

## BAB 3 PERENCANAAN DAN PERANCANGAN

### PENGUJIAN

#### 3.1 Identifikasi Lingkungan Pengujian

Identifikasi lingkungan pengujian meliputi pencarian informasi mengenai perangkat keras, perangkat lunak, jaringan, dan *tools* lainnya yang dilibatkan dalam pengujian.

##### 3.1.1 Perangkat Keras

Perangkat keras yang digunakan dalam pengujian pada tesis ini terbagi menjadi tiga: dua *server* dan satu *client*. Berikut adalah spesifikasi dari komputer *server*:

<b>Merk</b>	HP ProLiant ML350 G5
<b>CPU</b>	1 x Intel Xeon E5130 (Dual-Core, 2.0 GHz) 1 x Intel Xeon E5140 (Dual-Core, 2.33 GHz)
<b>RAM</b>	4 GB
<b>Disk</b>	2 x 146 GB (SAS, 15K RPM, RAID-1)
<b>NIC</b>	2 x Gigabit Ethernet

*Tabel 1: Spesifikasi komputer server.*

Dalam spesifikasi komputer *server* di atas keduanya memiliki dua *core* yang berarti memiliki dua CPU dalam satu *packaging*. Pada pengujian yang dilakukan di dalam tesis ini, sistem operasi dan perangkat lunak virtualisasi dijalankan tanpa fitur *multiprocessing* (sering disebut sebagai *symetric multiprocessing*, SMP)

karena dua alasan:

- Memudahkan dalam membandingkan masing-masing solusi virtualisasi dengan pada saat *native* tanpa memperhitungkan seberapa baik masing-masing perangkat lunak virtualisasi atau sistem operasi tersebut memanfaatkan fitur SMP.
- Solusi virtualisasi seperti Virtual Server tidak dapat memanfaatkan SMP pada mesin virtual.

Begitu juga dengan RAM, meskipun tersedia 4 GB masing-masing komputer dibatasi menjadi 1 GB. Ini artinya pada saat pengujian dengan komputer *native* sistem operasi diberikan 1 GB, sedangkan pada saat menggunakan virtualisasi masing-masing mesin virtual juga diberikan jumlah RAM yang sama, 1 GB.

	Tanpa virtualisasi ( <i>native</i> , 1 komputer)	Tanpa virtualisasi ( <i>native</i> , 2 komputer)	Dengan virtualisasi	
			Sistem operasi <i>host</i>	Sistem operasi <i>guest</i>
<b>CPU</b>	1	@1	1	1
<b>RAM</b>	2 GB	@1 GB	~	1 GB
<b>Disk</b>	~	~	~	1 GB

~ Sebesar sisa yang tersedia.

Tabel 2: Pembatasan spesifikasi komputer.

Selain komputer *server* pengujian ini juga melibatkan komputer yang berperan sebagai *client*. Komputer ini digunakan untuk melakukan *request* kepada aplikasi Web pada *Web server* (mengenai hal ini akan dibahas pada sub-bab 3.3). Spesifikasi tidak menjadi masalah selama komputer tersebut bisa membuat sejumlah *request* dan menerima balasannya sesuai yang dibutuhkan dalam pengujian. Spesifikasi komputer yang digunakan adalah:

<b>Merk</b>	IBM/Lenovo ThinkPad X60
<b>CPU</b>	1 x Intel Core Duo T2300 (Dual-Core, 1.66 GHz)
<b>RAM</b>	1 GB
<b>Disk</b>	60 GB
<b>NIC</b>	1 x Gigabit Ethernet

Tabel 3: Spesifikasi komputer client.

### 3.1.2 Perangkat Lunak

Perangkat lunak yang digunakan dalam pengujian ini terbagi empat, yaitu sistem operasi, aplikasi/program yang diujikan, aplikasi/program penguji, dan perangkat lunak virtualisasi itu sendiri.

#### 3.1.2.1 Solusi Virtualisasi

Berikut adalah daftar perangkat lunak virtualisasi yang digunakan di dalam pengujian:

- **Microsoft Hyper-V.** Hyper-V adalah solusi virtualisasi dari Microsoft yang tersedia bersama dengan sistem operasi Windows Server 2008. Microsoft juga memberikan secara cuma-cuma Microsoft Hyper-V Server 2008 yang sebagian orang menyebutnya sebagai varian dari Windows Server 2008 “Core” dengan Hyper-V sebagai satu-satunya *role* yang tersedia.

Hyper-V merupakan perangkat lunak virtualisasi bertipe *bare-metal* yang memerlukan CPU *x86-64* dan teknologi Intel VT-x atau AMD-V (hanya mendukung pendekatan *hardware-assisted virtualization*). Hyper-V mendukung sistem operasi *desktop/server* sejak Windows 2000 hingga

Windows server 2008 dan beberapa distribusi GNU/Linux.

