

Analisis *Overhead Server Cloud Infrastructure* pada *Proxmox VE Hypervisor*

I Gusti Ngurah Wikranta Arsa
Sistem Komputer, STMIK STIKOM Bali
Jln. Raya Puputan No.86 Renon Denpasar
arsa@stikom-bali.ac.id

Abstrak

Cloud infrastructure merupakan layanan *cloud* yang terfokus menyediakan layanan berupa infrastruktur teknologi informasi, seperti menyediakan sebuah *server* yang dapat dikonfigurasi dan digunakan sesuai dengan keinginan pengguna. NIST (*National Institute of Standards and Technology*) menjelaskan bahwa salah satu karakteristik dari *cloud computing* adalah *resource pooling*, dimana sumber daya IT terpusat disuatu *pool*. Untuk dapat menyediakan sebuah infrastruktur *cloud* maka dibutuhkan beberapa *server* yang handal. Permasalahan yang dihadapi penyedia layanan *cloud* infrastruktur adalah berapa besar *overhead* yang terjadi sebelum dan sesudah *server* konvensional dikonfigurasi menjadi *server* penyedia layanan *cloud* infrastruktur. Untuk mengetahuinya maka harus dilakukan pengujian dan analisis *server*. Metode penelitian dimulai dari investigasi sistem *cloud*, rancangan sistem *cloud* infrastruktur, rancangan pengujian sistem, implementasi sistem *cloud* infrastruktur, dan pengujian sistem. Hasil penelitian ini adalah sebuah sistem *cloud infrastructure* dengan menggunakan *proxmox ve* sebagai *hypervisor*nya dan perbandingan *overhead server cloud*.

Kata kunci: *cloud infrastructure, proxmox ve, overhead*

1. Pendahuluan

Cloud Computing adalah sebuah model komputasi yang mengutamakan elastisitas dan layanan yang terukur menggunakan teknologi internet sebagai basis layanannya. Menurut [1] dalam publikasinya di 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, server s, storage, applications, and service*) yang bisa dengan cepat dirilis tanpa adanya banyak interaksi dengan penyedia layanan. Sedangkan [2] menyatakan *cloud computing* adalah gaya dari komputasi yang terukur dan elastis berkaitan dengan kemampuan IT serta menyediakan layanan kepada pelanggan eksternal menggunakan teknologi internet.

Cloud Computing memiliki tiga model layanan, yaitu *Software as a Service (SaaS)*, *Platform as a Service (PaaS)*, dan *Infrastructure as a Service (IaaS)*. *IaaS* adalah salah satu model layanan yang menyediakan infrastruktur teknologi Informasi. Dalam pengembangannya, penyedia layanan harus memikirkan infrastruktur fisik yang handal. *Server* adalah salah satu komponen utama yang harus disediakan oleh sisi penyedia *IaaS*. Permasalahan yang dihadapi penyedia *IaaS* adalah menentukan berapa jumlah minimal *server* yang harus disediakan ataupun spesifikasi *server* minimal yang harus disediakan. Untuk mengetahuinya, dibutuhkan sebuah analisis *server* dengan skenario layanan infrastruktur yang ditentukan dan dimaksimalkan dalam sebuah *server* fisik.

Salah satu platform atau *hypervisor* (virtual mesin monitor, perangkat lunak, firmware, atau perangkat keras yang membuat dan menjalankan virtual mesin) yang dapat digunakan dalam mengembangkan *IaaS* adalah Proxmox VE (Virtual Environment). Proxmox VE adalah sebuah platform virtualisasi *open source* untuk menjalankan *appliance* dan mesin virtual [3] Proxmox VE juga dapat membuat *Grid* komputer, dimana dapat menyatukan beberapa komputer menjadi sebuah komputer tunggal. Namun, para penyedia layanan *IaaS* belum mengetahui seberapa handal Proxmox VE dalam manajemen infrastruktur *server* secara fisik. Untuk mengetahui kehandalan sebuah *hypervisor* maka dibutuhkan sebuah analisis kinerja yang disesuaikan dengan lingkungan penyedia layanan *IaaS*.

