HALAMAH JUDUL

ANALISIS PENGGUNAAN RESOURCE SERVER DAN WAKTU EKSEKUSI DATA PADA DANSGUARDIAN SERVER BERDASARKAN VARIASI BANDWIDTH

LAPORAN TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana

Strata-1 Teknik Informatika

Oleh :

Nama : Andika Kholifah Gilar Pratiwi

No.Mahasiswa : 07523309

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

YOGYAKARTA

2011

LEMBAR PENGESAHAN DOSEN PEMBIMBING

ANALISIS PENGGUNAAN RESOURCE SERVER DAN WAKTU
EKSEKUSI DATA PADA DANSGUARDIAN SERVER
BERDASARKAN VARIASI BANDWIDTH

TUGAS AKHIR

Oleh :

Nama : Andika Kholifah Gilar Pratiwi

No.Mahasiswa : 07 523 309

Yogyakarta, 30 November 2011

Pembimbing,

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

LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR

Saya yang bertandatangan di bawah ini,

Nama : Andika Kholifah Gilar Pratiwi

No. Mahasiswa : 07 523 309

Menyatakan bahwa seluruh komponen dan isi dalam Laporan Tugas Akhir ini adalah hasil karya saya sendiri.Apabila di kemudian hari terbukti bahwa ada beberapa bagian dari karya ini adalah bukan hasil karya saya sendiri, maka saya siap menanggung risiko dan konsekuensi apapun.

Demikian pernyataan ini saya buat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 30 November 2011

Andika Kholifah Gilar Pratiwi

LEMBAR PENGESAHAN DOSEN PENGUJI

ANALISIS PENGGUNAAN RESOURCE SERVER DAN WAKTU EKSEKUSI DATA PADA DANSGUARDIAN SERVER BERDASARKAN VARIASI BANDWIDTH

TUGAS AKHIR

Oleh :

Nama : Andika Kholifah Gilar Pratiwi

No.Mahasiswa : 07 523 309

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

Yogyakarta, 30 November 2011

Tim Penguji,

Tim Penguji,

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

Raden Teduh Dirgahayu Dr.S.T.,M.Sc.
Anggota I

Ahmad Munasir Raf'ie Pratama S.T.,M.I.T
Anggota II

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Teknologi Industri
Universitas Islam Indonesia

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

ii

HALAMAN PERSEMBAHAN

Kupersembahkan tugas akhirku ini kepada……

Allah SWT yang telah mendengarkan doa dan keluh kesahku
serta atas limpahan nikmat, karunia dan rahmat-Nya yang tiada henti-henti nya…..

Bapak dan Ibuku tercinta, atas doa, kasih sayang,
cinta, pengertian, dukungan moral, spiritual, dan kepercayaannya.....

Adik-adikku tersayang Ardita Permata Alfiani dan Adinda Arka Maulita yang
selalu mendukung...

Seseorang yang bernama Maulanasyah Abdul Djafar yang selalu menjadi
inspirasiku dan menemaniku selalu dalam suka dukaku....

Seluruh teman-teman di Lab SisJarkom tercinta,terimakasih banyak atas
dukungan,bantuan,motivasi dan persahabatannya.....

Semua teman-temanku atas dukungan dan bantuannya
semoga persahabatan kita tidak pernah terhenti.....

HALAMAN MOTTO

"Keyakinan merupakan satu-satunya penawar kegagalan yang diketahui orang!"

(Napoleon Hill, Think & Grow Rich)

“Tidak ada satupun di dunia ini, yang bisa di dapat dengan mudah.
Kerja keras dan doa adalah cara untuk mempermudahnya.”

(The Seven Habits of Highly Effective Teens)

“Trying is a part of failing. If you are afraid to fail then you're afraid to try.

(Mrs. Cunningham)

KATA PENGANTAR

Alhamdulillahi Rabbil'alamin. Puji dan syukur kehadirat 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 PENGGUNAAN RESOURCE SERVER DAN WAKTU EKSEKUSI DATA PADA DANSGUARDIAN SERVER BERDASARKAN VARIASI BANDWIDTH ".**

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. Allah SWT yang selalu memberi hamba nikmat sehat, islam, iman.
2. Yang tercinta Ayahanda (Widiyanto Kabul)dan Ibunda (Endang Pastorini Ary M S.Pd), serta adik-adikku (Ardita Permata Alfiani&Adinda Arka Maulita), semoga Allah SWT membalas amal budinya dengan kasih sayang yang berlimpah.
3. Bapak Syarif Hidayat ,S.Kom., M.I.T, selaku pembimbing yang telah meluangkan waktu untuk berdiskusi selama penyusunan tugas akhir.
4. Bapak Yudi Prayudi, S.Si., M.Kom., selaku Ketua Jurusan, Jurusan Teknik Informatika, Fakultas Teknologi Industri.
5. Seseorang yang bernama Maulanasyah Abdul Djafar, S.Kom terima kasih atas dorongan semangat dan doanya selama ini.

6. Sahabatku Isna, Bitana, Depi, dan semua yang mengenalku, terima kasih untuk semangat, bantuan dan waktunya selama kurang lebih 4 tahun di UII tercinta ini.
7. Sahabat-sahabatku tercinta di Lab SisJarkom, Citra, Bitana, Devi, Yopi, Irwan, Sugi, Rangga, Mas Lantip, Mas Bram, Mba nita, Mas Doni, Mas Rohmat, Nita Niti, Dodo, Rifcong, Gendro, Kiki, Mas Bro Valent, Irfan Geli, Hanafi Kentang, Olip, Amin dan Timur atas dukungan, bantuan, motivasi dan persahabatannya selama ini.
8. Keluarga besar informatika, khususnya INCLUDE.
9. Serta semua pihak yang telah turut membantu hingga selesainya penyusunan tugas akhir ini. Semoga Allah SWT membalas budi baik dan keikhlasannya, Amin.

Penulis menyadari sepenuhnya bahwa tugas akhir ini masih jauh dari kesempurnaan suatu tulisan ilmiah, oleh sebab itu dengan segala kerendahan hati penulis menerima kritik dan saran demi kesempurnaannya.

Akhir kata, semoga tugas akhir ini dapat memberikan suatu manfaat yang sebesar-besarnya bagi kita semua.

Yogyakarta, 30 November 2011

Penulis,

Andika Kholifah Gilar Pratiwi

SARI

Salah satu metode yang dapat digunakan untuk melakukan penyaringan informasi di internet adalah menggunakan aplikasi *Web Content Filtering.Danguardian merupakan* salah satu aplikasi yang sering digunakan.Selain karena kemudahan dalam pengimplementasiannya, *Dansguardian* juga bersifat gratis untuk penggunaannya.

Dalam pemanfaatan *Dansguardian* sebagai *Web Content Filtering*, terdapat banyak sekali situs yang sebenarnya diperlukan untuk ilmu pengetahuan, namun terdapat konten yang bisa disalahgunakan. Situs-situs tersebut dapat digolongkan menjadi *black list* dan *grey list*.

Bandwidth merupakan lebar pita untuk transfer data, yaitu jumlah data yang dapat dibawa dari sebuah titik ke titik lain dalam jangka waktu. Dalam hal ini, Dansguardian yang terdapat di tengah sebuah jaringan komputer, akan melakukan *filtering* terhadap seluruh data yang melewatinya yang menyebabkan paket data yang lewat harus mengantri untuk melakukan filtering. Sehingga antrian paket data akan memenuhi bandwidth yang terdapat pada jaringan komputer.

Untuk itu, perlu diketahui waktu saat paket data pada bandwidth menjadi sangat penuh dan menghambat jaringan komputer. *Filtering* oleh Dansguardian kemungkinan berpengaruh juga terhadap performa sebuah server yang dapat membuat server menjadi bekerja lebih saat melakukan *filtering* agar paket data dapat segera melewati server.

Kata kunci :*Web Content Filtering, Dansguardian, Bandwidth*

TAKARIR

bottle neck	penghambat
queue	antrian
content filtering	penyaringan konten
bandwidth management	pengaturan bandwidth
gateway	pintu keluar
Carrier	pembawa sinyal
Caching	penyimpanan object yang sudah di akses
Upadate	memperbaharui
Request	permintaan
outgoing traffic	lalu lintas jaringan yang menuju keluar
source IP	sumber IP
Banned	dilarang
content disposition	konten disposisi
factory default	standar aturan pabrik
Performance	kinerja
Source	sumber
Cache	tempat penyimpanan pada proxy
Delay	penundaan
memory usage	penggunaan memori
load average	rata-rata beban

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Teknologi *internet* terus dikembangkan untuk memudahkan setiap orang di dunia dapat terhubung satu dengan yang lainnya dengan cepat. Penyaringan informasi yang diperoleh dari *internet* perlu dilakukan apabila suatu pihak atau instansi tertentu menginginkan pelarangan akses terhadap beberapa jenis informasi tertentu dari *internet*. Salah satunya menggunakan proxy.

Salah satu fungsi proxy adalah sebagai *gateway*. *S*ebagai pintu keluar / masuk paket data yang ada dalam sebuah jaringan komputer, *gateway* juga harus bisa menjadi pelindung jaringan lokal dibawahnya dari bahaya yang terdapat di internet. Proxy melakukan filtering berdasarkan alamat web dan beberapa kata, tetapi tidak semua halaman web mengandung hal-hal ‘buruk’ untuk di akses. Sedangkan proxy, akan memblokir seluruh halaman tersebut tanpa kecuali.

Untuk mendukung fungsi proxy,di gunakan banyak macam web filtering normal seperti *Cyber Patrol, TinyproxyGuard, Net Nanny*, dan masih banyak lagi. Web filtering ini memiliki daftar situs yang ‘buruk’ sangat banyak. Jika *user* mencoba untuk mengakses situs yang terdapat pada *list* tersebut, situs yang di akses akan diblokir berdasarkan alamat web-nya.

Web mengalami perubahan yang sangat cepat dan tidak dapat di prediksi waktunya, bahkan *search engine* dari provider besar seperti *Google* atau *Yahoo* bahkan tidak tahu perubahan apa yang sedang dan akan terjadi. Hal ini membuat *filtering* dengan alamat web (URL) menjadi sulit jika harus dilakukan setiap waktu. Untuk melakukan *filtering* yang komprehensif hanya berdasarkan URL, yang dibutuhkan adalah sesuatu bisa memeriksa setiap halaman web yang di akses dari hal-hal yang 'buruk' seperti obat, *profanities*, kekerasan, pornografi, kemudian melakukan pemblokiran halaman tersebut jika isi di dalamnya di anggap tidak baik untuk di akses. Hal ini yang disebut dengan *'Content Filtering'*.

DansGuardian merupakan pemenang penghargaan *Open Source web content filter* yang saat ini berjalan pada *Linux, FreeBSD, OpenBSD, NetBSD, Mac OS X, HP-UX*, dan Solaris. Ini merupakan *Content Filtering* yang didasarkan pada banyak metode yang di dalamnya, termasuk *phrase matching, PICS filtering* dan *URL filtering*. Ini bukan merupakan *filtering* murni berdasarkan daftar situs yang dilarang untuk di akses. DansGuardian dirancang agar bisa lebih fleksibel dan memungkinkan untuk menyesuaikan *filtering* sesuai dengan kebutuhan.

Dansguardian merupakan aplikasi yang berjalan berjalan pada sisi server. Sebagai *gateway*, server yang akan dilewati oleh paket-paket data yang mengantri saat di akses oleh klien. Pada server yang terdapat Dansguardian, paket-paket tersebut akan mengalami ‘pemeriksaan’ oleh Dansguardian terlebih dahulu sebelum sampai ke *client.*

Pemeriksaan tersebut berpotensi menghambat kelancaran paket-paket data untuk segera sampai kepada klien yang mengaksesnya karena harus mengantri untuk ‘di periksa’ terlebih dahulu oleh Dansguardian *(bottle neck)*. Untuk itu, diperlukan ketepatan dalam menentukan *resource* dan pengaturan yang tepat oleh seorang *Network Administrator* agar Dansguardian dapat berjalan dengan baik dalam sebuah jaringan komputer.

Pengaturan yang tepat, dapat dilakukan jika seorang *Network Administrator* mengetahui apa saja yang akan terjadi jika mengaplikasikan Dansguardian dalam servernya. *Resource* yang tepat untuk server yang terdapat Dansguardian, *delay* yang mungkin terjadi, antrian (*queue*) yang bisa jadi akan menumpuk di server karena pemeriksaan yang di lakukan oleh Dansguardian patut untuk menjadi pertimbangan.

1.2 Rumusan Masalah

Masalah yang sering dihadapi dalam hal *web content filtering* adalah **mengetahui skema penggunaan server Dansguardian dan waktu eksekusi data berdasarkan variasi bandwidth pada sebuah jaringan komputer**. Keberadaan Dansguardian akan sangat berpengaruh karena akan dilakukan

“pemeriksaan” konten serta URL pada paket data yang melewatinya dengan sistem *queue.*

1.3 Batasan Masalah

Adapun batasan masalah yang di angkat dalam penelitian ini sebagai berikut :

1. Membangun *Web Content Filtering* server menggunakan *Dansguardian* pada sistem operasi *Linux CentOS-6.0.*
2. Mengatur mikrotik *routerboard* untuk sebagai *router* dan *bandwidth management.*
3. Halaman web yang digunakan untuk pengujian, berisi kata-kata yang akan di filter oleh Dansguardian.
4. Penilaian kinerja *Dansguardian* sebagai *Web Content Filtering* server meliputi beberapa parameter, yaitu bagaimana pengaruh penggunaan Dansguardian terhadap waktu eksekusi data dalam menyaring konten yang tidak pantas, serta penggunaan *resource* server (*memory usage* dan *load average*).

1.4 Tujuan Penelitian

Tujuan dari tugas akhir adalah menganalisa server Dansguardian dengan variasi bandwidth berdasarkan parameter waktu eksekusi data, *memory usage* dan *load average* untuk mengetahui penggunaan bandwidth yang optimal pada Dansguardian server.

1.5 Manfaat Penelitian

Melalui hasil analisis jaringan komputer dengan meletakkan *Dansguardian* di dalamnya, *network administrator* dapat meng-optimalkan kemampuan jaringan komputernya dalam hal *Web Content Filtering*, mengetahui penggunaan Dansguardian berdasarkan *bandwidth* dan *resource*nya, sehingga penggunaan sistem penyaringan informasi *internet* yang kita inginkan bekerja secara maksimal.

1.6 Metode Penelitian

1.6.1 Studi Pustaka

Studi pustaka digunakan untuk menggali informasi yang terkait dengan penelitian, yaitu melalui buku-buku dan internet yang berkaitan dengan penelitian. Meliputi pemilihan perangkat lunak yang di gunakan serta pencarian referensi mengenai tahapan dalam analisa efektifitas *Web Content Filtering* menggunakan *Dansguardian*.

1.6.2 Implementasi dan Konfigurasi Sistem

Analisa efektifitas Dansguardian dalam jaringan komputer di susun berdasarkan perolehan dari studi pustaka yang meliputi:

a. Desain Arsitektur Jaringan

   Tahap ini merupakan perancangan arsitektur jaringan komputer yang akan digunakan untuk melakukan analisa pada efektifitas *Dansguardian* yang diterapkan dalam sebuah jaringan komputer.

b. Desain Alur data proses aplikasi

   Tahap ini merupakan perancangan visualisasi data yang mengalir pada setiap proses yang terjadi pada masing-masing aplikasi dari awal hingga akhir.

c. Pengadaan Perangkat Keras

   Tahapan ini merupakan tahap pengadaan perangkat keras yang berupa mikrotik RB750, computer client serta komputer server yang nantinya akan dibuat sebagai *proxy, Dansguardian* dan web server.

d. Installasi dan Konfigurasi Sistem

   Tahapan ini merupakan tahap installasi dan konfigurasi dari masing-masing server dan sistem yang merupakan mikrotik *routerboard, Linux OS,* serta *Dansguardian*.

e. Analisis Kinerja Jaringan Komputer

   Tahapan ini merupakan tahap analisis dari seluruh jaringan komputer yang sudah dibuat berdasarkan topologinya dan sudah diletakkan *Dansguardian* di dalamnya.

f. Pengambilan Kesimpulan

Tahap selanjutnya adalah mengumpulkan data hasil analisis untuk dilihat hasilnya, kemudian mengambil kesimpulan dari hasil pengumpulan data tersebut untuk mengetahui *efficiency* dari *Dansguardian* sebagai *Web Content Filtering.*

1.7 Sistematika Penulisan

BAB I Pendahuluan

Membahas tentang latar belakang masalah, batasan masalah, rumusan masalah, tujuan penelitian serta manfaat dari penelitian dan metodologi yang di angkat menjadi materi laporan tugas akhir Analisis Efektifitas dan Resource Server Pada Dansguardian Sebagai Implementasi Web Content Filtering.

BAB II Landasan Teori

Membahas dasar-dasar teori yang digunakan dalam perancangan dan pembangunan sebuah *gateway* server sebagai *bandwidth management* dan Dansguardian sebagai *Web Content Filtering* yang kemudian akan di analisis efektifitas kinerjanya.

BAB III Metodologi

Memuat uraian tentang analisis masalah, gambaran umum sistem, analisis kebutuhan sistem yang mencakup kebutuhan perangkat keras, perangkat jaringan, dan perangkat lunak yang digunakan untuk membantu penyelesaian tugas akhir.

BAB IV Hasil dan Pembahasan

Memuat dokumentasi mulai dari tahap installasi, konfigurasi dan pengujian terhadap aplikasi, yaitu dengan melakukan analisis terhadap efektifitas kinerja *Dansguardian*, *resource* server dan penggunaan

bandwidth yang tepat, kemudian mengambil kesimpulan dari data yang di dapat.

BAB V Penutup

Memuat kesimpulan-kesimpulan dari seluruh rangkaian proses implementasi perangkat lunak, baik pada tahap analisis, perancangan, implementasi, terutama pada analisis efektifitas kinerja *Dansguardian* serta *resource* server dan *bandwidth* yang digunakan.

BAB 2

LANDASAN TEORI

2.1 Jaringan Komputer

2.1.1 Pengertian Jaringan Komputer

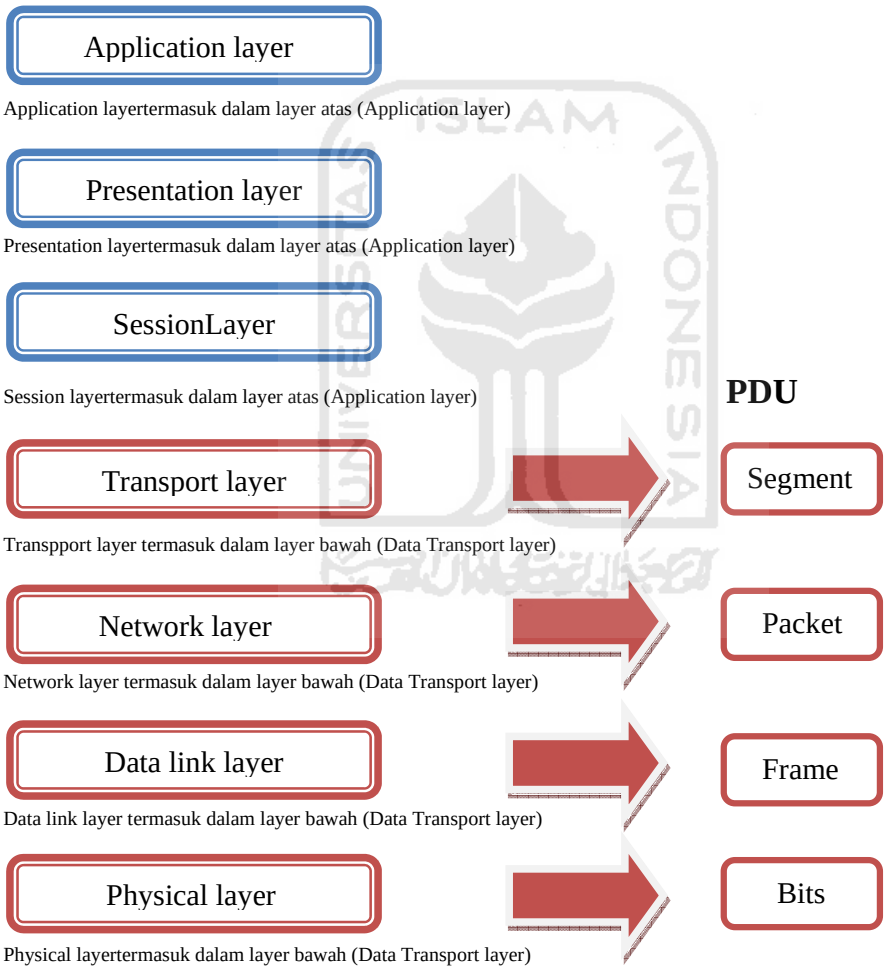
Jaringan komputer adalah dua atau lebih komputer yang saling terhubung dengan menggunakan media transmisi baik itu kabel maupun nirkabel, sehingga dapat berbagi sumber daya.

