

BAB 4 PENGUJIAN DAN ANALISIS

4.1 Pengujian

Pengujian terbagi lima bagian, pengujian pada saat *native* dan saat menggunakan virtualisasi dengan empat pendekatan yang berbeda. Ini artinya dari delapan solusi virtualisasi yang diujikan semua hasilnya dikelompokkan berdasarkan pendekatan yang digunakan. Berikut adalah daftar pengelompokan untuk pengujian:

1. Native
2. Operating system-level virtualization
 - Linux-VServer
 - OpenVZ
3. Paravirtualization
 - VMware ESXi
 - VMware Server
 - Xen
4. Hardware-assisted virtualization
 - Hyper-V
 - VirtualBox
 - Virtual Server

- VMware ESXi
- VMware Server
- Xen

5. Full virtualization

- VirtualBox
- Virtual Server
- VMware ESXi
- VMware Server

4.1.1 Native

Pada pengujian skenario pertama dengan satu komputer, program `httperf` dijalankan pada komputer *client* agar melakukan *requests* kepada komputer *server* selama 10 detik. Hasil keluaran `httperf` yang perlu diperhatikan adalah besarnya data yang ditransfer oleh Web server kepada `httperf`. Pada pengujian di sini, karena skenarionya adalah dua pengunjung yang melihat dua halaman berbeda dalam satu detik maka ada dua hasil data, yaitu 12759 *bytes* dan 26238 *bytes*. Meskipun besar data yang ditransfer ini bukan metrik yang digunakan dalam pengujian, tetapi informasi ini bisa menunjukkan apakah pengujian-pengujian selanjutnya telah berjalan dengan benar.

Apa yang menjadi metrik dalam skenario pertama adalah persentase penggunaan sumber daya CPU, sehingga program `sar` (atau program khusus dengan fungsi serupa untuk solusi tertentu; lihat bagian 3.1.2.4) perlu dijalankan

pada komputer *server* selama *requests* sedang berlangsung. Pada pengujian ini sendiri program *sar* hanya dijalankan selama 5 kali dengan interval setiap 1 detik (*sar -u 1 5*). Setelah selesai program *sar* menampilkan rata-rata penggunaan sumber daya CPU pada komputer *server* di mana ia dijalankan. Karena perbedaan antara pengujian *native* satu komputer dengan dua komputer hanyalah letak dari *database server* maka untuk konfigurasi dua komputer program *sar* juga dijalankan pada komputer untuk *database*.

Untuk skenario Kompresi atau Kompilasi, salah satu skenario dijalankan dengan cara seperti yang ditunjukkan pada Lampiran A sebanyak minimal tiga kali lalu diambil rata-ratanya. Setelah tugas utama selesai dijalankan program *time* menampilkan berapa waktu yang dibutuhkan untuk menjalankan tugas tersebut. Kedua skenario ini juga dijalankan hanya pada satu komputer karena tidak melibatkan jaringan.

	Skenario Aplikasi Web (dalam %CPU)	Skenario Kompresi (dalam detik)	Skenario Kompilasi (dalam detik)
Native (1 PC)	35	57	83
Native (2 PC, Web server)	34	-	-
Native (2 PC, database server)	1	-	-

Tabel 7: Hasil data pengujian *native*.

Seperti yang terlihat dalam Tabel 7 di atas, tingkat persentase penggunaan sumber daya CPU pada konfigurasi dua komputer, komputer *database server* ternyata hanya terpakai 1%. Nilai itu sendiri adalah hasil rata-rata dari nilai yang berfluktuasi, antara 0% hingga 2%. Bisa terlihat bahwa persentase penggunaan sumber daya CPU pada pengujian dengan konfigurasi satu komputer merupakan

gabungan dari konfigurasi dua komputer.

4.1.2 Operating System-Level Virtualization

Pada pengujian operating system-level virtualization ada dua solusi yang terlibat, Linux-VServer dan OpenVZ. Meskipun pada pendekatan ini *guest* menggunakan *kernel* yang sama dengan *host* (*x86-64*), tetapi apa saja yang ada di dalam *guest* (program, *library*, dll) tetap menggunakan versi *32-bit* (*x86*) sesuai dengan yang ditampilkan dalam Tabel 5.

	Skenario Aplikasi Web (dalam %CPU)	Skenario Kompresi (dalam detik)	Skenario Kompilasi (dalam detik)
Linux-VServer	36	58	84
OpenVZ	37	58	86

Tabel 8: Hasil data pengujian pendekatan OS-level virtualization.

