

# Analisis Migrasi Data LMS pada Klaster Kubernetes Antar-Public Cloud Menggunakan Backup dan Restore

Adimas Fachri Ranunegoro, Favian Dewanta, Bagus Aditya

Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

Kabupaten Bandung, Jawa Barat

[adimasfachri@student.telkomuniversity.ac.id](mailto:adimasfachri@student.telkomuniversity.ac.id) [favian@telkomuniversity.ac.id](mailto:favian@telkomuniversity.ac.id) [goesaditya@telkomuniversity.ac.id](mailto:goesaditya@telkomuniversity.ac.id)

Diterima: 21 Februari 2023. Disetujui: 28 Juli 2023. Dipublikasikan: 29 Juli 2023

*Abstract - Moodle is one of the media for learning management systems that are widely used today because non-face-to-face learning is unavoidable. Fluctuation of user traffics makes moodle suitable for deployment to a public cloud. Public clouds are easier to scale, especially when combined with a container orchestrator such as Kubernetes. However, there are times when it is necessary to migrate data on a Kubernetes cluster from a public cloud to another public cloud to mitigate disasters in a specific region in the public cloud. Moodle applications can be routed to different regions, for example from the Singapore region to Japan region, but this will cause high latency. This problem can be solved by migrating the kubernetes cluster on Google Cloud to the same region on Microsoft Azure as an alternative public cloud. That problem can be solved with a migration of a kubernetes cluster along with persistent volume data between public clouds from Google Cloud Platform to Microsoft Azure and vice versa using backup and restore methods. Then to perform the backup and restore process, velero is used together with the restic plugin to move data on persistent volumes that are outside the Kubernetes cluster. The experiment results show that Velero with the restic plugin can backup and restore persistent volumes outside the cluster. The larger the data size, the longer the backup, restore, and migration time will be. Backup and restore time for each incremental size of approximately 500 MB will increase the backup and restore time by approximately 10 seconds. Meanwhile, on the utility side, the amount of CPU usage during restore consumes more resources than when backup. At the time of backup, the maximum CPU spike was 3.5% at 3 GB data size in both public cloud clusters. Meanwhile, at the time of restore, the maximum CPU spike is 5% at 3 GB of data size.*

*Keywords: Kubernetes, Public Cloud, LMS, Migration*

Abstrak - Moodle menjadi salah satu media untuk learning management system yang banyak digunakan saat ini dikarenakan pembelajaran non-tatap muka yang tidak dapat dihindari. Trafik pengguna yang fluktuatif membuat moodle cocok untuk ditempatkan ke sebuah public cloud. Dengan public cloud, resource yang digunakan menjadi lebih mudah untuk dieskalasi terutama apabila dipadukan dengan container orchestrator seperti kubernetes. Namun, ada kalanya diperlukan migrasi data pada cluster kubernetes dari suatu public cloud menuju ke public cloud lain dengan tujuan untuk memitigasi terjadinya disaster pada suatu region di public cloud. Aplikasi moodle bisa saja dilakukan route ke region yang berbeda, misalnya dari region Singapur ke Jepang, tetapi hal tersebut akan menyebabkan latency yang lebih tinggi. Permasalahan tersebut dapat diatasi dengan melakukan migrasi cluster kubernetes pada Google Cloud menuju ke region yang sama di Microsoft Azure sebagai public cloud alternatif. Salah satu metode untuk melakukan migrasi suatu kubernetes cluster beserta data persistent volume antar public cloud adalah dengan menggunakan metode backup dan restore. Kemudian untuk melakukan proses backup dan restore, Velero digunakan bersamaan dengan plugin restic untuk memindahkan data pada persistent volume yang berada di luar cluster Kubernetes. Hasil eksperimen menunjukkan bahwa velero dengan plugin restic dapat melakukan backup dan restore persistent volume di luar cluster. Semakin besar ukuran data yang akan dimigrasi, maka semakin besar pula waktu yang diperlukan untuk melakukan backup, restore, dan migrasi. Waktu backup dan restore pada setiap ukuran data incremental sebesar kurang lebih 500 MB akan menambah waktu backup dan restore kurang lebih sebanyak 10 detik. Sedangkan dalam sisi utilitas jumlah CPU usage pada saat restore membutuhkan lebih banyak resource dibandingkan dengan saat backup. Pada saat backup maksimum spike CPU adalah sebesar 3.5% pada size data 3 GB di kedua klaster public cloud. Sedangkan pada saat restore maksimum spike CPU adalah sebesar 5% pada size data 3 GB.

*Kata kunci: Kubernetes, Public Cloud, LMS, Migration*

## I. PENDAHULUAN

Pandemi covid-19 yang telah berlangsung selama lebih dari 2 tahun belakangan ini telah mengakselerasi kebutuhan pembelajaran secara non-tatap muka. Pembelajaran non-tatap muka dapat dilakukan secara daring melalui *platform learning management system*, salah satu yang populer dan bersifat *open source* adalah Moodle [1][11]. *Public cloud* menjadi salah satu wadah yang cocok untuk melayani aplikasi moodle karena sifatnya yang fleksibel, efisien, aman, dan hemat biaya. Dengan layanan *public cloud*, server dapat dengan mudah untuk diskalakan secara otomatis sesuai kebutuhan dan membayar sesuai *resources* yang digunakan [2].

