tersebut, tentu saja tidak semua konten pantas diakses oleh beberapa kategori pengguna Internet, seperti informasi mengenai pornografi, kekerasan, judi, dan lain sebagainya. Untuk itulah maka perlu dilakukan penyaringan informasi dari Internet apabila suatu pihak atau instansi tertentu menginginkan pelarangan akses terhadap jenis informasi tertentu dari Internet.

Ada berbagai macam metode yang dapat digunakan untuk melakukan penyaringan informasi dari Internet, salah satunya adalah dengan menggunakan aplikasi *web filtering*. Aplikasi-aplikasi ini banyak sekali ragamnya. Ada yang dapat berjalan pada sistem operasi tertentu, ada juga yang dapat berjalan di berbagai sistem operasi. Ada yang bersifat gratis dalam penggunaannya, ada pula yang berbayar. Dari berbagai macam jenis aplikasi-aplikasi *web filtering* tersebut terdapat beberapa aplikasi yang paling populer digunakan, yaitu Squidguard dan Dansguardian. Selain karena kemudahan implementasinya, kedua aplikasi tersebut bersifat gratis dalam penggunaannya.

Squidguard dan Dansguardian merupakan aplikasi yang berjalan pada sisi server dan berjalan secara bersama dengan sebuah *web proxy* seperti squid. Sebagai *gateway*, server akan dilewati paket-paket data yang mengantri saat diakses oleh client. Pada server yang terdapat Squidguard atau Dansguardian, paket-paket data tersebut akan mengalami *filtering* oleh aplikasi tersebut sebelum paket-paket data tersebut diteruskan. Squidguard dan Dansguardian tentunya memiliki kinerja yang berbeda dalam melakukan proses *filtering* tersebut.

Secara umum, kinerja suatu *software* dapat dilihat berdasarkan *performance* dan *cost* dari *software* tersebut. *Performance* dinilai berdasarkan bagaimana efisiensi waktu dari *software* tersebut dalam menyelesaikan *task* yang diberikan (*processing time*). Semakin cepat *processing time* semakin baik. *Cost* (biaya) dinilai berdasarkan bagaimana penggunaan *resource hardware* oleh *software* dalam menyelesaikan *task* yang diberikan. Semakin rendah *resource hardware* yang dibutuhkan semakin baik. Dalam melakukan proses *filtering* oleh Squidguard dan Dansguardian tentu saja terdapat penggunaan *resource hardware* dan *processing time*. Kedua aplikasi tersebut memiliki mekanisme *web filtering* yang berbeda, sehingga diperkirakan penggunaan *resource hardware* dan *processing time* yang terpakai pun juga berbeda.

Berdasarkan hal tersebut, penulis termotivasi untuk melakukan perbandingan penggunaan *resource hardware*, dalam hal ini prosesor dan memori, dan *processing time* oleh aplikasi Squidguard dan Dansguardian. Dengan mencermati hasil dari penelitian ini diharapkan dapat dijadikan sebagai acuan oleh organisasi atau instansi tertentu dalam memilih aplikasi mana di antara kedua aplikasi tersebut yang ingin digunakan sebagai sistem penyaringan informasi Internet.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut diatas maka dapat ditarik rumusan masalah yaitu bagaimana perbandingan penggunaan *resource* prosesor dan memori, serta waktu eksekusi data antara Squidguard dan Dansguardian berdasarkan variasi kepadatan lalu lintas data dalam melakukan penyaringan informasi dari Internet.

1.3 Batasan Masalah

Optimalisasi pembahasan secara terperinci dapat diperoleh ketika tersedianya suatu batasan materi pada penyusunan tugas akhir ini. Dalam hal ini terdapat beberapa batasan masalah yaitu sebagai berikut:

1. Membangun *web filtering server* menggunakan Squidguard dan Dansguardian, masing-masing pada sistem operasi Linux Ubuntu Server.
2. Menggunakan virtualisasi dalam membangun *web filtering server* menggunakan VMware.
3. *Web filtering* yang diuji berupa *filtering* berdasarkan domain (*domain filtering*).
4. Penilaian kinerja Squidguard dan Dansguardian sebagai *web filtering server* meliputi beberapa parameter, yaitu penggunaan *resource server* (prosesor dan memori), serta waktu tanggap pengaksesan halaman.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengetahui kinerja aplikasi Squidguard dan Dansguardian dalam berbagai variasi kepadatan lalu lintas data berdasarkan parameter waktu eksekusi data, *memory usage*, dan *load averaged* dalam membangun *web filtering server*.

1.5 Manfaat Penelitian

Melalui hasil analisis dalam penelitian ini, *network administrator* dapat mengetahui bagaimana perbandingan kinerja Squidguard dan Dansguardian dalam berbagai variasi kepadatan lalu lintas data sehingga memberikan kemudahan dalam memilih aplikasi yang lebih baik dalam membangun *web filtering server*, sehingga *web filtering server* yang akan dibangun nantinya dapat sesuai dengan kebutuhan dan *reliable*.

1.6 Metodologi Penelitian

1.6.1 Metode Pengumpulan Data

Pada penelitian ini, metode pengumpulan data yang terkait dengan penelitian meliputi:

a. Metode Studi Pustaka

Metode studi pustaka, yaitu metode pengumpulan melalui buku-buku yang berkaitan dengan penelitian yang dilakukan.

b. Metode Studi Literatur Internet

Metode studi literatur Internet, merupakan metode pengumpulan melalui Internet yang dilakukan dengan tujuan pemilihan perangkat lunak yang akan diimplementasikan dalam membangun *web filtering server*, serta kebutuhan perangkat keras dan perangkat lunak pendukung lainnya agar sistem berjalan dengan baik.

1.6.2 Perencanaan Skenario Pengujian

Tahap ini merupakan tahap perencanaan bagaimana skenario pengujian yang akan dilakukan dalam penelitian berdasarkan perolehan dari hasil pengumpulan data, meliputi desain arsitektur jaringan, desain alur proses aplikasi, serta rancangan bagaimana proses pengujian akan dilakukan.

1.6.3 Implementasi Skenario Pengujian

Tahap ini merupakan tahap implementasi dari skenario pengujian yang telah dirancang sebelumnya, serta termasuk instalasi dan konfigurasi sistem.

1.6.4 Validasi Pengujian

Tahap ini merupakan tahap validasi terhadap konfigurasi sistem apakah sistem sudah berjalan dan mampu melakukan proses pengujian dengan benar.

1.6.5 Pengujian dan Pengumpulan Hasil

Pada tahap ini dilakukan proses pengujian dan pengumpulan data hasil dari pengujian.

1.6.6 Analisis Hasil

Tahap selanjutnya adalah analisis dari data-data hasil pengujian. Data-data hasil analisis ini akan dijadikan patokan untuk mengambil kesimpulan dalam penelitian ini.

1.7 Sistematika Penulisan

Sistematika penulisan laporan berguna untuk memberikan gambaran umum dari keseluruhan isi laporan. Sistematika penulisan dan garis besar isi laporan ini adalah sebagai berikut.

- BAB I Pendahuluan, berisi latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, serta sistematika penulisan laporan.
- BAB II Landasan Teori, memuat dasar teori yang membahas mengenai teori-teori yang berkaitan dengan *web filtering* menggunakan Squidguard dan Dansguardian.
- BAB III Metodologi, berisikan langkah-langkah membangun *web filtering server* termasuk instalasi dan konfigurasi sistem sesuai dengan aplikasi *web filtering* yang digunakan, validasi pengujian, serta metode pengujian dan analisis.
- BAB IV Hasil dan Pembahasan, berisi hasil pengujian dari kedua aplikasi *web filtering* tersebut serta analisis dari hasil pengujian.
- BAB V Kesimpulan dan Saran, memuat kesimpulan dari pengujian yang dilakukan terhadap kedua aplikasi *web filtering*. Selain itu juga berisi saran-saran dalam memilih aplikasi *web filtering* yang tepat untuk perusahaan-perusahaan atau organisasi yang ingin memiliki *web filtering server*.

BAB II

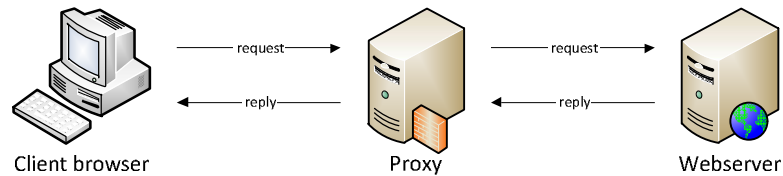
LANDASAN TEORI

2.1 Konsep Dasar Proxy

Proxy dapat dipahami sebagai pihak ketiga(sistem komputer atau aplikasi) yang berdiri ditengah-tengah antara kedua pihak yang saling berhubungan (*client* dan *server*) dan berfungsi sebagai perantara, sedemikian sehingga pihak pertama dan pihak kedua tidak secara langsung berhubungan, akan tetapi masing-masing berhubungan dengan perantara, yaitu Proxy.