Dari Tabel 8 dapat terlihat bahwa dalam ketiga skenario keduanya menunjukkan kinerja yang tidak jauh berbeda, hanya terpaut 1-2 poin, bahkan dalam skenario Kompresi hasilnya sama.

4.1.3 Paravirtualization

Pada tahun 2005 VMware mengeluarkan suatu spesifikasi antarmuka untuk virtualisasi, *Virtual Machine Interface (VMI)*, sebagai mekanisme komunikasi antara sistem operasi *guest* dengan perangkat lunak virtualisasi. Antarmuka ini kini memungkinkan produk dari VMware untuk melakukan virtualisasi dengan pendekatan *paravirtualization*, seperti yang dilakukan Xen.

	Skenario Aplikasi Web (dalam %CPU)	Skenario Kompresi (dalam detik)	Skenario Kompilasi (dalam detik)
VMware ESXi	42	58	107
VMware Server	42	61	108
Xen	38	58	92

Tabel 9: Hasil data pengujian pendekatan paravirtualization.

Hasilnya, pada Tabel 9 menunjukkan bahwa Xen, sebagai salah satu perintis pendekatan *paravirtualization* pada arsitektur x86, memiliki kinerja sedikit lebih baik dibanding solusi lainnya. Meskipun begitu, pada skenario Kompresi Xen memiliki kinerja yang sama dengan VMware ESXi, yang juga sedikit lebih baik dibanding VMware Server.

4.1.4 Hardware-Assisted Virtualization

Pada pengujian ini, solusi virtualisasi yang menawarkan pendekatan *hardware-assisted virtualization* sebenarnya ada lima: Hyper-V, VirtualBox, Virtual Server, VMware ESXi, dan VMware Server, akan tetapi VMware ESXi tidak bisa diikutsertakan. Ini dikarenakan VMware ESX/ESXi tidak mendukung sistem operasi 32-bit sebagai *guest* dengan pendekatan ini, hanya sistem operasi *guest* 64-bit yang didukung (Mattson, 2008).

	Skenario Aplikasi Web (dalam %CPU)	Skenario Kompresi (dalam detik)	Skenario Kompilasi (dalam detik)
Hyper-V	44	65	133
VirtualBox	46	68	151
Virtual Server	44	68	118
VMware ESXi	-	-	-
VMware Server	42	62	113
Xen	43	63	106

Tabel 10: Hasil data pengujian pendekatan hardware-assisted virtualization.

Hasil pengujian pada Tabel 10 menunjukkan VMware Server memiliki

kinerja sedikit lebih baik dibandingkan solusi lainnya. Hal menarik perhatian di sini adalah kinerja VirtualBox pada skenario Kompilasi yang memiliki selisih cukup jauh bila dibandingkan dengan VMware Server.

Satu hal yang bisa menjadi pertimbangan adalah, tidak seperti sistem GNU/Linux yang seluruh *daemon*-nya dimatikan, pada *default install* sistem operasi Windows Server 2008, ada beberapa *services* yang tetap berjalan karena diperlukan. *Services* ini terkadang menyebabkan tingkat penggunaan sumber daya CPU berfluktuasi antara 0% hingga 2%. Untuk pengujian ini VirtualBox dijalankan melalui perintah `VBoxHeadless`.

4.1.5 Full Virtualization

Pada pengujian ini, solusi virtualisasi yang menawarkan pendekatan *full virtualization* sebenarnya ada empat, yaitu VirtualBox, Virtual Server, VMware ESXi, dan VMware Server, tetapi dalam pengujian ini ternyata pada saat menggunakan Virtual Server, *kernel* sistem operasi *guest* tidak dapat berjalan. Dalam sebuah diskusi di *mailing list*, H. Peter Anvin, salah satu pengembang *kernel* Linux mengatakan, kalau ini adalah *bug* yang terdapat pada Virtual Server (Ehrhardt, 2008)

	Skenario Aplikasi Web (dalam %CPU)	Skenario Kompresi (dalam detik)	Skenario Kompilasi (dalam detik)
VirtualBox	48	78	165
Virtual Server	-	-	-
VMware ESXi	42	64	98
VMware Server	42	63	102

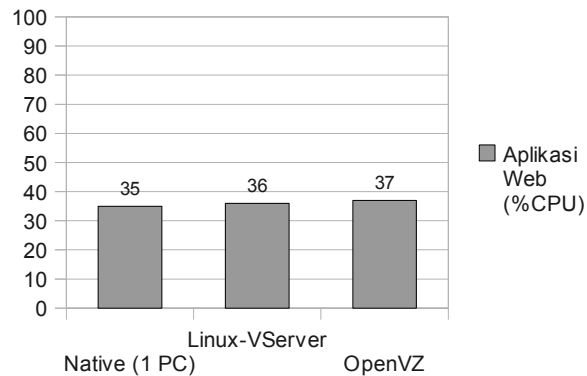
Tabel 11: Hasil data pengujian pendekatan *full virtualization*.

