

# PERANCANGAN, IMPLEMENTASI, DAN ANALISIS KINERJA VIRTUALISASI SERVER MENGGUNAKAN PROXMOX, VMWARE ESX, DAN OPENSTACK

Arief Arfriandi  
Magister Teknologi Informasi Universitas Gadjahmada Yogyakarta  
arfriandi35@gmail.com

## ABSTRACT

*Several types of processor has more than one cores, especially on the server. By looking at the potential core processors that have more than one, we can use it to run applications and services simultaneously using virtualization techniques on a computer server. The concept of high availability clusters contained in server virtualization can reduce costs and simplify management of information technology services. Accordingly, in this study measured against some server virtualization using overhead, and linearity to determine the performance of server virtualization. The method this research is by design, and implementation Proxmox Virtual Environment, VMWare ESXi and Openstack on multicore servers, and analyzing the performance of the server virtualization. The main results of these studies are design, and implementation server virtualization using Proxmox Virtual Environment, VMware ESXi, and Openstack and also a description of analysis, the performance of each model the server virtualization.*

*Keywords : Virtualization, High Availability Server, Overhead, Linearity*

## INTISARI

Banyak tipe *processor* yang mempunyai inti lebih dari satu, terutama pada *server*. Dengan melihat potensi *processor* yang mempunyai inti lebih dari satu tersebut, kita dapat memanfaatkannya untuk menjalankan aplikasi-aplikasi dan *services* secara bersamaan menggunakan teknik *virtualisasi* pada komputer *server*. Konsep *cluster high availability* yang terdapat pada *virtualisasi server* dapat mengurangi biaya dan menyederhanakan pengelolaan pelayanan teknologi informasi. Berdasarkan hal tersebut, dalam penelitian ini dilakukan pengukuran terhadap beberapa *virtualisasi server* menggunakan metode *overhead*, dan linearitas untuk mengetahui kinerja *virtualisasi server*. Metode yang digunakan adalah dengan cara perancangan, dan implementasi *Proxmox, vmware esx* dan *openstack* pada *server multicore*, dan dilakukan analisa kinerja dari *virtualisasi server* tersebut. Hasil utama dari penelitian ini adalah perancangan, dan implementasi *virtualisasi server* menggunakan *Proxmox, vmware*, dan *openstack* serta deskripsi analisa, hasil kinerja masing-masing model *virtualisasi server* tersebut.

Kata-kunci: *Virtualisasi, High Availability Server, Overhead, Linearitas*

## PENDAHULUAN

Penggunaan server dengan processor berinti banyak semakin banyak dijumpai, masing-masing vendor produsen processor mempunyai berbagai macam tipe processor dengan inti lebih dari satu. Processor dengan inti lebih dari satu mempunyai kemampuan yang cukup untuk melakukan berbagai macam proses secara bersamaan, akan tetapi belum semua aplikasi pada saat ini yang dapat memanfaatkan secara optimal prosesor berinti banyak (multiprosesor) tersebut. Berdasarkan pengamatan dari Tony Iams, analis senior di D.H. Brown Associates Inc, NY, server di sebagian besar organisasi hanya menggunakan 15-20% dari kapasitas sesungguhnya, tentu saja angka tersebut merupakan rasio yang jauh dari ideal. Dengan melihat potensi prosesor yang

mempunyai inti lebih dari satu tersebut, dapat kita manfaatkan untuk menjalankan aplikasi-aplikasi dan *services* secara bersamaan menggunakan teknik *virtualisasi* pada komputer *server*. Tanggapan vendor produsen processor terhadap semakin dibutuhkannya teknologi *virtualisasi* juga dibuktikan dengan menambahkan dukungan terhadap proses *virtualisasi* di dalam processor yang mereka produksi sehingga teknologi *virtualisasi* yang akan digunakan pada processor berinti banyak tersebut dapat memiliki kinerja yang optimal ketika menjalankan sistem operasi dan program aplikasi secara virtual.

Teknologi *virtualisasi server* yang diterapkan pada server dengan processor berinti lebih dari satu, dapat kita manfaatkan untuk menjalankan aplikasi-aplikasi dan

services secara virtualisasi. Konsep virtualisasi server merupakan paradigma baru dalam perkembangan teknologi. Hal ini dikarenakan virtualisasi server memungkinkan penggunaan satu perangkat keras untuk menjalankan beberapa sistem operasi dan services pada saat

yang sama, sehingga client dapat menggunakan sumber daya tersebut sesuai dengan kebutuhannya berbasis internet.

Virtualisasi server adalah penggunaan perangkat lunak yang memungkinkan satu perangkat keras untuk menjalankan beberapa sistem operasi dan services pada saat yang sama, sedangkan virtual server adalah penggunaan perangkat lunak yang memungkinkan banyak perangkat keras untuk menjalankan satu sistem secara terpadu. Teknologi virtualisasi server ini bertujuan untuk menghindari pemborosan daya proses yang mahal atau dengan kata lain meningkatkan efisiensi serta mengoptimalkan penggunaan processor berinti lebih dari satu. Penghematan lain adalah biaya listrik karena hanya menggunakan satu atau sedikit server saja.

