

**ANALISIS PERBANDINGAN KINERJA VIRTUAL SERVER  
DENGAN DEDICATED SERVER BERBASIS TEMPLATE  
UNIX DARI CACTI**

**LAPORAN TUGAS AKHIR**

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



**Disusun oleh:**

**Nama : Dimas Kuncoro**

**No. Mahasiswa : 05 523 011**

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

**2010**

# **HALAMAN JUDUL**

## **ANALISIS PERBANDINGAN KINERJA VIRTUAL SERVER DENGAN DEDICATED SERVER BERBASIS TEMPLATE UNIX DARI CACTI**

### **LAPORAN TUGAS AKHIR**

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



**Disusun oleh:**

**Nama : Dimas Kuncoro**

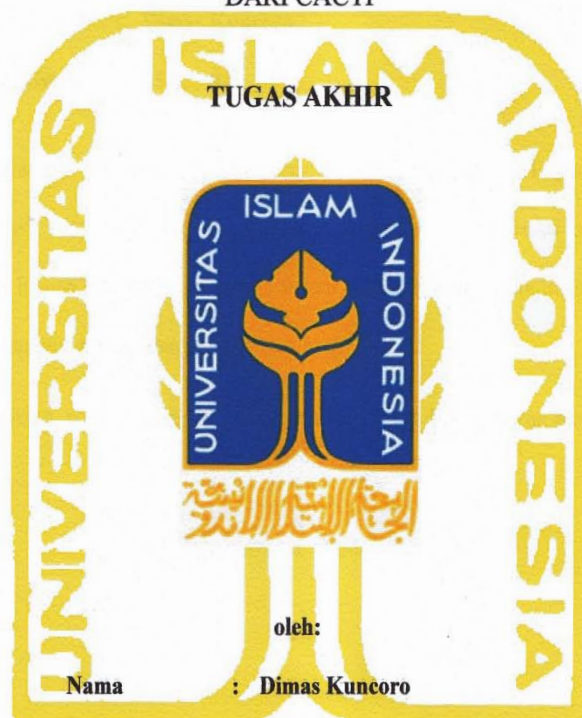
**No. Mahasiswa : 05 523 011**

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

**2010**

**LEMBAR PENGESAHAN DOSEN PEMBIMBING**

**ANALISIS PERBANDINGAN KINERJA VIRTUAL SERVER  
DENGAN DEDICATED SERVER BERBASIS TEMPLATE UNIX  
DARI CACTI**



oleh:  
Nama : Dimas Kuncoro

No. Mahasiswa : 05 523 011

Yogyakarta, 30 Januari 2010

Pembimbing,

  
Irving Vitra Papatungan

## **LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR**

Saya yang bertandatangan di bawah ini,

**Nama : Dimas Kuncoro**

**No. Mahasiswa : 05 523 011**

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 Januari 2010

Dimas Kuncoro

**LEMBAR PENGESAHAN DOSEN PENGUJI****ANALISIS PERBANDINGAN KINERJA VIRTUAL SERVER DENGAN  
DEDICATED SERVER BERBASIS TEMPLATE UNIX DARI CACTI****TUGAS AKHIR**

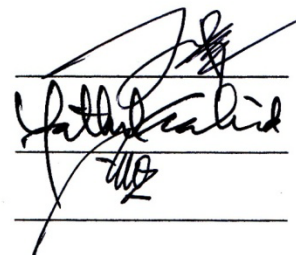
oleh:

**Nama : Dimas Kuncoro****No. Mahasiswa : 05 523 011**

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

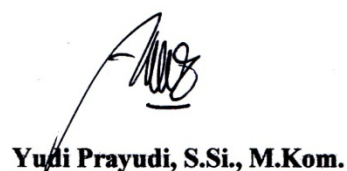
Yogyakarta, 30 Januari 2010

Tim Penguji,  
Irving Vitra Papatungan, ST., M.Sc.  
Ketua  
Fathul Wahid, ST., M.Sc.  
Anggota I  
Yudi Prayudi, S.Si., M.Kom.  
Anggota II



Mengetahui,

Ketua Jurusan Teknik Informatika  
Fakultas Teknologi Industri  
Universitas Islam Indonesia



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

## **HALAMAN PERSEMBAHAN**

Tugas akhir ini aku persembahkan kepada . . .

Allah SWT yang telah memberikan limpahan berkah dan rahmat, sehingga aku bisa menyelesaikan kewajibanku ini dengan baik..

Ayah dan Ibuku tercinta yang telah memberikan segala perhatian dan doanya selama penulisan tugas akhir ini..

Puput Dewanthy dan Puteri Artiningtyas, saudari-saudariku yang telah memberikan dorongan dan semangat untuk menyelesaikan tugas akhir ini..

Semua sobat-sobatku atas dukungan dan bantuan kalian dari awal aku kuliah sampai selesainya penulisan tugas akhir ini..

Kayra sebagai tempat aku menyegarkan diri dari kejenuhan-kejenuhan selama pengerjaan tugas akhir ini.. 1 2 3 4 Kay!!

## **HALAMAN MOTTO**

"Nothing impossible in this life"

"Keep hope, keep attempt, and keep pray "

- KAY

## KATA PENGANTAR



Alhamdulillah Rabbil'alamin. Puji dan syukur kehadiran Allah SWT yang telah melimpahkan rahmat, nikmat dan hidayah-Nya. Shalawat serta salam senantiasa tercurah kepada junjungan kita Rasulullah Muhammad SAW beserta para keluarga, sahabat serta para pengikutnya, sehingga terselesaikannya tugas akhir dengan judul **“ANALISIS PERBANDINGAN KINERJA VIRTUAL SERVER DENGAN DEDICATED SERVER BERBASIS TEMPLATE UNIX DARI CACTI”**

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. Yang tercinta Ayahanda, Aryanto dan Ibunda, Suhartini serta saudari-saudariku, Puput Dewanthy dan Puteri Artiningtyas, semoga Allah SWT membalas amal budi mereka dengan kasih sayang yang berlimpah.
2. Bapak Fathul Wahid, ST., M.Sc., sebagai Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.

3. Bapak Yudi Prayudi, S.Si., M.Kom., selaku Ketua Jurusan, Jurusan Teknik Informatika, Fakultas Teknologi Industri.
4. Bapak Irving Vitra Papatungan, selaku pembimbing yang telah meluangkan waktu untuk berkonsultasi selama penyusunan tugas akhir.
5. Sobat-sobatku dari Spala, Rendy, Rizal, Dody, Doko, Febri, Anggo, Didik, Ratman, Alief, Yudha, Iwan, Budi, Toni, Wisnu, Alm. Hairul, Vinah, Maya, Sendy, dll.
6. Sobat-sobatku dari Smanza, Usan, Poji, Sengget, Maradian, Diding, Bosari, Bombay, Vena, Tripang, Bima, Ridho, Impact 07', dll.
7. Sobat-sobatku dari ALIEN 05', anak-anak kelas A 05' khususnya.
8. Rekan-rekan KKN unit 45 angkatan 36.
9. Sobat-sobat yang banyak membantu dalam proses penyelesaian tugas akhir ini, khususnya Mas Fahmi, Asisten Lab SisJarkom, Haried, Yayak, Aris dipsenk, dan Yopi.
10. Bolang-bolang kontrakan Sulawesi4 7D, Rangga Sela, Booedy, Teguh Anakost, Rexa cihuy, Coach Omen, Bayu, ST., Fadli Datar, Ridwan Tan Malaka , Hasan '9'.
11. Rika Yuliani Safitri binti Jumono, my endless love.
12. Serta semua pihak yang telah turut membantu hingga selesainya penyusunan tugas akhir ini. Semoga Allah SWT membalas budi baik dan keikhlasannya, Amin.

Dalam penyusunan laporan ini, penulis menyadari masih banyak kekurangan sehingga masukan dan saran yang bersifat membangun demi menambah wacana dan wawasan keilmuan saat ini dan masa yang akan datang sangat diharapkan guna kesempurnaan laporan yang telah dibuat. Untuk itu, penulis mengucapkan terimakasih yang sebesar-besarnya kepada semua pihak yang telah membantu atas terselesaikannya tugas akhir ini, semoga Allah SWT membalas semua amal kebbaikannya.

Akhir kata, penulis mengharapkan semoga laporan ini dapat berdaya guna dan memberikan suatu manfaat yang sebesar-besarnya bagi kita semua.

Yogyakarta, 30 Januari 2010

Penulis

Dimas Kuncoro

## SARI

Tingginya inflasi dan melemahnya ekonomi dunia jadi penyebab banyak perusahaan terpaksa melakukan berbagai pemangkasan, mulai dari pengurangan jumlah karyawan hingga pengurangan biaya perjalanan bisnis, sehingga tidak mengejutkan bila para pimpinan perusahaan juga dituntut menekan pengeluaran serta memaksimalkan investasi *information technology* (TI) perusahaan.

*Virtualization* adalah tren yang belakangan banyak dibicarakan yang juga merupakan cara tepat untuk memangkas biaya *server* dan TI perusahaan. Jika sebelumnya departemen TI mengatur *server* berdasarkan fungsi dan kebutuhan, misalnya ada *server* khusus untuk *email*, dokumen, akunting dan bahkan *server* khusus untuk *printing*. Dengan virtualisasi, semua fungsi tersebut dapat dilakukan antar mesin sehingga jauh lebih efisien. Untuk skala kecil, penggunaan virtualisasi diprediksi menurunkan konsumsi listrik minimal 25 persen dibandingkan tanpa virtualisasi. Sementara untuk skala besar penghematan energi mencapai 80 persen. Karena itu, virtualisasi kini menjadi pilihan menarik bagi dunia usaha dan organisasi mulai dari yang berukuran kecil hingga besar. Selama masa krisis masih berlangsung, virtualisasi masih menjadi solusi yang relevan untuk membantu perusahaan tetap bertahan.

Dari hasil analisis setelah melakukan pengujian terhadap dua jenis infrastruktur *server* yang berbeda, dapat ditarik kesimpulan bahwa Virtual server mempunyai keunggulan dari Dedicated server, yaitu penggunaan sumber daya yang lebih rendah tetapi memberikan performa yang sama baik. Hal itu berarti infrastruktur Virtual server sangat tepat untuk instansi yang mengusung konsep *green bussines*, yaitu memaksimalkan investasi di bidang TI dan melakukan penghematan energi demi menjaga kelestarian lingkungan hidup.

Kata Kunci : *green bussines, information technology, virtualization, virtual server*

## TAKARIR

<i>All in one</i>	Mencakup semua dalam satu paket
<i>Benchmark</i>	Proses pengukuran dan perbandingan kinerja
<i>Data center</i>	Pusat penyimpanan data
<i>Dedicated server</i>	Server yang berdiri sendiri dalam satu mesin
<i>Downtime</i>	Waktu selama sistem mati
<i>Green business</i>	Konsep hemat energi
<i>Hot standby</i>	Metode penyimpanan data secara fisik
<i>Memory</i>	Tempat penyimpanan informasi proses
<i>Multitasking</i>	Beberapa proses bekerja bersamaan
<i>Server</i>	Sistem penyedia layanan pada jaringan komputer
<i>Virtual</i>	Maya/bukan yang sebenarnya
<i>Virtualization</i>	Membuat sesuatu secara virtual
<i>Virtual server</i>	Server yang dibangun secara virtual

## DAFTAR ISI

<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>LEMBAR PENGESAHAN DOSEN PEMBIMBING .....</b>	<b>ii</b>
<b>LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR .....</b>	<b>iii</b>
<b>LEMBAR PENGESAHAN DOSEN PENGUJI .....</b>	<b>iv</b>
<b>HALAMAN PERSEMBAHAN .....</b>	<b>v</b>
<b>HALAMAN MOTTO .....</b>	<b>vi</b>
<b>KATA PENGANTAR .....</b>	<b>vii</b>
<b>SARI .....</b>	<b>x</b>
<b>TAKARIR .....</b>	<b>xi</b>
<b>DAFTAR ISI .....</b>	<b>xii</b>
<b>DAFTAR GAMBAR .....</b>	<b>xv</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
<b>1.1 Latar Belakang Masalah .....</b>	<b>1</b>
<b>1.2 Rumusan Masalah .....</b>	<b>4</b>
<b>1.3 Batasan Masalah .....</b>	<b>5</b>
<b>1.4 Tujuan Penelitian .....</b>	<b>5</b>
<b>1.5 Manfaat Penelitian .....</b>	<b>5</b>
<b>1.6 Metodologi Penelitian .....</b>	<b>6</b>
1.6.1 Metodologi Pengumpulan Data .....	6
1.6.2 Metode Pengembangan Virtual Server .....	7
<b>1.7 Sistematika Penulisan .....</b>	<b>8</b>

<b>BAB II LANDASAN TEORI .....</b>	<b>10</b>
<b>2.1 Virtualisasi .....</b>	<b>10</b>
2.1.1 Pemahaman Tentang Virtualisasi .....	10
2.1.2 Perkembangan Virtualisasi .....	12
2.1.3 Virtualisasi pada Berbagai Platform .....	13
2.1.4 Kelebihan Menggunakan Virtualisasi .....	14
2.1.5 Cara Melakukan Virtualisasi .....	15
2.1.6 Perbandingan Virtual Server dengan Dedicated Server .....	16
<b>2.2 Server .....</b>	<b>18</b>
2.2.1 DHCP Server .....	18
2.2.2 DNS Server .....	19
2.2.3 Web Server .....	25
2.2.4 FTP Server .....	27
2.2.5 Proxy Server .....	30
<b>BAB III METODOLOGI .....</b>	<b>34</b>
<b>3.1 Analisis Kebutuhan Server .....</b>	<b>34</b>
<b>3.2 Metode Analisis .....</b>	<b>34</b>
<b>3.3 Hasil Analisis .....</b>	<b>34</b>
<b>3.4 Perangkat lunak yang dibutuhkan .....</b>	<b>35</b>
<b>3.5 Perangkat keras yang dibutuhkan .....</b>	<b>39</b>
<b>3.6 Instalasi Perangkat Lunak .....</b>	<b>40</b>
<b>3.7 Perancangan Network Diagram .....</b>	<b>40</b>
<b>3.8 Perancangan Sequence Diagram .....</b>	<b>42</b>

<b>BAB IV HASIL DAN PEMBAHASAN .....</b>	<b>44</b>
<b>4.1 Tahapan-tahapan Analisis .....</b>	<b>44</b>
<b>4.2 Konfigurasi dan Instalasi .....</b>	<b>46</b>
4.2.1 Konfigurasi DHCP Server .....	46
4.2.2 Konfigurasi Proxy Server .....	47
4.2.3 Konfigurasi DNS Server .....	48
4.2.4 Konfigurasi Web Server .....	49
4.2.5 Konfigurasi FTP Server .....	50
<b>4.3 Hasil Pengujian dan Pembahasan .....</b>	<b>51</b>
4.3.1 Memory Usage .....	52
4.3.2 Load Average .....	55
4.3.3 Ping Latency .....	57
4.3.4 Processes .....	60
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>63</b>
<b>5.1 Kesimpulan .....</b>	<b>65</b>
<b>5.2 Saran .....</b>	<b>64</b>
<b>5.3 Rekomendasi .....</b>	<b>64</b>
<b>DAFTAR PUSTAKA .....</b>	<b>67</b>

## DAFTAR GAMBAR

Gambar 2.1 Penggabungan beberapa <i>server</i> ke dalam satu PC server .....	11
Gambar 2.2 Komponen-komponen kerja DNS .....	21
Gambar 2.3 Struktur DNS .....	23
Gambar 2.4 Ilustrasi alur <i>request – response</i> .....	26
Gambar 2.5 Koneksi antar Proxy .....	35
Gambar 3.1 Network Diagram <i>virtual server</i> .....	41
Gambar 3.2 Sequence Diagram berbasis Cacti .....	42
Gambar 4.1 Menjalankan <i>service dhcpd</i> .....	47
Gambar 4.2 Menjalankan <i>service squid</i> .....	47
Gambar 4.3 Menjalankan <i>service named</i> .....	48
Gambar 4.4 Menjalankan <i>service httpd</i> .....	50
Gambar 4.5 Menjalankan <i>service vsftpd</i> .....	51
Gambar 4.6 Memory Usage DHCP-Proxy dedicated server .....	52
Gambar 4.7 Memory Usage DHCP-Proxy virtual server .....	53
Gambar 4.8 Memory Usage DNS-Web-FTP dedicated server .....	53
Gambar 4.9 Memory Usage DNS-Web-FTP virtual server .....	54
Gambar 4.10 Load Average DHCP-Proxy dedicated server .....	55
Gambar 4.11 Load Average DHCP-Proxy virtual server .....	56
Gambar 4.12 Load Average DNS-Web-FTP dedicated server .....	56
Gambar 4.13 Load Average DNS-Web-FTP virtual server .....	56
Gambar 4.14 Ping Latency DHCP-Proxy dedicated server .....	57

Gambar 4.15 Ping Latency DHCP-Proxy virtual server .....	58
Gambar 4.16 Ping Latency DNS-Web-FTP dedicated server .....	58
Gambar 4.17 Ping Latency DNS-Web-FTP virtual server .....	59
Gambar 4.18 Processes DHCP-Proxy dedicated server .....	60
Gambar 4.19 Processes DHCP-Proxy virtual server .....	60
Gambar 4.20 Processes DNS-Web-FTP dedicated server .....	61
Gambar 4.21 Processes DNS-Web-FTP virtual server .....	61

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Tingginya inflasi dan melemahnya ekonomi dunia jadi penyebab banyak perusahaan terpaksa melakukan berbagai pemangkasan, mulai dari pengurangan jumlah karyawan hingga pengurangan biaya perjalanan bisnis, sehingga tidak mengejutkan bila para pimpinan perusahaan juga dituntut menekan pengeluaran serta memaksimalkan investasi teknologi informasi (TI) perusahaan<sup>1</sup>.

