

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Penggunaan akses layanan berbasis web saat ini sangat penting bagi pengguna untuk mendapatkan informasi yang diinginkan secara cepat. Namun, layanan berbasis web sering mengalami kendala dengan beban *server* yang tidak dapat menampung *request* dalam jumlah besar sehingga *server* mengalami *down*. Jika *server down* dan tidak memiliki *backup* web *server* yang menjalankan layanan berbasis web tersebut tidak dapat diakses(Ardhi Permana *et al.*, n.d.).

Dampak yang akan terjadi jika *server down* adalah pengguna yang membutuhkan sebuah informasi didalam layanan basis web tersebut. Untuk menjalankan suatu layanan basis web membutuhkan sebuah *server* yang memadai sehingga pada saat menerima *request* dalam jumlah besar, server mampu mengatasinya(Komaruddin *et al.*, 2019).

Untuk mengatasi beban berlebih pada *server* yaitu dengan cara membuat *server load balancing* sebagai pembagi rata *request* dengan algoritma *round robin* sebelum diteruskan ke *server* dan membuat dua atau lebih *server* berisi *web server* yang sama dan meringankan kinerja web *server* dengan menggunakan di setiap *server* terpasang *docker* untuk meringankan proses kinerja *server*. Jika salah satu *server* ada yang mati maka *server* yang lainnya tetap berjalan(Riskiono and Pasha, 2020a, 2020b).

1.2. Rumusan Masalah

Bagaimana hasil perbandingan nilai *throughput*, *response time*, *request per second*, *downtime* sebelum dan sesudah menerapkan *load balancing* dengan pengujian *request* 5.000, 10.000, dan 15.000 ?

1.3. Batasan Masalah

Implementasi ini hanya menggunakan *load balancing* dengan perbandingan nilai:

- a) *Throughput* merupakan parameter untuk melihat performa jaringan pada kecepatan pengiriman paket yang dilakukan dengan mengirim dan memanfaatkan bandwidth yang ada.
- b) *Response Time* untuk mengukur kecepatan suatu *response* dapat menerima *request* dari klien.
- c) *Request per Second* melihat dengan mengirim paket per satuan detik dari klien yang kemudian dikirimkan ke web server untuk mengetahui kinerja server yang menggunakan sistem *load balancing*.
- d) *Downtime* untuk melihat periode waktu dimana sistem gagal menjalankan fungsi server utama.

1.4. Tujuan Penelitian

Mengetahui dan memperoleh peningkatan kinerja *server* yang menjalankan layanan sistem informasi dengan beban berlebih sebelum mengimplementasikan metode *load balancing* pada server.

1.5. Manfaat Penelitian

1. Memberikan solusi permasalahan dalam mengatasi server yang sering mengalami *down* pada server.
2. Meningkatnya performa kinerja server dengan mengimplementasikan *load balancing* dengan Algoritma *Roun Robin*.