Implementasi Dan *Profiling* Fungsi *Horizontal Pod Autoscaler* Pada Aplikasi *Web* Dalam Lingkungan Google Kubernetes Engine Dengan Metrik *Pod* Dan *Transaction*

1st Yusuf H. Manik
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
yusufheri@student.telkomuniverity.ac.id

2nd Adityas Widjajarto
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
adtwjrt@telkomuniversity.co.id

3rd Avon Budiyono
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
avonbudi@telkomuniversity.ac.id

***Abstrak*— Aspek skalabilitas aplikasi dapat diimplementasikan dengan manajemen *container* aplikasi yang handal seperti Kubernetes. Kubernetes menjadi pilihan yang umum untuk mengatur dan mengelola *container* aplikasi. Skalabilitas layanan *web* dalam Google Cloud Platform dapat menggunakan Google Kubernetes Engine. Pada penelitian ini dilakukan pengujian aspek skalabilitas terhadap *cluster* Kubernetes yang dibangun di atas Google Kubernetes Engine. Tujuan dari penelitian ini ialah melihat aspek skalabilitas dari *cluster* Kubernetes yang menggunakan *horizontal pod autoscaler* dan *cluster* tanpa *horizontal pod autoscaler.* Konfigurasi *horizontal pod autoscaler* yang digunakan pada *cluster* Kubernetes adalah utilisasi CPU 50 persen. Pengujian terhadap *cluster* dilakukan sebanyak tiga kali setiap variasi *user.* Variasi *user* dimulai dari 100 *user* hingga 1.000 *user* dengan kelipatan 100 *user.* Dari hasil pengujian terhadap *cluster* Kubernetes diperoleh hasil bahwa *cluster* dengan HPA memiliki aspek skalabilitas lebih baik daripada *cluster* non-HPA. Aspek skalabilitas yang dipertimbangkan yaitu jumlah *pod,* jumlah transaksi, *transaction rate, response time, longest transaction* dan *shortest transaction. Cluster* HPA memiliki jumlah transaksi dengan *transaction rate* lima kali lebih banyak, *response time* dua kali lebih cepat, waktu *longest transaction dan shortest transaction* sedikit lebih cepat daripada *cluster* non-HPA. Untuk pengembangan penelitian ini dapat dilakukan metode *autoscaling* berbeda, alat *load testing* bukan siege dan menggunakan parameter yang berbeda.**

***Kata kunci:* container, horizontal pod autoscaler, cluster, cpu. hpa.**

I. PENDAHULUAN

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 memilh 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, infrastuktur 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*.

II. DASAR TEORI

A. Kubernetes

Kubernetes merupakan platform *open-source* yang digunakan untuk melakukan manajemen workloads aplikasi yang dikontainerisasi, serta menyediakan konfigurasi dan otomatisasi secara deklaratif. Kubernetes berada di dalam ekosistem yang besar dan berkembang cepat. Layanan, dukungan dan juga *tools* kubernetes sudah tersedia secara meluas [1].

B. Docker

Docker adalah platform perangkat lunak untuk membuat, menguji, dan menerapkan aplikasi dengan cepat. Docker mengemas perangkat lunak ke dalam kontainer yang memiliki semua yang diperlukan perangkat lunak agar dapat berfungsi diantaranya pustaka, alat sistem, kode, dan waktu proses. Dengan menggunakan Docker, kita dapat dengan cepat menerapkan dan menskalakan aplikasi ke lingkungan apa pun dan yakin bahwa kode yang dibuat akan berjalan dengan baik [2].

C. Utilisasi CPU

Utilisasi CPU merupakan jumlah penggunaan komputasi CPU pada sebuah mesin komputer. Utilisasi CPU pada umumnya menjadi acuan performa atau KPI komputasi dari infrastuktur atau sistem. Utilisasi CPU memiliki ambang batas utilisasi yang terdiri dari 2 jenis yaitu *Warning Threshold* dan *Critical Threshold* [3]. *Warning Threshold* merupakan fase utilisasi CPU di atas 70 persen dan di bawah 90 persen. *Critical Threshold CPU Utilization* merupakan fase utilisasi CPU di atas 90 persen. Berdasarkan dua jenis ambang batas utilisasi CPU tersebut, maka utilisasi CPU yang berada dalam fase aman adalah utilisasi CPU di bawah 70 pesen.

D. Horizontal Pod Autoscaler (HPA)

Horizontal Pod Autoscaler (HPA) merupakan salah satu fitur *autoscaling* pada kubernetes secara horizontal dengan penambahan jumlah *pod* pada *cluster*. HPA akan melakukan duplikasi *pod* secara otomatis yang disesuaikan dengan *load* pada sistem. Cara kerja dari HPA yaitu dengan menunggu *trigger* ataupun informasi dari *metric server. Trigger* HPA dapat diatur pada konfigurasi HPA. *Trigger* HPA dapat berupa *resource* seperti CPU, memori atau metrik-metri lainnya. Ketika kondisi *trigger* terpenuhi maka dilakukan penambahan *pod* dan akan dilakukan pengurangan *pod* bila kondisi *trigger* sudah tidak terpenuhi. Mekanisme penambhan *pod* disebut *scale up* dan mekanisme pengurangan *pod* disebut *scale down.*

E. Load Testing

Load Testing merupakan teknik yang digunakan untuk menguji kemampuan sistem dalam merespon permintaan pengguna dalam berbagai kondisi *load.* Pengujian ini dilakukan untuk melihat perilaku dari aplikasi dan juga *server* ketika banyak *user* mengakses secara bersamaan. *Load Testing* mempermudah kerja para pengembang aplikasi dan juga administrator *server* dalam menguji kemampuan sistem dengan melakukan simulasi akses aplikasi tanpa melibatkan banyak orang untuk mengujinya [4]. Contoh *tools* yang bisa digunakan untuk *load testing* yaitu *Apache JMeter, Locust, LoadRunner, Stege* dan lain sebagainya.

F. Siege

Siege adalah alat yang digunakan untuk menguji dan mengukur kinerja server web. Siege dapat mensimulasikan sejumlah besar pengguna yang mengakses satu atau beberapa *URL* secara bersamaan. Siege memberikan informasi tentang jumlah klik, data yang ditransfer, waktu respons, dan *concurency*. Siege memungkinkan pengguna untuk menentukan jumlah pengguna untuk mengakses *URL*. Siege dibuat oleh Jeffrey Fulmer.

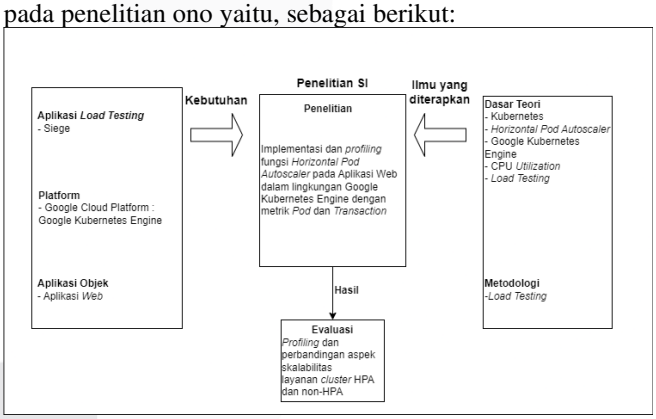
G. Cloud Computing

Cloud computing adalah model untuk memungkinkan akses jaringan dimanapun secara nyaman dan sesuai permintaan ke kumpulan sumber daya komputasi yang dapat dikonfigurasi bersamaan (misalnya, jaringan, *server*, penyimpanan, aplikasi, dan layanan lainnya). Sumber daya pada model *cloud* dapat disediakan disediakan dan dirilis dengan cepat dan mudah. Secara umum, terdapat tiga model layanan *cloud* yaitu IaaS, PaaS, dan SaaS [5]. Penyedia layanan *cloud* yang popular saat ini diantaranya AWS, GCP dan Microsoft Azure.

