

Perancangan dan Analisis Kinerja *Private Cloud Computing* dengan Layanan *Infrastructure-As-A-Service (IAAS)*

Wikranta Arsa*¹, Khabib Mustofa²

¹Program Studi Monodisiplin S2/S3 Ilmu Komputer UGM; Gedung S2/S3 Lt.4 FMIPA UGM Sekip Utara Bulaksumur Yogyakarta 55281, Telp/Fax:(0274)555133 CP:08121570379

²Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta

e-mail: *¹zember_ajuzz@yahoo.co.id, ²khabib@ugm.ac.id

Abstrak

Mesin server merupakan salah satu penunjang dan komponen utama yang harus ada dalam mengembangkan suatu karya ilmiah dengan berbasis web. Mahalnya server menjadi kendala mahasiswa/mahasiswi dalam menghasilkan suatu karya ilmiah. Konfigurasi server yang dapat dilakukan dimana saja dan kapan saja menjadi sebuah keinginan mendasar, selain pemesanan mesin yang mudah, cepat dan fleksibel. Untuk itu diperlukan sebuah sistem yang dapat menangani permasalahan tersebut. Cloud computing dengan layanan Infrastructure-As-A-Service (IAAS) dapat menyediakan sebuah infrastruktur yang handal. Untuk mengetahui kinerja sistem diperlukan suatu analisis performance antara server cloud (instance) dengan server konvensional. Hasil penelitian dari analisis kinerja private cloud computing dengan layanan Infrastructure-As-A-Service (IAAS) ini menunjukkan bahwa perbandingan kinerja satu server cloud atau server virtual cloud dengan satu server konvensional tidak jauh berbeda namun akan terlihat perbedaan kinerja yang signifikan jika dalam satu server node terdapat lebih dari satu server virtual dan sistem ini memberikan tingkat penggunaan resource server yang lebih maksimal.

Kata kunci—Cloud Computing, Infrastructure As-A-Service (IAAS), analisis Performance.

Abstract

Server machine is one of the main components in supporting and developing a web-based scientific work. The high price of the server to be the main obstacle in the student produced a scholarly work. Server configuration that can be done anywhere and anytime to be a fundamental desire, in addition to the booking engine is easy, fast, and flexible is also highly desirable. For that we need a system that can handle these problems. Cloud computing with Infrastructure-As-A-Service (IAAS) can provide a reliable infrastructure. To determine the performance of the system, we required a performance analysis of cloud server between conventional servers. Results of performance analysis of private cloud computing with Infrastructure-As-A-Service (IAAS) indicate that the cloud server performance comparison with conventional server is not too much different and the system resource usage level servers provide more leverage.

Keyword—Cloud Computing, Infrastructure As-A-Service (IAAS), Performance Analysis.

1. PENDAHULUAN

Komputer pada dasarnya adalah sebuah mesin yang digunakan untuk menyelesaikan perhitungan matematis atau komputasi matematis. Perkembangan teknologi menghasilkan beberapa metode komputasi, salah satunya adalah *cloud computing* yang menjadi semakin berkembang akhir-akhir ini.

Mell dan Grance [1], melalui NIST (*National Institute of Standards and Technology*) memberikan pengertian *cloud computing* adalah suatu model komputasi yang memberikan kemudahan, kenyamanan, dan sesuai dengan permintaan (*on-demand access*) untuk mengakses dan mengkonfigurasi sumber daya komputasi (*network, servers, storage, applications, and service*) yang bisa dengan cepat dirilis tanpa adanya banyak interaksi dengan penyedia layanan. NIST menetapkan bahwa terdapat lima karakteristik dari *Cloud*, tiga servis model, dan empat model pengembangan. Disini nampak NIST mengusulkan agar adanya sebuah ketetapan dan standarisasi tentang *cloud computing*. Menurut Tsai dan Lin [2], dalam penelitiannya tentang membangun *enterprise data center* menggunakan teknologi virtual dan *cloud* menyatakan bahwa dengan memanfaatkan *cloud computing* maka akan mendapatkan banyak keuntungan di antaranya adalah pemanfaatan *resource* yang lebih maksimal dan penghematan dalam sumber daya listrik. Selain itu *cloud computing* dapat membantu perusahaan meningkatkan kreasi dan penyampaian dari solusi teknologi informasi, dengan mengizinkan untuk mengakses layanan yang lebih fleksibel dan biaya yang efektif [3].

Salah satu model penyebaran *cloud computing* adalah *Private Cloud*. *Private Cloud* infrastruktur merupakan layanan *cloud* yang dioperasikan hanya untuk sebuah organisasi/perusahaan tertentu. Infrastruktur *cloud* ini biasanya dikelola oleh suatu organisasi atau oleh pihak ketiga. Lokasinya pun bisa *on-site* atau *off-site*. Biasanya organisasi dengan skala besar saja yang mampu memiliki/mengelola *private cloud* ini. Cearley [3], menyatakan bahwa dalam *public cloud computing*, penyedia pihak ketiga memberikan pelayanan kepada pelanggan eksternal sedangkan dalam *private cloud computing* teknologi informasi memberikan layanan kepada pelanggan internal. Tujuan utama penyebaran *private cloud* tidak menjual kapasitas melalui jaringan internet dengan *publicly accessible interfaces* tetapi memberikan pengguna lokal fleksibilitas dan kecepatan dari infrastruktur *private* ini. Ada beberapa fitur yang harus disediakan dalam *private cloud* agar memiliki fitur yang sama dengan fitur pada *public cloud*. Intinya *private cloud* nantinya bisa dikembangkan menjadi *hybrid cloud* [4].

