

BAB I

PENDAHULUAN

1.1. Latar Belakang

Infrastruktur layanan perlu di desain dengan memiliki standar availability dan minim gangguan sehingga layanan dapat berjalan dengan optimal dengan minimal layanan tidak dapat diakses (*downtime*). Layanan *cloud* dapat menanggapi banyak permintaan *request* dalam satu waktu. Ketika sebuah sistem mengalami kenaikan jumlah *request* sampai ribuan per hari dapat menyebabkan kinerja sistem menjadi sangat lambat karena kelebihan beban (*overload*), hal ini dapat menimbulkan banyak permasalahan, salah satunya adalah keluhan dari sisi pengguna (Ashari & others, 2016).

Penggunaan server dengan sistem terdistribusi membutuhkan suatu metode agar dapat membagi beban dengan merata disetiap server. Berbagai penelitian telah dilakukan untuk mengatur pembagian beban kerja pada server klustering agar dapat mengoptimalkan kinerja sistem yaitu dengan menerapkan metode load balancing. Penerapan load balancing dalam web server sangat penting dan dapat menjadi solusi dalam menangani beban server yang sibuk sehingga dapat meningkatkan skalabilitas pada sistem terdistribusi (Lukitasari & Oklilas, 2013).

Konsolidasi server adalah sebuah proses dalam melakukan sentralisasi beban kerja komputasi untuk mengurangi biaya, kompleksitas, jalur komunikasi, manajemen biaya dan untuk mengoptimalisasi dan simplikasi infrastruktur IT yang sedang berjalan dan untuk menghasilkan solusi investasi dan implementasi (Dutta & Mia, 2010). Lebih detail lagi konsolidasi server bertujuan untuk meningkatkan efisiensi pemanfaatan sumber daya komputer dengan salah satu caranya yaitu mengurangi jumlah server. Dengan konsolidasi server maka semua fungsi yang sebelumnya ditangani oleh beberapa server yang berbeda akan ditangani oleh sebuah server dengan kapasitas yang lebih besar. Namun dalam pengimplementasian konsolidasi server ini membutuhkan konfigurasi yang kompleks sehingga cukup menyulitkan bagi yang baru memulai menerapkannya. Salah

satu solusi dalam mengatasinya adalah menggunakan virtualisasi server karena mudah untuk diimplementasikan (Eisen, 2011).

Tidak seperti virtualisasi mesin, *docker container* tidak menggunakan *hardware* untuk virtualisasi. Program berjalan dalam *docker container* berhubungan langsung dengan *linux kernel* pada *host operating system*. Karena tidak ada tambahan lapisan antara program yang berjalan didalam *container* dengan sistem operasi pada komputer, sehingga tidak ada *resources* yang habis karena redundansi aplikasi atau simulasi *virtual hardware*. *Container* memungkinkan mengisolasi lingkungan program, sehingga program dapat berjalan tanpa gangguan dari permasalahan di sistem operasi. Dalam perkembangannya, kemudahan pengelolaan *container* menjadi basis untuk melakukan skalabilitas (Nickoloff, 2016).

Container sangat mungkin untuk melakukan proses *load balancing* dengan kemampuan dan kemudahannya. Fokus utama dalam penelitian ini adalah menganalisis kinerja sebuah perangkat raspberry pi yang menerapkan teknologi *container* untuk memberikan solusi *load balancing*.

1.2. Rumusan Masalah

Rumusan masalah yang dikaji adalah bagaimana rancangan arsitektur Raspberry Pi untuk kebutuhan container cluster, dan bagaimana performa cluster Raspberry Pi dalam *load balancing* web server.

1.3. Batasan Masalah

Adapun batasan-batasan masalah pada penelitian ini adalah :

1. Analisis yang dilakukan berupa analisis praktik/simulasi
2. Pembuatan cluster melibatkan Raspberry Pi Model B
3. Platform yang digunakan untuk implementasi adalah docker
4. Pengujian tidak membahas masalah keamanan.
5. Implementasi *load balancing* webfarm pada raspberry pi.
6. Parameter yang dianalisa untuk mengetahui perbaikan kinerja setelah implementasi *load balancing* adalah request http, memori dan processor.

1.4. Tujuan Penelitian

Mengimplementasikan dan mengembangkan infrastruktur Raspberry Pi di lingkungan STMIK AKAKOM untuk mengetahui performa dari cluster Raspberry Pi yang dibangun sebagai *load balancer*.

1.5. Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah diharapkan dapat menjadi bahan referensi untuk meningkatkan skalabilitas dari webserver, sehingga dapat meningkatkan akses (*request*) terhadap server. Hal ini dapat memungkinkan sistem informasi yang berjalan diatas webserver dapat meningkat secara ketersediaan layanannya.

1.6. Target Luaran

Target luaran dari penelitian ini adalah: *load balancing* web server dalam bentuk dokumentasi dan model perancangan infrastruktur dengan memanfaatkan teknologi *container docker*.