- **Linux-VServer.** Linux-VServer adalah salah satu solusi virtualisasi yang memberikan kemampuan *OS-level virtualization* pada *kernel* Linux. Linux-VServer didistribusikan sebagai perangkat lunak bebas dan dikembangkan oleh komunitas. Linux-VServer dapat berjalan pada kebanyakan arsitektur CPU yang didukung oleh *kernel* Linux, terutama *x86* dan *x86-64*.
- **OpenVZ.** OpenVZ, seperti Linux-VServer, adalah salah satu solusi virtualisasi yang memberikan kemampuan *OS-level virtualization* pada *kernel* Linux. OpenVZ didistribusikan sebagai perangkat lunak bebas dan dikembangkan oleh komunitas. Pengembangan OpenVZ juga didukung oleh perusahaan komersil, Parallels, dan menjadi basis dari salah satu produk mereka, Parallels Virtuozzo.
- **Sun xVM VirtualBox.** VirtualBox adalah perangkat lunak virtualisasi dari Sun Microsystem dengan tipe *hosted*. VirtualBox merupakan perangkat lunak bebas, tetapi versi *proprietary* (gratis) dari Sun memberikan beberapa fitur tambahan seperti Remote Desktop Protocol, USB, iSCSI, dll.

VirtualBox dapat berjalan pada CPU dengan arsitektur *x86* atau *x86-64* dan sistem operasi Windows, GNU/Linux, Mac OS X, atau Solaris sebagai *host*. VirtualBox dapat menggunakan pendekatan *full virtualization* maupun *hardware-assisted virtualization*, sedangkan *paravirtualization*

direncanakan di masa mendatang.

- **Microsoft Virtual Server.** Sebelum Hyper-V, Virtual Server adalah solusi virtualisasi untuk *server* satu-satunya dari Microsoft. Virtual Server merupakan perangkat lunak virtualisasi bertipe *hosted* dan bisa melakukan virtualisasi dengan pendekatan *full virtualization* atau *hardware-assisted virtualization*. Saat ini Virtual Server sudah bisa didapat secara gratis.

VirtualServer dapat berjalan pada CPU dengan arsitektur *x86* atau *x86-64*, tetapi hanya bisa menjalankan sistem operasi atau mesin virtual *x86*. Virtual Server mendukung sistem operasi sejak Windows XP hingga Windows Server 2008 sebagai *host*, sedangkan untuk *guest* Virtual Server mendukung sistem operasi *server* sejak Windows NT Server 4.0 hingga Windows Server 2003. Meskipun begitu Microsoft tidak menyarankan Windows XP atau Windows Vista sebagai sistem operasi *host* pada *server* produksi.

- **VMware ESX/ESXi.** ESX/ESXi adalah salah satu perangkat lunak virtualisasi dari VMware yang bertipe *bare-metal*. Apa yang membedakan ESX dengan ESXi adalah arsitektur dan manajemen operasi. Walaupun inti dari kedua perangkat lunak virtualisasi sama dan tidak bergantung pada sistem operasi tertentu untuk manajemen, tetapi ESX memerlukan sistem operasi GNU/Linux untuk melakukan manajemen. ESXi bisa dibidang sebagai ESX versi minimal tanpa fitur-fitur tambahan tertentu yang bisa didapat dengan gratis. Dalam tesis ini yang digunakan adalah

ESXi.

ESX/ESXi berjalan pada arsitektur CPU *x86* atau *x86-64*. ESX/ESXi dapat melakukan pendekatan virtualisasi *full virtualization*, *paravirtualization*, dan *hardware-assisted virtualization* (hanya pada sistem operasi 64-bit).

- **VMware Server.** VMware Server sebelumnya bernama VMware GSX Server dan merupakan produk utama dari VMware. VMware memberikan VMware Server secara cuma-cuma dengan harapan menjadi titik mula pengguna menuju VMware ESX.

VMware Server bertipe *hosted* dan mendukung CPU dengan arsitektur *x86* atau *x86-64* dengan sistem sistem operasi GNU/Linux dan Windows sebagai *host*. VMware Server dapat melakukan pendekatan virtualisasi *full virtualization*, *paravirtualization*, dan *hardware-assisted virtualization*.

- **Xen.** Xen adalah perangkat lunak virtualisasi yang awalnya dikembangkan di universitas Cambridge dan sekarang dikembangkan oleh komunitas sebagai perangkat lunak bebas. Selain itu dalam pengembangannya Xen juga didukung oleh banyak perusahaan TI terkemuka di dunia seperti Citrix, IBM, Intel, Hewlett-Packard, Novell, Red Hat, Sun Microsystems, dan Oracle.

Xen dapat berjalan pada arsitektur CPU *x86/x86-64* dan menjalankan sistem operasi di dalam mesin virtual dengan arsitektur yang sama. Pendekatan utama Xen adalah *paravirtualization*, tetapi sejak versi 3.0 Xen juga mendukung *hardware-assisted virtualization*. Xen merupakan

perangkat lunak virtualisasi dengan tipe *bare-metal*.