III. METODELOGI PENELITIAN

A. Model Konseptual Penelitian

Model konseptual ini bertujuan untuk memudahkan dalam melakukan identifikasi permasalahan yang ditemukan pada penelitian ono yaitu, sebagai berikut:

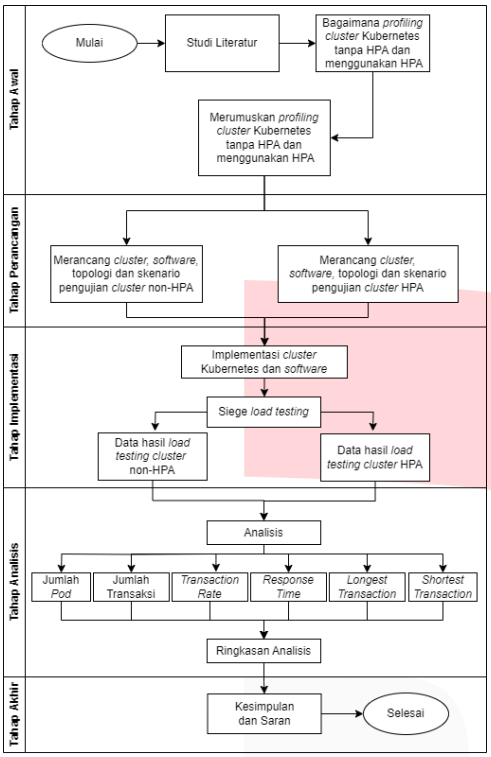


Gambar 1
Model Konseptual Penelitian

Dapat diketahui pada Gambar 1 Model Konseptual Penelitian terdapat 3 bagian diantaranya, kebutuhan, penelitian dan dasar ilmu yang diterapkan. Kebutuhan terdiri dari 3 bagian diantaranya aplikasi *load testing* yaitu *platform* pengujian yaitu Google Kubernetes Engine dan aplikasi objek yaitu aplikasi *web*. Ilmu yang diterapkan terdiri dari dasar teori dan metodologi. Dasar Teori yang digunakan diantaranya Kubernetes, *Horizontal Pod Autoscaler,* CPU *Utilization* dan *Load Testing*. Metodologi yang digunakan adalah *load testing*. Penelitian ini akan membahas implementasi implementasi dan *profiling* fungsi *Horizontal Pod Autoscaler* pada Aplikasi Web dalam lingkungan Google Kubernetes Engine dengan metrik *Pod* dan *Transaction*. Hasil yang diperoleh dari penelitian ini adalah *profiling* dan perbandingan aspek skalabilitas layanan *cluster* HPA dan non-HPA.

B. Sistematika Penelitian

Sistematika penelitian digunakan untuk penyelesaian masalah dalam menggambarkan alur penelitian yang akan dikerjakan sebagai gambaran dalam mecahkan masalah berikut:



Gambar 2
Sistematika Penelitian

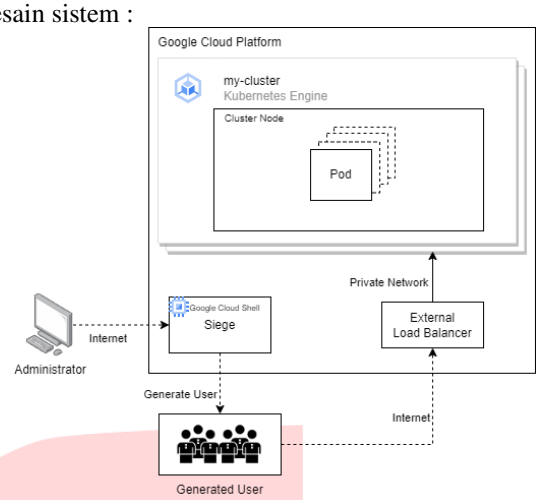
IV. EKSPERIMEN DAN DATA

A. Rancangan Sistem

Sebelum melakukan pengujian, dilakukan perancangan sistem untuk melakukan pengujian pada *cluster* Kubernetes dengan *Horizontal Pod Autoscaler* dan *cluster* Kubernetes tanpa *Horizontal Pod Autoscaler*

1. Rancangan Desain Sistem

Rancangan Desain Sistem dibuat sebagai panduan dalam implementasi dan pengujian. Rancangan ini membantu mendefinisikan kebutuhan komponen yang digunakan saat implementasi dan pengujian. Berikut gambar dari rancangan desain sistem :



Gambar 3
Rancangan Desain Sistem

2. Instrumen Fisik

Instrumen fisik yang digunakan pada penelitian ini adalah:

Tabel 1
Tabel Instrumen Fisik

Jumlah	**Perangkat**	**Spesifikasi**	
1	*Notebook* (Asus X555DG-X133D)	*Processor*	AMD A10-8700P Radeon R6, 10 Compute Cores 4C+6G 1.80 GHz
		Memory	8 GB DDR3L
		SSD	240 GB
		OS	Windows 10 Pro
		System Type	64-bit *operating system*, x64-*based processor*
2	*Cluster Node*	*Type*	e2-medium
		Processor	2 vCPU Intel Broadwell x86/64
		Memory	2 GB
		OS	Container-Optimized OS from Google
		Storage	25 GB

3. Instrumen Non-Fisik

Instrumen non-fisik yang digunakan pada penelitian ini adalah :

Tabel 2
Tabel Instrumen Non-Fisik

No.	Jenis	Nama	Versi
1	*Platform* Utama	Google Cloud Platform (GCP)	2023
2	Sistem Operasi	Windows 10 Pro	10.0.19045 Build 19045
		Container-Optimized OS from Google	1.6.18

3	Aplikasi Utama	Hello-app	1.0
4	Aplikasi Pendukung	Google Kubernetes Engine (*Control Plane*)	1.25.8-gke.1000
		Google Cloud CLI	latest
5	Aplikasi Pengukur	Siege	4.1.6

4. Daftar IP Address

IP *Address* yang digunakan pada penelitian ini, dilampirkan pada Tabel 3 Daftar IP address:

Tabel 3
Daftar IP address

No.	Jenis	*IP Address*
1	*Cluster (External Endpoint)*	10.80.3.7/*
2	*Cluster (Internal Endpoint)*	10.128.0.2/*
3	*Pod*	10.76.0.0/14
4	*Service*	10.80.0.0/20
5	*External Endpoint (Load Balancer)*	34.70.98.47:80

B. Skenario Pengujian

Berikut skenario pengujian yang dilakukan dapat dilihat pada tabel di bawah ini :

Tabel 4
Skenario Pengujian

Variabel	Non-HPA	HPA
CPU *request*	5 *milicore*	5 *milicore*
CPU *limit*	15 *milicore*	15 *milicore*
Konfigurasi HPA	Tidak ada	Utilisasi CPU = 50% Minimum pod = 1 Maksimum pod = 100 (fleksibel)
Variasi *user*	100, 200, 300, 400, 500, 600, 700, 800, 900 dan 1.000	100, 200, 300, 400, 500, 600, 700, 800, 900 dan 1.000
Waktu pengujian	60 detik	60 detik
Jumlah pengujian	3 kali pengujian setiap variasi *user*	3 kali pengujian setiap variasi *user*
Parameter pengujian	*transactions, transaction rate, response time, longest transaction* dan *shortest transaction.*	*transactions, transaction rate, response time, longest transaction* dan *shortest transaction.*

C. Pengujian

Pada tahapan ini, dimulai dengan pengujian dengan *cluster* kubernetes yang tidak menggunakan HPA. Selanjutnya, dilakukan pengujian dengan *cluster* kubernetes yang menggunakan HPA.

1. Pengujian *Cluster* Non-HPA

Pengujian *cluster* tanpa meggunakan HPA dilakukan dengan memberi *HTTP request* menggunakan *Siege* pada aplikasi *web* dengan jumlah *user* tertentu. Tahapan pengujian yang dilakukan adalah sebagai berikut :

a. Mengkonfigurasi *limit* dan *request* penggunaan *CPU* pada *pod* yaitu 15 *milicore* dan 5 *milicore*.
b. Menjalankan *load testing* menggunakan *siege* pada setiap variasi *user* sebanyak 3 kali. *Load testing* dijalankan secara bertahap mulai dari 100 *user* hingga 1000 *user.* Setiap proses *load testing* dilakukan dalam 60 detik.
c. Menghimpun data hasil pengujian yaitu parameter-parameter berikut ini diantaranya *transactions, transaction rate, response time, longest transaction* dan *shortest transaction.*

