BAB II
KAJIAN PUSTAKA

2.1 Penelitian Terkait

Penelitian ini mengacu pada beberapa referensi yang terkait dengan penggunaan metode *load balancing* dan *load aggregation* di lingkungan *cloud*. Berdasarkan pada penelitian yang ada, kedua metode tersebut umumnya digunakan untuk menangani permasalahan beban kerja *server* fisik dengan tujuan untuk meningkatkan kinerja *server* dan optimasi penggunaan sumber daya komputasi. Cara yang dilakukan untuk mencapai tujuan tersebut yaitu dengan memindahkan mesin virtual antar *server* fisik, atau yang dikenal dengan teknik *live migration*. Metode *load balancing* dan *load aggregation* dapat dilakukan baik secara statis maupun dinamis dengan menggunakan berbagai macam algoritma sesuai dengan kebutuhan operasional.

Penelitian terkait metode *load balancing*, salah satunya dilakukan oleh Akshay Chandak [CHA-12] dengan judul *"Dynamic Load Balancing of Virtual Machines using QEMU-KVM"*. Implementasi sistem dilakukan dengan menggunakan *threshold*, yaitu dengan mengkategorikan beban kerja ke dalam tiga bagian yaitu beban kerja rendah, beban kerja tinggi, dan beban kerja normal. Proses *load balancing* dijalankan ketika beban kerja *server* fisik tidak berada dalam batas beban kerja normal. Tujuan dari penelitian ini adalah untuk menyeimbangkan beban kerja pada seluruh *server* fisik yang ada dengan melakukan pemindahan mesin virtual berdasarkan beban kerja di setiap *server* fisik. Hasil pengujian sistem yang dilakukan [CHA-12] membuktikan bahwa beban kerja pada masing-masing *server* fisik dapat mencapai keseimbangan pada batas beban kerja normal. Proses *load balancing* ini dilakukan secara dinamis dengan menggunakan sebuah *policy engine* yang dijalankan setiap 25 menit sekali. Penelitian serupa juga pernah dilakukan oleh Terry C. Wilcox Jr. [WIL-09] dengan menggunakan *hypervisor* yang berbeda. *Hypervisor* sendiri merupakan jenis perangkat lunak yang digunakan untuk melakukan teknik virtualisasi.

Penelitian lain mengenai *load aggregation* juga telah dilakukan oleh sejumlah peneliti seperti Anton Beloglazov [BEL-10], Deep Mann [MAN-12],

Aziz Murtazaev [MUR-11], dan Daniel Versick [VER-10]. Salah satu penelitian oleh Daniel Versick dengan judul "*Reducing Energy Consumption by Load Aggregation with an Optimized Dynamic Live Migration of Virtual Machines*" membuktikan bahwa penggabungan beban kerja dari beberapa mesin virtual dapat mengoptimalkan penggunaan sumber daya komputasi dan sekaligus mengurangi jumlah *server* fisik yang digunakan. Selain itu, Deep Mann dalam penelitiannya juga melakukan penggabungan beban kerja mesin virtual dari beberapa *server* fisik dengan teknik *single threshold*. Penerapan metode *load aggregation* pada umumnya bertujuan untuk mengurangi jumlah konsumsi energi dengan cara optimasi penggunaan sumber daya komputasi *server* fisik.

2.2 *Cloud Computing*

Cloud computing dapat didefinisikan sebagai penggunaan berbagai aplikasi dan sumber daya komputasi melalui sebuah lingkungan jaringan tanpa memperhatikan kepemilikan dan manajemen. Dengan *cloud computing*, sumber daya komputasi dan data yang digunakan untuk menyelesaikan suatu pekerjaan tidak lagi disimpan dalam sebuah komputer personal, namun disimpan di suatu tempat agar dapat diakses dari manapun dan kapanpun [SCA-09].

