

HALAMAN JUDUL

**ANALISIS PERBANDINGAN
CLUSTERING WEB SERVER FULL VIRTUALIZATION DENGAN
CLUSTERING WEB SERVER PARAVIRTUALIZATION
SEBAGAI LOAD BALANCING**

LAPORAN TUGAS AKHIR

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



Disusun Oleh:

Nama: Muhammad Noor Hazman Syarip

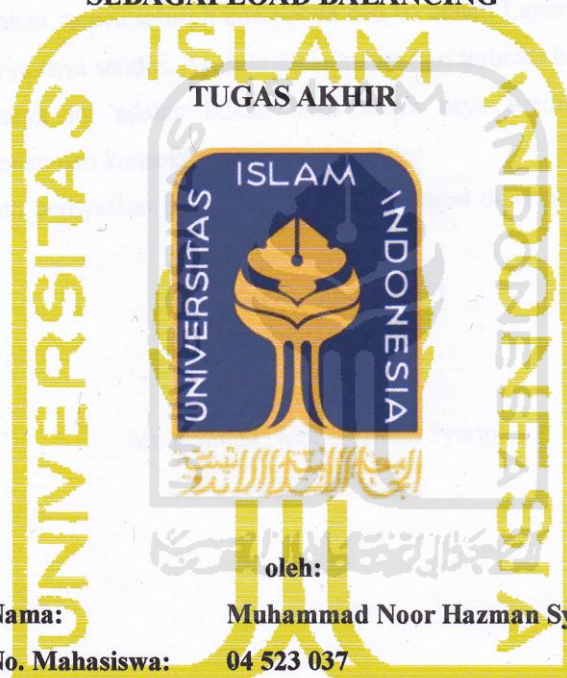
No. Mahasiswa: 04 523 037

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

2011

LEMBAR PENGESAHAN DOSEN PEMBIMBING

**ANALISIS PERBANDINGAN
CLUSTERING WEB SERVER FULL VIRTUALIZATION DENGAN
CLUSTERING WEB SERVER PARAVIRTUALIZATION
SEBAGAI LOAD BALANCING**



oleh:

Nama: Muhammad Noor Hazman Syarip

No. Mahasiswa: 04 523 037

Yogyakarta, Januari 2011

Pembimbing,

A handwritten signature in black ink, which appears to read 'Dirgahayu', is written over a horizontal line.

Teduh Dirgahayu, ST., M.Sc., Ph.D

LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR

Saya yang bertanda tangan dibawah ini,

Nama : Muhammad Noor Hazman Syarip

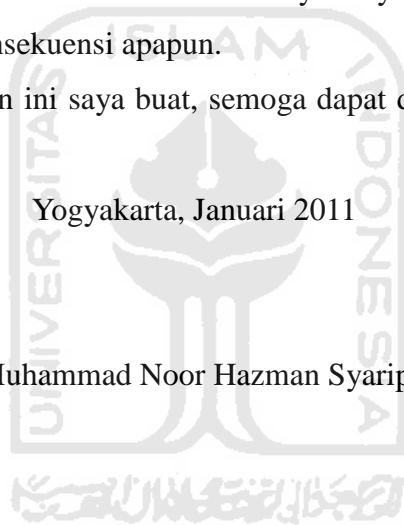
No. Mahasiswa : 04 523 037

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

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

Yogyakarta, Januari 2011

Muhammad Noor Hazman Syarip



LEMBAR PENGESAHAN DOSEN PENGUJI
ANALISIS PERBANDINGAN
CLUSTERING WEB SERVER FULL VIRTUALIZATION DENGAN CLUSTERING
WEB SERVER PARAVIRTUALIZATION
SEBAGAI LOAD BALANCING

TUGAS AKHIR

oleh:

Nama : Muhammad Noor Hazman Syarip

No. Mahasiswa : 04 523 037

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

Tim Penguji,
Teduh Dirgahayu, ST., M.Sc., Ph.D
Ketua

Syarif Hidayat, S.Kom., M.IT
Anggota I

Ari Sujarwo, S.Kom
Anggota II

Mengetahui,

Ketua Jurusan Teknik Informatika



Fakultas Teknologi Industri
Universitas Islam Indonesia

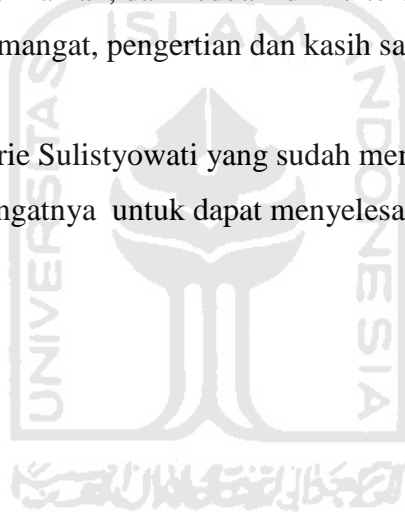
Widi Prayudi, S.Si., M.Kom

HALAMAN PERSEMBAHAN

Kupersembahkan tugas akhirku ini kepada...
ALLAH SWT yang telah memberiku ilmu, rizki dan nikmat-Nya
kepadaku yang tak henti-hentinya...

Ayah, Mamah, dan kedua Adikku tercinta
atas doa, semangat, pengertian dan kasih sayangnya...

Rizky Putrie Sulistyowati yang sudah memberikan
motivasi dan semangatnya untuk dapat menyelesaikan tugas akhir...



HALAMAN MOTTO

“Jangan hina pribadi anda dengan kepalsuan karena
dialah mutiara diri anda yang tak ternilai”

“Sabar dalam mengatasi kesulitan dan bertindak bijaksana dalam mengatasinya
adalah sesuatu yang utama”



KATA PENGANTAR



Alhamdulillah Rabbil'anlamin. Puji syukur kehadirat ALLAH SWT yang telah melimpahkan rahmat, nikmat dan hidayah-Nya. Shalawat dan salam senantiasa tercurah kepada junjungan kita Nabi akhir zaman Rasulullah Muhammad SAW beserta para keluarga, sahabat dan pengikutnya, sehingga dapat kami selesaikan tugas akhir dengan judul **“ANALISIS PERBANDINGAN CLUSTERING WEB SERVER FULL VIRTUALIZATION DENGAN CLUSTERING WEB SERVER PARAVIRTUALIZATION SEBAGAI LOAD BALANCING”**.

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 dan bantuan dari berbagai pihak. Untuk itu pada kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. ALLAH SWT yang selalu mendengarkan doa, harapan dan keluh kesah saya dan memberikan rahmat dan hidayah-Nya.
2. Kedua orang tua saya, Ayah dan Mama yang selalu mendukung dan membimbingku dalam menjalani hidup dan memberikan dorongan baik materi, moril dan kasih sayangnya tiada henti.
3. Bapak Teduh Dirgahayu, ST., M.Sc., Ph.D, dan Bapak Fathul Wahid ST., M.Sc selaku pembimbing saya dalam mengarahkan, membimbing dan meluangkan waktunya selama saya menyelesaikan tugas akhir.
4. Bapak Yudi Prayudi S.Si, M.Kom., selaku Kepala Jurusan, Jurusan Teknik Informatika, Fakultas Teknologi Industri.

5. Seluruh dosen Jurusan Teknik Informatika yang telah mengajar saya mulai dari semester pertama sampai akhirnya saya tutup teori.
6. Rizky Putrie Sulistyowati yang telah memberikan motivasi, semangat dan waktunya untuk tempat berkeluh kesah.
7. Sahabat terbaik 2004, Ino, Toro, Panji, Ari, dan seluruh teman-teman Jurusan Informatika UII 2003, 2004, 2005, 2006, 2007 yang tidak dapat ku sebutkan.
8. Sahabat terbaik Freak Ballers Indonesia, Iman, Erick, Dika, Avis, Goki, Om Sluw, Gaponk, Fahmi, Amri, Seno serta seluruh teman-teman bermain basketku.
9. Sahabat terbaik kawan-kawan Kamera Analog Jogjakarta.
10. Serta seluruh pihak yang telah membantu dalam menyelesaikan tugas akhir ini.

Penulis menyadari bahwa dalam penulisan dan pembuatan tugas akhir ini masih jauh dari kesempurnaan, oleh karena itu penulis menerima kritik dan saran demi kesempurnaannya tulisan ini.

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

Yogyakarta, 2011

Penulis,

Muhammad Noor Hazman Syarip

SARI

Era teknologi informasi seperti sekarang, kebutuhan akan akses Internet menjadi semakin dibutuhkan sebagai sarana komunikasi, bertukar pendapat maupun sebagai media pembelajaran. Perkembangan teknologi informasi ini sangat cepat dengan didukung perkembangan teknologi komputer dan terjangkaunya koneksi Internet. Dengan meningkatnya kebutuhan informasi dalam Internet semakin meningkat pula kinerja atau beban jaringan seperti *web server*.

Salah satu solusi murah yang dapat digunakan adalah dengan membangun *clustering web server* yang digunakan sebagai pembagi beban atau sering disebut *load balancer*. Seiring ditekannya investasi hardware dengan sistem *clustering*, virtualisasi merupakan pilihan yang tepat untuk mendukung investasi hardware tanpa mengeluarkan biaya besar. Adanya beberapa metode virtualisasi yang dapat digunakan dalam membangun server membuat sebuah pertanyaan, lebih baik mana antara *full virtualization* dengan *paravirtualization*.

Hasil akhir dari tugas akhir ini adalah mengetahui perbandingan metode virtualisasi *full virtualization* dengan *paravirtualization* sebagai *load balancing* pada *clustering web server*. Sehingga pengguna virtualisasi akan dapat memilih perangkat virtualisasi yang tepat guna.