2. Pengujian *Cluster* HPA

Pengujian *cluster* yang meggunakan HPA juga dilakukan dengan memberi *HTTP request* menggunakan *Siege* pada aplikasi *web* dengan jumlah *user* tertentu. Tahapan pengujian yang dilakukan adalah sebagai berikut :

a. Mengkonfigurasi *limit* dan *request* penggunaan *CPU* pada *pod* yaitu 15 *milicore* dan 5 *milicore*.
b. Membuat dan mengkonfigurasi HPA pada *cluster* Kubernetes dengan jumlah minimum *pod* ialah 1 *pod* dan jumlah maksimum *pod* ialah 100 *pod* atau fleksibel menyesuaikan kebutuhan *cluster* saat pengujian.
c. Menjalankan *load testing* menggunakan *siege* pada setiap variasi *user* sebanyak 3 kali. *Load testing* dijalankan secara bertahap mulai dari 100 *user* hingga 1000 *user.* Setiap proses *load testing* dilakukan dalam 60 detik.
d. Menghimpun data hasil pengujian yaitu parameter-parameter berikut ini diantaranya *transactions, transaction rate, response time, longest transaction* dan *shortest transaction.*

D. Hasil Pengujian

Berikut ini hasil pengujian *load testing* pada aplikasi *web* pada *cluster* yang tidak menggunakan *horizontal pod autoscaler.* Adapun hasil pengujian tersebut berupa parameter-parameter seperti jumlah transaksi, *transaction rate*, *response time*, *longest transaction* dan *shortest transaction.*

1. Hasil Pengujian *Cluster* Non-HPA

Berikut ini hasil pengujian *load testing* pada aplikasi *web* pada *cluster* yang tidak menggunakan *horizontal pod autoscaler* :

a. Jumlah Transaksi

Tabel 5
Jumlah Transaksi *Cluster* Non-HPA

User	1st	2nd	3rd
100	3199	3257	3148
200	3256	3492	3421
300	3646	3618	3529
400	3337	3665	3594
500	3268	3676	3599
600	3676	3714	3726

700	3496	3612	3757
800	3787	3642	3741
900	3806	3662	3481
1000	3592	3605	3605

b. *Transaction Rate*

Tabel 6 *Transaction Rate Cluster* Non-HPA

User	1st	2nd	3rd
100	53,07	53,76	52,29
200	53,95	57,56	56,56
300	59,9	60,11	57,9
400	54,73	60,41	58,95
500	53,75	60,73	59,09
600	61,06	61,36	61,93
700	58,19	59,18	62,55
800	62,48	60,44	61,75
900	62,67	60,84	57,26
1000	58,87	59,5	59,72

c. *Response Time*

Tabel 7 *Response Time Cluster* Non-HPA dalam detik

User	1st	2nd	3rd
100	1,86	1,82	1,87
200	3,51	3,31	3,4
300	4,76	4,86	4,89
400	6,83	6,25	6,34
500	8,65	7,64	7,78
600	8,82	9,05	8,99
700	11,28	10,44	10,3
800	11,04	11,33	11,08
900	12,36	12,59	13,41
1000	14,21	14,15	14,67

d. *Longest Transaction*

Tabel 8
Waktu *Longest Transaction Cluster* Non-HPA dalam detik

User	1st	2nd	3rd
100	4,5	5,31	5,59
200	11,2	8,29	10,29
300	9,6	13,2	20,21
400	16,09	11,54	24,09
500	19,85	21,9	16,7
600	15,89	17,8	16,29
700	18,7	16,34	16,9
800	18,8	18,8	21,5
900	22,4	18,8	32,5
1000	19,74	24,4	32,09

e. *Shortest Transaction*

Tabel 9 Waktu *Shortest Transaction Cluster* Non-HPA dalam detik

User	1st	2nd	3rd
100	0,4	0,41	0,4
200	0,41	0,41	0,41
300	0,41	0,41	0,41
400	0,41	0,41	0,41
500	0,41	0,41	0,4
600	0,4	0,41	0,4
700	0,41	0,41	0,4
800	0,38	0,38	0,39
900	0,47	0,38	0,38
1000	0,44	0,4	0,43

2. Hasil Pengujian *Cluster* HPA

Berikut ini hasil pengujian *load testing* pada *cluster* yang menggunakan *horizontal pod autoscaler* :

a. Jumlah *Pod*

Tabel 5
Jumlah Transaksi *Cluster* HPA

User	1st	2nd	3rd
100	4	15	28
200	4	16	28
300	4	13	49
400	4	16	63
500	4	17	33
600	4	9	24
700	8	27	100
800	14	45	172
900	6	10	26
1000	5	17	37

b. Jumlah Transaksi

Tabel 11
Jumlah Transaksi *Cluster* HPA

User	1st	2nd	3rd
100	2840	9140	13695
200	2940	10495	19791
300	3055	10961	26087
400	3246	19441	33100
500	2847	11487	31781
600	3142	12073	39567
700	3210	22697	58610
800	3071	34924	79333
900	2665	8976	27910
1000	3334	8620	37982

c. *Transaction Rate*

Tabel 12
Transaction Rate Cluster HPA

User	1st	2nd	3rd

100	47,28	150,95	224,8
200	48,51	174,63	326,21
300	50,56	180,96	431,9
400	53,64	323,69	543,51
500	47,13	191,23	520,83
600	51,58	198,41	650,88
700	52,1	373	961,61
800	50,86	577,64	1310,42
900	44,34	148,07	458,67
1000	55,2	142,83	630,09

d. *Response Time*

Tabel 13
Response Time Cluster Non-HPA dalam detik

User	1st	2nd	3rd
100	2,07	0,66	0,44
200	3,92	1,12	0,61
300	5,69	1,63	0,69
400	6,9	1,22	0,73
500	9,19	2,54	0,95
600	10,6	2,94	0,91
700	12,09	1,83	0,72
800	13,58	1,36	0,6
900	15,77	5,72	1,92
1000	15,98	6,1	1,56

e. *Longest Transaction*

Tabel 14
Waktu *Longest Transaction Cluster* Non-HPA dalam detik

User	1st	2nd	3rd
100	18,91	3,81	3,5
200	9,7	4,3	16
300	11,6	9,3	7,8
400	17,1	18,01	21,2
500	16,89	17	16,32
600	25,7	17,9	16,8
700	37,31	18	16,11
800	24,84	8,01	16,36
900	29,3	15,9	21,16
1000	26,5	23,7	16,59

f. *Shortest Transaction*

Tabel 15
Waktu *Shortest Transaction Cluster* Non-HPA dalam detik

User	1st	2nd	3rd
100	0,4	0,4	0,4
200	0,4	0,4	0,4
300	0,41	0,41	0,4
400	0,41	0,4	0,4
500	0,41	0,41	0,4
600	0,41	0,4	0,4
700	0,4	0,4	0,4
800	0,4	0,4	0,4
900	0,41	0,4	0,4
1000	0,41	0,4	0,4

V. ANALISIS

Pada tahap ini dilakukan identifikasi untuk mengambil *insight* dari data hasil pengujian yang diperoleh. Tujuan dari analisis adalah untuk mendapatkan profil atau karakter dari fungsi HPA pada *cluster* Kubernetes di *Google Kubernetes Engine*.

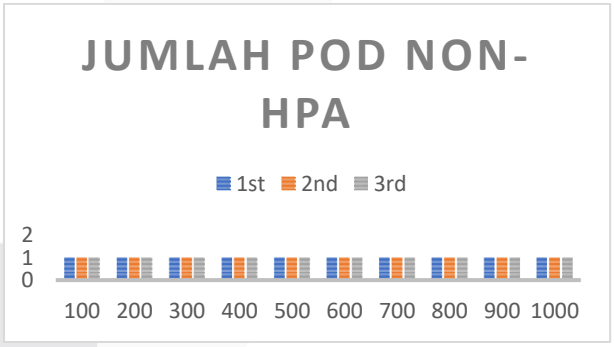
Sebelum masuk ke dalam pembahasan analisis, berikut cara membaca grafik pada bab analisis ini diantaranya :

- 1st berarti pengujian pertama *cluster* Non-HPA
- 2nd berarti pengujian kedua *cluster* Non-HPA
- 3nd berarti pengujian ketiga *cluster* Non-HPA
- 4th berarti pengujian pertama *cluster* dengan HPA
- 5th berarti pengujian kedua *cluster* dengan HPA
- 6th berarti pengujian ketiga *cluster* dengan HPA

A. Analisis Jumlah *Pod*

Berikut ini adalah tahapan analisis jumlah *pod* pada pengujian *cluster* non-HPA dan *cluster* dengan HPA.

