

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Seiring majunya perkembangan teknologi pada saat ini, terjadinya sebuah perkembangan pada teknologi dan informasi yang signifikan terutama pada kegiatan sehari-hari yang dapat dikerjakan dalam waktu singkat. Maka dari itu pengembangan teknologi berdasarkan internet pada saat ini lebih mengarah pada aplikasi *system*

Istilah *Web Services* sudah mulai berkembang dan mulai pada sudah banyak pula yang menggunakan. Dikarenakan kelebihan dari *web service* itu sendiri, diantaranya yaitu kemudahannya dalam pengiriman data, karena secara umum *web service* menggunakan protokol TCP/IP dan HTTP sebagai sarana komunikasinya. Seperti diketahui protokol HTTP merupakan protocol yang umum dipakai di internet. Selain kemudahannya dalam pengiriman data *web service* juga menggunakan format yang digunakan standar yaitu menggunakan format XML (*Extensible Markup Language*), sehingga dapat dipergunakan untuk mengintegrasikan berbagai system yang berbeda-beda platform maupun arsitektur.

Web service ini telah dibangun dengan salah satu infrastructure of a service (IAAS) yang akan digunakan pada tugas akhir ini yaitu openstack, dikarenakan openstack ini menyediakan platform komputasi awan *open source* di mana-mana untuk *cloud* publik maupun pribadi, juga merupakan perangkat lunak gratis. OpenStack ini juga memberikan fasilitas untuk menyebarkan mesin virtual (VM) dan instance lain yang menangani tugas yang berbeda untuk mengelola lingkungan *cloud*, untuk mengakses openstack ini pengguna harus terkoneksi dengan internet agar bisa memasuki dan menjalankan situs tersebut. Jadi pengguna hanya cukup menyediakan sebuah pc dan perangkat jaringan internet agar bisa terkoneksi dengan server internet dan menyimpan data di komputer server dan tidak harus menyediakan hard disk yang berkapasitas besar pada komputer tersebut. Maka dari itu, *platform* layanan harus online setiap saat untuk memberikan *service* kepada semua *client*, sedangkan platform juga pasti akan mengalami shutdown dan *restart* untuk keperluan *maintanance*. *Live migration* adalah suatu solusi yang dapat

mengurangi *downtime platform* saat dilakukan *maintenance* dengan memindahkan *service* ke *platform* cadangan dalam keadaan menyala. *Live migration* mampu membekukan aplikasi yang sedang berjalan, melakukan *record memory* sesuai dengan keadaan aplikasi saat itu, memindahkan hasil record ke platform cadangan, restore data record dari platform utama di platform cadangan, dan melanjutkan aplikasi berjalan sesuai dengan keadaan pada saat *checkpoint* dibuat. Solusi ini sangat berguna untuk menjaga ketersediaan layanan pada saat platform harus berhenti sejenak.

Memahami kinerja migrasi dibawah tekanan sistem dan jaringan kegagalan diperlukan untuk mencapai tujuan diatas. Total waktu migrasi tergantung pada aktivitas VM (*Virtual Machine*). Jika VM sangat aktif dan intensif memori, proses migrasi mungkin memerlukan waktu lebih lama untuk mentransfer data status VM. metode *live migration* yang paling umum saat ini dapat dengan mudah dibedakan berdasarkan iterasi yang berbeda yaitu terdapat metode *pre-copy* (Memori ditransfer terlebih dahulu dan kemudian eksekusi ditransfer) dan metode *post-copy* (Eksekusi ditransfer terlebih dahulu dan kemudian memori). kedua metode tersebut memiliki kelemahan dan kekuatan sendiri. Metode *pre copy* tidak dapat berkinerja baik dalam lingkungan intensif tulis. Ini menunjukkan peningkatan drastis dalam total waktu migrasi. Di sisi lain, *post-copy* juga tidak berkinerja baik dengan read beban intensif. maka dari itu Metode *live migration* yang akan dilakukan yaitu metode *hybrid*. Metode ini akan mengambil yang terbaik dari masing-masing metode *pre copy* dan *post-copy*. Metode ini dapat menunjukkan bahwa *pre-copy* dan *post-copy* dapat berkinerja baik dalam skenario ini[1]

Dalam tugas akhir ini penulis menyelidiki keandalan *live migration* di openstack, dengan meningkatkan tekanan system selama migrasi. untuk permasalahan terkait topik tugas akhir ini, sebelumnya sudah pernah diangkat sebagai tugas akhir juga dengan judul yang berbeda yaitu dengan judul "*LIVE MIGRATION PADA CLOUD COMPUTING DENGAN METODE HYBRID*" pada tugas akhir tersebut dilakukan penelitian terhadap *live migration* pada *cloud computing* dengan menggunakan *Kernel-based Virtual Machine (KVM)* berbeda halnya dengan penelitian yang dilakukan pada buku ini yaitu *live migration* pada *cloud computing* dengan menggunakan Openstack. disini penulis akan mensimulasikan beberapa kegagalan jaringan seperti *packet delay, jitter, Throughput*. Yang dapat menentukan dampak kegagalan jaringan pada kinerja migrasi. Hasil dapat digunakan untuk memandu administrator pusat

data dalam keputusan migrasi dan lokalisasi kesalahan. Selain itu, dapat membantu untuk menemukan kemacetan dan metode pengoptimalan untuk *live migration* di openstack.

