 ISSN: 1411-3201

IMPLEMENTASI CLOUD COMPUTING MENGGUNAKAN MODEL INFRASTRUCTURE AS A SERVICE UNTUK OPTIMALISASI LAYANAN DATA CENTER (Studi Kasus : UPT STMIK AMIKOM YOGYAKARTA)

Danang Setiyawan [1] , **Ahmad Ashari** [2] , **Syamsul A Syahdan** [3]

[1,3]*Magister Teknik Informatika STMIK AMIKOM Yogyakarta*
[2]*Magister Ilmu Komputer FMIPA Universitas Gadjah Mada Yogyakarta*
email : masdhanang@gmail.co[1], *ashari@ugm.ac.id*[2], *syamsul@amikom.ac.id*[3]

Abstraksi

Laboratorium komputer pada UPT STMIK AMIKOM Yogyakarta berjumlah banyak dan kapasitas pengguna yang besar maka kebutuhan layanan data center juga semakin tinggi. Untuk memenuhi kebutuhan pengembangan data center tersebut dibutuhkan peningkatan kapasitas komputasi, salah satunya adalah dengan cara pengadaan server baru. Namun terdapat konsekuensi dari keputusan tersebut, organisasi akan menghadapi beberapa masalah baru dalam pengelolaan server yang semakin bertambah yaitu biaya yang dihabiskan untuk keperluan tersebut cukup besar. Biaya yang paling besar adalah pada pembelian dan maintenance server. Pemanfaatan fungsi server juga digunakan untuk mendukung pembelajaran praktikum di laboratorium komputer, jadi selain biaya dan maintenance organisasi juga akan menghadapi permasalahan baru, yaitu utilisasi server yang rendah. Untuk itu cloud computing sebagai solusi yang tepat untuk di implementasikan di laboratorium komputer STMIK AMIKOM Yogyakarta agar layanan data center bisa optimal dari sisi jumlah layanan dan penggunaan sumberdaya server.

***Kata Kunci* :**
Data center, server, maintenance, utilisasi, cloud computing

Pendahuluan

Laboratorium komputer yang dimiliki STMIK AMIKOM Yogyakarta merupakan aset dan fasilitas terbesar dan dikelola oleh bagian UPT dengan jumlah empat belas ruang dengan berbagai spesifikasi perangkat dan platform *Operating System.* Infrastruktur jaringan laboratorium komputer sudah menggunakan perangkat dengan kecepatan gigabit ethernet dan memanfaatkan sistem Data Terpusat (*data center*) yang digunakan untuk menampung tugas atau *project* praktikum yang dikerjakan oleh mahasiswa. Layanan utama yang diberikan oleh *data center* adalah layanan *file server* berbasis web dan *file sharing* yang berjalan pada protokol *Microsoft SMB Protocol*.

Dengan jumlah laboratorium komputer yang banyak serta kapasitas pengguna yang besar maka kebutuhan layanan *data center* juga semakin tinggi. Untuk memenuhi kebutuhan pengembangan *data center* tersebut dibutuhkan peningkatan kapasitas komputasi, salah satunya adalah dengan cara pengadaan server baru. Namun terdapat konsekuensi dari keputusan tersebut, organisasi akan menghadapi beberapa masalah baru dalam pengelolaan server yang semakin bertambah yaitu biaya yang dihabiskan untuk keperluan tersebut cukup besar. Biaya yang paling besar adalah pada pembelian dan *maintenance* server. Selain biaya dan *maintenance* organisasi juga akan menghadapi permasalahan baru, yaitu utilisasi server yang [1].

Permasalahan yang lain adalah pemanfaatan fungsi-fungsi server untuk mendukung pembelajaran praktikum di laboratorium komputer.

Dengan permasalahan yang dihadapi, *cloud computing* menggunakan model *infrastructure as a service* sebagai solusi yang sesuai untuk di implementasikan dengan tujuan untuk optimalisasi layanan *data center* di UPT STMIK AMIKOM Yogyakarta.

Pada penelitian ini model *cloud computing* yang digunakan adalah model IaaS dan infrastruktur *cloud* yang digunakan adalah *private cloud* dan hanya membahas pada fitur aplikasi *cloud computing*.

Tinjauan Pustaka

Berbeda dengan penelitian yang dilakukan oleh Garnier(2010), yaitu melakukan desain dan implementasi virtualisasi untuk infrastuktur server di perusahaan. Tahap awal yang dilakukan oleh peneliti adalah melakukan survei tentang kondisi server dan melakukan analisa kebutuhan infrastruktur server yang sesuai dengan perusahaan yang digunakan sebagai obyek penelitian[2].

 ISSN: 1411-3201

Kesimpulan yang didapatkan dari hasil penelitian tersebut adalah Pertama, alokasi sumberdaya mesin server berupa prosesor dan memori dapat disesuaikan dengan beban kerja sehingga utilisasi perangkat keras mesin server menjadi optimal. Kedua, layanan yang diberikan server tidak mengalami gangguan ketika mesin server mengalami *maintenance*.

Perbedaan dengan penelitian ini yaitu pada tema yang diangkat oleh penulis merupakan pengembangan dari teknologi virtualisasi yaitu *cloud computing*. Tujuan akhir dari penelitian yang dilakukan penulis adalah infrastruktur *data center* dengan *cloud computing* yang *high availability*.