Pada saat ini banyak sekali metode perancangan virtualisasi server dengan tipe cluster high availability, antara lain menggunakan proxmox, vmware esx dan openstack yang merupakan tipe virtualisasi server yang free dan mudah dalam instalasi. Diantara beberapa tipe virtualisasi server tersebut terdapat perbedaan dalam hal kinerja. Hal ini dikarenakan beberapa tipe virtualisasi server tersebut dibuat oleh beberapa vendor yang berbeda.

### Tinjauan Pustaka

Beberapa penelitian yang berkaitan dengan *Cloud Computing* antara lain Zulhaidi dan Dipojono (2002), Penelitian ini membahas menjelaskan desain dan implementasi perangkat keras yang diperlukan untuk *cluster* pada *Cloud Computing*.

Penelitian lain yang berhubungan dengan *virtualisasi* juga dilakukan oleh Fauzi (2008). Dalam tesisnya Fauzi mencari tahu sejauh mana atau seberapa baik skalabilitas saat menggunakan *virtualisasi* jika dibandingkan dengan implementasi *native* dengan menggunakan satu pendekatan *virtualisasi*. Penelitian lain yang berkaitan dengan *virtualisasi server* yang lain Chen, dkk (2006). Penelitian ini bertujuan untuk menyediakan infrastruktur TI yang *reliable*, dalam arti bisa melakukan *update* dan *patch* tanpa mematikan atau *me-restart* mesin

*server* produksi yang sedang berjalan. Penelitian lain yang berhubungan dengan virtualisasi juga dilakukan oleh Benjamin, dkk (2007). Penelitian ini bertujuan untuk mengetahui skalabilitas empat tipe *virtualisasi server*, metode yang digunakan dalam pengukurannya adalah menggunakan metode *overhead*, *linearitas* dan *usability*.

Peneliti yang lain adalah Garnieri (2010), Penelitian ini menargetkan penggunaan *virtualisasi server* pada perusahaan besar, yang memfokuskan pada minimalisasi *downtime* pada saat *maintance* perangkat keras *server* dengan menggunakan aplikasi *VMware Infrastructure 3*. Selain itu peneliti Ardianto (2011), menjelaskan bagaimana merancang *virtualisasi appliances* dengan memanfaatkan metode *virtualisasi*.

Penelitian oleh Ardianto, Novan (2011), dalam penelitian tersebut berusaha dipadukan antara mesin *virtualisasi* dengan perangkat lunak yang sebenarnya maka akan terbentuklah *virtualisasi appliances*, dalam penelitian tersebut dirancang suatu *virtualisasi appliance* menggunakan metode *virtualisasi Proxmox* dan *virtualisasi box* untuk kemudian dianalisa kestabilannya. Kemudian mengimplementasikan *virtualisasi appliances* tersebut sehingga dapat dioperasikan dengan mudah di lingkungan *server* yang berbeda bahkan di OS yang berbeda. Faisal (2012), penelitian ini menekankan tentang faktor skalabilitas *private cloud computing* untuk layanan IAAS dan analisa kinerja dari *Cloud Computing*.

### Perumusan Masalah

Berdasarkan latar belakang diatas, maka didapat perumusan masalah yang akan diteliti yaitu bagaimana merancang, dan mengimplemestasikan *virtualisasi server* menggunakan *tipe* yang berbeda-beda yaitu *proxmox*, *vmware esx* dan *openstack*, serta menganalisa kinerja masing-masing *virtualisasi server* tersebut melalui pengukuran metrik skalabilitas yaitu menggunakan metode *overhead* dan *linearitas*.

### Tujuan Penelitian

Tujuan yang hendak dicapai dalam penelitian ini adalah :

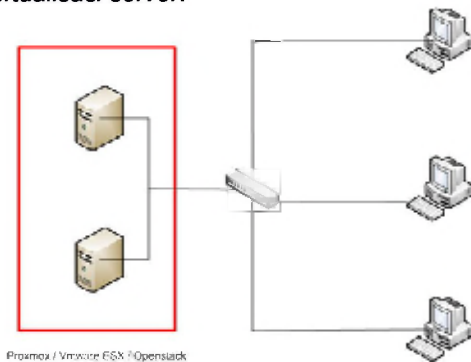
1. Merancang dan mengimplementasikan *virtualisasi server* menggunakan *proxmox VE 1.8*, *vmware esx* dan *openstack*
2. Menganalisa *virtualisasi server* yang berguna untuk mengetahui kinerja dari masing-masing tipe *virtualisasi server* tersebut melalui

pengukuran metrik skalabilitas yaitu menggunakan metode *overhead* dan linearitas.

#### Metode Penelitian

Penelitian ini merupakan sebuah penelitian observasi dengan mengimplementasikan sistem *virtualisasi server* dengan *Proxmox*, *vmware esx*, dan *openstack* yang merupakan tipe *cluster High Availability (HA)*. Penelitian observasi ini menggunakan metode analisis deskriptif, yang dilakukan dengan proses pengamatan langsung dan mempelajari observasi atas hasil yang terjadi dari analisis sistem ini. Pendekatan penelitian analisis deskriptif ini adalah untuk menggali informasi dari implementasi sistem *virtualisasi server* menggunakan *Proxmox*, *vmware esx*, dan *openstack*.