Infrastructure As A Services (IAAS) adalah sebuah layanan yang "menyewakan" sumberdaya teknologi informasi dasar, yang meliputi media penyimpanan, *processing power, memory*, sistem operasi, kapasitas jaringan dan lain-lain, yang dapat digunakan oleh *user* untuk menjalankan aplikasi yang dimilikinya [1]. *Data center* moderen dapat dikelola dengan *IAAS cloud*. Sistem operasi *cloud* menjadi komponen utamanya yang bertanggung jawab untuk mengelola infrastruktur fisik maupun virtual, mengatur penyediaan layanan dan penyebaran, serta menyediakan kemampuan untuk dapat mengakses dan menggunakan sumber daya virtual secara *remote cloud infrastructure* [5].

Dalam mengembangkan ilmu pengetahuan mahasiswa dituntut untuk menghasilkan sebuah karya yang dapat secara nyata diimplementasikan dalam kehidupan. Perkembangan internet dan *web* menuntut mahasiswa / mahasiswi untuk menghasilkan karya yang berbasis *web*. Mesin *server* merupakan salah satu penunjang dan komponen utama yang harus ada dalam mengembangkan suatu karya dengan berbasis *web*. Namun mesin *server* memiliki harga yang relatif mahal sehingga mahasiswa/mahasiswi terkendala dengan biaya pembelian *server*. Hal ini mengakibatkan mahasiswa/mahasiswi enggan melakukan sebuah penelitian yang memerlukan sebuah mesin *server*.

Universitas Gadjah Mada (UGM) memiliki kewajiban untuk memberikan fasilitas kepada mahasiswa/mahasiswi dalam hal ini adalah mesin *server*. UGM sendiri telah menyediakan laboratorium untuk memfasilitasi ini, namun terbatasnya jam kampus serta penggunaan laboratorium menimbulkan kendala lainnya, sehingga jika saja ada sebuah fasilitas

mesin *server* yang dapat diakses kapan saja dan dimanapun tentunya menambah minat mahasiswa untuk menghasilkan sebuah karya yang lebih maksimal tanpa harus menggunakan laboratorium yang tentunya penggunaannya terbatas oleh waktu.

UGM sebenarnya menyediakan *server* virtual yang dapat diakses dimanapun dan kapanpun, hanya saja dalam penyediaan *server* tersebut mahasiswa/mahasiswi harus datang dan bertemu langsung dengan pihak PSDI UGM sebagai pusat pengolahan data dan penyajian informasi. Hal ini dirasa masih sangat tradisional karena untuk mendapatkan sebuah *server* mahasiswa/mahasiswi harus datang dengan surat permohonan peminjaman, bertemu dengan *admin* yang membuat mesin, dan tentunya adanya prosedur-prosedur yang terbilang rumit untuk mendapatkan sebuah *server*.

Sistem yang ada saat ini belum bisa melayani mahasiswa/mahasiswi yang membutuhkan penambahan atau pengurangan kapasitas mesin *server* secara fleksibel dan otomatis. Hal ini diperlukan untuk memberikan porsi penggunaan mesin *server* yang sesuai dengan kebutuhan. Dimana mahasiswa/mahasiswi yang sebelumnya telah memiliki *server* dapat memesan mesin *server* yang lebih besar dari sebelumnya ataupun menghapus mesin sebelumnya untuk digantikan dengan mesin yang baru secara cepat.

Dari permasalahan yang ada peneliti ingin mencoba merancang *private cloud computing* dengan layanan *Infrastructure-As-A-Service (IAAS)* pada Universitas Gadjah Mada (UGM) yaitu menyediakan infrastruktur berupa virtual mesin *server*, serta melakukan analisis kinerja dari infrastruktur tersebut.

Analisis kinerja diperlukan untuk mengetahui sejauh mana kemiripan dan kemampuan kinerja dan performa dari mesin *server* yang dihasilkan oleh *Cloud Computing* dibandingkan dengan mesin *server* asli. Analisis Kinerja Sistem adalah salah satu pokok bahasan penting dalam menjaga suatu sistem itu berjalan sesuai dengan rencana atau tidak. Di zaman teknologi informasi ini, aktivitas analisis ini menjadi sangat penting, di semua lini pengembangan sistem, mulai dari perencanaan, proses pembangunan ataupun pada proses pemeliharaan [6].

2. METODE PENELITIAN

2.1 Asumsi dan batasan masalah

Beberapa asumsi dan batasan masalah digunakan untuk pemodelan permasalahan pada penelitian ini, antara lain:

- a) Perangkat lunak yang digunakan dalam implementasi infrastruktur *Private Cloud Computing* adalah sistem operasi Proxmox VE 2.3, dimana proxmox adalah sistem operasi turunan debian yang memiliki kernel yang dapat menunjang sistem *cloud computing* serta bersifat *open source*.
- b) Analisis uji kinerja yang diteliti meliputi *Turnaround Time*, *Response Time*, *Throughput*, dan *Resource-Utilization*.
- c) *Guest OS instance* yang digunakan berbasis linux.
- d) Proses instalasi perangkat keras maupun perangkat lunak tidak dibahas.
- e) Fokus penelitian pada aspek teknis pengembangan teknologi *Cloud Computing*, bukan pada aspek bisnis.
- f) Aspek keamanan belum dibahas dalam penelitian ini.