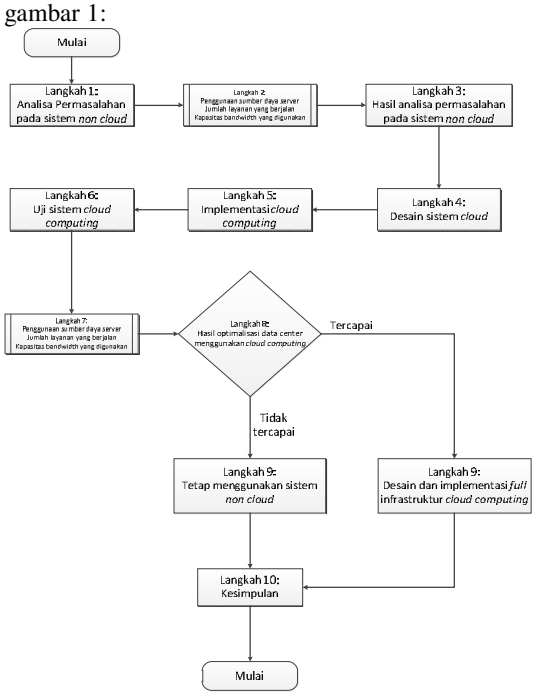
Sedangkan pada penelitian yang dilakukan oleh Januar, Prakasa, & Santiko(2012), penelitian tersebut para peneliti mencoba membuat analisa dan perencanaan *cloud computing* untuk perusahaan yang bergerak dibidang kimiawi (*oil field & industrial chemicals*). Tujuan penelitian tersebut mengambil inti permasalahan pada penghematan biaya, memudahkan operasional dan manajemen perusahaan. Permasalahan lain yang diangkat dalam penelitian tersebut adalah permasalahan teknis yang dihadapi yaitu serangan virus/trojan pada PC (*Personal Computer*) dan meningkatkan jaminan kehandalan dan aksesabilitas system[3]. Yang menjadi perbedaan antara penelitian yang dilakukan oleh para peneliti dengan penulis adalah, Januar (2010) memberikan solusi penggunaan *cloud computing* sebagai tindakan efisiensi biaya untuk membeli kebutuhan sumberdaya komputer (hardware, software) sedangkan tema yang diangkat oleh penulis menekankan optimalisasi perangkat server sebagai solusi karena ketidakseimbangan antara kemampuan sumberdaya server dengan *output* layanan *data center* UPT STMIK AMIKOM Yogyakarta.

Metode Penelitian

Metodologi penelitian yang dilakukan dalam pelaksanaan penelitian ini yaitu :

1. Tahap Studi Literatur
   Mengumpulkan bahan atau materi penelitian, berupa:
   - Studi pustaka dengan mengumpulkan berbagai literatur/referensi yang berhubungan dengan jaringan komputer (*networking*) terutama yang berhubungan dengan *cloud computing*. Referensi yang digunakan tidak hanya didapat dari perpustakaan (buku, jurnal ilmiah) tetapi juga dari internet.
   - Melakukan wawancara dengan pihak UPT STMIK AMIKOM Yogyakarta sebagai pengelola laboratorium komputer untuk mengetahui secara detail sistem yang berjalan dan permasalahan di *data center* UPT STMIK AMIKOM Yogyakarta.
   - Melakukan observasi/pengamatan langsung untuk mengetahui infrastruktur jaringan *data center* UPT STMIK AMIKOM Yogyakarta untuk digunakan sebagai materi analisis.
2. Tahap Analisa, Desain dan Implementasi

Langkah penelitian berikutnya adalah melakukan analisa, desain dan implementasi. Tahap ini dilakukan untuk mendapatkan hasil analisa yang lengkap, desain yang sesuai dan implementasi yang tepat berdasar pada hasil analisa dan desain *cloud computing* sehingga dapat digambarkan pada gambar 1:

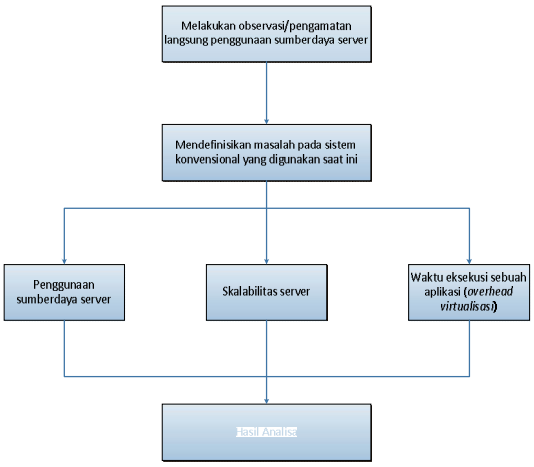


Gambar 1 Tahap analisa, desain dan implementasi

a. Tahap Analisa

Tahapan ini dimaksudkan untuk mengambil, mempelajari dan menganalisis data-data yang diperoleh dari observasi/pengamatan langsung pada perangkat-perangkat server untuk kemudian dilakukan proses perancangan sistem dan desain topologi. Pada tahap ini dilakukan analisis terhadap utilitas server yang digunakan, skalabilitas server dan waktu eksekusi sebuah aplikasi yang dijalankan pada sistem non-virtualisasi dengan aplikasi yang sama dijalankan pada sebuah mesin virtualisasi pada *data center* UPT STMIK AMIKOM Yogyakarta.

 ISSN: 1411-3201

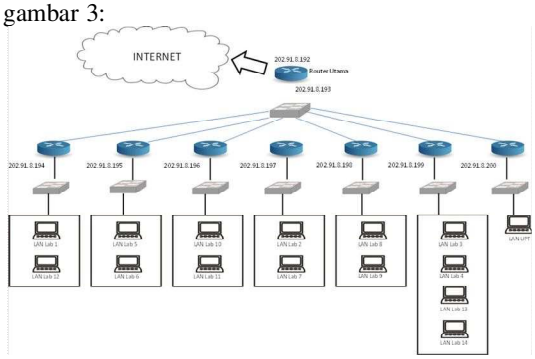


Gambar 2 Langkah-langkah analisis data untuk perancangan sistem *cloud*

Hasil dari tahapan-tahapan analisis yang dijelaskan pada gambar 1 digunakan sebagai dasar pertimbangan implementasi *cloud computing* untuk optimalisasi layanan *data center* di UPT STMIK AMIKOM Yogyakarta. Alur tahap analisa dijelaskan pada gambar 2.