Dalam analisa *virtualisasi server*, peneliti berusaha membandingkan dengan *server* tradisional atau aplikasi *server* yang tidak menggunakan *virtualisasi*, selain itu juga dibandingkan dengan dua *virtual machine* yang berisi aplikasi *web server* dan *ftp server* pada *server virtualisasi* sehingga diperoleh data pembandingan dalam melakukan analisa kinerja dan pengaruh implementasi aplikasi yang berbeda pada *virtualisasi server*.



Gambar 1. Arsitektur dengan *virtualisasi server*

Skenario pengujian *virtualisasi server* sebagai berikut :

1. Membuat 2 *virtual machine*, dilakukan *update* dan *upgrade* secara bersamaan.
2. Membuat 2 *virtual machine*, setiap *virtual machine* berfungsi sebagai *server* yang berbeda. *Virtual machine* 1 difungsikan sebagai *web server* sedangkan *virtual machine* 2 difungsikan sebagai *ftp server* kemudian dilakukan pengiriman *file* sebesar 706.803.376 B.

3. Membuat beberapa *virtual machine*, *virtual machine* 1 di *install* aplikasi *web server* dan *virtual machine* 2 di *install ftp server*, sementara itu *virtual machine* yang lain tetap dijalankan, akan tetapi tidak diberikan aplikasi.
4. Membuat beberapa *virtual machine*, *virtual machine* 1 di *install* *web server*, sedangkan *virtual machine* yang lain di *install* aplikasi *ftp server* yang sama, dan dijalankan secara bersamaan kemudian setiap penambahan *virtual machine*, dicoba *upload* sebuah *file* dengan ukuran 706.803.376 B ke semua *ftp server* yang telah di *install* pada *virtual machine* secara bersamaan.

Analisa *virtualisasi server* menggunakan pengukuran metrik skalabilitas yaitu menggunakan metode *overhead* dan linearitas.

#### a. Overhead

Untuk evaluasi *overhead virtualisasi* yang disebabkan mekanisme *virtualisasi* dilakukan dengan membandingkan waktu eksekusi sebuah aplikasi yang dijalankan pada sistem non *virtualisasi* ( $T_a$ ) dengan aplikasi yang sama dijalankan pada sebuah mesin *virtualisasi* ( $T_{av}$ ). *Overhead* mungkin saja bisa diabaikan untuk sebuah mesin *virtualisasi* dan bisa menjadi signifikan ketika jumlah mesin *virtualisasi* dijalankan secara bersamaan. Disamping itu dibandingkan pula  $T_a$  dengan  $T_{av}$  ketika sejumlah  $n$  mesin *virtualisasi* dijalankan secara bersamaan. Pada skenario ini hanya sebuah mesin *virtualisasi* yang menjalankan aplikasi. Sedangkan  $n-1$  mesin *virtualisasi* yang lain bebas dari aplikasi.

#### b. Linearitas

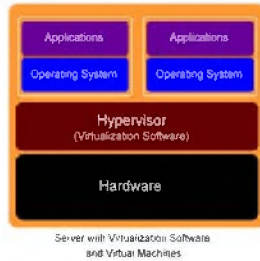
Untuk mengevaluasi perubahan skalabilitas pada saat jumlah mesin *virtualisasi* yang dijalankan meningkat, terlebih dahulu diukur waktu eksekusi sebuah aplikasi yang dijalankan pada mesin *virtualisasi*. Selanjutnya diukur waktu eksekusi aplikasi yang sama dijalankan secara bersamaan pada beberapa mesin *virtualisasi*.

### Landasan Teori

#### Pengertian *Virtualisasi*

*Virtualisasi / Virtualisasi* adalah sebuah teknik atau cara untuk membuat sesuatu dalam bentuk *virtualisasi*, tidak seperti kenyataan yang ada. *Virtualisasi* juga digunakan untuk mengemulasikan perangkat fisik komputer, dengan cara membuatnya

seolah-olah perangkat tersebut tidak ada (disembunyikan) atau bahkan menciptakan perangkat yang tidak ada menjadi ada.



Gambar 2. Konsep *Virtualisasi*

### Jenis-Jenis Pendekatan *Virtualisasi*

1. *Partial Virtualisasi*  
*Virtualisasi* parsial adalah bentuk *virtualisasi* pada sebagian dari perangkat keras. Perangkat lunak *virtualisasi* parsial akan mengemulasikan, seolah-olah perangkat komputer kita memiliki alat tersebut.
2. *Full Virtualisasi*  
*Virtualisasi* penuh berarti membuat seolah-olah ada komputer lain di dalam komputer. Dengan menginstal *Linux* dalam *Windows* Anda, demikian juga meng-*install Windows* dalam *Linux*.
3. *Hardware-assisted Virtualisation*  
Merupakan *virtualisation* yang didukung oleh *hardware*, jadi ada *hardware* khusus yang berguna untuk meningkatkan *performance* proses *virtualisasi*. *Hardware-assisted virtualisation* mempunyai *overhead* yang banyak, agar skalabilitas *guest OS* tidak terlalu turun, maka dibantu dengan *hardware*.