Nama	Pembuat	Host CPU	Guest CPU	Host OS	Guest OS	Pendekatan	Lisensi
Hyper-V	Microsoft	x86-64	x86, x86-64	None (bare-metal, Windows Server 2008 as <i>root partition</i> )	Windows 2000-Vista, Linux	Hardware assisted	Proprietary (free)
Linux-VServer	Community	x86, x86-64, etc.	Same as host	Linux	Same as host	OS-level	GPL v2
OpenVZ	Community & Parallels	x86, x86-64, etc.	Same as host	Linux	Same as host	OS-level	GPL v2
Sun xVM VirtualBox	Community & Sun Microsystems	x86, x86-64	x86, x86-64 (with hardware assisted)	Windows, Linux, Mac OS X, Solaris	Windows, Linux, *BSD, Solaris, etc	Full-virtualization & Hardware assisted	GPL v2, Proprietary (free)
Virtual Server	Microsoft	x86, x86-64	x86	Windows XP-Windows Server 2008	Windows NT-2003, Linux	Full-virtualization & Hardware assisted	Proprietary (free)
VMware ESX/ESXi Server	VMware	x86, x86-64	Same as host	None (bare-metal)	Windows, Linux, *BSD, Solaris, etc	Full-virtualization, Hardware assisted, and Paravirtualization	Proprietary /Proprietary (free)
VMware Server	VMware	x86, x86-64	Same as host	Windows, Linux	Windows, Linux, *BSD, Solaris, etc	Full-virtualization, Hardware assisted, and Paravirtualization	Proprietary (free)
Xen	Community & Citrix Systems	x86, x86-64, etc.	Same as host	None (bare-metal, Linux, Solaris, etc as <i>domain 0</i> )	Windows (with hardware assisted), Linux, *BSD, Solaris, etc	Paravirtualization & Hardware assisted	GPL

Tabel 4: Daftar solusi virtualisasi yang diujikan.

### 3.1.2.2 Sistem Operasi

Ada dua sistem operasi yang terlibat dalam pengujian ini, pertama GNU/Linux dan Microsoft Windows Server 2008. Distribusi yang digunakan untuk GNU/Linux itu sendiri adalah Debian 5.0 untuk *server* dan Ubuntu 8.10 untuk *client*.

Untuk solusi virtualisasi yang berasal dari Microsoft digunakan Windows

Server 2008 (x86-64) sebagai sistem operasi *host*. Untuk solusi virtualisasi lainnya, meskipun tersedia untuk sistem GNU/Linux dan Windows, sistem GNU/Linux (x86-64) yang digunakan sebagai sistem operasi *host*. Untuk sistem operasi *guest* semuanya disamakan menggunakan sistem GNU/Linux (x86).

Nama	Sistem operasi <i>host</i>	Sistem operasi <i>guest</i>
Native	Debian GNU/Linux 5.0 (x86)	None (no virtualization)
Hyper-V	None (bare-metal, Windows Server 2008 as root partititon, x86-64 )	Debian GNU/Linux 5.0 (x86)
Linux-VServer	Debian GNU/Linux 5.0 (x86-64)	Debian GNU/Linux 5.0 (x86)
OpenVZ	Debian GNU/Linux 5.0 (x86-64)	Debian GNU/Linux 5.0 (x86)
Sun xVM VirtualBox	Debian GNU/Linux 5.0 (x86-64)	Debian GNU/Linux 5.0 (x86)
Virtual Server R2	Windows Server 2008 (x86-64)	Debian GNU/Linux 5.0 (x86)
VMWare ESXi	None	Debian GNU/Linux 5.0 (x86)
VMWare Server	Debian GNU/Linux 5.0 (x86-64)	Debian GNU/Linux 5.0 (x86)
Xen	None (bare-metal, Debian 5.0 as <i>domain 0</i> , x86-64)	Debian GNU/Linux 5.0 (x86)

Tabel 5: Sistem operasi yang digunakan.

Beberapa perangkat lunak virtualisasi tertentu yang bertipe *bare-metal* memerlukan suatu mesin virtual khusus dengan sistem operasi tertentu yang digunakan untuk mengendalikan perangkat lunak virtualisasi yang bersangkutan. Dalam pengujian ini, untuk Xen, mesin virtual khusus (dalam istilah Xen disebut sebagai *domain 0*) itu sendiri terpasang Debian 5.0 (x85-64). Untuk Microsoft Hyper-V sistem operasi yang terpasang pada mesin virtual khusus (*Root Partition* dalam istilah Hyper-V) tersebut sudah pasti untuk saat ini adalah Windows Server 2008. Untuk VMware ESX/ESXi sendiri, karena bisa dikendalikan melalui jaringan menggunakan aplikasi *thick client* maka tidak diperlukan mesin virtual

tambahan.