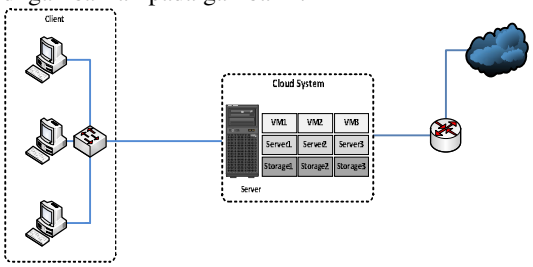
b. Desain Sistem

Desain sistem merupakan desain topologi yang digunakan untuk implementasi *cloud computing*. Desain topologi disesuaikan dengan kebutuhan infrastruktur jaringan *data center* di UPT STMIK AMIKOM Yogyakarta. Kondisi sekarang terdapat delapan server yang memiliki ip address private yang tergabung dalam satu switch. Topologi yang sudah berjalan dapat digambarkan seperti pada gambar 3:



Gambar 3 Topologi yang sudah tersedia

Dari desain topologi yang sudah berjalan diusulkan desain untuk mendukung *cloud computing* sehingga penggunaan jumlah server menjadi lebih sedikit dan jumlah layanan meningkat. Dalam konsep usulan tersebut memanfaatkan sumberdaya server dari yang *dedicated* menjadi *non-dedicated*. Penerapan *cloud computing* menggunakan *proxmox* akan diinstal dalam satu hardware server yang didalamnya memuat virtualisasi server dengan layanan yang berbeda. Dalam paket instalasi server *proxmox* sudah di ikutkan dua aplikasi yang dapat digunakan untuk melakukan virtualisasi yaitu *OpenVZ* dan *KVM*. Untuk menggunakan *KVM* server *proxmox* membutuhkan *motherboard* atau *CPU* yang mendukung teknologi virtualisasi yaitu *Intel VT* atau *AMD-V* dan infrastruktur yang digunakan sudah support teknologi tersebut. Untuk menggunakan *OpenVZ* yang kita perlukan adalah *OpenVZ Template* yang bisa di *download* dari *website OpenVZ*. Perbedaan yang mendasari antara *KVM* dan *OpenVZ* selain dari sisi teknologi adalah dari proses instalasi. Topologi yang diusulkan dapat di gambarkan pada gambar 4:



Gambar 1 Topologi yang diusulkan

c. Implementasi Sistem *Cloud Computing*

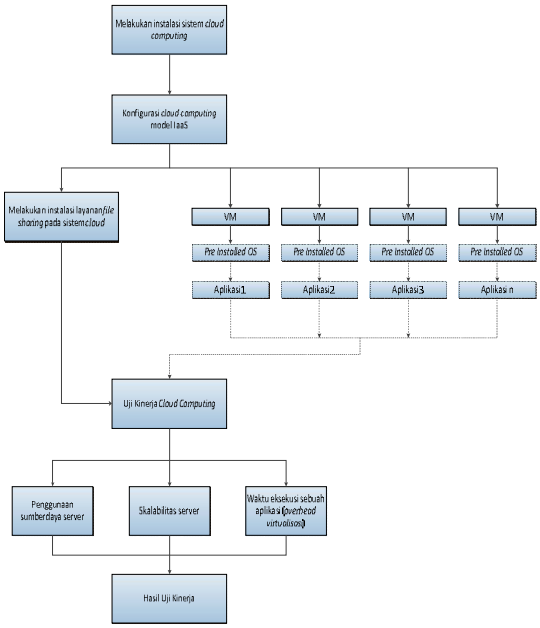
Tahapan ini dimaksudkan untuk implementasi *cloud computing* dari perancangan yang telah dipersiapkan. Implementasi dasar dilakukan dengan melakukan instalasi *file sharing* dan digunakan secara riil sebagai layanan *file sharing* laboratorium komputer STMIK AMIKOM Yogyakarta. Untuk mengukur kinerja layanan ini, server *cloud* diamati pengukuran metrik skalabilitas yaitu menggunakan metode *overhead* dan liniearitas.

Tahap berikutnya adalah konfigurasi model IaaS dengan mempersiapkan *pre installed OS*. Parameter yang digunakan dalam pengukuran tingkat optimalisasi adalah :

- Penggunaan sumberdaya server.
- Skalabilitas server, yaitu jumlah layanan yang mampu didistribusikan oleh sistem *cloud* yang diimplementasikan.
- Waktu eksekusi sebuah aplikasi yang dijalankan pada sistem *cloud.*

Dengan parameter yang digunakan pada simulasi diatas, langkah-langkah simulasi dilakukan skenario sesuai dengan gambar 5:

 ISSN: 1411-3201



Gambar 2 Langkah-langkah implementasi dan pengujian *cloud computing*

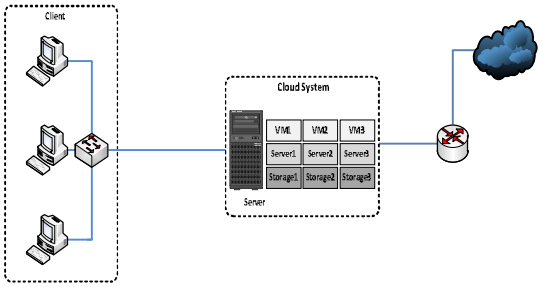
Hasil dan Pembahasan

Pengujian dan Pengukuran

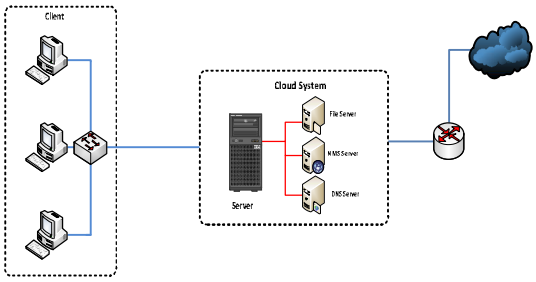
Pengujian yang dijalankan pada sistem *cloud* terdiri dari beberapa bagian skenario yang dijalankan pada server virtualisasi. Skenario pengujian dijalankan pada *native server* dan *cloud server* yang di implementasikan pada server yang identik. Analisa virtualisasi server menggunakan pengukuran metrik skalabilitas yaitu menggunakan metode *overhead* dan liniearitas.

Topologi yang diusulkan

Dari desain topologi yang sudah berjalan diusulkan desain untuk mendukung *cloud computing* sehingga penggunaan jumlah server menjadi lebih sedikit dan jumlah layanan meningkat. Dalam konsep usulan tersebut memanfaatkan sumberdaya server dari yang *dedicated* yang dijelaskan pada menjadi *no-dedicated* yang dijelaskan pada gambar 6 dan gambar 7 beribut:



Gambar 6 Topologi logic yang diusulkan



Gambar 7 Topologi fisik yang diusulkan

Pengujian Server Cloud

1. Analisa Overhead

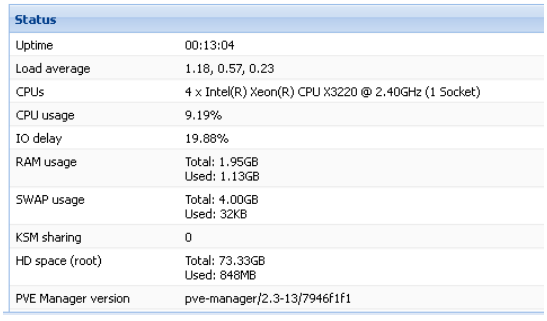
a. Analisa *overhead* pada skenario pengujian pertama

Pada tahap ini pengujian dilakukan perbandingan durasi waktu *upload file* ke *server* menggunakan aplikasi *easy file sharing*. Pada pengujian yang dilakukan dengan skenario pertama didapatkan hasil sesuai dengan tabel 1:

Tabel 1 Evaluasi *overhead* pada pengujian skenario pertama

beban	jumlah proses	*server native*	*cloud server*	
			cloud active	waktu (*sec*)
Upload File 62 MB	1	12.9	1	15.0
			2	13.5
			3	13.2
			4	13.6
			5	13.2
			6	13.8
			7	13.8
			8	14.2
			9	-
			10	-

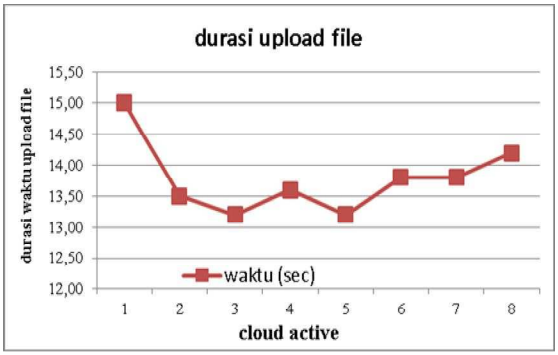
Dari hasil pengujian pada skenario pertama didapatkan hasil bahwa durasi waktu transfer *native server* lebih cepat dibandingkan dengan *cloud server*. Pengamatan lain adalah dilakukan pada server utama yang dijadikan sebagai server *cloud*. Penggunaan utilitas *server* sebesar 9.19% yang dijelaskan pada gambar 8:



Status	
Uptime	00:13:04
Load average	1.18, 0.57, 0.23
CPUs	4 x Intel(R) Xeon(R) CPU X3220 @ 2.40GHz (1 Socket)
CPU usage	9.19%
IO delay	19.88%
RAM usage	Total: 1.95GB Used: 1.13GB
SWAP usage	Total: 4.00GB Used: 32KB
KSM sharing	0
HD space (root)	Total: 73.33GB Used: 848MB
PVE Manager version	pve-manager/2.3-13/7946f1f1

Gambar 8 Utilitas CPU pada server utama

 ISSN: 1411-3201



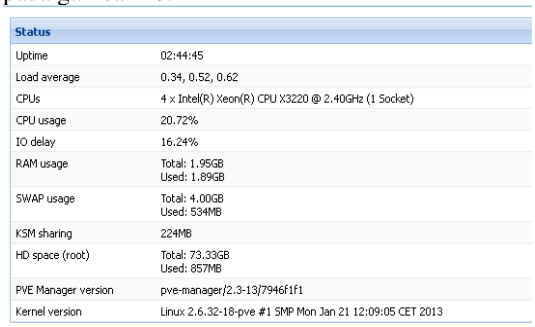
Gambar 9 Grafik Durasi *Upload file* per cloud

b. Analisa *overhead* pada skenario pengujian kedua
Pada tahap ini pengujian dilakukan perbandingan durasi waktu *upload file* ke *server* menggunakan aplikasi *easy file sharing*. Pada pengujian yang dilakukan dengan skenario pertama didapatkan hasil sesuai dengan tabel 2:

Tabel 2 Evaluasi *Overhead* pada pengujian skenario kedua

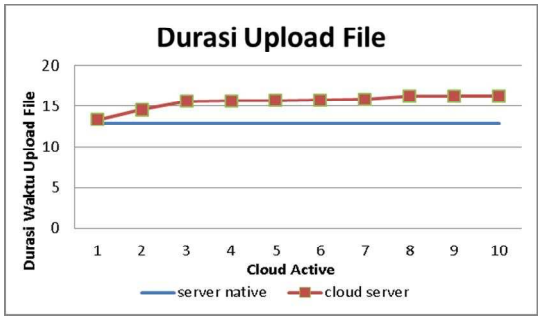
beban	jumlah proses	*server native*	*cloud server*	
			cloud active	waktu (*sec*)
Upload File 62 MB	1	12.9	1	13.29
			2	14.55
			3	15.59
			4	15.62
			5	15.67
			6	15.72
			7	15.81
			8	16.22
			9	16.24
			10	16.24

Dari hasil pengujian pada skenario kedua didapatkan durasi waktu transfer lebih tinggi dibanding *native server*. Pada pengujian kedua juga dilakukan pengamatan pada server utama dan didapatkan penggunaan utilitas *cpu* sebesar 20.72% pada gambar 10:



Status	
Uptime	02:44:45
Load average	0.34, 0.52, 0.62
CPUs	4 x Intel(R) Xeon(R) CPU X3220 @ 2.40GHz (1 Socket)
CPU usage	20.72%
IO delay	16.24%
RAM usage	Total: 1.95GB Used: 1.89GB
SWAP usage	Total: 4.00GB Used: 534MB
KSM sharing	224MB
HD space (root)	Total: 73.33GB Used: 857MB
PVE Manager version	pve-manager/2.3-13/7946f1f1
Kernel version	Linux 2.6.32-18-pve #1 SMP Mon Jan 21 12:09:05 CET 2013

Gambar 10 Utilitas CPU pada server utama



Gambar 11 Durasi Waktu *Upload* pada *Server Cloud*

Dari hasil pengujian menggunakan metode *overhead* dari skenario pertama dan kedua didapatkan hasil perbandingan pada setiap penambahan *virtual machine*.

Tabel 3 Perbandingan durasi transfer file pada pengujian *overhead*

Jumlah *cloud computing*	Skenario pertama (sec)	Skenario kedua (sec)
1	15.0	13.29
2	13.5	14.55
3	13.2	15.59
4	13.6	15.62
5	13.2	15.67
6	13.8	15.72
7	13.8	15.81
8	14.2	16.22
9	-	16.24
10	-	16.24

Dari hasil data diatas, maka dapat dihitung nilai *overhead* virtualisasi:

$T_a = 12.9$

$O_v = T_{av} - T_a$

$= 13.29 - 12.9$ (1)

$= 0.39$ sec

$= 390$ ms

$O_{vn} = T_{anv} - T_a$ n = 10 (2)

$= 16.24 - 12.9$

$= 3.34$ sec

$= 3340$ ms

Maka didapatkan hasil nilai degradasi server ketika menjalankan sepuluh *cloud server* sebesar:

$$\frac{Ov}{Ovn} x100\% = \frac{390}{3340} x\ 100\% = 11.67\% \quad (3)$$

Pada bagian percobaan ini, pada satu waktu baik server tradisional dan server virtual hanya ada satu proses yang berjalan. Namun pada *server cloud* jumlah mesin virtual *non-aplikasi* yang berjalan bersamaan secara bertahap ditingkatkan jumlahnya.

2. Analisa Linearitas

a. Analisa linearitas pada skenario pengujian pertama
Pada pengujian linearitas, jumlah proses mewakili jumlah *client* yang melakukan *upload* ke *server*. Hal ini dilakukan karena aplikasi *easy*

 ISSN: 1411-3201

file sharing tidak dapat di-*install* lebih dari satu pada satu *workstation*. Dari hasil pengujian yang dilakukan didapatkan hasil seperti table 4:

Tabel 4 Perbandingan uji linearitas pada skenario pertama

beban	jumlah *process*	*server native*	*cloud server*	
			cloud active	waktu (*sec*)
Upload File 62 MB	1	12.9	1	13.29
	2	12.9	2	14.55
	3	13.1	3	18.20
	4	13.2	4	18.62
	5	13.5	5	18.67
	6	14.2	6	19.12
	7	14.2	7	19.56
	8	14.6	8	22.22
	9	-	-	-
	10	-	-	-

Pada pengujian linearitas skenario pertama didapatkan hasil durasi waktu transfer pada server *native* tidak mengalami kenaikan yang signifikan jika dibandingkan dengan pengujian *overhead*. Hasil yang berbeda justru didapatkan pada pengujian yang dilakukan pada *server cloud*. Terjadi kenaikan yang signifikan dimulai pada jumlah *client* tiga. Semakin bertambahnya jumlah *server cloud* yang berjalan dan jumlah *client* yang melakukan *upload* membuat durasi waktu *transfer file* juga mengalami peningkatan.

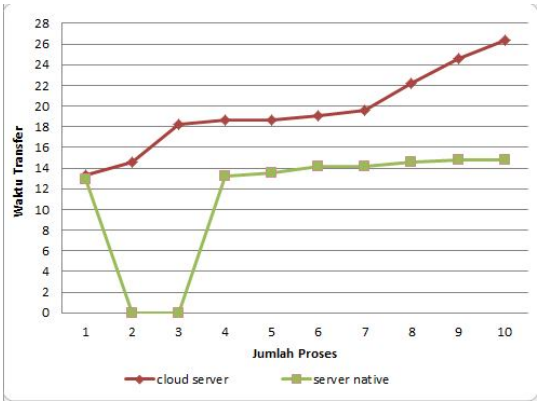
b. Analisa linearitas pada skenario pengujian kedua

Pada pengujian linearitas menggunakan skenario kedua, Dari hasil pengujian yang dilakukan didapatkan hasil seperti tabel 5 :