1. Jumlah *Pod* Non-HPA

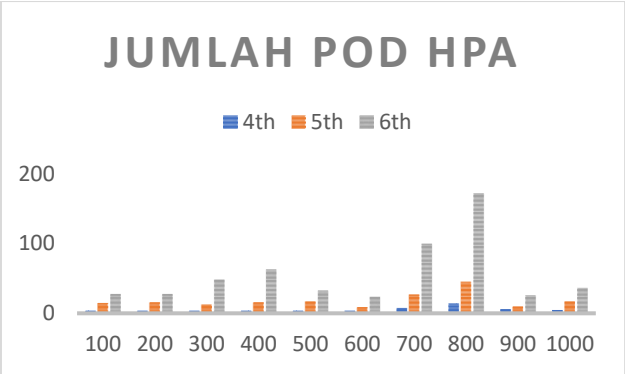


Gambar 4
Jumlah *Pod Cluster* Non-HPA

Analisis dari grafik jumlah *pod* di atas menunjukkan :

a. Jumlah *pod* pada saat pengujian *cluster* non-HPA adalah 1 *pod*.
b. Jumlah *pod* pada saat pengujian *cluster* non-HPA tidak mengalami perubahan baik meningkat ataupun menurun.

2. Jumlah *Pod* HPA

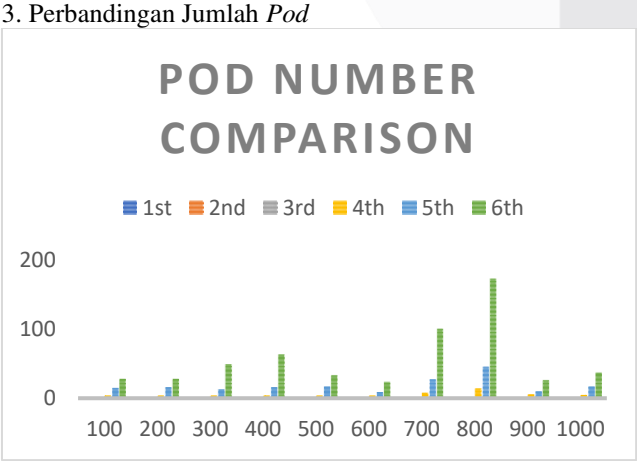


Gambar 5
Jumlah *Pod Cluster* HPA

Analisis dari grafik jumlah *pod* menunjukkan :

a. Peningkatan jumlah *user* saat pengujian cenderung meningkatkan jumlah *pod* pada *cluster* tetapi tidak teratur.
b. Jumlah *pod* pada saat pengujian *cluster* dengan HPA cenderung mengalami peningkatan tetapi sempat mengalami penurunan pada 500 - 600 *user* dan pada 900 – 1.000 *user*.
c. Perubahan jumlah *pod* konsisten naik pada dari pengujian pertama ke pengujian kedua di masing-masing variasi *user*.
d. Perubahan jumlah *pod* konsisten naik pada dari pengujian kedua ke pengujian ketiga di masing-masing variasi *user*.
e. Nilai minimum jumlah *pod* pada *cluster* dengan HPA adalah 4 *pod*. Jumlah *pod* tersebut terjadi pada pengujian pertama di 100 – 600 *user*.
f. Nilai maksimum jumlah *pod* pada *cluster* dengan HPA adalah 172 *pod*. Jumlah *pod* tersebut terjadi pada pengujian ketiga di 800 *user*.
g. Nilai rata-rata jumlah *pod* pada *cluster* dengan HPA dari pengujian pertama adalah 5.7 *pod*.
h. Nilai rata-rata jumlah *pod* pada *cluster* dengan HPA dari pengujian kedua adalah 18,5 *pod*.
i. Nilai rata-rata jumlah *pod* pada *cluster* dengan HPA dari pengujian ketiga adalah dan 56 *pod*.

3. Perbandingan Jumlah *Pod*



Gambar 6
Perbandingan jumlah *pod*

Analisis dari grafik perbandingan jumlah *pod* di atas menunjukkan :

a. Jumlah *pod* pada *cluster* non-HPA tetap yaitu 1 *pod*.
b. Jumlah *pod* pada *cluster* dengan HPA cenderung naik tetapi tidak teratur.

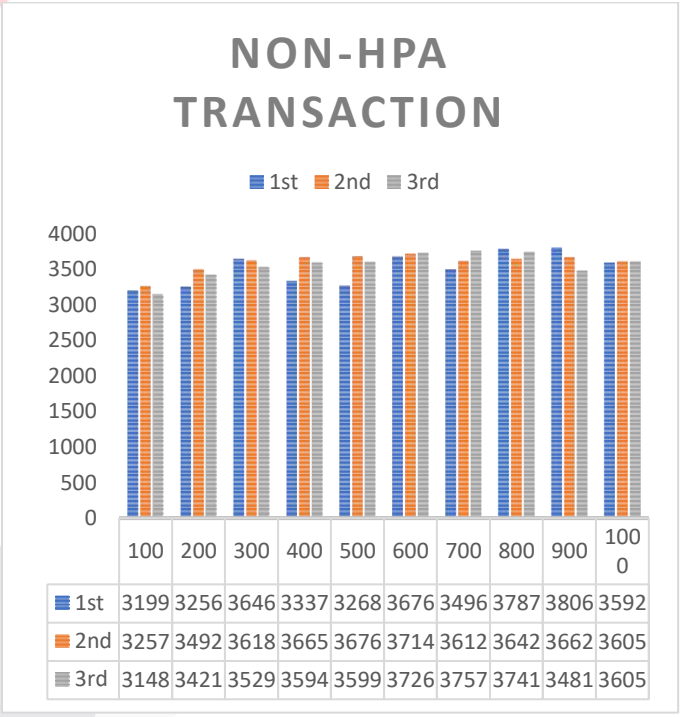
Dari penelusuran data jumlah *pod* diperoleh analisis sebagai berikut :

- Rata-rata jumlah *pod* dari tiga pengujian pada *cluster* HPA adalah 26,73 *pod*.
- Jumlah *pod* pada *cluster* HPA di pengujian pertama memiliki perbedaan signifikan dengan jumlah *pod*.pada *cluster* HPA pengujian kedua. Hal ini terlihat dari jumlah *pod* terbesar pada *cluster* non-HPA adalah 14 *pod* sedangkan pada *cluster* HPA 45 *pod*.
- Jumlah *pod* pada *cluster* HPA di pengujian kedua memiliki perbedaan signifikan dengan jumlah *pod*.pada *cluster* HPA pengujian ketiga. Hal ini terlihat dari jumlah *pod* terbesar pada *cluster* non-HPA adalah 45 *pod* sedangkan pada *cluster* HPA mencapai 172 *pod*.

B. Analisis Jumlah Transaksi

Berikut ini adalah tahapan analisis jumlah transaksi pada pengujian *cluster* non-HPA dan *cluster* dengan HPA :

1. Jumlah Transaksi Non-HPA



	100	200	300	400	500	600	700	800	900	1000
1st	3199	3256	3646	3337	3268	3676	3496	3787	3806	3592
2nd	3257	3492	3618	3665	3676	3714	3612	3642	3662	3605
3rd	3148	3421	3529	3594	3599	3726	3757	3741	3481	3605

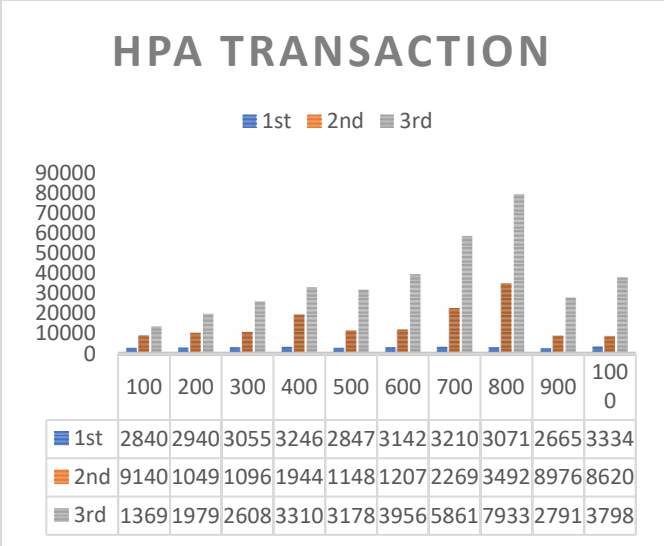
Gambar 7
Jumlah Transaksi Non-HPA

Analisis dari grafik jumlah transaksi di atas menunjukkan:

- Perubahan jumlah transaksi saat pengujian *cluster* non-HPA cenderung tidak teratur dan tidak dapat diprediksi.
- Nilai minimum jumlah transaksi pada *cluster* non-HPA adalah 3.148 transaksi. Nilai minimum tersebut terjadi di pengujian pertama dengan variasi 100 *user*.
- Nilai maksimum jumlah transaksi pada *cluster* non-HPA adalah 3.806 transaksi. Nilai maksimum tersebut terjadi di pengujian pertama dengan variasi 900 *user*.

