

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan infrastruktur digital yang pesat telah menjadikan internet sebagai bagian penting dalam kehidupan manusia setiap hari. Menurut survei dari Statista, pengguna internet harian pada 2021 di seluruh dunia sudah mencapai 4.9 miliar yang artinya sudah hampir dua sepertiga dari populasi global dengan rata-rata penggunaan lebih dari 140 menit. Kegiatan paling populer yang dilakukan pengguna di internet termasuk *instant messaging service*, *video streaming platform*, dan media sosial[1]. Melihat data ini tentu saja membuat *web server* memiliki peran yang penting sebagai pemberi layanan dari pemilik *website* kepada pengguna.

Dengan tingginya akses internet setiap harinya maka para penyedia layanan *web* diharuskan untuk memiliki *web server* yang mampu meladeni permintaan data yang banyak. Jika *web server* yang dipakai tidak mampu menerima lonjakan permintaan data maka akan menyebabkan layanan bermasalah. Masalah tersebut dapat diatasi dengan menggunakan *clustered web server* yang merupakan sekumpulan *server* mandiri yang terintegrasi baik secara *software* maupun *hardware* yang bekerja bersamaan untuk menjalankan layanan *website*. Selain lonjakan permintaan data ada juga masalah ketidakmerataan distribusi beban dalam *server* yang dapat menyebabkan *overload* pada salah satu *server* yang dapat mengurangi tingkat *availabilitas* sebuah *website*.

Untuk meningkatkan tingkat *availabilitas* pada *web server* diperlukan mekanisme *load balancing*. Sebuah *load balancing* yang efisien ditandai dengan penurunan selisih muatan antar *server* dari yang terberat dengan *server* yang teringan dengan memanfaatkan sumber daya yang ada dengan lebih baik. Selain itu, *load balancing* juga dapat mengurangi waktu yang diperlukan untuk melakukan pemrosesan permintaan dan menambah *respond time*[2]. Algoritma *round robin* merupakan algoritma statis yang bekerja dengan cara melakukan pembagian beban secara bergilir dan berurutan sehingga membentuk putaran[3]. Sedangkan untuk algoritma *least connection* merupakan algoritma dinamis yang bekerja dengan memindahkan beban dari *server* dengan beban tertinggi ke *server* dengan beban paling rendah[4]. Docker merupakan sebuah penyedia layanan *container* yang ringan dan mudah dioperasikan. Docker juga mengisolasi tiap *container* yang ada di dalamnya sehingga jika salah satu *container* dirubah

konfigurasinya tidak akan mengubah container lain[5]. Docker juga mampu mengatur banyak *host server* secara bersamaan menggunakan *mode swarm* yang membuatnya sangat cocok digunakan sebagai lingkungan virtual untuk *clustered web server*.

Berdasarkan latar belakang diatas, penulis bertujuan untuk melakukan implementasi *clustered web server* dengan menggunakan Docker Swarm dan melakukan konfigurasi *load balancing* untuk mengukur kinerja dari algoritma *round robin* dan algoritma *least connection* sebagai pertimbangan untuk para penyedia layanan *website* dalam membangun infrastruktur *web server*.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah disampaikan maka dapat dirumuskan masalah yaitu, bagaimana performa algoritma *round robin* dibandingkan dengan algoritma *least connection* sebagai *load balancer* pada *web server* pada lingkungan virtual Docker Swarm.

1.3 Batasan Masalah

Batasan masalah yang digunakan dalam penelitian ini meliputi hal-hal berikut :

1. Implementasi *load balancing* akan dilakukan pada *layer 7* menggunakan algoritma *round robin* dan algoritma *least connection*.
2. *Node server* menggunakan *virtual private server* dan berjumlah sebanyak 3 *node* dengan sistem operasi Ubuntu 20.
3. *Web service* yang digunakan dalam penelitian ini adalah Nginx.
4. Parameter untuk pengujian meliputi *throughput*, *elapsed time*, *response time*, *transcation rate*, dan *cpu utilization*.

1.4 Tujuan Penelitian

Tujuan yang penulis ingin capai dari penelitian ini adalah :

1. Sebagai syarat kelulusan dalam Program Studi Informatika di Universitas Amikom Yogyakarta.

2. Dapat merancang dan melakukan implementasi *web server* berbasis Docker Swarm.
3. Untuk mengetahui hasil dari implementasi algoritma *load balancing* pada *web server* yang telah dirancang.

1.5 Manfaat Penelitian

Manfaat yang ingin penulis dapatkan dari penelitian ini adalah :

1. Manfaat untuk penelitian pengembangan tentang *environment* Docker Swarm sebagai *web server*.
2. Dapat digunakan sebagai referensi untuk optimalisasi sumber daya pada *web server* dengan menerapkan algoritma *load balancing*.

1.6 Sistematika Penulisan

Untuk mempermudah dalam memahami tentang penulisan penelitian, maka penulis telah mengelompokan materi penulisan menjadi lima bab, yaitu :

BAB I PENDAHULUAN

Bab ini memuat latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini akan dibahas mengenai teori-teori dan konsep dasar yang melandasi permasalahan penelitian ini.

BAB III METODE PENELITIAN

Dalam bab ini berisi mengenai gambaran-gambaran umum penelitian dan perancangan dalam membangun sistem.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menjelaskan tentang langkah-langkah dalam implementasi sistem dan proses pengujian yang dilakukan dalam *load balancing* menggunakan algoritma *round robin* dan algoritma *least connection*.

BAB V KESIMPULAN DAN SARAN

Pada bab terakhir ini berisi kesimpulan dan saran yang dapat diberikan oleh penulis berdasarkan hasil penelitian yang telah dilakukan.