Kata kunci : *Load balancing, Clustering, Full Virtualization, Paravirtualization*

TAKARIR

<i>Back end server</i>	Server pendukung server utama
<i>Bottleneck.</i>	Kemacetan jaringan seperti leher botol
<i>Downtime</i>	Lamanya waktu maintenance
<i>Load Balancer</i>	Pembagi beban
<i>Freeware</i>	Perangkat lunak gratis
<i>Guest Computer</i>	Nama lain dari komputer virtual
<i>Hardware.</i>	Perangkat keras
<i>Host Computer</i>	Komputer induk
<i>Hypervisor</i>	Emulator sebagai pembentuk komputer virtual
<i>Interface</i>	Antarmuka
<i>Mainframe</i>	Super komputer
<i>Multitasking</i>	Multi tugas
<i>Open Source</i>	Sumber terbuka
<i>Shareware</i>	Perangkat lunak berbagi
<i>Software</i>	Perangkat lunak
<i>Throughput</i>	Byte yang dikirim dan diterima
<i>Virtual Machine</i>	Mesin virtual
<i>Web Server</i>	Penyedia halaman html

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN DOSEN PEMBIMBING	ii
LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR	iii
LEMBAR PENGESAHAN DOSEN PENGUJI	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTTO	vi
KATA PENGANTAR	vii
SARI	ix
TAKARIR	x
DAFTAR ISI	xi
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xv
BAB I	
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Metodologi Penelitian	3
1.6.1 Metode Pengumpulan Data	3
1.6.2 Metode Implementasi Perangkat	4
1.7 Sistematika Penulisan	4
BAB II	
DASAR TEORI	6
2.1 Virtualisasi	6

2.1.1	Jenis Virtualisasi	8
2.2	Full Virtualization	8
2.3	Paravirtualization	8
2.4	XEN	9
2.5	Web Server	10
2.5.1	Web Server Apache	11
2.6	Cluster Server	12
2.6.1	Cluster Load Balancing	13
 BAB III		
METODOLOGI PENELITIAN		15
3.1	Persiapan Instalasi	15
3.1.1	Perangkat Lunak yang Dibutuhkan	15
3.1.2	Perangkat Keras yang Dibutuhkan	15
3.2	Langkah Instalasi	16
3.3	Arsitektur Jaringan	17
3.4	Pengujian	19
3.1.1	Pengujian Server Tunggal	20
3.1.1	Pengujian Server dengan Load Balancing	20
 BAB IV		
HASIL DAN PEMBAHASAN		21
4.1	Proses Instalasi dan Konfigurasi	22
4.1.1	Instalasi CentOS 5 di Server	22
4.1.2	Instalasi Guest OS di Mesin Virtual	22
4.1.3	Instalasi Web Server	23
4.1.4	Instalasi dan Konfigurasi Load Balancing	26
4.1	Hasil Pengujian	27
4.1.1	Pengujian pada Server Tunggal	27
4.1.1	Pengujian pada Server Load Balancing	29

BAB V

KESIMPULAN	36
5.1 Kesimpulan	36
5.2 Saran	37
DAFTAR PUSTAKA	38
LAMPIRAN	39



DAFTAR TABEL

Tabel 3.1 Tabel Instalasi	17
Tabel 4.1 Koneksi menuju Server Tunggal	29
Tabel 4.2 Koneksi menuju Server Load Balancing	30
Tabel 4.3 Prosentase penghematan <i>paravirtualization</i> terhadap <i>full virtualization</i>	35



DAFTAR GAMBAR

Gambar 3.3.1 Topologi logika web server tunggal	17
Gambar 3.3.2 Topologi logika web server dengan load balancing full virtualization dan paravirtualization	18
Gambar 4.1 Daftar sistem operasi yang sudah diinstal	22
Gambar 4.2 Tampilan halaman web	26
Gambar 4.3 Metrik CPU usage	27
Gambar 4.4 Metrik free memory	28
Gambar 4.5 Metrik user time dan system time	29
Gambar 4.6 Grafik CPU usage	31
Gambar 4.7 Grafik user time	32
Gambar 4.8 Grafik system time	33
Gambar 4.9 Grafik Net I/O	34
Gambar 4.10 Grafik free memory	34

BAB I

PENDAHULUAN

1.1 Latar Belakang

Era teknologi informasi seperti sekarang, kebutuhan akan akses Internet menjadi semakin dibutuhkan sebagai sarana komunikasi, bertukar pendapat maupun sebagai media pembelajaran. Perkembangan teknologi informasi ini sangat cepat dengan didukung perkembangan teknologi komputer dan terjangkaunya koneksi Internet. Dengan meningkatnya kebutuhan informasi dalam Internet semakin meningkat pula kinerja atau beban jaringan seperti *web server*. Untuk mengatasi *down time* akibat peningkatan akses data koneksi ke *web server*, diperlukan perubahan fasilitas yang mampu mengendalikan situasi agar tidak terjadi *down time* yang cukup lama. Cara paling mudah adalah mengganti perangkat *web server* dengan perangkat yang memiliki spesifikasi tinggi yang diharapkan menambah kinerja suatu *web server* namun, solusi di atas membutuhkan investasi yang cukup mahal.

Salah satu solusi murah yang dapat digunakan adalah dengan membangun *clustering web server* yang digunakan sebagai pembagi beban atau sering disebut *load balancer*. Sistem *cluster* ini dibangun dengan membagi beban diantara server – server yang ada di dalam sebuah *cluster*, dengan kata lain komputer *cluster* adalah sekumpulan komputer yang saling terhubung dalam satu jaringan dan bekerja sama untuk mendukung suatu tugas yang biasa dikerjakan oleh komputer tunggal [GOZ02]. Dengan adanya komputer *cluster* ini sebuah tugas diselesaikan dengan mendistribusikan beban pekerjaan ke beberapa komputer. Teknik *clustering* untuk *load balancer* ini memang membutuhkan beberapa mesin komputer namun, spesifikasi mesin yang digunakan tidak harus mesin dengan spesifikasi tinggi, sehingga kita dapat memanfaatkan komputer yang sudah ada untuk membangun teknik *cluster* ini.

Seiring ditekannya investasi hardware dengan sistem *clustering*, virtualisasi merupakan pilihan yang tepat untuk mendukung investasi hardware tanpa

mengeluarkan biaya besar. Virtualisasi dapat diartikan sebagai pembuatan suatu bentuk atau versi virtual dari sesuatu yang bersifat fisik misalnya sistem operasi, perangkat penyimpanan, dan sumber daya jaringan [VAV10]. Dengan dipadunya clustering dengan virtualisasi diharapkan dapat memiliki infrastruktur jaringan yang hemat dan memiliki kinerja maksimal.

1.2 Rumusan Masalah

Berdasar dari latar belakang yang disebutkan di atas dapat diambil rumusan masalah sebagai berikut yaitu:

1. Bagaimana menganalisis perbandingan kinerja *clustering web server* sebagai *load balancing* pada model virtualisasi *full virtualization* dan *paravirtualization*?

1.3 Batasan Masalah

Dalam melakukan analisis perbandingan ini terdapat batasan masalah sebagai berikut:

1. Sistem operasi yang digunakan setiap server adalah Linux Centos 5
2. Virtualisasi menggunakan XEN untuk representasi virtualisasi *full virtualization* dan *paravirtualization*.
3. Secara logika server yang dibutuhkan ada 3 yaitu, sebuah web server virtual dengan load balancing di dalamnya dan didukung dua *backend web server* sebagai komputer pembagi beban.
4. Simulasi *Clustering Web Server* sebagai *Load Balancing* ini masih dalam *Local Area Network (LAN)*
5. Parameter metrik yang digunakan dalam pengukuran kinerja server yaitu *CPU usage*, *user time*, *system time*, dan *Net I/O*

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah menganalisis dan membandingkan kinerja virtualisasi web server dengan metode *full virtualization* dengan *paravirtualization* pada *clustering web server* sebagai *load balancing*.

1.5 Manfaat Penelitian

Dengan adanya penelitian ini dapat memberikan manfaat penggunaan virtualisasi pada web server untuk membangun *clustering web server* sebagai *load balancing* untuk meminimalkan biaya investasi dan menghindari sistem dari *bottleneck*.

1.6 Metodologi Penelitian

1.6.1 Metode Pengumpulan Data

Pada metode pengumpulan literatur ini, metode pengumpulan meliputi metode studi pustaka. Metode studi pustaka yaitu mengumpulkan literatur yang berkaitan dengan penelitian ini seperti buku, majalah, maupun jurnal yang berkaitan dengan penelitian. Metode pengumpulan data juga dapat melalui media Internet dengan tujuan mencari perangkat lunak yang akan digunakan dalam web server dan metode virtualisasi agar sistem berjalan dengan baik.

1.6.2 Metode Implementasi Perangkat

Metode implementasi ini disusun berdasar hasil yang didapat setelah mengumpulkan literatur. Metode ini meliputi:

a. Pengadaan perangkat keras.

Persiapan instalasi dilakukan dengan menyediakan perangkat keras dan perangkat lunak yang akan digunakan. Perangkat keras yang digunakan yaitu komputer yang terdiri dari sebuah web server dan dua buah komputer sebagai

backend server. Perangkat lunak yang digunakan adalah sistem operasi dan perangkat lunak untuk mendukung sebuah web server.

b. Instalasi dan konfigurasi.

Tahapan ini merupakan tahap instalasi sistem operasi, konfigurasi web server dan membangun *clustering* yang akan digunakan untuk *load balancing*

c. Pengujian

Parameter yang digunakan dalam pengujian adalah pengukuran prosesor, memori dan jaringan. Untuk pengujian dilakukan uji T dengan 30 kali pengujian pada server tunggal dan server *load balancing*. Pengujian dilakukan menggunakan perangkat lunak *httperf* agar menjamin beban yang diterima tiap server sama.

1.7 Sistematika Penulisan

Dalam penelitian ini sistematika penulisan dibagi menjadi lima bab yang terdiri dari:

Bab 1 menjelaskan mengenai latar belakang permasalahan yang akan dibahas. Pendahuluan meliputi latar belakang masalah, batasan masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika penulisan.

Bab 2 membahas tentang dasar – dasar teori yang digunakan dalam penelitian yang meliputi pengertian web server, virtualisasi, clustering server dan load balancing.

Bab3 menjelaskan langkah penelitian dari menjabarkan perangkat lunak dan perangkat keras yang digunakan serta pengumpulan data untuk mengkonfigurasi server yang kemudian dilakukan pengujian.

Bab 4 membahas konfigurasi dan instalasi komputer server dengan metode virtualisasi *full Virtualization* dan *paravirtualization*. Dalam bab ini juga membahas tentang membangun komputer *clustering* yang nantinya akan digunakan sebagai *load balancing*. Pada tahap pengujian akan dibahas mengenai analisis dan perbandingan kinerja *clustering web server full virtualization* dengan *clustering web server*

paravirtualization dengan melakukan simulasi menggunakan aplikasi *httperf*.