Tabel 5 Perbandingan uji linearitas pada skenario kedua

beban	jumlah *process*	*server native*	*cloud server*	
			cloud active	waktu (*sec*)
Upload File 62 MB	1	12.9	1	13.29
	2	12.9	2	14.55
	3	13.1	3	18.20
	4	13.2	4	18.62
	5	13.5	5	18.67
	6	14.2	6	19.12
	7	14.2	7	19.56
	8	14.6	8	22.22
	9	14.8	9	24.59
	10	14.8	10	26.38

Pada pengujian linearitas skenario kedua didapatkan hasil durasi waktu transfer yang linier pada server *native*. Hal ini terlihat pada jumlah client sembilan dan sepuluh memiliki durasi waktu *transfer* yang sama Pada pengujian yang dilakukan pada *cloud server*. Grafik dapat diamati pada gambar 12:



Gambar 12 Grafik Pengukuran Linearitas pada *Cloud Server*

Jika virtualisasi adalah konstan (tidak terikat pada jumlah mesin virtual), maka maksimum eksekusi aplikasi harus menjadi fungsi *affine* dari jumlah mesin virtual yang menjalankan aplikasi. Berdasarkan hasil pengukuran diperoleh linearitas :

$n = 10$

$t_{max} = o_v + t \times n$, dimana $o_v = 300$ ms (4)

$t = 13.29$ sec

$= 13290$ ms

Maka $t_{max} = 300 + 13290 \times 10$ (5)

$= 133200$ ms

3. Analisis Data Skalabilitas Server Virtual

Pada penelitian ini secara tipikal dilakukan penghitungan *overhead* virtualisasi untuk satu mesin virtual dibandingkan dengan sistem operasi dasar. Disamping itu juga ditampilkan data yang merepresentasikan skalabillitas sistem meliputi linearitas sistem dan degradasi kinerja ketika beberapa mesin virtual dijalankan dengan beban yang sama. Untuk pengukuran yang telah dilakukan diperoleh nilai *overhead* sebagai berikut :

$T_a = 12.9$ (6)

$O_v = 0.39$ sec

$= 390$ ms

$O_{vn} = 3.34$ sec

$= 3340$ ms

Degradasi kinerja server setelah sepuluh *cloud server* dijalankan : 11,67%.

Berdasarkan hasil observasi di atas dapat dilihat ketika satu virtual mesin dijalankan maka *overhead* virtualisasi setelah dibandingkan dengan eksekusi pada host sistem operasi bisa diabaikan karena nilainya sangat kecil yaitu 390 ms. Jika diamati pada grafik gambar 11 bahwa durasi waktu *transfer server cloud* mendekati linier dengan durasi waktu *transfer* pada *server native*.

 ISSN: 1411-3201

Hasil yang berbeda justru didapatkan ketika pengujian linearitas. Ketika sepuluh *client* melakukan *transfer* bersamaan ke *server cloud*, justru durasi waktu *transfer file* mengalami kenaikan yang signifikan jika dibandingkan dengan pengujian *overhead* dengan satu *client* yang terhubung. Hal ini tampak pada gambar 12 dimana jumlah client yang terhubung diatas enam durasi *transfer file* mulai mengalami kenaikan.

Degradasi kinerja meningkat ketika jumlah mesin virtual diaktifkan bertambah, yaitu mencapai 11,67%. Ketika jumlah mesin virtual yang aktif ditambah, terdapat perbedaan yang cukup jauh pada *server cloud*.

Pengukuran Penggunaan Sumber Daya pada *Server Native* dan *Server Virtual*

Pengukuran parameter dilakukan pada masing-masing *server native* dan *server cloud*. Pengukuran terhadap penggunaan sumber daya pada dua kondisi yaitu :

1. Pada saat server selesai *booting* dan *user-user* sudah login ke jaringan.

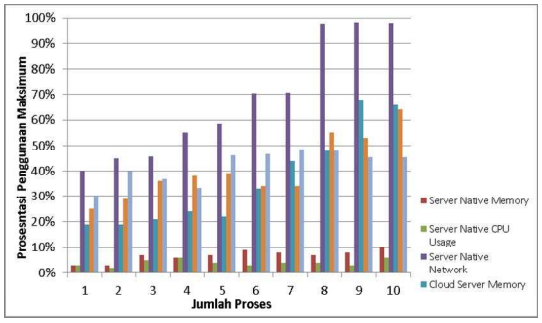
Pengukuran penggunaan sumber daya perangkat keras dilakukan dengan menggunakan aplikasi MRTG yang terinstal terpisah dari *server* yang diuji. Data yang diambil adalah penggunaan maksimum sumber daya dalam melakukan sebuah proses.

Hasil pengukuran yang diperoleh ketika *user-user* telah terhubung ke jaringan namun *server* belum menjalankan program aplikasi dapat dijelaskan pada tabel 6:

Tabel 6 Pengukuran maksimum penggunaan perangkat keras

jumlah process	Server Native			Cloud Server		
	Memory	CPU Usage	Network	Memory	CPU Usage	Network
1	3,00%	3,00%	20,12%	9,00%	20,00%	30,00%
2	3,00%	2,00%	25,05%	9,00%	21,00%	39,80%
3	7,00%	5,00%	25,70%	11,00%	22,00%	36,87%
4	6,00%	6,00%	25,02%	14,00%	26,00%	33,23%
5	7,00%	4,00%	28,56%	12,00%	29,00%	46,28%
6	9,00%	3,00%	50,40%	13,00%	24,00%	46,88%
7	8,00%	4,00%	60,70%	16,00%	24,00%	48,53%
8	7,00%	4,00%	67,83%	19,00%	25,00%	48,10%
9	8,00%	3,00%	68,20%	12,00%	23,00%	45,50%
10	10,00%	6,00%	68,00%	14,00%	24,00%	45,50%