Teknologi virtualisasi menawarkan ekspansi infrastruktur teknologi informasi ke tingkat yang lebih tinggi tanpa harus dihadapkan kepada ongkos investasi yang berlipat. Hal tersebut memungkinkan karena virtualisasi menyederhanakan cara memanfaatkan sumber daya komputer sehingga dapat digunakan untuk berbagai kebutuhan.

Sebelum ada virtualisasi, setiap aplikasi bisnis harus berjalan dengan *server* masing-masing sehingga dikenal ada *server* CRM (*Customer Relationship Management*), ERP (*Enterprises Resource Planning*), dan sejenisnya. Setiap penambahan aplikasi akan diikuti dengan penambahan mesin baru dan tentu kebutuhan ruangan baru. Dengan virtualisasi, penambahan ruang bisa dikatakan tidak diperlukan karena sebuah *server* dapat digunakan bersama-sama untuk banyak aplikasi. Penambahan mesin pun, jika diperlukan karena keterbatasan

---

<sup>1</sup> <http://www.antara.co.id/view/?i=1224815189&c=PRW&s=>

kapasitas, menggunakan sistem modular sehingga lebih cepat dan efisien. Ongkos untuk membayar lisensi perangkat lunak juga akan lebih kecil dengan jumlah *server* yang lebih sedikit<sup>2</sup>.

Menurut analisis dari Gartner, para vendor TI raksasa mulai mengalihkan bisnis mereka dari *server* besar ke *server-server* yang lebih kecil dengan mesin yang lebih efisien dan kompatibel satu dengan yang lain. Perusahaan harus mengeluarkan biaya yang sangat besar untuk pengadaan *server* dengan sistem operasi Windows yang standar saja. Belum lagi, karena kapasitas yang dibutuhkan sebenarnya hanya sekitar 20 persen dari kapasitas yang dimiliki oleh *server-server* canggih tersebut. Bagi para pemilik bisnis, ini berarti pemborosan<sup>1</sup>.

Selain itu persoalan pasokan energi di negeri ini, amatlah krusial. Baik PLN dan Pertamina, misalnya, selalu berada dalam ketidakpastian di sektor hulu (khususnya terkait fluktuasi harga minyak dunia), yang membuat layanan di hilir sulit stabil. Di satu pihak, wacana kepedulian lingkungan sebagai imbas *global warming* belakangan makin nyaring terdengar. Bukan hanya di kalangan pemerintahan, namun juga telah sampai ke masyarakat, hingga korporasi yang mengusung konsep *green business*.

Sektor teknologi informasi (TI) termasuk salah satu segmen yang tidak luput menerapkan strategi ekonomi hijau ini. Sebagai salah satu 'pemakan rakus' sektor energi tidak terbarukan, memang mutlak adanya kalangan TI terlibat. Praktiknya, dalam setahun terakhir, telah ada berbagai produk, gagasan, sampai usulan di dunia TI yang berbasis kepada upaya penyelamatan lingkungan hidup

---

<sup>2</sup> [http://irsyadi.com/itnews\\_detail.php?recordID=202](http://irsyadi.com/itnews_detail.php?recordID=202)

demi kemaslahatan umat manusia di masa mendatang itu. Gartner bahkan memprediksi separuh dari *data center* di dunia pada 2008 akan kekurangan kapasitas *power* dan *cooling* akibat krisis energi. Survei Sharing Vision pada 2007 juga menunjukkan bahwa mayoritas perusahaan di Indonesia justru menginginkan metode *hot standby* dalam *data center* miliknya, sekalipun mereka tahu krisis energi kerap terjadi di Indonesia. Yang lebih ironis, sebagaimana dirilis Aparture Research Institute 2007, 43 persen responden *data center* (dari kalangan perbankan, kesehatan, telekomunikasi, dan ritel) justru kehabisan tenaga listrik disetiap kondisi puncak. Angka tambahan ini naik dua kali lipat dari jumlah tahun sebelumnya<sup>3</sup>. Dengan kata lain, bayangkan jika perusahaan memiliki lima *server* dengan investasi ribuan dolar untuk membeli perangkat lunak, padahal mestinya satu *server* saja sudah bisa memenuhi kebutuhan perusahaan tersebut. Berarti kapasitas empat *server* yang lain tidak terpakai atau tidak diperlukan<sup>1</sup>.

*Virtualization* adalah tren yang belakangan banyak dibicarakan yang juga merupakan cara tepat untuk memangkas biaya *server* dan TI perusahaan. Jika sebelumnya departemen TI mengatur *server* berdasarkan fungsi dan kebutuhan, misalnya ada *server* khusus untuk *email*, dokumen, akunting dan bahkan *server* khusus untuk *printing*. Dengan virtualisasi, semua fungsi tersebut dapat dilakukan antar mesin sehingga jauh lebih efisien<sup>1</sup>.

Untuk skala kecil, penggunaan virtualisasi diprediksi menurunkan konsumsi listrik minimal 25 persen dibandingkan tanpa virtualisasi. Sementara untuk skala besar penghematan energi mencapai 80 persen. Karena itu, virtualisasi kini

---

<sup>3</sup> <http://www.detikinet.com/read/2009/06/23/132849/1152586/328/yuk-bangun-data-center-ramah-lingkungan>

menjadi pilihan menarik bagi dunia usaha dan organisasi mulai dari yang berukuran kecil hingga besar. Selama masa krisis masih berlangsung, virtualisasi masih menjadi solusi yang relevan untuk membantu perusahaan tetap bertahan<sup>2</sup>.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang masalah tersebut di atas, maka dapat ditarik rumusan masalah sebagai berikut, yaitu :

1. Bagaimana mengkonfigurasi server DHCP, Proxy, Web, DNS, dan FTP pada infrastruktur Virtual server ?
2. Bagaimana perbandingan kinerja dua infrastruktur *server* berbeda antara Dedicated server yang berdiri sendiri dengan Virtual server yang bersifat *all in one* pada *server* DHCP dan Proxy serta *server* Web, DNS, dan FTP dari segi *memory usage*, *load average*, *ping latency*, dan *processes* menggunakan Template Unix Cacti dengan tingkat trafik di bawah tujuh pengguna ?

## 1.3 Batasan Masalah

Pembangunan *server* berbasis virtualisasi dibuat berdasarkan pada kebutuhan dari hasil analisis. Pada pengembangannya pembangunan *server* berbasis virtualisasi ini dibatasi pada beberapa permasalahan, antara lain:

1. Paramater kinerja *server* yang dianalisis adalah dari segi penggunaan memori, besarnya *load average*, besarnya *ping latency*, dan keseluruhan jumlah *processes*.

2. Konfigurasi Virtual server sama dengan konfigurasi pada Dedicated server, sederhana berdasarkan kebutuhan UKM.
3. Jenis-jenis *server* yang divirtualisasi berupa DHCP Server, Proxy Server, DNS Server, Web Server, dan FTP Server.
4. Dilakukan pada waktu yang bersamaan tetapi tidak secara *real time* dan pada tingkat trafik yang rendah, yaitu di bawah tujuh pengguna.

#### **1.4 Tujuan Penelitian**

Tujuan penelitian dari tugas akhir ini adalah untuk menganalisis perbandingan kinerja dua jenis infrastruktur *server* dengan penerapan teknologi yang berbeda, yaitu Virtual server dengan Dedicated server dan hasil dari penelitian ini diharapkan dapat memberikan pemahaman yang mendalam mengenai kinerja *server* khususnya Virtual server.

#### **1.5 Manfaat Penelitian**

1. Memberikan gambaran teknologi *server* dengan dua jenis infrastruktur yang berbeda, yaitu Dedicated server dan Virtual server.
2. Mendapatkan hasil performa dari Dedicated server dan Virtual server pada tingkat trafik yang rendah.
3. Memberikan rekomendasi tentang bagaimana suatu infrastruktur Virtual server yang baik sesuai dengan kapasitas.

## 1.6 Metodologi Penelitian

Untuk mengidentifikasi masalah dan kebutuhan, serta persyaratan bagi solusi yang diimplementasikan dalam pengembangan Virtual server yang akan diujicoba perbandingannya dengan Dedicated server, sehingga diharapkan dapat memberikan solusi yang tepat dalam menganalisis hasil perbandingan kedua infrastruktur *server* tersebut. Dalam identifikasi ini tahap-tahap yang dilakukan berupa :

### 1.6.1 Metode Pengumpulan Data

#### a. Wawancara

Melakukan wawancara dari perancangan infrastruktur Virtual server yang akan dibangun, dalam hal ini dilakukan diskusi kelompok, diskusi dengan pakar, saling bertukar pendapat dalam penelitian, dan konsultasi pada dosen terkait dengan mata kuliah yang berbasis pada konsentrasi sistem dan jaringan komputer.

#### b. Studi kepustakaan

Studi kepustakaan dilakukan dalam proses pengumpulan data yang akurat untuk mendukung analisis dan memperkuat usulan pemecahan masalah yang dilakukan. Studi ini dilakukan di perpustakaan universitas dimana tersedia berbagai sumber informasi yang mendukung secara teoritis serta dari beberapa buku yang didapat dari berbagai tempat. Selain itu sumber informasi diperoleh dengan studi melalui media elektronik khususnya internet, dilakukan dalam upaya pencarian data tentang spesifikasi dari peralatan pendukung teknologi (termasuk kelebihan dan kekurangannya), infrastruktur *server* dan standarnya, teori pendukung, fitur,

aplikasi dan perancangan, serta bagaimana pengimplementasiannya, khususnya mengenai Virtual server.

### **1.6.2 Metode Pengembangan Virtual Server**

#### **a. Analisis Kebutuhan**

Analisis kebutuhan adalah analisis yang dilakukan untuk mengetahui kebutuhan perangkat keras dan perangkat lunak dalam proses pengembangan Virtual server.

#### **b. Pengembangan Virtual Server**

Pengembangan Virtual server merupakan tahapan yang dilakukan untuk membuat sebuah konsep infrastruktur untuk mengetahui konfigurasi yang dibutuhkan agar sesuai dengan apa yang diinginkan.

#### **c. Simulasi Pengujian Server**

Melakukan pengujian terhadap fungsi-fungsi *server* untuk mengetahui tingkat keberhasilan dari pengembangan *server* sebelumnya. Pengujian fungsi-fungsi *server* dilakukan pada Virtual server yang telah dikonfigurasi.

#### **d. Implementasi Server**

Setelah melakukan pengujian terhadap fungsi-fungsi *server* dan telah berhasil, *server* diimplementasikan untuk melihat hasil yang nantinya akan dianalisis untuk mengetahui perbandingan dari tugas akhir yang dikerjakan.

#### **e. Analisis Hasil Implementasi**

Diperoleh setelah pengujian dan implementasi *server* dilakukan untuk mengetahui hasil dari implementasi tersebut untuk kemudian dituliskan laporannya.

## 1.7 Sistematika Penulisan

Pembahasan tugas akhir ini disusun dalam lima bab yang akan diuraikan sebagai berikut :

### **BAB I : PENDAHULUAN**

Bab ini merupakan bagian pendahuluan dimana akan tercakup secara umum mengenai latar belakang penulisan tugas akhir, rumusan masalah tugas akhir ini, batasan-batasan masalah pengerjaan tugas akhir, tujuan penelitian yang akan dicapai, manfaat penelitian, metodologi yang dipakai dalam penyusunan laporan dan sistematika penulisan yang digunakan.

### **BAB II : LANDASAN TEORI**

Dalam bab ini akan dibahas mengenai tinjauan pustaka yang digunakan dan teori-teori dasar atau umum yang berhubungan dengan infrastruktur *server*, terutama pengertian-pengertian dan penjelasan mengenai hal-hal yang berhubungan dengan *server*, peralatan dan teknologinya, juga mengenai landasan dalam analisis, desain dan perancangan *server* khususnya Virtual server.

### **BAB III : METODOLOGI**

Pada bab ini akan dibahas mengenai hasil analisis yang telah dilakukan untuk mengetahui kebutuhan *server*, konsep dan perancangan infrastruktur dari Virtual server yang akan dikembangkan, dan bagaimana langkah-langkah melakukan instalasi

dan konfigurasi pada Virtual server tersebut. Adapun isi yang akan diuraikan pada hasil analisis kebutuhan *server* mencakup spesifikasi perangkat keras dan perangkat lunak yang digunakan, dan untuk konsepnya akan diberikan gambaran Network Diagram dan Sequence diagram disertai penjelasan infrastrukturnya.

#### **BAB IV : HASIL DAN PEMBAHASAN**

Pada bab ini berisi tentang hasil simulasi pengujian fungsi-fungsi *server* yang telah dikonfigurasi beserta implementasi infrastruktur *server* dan evaluasi dari pembahasan hasil analisis perbandingan tugas akhir yang dikerjakan.

#### **BAB V : KESIMPULAN DAN SARAN**

Memuat kesimpulan dari hasil skripsi yang didapat setelah melakukan survei tugas akhir, analisis, pengembangan dan evaluasi terhadap *server* yang dikembangkan, beserta beberapa saran dan rekomendasi yang dapat dijadikan pertimbangan untuk organisasi atau perusahaan serta bagi pihak-pihak yang berkepentingan.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Virtualisasi**

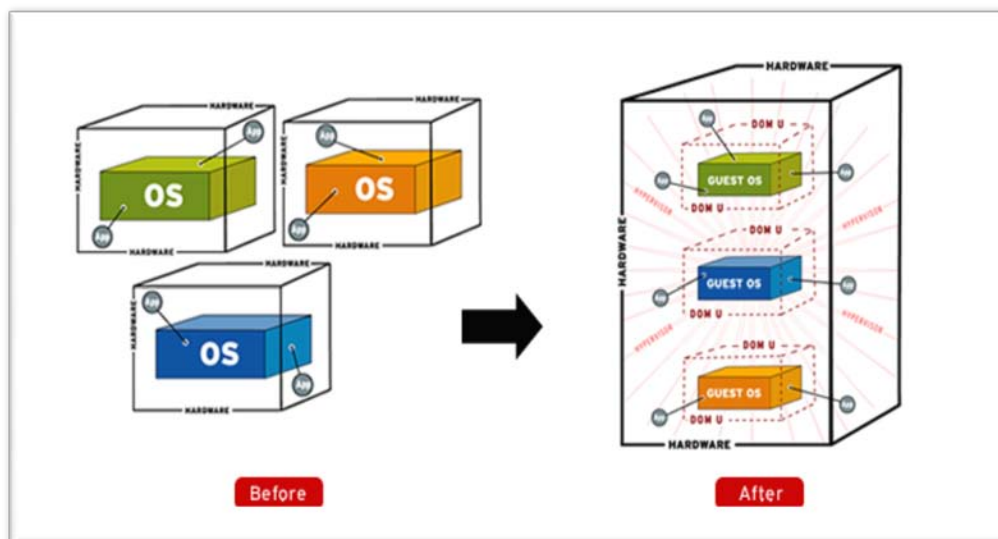
##### **2.1.1 Pemahaman Tentang Virtualisasi**

Teknologi ini sedang hangat menjadi pembicaraan di kalangan IT (*Information Technology*) saat ini. Sebenarnya teknologi ini sudah ada sejak akhir tahun 1960. Dengan perkembangan dunia IT yang berkembang pesat saat ini maka teknologi *virtualization* ini pun lebih disempurnakan. Dengan berkembangnya juga teknologi *hardware* yang ada saat ini, memungkinkan virtualisasi berjalan semakin baik dan sempurna. Satu prosesor kini terdiri dari berbagai *core* (inti) dan peningkatan L2 di prosesor tersebut yang semakin baik. Awalnya dimulai dari Hyperthreading dan Hypertransport, Dual Core, 4 Core, bahkan 8 Core di dalamnya.

