

**IN
REVIEW**



**IMPLEMENTASI *HIGH AVAILABILITY* RANCHER KUBERNETES
CLUSTER DENGAN MENGGUNAKAN ENVOY PROXY PADA PT
ASURANSI JIWA GENERALI INDONESIA**

TUGAS AKHIR

MANSURUDIN
41516120064

**UNIVERSITAS
MERCU BUANA
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MERCU BUANA
JAKARTA
2021**



**IMPLEMENTASI *HIGH AVAILABILITY* RANCHER KUBERNETES
CLUSTER DENGAN MENGGUNAKAN ENVOY PROXY PADA PT
ASURANSI JIWA GENERALI INDONESIA**

Tugas Akhir

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer

Oleh:

MANSURUDIN

41516120064

UNIVERSITAS
MERCU BUANA
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MERCU BUANA

JAKARTA

2021

LEMBAR PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini:

NIM : 41516120064

Nama : Mansurudin

Judul Tugas Akhir : **Implementasi *High Availability* Rancher Kubernetes Cluster Dengan Menggunakan Envoy Proxy Pada PT Asuransi Jiwa Generali Indonesia**

Menyatakan bahwa Laporan Tugas Akhir saya adalah hasil karya sendiri dan bukan plagiat. Apabila ternyata ditemukan didalam laporan Tugas Akhir saya terdapat unsur plagiat, maka saya siap untuk mendapatkan sanksi akademik yang terkait dengan hal tersebut.

Jakarta, 13 Februari 2021



Mansurudin

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR

Sebagai mahasiswa Universitas Mercu Buana, saya yang bertanda tangan di bawah ini :

Nama Mahasiswa : Mansurudin
NIM : 41516120064
Judul Tugas Akhir : Implementasi *High Availability Rancher*
Kubernetes Cluster Dengan Menggunakan Envoy
Proxy Pada PT Asuransi Jiwa Generali Indonesia

Dengan ini memberikan izin dan menyetujui untuk memberikan kepada Universitas Mercu Buana **Hak Bebas Royalti Noneksklusif** (*None-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul diatas beserta perangkat yang ada (jika diperlukan).

Dengan Hak Bebas Royalti/Noneksklusif ini Universitas Mercu Buana berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya.

Selain itu, demi pengembangan ilmu pengetahuan di lingkungan Universitas Mercu Buana, saya memberikan izin kepada Peneliti di Lab Riset Fakultas Ilmu Komputer, Universitas Mercu Buana untuk menggunakan dan mengembangkan hasil riset yang ada dalam tugas akhir untuk kepentingan riset dan publikasi selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, 13 Februari 2021


Mansurudin

SURAT PERNYATAAN LUARAN TUGAS AKHIR

Sebagai mahasiswa Universitas Mercu Buana, saya yang bertanda tangan di bawah ini :

Nama Mahasiswa : Mansurudin
NIM : 41516120064
Judul Tugas Akhir : Implementasi *High Availability Rancher*
Kubernetes *Cluster* Dengan Menggunakan Envoy
Proxy Pada PT Asuransi Jiwa Generali Indonesia

Menyatakan bahwa :

1. Luaran Tugas Akhir saya adalah sebagai berikut :

No	Luaran	Jenis		Status		
I	Publikasi Ilmiah	Jurnal Nasional Tidak Terakreditasi			Diajukan	✓
		Jurnal Nasional Terakreditasi		✓		
		Jurnal International Tidak Bereputasi			Diterima	
		Jurnal International Bereputasi				
	Disubmit/dipublikasikan di :	Nama Jurnal	: Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)			
		ISSN	:			
		Link Jurnal	:			
		Link File Jurnal Jika Sudah di Publish	:			

2. Bersedia untuk menyelesaikan seluruh proses publikasi artikel mulai dari submit, revisi artikel sampai dengan dinyatakan dapat diterbitkan pada jurnal yang dituju.
3. Diminta untuk melampirkan scan KTP dan Surat Pernyataan (Lihat Lampiran Dokumen HKI), untuk kepentingan pendaftaran HKI apabila diperlukan

Demikian pernyataan ini saya buat dengan sebenarnya.