Analisis *Overhead* adalah salah satu yang dapat dilakukan sebagai pengukuran kinerja. Dimana analisis *overhead* dapat memperlihatkan kinerja sebuah mesin konvensional dengan dikonfigurasi menjadi sebuah layanan dan dibandingkan dengan mesin virtual hasil dari kerja *hypervisor* (dalam hal ini

Proxmox VE) dan dibebankan dengan kerja yang sama. [4] melakukan analisis *overhead* dan merancang model *cloud infrastructure*, dalam penelitiannya menggunakan OpenStack sebagai *hypervisor*-nya. Dalam penelitiannya dihasilkan bentuk kurva *overhead* cenderung bersifat linier mendekati skalabilitas optimal, artinya virtualisasi dengan openstack mempunyai performance yang bagus, hampir mendekati skalabilitas *server* sebenarnya. Lain halnya dengan [5] melakukan analisis skalabilitas dari KVM (Kernel-base Virtualization Machine) hasilnya adalah KVM memiliki *overhead* jaringan virtualisasi yang sangat rendah, *overhead* CPU virtualisasi yang medium, dan *overhead* untuk hardisk virtualisasi yang sangat tinggi. Sedangkan [6] menganalisis kinerja dan merancang model *Cloud Computing* dengan layanan *IAAS*. Analisis dengan empat parameter uji yaitu *Turnaroud Time*, *Response Time*, *Resource Utilization*, dan *Throughput*. Hasilnya adalah dengan Proxmox VE dihasilkan performa satu mesin *server* dengan satu mesin konvensional didapatkan degradasi *turnaroud time*, *response time*, dan *troughput* sebesar 9,03%, 2,25%, dan 14,29%.

Dari latarbelakang yang telah dipaparkan, maka diambil sebuah judul penelitian analisis *overhead server cloud infrastructure* berbasis Proxmox VE *Hypervisor*. Penelitian ini akan merancang layanan *private cloud* infrastruktur dengan Proxmox VE, merancang skenario pengujian *overhead*, mulai dari analisis kinerja *server* konvensional dibandingkan dengan virtual mesin (VM), menyusun rekomendasi dalam menyediakan sebuah *server* fisik kepada penyedia layanan *cloud* infrastruktur dalam sebuah perbandingan.

2. Metode Penelitian

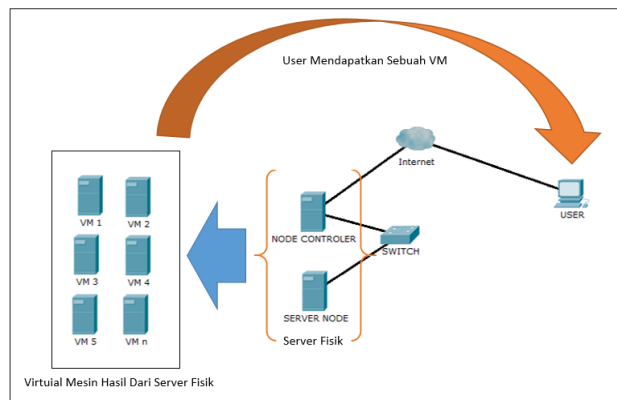
Penelitian dilaksanakan di Lab STIKOM Bali dikarenakan adanya peminjaman lab berupa mesin komputer yang nantinya dijadikan sebuah *server* penyedia layanan *cloud* infrastruktur.

2.1. Investigasi Sistem

Pada langkah awal dari penelitian ini adalah melakukan investigasi permasalahan. Dari hasil investigasi permasalahan ditemukan bahwa masalah yang didapat adalah berapa spesifikasi sebuah *server* yang diperlukan untuk membuat layanan *cloud* infrastruktur sesuai dengan layanan yang ditentukan. Spesifikasi ini diperlukan sebagai langkah awal dalam mempersiapkan infrastruktur layanan *cloud*. Analisis *overhead* adalah salah satu yang diperlukan untuk mengetahui sejauh mana perbedaan sebuah *server* bekerja dengan virtualisasi atau tanpa virtualisasi.