Kelas Layanan	Akses Utama & Alat Manajemen	Konten Layanan
SaaS	Peramban *Web*	***Cloud Applications*** *Social networks, Office suites, CRM, Video Processing*
PaaS	Lingkungan Pengembangan *Cloud*	***Cloud Platform*** Bahasa Pemrograman, *Frameworks, Mashups editors*, Data Terstruktur
IaaS	Manajer Infrastruktur Virtual	***Cloud Infrastructure*** *Compute Servers, Data Storage, Firewall, Load Balancer*

Gambar 2.1 Klasifikasi *Cloud Computing*
Sumber: [BUY-11:14]

2.2.1 Fitur *Cloud Computing*

Cloud computing membawa sejumlah fitur baru dibandingkan dengan paradigma komputasi lainnya, seperti yang dinyatakan oleh [WAN-08]:

- **Skalabilitas dan layanan *on-demand*** – *Cloud computing* menyediakan sumber daya dan layanan kepada pengguna sesuai permintaan. Selain itu, sumber daya dapat ditambah dan dikurangi dengan mudah di antara beberapa *data center*.
- **Antarmuka berbasis pengguna** – antarmuka *cloud* terletak pada lokasi yang independen dan dapat diakses oleh antarmuka pada umumnya seperti layanan *Web* dan peramban *Internet*.
- **Jaminan *Quality of Service* (QoS)** – *Cloud computing* memberi jaminan kualitas layanan kepada pengguna dalam hal performa perangkat keras seperti CPU, *bandwidth*, dan kapasitas *memory*.
- **Sistem Otonom** – Sistem *cloud computing* adalah sistem otonom yang dikelola secara transparan kepada pengguna. Namun, perangkat lunak dan data di dalam *cloud* dapat secara otomatis dibentuk dan digabungkan menjadi sebuah *platform* sederhana berdasarkan pada kebutuhan pengguna.

2.2.2 Layanan *Cloud Computing*

Berdasarkan pada beberapa model layanan yang terdapat di *cloud computing*, penelitian ini akan berfokus pada model *Infrastructure as a Service*. Hal ini dikarenakan masalah optimasi penggunaan sumber daya komputasi yang diangkat penulis sangat erat kaitannya dengan manajerial infrastruktur virtual. Dengan begitu, sistem virtualisasi *server* yang akan dibangun hanya diteliti dari sisi penggunaan infrastruktur *cloud* saja. Berikut penjelasan singkat beberapa model layanan c*loud computing* yang ada menurut [KEP-10]:

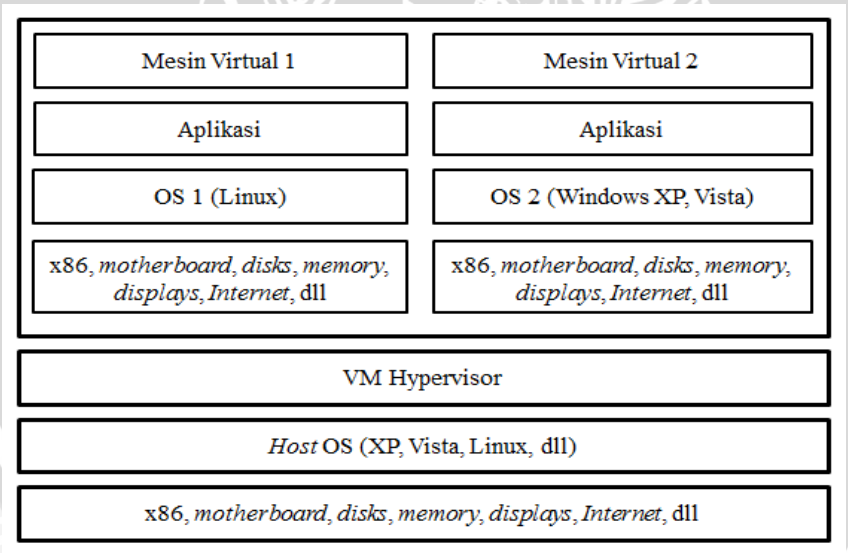
- ***Infrastructure as a Service*** – IaaS adalah cara untuk menjadikan infrastruktur *cloud computing* seperti *server*, penyimpanan data, jaringan, dan sistem operasi sebagai sebuah layanan *on-demand*.
- ***Platform as a Service*** – PaaS dapat didefinisikan sebagai *platform* komputasi yang memungkinkan penyediaan aplikasi *web* secara cepat dan

mudah tanpa harus memiliki dan memelihara perangkat lunak beserta infrastruktur yang berada di bawahnya.