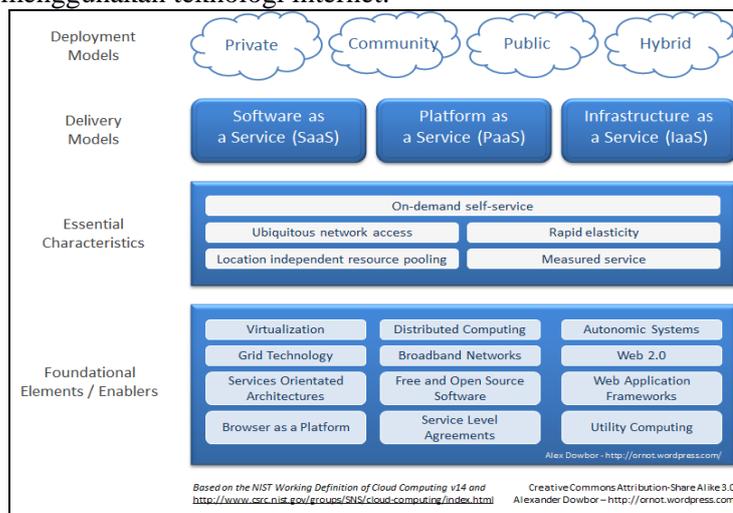
2.2 Dasar Teori

2.2.1 Cloud Computing

Cloud Computing adalah adalah suatu model komputasi yang memberikan kemudahan, kenyamanan, dan sesuai dengan permintaan (*on-demand access*) untuk mengakses dan mengkonfigurasi sumber daya komputasi (*network, servers, storage, applications, and service*) yang bisa dengan cepat dirilis tanpa adanya banyak interaksi dengan penyedia layanan [1].

Selain itu menurut Cearley [3], *cloud computing* adalah gaya dari komputasi yang

terukur dan elastis berkaitan dengan kemampuan IT serta menyediakan layanan kepada pelanggan eksternal menggunakan teknologi internet.



Gambar 1 *Framework Cloud* oleh NIST

Gambar 1 memperlihatkan lima karakteristik utama *Cloud Computing*, empat model *deployment* infrastruktur, tiga model *delivery*. Dalam Gambar 1 juga ditunjukkan bahwa dasar elemen dari teknologi *cloud* adalah virtualisasi, teknologi grid, arsitektur yang berorientasi *service*, komputasi terdistribusi, *broadband networks*, *open source*, dan lainnya.

2.2.2 Virtualisasi

Bittman [7], menjelaskan bahwa virtualisasi harus dilihat sebagai katalisator perubahan IT secara menyeluruh dan modernisasi. Virtualisasi merubah posisi IT menjadi lebih dari penyedia layanan untuk bisnis, membantu IT lebih sebagai penyedia layanan cloud dan bersiap menghadapi bisnis yang lebih baik dari cloud computing. Bittman [7], juga mengatakan virtualisasi sesuai dengan empat dari lima atribut layanan cloud computing yaitu: berbasis servis (service-base), skalabel dan elastis, berbagi (shared), diukur sesuai penggunaan.

Virtualisasi juga merupakan dasar untuk penyedia layanan komputasi awan yang menyediakan layanan infrastruktur, *Infrastructure-As-A-Services* (IAAS). Beberapa penyedia layanan IAAS seperti Amazon, GoGrid, GoDaddy.com dan Terremark Worldwide telah menggunakan mesin-mesin virtual sebagai dasar layanan komputasi awan [8].

Virtualisasi adalah suatu teknik atau cara untuk membuat sesuatu dalam bentuk virtual, tidak seperti kenyataan yang ada. Virtualisasi merupakan suatu aplikasi perangkat lunak untuk menstimulasikan sumber daya perangkat keras. Menurut Sundarajan dan Bhattacharya [9], virtualisasi adalah teknik dimana sumber daya perangkat keras yaitu prosesor, penyimpanan, I/O, dan jaringan pada satu atau lebih mesin dapat diubah melalui partisi *hardware/software*, *time sharing*, dan simulasi/emulasi ke beberapa lingkungan eksekusi, yang masing-masing dapat bertindak sebagai sistem yang lengkap.

2.2.3 Analisis Kinerja

Menurut Meier dkk [10], analisis kinerja adalah berkaitan dengan pencapaian *response time*, *throughput*, dan *resource-utilization level* yang memenuhi tujuan kinerja untuk proyek atau produk. *Response Time* disini mengacu pada seberapa cepat suatu sistem/aplikasi dapat merespon suatu aksi. Misalnya seberapa cepat menampilkan sebuah halaman *web*. *Throughput* menunjukkan banyaknya tugas yang dapat diselesaikan dalam satu satuan waktu. Misalnya *web server* harus bisa melayani minimal 10 pengunjung dalam satu detik. Sedangkan *Resource-utilization* menunjukkan tingkat penggunaan sumber daya perangkat keras tertentu. Misalnya pada saat jam sibuk *web server* tidak boleh memakai CPU *time* lebih dari 20% [11].

a) *Turnaround Time*

Dalam penelitian ini penguji mencatat waktu yang diperlukan oleh sistem *private cloud* untuk menyelesaikan proses kompresi dengan besar data yang akan digunakan sebesar 249 MB mulai dari satu VM sampai sepuluh VM. Kemudian akan dibandingkan dengan waktu proses menyelesaikan data yang sama pada mesin konvensional (*native*), kemudian dicari prosentase pencapaian kinerja dari *Turnaround Time* ini dan juga rata-rata dalam prosentase *Turnaround Time (TAT)* pada *private cloud computing* yang dirancang, dengan persamaan :

$$\text{Pencapaian kinerja TAT} = \frac{TAT_{VM} - TAT_{native}}{TAT_{VMn} - TAT_{native}} \times 100\% \quad (1)$$

$$\text{average TAT cloud} = \frac{\sum TAT_{VM}}{N} \quad (2)$$

Keterangan:

TAT_{vm} : *Turnaround Time* Virtual Mesin

TAT_{vmn} : *Turnaround Time* Virtual Mesin ke-n

TAT_{native} : *Turnaround Time* mesin konvensional / *native*

Untuk mengetahui *Turnaround time* dibutuhkan data berupa waktu yang digunakan sistem untuk memproses sebuah proses. Dalam hal ini penguji menggunakan waktu yang digunakan sistem untuk melakukan proses kompresi. Semakin kecil waktu *turnaround* semakin baik kinerja sebuah *server*.

b) *Response Time*

Response time adalah waktu interval waktu antara perintah input yang siap untuk terminal sistem sampai dengan adanya tanggapan kembali pada terminal. Untuk mendapatkan *response time* peneliti menggunakan waktu yang dicatat pada skenario *request*. Dimana dengan menggunakan *tool* *Httpperf* akan dilakukan 1000 *request* kepada *Virtual Mesin (VM)* pada *cloud* yang tentunya sudah dibuat menjadi sebuah *server*, *request* ini akan dilakukan mulai dari satu buah VM hingga 10 VM yang dilakukan bersamaan. Selain itu *request* juga akan dilakukan kedalam *server* konvensional yang dibuat identik dengan *server VM*, yang fungsinya sebagai pembanding. Waktu ini kemudian dicatat dalam sebuah tabel dan disajikan kedalam sebuah grafik, serta dicari pencapaian kinerja dan rata-rata *response time* pada *private cloud computing* yang dirancang, dengan persamaan :

$$\text{Pencapaian kinerja RPT} = \frac{RPT_{VM} - RPT_{native}}{RPT_{VMn} - RPT_{native}} \times 100\% \quad (3)$$

$$\text{average RPT cloud} = \frac{\sum RPT_{VM}}{N} \times 100\% \quad (4)$$

Keterangan:

RPT_{vm} : *Response Time* Virtual Mesin

RPT_{vmn} : *Response Time* Virtual Mesin ke-n

RPT_{native} : *Response Time* mesin konvensional / *native*

c) *Throughput*

Throughput jumlah atau banyaknya pekerjaan yang dapat dilakukan dalam satuan waktu tertentu. Ekspresi nilainya dengan cara [6]:

1. Jumlah program yang diproses per satuan waktu.
2. Jumlah data yang diproses per satuan waktu.
3. Jumlah transaksi yang diproses per satuan waktu.

Sistem *Throughput* biasanya lebih merupakan nilai teoritis dari pada kemampuan

yang tersedia (*capacity*). *Throughput* mempengaruhi beberapa faktor [6]:

1. Karakteristik workload (beban kerja) yang akan dievaluasi.
2. Karakteristik perangkat keras dan perangkat lunak sistem.
3. Kemungkinan digunakan overlapping untuk banyak komponen.
4. Algoritma yang digunakan.
5. Kecepatan perangkat keras dan perangkat lunak sistem.

Untuk mendapatkan nilai *throughput* diperoleh dari persamaan berikut ini:

$$P = \frac{S}{T} \quad (5)$$

Keterangan:

P : *Throughput*

S : Ukuran file dalam bit

T : Waktu yang digunakan untuk melakukan transfer suatu file

Penelitian ini dilakukan dengan pencatatan yang dilakukan mulai dari satu VM sampai sepuluh VM pada *cloud*, dan pada *server* konvensional. Selanjutnya dari nilai *throughput* ini dicari penurunan kinerja *throughput* dengan membandingkan nilai *throughput* dari VM dengan nilai *throughput* pada *server* konvensional dengan persamaan :

$$P_d = P_{vm} - P_k \quad (6)$$

$$P_{dv} = P_{vmn} - P_k \quad (7)$$

Keterangan:

P_d : Pencapaian kinerja *Throughput*

P_{vm} : *Throughput* Virtual Mesin

P_k : *Throughput* mesin konvensional

P_{dv} : Pencapaian kinerja *Throughput* Virtual mesin

P_{vmn} : *Throughput* virtual mesin ke-n

Dari persamaan (6) dan persamaan (7) didapatkan pencapaian kinerja *Throughput* dengan persamaan :

$$\text{Pencapaian kinerja } \textit{Throughput} = \frac{P_d}{P_{dv}} \times 100\% \quad (8)$$

Persamaan (8) digunakan untuk memperoleh presentase penurunan kinerja pada sistem dengan menjalankan 10 mesin virtual. Semakin besar nilai *throughput* semakin baik kinerja sebuah *server*.

d) *Resource-Utilization*

Menunjukkan tingkat penggunaan sumber daya perangkat keras tertentu seperti penggunaan CPU dan memori. Tujuannya adalah melihat seberapa sibuk *resource* yang dimiliki dalam menjalankan sistem yang berjalan di atasnya. Dalam penelitian ini penulis melihat presentase penggunaan CPU dan memori pada *cloud* mulai dari satu VM yang berjalan dengan OS hingga sepuluh VM dengan OS saja yang berjalan. Selanjutnya dilihat penggunaan CPU dan Memori pada satu VM dengan OS dan proses kompresi yang berjalan sampai dengan sepuluh VM yang direncanakan. Selain itu dilihat juga perbandingannya dengan mesin konvensional (*native*).

2.2.4 *Proxmox*

Proxmox VE adalah sebuah *platform* virtualisasi *open source* yang digunakan untuk menjalankan mesin virtual. Proxmox VE memiliki fitur andalan yaitu *backup*, *restore*, dan *live migration* [12].