### 3.1.2.3 Aplikasi/Program

Pada skenario pengujian yang akan dibahas pada sub-bab 3.3, ada tiga skenario yang membutuhkan program/aplikasi tertentu agar skenario tersebut bisa dijalankan. Meskipun begitu, untuk memenuhi kebutuhan masing-masing skenario tersebut tidak hanya sebatas pada tiga program. Masing-masing program yang digunakan membutuhkan program lainnya agar bisa digunakan sebagai alat penguji, tetapi yang menjadi tugas utama (*main task*) dari masing-masing skenario pengujian adalah:

- **WordPress.** WordPress adalah suatu aplikasi Web untuk penerbitan (*publishing*) yang populer digunakan sebagai aplikasi untuk *blogging*. WordPress adalah aplikasi bebas dan *open source*. WordPress sebagian besar kodenya dibuat menggunakan bahasa pemrograman PHP dan MySQL sebagai basis datanya. Ini berarti WordPress membutuhkan sebuah *Web server* dengan interpreter PHP dan *database server* MySQL terpasang agar bisa digunakan.
- **bzip2.** bzip2 adalah algoritma/program untuk melakukan kompresi data dengan menggunakan algoritma Burrows-Wheeler dan Huffman. bzip2 merupakan algoritma dan program yang berlisensi bebas.
- **gcc.** gcc, kependekan dari GNU Compiler Collection, adalah sistem *compiler* yang dibuat oleh GNU Project yang mendukung berbagai macam bahasa pemrograman. Pada kebanyakan sistem operasi *UNIX-like*,

Universitas Indonesia

khususnya GNU/Linux, gcc sering digunakan sebagai *compiler* standar.

### 3.1.2.4 Alat Pengujian

Alat pengujian terbagi dua, alat untuk melakukan simulasi beban dan alat untuk mendapatkan data dari metrik. Alat untuk melakukan simulasi beban hanya diperlukan untuk skenario Aplikasi Web, sedangkan alat untuk mendapatkan data metrik diperlukan pada semua skenario.

- **httperf.** `httperf` adalah program untuk mengukur kinerja dari *Web server* yang dibuat oleh David Mosberger dari HP Labs. `httperf` menyediakan fitur yang fleksibel dalam pembuatan beban kerja pada *Web/HTTP server*. `httperf` dapat mensimulasikan berbagai macam beban kerja sesuai dengan parameter yang diberikan kepadanya.

Program ini diperlukan untuk mensimulasikan beban kerja pada skenario Aplikasi Web. Versi yang digunakan adalah sama seperti yang tersedia pada repositori Ubuntu 8.10.

- **sar.** `sar` yang merupakan bagian dari paket program `sysstat` digunakan untuk mengambil informasi mengenai aktivitas sistem. `sar` dapat digunakan untuk mendapatkan kecepatan transfer perangkat *I/O* seperti jaringan atau *disk*, tingkat penggunaan RAM atau CPU, dan sebagainya.

Untuk solusi virtualisasi yang bukan berbentuk *bare-metal* dan berjalan pada sistem operasi GNU/Linux program ini dapat digunakan untuk melihat tingkat penggunaan sumber daya CPU. Selain itu program ini juga

digunakan pada saat *native*. Versi yang digunakan adalah sama seperti yang tersedia pada repositori Debian 5.0

Pengambilan metrik tingkat penggunaan sumber daya CPU menggunakan program ini dilakukan dengan cara mengurangi 100% (mewakili seluruh sumber daya CPU yang tersedia) dikurangi oleh hasil rata-rata pada kolom %idle (persentase CPU dalam keadaan *idle*).

- **perfmon.exe.** `perfmon.exe` adalah program untuk sistem operasi Windows yang berfungsi kurang-lebih sama seperti `sar`. Untuk solusi virtualisasi berjenis *hosted* pada sistem operasi Windows program ini bisa digunakan langsung apa adanya, karena *by default* tingkat penggunaan sumber daya CPU ditampilkan oleh program ini.
- **xentop & esxtop.** `xentop` dan `esxtop` adalah program yang serupa dengan program `top` yang tersedia pada sistem operasi UNIX, yaitu untuk menampilkan informasi sistem secara ringkas. `xentop` dan `esxtop` digunakan untuk melihat tingkat penggunaan sumber daya CPU pada Xen dan VMware ESX/ESXi karena keduanya merupakan solusi virtualisasi bertipe *bare-metal*.
- **time.** Program `time` digunakan untuk menjalankan program lainnya. Setelah selesai program ini akan menampilkan informasi waktu mengenai program yang berjalan tersebut.