2.2. Desain Sistem

Desain sistem adalah tahap selanjutnya setelah tahap investigasi dan pengumpulan data dilakukan. Dalam tahap ini dilakukan desain sistem *cloud* infrastruktur dan desain pengujian sistem nantinya. Pada tahap desain sistem dilakukan rancangan topologi dari infrastruktur *cloud*. Topologi yang dirancangan dapat dilihat pada Gambar 1



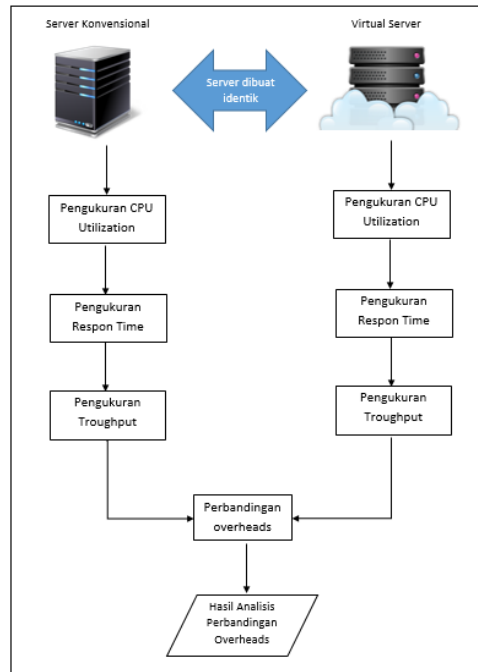
Gambar 1 Rancangan Topologi Cloud Infrastruktur

Dalam gambar 1 terlihat bagaimana topologi jaringan dan peletakan dari *server* fisik. Dalam topologi terlihat satu buah *node controller* yang digunakan sebagai *server* control untuk semua *server* fisik yang terhubung menjadi satu. Kemudian dari dua *server* fisik tersebut akan dibuat *server* virtual didalamnya memanfaatkan Proxmox VE sebagai *hypervisor*-nya. *Server* virtual inilah yang menjadi layanan infrastruktur *cloud* nantinya. User akan mendapatkan sebuah virtual mesin (vm) yang dikonfigurasi. Sesuai dengan skenario pengujian, dimana membandingkan *server* konvensional dengan *server* virtual. 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.

2.3. Rancangan Pengujian Sistem

Tahap terakhir adalah pengujian sistem, dalam tahap ini dibuat rancangan pengujian dari sistem untuk menganalisis *overhead* antara *server* fisik dengan *server* virtual. Rancangan scenario pengujian dapat dilihat pada Gambar 2



Gambar 2 Rancangan Skenario Pengujian

Pada Gambar 2 terlihat dua *server* konvensional dan *server* virtual, *server* tersebut akan dibuat seidentik mungkin dengan sistem operasi yang sama. Kemudian mulai dilakukan kompresi, *respon time*, dan *troughput*. Kemudian dari sana akan dilihat perbandingan *server* konvensional dengan *server* virtual.

Kemudian satu *server* konvensional juga akan dibandingkan jika mengerjakan sejumlah *request* misalnya 500 *request* dengan sebuah *server* cloud yang didalamnya dibuat 5 *server* virtual kemudian tiap vm mengerjakan 100 *request*. Jika dilihat jumlah *request* *server* sama yaitu 500 *request* namun *server* cloud terbagi ke 5 virtual *server*.

Selanjutnya akan dilakukan juga pengujian berapa tingkat penggunaan *resource* maksimal pada sebuah *server* konvensional jika didalamnya dibangun *server* virtual, dalam hal ini *server* akan dibuat sebagai *server* web. Kemudian akan dihitung menggunakan rumus (1) dan (2) (Nggilu, 2012).

$$Ov = Tav - Ta \quad (1)$$

$$Ovn = Tanv - Ta \quad (2)$$

Keterangan :