Data hasil pengujian dengan pendekatan *full virtualization* pada Tabel 11 di atas menunjukkan kinerja VirtualBox pada skenario Aplikasi Web dan Kompresi memperlihatkan perbedaan yang cukup menarik perhatian jika dibandingkan dengan kedua solusi lainnya. Ini karena VirtualBox menunjukkan kinerja yang lebih rendah, walaupun tidak terlalu jauh, tapi cukup menarik perhatian. Bahkan pada skenario Kompilasi VirtualBox menunjukkan perbedaan lebih jauh lagi yang mengundang pertanyaan dan akhirnya mendorong dilakukannya pengujian ulang (dibahas pada sub-bab 4.5). Untuk pengujian ini VirtualBox juga dijalankan melalui perintah `VBoxHeadless`.

4.2 Analisis

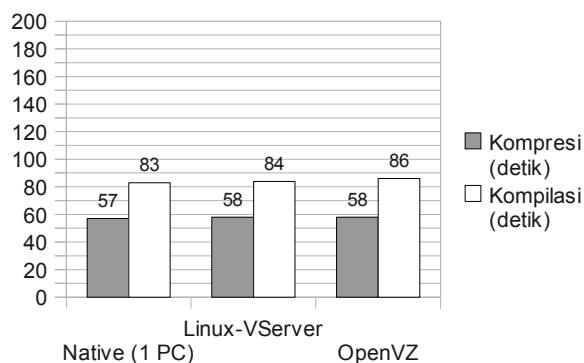
4.2.1 Operating System-Level Virtualization

Dari hasil pengujian pada Gambar 11, untuk skenario Aplikasi Web, Linux-VServer hanya menunjukkan peningkatan 1% dan OpenVZ menunjukkan peningkatan 2% dibandingkan persentase penggunaan sumber daya CPU pada saat *native*.



Gambar 11: Grafik perbandingan pendekatan OS-level virtualization dengan native (skenario Aplikasi Web).

Sedangkan untuk skenario Kompresi keduanya, Linux-VServer dan OpenVZ, hanya 1 detik lebih lama dibandingkan saat *native*, menjadi 58 detik. Untuk skenario Kompilasi Linux-VServer juga lebih lama 1 detik dibandingkan saat *native*, menjadi 84 detik, tetapi OpenVZ mulai menunjukkan perbedaannya untuk skenario Kompilasi ini, yaitu lebih lama 3 detik dibandingkan *native* menjadi 86 detik.



Gambar 12: Grafik perbandingan pendekatan OS-level virtualization dengan native (skenario Kompresi & Kompilasi).

Tabel 12 menunjukkan persentase perbedaan kinerja dengan implementasi *native*. Kecuali pada skenario Kompresi, di situ ditunjukkan bahwa OpenVZ

persentase-persentase perbedaannya lebih besar dibandingkan Linux-VServer. Sebagai kesatuan sebagai solusi virtualisasi dengan pendekatan *operating system-level virtualization*, Linux-VServer dan OpenVZ, menunjukkan persentase rata-rata perbedaan kinerja dengan implementasi *native* kurang dari 5% pada skenario Aplikasi Web, kurang dari 2% pada skenario Kompresi, dan kurang dari 3% pada skenario Kompilasi. Persentase perbedaan yang kurang dari 5% pada seluruh skenario dibanding pada saat *native* adalah perbedaan yang cukup kecil.

	Linux-VServer	OpenVZ	Rata-Rata Virtualiasi
Aplikasi Web	2,85%	5,71%	4,29%
Kompersi	1,75%	1,75%	1,75%
Kompilasi	1,20%	3,61%	2,41%

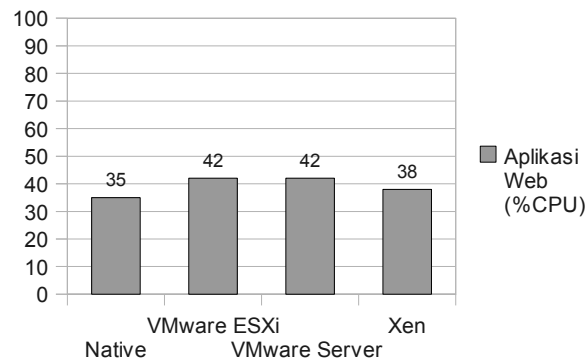
Tabel 12: Persentase perbedaan kinerja pendekatan OS-level virtualization dengan native.