Mengetahui
Dosen Pembimbing TA



Muhammad Rifqi, S.Kom., M.Kom

Jakarta, 13 Februari 2021



Mansurudin

LEMBAR PERSETUJUAN

NIM : 41516120064
Nama : Mansurudin
Judul Tugas Akhir : Implementasi *High Availability* Rancher
Kubernetes *Cluster* Dengan Menggunakan Envoy
Proxy Pada PT Asuransi Jiwa Generali Indonesia

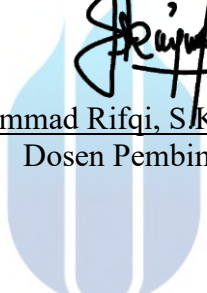
Tugas Akhir ini telah diperiksa dan disetujui

Jakarta, 17 Januari 2021

Menyetujui,



(Muhammad Rifqi, S.Kom, M.Kom)
Dosen Pembimbing



UNIVERSITAS
MERCU BUANA

LEMBAR PERSETUJUAN PENGUJI

NIM : 41516120064
Nama : Mansurudin
Judul Tugas Akhir : Implementasi *High Availability* Rancher Kubernetes
Cluster Dengan Menggunakan Envoy Proxy Pada
PT Asuransi Jiwa Generali Indonesia

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 09 Februari 2021



(Desi Ramayanti, S.Kom, MT)

LEMBAR PERSETUJUAN PENGUJI

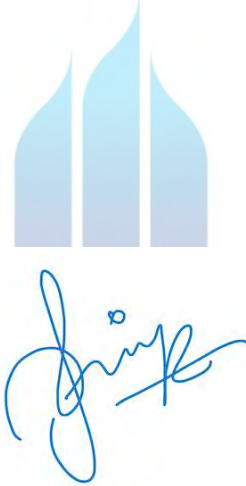
NIM : 41516120064

Nama : Mansurudin

Judul Tugas Akhir : Implementasi *High Availability* Rancher Kubernetes
Cluster Dengan Menggunakan Envoy Proxy Pada
PT Asuransi Jiwa Generali Indonesia

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 09 Februari 2021



UNIVERSITAS
MERCU BUANA

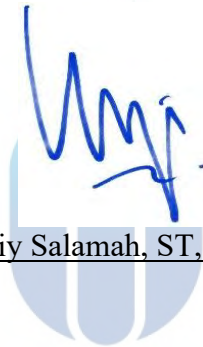
(Afiyati, S.Si, MT)

LEMBAR PERSETUJUAN PENGUJI

NIM : 41516120064
Nama : Mansurudin
Judul Tugas Akhir : Implementasi *High Availability* Rancher Kubernetes
Cluster Dengan Menggunakan Envoy Proxy Pada PT
Asuransi Jiwa Generali Indonesia

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 23 Februari 2021



(Umniy Salamah, ST, MMSI)

LEMBAR PENGESAHAN

NIM : 41516120064
Nama : Mansurudin
Judul Tugas Akhir : Implementasi *High Availability Rancher Kubernetes Cluster*
Dengan Menggunakan Envoy Proxy Pada PT Asuransi Jiwa
Generali Indonesia

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 09 Februari 2021

Menyetujui,



(Muhammad Rifqi, S.Kcm, M.Kom)

Dosen Pembimbing

Mengetahui,



(Diky Firdaus, S.Kom, MM)

Koord. Tugas Akhir Teknik Informatika



(Desi Ramayanti, S.Kom, MT)

Ka. Prodi Teknik Informatika

ABSTRAK

Nama : Mansurudin
NIM : 41516120064
Pembimbing TA : Muhammad Rifqi, S.Kom, M.Kom
Judul : Implementasi *High Availability* Rancher Kubernetes Cluster Dengan Menggunakan Envoy Proxy Pada PT Asuransi Jiwa Generali Indonesia

