

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Biaya yang relatif rendah dan sarana yang efektif untuk menyebarkan informasi dan komunikasi tentang penggunaan internet, menyebabkan peningkatan trafik yang signifikan [1]. Salah satunya adalah pengguna yang mengirimkan beragam *request* kepada *web server*. Dengan banyaknya *request*, dapat meningkatkan beban kerja *web server* sehingga pengiriman informasi yang diminta menjadi kurang optimal. Beban kerja pada *web server* dapat meningkat pada *server* yang unggul sekalipun karena terjadinya penumpukan *request* pada *server*. *Server* tidak mampu menangani permintaan dari pengguna dan menyebabkan *overload*, *server* akan terputus dan perlu waktu untuk mengkonfigurasi ulang *server* supaya dapat melayani permintaan pengguna. Mengingat *server* yang berfungsi untuk melayani *client* mengharuskan *server* untuk terus menerus melayani permintaan dari *client* [2].

Untuk mengantisipasi kegagalan koneksi ketika *server down* atau pada saat *request* melebihi kemampuan maka membutuhkan *server* dengan ketersediaan tinggi atau *high availability server* yang mampu menuruti semua permintaan pengguna dengan menerapkan konsep *clustering*. Konsep *clustering* merupakan teknologi yang menggabungkan beberapa *server* yang beroperasi dalam waktu yang sama. Dalam hal ini *failover clustering* menggunakan *haproxy* menjadi solusi dari permasalahan tersebut. Ketika *server* mengalami kegagalan (*down*) atau *request* pengguna melebihi kemampuan, *server* cadangan akan mengambil alih fungsi dari *server* utama sehingga pengguna tetap dapat terkoneksi dengan *server* meskipun sistem pada *server* utama berhenti bekerja (*down*) [3]. *Failover* diimplementasikan pada *server*, *server* diduplikasi untuk digabungkan menjadi menjadi satu *cluster* dengan *server* utama [4]. Jenis *failover* yang digunakan yaitu *active pasive failover* karena penelitian ini hanya menggunakan satu *server* aktif yang berfungsi untuk melayani *client*. *Haproxy* bertugas sebagai *server* yang diakses oleh pengguna karena pengguna tidak dapat mengakses secara langsung *web server*. Oleh karena itu *haproxy* akan meneruskan ke *web server*

cadangan yang sedang *standby* saat terjadi kegagalan *server*. Pada tahun 2020, Pribadi, dkk., [5] membuktikan dengan menggunakan *failover clustering* dalam mencapai *high availability*. Penelitian [5] menunjukkan hasil yang bagus dalam hal *availability* ketika *server* sedang *down*. Namun penelitian Pribadi, dkk., menggunakan jenis *server* fisik sehingga, ketika menggunakan *server* fisik akan memakan waktu yang lama untuk konfigurasi. Selain itu *server* fisik membutuhkan perangkat fisik baru untuk menambah kinerja *server*. Di lain sisi, jika menggunakan *server cloud* peningkatan kinerja *server* dapat dilakukan penambahan atau pengurangan spesifikasi perangkat dengan mudah dan fleksibel.

Dari permasalahan yang telah disebutkan, penulis mengambil judul “**ANALISIS KINERJA *FAILOVER CLUSTERING* PADA *WEB SERVER* MENGGUNAKAN *HAPROXY***”. Penelitian ini akan membahas cara kerja *failover clustering* dalam pengambil alihan fungsi *server* utama ke *server* cadangan agar pengguna dapat selalu menikmati layanan. Parameter yang diteliti seperti *availability*, QoS (*Throughput, Packet loss, Delay, Jitter*) dan CPU *usage* pada *server cloud*.

1.2 RUMUSAN MASALAH

Rumusan masalah yang terdapat pada penelitian ini yaitu :

1. Bagaimana performansi dari *failover clustering* saat terjadinya kegagalan pada *server* berdasarkan parameter *availability*, QoS, CPU *usage*?

1.3 BATASAN MASALAH

Batasan masalah yang terdapat pada penelitian ini yaitu:

1. Implementasi menggunakan *failover clustering* menggunakan *haproxy*.
2. Implementasi *web server* menggunakan *apache*.
3. Menggunakan 2 buah *server* yaitu *server* utama dan *server* cadangan.
4. Parameter yang diamati adalah nilai *Availability*, QoS, dan CPU *Usage*.
5. Tidak membahas lebih jauh tentang protokol *routing*.
6. Menggunakan *openstack* sebagai *cloud computing* untuk instalasi *web server* dan Menggunakan *Linux* jenis *ubuntu* sebagai sistem operasi.

1.4 TUJUAN

Tujuan yang terdapat pada penelitian ini yaitu:

1. Menganalisis parameter *availability*, *response time*, *down time*, QoS, dan CPU *usage* pada *web server* berdasarkan *failover clustering* menggunakan *haproxy* pada *server cloud* sehingga dapat memberikan sistem layanan yang terbaik untuk *user*.

1.5 MANFAAT

Manfaat yang diperoleh dari penelitian ini yaitu ketika *server* utama mengalami kegagalan maka dengan adanya *failover* yang berjalan dampak negatif yang dialami pengguna akan berkurang. Pengguna tetap dapat mengakses layanan *web server* meskipun beban kerja pada *web server* meningkat. Mengantisipasi kegagalan koneksi ketika *server down* atau pada saat terjadinya serangan ataupun ketersediaan tinggi karena meningkatnya permintaan akan kebutuhan informasi dalam internet.

1.6 SISTEMATIKA PENULISAN

Penelitian ini terbagi menjadi beberapa bab. Bab 1 yaitu pendahuluan yang berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian. Bab 2 yaitu dasar teori yang membahas tentang kajian pustaka beserta teori-teori yang berkaitan dengan penelitian diantaranya adalah *high availability*, *failover clustering*, *web server*, *server*, *haproxy*, *response time*, *downtime*, *workload*, QoS (*throughput*, *delay*, *packet loss*, *jitter*), *virtual box*, *Openstack*, *apache*, dan *Ubuntu*. Bab 3 yaitu metode penelitian membahas mengenai perangkat keras dan lunak yang digunakan dalam penelitian, alur penelitian, topologi jaringan, dan skenario pengujian dalam penelitian. Bab 4 analisis dan pembahasan membahas mengenai hasil dari pengujian yang dilakukan berdasarkan parameter *availability*, QoS (*throughput*, *delay*, *packet loss*, *jitter*), dan CPU *usage*. Bab 5 penutup membahas mengenai kesimpulan dan saran dari penelitian yang dilakukan.