Proxy server bekerja dengan mendengarkan request dari client internal dan mengirim request tersebut ke jaringan eksternal seolah-olah proxy server itu sendiri yang menjadi client. Pada waktu proxy server menerima respon dan server publik, ia memberikan respon tersebut ke client yang asli seolah-olah ia public server.

Proxy server mempunyai 3 fungsi utama, yaitu *connection sharing*, *filtering*, dan *caching*. Dalam suatu jaringan lokal yang terhubung ke jaringan lain atau internet, pengguna tidak langsung berhubungan dengan jaringan luar atau internet, tetapi harus melewati suatu *gateway*, yang bertindak sebagai batas antara jaringan lokal dan jaringan luar. *Gateway* ini sangat penting, karena jaringan lokal harus dapat dilindungi dengan baik dari bahaya yang mungkin berasal dari internet, dan hal tersebut akan sulit dilakukan bila tidak ada garis batas yang jelas jaringan lokal dan internet. *Gateway* juga bertindak sebagai titik dimana sejumlah koneksi dari pengguna lokal akan terhubung kepadanya, dan suatu koneksi ke jaringan luar juga terhubung kepadanya. Dengan demikian, koneksi dari jaringan lokal ke internet akan menggunakan sambungan yang dimiliki oleh *gateway* secara bersama-sama (*connection sharing*). Dalam hal ini, *gateway* adalah juga sebagai *proxy server*, karena menyediakan layanan sebagai perantara antara jaringan lokal dan jaringan luar atau internet. Cara kerja proxy dapat dilihat pada gambar 2.1.



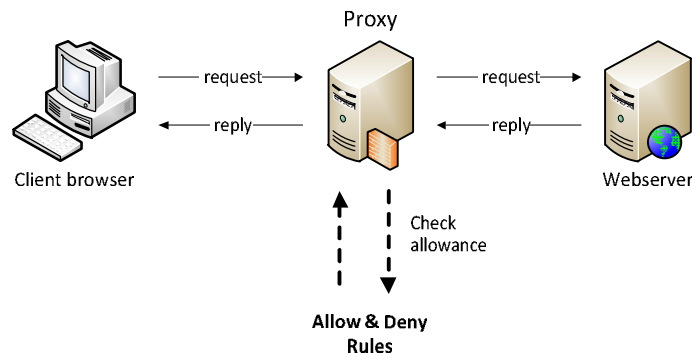
Gambar 2.1Diagram cara kerja proxy

Proxy bekerja dalam berbagai jenis protokol komunikasi jaringan dan dapat berada pada level-level yang berbeda pada hirarki layer protokol komunikasi jaringan. Suatu perantara dapat saja bekerja pada layer Data-Link, layer Network dan layer Transport, maupun layer Aplikasi dalam hirarki layer komunikasi jaringan menurut OSI. Namun pengertian proxy server sebagian besar adalah untuk menunjuk suatu server yang bekerja sebagai proxy pada layer Aplikasi.

2.2 Web Filtering

Web filtering biasanya digunakan oleh organisasi seperti perusahaan atau sekolah untuk mencegah pengguna komputer dari pengaksesan terhadap situs *web* yang tidak pantas ataupun demi keamanan jaringan, seperti pencegahan pengaksesan terhadap *malware host*.

Filtering rules (aturan penyaringan) biasanya diatur oleh departemen IT dan dapat diimplementasikan melalui perangkat lunak pada komputer individu atau pada titik pusat pada jaringan seperti pada *proxy server* atau *internet router*. Tergantung dari kecanggihan sistem yang digunakan, dimungkinkan bagi para pengguna komputer untuk memiliki tingkat akses internet yang berbeda. Pada Gambar 2.2 dapat dilihat pengimplementasian *web filtering* pada *proxy server*.



Gambar 2.2Diagram web filtering dengan proxy

Perangkat lunak untuk *web filtering* juga terkadang digunakan pada komputer rumah untuk membatasi akses ke situs yang tidak pantas untuk anak-anak yang menggunakan komputer tersebut. Perangkat lunak tersebut biasanya dikenal dengan *Parental Control Software*.

2.3 Squid

Squid adalah sebuah *daemon* yang digunakan sebagai *proxy server* dan *web cache*. Squid memiliki banyak jenis penggunaan, mulai dari mempercepat server *web* dengan melakukan *caching* permintaan yang berulang-ulang, *caching* DNS, *caching* situs *web*, dan *caching* pencarian komputer di dalam jaringan untuk sekelompok komputer yang menggunakan sumber daya jaringan yang sama, hingga pada membantu keamanan dengan cara melakukan penyaringan (*filter*) lalu lintas. Meskipun seringnya digunakan untuk protokol HTTP dan FTP, Squid juga menawarkan dukungan terbatas untuk beberapa protokol lainnya termasuk *Transport Layer Security* (TLS), *Secure Socket Layer* (SSL), *Internet Gopher*, dan HTTPS. Versi Squid 3.1 mencakup dukungan protokol IPv6 dan *Internet Content Adaptation Protocol* (ICAP).

Squid pada awalnya dikembangkan oleh Duane Wessels sebagai "*Harvest object cache*", yang merupakan bagian dari proyek Harvest yang dikembangkan di University of Colorado at Boulder. Pekerjaan selanjutnya dilakukan hingga selesai di University of California, San Diego dan didanai melalui National Science

Foundation. Squid kini hampir secara eksklusif dikembangkan dengan cara usaha sukarela.

Squid umumnya didesain untuk berjalan di atas sistem operasi keluarga UNIX, meski Squid juga bisa berjalan di atas sistem operasi Windows. Karena dirilis di bawah lisensi GNU General Public License, maka Squid merupakan perangkat lunak bebas.

Squid dapat berjalan di atas sistem-sistem operasi berikut:

- a. AIX
- b. BSDI
- c. Digital Unix
- d. FreeBSD
- e. HP-UX
- f. IRIX
- g. Linux
- h. Mac OS X
- i. NetBSD
- j. NeXTStep
- k. OpenBSD
- l. SCO OpenServer
- m. Solaris
- n. UnixWare
- o. Windows

2.4 SquidGuard

SquidGuard adalah *plugin* untuk squid yang memiliki fitur kombinasi dari *filter*, *redirector*, dan *access controller*. Aplikasi ini bersifat :

1. *Free*(GPLv2)
2. Sangat fleksibel (dalam hal opsi konfigurasi)
3. Mudah diinstal
4. *Portable*, dapat diinstal di berbagai platform UNIX

SquidGuard dapat digunakan untuk :

1. Membatasi akses *web* untuk beberapa *user* ke suatu daftar *server web* ataupun URL.
2. Memblokir akses ke beberapa *server web* ataupun URL yang tercantum dalam daftar hitam (*blacklist*) untuk beberapa *user*.
3. Memblokir akses ke URL menggunakan pencocokan berdasarkan daftar ekspresi reguler atau kata-kata untuk beberapa *user*.
4. Memaksa penggunaan nama domain / melarang penggunaan alamat IP dalam URL.
5. *Redirect* URL yang diblokir ke halaman info berbasis CGI.
6. *Redirect* *unregistered user* (user yang belum terdaftar) ke halaman registrasi.
7. *Redirect* pengunduhan objek ke salinan lokal.
8. Memiliki aturan akses yang berbeda berdasarkan waktu, hari, minggu, tanggal, dan lain-lain.
9. Memiliki aturan yang berbeda untuk kelompok pengguna yang berbeda.
10. Dan lain sebagainya.

SquidGuard memiliki banyak opsi konfigurasi yang memungkinkan:

1. **menentukan rentang waktu yang berbeda** dengan berbagai kombinasi:
 - a. waktu dalam hari (00:00-08:00, 17:00-24:00)
 - b. hari (Senin, Rabu, Minggu)
 - c. tanggal (1999/05/13)
 - d. rentang tanggal (1999-04-01-1999-04-05)
 - e. tanggal wildcard (*- *- 01-01 05-17 12-25 *-)
2. **kelompok sumber** (pengguna / klien) dalam kategori yang berbeda seperti "manajer", "karyawan", "guru", "siswa", "pelanggan", "tamu", dan lain-lain, misalnya:
 - a. berdasarkan rentang alamat IP (172.16.0.0/12, 172.16.0.0/255.240.0.0, 172.16.0.11-172.16.0.35)
 - b. daftar alamat ip (172.16.134.54 172.16.156.23 ...)
 - c. daftar domain (foo.bar.com ...)

- d. daftar id user(*weho sdgh dfhj ASEF ...*)
- 3. **kelompok tujuan** (URL / server) dalam kategori yang berbeda seperti "lokal", "pelanggan", "vendor", "banner", "dilarang" dan lain-lain, misalnya:
 - a. domain, termasuk subdomain (*foo.bar.com*)
 - b. host(*host.foo.bar.com*)
 - c. URL direktori, termasuk subdirektori (*foo.bar.com/suatu/direktori*)
 - d. File URL (*foo.bar.com/tempat/file.html*)
 - e. ekspresi reguler (*((expr1|expr2|...))*)
- 4. **rewrite / redirect URL**
- 5. **mendefinisikan access control lists** (ACL)
- 6. **memiliki selective logging**

2.5 Dansguardian

Dansguardian adalah sebuah aplikasi *active web content filter*. Aplikasi ini dirancang untuk mengontrol *website* mana yang boleh diakses dan mana yang tidak. Dansguardian hanya dapat berjalan pada server berbasis Unix atau GNU/Linux.