Sebuah perusahaan yang peka akan kebutuhan IT tentunya akan memiliki berbagai *server* sesuai dengan kebutuhannya. Mulai dari File Server, DNS Server, Proxy Server maupun DHCP Server, dan lain-lain. Tentunya mereka pun harus menyediakan beberapa PC (*Personal Computer*) sekelas *server* untuk membangun itu semua. Jika diperhatikan utilitas dari beberapa jenis *server* tersebut, tentunya tidak akan efisien dan tentunya memakan tempat dan membutuhkan daya yang besar juga. Contohnya DNS (*Domain Name System*) Server dan DHCP (*Dynamic Host Configuration Protocol*) Server. Utilisasi dari kedua *server* tersebut tidaklah

begitu besar yang sebetulnya dapat dilakukan oleh PC biasa. Namun PC biasa memiliki keterbatasan jika digunakan pada *lifetime* yang lama.

Solusi yang biasa dilakukan adalah menggabungkan beberapa kegunaan *server* tersebut ke dalam sebuah PC Server yang handal dan memiliki spesifikasi tinggi, seperti yang ditunjukkan pada gambar 2.1. Namun akan timbul permasalahan baru yaitu keamanan, dengan penggabungan beberapa *server* tentunya akan semakin banyak *port* yang terbuka dalam *server* tersebut. Hal ini semakin besar kemungkinan *server* tersebut untuk dieksploitasi. Tentunya ini membutuhkan manajemen keamanan yang baik terhadap *server* tersebut. Satu permasalahan lagi jika *server* tersebut melakukan *update* yang memerlukan *restart*, maka *service server* yang lain akan ikut mati. Teknologi *virtualization* inilah solusi dari permasalahan tersebut<sup>4</sup>.



Gambar 2.1 Penggabungan beberapa *server* ke dalam satu PC server  
(<http://www.trendigital.com/Common/NewsImage.ashx?Id=321&w=200&h=200>)

<sup>4</sup> <http://www.trendigital.com/Website/News.aspx?id=321>

### **2.1.2 Perkembangan virtualisasi**

#### **Tahun 1960 – 1980**

Dimulai pada pertengahan 1960an saat IBM mengembangkan CP-40 (versi pertama dari CP/CMS) yang mulai digunakan kalangan industri pada Januari 1967. Tahun 1970an menandakan era penggunaan *mainframe*. Dimana *mainframe* dibuat menjadi beberapa partisi *virtual machine* untuk memungkinkan dijalankannya *multitasking*. Dengan mahalnya biaya yang diperlukan untuk *mainframe*, virtualisasi dimaksudkan untuk memberikan hasil maksimal pada investasi yang telah dikeluarkan.

#### **Tahun 1980 - 1990**

Pada tahun 1985 Lotus Computing Corporation berkerja-sama dengan AT&T mengeluarkan Intel 80286 sebuah *virtual machine monitor*. Era 90an terjadi pergeseran tren yang cukup signifikan dimana para pengguna komputer (khususnya perusahaan yang sedang berkembang) lebih memilih *server* secara terdistribusi daripada menjalankan virtualisasi di *mainframe*, tentunya ini dilakukan dengan pertimbangan biaya ditambah makin tersedianya komputer dengan harga yang lebih terjangkau. Tetapi teknologi *virtual* tetap dikembangkan, ini dibuktikan dengan diluncurkannya VMware pada tahun 1999. Pada era ini jugalah *desktop* dan *server x86* mengukuhkan posisinya sebagai standar *server* di dunia industri.

#### **Tahun 2000an**

Juni 2001 Connectix mengeluarkan versi pertama dari Virtual PC untuk Windows, diikuti VMware meluncurkan produk virtualisasi untuk *server x86*

sebulan sesudahnya (Juli). Pada Februari 2003 Microsoft meluncurkan salah satu produk virtualisasinya: Microsoft Virtual Server. Pada November 2003, Microsoft melakukan gebrakan dengan meluncurkan Microsoft Virtual PC, dimana telah menggunakan teknologi virtualisasi di tingkat mesin.

### **Tahun 2008**

Microsoft meluncurkan Hyper-V dengan nama Viridian, teknologi virtualisasi berbasis sistem x64. Pada bulan April VMware mengeluarkan VMware Workstation 6.5 beta<sup>5</sup>.

#### **2.1.3 Virtualisasi pada berbagai *platform***

Untuk mendukung virtualisasi ini tentunya digunakan perangkat lunak khusus untuk membuat virtualisasi dari suatu sistem operasi. Selain perangkat lunak, tentunya perangkat keras juga harus mendukung virtualisasi secara penuh, terutama prosesor. Perangkat lunak untuk virtualisasi di Linux cukup banyak pilihannya, dari yang berbayar sampai yang gratisan. Umumnya perangkat lunak tersebut juga ada di *platform* sistem operasi Windows.

Untuk yang berbayar ada VMware yang sudah terkenal sejak dulu untuk melakukan virtualisasi yang sangat baik. Tersedia untuk *platform* sistem operasi Windows dan Linux. Saat ini juga telah mendukung prosesor yang mendukung virtualisasi, baik dari yang berbasis Intel maupun AMD. Untuk perangkat lunak virtualisasi yang gratis, ada VirtualBox dan Qemu. Kedua-duanya juga sudah mendukung virtualisasi pada prosesor saat ini dan tentunya dapat dikonsumsi

---

<sup>5</sup> <http://wss-id.org/blogs/leonardo/archive/2008/07/17/TeknologiVirtualisasi.aspx>

gratis oleh siapa saja. Fitur yang ditawarkan juga tidak kalah dengan yang berbayar. Terutama VirtualBox yang sangat mirip GUI *interfacenya* dengan VMware. Untuk Qemu memerlukan instalasi *kernel* khusus untuk mengaktifkan *mode virtualization full* pada prosesor yang digunakan, sementara VirtualBox tidak memerlukannya, karena *defaultnya* sudah mendukung prosesor yang mendukung virtualisasi.

Teknologi pada prosesor ini memungkinkan sistem operasi yang kita emulasi atau yang ada di dalam virtualisasi dapat menggunakan atau mengakses prosesor secara langsung seperti menggunakan sistem operasi langsung tanpa divirtualisasi. Sehingga sistem operasi yang divirtualisasi akan memiliki kecepatan, fitur dan kompatibilitas terhadap prosesor yang sama jika menggunakan sistem operasi yang tidak dalam perangkat lunak virtualisasi<sup>6</sup>.

#### **2.1.4 Kelebihan menggunakan virtualisasi :**

1. **Biaya investasi *hardware*.** Investasi *hardware* dapat ditekan lebih rendah karena virtualisasi hanya mendayagunakan kapasitas yang sudah ada. Tak perlu ada penambahan perangkat komputer dan peralatan secara fisik. Kalaupun ada penambahan kapasitas *harddisk* dan *memory*, itu lebih ditujukan untuk mendukung stabilitas kerja komputer induk, yang jika dihitung secara finansial, masih jauh lebih hemat dibandingkan investasi *hardware* baru.
2. **Kemudahan *recovery*.** *Server-server* yang dijalankan di dalam sebuah *virtual machine* dapat disimpan dalam satu buah *image* yang berisi seluruh konfigurasi

---

<sup>6</sup> <http://www.trendigital.com/Website/News.aspx?id=653>

sistem. Jika suatu saat *server* tersebut *crash*, kita tidak perlu melakukan instalasi dan konfigurasi ulang. Cukup mengambil salinan *image* yang sudah disimpan, *merestore* data hasil *backup* terakhir dan *server* berjalan seperti sedia kala. Hemat waktu, tenaga dan sumber daya.

3. **Kemudahan *deployment*.** Virtual server dapat dikloning sebanyak mungkin dan dapat dijalankan pada mesin lain dengan mengubah sedikit konfigurasi. Mengurangi beban kerja para staf IT dan mempercepat proses implementasi suatu sistem.

### 2.1.5 Cara Melakukan Virtualisasi

Untuk memulai sistem virtualisasi, sebelumnya para penyedia sistem ini harus melakukan investasi waktu dan dana serta layanan profesional. Namun seiring makin banyaknya industri yang menggunakan sistem ini, sekarang sudah tersedia standar yang makin memudahkan perusahaan melakukan virtualisasi.

#### 1. **Membuka kunci *server* untuk virtualisasi**

Setelah menentukan *server* atau *software* mana yang akan divirtualisasi (atau diduplikasi), pilih komponen mana yang akan dijalankan oleh *software* tersebut. Jika diperlukan untuk membeli *hardware server* yang baru, bandingkan pilihan yang ditawarkan secara teliti agar sesuai dengan spesifikasi yang diperlukan. Setiap *server virtual* secara fisik terlihat seperti komputer biasa yang berdiri sendiri. Kenyataannya, kebanyakan *server virtual* menggunakan hanya satu *server* yang digunakan bersama.

## 2. Mudah digunakan dan digabungkan dengan yang lain

Penggunaan virtualisasi diawali dengan fungsi baru dari *server* lama yang dimiliki korporasi, melakukan sambungan komponen perangkat keras, lalu melakukan integrasi *networking* dan sistem penyimpanan. Saat perangkat keras telah siap, pasang *software* virtualisasi, kemudian pindahkan aplikasi yang ada ke sistem *virtual*. Dengan menggunakan *software* virtualisasi yang dapat secara langsung dipasang pada *server*, tidak dibutuhkan *host operating system* tersendiri, sehingga memudahkan *vendor* maupun perusahaan dalam melakukan integrasi atas perangkat yang telah ada<sup>7</sup>.

### 2.1.6 Perbandingan Virtual Server dengan Dedicated Server

#### 1. Virtual Server:

##### a. Optimalisasi biaya:

Dengan Virtual server maka jumlah *server* yang digunakan pun dapat dikurangi, ini akan berpengaruh pada jumlah ruangan, daya listrik dan pendinginan.

##### b. Optimalisasi infrastruktur:

Dengan virtualisasi kita dapat memaksimalkan kinerja *server*, dengan demikian hasil dari investasi pun menjadi maksimal.

##### c. Fleksibilitas dalam *maintenance*:

Dengan virtualisasi akan memudahkan para *admin* untuk melakukan *maintenance* infrastruktur *server*, ini berdampak pada waktu yang dibutuhkan untuk *maintenance*.

---

<sup>7</sup> <http://teknologitinggi.wordpress.com/2008/10/24/pentingnya-virtualisasi-server-dalam-pengolahan-data/>

d. Fleksibilitas dalam *downtime server*:

Dengan teknologi *virtual*, *error* atau *downnya* salah satu Virtual server tidak berdampak pada Virtual server yang lain. Namun untuk teknologi sebelumnya apabila terjadi *error* pada *host operating system* maka akan berdampak pada *guest operating systemnya*.

## 2. Dedicated Server:

a. Meningkatnya biaya infrastruktur :

Banyaknya jumlah *server* yang digunakan dalam Dedicated server berpengaruh juga pada biaya yang harus disediakan.

b. *Server* tidak digunakan secara optimal :

Dengan dijalankannya satu aplikasi per *server*, besar kemungkinan *server* akan tidak digunakan secara maksimal.

c. Tingginya biaya *maintenance* :

Biaya yang dikeluarkan untuk *maintenance* pun akan berlipat berdasarkan jumlah *server* yang ada.

d. *Server independent* :

Dengan terdistribusinya *server* maka *downtime* salah satu *server* tidak akan berpengaruh pada *server* yang lain<sup>5</sup>.

## 2.2 Server

Dalam jaringan komputer, peranan *server* sangatlah penting sebagai penyedia layanan dan pengatur kebijakan-kebijakan dalam jaringan komputer tersebut. Tanpa *server*, manajemen pada jaringan komputer tersebut akan menjadi

kacau dan tidak beraturan. Terutama lima *server* utama yang dibahas dari bab 2.2.1 sampai bab 2.2.5 yang bersumber dari “Modul Praktikum Jaringan Komputer Jurusan Teknik Informatika, Universitas Islam Indonesia Semester Genap 2006/2007” sebagai berikut:

### **2.2.1 DHCP Server**

Untuk mengadministrasi sebuah jaringan kecil, pemberian IP (*Internet Protocol*) secara manual (*static*) sangat memudahkan bagi administrator jaringan. Namun jika jaringan sudah mulai luas kemungkinan untuk menggunakan IP yang sama akan lebih besar sehingga menyebabkan konflik. Dengan dasar ini maka penggunaan DHCP (*Dynamic Host Configuration Protocol*) Server sangat dianjurkan.

DHCP adalah layanan yang secara otomatis memberikan nomor IP kepada komputer yang memintanya. Komputer yang memberikan nomor IP disebut sebagai DHCP Server, sedangkan komputer yang meminta nomor IP disebut sebagai DHCP Client. Dengan demikian administrator tidak perlu lagi harus memberikan nomor IP secara manual pada saat konfigurasi TCP/IP, tapi cukup dengan memberikan referensi kepada DHCP Server. Layanan ini bisa juga diberikan kepada suatu komputer dengan kriteria tertentu, misalnya saja berdasarkan alamat MAC Address Network Card ataupun berdasarkan *hostnamanya*.

Sebagai contoh, jika terdapat sebuah DHCP Server dengan *range IP* 192.168.1.100 sampai dengan 192.168.10.200, maka setiap komputer yang terhubung pada jaringan tersebut dan mengaktifkan penggunaan DHCP, maka

DHCP Server akan memberikan alamat IP pada *range* diatas yaitu antara 100 sampai dengan 200. Biasanya DHCP Server memberikan IP pada *range* paling atas terlebih dahulu. Jika pada contoh kita ini baru satu yang menggunakan DHCP maka kemungkinan besar mendapat IP 192.168.1.200. Jika pada jaringan tersebut terdapat sebuah komputer dengan IP Statik dan masih dalam *range* dari IP DHCP Server maka DHCP Server tidak akan menggunakan IP tersebut untuk diberikan kepada pengguna DHCP yang lain.

### 2.2.2 DNS Server

#### 1. Konsep DNS

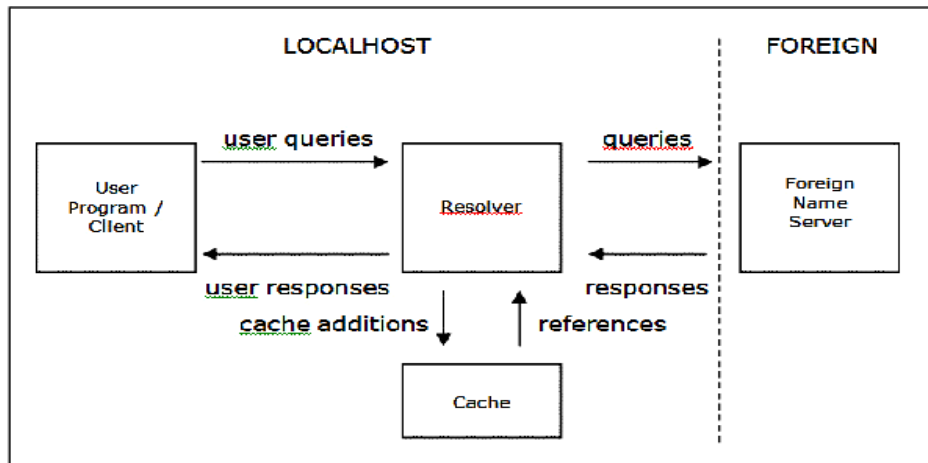
Domain Name System (DNS) adalah *distribute database system* yang digunakan untuk pencarian nama komputer (*name resolution*) di jaringan yang menggunakan TCP/IP (*Transmission Control Protocol/Internet Protocol*). DNS biasa digunakan pada aplikasi yang terhubung ke Internet seperti *web browser* atau *e-mail*, dimana DNS membantu memetakan *hostname* sebuah komputer ke IP address. Selain digunakan di Internet, DNS juga dapat diimplementasikan ke *private network* atau Intranet dimana DNS memiliki keunggulan seperti:

- a. Mudah, DNS sangat mudah karena *user* tidak lagi direpotkan untuk mengingat IP address sebuah komputer cukup *host name* (nama komputer).
- b. Konsisten, IP address sebuah komputer bisa berubah tapi *hostname* tidak berubah.
- c. Sempel, *user* hanya menggunakan satu nama *domain* untuk mencari baik di Internet maupun di Intranet.

DNS dapat disamakan fungsinya dengan buku telepon. Dimana setiap komputer di jaringan Internet memiliki *hostname* (nama komputer) dan Internet Protocol (IP) address. Secara umum, setiap klien yang akan mengkoneksikan komputer yang satu ke komputer yang lain, akan menggunakan *hostname*. Lalu komputer anda akan menghubungi DNS Server untuk mencek *hostname* yang anda minta tersebut berapa IP addressnya. IP address ini yang digunakan untuk mengkoneksikan komputer anda dengan komputer lainnya.