- Nilai rata-rata dari keseluruhan jumlah transaksi pada *cluster* non-HPA adalah 3.553,57 transaksi.

2. Jumlah Transaksi HPA



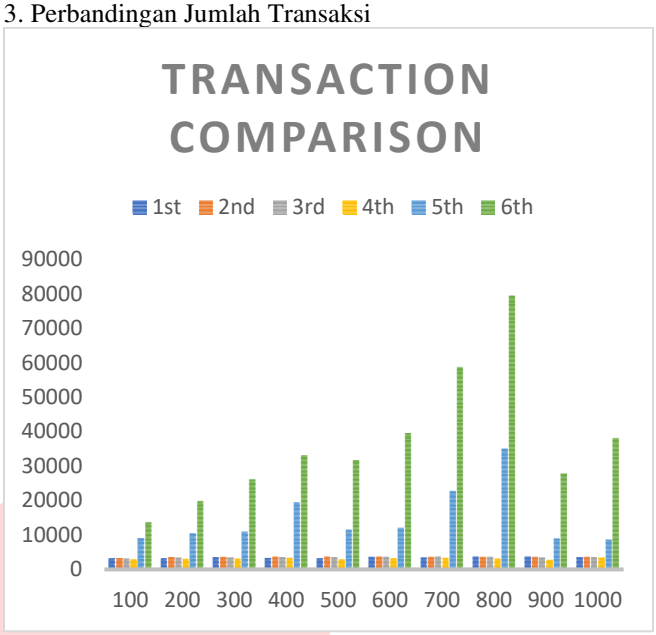
	100	200	300	400	500	600	700	800	900	1000
1st	2840	2940	3055	3246	2847	3142	3210	3071	2665	3334
2nd	9140	1049	1096	1944	1148	1207	2269	3492	8976	8620
3rd	1369	1979	2608	3310	3178	3956	5861	7933	2791	3798

Gambar 8
Jumlah Transaksi HPA

Analisis dari grafik jumlah transaksi menunjukkan:

a. Perubahan jumlah transaksi pada cenderung naik secara keseluruhan.
b. Terjadi penurunan signifikan pada variasi 900 dan 1.000 *user*.
c. Nilai minimum dari jumlah transaksi pengujian pertama pada *cluster* dengan HPA adalah 2.665 *user*. Transaksi tersebut terjadi pada variasi 900 *user*.
d. Nilai minimum dari jumlah transaksi pengujian kedua pada *cluster* dengan HPA adalah 8.620 *user*. Transaksi tersebut terjadi pada variasi 1.000 *user*.
e. Nilai minimum dari jumlah transaksi pengujian ketiga pada *cluster* dengan HPA adalah 13.695 *user*. Transaksi tersebut terjadi pada variasi 100 *user*.
f. Nilai maksimum dari jumlah transaksi pengujian pertama pada *cluster* dengan HPA adalah 3.334 *user*. Transaksi tersebut terjadi pada variasi 1.000 *user*.
g. Nilai maksimum dari jumlah transaksi pengujian kedua pada *cluster* dengan HPA adalah 34.924 *user*. Transaksi tersebut terjadi pada variasi 800 *user*.
h. Nilai maksimum dari jumlah transaksi pengujian ketiga pada *cluster* dengan HPA adalah 79.333 *user*. Transaksi tersebut terjadi pada variasi 800 *user*.
i. Nilai rata-rata jumlah transaksi keseluruhan dari pengujian pertama, kedua sampai pengujian ketiga secara berurut adalah 3.035, 14.881, dan 36.786 transaksi.

3. Perbandingan Jumlah Transaksi



Gambar 9
Perbandingan Jumlah Transaksi

Analisis dari grafik perbandingan jumlah transaksi di atas menunjukkan :

1. Jumlah transaksi pada pengujian *cluster* non-HPA menunjukkan fluktuasi data yang relatif kecil.
2. Jumlah transaksi pada pengujian *cluster* dengan HPA memiliki kecenderungan naik dengan selisih data yang besar.

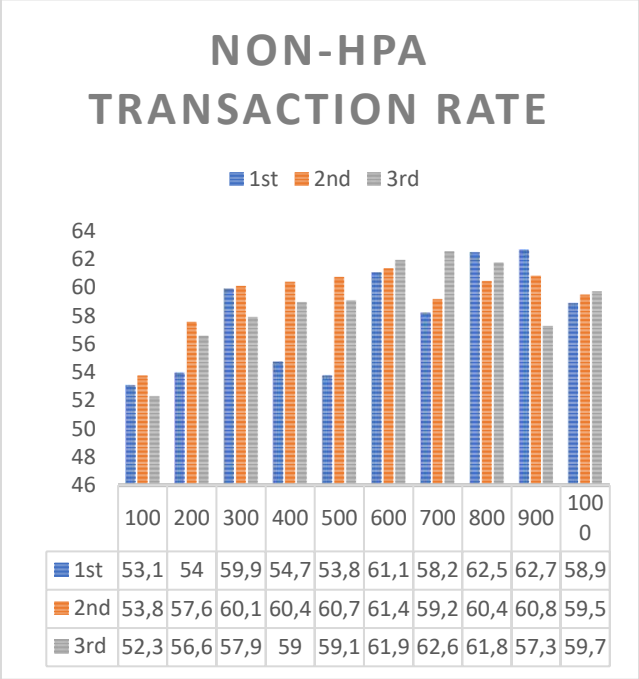
Dari penelusuran data jumlah transaksi diperoleh analisis sebagai berikut :

1. Rata-rata jumlah transaksi dari tiga pengujian pada *cluster* non-HPA adalah 3.553,57 transaksi sedangkan pada *cluster* HPA adalah 18.234 transaksi.
2. Jumlah transaksi pada *cluster* non-HPA memiliki interval data yang kecil yaitu dimulai dari 3.148 sampai 3.806 transaksi dengan rata-rata 3.554 transaksi.
3. Jumlah transaksi pada *cluster* dengan HPA memiliki interval data yang besar yaitu mulai dari 2.665 sampai 79.333 transaksi dengan rata-rata 18.234 transaksi
4. Pengujian pertama pada *cluster* dengan HPA memiliki jumlah transaksi yang cenderung sama dengan jumlah transaksi ketiga pengujian pada *cluster* non-HPA.
5. Pengujian kedua dan ketiga pada *cluster* dengan HPA punya data yang naik signifikan.

C. Analisis *Transacton Rate*

Berikut ini adalah tahapan analisis *transaction rate* pada pengujian *cluster* non-HPA dan *cluster* dengan HPA. Analisis ini menunjukkan bagaimana data kecepatan atau jumlah transaksi per detik saat pengujian dilakukan.

1. *Transaction Rate* Non-HPA



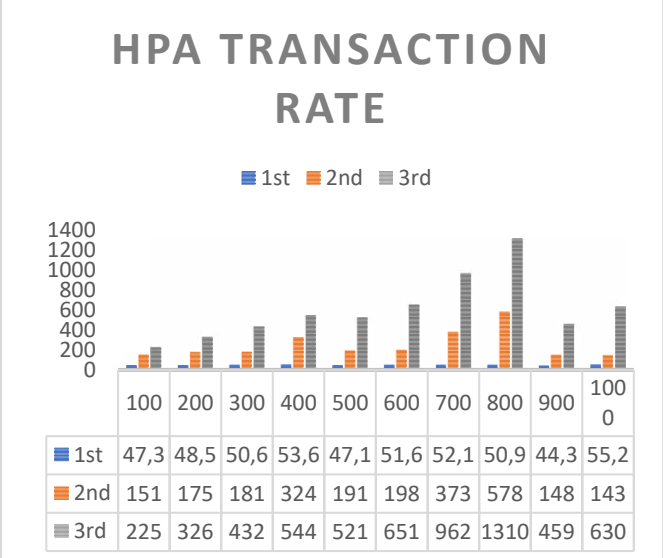
	100	200	300	400	500	600	700	800	900	1000
1st	53,1	54	59,9	54,7	53,8	61,1	58,2	62,5	62,7	58,9
2nd	53,8	57,6	60,1	60,4	60,7	61,4	59,2	60,4	60,8	59,5
3rd	52,3	56,6	57,9	59	59,1	61,9	62,6	61,8	57,3	59,7

Gambar 10
Transaction Rate Non-HPA

Analisis dari grafik jumlah transaksi menunjukkan:

a. Pengujian pertama menunjukkan perubahan yang tidak teratur dan tidak dapat diprediksi.
b. Pengujian kedua menunjukkan perubahan yang cenderung naik.
c. Pengujian ketiga menunjukkan perubahan yang cenderung naik.
d. Nilai rata-rata dari pengujian pertama, kedua dan pengujian ketiga secara berurut adalah 57,87, 59,39 dan 58,60 transaksi per detik. Nilai rata-rata menunjukkan performa *cluster* yang cukup stabil dalam aspek *transaction rate*.