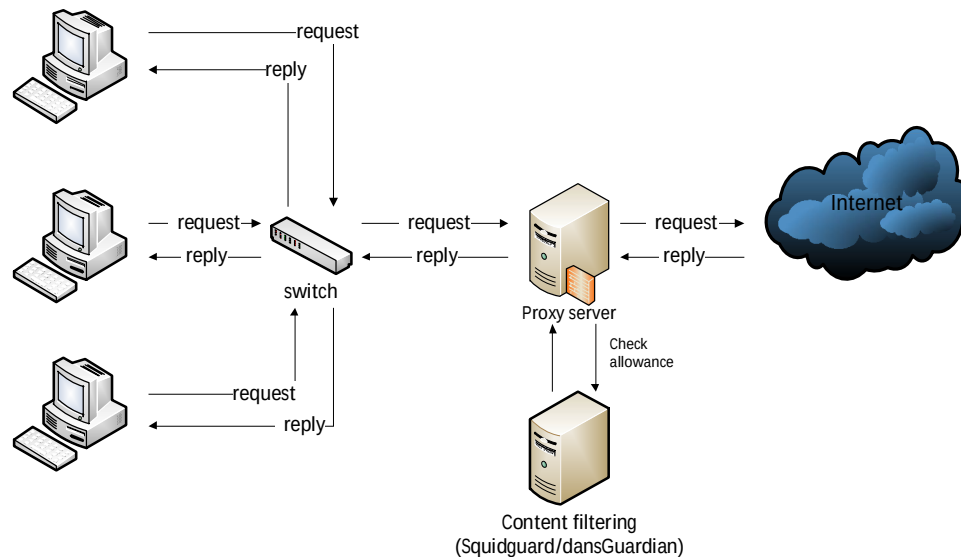
Dansguardian didistribusikan dibawah lisensi GPLv2 *free software licence*, dan ditulis menggunakan bahasa pemrograman C++. Dansguardian terutama dapat berjalan pada sistem GNU/Linux dan berbasis Unix lainnya. Aplikasi ini dapat digunakan secara *command line* maupun *web base*, dan dimaksudkan untuk berjalan secara konjungsi bersama sebuah *web proxy* seperti Squid.

Dalam proses penyaringannya, dansguardian menggunakan beberapa metode:

1. URL and *domain filtering*, pencocokan terhadap URL dan domain yang diakses dengan daftar URL dan domain yang boleh diakses atau tidak sesuai dengan konfigurasi yang dibuat

2. *content phrase filtering*, pengecekan konten dari sebuah halaman terhadap frase-frase yang sering berasosiasi pada pornografi, kekerasan, judi, dan konten yang tidak diinginkan lainnya
3. PICS (Platform for Internet Content Selection) filtering
4. MIME filtering
5. file extension filtering
6. *POST limiting*, memungkinkan untuk memblokir atau membatasi *web upload*

Diagram web filtering dengan menggunakan Squidguard ataupun Dansguardian dapat dilihat pada Gambar 2.3.



Gambar 2.3 Diagram web filtering dengan Squidguard/Dansguardian

2.6 Webservice Stress Tool

Webservice Stress Tool adalah aplikasi yang digunakan untuk pengujian tingkat stress pada suatu aplikasi web yang melewati media http/https pada waktu yang bersamaan. Pengujian ini berlangsung sendiri-sendiri pada setiap user yang diatur untuk mengakses web tersebut mulai dari loading gambar bingkai dan sebagainya.

Pada pengujiannya kita dapat mengatur jenis profil dari user pengakses web ini, semisal ada yang melihat tab download ada yang login ada yang melakukan poling. Setiap user akan direkam dan dilakukan penganalisaan apakah web itu dapat bertahan pada pengujian serempak dan melakukan pelaporan atas proses tersebut.

Jenis tes / fungsi yang dapat dilakukan oleh Webserver Stress Tool antara lain:

1. Performance Tests, digunakan untuk menguji setiap bagian dari atau aplikasi web apakah lambat dan bagian mana yang harus dipercepat, biasanya pada beberapa pengujian web yang menggunakan satu skrip per *page* dapat lebih cepat dalam pengaksesannya.
2. Load Tests, pengujian untuk memperhitungkan lalu lintas data yang dapat didukung web yang diuji, pengujian ini akan menyamakan dengan lingkungan asli untuk web yang belum di hosting di internet semisal pada server lokal.
3. Stress Tests, mensimulasikan serangan “brute force” yang akan terjadi pada *web server*, biasanya pada kenyataannya digunakan untuk mensiasati situasi yang mungkin terjadi pada *web* ketika dihostingkan.
4. Ramp Tests, digunakan untuk menghitung berapa banyak *user* yang dapat mengaksesnya sebelum terjadinya pesan error.
5. Berbagai pengujian lainnya yang memungkinkan, seperti *database deadlock*, *semaphores*, dan lainnya.

Aplikasi ini akan menghitung:

1. Waktu mengklik. Aplikasi ini akan mensimulasikan pengklikan apa saja yang terdapat dalam halaman *web* tersebut yang selanjutnya setiap aksi pengklikan akan dihitung jeda waktunya dan pengiriman yang dapat dilakukan oleh *server* dalam waktu bersamaan.
2. Rata-rata waktu pengklikan, perhitungan nilai rata-rata per URL, per *user* atau per *website*.

3. Waktu untuk DNS, waktu yang digunakan oleh domain url pada saat digunakan pengguna ketika mengakses langsung dns *server*..
4. Waktu untuk terhubung, waktu yang digunakan untuk mengkoneksikan dengan *server*.
5. Waktu untuk byte pertama(TFB), waktu yang diperlukan diantara inisiasi permintaan dan penerimaan dihitung dimulai dari pengiriman byte pertama dari *server*.
6. Waktu permintaan(*Request Time*), waktu yang diperlukan untuk permintaan satu halaman http (seperti: HTML *page*, gambar, bingkai, dsb)
7. *User/Server Bandwidth*, perhitungan atas *bandwidth* yang digunakan *server* dan *user* untuk dapat saling berhubungan.
8. Pengiriman permintaan, angka pengiriman perhitungan selama waktu di*server*.
9. Penerimaan permintaan, angka yang diterima dari pengiriman data dari *server*.
10. Membuka permintaan, angka yang yang digunakan untuk menghitung permintaan yang berhasil dibuka.
11. Tingkat error, angka kegagalan dari permintaan per periode waktu , per user maupun per url.

Webserver Stress Tool akan menganalisis setiap elemen kedalam spesifik data test yang selanjutnya dapat disimpan kedalam log CSV-format agar mudah dilihatnya.

BAB III

METODOLOGI

3.1 Skenario

Dalam pengujian akan digunakan satu buah alamat domain sebagai target. Domain target akan dimasukkan ke dalam file *blacklist* (dimana file *blacklist* yang digunakan pada penelitian ini terdiri dari 800.220 domain) sehingga ketika domain tersebut diakses pasti akan diblok oleh Squidguard/Dansguardian. Pengujian terhadap domain tersebut akan dilakukan dalam 3 kondisi berbeda, yaitu :

1. Kondisi *best case*, dimana domain tersebut ditempatkan di awal file *blacklist*(baris 1).
2. Kondisi *average case*, dimana domain tersebut ditempatkan di tengah file *blacklist*(baris 400.110).
3. Kondisi *worst case*, dimana domain tersebut ditempatkan di akhir file *blacklist*(baris 800.220).

Ketiga skenario tersebut dapat dilihat pada tabel 3.1

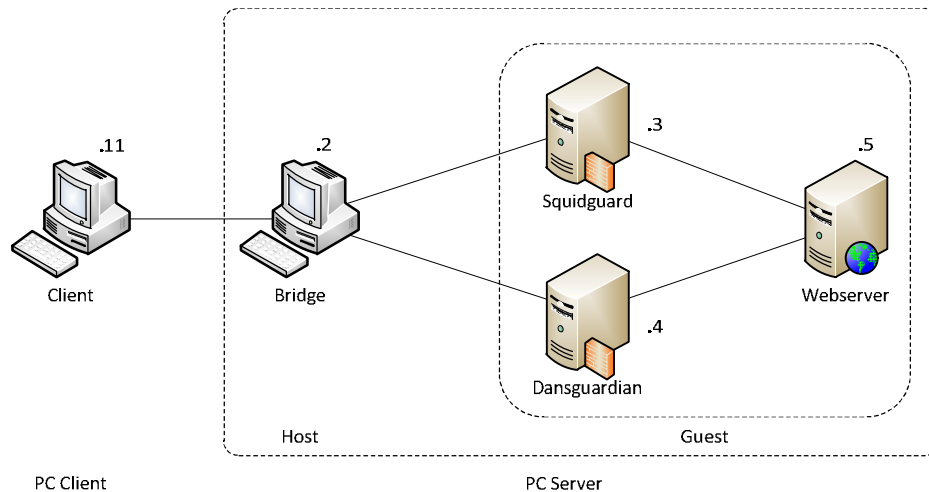
Tabel 3.1Tabel skenario pengujian

Skenario Pengujian	Letak domain target pada file <i>blacklist</i>
Best Case	Awal
Average Case	Tengah
Worst Case	Akhir

Proses pencarian dan pencocokan domain target dengan database domain merupakan proses utama yang dilakukan oleh Squidguard dan Dansguardian dalam melakukan web filtering. Dengan dilakukan pengujian pada skenario yang berbeda ini diharapkan dapat meminimalisir ketidakakuratan data akibat perbedaan algoritma pencarian yang digunakan antara Squidguard dan Dansguardian, karena nilai yang diambil nantinya adalah nilai rata-rata dari ketiga skenario ini.