Bab 5 memuat kesimpulan yang merupakan rangkuman dari hasil analisis dan perbandingan kinerja *clustering web server full virtualization* dengan *clustering web server paravirtualization*. Saran untuk memperbaiki atau mengembangkan penelitian ini juga dimuat pada bab ini.



BAB II

DASAR TEORI

2.1 Virtualisasi

Virtualisasi adalah salah satu cara untuk memangkas biaya server dan sistem informasi di suatu perusahaan. Karena tidak harus memiliki beberapa mesin server berspesifikasi tinggi maka virtualisasi semakin menjadi pilihan menarik bagi dunia usaha dan organisasi mulai dari yang skala kecil hingga skala besar. Virtualisasi komputer ini sudah digunakan sejak tahun 1960 dan telah diaplikasikan ke beberapa aspek komputer. Secara teknis teknologi virtualisasi ini adalah menyembunyikan detail teknis komputer melalui enkapsulasi [VMW06]. Dengan kata lain virtualisasi komputer adalah komputer yang berjalan di dalam komputer yang di implementasikan oleh perangkat lunak yang berfungsi untuk mengemulasikan keseluruhan sistem hardware mulai dari prosesor sampai network interfacenya [VAV 10].

Untuk mendukung virtualisasi ini tentunya harus didukung dengan *software* yang mendukung virtualisasi dari suatu sistem operasi. Perangkat lunak yang mendukung virtualisasi sekarang ini sudah banyak dapat ditemui dan di unduh secara gratis. Sistem virtualisasinya pun sudah dapat digunakan di berbagai platform baik Windows, Unix maupun Linux. Banyak pilihan perangkat lunak virtualisasi yang dapat digunakan, untuk yang berbayar dapat menggunakan VMWare yang tersedia untuk Windows dan Linux dan untuk yang gratis dapat menggunakan XEN, Sun VirtualBox, Qemu dan VMWare ESXi. Walaupun dengan adanya perbedaan antar berbayar dan gratis produk virtualisasi tersebut harus dapat mendukung virtualisasi yang berbasis prosesor Intel maupun AMD. Teknologi pada prosesor inilah yang memungkinkan sistem virtual pada komputer dapat mengakses prosesor secara langsung seperti tanpa menggunakan virtualisasi. Sehingga sebuah mesin virtual memiliki kecepatan, fitur dan kompatibilitas yang sama dengan komputer yang berada sebagai komputer induk dalam mesin virtual [VMW06].

Setiap teknologi pada prosesor ini memiliki nama yang berbeda tergantung pada vendor prosesor. Teknologi yang digunakan Intel adalah *Virtualization Technology* (Intel VT), sedangkan untuk AMD memiliki teknologi virtualisasi yang disebut *AMD Virtualization* (AMD-V) [VMW06]. Ada beberapa macam virtualisasi yang dapat diimplementasikan kedalam beberapa bentuk, antara lain [VAV10]:

Network Virtualization.

Memory Virtualization.

Grid Computing.

Storage Virtualization.

Platform Virtualization.

Pada penelitian yang dilakukan oleh VMware pada usaha kecil dan menengah (UKM), virtualisasi memberikan keuntungan sebagai berikut [VMW09]:

- 2 Meminimalkan dan memudahkan kerja *maintenance* para IT Administrator, karena sistem virtual hanyalah versi virtual dari server sebenarnya.
- 3 Kemudahan backup dan proteksi data. Server virtual dapat disimpan pada satu *image* yang berisi dari konfigurasi sistem sehingga apabila ada salah satu sistem yang rusak dapat dengan mudah menyalin *image* yang sudah *backup* sebelumnya.
- 4 Memudahkan mengembangkan aplikasi tanpa takut sistem menjadi rusak karena bekerja dalam ruang lingkup virtual.
- 5 Kemudahan untuk merespon perubahan kebutuhan bisnis.
- 6 Pengurangan biaya investasi *hardware*.

2.1.2 Jenis Virtualisasi

Dalam sebuah mesin yang digunakan untuk virtualisasi dibagi menjadi dua, yaitu *Host Computer* yang merupakan komputer induk yang memiliki *virtual machine* dan *Guest Computer* sebagai komputer virtual yang berjalan pada komputer induk. Untuk mensimulasikan sebuah mesin kedalam virtualisasi diatas sistem virtualisasi sebuah *virtual machine* memiliki beberapa teknik yang digunakan, yaitu [LIN06]:

1. *Full Virtualization*. Virtualisasi penuh seluruh hardware yang digunakan dalam mesin virtual sehingga menyerupai sebuah mesin yang sebenarnya.
2. *Paravirtualization*. Teknik ini merupakan teknik virtualisasi yang memiliki interface perangkat lunak pada mesin virtual yang sama namun tidak identik dengan mesin yang sebenarnya.
3. *Emulation*. Teknik pada *virtual machine* ini membolehkan suatu program komputer berjalan disebuah platform yang berbeda baik secara arsitektur komputer maupun platformnya.

2.2 Full Virtualization

Full virtualisasi adalah virtualisasi penuh seluruh hardware yang digunakan dalam mesin virtual sehingga menyerupai sebuah mesin yang sebenarnya. Pada virtualisasi penuh, *virtual machine* menyediakan layanan fisik komputer seperti perangkat memori, kartu jaringan, termasuk juga bios virtual. Hasilnya adalah sebuah sistem yang mampu mengeksekusi semua perangkat lunak pada perangkat keras yang bisa dijalankan pada *virtual machine* [VMW07]. Produk yang mengembangkan virtualisasi penuh di Linux salah satunya adalah VirtualBox dan XEN.

2.3 Paravirtualization

Paravirtualisasi adalah upaya untuk memberikan layanan langsung dari hardware. Teknik ini merupakan teknik virtualisasi yang memiliki interface perangkat lunak pada mesin virtual yang sama namun tidak identik dengan mesin yang sebenarnya [LIN06]. *Interface* yang diubah pada teknik ini untuk mengurangi porsi

waktu eksekusi *guest OS* dalam melakukan operasi yang berjalan dalam lingkungan virtual dibandingkan lingkungan non-virtual [VMW07]. Teknik *Paravirtualization* menyediakan kait untuk memungkinkan *Guest OS* dan *Host OS* untuk meminta dan mengakses tugas-tugas dalam domain virtual. Salah satu software yang mendukung paravirtualisasi adalah XEN, dimana sebelumnya fasilitas virtualisasi dengan metode *paravirtualization* ini ditawarkan oleh VM IBM pada tahun 1972. XEN merupakan sebuah *virtual machine* yang dikembangkan oleh University of Cambridge. Saat ini perkembangan XEN sudah mulai ditanamkan pada distribusi Linux seperti pada OpenSUSE dan RedHat [LIN06].

2.4 XEN

Teknologi virtualisasi yang dikembangkan oleh Keir Fraser dan Ian Pratt ini merupakan aplikasi virtualisasi yang dapat menggunakan 2 metode, yaitu *full virtualization* dan *paravirtualization*. XEN dapat beroperasi pada berbagai arsitektur CPU tanpa mendukung virtualisasi penuh. Aplikasi ini berjalan diantara perangkat keras dan sistem operasi yang dinamakan *hypervisor*. Lapisan abstraksi tersebut memungkinkan setiap *host computer* untuk menjalankan mesin virtualnya. Proyek *open source* yang dikembangkan pertama kali di Universitas Cambridge ini, teknologi virtualisasinya sudah dikembangkan oleh beberapa vendor internasional seperti AMD, Intel, Cisco, Dell, HP, IBM, Mellanox, Network Appliance, Novell, Red Hat, Samsung, SGI, Sun, Unisys, Veritas, Voltaire, dan Citrix. Dengan dukungan hardware yang baik yang mendukung hampir semua driver perangkat Linux, teknologi XEN dapat menjalankan sampai 32 CPU virtual dari setiap mesin virtualnya [TAK10].

2.5 Web Server

Web server merupakan tulang punggung dari *world wide web* atau yang sering disingkat dengan WWW. Sebuah web server merupakan pelayan bagi *client* yang meminta sebuah halaman web yang diakses melalui *web browser*. Tidak hanya berupa

teks, *web server* juga dapat memberikan permintaan gambar dari *client*. Komunikasi antara web server dan *client* ini menggunakan protokol yang disebut HTTP (*Hyper Text Transfer Protocol*) yang akhirnya dapat sampai pada web browser berupa teks maupun gambar yang diminta. Secara garis besar web server hanya memproses masukan yang diperolehnya dari web clientnya [AZM02].

Halaman web pastilah disusun sedemikian rupa sehingga dapat dinikmati oleh clientnya. Untuk membuat halaman – halaman web tersebut dalam sebuah web server dapat diisi dengan halaman yang memiliki format HTML, SGML, PHP, PHP3, CGI dan lain sebagainya, tergantung dengan kebutuhan web server yang akan dikelola. Membangun web server yang handal haruslah memiliki perangkat lunak dan perangkat keras yang memadai. Salah satu hal yang paling penting untuk diperhatikan adalah pemilihan perangkat lunak. Untuk memilih perangkat lunak ada beberapa pertimbangan sebagai berikut [AZM02]:

1. Lisensi dari perangkat lunak yang digunakan.
2. Kemudahan instalasi, konfigurasi dan perawatannya
3. Kemampuan perangkat lunak dalam mengelola web server.
4. Besar *memory* dan sumber daya yang dibutuhkan oleh perangkat lunak tersebut agar dapat berfungsi dengan baik.
5. Prospek perangkat lunak di masa yang akan datang.
6. Dukungan teknis dan dokumen yang diberikan oleh perangkat lunak apabila terjadi kerusakan.

Saat ini perangkat lunak web server yang memiliki kriteria di atas sudah banyak di temui di Internet dari yang berbayar, *shareware*, dan *freeware*. Dari sekian banyak perangkat lunak web server yang tersedia salah satunya adalah web server Apache yang merupakan perangkat lunak web server gratis yang memiliki dukungan teknis dan dokumentasi yang lengkap serta *powerful*.