Container merupakan sebuah teknologi virtualisasi terbaru yang memudahkan para *system administrator* untuk mengelola berbagai aplikasi pada server. Dengan perpaduan teknologi Docker dan Kubernetes sebagai *cluster management container* berbasis *open source*, *system administrator* mampu mengelola aplikasi *microservices* dengan jumlah yang besar dalam satu *console*. Namun permasalahannya dalam memberikan layanan dan informasi pada *cluster existing* pada PT Asuransi Jiwa Generali Indonesia yang telah dijadikan sebagai penelitian masih belum menerapkan sistem *high availability*, mengakibatkan pengaruh besar terhadap *deployment* suatu aplikasi. Oleh karena itu, dengan Implementasi *High Availability* Kubernetes Cluster Rancher Dengan Menggunakan Envoy Proxy merupakan salah satu solusi yang dirancang guna mendukung ketersediaan layanan *management container* pada proses *deployment* aplikasi oleh *user* maupun *developer*. Hasil yang di harapkan dalam ujicoba yang akan dilakukan dengan mencoba untuk mematikan salah satu *service* Rancher pada salah satu *node worker*, menunjukan bahwa tingkat ketersediaan layanan mengalami peningkatan dengan tidak adanya masalah yang muncul ketika mengakses cluster pada saat *deployment* walaupun salah satu *node worker* error.

Kata kunci:

container, docker, kubernetes, rancher, high availability, failover

ABSTRACT

Name : Mansurudin
Student Number : 41516120064
Counsellor : Muhammad Rifqi, S.Kom, M.Kom
Title : Implementasi *High Availability* Rancher Kubernetes Cluster Dengan Menggunakan Envoy Proxy Pada PT Asuransi Jiwa Generali Indonesia

Container is the latest virtualization technology that makes it easy for system administrators to manage various applications on the server. With the combination of Docker and Kubernetes technology as an open source cluster management container, system administrators are able to manage a large number of microservices applications on one console. However, the problem is in providing services and information to the clusters existing at PT Asuransi Jiwa Generali Indonesia, which has been used as research still not implementing a high availability system, resulting in a major influence on the deployment of an application. Therefore, with the Implementation of High Availability Kubernetes Cluster Rancher Using Envoy Proxy is one of the solutions designed to support the availability of container management services in the application deployment process by user and developer. The expected results in tests that will be carried out by trying to shut down one of the Rancher services on one of the worker nodes, shows that the level of service availability has increased with no problems that arise when accessing the cluster during deployment even though one of the worker nodes has an error.

Key words:

container, docker, kubernetes, rancher, high availability, failover

KATA PENGANTAR

Puji syukur kehadiran Allah SWT karena atas rahmat dan hidayah-Nya lah penulis dapat menyelesaikan penulisan laporan Tugas Akhir yang berjudul *Implementasi High Availability Rancher Kubernetes Cluster Dengan Menggunakan Envoy Proxy Pada PT Asuransi Jiwa Generali Indonesia* tepat pada waktunya. Penulis menyadari bahwa tanpa bantuan dan bimbingan dari bapak Muhammad Rifqi, S.Kom, M.Kom tidak akan mampu menyelesaikan penyusunan penulisan laporan ini. Oleh karena itu, kritik dan saran sangat kami harapkan demi adanya perbaikan di masa yang akan datang. Dalam menyelesaikan laporan tugas akhir ini, bukanlah dari hasil kerja keras penulis sendiri, tetapi laporan ini terselesaikan berkat bantuan serta bimbingan dari berbagai pihak. Dalam kesempatan ini kami tidak lupa menyampaikan terima kasih kepada semua pihak yang telah berpartisipasi dan memberi motivasi baik berupa moril maupun materi dalam menyelesaikan laporan ini. Dengan tulus hati Penulis sampaikan terima kasih banyak kepada :