- ***Software as a Service*** – SaaS adalah pengiriman model perangkat lunak dimana perangkat lunak tersebut terkait dengan data komputer yang terpusat di *cloud*. SaaS biasanya diakses oleh pengguna dengan menggunakan berbagai perangkat melalui *web browser*.

2.3 Virtualisasi

Virtualisasi adalah suatu teknik yang memungkinkan sumber daya fisik tunggal (seperti *server*, sistem operasi, aplikasi, atau perangkat penyimpanan) muncul sebagai beberapa sumber daya logis; atau membuat beberapa sumber daya fisik (seperti perangkat penyimpanan atau *server*) muncul sebagai sumber daya logis tunggal. Teknik virtualisasi pada dasarnya memungkinkan sebuah komputer untuk melakukan pekerjaan dari beberapa komputer, dengan berbagi sumber daya dari perangkat keras tunggal di beberapa lingkungan [MYI-10].



Gambar 2.2 Arsitektur Virtualisasi
Sumber: [ALI-11]

Dalam teknik virtualisasi, terdapat dua cara yang paling umum digunakan yaitu *full virtualization* dan *paravirtualization*. *Full virtualization* memungkinkan *hypervisor* mengontrol sumber daya perangkat keras dan kemudian mengemulasikannya secara penuh ke sistem operasi mesin virtual. Dengan teknik ini, tidak

ada modifikasi sistem operasi mesin virtual yang diperlukan. Sedangkan dalam *paravirtualization*, *hypervisor* mengontrol sumber daya perangkat keras dan menyediakan API ke sistem operasi mesin virtual agar proses pengaksesan dapat dilakukan ke perangkat keras tersebut. Teknik ini memerlukan modifikasi sistem operasi mesin virtual [CHA-12].

2.3.1 Karakteristik Virtualisasi

Teknik virtualisasi *server* yang telah banyak digunakan di lingkungan *cloud* memiliki sejumlah kelebihan dan permasalahan. Kelebihan yang didapatkan dari penggunaan teknik virtualisasi *server* menurut [LEJ-08] antara lain:

- Peningkatan utilisasi perangkat keras,
- Peningkatan efektifitas manajemen infrastruktur *server*,
- Perbaikan proses pemulihan kegagalan sistem komputer,
- Pengurangan kebutuhan konsumsi energi,
- Pengurangan kebutuhan tempat dan beban jaringan,
- Serta pengurangan biaya tes dan pengembangan perangkat lunak.

Selain memiliki aspek keuntungan, teknik virtualisasi *server* juga memiliki beberapa permasalahan yang dapat muncul apabila tidak direncanakan dengan baik. Beberapa hal yang perlu diperhatikan menurut [MAR-06] dan [GAR-10] antara lain:

- **Daya Listrik** – Walaupun secara umum virtualisasi *server* akan mengurangi biaya penggunaan daya listrik, dibutuhkan perencanaan yang matang terkait dengan instalasi kelistrikan. Sebuah *server* fisik yang di dalamnya terdapat banyak mesin virtual, akan mengkonsumsi energi listrik yang relatif lebih banyak daripada *server* fisik normal.
- ***Backup*** – Dalam infrastruktur virtualisasi *server*, ukuran data yang perlu di-*backup* sangat besar sehingga diperlukan sistem *backup* yang dapat mengirimkan data dalam jumlah besar dengan cara yang cepat.
- **Jaringan** – Virtualisasi *server* dapat menyebabkan lalu lintas paket data yang sangat tinggi pada kartu jaringan dan *port switch*. Penuhnya *buffer* pada *switch* dapat berakibat pada hilangnya paket sehingga diperlukan pengiriman ulang paket yang hilang tersebut.