## 2. Komponen Kerja DNS

Untuk memahami cara kerja DNS terlebih dahulu harus dipahami tentang komponen-komponen yang bekerja di dalamnya. Gambar 2.2 berikut mengilustrasikan komponen-komponen DNS secara sederhana:



Gambar 2.2 Komponen-komponen kerja DNS

a. Keterangan:

- *Cache*: media penyimpanan sementara.
- *Resolver*: bagian dari program aplikasi yang berfungsi menjawab pertanyaan tentang *domain (local DNS)*.

b. Cara kerja *resolver*:

- *Resolver* menjawab pertanyaan dari *client* meliputi dua cara.
- *Resolver* memeriksa *cache*-nya, jika jawaban yang diminta terdapat di dalam *cache*-nya, maka *resolver* kemudian akan memberikan jawaban berupa IP address dari alamat yang dimaksud.
- Jika tidak terdapat jawaban dalam *cache*, maka *resolver* akan bertanya pada DNS Server serta menginterpretasikan hasilnya. Pertanyaan kepada DNS Server membuat DNS Server menjadi *defaultnya* memberikan jawaban atas pertanyaan itu. Jika informasi pada *zona file* dari DNS Server tidak ditemui, maka DNS Server akan mengirimkan *message failure* (pesan gagal) kepada *client*. Kemudian *client* akan menghubungi DNS Server yang lain untuk mencari jawaban atas pertanyaan tersebut. Jika berhasil ditemukan, maka DNS Server yang dihubungi *client* tersebut akan memberikan jawaban berupa IP address dari alamat yang dimaksud kepada DNS Server lokal untuk kemudian dikirim ke *client* yang *request*-nya. Proses tersebut disebut dengan *Forward Lookup Query*, yaitu permintaan dari *client* dengan cara memetakan nama komputer (*host*) ke IP address.

3. **Struktur hierarki DNS**

DNS mengatur proses penamaan komputer secara hierarkis dalam struktur pohon (*tree*). Struktur tersebut terbentuk dari sebuah *node* (titik) yang memiliki *subnode*, masing-masing *subnode* memiliki lagi *subnode* di bawahnya dan begitu seterusnya. *Node-node* tersebut diberi label sehingga disebut *domain*.

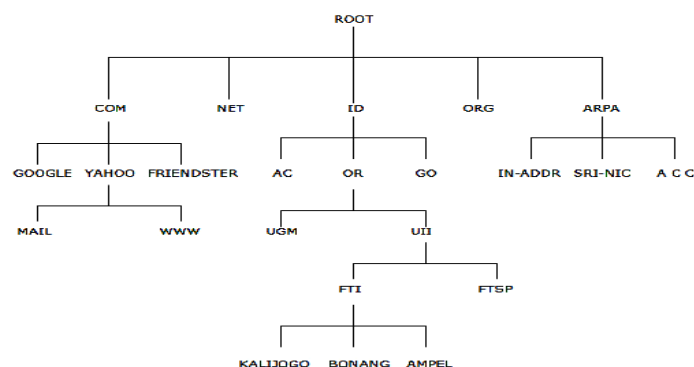
*Domain* ditentukan berdasarkan tingkatan kemampuan yang ada di struktur hierarki yang disebut dengan *level*, yang terbagi menjadi beberapa bagian di antaranya:

a. *Root Level Domains*: *Domain* ditentukan berdasarkan tingkatan kemampuan yang ada di struktur hierarki yang disebut dengan *level*. *Level* paling atas pada struktur hierarki disebut dengan *root domain*. *Root domain* diekspresikan berdasarkan periode dimana lambang untuk *root domain* adalah dot (“.”).

b. *Second Level Domains*: *Second level domains* dapat berisi *host* dan *domain* lain, yang disebut dengan *subdomain*. Untuk contoh: *domain* Mahasiswa, mahasiswa.com terdapat komputer (*host*) seperti server1.mahasiswa.com dan *subdomain* training.mahasiswa.com. *Subdomain* training.mahasiswa.com juga terdapat komputer (*host*) seperti client1.training.mahasiswa.com.

c. *Host Names*: *Domain name* yang digunakan dengan *hostname* akan menciptakan *Fully Qualified Domain Name* (FQDN) untuk setiap komputer. Sebagai contoh, jika terdapat fileserver1.detik.com, dimana fileserver1 adalah *hostname* dan detik.com adalah *domain name*.

Gambar 2.3 berikut merupakan ilustrasi sederhana mengenai struktur DNS:



Gambar 2.3 Struktur DNS

#### 4. Studi kasus DNS

Seorang mahasiswa ingin mengakses [www.facebook.com](http://www.facebook.com). Proses yang terjadi agar mahasiswa tersebut dapat menghubungi komputer [www.facebook.com](http://www.facebook.com) adalah:

- a. DNS Client menghubungi DNS Server lokal untuk mendapatkan IP address dari [www.facebook.com](http://www.facebook.com).
- b. DNS Server lokal memeriksa *cache*-nya, jika akses tersebut sudah pernah dilakukan sebelumnya, maka DNS Server lokal akan mengambil IP address [www.facebook.com](http://www.facebook.com) dari data *cache*-nya.
- c. Jika dalam data *cache* tidak ditemukan informasi yang dicari, maka kemudian DNS Server lokal akan mengirimkan *message failure* kepada *client* bahwa IP address dari alamat yang diminta tidak ditemukan.
- d. Kemudian *client* melakukan *request* kepada DNS Server tertinggi yaitu '.' (dot) atau *root server*. Dan hasilnya akan dikembalikan lagi ke *client*.
- e. Jika tidak ditemukan lagi, maka *client* akan menghubungi DNS Server .com (*Second Level Domains*). Hasilnya juga akan dikembalikan lagi ke *client*.
- f. Kemudian jika belum dapat ditemukan, *client* akan menghubungi DNS Server [www.facebook.com](http://www.facebook.com).
- g. DNS Server [www.facebook.com](http://www.facebook.com) memberikan IP address dari *subdomain* [www.facebook.com](http://www.facebook.com).
- h. IP address tersebut kemudian dikirimkan kembali ke DNS Server lokal untuk diberikan kepada *browser* dan kemudian dicatat dalam data *cache* DNS Server lokal.

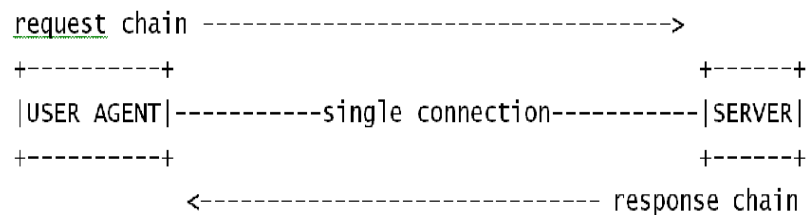
- i. *Browser* mengarah ke IP address yang dimaksud untuk mengakses komputer pada IP address tersebut.
- j. Mahasiswa tersebut akhirnya bisa Facebook-an.

### 2.2.3 Web Server

Web Server adalah *software server* yang menjadi tulang belakang dari World Wide Web (WWW). Web Server menunggu permintaan dari *client* yang menggunakan *browser* seperti Netscape Navigator, Internet Explorer, Mozilla Firefox, dan program *browser* lainnya. Jika ada permintaan dari *browser*, maka Web Server akan memproses permintaan itu dan kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke *browser*. Data ini mempunyai format yang standar disebut dengan format SGML (*Standard General Markup Language*). Data yang berupa format ini kemudian akan ditampilkan oleh *browser* sesuai dengan kemampuan *browser* itu. Contohnya adalah bila data yang dikirim berupa data gambar, *browser* yang hanya mampu menampilkan teks tidak akan mampu menampilkannya dan jika ada akan menampilkan alternatifnya saja.

Web Server, untuk berkomunikasi dengan *clientnya* (*web browser*) mempunyai protokol sendiri yaitu HTTP (*HyperText Transfer Protocol*). Dengan protokol ini, komunikasi antar Web Server dengan *clientnya* (*browser*) dapat saling dimengerti dan lebih mudah. Seperti telah dijelaskan di atas, standar format data pada *World Wide Web* adalah SGML. Tapi sudah menjadi hal yang umum bahwa para pengguna Internet lebih banyak menggunakan format HTML (*HyperText Markup Language*) karena penggunaannya yang lebih sederhana dan mudah dipelajari.

HTTP merupakan protokol yang bekerja pada lapisan aplikasi pada layer OSI dan secara sederhana dapat didefinisikan sebagai sekumpulan aturan untuk menukar data pada *World Wide Web*. Ide dasar dari HTTP adalah bagaimana sebuah data dapat berisi suatu referensi pada *file* yang bersangkutan. Sebenarnya HTTP juga dapat digunakan sebagai *generic protocol* untuk melakukan komunikasi antara *user agents*, *proxy*, *gateway* dengan protokol Internet lainnya seperti SMTP, FTP, Gopher, NNTP, dan lain sebagainya. Gambar 2.4 di bawah ini mengilustrasikan secara sederhana bagaimana alur *request response* antara *user agent* dengan *server*.



Gambar 2.4 Ilustrasi alur *request - response*

Kata *HyperText* mempunyai arti bahwa seorang pengguna Internet dengan *web browser*nya dapat membuka dan membaca dokumen-dokumen yang ada dalam komputernya atau bahkan komputer yang jauh tempatnya sekalipun. Hal ini memberikan cita rasa dari suatu proses yang *tridimensional*, artinya pengguna Internet dapat membaca dari satu dokumen ke dokumen yang lain hanya dengan mengklik beberapa bagian dari halaman-halaman dokumen (*web*) itu.

Proses yang dimulai dari permintaan *webclient* (*browser*), diterima Web Server, diproses, dan dikembalikan hasil prosesnya oleh Web Server ke *webclient* lagi dilakukan secara transparan. Setiap orang dapat dengan mudah mengetahui

apa yang terjadi pada tiap-tiap proses. Secara garis besarnya Web Server hanya memproses semua masukan yang diperolehnya dari *webclient*nya.

Untuk membuat sebuah Web Server, maka kita akan menemukan berbagai macam persoalan, dimulai dari pemilihan *software web browser* mana yang paling sesuai kebutuhan, apa spesifikasi perangkat keras yang dibutuhkan, bagaimana kondisi interkoneksi jaringan Internet yang ada, dan lain sebagainya. Belum lagi termasuk bagian pembuatan halaman-halaman *web*nya, mau menggunakan format apa (HTML, SGML, PHP, PHP3, CGI, dan lain-lain). Hal yang paling utama dalam proses pembuatan Web Server adalah memilih perangkat lunak mana yang akan digunakan sebagai Web Server kita.

#### **2.2.4 FTP Server**

##### **1. Pemahaman FTP Server**

FTP merupakan singkatan dari File Transfer Protocol adalah suatu jalur atau media atau alat untuk untuk proses *transfer file* dari suatu *server* ke *server* lain dalam jaringan Internet yang menggunakan *port* 21 dan 20. Salah satu aktifitas utama yang ditemukan dalam Internet adalah *transfer file*. Setiap menit dalam suatu hari, *user* Internet *mendownload file* dari berbagai *website* dan *webmaster* *mengupload file* dan halaman *web* ke dalam *websitenya*.

Metode yang paling umum digunakan untuk *transfer file* pada Internet adalah dengan menggunakan FTP. Tujuan penggunaan FTP ini adalah sebagai *platform independent* untuk *transfer file* pada Internet yang berbasiskan arsitektur *client/server*. Untuk menggunakan FTP dibutuhkan sebuah program FTP Client dan program FTP Server. FTP Server akan menyimpan atau sebagai tempat *file*

yang akan diakses selama transfer terjadi dan FTP Client akan menghubungi FTP Server untuk mengirim atau mengambil *file* dari *server*.

FTP dibagi ke dalam dua buah *channel* yang berbeda, satu digunakan untuk melayani data (TCP-Port 20) dan yang lain digunakan sebagai kontrol (TCP-Port 21). Koneksi pada FTP melibatkan empat langkah, yaitu:

- a. Autentikasi *user*
- b. Membangun *control channel*
- c. Membangun *data channel*
- d. Memutus koneksi

FTP menggunakan koneksi kontrol TCP (*Transmission Control Protocol*) sebagai protokol transmisi yang akan memastikan kedatangan data pada penerima. Sehingga FTP tidak akan memusingkan tentang *packet loss* atau *error checking* selama transfer data. Secara sederhana dapat dikatakan bahwa TCP menggaransi bahwa paket data akan tiba hanya sekali tanpa *error* dengan urutan yang benar. Ada beberapa paket yang dapat digunakan untuk membuat FTP Server, antara lain *proftpd*, *vsftpd*, dan *ftpd*.

Setiap paket FTP Server mempunyai kelebihan dan kekurangannya masing-masing. FTP server terdiri dari dua bagian, yaitu : *server*, yang menanggapi permintaan klien seperti mengirim *file*, mendaftar direktori yang ada, dan lain-lain. Dan *filesystem* yang merupakan tempat FTP Server membaca *file* dan direktori yang ada. Program implementasi standar UNIX untuk menjalankan FTP server adalah *ftpd* (*ftp daemon*).

Ftpd berjalan di bawah inetd, UNIX *superserver*. Inetd mendengarkan permintaan pada sejumlah *port*, termasuk *port* kontrol FTP, TCP port 21. Ketika sebuah FTP Client pertama kali mencoba untuk berhubungan dengan sebuah FTP Server, dia mengirimkan paket ke *port* 21 pada *host* tujuannya. Kemudian inetd menerima paket tersebut, menentukan dari nomor *port* tujuan bahwa layanan FTP perlu dijalankan, dengan melakukan proses *forking* dan *executing* pada ftpd. Jika pada saat yang bersamaan ada klien lain yang ingin berhubungan dengan FTP Server, maka inetd akan meminta salinan dari ftpd yang sudah berjalan untuk menanganinya.

Untuk meningkatkan keamanan dan layanan FTP Server dengan proftpd, maka perlu dikombinasikan dengan mysql sebagai *database user* FTP. Sehingga semua pengguna FTP tersimpan pada basisdata di *mysql server*. Jika pengguna yang telah terdaftar ingin mengakses FTP Server maka akan diminta *username* dan *password* yang akan diautentikasi dengan *data user* yang ada di basisdata, sehingga tingkat keamanan lebih terjamin.

## 2. **User Dalam FTP**

Dalam penggunaan FTP dikenal dua akses bagi pemakai, yaitu:

### a. Regular FTP

Secara *default user* Linux dapat mengkopi *file* dari dan ke direktori *homenya* masing-masing menggunakan FTP Client dengan menggunakan *username* dan *password* sebagai syarat *loginnya*. *File-file* yang dapat diupload/download diletakkan di direktori *home* masing-masing *user* tersebut. *Username* dan *password* yang digunakan untuk *login* biasa di komputer tujuan. Kerugian

penggunaan Reguler FTP adalah tidak cocok untuk mendistribusikan *software* atau apapun yang umum digunakan oleh orang kebanyakan, karena untuk setiap orang harus memiliki *username* pada FTP Server.

#### b. Anonymous FTP

Adalah metode yang digunakan untuk mengakses FTP Server tanpa perlu memiliki *username* pada *server* tersebut. *Login* pada FTP Server dengan menggunakan *username* “*anonymous*” dengan *password* alamat *email* atau tanpa *password* sama sekali. *File-file* yang dapat diakses *user* ini diletakkan di direktori */var/ftp/*, sedangkan untuk *upload*, direktori yang disediakan adalah */var/ftp/pub*.

### 2.2.5 Proxy Server

#### 1. Pengertian proxy

Secara terminologi istilah Proxy dapat diartikan sebagai seseorang/lembaga yang bertindak sebagai perantara atau atas nama dari orang lain/lembaga dalam suatu hal. Namun dalam dunia jaringan komputer, istilah Proxy berarti komputer yang berfungsi sebagai perantara antara *client* dan *server* dalam jaringan komputer. Proxy dapat dipahami sebagai pihak ketiga yang berdiri di tengah-tengah antara kedua pihak yang saling berhubungan dan berfungsi sebagai perantara, sedemikian sehingga pihak pertama dan pihak kedua tidak secara langsung berhubungan, akan tetapi masing-masing berhubungan dengan perantara, yaitu Proxy.

Sebuah analogi, bila seorang mahasiswa meminjam buku di perpustakaan, kadang mahasiswa tersebut tidak diperbolehkan langsung mencari dan mengambil sendiri buku yang diinginkan dari rak, tetapi kita meminta buku tersebut kepada

petugas, tentu saja dengan memberikan nomor atau kode bukunya, dan kemudian petugas tersebut yang akan mencarinya dan mengambil bukunya.

Dalam kasus di atas, petugas perpustakaan tersebut telah bertindak sebagai perantara atau Proxy. Petugas tersebut juga bisa memastikan dan menjaga misalnya, agar mahasiswa hanya bisa meminjam buku untuk mahasiswa, dosen boleh meminjam buku semua buku, atau masyarakat umum hanya boleh meminjam buku tertentu.

Mungkin proses tersebut menjadi lebih lama dibandingkan bila langsung mencari dan mengambil sendiri buku yang diinginkan. Namun bila saja setiap kali petugas mencari dan mengambil buku untuk seseorang, petugas juga membuat beberapa salinan dari buku tersebut sebelum memberikan bukunya kepada orang yang meminta, dan menyimpannya di atas meja pelayanan, maka bila ada orang lain yang meminta buku tertentu, sangat besar kemungkinan buku yang diminta sudah tersedia salinannya di atas meja, dan petugas tinggal memberikannya langsung. Hasilnya adalah layanan yang lebih cepat dan sekaligus keamanan yang baik.

