

# BAB I PENDAHULUAN

## I.1 Latar Belakang

Sebagai wadah dari sebuah layanan *Content Management System* (CMS), *cloud* menjadi salah satu pilihan untuk membangun layanan CMS dan menyimpan data dari layanan CMS yang dibuat. Berdasarkan hasil survei yang dilakukan oleh Flexera, mengenai tren *cloud computing* 2020. Google Cloud Platform (GCP) menjadi salah satu dari tiga *platform cloud* yang paling banyak digunakan, dan paling besar peningkatannya dari tahun 2019 ke tahun 2020. Tingkat adopsi Google naik sebesar 70%, peningkatan yang cukup besar dari 2019 (Flexera, 2020). GCP merupakan salah satu layanan yang menyatukan beberapa layanan *cloud computing* yang ditawarkan oleh Google. Berdasarkan data dan hasil survei flexera mengenai tren cloud computing 2020. Organisasi semakin memanfaatkan banyak layanan *Platform as a Service* (PaaS) dari penyedia *cloud*. Tiga teratas adalah *Relational Database as a Service* (DBaaS), *container-as-a-service* dan *data warehouse* (Flexera, 2020). Organisasi mendorong pergerakan ini karena meningkatnya minat mereka dalam memanfaatkan *container* untuk mempercepat *deployment*, skala operasi, dan meningkatkan efisiensi beban kerja yang berjalan di *cloud* (Flexera, 2020).

Untuk mengoptimalkan *container* dibutuhkan *tools* untuk *management container*, salah satunya adalah Kubernetes. Kubernetes merupakan *platform open-source* yang diperkenalkan oleh Google pada tahun 2014 untuk mengelola aplikasi *container* di *cluster machines* (Jawarneh, 2019). Kubernetes dipilih karena Kubernetes adalah *tools* paling banyak tersebar dipasar (Jawarneh, 2019). Kubernetes didasarkan pada pola arsitektur *master/slave*. *Master node* mewakili bidang kontrol *cluster*, dan direplikasi untuk menjamin *high availability* dan toleransi kesalahan mengeksploitasi *scheduler layer* (Jawarneh, 2019). *Node slave* (dikenal sebagai *minions*) adalah *node – node* tempat aplikasi *container* dieksekusi (Jawarneh, 2019). Arsitektur ini membantu *developer* untuk melakukan *monitoring*, dan *scheduling* terhadap aplikasi yang berada dalam *container*. Selanjutnya jika layanan CMS yang diakses oleh *client* bersifat *public cloud*, maka

dibutuhkan suatu wadah untuk membantu seorang *developer* dalam melakukan pengembangan layanan CMS sebelum mulai mengimplementasikannya pada *public cloud*. VMware menjadi salah satu *platform* yang bisa dipilih oleh *developer*.

Tentu pada setiap layanan perlu adanya *maintenance*, baik untuk mengoptimalkan performa layanan atau penyebab lainnya. Ini menyebabkan layanan CMS harus dimatikan untuk sementara waktu, hal ini tentu akan memberikan *downtime* yang beragam sesuai dengan lamanya waktu *maintenance*. Terkadang adanya suatu *disaster* dapat menyebabkan layanan CMS tidak dapat diakses dan data dari layanan CMS tersebut hilang karena terjadinya *disaster*. *Backup* dan *restore* merupakan suatu solusi yang dapat membantu proses *maintenance* agar *downtime* yang terjadi dapat dipersingkat, serta dapat berguna ketika terjadi sebuah *disaster*. *Backup* biasanya merupakan duplikat data yang diperlukan jika terjadi kehilangan atau pemulihan data (Alshammari dkk., 2016). *Restore* merupakan proses mengembalikan kembali sebuah data atau *file* ke tempat semula (Solomon, 2014).

Dalam tugas akhir ini dilakukan penelitian untuk meningkatkan mekanisme *backup* dan *restore* dari *platform* GCP ke *platform* VMware, begitupun sebaliknya. Peneliti menggunakan *Docker* sebagai *container*, dan Kubernetes sebagai *container orchestration*. Menggunakan Velero sebagai alat bantu untuk melakukan *backup* dan *restore*, NFS sebagai protokol untuk mendistribusikan data milik layanan CMS agar bisa diakses oleh kedua *platform*, serta menggunakan Pritunl sebagai alat untuk membuat jaringan *Virtual Private Network* (VPN) agar *platform* VMware bisa terhubung dengan NFS di GCP.