2.1.2 Cara Kerja Jaringan Komputer

Bagian utama dalam jaringan komputer adalah sistem komunikasi data.Sistem komunikasi data membutuhkan media sebagai pembawa sinyal(*carrier*), yang dapat berupa kabel, *fiber optic* dan juga melalui udara (sinyal gelombang radio).Untuk menerima dan mengirimkan sinyal dari media perantara tersebut menggunakan NIC (*Network Interface Card*) yang terdapat di dalam sebuah perangkat komputer. Setiap NIC memiliki alamat khusus yang disebut sebagai MAC (*Media Acces Control*) address, MAC addres terdiri dari kode heksa desimal yang berbeda antara NIC satu dengan lainnya. MAC address menghindarkan data bertabrakan ketika berkomunikasi ada jaringan. Jika ada paket data yang akan dikirimkan maka sebelumnya akan melihat apakah di dalam jaringan ada yang sedang mengirimkan paket data atau tidak, jika jaringan tidak sedang digunakan (melakukan pengiriman paket data) maka node akan langsung mengirimkan paket data ke komputer tujuan.

Model referensi OSI(*Open System Interconnection*) merupakan salah satu arsitektur jaringan komputer yang di buat oleh ISO (*International Standarization Organization*) untuk memecahkan permasalahan kompatibilitas *devices* antar vendor. Model referensi OSI mengidentifikasikan semua proses yang di butuhkan untuk melakukan komunikasi dan membaginya ke dalam kelompok secara logika yang disebut *layer*. OSI merupakan petunjuk bagi para pengembang aplikasi dalam membuat dan mengimplementasikan aplikasinya pada sebuah jaringan.

OSI terdiri dari tujuh *layer*, yang secara umum terbagi dalam dua kelompok, yaitu layer atas (*Application Layer*) dan *layer* bawah (*Data Transport Layer*). Lapisan yang tergolong dalam *layer* atas mendefinisikan bagaimana aplikasi pada sebuah *host* akan berkomunikasi dengan *host* lainnya, sedangkan *layer* bawah mendefinisikan bagaimana data terkirim dari suatu *host* ke *host* lainnya. Proses komunikasi dan pertukaran informasi dari tiap-tiap layer menggunakan sebuah *protocol* yang disebut sebagai *Protocol Data Unit* (PDU).



Gambar 2.1 Gambar referensi model OSI

Berikut penjelasan dari gambar 2.1 Susunan Layer Referensi Model OSI:

a. *Application Layer*

   Application layer berfungsi sebagai interface antara user dan komputer. A*pplication layer* bertanggung jawab untuk mengidentifikasi ketersediaan dari partner komunikasi. *Application layer* menentukan identitas dan ketersediaan dari partner komunikasi untuk sebuah aplikasi dengan data yang dikirim.

b. *Presentation Layer*

   Presentation layer berfungsi menyediakan sistem penyajian data ke *application layer*, *layer* ini juga memberikan layanan untuk konversi, *sintaks, format* dan *enkripsi data,* dengan menampilkan informasi dalam bentuk teks maupun grafis.

c. *Session Layer*

   Session layer berfungsi dan bertangung jawab untuk mengkoordinasikan jalannya komunikasi antarsistem serta layer ini juga dapat mengendalikan dialog antar komputer.

d. *Transport Layer*

   Transport Layer berfungsi untuk melakukan pengemasan data *upper layer* (*layer* atas) ke dalam bentuk segment, secara opsional menjamin proses pengiriman data yang di andalkan dalam menjaga keutuhan transmisi data. Proses pengiriman pada layer ini dapat dilakukan dengan 2 mekanisme, yaitu :*Connection-Oriented* dan *Connection-Less.*

e. *Network Layer*

   Network Layer berfungsi untuk melakukan pengemasan data berupa segment yang di terima dari *transport layer* yang akan dikemas ke dalam bentuk *paket.* Ketika *paket* diterima oleh *interface* sebuah *router,* maka alamat tujuan tidak ditemukan maka *paket* tersebut akan dibuang. Layer ini juga bertanggung jawab untuk melakukan mekanisme *routing* melalui *inter network.*

f. *Data Link Layer*

Data link layer mengatur topologi jaringan. *Error notification* dan *flow control.* Lapisan ini juga menyediakan fasilitas alamat perangkat keras dan mengolah paket dari lapisan *network* yang masa paket tersebut akan dibungkus oleh *datalink layer* ke dalam sebuah frame dengan menambahkan informasi mengenai MAC address yang di tuju dan alamat asal.

g. *Physical Layer*

Tanggung jawab dari layer ini adalah melakukan pengiriman dan penerimaan bit. *Physical layer* secara langsung menghubungkan media komunikasi yang berbeda-beda. Lapisan ini bertanggungjawab untuk mengaktifkan dan mengatur anatar muka fisik dari jaringan komputer.

2.2 Konsep Dasar Proxy

Proxy merupakan pihak ketiga yang berdiri di tengah-tengah antara kedua pihak yang saling berhubungan dan berfungsi sebagai perantara dalam sebuah jaringan komputer, sehingga seorang *client* tidak langsung terhubung ke internet. Proxy bekerja dalam berbagai jenis protocol komunikasi jaringan.

Proxy memiliki 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. Selain sebagai pintu keluar / masuk paket data yang ada dalam sebuah jaringan komputer, gateway juga harus bisa menjadi pelindung jaringan lokal dibwahnya dari bahaya yang terdapat di internet.

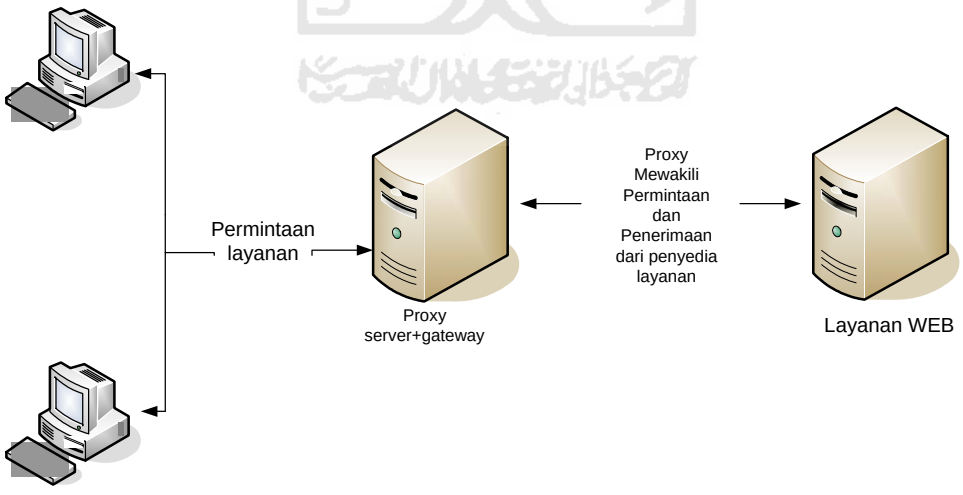
Dengan demikian, koneksi dari jaringan lokal ke internet akan menggunakan sambungan yang dimiliki oleh gateway secara bersama-sama (*connection sharing*). Maka, dalam hal ini, gateway juga berfungsi sebagai proxy server,

karena menyediakan layanan perantara antara jaringan lokal dan jaringan luar atau internet.

Proxy server sebagai sebuah perantara dapat bekerja pada *data link layer*, *network layer*, *transport layer* maupun *application layer* dalam hierarki *layer* komunikasi jaringan menurut OSI. Tetapi, pengertian proxy server dan fungsinya sebagian besar berada pada *application layer*.Proxy server dapat berjalan pada banyak aplikasi antara lain :

1. HTTP Proxy
2. Web Proxy untuk protocol HTTP / Web
3. FTP Proxy
4. SMTP Proxy/POP Proxy untuk e-mail
5. NNTP proxy untuk Newsgroup
6. RealAudio/RealVideo Proxy untuk multimedia streaming.

Kemampuan proxy server sangat baik dalam penyimpanan catatan (*logging*) dari trafik jaringan, dan dapat digunakan untuk memastikan bahwa koneksi untuk jenis trafik tertentu harus selalu tersedia.



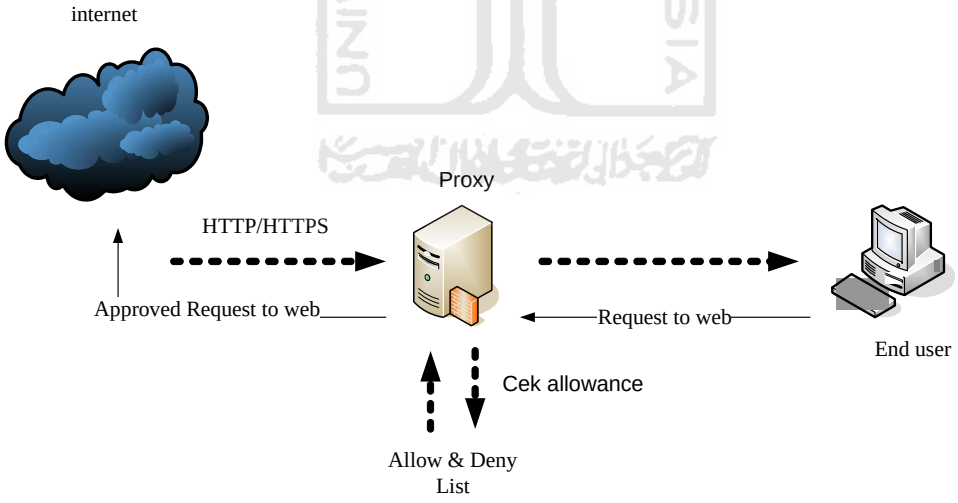
Gambar 2.2 Gambar kinerja layanan Proxy Server

2.2.1 Cache

Fungsi proxy server yang sangat penting adalah untuk *caching. Caching* di sini diartikan sebagai penyimpanan internet object (gambar/halaman web) dari suatu web site yang sudah pernah diakses, sehingga bila akan mengakses objek yang sama di internet, tidak perlu mengambil dari internet, tetapi cukup dari proxy karena sudah disimpan. Dengan adanya *caching* ini, *bandwidth* yang dipakai akan lebih hemat, dan dapat mempercepat akses ke web site. Begitu juga bila web server yang dituju ternyata mati atau mengalami gangguan, client tetap bisa mengaksesnya.

Proses *caching* ini juga tidak terlihat bagi pengguna (transparan), karena bagi pengguna tidak tampak siapa sebenarnya yang memberikan obyek yang dimintanya, apakah dari proxy server yang mengambil dari cache-nya atau server asli di Internet. Dari sisi pengguna, semua akan nampak sebagai balasan langsung dari Internet.

2.3 *Web Filtering*



Gambar 2.3 Diagram *Web Filtering*

Proxy server yang dikonfigurasi secara benar, akan meningkatkan performa dan sekuritinya. Karena proxy bekerja pada layer aplikasi (*OSI Layer*), maka

content URL filtering yang dilakukan oleh proxy lebih cerdas daripada *packet filtering* pada firewall .Proxy web server dapat mengecek URL permintaan akses keluar (*outgoing request*) untuk halaman web dengan memeriksa pesan HTTP, GET dan POST. Dengan kemampuan ini, administrator dapat melarang atau mengijinkan akses ke domain tertentu. Firewall bisaa tidak dapat melihat nama domain di dalam pesan tersebut, karena firewall hanya memeriksa *header* dari paket data.

2.4 *Domain Filtering*

Domain filtering merupakan penyaringan informasi yang menggunakan daftar domain yang berbahaya dalam *list*-nya. Namun, *domain filtering* memiliki kelemahan yang patut dijadikan pertimbangan untuk mengadakan penelitian lebih lanjut terhadap optimalisasi *filtering* dalam sebuah jaringan komputer yang terhubung dengan internet yaitu:

a. *Domain filtering* membutuhkan sumber daya yang besar untuk melakukan pengecekan pada setiap domain.
b. Perubahan terhadap nama domain di dunia maya yang tidak dapat diperkirakan atau di prediksi kapan akan terjadi.
c. Nama domain yang pada awalnya berisi konten yang dianggap tidak berbahaya, bisa berubah menjadi berbahaya kapan saja.
d. Berdasarkan penelitian yang dilakukan oleh mahasiswa Politeknik Elektronika Negeri Surabaya dengan menggunakan teknik pendeteksian gambar porno, filtering dilakukan pada data yang melewati sebuah jaringan komputer yang terhubung ke internet. Sebuah server yang digunakan untuk memproses data yang ada di dalam jaringan, tingkat pendeteksian gambar porno mencapai 60 %. Sedangkan tingkat kesalahan (*false*) hanya 10 %. Proses pendeteksian ini memerlukan kemampuan komputansi yang tinggi. Gambar yang berukuran 200.000 *pixel*, memerlukan waktu paling sedikit 3 detik untuk dapat diproses. (Utama,A.C.2010)

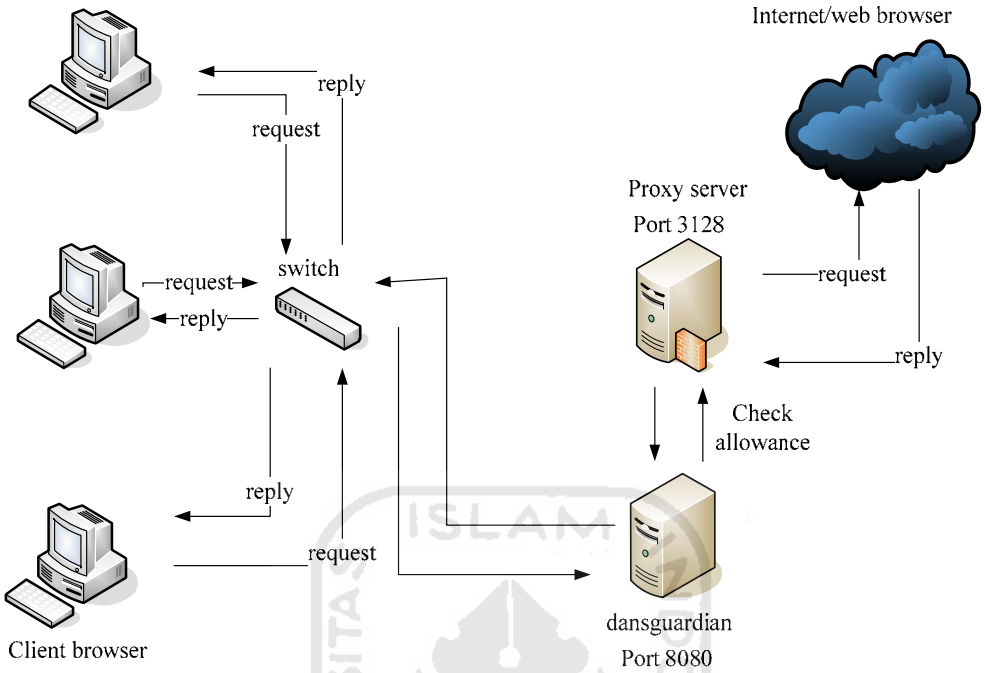
2.5 Dansguardian

Internet tidak jarang memberi dampak yang buruk bagi masyarakat jika disalahgunakan. Bahkan saat ini, anak usia dini yang dengan leluasa dapat mengakses situs yang berbau negatif dengan bebas, tentunya hal ini dapat merusak moral apabila content-content negatif tersebut tidak segera di filter. Setiap penyedia layanan internet, tentunya mempunyai kewajiban moral untuk menutup akses ke situs-situs negatif tersebut.

Saat ini sudah banyak cara yang bisa dilakukan untuk melakukan pemblokiran terhadap situs-situs negatif, baik salah satunya menggunakan openDNS atau menggunakan DNS Nawala. Bisa juga dengan memanfaatkan fitur optional yang ada pada Tinyproxy. Saat ini Dansguardian merupakan aplikasi yang banyak digunakan untuk memfilter akses internet .

Dansguardian merupakan sebuah *software* yang di rancang untuk mengontrol *user* saat mengakses sebuah website dari internet, termasuk filtering dari virus dan terdapat fitur monitoring. Dansguardian hanya dapat digunakan pada sistem operasi Unix atau GNU Linux seperti komputer server dan melindungi banyak komputer dengan sistem operasi yang berbeda-beda. Dansguardian sering digunakan oleh sekolah, bisnis dan ISP sebagai penyedia layanan internet.

DansGuardian merupakan pemenang penghargaan *Open Source web content filter* yang saat ini berjalan pada *Linux, FreeBSD, OpenBSD, NetBSD, Mac OS X, HP-UX*, dan Solaris. Ini merupakan *Content Filtering* yang didasarkan pada banyak metode yang di dalamnya, termasuk *phrase matching, PICS filtering* dan *URL filtering.* Ini bukan merupakan filtering murni berdasarkan daftar situs yang dilarang untuk di akses. DansGuardian dirancang agar bisa lebih fleksibel dan memungkinkan untuk menyesuaikan filtering sesuai dengan kebutuhan.



Gambar 2.4 Diagram Dansguardian

Dansguardian merupakan aplikasi yang berjalan berjalan pada sisi server. Sebagai gateway, server yang akan dilewati oleh paket-paket data yang mengantri saat di akses oleh klien. Pada server yang terdapat Dansguardian, paket-paket tersebut akan mengalami ‘pemeriksaan’ oleh Dansguardian terlebih dahulu sebelum sampai ke client.

Pemeriksaan tersebut berpotensi menghambat kelancaran paket-paket data untuk segera sampai kepada klien yang mengaksesnya karena harus mengantri untuk ‘di periksa’ terlebih dahulu oleh Dansguardian *(bottle neck)*. Untuk itu, diperlukan pengaturan yang tepat oleh seorang *Network Administrator* agar Dansguardian dapat berjalan dengan baik dalam sebuah jaringan komputer.

Pengaturan yang tepat, dapat dilakukan jika seorang *Network Administrator* mengetahui apa saja yang akan terjadi jika mengaplikasikan Dansguardian dalam servernya. *Resource* yang tepat untuk server yang terdapat Dansguardian, delay yang mungkin terjadi, antrian (*queue*) yang bisa jadi akan menupuk di server

karena pemeriksaan yang di lakukan oleh Dansguardian patut untuk menjadi pertimbangan.

2.5.1 Cara Kerja Dansguardian

DansGuardian berdiridi antara web browser di sisi client dan proxy server, sehingga menghalangi serta memodifikasi komunikasi antar keduanya. Tinyproxy bekerja pada port 8888 dan DansGuardian bekerja pada port 8080.Saat terdapat *reques t*dari client pada port 8080 Dansguardian, *request* tersebut akan diteruskan melewati Tinyproxy pada port 8888.

Pengalihan *request* data agar langsung melewati Dansguardian dilakukan dengan mengarahkan browser pada port 8080, atau dengan menggunakan *transparent proxy* dalam Tinyproxy dan mengarahkan *outgoing traffic* pada firewall pada port 80 ke port 8080 di *localhost* (jika pada DansGuardian server terdapat pada *firewall*).

Setelah client mengirimkan *request* pada DansGuardian. DansGuardian akan memeriksa *header* dari *request* seperti *username, source IP* (sumber IP), URL, dan status POST. Filter yang tepat diterapkan (*banned user, exception user, banned URL, exception URL, banned IP, exception IP*). Jika ada ukuran POST untuk *upload*, juga akan diperiksa. Jika semua baik, *request* tersebut akan diteruskan ke Tinyproxy yang kemudian mengambil file dari Internet.

Tinyproxy hanya melewatkan *header* dari *request* kembali ke DansGuardian yang kemudian memeriksanya berdasarkan *MIME-types*, *content disposition* (nama file). DansGuardian kemudian melakukan filter untuk melarang *MIME-types*, dan *file extensions*.

