

# BAB 1

## PENDAHULUAN

### 1.1 LATAR BELAKANG

Seiring dengan perkembangan teknologi, jumlah pengguna layanan web semakin meningkat. Hal ini menyebabkan situs – situs web populer memiliki jumlah traffic yang tinggi. Kegiatan atau acara tertentu juga dapat menyebabkan naiknya jumlah traffic web suatu organisasi. Peningkatan jumlah traffic menyebabkan kerja *server* yang melayani permintaan menjadi semakin berat. Akibatnya *server* menurun dan sering terjadi gangguan pada layanan – layanan web tersebut. Hal ini dapat mengakibatkan sistem ataupun situs web tersebut mati/down. Menurut *Thamrin (2011)*, dalam jurnalnya yang berjudul perancangan *tools* berbasis python untuk memantau keaktifan *server*. *Server* yang mati dapat mengganggu layanan yang diberikannya.

Salah satu solusi untuk menangani masalah tersebut adalah dengan menggunakan sebuah *server* yang handal dengan performa yang tinggi. Ren, dkk (2012) berpendapat bahwa teknologi *load balancing* dapat membagi beban permintaan ke beberapa *webservice* sehingga teknologi ini memiliki peranan penting dalam mencapai penggunaan sumber daya bersama yang efektif. *Load balancing* diperlukan untuk efisiensi kegunaan sumber daya komputer dalam sistem paralel dan terdistribusi. Tujuannya untuk membagikan beban secara adil ke seluruh nodes yang ada. Nodes menerapkan algoritma dinamis kedalam rancangannya sehingga *load balancing* dapat merubah algoritma sesuai dengan kondisi yang ada.

*Dita Alamanda Putri (2020)* melakukan Analisis Perbandingan Algoritma *Round robin* dan *LeastConnection* Berbasis *HaProxy* untuk *Load balancing Webservice*. Hasilnya menunjukkan *load balancing* berbasis *HaProxy* menggunakan algoritma *least connection* lebih unggul daripada algoritma *round robin* untuk komponen *throughput*, *request time* dan *request loss* dalam percobaan penelitian sebelumnya berfokus pada membandingkan kinerja algoritma *round robin* dan algoritma *least connection*. Sehingga membuat penulis untuk melakukan penelitian

mengenai Analisis Algoritma Roundrobin dan *Source ip hash* Pada Pengaturan Kinerja *Load balancing Webserver*.

Penelitian ini bertujuan untuk mengimplementasikan *load balancing* pada *webserver* agar dapat menyeimbangkan kinerja sistem. Manfaat dari penelitian ini adalah untuk mengetahui cara kerja algoritma *round robin* dan *source ip hash*.

## 1.2 RUMUSAN MASALAH

1. Berdasarkan latar belakang tersebut, maka dalam penelitian ini penulis merumuskan masalah sebagai berikut :
2. Bagaimana mengimplementasikan *Load balancing* pada dua *webserver* menggunakan aplikasi *HaProxy* dengan algoritma *round robin* dan algoritma *source ip hash*.
3. Menganalisa cara kerja *HaProxy* dengan menggunakan algoritma *round robin* dengan menggunakan algoritma *source ip hash*. Berdasarkan hasil pengujian *CPU Load*, *HaProxy* Statistik, *Byte/Hit*, *Hit/Second*, ping dan *throughput*.

## 1.3 TUJUAN

Berdasarkan rumusan masalah yang telah diuraikan di atas, maka tujuan dari penelitian ini adalah sebagai berikut :

1. Mengetahui kekurangan dan kelebihan penggunaan *load balancing* dengan menggunakan algoritma *round robin* dan algoritma *source ip hash*.
2. Mengimplementasikan *load balancing* pada *router* agar dapat lebih efektif dalam menyetarakan beban *bandwidth* pada *webserver*.
3. Sistem ini dibuat untuk menyelesaikan masalah beban *resource* pada *webserver* dan jika terjadi masalah pada salah satu *webserver*.

## 1.4 MANFAAT

Manfaat yang ingin dicapai dari penelitian ini adalah memberikan solusi untuk menangani masalah pada *webserver* yang tidak stabil dengan menggunakan tiga *webserver* sehingga beban traffic di tiga *webserver* tersebut bisa terjaga keseimbangannya.

## 1.5 BATASAN MASALAH

Untuk menghindari pembahasan yang lebih luas terkait dengan implementasi *load balancing* menggunakan aplikasi *HaProxy* dengan algoritma *round robin* dan algoritma *source ip hash* pada tiga *webserver*, penulis membatasi masalah - masalah yang ada pada penelitian ini, antara lain :

1. Tidak membahas segi keamanan terhadap jaringan.
2. Menggunakan aplikasi *HaProxy* dengan algoritma *round robin* dan algoritma *source ip hash* pada tiga *webserver*.
3. Pengujian menggunakan *apache benchmark* untuk mengetahui *resource server* saat diakses *CPU Load*, *HaProxy* Statistik, *Byte/Hit*, *Hit/Second*, ping dan *throughput*.