Hal lain yang bisa menjadi pertimbangan di sini adalah bahwa OpenVZ menawarkan fitur lebih lengkap dibandingkan Linux-VServer. Salah satunya adalah *live migration* yang memungkinkan suatu mesin virtual untuk dipindahkan ke komputer lain tanpa mematikan mesin virtual tersebut. Ini menunjukkan seolah-olah fitur lebih yang ditawarkan OpenVZ harus dibayar dengan sedikit penurunan dalam kinerjanya.

4.2.2 Paravirtualization

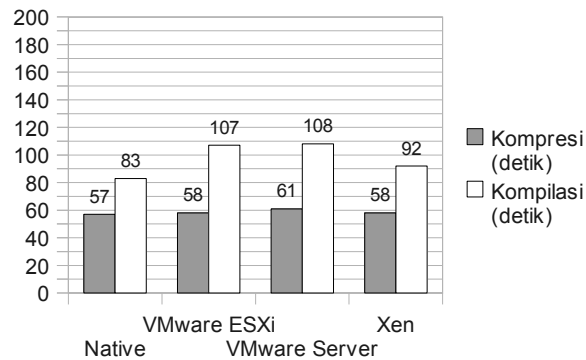
Meskipun bukan yang pertama, Xen, sebagai salah satu perintis dan pendorong utama digunakannya *paravirtualization* pada arsitektur x86/x86-64 menunjukkan keunggulannya di sini. Dalam melakukan virtualisasi pada skenario pertama Xen hanya terpaut 3% dalam penggunaan sumber daya CPU dibanding

saat native, yaitu meningkat menjadi 38%. Dua solusi lainnya dari VMware menunjukkan peningkatan yang sama, sebesar 7%, menjadi 42% (Gambar 13).



Gambar 13: Grafik perbandingan pendekatan paravirtualization dengan native (skenario Aplikasi Web).

Sesuai Gambar 14, untuk skenario Kompresi VMware ESXi dan Xen hanya membutuhkan waktu 1 detik lebih lama dibandingkan saat *native*, yaitu 58 detik, sedangkan VMware Server 4 detik lebih lama menjadi 61 detik. Pada skenario Kompilasi, Xen menunjukkan lagi kelebihanannya dengan hanya memakan waktu lebih lama 9 detik dibanding saat *native* menjadi 92 detik. Untuk VMware ESXi dan VMware Server, berturut-turut, membutuhkan waktu 107 detik dan 108 detik, lebih lama 24 detik dan 25 detik dibanding saat *native*.



Gambar 14: Grafik perbandingan pendekatan paravirtualization dengan native (skenario Kompresi & Kompilasi).

Tabel 13 menunjukkan, persentase perbedaan kinerja antara VMware ESXi dan VMware Server terhadap implementasi *native* hampir sama, kecuali pada skenario Kompresi. Dimana VMware ESXi hanya 1,75% sedangkan VMware Server 7,01%. Xen sendiri, untuk skenario Kompresi persentase perbedaannya terhadap *native* sama seperti VMware ESXi, yaitu 1,75%. Untuk skenario lainnya, persentase perbedaan Xen dengan saat *native* kurang dari setengah yang dimiliki dua solusi lainnya, yaitu 8,5% untuk skenario Aplikasi Web dan 10,84% untuk skenario Kompilasi.

	VMware ESXi	VMware Server	Xen	Rata-Rata Virtualisasi
Aplikasi Web	20,00%	20,00%	8,57%	16,19%
Kompresi	1,75%	7,01%	1,75%	3,51%
Kompilasi	28,92%	30,12%	10,84%	23,29%

Tabel 13: Persentase perbedaan kinerja pendekatan paravirtualization dengan *native*.