Dari hasil pengamatan yang dilakukan sesuai dengan tabel 6 maka dapat digambarkan grafik sesuai dengan gambar 13:



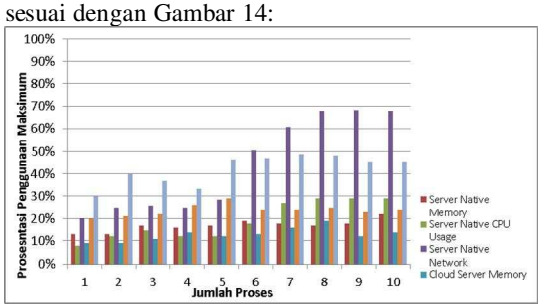
Gambar 13 Penggunaan Maksimum Sumber Daya Perangkat Keras

Untuk pengukuran berikutnya dilakukan *upload file* sebesar 62 MB ke *server* yang dilakukan secara bertahap sebanyak sepuluh kali pengambilan data untuk diambil nilai rata-ratanya. Tabel 7 menunjukkan hasil rata-rata pengambilan data yang telah dilakukan sesuai dengan jumlah *user* yang melakukan akses ke aplikasi pada *server native* dan *server cloud*.

Tabel 7 Pengukuran maksimum penggunaan perangkat keras

jumlah process	Server Native			Cloud Server		
	Memory	CPU Usage	Network	Memory	CPU Usage	Network
1	13,00%	8,00%	40,12%	19,00%	25,00%	33,50%
2	13,00%	12,00%	45,05%	19,00%	29,00%	59,21%
3	17,00%	15,00%	45,70%	21,00%	36,00%	66,26%
4	16,00%	12,00%	55,02%	24,00%	38,00%	73,23%
5	17,00%	12,00%	58,56%	22,00%	39,00%	76,28%
6	19,00%	18,00%	70,40%	33,00%	34,00%	76,88%
7	18,00%	27,00%	70,70%	44,00%	34,00%	78,53%
8	17,00%	29,00%	87,83%	48,00%	55,00%	78,10%
9	18,00%	29,00%	88,20%	68,00%	53,00%	95,50%
10	22,00%	29,00%	88,00%	66,00%	64,00%	95,50%

Dari hasil pengamatan yang dilakukan sesuai dengan tabel 7 maka dapat digambarkan grafik sesuai dengan Gambar 14:



Gambar 14 Penggunaan Maksimum Sumber Daya Perangkat Keras

2. Pada saat *server* telah menjalankan aplikasi dan diakses oleh *user*.

Pembahasan

Berdasarkan hasil observasi diatas dapat dilihat bahwa pada setiap penambahan *server cloud* pada pengujian *overhead* penggunaan sumber daya

 ISSN: 1411-3201

server terlihat stabil walaupun ketika dilihat hasil observasi secara terperinci, terdapat fluktuasi ketika *server cloud* dijalankan. Pada pengujian *overhead*, penggunaan utilitas *cpu* pada *server cloud* terlihat lebih tinggi jika dibandingkan dengan utilitas *cpu* pada *native server*. Nilai tertinggi pada *server cloud* didapatkan pada server NS.

Pada pengujian linearitas skenario pertama dan skenario kedua hasil yang didapatkan tampak tidak ada peningkatan yang signifikan jika dibandingkan dengan pengujian *overhead* di skenario yang sama. Utitilitas *cpu* pada *native server* mendapatkan hasil yang sama dengan pengujian *overhead*.

Pada pengujian waktu eksekusi di masing-masing *server cloud*, durasi waktu *startup* jika dibandingkan dengan *native server* hasil yang didapatkan lebih pendek dengan perbandingan sesuai dengan tabel 8:

Tabel 8 Tabel perbandingan waktu startup pada pengujian overhead dan linearitas

Native Server	*Server cloud*	Durasi waktu *startup* (detik)		Keterangan
		Overhead	linearitas	
169	1	41	42	NS Server
169	2	43	43	MRTG Server
169	3	44	47	easy file sharing
169	4	35	51	easy file sharing
169	5	35	35	easy file sharing
169	6	38	50	easy file sharing
169	7	41	41	easy file sharing
169	8	51	51	easy file sharing
169	9	116	92	easy file sharing
169	10	104	109	easy file sharing

Hasil observasi diatas dapat dilihat bahwa setiap penambahan server cloud menggunakan metode overhead maupun linearitas relatif stabil. Hal ini dikarenakan *proxmox virtual environment* mampu berjalan dengan dua model yaitu *OpenVZ* atau *container base virtualization* dan *KVM* atau *kernel base virtualization* sehingga pengoperasiannya dapat berbagi *kernel* antara server utama (*Proxmox Virtual Environment*) dengan *guest* (*server cloud*).

Dari penjelasan berdasarkan pengukuran di atas dapat dilihat bahwa *server virtual* mampu menggunakan *memory* dan *CPU* lebih optimal. Pada *server virtual* terdapat proses kontrol yang bersifat dinamik terhadap jumlah alokasi *memory*. Sistem akan mengatur alokasi *memory* ke mesin *virtual* secara otomatis bergantung pada beban sistem. Teknologi ini juga menyediakan kontrol dinamik terhadap *execution rate* dan *processor assignment* yang telah dijadwalkan. *Scheduler* ini melakukan *automatic load balancing* pada sistem *multiprocessor*.

