

BAB I PENDAHULUAN

1.1 Latar Belakang

Di era perkembangan teknologi digital saat ini, terjadi peningkatan signifikan pengguna layanan berbasis web yang menyebabkan lonjakan *request* pada server. Namun, seiring meningkatnya pengguna yang mengakses, server dapat mengalami kelebihan beban. Kondisi kelebihan beban pada gilirannya dapat menyebabkan penurunan kinerja, waktu respons yang lambat, dan bahkan kegagalan akses bagi pengguna atau yang dikenal juga dengan istilah *overload request* [1]. Perkembangan ini menuntut adanya sistem server yang andal dan efisien yang dapat menangani peningkatan lalu lintas (*traffic*) pengguna tanpa mengorbankan kualitas performa layanan. Salah satu solusi utama untuk mengatasi masalah ini adalah penggunaan mekanisme *load balancing*. *Load balancing* merupakan metode untuk mendistribusikan *traffic* jaringan secara merata di berbagai server, yang bertujuan untuk mengoptimalkan penggunaan sumber daya, memaksimalkan *throughput*, meminimalkan waktu respons, dan meningkatkan ketersediaan sistem secara keseluruhan [2].

Load balancing memiliki beberapa algoritma untuk menentukan cara pendistribusian beban, di antaranya adalah *Least Connection* dan *Round Robin*. Dengan algoritma *Least Connection*, permintaan baru akan diteruskan ke server dengan jumlah koneksi aktif paling sedikit. Sedangkan dengan algoritma *Round Robin*, permintaan didistribusikan dalam grup secara bergantian ke setiap server yang tersedia [2]. Dalam algoritma *Least Connection*, distribusi beban kerja dilakukan berdasarkan pemantauan kondisi aktual setiap server. *Request* berikutnya akan diterima oleh server dengan jumlah koneksi aktif terendah. Sementara itu, algoritma *Round Robin* bekerja dengan konsep antrian sederhana, di mana setiap *request* dari klien didistribusikan secara berurutan ke setiap server yang tersedia tanpa mempertimbangkan beban server saat ini.

Salah satu platform *cloud* terkemuka yang menyediakan layanan *load balancing* ini adalah Microsoft Azure. Layanan *Load Balancer* dirancang untuk mendistribusikan lalu lintas jaringan yang masuk ke berbagai *web server* secara merata [3]. Pentingnya mekanisme semacam ini di lingkungan *cloud* ditekankan oleh kebutuhan untuk mengelola sumber daya virtual secara dinamis dan efisien guna menjamin *Quality of Service* (QoS) yang konsisten bagi pengguna akhir [4]. Layanan ini mendukung berbagai algoritma untuk menentukan bagaimana permintaan diteruskan ke server, termasuk di antaranya adalah *Least Connection* dan *Round Robin*.

Berdasarkan permasalahan tersebut, Tugas Akhir ini menjadi relevan dilakukan guna melakukan perbandingan terhadap performa *load balancing* pada *web server* di Microsoft Azure menggunakan algoritma *Least Connection* dan *Round Robin*. Perbandingan ini akan

difokuskan pada parameter-parameter kunci seperti waktu respons, *throughput*, dan stabilitas koneksi.

1.2 Rumusan Masalah

Adapun rumusan masalah dalam Tugas Akhir ini, sebagai berikut:

1. Bagaimana performa algoritma *Least Connection* dan *Round Robin* dalam *Load Balancing* pada *web server* di Microsoft Azure berdasarkan parameter yang diukur?
2. Sejauh mana algoritma *Least Connection* dan *Round Robin* dapat meningkatkan efisiensi distribusi *traffic* dan menjaga stabilitas koneksi pada *web server* di Microsoft Azure?
3. Apa kelebihan dan keterbatasan algoritma *Least Connection* dan *Round Robin* dalam menangani *traffic* pengguna pada Microsoft Azure?
4. Apa saja perbedaan hasil perbandingan performa *load balancing* dari kedua algoritma?