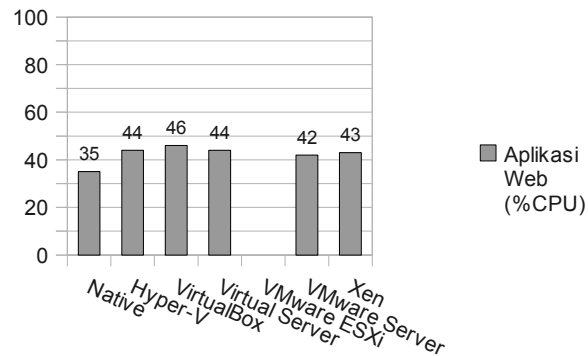
Persentase perbedaan rata-rata ketiganya terhadap implementasi *native* menunjukkan, 16,17% untuk skenario Aplikasi Web, 3,50% untuk skenario Kompresi, dan 23,28% untuk skenario Kompilasi. Dengan persentase

perbedaannya yang kurang dari 20% untuk skenario Aplikasi Web, selisih yang dihasilkan oleh pendekatan ini dianggap masih cukup rendah. Mengingat apa yang telah disebutkan sebelumnya, bahwa rata-rata komputer *server* masih banyak yang *underutilized* dan hanya terpakai kurang dari seperlima dari sumber daya yang ada (VMware, 2008; Sun, 2008).

4.2.3 Hardware-Assisted Virtualization

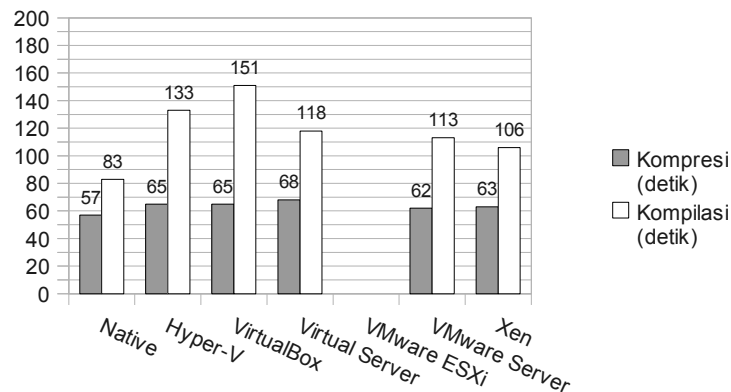
Dalam pengujian dengan pendekatan ini yang terlihat pada Gambar 15, hasil dari skenario Aplikasi Web kedua solusi dari Microsoft, Hyper-V dan Virtual Server, menunjukkan penggunaan sumber daya CPU yang sama, 44%, atau naik 9% dari implementasi *native* yang 35%. VirtualBox naik 11% menjadi 46%, VMware Server menunjukkan kenaikan terendah yang hanya 7% menjadi 42%, dan Xen naik 8% menjadi 43%.

Seperti yang telah disebutkan pada sub-bab Pengujian, khususnya butir 4.14, sistem operasi Windows Server 2008 yang digunakan Hyper-V dan Virtual Server memiliki beberapa *services* yang berjalan dibelakang layar. Sebagian sifatnya *mandatory* dan terkadang memakan sumber daya CPU hingga 1%-2%. Untuk sebagian kalangan ini bisa dijadikan bahan pertimbangan baru dalam menilai kedua solusi tersebut.



Gambar 15: Grafik perbandingan pendekatan hardware-assisted virtualization dengan native (skenario Aplikasi Web).

Gambar 16, seperti yang telah disebutkan pada butir 4.1.4, menunjukkan kinerja VirtualBox yang rendah dan cukup menarik perhatian. Ini dikarenakan pada skenario Kompilasi VirtualBox memiliki selisih waktu cukup jauh dibandingkan dengan solusi virtualisasi dengan kinerja cukup baik di sini, yaitu VMware Server dan Xen. VirtualBox membutuhkan waktu 38 detik lebih lama dari VMware Server yang hanya 113 detik. Meskipun tidak sejauh VirtualBox, kinerja rendah yang ditunjukkan Hyper-V juga cukup menarik perhatian, 20 detik lebih lama dibandingkan VMware Server. Perbedaan ini sudah tentu jauh lebih buruk lagi kalau dibandingkan dengan pada saat *native* yang hanya membutuhkan 83 detik.



Gambar 16: Grafik perbandingan pendekatan hardware-assisted virtualization dengan native (skenario Kompresi & Kompilasi).

Tabel 14 bisa terlihat, persentase perbedaan rata-rata seluruh pendekatan hardware-assisted virtualization menunjukkan peningkatan yang lebih dibanding rata-rata pendekatan *paravirtualization*. Pada skenario Kompilasi yang sebelumnya pada pendekatan *OS-level virtualization* dan *paravirtualization* hanya menunjukkan perbedaan kurang dari 4% dibanding saat *native*, di sini bisa meningkat lebih dari tiga kali menjadi 15,34%.

