

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Dari penelitian yang dilakukan, terdapat dua poin kesimpulan yang dapat diambil dari hasil penelitian.

1. Analisis pembuatan *container* pada *platform* GCP, AWS, dan Azure.

Pada penelitian ini menghasilkan analisis pembuatan *container* pada *platform* GCP, AWS, dan Azure yang dibangun dengan beberapa tahapan yaitu pembuatan Kubernetes *cluster*, *pen-deploy-an* aplikasi dengan *yaml file*, dan *instalasi* Prometheus & Grafana sebagai *software* yang membantu untuk mengumpulkan parameter dan data pengujian.

2. Perbandingan performa *container* pada Kubernetes *service* di *platform* GCP, AWS, dan Azure.
  - a. GCP adalah *platform* Kubernetes terbaik dalam penelitian ini, karena memiliki performa yang unggul dalam hal *execution time*, efisiensi memori, efisiensi jaringan, dan *performance per price*. GCP menunjukkan kinerja yang cepat dan hemat *resource* dalam menjalankan aplikasi yang *dicontainerisasi* di lingkungan Kubernetes.
  - b. AWS adalah *platform* Kubernetes terburuk dalam penelitian ini, karena memiliki performa yang buruk dalam hal *execution time*, efisiensi memori, efisiensi jaringan, dan *performance per price*. AWS menunjukkan kinerja yang lambat dan boros sumber daya dalam menjalankan aplikasi yang *dicontainerisasi* di lingkungan kubernetes.
  - c. Azure adalah *platform* Kubernetes yang moderat dalam penelitian ini, karena memiliki performa yang seimbang dalam hal *execution time*, efisiensi memori, efisiensi jaringan, dan *performance per price*. Azure menunjukkan kinerja yang cukup baik dan cukup hemat sumber daya dalam menjalankan aplikasi yang *dicontainerisasi* di lingkungan Kubernetes.

Maka, berdasarkan hasil dari pengujian, *platform* Kubernetes *service* terbaik ditempati GCP dikarenakan unggul dalam berbagai aspek performa.

## 5.2 Implikasi

Setelah melakukan analisis perbandingan performa *container* pada ketiga CSPs, maka dapat dikemukakan beberapa implikasi sebagai berikut:

### 1. Seleksi *Platform*

Dengan adanya analisis performa dapat membantu suatu organisasi dalam memilih *platform cloud* yang paling cocok untuk diterapkan sesuai dengan kebutuhan mereka.

### 2. Optimalisasi *Cost*

Suatu organisasi dapat mengidentifikasi CSPs mana yang menawarkan keseimbangan terbaik antara performa dan biaya untuk aplikasi dalam *container* mereka yang memiliki potensi menghasilkan penghematan yang signifikan.

### 3. Alokasi *Resource*

Dengan adanya analisis ini dapat mengungkapkan bagaimana perbedaan platform saat mengkonfigurasi Kubernetes memengaruhi penggunaan *resources* seperti CPU, *memory*, dan *storage*. Informasi ini akan membantu suatu organisasi mengoptimalkan alokasi *resources*.

## 5.3 Rekomendasi

Rekomendasi untuk penelitian selanjutnya adalah melakukan *benchmarking* pada setiap CSPs, *deploy* aplikasi lebih beragam, ataupun membandingkan Kubernetes yang berjalan pada *bare-metal server* dan CSPs. Hal ini akan memberikan hasil yang berbeda.

Rekomendasi lainnya adalah dalam penggunaan CSPs lebih baik segera menghapus *cluster* atau proyek setelah selesai untuk mengurangi beban biaya. Karena terutama AWS menghitung beban biaya perjam dan setiap 1 bulan akan mendapat tagihan pembayaran jika tidak membayar maka akun akan di *suspend* dan tidak bisa lagi digunakan. Lalu untuk pelajar ataupun mahasiswa yang tertarik dengan CSPs terutama GCP, AWS, dan Microsoft Azure dan ingin belajar namun mengalami keterbatasan materi GCP merupakan pilihan terbaik.