2. *Transaction Rate* HPA



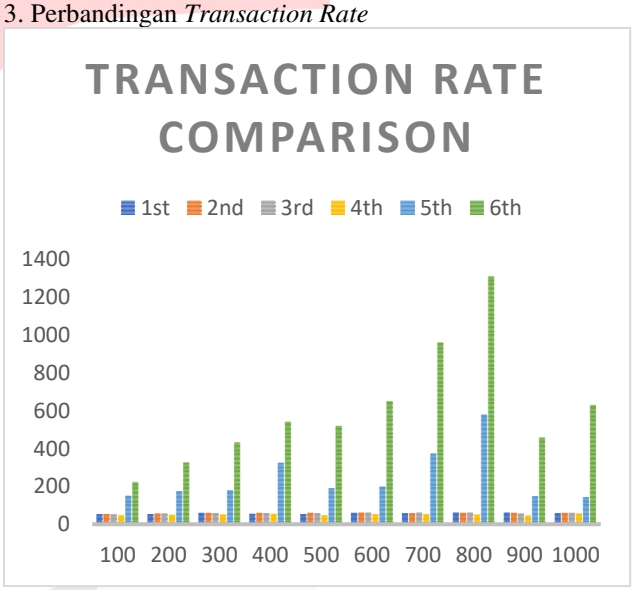
	100	200	300	400	500	600	700	800	900	1000
1st	47,3	48,5	50,6	53,6	47,1	51,6	52,1	50,9	44,3	55,2
2nd	151	175	181	324	191	198	373	578	148	143
3rd	225	326	432	544	521	651	962	1310	459	630

Gambar 11
Transaction Rate HPA

Analisis dari grafik jumlah transaksi di atas menunjukkan:

a. Peningkatan *user* dalam pengujian ini menunjukkan perubahan yang cenderung naik dari variasi 100-800 *user* lalu menurun signifikan di 900-1.000 *user*.
b. Perubahan data dari pengujian pertama ke pengujian kedua sangat signifikan.
c. Perubahan data dari pengujian kedua ke pengujian ketiga juga sangat signifikan.
d. Pada pengujian pertama data *transaction rate* cenderung stabil di rentang waktu 44,34 – 55,2 detik.
e. Pada pengujian kedua di 100 – 800 *user* cenderung mengalami kenaikan kemudian berubah turun di 900 – 1000 *user*.
f. Pada pengujian ketiga di 100 – 800 *user* cenderung mengalami kenaikan kemudian turun naik di 900 – 1000 *user*.

3. Perbandingan *Transaction Rate*



Gambar 12
Perbandingan *Transaction Rate*

Analisis dari grafik perbandingan jumlah *transaction rate* menunjukkan :

c. Nilai *transaction rate* pada *cluster* non-HPA memiliki kecenderungan perubahan yang kecil.
d. Nilai *transaction rate* pada *cluster* dengan HPA cenderung naik signifikan di 100-800 *user* tapi turun di 800-1.000 *user*.

Dari penelusuran data jumlah *transaction rate* diperoleh analisis sebagai berikut :

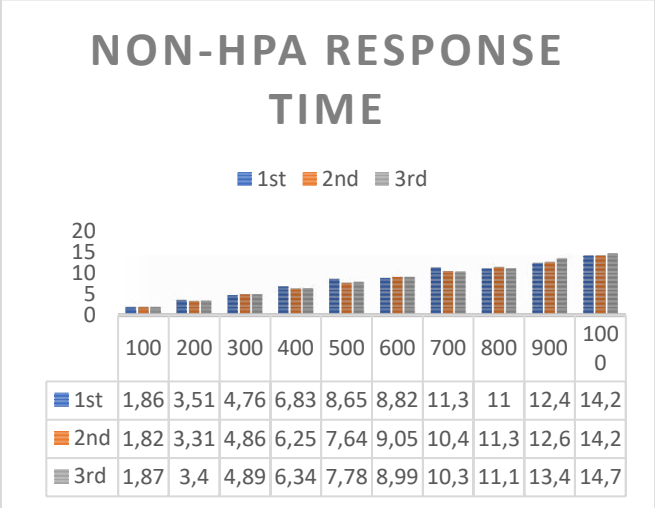
e. Rata-rata nilai *transaction rate* dari tiga pengujian pada *cluster* non-HPA adalah 58,69 transaksi per detik sedangkan pada *cluster* HPA adalah 300,72 transaksi per detik.
f. Interval data *transaction rate* non-HPA relatif kecil yaitu 52,29 sampai 62,67 transaksi per detik.
g. Interval data *transaction rate* HPA relatif besar yaitu 47,13 sampai 1.310,42 transaksi per detik.

h. Data pengujian pertama *transaction rate* HPA lebih kecil dibandingkan dengan data *transaction rate* pada non-HPA.

D. Analisis *Response Time*

Berikut ini adalah tahapan analisis *response time* pada pengujian *cluster* non-HPA dan *cluster* dengan HPA. Analisis ini menunjukkan data rata-rata waktu respon setiap transaksi saat pengujian dilakukan.

1. *Response Time* Non-HPA



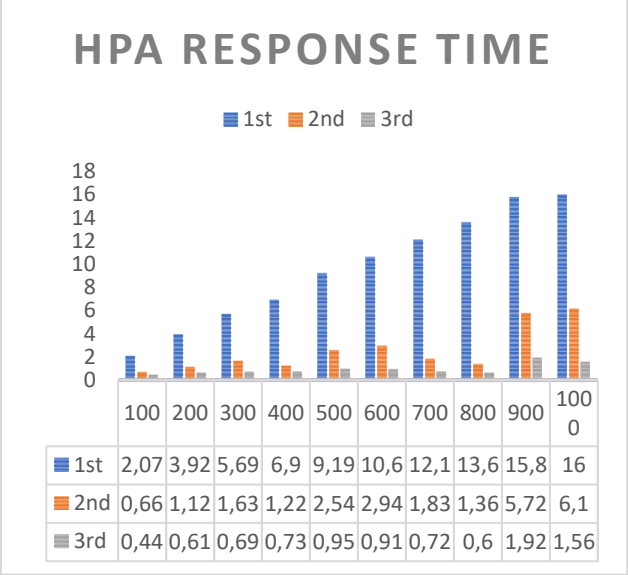
	100	200	300	400	500	600	700	800	900	1000
1st	1,86	3,51	4,76	6,83	8,65	8,82	11,3	11	12,4	14,2
2nd	1,82	3,31	4,86	6,25	7,64	9,05	10,4	11,3	12,6	14,2
3rd	1,87	3,4	4,89	6,34	7,78	8,99	10,3	11,1	13,4	14,7

GAMBAR 13
Response Time Non-HPA

Analisis dari grafik data *response time* di atas menunjukkan:

a. Peningkatan jumlah *user* pada *cluster* non-HPA menyebabkan *response time* meingkat juga.
b. Selisih data antar pengujian juga relatif kecil dan cenderung stabil.
c. Nilasi rata-rata pada pengujian pertama, kedua dan ketiga pada *cluster* non-HPA secara berurut adalah 8,33 detik, 8,14 detik dan 8,27 detik.

2. *Response Time* HPA



	100	200	300	400	500	600	700	800	900	1000
1st	2,07	3,92	5,69	6,9	9,19	10,6	12,1	13,6	15,8	16
2nd	0,66	1,12	1,63	1,22	2,54	2,94	1,83	1,36	5,72	6,1
3rd	0,44	0,61	0,69	0,73	0,95	0,91	0,72	0,6	1,92	1,56

GAMBAR 14
Response Time HPA

Analisis dari grafik data *response time* di atas menunjukkan:

d. Peningkatan jumlah *user* pada *cluster* HPA menyebabkan *response time* meingkat juga.
e. Nilai *response time* pada pengujian pertama ke pengujian kedua dan pengujian kedua mengalami penurunan *response time* yang signifikan.
f. Nilasi rata-rata pada pengujian pertama, kedua dan ketiga pada *cluster* non-HPA secara berurut adalah 9,58 detik, 2,51 detik dan 0,91 detik.

