

# BAB I PENDAHULUAN

## I.1 Latar Belakang

Perkembangan internet berjalan beriringan dengan perkembangan teknologi *software* dan juga *hardware*. Hal ini diperlukan agar para pengguna mendapatkan layanan dengan lebih mudah dan lebih cepat. Para pengembang dan tenaga ahli IT terus berinovasi dalam mewujudkan sebuah sistem dan layanan yang dapat diandalkan. Salah satu inovasi tersebut ialah munculnya teknologi virtualisasi. Teknologi virtualisasi yang telah banyak diterapkan yaitu *virtual machine* dan *container*.

Docker merupakan sebuah platform aplikasi yang mewadahi aplikasi dan komponen-komponen pendukungnya di dalam sebuah *container*. Docker membantu para *developer* dalam proses pengembangan aplikasi dengan teknologi *container* untuk dapat membangun, mengemas, dan menjalankan aplikasi secara lengkap beserta komponen lainnya yang dibutuhkan untuk menjalankan sebuah layanan atau aplikasi (Docker, 2021). Teknologi *Container* pada Docker membuat aplikasi dan layanan menjadi efisien sehingga banyak aplikasi maupun platform lebih memilih untuk menggunakan *Container* seperti untuk *web service*, *big data*, *internet of things* dan lain sebagainya. Dengan meningkatnya penggunaan *container*, diperlukan platform yang dapat mengelola banyak *container* dalam menjalankan *workload* aplikasi. Dari kebutuhan ini, muncul *container orchestration* yaitu Kubernetes.

Kubernetes merupakan platform manajemen *workload* untuk aplikasi yang dikontainerisasi. dan menyediakan konfigurasi otomatis secara deklaratif. Kubernetes melakukan *deployment* kontainer ke *server* secara otomatis dan meningkatkan produktifitas dalam tim pengembang aplikasi. Dengan begitu, Kubernetes merupakan solusi dari masalah *deployment* aplikasi secara manual ke *server* yang memakan waktu lebih banyak dan sulit dalam urusan skalabilitas (Kubernetes, 2021).

Kubernetes dapat menambah dan mengurangi sumber daya yang dibutuhkan untuk menjalankan aplikasi dengan mudah dan cepat dengan fitur *autoscaling*.

Salah satu fitur *autoscaling* pada kubernetes ialah *horizontal pod autoscaler*. *Horizontal pod autoscaler* akan melakukan penyesuaian otomatis pada *workload* untuk memenuhi permintaan atau kebutuhan aplikasi berdasarkan metrik-metrik tertentu.

Di samping itu, menurut (Chang et al., 2017), kubernetes dapat secara dinamis memantau kebutuhan sumber daya dan penggunaan aplikasi yang sedang berjalan, dan kemudian menyesuaikan sumber daya yang disediakan ke kontainer yang dikelola. Dengan begitu, infrastruktur yang digunakan oleh kubernetes dapat mengakomodasi layanan aplikasi sesuai dengan jumlah permintaan pengguna.

Dalam penelitian ini, akan diimplementasikan sebuah sistem kontainer kubernetes pada layanan *cloud*. infrastruktur yang dibangun akan menggunakan perangkat dan layanan dari penyedia *cloud*. Setelah melakukan implementasi, dilakukan analisis dan pengujian terhadap fungsi *scale* pada *horizontal auto scaling* (HPA). Pengujian fungsi *scale* akan mengacu kepada beberapa parameter dan metrik seperti jumlah *pod*, jumlah transaksi, *transaction rate*, *response time*, *longest transaction* dan *shortest transaction*.

## **I.2 Perumusan Masalah**

Rumusan masalah yang mendasari penelitian ini adalah:

- a. Bagaimana implementasi aplikasi *web* pada *cluster* Kubernetes ?
- b. Bagaimana cara melakukan *profiling* aspek skalabilitas pada *cluster* Kubernetes?
- c. Bagaimana meningkatkan skalabilitas pada *cluster* Kubernetes ?

## **I.3 Tujuan Penelitian**

Berdasarkan rumusan masalah yang telah dikaji, tujuan dari penelitian ini adalah sebagai berikut:

- a. Implementasi aplikasi *web* pada *cluster* Kubernetes non-HPA dan *cluster* Kubernetes HPA dalam Lingkungan Google Kubernetes Engine.
- b. Melakukan *profiling cluster* Kubernetes non-HPA dan *cluster* Kubernetes HPA melalui analisis hasil metode *load testing* dengan parameter jumlah

*pod*, jumlah transaksi, *transaction rate*, *response time*, *longest transaction* dan *shortest transaction*.

- c. Meningkatkan skalabilitas *cluster* kubernetes dengan implementasi *Horizontal Pod Autoscaler* pada *cluster* Kubernetes dengan menerapkan konfigurasi utilisasi CPU.

#### **I.4 Batasan Penelitian**

Adapun batasan dari penelitian ini adalah sebagai berikut:

- a. Penelitian ini menggunakan platform simulasi pada *Google Cloud Platform* dan tidak membahas struktur aplikasi.
- b. Penelitian ini berfokus pada pengukuran kualitas fungsi *scale* dengan HPA dengan beberapa parameter seperti *pod number*, *transactions*, *transaction rate*, *average response time*, *longest transaction* dan *shortest transaction*.

#### **I.5 Manfaat Penelitian**

Manfaat penelitian ini adalah sebagai berikut :

- a. Manfaat Teoritis

Penelitian ini dapat berkontribusi menambah pengetahuan tentang fungsi *autoscaler* dalam aspek skalabilitas *cluster* Kubernetes.

- b. Manfaat Teknis

Penelitian ini dapat memberi gambaran teknis penggunaan fitur *autoscaling* pada *cluster* Kubernetes.

#### **I.6 Sistematika Penulisan**

Sistematika penulisan yang digunakan pada penelitian ini adalah :

### **Bab I**

### **Pendahuluan**

Bab ini menjelaskan latar belakang penelitian, rumusan masalah, tujuan penelitian, batasan penelitian, manfaat penelitian dan sistematika penulisan.

