

# BAB I PENDAHULUAN

## 1.1. Latar Belakang Masalah

Kerugian yang terjadi dalam layanan *website* adalah tidak efisiennya penyediaan sumber daya *server* untuk melayani sebuah kunjungan. Umumnya penyedia layanan *website* menambah kuantitas *server* untuk menyesuaikan kebutuhan tertinggi pada suatu *website*. Namun, hal tersebut kurang efisien ketika kunjungan di *website* sedang rendah. Permasalahan ini dapat diatasi dengan memanfaatkan fitur *auto-scaling* pada *cloud computing*. Fitur ini dapat menyesuaikan jumlah *server* berdasarkan volume kunjungan. Ketika kunjungan meningkat, *auto-scaling* secara otomatis menambah jumlah *server*. Begitu pula sebaliknya, *auto-scaling* secara alternatif menghapus *server* jika terjadi penurunan kunjungan [3]. Berdasarkan hal tersebut *Openstack* menghadirkan *Senlin* untuk menjalankan fitur *auto-scaling*.

*Openstack* merupakan *cloud computing* platform yang mengelola sumber daya *compute*, *network*, dan *storage* dengan mengizinkan penyediaan permintaan sumber daya virtual melalui *self-service portal* [25]. *Openstack* adalah *software open-source*, dimana semua *user* dapat mengakses *source-code Openstack*. *Source-code* tersebut dapat dimodifikasi sesuai kebutuhan *user*, dan dengan bebas membagikan modifikasi tersebut kepada *user* lain. Oleh karena itu *Openstack* sangat fleksibel dan *powerful* sebagai sistem *cloud computing*. Serta dengan menambahkan layanan *Senlin* maka sistem *cloud computing* mampu membuat dan mengoperasikan *cluster* dengan objek serupa. *Senlin* dapat mengelola *deployments* dan *orchestration* kumpulan sumber daya berskala besar dengan lebih mudah [19].

Penelitian sebelumnya tentang implementasi *vertical auto-scaling* pada *container* dan *virtual machine* menunjukkan kombinasi dari sistem tersebut memiliki kinerja yang baik [2]. Namun penggunaan *vertical auto-scaling* tidak fleksibel karena sebuah komputer memiliki jumlah *core* dan *memory* yang terbatas, sehingga *vertical auto-scaling* memiliki batas maksimum dalam mengoptimalkan sumber daya *server*. Penelitian lainnya tentang implementasi *auto-scaling* pada layanan *VoIP* menunjukkan penggunaan metode *predictive auto-scaling* mampu

meningkatkan *availability* layanan dan meningkatkan efisiensi penggunaan *memory* [1]. Namun penggunaan *predictive auto-scaling* lemah terhadap lonjakan kunjungan di waktu abnormal seperti adanya acara besar yang diselenggarakan di luar waktu kerja. Oleh karena itu, Tugas Akhir ini meneliti tentang skala lebih luas, seperti penggunaan beberapa *server* yang dapat aktif ketika sumber daya *computing* mencapai batas *threshold*.

Penelitian Tugas Akhir ini menggunakan sistem *horizontal auto-scaling* metode *reactive* yang dapat menambah atau menghapus jumlah *server* berdasarkan *threshold* yang diberikan. Dibandingkan dengan sistem *vertical auto-scaling* maupun sistem *predictive auto-scaling*, penelitian ini lebih fleksibel digunakan pada berbagai macam *computer server* dan waktu kunjungan. Penelitian ini menerapkan *Senlin* pada *Openstack* untuk mengoptimalkan sumber daya *computing* pada layanan *web server*.

## **1.2. Rumusan Masalah**

Adapun rumusan masalah dari Tugas Akhir ini, sebagai berikut:

1. Bagaimana desain dan implementasi sistem *auto-scaling* pada *Openstack*?
2. Bagaimana kinerja *auto-scaling* di parameter *cpu-usage*, *throughput*, *response time*, dan *request loss* pada *web server*?

## **1.3. Tujuan dan Manfaat**

Tujuan dari Tugas Akhir ini adalah mengimplementasikan *auto-scaling* untuk menambah atau menghapus jumlah *server* berdasarkan *cpu-usage*. Serta mengetahui kinerja *auto-scaling* di parameter *throughput*, *response time*, dan *request loss* pada *web server*.

Adapun manfaat Tugas Akhir ini adalah dapat mengimplementasikan *auto-scaling* untuk menambah atau menghapus jumlah *server* berdasarkan *cpu-usage*. Serta mengetahui efektivitas sistem *auto-scaling* pada parameter *throughput*, *response time*, dan *request loss* dengan melihat efisiensi sumber daya *web server*.

#### 1.4. Batasan Masalah

Dalam Tugas Akhir ini, dilakukan pembatasan masalah sebagai berikut:

1. Implementasi *horizontal auto-scaling* diterapkan pada *Openstack Cloud*.
2. Parameter pengujian yang digunakan meliputi *cpu-usage*, *throughput*, *response time*, dan *request loss*.
3. Pengujian dilakukan menggunakan jaringan lokal IPV4.
4. *Cloud computing* yang dibangun hanya sebatas *private cloud*.
5. *NFS-server* hanya *mount directory* tanpa *authorization*.
6. Tidak membahas keamanan jaringan pada *server*.

#### 1.5. Metode Penelitian

Metodologi yang digunakan pada penelitian ini sebagai berikut:

1. Studi Literatur  
Mencari informasi berkaitan dengan *auto-scaling Openstack* berupa jurnal, dokumentasi resmi, dan diskusi untuk menunjang selesainya penelitian ini.
2. Perancangan Sistem  
Melakukan perancangan sistem sesuai dengan tujuan yang ingin dicapai pada penelitian ini.
3. Pengujian  
Melakukan pengujian terhadap sistem yang telah dirancang dengan skenario dan parameter uji yang telah ditentukan.
4. Analisis Kerja Sistem  
Menganalisis hasil dari pengujian yang dikerjakan sesuai dengan skenario yang telah ditetapkan.
5. Kesimpulan  
Tahap terakhir dari keseluruhan pengerjaan Tugas Akhir, dimana pada tahap ini ditarik kesimpulan tentang analisis yang telah dilakukan.

## 1.6. Jadwal Pelaksanaan

Adapun target penyelesaian Tugas Akhir yang dijadikan sebagai tolak ukur pencapaian pekerjaan adalah sebagai berikut:

Tabel 1.1 Jadwal dan *Milestone*.

No.	Deskripsi Tahapan	Durasi	Tanggal Mulai	Tanggal Selesai	<i>Milestone</i>
1	Studi Literatur	4 bulan	1 Oct 2021	31 Jan 2022	Mendapat sumber penelitian
2	Desain Sistem	1 bulan	1 Jan 2022	31 Jan 2022	Perancangan simulasi TA
3	Implementasi <i>Openstack</i>	4 bulan	1 Feb 2022	28 Mei 2022	Mendapat hasil perancangan TA
4	Penyusunan buku TA	2 bulan	18 Mei 2022	18 Jul 2022	Buku TA selesai