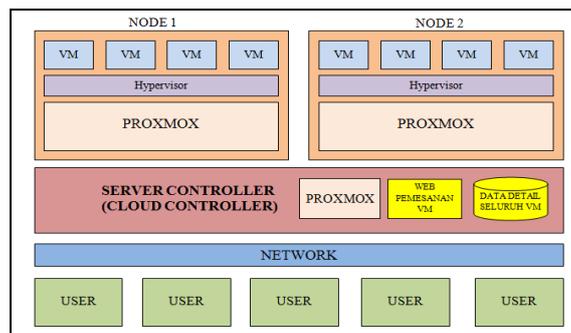
Proxmox VE merupakan sebuah proyek *open source* yang dikembangkan dan dimaintain oleh *Proxmox Server Solution GmbH* di Jerman. Proxmox VE adalah sebuah *platform* virtualisasi *open source* untuk menjalankan *appliance* dan mesin virtual [13].

Proxmox menggunakan kernel Linux dan didasarkan pada distro GNU/Linux Debian. Source code Proxmox VE dirilis berdasarkan GNU Affero General Public License, versi 3 (GNU AGPL,v3). Yang berarti bebas untuk memeriksa *source code* setiap saat dan berkontribusi pada Proxmox [14]. Proxmox VE menggunakan KVM (Kernel-based Virtualization Machine). *Open source hypervisor* KVM adalah salah satu solusi *full virtualization* untuk linux dalam perangkat keras x86 yang berisi ekstensi virtualisasi (intel VT atau AMD-V). Ini adalah modul kernel yang ditambahkan kedalam mainline Linux. Dengan KVM memudahkan dalam menjalankan beberapa virtual mesin dengan *unmodified* Linux atau *images* Windows.

Proxmox VE juga menggunakan OpenVZ Virtualization . OpenVZ adalah container-based virtualization untuk Linux. OpenVZ menciptakan keamanan yang bertingkat, kontainer Linux yang terisolasi (atau lebih dikenal sebagai VE atau VPS) didalam satu *server* fisik yang memungkinkan utilisasi *server* yang lebih dan memastikan bahwa aplikasi tidak konflik.

2.3 Analisis dan Perancangan

Gambar 2 memperlihatkan bahwa *user* berada pada lapisan paling bawah. Lapisan selanjutnya adalah *network* yang menghubungkan antara *user* dengan *server controller*. *Server controller* adalah pengatur dari seluruh *server- server node* yang terhubung. Pada *controller* ini terdapat *web* pemesanan VM yang menjadi *interface* untuk *user* yang akan melakukan pemesanan dan pembuatan mesin *server*. *Controller* juga menyimpan seluruh data detail VM yang telah dibuat *user*. Fungsinya adalah menghindari adanya duplikasi VM. Selanjutnya pada lapisan terakhir adalah *node*. *Node* adalah sebuah *server* fisik yang saling terhubung dimana disinilah nantinya tersimpan *server* yang dimiliki oleh masing-masing *user*. Dalam *node* diinstallkan sistem operasi Proxmox. Sistem operasi ini memiliki *hypervisor* yang mengatur VM yang telah dibuat.



Gambar 2 Rancangan Arsitektur Private Cloud Computing

User yang selanjutnya bertindak sebagai *client* melakukan pendaftaran melalui sebuah *web* yang telah disiapkan. Pendaftaran ini sebenarnya adalah *web* aplikasi pendaftaran biasa. *Web* ini ditanam pada *server controller* sesuai dengan Gambar 2. Fungsi utama dari *web* ini adalah mempermudah *admin* untuk melakukan pembuatan *server*. *Admin* hanya mengecek permintaan dan dengan satu tombol, mesin akan terbuat secara otomatis. Selanjutnya setelah mesin terbuat maka disinilah peranan dari *cloud* ini. Mesin yang telah dibuat dapat diakses menggunakan SSH oleh *user*. *User* secara bebas diberikan mengkonfigurasi dan mengkasess *server* yang dimilikinya sesuai dengan permintaan mesin yang dilakukan pada *web* pendaftaran sebelumnya. *User* dapat mengakses dimanapun dan kapanpun asalkan terhubung dengan

internet. Sementara *admin* dapat memantau dan melakukan pembuatan mesin dimanapun. *Admin* juga dapat melihat aktifitas seluruh *server node* dan *VM* dengan bantuan proxmox.

3. HASIL DAN PEMBAHASAN

3.1 Hasil

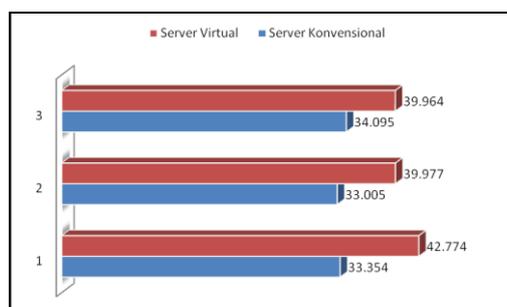
Implementasi sistem dalam penelitian menghasilkan sebuah sistem untuk memenuhi kebutuhan sebuah mesin *server*, dalam hal ini sebuah mesin virtual. Sistem ini diharapkan memberikan kemudahan kepada *user* dan *admin* dalam pengelolaan *server*. *User* dapat secara langsung memesan sebuah *server* dan mengkonfigurasinya dan *admin* dapat melakukan seleksi dengan mudah untuk menerima permintaan *server*. Dalam pembuatan mesin *admin* juga dipermudah dengan sistem ini, dimana *admin* hanya melakukan penerimaan mesin dan secara otomatis mesin terbuat.