1.3 Tujuan

Adapun tujuan dari Tugas Akhir, sebagai berikut:

1. Meninjau performa algoritma *least connection* dan *round robin* dalam *load balancing* pada *web server* di Microsoft Azure berdasarkan parameter yang diukur seperti waktu respons, *throughput*, dan stabilitas koneksi.
2. Merancang dan mengonfigurasi sistem supaya beban kerja di distribusi secara efisien.
3. Meninjau kelebihan dan keterbatasan algoritma *Least Connection* dan *Round Robin* dalam menangani *traffic* pengguna di Microsoft Azure.
4. Membandingkan hasil performa *load balancing* dari kedua algoritma.

1.4 Cakupan Pengerjaan

Ruang lingkup pengerjaan Tugas Akhir dibatasi untuk memastikan pembahasan yang fokus dan mendalam. Cakupan pengerjaan ini meliputi:

1.4.1 Platform dan Layanan

Penelitian ini secara khusus menggunakan platform *cloud* Microsoft Azure. Layanan utama yang digunakan adalah *Load Balancer* untuk mengimplementasikan mekanisme *load balancing* pada *web server*.

1.4.2 Algoritma yang ditinjau

Peninjauan performa difokuskan hanya pada dua jenis algoritma *load balancing*, yaitu *Least Connection* dan *Round Robin*.

1.4.3 Parameter Pengujian

Kinerja kedua algoritma diukur dan ditinjau berdasarkan parameter utama seperti waktu respons, *throughput*, dan stabilitas koneksi.

1.4.4 Lingkungan Implementasi

Sistem yang dibangun terdiri dari 3 *Virtual Machine* (VM) yang berperan sebagai *web server* dalam 2 *load balancer* yang berbeda untuk menerima distribusi beban dari *load balancer*.

1.4.5 Klien dan Permintaan

Jumlah klien yang digunakan dalam pengujian hanya dua klien, sehingga jumlah *request* klien yang masuk dari masing-masing klien telah diatur secara spesifik. Kemudian koneksi dan *request* yang diuji untuk setiap algoritma telah ditentukan secara spesifik.

1.5 Tahapan Pengerjaan

Tahapan pengerjaan dari Tugas Akhir ini adalah sebagai berikut:

1.5.1 Studi Literatur

Tahap ini meliputi pencarian dan pengkajian dari referensi ilmiah, baik buku, artikel, jurnal, e-jurnal, maupun internet terkait teknologi yang digunakan. Fokus utamanya adalah konsep *load balancing*, cara kerja algoritma *Least Connection* dan *Round Robin*, serta arsitektur dan layanan pada Microsoft Azure.

1.5.2 Perancangan Sistem

Merancang arsitektur sistem *load balancing* yang akan diimplementasikan di Microsoft Azure. Tahapan ini mencakup penentuan jumlah VM yang akan digunakan sebagai *web server*, konfigurasi *Virtual Network* (VNet), dan skema pengujian untuk kedua algoritma.

1.5.3 Implementasi Sistem

Pada tahap ini mengimplementasikan arsitektur yang telah dirancang. Membuat dan mengonfigurasi VM, menginstal *web server* pada setiap VM, dan melakukan setup *load balancer* untuk mendistribusikan *traffic* menggunakan algoritma *least connection* dan *round robin*.

1.5.4 Pengujian dan Pengambilan Data

Pada tahap ini melakukan pengujian performa pada sistem yang telah diimplementasikan. Pengujian dilakukan dengan memberikan simulasi beban *traffic* ke *web server* melalui *load balancer*. Data mengenai waktu respons,

throughput, dan stabilitas koneksi untuk setiap skenario algoritma dikumpulkan dan dipantau.

1.5.5 Peninjauan Hasil

Meninjau data yang telah dikumpulkan dari hasil pengujian untuk membandingkan performa antara algoritma *Least Connection* dan *Round Robin*. Peninjauan ini bertujuan untuk menentukan efisiensi, kelebihan, dan keterbatasan dari masing-masing algoritma.