2.5.1 Web Server Apache

Web server Apache bisa dikatakan web server yang paling banyak dipakai di dunia. Program ini pertamakali didesain untuk sistem operasi pada lingkungan UNIX. Apache dimulai dari seorang developer NCSA httpd (National Center Supercomputing Application) yaitu Rob McCool. Nama Apache diambil dari kata *Apache* server perbaikan yang penuh perbaikan atau *patch* dari NCSA httpd v1.3. Saat ini web server Apache sudah dapat digunakan pada Windows dan semakin dipergunakan secara luas. Hal lain yang menjadikan Apache banyak digunakan adalah programnya yang gratis dan relatif stabil [AZM02].

Apache mempunyai program pendukung dalam sebuah web server guna memberikan layanan yang baik kepada penggunanya, antara lain[AZM02]:

1. Kontrol Akses

Kontrol ini dapat dijalankan berdasarkan nama host atau nomor IP

2. CGI (*Common Gateway Interfaces*)

CGI mengatur layanan komunikasi *client* dan server. Web server Apache didukung oleh PERL (*Practical Extraction and Report Language*) yang ditempatkan oleh Apache sebagai modul yang dikenal dengan modul *mod_perl*.

3. PHP (*Personal Home Page/PHP Hypertext Processor*)

PHP adalah bahasa pemrograman web yang dapat bekerja pada web server Apache yang memiliki kemampuan untuk memproses data dinamis yang ditempatkan Apache pada modul *mod_php*.

4. SSI (*Server Side Includes*)

Server-side includes digunakan untuk menyisipkan dokumen atau mengeksekusi dokumen tertentu dan menampilkan hasilnya. Sebagai contoh dengan SSI ini dapat menampilkan informasi tanggal dan waktu secara otomatis sesuai setting waktu system.

2.6 Cluster Server

Yang dimaksud dengan komputer *cluster* adalah sekumpulan komputer yang berada dalam satu jaringan dan bekerja sama dalam mengatasi tugas atau beban yang biasa ditangani oleh komputer tunggal. Di dalam suatu komputer *cluster* terdapat satu komputer sebagai komputer *cluster manager* dan beberapa komputer lainnya sebagai *cluster node*. *Cluster manager* adalah sebagai pebagi tugas kepada komputer node atau cluster node untuk diselesaikan sesuai dengan konfigurasi yang telah dilakukan sebelumnya. Ada beberapa jenis *clustering* yaitu [GOZ02]:

1. *Shared processing*, yaitu sistem cluster yang menggabungkan beberapa kemampuan komputer yang ada sehingga memberikan kemampuan komputasi yang lebih besar untuk menyelesaikan suatu pekerjaan.
2. *Load balancing*, adalah tipe clustering yang mirip dengan tipe *shared processing*, namun perbedaannya adalah setiap *cluster node* hanya menyelesaikan tugas yang diberikan *cluster manager* sebagai pembagi beban tugas yang sudah didistribusikan.
3. *Failover*, tipe dari cluster yang berfungsi sebagai pengambil alih proses dari sebuah node yang *down*, sehingga proses yang diminta terselesaikan dengan baik.
4. *High availability*, metode ini dapat dicapai apabila seluruh sistem yang diimplementasikan diatas dijaga baik dari hardware maupun software.
5. Sistem ini dapat bekerja sebagai *shared processing*, *load balancing* dan *failover*.

2.6.1 Cluster Load Balancing

Dalam *clustering* jaringan komputer, *load balancing* adalah sebuah teknik untuk mendistribusikan beban kerja secara merata di dua atau lebih komputer, link

jaringan, CPU, hard drive, atau sumber daya lainnya, untuk mendapatkan pemanfaatan sumber daya yang optimal, memaksimalkan *throughput*, meminimalkan waktu respon, dan menghindari *overload*.

Load balancing adalah sebuah konsep yang berfungsi sebagai penyeimbang beban pada infrastruktur TI. Layanan load balancing ini memungkinkan akses sumberdaya dalam jaringan dapat didistribusikan ke beberapa host lain sehingga beban tidak terpusat dan kerja jaringan semakin ringan. Solusi yang ada pada *load balancing* adalah dengan membagi beban. Hal ini umumnya digunakan untuk komunikasi internal di cluster komputer, terutama untuk server cluster. Jika beban berlebih pada sebuah server, maka beban server sekunder mengambil beberapa sementara yang lain masih memproses permintaan Adapun kelebihan load balancing antara lain [RIJ05]:

1. Menjamin reliabilitas layanan sehingga pekerjaan dapat dilakukan dengan baik dan lancar.
2. Skalabilitas ketersediaan sumberdaya meningkatkan kerja server sehingga tidak terjadi *downtime* yang lama.

Dalam server yang baik terdapat beberapa parameter-parameter yang memiliki matrik-matrik yang dapat diukur, seperti[CIL07]:

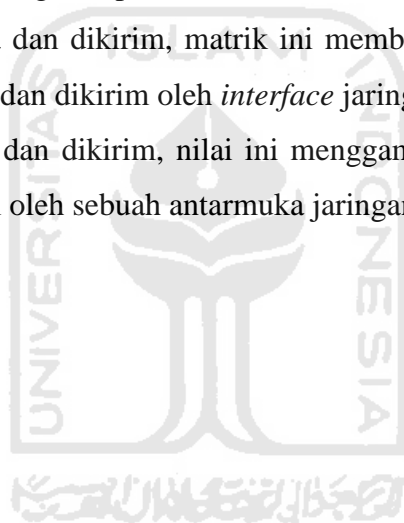
1. Parameter prosesor, berikut adalah matrik yang ada dalam prosesor:
 - a) *CPU utilization*, yaitu pemanfaatan keseluruhan prosesor. Apabila penggunaan lebih dari 80% secara berkelanjutan dapat dipastikan prosesor tersebut mengalami *bottleneck*.
 - b) *User time*, melihat presentasi CPU yang digunakan dalam penggunaan sistem saat bekerja.
 - c) *System time*, pada metrik ini menggabungkan presentase CPU yang bekerja di dalam kernel, semakin baik sistem semakin kecil waktu yang dihabiskan di dalam kernel untuk menghindari kemacetan jaringan.
 - d) *Load average*, yaitu nilai rata-rata *task running* dan *task interruptable* proses pada CPU.

2. Parameter pada *memory* meliputi metrik:

- a) *Free memory*, yang berarti memori yang tidak terpakai atau memori bebas.
- b) *Swap usage*, yang berkaitan dengan memori virtual untuk mencegah terjadinya *bottleneck*.
- c) Memori aktif dan memori tidak aktif, dengan melihat ini maka dapat diketahui informasi tentang pemakaian memori dalam sebuah sistem.

3. Parameter jaringan yang meliputi:

- a) Paket yang diterima dan dikirim, matrik ini memberitahukan tentang jumlah paket yang diterima dan dikirim oleh *interface* jaringan.
- b) Byte yang diterima dan dikirim, nilai ini menggambarkan jumlah byte yang diterima dan dikirim oleh sebuah antarmuka jaringan yang diberikan.



BAB III

METODOLOGI PENELITIAN

3.1 Persiapan Instalasi

3.1.1 Perangkat Lunak yang Dibutuhkan

Perangkat lunak yang digunakan dalam analisis ini adalah sebagai berikut:

1. Linux CentOS 5 sebagai sistem operasi.
2. XEN sebagai representasi *full virtualization* dan *para virtualization*.
3. Apache sebagai web server.
4. Paket Balance-3.35-1.el.rf.i386.rpm
5. Httperf, untuk mengukur kinerja web server.

3.1.2 Perangkat Keras yang Dibutuhkan.

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

1. Komputer Server

Komputer server adalah komputer yang dapat digunakan sebagai web server tunggal dan sebagai *host operating system* pada mesin virtual *full virtualization* dan *paravirtualization*, perangkat keras sebagai berikut:

- a) Prosesor dengan kecepatan 3,01 Ghz.
- b) Memori 1 GB DDR3
- c) Media penyimpanan sebesar 40 GB.
- d) DVD-ROM.

2. *Real Back end server 1*

Komputer *back end server* digunakan sebagai komputer pembagi beban web server *load balancing*, dengan spesifikasi sebagai berikut:

- a) Prosesor dengan kecepatan 2,7 Ghz.
- b) Memori 1 GB DDR2

- c) Media penyimpanan sebesar 20GB.
- d) Kartu jaringan.

3. *Real Back end server 2*

Komputer *back end server* digunakan sebagai komputer pembagi beban web server *load balancing*, dengan spesifikasi sebagai berikut:

- a) Prosesor dengan kecepatan 2.0 Ghz.
- b) Memori 512 MB DDR2
- c) Media penyimpanan sebesar 20 GB.
- d) Kartu jaringan.

4. *Server load balancer full virtualization* dan *paravirtualization*

Server virtual ini berfungsi sebagai server pengatur pembagian beban menuju *back end server*, kedua server *load balancing* memiliki spesifikasi sebagai berikut:

- a) Memiliki 1 Core Prosesor bagian dari *Host Operating System*.
- b) Memori 512 MB
- c) Media penyimpanan 10 GB

5. Switch

3.2 Langkah Instalasi

Berikut ini adalah instalasi yang diperlukan dalam setiap komputer.

Tabel 3.1 Tabel Instalasi.