Google Cloud Platform (GCP) dan Microsoft Azure merupakan dua dari tiga *public cloud* yang paling banyak digunakan berdasarkan *market share* pada Q1 2021 [3]. Dengan semakin merambahnya layanan *cloud* maka organisasi mendorong untuk beralih dari penggunaan *virtual private server* menuju *container* karena dapat meningkatkan efisiensi *workload* yang berjalan di *cloud* [4]. Untuk mengoptimalkan *container*, dibutuhkan *tool* untuk melakukan orkestrasi *container*. Salah satu *tool* orkestrasi *container* yang *open source* dan didukung langsung oleh Google adalah Kubernetes [5]. Arsitektur kubernetes menerapkan konsep *master node* dan *worker node*. Dengan arsitektur tersebut, apabila salah satu *pod* berhenti berjalan pada *worker node*, *master node* akan secara otomatis memerintahkan *worker node* untuk membuat *pod* replika baru sebagai pengganti *pod* yang berhenti berjalan [14]. Dengan konsep tersebut, arsitektur kubernetes cocok untuk diterapkan pada *learning management system* seperti moodle yang membutuhkan *high availability*.

*Public Cloud* memberi kemudahan dalam membangun arsitektur kubernetes. Kemudahan tersebut membuat suatu *cluster* kubernetes dapat dibuat secara mudah dan kapan saja bisa diberhentikan [12]. Terlepas dari kemudahan membangun suatu *cluster* kubernetes pada *public cloud*, ada kalanya perlu melakukan migrasi suatu *cluster* kubernetes dari suatu *public cloud* menuju ke *public cloud* lainnya. Salah satu tujuan dari migrasi tersebut yaitu untuk memitigasi terjadinya *disaster* pada suatu *region* di *public cloud*. Misalnya ketika *public cloud* seperti Google Cloud mengalami *disaster* terhadap salah satu *region* mereka, maka aplikasi moodle yang ditempatkan pada *region* tersebut tidak dapat diakses [6]. Aplikasi pada *cluster* kubernetes tersebut bisa saja dilakukan route

ke *region* yang berbeda, namun hal tersebut akan menyebabkan *latency* yang lebih tinggi. Permasalahan tersebut dapat diatasi dengan memigrasi *cluster* kubernetes pada Google Cloud menuju ke *region* yang sama di Microsoft Azure sebagai *public cloud* alternatif. Salah satu cara untuk melakukan migrasi dari Google Cloud menuju ke Microsoft Azure adalah dengan menggunakan metode *backup* dan *restore* [7][13]. Namun, terdapat tantangan untuk melakukan *backup* suatu *cluster* kubernetes yang memiliki *database stateful* seperti moodle. *Database* harus diletakan di luar *cluster* untuk tetap menjaga agar data tidak ikut hilang apabila suatu *pod* moodle pada *cluster* mengalami masalah [8].

Velero merupakan salah satu *tool* yang dapat digunakan untuk melakukan *backup* dan *restore* pada *cluster* kubernetes, namun dibutuhkan *plugin* restic untuk dapat melakukan *backup* dan *restore volume database* di luar *cluster* [9]. Pada penelitian sebelumnya [10], velero digunakan untuk *backup* dan *restore* pada *cluster* kubernetes, namun hanya *object* di dalam *cluster* yang dapat dilakukan *backup* dan *restore*, tidak dengan *volume* yang terdapat di luar *cluster* dikarenakan penelitian tersebut tidak menggunakan *plugin* restic. Kemudian penelitian tersebut tidak melakukan skema migrasi antar *public cloud* melainkan antar *on-premise* ke *public cloud*.

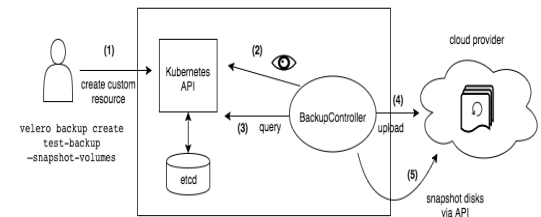
Oleh karena itu, dalam penelitian ini penulis mengimplementasikan dan menganalisis migrasi kubernetes *cluster* beserta *persistent volume* dari Google Cloud menuju ke Microsoft Azure menggunakan mekanisme *backup* and *restore*. Skema *backup* dan *restore* menggunakan velero dengan *plugin* restic sebagai *tool* bantu untuk *backup* *cluster* Kubernetes beserta *persistent volume* di luar *cluster* ke dalam Google Cloud Storage Bucket lalu memigrasi *object backup* tersebut ke Azure Blob Storage sehingga *cluster* Kubernetes pada Azure dapat melakukan *restore* *cluster* yang sudah dicadangkan oleh GCP beserta dengan *persistent volume*.