Ov = *overhead*

Ovn = *overhead* virtualisasi

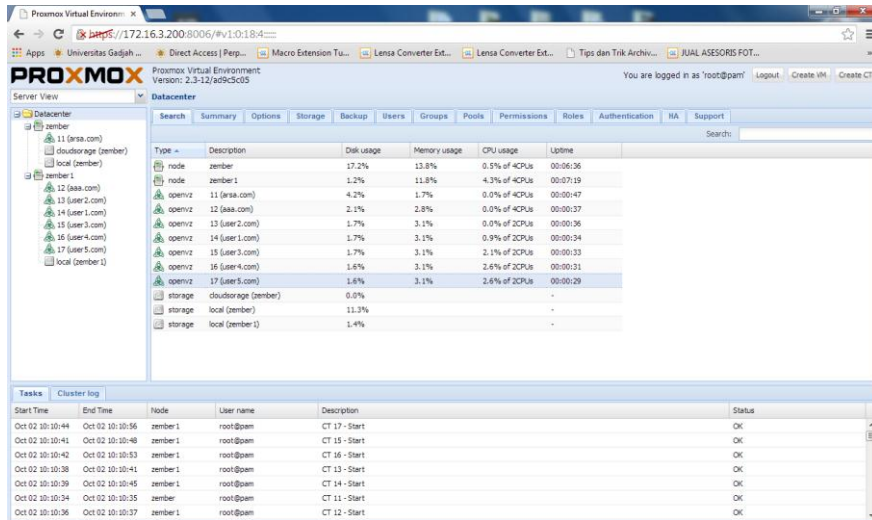
Ta = Waktu eksekusi pada sistem operasi non-virtual

Tav = Waktu eksekusi pada sistem operasi virtual

Tanv = waktu eksekusi pada sistem operasi virtual pada jumlah vm ke n

3. Hasil dan Pembahasan

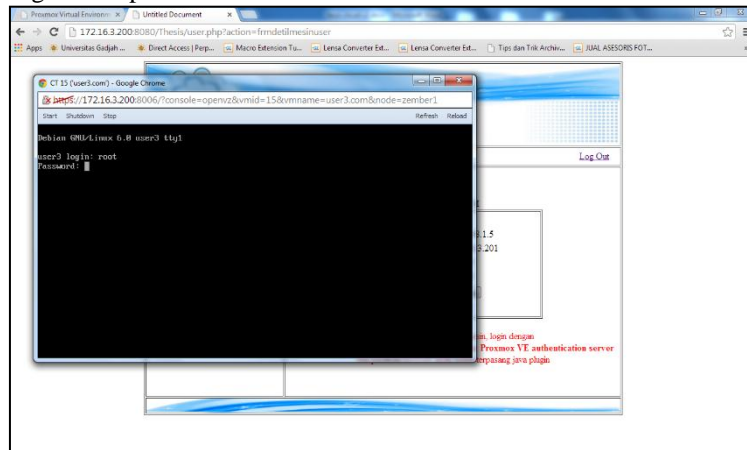
Pada tahap awal dilakukan proses instalasi sebuah *server* menggunakan Proxmox VE, dalam hal ini Proxmox bertindak sebagai *Hypervisor*. Gambar 3 memperlihatkan tampilan Proxmox VE yang telah terinstal serta bagaimana admin dapat leluasa mengelola seluruh mesin ada pada setiap *node*. Dalam hal ini *server* dengan nama zember bertindak sebagai *server controller* sekaligus menjadi sebuah *node*, sedangkan *server* zember1 menjadi *node*. Dalam gambar juga diperlihatkan seluruh virtual mesin yang berjalan diatas kedua *server* fisik tersebut



Gambar 3 Tampilan Proxmox di Web Browser

Selanjutnya dari sana dilakukan pembuatan mesin virtual yang akan dibandingkan dengan server konvensional. Virtual mesin dibuat dengan menggunakan GUI (Grafic User Interface) dari Proxmox dengan dikonfigurasi menjadi server web.

Proxmox VE memiliki fitur *View console* yang merupakan fitur spesial, dimana dengan fitur ini dapat mengakses vnc dari mesin dengan web browser. Fitur ini akan berjalan jika plugin java applet sudah terpasang dengan baik pada browser.