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

bekerja sebagai Proxy pada layer Aplikasi, meskipun juga akan dibahas mengenai Proxy pada tingkat sirkuit.

Dalam suatu jaringan lokal yang terhubung ke jaringan lain atau Internet, pengguna tidak langsung berhubungan dengan jaringan luar atau Internet, tetapi harus melewati suatu *gateway*, yang bertindak sebagai batas antara jaringan lokal dan jaringan luar. *Gateway* ini sangat penting, karena jaringan lokal harus dapat dilindungi dengan baik dari bahaya yang mungkin berasal dari Internet, dan hal tersebut akan sulit dilakukan bila tidak ada garis batas yang jelas jaringan lokal dan Internet.

*Gateway* juga bertindak sebagai titik dimana sejumlah koneksi dari pengguna lokal akan terhubung kepadanya, dan suatu koneksi ke jaringan luar juga terhubung kepadanya. Dengan demikian, koneksi dari jaringan lokal ke Internet akan menggunakan sambungan yang dimiliki oleh *gateway* secara bersama-sama (*connection sharing*). Dalam hal ini, *gateway* adalah juga sebagai Proxy Server, karena menyediakan layanan sebagai perantara antara jaringan lokal dan jaringan luar atau Internet.

## 2. Fungsi Proxy

Analogi di atas menjelaskan konsep dan fungsi dasar dari suatu Proxy dalam komunikasi jaringan komputer dan Internet. Proxy Server mempunyai 3 fungsi utama, yaitu :

### a. Firewall

Proxy Server yang dikonfigurasi secara benar, akan meningkatkan performa dan keamanannya. Karena Proxy bekerja pada layer aplikasi (dalam OSI Layer),

maka *filtering* yang dilakukan oleh Proxy lebih “cerdas” daripada *firewall* biasa. Proxy Web Server dapat mengecek URL dari *outgoing request* (permintaan akses keluar) untuk halaman *web* dengan memeriksa pesan HTTP, GET, dan POST. Dengan kemampuan ini, administrator dapat melarang atau mengizinkan akses ke *domain* tertentu. *Firewall* biasa, tidak dapat melihat nama *domain* di dalam pesan tersebut, karena *firewall* hanya memeriksa *header* dari paket data.

#### b. Gateway

Untuk dapat mengakses Internet, sebuah komputer harus memiliki sebuah IP Public. Hal ini menjadi mustahil karena IP yang tersedia di dunia sangat terbatas, sehingga untuk dapat mengakses Internet secara bersama-sama dengan menggunakan satu IP Public, dibutuhkan komputer yang memiliki IP Public, yang digunakan sebagai *gateway* komputer-komputer lain. Dalam hal ini Proxy Server juga berfungsi sebagai *gateway*. *Server* ini mempunyai dua *interface*, satu untuk antarmuka dengan Internet dan satu untuk antarmuka dengan jaringan lokal.

#### c. Cache

Fungsi Proxy Server yang lain adalah untuk *web caching*. *Caching* di sini diartikan sebagai penyimpanan *internet object* (gambar/halaman *web*) dari suatu *website* 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 *website*. Begitu pula jika Web Server yang dituju ternyata mati atau mengalami gangguan, *client* tetap bisa mengaksesnya.

### 3. Koneksi antar Proxy

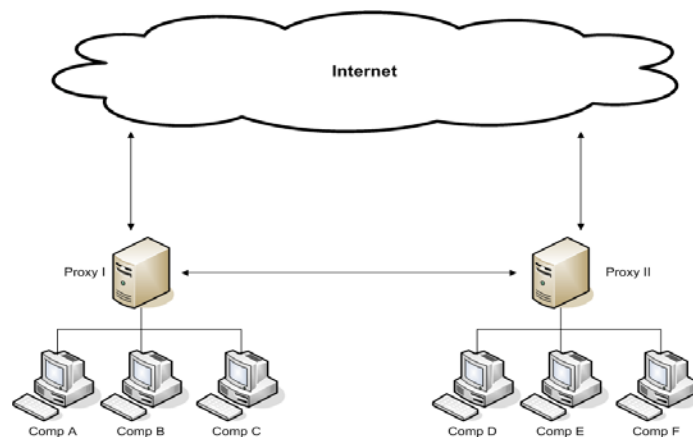
Sebuah Proxy Server dapat dikoneksikan dengan Proxy Server lain. Sedangkan hubungan antar *server* tersebut ada dua macam:

#### a. Sibling

Seperti ditunjukkan pada gambar 2.5, bila komputer A, komputer B, komputer C hendak mengakses Internet, maka ia akan mencari objek yang dicari apakah sudah tersimpan di Proxy I. Bila Proxy tidak menyimpan objek yang dicari, maka Proxy I akan mencari objek yang dicari di Proxy II. Bila tidak ada juga, maka Proxy I akan *download* langsung dari Internet.

#### b. Parent-Child

Seperti ditunjukkan pada gambar 2.5, bila komputer A, komputer B, komputer C hendak mengakses Internet, maka ia akan mencari objek yang dicari apakah sudah tersimpan di Proxy I. Bila Proxy tidak menyimpan objek yang dicari, maka Proxy I akan mencari objek yang dicari di Proxy II. Bila tidak ada juga, maka Proxy II akan *download*kannya langsung dari Internet untuk kemudian diberikan pada Proxy I.



Gambar 2.5 Koneksi antar Proxy

## **BAB III**

### **METODOLOGI**

#### **3.1 Analisis Kebutuhan Server**

Analisis kebutuhan *server* merupakan bagian yang penting dalam rancang bangun sebuah sistem *server*, karena tahap inilah konsep awal sebuah *server* akan dibangun sehingga kita bisa mendapatkan rancangan awal *server* yang akan kita buat serta langkah-langkah apa saja yang akan kita lakukan dalam proses implementasi *server*.

#### **3.2 Metode Analisis**

Metode yang digunakan dalam implementasi Analisis Perbandingan Kinerja Virtual Server dengan Dedicated Server Berbasis Template Unix dari cacti adalah dengan merancang skema jaringannya serta membuat simulasi pengujian fungsi *server* secara *virtual*.

#### **3.3 Hasil Analisis**

Dalam simulasi ini dibangun tiga buah mesin *virtual* yang terdiri dari satu buah mesin *server* yang di dalamnya dibangun DHCP server dan Proxy server, satu buah mesin *server* yang di dalamnya dibangun DNS server, Web server, dan FTP server, dan satu buah *benchmarking server* sebagai *throughput* hasil simulasi pengujian *server*.

Untuk uji coba pengaksesan digunakan satu buah komputer klien terpisah dan satu klien dalam komputer yang sama dengan mesin *virtual*. Untuk

melakukan akses puluhan hit ke *server* yang dilakukan secara simulasi dengan menggunakan *tools* yang mampu mengakses *server* sebanyak proses yang diinginkan.

### 3.4 Perangkat lunak yang dibutuhkan

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

1. Windows Vista Home Basic SP1 sebagai sistem operasi inti
2. VMware Workstation 6 sebagai representasi Virtual Machine
3. Fedora 3 sebagai sistem operasi pada Virtual server
4. Ubuntu Server sebagai representasi sistem operasi *benchmarking server*
5. Cacti untuk menilai *throughput* paket data dan CPU *utilization*
6. MySql
7. Apache2
8. PHP5
9. PHPMyAdmin
10. Squid sebagai Proxy server

*Software* merupakan sebuah aplikasi yang akan diinstal ke dalam sistem operasi. Untuk mengimplementasikan Proxy server menggunakan aplikasi yang dinamakan Squid. Squid merupakan Proxy server dengan kemampuan luar biasa. Selain mampu membagi akses Internet ke banyak pengguna (*router*) dan menyimpan aktivitas pengguna dalam *cache* (agar akses informasi lebih cepat diperoleh), Squid juga mampu mengelola lalu lintas permintaan data dari pengguna ke Internet dan membatasinya jika diperlukan.

Pengelolaan tersebut kadangkala diperlukan karena manfaat Internet secara bersama-sama seringkali tidak dapat dirasakan secara optimal. Lalu lintas permintaan data dari Internet untuk masing-masing pengguna tidak diatur berdasarkan kebutuhan dan kondisi *bandwidth* yang dimiliki, bahkan pemanfaatan Internet lebih banyak didominasi oleh pengguna yang tidak berhak. Untuk itu diperlukan suatu manajemen agar Internet bersama dapat dinikmati oleh pengguna dengan adil berdasarkan kuotanya masing-masing. Setiap pengguna memperoleh hak akses Internet sesuai dengan bidang pekerjaannya atau kebutuhan yang seharusnya diperoleh.

#### 11. Dhcpd sebagai DHCP server

Dhcpd adalah perangkat lunak yang digunakan untuk membangun sebuah DHCP server. Dhcpd mampu memberikan layanan yang secara otomatis memberikan nomor IP kepada komputer yang memintanya. Setelah dhcpd memberikan nomor IP, maka dhcpd meminjamkan (*lease*) nomor IP yang ada ke DHCP *client* dan mencoret nomor IP tersebut dari daftar *pool*. Nomor IP diberikan bersama dengan *subnet mask* dan *default gateway*. Jika tidak ada lagi nomor IP yang dapat diberikan, maka *client* tidak dapat menginisialisasi TCP/IP, dengan sendirinya tidak dapat tersambung pada jaringan tersebut. Setelah periode waktu tertentu, maka pemakaian DHCP *client* tersebut dinyatakan selesai dan *client* tidak memperbaharui permintaan kembali, maka nomor IP tersebut dikembalikan kepada dhcpd, dan dhcpd dapat memberikan nomor IP tersebut kepada *client* yang membutuhkan. Lama periode ini dapat

ditentukan dalam menit, jam, bulan atau selamanya. Jangka waktu tersebut disebut dengan *leased period*.

#### 12. Apache sebagai Web server

Apache merupakan salah satu perangkat lunak yang biasa digunakan untuk layanan Web server. Apache dapat didownload secara gratis dari web resmi yaitu <http://www.apache.org>. Perangkat lunak Apache merebut pasar dunia, karena Apache merupakan perangkat lunak yang *open source* dan fleksibel terhadap berbagai sistem operasi (Windows dan Linux). Apache merupakan turunan dari Web server yang dikeluarkan oleh NCSA HTTPd pada tahun 1995. Kelebihan dari Web server Apache :

- a. *Open source* dan *freeware*.
- b. Mampu berjalan pada berbagai *platform* sistem operasi
- c. Instalasi dan konfigurasi cukup mudah.

#### 13. Vsftpd sebagai FTP server

Vsftpd merupakan salah satu aplikasi yang digunakan untuk membuat *server* FTP. Aplikasi yang lainnya : proftpd, pure-ftp, wu-ftp, dan lain sebagainya. Keunggulan vsftpd dibanding aplikasi FTP yang lain :

- a. *Freeware*
- b. Mudah instalasi dan konfigurasi
- c. Lebih mudah, aman, dan stabil

#### 14. Bind sebagai DNS server

Pada sistem operasi UNIX, DNS diimplementasikan dengan menggunakan *software* Berkeley Internet Name Domain (BIND). BIND ini memiliki dua

sisi, yaitu sisi *client* dan sisi *server*. Sisi *client* disebut *resolver*. *Resolver* ini bertugas membangkitkan pertanyaan mengenai informasi *domain name* yang dikirimkan kepada sisi *server*. Sisi *server* BIND ini adalah sebuah *daemon* yang disebut **named**. Ia yang akan menjawab *query-query* dari *resolver* yang diberikan kepadanya.

Pada saat BIND dijalankan, ia memiliki 4 modus operasi, yaitu :

*a. Resolver-only*

Komputer hanya membangkitkan *query* informasi *domain name* kepada sebuah DNS server dan tidak menjalankan fungsi DNS server.

*b. Caching-only*

Komputer menjalankan fungsi *name server* tetapi tidak memiliki *database* DNS server. Ia hanya mempelajari jawaban-jawaban *query* yang diberikan oleh *remote* DNS server dan menyimpannya dalam memori. Data-data dalam memori tersebut akan digunakan untuk menjawab *query* selanjutnya yang diberikan kepadanya.

*c. Primary server*

Komputer menjalankan fungsi *name server* berdasarkan *database* yang dimilikinya. *Database* ini dibangun oleh administrator DNS. Server ini menjadi *authoritative source* bagi domain tertentu.

*d. Secondary server*

Komputer menjalankan fungsi *name server* berdasarkan *database* yang diambil dari *primary server*. Proses pengambilan *file database* ini sering

disebut *zone file transfer*. Ia juga menjadi *authoritative source* bagi domain tersebut.

### 3.5 Perangkat keras yang dibutuhkan

Perangkat keras penunjang yang diperlukan pada proses menganalisis perbandingan kinerja antara Virtual Server dengan Dedicated Server adalah sebagai berikut:

#### 1. Komputer Virtual Server

Komputer sejumlah satu unit yang diperuntukkan sebagai Virtual server, dengan spesifikasi sebagai berikut:

- a. Prosesor Intel DualCore 2,17 GHz
- b. Memori 2 GB DDR2
- c. Media penyimpanan sebesar 160 GB HDD
- d. DVD-ROM
- e. Kartu jaringan

#### 2. Komputer klien

Komputer yang akan difungsikan sebagai klien untuk melakukan pengujian, yang memiliki spesifikasi komputer standar yang dapat terkoneksi ke jaringan komputer. Komputer klien yang akan digunakan sejumlah 2 buah.

#### 3. Kebutuhan *virtual machine*

Perangkat lunak yang digunakan sebagai *virtual machine* adalah VMware Workstation 6 dengan sistem operasi Fedora Core 3 untuk Virtual server dan Ubuntu Server untuk *benchmarking server*, dengan kebutuhan spesifikasi sebagai berikut:

- a. Mempunyai tempat penyimpanan 4 GB
- b. Mempunyai memori 512 MB
- c. Jaringan terkoneksi secara *bridged*

### **3.6 Instalasi Perangkat Lunak**

Tahapan-tahapan yang dilakukan dalam proses instalasi pada perangkat lunak terkait adalah sebagai berikut:

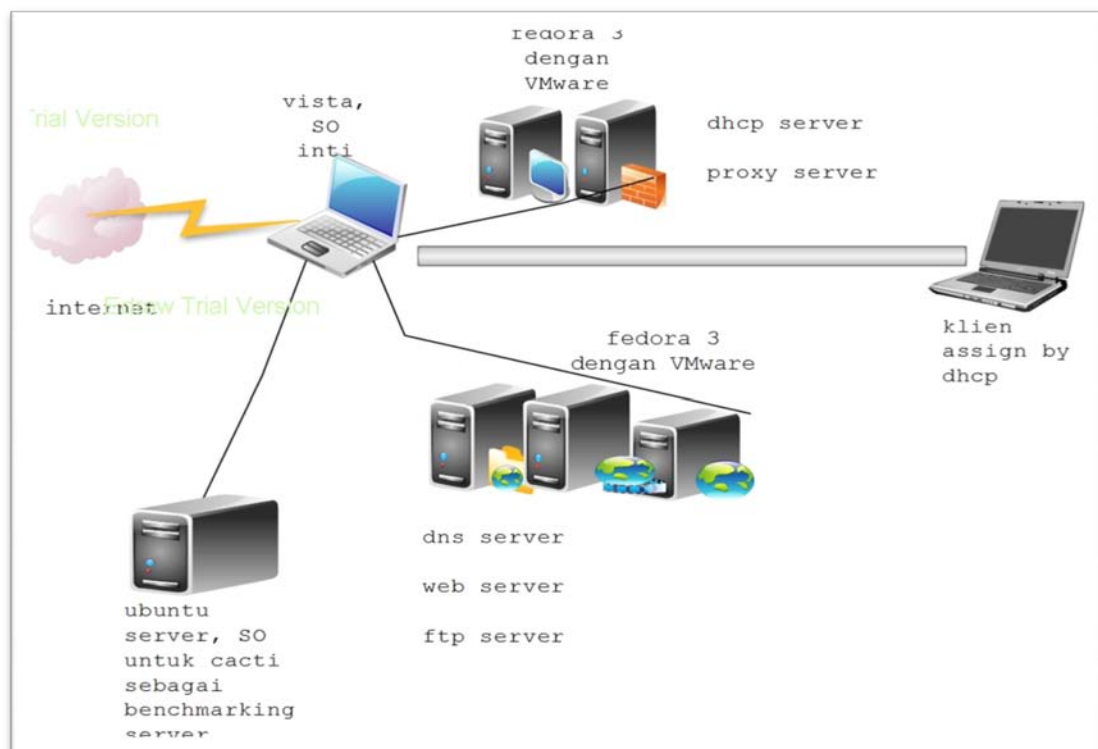
1. Instalasi sistem operasi Windows Vista Home Basic SP1 pada komputer *server* sebagai sistem operasi inti.
2. Instalasi VMware Workstation 6 pada komputer *server*.
3. Instalasi sistem operasi Fedora Core 3 pertama pada VMware sebagai Proxy server dan DHCP server.
4. Instalasi Proxy server dan DHCP server pada Fedora pertama.
5. Instalasi sistem operasi Fedora Core 3 kedua pada VMware sebagai Web server, DNS server, dan FTP server.
6. Instalasi Web server, DNS server, dan FTP server pada Fedora kedua.
7. Konfigurasi seluruh *server* yang telah diinstal.
8. Instalasi sistem operasi Ubuntu Server pada Vmware sebagai *benchmarking server*.
9. Instalasi Cacti pada Ubuntu Server sebagai *benchmarking tool*.

