

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring berjalannya waktu, *internet* memiliki perkembangan yang sangat pesat dari tahun ke tahun. Layanan yang ditawarkan sudah sangat kompleks dan meluas, seperti layanan aplikasi *web*, layanan media penyimpanan *online*, layanan *cloud computing*, *streaming video*, *game online*, dan masih banyak lagi. Tentu saja dengan berkembangnya *internet* dapat mempengaruhi dampak suatu infrastruktur dan skalabilitas jaringan yang kompleks pula, dimana paket data melewati berbagai banyak trafik yang dikirimkan ke *server* dalam waktu sangat cepat dan bahkan dapat menampung jumlah data yang sangat besar. Dalam hal ini perlu dipertimbangkan jika suatu kasus pada *server* sewaktu-waktu mengalami kondisi *overload* ataupun *down*. Jika *server* mengalami kondisi *down*, maka *server* tidak dapat menjalankan aplikasi-aplikasi yang disediakan oleh *server* tersebut tidak tersedia. Oleh karena itu, perlu dibutuhkan solusi yang tepat untuk mengatasi masalah tersebut dengan cara menggunakan metode *load balancing* dan membangun suatu kluster *server* menggunakan platform *Kubernetes* agar dapat mengurangi penggunaan sumber daya perangkat berlebih dan mampu meringkas konfigurasi terhadap aplikasi agar dapat berjalan dengan baik.

Load balancing merupakan salah satu cara terbaik untuk mengatasi permasalahan tersebut, dimana *load balancing* akan melakukan pembagian beban kerja suatu paket data yang dikirimkan dari klien dengan menggunakan algoritma tertentu agar trafik data yang dilewati akan seimbang, serta mampu meningkatkan skalabilitas dan ketersediaan suatu layanan dari *server*. Setiap algoritma *load balancing* memiliki karakteristik masing-masing. Sebagai contoh penggunaan algoritma *Round Robin*, dimana algoritma *Round Robin* akan melakukan pembagian beban secara bergiliran sehingga membentuk seperti lingkaran. Kemudian pada *Kubernetes* mampu meminimalisir penggunaan *resource* yang berlebih pada *server*, dikarenakan *Kubernetes* menggunakan teknik *Container Orchestration*, dimana aplikasi-aplikasi diisolasi dalam bentuk *container* yang berisi berupa *library* pendukung dan memberikan tingkat konfigurasi dengan mengotomasi dan *management* penggunaan *container*. Tentu saja hal tersebut lebih efisien dalam konsumsi penggunaan sumber daya perangkat seperti *CPU* dan *Memory*

jika dibandingkan dengan penggunaan aplikasi atau *library* yang tertanam secara langsung pada *server* dan *virtual machine*. Aplikasi berbasis *container* bersifat praktis, sehingga dengan mudah untuk dilakukan *deployment* dan dijalankan berbagai macam lingkungan komputasi satu dengan yang lain seperti layanan *cloud computing* ataupun *On Premise*.

Terdapat pada penelitian [1] penggunaan *Load Balancing Web Server* menggunakan teknologi *Software Defined Network (SDN)* dan menggunakan algoritma yaitu *Source IP Hash* dengan jumlah request yang lebih sedikit sehingga tidak memberikan hasil yang maksimal pada pengujian nya. Pada penelitian [2] menggunakan algoritma *Least Connection* dan *Round Robin* sebagai pembandingan dan diterapkan pada *Docker Swarm* sebagai kluster *server*, namun untuk hasil pengujian yang didapat belum maksimal pada *Least Connection*. Pada penilitan [3] menganalisis performansi *Least Connection* dan *IP Hash*, namun tidak melakukan pengukuran dari penggunaan *CPU*, sehingga tidak dapat mengetahui berapa penggunaan *CPU* ketika menggunakan algoritma tersebut. Pada Penelitian [4] dilakukan pembuatan aplikasi layanan web E-Commerce pada platform *docker*, selanjutnya akan dilakukan simulasi untuk melihat performa dari sistem yang telah dikembangkan berdasarkan parameter *throughput*, *response time*, *CPU utilization*, dan *memory utilization*.

Pada Proyek Akhir ini dilakukan dua algoritma sebagai pembandingan, yaitu algoritma *Least Connection* dan *IP Hash* yang akan diimplementasikan pada platform *kubernetes* menggunakan salah satu layanan *cloud computing* yaitu *Oracle Cloud Infrastrucutre*. Kluster tersebut terdiri dari 3 *virtual server* sebagai *worker node* atau *backend server* dimana melakukan *processing* dari layanan aplikasi web Wordpress sekaligus *MySQL* yang diisolasi dalam bentuk objek *Pod* pada masing-masing *worker node*. Selanjutnya dilakukan dengan kombinasi layanan penyimpanan pada *oracle cloud* menggunakan *File Storage* sebagai media penyimpanan data dari aplikasi *pod* pada *kubernetes cluster*. Hal tersebut dilakukan untuk mengurangi penggunaan disk serta dapat mencegah terjadinya hilang data pada saat suatu kasus server tersebut atau *pod* tidak berfungsi dengan baik. Kemudian, dibutuhkan *virtual client* yang terinstall pada *Vmware Workstation* sebagai wadah untuk menjalankan simulasi *load testing* untuk membangkitkan trafik beban *request HTTP* dengan jumlah yang besar menuju ke *load balancer* dengan bantuan aplikasi yaitu *Apache JMeter*. Kemudian *load balancer* melakukan pembagian beban trafik permintaan dari client berdasarkan algoritma yang