Gambar 4 Akses Mesin Dengan Browser

Gambar 4 memperlihatkan bagaimana dapat secara langsung mengkonfigurasi mesin server dengan web browser. Gambar 4 merupakan contoh virtual mesin yang dihasilkan oleh proxmox dengan sistem operasi debian 6.0.

Untuk memperoleh data *response time* akan dilakukan proses *request* kedalam server. *Request* yang dilakukan adalah permintaan halaman *index.html*. dimana sebelumnya mesin akan dikonfigurasi sehingga dapat menyediakan halaman ini. Halaman ini adalah halaman default ketika kita berhasil menginstal *web server* apache. Untuk *request* dijalankan dengan memanfaatkan *tool Httpperf*. Dengan perintah membuat seratus koneksi dengan setiap koneksi melakukan sepuluh *request* perdetik sehingga total *request* adalah seribu *request*. Tabel 1 memperlihatkan hasil dari keluaran *httperf* berupa waktu respon yang dibutuhkan satu vm hingga sepuluh vm, serta waktu respon dari mesin konvensional dalam mili detik (*milisecond*).

Sesui dengan rumus (1) maka dilakukan *overhead* waktu respon yaitu :

$$Ov = Tav - Ta$$

$$Ov = 7,6 \text{ ms} - 7,4 \text{ ms}$$

$$Ov = 0.2 \text{ ms}$$

Sesui dengan rumus (1) maka dilakukan *overhead* waktu respon yaitu :

$$Ovn = Tanv - Ta \text{ (mesin ke-10)}$$

$$Ovn = 39,6 - 7,4$$

$$O_{vn} = 32,2 \text{ ms}$$

$$\text{Capaian kerja} \frac{O_v}{O_{vn}} \times 100\% = \frac{0,2}{32,2} \times 100\% = 0,62\%$$

Tabel 1 Hasil *Respon Time*

SKENARIO	RESPON SERVER KONVENSIONAL	VIRTUAL MESIN KE-	RESPON VM (ms)
httpperf 100 koneksi dengan 10 request tiap koneksi pada 1 mesin	7.4	1	7.6
httpperf 100 koneksi dengan 10 request tiap koneksi pada 10 mesin instance dan 10 user yang mengakses masing-masing instance secara bersamaan		1	32.3
		2	38.8
		3	39.4
		4	39.9
		5	41.7
		6	40.9
		7	39.7
		8	40.1
		9	39.6
		10	39.6

Throughput diperoleh dari percobaan *request* yang dilakukan sebelumnya dengan menggunakan *tool httpperf*. Selanjutnya dari hasil pengujian *throughput* sesuai dengan tabel 2 menggunakan rumus (1) dan rumus (2) didapat hasil sebagai berikut :

Sesui dengan rumus (1) maka dilakukan *overhead Throughput* yaitu :

$$O_v = T_{av} - T_a$$

$$O_v = 61,7 - 64,50$$

$$O_v = -2,8$$

Sesui dengan rumus (1) maka dilakukan *overhead Throughput* yaitu :

$$O_{vn} = T_{anv} - T_a \text{ (mesin ke-10)}$$

$$O_{vn} = 10,8 - 64,50$$

$$O_{vn} = -53,7$$

$$\text{Capaian kerja} \frac{O_v}{O_{vn}} \times 100\% = \frac{-2,8}{-53,7} \times 100\% = 5,21\%$$

Tabel 2 Hasil Pengujian *Throughput*

SKENARIO	THROUGHPUT SERVER KONVENSIONAL	VIRTUAL MESIN KE-	THROUGHPUT (KB/s)
httpperf 100 koneksi dengan 10 request tiap koneksi pada 1 mesin	64.50	1	61.7
httpperf 100 koneksi dengan 10 request tiap koneksi pada 10 mesin instance dan 10 user yang mengakses masing-masing instance secara bersamaan		1	13
		2	11
		3	10.7
		4	10.4
		5	10.3
		6	10.3
		7	10.4
		8	10.5
		9	10.6
		10	10.8