Apabila suatu proses dijalankan dengan menggunakan program `time`, maka setelah proses tersebut selesai program `time` akan melaporkan

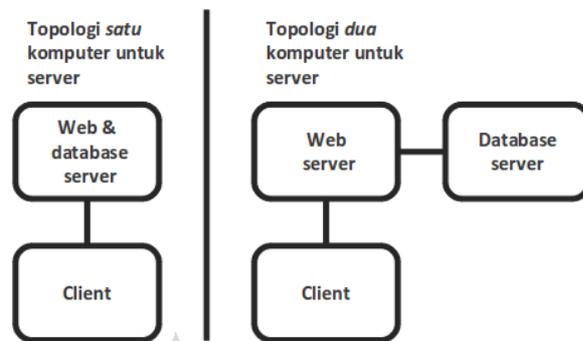
berapa lama proses tersebut berjalan (*real elapsed time*). Selain itu *by default* program `time` juga akan menampilkan berapa lama CPU bekerja pada *system/user mode* selama proses yang bersangkutan berjalan.

### 3.1.3 Jaringan

Dalam pengujian ada dua bagian utama, pengujian tanpa virtualisasi (*native*) dan dengan virtualisasi. Pada pengujian tanpa virtualisasi ada dua macam topologi jaringan. Topologi jaringan yang dibahas di sini hanya digunakan pada pengujian yang melibatkan aplikasi Web, karena skenario pengujian lainnya tidak melibatkan jaringan.

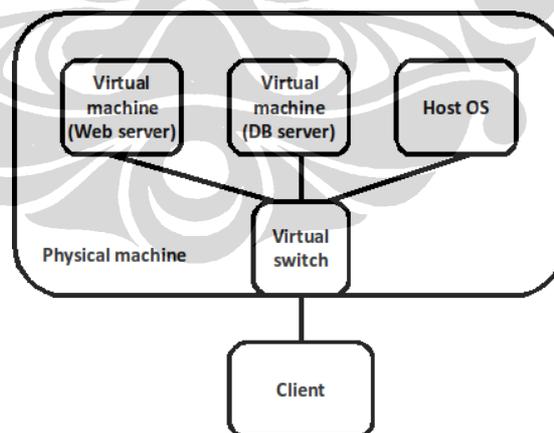
Pengujian yang melibatkan aplikasi Web membutuhkan dua buah *server* (program *server*, bukan komputer *server*). Masing-masing *server* bisa dipasang pada satu komputer atau pada dua komputer yang berbeda. Untuk konfigurasi satu komputer maka topologi jaringan hanya berupa *point-to-point* antara dua komputer, komputer *server* dan komputer *client*.

Pada konfigurasi dua komputer, *Web server* dan *database server* masing-masing terpasang pada dua komputer dan saling terhubung secara *point-to-point*. Komputer *client* juga terhubung langsung secara *point-to-point* dengan komputer Web server. Ini artinya komputer *Web server* minimal harus memiliki dua buah *NIC* (*network interface controller*), untuk *database server* dan untuk *client*.



Gambar 9: Topologi jaringan tanpa virtualisasi.

Untuk pengujian pada saat menggunakan virtualisasi, pada dasarnya topologi yang digunakan berupa satu atau lebih mesin virtual yang terhubung pada satu *bridge/switch*. Walaupun mesin virtual terhubung dengan salah satu *port* dari *switch* virtual, tetapi karena *port* fisik komputer digunakan sebagai salah satu *port* dari *switch* virtual tersebut, maka ini digunakan sebagai jalan menuju “dunia luar”. *Port* fisik inilah yang terhubung dengan komputer *client*.



Gambar 10: Topologi jaringan dengan virtualisasi.

Pada skenario Kompresi dan Kompilasi (dijelaskan pada sub-bab 3.3) hanya satu mesin virtual yang terlibat, sehingga hanya satu mesin virtual yang dibuat

dan berjalan. Jika dibandingkan dengan Gambar 10, pada skenario kedua dan ketiga hanya ada satu gambar mesin virtual.

### 3.2 Kriteria Penerimaan Kinerja

Meier dkk. (2007) mengatakan bahwa umumnya ada tiga hal yang menjadi sumber perhatian dalam suatu uji kinerja:

- **Response time.** *Response time* mengacu pada seberapa cepat suatu sistem/aplikasi dapat merespon suatu aksi. Misalnya, halaman Web harus bisa tampil kurang dari lima detik.
- **Throughput.** *Throughput* menunjukkan banyaknya tugas yang bisa dilakukan dalam satu satuan waktu. Misalnya, *Web server* harus bisa melayani minimal 10 pengunjung dalam satu detik.
- **Resource utilization.** *Resource utilization* menunjukkan tingkat penggunaan sumber daya perangkat keras tertentu. Misalnya, pada saat jam sibuk *Web server* tidak boleh memakan CPU *time* lebih dari 20%.