Percobaan dilakukan dengan membandingkan sebuah *server* virtual yang berjalan dalam sistem ini dengan sebuah *server* fisik. Spesifikasi *server* dibuat identik dengan sistem operasi linux ubuntu 12.04, selanjutnya dicari perbandingannya dengan empat parameter yaitu *turnaround time*, *response time*, *throughput*, dan *resource-utilization*. Sebelum melakukan percobaan dilakukan pembuatan 10 mesin virtual mesin kedalam sistem dan diletakkan dalam satu *node* yaitu zember1

Tabel 1 adalah hasil dari salah satu pengujian yaitu kompresi. Dalam Tabel 1 terlihat tiga kali pengujian. Dimana hasil pengujian pada Tabel 1 akan dijadikan sumber data dalam penghitungan penurunan kinerja mesin *cloud*.

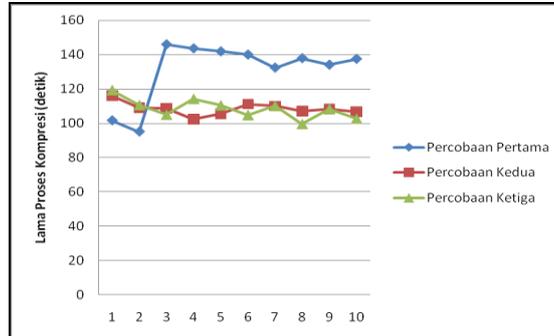
Tabel 1 Hasil Pengukuran Proses Kompresi

SKENARIO	Lama Proses Pada Server Native Percobaan 1 (s)	Lama Proses Pada Server Native Percobaan 2 (s)	Lama Proses Pada Server Native Percobaan 3 (s)	SERVER INSTANCE				
				VM (s)	Lama Proses Percobaan 1 (s)	Lama Proses Percobaan 2 (s)	Lama Proses Percobaan 3 (s)	Rata-rata (s)
Proses Kompresi 249 MB	33.354	33.005	34.095	1	42.774	39.977	39.964	40,91
Proses Kompresi 249 MB kepada seluruh VM hampir bersamaan				1	101.561	116.082	119.321	112.32
				2	95.015	108.921	110.675	104.87
				3	145.975	108.769	105.093	119.95
				4	143.644	102.409	114.264	120.11
				5	141.859	105.693	110.495	119.35
				6	139.988	111.195	104.962	118.72
				7	132.393	110.045	110.224	117.55
				8	137.920	107.205	99.702	114.94
				9	134.079	108.370	108.423	116.96
				10	137.513	106.672	102.846	115.68



Gambar 3 Grafik Perbedaan Lama Proses Kompresi Mesin Konvensional Dengan Satu Mesin Virtual

Gambar 3 memperlihatkan grafik perbedaan dari mesin virtual dengan mesin konvensional. Terlihat mesin konvensional memiliki catatan waktu yang lebih baik dari pada *server* virtual. Hal ini wajar terjadi karena mesin virtual tentunya memiliki proses yang lebih panjang untuk sampai pada CPU.



Gambar 4 Grafik Lama Proses Kompresi Pada 10 Mesin Virtual

Gambar 4 memperlihatkan lama proses yang dibutuhkan untuk melakukan proses kompresi. Dari gambar terlihat bahwa dari tiga kali percobaan cenderung memiliki bentuk grafik yang hampir sama dimana mesin kedua memiliki catatan waktu yang lebih baik dari pada mesin yang lainnya dan cenderung ketika mesin diatas 3 memiliki catatan waktu yang tidak jauh berbeda atau lama proses kompresi cenderung stabil.

3.2 Pembahasan

Pada penelitian ini telah dilakukan pengujian dengan melakukan proses kompresi dan *request page* dengan menggunakan *httperf*. Dari pengukuran didapatkan data-data yang selanjutnya menjadi sumber data dalam perhitungan pencapaian kinerja mesin virtual dengan hasil sebagai berikut:

Dari persamaan (1) dan (2), maka diperoleh pencapaian kinerja *turnaroud time* dan rata-rata dari *turnaroud time* mesin virtual dalam sistem sebagai berikut:

Turnaroud Time Virtual Mesin : $TAT_{vm} = 40,91$

Turnaroud Time Virtual Mesin ke-10 : $TAT_{vmn} = 115,68$

Turnaroud Time mesin konvensional / *native* : $TAT_{native} = 33,485$ (rata-rata)

Dengan persamaan (1) didapatkan nilai pencapaian kinerja Turnaroud Time (TAT)

$$\text{pencapaian kinerja TAT} = \frac{TAT_{VM} - TAT_{native}}{TAT_{VMn} - TAT_{native}} \times 100\%$$

$$\text{pencapaian kinerja TAT} = \frac{40,91 - 33,485}{115,68 - 33,485} \times 100\%$$

$$\text{pencapaian kinerja TAT} = 9,03 \%$$

Dengan persamaan (2) didapat nilai rata-rata Turnaroud Time (TAT)

$$\text{average TAT cloud} = \frac{\sum TAT_{VM}}{N}$$

$$\text{average TAT cloud} = 1160,44 : 10 = 116,04 \text{ detik}$$

Dengan persamaan yang sama dilihat Pencapaian TAT saat 5 VM saja yang melakukan kompresi dan 5 VM lainnya dalam keadaan idle.