Server Utama	<i>Back End Server 1</i>	<i>Back End Server 2</i>
1. OS Linux CentOS 5 2. XEN	1. OS Linux CentOS 5 2. Apache web server	1. OS Linux CentOS 5 2. Apache web server

3.3 Arsitektur Jaringan

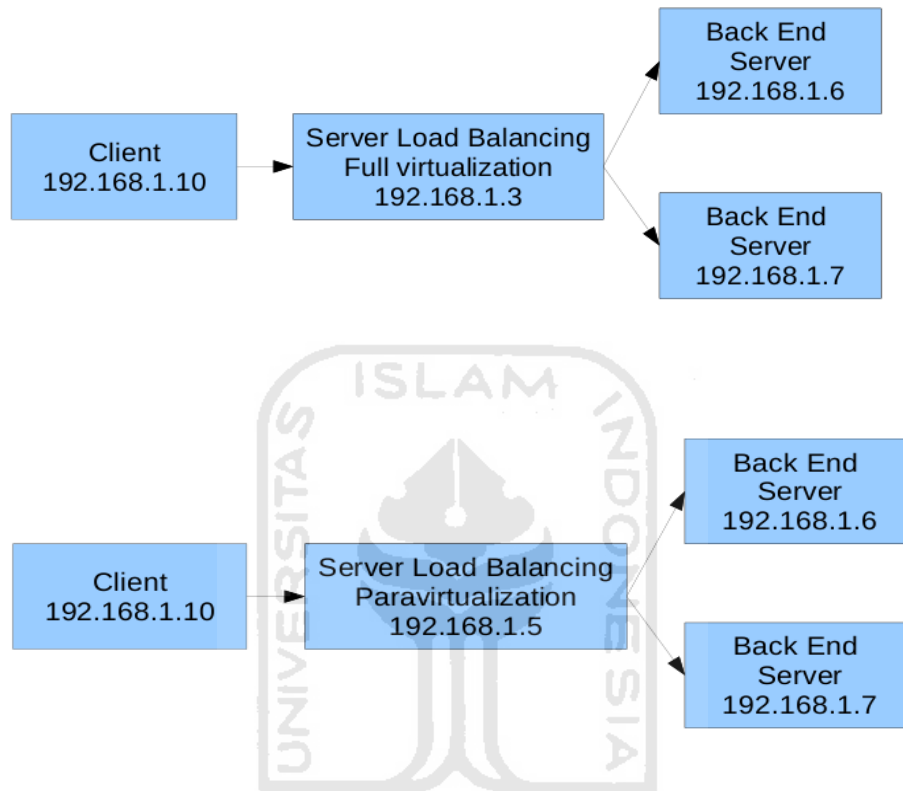
Pada perancangan arsitektur jaringan *load balancing* menggunakan tiga buah server. Server pertama adalah server utama dimana instalasi web server virtual dengan metode *full virtualization* dan *paravirtualization* dilakukan. Untuk mendukung proses *load balancing* digunakan pula dua buah *back end server* yang sudah diinstal *apache web server*. Server-server tersebut masih dalam satu jaringan lokal yang digunakan untuk simulasi perbandingan kinerja *load balancing* virtual menggunakan *httperf* pada PC penguji. Secara logika dan fisik digambarkan seperti berikut:

1. Topologi Logika Server Tunggal



Gambar 3.3.1. Topologi Server Tunggal, Virtual Web Server dengan Full

2. Topologi Logika *Virtual Web Server* dengan *Full Virtualization* dan *Paravirtualization*



Gambar 3.3.2. Topologi Virtual Web Server dengan Full Virtualization dan Paravirtualization

3.4 Pengujian

Pengujian yang akan dilakukan dalam tugas akhir ini dilakukan pada web server virtual yang digunakan sebagai server load balancing pada jaringan lokal. Pengujian dilakukan berdasarkan parameter prosesor, memori dan jaringan. Pengujian dilakukan secara bertahap dari 100 *client* sampai 1000 *client* dengan kenaikan bertahap. Setiap pengujian dilakukan sebanyak 30 kali kemudian diambil nilai rata-rata menggunakan Uji T. Metrik prosesor yang akan diukur adalah *CPU usage*, *user time* dan *system time*, untuk parameter memori adalah memori yang tidak aktif kemudian byte I/O pada parameter jaringan.

Cara Pengujiannya adalah sebagai berikut:

1. Prosesor,
Metrik pada parameter prosesor yang diuji adalah:
 - a) *CPU usage* yang dapat dilihat dari monitor sistem yang terdapat pada virtual machine manager.
 - b) *System time* akan diuji dengan program *httperf*.
 - a) *User time* yang akan diuji dengan program *httperf*.
2. Memory
Dalam parameter memori metrik yang dilihat adalah:
 - a) Memori tidak terpakai, dengan perintah dasar linux “top”
3. Jaringan
Metrik yang akan dilihat pada parameter jaringan adalah:
 - a) Byte I/O yang akan dilihat melalui program *httperf*.

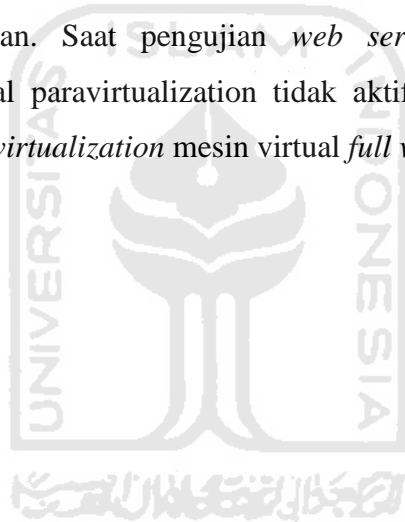
Panggilan dilakukan oleh *client* yang memiliki IP 192.168.1.10 menuju web server *load balancer* yang ditanam pada komputer *real server* di alamat IP 192.168.1.100 dimana masing-masing memiliki alamat IP 192.168.1.3 untuk *web server load balancer full virtualization* dan pada alamat IP 192.168.1.5 untuk *web server load balancer para virtualization* yang kemudian dibantu oleh dua *back end server* yang memiliki alamat IP 192.168.1.6 dan 192.168.1.7.

3.4.1 Pengujian Server Tunggal

Pengujian ini bertujuan untuk menguji kemampuan web server virtual tanpa didukung oleh *back end server*. Pada server tunggal *service httpd* harus dijalankan agar server dapat menampilkan halaman web.

3.4.2 Pengujian Server dengan Load Balancing.

Pengujian ini bertujuan untuk menguji kemampuan *load balancing* pada *web server load balancing full virtualization* dan *paravirtualization* yang terhubung dengan 2 *real back end server*. Pada proses pengujian ini *service httpd* pada kedua server virtual dinonaktifkan. Saat pengujian *web server load balancing full virtualization*, mesin virtual *paravirtualization* tidak aktif apabila pengujian *web server load balancing paravirtualization* mesin virtual *full virtualization* tidak aktif.



BAB IV

HASIL DAN PEMBAHASAN

4.1 Proses Instalasi dan Konfigurasi

Langkah langkah yang dilakukan dalam instalasi dan konfigurasi daam pembuatan tugas akhir ini adalah sebagai berikut:

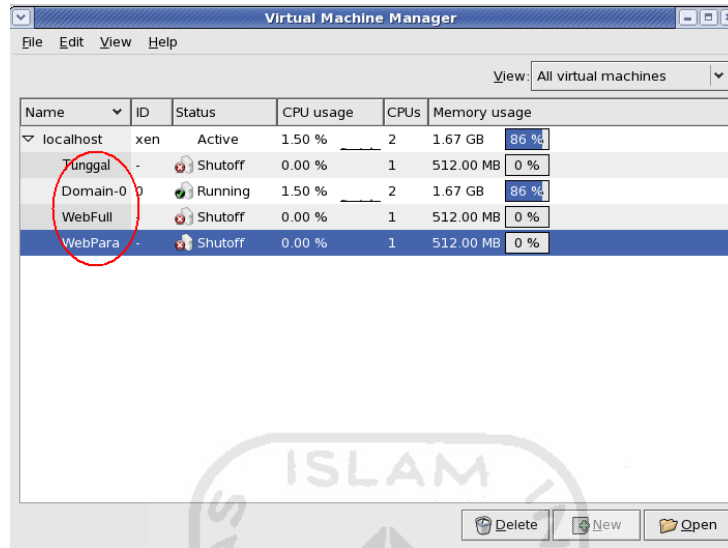
4.1.1 Instalasi CentOS 5 di Server

CentOS 5 adalah sistem operasi yang digunakan pada penelitian ini. Sistem operasi tersebut menjadi sistem operasi server utama dan 2 *back end server*. Berikut adalah langkah instalasi CentOS 5. Proses instalasi dapat dilakukan dari mode teks maupun grafis.

4.1.2 Instalasi Guest OS di Mesin Virtual

Mesin virtual pada CentOS 5 sudah terinstal apabila kita memilih paket server dan paket virtualisasi pada saat proses instalasi. XEN adalah mesin virtual yang akan digunakan pada penelitian ini. Langkah langkah konfigurasi mesin virtual XEN adalah sebagai berikut:

1. Pertama kali cek apakah XEN sudah benar terinstal pada saat instalasi dengan baik.
2. Apabila sudah terinstall langkah selanjutnya adalah membuat mesin virtual dengan metode *full virtualization* dan *paravirtualization*.
3. Proses instalasi *guest operating system* pada XEN akan ditanyakan beberapa informasi, yaitu nama mesin virtual, metode virtualisasi, lokasi *storage* atau lokasi instalasi mesin virtual, yang terakhir memori dan alokasi CPU pada mesin virtual. Untuk instalasi dengan metode *full virtualization* dapat dilakukan melalui CD-ROM maupun *iso image*. sedangkan untuk metode *paravirtualization* hanya bisa dilakukan melalui media *network install tree* (FTP, HTTP, atau NFS).
4. Apabila sukses pada *interface Virtual Machine Manager* akan muncul nama mesin virtual yang sudah diinstal seperti gambar 4.1



Gambar 4.1 Daftar sistem operasi yang sudah diinstal

- Konfigurasi IP pada mesin virtual. Pada penelitian ini mesin virtual dengan metode *full virtualization* diberi IP 192.168.1.3 dan mesin virtual dengan metode *paravirtualization* diberi IP 192.168.1.5. file yang dikonfigurasi adalah `/etc/sysconfig/network-scripts/eth0`, tambahkan files seperti dibawah ini:

```
#nano /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
BOOTPROTO=none
HWADDR=40:61:86:f4:c5:0a
ONBOOT=yes
DHCP_HOSTNAME=xnor.web.id
IPADDR=192.168.1.5
NETMASK=255.255.255.0
TYPE=Ethernet
```

4.1.3 Instalasi Web Server

Langkah yang dilakukan untuk instalasi web server adalah sebagai berikut:

- Apache web server* sudah disediakan dalam paket instalasi Linux CentOS.