3.2 Topologi

Skema jaringan yang akan digunakan dalam pengujian dapat dilihat pada Gambar 3.1 berikut:



Gambar 3.1 Desain skema jaringan pengujian pada network 10.0.0.0/24

Alamat jaringan yang digunakan adalah 10.0.0.0/24 dengan alokasi IP sebagai berikut:

- a. PC Client : 10.0.0.11/24
- b. Host PC Server (bridge) : 10.0.0.2/24
- c. Server Squidguard : 10.0.0.3/24
- d. Server Dansguardian : 10.0.0.4/24
- e. Webserver : 10.0.0.5/24

3.3 Analisis Kebutuhan

Untuk mengakomodasi skenario dan topologi yang telah dirancang untuk pengujian, akan digunakan perangkat keras dan perangkat lunak yang dapat memenuhi kebutuhan sistem.

3.3.1 Perangkat Keras

Perangkat keras penunjang yang digunakan pada proses membandingkan Squidguard dengan Dansguardian dalam implementasi web filtering adalah sebagai berikut:

1. Komputer server dan client

Dalam pengujian ini akan digunakan dua unit komputer. Satu unit akan digunakan sebagai client (PC Client) dan satu unit lagi akan dibuat sebagai komputer server (PC Server). Spesifikasi untuk PC Client sebagai berikut:

- a. Prosessor dengan kecepatan 2,1 GHz
- b. Memori 2 GB
- c. Media penyimpanan sebesar 250 GB

Pada PC Server dibuat virtualisasi tiga buah *guest operating system*, yaitu server Squidguard, server Dansguardian, dan *Webserver*, sedangkan *host operating system* akan digunakan sebagai *bridge* antara *guest operating system* dengan komputer *client*. Spesifikasi PC Server ini sebagai berikut:

- a. Prosessor dengan kecepatan 2,53 GHz
- b. Memori 4 GB
- c. Media penyimpanan sebesar 640 GB

2. Virtual machine

Setiap *virtual machine*, baik itu untuk server Squidguard, server Dansguardian, maupun *Webserver*, akan mempunyai spesifikasi sebagai berikut:

- a. Prosessor dengan kecepatan 1 GHz, 1 core
- b. Memori 512 MB
- c. Media penyimpanan sebesar 10 GB
- d. Jaringan terkoneksi secara *bridge*

3.3.2 Perangkat Lunak

Perangkat lunak yang digunakan dalam tugas akhir ini adalah sebagai berikut:

1. Ubuntu server 10.04 sebagai sistem operasi server.
2. VMware Workstation sebagai software virtualisasi.
3. Squid sebagai *proxy server*.
4. Squidguard dan Dansguardian sebagai software *web filtering*.
5. Apache sebagai *webserver*.
6. Webserver stress tool digunakan untuk pengujian kinerja server *web filtering* dengan menyimulasikan *request* halaman web oleh banyak *user* pada saat bersamaan.

Sistem operasi dan aplikasi yang digunakan pada masing-masing server serta konfigurasi yang dilakukan dapat dilihat pada tabel 3.2

Tabel 3.2 Tabel perangkat lunak yang digunakan pada server

Mesin	OS	Aplikasi	Konfigurasi
Squidguard	Ubuntu	Squid Squidguard	Squid dikonfigurasi agar menggunakan Squidguard untuk proses filtering. Buat database blacklist. Konfigurasi Squidguard agar menggunakan database blacklist yang telah dibuat.
Dansguardian	Ubuntu	Squid Dansguardian	Squid dikonfigurasi agar menggunakan Dansguardian untuk proses filtering. Buat database blacklist. Konfigurasi Dansguardian agar menggunakan database blacklist yang telah dibuat.
Webserver	Ubuntu	Apache	Konfigurasi apache sebagai domain target

3.4 Instalasi dan Konfigurasi pada Server

3.4.1 Instalasi

Untuk cara penginstalasian software pada sistem ada beberapa cara yang bisa digunakan, yaitu bisa dengan cara kompilasi *source code*, instalasi menggunakan paket **.deb*, atau instalasi secara online ke *repository*. Untuk pengujian ini semua instalasi software yang dibutuhkan oleh sistem menggunakan cara instalasi melalui *repository* online.

a. Instalasi Proxy Server di kedua server

Pada pengujian ini menggunakan *software* Squid sebagai proxy server. Instalasi program squid proxy server secara online menggunakan *apt-get install squid* melalui terminal.

```
root@proxy:~# apt-get install squid
```

b. Instalasi Squidguard

Instal Squidguard pada server Squidguard menggunakan perintah *apt-get install squidguard* pada terminal.

```
root@proxy:~# apt-get install squidguard
```

c. Instalasi Dansguardian

Instal Dansguardian pada server Dansguardian menggunakan perintah *apt-get install dansguardian* pada terminal.

```
root@proxy:~# apt-get install dansguardian
```

d. Instalasi Apache webserver

Pada webserver akan digunakan Apache2 sebagai software webserver. Instal Apache pada webserver menggunakan perintah *apt-get install apache2* pada terminal.

```
root@webserver:~# apt-get install apache2
```

3.4.2 Konfigurasi

Untuk pengujian dalam penelitian ini, penulis hanya akan menggunakan satu kategori saja untuk database domain yang di-*blacklist*, yaitu kategori porn, baik pada Squidguard maupun pada Dansguardian. Database tersebut akan menggunakan database *blacklist* milik shallalist yang dapat diunduh secara gratis melalui www.shallalist.de.

a. Konfigurasi Pada Server Squidguard

1. Buat database daftar domain yang akan di blok (*blacklist*). Perintah yang digunakan:

```
root@proxy:~# cd /tmp
root@proxy:~# wget www.shallalist.de/Downloads/shallalist.tar.gz
root@proxy:~# tar -xvzf shallalist.tar.gz
root@proxy:~# cp -rf/tmp/BL/porn /var/lib/squidguard/db/
```

Pada perintah tersebut, database dari shallalist didownload ke direktori **/tmp** menggunakan perintah **wget**. Database yang berbentuk arsip *.tar.gz tersebut diekstrak, lalu direktori porn di-copy-kan ke direktori database squidguard yaitu di **/var/lib/squidguard/db/**

2. Dilakukan konfigurasi pada file konfigurasi squidguard. Penulis menggunakan teks editor **nano**.

```
root@proxy:~# nano /etc/squid/squidGuard.conf
```

file konfigurasi tersebut diubah menjadi seperti berikut:

```
# CONFIG FILE FOR SQUIDGUARD
dbhome /var/lib/squidguard/db
logdir /var/log/squid

dest porn {
    domainlist porn/domains
}

acl {
    default {
        pass !porn all
        redirect http://localhost/block.html
    }
}
```

dbhome : konfigurasi letak file database

- logdir : konfigurasi letak file log
- destporn : pendefinisian kategori *porn*. Pada bagian ini bisa digunakan domainlist jika isi database berupa daftar domain, atau urllist jika isi file database berupa daftar url.
- acl : Access Control List. Pada bagian ini ditentukan apa yang akan dilakukan terhadap kategori yang telah didefinisikan sebelumnya. Pada konfigurasi di atas, semua domain diperbolehkan untuk diakses, kecuali domain-domain yang terdapat pada kategori porn. Jika domain yang diakses client termasuk dalam kategori yang diblok, maka akan di-*redirect* ke halaman block.html.