### **3.7 Perancangan Network Diagram**

Network Diagram adalah sebuah pemetaan jaringan yang dilakukan seorang administrator untuk merepresentasikan kegiatan-kegiatan yang akan dilakukan. Network Diagram sangat berguna ketika seorang administrator akan menyusun

sebuah jaringan komputer, karena semua aktifitas akan ditampilkan di dalam Network Diagram.

Berikut merupakan gambaran umum dari Network diagram yang akan dibangun:



Gambar 3.1 Network Diagram *virtual server*

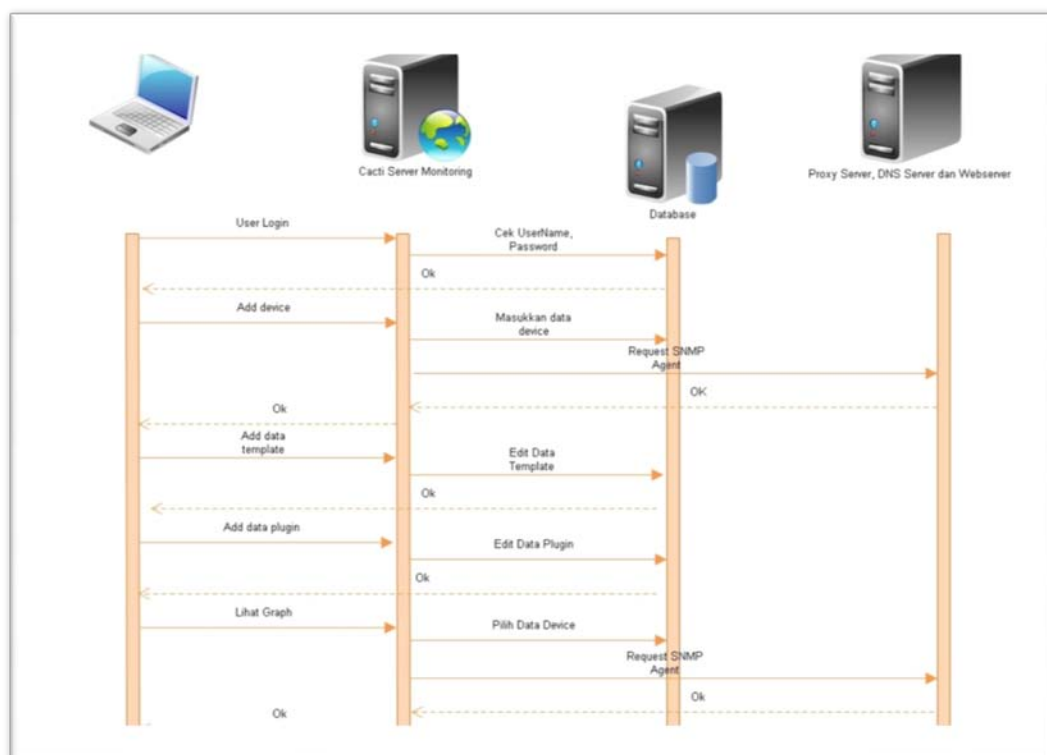
Seperti dapat dilihat pada Gambar 3.1, bahwa semua *server* dibangun dalam satu mesin fisik. Di dalam mesin fisik tersebut terdapat tiga mesin *virtual* yang masing-masing digunakan untuk sistem operasi fedora 3 dengan DHCP server dan Proxy server, sistem operasi fedora 3 dengan DNS server, Web server, dan FTP server, serta sistem operasi Ubuntu server dengan Cacti sebagai *benchmarking tool*. Semua *server* tersebut terhubung melalui sistem operasi Vista sebagai komputer inti secara *bridged* dan dihubungkan kepada Internet dengan menggunakan kabel RJ-45 melalui gateway Lab Sisjarkom. Klien yang digunakan

untuk menguji fungsi *server* dan meng-*hit request* dapat terhubung secara *host-only* dengan *virtual machine* atau melalui kabel LAN atau melalui *hotspot* kampus, dan untuk memonitoring *request* dari klien digunakan Cacti sebagai *benchmarking* hasil analisis nantinya.

### 3.8 Perancangan Sequence Diagram

Sequence Diagram merupakan gambaran interaksi antar objek dalam dan di sekitar sistem yang saling berkomunikasi menggunakan pesan dan memiliki parameter waktu. Fungsi utama Sequence Diagram yaitu untuk menggambarkan skenario atau urutan langkah-langkah yang dilakukan sebagai respon yang memicu aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan keluaran apa yang dihasilkan.

Berikut merupakan gambaran umum dari Sequence Diagram berbasis Cacti:



Gambar 3.2 Sequence Diagram berbasis Cacti

Pada Gambar 3.2 dapat dilihat bahwa cara kerja dari Cacti ini adalah dapat melakukan proses *login user*, dengan melakukan proses autentifikasi *user* terhadap *database* yang dimiliki. Dapat dilihat proses *add data device* juga melibatkan *database* yang berfungsi untuk menyimpan semua data yang berkaitan dengan *server* yang dimonitoring. Selain itu Cacti juga dapat melakukan proses penambahan modul *plugin*.

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Tahapan-tahapan Analisis

Dalam implementasi analisis perbandingan kinerja Virtual Server dengan Dedicated Server berbasis Template Unix dari Cacti ini, akan dibagi menjadi tiga tahapan proses, yaitu tahap instalasi, tahap pengujian, dan tahap pembahasan hasil pengujian/analisis. Tahap instalasi dan konfigurasi sendiri terbagi menjadi sembilan tahap, yaitu sebagai berikut :

1. Tahap instalasi Vmware Workstation 6.

Pada tahap ini dilakukan instalasi perangkat lunak VMware Workstation 6 sebagai *virtual machine* yang akan menjalankan *guest OS*. Ada tiga *virtual machine* yang akan dibuat, yaitu dua VMware untuk Virtual server dan satu VMware untuk *benchmarking server*.

2. Tahap instalasi sistem operasi Fedora 3.

Pada tahap ini dilakukan instalasi sistem operasi Fedora 3 sebagai *guest OS*. Sistem operasi Fedora 3 ini digunakan untuk membuat *server-server* yang akan dianalisis, yaitu DHCP server, Proxy server, DNS server, Web server, dan FTP server.

3. Tahap instalasi Ubuntu server.

Pada tahap ini dilakukan instalasi Ubuntu server pada *virtual machine* untuk menjalankan Cacti sebagai *benchmarking server*.

4. Tahap pembuatan DHCP server.

Pada tahap ini dilakukan proses pembuatan DHCP server dengan `dhcpd` sebagai aplikasinya. DHCP server dibuat pada *virtual machine* pertama. Pembuatan DHCP server ini menggunakan konfigurasi-konfigurasi dasar. Hal khusus yang ditambahkan adalah protokol SNMP, agar dapat dimonitoring oleh Cacti.

5. Tahap pembuatan Proxy server.

Pada tahap ini dilakukan proses pembuatan Proxy server dengan Squid sebagai aplikasinya. Server Proxy ini dibuat dalam satu mesin *virtual* dengan DHCP server (*virtual machine* pertama), hal ini dikarenakan utilitas dan fungsi mereka yang saling berkaitan.

6. Tahap pembuatan DNS server.

Pada tahap ini dilakukan proses pembuatan DNS server dengan BIND sebagai aplikasinya. DNS server dibuat pada *virtual machine* kedua. Protokol SNMP juga diaktifkan pada mesin *virtual* ini.

7. Tahap pembuatan Web server.

Pada tahap ini dilakukan proses pembuatan Web server dengan Apache sebagai aplikasinya. Web server ini dibuat dalam satu mesin *virtual* dengan DNS server (*virtual machine* kedua), hal ini dikarenakan utilitas dan fungsi mereka yang saling berkaitan.

8. Tahap pembuatan FTP server.

Pada tahap ini dilakukan proses pembuatan FTP server dengan `vsftpd` sebagai aplikasinya. FTP server juga dibuat pada *virtual machine* kedua.

## 9. Tahap instalasi Cacti.

Pada tahap ini dilakukan instalasi Cacti pada Ubuntu server sebagai *benchmarking server* yang akan memonitoring dua mesin Virtual server dan Dedicated server yang telah dibuat sebelumnya. Pada tahap ini meliputi penginstalan paket-paket pendukung Cacti, seperti Mysql, Apache2, PHP5, dan PHPMyAdmin.

## 4.2 Konfigurasi dan Instalasi

### 4.2.1 Konfigurasi DHCP server

Lakukan proses instalasi paket, dikarenakan proses instalasi ini lakukan secara *online*.

Berikut perintahnya adalah :

```
#rpm -qa | grep dhcpd
#rpm -ivh dhcp-3.0.1-11.i386.rpm
```

Selanjutnya setelah terinstal, salin contoh konfigurasi dari dhcpd yang telah ada ke dalam konfigurasi dhcpd yang akan kita buat.

Berikut perintahnya adalah :

```
#cp /usr/share/doc/dhcp-3.0.1/dhcpd.conf.sample /etc/dhcpd.conf
```

Setelah disalin, masuk ke file konfigurasi utama dari dhcpd.conf dan kemudian edit konfigurasi dhcpd tersebut.

Berikut perintahnya adalah :

```
#vi /etc/dhcpd.conf
```

Langkah selanjutnya adalah menjalankan *service* dari dhcpd.

Berikut perintah nya adalah :

```
#service dhcpd start
```

Apabila tampilannya seperti pada gambar 4.1, maka sudah bisa dipastikan bahwa *service* dari *dhcpd* telah berjalan dengan baik.

```
[root@router_proxy_dhcp ~]# service dhcpd start
Starting dhcpd: [ OK ]
[root@router_proxy_dhcp ~]# _
```

Gambar 4.1 Menjalankan *service dhcpd*

#### 4.2.2 Konfigurasi Proxy server

Lakukan proses instalasi paket, dikarenakan proses instalasi ini lakukan secara *online*. Berikut perintahnya adalah :

```
#rpm -qa | grep squid
#rpm -ivh squid-2.5.STABLE6-3
```

Edit hak akses file *squid.conf*, berikut perintahnya adalah :

```
#chmod 740 /etc/squid/squid.conf
```

Lakukan konfigurasi (edit file) pada *squid.conf*, berikut perintahnya adalah :

```
#vi /etc/squid/squid.conf
```

Langkah selanjutnya adalah menjalankan *service* dari *squid*.

Berikut perintahnya adalah :

```
#service squid start
```

Apabila tampilannya seperti pada gambar 4.2, maka sudah bisa dipastikan bahwa *service* dari *squid* telah berjalan dengan baik.

```
[root@router_proxy_dhcp ~]# service squid start
Starting squid: . [ OK ]
```

Gambar 4.2 Menjalankan *service squid*

### 4.2.3 Konfigurasi DNS server

Untuk mengimplementasikan DNS digunakan perangkat lunak BIND. BIND bekerja menggunakan *daemon* yang terkenal dengan sebutan **named**. **Named** bekerja secara *background* dan mendengarkan permintaan *client* pada *port* 53.

Langkah awal adalah dengan menginstal paket rpm-nya.

Berikut perintahnya adalah sebagai berikut:

```
#rpm -ivh bind-libs-9.2.4-2.i386.rpm bind-utils-9.2.4-2.i386.rpm
bind-9.2.4-2.i386.rpm bind-chroot-9.2.4-2.i386.rpm
```

Lakukan konfigurasi (edit file) pada `named.conf`, berikut perintahnya adalah :

```
#vi /var/named/chroot/etc/named.conf
```

Lakukan konfigurasi (edit file) pada *reverse file* (`db.192.168.10`), berikut perintahnya adalah :

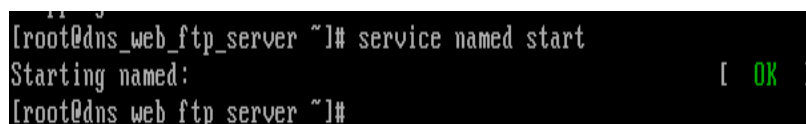
```
#vi /var/named/chroot/var/named/db.192.168.10
```

Setelah itu lakukan konfigurasi pada file `resolv.conf`.

Berikut perintahnya adalah :

```
#vi /etc/resolv.conf
```

Apabila tampilannya seperti pada gambar 4.3, maka sudah bisa dipastikan bahwa *service* dari `named` telah berjalan dengan baik.



```
[root@dns_web_ftp_server ~]# service named start
Starting named: [ OK ]
[root@dns_web_ftp_server ~]# _
```

Gambar 4.3 Menjalankan *service* named

#### 4.2.4 Konfigurasi Web server

Web server merupakan *server* yang digunakan untuk menjalankan *web*. Untuk menjalankan Web server sendiri diperlukan perangkat lunak pendukung lainnya, seperti Apache. Paket yang dibutuhkan untuk menginstal Apache antara lain:

- `apr-0.9.4-23.i386.rpm`
- `apr-util-0.9.4-17.i386.rpm`
- `httpd-2.0.52-3.i386.rpm`
- `httpd-suexec-2.0.52-3.i386.rpm`

Untuk melakukan penginstalan lakukan langkah-langkah sebagai berikut:

Cek apakah sudah terdapat paket Apache pada sistem operasi.

Berikut perintahnya adalah:

```
#rpm -qa | grep namapaket
```

Jika belum ada, instal paket tersebut dengan :

```
#rpm -ivh apr-0.9.4-23.i386.rpm apr-util-0.9.4-17.i386.rpm httpd-  
2.0.52-3.i386.rpm httpd-suexec-2.0.52-3.i386.rpm
```

Setelah itu lakukan konfigurasi pada file `httpd.conf`.

Berikut perintahnya adalah :

```
#vi /etc/httpd/conf/httpd.conf
```

Pastikan terdapat baris:

```
DocumentRoot "/var/www/html"
```

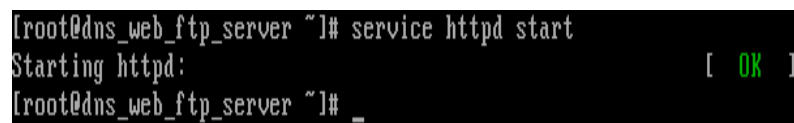
Baris di atas berarti bahwa lokasi penyimpanan dokumen *web* akan diletakkan pada direktori `/var/www/html/`.

Buat sebuah dokumen **index.html** sederhana sebagai file ujicoba yang akan ditempatkan pada `/var/www/html/`.

Berikut perintahnya adalah :

```
#vi /var/www/html/index.html
```

Apabila tampilannya seperti pada gambar 4.4, maka sudah bisa dipastikan bahwa *service* dari `httpd` telah berjalan dengan baik.



```
[root@dns_web_ftp_server ~]# service httpd start
Starting httpd: [ OK ]
[root@dns_web_ftp_server ~]# _
```

Gambar 4.4 Menjalankan *service* `httpd`

#### 4.2.5 Konfigurasi FTP server

Langkah pertama yang dilakukan adalah mengecek apakah sudah terdapat paket Apache pada sistem operasi.

Berikut perintahnya adalah :

```
#rpm -qa | grep vsftpd
```

Jika belum ada, instal paket tersebut dengan perintah sebagai berikut :

```
#rpm -ivh vsftpd-2.0.1-5.i386.rpm
```

Setelah itu lakukan konfigurasi pada file `vsftpd.conf`.

Berikut perintahnya adalah :

```
#vi /etc/vsftpd/vsftpd.conf
```

Hilangkan tanda pagar (#) untuk mengaktifkan kebijakan-kebijakan seperti :

```
anonymous_enable = YES
```

- mengijinkan *login* secara *anonymous*

```
local_enable = YES
```

- mengijinkan *login* FTP di satu komputer

```
write_enable = YES
```

- mengizinkan melakukan penulisan pada file-file di *server*

```
anon_upload_enable = YES
```

- mengizinkan *upload* file oleh *anonymous*

```
anon_mkdir_write_enable = YES
```

- mengizinkan *user anonymous* membuat direktori

Apabila tampilannya seperti pada gambar 4.5, maka sudah bisa dipastikan bahwa *service* dari *vsftpd* telah berjalan dengan baik.

```
[root@dns_web_ftp_server ~]# service vsftpd start
Starting vsftpd for vsftpd: [ OK ]
[root@dns_web_ftp_server ~]# _
```

Gambar 4.5 Menjalankan *service vsftpd*

### 4.3 Hasil pengujian dan pembahasan

Pengujian ini dilakukan terhadap *server-server* yang telah dibangun sebelumnya dengan menggunakan *Cacti* sebagai *monitoring tool*. Keadaan pada saat monitoring tidak secara *real time*, hal ini dikarenakan *interval pool* (jarak waktu) maksimal pada *Cacti* adalah satu menit, yang berarti *Cacti* menampilkan grafik monitoring satu menit sebelumnya. Monitoring *server-server* ini dilakukan pada waktu yang bersamaan, yaitu pada tanggal 28 Desember 2009 selama 13 jam 30 menit. Grafik-grafik yang *dicapture* antara lain adalah sebagai berikut :

#### 1. Memory Usage

Besarnya memori yang digunakan pada saat *swapping* dan pada saat tidak digunakan dalam satuan *kilobytes*.