1. Allah Yang Maha Pemberi Petunjuk, yang sudah memberikan petunjuk-Nya kepada penulis sehingga dapat menyelesaikan Tugas Akhir ini,
2. Orang tua penulis, yang telah memberikan dukungan baik moril maupun materil serta do'a yang senantiasa beliau selalu haturkan kepada penulis demi kelancaran penyusunan laporan ini,
3. Bapak Muhammad Rifqi, S.Kom, M.Kom selaku dosen pembimbing yang telah membimbing selama penyusunan penelitian ini,
4. Ibu Desi Ramayanti, S.Kom, MT selaku Ketua Prodi Teknik Informatika Universitas Mercu Buana,
5. Bapak Diky Firdaus, S.Kom, MM selaku Koordinator Tugas Akhir Teknik Informatika Universitas Mercu Buana,
6. Seluruh bapak dosen Prodi Teknik Informatika Universitas Mercu Buana,
7. Bapak Petrus Rio Purwanto selaku *IT Operation Head* PT Asuransi Jiwa Generali Indonesia yang selalu memberikan arahan agar mendapatkan hasil yang diharapkan selama implementasi penelitian,
8. Ghania Felisha selaku seseorang yang spesial di hati penulis yang senantiasa menemani, memberi dukungan, mendoakan, dan penyemangat dalam

proses penyelesaian penelitian ini. Terima kasih telah menjadi wanita terindah yang mendukung setiap keputusan penulis. Semoga kesuksesan dan karunia-Nya selalu tertuju padamu.

9. Keluarga penulis yang senantiasa selalu mendukung selama penyusunan penelitian ini,
10. Teman-teman Mahasiswa dan Mahasiswi Angkatan 30 Universitas Mercu Buana,
11. Saudara dan teman-teman yang selalu membantu ketika menghadapi kesulitan,
12. Serta kepada semua pihak yang terlibat dan tidak dapat disebutkan satu persatu semoga Allah SWT membalas kebajikannya.

Akhir kata, penulis berharap semoga penelitian ini bias bermanfaat bagi para pembaca dan mendorong penelitian-penelitian berikutnya.

Jakarta, 13 Februari 2021

Penulis



DAFTAR ISI

HALAMAN SAMPUL.....	i
HALAMAN JUDUL	i
LEMBAR PERNYATAAN ORISINALITAS	ii
SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR... iii	
SURAT PERNYATAAN LUARAN TUGAS AKHIR..... iv	
LEMBAR PERSETUJUAN	v
LEMBAR PERSETUJUAN PENGUJI	vi
LEMBAR PENGESAHAN	ix
ABSTRAK	x
ABSTRACT	xi
KATA PENGANTAR.....	xii
DAFTAR ISI.....	xiv
NASKAH JURNAL	1
1. PENDAHULUAN	1
2. TINJAUAN PUSTAKA	1
2.1. Docker	1
2.2. Kubernetes.....	1
2.3. Rancher	2
2.4. Envoy Proxy.....	2
3. METODOLOGI PENELITIAN.....	2
3.1. Identifikasi Masalah	3
3.2. Studi Literatur	3
3.3. Perancangan Sistem	3
3.4. Implementasi	4
3.4.1. Persiapan Server.....	4
3.4.1. Konfigurasi Server	4
3.4.2. Instalasi Rancher	4
3.5. Pengujian dan Analisis.....	5
3.6. Kesimpulan dan Saran.....	5
4. HASIL DAN PEMBAHASAN.....	5
4.1. Deskripsi Sistem	5
4.2. <i>High Availability Rancher</i>	5
1. <i>Rancher Login</i>	5