## 1.2 Rumusan Masalah

Dari latar belakang yang sudah dijelaskan sebelumnya, adapun rumusan masalah dari Tugas Akhir adalah sebagai berikut :

1. Bagaimana konsep membangun suatu infrastruktur *web service* agar dapat melakukan *live migration*?
2. Bagaimana kinerja *live migrate* instance antar 2 host berdasarkan *migration time*, *downtime*, ukuran *data transfer* pada openstack?
3. Bagaimana pengaruh dari *live migration* terhadap kualitas layanan yang sedang berjalan pada server virtual?
4. Bagaimana performansi (*Quality of Service*) jaringan khususnya *Delay*, *Throughput*, *Jitter*, *packet loss*?

## 1.3 Tujuan dan Manfaat

Berdasarkan permasalahan yang telah dijelaskan, maka tujuan penelitian sebagai berikut :

1. Mengimplementasikan suatu system *webservice* yang memiliki kemampuan *live migration*
2. Menguji kinerja *live migration* berdasarkan parameter *migration time*, ukuran *data transfer* dan downtime pada mesin virtual
3. Mengukur dan melakukan pengujian performansi sistem yang dibuat yang berupa *Delay*, *Throughput*, *Jitter*, *packet loss*
4. memahami pengaruh dari *live migration* terhadap kualitas layanan yang sedang berjalan pada server virtual?

Adapun manfaat penelitian Tugas Akhir ini adalah sebagai berikut: :

1. Memberikan implementasikan suatu *system webservice* yang memiliki kemampuan *live migration*

2. Memberikan referensi kepada pengembang lain mengenai implementasi sistem serupa agar dapat meningkatkan performansi.

#### 1.4 Batasan Masalah

Dalam Tugas Akhir ini terdapat beberapa hal yang harus dibatasi untuk memberi fokus kepada objek yang dikerjakan, diantaranya :

1. Berbasis hanya untuk *live migration* pada *cloud computing*.
2. Parameter *live migration* yang digunakan adalah *downtime*, *migration time*, *data transfer*.
3. Tidak membahas tentang keamanan system.
4. *Hypervisor* menggunakan VMware Workstation
5. *Virtual Machine* dibangun dengan menggunakan system operasi Ubuntu server 20.04.
6. Instalasi Openstack yang digunakan adalah Openstack versi Xena
7. *Storage* yang digunakan adalah Ceph.

#### 1.5 Metode Penelitian

Metodologi yang digunakan untuk menyelesaikan penulisan Tugas Akhir ini adalah sebagai berikut:

#### 1.6 Sistematika Penulisan

Sistematika penulisan laporan adalah sebagai berikut:

- Bab 1 Pendahuluan

Bab ini berisi latar belakang, permasalahan, tujuan, metode penelitian, dan sistematika penulisan.

- Bab 2 Tinjauan Pustaka

Bab ini berisi penjelasan teori, alat, dan perlengkapan yang digunakan.

- Bab 3 Skema Simulasi

Bab ini berisi alur kerja dan alur perancangan sistem.

- Bab 4 HASIL SIMULASI DAN ANALISIS

Bab ini berisi langkah simulasi dan pengujian yang dilakukan, hasil pengujian, dan analisis dari hasil pengujian yang didapat.

- Bab 5 KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dan saran tugas akhir ini.

### 1.7 Jadwal Pelaksanaan

Berisi jadwal pelaksanaan pengerjaan Tugas Akhir.

**Tabel 1.1.** Tabel Milestone

No.	Deskripsi Tahapan	Durasi	Tanggal Selesai	Milestone
1.	Pemilihan Topik	1 Bulan	22 September 2021	Penentuan Judul
2.	Desain Sistem	1 Bulan	10 Oktober 2021	Diagram Blok dan Spesifikasi input
3.	Pemilihan Komponen	1 Bulan	22 November 2021	List Komponen yang digunakan seperti node controller -compute, node compute, dan storage ceph
4.	Perancangan dan Implementasi Sistem	6 Bulan	30 Mei 2022	Membangun node controller-compute, node compute dan storage ceph pada vm, Membangun openstack
5.	Penyusunan laporan dan buku TA	2 Bulan	18 Juli 2022	Buku TA selesai