Untuk melihat apakah *apache* sudah terinstal pada terminal ketikkan perintah sebagai berikut:

```
# rpm -qa | grep httpd
httpd-2.2.3-43.el5.centos
system-config-httpd-1.3.3.3-1.el5
httpd-manual-2.2.3-43.el5.centos
```

2. Apabila tampilan pada terminal tidak menunjukkan baris diatas maka apache belum diinstal dengan baik. Untuk Instalasi secara online cukup ketikkan perintah pada terminal sebagai berikut:

```
#yum install httpd
```

Setelah instalasi selesai jalankan dengan perintah pada terminal:

```
#!/sbin/service httpd start
Starting httpd: [ OK ]
```

3. Buat halaman `index.html` untuk streaming video dengan nano dengan konfigurasi seperti di bawah ini:

```
#nano /var/www/html/index.html
<html>
<head>
<script type="text/javascript" src="flowplayer-
3.2.4.min.js">
</script>
<link rel="stylesheet" type="text/css" href="style.css">
<title>Video Lucu</title>
</head>
<body>
<div id="page">
<h1>Music Humor</h1>
<p>Music Video OnLine</p>
<a href="http://192.168.1.100/Screenshot.jpg"
style="display:block;width:520px;height:330px"
```

```

id="player"> </a>
<script>flowplayer("player", "../flowplayer-3.2.5.swf");
</script>
</div>
</body>
</html>

```

5. Agar tidak terlalu statis halaman web dipercantik dengan sedikit tampilan CSS. Buat script cssnya pada /var/www/html sebagai berikut:

```

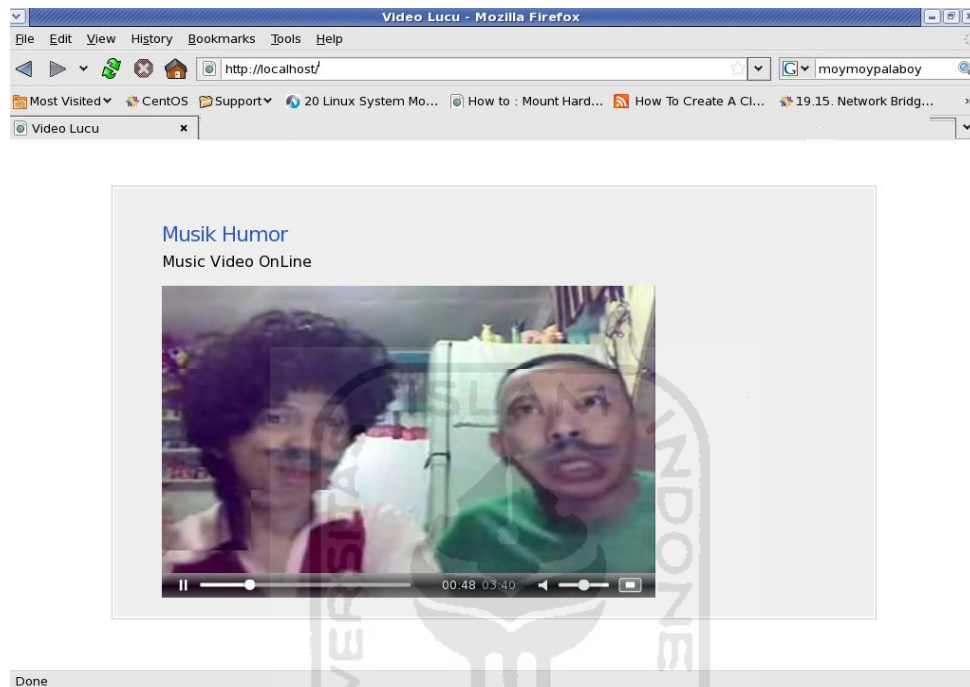
#nano /var/www/html/style.css
body {
background-color:#fff;
font-family:"Lucida Grande","bitstream vera
sans","trebuchet ms",verdana,arial;
text-align:center;
}
#page {
background-color:#efefef;
width:600px;
margin:50px auto;
padding:20px 150px 20px 50px;
min-height:600px;
border:2px solid #fff;
outline:1px solid #ccc;
text-align:left;
}
h1, h2 {
letter-spacing:-1px;
color:#2D5AC3;
font-weight:normal;
margin-bottom:-10px;

```

```
}  
h1 {  
font-size:22px;  
}  
h2 {  
font-size:18px;  
}  
.less {  
color:#999;  
font-size:12px;  
}  
a {  
color:#295c72;  
}
```



Hasil dari pemrograman web menggunakan HTML diatas adalah seperti gambar pada gambar 4.2 dibawah ini:



Gambar 4.2 Tampilan halaman web

4.1.4 Instalasi dan Konfigurasi Load Balancing Web Server pada Virtual Machine

Langkah yang dilakukan untuk instalasi *load balancing* pada mesin virtual adalah sebagai berikut:

1. *Download* paket untuk melakukan load balancing pada website <http://packages.sw.be/balance/>
2. Untuk melakukan load balancng ketikkan perintah dibawah ini:

```
#/usr/sbin/balance -f http 192.168.1.6 192.168.1.7
```
3. Untuk melihat koneksi apakah backend server sudah terdaftar ketikkan perintah dibawah ini:

```
#/usr/sbin/balance -f -d http 192.168.1.6 192.168.1.7
```

4.2 Hasil Pengujian

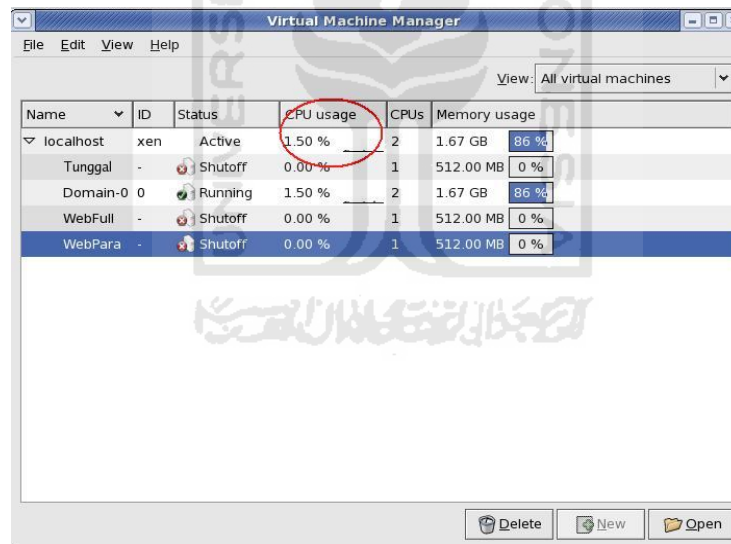
Pengujian dilakukan berdasarkan parameter yang telah ditentukan sebelumnya. Pengujian dilakukan pada web server tunggal dan web server full virtualization dan paravirtualization sebagai load balancing. Berikut ini adalah hasil analisis yang telah dilakukan:

4.2.1 Pengujian pada Server Tunggal

Parameter yang diuji pada server tunggal adalah parameter prosesor, *memory* dan jaringan. Berikut ini adalah metrik metrik yang diuji pada setiap parameter.

1. Metrik *CPU usage*.

Metrik *CPU usage* dilihat melalui *Virtual Machine Manager*. Pengujian metrik cpu usage dilakukan saat program *httperf* dijalankan dan diambil nilai tertinggi saat program *httperf* dijalankan.



Name	ID	Status	CPU usage	CPUs	Memory usage
localhost	xen	Active	1.50 %	2	1.67 GB 86 %
Tunggal	-	Shutoff	0.00 %	1	512.00 MB 0 %
Domain-0 0	-	Running	1.50 %	2	1.67 GB 86 %
WebFull	-	Shutoff	0.00 %	1	512.00 MB 0 %
WebPara	-	Shutoff	0.00 %	1	512.00 MB 0 %

Gambar 4.3 Metrik CPU usage

2. Metrik *free memory*

Untuk melihat metrik ini dapat dilakukan dengan menggunakan perintah *top* pada terminal Linux.

```

xnor@xnor:~
File Edit View Terminal Tabs Help
top - 10:22:32 up 37 min, 2 users, load average: 0.32, 0.20, 0.12
Tasks: 163 total, 1 running, 162 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.8%us, 0.0%sy, 0.0%ni, 98.7%id, 0.3%wa, 0.0%hi, 0.0%si, 0.2%st
Mem: 1755136k total, 752792k used, 1002344k free, 36260k buffers
Swap: 1966072k total, 0k used, 1966072k free, 496988k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM     TIME+  COMMAND
 3838 root        15   0 108m  21m  7168  S   1.3   1.2   0:40.41 Xorg
 4116 root        15   0 114m  26m  11m   S   0.3   1.6   0:29.11 /usr/share/virt
11823 xnor        15   0 2352 1056  804  R   0.3   0.1   0:00.03 top
   1 root        15   0 2088  644  556  S   0.0   0.0   0:00.34 init
   2 root        RT  -5   0    0    0  S   0.0   0.0   0:00.01 migration/0
   3 root        34  19   0    0    0  S   0.0   0.0   0:00.00 ksoftirqd/0
   4 root        RT  -5   0    0    0  S   0.0   0.0   0:00.00 watchdog/0
   5 root        RT  -5   0    0    0  S   0.0   0.0   0:00.00 migration/1
   6 root        34  19   0    0    0  S   0.0   0.0   0:00.00 ksoftirqd/1
   7 root        RT  -5   0    0    0  S   0.0   0.0   0:00.00 watchdog/1
   8 root        10  -5   0    0    0  S   0.0   0.0   0:00.01 events/0
   9 root        10  -5   0    0    0  S   0.0   0.0   0:00.00 events/1
  10 root        10  -5   0    0    0  S   0.0   0.0   0:00.00 khelper
  11 root        10  -5   0    0    0  S   0.0   0.0   0:00.00 kthread
  13 root        17  -5   0    0    0  S   0.0   0.0   0:00.00 xenwatch
  14 root        17  -5   0    0    0  S   0.0   0.0   0:00.00 xenbus
  17 root        10  -5   0    0    0  S   0.0   0.0   0:00.02 kblockd/0

```

Gambar 4.4 Metrik *free meory*

3. Metrik *user time* dan *system time*
 Metrik *user time* dapat dilihat dari hasil output program *htpperf*
4. Metrik Net I/O
 Metrik Net I/O atau yang dikenal juga dengan *throughput* dapat dilihat dari hasil output program *htpperf*.

```

r=4096 --recv-buffer=16384 --num-conns=100 --num-calls=1
Maximum connect burst length: 1

Total: connections 100 requests 100 replies 100 test-duration 1.238 s

Connection rate: 80.7 conn/s (12.4 ms/conn, <=1 concurrent connections)
Connection time [ms]: min 0.1 avg 12.4 max 1017.2 median 0.5 stddev 102.3
Connection time [ms]: connect 0.3
Connection length [replies/conn]: 1.000

Request rate: 80.7 req/s (12.4 ms/req)
Request size [B]: 66.0

Reply rate [replies/s]: min 0.0 avg 0.0 max 0.0 stddev 0.0 (0 samples)
Reply time [ms]: response 12.1 transfer 0.0
Reply size [B]: header 245.0 content 444.0 footer 0.0 (total 689.0)
Reply status: 1xx=0 2xx=100 3xx=0 4xx=0 5xx=0

CPU time [s]: user 0.32 system 0.87 (user 25.8% system 70.1% total 95.9%)
Net I/O: 59.5 KB/s (0.5*10^6 bps)

Errors: total 0 client-timo 0 socket-timo 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0
[root@xnor xnor]#

```

Gambar 4.5 Metrik *user time* dan *system time*

Pada tabel dibawah ini dapat dilihat tabel hasil output yang sudah diuji dengan metode uji t dengan metrik *CPU usage*, *user time*, *system time*, dan NET I/O pada kondisi 100 koneksi sampai 1000 koneksi.