## II. METODE PENELITIAN

Rancangan sistem pada penelitian ini berfungsi sebagai konsep keseluruhan mengenai komponen-komponen penyusun yang mendukung pengujian yang dilakukan sesuai dengan gambar 1. Pada gambar 1 terdapat dua buah *cluster* Kubernetes, yaitu pada Google Cloud dan Microsoft Azure yang di dalam kedua *cluster* tersebut masing-masing di-*install tool* velero . Google Cloud menjadi

cluster utama sedangkan Microsoft Azure menjadi cluster penampung migrasi dan sebaliknya. Pada cluster Kubernetes yang berada pada Google Cloud, velero client akan menjalankan perintah pada master node melalui API untuk melakukan backup pada resources.

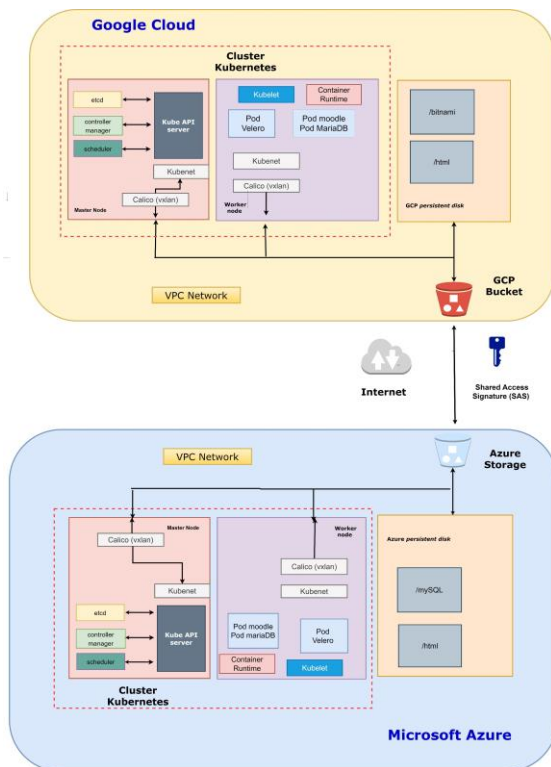
Pada sistem ini resources yang dilakukan backup adalah seluruh object cluster, Moodle, MariadDB, service pada setiap pod, beserta persistent volume di luar cluster. API server yang bertugas menjadi penghubung antar master node dan worker node kemudian meneruskan perintah kepada pod velero untuk melakukan backup pada resources. Container network interface (service) berfungsi sebagai plugin jaringan untuk menghubungkan pod Moodle dan MariadDB yang berada dalam satu jaringan kubernetes cluster [15]. Kemudian resources yang telah dilakukan backup oleh velero disimpan ke dalam bucket GCP melalui adapter fisik dan resources tersebut dikirim melalui jaringan internet dengan HTTPS file transfer protocol dan shared access signature sebagai autentikasi tambahan untuk dikirimkan ke dalam Azure storage dan sebaliknya. Setelah itu cluster Kubernetes melakukan restore dengan mengambil backup resource yang telah dikirim oleh GCP bucket/Azure Blob Storage dan data tersebut disimpan pada Azure Blob Storage di Microsoft Azure atau GCP bucket di Google Cloud. Setelah objek resources dilakukan restore, pod, object, dan persistent volume akan



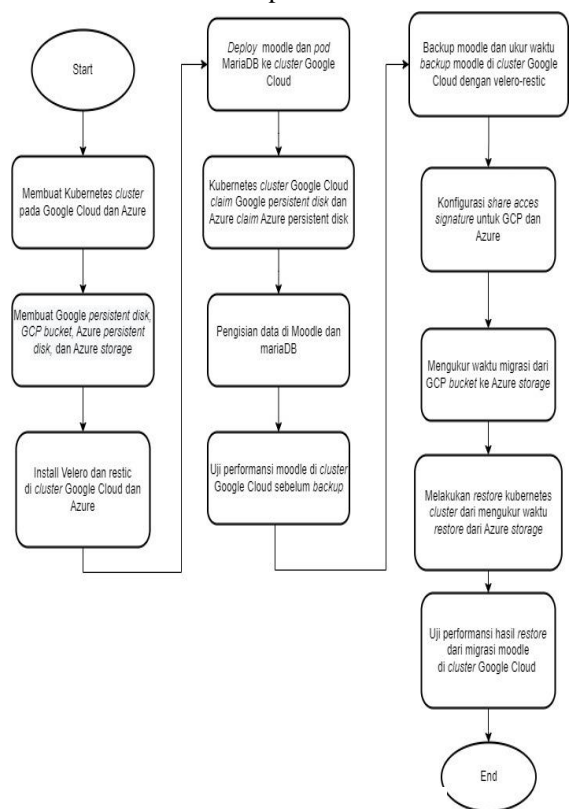
Gambar 2, Mekanisme Velero

mulai dilakukan restore sampai berhasil dengan dinyatakan dengan message completed.