3. Perbandingan *Response Time*

GAMBAR 15
Perbandingan *Response Time*

Analisis dari grafik perbandingan data *response time* di atas menunjukkan :

i. Peningkatan jumlah *user* menyebabkan nilai *response time* semakin besar. Hal ini terjadi pada *cluster* non-HPA maupun *cluster* HPA.
j. Tiga kali pengujian pada *cluster* non-HPA menghasilkan nilai *response time* yang cenderung tetap tetapi pada *cluster* HPA menurun signifikan.

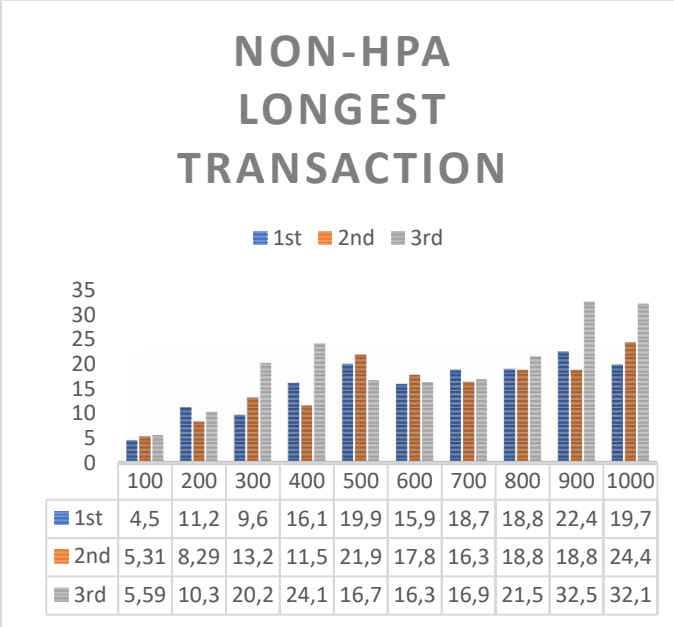
Dari penelusuran data jumlah *response time* diperoleh analisis sebagai berikut :

k. Rata-rata nilai *response time* dari tiga pengujian pada *cluster* non-HPA adalah 8,25 detik sedangkan pada *cluster* HPA adalah 4,33 detik.

E. Analisis *Longest Transaction*

Berikut ini adalah tahapan analisis *longest transaction* pada pengujian *cluster* non-HPA dan *cluster* dengan HPA. Analisis ini menunjukkan data waktu terpanjang atau terlama sebuah transaksi dari seluruh transaksi yang berjalan saat pengujian dilakukan.

1. *Longest Transaction* Non-HPA



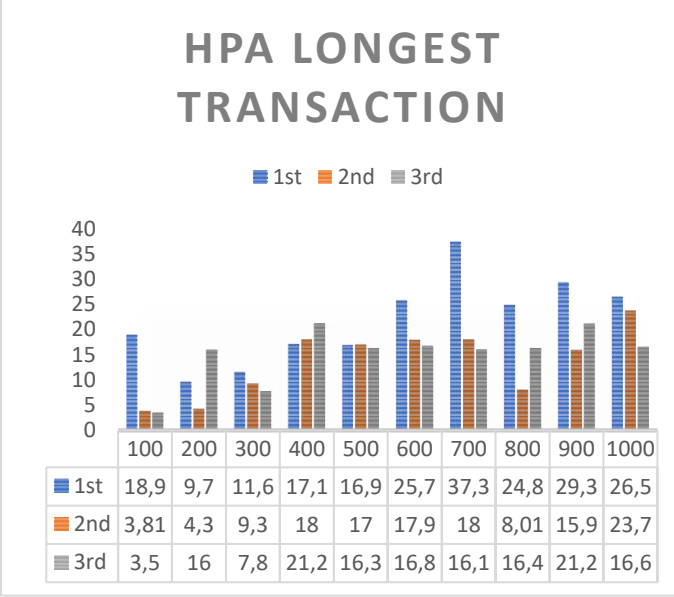
	100	200	300	400	500	600	700	800	900	1000
1st	4,5	11,2	9,6	16,1	19,9	15,9	18,7	18,8	22,4	19,7
2nd	5,31	8,29	13,2	11,5	21,9	17,8	16,3	18,8	18,8	24,4
3rd	5,59	10,3	20,2	24,1	16,7	16,3	16,9	21,5	32,5	32,1

GAMBAR 16
Longest Transaction Non-HPA

Analisis dari grafik data *longest transaction* di atas menunjukkan :

a. Peningkatan *user* cenderung meningkatkan nilai *longest transaction.*
b. Pengujian berkesinambungan selama tiga kali juga cenderung meningkatkan nilai *longest transaction.*
c. Nilai minimum *longest transaction* pada *cluster* non-HPA adalah 4,5 detik.
d. Nilai maksimum *longest transaction* pada *cluster* non-HPA adalah 32,5 detik.

2. *Longest Transaction* HPA



	100	200	300	400	500	600	700	800	900	1000
1st	18,9	9,7	11,6	17,1	16,9	25,7	37,3	24,8	29,3	26,5
2nd	3,81	4,3	9,3	18	17	17,9	18	8,01	15,9	23,7
3rd	3,5	16	7,8	21,2	16,3	16,8	16,1	16,4	21,2	16,6

GAMBAR 17
Longest Transaction HPA

Analisis dari grafik data *longest transaction* tas menunjukkan:

e. Peningkatan *user* dan nilai *longest transaction* cenderung tidak teratur.
f. Pengujian berkesinambungan selama tiga kali juga tidak membuat relasi pada nilai *longest transaction.*
g. Nilai minimum *longest transaction* pada *cluster* non-HPA adalah 4,3 detik.
h. Nilai maksimum *longest transaction* pada *cluster* non-HPA adalah 37,31 detik.

3. Perbandingan *Longest Transaction*

GAMBAR 18 Perbandingan *Longest Transaction*

Analisis dari grafik perbandingan nilai *longest transaction* di atas menunjukkan :

l. Nilai *longest transaction* pada *cluster* non-HPA memiliki kecenderungan perubahan naik karena peningkatan jumlah *user*.
m. Nilai *longest transcation* pada *cluster* HPA memiliki pola data yang tidak teratur dan tidak dapat diprediksi.

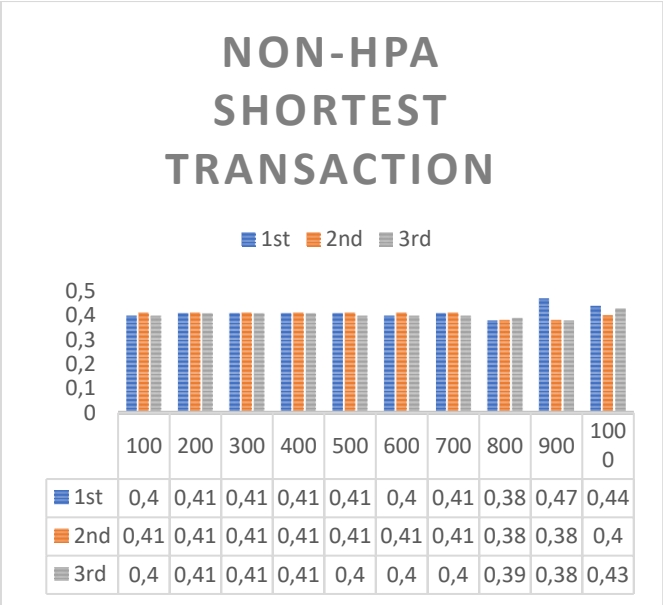
Dari penelusuran data *longest transaction* diperoleh analisis sebagai berikut :

- Data *longest transaction* pada *cluster* non-HPA lebih dapat diprediksi dibandingkan dengan data *longest transaction* pada *cluster* HPA.
- Rata-rata waktu *longest transaction* dari tiga pengujian pada *cluster* non-HPA adalah 16,98 detik sedangkan pada *cluster* HPA adalah 16,85 detik.