	Hyper-V	VirtualBox	Virtual Server	VMware ESXi	VMware Server	Xen	Rata-Rata Virtualisasi (tanpa VMware ESXi)
Aplikasi Web	25,71%	31,43%	25,71%	-	20,00%	22,86%	25,14%
Kompresi	14,03%	14,04%	19,30%	-	8,77%	10,53%	13,33%
Kompilasi	60,24%	81,93%	42,17%	-	36,14%	27,71%	49,64%

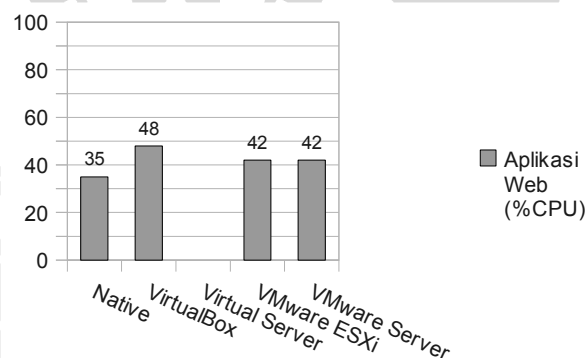
Tabel 14: Persentase perbedaan pendekatan kinerja hardware-assisted virtualization dengan native.

Pada dasarnya rata-rata persentase perbedaan solusi-solusi virtualisasi pada pendekatan *hardware-assisted virtualization* ini lebih tinggi dibandingkan *paravirtualization*, seperti pada skenario Kompresi dan Kompilasi yang bisa lebih

dari dua kali lipat. Meskipun begitu, kalau bukan karena kinerja buruk Hyper-V dan VirtualBox rata-rata persentase perbedaan pada skenario Kompilasi bisa lebih rendah dari 40%.

4.2.4 Full Virtualization

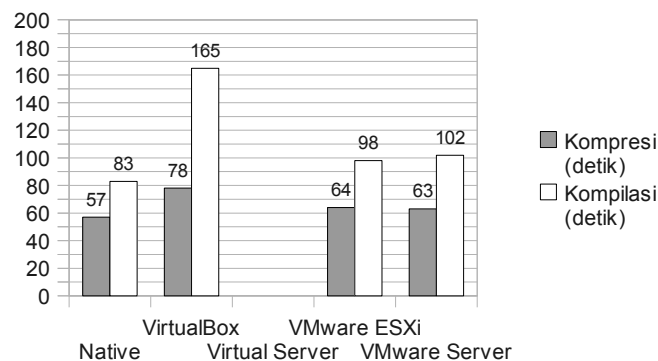
Dalam pendekatan ini, Gambar 17 menunjukkan bahwa VirtualBox memiliki selisih kinerja yang cukup kontras bila dibandingkan dengan implementasi *native*. Pada skenario Aplikasi Web VirtualBox menggunakan sumber daya CPU 13% lebih tinggi dibandingkan implementasi *native*, yaitu 48%, sedangkan kedua solusi lainnya dari VMware hanya 7% lebih tinggi, yaitu pada 42%.



Gambar 17: Grafik perbandingan pendekatan full virtualization dengan native (skenario Aplikasi Web).

Gambar 18 masih menunjukkan perbedaan yang cukup kontras untuk VirtualBox. Untuk skenario Kompresi VirtualBox membutuhkan waktu lebih lama 21 detik dibandingkan pada saat *native*, yaitu 78 detik, sedangkan untuk skenario Kompilasi VirtualBox membutuhkan hampir dua kali lipat lebih lama dibanding

pada saat *native*, yaitu 165 detik atau naik 82 detik. Untuk VMware ESXi dan VMware Server sendiri, keduanya menunjukkan kinerja yang tidak jauh berbeda dan cukup dapat diterima dibandingkan dengan pada saat *native*.



Gambar 18: Grafik perbandingan pendekatan full virtualization dengan native (skenario Kompresi & Kompilasi).

Seperti yang terlihat pada Tabel 15, kedua solusi VMware masih menunjukkan persentase perbedaan kinerja dengan implementasi *native* yang masih bisa diterima. Meskipun untuk skenario Aplikasi Web dan Kompresi cukup bisa diterima, tetapi pada skenario Kompilasi VirtualBox membutuhkan waktu hampir dua kali lipat dirasa terlalu tinggi.