Turnaroud Time Virtual Mesin : $TAT_{vm} = 40,91$

Turnaroud Time Virtual Mesin ke-5 : $TAT_{vmn} = 56,327$

Turnaroud Time mesin konvensional / *native* : $TAT_{native} = 33,485$ (rata-rata)

Dengan persamaan (1) didapatkan nilai pencapaian kinerja Turnaroud Time (TAT)

$$\text{Pencapaian kinerja TAT} = \frac{TAT_{VM} - TAT_{native}}{TAT_{VMn} - TAT_{native}} \times 100\%$$

$$\text{Pencapaian kinerja TAT} = \frac{40,91 - 33,485}{56,327 - 33,485} \times 100\%$$

$$\text{Pencapaian kinerja TAT} = 32,506\%$$

Dari persamaan (3) dan (4) pada bab 4, maka diperoleh pencapaian kinerja *response time* dan rata-rata dari *response time* mesin virtual dalam sistem sebagai berikut:

Response Time Virtual Mesin : $RPT_{vm} = 0,4$

Response Time Virtual Mesin ke-10 : $RPT_{vmn} = 1,47$

Response Time mesin konvensional / *native* : $RPT_{native} = 0,3$

Dengan persamaan (3) didapatkan nilai pencapaian kinerja *response time* (RPT)

$$\text{pencapaian kinerja RPT} = \frac{RPT_{VM} - RPT_{native}}{RPT_{VMn} - RPT_{native}} \times 100\%$$

$$\text{pencapaian kinerja RPT} = \frac{0,4 - 0,3}{1,47 - 0,3} \times 100\%$$

$$\text{pencapaian kinerja RPT} = 8,55 \%$$

Dengan persamaan (4) didapat nilai rata-rata *response time* (RPT)

$$\frac{\sum RPT_{VM}}{N}$$

$$\text{average RPT cloud} = \frac{\sum RPT_{VM}}{N}$$

$$\text{average RPT cloud} = 14,87 : 10 = 1,487 \text{ ms}$$

Dari persamaan (6), (7), dan (8) pada bab 4, maka diperoleh nilai pencapaian kinerja *throughput* seperti berikut:

Throughput Virtual Mesin : $P_{vm} = 1760 \text{ KB/s}$

Throughput mesin konvensional : $P_k = 1764,67 \text{ KB/s}$

Throughput virtual mesin ke-10 : $P_{vm10} = 418,67 \text{ KB/s}$

Dari persamaan (6) dan (7) diperoleh :

$$P_d = P_{vm} - P_k \\ = 1760 - 1764,67 = -4,67$$

$$P_{dv} = P_{vmn} - P_k \\ = 418,67 - 1764,67 = -1346$$

Dengan persamaan (8) maka diperoleh presentase pencapaian kinerja *throughput*:

$$\text{Pencapaian kinerja throughput} = \frac{P_d}{P_{dv}} \times 100\% = \frac{-4,67}{-1346} \times 100\% = 0,35\%$$

Dari observasi diatas dapat dilihat pencapaian kinerja *Turnaroud time*, *response time*, dan *throughput* dari satu mesin virtual dengan mesin konvensional memiliki perbedaan yang tidak terlalu jauh, namun ketika terjadi peningkatan mesin virtual pada satu *node* maka terjadi pencapaian kinerja *turnaround time* mencapai 9,03 %, *response time* sebesar 8,55 %, dan *throughput* sebesar 0,35 %. Perbedaan *Turnaroud time*, *response time*, dan *throughput* sebesar 7,425 detik, 0,1 milidetik, dan 4,67 KB/s (diperoleh dari pengurangan nilai *turnaroud time*, *respon time*, dan *throughput* mesin virtual dengan mesin konvensional).

Penggunaan *resource* mesin konvensional sebelum dan sesudah melakukan proses kompresi terjadi peningkatan CPU dari 1,3 % menjadi 50% dan penggunaan memori dari 366 MB menjadi 373 MB terjadi kenaikan penggunaan memori sebesar 7 MB. Sedangkan mesin virtual menggunakan CPU dari 0,7% menjadi 100% dan memori 62 MB menjadi 69 MB terjadi peningkatan penggunaan memori sebesar 7 MB. Dari sisi *server* fisik *node* penggunaan *resource* dari satu mesin yang berjalan diatasnya menggunakan CPU dari 1,3% menjadi 100% pada *core* ke empat. Sedangkan memori dari 376 MB menjadi 384 MB terjadi peningkatan penggunaan memori sebesar 8 MB.

Penggunaan *resource* pada *node* dengan satu virtual mesin yang aktif sampai sepuluh virtual mesin yang aktif dalam keadaan *idle*, penggunaan CPU 1% menjadi 2% sedangkan memori dari 376 MB menjadi 468 MB terjadi peningkatan penggunaan memori sebesar 92 MB.

Sedangkan ketika proses kompresi dijalankan secara bersamaan ke sepuluh virtual mesin maka penggunaan CPU dari 2% menjadi 100% pada seluruh *core* dan memori dari 468 MB menjadi 646 MB terjadi kenaikan penggunaan memori sebesar 178 MB. Sedangkan penggunaan CPU pada sepuluh mesin dalam keadaan *idle* dalam *node* 0% meningkat ketika proses kompresi secara bersamaan sebesar 50% dan penggunaan memori dari 60 MB menjadi 65 MB terjadi peningkatan kurang lebih sebanyak 5 MB.

Nilai *resource* ini didapatkan dari *tool htop*, sehingga keuntungan utama yang ditawarkan oleh penggunaan teknologi *virtual* adalah menjanjikan infrastruktur yang dapat diandalkan dan memungkinkan penggunaan *resource* yang maksimal dari sebuah *server*.