2.	<i>Rancher Cluster Dashboard</i>	5
3.	<i>Rancher Cluster</i>	5
4.	<i>Rancher Node</i>	5
5.	<i>Rancher Storage</i>	6
6.	<i>Rancher Project</i>	6
7.	<i>Rancher Members</i>	6
8.	<i>Envoy Proxy Server</i>	6
4.3.	<i>Testing High Availability</i>	6
5.	KESIMPULAN	6
	DAFTAR PUSTAKA	7
	KERTAS KERJA	8
	BAGIAN 1. LITERATUR REVIEW	9
	BAGIAN 2 ANALISIS DAN PERANCANGAN	12
	BAGIAN 3 KONFIGURASI	16
	BAGIAN 4 TAHAPAN EKSPERIMEN	29
	BAGIAN 5 HASIL SEMUA EKSPERIMEN	30



NASKAH JURNAL

1. PENDAHULUAN

Perkembangan teknologi saat ini yang *up to date* memberikan pengaruh besar pada kehidupan di era digital saat ini. Tak dapat disangkal lagi bahwa perkembangan informasi dan komunikasi yang telah berlangsung begitu cepat, telah menyebabkan sejumlah perubahan yang besar pada masyarakat. Fenomena ini membuat tren rekayasa perangkat lunak bergeser dari program aplikasi desktop ke aplikasi *web* di bawah *program* (Rudiarto, 2018). Sehingga mendorong penyedia layanan untuk membangun sistem dengan *availability* yang sangat tinggi.

High availability cluster diterapkan untuk meningkatkan ketersediaan layanan penggunaan *cluster*. Pada metode *clustering* server ini terdapat dua fungsi yaitu sebagai *failover cluster* dan *load balancing cluster*. Fungsi *failover* bekerja ketika salah satu server bagian dari *cluster* (*node*) mengalami kerusakan maka beban kerja akan digantikan oleh server yang lain, sehingga layanan tidak mengalami gangguan (Rahmatulloh and MSN, 2017). *Container* sangat ideal untuk mengemas layanan *microservices* karena mengisolasi *package* aplikasi, *library*, dan sistem operasi, selain itu *container* berukuran kecil dan tidak mengalami *overhead* seperti permasalahan yang ditemukan pada mesin virtualisasi ketika melakukan proses *deployment* aplikasi.

Pada penelitian sebelumnya, yang sudah dilakukan oleh (Sumbogo, Data and Siregar, 2018), hanya melakukan penerapan *failover* dan *autoscaling* pada *container* sehingga jika terjadi masalah pada sistem inti rancher kubernetes maka *cluster* tidak dapat di kelola dengan baik karena tidak ada sistem *failover* pada sistem kubernetes tersebut. Kendala lain yang akan muncul yaitu rancher kubernetes tidak akan mampu menginformasikan *autoscale* kepada *container* dikarenakan kerusakan sistem api terhadap kubernetes, karena setiap *container* akan menggunakan objek-objek Kubernetes API untuk merepresentasikan *state* yang diinginkan. Salah satu solusi yang dapat diimplementasikan untuk mengatasi masalah tersebut adalah dengan menciptakan *high availability server* menggunakan teknologi *web server clustering* (Pribadi, Pn and Irwansyah, 2020). Untuk mengatasi

permasalahan-permasalahan pada penelitian sebelumnya, maka selain *autoscaling* perlu metode *high availability* Rancher kubernetes *cluster* yang akan bekerja untuk mengelola kubernetes *cluster* pada *node* dan melakukan *failover* jika salah satu *node* kubernetes sistem terjadi masalah atau sedang dalam *maintenance*. Sehingga ketika pengguna melakukan pembuatan *workload*, *autoscale* maupun pembuatan *ingress* terhadap *container*, layanan akan selalu tersedia.

Penelitian ini berfokus melakukan penerapan *high availability* pada rancher kubernetes *cluster*, dimana akan menjawab pengaruh implementasi *failover* terhadap ketersediaan layanan Rancher kubernetes *cluster*. Dimana fokus *failover* terletak pada peran Envoy Proxy yang bertindak sebagai *load balancer* terhadap rancher kubernetes *cluster* rancher.