Untuk melakukan backup persistent data dengan velero di luar cluster perlu ditambahkan plugin restic dengan menambahkan flag --use-estic saat instalasi velero [16]. Setelah itu, juga diperlukan anotasi nama persistent data yang sudah dibuat di luar cluster agar restic dapat mengenali persistent data yang akan diikutsertakan untuk dilakukan backup. Velero tidak mengikutsertakan backup persistent volume yang berada di luar cluster apabila tidak menambahkan flag --use-estic untuk instalasi plugin restic di velero. Setelah backup dilakukan, diperlukan restore hasil backup Kubernetes cluster untuk membuktikan bahwa tanpa plugin restic di velero. Setelah backup dilakukan, diperlukan restore hasil backup Kubernetes cluster untuk membuktikan bahwa tanpa plugin restic di velero. Setelah dilakukan backup kemudian akan dilakukan restore untuk membuktikan persistent data di luar cluster dapat muncul saat dilakukan

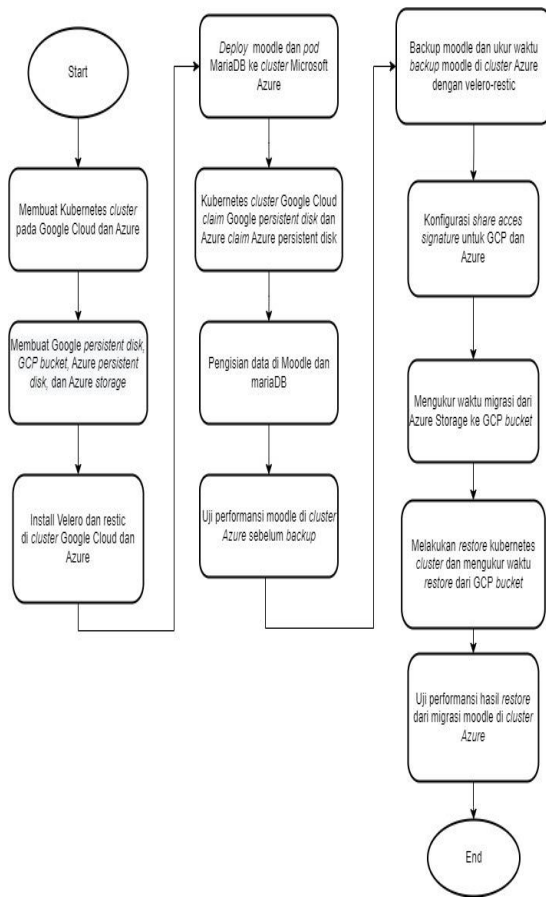


Gambar 1. Rancangan Desain Penelitian



Gambar 3. Diagram Alir Migrasi dari GCP

restore. Oleh karena itu, *plugin* restic berperan penting untuk melakukan *backup persistent* data di luar *cluster*.



Gambar 4. Diagram Alir Migrasi dari Azure ke GCP

TABEL I. DESAIN SPESIFIKASI *CLUSTER* GOOGLE CLOUD PLATFORM

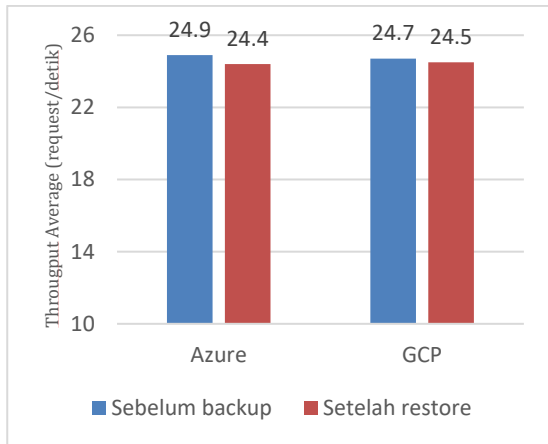
No	Node	Spesifikasi
1	Master	Manage by Google
2	Worker	<b>Memory:</b> 16 GB <b>SSD:</b> 10 GB <b>Number Processor :</b> 4 <b>System Type:</b> 64-bit, x64-based processor <b>Operating System :</b> Linux Ubuntu 20.04 <b>Location :</b> East-US
3	GCP persistent disk	SSD: 30 GB

TABEL II. DESAIN SPESIFIKASI *CLUSTER* AZURE

	Node	Spesifikasi
1	Master	Manage by Azure  <b>Memory:</b> 16 GB <b>SSD:</b> 10 GB <b>Number Processor :</b> 4 <b>System Type:</b> 64-bit, x64-based processor <b>Operating System :</b> Linux Ubuntu 20.04 <b>Location :</b> East-US
2	Worker	
3	GCP persistent disk	SSD: 30 GB