Pada pengujian di sini tujuan utamanya adalah mencari tahu karakteristik kinerja dari beberapa solusi virtualisasi dengan pendekatannya yang berbeda-beda, oleh karena itu tidak ada target kinerja tertentu. Untuk ketiga skenario yang akan dijelaskan pada sub-bab 3.3 sendiri ada dua hal yang menjadi perhatian, yaitu *resource utilization* dan *response time*.

### 3.3 Perencanaan dan Perancangan dari Pengujian

Perencanaan dan perancangan uji kinerja menentukan skenario yang

diinginkan, menentukan variasi yang digunakan, menentukan data untuk pengujian, dan menentukan metrik yang diambil. Karena dalam pengujian pada tesis ini variasinya terdapat pada (kinerja) beberapa solusi virtualisasi, terutama pendekatannya, maka hanya sisanya akan dibahas disini.

### 3.3.1 Skenario

Pengujian yang dilakukan di sini terbagi menjadi tiga skenario, dan melayani halaman Web dinamis, kompresi dan kompilasi. Ketiga skenario tersebut dijalankan pada saat *native* dan pada masing-masing solusi virtualisasi. Ini termasuk setiap pendekatan yang disediakan oleh solusi tersebut, karena seperti yang ditunjukkan Tabel 4, solusi tertentu menyediakan lebih dari satu pendekatan virtualisasi.

#### 3.3.1.1 Aplikasi Web

Skenario pertama ini dipilih karena salah satu peran yang sering ditemukan pada komputer *server* adalah melayani aplikasi Web. Aplikasi yang digunakan dalam skenario ini, seperti yang telah disebutkan sebelumnya, adalah Wordpress. Wordpress cukup menggambarkan struktur atau bentuk dari banyak aplikasi Web seperti situs berita dan forum diskusi, di mana akan ada daftar dari judul *entry*, konten dari *entry* dan komentar terhadap *entry* tersebut. Versi Wordpress yang digunakan dalam pengujian ini adalah versi 2.7.1, sedangkan versi *Web server* (Apache), *database server* (MySQL), dan PHP *interpreter* sama dengan yang tersedia pada repositori Debian 5.0 (versinya akan selalu sama).

Pada saat *native* skenario ini melibatkan maksimal dua komputer, untuk *Web server* dan *database server*, yang artinya pada saat menggunakan virtualisasi juga melibatkan dua mesin virtual. Pengujian skenario ini saat *native* menggunakan satu komputer sekaligus untuk *Web server* dan *database server* juga dilakukan di sini.

Wordpress diisi dengan dua belas *entry*, dan masing-masing *entry* diisi dengan beberapa komentar. Seluruh daftar *entry* pada halaman utama dibagi menjadi dua halaman, artinya enam *entry* setiap halaman. Data *dummy* yang digunakan untuk mengisi *entry* dan komentar dapat dilihat pada Lampiran C.

Simulasi kasus yang digunakan dalam skenario ini berupa seorang pengunjung yang membuka *entry* terbaru dan seorang lagi melihat daftar *entry* pada halaman dua. Keduanya melakukan *request* secara bersamaan dalam satu detik. Selama simulasi dilakukan, akan dilihat tingkat penggunaan sumber daya CPU yang digunakan untuk melayani dua *request* per detik tersebut. Tingkat penggunaan sumber daya CPU yang merupakan data metrik dari skenario ini diambil minimal lima kali lalu diambil rata-ratanya. Karena beberapa perangkat lunak virtualisasi berbentuk *bare-metal*, maka cara untuk mendapatkan data metrik dalam skenario ini memerlukan cara berbeda-beda dan akan dibahas pada butir 3.3.2.

Tidak ada alasan khusus dalam mensimulasikan dua pengunjung dalam satu detik selain menyisakan ruang sumber daya tidak terpakai yang cukup pada saat menggunakan virtualisasi nanti. Ini dikarenakan dalam pengujian awal saat *native*

konfigurasi ini pada Web server memakan sekitar sepertiga dari seluruh sumber daya CPU. Mengingat saat menggunakan virtualisasi akan ada *overhead* dari proses virtualisasi itu sendiri, maka konfigurasi ini memberikan cukup ruang dan juga memberikan hasil data yang cukup *visible*.

### 3.3.1.2 Kompresi

Pada skenario kedua ini diujikan sebuah program kompresi `bzip2` dalam melakukan kompresi berkas-berkas. Apa yang menjadi data dan dikompresi dalam skenario ini adalah kode sumber *kernel* sistem operasi Linux yang besarnya mencapai 317 MB (hasil perhitungan program `du -h`). Versi kernel yang digunakan adalah sama dengan yang ada pada repositori Debian 5.0, versi 2.6.26 (versinya akan selalu sama).