### Cluster

*Cluster* merupakan suatu set individual yang terhubung melalui perangkat keras dan perangkat lunak khusus, menyajikan gambar sistem tunggal untuk para penggunanya (Vrenios, 2002). Tiga tipe *cluster* yang dominan yaitu:

1. *High Performance Computing (HPC)*. Secara umum, tipe *cluster HPC* ditujukan pada bagaimana suatu proses komputasi dapat dipercepat, dengan demikian task dapat diselesaikan dengan lebih cepat. Contoh *cluster* jenis ini adalah *MPI, DSM, PVM*.
2. *High Availability (HA)*. Secara umum, tipe *cluster* ini ditujukan agar program yang dijadikan di atasnya bisa terus berjalan, sekalipun salah satu *node hang* atau *down*. Contohnya adalah fasilitas *cluster* pada *Proxmox*.

3. *Load-Balancing (LB)*. Secara umum, tipe *cluster* ini beroperasi dengan mendistribusikan beban pekerjaan secara merata melalui beberapa *node* yang bekerja di belakang (*back-end node*). Umumnya *cluster* ini akan dikonfigurasi sedemikian rupa dengan *front-end load-balancing redundant*.

### Overhead dan Linearitas

Overhead pada virtualisasi server adalah seberapa sering dan lamanya waktu yang dibutuhkan oleh hypervisor untuk menyelesaikan suatu proses dan menjalankan kembali virtual machine. Pada virtualisasi server, pengujian overhead dijalankan dengan memperbanyak virtual machine tanpa aplikasi. Linearitas pada virtualisasi server dapat diartikan sama dengan overhead, akan tetapi dalam pengujian linearitas, virtual machine yang ditambahkan diberikan aplikasi yang sama.

### IMPLEMENTASI SISTEM

#### Memeriksa fitur *Virtualisasi* Pada Processor

*Processor* yang mendukung *virtualisasi* dapat diperiksa melalui bios mainboard atau diperiksa secara *online* melalui *website* pabrik *processor* tersebut, dengan melihat apakah seri *processor* yang digunakan termasuk dalam daftar yang mendukung *virtualisasi*. Cara yang lain bisa dengan mengetikkan beberapa perintah di *console linux* :

```
egrep '(vmx|svm)' /proc/cpuinfo
```

Apabila hasilnya ada kata *vmx* jika menggunakan *processor Intel-VT* atau *svm* jika menggunakan *processor AMD-V*, berarti *processor* yang digunakan telah mendukung dan dilengkapi teknologi *virtualisasi* secara *hardware*.

#### *Proxmox Virtual Environment*

Pada skenario penelitian ini digunakan *Proxmox VE* versi 1.8 yang dapat diunduh pada *website resmi Proxmox Virtual Environment* di alamat <http://www.proxmox.com/products/proxmox-ve>. Beberapa hal yang perlu diperhatikan terkait *Proxmox*:

1. *Proxmox* disediakan hanya untuk mesin 64 bit, sehingga tidak bisa digunakan untuk mesin 32 bit.
2. Pada saat instalasi, *Proxmox* diinstalasikan langsung dari CD dan akan menghapus seluruh isi *harddisk*.

## VMware ESXi 5.0

Tipe *virtualisasi server* yang digunakan selain *proxmox virtual environment* adalah *VMware ESXi 5.0*. Beberapa hal yang perlu diperhatikan dalam melakukan instalasi dan implementasi *VMware ESXi 5.0* :

1. *VMware ESXi 5.0* dapat diperoleh di [http://downloads.vmware.com/d/info/datacenter\\_cloud\\_infrastructure/vmware\\_vsphere\\_hypervisor\\_esxi/5\\_0?cmd=displayKC&externalId=1993](http://downloads.vmware.com/d/info/datacenter_cloud_infrastructure/vmware_vsphere_hypervisor_esxi/5_0?cmd=displayKC&externalId=1993)
2. Dalam implementasi *VMware ESXi 5.0*, tidak memerlukan *base sistem operasi* dalam proses instalasi.
3. *VMware ESXi 5.0* merupakan sistem operasi yang berjalan mandiri, tanpa bergantung pada sistem operasi lain.
4. *VMware ESXi 5.0* membutuhkan *software tambahan VMware vSphere Client 5.0* untuk mengelola *virtualisasi* pada *VMware ESXi 5.0*.

## Openstack

*Openstack* yang diinstal, menggunakan sistem operasi *Ubuntu server 64 bit* versi 11.10 (*oneiric-ocelot*), konfigurasi teknis dapat dilihat pada Tabel 1. Pada *OpenStack*, volume yang akan dipakai menggunakan partisi *LVM (Logical Volume Management)*, untuk membuat partisi *LVM* pada *Ubuntu 11.10* dapat dilakukan ketika instalasi sistem.