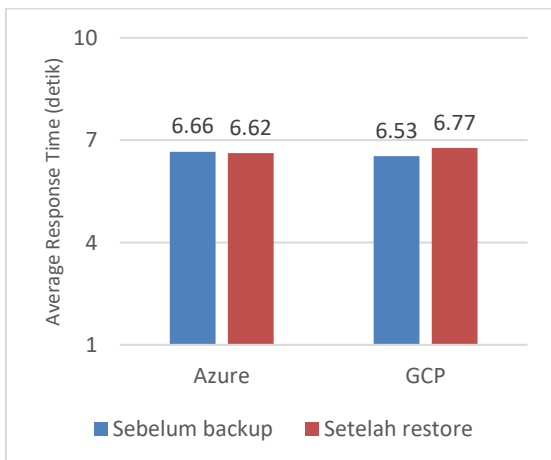
### III. HASIL DAN PEMBAHASAN

Pengujian performansi dilakukan dengan memberikan 500 target *threads* dengan 5 detik *ramp-up* yang berarti akan ada 100 *threads* yang ditambahkan setiap detiknya. Hal tersebut dapat dilakukan dengan menggunakan aplikasi ApacheJMeter sebagai *traffic generator*. Parameter yang akan diamati adalah *throughput*, *response time over time*, *max hits per seconds*, *max response code per second*, dan *error rate*. Pengujian dilakukan menggunakan dua skenario yaitu pada saat sebelum *backup* dan setelah *restore* di *cluster* Google Cloud dan Microsoft Azure. Pengujian dilakukan tiga puluh kali dan diambil rata-rata dari tiga puluh percobaan tersebut. Mengukur waktu *backup*, *restore*, dan migrasi data LMS antar GCP Bucket dan Azure Blob Storage. *Backup*, *restore*, dan migrasi dilakukan dengan dua skenario, pertama saat migrasi data LMS dari GCP ke Azure dan kedua saat migrasi data LMS dari Azure ke GCP. Sampel *size* data *dummy* untuk pengujian adalah sebesar kurang lebih 500 MB *incremental* untuk menguji waktu *backup*, *restore*, dan migrasi di ukuran *file* yang berbeda. Pengujian dilakukan selamat tiga puluh kali dan diambil rata-rata dari tiga puluh percobaan tersebut. Pengujian utilisasi dari *cluster* Kubernetes di Google Cloud dan Microsoft Azure menggunakan *monitoring* yang telah disediakan oleh Google Cloud dan Microsoft Azure. Pengujian dilakukan menggunakan dua skenario yaitu utilitas saat sebelum *backup* dan setelah *restore* di kedua *cluster*.



Gambar 5. Throughput Kubernetes Kluster pada Azure dan GCP

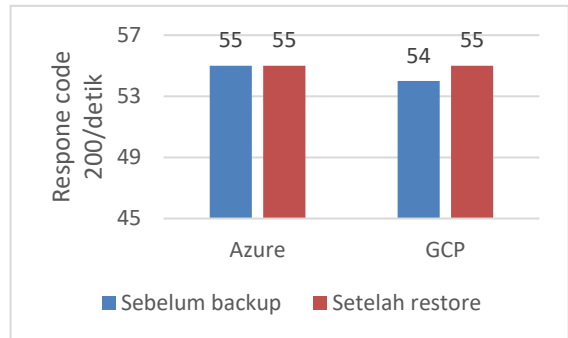
Nilai parameter QoS dalam slide ini diperoleh dari laporan ringkasan ApacheJMeter. Parameter yang ingin disoroti adalah *throughput*. Throughput dalam hal ini adalah jumlah *thread* atau *request* yang diinput ke ApacheJmeter dibagi dengan total waktu semua *thread* atau *request* yang berhasil dikirim kembali ke *client*. perbandingan rata-rata *throughput* pada saat sebelum *backup* dan setelah *restore* tidak berbeda jauh. Ada beberapa hal yang dapat mempengaruhi hasil parameter ini, salah satunya kecepatan dari jaringan internet yang digunakan ketika melakukan pengujian.



Gambar 6. Response Time Kubernetes Kluster pada Azure dan GCP

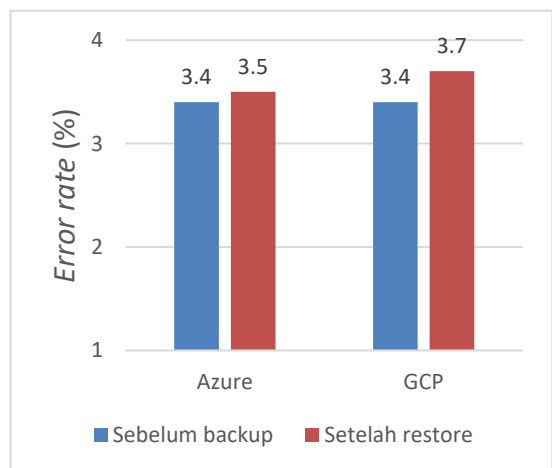
*Response time* adalah jumlah waktu yang dibutuhkan untuk setiap *request* dari *client* yang meminta *request*, saat *request* diproses di server, dan saat server memberikan *feedback* dari *request* yang telah dikirimkan oleh *client*. Pengambilan data dari grafik di atas adalah rata-rata *response time* 500 *thread* pada setiap skenario. Dari grafik tersebut nilai *response time* pada skenario sebelum *backup* dan setelah *restore* mengalami peningkatan setiap detiknya, hal ini terjadi karena *ramp up user* yang diproses meningkat sebesar 100 user setiap detiknya.