3. Pastikan user squid, dalam hal ini proxy, sebagai pemilik file config, db, dan log dari squidguard. Hal ini perlu dilakukan agar user proxy dapat mengakses dan memodifikasi file-file tersebut.

```
root@proxy:~# chown proxy:proxy /etc/squid/squidGuard.conf
root@proxy:~# chown -R proxy:proxy /var/lib/squidguard/db
root@proxy:~# chown -R proxy:proxy /var/log/squid/
```

Opsi `-R` pada perintah di atas digunakan agar perintah `chown` berlaku secara rekursif kepada seluruh file dalam direktori `/var/lib/squidguard/db` dan `/var/log/squid/`

4. Inisialisasi blacklist dengan mengkonversikan file text blacklist menjadi file database.

```
root@proxy:~# sudo squidGuard -C all
```

5. Dilakukan konfigurasi pada squid

```
root@proxy:~# nano /etc/squid/squid.conf
```

- a. Pada baris **http_access allow localnet**, dihilangkan tanda `#` di depannya untuk mengaktifkan rule ini.
- b. Baris berikut ditambahkan pada bagian `url_rewrite`

```
url_rewrite_program /usr/bin/squidGuard -c /etc/squid/squidGuard.conf
```

6. Perintah agar squid membaca konfigurasi yang baru

```
root@proxy:~# squid -k reconfigure
```

b. Konfigurasi Dansguardian

1. Dilakukan perubahan pada file konfigurasi dansguardian menggunakan text editor

```
root@proxy:~# nano /etc/dansguardian/dansguardian.conf
```

- a. Ditambahkan tanda # diawal baris **“UNCONFIGURED - Please remove this line after configuration”** agar baris tersebut menjadi komentar dan tidak dieksekusi (atau bisa juga dihapus).

```
# UNCONFIGURED - Please remove this line after configuration
```

- b. Baris **“weightedphrasemode”** diganti valuenya menjadi 0.

```
weightedphrasemode = 0
```

Hal ini dilakukan untuk mematikan fitur weightedphrasemode pada Dansguardian.

2. Sama seperti pada konfigurasi squidguard, database blacklist yang akan digunakan dalam pengujian ini diambil dari www.shallalist.de dan hanya satu kategori saja yang akan digunakan yaitu kategori *porn*. File tersebut di-*download* lalu diekstrak, kemudian direktori *porn* di-*copy* ke direktori database Dansguardian

```
root@proxy:~# cd /tmp
root@proxy:~# wget www.shallalist.de/Downloads/shallalist.tar.gz
root@proxy:~# tar -xvzf shallalist.tar.gz
root@proxy:~# cp -rf /tmp/BL/porn /etc/dansguardian/lists/blacklist
```

3. Dilakukan konfigurasi pada file `/etc/dansguardian/lists/bannedsitelist`

```
root@proxy:~# nano /etc/dansguardian/lists/bannedsitelist
```

Kategori porn diaktifkan dengan menghapus tanda `$(uncomment)` pada baris `.Include</etc/dansguardian/blacklists/porn/domains>`

```
#.Include</etc/dansguardian/lists/blacklists/onlinepayment/domains>
#.Include</etc/dansguardian/lists/blacklists/personalfinance/domains>
#.Include</etc/dansguardian/lists/blacklists/pets/domains>
.Include</etc/dansguardian/lists/blacklists/porn/domains>
#.Include</etc/dansguardian/lists/blacklists/proxy/domains>
#.Include</etc/dansguardian/lists/blacklists/publicite/domains>
#.Include</etc/dansguardian/lists/blacklists/redirector/domains>
```

4. Daemon Dansguardian direstart agar menggunakan konfigurasi terbaru.

```
root@proxy:~# /etc/init.d/dansguardian reload
```

3.4.3 Validasi Pengujian

Setelah instalasi dan konfigurasi sistem selesai, dilakukan validasi apakah sistem sudah berjalan dan mampu melakukan proses pengujian dengan benar. Validasi dilakukan baik terhadap Squidguard maupun Dansguardian.

Proses validasi ini dilakukan dengan cara :

1. Pada browser di komputer client, settingan proxy diarahkan ke alamat IP dan port Squidguard/Dansguardian.
2. Menggunakan browser tersebut, dicoba akses ke alamat domain atau URL yang tidak dilarang dalam *filtering rules* Squidguard/Dansguardian. Pada proses ini seharusnya browser dapat mengakses web yang dituju.
3. Coba lagi akses ke alamat domain atau URL yang dilarang dalam *filtering rules* Squidguard/Dansguardian. Pada proses ini seharusnya browser tidak dapat mengakses web yang dituju.

Pada langkah 2 dan 3 hasil yang didapatkan sesuai dengan yang diharapkan. Konfigurasi sistem sudah valid dan proses pengujian bisa dilakukan.

3.5 Pengujian

3.5.1 Metode

Proses pengujian untuk masing-masing skenario dalam penelitian ini akan dilakukan dengan beberapa kondisi lalu lintas data, yaitu berdasarkan jumlah *client* yang mengakses *server* secara bersamaan. Dengan menggunakan

Webserver Stress Tool sebagai *user* simulator, disimulasikan sejumlah *user* melakukan *request* ke suatu halaman *web* (melalui *web filtering server*) secara bersamaan. Kondisi lalu lintas data yang akan diujikan akan dimulai dari 500 user hingga mencapai 5000 user dengan kelipatan 500. Masing-masing kondisi pengujian dilakukan sebanyak 5 kali, dan pada saat proses tersebut berlangsung akan dicatat dan diambil nilai rata-rata dari parameter-parameter yang digunakan untuk mengukur kinerja squidguard dan dansguardian.

Kondisi lalu lintas data ini dipilih karena berdasarkan beberapa percobaan yang penulis lakukan sebelum penelitian ini dilakukan, kondisi lalu lintas data dengan selisih 500 user merupakan kondisi yang cukup terlihat perbedaannya sehingga lebih mudah untuk dilakukan analisis nantinya.

Pada saat proses pengujian tiap-tiap *server web filtering* berlangsung, akan diusahakan dalam kondisi yang sama. Ketika server Squidguard diuji, *virtual machine* yang dinyalakan hanya mesin Squidguard dan mesin Webserver, sedangkan mesin Dansguardian dimatikan. Begitu pula sebaliknya, ketika server Dansguardian diuji, *virtual machine* yang dinyalakan hanya mesin Squidguard dan mesin Webserver, sedangkan mesin Squidguard dimatikan.

3.5.2 Parameter Pengujian

Menurut Ciliendo dan Kunimasa(2007), performa sebuah server dapat ditinjau dari beberapa parameter. Setiap parameter tersebut memiliki metrik-metrik sebagai pengukur kinerja server. Metrik-metrik tersebut diantaranya adalah:

1. Metrik *Memory usage* untuk parameter penggunaan memory.
2. Metrik *Load average* untuk parameter penggunaan prosesor.
3. Metrik rata-rata waktu eksekusi data saat user mengakses untuk parameter *time*.

Cara pengambilan nilai metrik untuk masing-masing parameter tersebut dalam pengujian ini adalah sebagai berikut:

1. Prosesor

Utilitas CPU akan dilihat berdasarkan *load average*. *Load average* merupakan beban rata-rata sebuah sistem didalam menjalankan banyak proses pada sebuah server. Nilai load average berkaitan erat dengan berapa banyak core CPU pada sebuah server. Jika *load average* bernilai 1.00 maka utilitas CPU adalah 100% pada CPU 1 core, jika bernilai 2.00 maka utilitas CPU adalah 100% pada CPU dengan 2 core, demikian seterusnya. Jika load average bernilai 1.70 pada CPU dengan 1 core, maka itu artinya CPU kelebihan beban sebesar 70% daripada beban kerja total yang dapat ditangani oleh CPU tersebut.

Semakin kecil nilai *load average*, maka semakin bagus performa dari server tersebut. Untuk melihat data *load average* akan digunakan perintah “TOP”. Data parameter ini diambil di mesin server Squidguard/Dansguardian.

2. Memori

Penggunaan memori akan dilihat berdasarkan jumlah memori yang terpakai (*memory used*). Semakin rendah memori yang dipakai semakin baik Untuk melihat data memori yang dipakai akan digunakan perintah “TOP”. Data parameter ini diambil di mesin server Squidguard/Dansguardian.

Contoh perintah “TOP” untuk mengambil nilai *load average* dan *memory used* dapat dilihat pada Gambar 3.2.

```

gopar@dansguardian: ~
top - 02:06:52 up 16 min, 1 user, load average: 0.20, 0.07, 0.01
Tasks: 110 total, 1 running, 109 sleeping, 0 stopped, 0 zombie
Cpu(s): 1.0%us, 0.7%sy, 98.3%id, 0.0%wa, 2.3%hi, 10.3%si, 0.0%st
Mem: 508392k total, 145932k used, 362460k free, 11324k buffers
Swap: 492536k total, 0k used, 492536k free, 78832k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 1093 dansguar  20   0 35716 21m 1064  S   7.3   4.3   0:00.97  dansguardian
 1094 dansguar  20   0 35716 21m 1064  S   3.3   4.3   0:00.29  dansguardian
   890 proxy    20   0  7796 5200 1764  S   1.0   1.0   0:00.76  squid
 1095 dansguar  20   0 35716 21m 1064  S   0.3   4.3   0:00.07  dansguardian
 1097 dansguar  20   0 35716 21m 1064  S   0.3   4.3   0:00.02  dansguardian
    1 root      20   0  2776 1652 1220  S   0.0   0.3   0:00.80  init
    2 root      20   0    0    0    0  S   0.0   0.0   0:00.00  kthreadd
    3 root      20   0    0    0    0  S   0.0   0.0   0:00.00  migration/0
    4 root      20   0    0    0    0  S   0.0   0.0   0:00.00  ksoftirqd/0
    5 root      20   0    0    0    0  S   0.0   0.0   0:00.00  watchdog/0
    6 root      20   0    0    0    0  S   0.0   0.0   0:00.12  events/0
    7 root      20   0    0    0    0  S   0.0   0.0   0:00.00  cpuset
    8 root      20   0    0    0    0  S   0.0   0.0   0:00.00  khelper
    9 root      20   0    0    0    0  S   0.0   0.0   0:00.00  netns
   10 root      20   0    0    0    0  S   0.0   0.0   0:00.00  async/mgr
   11 root      20   0    0    0    0  S   0.0   0.0   0:00.00  pm
   12 root      20   0    0    0    0  S   0.0   0.0   0:00.00  sync_supers