#### 2. Load Average

Rata-rata jumlah proses yang sedang berjalan dan diakses.

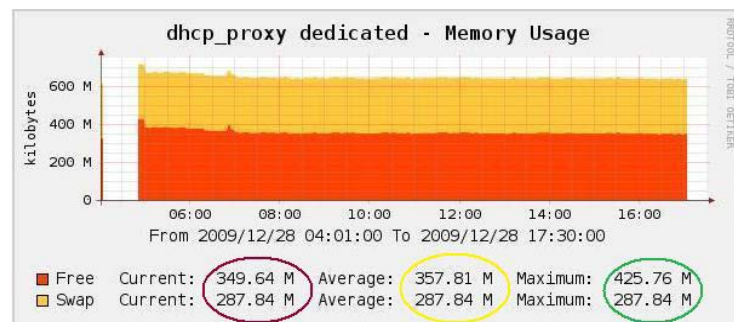
### 3. Ping Latency

Besarnya waktu tunda dalam satuan *milliseconds* yang dilakukan pada mesin *server* tersebut.

### 4. Processes

Besarnya jumlah proses saat *server* dalam keadaan hidup.

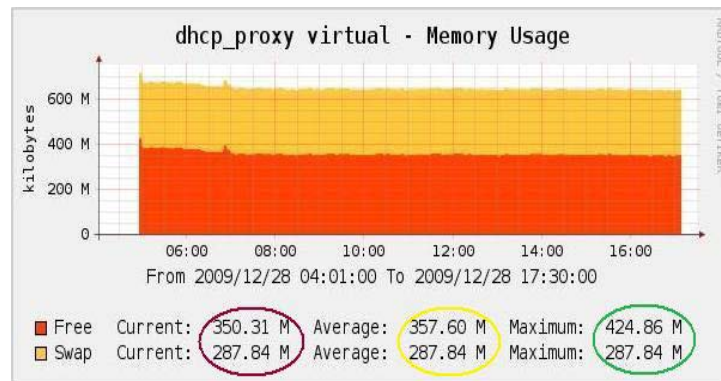
#### 4.3.1 Memory Usage



Gambar 4.6 Memory Usage DHCP-Proxy dedicated server

Keterangan gambar 4.6 :

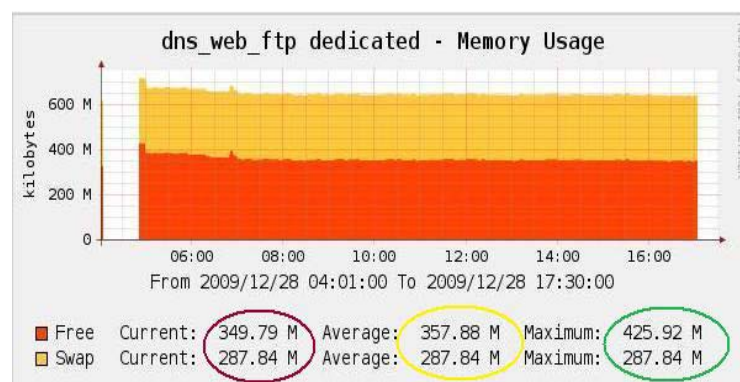
- Pada saat pengukuran, seperti ditunjukkan lingkaran merah pada gambar: besarnya memori yang digunakan proses *swapping* adalah sebesar 287.84MB, sedangkan pada saat *free* adalah sebesar 349.64MB.
- Rata-rata besarnya memori, seperti ditunjukkan lingkaran kuning pada gambar: pada saat melakukan proses *swapping* adalah sebesar 287.84MB, sedangkan pada saat *free* adalah sebesar 357.81MB.
- Penggunaan memori tertinggi, seperti ditunjukkan lingkaran hijau pada gambar: pada saat melakukan proses *swapping* adalah sebesar 287.84MB, sedangkan pada saat *free* adalah sebesar 425.76MB.



Gambar 4.7 Memory Usage DHCP-Proxy virtual server

Keterangan gambar 4.7 :

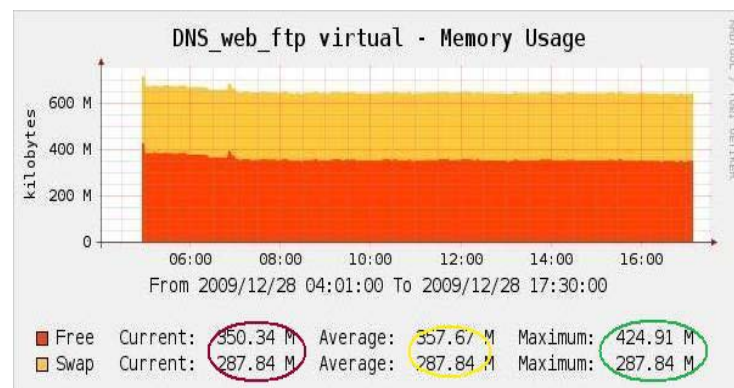
- Pada saat pengukuran, seperti ditunjukkan lingkaran merah pada gambar: besarnya memori yang digunakan proses *swapping* adalah sebesar 287.84MB, sedangkan pada saat *free* adalah sebesar 350.31MB.
- Rata-rata besarnya memori, seperti ditunjukkan lingkaran kuning pada gambar: pada saat melakukan proses *swapping* adalah sebesar 287.84MB, sedangkan pada saat *free* adalah sebesar 357.60MB.
- Penggunaan memori tertinggi, seperti ditunjukkan lingkaran hijau pada gambar: pada saat melakukan proses *swapping* adalah sebesar 287.84MB, sedangkan pada saat *free* adalah sebesar 424.86MB.



Gambar 4.8 Memory Usage DNS-Web-FTP dedicated server

Keterangan gambar 4.8 :

- Pada saat pengukuran, seperti ditunjukkan lingkaran merah pada gambar: besarnya memori yang digunakan proses *swapping* adalah sebesar 287.84MB, sedangkan pada saat *free* adalah sebesar 349.79MB.
- Rata-rata besarnya memori, seperti ditunjukkan lingkaran kuning pada gambar: pada saat melakukan proses *swapping* adalah sebesar 287.84MB, sedangkan pada saat *free* adalah sebesar 357.88MB.
- Penggunaan memori tertinggi, seperti ditunjukkan lingkaran hijau pada gambar: pada saat melakukan proses *swapping* adalah sebesar 287.84MB, sedangkan pada saat *free* adalah sebesar 425.92MB.



Gambar 4.9 Memory Usage DNS-Web-FTP virtual server

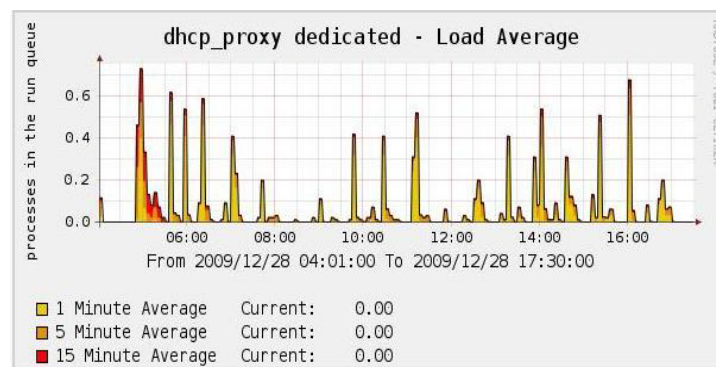
Keterangan gambar 4.9 :

- Pada saat pengukuran, seperti ditunjukkan lingkaran merah pada gambar: besarnya memori yang digunakan proses *swapping* adalah sebesar 287.84MB, sedangkan pada saat *free* adalah sebesar 350.34MB.
- Rata-rata besarnya memori, seperti ditunjukkan lingkaran kuning pada gambar: pada saat melakukan proses *swapping* adalah sebesar 287.84MB, sedangkan pada saat *free* adalah sebesar 357.67MB.

- c. Penggunaan memori tertinggi, seperti ditunjukkan lingkaran hijau pada gambar: pada saat melakukan proses *swapping* adalah sebesar 287.84MB, sedangkan pada saat *free* adalah sebesar 424.91MB.

Besarnya memori utama (memori fisik) pada keempat *server* tersebut adalah 512MB, dan besarnya memori *swap* adalah dua kali memori utama yaitu sebesar 1GB. Dilihat dari keempat gambar tersebut, semua penggunaan besarnya memori *swap* adalah sama besar dan penggunaan memori utama yang tidak jauh berbeda, hal ini dikarenakan tidak banyaknya program yang berjalan sehingga tidak sampai melebihi kapasitas memori utama. Jadi, semakin besar nilai proses *swap* semakin kurang baik, karena terlalu banyaknya program yang berjalan dan kapasitas memori utama yang tidak cukup untuk menampung.

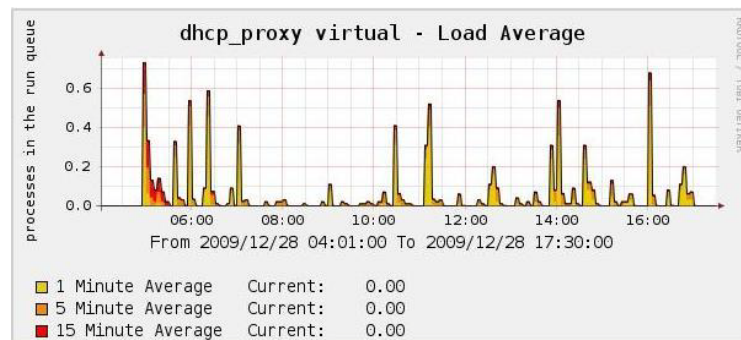
#### 4.3.2 Load Average



Gambar 4.10 Load Average DHCP-Proxy dedicated server

Keterangan gambar 4.10 :

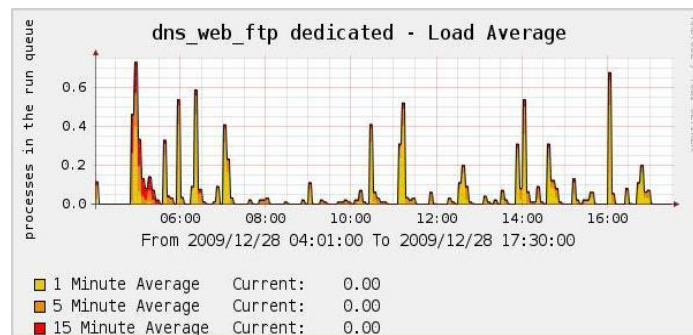
- Rata-rata besarnya load: di atas 0.2 bits per second
- Akses load tertinggi: 0.8 bits per second



Gambar 4.11 Load Average DHCP-Proxy virtual server

Keterangan gambar 4.11 :

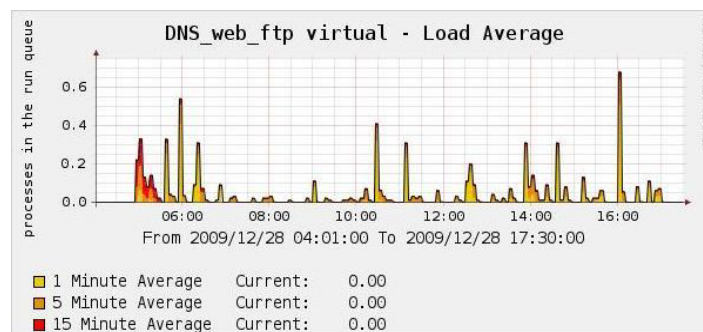
- Rata-rata besarnya load: di bawah 0.2 bits per second
- Akses load tertinggi: 0.8 bits per second



Gambar 4.12 Load Average DNS-Web-FTP dedicated server

Keterangan gambar 4.12 :

- Rata-rata besarnya load: di bawah 0.2 bits per second
- Akses load tertinggi: 0.8 bits per second



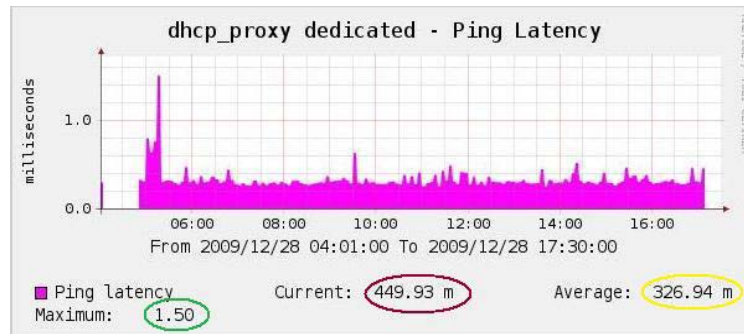
Gambar 4.13 Load Average DNS-Web-FTP virtual server

Keterangan gambar 4.13 :

- a. Rata-rata besarnya load: di bawah 0.2 bits per second
- b. Akses load tertinggi: 0.7 bits per second

Rata-rata besarnya load tergantung pada seberapa sering klien mengakses *server* tersebut. Semakin besar rata-rata nilai load pada satu *server*, berarti semakin sering *server* tersebut diakses oleh klien. Nilai akses load tertinggi terjadi ketika banyak klien mengakses *server* tersebut pada waktu yang bersamaan. Jadi, semakin tinggi akses load semakin baik, semakin banyak klien yang mengakses *server* tersebut pada waktu yang bersamaan.

### 4.3.3 Ping Latency

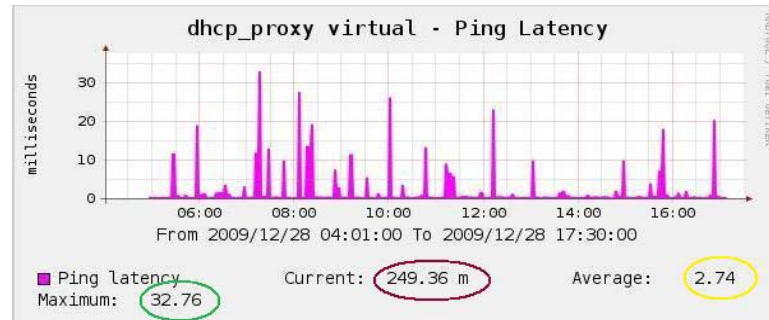


Gambar 4.14 Ping Latency DHCP-Proxy dedicated server

Keterangan gambar 4.14 :

- a. Pada saat pengukuran, seperti ditunjukkan lingkaran merah pada gambar: besarnya *ping latency* adalah sebesar 449.93 milliseconds.
- b. Rata-rata besarnya *ping latency*, seperti ditunjukkan lingkaran kuning pada gambar: adalah sebesar 326.94 milliseconds

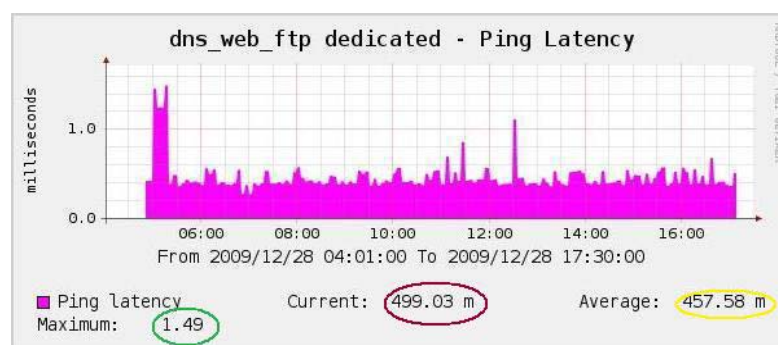
- c. Besarnya *ping latency* tertinggi, seperti ditunjukkan lingkaran hijau pada gambar: adalah sebesar 1.50 seconds



Gambar 4.15 Ping Latency DHCP-Proxy virtual server

Keterangan gambar 4.15 :

- Pada saat pengukuran, seperti ditunjukkan lingkaran merah pada gambar: besarnya *ping latency* adalah sebesar 249.36 milliseconds.
- Rata-rata besarnya *ping latency*, seperti ditunjukkan lingkaran kuning pada gambar: adalah sebesar 2.74 seconds
- Besarnya *ping latency* tertinggi, seperti ditunjukkan lingkaran hijau pada gambar: adalah sebesar 32.76 seconds

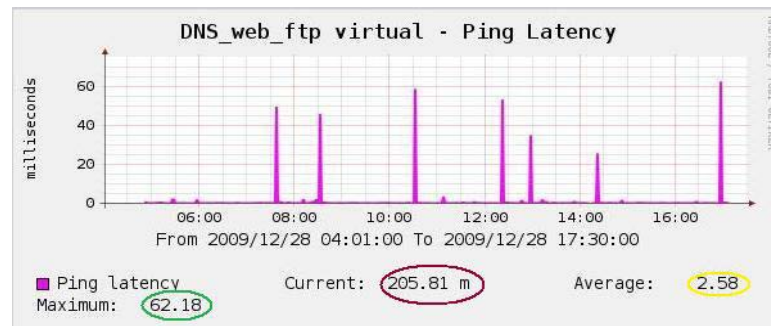


Gambar 4.16 Ping Latency DNS-Web-FTP dedicated server

Keterangan gambar 4.16 :

- Pada saat pengukuran, seperti ditunjukkan lingkaran merah pada gambar: besarnya *ping latency* adalah sebesar 449.03 milliseconds.