	VirtualBox	Virtual Server	VMware ESXi	VMware Server	Rata-Rata Virtualisasi (tanpa Virtual Server)
Aplikasi Web	37,14%	-	20,00%	20,00%	25,71%
Kompresi	36,84%	-	12,28%	10,53%	19,88%
Kompilasi	98,80%	-	18,07%	22,89%	46,58%

Tabel 15: Persentase perbedaan pendekatan full virtualization dengan native.

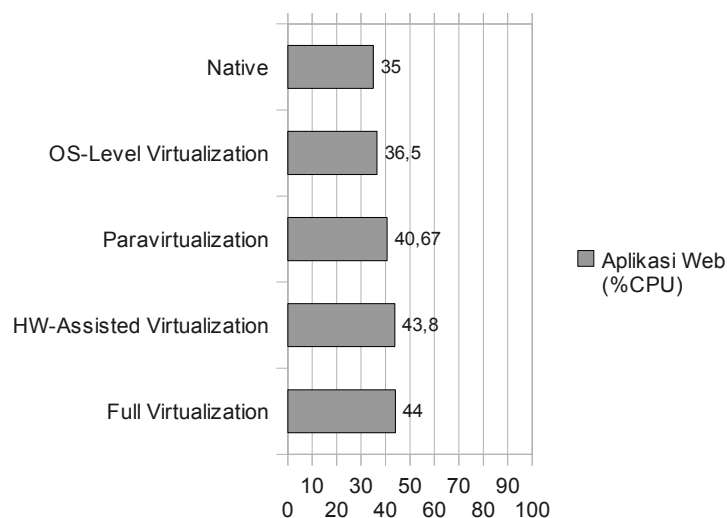
Tergantung skenario yang diujikan, pada kasus tertentu salah satu—*full virtualization* atau *hardware-assisted virtualization*—akan lebih baik dari yang lainnya, tetapi bisa sebaliknya pada skenario yang lain seperti yang dengan jelas

ditunjukkan oleh VMware Server. Sehingga kurang-lebih persentase perbedaan kinerja rata-rata seluruh solusi virtualisasi dengan pendekatan *full virtualization* sama dengan *hardware-assisted virtualization* dan masih dapat diterima. Meskipun begitu, tanpa VirtualBox rata-rata tersebut bisa lebih baik lagi.

Untuk sebagian pihak, persentase perbedaan kinerja yang bisa lebih dari 50% terlihat besar, tetapi apabila suatu organisasi memiliki komputer dengan tingkat penggunaan sumber daya CPU yang hanya 10%, maka peningkatan 50% pada saat menggunakan virtualisasi berarti akan membuatnya menjadi 15%. Ini juga artinya masih tersisa 85% sumber daya CPU yang tersedia untuk konsolidasi.

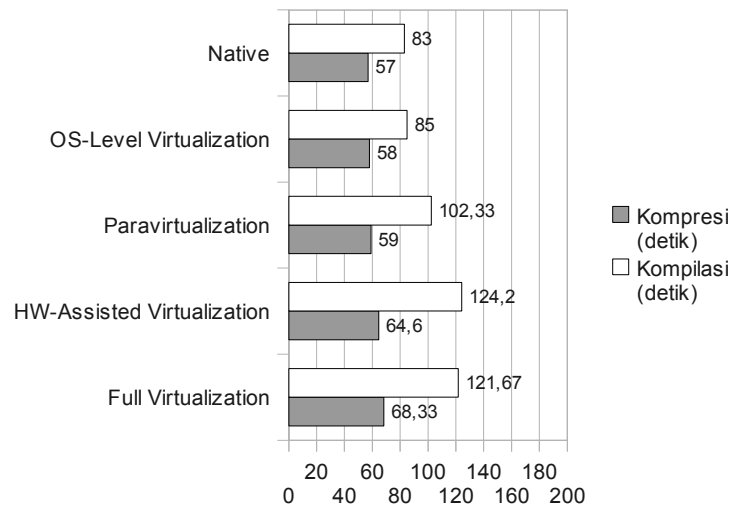
4.3 Analisis Tren Kinerja

Pada Gambar 19 di bawah bisa dilihat gambaran besar dari rata-rata perbedaan kinerja yang dihasilkan dari masing-masing pendekatan virtualisasi untuk skenario Aplikasi Web. Pendekatan *operating system-level virtualization* tidak mengambil sumber daya CPU lebih tinggi dari 2% (36,5%) dibandingkan pada saat *native* (35%). Pendekatan *paravirtualization* mengambil sumber daya CPU lebih banyak lagi tetapi tidak sampai 6% lebih tinggi (40,67%). Pendekatan *hardware-assisted virtualization* dan *full virtualization* hampir sama, berturut-turut, membutuhkan sumber daya CPU lebih tinggi 8,8% (43,8%) dan 9% (44%).