Tinyproxy melewatkan *document body* kembali ke DansGuardian yang kemudian di de-kompresi (jika berasal dari web server memiliki *gzip* atau *deflateplugin*) kemudian menghasilkan dua salinan file, yaitu HTML dan *whitespace removed* serta *original file*. O*riginal file* akan mencari PICS *labeling*, kemudian mencari *phrases*. Proses filtering dinyatakan selesai saat rating PICS dan *phrases* ditemukan dalam halaman web.

Jika pengecekan sudah selesai, maka *header* dan *body* akan kembali ke *client browser* sesuai dengan *request*nya dan hasil pengecekan Dansguardian. Pada setiap tahap pengecekan dalam Dansguardian, jika *page* atau *file* harus di blokir, proses pengembalian *request* dari internet / web server ke client, akan ditolak dan client akan mendapat pesan peringatan *Access Denied* dari Dansguardian.

BAB 3

METODOLOGI

3.1 Analisis Masalah

Data dari Direktorat Pembinaan Sekolah Menengah Kejuruan, Direktorat Jendral Manajemen Pendidikan Dasar Dan Menengah Kementrian Pendidikan NasionaL menunjukan bahwa lebih dari 40% aktifitas internet disalahgunakan. Dalam sebuah instansi yang memiliki fasilitas internet dengan akses yang bebas tanpa ada *filtering*, tidak menutup kemungkinan kebebasan akses tersebut digunakan untuk membuka situs-situs yang tidak berhubungan dengan pekerjaan. Salah satunya bisa jadi situs porno, *warez* dan *hacking*. (Gunawan,G.2009)

Jika terus dibiarkan, bukan tidak mungkin produktifitas pekerja akan menurun dan kebutuhan *bandwidth* akan meningkat. Akan berakibat kurang baik lagi apabila dari situs tersebut, menyebabkan komputer terinveksi *virus/ Trojan /spyware.*

Untuk mengatasi hal ini, tidak mudah menegur *user* satu persatu. Salah satu solusinya adalah menggunakan aplikasi *content filter*.

Dansguardian sebagai *web content filtering* yang digunakan, bekerja berdasarkan isi dari situs, bila isi dari suatu situs dinilai tidak boleh diakses maka situs tersebut akan diblok secara otomatis.

Pemasangan Dangsguardian dalam sebuah jaringan komputer, akan berpengaruh pada *speed* atau kecepatan transfer data dalam jaringan tersebut. Masalah yang akan dihadapi dalam penerapan Dansguardian adalah saat keberadaan Dansguardian menjadi penghambat (*bottle neck*) bagi sebuah jaringan komputer.

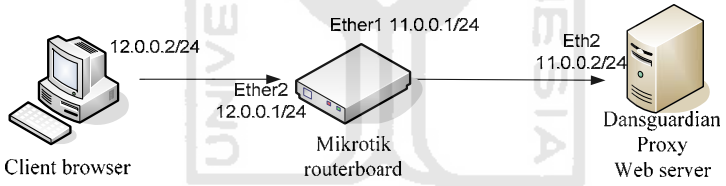
Penggunaan aplikasi Dansguardian sebagai *web content filtering* yang diletakkan di antara *gateway* dan *client*, sudah tentu dilewati oleh paket-paket data yang di akses oleh *client*. Dansguardian akanmelakukan “pemeriksaan” konten serta URL pada paket data yang melewatinya dengan sistem *queue*.

Dari permasalahan tersebut akan dibuat suatu penyelesaian, dan permasalahanya adalah bagaimana membuat sistem filtering yang maksimal pada *bandwidth* tertentu serta pengaruh Dansguardian terhadap *resource* server yang digunakan.

Dalam analisis ini, *bandwidth* yang digunakan berbeda-beda, agar dapat ditentukan berdasarkan fakta pada bandwidth tertentu, Dansguardian akan menjadi penghambat serta seberapa besar pengaruh Dansguardian terhadap *resource* yang digunakan.

3.2 Gambaran Umum Sistem

Implementasi mekanisme jaringan yang digunakan adalah jaringan lokal.



Gambar 3.1 Desain topologi jaringan komputer dengan Dansguardian

Terdapat 2 server dalam jaringan, yaitu mikrotik sebagai *gateway* yang bertugas sebagai penghubung client ke dansguardian server dan sebuah komputer sebagai Dansguardian dan web server. Pengujian dilakukan dengan mengakses web server yang berada di dalam server.

Mikrotik routerboard berfungsi sebagai *bandwidth management*. Mikrotik router inilah yang nanti akan memberikan aturan besarnya *bandwidth* yang digunakan. Pada mikrotik router, digunakan dua interface yaitu:

1. Ether1 *interface* yang terhubung dengan Dansguardian dan web server, dengan alamat IPv4 static.

2. Ether2 *interface* yang terhubung dengan jaringan lokal, dengan alamat IPv4 static.

Sedangkan *web content filtering* dengan Dansguardian terhubung langsung dengan jaringan lokal melalui mikrotik yang terdiri dari satu buah *client*. Komputer ini menggunakan satu *interface* yaitu:

1. Eth4 adalah *interface* yang terhubung dengan jaringan lokal melalui routerboard dengan alamat IPv4 static.

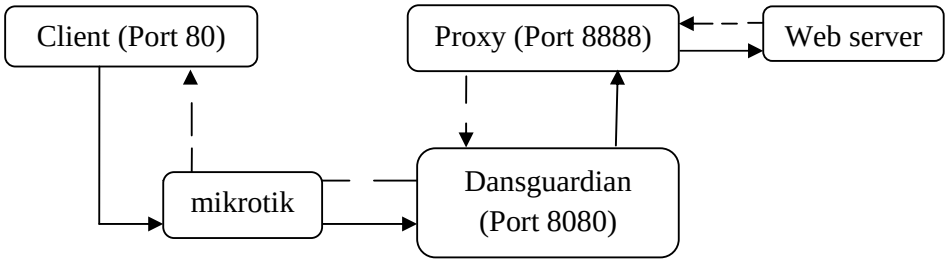
Tabel 3.1 Alokasi alamat IP

Nama Perangkat	Interface	Alamat IP
Dansgaurdian Server	eth2	11.0.0.2/24
Mikrotik Routerboard	Ether1	11.0.0.1/24
	Ether2	12.0.0.1/24
PC Client	local ethernet	12.0.0.2/24

Komputer client memiliki satu buah *interface* yang akan diberikan IPv4 *static* yang satu jaringan dengan *interface* Ether2 yang dimiliki Mikrotik *routerboard*. Hal ini bertujuan agar *client* mendapatkan koneksi melalui *routerboard* menuju Dansguardian dan web server. *Routerboard* bertugas untuk melakukan *routing* paket yang diminta oleh komputer *client*.

3.2.1 Desain Alur Data dan Proses

Gambar 3.2 merupakan gambar yang menjelaskan bagaimana alur data yang akan dilewati ketika sistem sudah terbentuk. Di dalam sistem tersebut terdapat mikrotik server yang berfungsi mengatur besarnya jumlah bandwidth (*bandwidth management*) yang ditentukan untuk analisis.



Gambar 3.2 Desain alur data

3.3 Analisis dan Persiapan Kebutuhan Sistem

3.3.1 Kebutuhan Perangkat Keras

Berdasarkan gambaran umum sistem, sudah jelas terlihat bahwa perangkat-perangkat keras yang dibutuhkan untuk menunjang implementasi dari semua fitur-fitur yang ada pada sistem adalah sebagai berikut:

1. *Routerboard* mikrotik sebagai server *gateway* langsung ke internet/web server. Routerboard ini merupakan sebuah router yang di dalamnya sudah terdapat mikrotik OS yang siap pakai. Difungsikan sebagai *gateway* dari jaringan lokal menuju Dansguardian server dan sebagai *banwidth management* untuk jaringan lokal. Routerboard yang digunakan adalah seri RB750 mikrotik yang memilki 5 port Ethernet.
2. Komputer sebagai Dansguardian dan proxy server

   Dalam komputer ini,terdapat Dansguardian dan proxy server, serta Dansguardian yang akan di analisis. Spesifikasi yang digunakan adalah :

   a. Prosesor 1.6 GHz

   b. Memori 512 MB

   c. harddisk dengan kapasitas 80 GB

   d. Network Interface Card 2 buah
3. Komputer sebagai *client*

   Komputer yang digunakan untuk pengujian.

   a. Prosesor 1.6 GHz

   b. Memori 2 GB

c. harddisk dengan kapasitas 160 GB

d. Network Interface Card 1 buah

3.3.2 Kebutuhan Perangkat Lunak

1. Mikrotik

   MikroTik RouterOS™ merupakan sistem operasi dan perangkat lunak yang dapat digunakan pada komputer PC menjadi sebuah komputer router *network* yang handal, mencakup berbagai fitur yang dibuat untuk ip *network* dan jaringan *wireless*. MikroTik didesain untuk memberikan kemudahan bagi penggunanya. Administrasinya bisa dilakukan melalui *Windows Application (WinBox)*. PC yang akan dijadikan router mikrotik pun tidak memerlukan resource yang cukup besar untuk penggunaan standar, misalnya hanya sebagai *gateway*. MikroTik RouterOS yang berbentuk software yang dapat di-*download* diwww.mikrotik.com merupakan jenis Mikrotik OS yang dapat diinstal pada komputer. Sedangkan *BUILT-IN Hardware* MikroTik merupakan jenis Mikrotik dalam bentuk perangkat keras yang khusus dikemasdalam *board router* yang didalamnya sudah terinstal MikroTik RouterOS.

2. Winbox

   Winbox merupakan sebuah aplikasi yang memungkinkan kita melakukan konfigurasi Mikrotik RouterOS menggunakan *graphic interface* yang cepat dan sederhana. Winbox merupakan aplikasi biner Win32 (Windows), Winbox berjalan pada sistem operasi Windows, akan tetapi dapat dijalankan pada sistem operasi Linux dan Mac OS dengan menggunakan program bantu Wine. Semua fungsi *interface* Winbox dibuat sedekat mungkin dengan fungsi Console, itu sebabnya tidak ada bagian Winbox di manual. Terdapat beberapa konfigurasi tingkat *advanced* dan konfigurasi sistem yang penting tidak dapat dilakukan dari winbox, seperti perubahan MAC *address* pada sebuah antarmuka atau mem-*factory default* Mikrotik.

3. Dansguardian

   Dansguardian merupakan aplikasi yang berjalan berjalan pada sisi server. Sebagai gateway, server yang akan dilewati oleh paket-paket data yang mengantri saat di akses oleh klien. Pada server yang terdapat Dansguardian, paket-paket tersebut akan mengalami 'pemeriksaan' oleh Dansguardian terlebih dahulu sebelum sampai ke client. DansGuardian dirancang agar bisa lebih fleksibel dan memungkinkan untuk menyesuaikan filtering sesuai dengan kebutuhan. Dansguardian melengkapi kerja proxy yang belum dapat mengimbangi perubahan-perubahan yang sangat cepat pada web di sunia *cyber*. Filtering yang dilakukan oleh dansguardian, bukan hanya pada URL, melainkan berdasarkan *content* yang terdapat di dalam sebuah web atau biasa disebut dengan *web content filtering.*

4. *Tinyproxy*

   Tinyproxy merupakan sebuah HTTP proxy server daemon untuk sistem operasi POSIX. Tinyproxy dirancang agar berjalan pada sistem seperti Unix. Dirilis di bawah GNU *General Public License,* Tinyproxy adalah *free application* dan telah dikembangkan untuk beberapa tahun. Saat ini sedang dipertahankan pada Banu sebagai proyek yang dapat diakses publik. Tinyproxy memerlukan sangat sedikit *system resources.* Dengan demikian, Tinyproxy dapat dijalankan pada mesin yang lebih tua, atau pada alat jaringan seperti router *broadband* berbasis Linux, tanpa dampak yang nyata pada kinerja.

5. *Webserver Stress Tool 7*

   Webserver Stress Tool merupakan aplikasi yang digunakan untuk pengujian tingkat stress pada suatu aplikasi web yang melewati media *http/https* pada waktu yang bersamaan. Pengujian ini berlangsung masing-masing pada setiap *user* yang diatur untuk mengakses web tersebut mulai dari *loading* gambar bingkai dan sebagainya. Pada pengujiannya dapat mengatur jenis profil dari *user* pengakses web. 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* :

a. *Performance Tests*

   Test ini merupakan permintann URL tunggal dari server web atau aplikasi web untuk mengidentifikasi dan menemukan elemen yang mungkin bertanggung jawab atas kinerja lebih lambat dari yang diharapkan. Tes ini memberikan kesempatan unik untuk mengoptimalkan pengaturan konfigurasi server atau aplikasi dengan berbagai implementasi pengujian halaman web tunggal/script untuk mengidentifikasi kode tercepat atau pengaturan.

b. *Load Tests*

   Untuk tes ini,cukup memasukan URL, jumlah pengguna, dan waktu antara (*delay*) klik lalu lintas situs web .

c. *Stress Tests*

   Tes ini merupakan simulasi dari “kekerasan” serangan yang berlaku dengan tujuan memberikan beban berlebihan ke server web.Jenis “kekuatan fisik” ini dapat disebabkan oleh lonjakan besar dalam kegiatan pengguna (misalnya dengan penambahan iklan baru). Ini adalah tes untuk menemukan ambang batas lalu lintas untuk sebuah server web.

d. *Ramp Tests*

   Tes ini menggunakan peningkatan jumlah pengguna selama jangka waktu yang diberikan untuk menentukan jumlah maksimum pengguna saat web server dapat mengakomodasi sebelum menghasilkan pesan error.

e. Webserver Stress Tool memiliki tes lainnya yang bisa memberikan wawasan lebih lanjut tentang situs web, misalnya untuk menentukan halaman web yang dapat diminta secara bersamaan tanpa masalah seperti database,semaphore, dan lainnya. Dengan simulasi yang dihasilkan oleh permintaan HTTP ratusan atau bahkan ribuan pengguna secara simultan, aplikasi ini dapat menguji kinerja sebuah

web server di bawah beban normal dan berlebihan untuk memastikan bahwa informasi penting dan layanan yang tersedia pada *end-users*sesuai dengan yangdiharapkan. Rincian detail *test-log* dan beberapa grafik yang mudah dibaca membuat hasil analisis lebih cepat di dapat. *Webserver Stress Tool* untuk Windows (2000/XP/2003/Vista/7/2008) dapat mengirim hampir semua server HTTP (halaman statis misalnya, JSP / ASP, atau CGIS) untuk *performance*, beban (*load*), dan *stress-tests*. (Hendrawan,D.2010)

6. *Wine*

   Wine merupakan sebuah perangkat lunak pada sistem operasi GNU/Linux, yang dapat digunakan untuk menjalankan aplikasi berbasis sistem operasi *Microsoft Windows* secara langsung (*native*) pada sistem operasi GNU/Linux. *Wine* bekerja pada level *compability layer*, dengan menggunakan *daemon / service*/ layanan bernama *wineserver*, dimana *wineserver* ini akan menggantikan fungsi dasar *windows* (bertindak seolah-olah sebagai windows asli) untuk melayani aplikasi-aplikasi yang membutuhkan akses ke *registry windows* dan akses ke pustaka-pustaka *windows*, serta mengintegrasikan aplikasi yang dijalankan ke X Windows sistem milik GNU/Linux. Meskipun *wine* dapat bertindak selayaknya *windows* dengan menerapkan/meniru beberapa fungsi dasar *windows*, sampai sekarang *wine* belum bisa mengemulasikan *driver windows* asli secara langsung. Berbeda dengan *virtual machine* yang mensimulasikan seluruh sistem operasi dalam sebuah lingkungan perangkat keras virtual, aplikasi *windows* yang dijalankan dengan *wine* dianggap sama dengan aplikasi GNU/Linux lainnya, sehingga tidak membutuhkan memori sebanyak *virtual machine* yang berdampak pada eksekusi program di dalam *wine* menjadi lebih cepat. Dengan menggunakan *wine* jika ada aplikasi windows yang mengalami kerusakan atau *crash* penanganannya akan lebih mudah.

7. Web Server

   Setiap kali sebuah web browser berhubungan dengan suatu situs web, sebetulnya ia terhubung dengan sebuah web server. Server tersebut mendengarkan *request* pada jaringan dan memberikan jawaban berupa data tertentu kepada *client* atau pengirim permintaan.Web server atau HTTP adalah sebuah program yang melayani koneksi HTTP (*Hyper Text Transfer Protokol*). Web server bekerja berdasarkan request-response, yaitu ketika HTTP *client* (misalnya web browser) membangun koneksi dengan server dengan mengirimkan *request* (permintaan) kepada server, maka server akan merespon dengan mengolah permintaan tersebut kemudian mengirimkan data sesuai yang diminta oleh HTTP client. Format data yang dikirimkan oleh HTTP server mengunakan format SGML (*Standard General Markup Languange*), data ini akan ditampilkan oleh HTTP client (*browser*) sesuai dengan fasilitas yang dimilikinya. Misal data yang dikirimkan berupa data gambar, sedangkan browser yang dipakai hanya mampu menampilkan data berupa teks (misal, *lynx* dan *links*) maka browser tidak akan mampu menampilkan data tersebut secara sempurna, atau hanya akan ditampilkan alternatifnya saja. Koneksi ke web server dilakukan melalui protokol yang dikenal dengan HTTP (*Hypertext Transfer Protokol*) yang secara default menggunakan port 80 (nilai port dapat diubah). HTTP merupakan protokol yang bekerja pada lapisan aplikasi (*application layer*) dan secara sederhana dapat didefinisikan sebagai sekumpulan aturan untuk tukar menukar data pada *world wide web* (www). Ide dasar dari HTTP adalah bagaimana sebuah file dapat berisi suatu referensi pada file yang lain dengan sebuah transfer permintaan file yang bersangkutan.