Dengan mengumpulnya pengguna setiap detik, waktu yang dibutuhkan untuk menyelesaikan permintaan menjadi lebih lama disetiap detik. Pada gambar 6 terlihat bahwa rata-rata *response time* pada skenario sebelum *backup* dan setelah *restore* tidak memiliki perbedaan yang signifikan.



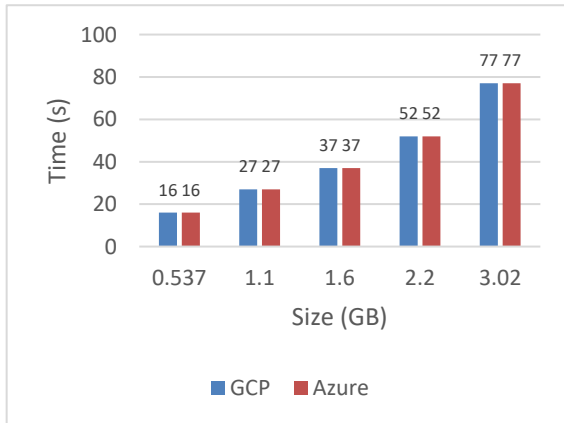
Gambar 7. Response Code 200/detik Kubernetes Kluster pada Azure dan GCP

Pada gambar 7 terlihat bahwa *response code 200* yang diterima pada cluster GCP dan Azure stabil. *Response code 200* adalah permintaan yang berhasil diterima klien dari server. Pada grafik tersebut terlihat bahwa rata-rata perbandingan *response code 200* pada skenario sebelum backup dan setelah restore cukup strabil. Dalam skenario ini juga ditunjukkan bahwa sebelum cluster melakukan backup, *cluster* tidak begitu signifikan mempengaruhi kesuksesan response saat *client* melakukan request pada Kubernetes cluster baik di Microsoft Azure dan GCP.



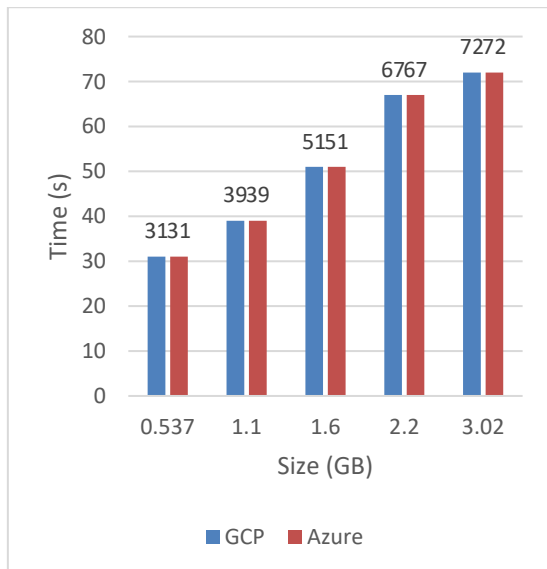
Gambar 8. Error Rate Kubernetes Kluster pada Azure dan GCP

Nilai parameter *error rate* didapatkan dari hasil *summary report* dari aplikasi Apache Jmeter. Ada beberapa hal yang dapat mempengaruhi hasil dari parameter tersebut, salah satunya kecepatan dan kestabilan jaringan internet yang digunakan ketika melakukan pengujian. Setelah dilakukan pengujian selama 20 kali di kedua skenario, pada gambar 8 terlihat bahwa rata-rata *error code* pada setelah *restore* sedikit lebih besar dibandingkan dengan sebelum *backup*.



Gambar 9. Backup Time Pada Kedua Public Cloud

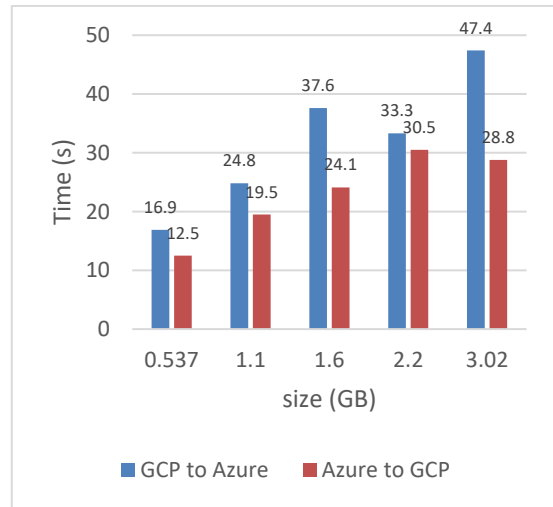
Pada gambar 9 terlihat bahwa *backup time* di kedua *cluster* tidak mengalami perbedaan, hal itu terjadi dikarenakan *machine/resource* dan *tier storage* yang digunakan untuk membangun *cluster* di kubernetes menggunakan spesifikasi yang sama.