```

Gambar 3.2 Contoh pengujian *memory usage* dan *load average*

3. Time

Time yang dimaksud adalah jeda waktu antara *request* yang dikirim *user* dengan respon yang diterima *user* dari *server*. Data jeda waktu akses ini akan diambil nilai rata-rata dari *log* Webserver Stress Tool pada mesin *client*. Jika rata-rata waktu akses rendah berarti sistem bekerja dengan baik. Contoh rata-rata waktu akses dari 2000 user mengakses secara bersamaan dapat dilihat pada Gambar 3.3.

```

Completed Clicks: 542 with 0 Errors (=0.00%)
Average Click Time for 2,000 Users: 352 ms
Successful clicks per Second: 48.17 (equals 173,413.66 Clicks per Hour)

Results of period #2 (from 13 sec to 25 sec):
*****
Completed Clicks: 1367 with 0 Errors (=0.00%)
Average Click Time for 2,000 Users: 582 ms
Successful clicks per Second: 118.35 (equals 426,069.76 Clicks per Hour)

Results of complete test
*****

** Results per URL for complete test **

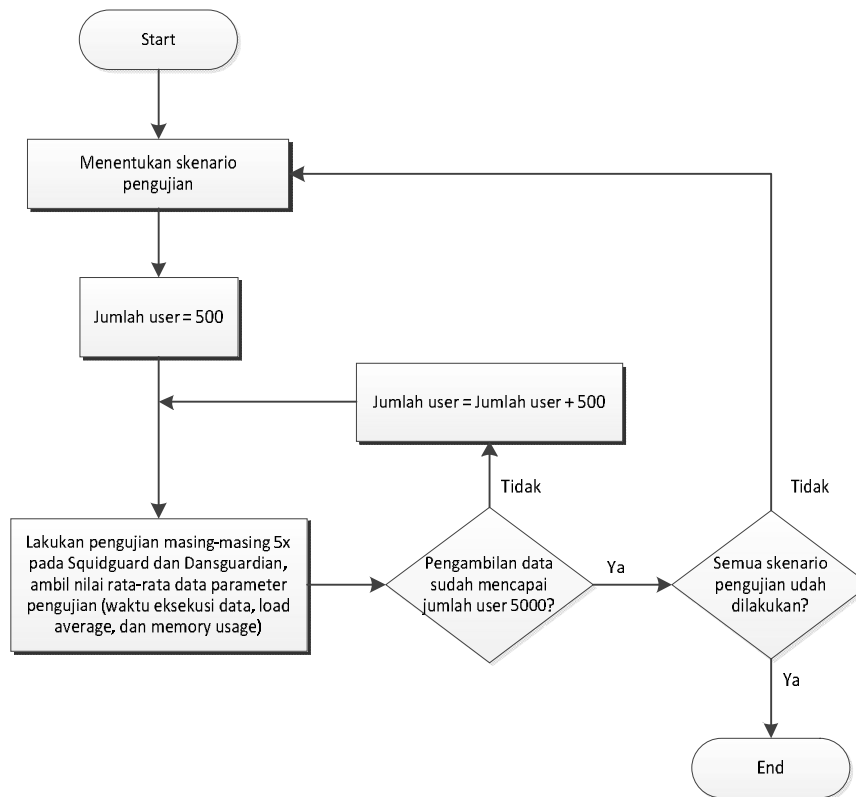
URL #1 (): Average Click Time 517 ms, 1,909 Clicks, 0 Errors

Total Number of Clicks: 1,909 (0 Errors)
Average Click Time of all URL: 517 ms
  
```

Gambar 3.3 Contoh pengujian rata-rata waktu eksekusi data

3.5.3 Alur Pengujian dan Pengambilan Data

Secara garis besar, alur pengujian dan pengambilan data dalam pengujian ini dapat dilihat pada gambar 3.4.



Gambar 3.4 Alur metode pengambilan data

3.6 Analisis

Data dari hasil pengujian dicatat dan diolah dengan cara mencari nilai rata-rata tiap parameter pengujian pada masing-masing kondisi pengujian untuk kemudian dianalisis. Nilai rata-rata tersebut akan ditampilkan dalam bentuk tabel dan grafik. Dari tabel dan grafik yang didapat kemudian dilakukan uraian dan penafsiran, bagaimana trend dari masing-masing server yang diuji berdasarkan bentuk grafik, serta ditarik kesimpulan bagaimana perbandingan kinerja Squidguard dan Dansguardian berdasarkan perbandingan nilai rata-rata tiap parameter pada tabel dan grafik tersebut.

Langkah akhir dalam menganalisis data adalah memberi saran dari hasil perbandingan yang telah dilakukan.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Tabel Hasil Pengujian

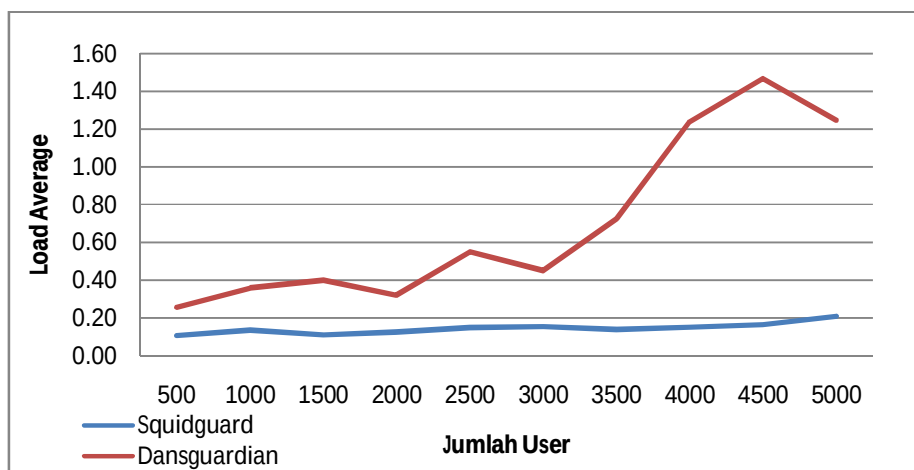
4.1.1 Menggunakan parameter *load average*

Tabel 4.1 menunjukan perbandingan rata-rata load average antara Squidguard dan Dansguardian pada masing-masing kondisi banyaknya jumlah user yang mengakses.

Tabel 4.1 Nilai rata-rata hasil pengujian berdasarkan parameter load average

Jumlah User	Squidguard	Dansguardian
500 user	0.11	0.26
1000 user	0.14	0.36
1500 user	0.11	0.40
2000 user	0.13	0.32
2500 user	0.15	0.55
3000 user	0.16	0.45
3500 user	0.14	0.72
4000 user	0.15	1.24
4500 user	0.17	1.47
5000 user	0.21	1.25

Dari tabel 4.1 menghasilkan grafik seperti pada gambar 4.1



Gambar 4.1 Grafik perbandingan *load average*

Berdasarkan tabel 4.1 dan Gambar 4.1 dapat dilihat bahwa *load average* pada server Squidguard dan server Dansguardian terdapat perbedaan yang sangat signifikan.

Pada server Squidguard nilai *load average* cukup kecil, berkisar antara 0,11 hingga 0,21, dengan perubahan nilai yang cenderung stabil dan tidak signifikan. Meski pada beberapa titik terdapat fluktuasi, akan tetapi secara umum grafik cenderung mengalami peningkatan.

Pada server Dansguardian nilai *load average* lebih tinggi dibandingkan pada server Squidguard pada seluruh kondisi jumlah *user*. Grafik yang terbentuk pun sangat berbeda dengan grafik Squidguard. Pada server Dansguardian perubahan nilai cenderung lambat pada kondisi awal yaitu antara 500 – 3000 *user* (meskipun terdapat fluktuasi), diikuti dengan kenaikan secara drastis pada titik 3000 *user* hingga mengalami kejenuhan pada titik 4500 *user*. Kemudian grafik menunjukkan kecenderungan penurunan pada titik 5000 *user*.