Tabel 1. Spesifikasi teknis *master server openstack*

Keterangan	Spesifikasi Teknis
Hostname	<i>mastervserver</i>
Komponen pada <i>openstack</i>	<i>nova-compute, nova-network, nova-volume, dan nova-scheduler</i>
Konfigurasi <i>ethernet</i>	<i>eth0 (jaringan lokal)</i>
Alamat <i>Ip</i>	<i>eth0 – 10.10.10.100</i>
Gateway	10.10.10.1
DNS	103.23.100.9

## Pengujian dan pengukuran

Pengujian yang dijalankan pada *virtual machine* terdiri dari beberapa bagian skenario yang dijalankan pada *server virtualisasi*. Skenario pengujian dijalankan pada *native server* dan ketiga tipe *virtualisasi*

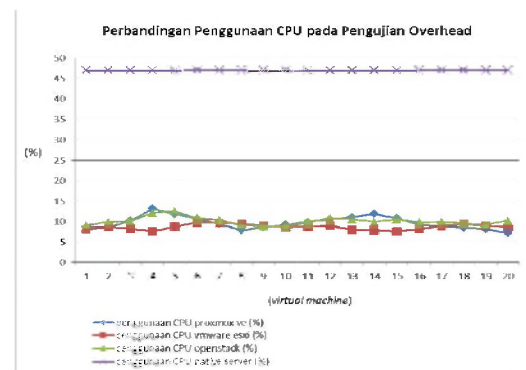
*server* yang di implementasikan pada 3 buah *server* yang identik.

## Hasil Pengujian dan pengukuran

### 1. Pengujian Overhead

Tabel 2. Perbandingan utilitas *CPU* antara *native server* dengan *virtualisasi server* pada pengujian skenario ke tiga

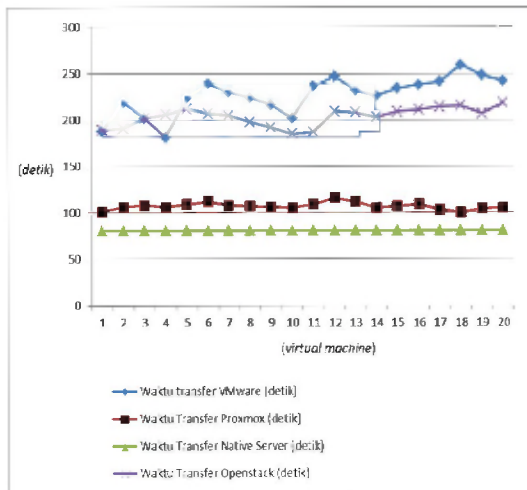
Jumlah <i>virtual machine</i>	Utilitas <i>CPU</i> (%)			
	Native <i>Server</i>	Proxmox VE 1,8	VMware ESXi 5.0	Openstack
1	47	8,6	8,12	9,1
2		8,7	8,77	9,9
3		10,2	8,25	10,1
4		13,2	7,68	12,2
5		11,8	8,78	12,6
6		10,7	9,81	10,8
7		9,5	9,68	10,4
8		8	9,45	9,1
9		8,7	9,01	9
10		9,3	8,61	8,9
11		9,9	8,87	9,8
12		10,6	9,01	10,9
13		11,1	8,01	10,6
14		11,9	7,89	9,9
15		10,8	7,69	10,6
16		9,2	8,21	9,8
17		8,9	8,98	9,9
18		8,6	9,37	9,6
19		8,2	9,03	9,2
20		7,3	8,74	10,3



Gambar 3. Grafik perbandingan utilitas *CPU* antara *native server* dengan *virtualisasi server* pada pengujian skenario ke tiga

Tabel 3. Perbandingan waktu transfer file antara native server dengan virtualisasi server pada pengujian skenario ke tiga

Jumlah virtual machine	Waktu Transfer File (detik)			
	Native Server	Proxmox VE 1,8	VMware ESXi 5.0	Openstack
1	81	101	188	189
2		106	219	191
3		108	202	202
4		106	181	206
5		109	224	212
6		112	239	207
7		108	230	205
8		107	224	198
9		106	216	192
10		105	202	185
11		109	236	187
12		116	247	209
13		112	232	208
14		105	226	203
15		107	234	209
16		109	238	211
17		103	241	214
18		100	259	215
19		104	248	207
20		105	242	218

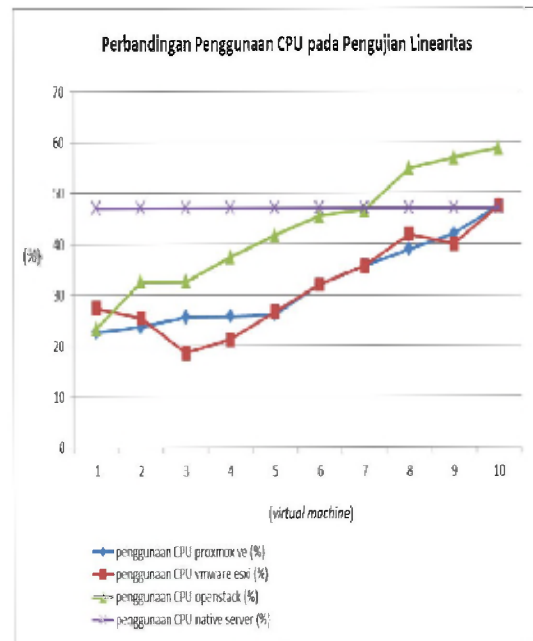


Gambar 4. Grafik perbandingan waktu transfer antara native server dengan virtualisasi server pada pengujian skenario ke tiga

2. Pengujian Linearitas

Tabel 4. Perbandingan utilitas CPU antara native server dengan virtualisasi server pada pengujian skenario ke empat