3.3.3 Instalasi dan Konfigurasi Sistem

Semua kebutuhan perangkat lunak di-*install* pada mesin dengan metode yang berbeda-beda. Secara umum paket *installer* adalah berbentuk *binary* dan *source*. Pemilihan dua jenis paket tersebut akan didasarkan pada kebutuhan

sistem. Perangkat-perangkat lunak akan disertakan pada dua buah mesin yang berbeda, yaitu Routerboard dan Server. Berikut adalah detil pembagian perangkat-perangkat lunak pada masing-masing mesin serta instalasi dan konfigurasi secara umum:

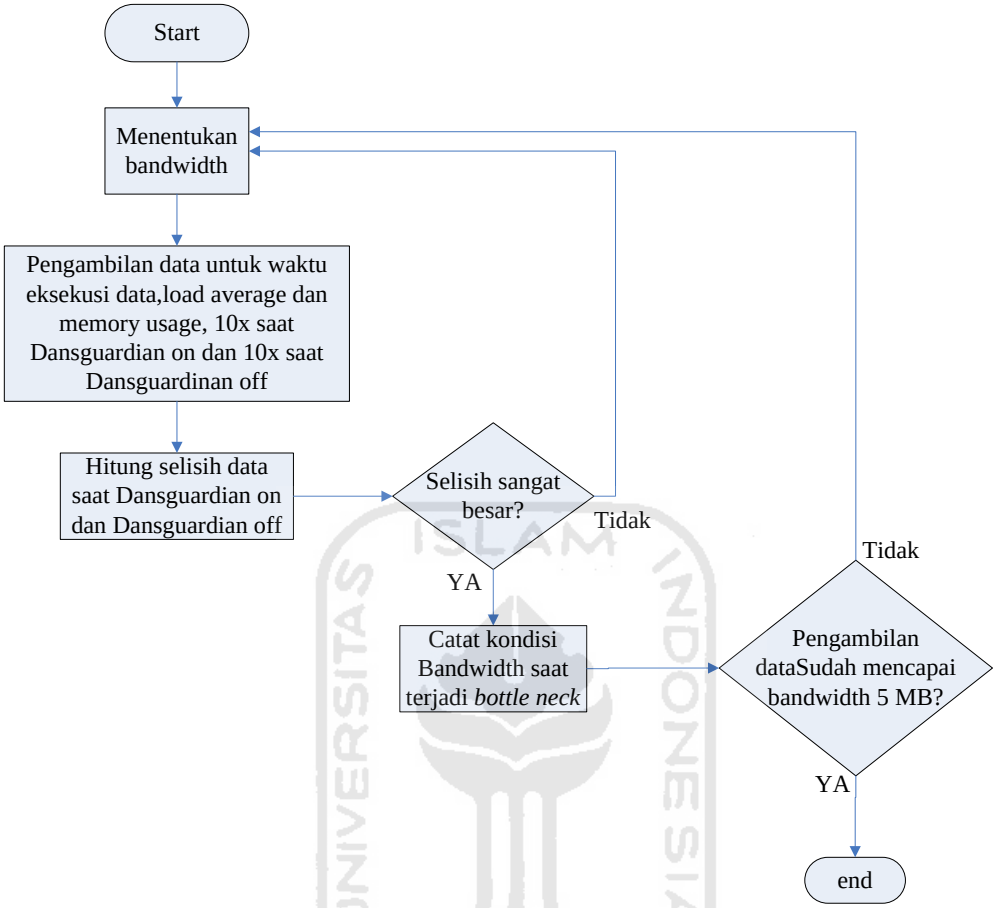
1. Mikrotik *Routerboard*
   a. Mikrotik *routerboard* merupakan sebuah *hardware* yang berfungsi sebagai *router* dan di dalamnya sudah terdapat Mikrotik OS yang siap digunakan.
   b. Konfigurasi yang dilakukan dengan mengatur IP static pada masing-masing NIC. Untuk *routing* dan *management bandwidth* dilakukan melalui *winbox application*.
2. Winbox
   a. Jenis paket yang digunakan *Winbox application* adalah *binary* dengan ekstensi .exe dan berjalan dengan bantuan *wine application.*
   b. Konfigurasi yang dilakukan terkait dengan *routing* dan *bandwidth management* pada mesin mikrotik yang sudah ter*install.*
3. Web Server
   a. Konfigurasi yang dilakukan adalah membuat 2 halaman HTML yang terdiri dari halaman kosong dan halaman yang berisi penuh dengan tulisan atau content.
4. Dansguardian
   a. Jenis paket yang digunakan adalah tipe *binary* yang diinstall dengan menggunakan perintah yum.
   b. Konfigurasi yang dilakukan adalah mengatur daftar domain maupun URL yang boleh dan tidak boleh di akses oleh *client.*
5. Tinyproxy
   a. Jenis paket yang digunakan adalah tipe *binary* yang diinstal dengan menggunakan paket *source* yang bisa diunduh di Internet. Dalam instalasi paket *source* akan dibutuhkan beberapa cara khusus, yaitu dengan menggunakan perintah `rpm –ivh` (nama paket)
   b. Konfigurasi yang dilakukan adalah terkait dengan

1. *Port* yang digunakan untuk berhubungan dengan client
2. *user* dan *group* pengelola aplikasi
3. penentuan ukuran dan lokasi *cache*
4. metode penyimpanan *cache*
5. ukuran maksimal dan minimal objek yang akan disimpan pada *cache*
6. Penentuan ukuran *memory* yang digunakan oleh aplikasi (Bayuputra, B.2011)

6. *Webserver Stress Tool*
   a. Jenis paket yang digunakan *Webserver Stress Tool application* adalah *binary* dengan ekstensi .exe dan berjalan pada sistem operasi Windows
   b. Konfigurasi yang dilakukan terkait denganjumlah *user, delay, url.*
7. *Wine*
   a. Jenis paket yang digunakan adalah tipe *binary* yang diinstall dengan menggunakan perintah `yum install`.
   b. Konfigurasi yang dilakukan adalah terkait dengan aplikasi yang berbasis *windows,* dalam hal ini winbox, yang memiliki ekstensi .exe dan dijalankan pada sistem operasi berbasis linux.

3.3.4 Metode Analisis

Data hasil analisa yang diperoleh dengan menggunakan *user simulation* dengan *Webserver Stress Tool 7.2.2.261* untuk melakukan request 1000 user secara langsung menuju server. Setelah itu, dilakukan perbandingan pengaruh Dansguardian dalam jaringan komputer dalam hal kecepatan pelayanan data, *memory usage* dan *load average* dengan bandwidth yang berbeda beda.



Gambar 3.3 Alur Metode Pengambilan Data

BAB 4

HASIL DAN PEMBAHASAN

4.1 Implementasi Secara Umum

Implementasi sistem merupakan tahap dimana sistem mampu diaplikasikan dalam keadaan yang sesungguhnya. Dari implementasi ini akan diketahui apakah sistem yang dibuat dapat berjalan dengan baik atau tidak. Serta apakah sistem menghasilkan *output* yang sesuai dengan perancangan yang telah dibuat.

4.2 Tahapan Implementasi Perangkat Lunak

Dalam pengaplikasian perangkat lunak ini, terdapat beberapa tahap yang terdiri dari :

a. Installasi *wine*

   Pada tahap ini dilakukan installasi *wine application* yang digunakan untuk menjalankan aplikasi *winbox.exe* untuk mempermudah konfigurasi pada mikrotik server.

b. Konfigurasi mikrotik routerboard

   Pada tahap ini dilakukan konfigurasi server mikrotik dengan menggunakan *hardware* mikrotik *routerboard* jenis RB750.

c. Konfigurasi web server

   Pada tahap ini dilakukan konfigurasi web server versi httpd-2.2.15-5.el6.centos.i686 yang digunakan untuk membuat halaman web yang akan di analisa.

d. Instalasi dan konfigurasi Tinyproxy

   Pada tahap ini dilakukan installasi dan konfigurasi Tinyproxy proxy versi 1.8.2-1.el6.i686yang digunakan untuk mengarahkan client agar melewati Dansguardian server saat mengakses data.

e. Instalasi dan konfigurasi Dansguardian

Pada tahap ini dilakukan installasi dan konfigurasi Dansguardian-2.10.1.1-1.el5.ef.i386 yang digunakan sebagai *web content filtering* agar semua situs yang di akses oleh client, dapat dilindungi dari hal-hal yang tidak pantas untuk di akses.

f. Konfigurasi IP Address CentOS

Pada tahap ini dilakukan konfigurasi IP Address static pada CentOS agar server dapat terkoneksi menuju jaringan lokal sesuai dengan *interface* yang ada.

g. Konfigurasi *iptables*

Pada tahap ini dilakukan konfigurasi terhadap `iptables` agar client melakukan *redirect* beberapa port agar proxy dan Dansguardian bisa terkoneksi.

h. *Webserver Stress Tool*

Pada tahap ini dilakukan installasi *Webserver Stress Tool* yang akan digunakan untuk memonitoring *traffic* dan melakukan analisa yang terjadi pada jaringan komputer saat client mengakses data melalui server yang ada.

4.3 Implementasi Hasil Perancangan

4.3.1 Installasi *wine application*

1. Dalam tahap ini dibutuhkan koneksi internet agar installasi dapat dilakukan dengan menggunakan perintah `yum.`
2. Lakukan installasi repository melalui terminal sebagai berikut:

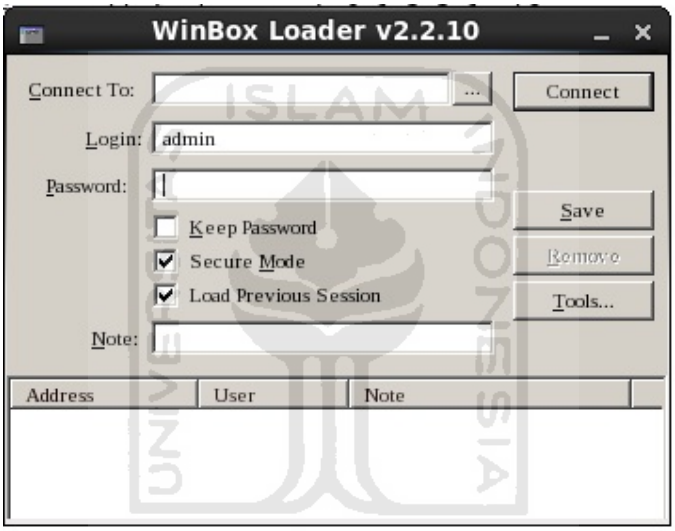
```
[root@localhost~]# rpm –Uvh http://download.fedora.redhat.com/pub
/epel/6/i386/epel/release-6-5.noarch.rpm
```

3. Lakukan installasi paket melalui terminal dengan perintah `yum install wine.`

```
[root@localhost ~]#yum install wine
```

4. Untuk menjalankan aplikasi winbox.exe yang terdapat pada Desktop CentOS melalui *wine application,* terlebih dahulu pindah ke direktori Desktop pada `/home/diqa/Desktop/`

Setelah pindah direktori, maka winbox.exe akan berjalan pada CentOS melalui aplikasi wine.



Gambar 4.1 *Winbox.exe* pada CentOS

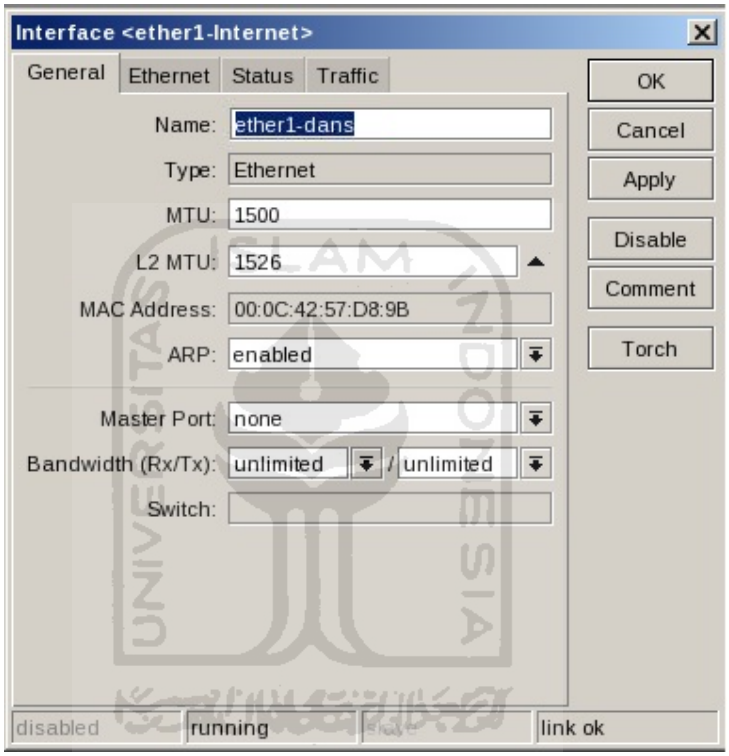
4.3.2 Konfigurasi mikrotik *Routerboard* RB750

1. Dalam tahap ini dibutuhkan mikrotik OS yang sudah tersedia pada Routerboard RB750 dan aplikasi winbox untuk melakukan konfigurasi secara GUI.
2. Untuk menjalankan aplikasi *winbox.exe* pada CentOS, menggunakan perintah `wine winbox` pada terminal.
3. Lakukan konfigurasipada bagian:
   a. *Identity* : mendefinisikan nama server mikrotik. Isikan dengan nama sesuai dengan yang di inginkan.

b. *Interface* : mendefinisikan nama *interface* yang digunakan. Pada bagian *interface list* lakukan konfigurasi pada bagian dibawah ini :

Ether1

Untuk koneksi yang digunakan menuju dansguardian. Lakukan konfigurasi sebagai berikut :



Gambar 4.2 Konfigurasi *interface* ether1

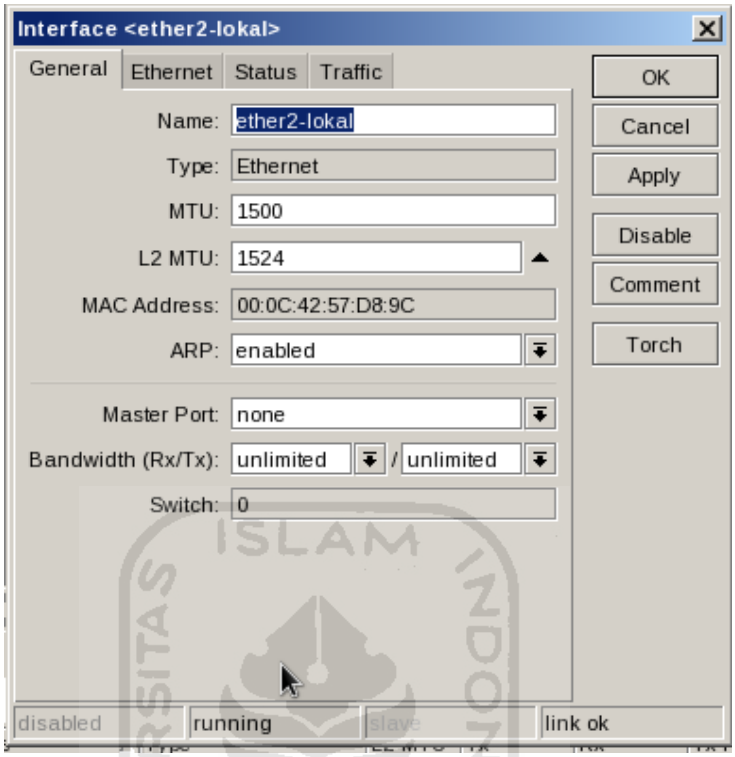
Keterangan

Name : ether1-dans

ARP : *enabled*

Ether2

Untuk koneksi yang digunakan menuju lokal. Lakukan konfigurasi sebagai berikut :



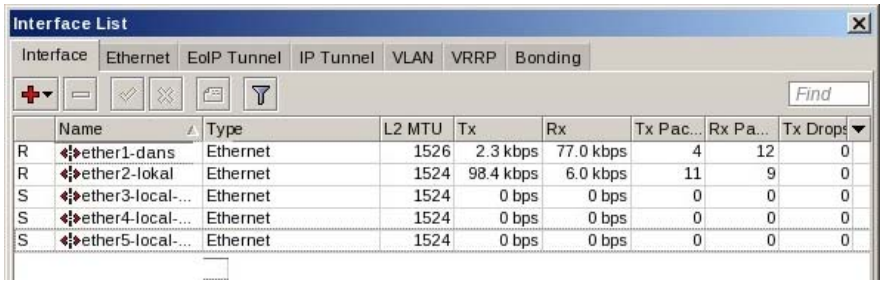
Gambar 4.3 Konfigurasi *interface* ether2

Keterangan

Name : ether2-lokal

ARP : *enabled*

Maka, hasil konfigurasi *interface* ether1 dan ether 2 seperti gambar di bawah ini :

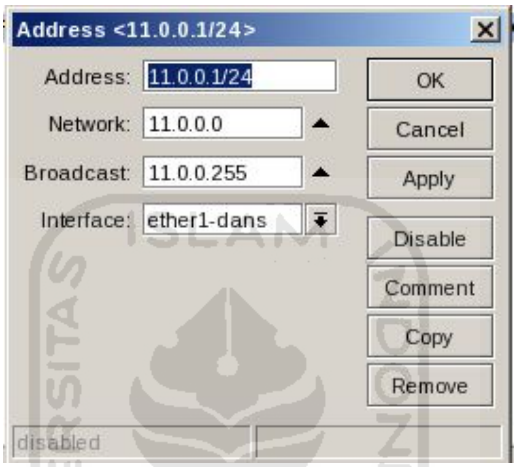


Gambar 4.4 *Interface List*

c. *IP Address*: mendefinisikan alamat *interface* yang digunakan untuk melakukan koneksi. Pada bagian *address list* lakukan konfigurasi pada bagian dibawah ini :

***Address* 1**

Untuk koneksi yang digunakan menuju dansguardian. Lakukan konfigurasi sebagai berikut :



Gambar 4.5 Konfigurasi *Address* Dansguardian

Keterangan :

Address : 11.0.0.1/24

Network : 11.0.0.0

Broadcast : 11.0.0.255

Interface : ether1-dans

***Address* 2**

Untuk koneksi yang digunakan menuju lokal. Lakukan konfigurasi sebagai berikut :

Gambar 4.6 Konfigurasi *Address* lokal

Keterangan :

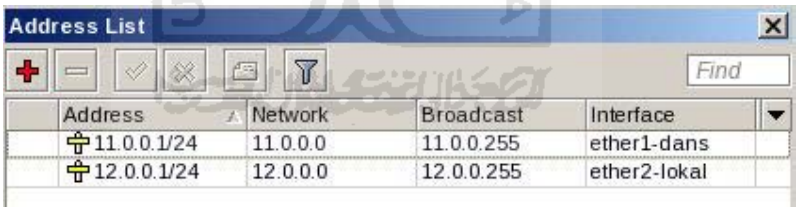
Address : 12.0.0.1/24

Network : 12.0.0.0

Broadcast : 12.0.0.255

Interface : ether2-lokal

Maka, hasil konfigurasi *address* seperti gambar di bawah ini :

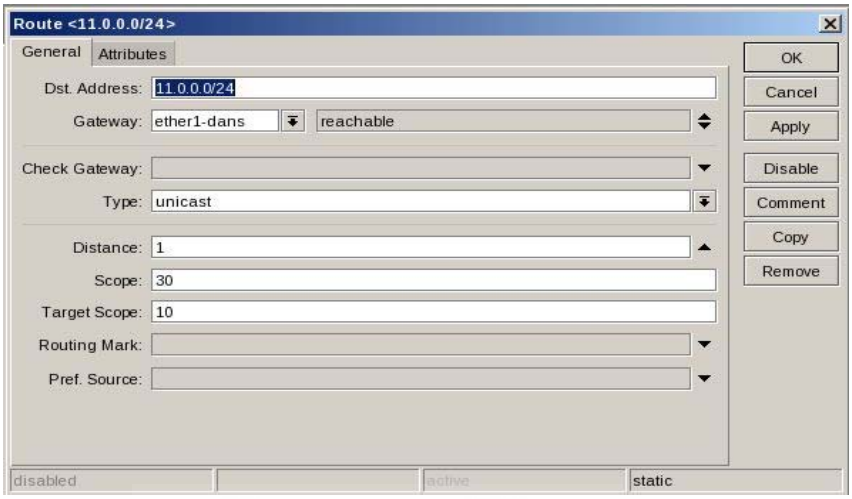


Gambar 4.7 *Address List*

d. *IP Routes*: mendefinisikan *routing* pada mikrotik agar dapat terkoneksi menuju dansguardian dan lokal. Pada bagian *route list* lakukan konfigurasi pada bagian dibawah ini :

Route 1

Untuk koneksi yang digunakan menuju Dansguardian. Lakukan konfigurasi sebagai berikut :



Gambar 4.8 Konfigurasi *route* menuju Dansguardian

Keterangan :

Dst.Address : 11.0.0.0/24

Gateway : ether1-dans

Maka, hasil konfigurasi *address* seperti gambar di bawah ini :

Gambar 4.9 *Route List*

Lakukan tes koneksi ke client (12.0.0.2) melalui terminal pada winbox. Klik menu *New Terminal* pada winbox, kemudian ketikkan perintah `ping 12.0.0.2.` Apabila `ping` mendapat *reply* seperti gambar dibawah ini, mikrotik sudah terhubung ke client dan sudah siap digunakan.

```
[admin@MikroTik] > ping 12.0.0.2
12.0.0.2 64 byte ping: ttl=128 time=6 ms
12.0.0.2 64 byte ping: ttl=128 time=2 ms
12.0.0.2 64 byte ping: ttl=128 time=126 ms
12.0.0.2 64 byte ping: ttl=128 time=10 ms
12.0.0.2 64 byte ping: ttl=128 time=10 ms
12.0.0.2 64 byte ping: ttl=128 time=9 ms
12.0.0.2 64 byte ping: ttl=128 time=10 ms
12.0.0.2 64 byte ping: ttl=128 time=10 ms
12.0.0.2 64 byte ping: ttl=128 time=10 ms
12.0.0.2 64 byte ping: ttl=128 time=10 ms
12.0.0.2 64 byte ping: ttl=128 time=10 ms
12.0.0.2 64 byte ping: ttl=128 time=11 ms
12.0.0.2 64 byte ping: ttl=128 time=13 ms
12.0.0.2 64 byte ping: ttl=128 time=10 ms
12.0.0.2 64 byte ping: ttl=128 time=10 ms
12.0.0.2 64 byte ping: ttl=128 time=10 ms
12.0.0.2 64 byte ping: ttl=128 time=10 ms
12.0.0.2 64 byte ping: ttl=128 time=11 ms
12.0.0.2 64 byte ping: ttl=128 time=11 ms
12.0.0.2 64 byte ping: ttl=128 time=9 ms
12.0.0.2 64 byte ping: ttl=128 time=9 ms
```