digunakan sehingga request tersebut akan diteruskan menuju *worker node* untuk melayani permintaan dari client selanjutnya. Setelah melakukan pengujian pada dua algoritma tersebut, dilakukan analisis lebih lanjut terkait dengan penggunaan algoritma *Least Connctcion* dan *IP Hash* dengan mengukur performa *server* seperti *CPU Utilization* pada masing-masing *worker node*. Karena hal tersebut merupakan metric pengukuran paling penting untuk mengukur kinerja dan dapat menguji performa *server*. Kemudian dilanjut dengan mengukur performa aplikasi yang digunakan dengan pengukuran *response time*, *throughput*, dan *request loss* agar mengetahui seberapa baik masing-masing *worker node* memberikan layanan aplikasi web saat sedang proses permintaan oleh client.

1.2 Tujuan dan Manfaat

Adapun tujuan dari penulisan Proyek Akhir ini, sebagai berikut.

1. Mengimplementasikan perancangan *load balancing* menggunakan algoritma *Least Connection* dan *IP Hash*.
2. Membuat perancangan sistem *load balancing* pada layanan *Cloud Computing* menggunakan platform *Kubernetes*.
3. Mengukur serta menganalisis hasil dari perancangan sistem *load balancing* dan melakukan perbandingan performa terhadap algoritma *Least Connection* dan *IP Hash*.

Manfaat dari penulisan Proyek Akhir ini, sebagai berikut.

1. Dapat merancang dan mengimplementasikan sistem *load balancing* pada layanan *Cloud Computing*.
2. Membandingkan prinsip kerja dari algoritma *Least Connection* dan *IP Hash* pada pengimplementasian *load balancing*.
3. Dapat mengoperasikan layanan *Cloud Computing* beserta fitur-fitur yang ditawarkan dengan baik dan efisien.

1.3 Rumusan Masalah

Adapun rumusan masalah dari Proyek Akhir ini, sebagai berikut.

1. Bagaimana menerapkan infrastruktur jaringan berbasis *Cloud Computing* menggunakan platform *Kubernetes* secara publik?

2. Bagaimana merancang sistem *load balancing* menggunakan *Least Connection* dan *IP Hash*?
3. Bagaimana menentukan kualitas layanan terbaik yang dihasilkan oleh suatu aplikasi menggunakan platform *Kubernetes*?
4. Bagaimanakah hasil akhir analisa yang dihasilkan dari perancangan *load balancing* dengan algoritma *Least Connection* dan *IP Hash* menggunakan platform *Kubernetes*.

1.4 Batasan Masalah

Adapun batasan masalah dari Proyek Akhir ini, sebagai berikut.

1. Menggunakan fitur layanan *load balancer* yang disediakan oleh *Oracle Cloud Infrastructure*.
2. Perancangan dan implementasi menggunakan 3 *worker node* dengan spesifikasi yang terbatas.
3. Fokus terhadap analisa performansi dari aplikasi layanan web *wordpress* dan penggunaan *resource* masing-masing *worker node* pada *Kubernetes*.
4. Hanya menggunakan layanan aplikasi web dengan protokol *HTTP*.
5. Tidak melakukan perancangan keamanan jaringan.

1.5 Metodologi

Adapun metodologi pada penelitian Proyek Akhir ini, sebagai berikut.

1. Studi Literatur

Studi literatur dilakukan dengan mengumpulkan literatur-literatur dan kajian-kajian yang berkaitan dengan permasalahan yang ada pada penelitian Proyek Akhir ini, baik berupa buku referensi, artikel, maupun *e-journal* yang berhubungan dengan topik pembahasan.

2. Perancangan dan Implementasi

Melakukan perancangan sistem dan implementasi dari masing-masing komponen yang digunakan seperti Load Balancer, Kubernetes Cluster, topologi jaringan, dan algoritma yang digunakan.

3. Pengujian dan Analisis

Pengujian dilakukan untuk melakukan ujicoba terhadap sistem yang telah dirancang dengan skenario beserta parameter-parameter pengukuran yang telah

ditentukan. Kemudian, dilakukan analisis lebih lanjut terhadap hasil dari pengujian yang telah dilakukan sesuai dengan skenario yang telah dirancang.

4. Penarikan Kesimpulan

Tahap akhir dari keseluruhan pengerjaan proyek akhir, sehingga dapat ditarik berupa kesimpulan tentang analisis yang telah dilakukan.

1.6 Sistematika Penulisan

Dalam penulisan Proyek Akhir terdiri atas lima bab, dengan keterangan sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini berisi latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah, metodologi penelitian, serta sistematika penulisan.

BAB II DASAR TEORI

Pada bab ini membahas tentang teori pendukung pengerjaan Proyek Akhir, seperti Kubernetes, Teori Load Balancing beserta Algoritma Least Connection dan IP Hash.

BAB III PERANCANGAN SISTEM

Pada bab ini membahas tentang deskripsi Proyek Akhir, alur pengerjaan Proyek Akhir.

BAB IV PENGUJIAN DAN ANALISIS

Pada bab ini membahas tentang pengujian serta analisis dari hasil sistem perancangan Proyek Akhir.

BAB V PENUTUP

Pada bab ini membahas tentang kesimpulan dari pengerjaan Proyek Akhir dan saran untuk pembaca yang akan mengambil penelitian dengan topik yang sama.