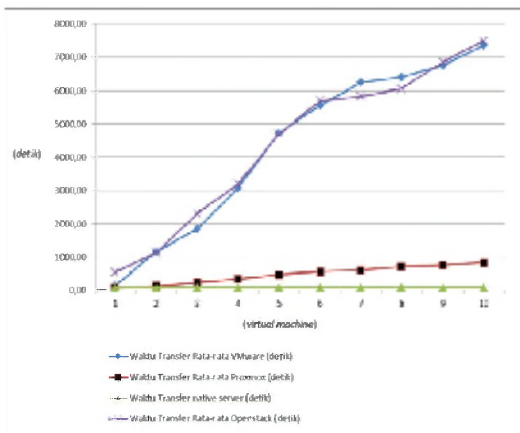
Jumlah Ftp Server	Utilitas CPU (%)			
	Native Server	Proxmox VE 1,8	VMware ESXi 5.0	Openstack
1	47	22,58	27,41	23,34
2		23,73	25,43	32,56
3		25,6	18,57	32,62
4		25,77	21,23	37,38
5		26,11	26,61	41,67
6		31,98	31,98	45,44
7		35,64	35,64	46,63
8		38,86	41,86	54,86
9		41,96	39,96	56,92
10		47,34	47,34	58,74



Gambar 5. Grafik perbandingan utilitas CPU antara native server dengan virtualisasi server pada pengujian skenario ke empat

Tabel 5. Perbandingan waktu transfer file antara *native server* dengan *virtualisasi server* pada pengujian skenario ke empat

Jumlah Server	Waktu Transfer Rata-rata (detik)			
	Native Server	Proxmox VE 1,8	VMware ESXi 5.0	Openstack
1	81	76,00	139,00	535,00
2		133,50	1151,50	1129,00
3		243,00	1840,67	2303,67
4		346,00	3071,00	3194,25
5		474,20	4723,40	4698,00
6		559,50	5556,83	5695,17
7		603,71	6245,57	5832,00
8		708,63	6402,50	6059,50
9		749,56	6762,89	6855,44
10		826,50	7355,10	7506,00



Gambar 6. Grafik perbandingan waktu transfer antara *native server* dengan *virtualisasi server* pada pengujian skenario ke empat

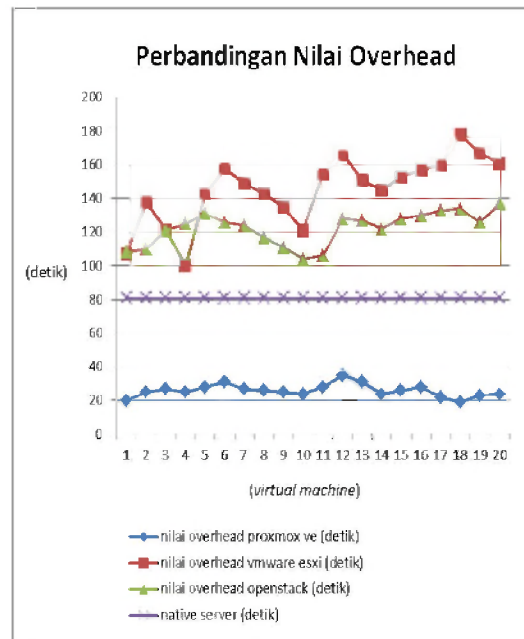
**PEMBAHASAN**

Berdasarkan hasil pengukuran diperoleh nilai-nilai untuk masing-masing server sesuai dengan peran masing-masing :

**Analisa Overhead**

Tabel 6. Perbandingan hasil analisa overhead pada setiap penambahan *virtual machine*

Jumlah virtual machine	overhead proxmox (detik)	overhead vmware (detik)	overhead openstack (detik)	Native server (detik)
1	20	107	108	81
2	25	138	110	81
3	27	121	121	81
4	25	100	125	81
5	28	143	131	81
6	31	158	126	81
7	27	149	124	81
8	26	143	117	81
9	25	135	111	81
10	24	121	104	81
11	28	155	106	81
12	35	166	128	81
13	31	151	127	81
14	24	145	122	81
15	26	153	128	81
16	28	157	130	81
17	22	160	133	81
18	19	178	134	81
19	23	167	126	81
20	24	161	137	81



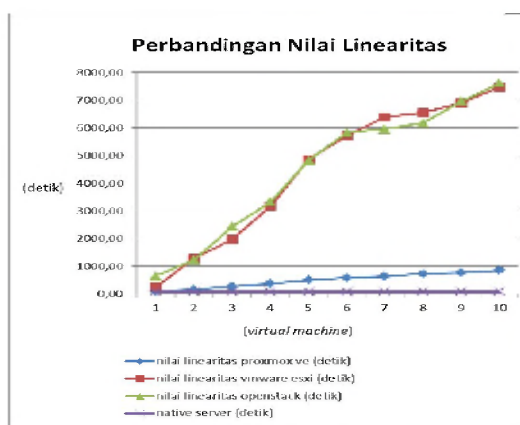
Gambar 7. Grafik perbandingan hasil analisa overhead pada setiap penambahan *virtual machine*

Berdasarkan hasil observasi diatas dapat dilihat bahwa setiap penambahan *virtual machine*, nilai *overhead* relatif stabil pada setiap tipe *virtualisasi* meskipun jika dilihat hasil observasi secara terperinci, terdapat fluktuasi pada beberapa titik ketika menjalankan *virtual machine*. Jika dibandingkan dengan *native server*, nilai *overhead* pada tipe *virtualisasi* menggunakan *vmware* dan *openstack* lebih tinggi daripada *native server*, waktu *overhead* di kisaran lebih dari waktu transfer file pada *native server* sebesar 81 detik. Ketika dilakukan pengukuran *overhead*, utilitas CPU atau penggunaan CPU pada semua tipe *virtualisasi* terlihat lebih hemat dengan nilai lebih rendah daripada utilitas CPU pada *native server* sebesar 47 %.