- **Performa** - Konsolidasi *server* akan meningkatkan utilisasi prosesor, sehingga disk I\O *request* akan semakin meningkat. Hal ini berpotensi untuk menimbulkan *bottleneck,* karena kecepatan akses disk jauh lebih lambat dibandingkan dengan kecepatan prosesor. Dalam hal ini, diperlukan sistem penyimpanan data yang baik dan kapasitas *memory* yang cukup.

2.3.2 *Live Migration*

Live migration adalah salah satu fitur unggulan yang hampir dimiliki oleh semua *hypervisor* mesin virtual. Dalam sistem virtualisasi *server* yang akan dibangun, *live migration* digunakan untuk memindahkan *server* dari satu mesin fisik ke mesin fisik lainnya. Ketika proses *live migration* terjadi, klien sama sekali tidak dilibatkan dalam proses ini sehingga mereka tetap dapat mengakses layanan secara normal tanpa merasakan adanya perpindahan *server*. Secara umum, *live migration* digunakan untuk menangani proses *load balancing*, pemeliharaan perangkat lunak, *upgrading* perangkat keras, dan *upgrading* perangkat lunak.

Berikut merupakan urutan algoritma umum *live migration* mesin virtual antar satu mesin fisik ke mesin fisik lainnya menurut [LUB-07]:

1. Permintaan migrasi diterima (migrasi *VM* dari mesin A ke mesin B)
    1) Membuat perintah-perintah eksternal
    2) Menghubungkan dan mengirimkan *header*
    3) Mengalokasikan sumber daya dan pengaturan
2. Pemindahan *memory*
    1) Proses pertama, transfer seluruh *memory pages* (iterasi pertama)
    2) Untuk iterasi selanjutnya, transfer seluruh *dirty pages* dari iterasi i-1
    3) Proses dilakukan terus menerus hingga selesai.
3. Menghentikan mesin virtual
4. Pemindahan *VM state*
    1) Setiap perangkat memindahkan *state* masing-masing.
    2) *Dirty pages* dari iterasi terakhir dipindahkan.
5. Melanjutkan mesin virtual
    - Pada mesin B, jika proses migrasi berhasil

- Mengirimkan paket *broadcast* melalui *Ethernet* untuk memberikan informasi lokasi mesin virtual yang baru.

- Pada *localHost* mesin A jika proses migrasi gagal.

2.4 Metode Penanganan Beban Kerja

Beberapa teknik yang digunakan dengan tujuan peningkatkan kinerja sistem dapat memberikan efek yang merugikan, khususnya pada konsumsi energi. Sebagai contoh, dalam sebuah *cluster server*, *load balancing* digunakan untuk membagi beban kerja agar kinerja maksimum dari sebuah sistem dapat tercapai. Namun, beban kerja yang didistribusikan pada banyak *server* fisik, untuk tingkat penggunaan yang rendah, akan menjadikan konsumsi energi tidak sebanding dengan jumlah pekerjaan yang dilakukan [MIT-10]. Solusi yang mungkin dilakukan adalah dengan memaksimalkan beban kerja pada satu *server* fisik. Namun, solusi tersebut tentu saja akan mengurangi kinerja karena pemanfaatan *server* yang melebihi batas optimal [SRI-08].