Kesimpulan dan Saran

Kesimpulan

Setelah melakukan ujicoba untuk implementasi *cloud computing* menggunakan model *infrastructure as a service* untuk optimalisasi layanan *data center*, beberapa kesimpulan bisa didapat antara lain:

1. Dari pengujian yang dilakukan didapatkan hasil server yang dimiliki oleh UPT STMIK AMIKOM Yogyakarta mampu untuk diimplementasikan sebagai server *cloud computing*. Dengan implementasi ini server lain dapat digunakan untuk aplikasi lain sehingga pelayanan UPT STMIK AMIKOM Yogyakarta dapat ditingkatkan.
2. Pada perbandingan analisa hasil pengukuran *overhead*, nilai *overhead* relatif stabil pada setiap tipe *server cloud*. Jika dilihat secara terperinci, terdapat fluktuasi pada beberapa titik ketika menjalankan *server cloud*.
3. Pada berbandingan analisa hasil pengukuran linearitas, nilai linearitas pada setiap *server cloud* relatif stabil walaupun terdapat fluktuasi di beberapa titik. Kenaikan secara signifikan justru terjadi ketika proses *transfer file* dari komputer *client* ke *server cloud* yang diinstal aplikasi *easy file sharing*.
4. Pada implementasi *Proxmox Virtual Environment*. Administrator dapat dengan leluasa melakukan konfigurasi sumber daya server untuk *virtual server* seperti jumlah *cpu*, jumlah *memory*, kapasitas *harddisk*.
5. Utilitas *cpu* pada semua tipe *server cloud* ketika dilakukan pengukuran linearitas terjadi kenaikan walaupun tidak signifikan. Tetapi jika dibandingkan dengan utilitas *cpu* pada *native server*, semua tipe *server cloud* lebih hemat.
6. Dari data yang didapatkan, durasi waktu *transfer file* pada *server cloud* lebih lambat jika dibandingkan dengan waktu *transfer file* pada *native server*. Hal ini dikarenakan masing-masing *server cloud* harus berbagi sumberdaya dalam melakukan akses input dan output.
7. Dari hasil pengukuran dan analisa didapatkan hasil bahwa optimalisasi layanan data center di UPT STMIK AMIKOM Yogyakarta dapat tercapai. Hal ini dibuktikan dengan satu *server* mampu menjalankan lebih dari satu *virtual server*.

Saran

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

1. Pada penelitian ini dilakukan dengan satu buah *server*, sehingga apabila secara fisik mesin tersebut rusak atau *error* maka akan semua sistem yang berjalan diatasnya akan *fail*. Hal ini dapat diatasi dengan membuat mekanisme *redundant server* atau *fail over server* sebagai cadangan.
2. Jika menggunakan dan menjalankan lebih banyak *service* pada *server cloud*, semakin banyak inti prosesor dan semakin besar

 ISSN: 1411-3201

kapasitas *RAM* yang digunakan akan lebih baik dalam kestabilan *server cloud* secara keseluruhan.

3. Dengan penggunaan *cloud computing* akan lebih efektif jika sistem penyimpanan menggunakan data terpusat. Hal ini bisa dikembangkan dengan teknologi *SAN Storage* maupun *NAS Storage*.
4. Peningkatan layanan *data center* di UPT STMIK AMIKOM Yogyakarta saat ini terbatas pada penambahan layanan *DNS Server* dan *MRTG Server*. Menambah *virtual appliance* pada *server cloud* diperlukan agar penggunaan teknologi *cloud computing* menjadi lebih maksimal.

Daftar Pustaka

[1] Suryono, T., & Afif, M. F., 2013, *Pembuatan Prototype Virtual Server Menggunakan Proxmox Ve Untuk Optimalisasi Resource Hardware Di Noc Fkip Uns. Ijns - Volume 1 Nomor 1 – November 2012, 56.*

[2] Garnier, M., 2010, *Desain Dan Implementasi Virtualisasi Server Di Pt Thiess Contractors Indonesia*, Skripsi tidak terpublikasi.

[3] Januar, J., Prakasa, A., & Santiko, D., 2012, *Analisis Dan Perancangan Cloud Computing Untuk Meningkatkan Kinerja Pt. Rama Kimindo Mulia,* Skripsi tidak terpublikasi.

Biodata Penulis

Danang Setiyawan, memperoleh gelas sarjana pada jurusan Teknik Informatiak STMIK AMIKOM Yogyakarta dan lulus tahun 2009 dan saat ini menjadi mahasiswa Magister Teknik Informatika Pasca Sarjana STMIK AMIKOM Yogyakarta.

Ahmad Ashari, Memperoleh gelar sarjana di Jurusan Fisika FMIPA UGM, Program Studi Elektronika dan Instrumentasi. Lulus tahun 1988, memperoleh gelar master pada Program Studi Ilmu Komputer Program Pascasarjana UI, Bidang pengutamaan Arsitektur Komputer dan Jaringan. Lulus tahun 1992, memperoleh gelar doktor di Vienna University of Technology Austria, bidang Informatik. Lulus tahun 2001 dan saat ini menjadi staf pengajar (dosen) pada Program Studi Elektronika dan Instrumentasi (Elins), jurusan Ilmu Komputer dan Elektronika FMIPA UGM, sejak tahun 1989, Staf pengajar (dosen) pada Program Studi Ilmu Komputer (Ilkom) jurusan Ilmu Komputer dan Elektronika Fakultas MIPA UGM, sejak tahun 1992. Staf pengajar (dosen) pada Program Pasca Sarjana UGM bidang studi Ilmu Komputer dan Fisika, sejak tahun 2002 dan dosen di Magister Teknik Informatika Pasca Sarjana STMIK AMIKOM Yogyakarta.

***Syamsul A Syahdan*,** Memperoleh gelar master pada ilmu komputer FMIPA UGM lulus tahun 2004 dan saat ini menjadi Dosen di Magister Teknik Informatika Pasca Sarjana STMIK AMIKOM Yogyakarta.