### Analisa Linearitas

Tabel 7. Perbandingan hasil analisa linearitas pada setiap penambahan *virtual machine*

Jumlah ftp server	linearitas proxmox (detik)	linearitas vmware (detik)	linearitas openstack (detik)	native server (detik)
1	96,00	246,00	643,00	81,00
2	158,50	1289,50	1239,00	81,00
3	270,00	1961,67	2424,67	81,00
4	371,00	3171,00	3319,25	81,00
5	502,20	4866,40	4829,00	81,00
6	590,50	5714,83	5821,17	81,00
7	630,71	6394,57	5956,00	81,00
8	734,63	6545,50	6176,50	81,00
9	774,56	6897,89	6966,44	81,00
10	850,50	7476,10	7610,00	81,00



Gambar 8. Grafik perbandingan hasil analisa linearitas pada setiap penambahan *virtual machine*

Berdasarkan hasil observasi diatas dapat dilihat bahwa setiap penambahan *virtual machine*, nilai linearitas relatif naik pada setiap tipe *virtualisasi* akan tetapi kenaikan secara signifikan hanya terjadi pada tipe *virtualisasi vmware* dan *openstack* ketika mulai diaktifkan ftp server yang kedua dan dikirimkan file sebesar 706.803.376 B pada masing-masing ftp server tersebut. Pada tipe *virtualisasi proxmox* ve kenaikan waktu transfer terjadi secara konstan. Jika dibandingkan dengan *native server*, nilai linearitas pada semua tipe *virtualisasi* lebih tinggi daripada *native server*. Pada pengukuran linearitas yang telah dilakukan, utilitas CPU atau penggunaan CPU pada semua tipe *virtualisasi* terlihat lebih hemat dengan nilai lebih rendah daripada utilitas CPU pada *native server* sebesar 47 %, akan tetapi ketika mulai diaktifkan sebanyak 8 ftp server, utilitas CPU tipe *virtualisasi vmware* meningkat melebihi utilitas CPU pada *native server*, sedangkan pada tipe *virtualisasi proxmox* dan *openstack* ketika mulai diaktifkan sebanyak 10 ftp server, tercatat hampir sama dengan *native server* yang berkisar pada kisaran 47 %. Hasil analisa dari pengukuran tersebut membuktikan bahwa *virtualisasi server* menggunakan *Proxmox Virtual Environment* nilai *overhead* dan linearitas lebih rendah jika dibandingkan *virtualisasi server VMware ESXi* dan *OpenStack*, hal ini karena pada *Proxmox Virtual Environment* menggunakan *virtual machine* dengan *OpenVZ* atau *container-based virtualization* yang hanya dapat menjalankan sistem operasi berbasis linux sehingga pengoperasiannya dapat berbagi kernel antara *host (Proxmox Virtual Environment)* dengan *guest (virtual machine)*. Hal itu menjadikan *virtual machine* dengan *openVZ* berjalan lebih ringan, sedangkan *VMware* dan *openstack* menggunakan *virtual machine KVM (Kernel-based Virtual Machine)* yang dapat menjalankan sistem operasi apapun termasuk Windows sehingga pengoperasiannya tidak dapat berbagi kernel antara *host* dengan *guest (virtual machine)*.

### KESIMPULAN

Dari hasil penelitian yang telah dianalisis, dapat ditarik kesimpulan sebagai berikut:

1. Telah berhasil dirancang dan diimplementasikan virtualisasi server menggunakan tiga tipe virtualisasi server menggunakan 3 buah server identik.