F. Analisis *Shortest Transaction*

Berikut ini adalah tahapan analisis *shortest transaction* pada pengujian *cluster* non-HPA dan *cluster* dengan HPA. Analisis ini menunjukkan data waktu terpendek atau tercepat sebuah transaksi dari seluruh transaksi yang berjalan saat pengujian dilakukan.

1. *Shortest Transaction* Non-HPA



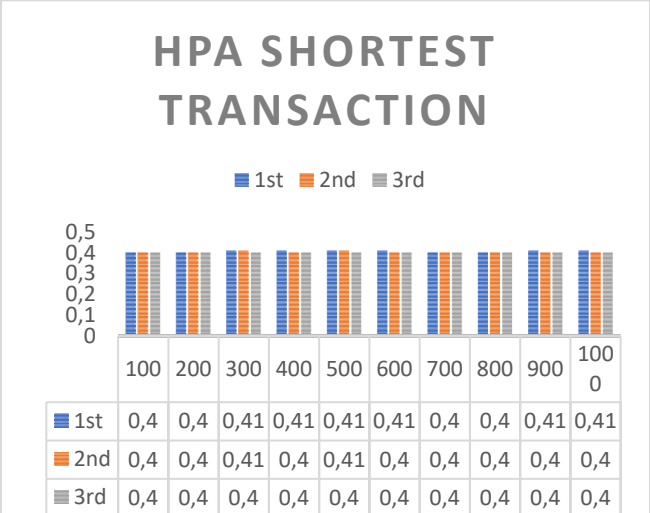
	100	200	300	400	500	600	700	800	900	1000
1st	0,4	0,41	0,41	0,41	0,41	0,4	0,41	0,38	0,47	0,44
2nd	0,41	0,41	0,41	0,41	0,41	0,41	0,41	0,38	0,38	0,4
3rd	0,4	0,41	0,41	0,41	0,4	0,4	0,4	0,39	0,38	0,43

GAMBAR 19
Shortest Transaction Non-HPA

Analisis dari grafik data *shortest transaction* di atas menunjukkan :

n. Data *shortest transaction* cenderung stabil.
o. Nilai minimum *shortest transaction* pada *cluster* non-HPA adalah 0,38 detik.
p. Nilai maksimum *shortest transaction* pada *cluster* non-HPA adalah 0,47 detik.

2. *Shortest Transaction* Non-HPA



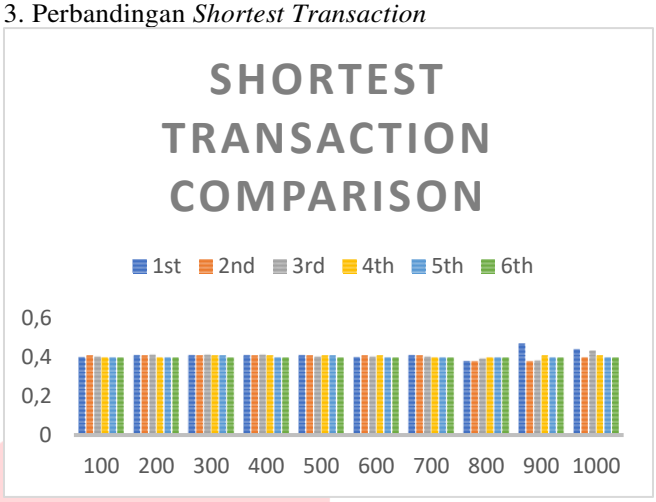
	100	200	300	400	500	600	700	800	900	1000
1st	0,4	0,4	0,41	0,41	0,41	0,41	0,4	0,4	0,41	0,41
2nd	0,4	0,4	0,41	0,4	0,41	0,4	0,4	0,4	0,4	0,4
3rd	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4

GAMBAR 20
Shortest Transaction HPA

Analisis dari grafik data *shortest transaction* menunjukkan :

q. Data *shortest transaction* stabil pada interval 0,4 detik – 0,41 detik.
r. Nilai minimum *shortest transaction* pada *cluster* HPA adalah 0,4 detik.
s. Nilai maksimum *shortest transaction* pada *cluster* HPA adalah 0,41 detik.

3. Perbandingan *Shortest Transaction*



GAMBAR 21
Perbandingan *Shortest Transaction*

Analisis dari grafik perbandingan nilai *shortest transaction* di atas menunjukkan

t. Nilai *shortest transaction* pada *cluster* non-HPA dan *cluster* HPA memiliki data yang cenderung stabil.

Dari penelusuran data *shortest transaction* diperoleh analisis sebagai berikut :

u. Rata-rata waktu *shortest transaction* dari tiga pengujian pada *cluster* non-HPA adalah 0,41 detik sedangkan pada *cluster* HPA adalah 0,4 detik.
v. Interval data pada *cluster* HPA lebih kecil daripada *cluster* non-HPA yaitu 0,4 detik – 0,41 detik sedangkan interval data pada *cluster* non-HPA adalah 0,38 detik – 0,47 detik.

VI. KESIMPULAN

A. Kesimpulan

Berdasarkan tahapan dan data hasil pengujian dalam penelitian ini, didapatkan beberapa simpulan diantaranya :

1. Penerapan *horizontal pod autoscaler* pada *cluster* Kubernetes dengan konfigurasi utilisasi CPU di 50 persen terbukti meningkatkan aspek skalabilitas layanan pada cluster.
2. Menguji kemampuan *cluster* Kubernetes dengan metode *load testing* menggunakan aplikasi *siege*. Pengujian dilakukan dengan konfigurasi *request* CPU *pod* yaitu 5 *milicore* dan *limit* CPU *pod* yaitu 15 *milicore*. *Load Testing* dilakukan sebanyak tiga kali pada masing-masing variasi *user*. Variasi *user* berjumlah 10 variasi dimulai dari 100, 200, 300,400, 500, 600, 700, 800, 900 hingga 1.000 *user*.
3. Aspek skalabilitas pada *cluster* Kubernetes yang menggunakan HPA lebih baik dibandingkan *cluster* non-HPA. Dalam aspek jumlah transaksi, *cluster* HPA menghasilkan jumlah transaksi 5 kali lebih banyak yaitu 18.234 transaksi sedangkan pada *cluster* non-HPA yaitu 3.553,57 transaksi. Dalam aspek *transaction rate*, *cluster* HPA menghasilkan transaksi 5 kali lebih banyak yaitu 300,72 transaksi per detik sedangkan pada *cluster* non-

HPA yaitu 58,69 transaksi per detik. Dalam aspek *response time*, *cluster* HPA 2 kali lebih cepat yaitu 4,33 detik sedangkan pada *cluster* non-HPA yaitu 8,25 detik. Dalam aspek waktu *longest transaction, Cluster* HPA sedikit lebih cepat dari *cluster* non-HPA dengan selisih waktu 0,13 detik. Dalam aspek *shortest transaction, Cluster* HPA sedikit lebih stabil cepat dari *cluster* non-HPA dengan selisih waktu 0,01 detik.

B. Saran

Berdasarkan hasil analisis dan pengujian yang telah dilakukan, berikut berupa saran yang dapat disampaikan:

1. Implementasi fitur *autoscaling* dapat menggunakan metode yang berbeda seperti *vertical pod autoscaler* atau dengan *custom metric.*
2. Melakukan *load testing* dengan aplikasi *load testing* yang berbeda dari *siege.* Selain aplikasi *load testing*, konfigurasi pada *cluster* Kubernetes dapat dikembangkan dari beberapa aspek seperti spesifikasi *node,*
3. Mengukur aspek skalabilitas pada *cluster* Kubernetes dengan parameter pengukuran yang berbeda untuk menambah sudut pandang dan pembahasan dalam penelitian ini.

REFERENSI

[1] *Production-Grade Container Orchestration*. (2021). Kubernetes. Retrieved December 2, 2021, from https://kubernetes.io/

[2] *Apa itu Docker? | AWS*. (2022). Amazon Web Services, Inc. https://aws.amazon.com/id/docker/

[3] Cockcroft, A. (2001). *Capacity planning for internet services: Quick Planning Techniques for High Growth Rates*.

[4] Permatasari, D. I. (2020). Pengujian Aplikasi menggunakan metode Load Testing dengan Apache JMeter pada Sistem Informasi Pertanian. *Jurnal Sistem Dan Teknologi Informasi (JUSTIN)*, *8*(1), 135. https://doi.org/10.26418/justin.v8i1.34452

[5] P. Mell and T. Grance, "*The NIST Definition of Cloud computing Recommendations of the National Institute of Standards and Technology*," NIST Spec. Publ. 800-145, 2011.