2. TINJAUAN PUSTAKA

2.1. Docker

Menurut F Adiputra dalam jurnal “*Container dan Docker: Teknik Virtualisasi Dalam Pengelolaan Banyak Aplikasi*”(Adiputra, 2015) Docker merupakan suatu platform berbasis *open source* bagi pengembang perangkat lunak dan pengelolaan sistem jaringan untuk membangun dan menjalankan berbagai aplikasi terdistribusi. Definisi tersebut membawa pengertian praktis bahwa Docker menggunakan arsitektur client-server. Dimana client dan docker berkomunikasi dengan *daemon* docker, yang melakukan suatu tindakan untuk membangun, menjalankan, dan mendistribusikan *container* docker (Dwiyatno, Rakhmat and Gustiawan, 2020). Dengan penggunaan docker, memungkinkan aplikasi aplikasi akan terisolasi dan terjaga keamanannya didalam *container*, dengan menjalankan banyak *container* secara bersamaan pada host yang sama tanpa mengganggu *hypervisor* (Nugroho, 2018).

2.2. Kubernetes

Kubernetes merupakan sebuah platform *open source* yang dirancang untuk *cluster management container*. Dikembangkan dari berbagai pengalaman mengenai pengaturan

container yang diakumulasi oleh Google dan ide-ide terbaik dari komunitas (Indrawati *et al.*, 2019). Kubernetes menggunakan Docker images sebagai dasar untuk membangun aplikasi di atas *container*. Kubernetes menyediakan layanan untuk pendudukan penyebaran berbasis konten dalam *cloud Platform-as-a-Service* (PaaS), dengan fokus khusus pada sistem berbasis *cluster*. Kubernetes terbagi menjadi dua bagian yaitu *master node* dan *worker node* yang memiliki fungsi yang berbeda. *Master node* pada Kubernetes berfungsi sebagai master yang fungsinya mengontrol keseluruhan unit dan *cluster*, mengatur *workload* serta komunikasi antar sistem. Sedangkan pengertian *worker node* atau yang dikenal sebagai pekerja atau *minion*, merupakan mesin tempat *container* (beban kerja) digunakan (Yedutun, Noertjahyana and Palit, 2019).

2.3. Rancher

Rancher adalah platform manajemen *container* Kubernetes *cluster*. Rancher menyediakan API untuk otomatisasi pembuatan *container*. API merupakan software *interface* yang terdiri atas kumpulan instruksi yang disimpan dalam bentuk *library* dan menjelaskan bagaimana agar suatu *software* dapat berinteraksi dengan *software* lain (Arianto, Mukhammad Agus; Munir, Sirojul; Khotimah, 2016). Rancher memungkinkan organisasi menjalankan *container* dalam sistem *production*. Dengan Rancher, pengorganisasian tidak lagi harus membangun *platform* layanan *container* dari awal menggunakan seperangkat teknologi *open source* yang berbeda. Rancher mengelola seluruh perangkat lunak yang diperlukan untuk mengelola seperti *scale up*, *redeploy* and *create container* dalam sistem Kubernetes *cluster*.

2.4. Envoy Proxy

Envoy Proxy memberikan berbagai manfaat seperti *Service Discovery*, load balancing lanjutan dan HTTP *Routing* bersamaan dengan ketahanan terhadap layanan dengan percobaan ulang otomatis, pembatasan kecepatan dan pemutusan sirkuit (Dattatreya Nadig, 2019). Envoy Proxy dibangun berdasarkan solusi pembelajaran proxy lainnya seperti Nginx, HAProxy, *hardware load balancer* dan *cloud based load balancer*. Envoy Proxy juga mendukung

protokol lain seperti gRPC, Kafka dan Redis. Envoy juga memiliki fitur *health checking* terhadap *service* yang ada, yang menjadikan Envoy Proxy mampu untuk mengumpulkan metrik terhadap lalu lintas data pada Layer 7 (L7) seperti HTTP, dimana HTTP adalah protokol yang menyampaikan berbagai informasi dari server situs *web* untuk ditampilkan kepada pengguna atau pengguna melalui *browser web* (Rudiarto, 2017). Algoritma Round-Robin dipilih karena merupakan algoritma yang bekerja dengan cara mendistribusikan tiap *request* yang datang ke *pool server-server backend* yang sebenarnya, mirip dengan algoritma round-robin DNS (Triangga, Faisal and Lubis, 2019).