Kode sumber *kernel* Linux diunduh dari repositori lalu setelah terpasang berkas arsip berada pada `/usr/src/linux-source-2.6.26.tar.bz2`. Berkas arsip ini diekstrak dan kemudian seluruh berkas di dalamnya kembali dikompres menggunakan program `bzip2` setelah sebelumnya diarsipkan kedalam satu berkas menggunakan program `tar`. Pengarsipan ini dilakukan karena program `bzip2` hanya bisa melakukan kompresi pada berkas tunggal. Lama dari proses kompresi adalah yang menjadi metrik dalam skenario ini, dan akan diambil minimal sebanyak tiga kali dan dihitung rata-ratanya.

### 3.3.1.3 Kompilasi

Dalam skenario ketiga ini kode sumber dari *kernel* Linux yang diekstrak

pada skenario Kompresi akan dikompilasi. Ini artinya kode sumber *kernel* Linux menjadi data dalam skenario ini. Lama dari proses kompilasi adalah yang menjadi metrik dalam skenario ini, dan akan diambil minimal sebanyak tiga kali dan dihitung rata-ratanya.

Kompilasi *kernel* Linux dilakukan menggunakan parameter “*allnoconfig*”. Parameter ini digunakan untuk menghindari terkompilasinya modul-modul dari *kernel* yang sifatnya *optional*.

### 3.4 Konfigurasi Lingkungan Pengujian

Dalam melakukan pengujian ini ada beberapa hal yang perlu diperhatikan pada perangkat keras, perangkat lunak, jaringan dan alat pengujian. Ini dilakukan karena semua itu tidak bisa digunakan begitu saja tanpa memiliki konfigurasi sesuai dengan apa yang direncanakan dan dirancang.

#### 3.4.1 Perangkat Keras

Beberapa hal mengenai perangkat keras yang perlu menjadi perhatian dalam pengujian ini adalah:

- Komputer yang digunakan untuk *Web server* pada konfigurasi dua komputer (*native*) adalah komputer dengan CPU Intel Xeon E5140 (2.33 GHz). Komputer yang sama juga digunakan pada konfigurasi satu komputer, termasuk saat menggunakan virtualisasi. Ini artinya komputer *server* dengan CPU Intel Xeon E5130 (2.0 GHz) hanya digunakan sekali saja, yaitu pada saat pengujian *native* dua komputer sebagai *database*

*server*.

- Pada saat *native* maupun sebagai sistem operasi *host* (atau *domain 0* untuk Xen), *kernel* Linux/Xen diberi parameter `nosmp maxcpus=1` sehingga hanya satu inti CPU yang digunakan. Begitu juga pada saat menggunakan Windows sebagai *host* (atau *Root Partition* untuk Hyper-V), parameter `/onecpu /numproc=1` digunakan untuk mendapatkan hasil yang sama. Untuk melakukan hal yang sama pada VMware ESX/ESXi, bisa dilakukan secara manual melalui program *thick client* agar ia hanya menggunakan salah satu *core* CPU saja.
- Khusus pada saat *native* dengan konfigurasi satu komputer RAM yang digunakan dibatasi sebesar 2 GB, sehingga *kernel* Linux diberi parameter `mem=2G`. Pada konfigurasi dua komputer masing-masing diberi parameter `mem=1G` agar setiap komputer hanya menggunakan RAM sebesar 1 GB.

Pada saat sebagai sistem operasi *guest*, *kernel* Linux tidak perlu diberikan parameter tertentu, karena semuanya diatur secara manual melalui perangkat lunak virtualisasi sesuai dengan konfigurasi yang telah disebutkan pada Tabel 2.

### 3.4.2 Perangkat Lunak

Untuk solusi virtualisasi, format dari *disk image* (sebagai *virtual disk*) yang digunakan adalah format *default* dari masing-masing solusi. Solusi virtualisasi dari VMware menggunakan format VMDK, dari Microsoft menggunakan format VHD, VirtualBox menggunakan format VDI, dan Xen menggunakan format *raw*. Pada solusi dengan pendekatan *operating system-level virtualization* tidak diperlukan *disk image*, karena seluruh berkas sistem operasi *guest* akan disimpan pada subdirektori *default* yang khusus disediakan untuk sistem operasi *guest* masing-masing solusi.

Untuk sistem operasi, seluruh instalasi Debian 5.0 dilakukan melalui *CD image* “netinst” pada partisi yang menggunakan sistem berkas *ext2*, sehingga sistem yang terpasang adalah sistem yang sangat minimal. Hal lain yang perlu

diketahui dari instalasi Debian 5.0 ini adalah bahwa beberapa *daemon*<sup>2</sup>, yaitu *acpid*, *cron*, *dhcpcd*, dan *rsyslog* dimatikan. Selain tidak diperlukan dalam pengujian, ini dilakukan untuk mengurangi kemungkinan program tersebut mempengaruhi hasil pengujian. Pada instalasi Debian 5.0 juga tidak dialokasikan berkas/partisi untuk *swap*, sedangkan pada instalasi sistem operasi Windows tidak ada konfigurasi khusus setelah instalasi selesai. Ini artinya alokasi berkas untuk *swap* atau *virtual memory* diserahkan seluruhnya pada sistem operasi Windows Server 2008.