Gambar 19: Grafik perbandingan rata-rata kinerja dari pendekatan-pendekatan virtualisasi (skenario Aplikasi Web).

Untuk skenario Kompresi dan Kompilasi rata-rata kinerja masih menunjukkan tren/pola kinerja yang mirip dengan skenario Aplikasi Web. Pendekatan *operating system-level virtualization* membutuhkan waktu sedikit lebih lama 1-2 detik dibanding pada saat *native* untuk kedua skenario dan setelah itu diikuti pendekatan *paravirtualization* (59 detik untuk Kompresi dan 102,33 detik untuk Kompilasi). Untuk pendekatan *hardware-assisted virtualization* dan *full virtualization*, pada skenario Kompresi dan Kompilasi, perbedaan keduanya dengan pendekatan *paravirtualization* semakin terlihat lagi, terutama untuk skenario Kompilasi (Gambar 20). Pada skenario Kompresi *hardware-assisted virtualization* dan *full virtualization*, berturut-turut, membutuhkan 64,6 detik dan 68,33, sedangkan pada skenario Kompilasi keduanya berturut-turut membutuhkan 124,2 detik dan 121,67 detik.



Gambar 20: Grafik perbandingan rata-rata kinerja dari pendekatan-pendekatan virtualisasi (skenario Kompresi dan Kompilasi).

Berdasarkan hasil pengujian dan analisis di atas, kinerja *hardware-assisted virtualization* dibandingkan dengan kinerja pendekatan *full virtualization* menunjukkan hasil kinerja yang bercampur. Pada skenario Kompresi pendekatan *hardware-assisted virtualization* bisa lebih baik dibandingkan *full virtualization*, tetapi hasil sebaliknya ditunjukkan pada skenario Kompilasi. Menurut VMware (2007), kinerja pendekatan *hardware-assisted virtualization* yang tidak sesuai harapan ini dikarenakan *overhead* transisi antara *root mode* dan *non-root mode* yang terlalu tinggi.

4.4 Temuan Lain

Dalam pengujian dengan pendekatan *hardware-assisted virtualization* dan *full virtualization*, VirtualBox menunjukkan kinerja lebih rendah dan cukup menarik perhatian. Terutama pada skenario Kompilasi yang selisih kinerjanya

cukup jauh dibandingkan solusi lainnya. Ini akhirnya mendorong untuk dilakukannya pengujian menggunakan versi VirtualBox sebelumnya, yaitu versi 2.1 (versi yang diujikan di atas adalah terbaru saat pengujian dilakukan, yaitu versi 2.2).

	Skenario Aplikasi Web (dalam %CPU)	Skenario Kompresi (dalam detik)	Skenario Kompilasi (dalam detik)
VirtualBox (Hardware-Assisted virtualization)	46	68	165
VirtualBox (Full virtualization)	57	74	159

Tabel 16: Hasil uji ulang VirtualBox.

Dari hasil pengujian tambahan untuk VirtualBox ternyata versi 2.1 ini menunjukkan hasil yang lebih mengejutkan untuk skenario Aplikasi Web. Meskipun pada saat menggunakan pendekatan *hardware-assisted virtualization* hasilnya sama dengan versi 2.2 (menggunakan 46% sumber daya CPU), tetapi pada saat menggunakan pendekatan *full virtualization* hasilnya turun menjadi 57%. Ini artinya ada peningkatan penggunaan sumber daya CPU sebesar 9%, tetapi untuk skenario Kompresi dan Kompilasi, versi 2.1 ini justru menunjukkan sedikit peningkatan.

Hasil di atas, selain menunjukkan penurunan kinerja cukup signifikan pada skenario Aplikasi Web dengan pendekatan *full virtualization*, juga menunjukkan pola kinerja kurang-lebih sama untuk skenario Kompresi dan Kompilasi (terutama kinerja skenario Kompilasi memiliki selisih cukup jauh dibanding solusi lainnya). Karena hal tersebut, maka diputuskan data yang digunakan adalah tetap data yang dihasilkan oleh VirtualBox versi 2.2.