3. METODOLOGI PENELITIAN

Penelitian dilakukan berdasarkan pengembangan terhadap *existing rancher cluster container* dengan *single server*. Lalu membuat rancher Kubernetes *cluster* baru dengan penambahan beberapa fitur seperti *failover*, *high availability* dan sistem *monitoring* pada sistem rancher *cluster*. Adapun metode pengumpulan data pada penelitian ini adalah dengan melaksanakan observasi dan wawancara di lingkungan PT Asuransi Jiwa Generali Indonesia, Jakarta Selatan.

Seperti yang telah disebutkan di dalam penelitian lainnya bahwa Implementasi *high availability* dengan penggunaan tiga buah *node web server cluster* dapat melayani permintaan client (*request*) jauh lebih baik dibandingkan dengan *single server* (Rivaldy Arif Pratama1, Ratna Mayasari, S.T., M.T.2, Danu Dwi Sanjoyo, S.T., M.T.3 1,2, 2018).

Dengan penggunaan *software load balancer* untuk mendistribusikan *traffic* ke tiap tiap *node worker* maka ketersediaan layanan rancher *cluster* akan meningkat dan terhindar dari *error* pada saat *deployment* aplikasi ke dalam Kubernetes *cluster*. Seperti yang telah disebutkan di penelitian lainnya pun bahwa dengan menerapkan *load balancing* dapat menunjukkan pembagian beban resource seperti memori, dan processor dari masing-masing *container* (Nugroho and Kartadie, 2016).

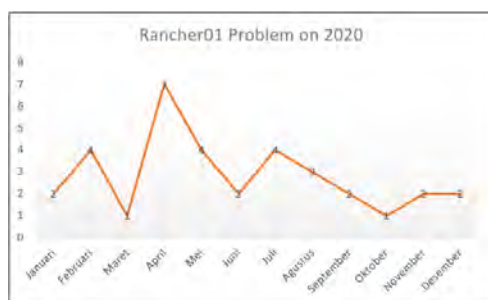
Dalam melakukan tahapan penelitian ini, dibutuhkan sebuah metodologi yang dapat digunakan sebagai acuan tahapan dalam penelitian. Berikut adalah alur metodologi yang digunakan:



Gambar 1. Tahapan Penelitian

3.1. Identifikasi Masalah

Masalah yang diteliti dalam penelitian ini adalah bagaimana cara untuk merancang sistem *high availability* pada rancher kubernetes *cluster* yang mampu meningkatkan ketersediaan layanan pengelolaan aplikasi yang ada di dalam *cluster*. Pada Gambar 2, merupakan data error rancher kubernetes *cluster existing* yang diambil dari seluruh *user complaint* error pada tahun 2020 terhadap rancher kubernetes *cluster*. Dari inti permasalahan tersebut kemudian diidentifikasi langkah-langkah yang dapat digunakan dalam penyelesaian permasalahan tersebut. Hasil dari identifikasi permasalahan ini kemudian dijabarkan solusi-solusi yang memungkinkan untuk digunakan dan diterapkan. Setelah berhasil menentukan permasalahan yang diteliti kemudian dilanjutkan dengan mencari tujuan penelitian.



Gambar 2. Data Error Rancher Existing

3.2. Studi Literatur

Mempelajari literatur dari beberapa bidang ilmu yang sesuai dengan penelitian diantaranya adalah : Docker, Kubernetes, Rancher dan Envoy Proxy. Literatur tersebut diambil dari, jurnal penelitian sebelumnya dan dari dokumentasi yang ditemukan selama proses pengerjaan penelitian. Mempelajari literatur dari beberapa bidang ilmu yang sesuai dengan penelitian diantaranya adalah : Docker, Kubernetes, Rancher dan Envoy Proxy. Literatur tersebut diambil dari, jurnal penelitian sebelumnya dan dari dokumentasi yang ditemukan selama proses pengerjaan penelitian.