4. KESIMPULAN

Kesimpulan yang diperoleh dalam penelitian perancangan dan analisis kinerja *private cloud computing* dengan layanan *infrastructure as a service* (IAAS) yang diimplementasikan pada laboratorium S2/S3 Ilmu Komputer Universitas Gadjah Mada adalah:

1. Telah berhasil dibangun sebuah sistem *private cloud computing* dengan layanan *Infrastructure as a service* (IAAS) dalam menyediakan sebuah mesin *server*.
2. Telah berhasil menyediakan sistem yang dapat melayani pembuatan mesin *server*, dan memungkinkan melakukan penambahan atau pengurangan kapasitas mesin secara fleksibel dan otomatis.
3. Pada pengukuran performa virtual mesin *server cloud* dan mesin konvensional yang arsitektur jaringannya sesuai rancangan pengujian didapatkan presentase pencapaian kinerja *turnaroud time*, *response time*, dan *throughput* sebesar 9,03%, 8,55%, dan 0,35% dengan jumlah mesin virtual yang dibuat dalam satu *node* sebanyak sepuluh.
4. Peningkatan waktu *turnaroud* sebesar 7,425 detik dan waktu *response* sebesar 0,1 milidetik, sedangkan terjadi penurunan nilai *troughput* sebesar 4,67 KB/s, nilai didapat dari hasil perbandingan satu buah mesin *server* virtual pada sistem *cloud* dengan satu buah mesin *server* konvensional.
5. Dari hasil pada poin tiga dan empat dapat dilihat *server* konvensional lebih baik dari *server cloud*, namun perbedaan *turnaroud time*, *response time*, dan *throughput server cloud* tidaklah terlalu jauh berbeda dari *server* konvensional jika dalam satu *node* dibuatkan satu *server* virtual, namun akan berbeda jauh jika dalam satu *node* dibuatkan lebih dari satu *virtual* mesin (dapat dilihat pada kesimpulan poin 3) dimana pencapaian kinerja mesin menjadi sangat kecil. Sehingga *server cloud* dirasa dapat dipertimbangkan untuk diimplementasikan jika memiliki *resource node* yang memadai, hal ini dapat dilihat dari hasil pencapaian kinerja *turnaround time* ketika lima mesin virtual yang melakukan kompresi pada satu *node* meningkat mencapai 32, 506%
6. Dengan sistem *cloud* menggunakan *proxmox* telah berhasil mengoptimalkan penggunaan *resource* sebuah mesin *server* hingga penggunaan CPU mencapai 100% pada seluruh *core* dan penggunaan memori meningkat sebanyak 178 MB ketika proses kompresi pada sepuluh virtual mesin yang berjalan. Penggunaan sistem *cloud* dengan *proxmox cluster* ini memberikan kemudahan dalam hal penambahan infrastruktur, cukup dengan menginstal mesin baru dengan sistem operasi *proxmox* kemudian letakkan pada satu jaringan dan tinggal ditambahkan pada *cluster* yang tersedia maka mesin siap untuk digunakan. Hal ini dapat dijalankan tanpa harus mematikan *server* yang lainnya.

5. SARAN

Adapun saran-saran terhadap penelitian perancangan dan analisis kinerja *private cloud computing* dengan layanan *infrastructure as a service* (IAAS) agar kedepannya penelitian ini lebih baik adalah :

1. Penggunaan sistem operasi pada *instance* dapat dikembangkan menjadi *multi operating sistem* tidak hanya terbatas dalam satu basis sistem operasi saja.

2. Membuat sistem ini terintegrasi dengan *database* yang dimiliki UGM sehingga nantinya seluruh data *user* mengambil dari *database* UGM.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada semua pihak laboratorium komputasi S2/S3 Ilmu Komputer UGM yang telah memberikan fasilitas berupa peminjaman server sehingga penelitian ini bisa selesai sesuai dengan rencana.

DAFTAR PUSTAKA

- [1] Mell, P. dan Grance, T., 2011, The NIST Definition of Cloud Computing, *NIST Special Publication 800-145*.
- [2] Tsai, D.R. dan Lin, S.W., 2010, Building Enterprise Data Centers Using Virtual and Cloud, *IEEE*, 716–718.
- [3] Cearley, D.W., 2010, Cloud Computing Key Initiative Overview, *Gartner, Inc*, Stamford.
- [4] Sotomayor, B., Montero R.S., Llorente, I.M. dan Foster, I., 2009, Virtual Infrastructure Management in Private and Hybrid Clouds, *IEEE internet Computing*, 14-22.
- [5] Vozmediano, R.M, Montero R.S. dan Llorente, I.M., 2011, IAAS Cloud Architecture: From Virtualized DataCenters to Federated Cloud Infrastructures, *IEEE*.
- [6] Iqbal, M., 2011, *Analisis Kinerja Sistem Pendekatan Teori dan Praktek*, Gunadarma, Depok.
- [7] Bittman, T.J., 2009, Server Virtualization: One Path That Leads to Cloud Computing, *Gartner RAS Core Research Note G00171730*.
- [8] Santoso, B.I., 2012, Perkembangan Virtualisasi, *CloudIndonesia Komunitas Cloud Computing Indonesia*.
- [9] Sundarrajan, S. dan Bhattacharya, S., 2006, Xen and Server Consolidation, *Infosys Building Tomorrow's Enterprise*, Bangalore.
- [10] Meier, J.D., Farre, C., Bansode, P., Barber, S. dan Rea, D., 2007, Performance Testing Guidance for Web Applications, *Microsoft Patterns and Practice 2007*.
- [11] Rasian, R. dan Mursanto, P., 2009, Perbandingan Kinerja Pendekatan Virtualisasi, *Journal of Information System*, Vol 5(2), 90-99.
- [12] Pehrson, B., Jonsson, S., dan Nungu, A., 2010, Proxmox Virtual Environment, *CareNet*.
- [13] Purbo, O.W., 2012, *Membuat Sendiri Cloud Computing Server Menggunakan Open Source*, Penerbit Andi, Jakarta.
- [14] Anonim, 2013, Proxmox, <http://www.proxmox.com/proxmox-ve/features>, diakses 9 Juli 2013.