Gambar 4.10 *Ping test*

4.3.3 Konfigurasi Web Server

1. Dalam tahap ini dilakukan konfigurasi web server menggunakan httpd-2.2.15-5.el6.centos.i686 dengan membuka `file /etc/httpd/conf/httpd.conf`

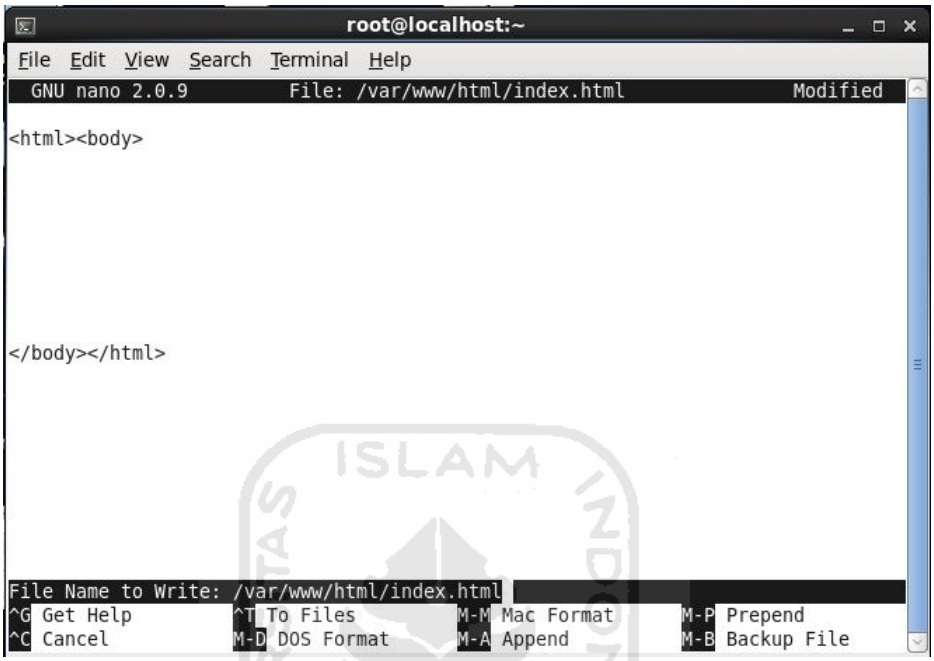
   Pastikan terdapat baris `/var/www/html` yang menunjukan bahwa dokumen web akan diletakkan pada direktori /var/www/html.

```
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/var/www/html"
```

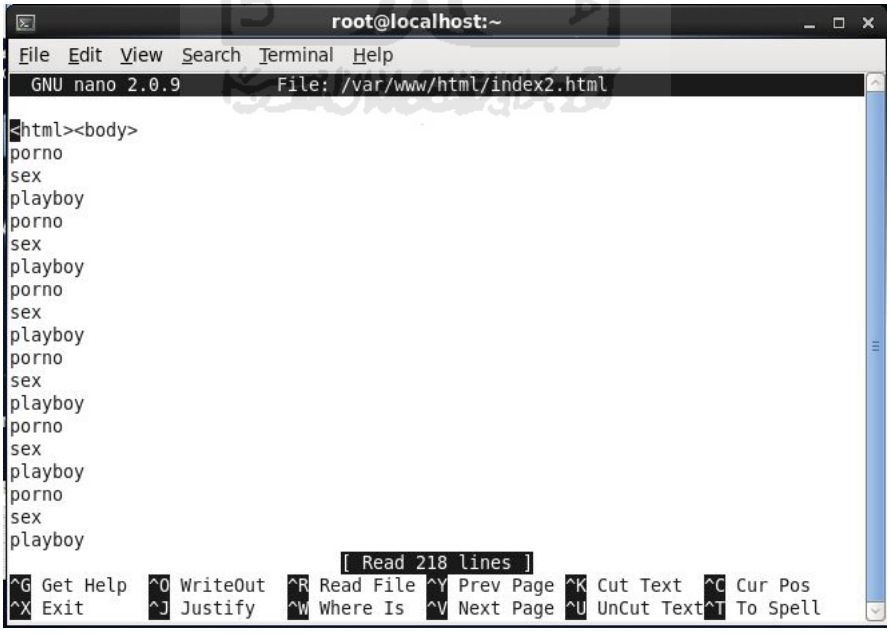
   Gambar 4.11 File`/etc/httpd/conf/httpd.conf`

2. Buatlah sebuah dokumen index.html dan index2.html pada direktori /var/www/html.

Isikan masing-masing file seperti gambar dibawah ini.



Gambar 4.12 File `/var/www/html/index.html`



Gambar 4.13 File `/var/www/html/index2.html`

3. Jalankan daemon httpd menggunakan perintah `service httpd start`.

```
[root@localhost Desktop]# service httpd start
```

4.3.4 Konfigurasi proxy server

1. Dalam tahap ini dilakukan installasi dan konfigurasi proxy menggunakan tinyproxy-1.8.2-1.el6.i686 yang dapat di install secara online dengan perintah `yum install tinyproxy`.

```
[root@localhost Desktop]# yum install tinyproxy
```

2. Kemudian lakukan konfigurasi proxy dengan menambahkan *access list* pada file `/etc/tinyproxy/tinyproxy.conf`.

```
[root@localhost Desktop]# nano /etc/tinyproxy/tinyproxy.conf
```

Konfigurasi file`/etc/tinyproxy/tinyproxy.conf`:

```
1#User tinyproxy
2#Group tinyproxy
3#Port 8888
4#Timeout 600
5#StatFile "/usr/share/tinyproxy/stats.html"
5#LogFile "/var/log/tinyproxy/tinyproxy.log"
6#PidFile "/var/run/tinyproxy/tinyproxy.pid"
7#MaxClients 100
8#MinSpareServers 5
9#MaxSpareServers 20
10#StartServers 10
11#Allow 127.0.0.1
12#Allow 12.0.0.2
```

Konfigurasi di atas merupakan konfigurasi default dari sebagian isi file `tinyproxy.conf`. Bagian yang harus diperhatikan adalah `Port 8888` yang menandakan port proxy yang aktif / digunakan oleh tiny proxy secara *default*. Sedangkan `Allow 12.0.0.2` merupakan *access list* yang diberikan kepada *IP Address client* untuk mengakses internet melalui proxy server.

3. Setelah selesai melakukan konfigurasi, jalankan *service* tinyproxy dengan perintah `service tinyproxy start.`

```
[root@localhost Desktop]#service tinyproxy start
```

4.3.5 Konfigurasi Dansguardian server

1. Dalam tahap ini dilakukan installasi dan konfigurasi Dansguardian menggunakan Dansguardian-2.10.1.1-1.el5.ef.i386 yang dapat di install secara online dengan perintah `rpm –Uvh.`

```
[root@localhost~]#rpm –Uvh http://dag.wieers.com/rpm/packages/
dansguardian/dansguardian-2.10.1.1-1.el5.ef.i386.rpm
```

2. Kemudian lakukan beberapa konfigurasi yang terdapat pada file `/etc/dansguardian/dansguardian.conf`

```
[root@localhost~]#gedit/etc/dansguardian/dansguardian.conf
```

Lakukan konfigurasi pada bagian filterip, fileport, proxyip, dan proxyport sebagai berikut:

```
1#filterip = 11.0.0.2
2#filterport = 8080
3#proxyip = 127.0.0.1
4#proxyport = 8888
```

Keterangan :

Baris 1 :IP server dimana dansguardian di install

Baris 2 : Listen Port Dansguardian

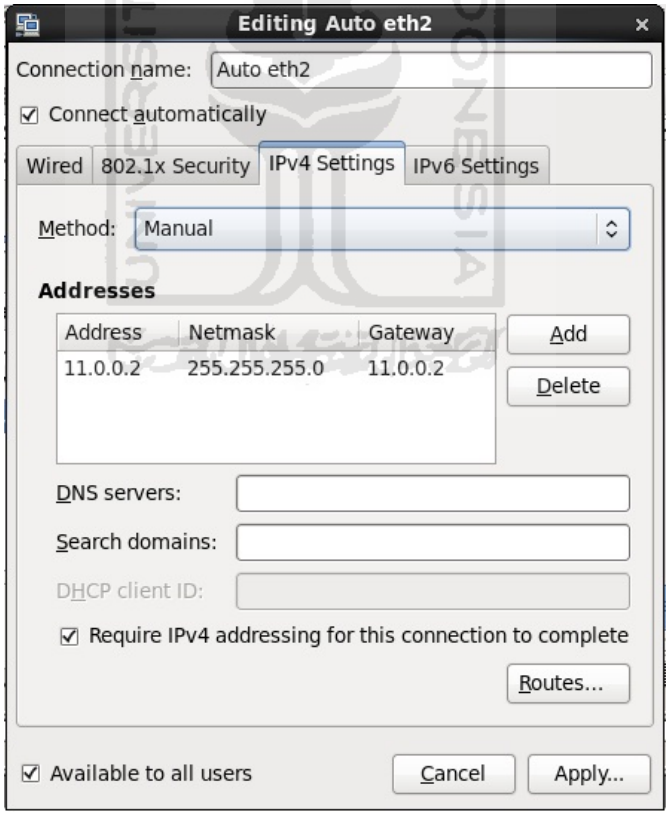
Baris 3 : *Default* IP *address*proxy Dansguardian

Baris 4 : Port Proxy

3. Setelah selesai melakukan konfigurasi, jalankan *service* tinyproxy dengan perintah `service dansguardian start.`

4.3.5 Konfigurasi IP Address pada CentOS

1. Dalam tahap ini dilakukan konfigurasi IP static pada CentOS pada interface eth2.
2. Klik menu *system* pada taskbar, pilih *preferences* kemudian pilih *network connections.*
3. Pada menu *network connections* terdapat *list interface* yang terdapat di komputer. Karena menggunakan eth2, klik pada eth2, pilih tab *IPv4 setting*. Pada bagian *method* pilih manual, lalu klik Add untuk menambahkan alamat IP sebagai berikut :



Gambar 4.14 Konfigurasi *IP Address* eth2

Jika sudah selesai melakukan konfigurasi, klik *Apply* dan masukan *password* root untuk autentifikasi oleh sistem.

4. Tutup menu *network connections* setelah selesai. *Restart network* untuk memastikan alamat *network* sudah berubah menjadi seperti konfigurasi yang baru saja dilakukan melalui terminal dengan perintah `service netwok restart`.Untuk memeriksa alamat IP yang sudah di konfigurasi ketikkan perintah `ifconfig`.

```
[root@localhost~]#service network restart
```

```
[root@localhost ~]# ifconfig
eth2 Link encap:Ethernet  HWaddr 00:0C:29:19:0D:8B
     inetaddr:11.0.0.2 Bcast:11.255.255.255  Mask:255.255.255.0
```

Alamat `11.0.0.2` digunakan untuk menghubungkan server dengan jaringan lokal melalui mikrotik server.

4.3.6 Konfigurasi *iptables*

1. Tambahkan rule iptables untuk mengarahkan koneksi agar paket yang lewat di jaringan harus melewati Dansguardian terlebih dahulu.

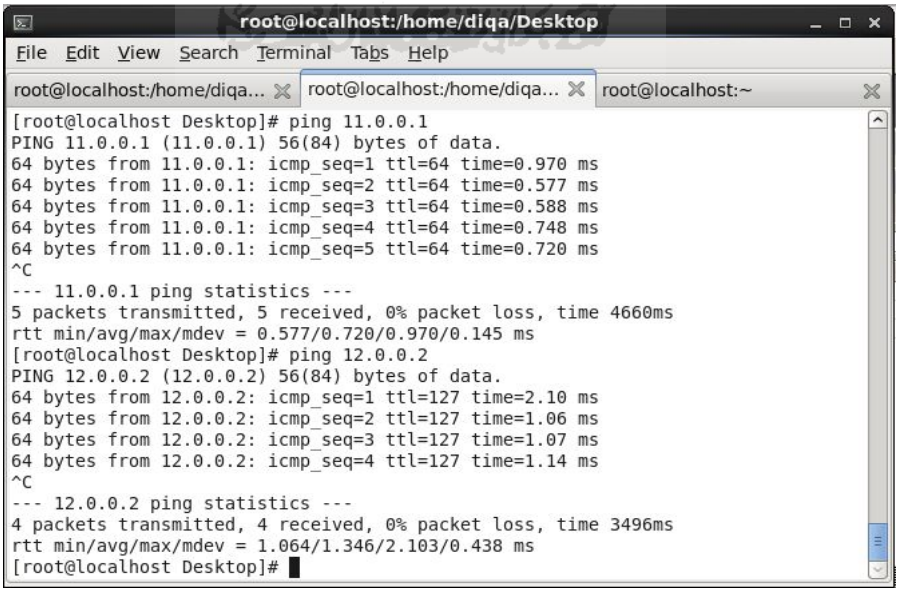
```
iptables --table nat -A PREROUTING -s 12.0.0.0/24  -p
tcp -m tcp --dport 80 -j REDIRECT --to-ports 8080
```

2. Ubah file `ip_forward` untuk memforward paket yang melewati server yang terdapat pada `/proc/sys/net/ipv4/ip_forward` menjadi 1.

```
[root@localhost~]#nano /proc/sys/net/ipv4/ip_forward
```

Gambar 4.15 Konfigurasi *ip_forward*

3. Lakukan test koneksi dari server ke alamat IP mikrotik (11.0.0.1) dan client (12.0.0.2)



Gambar 4.16 Ping test menuju 11.0.0.1 dan 12.0.0.2

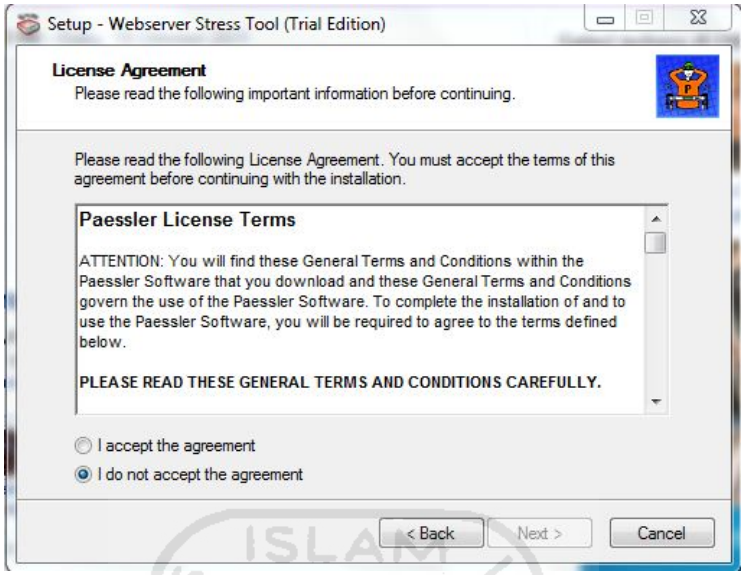
4. Jika seluruh koneksi sudah dipastikan terhubung, maka jaringan komputer sudah siap untuk digunakan.

4.3.7 Installasi Webserver Stress Tool

1. Klik dua kali pada webstress.exe yang telah di download sehingga muncul tampilan awal halaman installasi seperti pada gambar 4.17 dibawah ini.

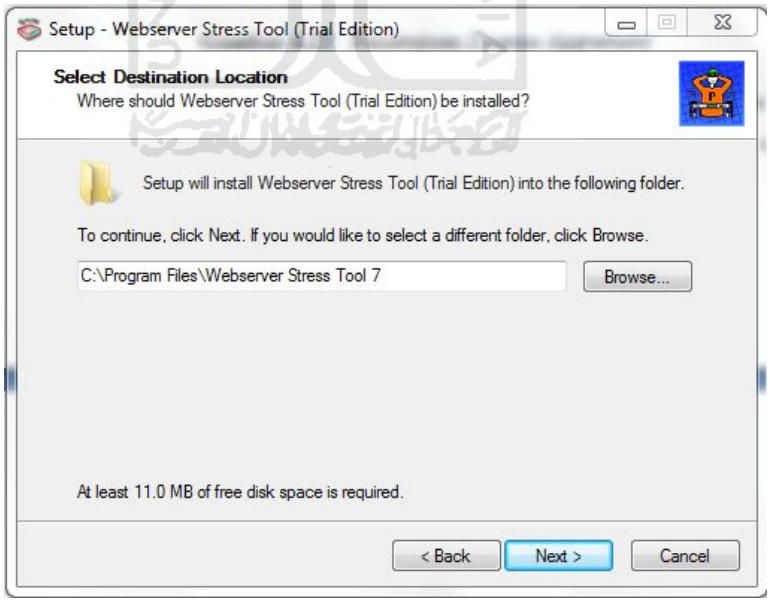
Gambar 4.17 Tampilan awal installasi webstress

2. Pada gambar 4.18 merupakan tampilan license aggrement. Klik opsi yang atas setelah itu klik next kembali.



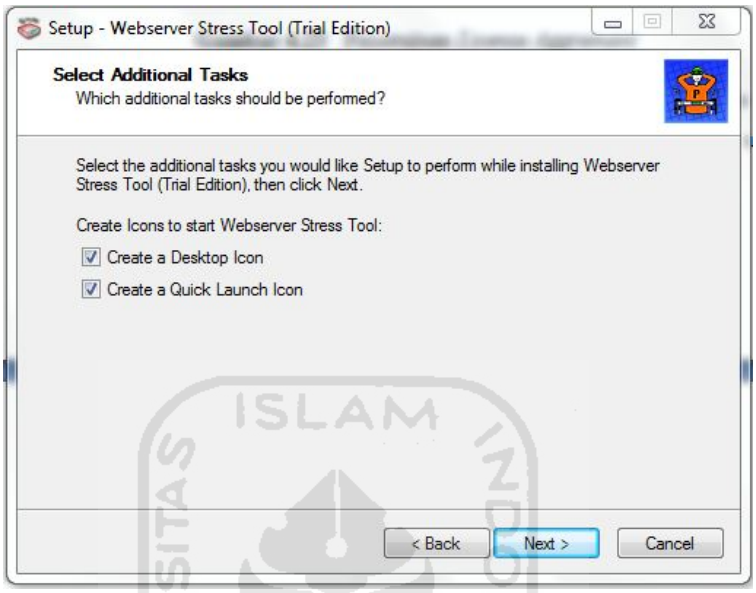
Gambar 4.18 Persetujuan Lisence Agreement

3. Pada gambar 4.19 akan muncul tampilan pilihan tempat dimana kita akan meletakkan hasil installasi tersebut. Setelah itu klik next untuk melanjutkan ke langkah selanjutnya.



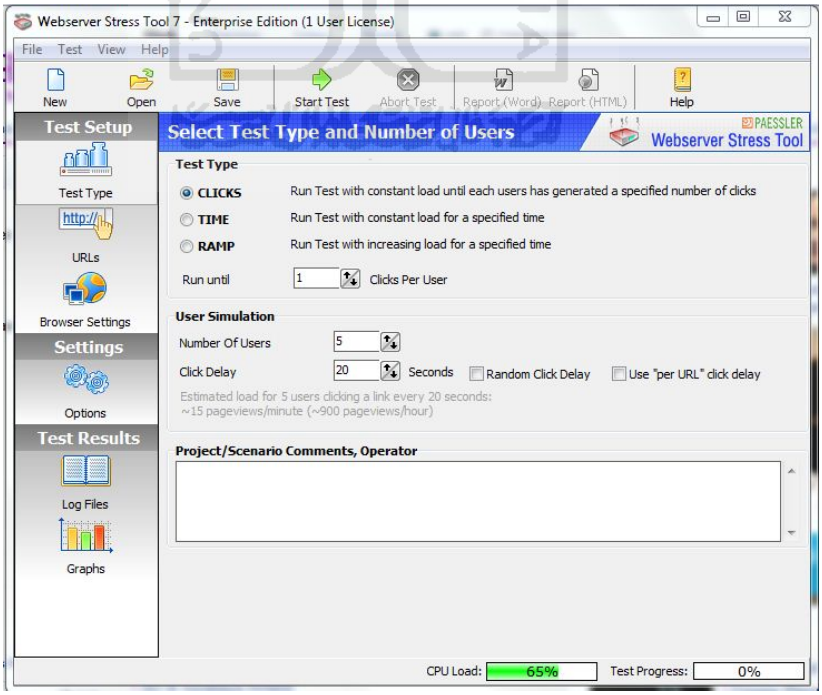
Gambar 4.19 Peletakan hasil installasi

4. Pada gambar 4.20 muncul tampilan berikutnya yaitu persetujuan untuk membuat shortcut. Jika setuju langsung next untuk melanjutkan.