2. Pada perbandingan analisa hasil pengukuran overhead, nilai overhead relatif stabil pada setiap tipe virtualisasi meskipun jika dilihat hasil observasi secara terperinci, terdapat fluktuasi pada beberapa titik ketika menjalankan virtual machine.
3. Grafik overhead pada setiap penambahan virtual machine cenderung bersifat linier dengan sedikit peningkatan.
4. Grafik linearitas pada setiap penambahan virtual machine, nilai linearitas relatif naik pada setiap tipe virtualisasi akan tetapi kenaikan secara signifikan hanya terjadi pada tipe virtualisasi vmware dan openstack ketika mulai diaktifkan ftp server yang kedua dan dikirimkan file sebesar 706.803.376 B pada masing-masing ftp server.
5. Pada tipe *virtualisasi proxmox* ve kenaikan waktu transfer terjadi secara konstan. Jika dibandingkan dengan *native server*, nilai linearitas pada semua tipe *virtualisasi* lebih tinggi daripada *native server*. Hal ini membuktikan bahwa penambahan *virtual machine* yang diberikan suatu aplikasi dan dijalankan bersama-sama dengan aplikasi yang sama, akan menyebabkan penurunan kinerja secara keseluruhan pada masing-masing tipe *virtualisasi server*.
6. Utilitas CPU atau penggunaan CPU pada semua tipe *virtualisasi* ketika dilakukan pengukuran *overhead*, terlihat lebih hemat dengan nilai lebih rendah daripada utilitas CPU pada *native server* sebesar 47 %,
7. Utilitas CPU atau penggunaan CPU pada semua tipe *virtualisasi* ketika dilakukan pengukuran linearitas, terlihat lebih hemat dengan nilai lebih rendah daripada utilitas CPU pada *native server* sebesar 47 %, akan tetapi ketika mulai diaktifkan sebanyak 8 *ftp server*, utilitas CPU tipe *virtualisasi vmware* meningkat melebihi utilitas CPU pada *native server*, sedangkan pada tipe *virtualisasi proxmox* dan *openstack* ketika mulai diaktifkan sebanyak 10 *ftp server*, tercatat hampir sama dengan *native server* yang berkisar pada kisaran 47 %. Hal ini membuktikan bahwa seiring dengan penambahan *virtual machine* yang diberikan aplikasi dan dijalankan secara bersamaan akan menyebabkan semakin

besar utilitas CPU yang digunakan pada masing-masing tipe *virtualisasi*.

8. Hasil analisa dari pengukuran tersebut membuktikan bahwa *virtualisasi server* menggunakan *Proxmox Virtual Environment* nilai *overhead* dan linearitas lebih rendah jika dibandingkan *virtualisasi server VMware ESXi* dan *OpenStack*, hal ini karena pada *Proxmox Virtual Environment* menggunakan *virtual machine* dengan *OpenVZ* atau *container-based virtualization*, sedangkan *VMware* dan *openstack* menggunakan *virtual machine KVM (Kernel-based Virtual Machine)* yang dapat menjalankan sistem operasi apapun termasuk *Windows*.

#### Saran

Saran yang dapat dikembangkan dalam penelitian lebih lanjut antara lain sebagai berikut:

1. Kekurangan dari *virtualisasi server* adalah mengumpulkannya semua *service* pada 1 mesin, sehingga apabila secara fisik mesin tersebut rusak atau error maka akan semua sistem yang berjalan di atasnya akan *fail*. Hal ini dapat diatasi dengan membuat mekanisme *redundant server* atau *fail over server* sebagai cadangan.
2. Teknologi *virtualisasi* dapat lebih dioptimalkan pada organisasi atau perusahaan yang mempunyai anggaran biaya sedikit, dalam pengembangan jaringan *server*.
3. Jika menggunakan dan menjalankan lebih banyak *service* pada *virtual machine*, semakin banyak inti prosesor yang digunakan akan lebih baik dalam kestabilan *virtualisasi server* secara keseluruhan.

#### DAFTAR PUSTAKA

- Ahmad. 2008. Archive for the 'virtualization' Category.  
<http://achmad.glclearningcenter.com/category/teknologi/virtualization/>  
 (diakses tanggal 22 Oktober 2012).
- Benjamin, Q., Vincent, N., Franck, C. 2006. Scalability Comparison of Four Host Virtualization Tools.
- Chen, H., Chen, R., Zhang, F., Zang, B., dan Yew, P. 2006. Live Updating Operating Systems Using Virtualization. Prosiding 2nd international Conference

- on Virtual Execution Environments (VEE'06). Ottawa
- D.H. Brown Associates, Inc. 2001. VMware: Tool for Server Consolidation. <http://www.vmware.com/pdf/vmware-dhbrown.pdf> (diakses pada tanggal 12 November 2011).
- Fauzi, H. 2008. Perbandingan Kinerja Server Melalui Virtualization Xen Pada Lingkungan Terbatas. Universitas Indonesia. MTI UI. Depok
- Garnieri, H, M. 2010. Desain dan Implementasi Virtualisasi Server di PT Thiess Contractors Indonesia. Yogyakarta
- Leung, F., Neiger, G., Rodgers, D., Santoni, A., dan Uhlig, R. 2006, Intel Virtualization Technology: Hardware Support for Efficient Process. <http://www.intel.com/technology/itj/2006/v10i3/> (diakses tanggal 10 Desember 2011).
- Lutfie. 2008. Virtualization The Series. <http://wssid.org/blogs/lutfie/archive/tags/Virtualization/default.aspx>. (diakses tanggal 15 Oktober 2011).
- Muli, B., Michael, F., Eran, R., Avishay, T. 2009. Adding Advanced Storage Controller Functionality via Low-Overhead Virtualization.
- Nggilu, F. 2012. Analisis Overhead Sebagai Salah Satu Faktor Skalabilitas Private Cloud Computing Untuk Layanan IAAS.
- Purbo, O, W. 2011. Proxmox. <http://opensource.telkomspeedy.com/wiki/index.php/Proxmox>. (diakses tanggal 24 September 2011).
- Rasian, Rio., Mursanto, Petrus. 2009. Perbandingan Kinerja Pendekatan Virtualisasi. Jurnal Sistem Informasi MTI-UI Vol 5, No 2. Depok
- Sugianto, Masim. 2010. Panduan Virtualisasi & Linux High Availability Server. Bekasi