Instalasi paket-paket perangkat lunak baik saat *native* maupun saat dengan virtualisasi (untuk *guest*) semuanya dilakukan dengan mengetikkan perintah:

- `apt-get install wordpress`, yang memasang Wordpress termasuk *dependency*-nya seperti *Web server* Apache beserta *interpreter* PHP. Aplikasi Wordpress itu sendiri di-*upgrade* sehingga menggunakan versi yang sama seperti yang telah disebutkan pada bagian 3.3.1.1, yaitu versi 2.7.1.
- `apt-get --no-install-recommends install mysql-server`, yang memasang *database server* MySQL. Parameter "`--no-install-recommends`" diberikan untuk menghindari terpasangnya *mail server* Exim yang disarankan untuk dipasang bersamaan dengan MySQL tetapi tidak diperlukan dalam pengujian ini.

Baris `skip-name-resolve` ditambahkan pada berkas konfigurasi MySQL (`/etc/my.cnf`) karena pengujian ini tidak menggunakan DNS. Selain itu *option* `bind-address` dimatikan agar MySQL bisa diakses dari komputer lain.

- `apt-get install linux-source-2.6.26 build-essential`, instalasi kode sumber dari *kernel* Linux termasuk kompiler GCC ini dilakukan untuk memenuhi skenario pengujian kedua dan ketiga, kompresi dan kompilasi.

---

<sup>2</sup> Program yang berjalan di belakang layar tanpa kendali langsung dari pengguna. Pada sistem operasi Windows program serupa disebut sebagai *service*.

### 3.4.3 Jaringan

Selain konfigurasi untuk pengujian *native* dua komputer, konfigurasi atau topologi jaringan pada pengujian ini cukup jelas, seluruh komputer berada pada satu jaringan yang sama, sehingga alamat IP apapun yang digunakan tidak menjadi masalah. Pada pengujian di sini digunakan konfigurasi seperti yang terlampir dalam tabel berikut:

	Native (1 PC)	Native (2 PC)	Dengan Virtualisasi (Skenario 1)	Dengan Virtualisasi (Skenario 2 & 3)
Server	192.168.0.1/24	- 192.168.0.1/24 (Web) - 10.0.0.2/8? (DB)	- 192.168.0.1/24 (host) - 192.168.0.111/24 (Web, guest) - 192.168.0.222/24 (DB, guest)	- 192.168.0.1/24 (host) - 192.168.0.222/24 (guest)
Client	192.168.0.2/24	192.168.0.2/24	192.168.0.2/24	192.168.0.2/24

Tabel 6: Alokasi alamat IP untuk pengujian.

### 3.4.4 Alat Pengujian

Ada beberapa hal yang perlu diketahui mengenai konfigurasi khusus untuk alat pengujian adalah:

- `perfmon.exe` *by default* menampilkan tingkat penggunaan sumber daya CPU pada komputer di mana sistem operasi berjalan. Hyper-V yang bertipe *bare-metal*, di mana manajemen dilakukan melalui *Root Partition* (juga merupakan mesin virtual khusus), maka `perfmon.exe` akan menampilkan tingkat penggunaan sumber daya CPU (virtual) hanya sebatas pada mesin virtual tersebut.

Untuk mendapatkan tingkat penggunaan sumber daya CPU keseluruhan

pada komputer fisik seseorang hanya perlu menambahkan *counter* “\Hyper-V Hypervisor Logical Processor(\_Total)\% Total Run Time” pada `perfmon.exe`.

- Pada VMware ESXi, walaupun program *thick client* bisa menampilkan tingkat penggunaan sumber daya CPU tetapi program ini tidak menampilkan secara *real-time*, atau paling tidak, tidak seketika seperti yang diharapkan. Untuk itu diperlukan program lain untuk melakukan ini. Untuk itu program `esxtop` yang digunakan dalam pengujian ini. Untuk menggunakan `esxtop` bisa dilakukan dengan menekan tombol Alt+F1 pada komputer di mana ESXi berjalan untuk membuka *console* tersembunyi. Lalu setelah mengetikkan “unsupported” sebuah *console* yang mirip *shell* pada sistem operasi UNIX/UNIX-like akan tampil, dari situ `esxtop` bisa dijalankan langsung dengan mengetikkan “esxtop”.

### 3.5 Implementasi Rancangan Pengujian

Suatu implementasi dari rancangan pengujian biasanya diwujudkan dengan membuat program/*script* dari skenario yang dilakukan. Dalam tesis ini ketiga skenario cukup sederhana sehingga bisa dijalankan hanya dengan beberapa baris perintah, tetapi untuk skenario pertama penulis membuatnya dalam sebuah *script* untuk kemudahan dan dapat dilihat pada Lampiran A.