Gambar 10. Restore Time Pada Kedua Public Cloud

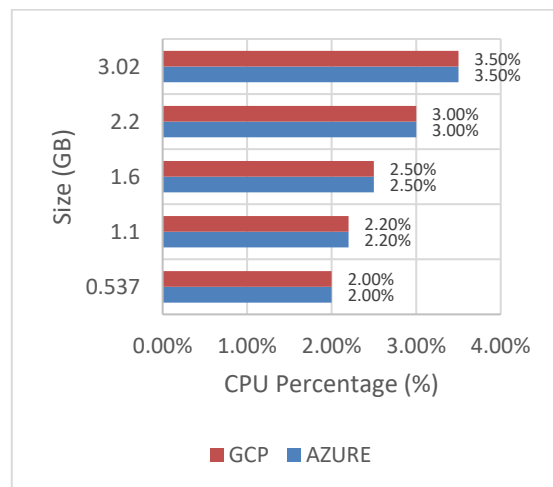
Pada gambar 10 *restore time* di kedua *cluster* juga relatif sama dikarenakan spesifikasi *machine/resource* di kedua *cluster* tidak dibedakan.

Hal itu menunjukkan bahwa waktu *backup* dan *restore* bergantung pada *machine/resource* di kubernetes *cluster* yang ingin dibangun.



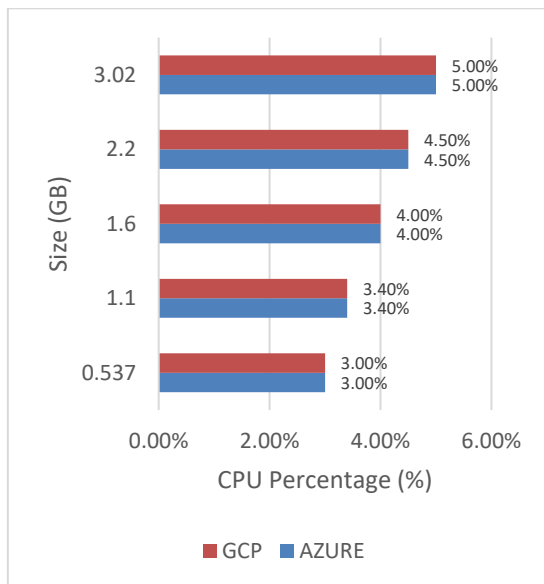
Gambar 11. Migration Time Pada Kedua Public Cloud

Pengujian migrasi antar kluster dilakukan sebanyak 30 kali percobaan di setiap ukuran data yang dimigrasi, lalu diambil waktu rata-rata migrasi antar kluster. Hal itu dilakukan dikarenakan kecepatan transfer antar *cloud provider* bergantung pada situasi jaringan antar *data center* dari kedua *cloud provider* tersebut saat dilakukan pengambilan data. Pada gambar 11 terlihat bahwa kecepatan migrasi dari Microsoft Azure ke Google Cloud relatif lebih cepat dibandingkan dengan Google Cloud ke Azure. Data yang digunakan untuk perbandingan dilakukan secara *incremental* kurang lebih sebesar 500 MB untuk mengestimasi berapa waktu yang diperlukan untuk memindahkan suatu kluster Kubernetes antar Google Cloud ke Azure dan sebaliknya.



Gambar 12. Utilisasi CPU Pada Kedua Public Cloud Saat Backup

Saat melakukan *backup cluster* Kubernetes beserta *persistent volume* menggunakan *velero-estic*, terlihat pada gambar 12 terdapat beberapa *spike* penggunaan CPU di setiap *size* data, di mana maksimal *spike* sebesar 3.5% pada saat melakukan *backup* dengan ukuran 3.02 GB data. Hal ini membuktikan bahwa *spike* CPU akan semakin tinggi apabila ukuran data yang dilakukan *backup* semakin besar.



Gambar 13. Utilisasi CPU Pada Kedua Public Cloud Saat Restore

Pada saat *velero-estic* melakukan *restore* dari hasil *backup* di *cluster* GCP, terlihat pada gambar 13 terdapat *spike* penggunaan CPU maksimal sebesar 5% pada saat *backup* 3.02 GB data. Hal ini membuktikan bahwa *spike* CPU akan semakin tinggi apabila ukuran data yang dilakukan *restore* semakin besar.

