

I PENDAHULUAN

1.1 Latar Belakang

Seiring berkembangnya teknologi dan informasi komputer, perangkat lunak berbasis *web* berkembang sangat pesat. Hanya dengan menggunakan *web browser*, aplikasi ini dapat dengan mudah diakses dari berbagai macam perangkat komputer. Oleh karena itu, kemudahan proses *deployment* aplikasi *web* seperti *database*, *dependency*, dan *environment* ke *server* sangat dibutuhkan.

Secara umum, teknik *deployment* aplikasi *web* terbagi menjadi dua. Pertama, dengan cara melakukan instalasi aplikasi *web* beserta *dependency*-nya ke dalam *server*. Teknik ini terbilang sederhana, cepat, dan mudah. Namun, teknik ini mempunyai kendala jika men-*deploy* beberapa aplikasi karena setiap aplikasi mempunyai *dependency* dengan paket versi tertentu. Hal ini dapat menyebabkan terjadinya kesalahan atau *bug* karena terdapat aplikasi atau paket aplikasi yang bergantung pada aplikasi lain. Masalah tersebut dikenal dengan istilah *dependency hell*.

Kendala tersebut dapat diatasi dengan menerapkan teknik kedua, yaitu dengan cara melakukan virtualisasi *server*. Virtualisasi akan membungkus setiap aplikasi beserta *dependency*-nya ke dalam sebuah *virtual machine*. Dengan demikian, skalabilitas dapat meningkat karena setiap *virtual machine* berjalan menggunakan sumber daya yang berbeda dan tidak saling memengaruhi. Namun, virtualisasi berbasis *hypervisor* ini membutuhkan sumber daya yang tidak sedikit karena setiap *virtual machine* menjalankan *kernel* dan *operating system* yang terpisah dari *host*. Untuk mengatasi kekurangan tersebut, dibutuhkan teknologi yang dapat melakukan virtualisasi secara ringan sehingga mampu memenuhi kebutuhan aplikasi dengan sumber daya yang minimal. Teknologi virtualisasi ini menggunakan *container* sebagai basisnya. Salah satu perangkat lunak yang menerapkan *container* adalah *docker*.

Docker merupakan salah satu perangkat lunak populer yang mengimplementasi konsep *container*. Berbeda dengan *virtual machine*, *docker* tidak melakukan instalasi *operating system* di dalam *container*. *Docker* menerapkan konsep *sharing-kernel* yang mana menggunakan *kernel host* sehingga membuat penggunaan sumber daya menjadi efektif dan efisien. (Sardi 2017)

PT GLOBAL TEKNO INTERNUSA mempunyai *service* yang berjalan dalam virtualisasi berbasis *hypervisor*. Untuk beberapa aplikasi *web*, PT GLOBAL TEKNO INTERNUSA masih menggunakan *shared-hosting*. PT GLOBAL TEKNO INTERNUSA mempunyai rencana untuk melakukan migrasi semua aplikasi *web* dari *shared-hosting* ke *local server* dan diterapkan dalam virtualisasi. Dalam virtualisasi *virtual machine*, proses *boot* akan terasa lama karena virtualisasi akan menjalankan *guest operating system* sebanyak *virtual machine* yang berjalan. Untuk mengatasi hal tersebut, penggunaan *virtual machine* dapat digantikan dengan *container*, salah satunya dengan menerapkan *docker*. Dengan diterapkannya *docker*, diharapkan mampu mempercepat proses *boot* dan meningkatkan kinerja *server* dalam menjalankan virtualisasi.

1.2 Rumusan Masalah

- Bagaimana cara melakukan virtualisasi *server* menggunakan *docker*?



2

- b. Bagaimana kinerja *server* setelah *docker* berhasil diimplementasi?

1.3 Tujuan

- a. Dapat melakukan virtualisasi *server* berbasis *container* dengan *docker*.
b. Dapat membuktikan bahwa dengan menggunakan *docker*, kinerja *server* menjadi lebih ringan.

1.4 Ruang Lingkup

- a. Menggunakan Linux Ubuntu versi 20.04 LTS dan Windows 10 sebagai *operating system*.
b. Menggunakan *Docker Hub* sebagai *container registry*.
c. Menggunakan Apache JMeter dan *stacer* sebagai alat pengujian.
d. Melakukan *deploy* aplikasi *web* beserta *dependency* ke dalam *docker container*.
e. Melakukan perbandingan nilai penggunaan *physical resources* CPU dan RAM antara *docker* dan *VirtualBox*.
f. Proses penerapan *docker* hanya sebatas virtualisasi dan membandingkan nilai penggunaan *physical resources*, tidak mencakup analisis hasil pengujian dan *hardening* ataupun *security*.