- b. Rata-rata besarnya *ping latency*, seperti ditunjukkan lingkaran kuning pada gambar: adalah sebesar 457.58 milliseconds
- c. Besarnya *ping latency* tertinggi, seperti ditunjukkan lingkaran hijau pada gambar: adalah sebesar 1.49 seconds



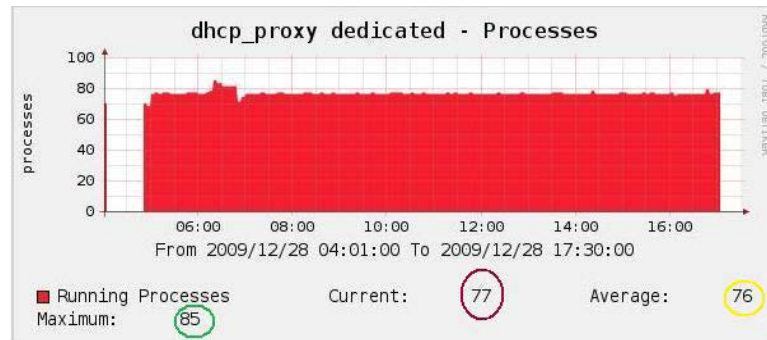
Gambar 4.17 Ping Latency DNS-Web-FTP virtual server

Keterangan gambar 4.17 :

- a. Pada saat pengukuran, seperti ditunjukkan lingkaran merah pada gambar: besarnya *ping latency* adalah sebesar 205.81 milliseconds.
- b. Rata-rata besarnya *ping latency*, seperti ditunjukkan lingkaran kuning pada gambar: adalah sebesar 2.58 seconds
- c. Besarnya *ping latency* tertinggi, seperti ditunjukkan lingkaran hijau pada gambar: adalah sebesar 62.18 seconds

Ping latency merupakan besarnya waktu tunda yang dilakukan terhadap *server*. Berbeda dengan *load average* yang mengakses layanan pada *server*, *ping latency* hanya melakukan *request* untuk mengecek apakah koneksi secara LAN tersambung atau tidak. Semakin besar nilai *ping latency*-nya, semakin kurang baik yang berarti *server* tersebut memiliki waktu tunda dan kemampuan menangkap *ping request* yang rendah sehingga membuat adanya antrian *request*.

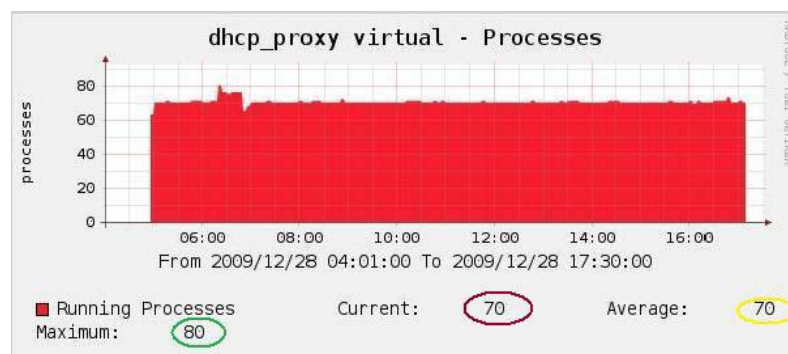
#### 4.3.4 Processes



Gambar 4.18 Processes DHCP-Proxy dedicated server

Keterangan gambar 4.18 :

- Pada saat pengukuran, seperti ditunjukkan lingkaran merah pada gambar: besarnya *processes* adalah sebesar 77 processes.
- Rata-rata banyaknya *processes*, seperti ditunjukkan lingkaran kuning pada gambar: adalah sebesar 76 processes.
- Banyaknya *processes* tertinggi, seperti ditunjukkan lingkaran hijau pada gambar: adalah sebesar 85 processes.

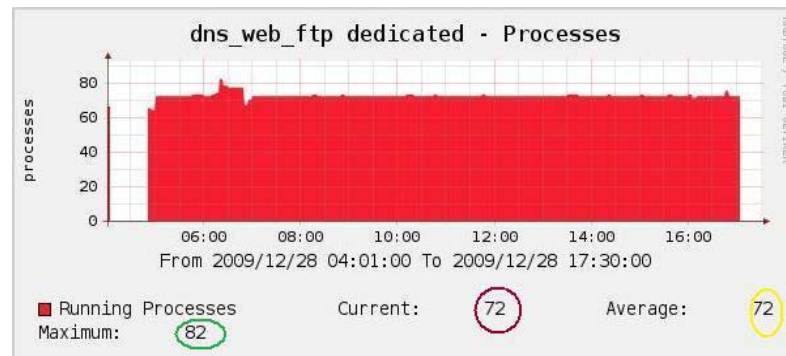


Gambar 4.19 Processes DHCP-Proxy virtual server

Keterangan gambar 4.19 :

- Pada saat pengukuran, seperti ditunjukkan lingkaran merah pada gambar: besarnya *processes* adalah sebesar 70 processes.

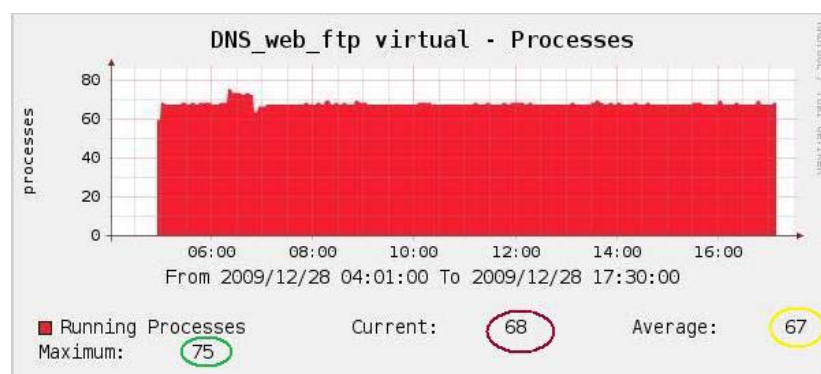
- b. Rata-rata banyaknya *processes*, seperti ditunjukkan lingkaran kuning pada gambar: adalah sebesar 70 *processes*.
- c. Banyaknya *processes* tertinggi, seperti ditunjukkan lingkaran hijau pada gambar: adalah sebesar 80 *processes*.



Gambar 4.20 Processes DNS-Web-FTP dedicated server

Keterangan gambar 4.20 :

- a. Pada saat pengukuran, seperti ditunjukkan lingkaran merah pada gambar: besarnya *processes* adalah sebesar 72 *processes*.
- b. Rata-rata banyaknya *processes*, seperti ditunjukkan lingkaran kuning pada gambar: adalah sebesar 72 *processes*.
- c. Banyaknya *processes* tertinggi, seperti ditunjukkan lingkaran hijau pada gambar: adalah sebesar 82 *processes*.



Gambar 4.21 Processes DNS-Web-FTP virtual server

Keterangan gambar 4.21 :

- a. Pada saat pengukuran, seperti ditunjukkan lingkaran merah pada gambar: besarnya *processes* adalah sebesar 68 processes.
- b. Rata-rata banyaknya *processes*, seperti ditunjukkan lingkaran kuning pada gambar: adalah sebesar 67 processes.
- c. Banyaknya *processes* tertinggi, seperti ditunjukkan lingkaran hijau pada gambar: adalah sebesar 75 processes.

Grafik *processes* ini menggambarkan akumulasi keseluruhan performa pada *server* yang dimonitoring, baik itu kinerja *hardware* maupun kinerja program yang dijalankan. Sebagai contoh, besarnya nilai terkait dengan *memory usage*, *load average*, dan *ping latency* ketika *server* dijalankan. Semakin besar nilainya, semakin kurang baik karena pengguna *resources* yang besar juga.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan hasil pengembangan, pengujian, dan analisis akhir dari perbandingan kinerja Virtual server dengan Dedicated server menggunakan Template Unix Cacti , maka dapat ditarik beberapa kesimpulan sebagai berikut :

1. Instalasi dan konfigurasi pada Virtual server tidak berbeda dengan Dedicated server, yang membedakan adalah spesifikasi perangkat keras yang dibutuhkan dari kedua *server* tersebut.
2. Berdasarkan hasil monitoring, besarnya *memory usage* pada Virtual server dan Dedicated server tidak jauh berbeda. Hal ini dikarenakan tidak banyaknya program yang dijalankan, sehingga perbedaan yang signifikan tidak kelihatan.
3. Berdasarkan hasil monitoring, besarnya *load average* pada Dedicated server lebih besar daripada *load average* pada Virtual server. Hal ini dikarenakan kemampuan Dedicated server yang mampu menangkap *request* dari pengguna lebih baik daripada Virtual server.
4. Berdasarkan hasil monitoring, besarnya rata-rata *ping latency* pada Dedicated server lebih sering daripada Virtual server. Hal ini berarti Virtual server lebih baik dalam menangkap *ping request*, dan hal ini dapat dikarenakan topologi *monitoring server* yang berada satu mesin dengan Virtual server.

5. Berdasarkan hasil monitoring, besarnya nilai *processes* pada Virtual server lebih rendah daripada nilai *processes* pada Dedicated server yang berarti program yang dijalankan dan konsumsi sumber daya pada Virtual server lebih sedikit, tetapi mempunyai utilitas yang sama besar.
6. Pada tingkat trafik yang sedikit yaitu di bawah tujuh pengguna, Virtual server dengan kebutuhan perangkat yang lebih rendah mempunyai performa yang sama baik dengan Dedicated server.

## 5.2 Saran

Saran yang dapat diberikan setelah kegiatan pengujian dan hasil analisis perbandingan *server* ini adalah :

1. Untuk Virtual server hendaknya menggunakan spesifikasi komputer yang mempunyai memori besar, hal ini dikarenakan kinerja *Host Operating System* yang harus membagi-bagi proses (*swapping*) kepada *Guest Operating System* yang ada.
2. Untuk analisis selanjutnya, hendaknya dilakukan pada organisasi atau instansi dengan tingkat trafik menengah ke atas dan monitoring secara lebih *real time*.

## 5.3 Rekomendasi

Berdasarkan hasil analisis akhir dari perbandingan Virtual server dengan Dedicated server, maka rekomendasi yang dapat diberikan sebagai bahan pertimbangan untuk menggunakan Virtual server adalah sebagai berikut :

1. Lebih hemat *server* dengan menggabungkan beberapa *server* dalam satu mesin *virtual* sesuai dengan utilitas *server* tersebut.

Jumlah *server* dapat ditekan sampai 50% lebih sedikit atau bahkan lebih dari 50%. Dampaknya panas di *datacenter* jadi berkurang, maka alat pendingin yang digunakan juga tidak sebanyak biasanya. Hal ini juga bisa dikatakan sebagai salah satu contoh *green bussines*, karena melakukan penghematan sumber daya dan ikut menjaga kelestarian Bumi. Selain itu ruangan di *datacenter* adalah sesuatu yang sangat mahal karena fasilitas yang ada di ruangan tersebut (alat pendingin, lantai khusus, alat keamanan, kabel khusus, dan lain-lain), sehingga dapat menghemat penggunaan tempat di *datacenter* dan hanya memerlukan daya listrik dalam jumlah yang jauh lebih sedikit.

2. Membuat *backup* untuk setiap Virtual server yang dibangun.

Menggabungkan beberapa jenis *server* dalam satu mesin berarti membuat berbagai layanan dalam satu tempat, dan ketika *server* tersebut melakukan *update* yang memerlukan *restart*, maka layanan *server* yang lain akan ikut *down*, oleh sebab itu diperlukannya *backup-an server* ini agar layanan *server* tetap dapat dijalankan. Selain itu apabila terjadi kerusakan *hardware*, *server* dapat dipindahkan ke virtualisasi *server* yang baru tanpa proses pemulihan secara khusus, hasilnya pengurangan *downtime* yang sangat besar.

3. Mengoptimalkan spesifikasi memori komputer dibanding dengan kebutuhan perangkat keras lainnya.

Cara kerja *virtual machine* yang melakukan *swapping* (membagi-bagi proses) menuntut sebuah komputer untuk memiliki memori yang sangat baik

dibanding dengan perangkat keras lainnya, seperti prosesor dan *harddisk* sebagai contoh. Pada saat melakukan *swapping*, kecepatan memori bisa seratus kali lebih cepat daripada kecepatan *harddisk* dan rata-rata penggunaan CPU di bawah 10%.

4. Menggunakan perangkat lunak (Virtual Machine) yang paling baik.

Untuk memilih perangkat lunak (Virtual Machine) yang paling baik, beberapa hal-hal yang perlu dipertimbangkan adalah sebagai berikut :

- a. Perangkat lunak bersifat gratis.
- b. Kemudahan untuk instalasi dan konfigurasi.
- c. Kemudahan untuk menambah atau mengubah periperalnya.
- d. Kemampuan perangkat lunak tersebut.
- e. Besar ruang yang dibutuhkan untuk menyimpan data-data minimal yang dibutuhkan agar perangkat lunak berfungsi dengan baik.
- f. Prospek perangkat lunak tersebut di masa yang akan datang.
- g. Performasi dan konsumsi sumber daya yang digunakan perangkat lunak tersebut.
- h. Fasilitas apa yang mampu didukung perangkat lunak tersebut.
- i. Dukungan teknis (mempunyai situs atau milis aktif).
- j. Dukungan *platform* (jenis sistem operasi apa saja yang bisa menjalankan perangkat lunak tersebut).
- k. Dukungan terhadap *third party* (apakah perangkat lunak ini dapat ditambahkan perangkat lunak tambahan sebagai pelengkap).

## DAFTAR PUSTAKA

- Tim Penyusun Lab Sisjarkom. 2006. *Modul Praktikum Sistem Operasi 2006*. Yogyakarta: Jurusan Teknik Informatika, Universitas Islam Indonesia.
- Tim Penyusun Lab Sisjarkom. 2007. *Modul Praktikum Jaringan Komputer Semester Genap 2006/2007*. Yogyakarta: Jurusan Teknik Informatika, Universitas Islam Indonesia.
- Anonim. 2008. *Pentingnya virtualisasi server dalam pengolahan data*. Tersedia di <http://teknologitinggi.wordpress.com/2008/10/24/pentingnya-virtualisasi-server-dalam-pengolahan-data/> (terakhir diakses pada tanggal 8 Oktober 2009).
- Leonardo. 2008. *Teknologi Virtualisasi*. Tersedia di <http://wss-id.org/blogs/leonardo/archive/2008/07/17/TeknologiVirtualisasi.aspx> (terakhir diakses pada tanggal 8 Oktober 2009).
- Mahayana, Dimitri. 2009. *Yuk, bangun data center ramah lingkungan*. Tersedia di <http://www.detikinet.com/read/2009/06/23/132849/1152586/328/yuk-bangun-data-center-ramah-lingkungan> (terakhir diakses pada tanggal 10 Juli 2009).
- Malik, Yaj. 2008. *Surganya pusat pengolahan data*. Tersedia di <http://www.antara.co.id/view/?i=1224815189&c=PRW&s=> (terakhir diakses pada tanggal 8 Oktober 2009).

Ramansyah, Doni. 2008. *Virtualization, tren IT saat ini*. Tersedia di <http://www.trendigital.com/Website/News.aspx?id=321> (terakhir diakses pada tanggal 8 Oktober 2009).

Ramansyah, Doni. 2009. *Virtualization di Linux*. Tersedia di <http://www.trendigital.com/Website/News.aspx?id=653> (terakhir diakses pada tanggal 8 Oktober 2009).

Wahyuni, Tri. 2008. *Virtualisasi, bukan tren tapi kebutuhan*. Tersedia di [http://irsyadi.com/itnews\\_detail.php?recordID=202](http://irsyadi.com/itnews_detail.php?recordID=202) (terakhir diakses pada tanggal 8 Oktober 2009).