#### IV. KESIMPULAN DAN SARAN

Berdasarkan hasil pengujian mekanisme *backup* dan *restore* dengan *persistent volume* pada Kubernetes *cluster* di Google Cloud dan Kubernetes *cluster* di Microsoft Azure pada *zone* yang sama, mekanisme *backup* dan *restore* dapat dijalankan dengan sukses menggunakan *third party* *Velero-estic* dengan dibantu GCP *bucket* sebagai *object store* di GCP dan Azure *blob storage* untuk Azure. Dengan menggunakan GCP *bucket* dan Azure *blob storage*, setiap kubernetes *cluster* dapat melakukan *restore* menggunakan objek yang sudah dilakukan *backup* oleh setiap *source* migrasi ke *object storage* yang telah ditransfer sebelumnya tanpa adanya data *corruption* saat proses migrasi. Jumlah CPU *usage* pada saat *restore* memakan lebih banyak *resource* dibandingkan dengan saat *backup*. Pada saat *backup*

maksimal *spike* CPU adalah sebesar 3.5% pada *Size* data 3 GB di kedua kluster *public cloud*. Sedangkan pada saat *restore* maksimal *spike* CPU adalah sebesar 5% pada *Size* data 3 GB. Semakin besar ukuran data, maka akan memerlukan *resource* CPU yang lebih besar serta waktu *backup*, *restore*, dan *migration time* yang lebih lama. Waktu *backup* dan *restore* pada setiap ukuran *incremental* sebesar kurang lebih 500 MB akan menambah waktu *backup* dan *restore* kurang lebih sebanyak 10 detik. Waktu *backup* dan *restore* pada setiap ukuran data pada kubernetes kluster di kedua *public cloud* sama dikarenakan *resource hardware* yang digunakan sama.

#### V. REFERENSI

- [1] L. L. Salekhova, K. S. Grigorieva, dan T. A. Zinnurov, "Using LMS moodle in teaching CLIL: A case study," *Proc. - Int. Conf. Dev. eSystems Eng. DeSE*, vol. October-2019, hal. 393–395, Okt 2019, doi: 10.1109/DESE.2019.00078.
- [2] "What is Cloud Computing? | Google Cloud." <https://cloud.google.com/learn/what-is-cloud-computing> (diakses Nov 04, 2021).
- [3] "Global cloud IT infrastructure vendor revenue 2021 | Statista." <https://www.statista.com/statistics/443205/vendors-quarterly-revenue-of-global-cloud-it-infrastructure-industry/> (diakses Des 09, 2021).
- [4] Q. Zhang, L. Liu, C. Pu, Q. Dou, L. Wu, dan W. Zhou, "A Comparative Study of Containers and Virtual Machines in Big Data Environment," *IEEE Int. Conf. Cloud Comput. CLOUD*, vol. 2018-July, hal. 178–185, Sep 2018, doi: 10.1109/CLOUD.2018.00030.
- [5] I. M. Al Jawarneh *et al.*, "Container Orchestration Engines: A Thorough Functional and Performance Comparison," *IEEE Int. Conf. Commun.*, vol. 2019-May, Mei 2019, doi: 10.1109/ICC.2019.8762053.
- [6] "Google Cloud data center in London faces outage on UK's hottest day | Reuters." <https://www.reuters.com/technology/google-cloud-data-center-london-faces-outage-uks-hottest-day-2022-07-19/> (diakses Nov 02, 2022).
- [7] M. M. Alshammari, A. A. Alwan, dan I. Alshaikhli, "Data recovery and business continuity in cloud computing: a review of the research literature," *undefined*, 2016.
- [8] "Volumes | Kubernetes." <https://kubernetes.io/docs/concepts/storage/volumes/> (diakses Nov 03, 2022).
- [9] "Velero Docs - Cluster migration." <https://velero.io/docs/v1.9/migration-case/> (diakses Nov 02, 2022).
- [10] L. Agung, R. Sanjaya, M. T. Kurniawan, A. Widjajarto, F. R. Industri, "Desain dan Analisis Mekanisme Backup dan Restore CMS Pada Kluster Kubernetes?" U. Telkom 2020.
- [11] "About Moodle - MoodleDocs." [https://docs.moodle.org/311/en/About\\_Moodle](https://docs.moodle.org/311/en/About_Moodle) (diakses Nov 04, 2021).

- [12] A. Gupta, P. Goswami, N. Chaudhary, dan R. Bansal, "Deploying an Application using Google Cloud Platform," *2nd Int. Conf. Innov. Mech. Ind. Appl. ICIMIA 2020 - Conf. Proc.*, hal. 236–239, Mar 2020, doi: 10.1109/ICIMIA48430.2020.9074911.
- [13] "What is Azure—Microsoft Cloud Services | Microsoft Azure."  
<https://azure.microsoft.com/en-us/overview/what-is-azure/> (diakses Des 01, 2021).
- [14] E. Kim, K. Lee, dan C. Yoo, "On the Resource Management of Kubernetes," *Int. Conf. Inf. Netw.*, vol. 2021-January, hal. 154–158, Jan 2021, doi: 10.1109/ICOIN50884.2021.9333977.
- [15] A. Performansi, P. Onos, B. Qirom, S. N. Hertiana, dan R. M. Negara, "Jaringan Kubernetes Performance Analysis of the Use of Onos Sona-Cni in the Kubernetes Network," vol. 8, no. 1, hal. 163–173, 2021.
- [16] "Velero." <https://velero.io/> (diakses Des 09, 2021).