Tabel 4.1 Koneksi menuju server tunggal

Metrik	Mean dari 30 kali pengujian				
	100 Koneksi	300 Koneksi	500 Koneksi	700 Koneksi	1000 Koneksi
<i>CPU usage</i>	3,0183	4,4730	4,5503	4,6443	4,9
<i>System time</i>	0,840	0,2140	0,3627	0,4983	5,92
<i>User time</i>	0,5790	1,7467	2,9453	4,1467	0,73
<i>Net I/O</i>	119,0167	119,1500	119,1500	119,157	119,158

4.2.2 Pengujian Server Load Balancing

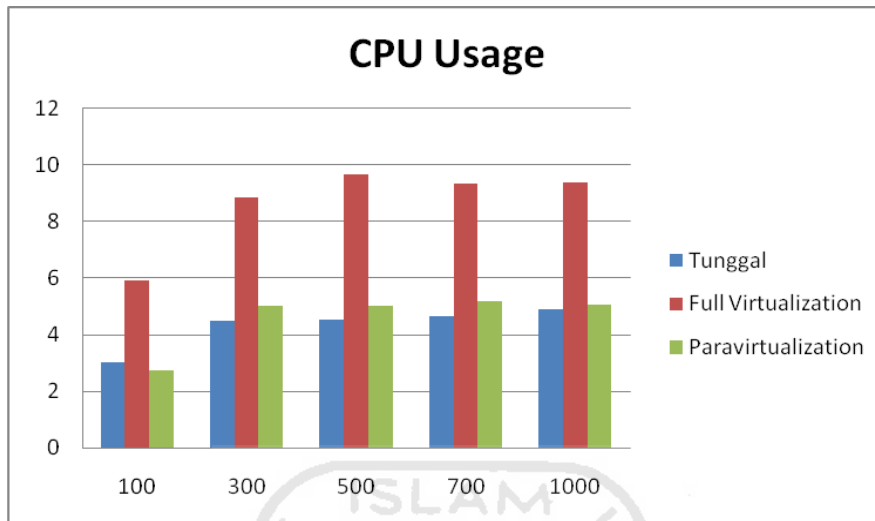
Pada pengujian server load balancing dilakukan pada server load balancing yang dipasang pada mesin virtual full virtualization dan paravirtualization. Pengujian dilakukan sebanyak 30 kali secara bertahap dari 100 koneksi sampai 1000 koneksi http dalam jaringan lokal dengan menggunakan program *httperf*. Proses pengujian dapat dilihat pada grafik yang dibawah ini. Metrik yang diukur yaitu *CPU usage*, *user time*, *system time*, NET I/O dan memory yang tidak terpakai. Untuk pengukuran memori dilakukan melalui terminal Linux dengan perintah *top* karena metrik memori

tidak ditampilkan pada keluaran *httperf*. Setelah dilakukan pengujian, kemudian dilakukan Uji T yang keluarannya ditampilkan seperti pada tabel 4.2 dari metrik CPU usage, user time, system time dan Net I/O.

Tabel 4.2 Koneksi menuju server *load balancing*

Metrik	Mean dari 30 kali pengujian									
	Full Virtualization					Paravirtualization				
	100 Koneksi	300 Koneksi	500 Koneksi	700 Koneksi	1000 Koneksi	100 Koneksi	300 Koneksi	500 Koneksi	700 Koneksi	1000 Koneksi
<i>CPU Usage</i>	5,9210	8,8577	9,6653	9,3427	9,3967	2,7400	5,0110	5,0123	5,1680	5,0767
<i>System time</i>	0,5843	1,7657	2,9467	4,1680	5,9677	0,5777	1,7637	2,9527	4,1150	5,9213
<i>User time</i>	0,730	0,2197	0,3467	0,4887	0,6730	0,763	0,2183	0,3577	0,5000	0,7150
<i>Net I/O</i>	113,0	113,033	113,0	113,00	113,00	113,0	113,200	113,00	113,00	113,00

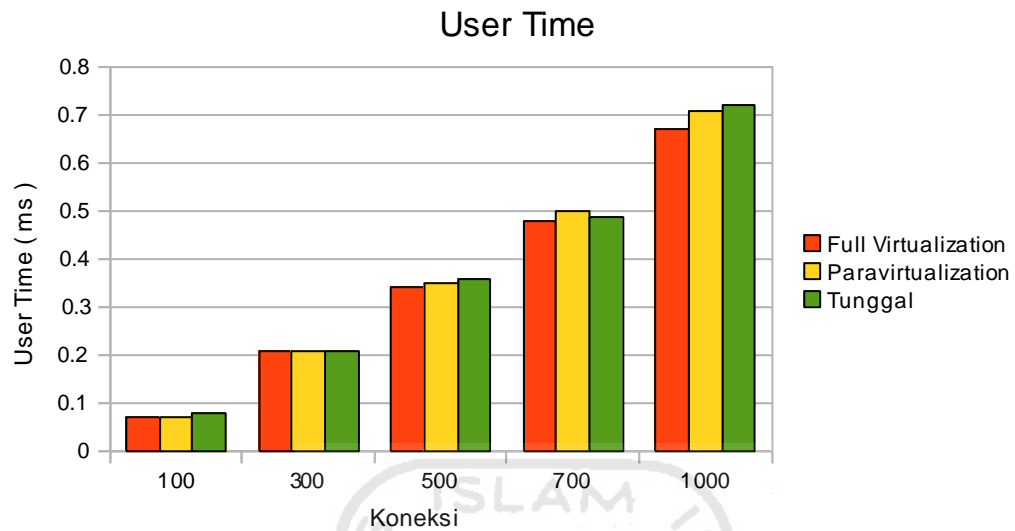
Hasil Uji T dari 100 koneksi sampai 1000 koneksi menuju web server load balancer dapat dilihat untuk koneksi terendah yaitu 100 koneksi menuju *load balancer full virtualization*, membutuhkan rata-rata 5,9210% penggunaan CPU, 0,0730 ms pada *user time*, 0,5843 ms pada *system time* dan menghasilkan net i/o sebesar 113 Kbps sedangkan untuk *paravirtualization* membutuhkan 2,740% penggunaan CPU, 0,763 ms pada *user time*, 0,5777 ms pada *system time* dan menghasilkan NET I/O sebesar 113 Kbps. Dari tabel diatas dapat dilihat pula uji t dari 300 koneksi sampai dengan 1000 koneksi menuju *load balancer* dengan *full virtualization* maupun *paravirtualization*. Dari tabel diatas dapat dilihat bahwa metode virtualisasi *paravirtualization* lebih hemat penggunaan CPU, *user time* dan *system time* yang lebih singkat daripada virtualisasi dengan *full virtualization*, walaupun menghasilkan Net I/O yang hampir sama. Selain dengan tabel, grafik metrik *CPU usage*, *user time*, *system time* dan Net I/O untuk 100 koneksi http menuju *web server load balancer* adalah sebagai berikut:



Gambar 4.6 Grafik *CPU usage*

Pada metrik *CPU usage* hal yang perlu diperhatikan adalah penggunaan *CPU* tidak boleh lebih dari 80% karena akan memicu terjadinya *bottleneck* [CIL07]. Dari grafik *CPU usage* diatas dapat diketahui bahwa penggunaan *CPU* pada virtualisasi dengan metode *paravirtualization* lebih baik dari metode *full virtualization*. Karena semakin kecil penggunaan *CPU usage* meminimalkan terjadinya *bottleneck* pada sistem. Untuk mencari berapa besar prosentase *mean* perbandingan *CPU usage* antara virtualisasi dengan metode *full virtualization* dengan *paravirtualization* dapat digunakan rumus sebagai berikut:

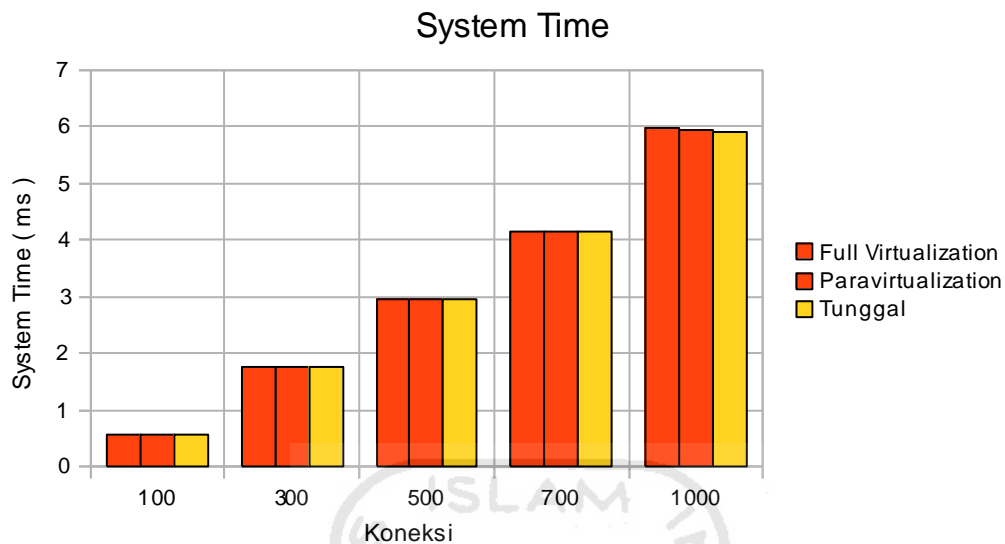
$$\frac{\text{cpu usage full virtualization} - \text{cpu usage paravirtualization}}{\text{cpu usage full virtualization}} \times 100\% \quad (4.1)$$



