

BABI

PENDAHULUAN

1.1 Latar Belakang Masalah

Docker adalah alat yang dirancang untuk memudahkan pengembang membuat, menyebarkan, dan menjalankan aplikasi dengan menggunakan kontainer. Memungkinkan pengembang untuk menerapkan aplikasi sebagai satu paket, yang terdiri dari semua pustaka yang diperlukan, alat sistem, dan kode. Sekarang pengembang dapat yakin bahwa aplikasi akan berjalan di komputer lain (Docker, 2021). Lingkungan eksekusi aplikasi berbagai kernel sistem operasi tetapi dijalankan secara terpisah satu sama lain. *Docker* telah menjadi standar *de-facto* dalam hal implementasi berbasis kontainer.

Dari implementasi skala kecil hingga aplikasi perusahaan skala besar, Teknologi *Docker* menggunakan kernel Linux dan fitur-fiturnya seperti *Cgroups*, yang mengatur isolasi dan penggunaan sumber daya sistem, seperti CPU dan memori, untuk sekelompok proses dan ruang nama, yang membungkus satu set sumber daya sistem dan menyajikannya ke suatu proses agar terlihat seperti mereka berdedikasi pada proses itu untuk memisahkan proses agar dapat berjalan secara mandiri. Kemandirian ini adalah tujuan kemampuan container untuk menjalankan banyak proses dan aplikasi secara terpisah dari satu sama lain untuk memanfaatkan infrastruktur dengan lebih baik sambil tetap mempertahankan keamanan yang dimiliki dengan sistem terpisah.

Sebagian besar aplikasi bisnis terdiri dari beberapa komponen terpisah diantaranya: server web, *database*, *cache* dalam memori. Wadah yang digunakan memungkinkan untuk menyusun bagian-bagian ini menjadi unit fungsional individu. Setiap bagian dapat dipelihara, diperbarui, ditukar, dan dimodifikasi secara independen dari yang lain. Ini pada dasarnya adalah model layanan mikro dari desain aplikasi. Dengan membagi fungsionalitas aplikasi menjadi layanan mandiri yang terpisah. Wadah yang ringan dan *portable* memudahkan pembuatan dan pemeliharaan aplikasi berbasis layanan mikro.

Teknologi virtualisasi telah mengubah arah revolusi industri komputer dengan cara penurunan biaya modal dan biaya operasi. Ketersediaan suatu virtualisasi juga akan meningkatkan ketersediaan layanan yang lebih tinggi dan mekanisme perlindungan suatu data (Evans, 2018). Dalam penerapan virtual ini akan menjadikan sebuah kemudahan untuk menjalankan dan penerapan suatu mekanisme implementasi *hosting* menggunakan *container* di suatu virtualisasi aplikasi.

Virtualisasi merupakan sebuah komputer yang menjalankan sebuah sistem operasi secara bersamaan. Aplikasi yang berjalan di mesin virtual dapat berjalan seolah-olah mereka berada di dalam mesin khusus yang mereka sendiri, di mana sistem operasi, *libraries*, dan program lainnya bersifat unik dan untuk sistem virtualisasi tidak terhubung ke sistem operasi host yang berada di bawahnya (Dimas Setiawan Afis, 2019).

Hosting adalah suatu website yang akan menampilkan di dunia maya dengan cara mengalokasikan ruang pada web server dan menyimpan *file-file* yang akan digunakan pada web tersebut agar dapat dilihat secara online. *hosting* yang

tepat berarti memiliki akses ke alokasi sumber daya yang tepat untuk menjaga situs web Anda memuat dengan cepat dan andal bagi pengunjung.

Dalam penelitian ini, penulis bermaksud untuk melakukan analisa mendalam mengenai opini dalam penggunaan *hosting* pada *container*. Dengan menggunakan *Docker container* yang telah disediakan dan dapat diklasifikasikan dan divisualisasikan. Nantinya, hasil dari itu akan diterapkan pada Sistem Operasi Linux Ubuntu Server 18.04 LTS di *Virtual Private Server* dengan IP *public*.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas dapat dirumuskan permasalahan yang dihadapi yaitu:

- 1) Bagaimana cara konfigurasi system didalam sebuah Docker container?
- 2) Bagaimana proses konfigurasi dan efisiensi pembangunan webserver yang diterapkan pada Docker container?
- 3) Bagaimana implementasi reverse proxy pada pembangunan system multiple web server pada Docker container?
- 4) Bagaimana cara melakukan checking performance webserver?

1.3 Batasan Masalah

Batasan masalah yang akan dibahas dalam cakupan pembangunan konfigurasi Reverse Proxy di Container adalah :

- 1) Konsep Konfigurasi pada jaringan public
- 2) Sistem dibangun diatas Docker dan menggunakan citra Ubuntu 20.04 LTS.

- 3) Perencanaan konfigurasi yang menerapkan reverse proxy di sebuah container.
- 4) Memberikan management Docker Container untuk sebuah monitoring kontainer yang sedang berjalan.
- 5) Skema pengujian analisis performance load yang menggunakan hosting biasa dengan yang menggunakan docker container dan reverse proxy.

1.4 Tujuan

Adapun tujuan dari penelitian yang dilakukan oleh penulis dengan judul “Membangun Konfigurasi Reverse Proxy Apache Web Server Di Docker Container”:

- 1) Meringankan beban kinerja sebuah server utama
- 2) Penerapan Docker Container untuk memberikan kemudahan dalam membangun aplikasi didalam sebuah Docker container.
- 3) Implementasi metode reverse proxy untuk penerapan system pembangunan multiple webservers di dalam sebuah container.

1.5 Manfaat

Adapun manfaat yang dapat diambil dari pembangunan system “Membangun Konfigurasi Reverse Proxy Apache Web Server Di Docker Container” sebagai berikut :

1. Bagi Penulis.

Manfaat yang diperoleh dalam melakukan pembangunan system multiple webserver yang menerapkan metode reverse proxy di dalam docker container adalah sebagai berikut :

- a. Lebih mengetahui tentang teknologi container yang sedang banyak digunakan oleh instansi – instansi dikalangan besar.
- b. Sebagai penerapan mengimplementasikan pengetahuan yang didapatkan dalam mata kuliah jaringan selama perkuliahan.
- c. Sebagai sarana penyelesaian tugas akhir atau skripsi.

2. Bagi Universitas

- a. Mengetahui kemampuan mahasiswa dalam menguasai materi yang telah diberikan selama proses perkuliahan.

3. Bagi Universal

- a. Dapat diterapkan dalam sebuah perkantoran dan instansi lainnya.
- b. Konsep dalam sebuah container memberikan kemudahan dalam konfigurasi dari beberapa aplikasi yang akan digunakan dalam sebuah instansi.
- c. Memberikan sebuah keamanan yang diberikan oleh konfigurasi reverse proxy dan container tersebut.