Gambar 4.20 Pembuatan *shortcut*

5. Pada gambar 4.21 ini adalah tampilan awal dari web stress toll



Gambar 4.21 Tampilan awal Webserver Stress Tool

4.4 Analisis Data

Analisis data pada penelitian ini menggunakan Webserver Stress Tool sebagai *user simulator* yang mensimulasikan 1000 user dengan aturan untuk masing-masing user adalah 3 kali klik dan delay 1 detik untuk mengakses data pada saat bersamaan. Sedangkan parameternya menggunakan :

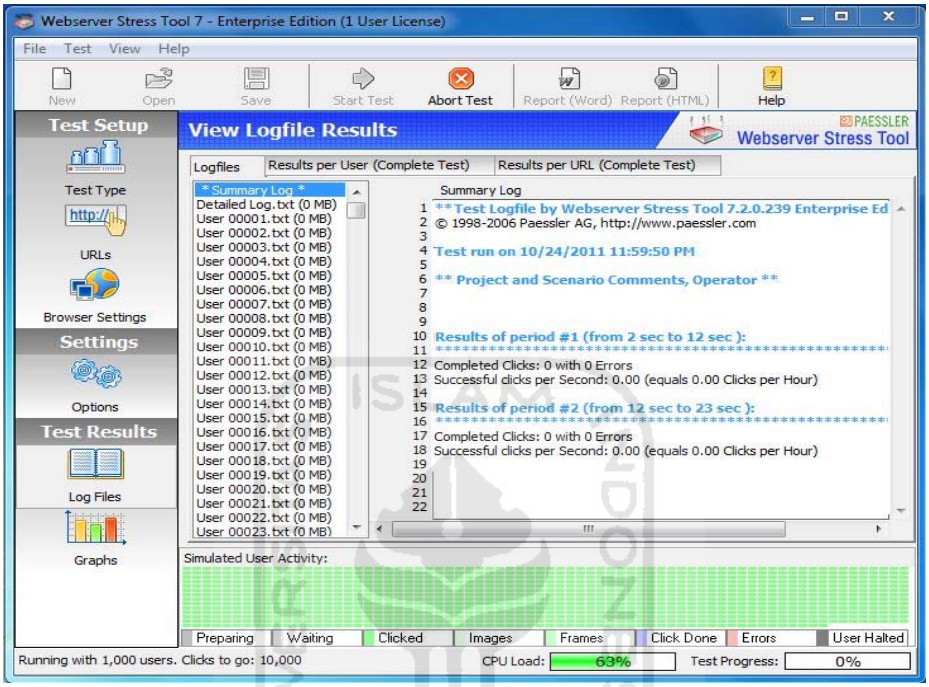
a. Rata-rata waktu eksekusi data saat user mengakses
b. *Memory usage* dan *load average*

Parameter *memory usage* dan *load average* digunakan untuk mengetahui kinerja server saat menggunakan Dansguardian dan tanpa Dansguardian. *Memory usage* merupakan penggunaan RAM pada computer saat aplikasi berjalan pada computer. RAM berfungsi menyimpan instruksi-instruksi yang dibutuhkan oleh processor sehingga aplikasi dapat berjalan dengan baik. Jika pada memori pada RAM sudah penuh atau hampir habis, makan processor akan mencari instruksi yang dibutuhkannya ke tempat lainnya seperti *harddisk, cd-rom, flash disk* dan system penyimpanan lainnya kemudian dikirim ke RAM, proses ini dinamakan *loading.* Jika aplikasi Dansguardian berjalan pada sistem dan melakukan filtering terhadap seluruh paket data yang melewati server, prosessor akan menjalankan mekanisme yang sudah dijelaskan di awal dan mengambil instruksi agar Dansguardian dapat berjalan dengan baik, semakin banyak paket data yang akan lewat, *memory usage* pada RAM akan melayani permintaan instruksi agar dapat menjalankan Dansguardian dengan baik.

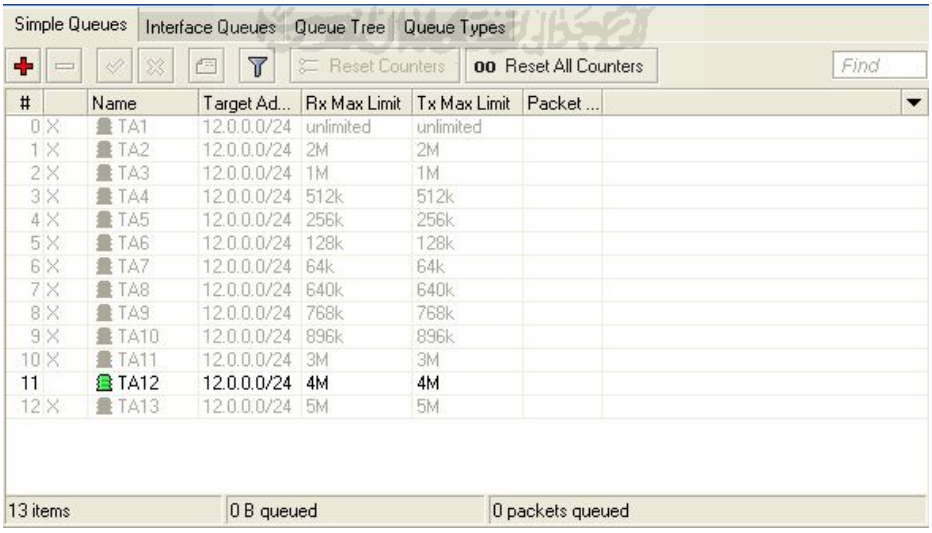
Load average merupakan rata-rata beban yang sedang dijalankan oleh computer, jika saat Dansguardian dinyalakan dan melakukan filtering, beban pada computer akanberubah sesuai dengan kinerja yang sedang dilakukan oleh aplikasi yang sedang berjalan.

Pengujian dilakukan menggunakan bandwidth yang berbeda-beda untuk mendapatkan data yang dibutuhkan. Berikut adalah gambar 4.22 yang merupakan proses Webserver Stress Tool saat melakukan simulasi client dan

gambar 4.23 merupakan pengaturan bandwidth yang sudah di buat pada mikrotik.



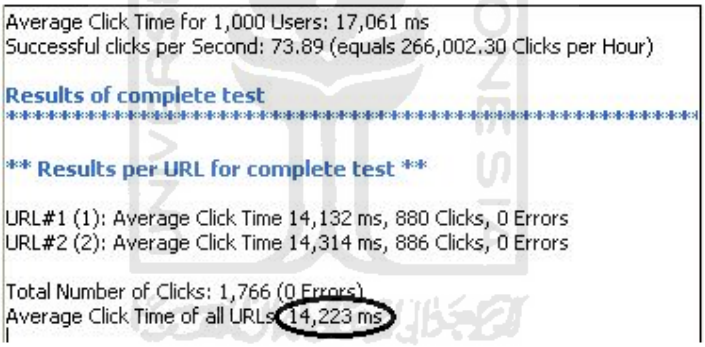
Gambar 4.22 Proses simulasi Webserver Stress Tool



Gambar 4.23 *Queue list*

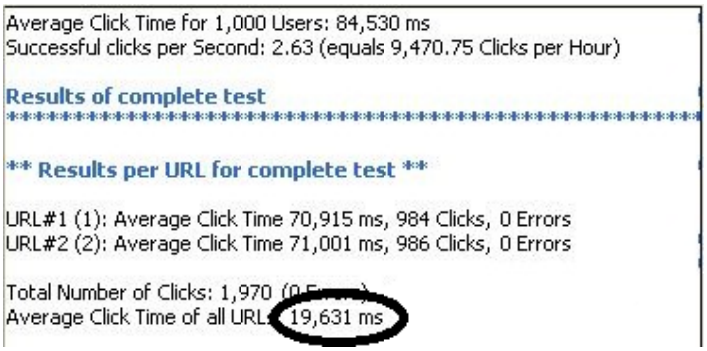
4.4.1 Pengujian dengan parameter rata-rata waktu eksekusi data

Pengujian dilakukan menggunakan web stress tool untuk mengetahui perbedaan lamanya waktu eksekusi data sebelum dan sesudah menggunakan Dansguardian pada server. Pengujian ini akan membandingkan hasil pengujian menggunakan web stress tool client menuju web server melalui server Dansguardian. Percobaan ini dilakukan sebanyak 10 kali pengujian pada masing – masing bandwidth menggunakan web stress tool saat sistem menggunakan Dansguardian dan tanpa Dansguardian, kemudian di ambil nilai rata-rata secara keseluruhannya pada bandwidth 5Mbps, 4Mbps, 3Mbps, 2Mbps, 1Mbps, 896 kbps, 768 kbps, 640kbps, 512kbps, 256 kbps, 384 kbps, 128kbps dan 64kbps. Pada gambar 4.24 menunjukan hasil ping sebelum menggunakan Dansguardian.



Gambar 4.24 Rata-rata waktu eksekusi sebelum menggunakan Dansguardian

Pada gambar 4.24 dapat dilihat hasil waktu rata-rata adalah 14,223ms. Hal ini tentunya berbeda dengan hasil pengujian ketika menggunakan Dansguardian. Pada gambar 4.25 menunjukan rata-rata setelah menggunakan Dansguardian.

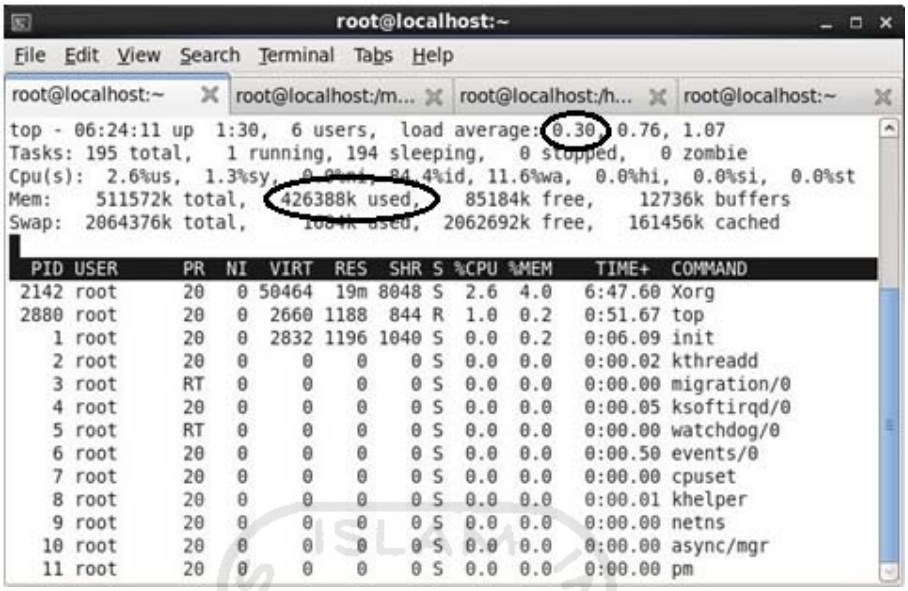


Gambar 4.25 Rata-rata waktu eksekusi setelah menggunakan Dansguardian

Pada gambar 4.25 dapat dilihat hasil rata-rata setelah menggunakan Dansguardian adalah 19,631 ms. Terjadi perbedaan lama waktu pengiriman data seperti yang ditunjukan pada dua gambar di atas.

4.4.2 Pengujian dengan parameter *memory usage* dan *load average*

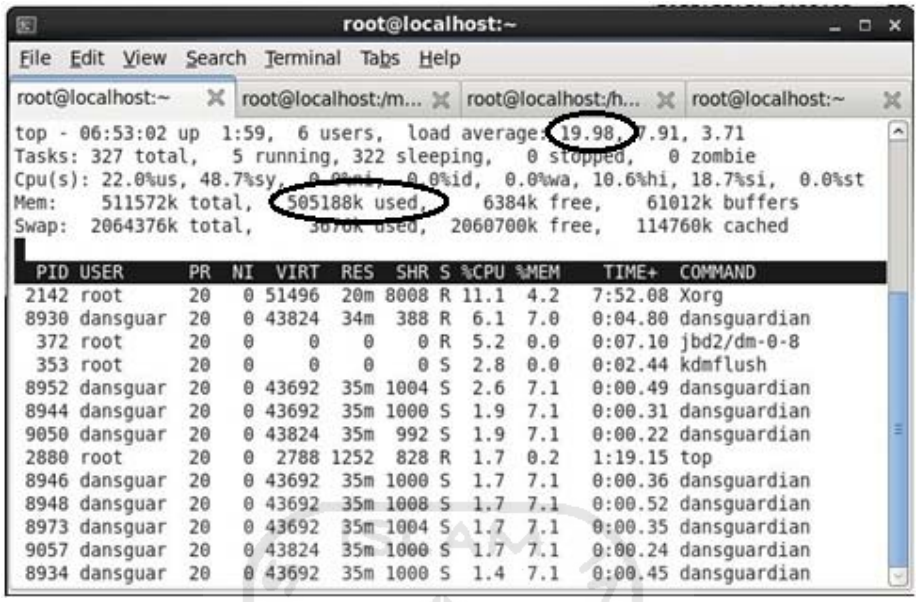
Pengujian pada server dilakukan untuk mengetahui perbedaan penggunaan *memory* dan *load average* sebelum dan sesudah menggunakan Dansguardian pada server. Pengujian ini akan membandingkan hasil pengujian *memory usage* dan *load average* menggunakan perintah `top` pada server Dansguardian yang sudah ada. Pengujian dilakukan dengan 10 kali pengujian pada masing – masing bandwidth menggunakan perintah `top` saat sistem menggunakan Dansguardian dan tanpa Dansguardian pada bandwidth 5Mbps, 4Mbps, 3Mbps, 2Mbps, 1Mbps, 896 kbps, 768 kbps, 640kbps, 512kbps, 256 kbps, 384kbps, 128kbps dan 64kbps., untuk di ambil nilai rata-rata secara keseluruhannya. Pada gambar 4.26 menunjukan *memory usage* dan *load average* sebelum menggunakan Dansguardian.



Gambar 4.26 Monitoring sebelum menggunakan Dansguardian

Pada gambar 4.32 di atas menunjukan penggunaan memori dan *load average* pada saat proses berjalan. Pada saat proses berjalan, memory yang digunakan oleh server dalam melayani permintaan adalah 426388k dari memory yang ada dan *load average*nya adalah 0,30.

Sedangkan pada gambar 4.27 di bawah ini menunjukan penggunaan memori dan *load average* pada saat proses berjalan setelah menggunakan Dansguardian. Pada saat proses berjalan, memory yang digunakan oleh server dalam melayani permintaan adalah 505188k dari memory yang ada dan *load average*nya adalah 19,98.



Gambar 4.27 Monitoring setelah menggunakan Dansguardian

4.5 Tabel Analisis Data

4.5.1 Menggunakan parameter rata-rata waktu eksekusi data

Tabel 4.1 sampai tabel 4.14 serta gambar 4.28 menunjukan perbandingan rata – rata waktu eksekusi data pada masing-masing bandwidth .

Tabel 4.1 Hasil tes pada bandwidth bandwidth 64 kbps

Pengujian	Dansguardian off	Dansguardian on
1	14.15 ms	16.79 ms
2	8.39 ms	15.25 ms
3	12.58 ms	15.58 ms
4	14.34 ms	16.03 ms
5	8.49 ms	15.32 ms
6	8.84 ms	18.73 ms
7	13.46 ms	14.50 ms
8	12.40 ms	15.27 ms
9	10.00 ms	19.07 ms
10	10.78 ms	15.25 ms
RATA-RATA	11.34 ms	16.18 ms

Tabel 4.2 Hasil tes pada bandwidth 128 kbps

Pengujian	Dansguardian off	Dansguardian on
1	9.07 ms	15.52 ms
2	7.91 ms	15.99 ms
3	13.30 ms	20.12 ms
4	12.81 ms	13.55 ms
5	14.00 ms	19.31 ms
6	13.48 ms	17.60 ms
7	15.33 ms	15.31 ms
8	11.97 ms	15.38 ms
9	14.09 ms	16.55 ms
10	13.99 ms	14.49 ms
RATA-RATA	12.59 ms	16.38 ms

Tabel 4.3 Hasil tes pada bandwidth 256 kbps

Pengujian	Dansguardian off	Dansguardian on
1	14.29 ms	16.74 ms
2	13.71 ms	16.81 ms
3	12.22 ms	18.35 ms
4	12.37 ms	15.18 ms
5	10.13 ms	17.18 ms
6	11.38 ms	17.20 ms
7	13.32 ms	13.59 ms
8	11.45 ms	17.43 ms
9	11.43 ms	14.53 ms
10	14.42 ms	16.98 ms
RATA-RATA	12.47 ms	16.40 ms

Tabel 4.4 Hasil tes pada bandwidth 384 kbps

Pengujian	Dansguardian off	Dansguardian on
1	11.11 ms	18.13 ms
2	19.23 ms	17.54 ms
3	14.25 ms	15.50 ms
4	10.17 ms	17.91 ms
5	9.27 ms	18.90 ms
6	8.71 ms	17.29 ms
7	11.27 ms	17.96 ms
8	17.21 ms	16.59 ms
9	10.38 ms	14.99 ms
10	13.41 ms	10.53 ms
RATA-RATA	12.50 ms	16.53 ms

Tabel 4.5 Hasil tes pada bandwidth 512 kbps

Pengujian	Dansguardian off	Dansguardian on
1	11.37 ms	11.53 ms
2	11.80 ms	18.49 ms
3	15.01 ms	18.41 ms
4	15.22 ms	18.32 ms
5	11.87 ms	18.90 ms
6	10.57 ms	19.78 ms
7	12.57 ms	19.45 ms
8	9.56 ms	18.29 ms
9	11.57 ms	15.83 ms
10	9.96 ms	17.70 ms
RATA-RATA	11.95 ms	17.67 ms

Tabel 4.6 Hasil tes pada bandwidth 640 kbps

Pengujian	Dansguardian off	Dansguardian on
1	10.69 ms	18.36 ms
2	9.76 ms	17.26 ms
3	11.94 ms	18.13 ms
4	12.12 ms	17.25 ms
5	11.47 ms	16.39 ms
6	14.61 ms	19.15 ms
7	12.29 ms	18.10 ms
8	11.82 ms	17.09 ms
9	12.70 ms	18.13 ms
10	11.62 ms	17.11 ms
RATA-RATA	11.90 ms	17.70 ms

Tabel 4.7 Hasil tes pada bandwidth 768 kbps

Pengujian	Dansguardian off	Dansguardian on
1	12.61 ms	18.25 ms
2	14.94 ms	16.77 ms
3	11.97 ms	18.53 ms
4	10.50 ms	18.26 ms
5	12.94 ms	19.50 ms
6	11.90 ms	18.78 ms
7	10.72 ms	18.34 ms
8	9.83 ms	16.39 ms
9	12.78 ms	17.49 ms
10	11.88 ms	16.66 ms
RATA-RATA	12.01 ms	17.90 ms

Tabel 4.8 Hasil tes pada bandwidth 896 kbps

Pengujian	Dansguardian off	Dansguardian on
1	11.24 ms	18.37 ms
2	11.98 ms	17.83 ms
3	14.34 ms	17.86 ms
4	12.71 ms	17.96 ms
5	10.97 ms	15.28 ms
6	8.92 ms	19.55 ms
7	11.86 ms	18.21 ms
8	11.70 ms	17.51 ms
9	10.81 ms	17.61 ms
10	10.60 ms	19.53 ms
RATA-RATA	11.51 ms	17.97 ms

Tabel 4.9 Hasil tes pada bandwidth 1 MB

Pengujian	Dansguardian off	Dansguardian on
1	11.76 ms	24.28 ms
2	11.47 ms	23.44 ms
3	14.22 ms	22.79 ms
4	6.72 ms	22.86 ms
5	6.40 ms	24.85 ms
6	10.97 ms	21.89 ms
7	13.81 ms	22.62 ms
8	9.41 ms	21.91 ms
9	12.21 ms	23.56 ms
10	10.21 ms	20.31 ms
RATA-RATA	10.72 ms	22.85 ms