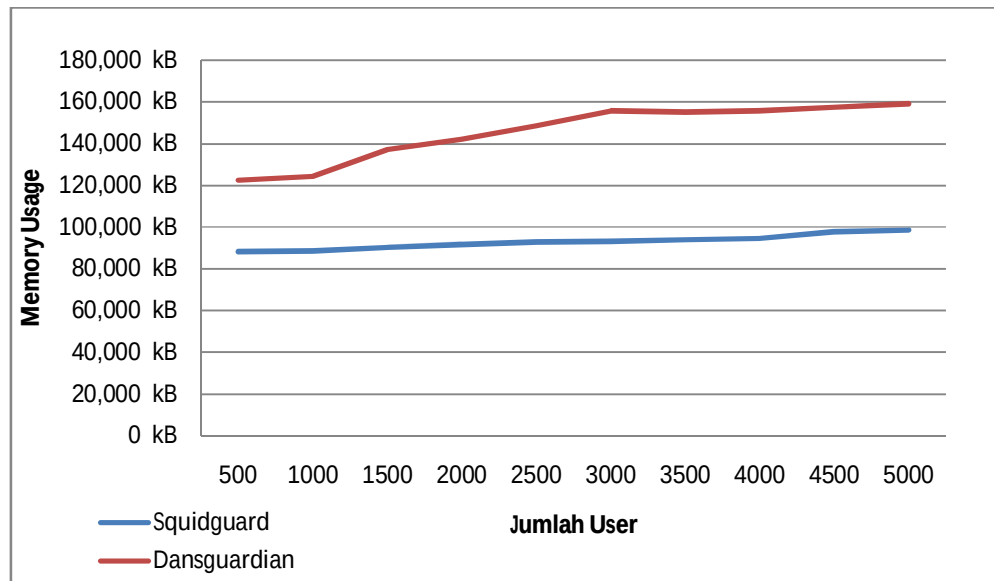
4.1.2 Menggunakan parameter *memory usage*

Tabel 4.2 menunjukkan perbandingan rata-rata *memory usage* antara Squidguard dan Dansguardian pada masing-masing kondisi banyaknya jumlah *user* yang mengakses.

Tabel 4.2 Nilai rata-rata hasil pengujian berdasarkan parameter *memory usage*

Jumlah User	Squidguard	Dansguardian
500 user	88,217 kB	122,625 kB
1000 user	88,421 kB	124,413 kB
1500 user	90,371 kB	137,313 kB
2000 user	91,764 kB	142,174 kB
2500 user	92,770 kB	148,401 kB
3000 user	93,133 kB	155,747 kB
3500 user	94,036 kB	154,975 kB
4000 user	94,479 kB	155,757 kB
4500 user	97,649 kB	157,363 kB
5000 user	98,588 kB	159,111 kB

Dari tabel 4.2 menghasilkan grafik seperti pada gambar 4.2



Gambar 4.2 Grafik perbandingan *memory usage*

Berdasarkan tabel 4.2 dan Gambar 4.2 dapat dilihat bahwa *memory usage* pada server Squidguard dan server Dansguardian juga terdapat perbedaan yang cukup signifikan.

Penggunaan *memory usage* pada server Squidguard hanya berkisar 88.217kB – 98.588 kB sedangkan penggunaan *memory usage* pada server Dansguardian lebih tinggi, yaitu berkisar antara 122,625 kB - 159,111 kB.

Peningkatan penggunaan *memory* pada server Squidguard cenderung lebih stabil dan tidak terlalu signifikan. Pada server Dansguardian peningkatan dari 500 ke 100 user tidak terlalu tinggi, tetapi peningkatan *memory usage* dari 1000 user hingga 3000 user mengalami peningkatan yang cukup tinggi, diikuti dengan sedikit penurunan dari 3000 user ke 3500 user. Mulai dari 3500 user ke 5000 user kembali terjadi peningkatan penggunaan *memory usage* yang stabil dan tidak terlalu signifikan. Peningkatan penggunaan *memory usage* ini tentunya dipengaruhi oleh semakin banyaknya jumlah *request filtering* dari user, sehingga semakin banyak pula data yang harus disimpan memori untuk kemudian disuplai ke CPU untuk dilakukan proses *filtering*.

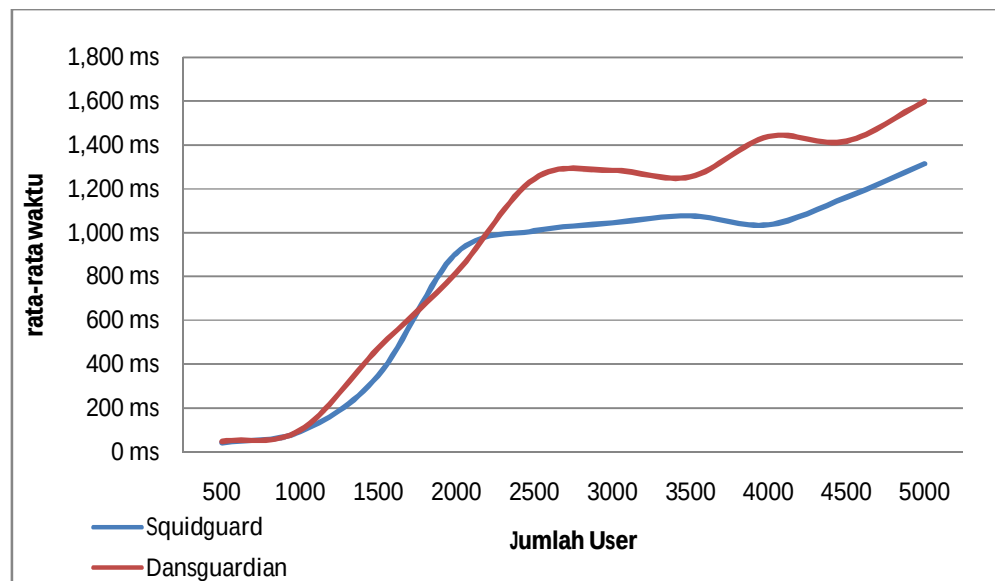
4.1.3 Menggunakan parameter rata-rata waktu eksekusi data

Tabel 4.3 menunjukkan perbandingan rata-rata waktu eksekusi data antara Squidguard dan Dansguardian pada masing-masing kondisi banyaknya jumlah user yang mengakses.

Tabel 4.3 Nilai rata-rata hasil pengujian berdasarkan parameter rata-rata waktu eksekusi data

Jumlah User	Squidguard	Dansguardian	Selisih
500 user	43 ms	48 ms	5 ms
1000 user	93 ms	99 ms	5 ms
1500 user	348 ms	472 ms	124 ms
2000 user	905 ms	817 ms	89 ms
2500 user	1,008 ms	1,246 ms	238 ms
3000 user	1,045 ms	1,284 ms	239 ms
3500 user	1,077 ms	1,255 ms	178 ms
4000 user	1,036 ms	1,438 ms	402 ms
4500 user	1,161 ms	1,417 ms	256 ms
5000 user	1,314 ms	1,600 ms	286 ms

Dari tabel 4.3 menghasilkan grafik seperti pada gambar 4.3



Gambar 4.3 Grafik perbandingan rata-rata waktu eksekusi data

Dari data pada Tabel 4.3 dan Gambar 4.3 menunjukkan nilai hasil pengujian rata-rata waktu eksekusi data tidak terlalu jauh berbeda ketika diuji dengan jumlah user 500 – 2000 user dimana selisih waktu nya berkisar antara 5 ms – 124 ms. Perbedaan yang cukup signifikan terjadi ketika dimulai dari pengujian dengan 2500 user dimana selisih antara kedua server lebih dari 200 ms.

Jika dilihat dari perubahan rata-rata waktu eksekusi data, pada kedua server memiliki kemiripan pada bentuk grafik perkembangan. Perkembangan diawal lambat, diikuti kenaikan secara drastis pada satu titik, hingga pada titik tertentu kenaikan kembali melambat, meski terdapat fluktuasi di beberapa titik.

4.2 Analisis Hasil Pengujian

4.2.1 Berdasarkan data parameter *load average*

Jika dilihat data-data hasil pengujian kedua server untuk parameter *load average*, penggunaan *resource* CPU oleh Squidguard lebih baik dibanding Dansguardian pada seluruh kondisi pengujian.

Nilai *load average* pada Squidguard cukup kecil dengan nilai rata-rata maksimum 0.21 yaitu pada kondisi pengujian dengan 5000 user. Ini berarti beban kerja CPU pada kondisi tersebut rata-rata hanya 21% dari seluruh beban kerja yang dapat ditangani oleh CPU server.

Pada server Dansguardian dengan kondisi pengujian 500 user hingga 3500 user nilai rata-rata *load average* menunjukkan beban kerja pada CPU masih dalam tahap aman karena hanya berkisar antara 26% - 72% dari beban kerja yang dapat ditangani oleh CPU server. Tetapi pada kondisi pengujian 4000, 4500, dan 5000 user nilai rata-rata *load average* melebihi 1.0. Nilai-nilai pada ketiga jumlah user tersebut masing-masing adalah 1.24, 1.47, dan 1.25. Ini menunjukkan bahwa CPU server, yang hanya memiliki 1 core, mengalami kelebihan beban kerja pada masing-masing kondisi sebesar 24%, 47%, dan 25% dari seluruh beban kerja yang dapat ditangani oleh CPU server. Hal ini dimungkinkan terjadi karena pada kondisi pengujian tersebut dengan jumlah data yang harus diproses sangat banyak, Dansguardian membutuhkan banyak *resource* CPU melebihi kemampuan CPU

untuk memproses semuanya sekaligus, sehingga sebagian data tersebut harusmengantri untuk diproses.