Berdasarkan [VER-10], terdapat dua metode yang dapat digunakan di lingkungan *cloud computing* untuk mengatasi permasalahan beban kerja *server* fisik, yaitu *load balancing* dan *load aggregation*/*server consolidation*. Berikut tabel yang menjelaskan karakteristik dari penggunaan kedua metode tersebut:

Tabel 3.1 Penggunaan Metode *Load Balancing* dan *Load Aggregation*

Tujuan	***Server consolidation***	***Load balancing***
Kapan dilakukan migrasi?	Terdapat banyak mesin fisik dengan tingkat utilisasi rendah	Terjadi ketidakseimbangan beban kerja pada banyak mesin fisik
Mesin Virtual mana yang dimigrasi?	Mesin virtual dari mesin fisik dengan beban kerja rendah	Mesin virtual dari mesin fisik dengan beban kerja tinggi
Ke mana mesin virtual dipindahkan?	Mesin fisik dengan beban kerja tinggi	Mesin fisik dengan beban kerja rendah

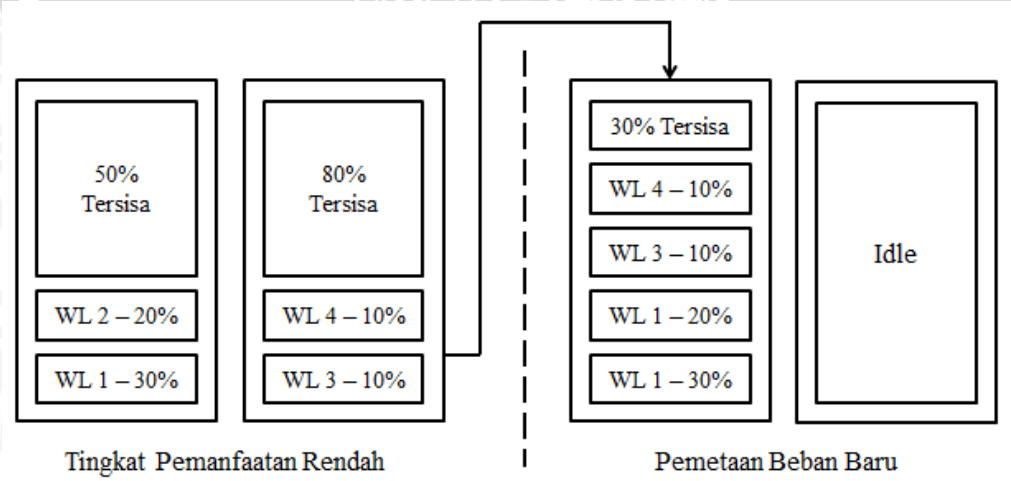
Sumber: [DAS-12]

2.4.1 *Load Balancing*

Load balancing adalah metode untuk melakukan pemetaan pekerjaan ke sumber daya fisik. Dengan kata lain, *load balancing* adalah metode untuk memutuskan pekerjaan yang harus dialokasikan pada sumber daya *server* fisik yang tepat [TER-09]. Pada sistem virtualisasi *server* ini, metode *load balancing* digunakan dengan tujuan untuk menyeimbangkan beban kerja tinggi CPU ke beberapa *server* fisik. Beban kerja tersebut didistribusikan ke *server* fisik lain dengan cara memindahkan mesin virtual sesuai kebutuhan. Dengan metode *load balancing*, seluruh mesin virtual yang berada di dalam sistem virtualisasi *server* dapat terus beroperasi tanpa terganggu oleh beban kerja tinggi CPU *server* fisik.

2.4.2 *Load Aggregation*

Ide dari *load aggregation* muncul berdasarkan pada pengamatan yang membuktikan bahwa banyak *server* fisik tidak optimal dalam memanfaatkan sumber daya *server* fisik yang tersedia. Penggabungan beban atau dengan cara konsolidasi *server*, mengurangi total energi yang dikonsumsi karena *server* fisik yang ada tidak menggunakan energi secara proporsional, yaitu sejumlah besarnya energi yang dikonsumsi pada tingkat penggunaan yang rendah [VER-08]. Konsolidasi mesin virtual ke sebuah *server* fisik memberikan peluang untuk mengurangi konsumsi energi secara menyeluruh karena *server* fisik hanya dioperasikan sesuai dengan kebutuhan permintaan.



Gambar 2.3 Skema Penggabungan Beban Kerja
Sumber: [FEL-11]