Tabel 4.10 Hasil tes pada bandwidth 2 MB

Pengujian	Dansguardian off	Dansguardian on
1	8.80 ms	23.11 ms
2	8.61 ms	23.41 ms
3	7.98 ms	25.16 ms
4	9.89 ms	23.04 ms
5	9.93 ms	24.35 ms
6	9.97 ms	25.06 ms
7	9.84 ms	25.15 ms
8	8.98 ms	26.06 ms
9	9.75 ms	25.60 ms
10	9.64 ms	24.11 ms
RATA-RATA	9.34 ms	24.50 ms

Tabel 4.11 Hasil tes pada bandwidth 3 MB

Pengujian	Dansguardian off	Dansguardian on
1	3.93 ms	26.06 ms
2	4.85 ms	23.46 ms
3	4.51 ms	26.80 ms
4	3.54 ms	24.14 ms
5	5.14 ms	24.11 ms
6	5.97 ms	25.11 ms
7	5.76 ms	25.39 ms
8	5.23 ms	26.16 ms
9	5.39 ms	26.34 ms
10	4.80 ms	26.45 ms
RATA-RATA	4.91 ms	25.40 ms

Tabel 4.12 Hasil tes pada bandwidth 4 MB

Pengujian	Dansguardian off	Dansguardian on
1	5.26 ms	26.16 ms
2	4.10 ms	24.46 ms
3	3.44 ms	25.64 ms
4	4.14 ms	26.79 ms
5	5.21 ms	28.64 ms
6	3.63 ms	26.25 ms
7	5.29 ms	26.16 ms
8	5.06 ms	27.31 ms
9	4.21 ms	26.01 ms
10	5.04 ms	26.80 ms
RATA-RATA	4.54 ms	26.42 ms

Tabel 4.13 Hasil tes pada bandwidth 5 MB

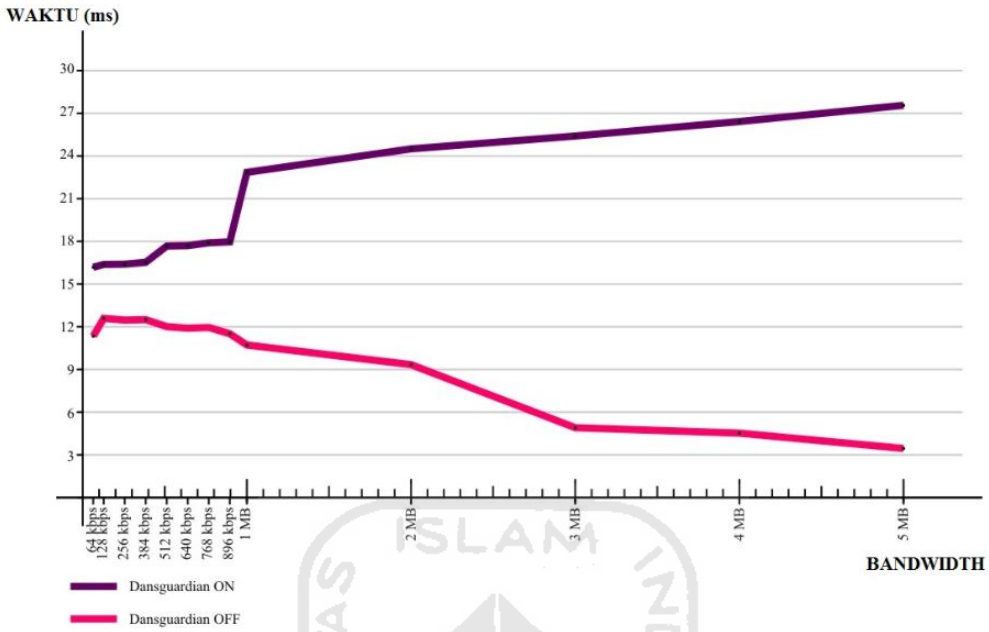
Pengujian	Dansguardian off	Dansguardian on
1	3.14 ms	26.23 ms
2	5.12 ms	27.52 ms
3	3.13 ms	28.95 ms
4	4.46 ms	26.67 ms
5	3.16 ms	27.40 ms
6	2.11 ms	25.51 ms
7	3.39 ms	26.84 ms
8	3.20 ms	29.96 ms
9	3.01 ms	28.55 ms
10	3.78 ms	27.98 ms
RATA-RATA	3.45 ms	27.56 ms

Tabel 4.14 Rata-rata hasil pengujian pada seluruh bandwidth

Bandwidth	Dansguardian off	Dansguardian on
64 kbps	11.34 ms	16.18 ms
128 kbps	12.59 ms	16.38 ms
256 kbps	12.47 ms	16.40 ms
384 kbps	12.50 ms	16.53 ms
512 kbps	11.95 ms	17.67 ms
640 kbps	11.90 ms	17.70 ms
768 kbps	12.01 ms	17.90 ms
896 kbps	11.51 ms	17.97 ms
1 MB	10.72 ms	22.85 ms
2 MB	9.34 ms	24.50 ms
3 MB	4.91 ms	25.40 ms
4 MB	4.54 ms	26.42 ms
5 MB	3.45 ms	27.56 ms

Data rata-rata waktu eksekusi data baik sebelum maupun sesudah pengimplementasian Dansguardian mengalami perubahan hal ini disebabkan perbedaan penggunaan bandwidth pada jaringan komputer dan pemeriksaan / filtering yang dilakukan oleh Dansguardian.

Beradasarkan tabel 4.14 perbandingan waktu pengiriman paket baik sebelum dan sesudah menggunakan Dansguardian menunjukan perbedaan waktu pada setiap bandwidth .Dilihat dari tabel 4.14, saat servis Dansguardian dijalankan pada server, pada bandwidth 1Mbps, perbandingan waktu rata-rata eksekusi data antar sebelum dan sesudah menggunakan Dansguardian menjadi sangat jauh.



Gambar 4.28 Grafik Perbandingan hasil test waktu eksekusi data

Dari Gambar 4.28 dapat dilihat bahwa mulai bandwidth 1 Mbps, waktu eksekusi mengalami kenaikan yang signifikan. Saat bandwidth ditambah menjadi 2 Mbps, waktu eksekusi data menjadi semakin lambat. Jika di hitung presentase kenaikan waktu eksekusi datanya, maka di dapatkan hasil sebagai berikut.

$$\text{64 kbps - 128 kbps} = \frac{16.38 \text{ ms} - 16.18 \text{ ms}}{16.18 \text{ ms}} \times 100 = 1.25 \%$$

$$\text{128 kbps - 256 kbps} = \frac{16.40 \text{ ms} - 16.38 \text{ ms}}{16.38 \text{ ms}} \times 100 = 0.1 \%$$

$$\text{256 kbps - 384 kbps} = \frac{16.53 \text{ ms} - 16.40 \text{ ms}}{16.40 \text{ ms}} \times 100 = 0.8 \%$$

$$\text{384 kbps - 512 kbps} = \frac{17.67 \text{ ms} - 16.53 \text{ ms}}{16.53 \text{ ms}} \times 100 = 6.9 \%$$

$$\text{512 kbps - 640 kbps} = \frac{17.70 \text{ ms} - 17.67 \text{ ms}}{17.67 \text{ ms}} \times 100 = 0.1 \%$$

$$\text{640 kbps - 768 kbps} = \frac{17.90 \text{ ms} - 17.70 \text{ ms}}{17.70 \text{ ms}} \times 100 = 1.1 \%$$

$$\text{768 kbps - 896 kbps} = \frac{17.97 \text{ ms} - 17.90 \text{ ms}}{17.90 \text{ ms}} \times 100 = 0.4 \%$$

896 kbps - 1 Mbps = $\frac{22.85 \text{ ms} - 17.97 \text{ ms}}{17.97 \text{ ms}} \times 100 =$ 27 %

1 Mbps - 2 Mbps = $\frac{24.50 \text{ ms} - 22.85 \text{ ms}}{22.85 \text{ ms}} \times 100 =$ 7.2 %

2 Mbps - 3 Mbps = $\frac{25.40 \text{ ms} - 24.50 \text{ ms}}{24.50 \text{ Ms}} \times 100 =$ 3.7 %

3 Mbps - 4 Mbps = $\frac{26.42 \text{ ms} - 25.40 \text{ ms}}{25.40 \text{ Ms}} \times 100 =$ 4 %

4 Mbps - 5 Mbps = $\frac{27.56 \text{ ms} - 26.42 \text{ ms}}{26.42 \text{ Ms}} \times 100 =$ 4.3 %

Kenaikan waktu eksekusi data dari bandwidth 896 kbps ke 1 Mbps mencapai 27%, sedangkan pada bandwidth 64 kbps – 896 kbps hanya berkisar antara 0,1% – 1,25 % dan pada bandwidth 1 Mbps ke atas, kenaikan waktu eksekusi data berkisar antara 3% - 7%.

Filtering yang dilakukan oleh Dansguardian mulai dilakukan setelah file / data yang di *request* oleh *user* diterima oleh Dansguardian. Misal, jika user meminta data A, maka data A diterima terlebih dahulu oleh Dansguardian, baru kemudian dilakukan pemeriksaan dengan rule yang sudah ditentukan.

Karena data yang akan di filter harus diterima terlebih dahulu, maka akan terdapat antrian data pada jaringan yang menunggu untuk diperiksa. Antrian tersebut tent saja akan menggunakan bandwidth yang tersedia.

Pada bandwidth 1 Mbps, data yang menunggu untuk diperiksa menjadi sangat banyak dan memenuhi *traffic* pada bandwidth yang tersedia. Pada saat itulah, peningkatan waktu eksekusi data menjadi sangat tinggi.

4.5.2 Menggunakan parameter *memory usage*

Tabel 4.15 sampai tabel 4.28 serta gambar 4.29 menunjukan perbandingan *memory usage* pada masing-masing bandwidth

Tabel 4.15 Hasil tes *memory usage* pada bandwidth 64 kbps

Pengujian	Dansguardian off	Dansguardian on
1	401,760 kB	489,252 kB
2	400,774 kB	490,764 kB
3	403,220 kB	476,364 kB
4	402,476 kB	479,984 kB
5	426,388 kB	484,548 kB
6	474,292 kB	486,276 kB
7	473,492 kB	487,888 kB
8	504,790 kB	488,756 kB
9	504,296 kB	491,468 kB
10	503,692 kB	493,088 kB
RATA-RATA	449,518 kB	486,839 kB

Tabel 4.16 Hasil tes *memory usage* pada bandwidth 128 kbps

Pengujian	Dansguardian off	Dansguardian on
1	497,332 kB	485,112 kB
2	504,196 kB	488,876 kB
3	492,880 kB	482,428 kB
4	456,316 kB	484,904 kB
5	460,872 kB	485,240 kB
6	466,376 kB	489,376 kB
7	468,072 kB	491,792 kB
8	471,468 kB	493,804 kB
9	480,396 kB	484,060 kB
10	426,772 kB	498,068 kB
RATA-RATA	472,468 kB	488,366 kB

Tabel 4.17 Hasil tes *memory usage* pada bandwidth 256 kbps

Pengujian	Dansguardian off	Dansguardian on
1	459,832 kB	489,080 kB
2	468,024 kB	478,498 kB
3	475,448 kB	481,294 kB
4	469,736 kB	487,566 kB
5	475,472 kB	493,258 kB
6	478,664 kB	493,298 kB
7	482,268 kB	497,188 kB
8	484,864 kB	491,494 kB
9	487,600 kB	497,696 kB
10	491,772 kB	491,496 kB
RATA-RATA	477,368 kB	490,087 kB

Tabel 4.18 Hasil tes *memory usage* pada bandwidth 384 kbps

Pengujian	Dansguardian off	Dansguardian on
1	458,219 kB	481,221 kB
2	492,364 kB	497,241 kB
3	469,529 kB	491,210 kB
4	484,929 kB	488,116 kB
5	459,572 kB	489,116 kB
6	491,459 kB	498,536 kB
7	495,439 kB	494,125 kB
8	497,295 kB	497,168 kB
9	487,229 kB	478,216 kB
10	485,259 kB	493,216 kB
RATA-RATA	482,129 kB	490,817 kB

Tabel 4.19 Hasil tes *memory usage* pada bandwidth 512 kbps

Pengujian	Dansguardian off	Dansguardian on
1	498,020 kB	493,296 kB
2	451,052 kB	495,932 kB
3	504,608 kB	486,652 kB
4	445,864 kB	497,884 kB
5	463,508 kB	492,464 kB
6	480,212 kB	494,664 kB
7	485,392 kB	483,028 kB
8	497,488 kB	485,896 kB
9	503,560 kB	499,608 kB
10	504,164 kB	498,268 kB
RATA-RATA	483,387 kB	492,769 kB

Tabel 4.20 Hasil tes *memory usage* pada bandwidth 640 kbps

Pengujian	Dansguardian off	Dansguardian on
1	451,220 kB	495,601 kB
2	485,223 kB	492,138 kB
3	496,215 kB	493,167 kB
4	443,044 kB	495,215 kB
5	527,125 kB	486,127 kB
6	497,116 kB	497,533 kB
7	503,262 kB	499,225 kB
8	464,213 kB	494,722 kB
9	485,215 kB	498,213 kB
10	503,489 kB	486,217 kB
RATA-RATA	485,612 kB	493,816 kB

Tabel 4.21 Hasil tes *memory usage* pada bandwidth 768 kbps

Pengujian	Dansguardian off	Dansguardian on
1	471,749 kB	495,216 kB
2	442,146 kB	492,856 kB
3	464,213 kB	487,825 kB
4	499,213 kB	488,425 kB
5	484,113 kB	497,126 kB
6	504,114 kB	492,456 kB
7	503,845 kB	495,165 kB
8	485,189 kB	498,215 kB
9	527,396 kB	498,216 kB
10	496,138 kB	494,128 kB
RATA-RATA	487,812 kB	493,963 kB

Tabel 4.22 Hasil tes *memory usage* pada bandwidth 896 kbps

Pengujian	Dansguardian off	Dansguardian on
1	491,746 kB	498,635 kB
2	491,237 kB	500,322 kB
3	489,274 kB	492,150 kB
4	491,746 kB	502,256 kB
5	492,856 kB	492,254 kB
6	480,157 kB	471,325 kB
7	493,496 kB	503,246 kB
8	491,274 kB	499,314 kB
9	491,466 kB	492,145 kB
10	491,865 kB	498,213 kB
RATA-RATA	490,512 kB	494,986 kB

Tabel 4.23 Hasil tes *memory usage* pada bandwidth 1MB

Pengujian	Dansguardian off	Dansguardian on
1	492,332 kB	503,148 kB
2	493,656 kB	495,690 kB
3	491,488 kB	502,297 kB
4	492,044 kB	498,798 kB
5	491,424 kB	497,458 kB
6	495,228 kB	505,097 kB
7	493,036 kB	505,352 kB
8	492,780 kB	505,100 kB
9	493,160 kB	478,198 kB
10	493,440 kB	479,716 kB
RATA-RATA	492,859 kB	497,085 kB

Tabel 4.24 Hasil tes *memory usage* pada bandwidth 2MB

Pengujian	Dansguardian off	Dansguardian on
1	489,100 kB	505,188 kB
2	502,408 kB	499,180 kB
3	505,106 kB	504,280 kB
4	502,128 kB	505,656 kB
5	498,152 kB	492,544 kB
6	498,160 kB	502,336 kB
7	486,278 kB	491,228 kB
8	500,354 kB	492,560 kB
9	493,236 kB	502,082 kB
10	499,196 kB	493,064 kB
RATA-RATA	497,412 kB	498,812 kB

Tabel 4.25 Hasil tes *memory usage* pada bandwidth 3MB

Pengujian	Dansguardian off	Dansguardian on
1	505,831 kB	508,429 kB
2	505,598 kB	508,459 kB
3	503,999 kB	505,945 kB
4	503,987 kB	508,721 kB
5	505,697 kB	508,929 kB
6	504,995 kB	508,735 kB
7	505,869 kB	507,127 kB
8	505,966 kB	509,308 kB
9	505,979 kB	507,913 kB
10	505,776 kB	508,813 kB
RATA-RATA	505,370 kB	508,238 kB

Tabel 4.26 Hasil tes *memory usage* pada bandwidth 4MB

Pengujian	Dansguardian off	Dansguardian on
1	505,998 kB	505,985 kB
2	505,933 kB	508,669 kB
3	505,723 kB	508,859 kB
4	505,853 kB	506,429 kB
5	506,939 kB	509,948 kB
6	507,921 kB	509,315 kB
7	504,828 kB	509,644 kB
8	506,863 kB	508,919 kB
9	504,875 kB	509,984 kB
10	503,964 kB	505,547 kB
RATA-RATA	505,890 kB	508,330 kB

Tabel 4.27 Hasil tes *memory usage* pada bandwidth 5MB

Pengujian	Dansguardian off	Dansguardian on
1	505,876 kB	508,976 kB
2	507,436 kB	509,953 kB
3	505,123 kB	508,955 kB
4	505,498 kB	509,991 kB
5	503,128 kB	508,995 kB
6	505,489 kB	507,983 kB
7	502,357 kB	509,958 kB
8	505,634 kB	507,998 kB
9	504,845 kB	508,872 kB
10	505,879 kB	509,993 kB
RATA-RATA	505,127 kB	509,167 kB

Tabel 4.28 Rata-rata hasil pengujian *memory usage* pada seluruh bandwidth

Bandwidth	Dansguardian off	Dansguardian on
64 kbps	449,518 kB	486,839 kB
128 kbps	472,468 kB	488,366 kB
256 kbps	477,368 kB	490,087 kB
384 kbps	482,129 kB	490,817 kB
512 kbps	483,387 kB	492,769 kB
640 kbps	485,612 kB	493,816 kB
768 kbps	487,812 kB	493,963 kB
896 kbps	490,512 kB	494,986 kB
1 MB	492,859 kB	497,085 kB
2 MB	497,412 kB	498,812 kB
3 MB	505,370 kB	508,238 kB
4 MB	505,890 kB	508,330 kB
5 MB	505,127 kB	509,167 kB

Dari Tabel 4.28 dapat dilihat bahwa penggunaan *memory usage* saat menggunakan Dansguardian pada server, tidak mengalami kenaikan yang signifikan. Jika di hitung presentase penggunaan *memory usage* pada tiap bandwidth, maka di dapatkan hasil sebagai berikut.

$$\text{64 kbps - 128 kbps} = \frac{488{,}366 \text{ kB} - 486{,}839 \text{ kB}}{486{,}839 \text{ kB}} \times 100 = 0.31\ \%$$

$$\text{128 kbps - 256 kbps} = \frac{490{,}087 \text{ kB} - 488{,}366 \text{ kB}}{488{,}366 \text{ kB}} \times 100 = 0.4\ \%$$

$$\text{256 kbps - 384 kbps} = \frac{490{,}817 \text{ kB} - 490{,}087 \text{ kB}}{490{,}087 \text{ kB}} \times 100 = 0.1\ \%$$

$$\text{384 kbps - 512 kbps} = \frac{492{,}769 \text{ kB} - 490{,}817 \text{ kB}}{490{,}817 \text{ kB}} \times 100 = 0.4\ \%$$

$$\text{512 kbps - 640 kbps} = \frac{493{,}816 \text{ kB} - 492{,}769 \text{ kB}}{492{,}769 \text{ kB}} \times 100 = 0.2\ \%$$

$$\text{640 kbps - 768 kbps} = \frac{493{,}963 \text{ kB} - 493{,}816 \text{ kB}}{493{,}816 \text{ kB}} \times 100 = 0\ \%$$

$$\text{768 kbps - 896 kbps} = \frac{494{,}986 \text{ kB} - 493{,}963 \text{ kB}}{493{,}963 \text{ kB}} \times 100 = 0.2\ \%$$

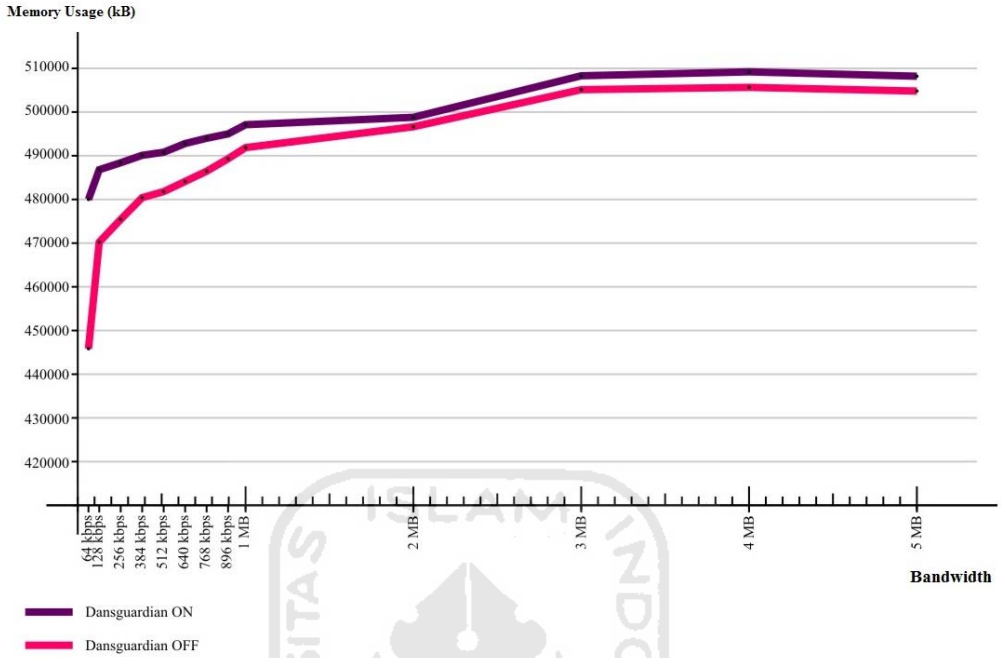
$$\text{896 kbps - 1 Mbps} = \frac{497{,}085 \text{ kB} - 494{,}986 \text{ kB}}{494{,}986 \text{ kB}} \times 100 = 0.4\ \%$$

$$\text{1 Mbps - 2 Mbps} = \frac{498{,}812 \text{ kB} - 497{,}085 \text{ kB}}{497{,}085 \text{ kB}} \times 100 = 0.3\ \%$$

$$\text{2 Mbps - 3 Mbps} = \frac{508{,}238 \text{ kB} - 498{,}812 \text{ kB}}{498{,}812 \text{ kB}} \times 100 = 1.9\ \%$$

$$\text{3 Mbps - 4 Mbps} = \frac{508{,}330 \text{ kB} - 508{,}238 \text{ kB}}{508{,}238 \text{ kB}} \times 100 = 0\ \%$$

$$\text{4 Mbps - 5 Mbps} = \frac{509{,}167 \text{ kB} - 508{,}330 \text{ kB}}{508{,}330 \text{ kB}} \times 100 = 0.2\ \%$$



Gambar 4.29 Grafik Perbandingan *memory usage*

Data *memory usage* baik sebelum maupun sesudah pengimplementasian Dansguardian mengalami perubahan hal ini disebabkan perbedaan penggunaan bandwidth pada jaringan komputer dan pemeriksaan / filtering yang dilakukan oleh Dansguardian.