## **I.2 Rumusan Masalah**

Berdasarkan uraian permasalahan yang telah dijelaskan pada latar belakang, berikut merupakan permasalahan yang dikaji dalam penelitian ini adalah sebagai berikut:

1. Bagaimana mendesain suatu sistem yang dapat menunjang mekanisme *backup* dan *restore* yang berfungsi untuk menyediakan layanan *high availability* serta menjadi upaya mengatasi *disaster* dengan mekanisme *backup* dan *restore*?

2. Bagaimana cara melakukan mekanisme *backup* dan *restore* pada *platform* yang berbeda?
3. Bagaimana mengetahui nilai *Quality of Service* pada mekanisme *backup* dan *restore* pada kedua *cluster*?

### **I.3 Tujuan Penelitian**

Adapun tujuan yang ingin dicapai dari penelitian ini adalah sebagai berikut:

1. Mengimplementasikan mekanisme *backup* dan *restore* untuk mengurangi *downtime* yang terjadi serta mengupayakan pencegahan ketika terjadi *disaster*
2. Mengimplementasikan Velero sebagai *tool* untuk melakukan *backup* dan *restore* terhadap suatu layanan
3. Mengidentifikasi nilai *delay*, *packet loss*, *throughput*, dan *jitter* untuk melakukan *backup* dan *restore*

### **I.4 Manfaat Penelitian**

Manfaat dari penelitian ini diharapkan meliputi dua aspek yaitu:

1. Manfaat untuk mahasiswa dan universitas yaitu peneliti berharap hasil dari penelitian ini dapat memberikan kontribusi untuk mahasiswa lainnya yang memiliki masalah atau penelitian yang berhubungan dengan permasalahan yang diteliti oleh peneliti
2. Manfaat dari aspek keilmuan dan industri penyedia layanan yaitu peneliti berharap hasil dari penelitian ini dapat berkontribusi untuk menunjang pembangunan atau pengembangan infrastruktur layanan berbasis *container*, sehingga dapat membantu mengurangi *downtime* yang terjadi serta bisa menjadi upaya pencegahan ketika terjadi *disaster*

### **I.5 Batasan Penelitian**

Adapun batasan dalam penelitian ini sebagai berikut:

1. Implementasi *container* menggunakan *tool open source* yaitu Docker serta menggunakan Kubernetes sebagai *container orchestration*
2. Terdapat dua *platform* yang digunakan, yaitu Google Cloud Platform dan VMware Workstation

3. Solusi untuk mekanisme *backup* dan *restore* menggunakan *third party* yaitu Velero serta terbatas pada objek yang di-*backup* atau di-*restore* oleh Velero saja, tidak dengan data dari layanan yang dibuat
4. *Object store* terbatas pada pengimplementasian GCP *bucket*

## **I.6 Sistematika Penulisan dan Bahasa**

Sistematika penulisan dan bahasa terbagi menjadi beberapa bab dari pokok pembahasan, secara umum dapat dijabarkan sebagai berikut:

1. BAB I – Pendahuluan, bab ini berisi penjelasan mengenai latar belakang, rumusan masalah, tujuan penelitian, batasan penelitian, dan manfaat penelitian penulis tentang desain dan analisis mekanisme *backup* dan *restore* pada Kubernetes untuk layanan CMS berdasarkan aspek performa jaringan, dan juga pada bab ini berisi sistematika penulisan dan bahasa.
2. BAB II – Landasan Teori, berisi penjelasan kajian – kajian literatur pendukung untuk riset dan beberapa *related work* yang pernah dilakukan oleh peneliti sebelumnya.
3. BAB III – Metodologi Penelitian, berisi penjelasan mengenai langkah-langkah dan tahapan yang dilakukan untuk menyelesaikan permasalahan dengan gambaran konseptual serta sistematika penelitian sebagai pemecahan masalah.
4. BAB IV – Analisis, berisi tentang model dari penelitian yang akan dilaksanakan, serta analisis hasil simulasi penerapan riset.
5. BAB V – Implementasi dan Pengujian, berisi tentang implementasi pembuatan logika, pengujian, menganalisa dari hasil analisis yang sudah dilakukan.
6. BAB VI – Kesimpulan dan Saran, berisi mengenai kesimpulan dari hasil penelitian berdasarkan pengolahan dan analisis data yang menjawab tujuan awal dari penelitian ini.