Untuk mendapatkan *turnaroud time* akan dilakuak proses kompresi, dengan mengkompresi file yang besarnya 249 MB yang sampel datanya diambil dari *folder var* pada salah satu *vm*. Kemudian semua *instance* akan di-*copy*-kan file tersebut termasuk juga *server* konvensional yang menjadi pembanding. Kemudian akan dibandingkan satu *vm* yang berjalan pada *server node* dengan *server* konvensional. Kemudian kembali dilakukan proses kompresi namun kali ini dilakukan kepada semua *vm* pada *server node* secara hampir bersamaan. Tabel 6.1 memperlihatkan hasil dari pengujian proses kompresi. Tabel 3 memperlihatkan lama proses yang dibutuhkan untuk melakukan kompresi data dengan besar 249 MB. Untuk mengetahui waktu yang diperlukan digunakan perintah *time* diikuti dengan perintah kompresi. Data dikompres menjadi *tar.bz2*.

Selanjutnya dari hasil pengujian Kompresi sesuai dengan tabel 2 menggunakan rumus (1) dan rumus (2) didapat hasil sebagai berikut :

Sesui dengan rumus (1) maka dilakukan *overhead* Kompresi yaitu :

$$Ov = Tav - Ta$$

$$Ov = 39,964 - 33,354$$

$$Ov = 6,61$$

Sesui dengan rumus (1) maka dilakukan *overhead* Kompresi yaitu :

$$Ovn = Tanv - Ta \text{ (mesin ke-10)}$$

$$Ovn = 102,846 - 33,354$$

$$Ovn = 69,492$$

$$\text{Capaian kerja } \frac{Ov}{Ovn} \times 100\% = \frac{6,61}{69,492} \times 100\% = 9,51\%$$

Tabel 3 Hasil Pengujian Kompresi

SKENARIO	KOMPRESI SERVER KONVENSIONAL	VIRTUAL MESIN KE-	KOMPRESI (s)
Proses Kompresi 249 MB	33.354	1	39.964
Proses Kompresi 249 MB kepada seluruh VM hampir bersamaan		1	119.321
		2	110.675
		3	105.093
		4	114.264
		5	110.495
		6	104.962
		7	110.224
		8	99.702
		9	108.423
		10	102.846

4. Simpulan

Dari penelitian yang dilakukan didapat beberapa kesimpulan yaitu, Proxmox dapat digunakan sebagai *hypervisor* untuk menghasilkan *infrastructure server cloud*. Dari hasil pengujian *overhead* didapat *overhead* sebuah mesin konvensional dengan satu mesin *virtual respon time, throughput*, kompresi data berturut-turut adalah 0,2 ms, -2,8 KB/s, 6,61 s. *overhead* antara *server* konvensional dengan satu *server* virtual tidak terlalu besar, namun ketika dilakukan pembuatan mesin virtual 10 VM memiliki perbedaan yang signifikan dimana capaian kerja virtual mesin dengan jumlah sepuluh *respon time* 0,62%, *throughput* 5,21%, dan kompresi 9,51%.

Daftar Pustaka

- [1] Mell, P. and Grance, T., 2011, The NIST Definition of *Cloud Computing*, *NIST Special Publication 800-145*.
- [2] Cearley, D.W., 2010, *Cloud Computing Key Initiative Overview*, *Gartner, Inc*, Stamford.
- Purbo, O.W., 2012, *Membuat Sendiri Cloud Computing Server Menggunakan Open Source*, Penerbit Andi, Jakarta.
- [3] Nggilu, F.S., 2012, *Analisis Overhead Sebagai Salah Satu Faktor Skalabilitas Private Cloud Computing untuk Layanan IaaS*, *Tesis*, Program Studi Magister Teknologi Informasi Fakultas Teknik UGM, Yogyakarta.
- [4] Rahma, F., Adji, T.B. and Widyawan, W., 2013. Scalability Analysis of KVM-Based Private *Cloud For IaaS*. *International Journal of Cloud Computing and Services Science*, 2(4), p.288.
- [5] Arsa, W. and Mustofa, K., 2014. Perancangan dan Analisis Kinerja Private *Cloud Computing* dengan Layanan *Infrastructure-As-A-Service (IAAS)*. *IJCCS-Indonesian Journal of Computing and Cybernetics Systems*, 8(2), pp.165-176.