4.2.2 Berdasarkan data parameter *memory usage*

Jika dilihat data-data hasil pengujian kedua server untuk parameter *memory usage*, penggunaan *resource* memory oleh Squidguard juga lebih baik dibanding Dansguardian pada seluruh kondisi pengujian. Jika dihitung persentase perbandingan penggunaan memori pada server Squidguard terhadap penggunaan memori pada server Dansguardian maka didapat hasil sebagai berikut.

$$500 \text{ user} = \frac{88,217 \text{ kB}}{122,625 \text{ kB}} \times 100\% = 71.94 \%$$

$$1000 \text{ user} = \frac{88,421 \text{ kB}}{124,413 \text{ kB}} \times 100\% = 71.07 \%$$

$$1500 \text{ user} = \frac{90,371 \text{ kB}}{137,313 \text{ kB}} \times 100\% = 65.81 \%$$

$$2000 \text{ user} = \frac{91,764 \text{ kB}}{142,174 \text{ kB}} \times 100\% = 64.54 \%$$

$$2500 \text{ user} = \frac{92,770 \text{ kB}}{148,401 \text{ kB}} \times 100\% = 62.51 \%$$

$$3000 \text{ user} = \frac{93,133 \text{ kB}}{155,747 \text{ kB}} \times 100\% = 59.80 \%$$

$$3500 \text{ user} = \frac{94,036 \text{ kB}}{154,975 \text{ kB}} \times 100\% = 60.68 \%$$

$$4000 \text{ user} = \frac{94,479 \text{ kB}}{155,757 \text{ kB}} \times 100\% = 60.66 \%$$

$$4500 \text{ user} = \frac{97,649 \text{ kB}}{157,363 \text{ kB}} \times 100\% = 62.05 \%$$

$$5000 \text{ user} = \frac{98,588 \text{ kB}}{159,111 \text{ kB}} \times 100\% = 61.96 \%$$

$$\text{Rata-rata} = 64.10 \%$$

Dari perhitungan tersebut rata-rata perbandingan penggunaan memori pada server Squidguard terhadap server Dansguardian sebesar 64.10%. Ini berarti penggunaan memori pada server Squidguard rata-rata 35.90% lebih kecil bila dibandingkan dengan penggunaan memori pada server Dansguardian.

4.2.3 Berdasarkan data parameter waktu eksekusi data

Dari data-data hasil pengujian kedua server untuk parameter rata-rata waktu eksekusi data, secara umum Squidguard lebih baik dibanding Dansguardian, meskipun terdapat satu kondisi pengujian di mana Dansguardian lebih baik dari pada Squidguard yaitu pada kondisi 2000 user.

Dari data-data tersebut selisih waktu terbesar antara Squidguard dan Dansguardian adalah 402 ms atau 0.4 second. Meskipun secara umum rata-rata waktu eksekusi data Squidguard lebih cepat dibandingkan Dansguardian, tetapi perbedaan waktu terbesar yang hanya 0.4 second tersebut tidak akan begitu terasa perbedaannya oleh user.

4.2.4 Hasil analisis secara umum

Dari hasil pengujian berdasarkan parameter waktu eksekusi data serta penggunaan *resource* memori dan CPU dapat disimpulkan bahwa kinerja Squidguard lebih baik dibandingkan Dansguardian karena *cost* yang digunakan oleh Squidguard dalam membangun *web filtering* lebih rendah daripada Dansguardian dengan *performance* yang tidak jauh berbeda.

Jika manajemen memutuskan untuk mengimplementasikan *web filtering* dengan menggunakan Squidguard, maka keuntungan yang didapatkan adalah *cost* yang dibutuhkan lebih kecil dibanding Dansguardian. Tetapi kerugiannya adalah fitur yang dimiliki Squidguard hanya terbatas pada *web filtering* berdasarkan domain (*domain filtering*), sehingga tidak bisa digunakan untuk *web filtering* berdasarkan konten (*content filtering*).

Sebaliknya, jika manajemen memutuskan untuk mengimplementasikan *web filtering* dengan menggunakan Dansguardian, maka akan dibutuhkan *cost* yang lebih besar dibandingkan Squidguard. Tetapi keuntungannya adalah fungsi

filtering pada Dansguardian memiliki potensi untuk dikembangkan tidak hanya sebagai *domain filtering* saja tetapi juga dapat melakukan *content filtering*. Dengan potensi tersebut, jika organisasi memiliki resource yang besar, maka lebih baik menggunakan Dansguardian.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan tentang perbandingan Squidguard dan Dansguardian berdasarkan parameter *load average*, *memory usage*, dan waktu eksekusi data, maka dapat ditarik kesimpulan sebagai berikut:

1. Melihat hasil pengujian dalam tugas akhir ini, dapat dilihat bahwa *cost* yang digunakan oleh Squidguard lebih rendah daripada Dansguardian, sehingga dapat disimpulkan bahwa Squidguard lebih baik daripada Dansguardian dalam membangun *web filtering* berdasarkan domain (*domain filtering*).
2. Dari hasil pengujian berdasarkan parameter *load average* dengan menggunakan spesifikasi komputer yang digunakan dalam tugas akhir ini yaitu CPU 1 GHz 1 core, menunjukkan bahwa pada server Squidguard masih dapat menangani kondisi pengujian dengan baik karena nilai *load average* pada seluruh kondisi pengujian masih cukup rendah. Pada server Dansguardian *load average* masih dalam tahap aman hanya pada kondisi pengujian 500 – 3500 user, sedangkan ketika pada kondisi pengujian 4000 – 5000 user nilai *load average* menunjukkan CPU mengalami kelebihan beban.
3. Dari hasil pengujian berdasarkan parameter *memory usage* dengan menggunakan spesifikasi komputer yang digunakan dalam tugas akhir ini yaitu memori 512 MB (524,288 kB), menunjukkan bahwa kedua server masih dapat menangani kondisi pengujian dengan baik karena *memory usage* pada server Squidguard hanya berkisar antara 88.217kB – 98.588 kB dan pada server Dansguardian berkisar antara 122,625 kB - 159,111 kB, masih cukup banyak memori yang tidak terpakai.
4. Meskipun rata-rata waktu eksekusi data dari hasil penelitian menunjukkan secara umum Squidguard lebih baik dibandingkan Dansguardian, tetapi secara *user experience* perbedaan antara kedua server tersebut tidak terlalu kentara

dirasakan oleh user karena nilai terbesar selisih waktu rata-rata eksekusi data antara kedua server adalah hanya 0.4 second.

5.2 Saran

Setelah melihat hasil kesimpulan diatas, berikut ini adalah saran yang mungkin berguna dan dapat dipakai oleh organisasi yang ingin menggunakan *web filtering* :

1. Jika organisasi memiliki resource yang terbatas, maka lebih baik menggunakan Squidguard untuk diimplementasikan sebagai *web filtering* karena penggunaan resource server serta waktu eksekusi data lebih kecil dibandingkan Dansguardian.
2. Jika organisasi memiliki resource yang besar, lebih baik menggunakan Dansguardian untuk diimplementasikan sebagai *web filtering* karena fitur filtering yang dimiliki Dansguardian lebih banyak.

DAFTAR PUSTAKA

- Alvey, Robert, 2004. *The Art of Web Filtering* (On-line) Available at www.sans.org/reading_room/whitepapers/bestprac/art-web-filtering_1375, diakses tanggal 16 Maret 2013.
- Barron, Daniel, 2011. *DansGuardian - True Web Content Filtering for All* (On-line) Available at <http://dansguardian.org/?page=whatisdg>. diakses tanggal 1 Agustus 2012.
- Ciliendo danKunimasa, 2007. *Linux Performance and Tuning Guidelines* (On-line) Available at <http://www.redbooks.ibm.com/abstracts/redp4285.html>, diakses tanggal 3 Agustus 2012.
- Gunther,Neil J., 2010. *UNIX Load Average Part 1: How It Works* (On-line) Available at <http://www.teamquest.com/pdfs/whitepaper/ldavg1.pdf>, diakses tanggal 5 Juni 2013.
- Liberian Geek, 2010. *Install, Configure and Manage DansGuardian In Ubuntu 10.04 Lucid Lynx via Web Content Control* (On-line) Alailable at <http://www.liberiangeek.net/2010/08/install-configure-manage-dansguardian-ubuntu-10-04-lucid-lynx-web-content-control/>, diakses tanggal 10 September 2012.
- Rucker, John C., 2006. *Ubuntu and DansGuardian for Filtering and Public Wireless Access* (On-line) Available at http://www.branchdistrictlibrary.org/professional/ubuntu_and_dansguardian.php, diakses tanggal 10 September 2012.
- Shalla Secure Services KG, 2007. *About squidGuard* (On-line) Available at <http://www.squidguard.org/about.html>, diakses tanggal 21 September 2012.

Walker, Ray, 2006. *Examining Load Average* (On-line) Available at <http://www.linuxjournal.com/article/9001>, diakses tanggal 5 Juni 2013.

LAMPIRAN