Gambar 4.7 Grafik User time

Gambar 4.7 menunjukkan grafik user time pada tahapan 100 koneksi sampai 1000 koneksi pada server dengan virtualisasi metode *full virtualization*, *paravirtualization* dan server tunggal. *User time* yang baik adalah *user time* yang memiliki waktu yang tinggi yang menandakan suatu sistem bekerja dengan baik [CIL07]. Untuk menghitung prosentase perbandingan *mean user time* dapat dihitung dengan menggunakan rumus sebagai berikut:

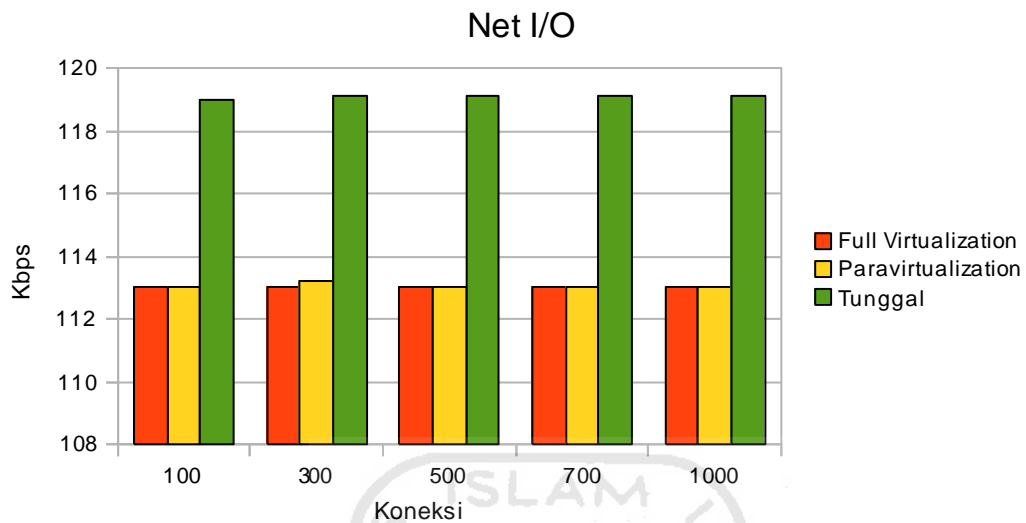
$$\frac{\text{user time paravirtualization} - \text{usertime full virtualization}}{\text{cpu usage full virtualization}} \times 100\% \quad (4.2)$$



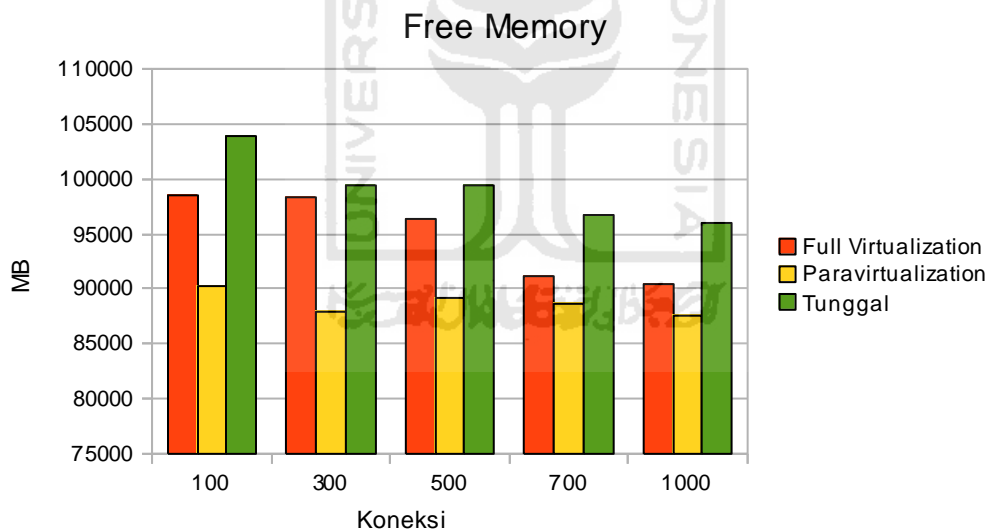
Gambar 4.8 Grafik system time

System time adalah waktu yang dibutuhkan oleh kernel dalam melakukan pemrosesan. Tingginya waktu pada system time dapat menimbulkan *bottleneck* pada jaringan dan membuat *driver stack* [CIL07]. Dapat dilihat pada grafik diatas system time paravirtualisasi membutuhkan waktu yang kecil dibandingkan dengan system time pada metode *full virtualisasi*. Prosentase perbandingan *mean* antara kedua metode virtualisasi tersebut dapat dihitung dengan rumus sebagai berikut:

$$\frac{\text{system time full virtualization} - \text{system time paravirtualization}}{\text{system time full virtualization}} \times 100\% \quad (4.3)$$



Gambar 4.8 Grafik Net I/O



Gambar 4.9 Grafik free memory

Pada grafik Net I/O atau yang biasa disebut dengan *throughput* kedua metode virtualisasi menghasilkan output yang sama. Namun, pada grafik metrik *free memory* kembali terdapat perbedaan antara virtualisasi pada metode *full virtualization* dengan metode *paravirtualisasi*. Dapat dilihat pada grafik *free memory*, *full virtualization* memiliki *free memory* lebih banyak dibanding dengan *paravirtualization*. Prosentase

perbandingan mean *free memory* dari kedua metode virtualisasi tersebut dapat dihitung dengan rumus sebagai berikut:

$$\frac{\text{free memory full virtualization} - \text{free memory paravirtualization}}{\text{free memory full virtualization}} \times 100\% \quad (4.4)$$

Dari perhitungan prosentase *CPU usage*, *user time*, dan *system time*, dapat diketahui bahwa virtualisasi dengan metode paravirtualisasi lebih baik dalam penggunaan masing-masing metrik. Pada table 4.3 dibawah dapat dilihat hasil prosentase penghematan pada virtualisasi metode *paravirtualization*.

Tabel 4.3 Prosentase penghematan *paravirtualization* terhadap *fullvirtualization*

Koneksi	<i>CPU usage</i>	<i>User time</i>	<i>System time</i>	<i>Free memory</i>
100 Koneksi	53,7 %	4,1%	1,7%	-8,8%
300 Koneksi	43,3%	0%	0,1%	-12%
500 Koneksi	48,1%	2,9%	0,3%	-7,8%
700 Koneksi	44,7%	4,1%	0,2%	-3%
1000 Koneksi	46%	5,9%	6,7%	-3%

Tabel diatas menunjukkan *clustering web server load balancing paravirtualization* menghemat cpu usage antara 43,3% sampai 53,7%. Untuk user time dapat menghemat 2,9% sampai 5,9% dan yang terakhir *clustering web server load balancing paravirtualization* menghemat 0,2% sampai 6,7% pada system time. Dari tabel diatas *free memory* pada *full virtualization* menghemat dari 3% sampai 12% penggunaan memory pada saat terjadi koneksi bertahap dari 100 sampai 1000 koneksi.

BAB V

KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil penelitian *clustering load balancing web server dengan full virtualization dan paravirtualization* maka dapat ditarik kesimpulan sebagai berikut:

1. Implementasi *clustering load balancing web server* pada mesin virtual *full virtualization* dan *paravirtualization* berhasil diterapkan dan tidak menutup kemungkinan apabila digunakan dalam jaringan internet.
2. Menurut pengujian besarnya CPU usage, *cluster load balancing dengan paravirtualization* lebih sedikit menghabiskan CPU usage. Pada metode *paravirtualization* dapat menghemat sampai 53,7% penggunaan CPU.
3. Dari hasil pengujian user time tampak pemrosesan pada *paravirtualization* lebih baik sampai 5,9% dibandingkan dengan *full virtualization*.
4. Nilai system time pada metode virtualisasi *paravirtualization* menghemat sampai 6,7% dari nilai system time metode virtualisasi *full virtualization*
5. Penggunaan memori pada load balancer web server *full virtualization* lebih sedikit dibandingkan dengan metode *paravirtualization*. Hal ini terbukti dari hasil pengujian dari 100 sampai 1000 koneksi menuju kedua web server tersebut.
6. Setelah dilakukan pengujian antara kedua metode virtualisasi tersebut dalam membangun *clustering web server load balancing*, metode virtualisasi *paravirtualization* lebih baik daripada metode virtualisasi *full virtualization*, karena kernel *guest OS* pada *paravirtualization* menyerupai kernel *host OS* sehingga mesin *guest OS* *paravirtualization* dapat berkomunikasi layaknya menggunakan hardware dari *host OS*.
7. Memory yang tidak terpakai dari *full virtualization* terbukti lebih baik sampai 12%, hal ini terjadi karena memory pada metode *full virtualization* berdiri sendiri berbeda dengan *paravirtualization* yang masih berbagi *memory* dengan

Host OSnya.

8. Melihat pengujian web server tunggal dengan *web server cluster server load balancing* kinerja yang dihasilkan masih lebih baik *web server* tunggal. Adapun perbedaan ini dikarenakan penelitian ini baru pada tahap menggunakan *load balancing* yang berjalan pada tingkat aplikasi dan keterbatasan perangkat yang ada.

5.2 Saran

Melihat kesimpulan diatas berikut ini adalah saran yang mungkin dapat digunakan untuk penelitian selanjutnya adalah bagaimana cara untuk meningkatkan performa *web server cluster load balancing* pada lingkungan virtual agar lebih baik dibandingkan server tunggal.