Penigkatan pengunaan *memory usage* dari seluruh bandwidth mencapai hanya berkisar antara 0% – 1,9 %. Beradasarkan tabel 4.28 dan hasil perhitungan presentase penigkatan penggunaan *memory usage* cenderung stabil dan tidak mengalami perubahan yang signifikan.

Karena Dansguardian berjalan secara *daemonized*, jadi meskipun lalu lintas atau *traffic* pada jaringan tersebut penuh, namun proses filtering yang dilakukan tidak memakan banyak *memory* yang tersedia.

4.5.3 Menggunakan parameter *load average*

Tabel 4.29 sampai tabel 4.42 serta gambar 4.30 menunjukan perbandingan *load average* pada masing-masing bandwidth.

Tabel 4.29 Hasil tes *load average* pada bandwidth 64 kbps

Pengujian	Dansguardian off	Dansguardian on
1	0.30 second	19.98 second
2	0.54 second	9.52 second
3	2.15 second	24.22 second
4	0.69 second	24.08 second
5	3.47 second	31.34 second
6	2.26 second	24.05 second
7	7.06 second	27.07 second
8	1.52 second	19.99 second
9	0.43 second	22.78 second
10	3.10 second	22.77 second
RATA-RATA	2.15 second	22.58 second

Tabel 4.30 Hasil tes *load average* pada bandwidth 128 kbps

Pengujian	Dansguardian off	Dansguardian on
1	3.38 second	11.48 second
2	1.28 second	10.29 second
3	4.18 second	26.15 second
4	2.19 second	29.26 second
5	0.92 second	21.22 second
6	4.85 second	22.20 second
7	2.18 second	26.14 second
8	1.44 second	30.84 second
9	2.73 second	25.21 second
10	0.70 second	24.54 second
RATA-RATA	2.39 second	22.73 second

Tabel 4.31 Hasil tes *load average* pada bandwidth 256 kbps

Pengujian	Dansguardian off	Dansguardian on
1	1.56 second	32.91 second
2	1.18 second	19.18 second
3	0.88 second	25.21 second
4	2.52 second	22.02 second
5	5.47 second	20.11 second
6	2.53 second	21.32 second
7	2.21 second	27.88 second
8	1.76 second	20.76 second
9	0.48 second	21.48 second
10	7.26 second	21.85 second
RATA-RATA	2.59 second	23.27 second

Tabel 4.32 Hasil tes *load average* pada bandwidth 384 kbps

Pengujian	Dansguardian off	Dansguardian on
1	1.24 second	20.60 second
2	7.36 second	18.38 second
3	2.38 second	26.78 second
4	2.61 second	21.02 second
5	1.67 second	21.40 second
6	2.49 second	22.51 second
7	1.90 second	23.60 second
8	5.12 second	24.51 second
9	0.34 second	31.71 second
10	1.16 second	22.85 second
RATA-RATA	2.63 second	23.34 second

Tabel 4.33 Hasil tes *load average* pada bandwidth 512 kbps

Pengujian	Dansguardian off	Dansguardian on
1	2.24 second	21.33 second
2	0.49 second	16.32 second
3	7.83 second	23.89 second
4	2.41 second	30.91 second
5	5.31 second	21.76 second
6	2.11 second	37.03 second
7	0.86 second	15.99 second
8	0.17 second	24.96 second
9	0.42 second	23.46 second
10	5.90 second	20.02 second
RATA-RATA	2.77 second	23.57 second

Tabel 4.34 Hasil tes *load average* pada bandwidth 640 kbps

Pengujian	Dansguardian off	Dansguardian on
1	5.38 second	23.50 second
2	1.70 second	36.09 second
3	7.41 second	22.77 second
4	2.26 second	23.66 second
5	2.60 second	16.20 second
6	2.46 second	21.40 second
7	0.18 second	21.28 second
8	0.38 second	16.98 second
9	0.32 second	23.55 second
10	5.74 second	31.51 second
RATA-RATA	2.84 second	23.69 second

Tabel 4.35 Hasil tes load *average* pada bandwidth 768 kbps

Pengujian	Dansguardian off	Dansguardian on
1	2.80 second	22.59 second
2	2.30 second	16.13 second
3	2.59 second	22.25 second
4	0.45 second	23.38 second
5	7.15 second	22.25 second
6	0.37 second	31.49 second
7	5.30 second	23.84 second
8	5.29 second	17.92 second
9	0.28 second	22.57 second
10	2.30 second	35.19 second
RATA-RATA	2.88 second	23.76 second

Tabel 4.36 Hasil tes *load average* pada bandwidth 896 kbps

Pengujian	Dansguardian off	Dansguardian on
1	2.38 second	24.01 second
2	2.09 second	18.32 second
3	3.22 second	29.91 second
4	2.26 second	23.51 second
5	2.21 second	18.38 second
6	1.36 second	21.34 second
7	2.94 second	20.30 second
8	5.89 second	28.24 second
9	3.34 second	35.12 second
10	3.26 second	19.23 second
RATA-RATA	2.90 second	23.84 second

Tabel 4.37 Hasil tes *load average* pada bandwidth 1MB

Pengujian	Dansguardian off	Dansguardian on
1	2.10 second	18.32 second
2	2.54 second	19.54 second
3	3.44 second	24.46 second
4	2.36 second	29.49 second
5	3.12 second	30.59 second
6	3.11 second	36.24 second
7	3.15 second	23.98 second
8	5.99 second	18.45 second
9	3.37 second	19.31 second
10	2.73 second	22.18 second
RATA-RATA	3.19 second	24.26 second

Tabel 4.38 Hasil tes *load average* pada bandwidth 2MB

Pengujian	Dansguardian off	Dansguardian on
1	2.77 second	19.74 second
2	3.27 second	22.16 second
3	3.51 second	30.51 second
4	3.15 second	24.53 second
5	3.15 second	19.42 second
6	6.89 second	23.57 second
7	2.48 second	19.41 second
8	2.64 second	36.51 second
9	3.28 second	19.63 second
10	3.55 second	30.60 second
RATA-RATA	3.47 second	24.61 second

Tabel 4.39 Hasil tes *load average* pada bandwidth 3MB

Pengujian	Dansguardian off	Dansguardian on
1	3.56 second	24.18 second
2	3.63 second	23.21 second
3	2.56 second	36.41 second
4	3.19 second	24.51 second
5	3.60 second	29.45 second
6	3.17 second	19.35 second
7	3.27 second	19.10 second
8	2.97 second	20.25 second
9	6.49 second	20.33 second
10	3.61 second	30.67 second
RATA-RATA	3.61 second	24.75 second

Tabel 4.40 Hasil tes *load average* pada bandwidth 4MB

Pengujian	Dansguardian off	Dansguardian on
1	2.49 second	36.51 second
2	3.16 second	20.44 second
3	3.50 second	29.41 second
4	3.74 second	21.41 second
5	3.57 second	23.79 second
6	6.43 second	29.51 second
7	4.18 second	24.71 second
8	3.31 second	19.42 second
9	4.18 second	19.35 second
10	3.68 second	23.61 second
RATA-RATA	3.82 second	24.82 second

Tabel 4.41 Hasil tes *load average* pada bandwidth 5MB

Pengujian	Dansguardian off	Dansguardian on
1	3.65 second	21.23 second
2	6.47 second	23.11 second
3	2.99 second	21.90 second
4	4.57 second	30.41 second
5	2.56 second	19.41 second
6	4.81 second	35.81 second
7	2.79 second	23.61 second
8	3.91 second	24.61 second
9	3.89 second	19.14 second
10	3.91 second	29.61 second
RATA-RATA	3.96 second	24.88 second

Tabel 4.42 Rata-rata hasil pengujian *load average* pada seluruh bandwidth

Bandwidth	Dansguardian off	Dansguardian on
64 kbps	2.15 second	22.58 second
128 kbps	2.39 second	22.73 second
256 kbps	2.59 second	23.27 second
384 kbps	2.63 second	23.34 second
512 kbps	2.77 second	23.57 second
640 kbps	2.84 second	23.69 second
768 kbps	2.88 second	23.76 second
896 kbps	2.90 second	23.84 second
1 MB	3.19 second	24.26 second
2 MB	3.47 second	24.61 second
3 MB	3.61 second	24.75 second
4 MB	3.82 second	24.82 second
5 MB	3.96 second	24.88 second

Dari Tabel 4.42 dapat dilihat bahwa *load average* saat menggunakan Dansguardian pada server, pada setiap bandwidth mengalami perbedaan yang signifikan. Namun, peningkatan *load average* tetap stabil dan tidak

signifikan meskipun mengalami perbandingan yang sangat jauh dengan keadaan saat server tidak menggunakan Dansguardian. Jika di hitung presentase peningkatan *load average* pada tiap bandwidth, maka di dapatkan hasil sebagai berikut.

64 kbps - 128 kbps = $\frac{22.73 \text{ s} - 22.58 \text{ s}}{22.58 \text{ S}}$ x 100 = 0.68 %

128 kbps - 256 kbps = $\frac{23.27 \text{ s} - 22.73 \text{ s}}{22.73 \text{ S}}$ x 100 = 2.4 %

256 kbps - 384 kbps = $\frac{23.34 \text{ s} - 23.27 \text{ s}}{23.27 \text{ S}}$ x 100 = 0.3 %

384 kbps - 512 kbps = $\frac{23.57 \text{ s} - 23.34 \text{ s}}{23.34 \text{ S}}$ x 100 = 1 %

512 kbps - 640 kbps = $\frac{23.69 \text{ s} - 23.57 \text{ s}}{23.57 \text{ S}}$ x 100 = 0.5 %

640 kbps - 768 kbps = $\frac{23.76 \text{ s} - 23.69 \text{ s}}{23.69 \text{ S}}$ x 100 = 0.3 %

768 kbps - 896 kbps = $\frac{23.84 \text{ s} - 23.76 \text{ s}}{23.76 \text{ S}}$ x 100 = 0.3 %

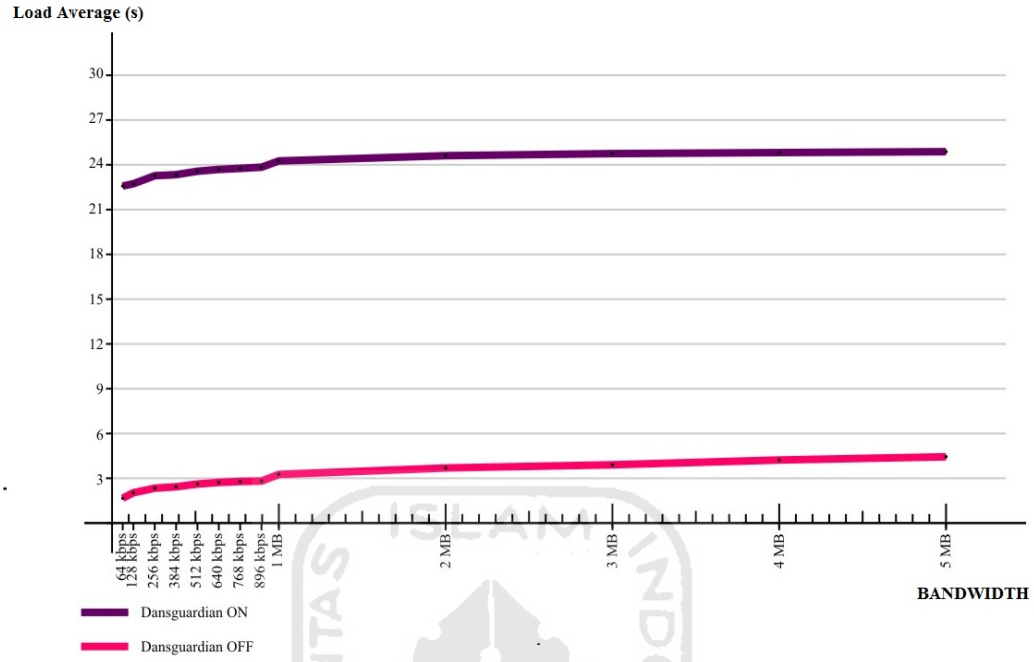
896 kbps - 1 Mbps = $\frac{24.26 \text{ s} - 23.84 \text{ s}}{23.84 \text{ S}}$ x 100 = 1.8 %

1 Mbps - 2 Mbps = $\frac{24.61 \text{ s} - 24.26 \text{ s}}{24.26 \text{ S}}$ x 100 = 1.5 %

2 Mbps - 3 Mbps = $\frac{24.75 \text{ s} - 24.61 \text{ s}}{24.61 \text{ S}}$ x 100 = 0.6 %

3 Mbps - 4 Mbps = $\frac{24.82 \text{ s} - 24.75 \text{ s}}{24.75 \text{ S}}$ x 100 = 0.3 %

4 Mbps - 5 Mbps = $\frac{24.88 \text{ s} - 24.82 \text{ s}}{24.82 \text{ S}}$ x 100 = 0.3 %



Gambar 4.30 Grafik Perbandingan *load average*

Data *load average* baik sebelum maupun sesudah pengimplementasian Dansguardian mengalami perubahan hal ini disebabkan perbedaan penggunaan bandwidth pada jaringan komputer dan pemeriksaan / filtering yang dilakukan oleh Dansguardian.

Peningkatan *load average* dari seluruh bandwidth mencapai hanya berkisar antara 0,3% – 1,24 %. Beradasarkan tabel 4.42 dan hasil perhitungan presentase penigkatan *load average* cenderung stabil dan tidak mengalami peningkatan yang signifikan.

Dansguardian berjalan secara *daemonized*, yang memungkinkan penanganan lalu lintas data yang tinggi tanpa harus banyak memakan *resource* pada server.

Oleh karena itu, meskipun antrian data yang terjadi saat menggunakan Dansguardian, membuat server terbebani dengan jumlah data yang sudah diterima dari internet, namun masih mengantri untuk di filter. Namun,

peningkatan beban pada tetap stabil, walaupun terjadi perbedaan yang sangat jauh saat menggunakan Dansguardian dan sebelum menggunakan Dansguardian.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan analisis dari masing-masing bandwidth dengan parameter rata-rata waktu eksekusi data, *memory usage* dan *load average*, maka dapat ditarik kesimpulan sebagai berikut :

1. Pada Bandwidth 1 Mbps dengan spesifikasi server yang digunakan, Dansguardian mulai menjadi *bottle neck* (penghambat) transfer data pada jaringan komputer karena pada bandwidth ini, perbandingan waktu eksekusi data antara sebelum dan sesudah menggunakan Dansguardian sangat signifikan dan penigkatan waktu eksekusi mencapai 27%.
2. Dansguardian sangat berpengaruh terhadap kecepatan data dalam jaringan komputer karena proses filtering yang dilakukannya.
3. Pada bandwidth dibawah 1Mbps dalam jaringan komputer, perbedaan waktu eksekusi antara sebelum dan sesudah menggunakan Dansguardian tidak signifikan.
4. Penggunaan memory dan *load average* pada server Dansguardian cukup stabil pada semua bandwidth .
5. Jika memiliki bandwidth yang terbatas pada jaringan computer yang tersedia, di sarankan untuk menggunakan Dansguardian pada servernya.

5.2 Saran

Setelah melihat hasil kesimpulan di atas, berikut adalah saran yang mungkin berguna dan dapat digunakan jika ingin melakukan perbandingan efektifitas bandwidth dan resource server pada Dansgaurdian :

1. Menghitung pola perubahan skema penggunaan Dansguardian berdasarkan CPU Power pada server.

DAFTAR PUSTAKA

"____",2011, *DansGuardian true web content filtering for all*,[ONLINE], (*http://dansguardian.org/?page=whatisdg,http://dansguardian.org/?page=dg flow*, di akses tanggal 1 Mei 2011)

Bayuputra, B.2011. *Konfigurasi jarak jauh pemblokiran URL Website menggunakan SMS*. Skripsi, tidak diterbitkan. Yogyakarta: Fakultas Teknologi Industri Universitas Islam Indonesia.

Gunawan,G.,2009, *URL Filter Dengan PROXY:DANSGUARDIAN* [ONLINE],(*http://www.ditpsmk.net/?page=artikel;86&mode=poll&guest_4d 20e77e9e7cc,* diakses tanggal 3 Mei 2011)

Hendrawan, D.2010. *Analisis Kinerja WEB Server Apache, Nginx, HS dan Tomcat Dengan Menggunakan Webserver Stress Tool (Webstress) dan Httperf*. Skripsi, tidak diterbitkan. Yogyakarta: Fakultas Teknologi Industri Universitas Islam Indonesia.

Laboratorium Sistem dan Jaringan Komputer. 2011. *Modul Praktikum Jaringan Komputer*. Yogyakarta. Universitas Islam Indonesia.

Nugroho, L.A.2011. *Remastering Distro Ubuntu Uuntuk Gateway, Distributed Filtering dan Security*. Skripsi, tidak diterbitkan. Yogyakarta: Fakultas Teknologi Industri Universitas Islam Indonesia.

Sujarwo,Y.H. 2009. *Analisis Perbandingan Network Monitoring Server Berbasis Cecti dan Zenoss*. Skripsi, tidak diterbitkan. Yogyakarta: Fakultas Teknologi Industri Universitas Islam Indonesia.

Utama, A.C.,2010, *Filtering Gambar Dan Video Porno Pada Jaringan* [ONLINE] , (*http://www.eepis-its.edu/uploadta/abstrakdetail.php?id=1041*, diakses tanggal 3 Mei 2011)

Wicaksono, A. 2011. *Analisis Keamanan dan Performa Ttransfer Data Pada Wide Area Network Menggunakan Gre Over IPSEC*. Skripsi, tidak diterbitkan. Yogyakarta: Fakultas Teknologi Industri Universitas Islam Indonesia.