3.3. Perancangan Sistem

Penganalisaan kebutuhan sistem dibutuhkan untuk persiapan sebelum melakukan implementasi dari sebuah sistem yang sedang diteliti. Kondisi rancher *cluster* saat ini seperti pada Gambar 2, menggunakan satu buah server untuk menjalankan rancher. Jika ada problem terjadi seperti *disk pressure*, penggunaan memori dan cpu yang tinggi sehingga terjadi error pada server rancher, maka dapat mengakibatkan *user* tidak dapat mengakses rancher dan menghambat *deployment* aplikasi.



Gambar 3. Existing Rancher Cluster

Tabel 1. Spesifikasi Server Rancher Existing

Node	Qty	CPU	RAM	Storage
Rancher	1	2 Core	4 GB	20 GB
Worker	5	4 Core	16 GB	20 GB
Control Plane	3	2 Core	8 GB	20 GB
ETCD	3	2 Core	8 GB	20 GB

3.4. Implementasi

Tahapan Implementasi merupakan tahap lanjutan setelah tahap perancangan. Pada tahap ini sistem akan dibangun berdasarkan perancangan yang telah dibuat sebelumnya untuk membangun sistem *high availability* rancher kubernetes *cluster*. Adapun tahapan-tahapan dalam implementasi adalah seperti pada gambar berikut :



Gambar 4, Tahapan Implementasi

3.4.1. Persiapan Server

Penulis menyiapkan beberapa kebutuhan seperti server dengan sistem operasi Linux CentOS 7 dengan spesifikasi sebagai berikut:

Tabel 2. Spesifikasi Rancher HA

Node	Qty	CPU	RAM	Storage
Rancher	3	2 Core	5 GB	50 GB
Worker	4	4 Core	16 GB	50 GB
Control Plane	3	2 Core	4 GB	50 GB
ETCD	3	2 Core	4 GB	50 GB

Tabel 3. Spesifikasi Envoy Proxy Server

Node	Qty	CPU	RAM	Storage
Envoy Proxy	1	2 Core	4 GB	50 GB

```

[sysgari@GNRIDLCTSYS01 rancher-dev]$ cat rancher-images.txt
rancher/coreos-etcd:v3.4.3-rancher1
rancher/rke-tools:v0.1.66
rancher/k8s-dns-kube-dns:1.15.0
rancher/k8s-dns-dnsmasq-nanny:1.15.0
rancher/k8s-dns-sidecar:1.15.0
rancher/cluster-proportional-autoscaler:1.7.1
rancher/coredns-dnsmasq:1.6.5
rancher/k8s-dns-node-cache:1.15.7
rancher/hyperkube:v1.17.14-rancher1
rancher/coreos-flannel:v0.12.0
rancher/flannel-cni:v0.3.0-rancher6
rancher/calico-cni:v3.13.4
rancher/calico-kube-controllers:v3.13.4
rancher/calico-cilium:v3.13.4
rancher/calico-pod2daemon-flexvol:v3.13.4
weaveworks/weave-kube:2.6.4
weaveworks/weave-npc:2.6.4
rancher/pause:3.1
rancher/nginx-ingress-controller:nginx-0.35.0-rancher2
rancher/nginx-ingress-controller-defaultbackend:1.5-rancher1
rancher/metrics-server:v0.3.6
[sysgari@GNRIDLCTSYS01 rancher-dev]$
  
```

Gambar 4. Rancher Images Requirements

3.4.1. Konfigurasi Server

Konfigurasi server untuk kebutuhan Rancher Kubernetes *Cluster* adalah sebagai berikut :

```

# If you intend to deploy Kubernetes in an air-gapped environment,
# please consult the documentation on how to configure custom RKE images.
nodes:
- address: 192.168.101.67
  port: "22"
  role:
  - controlplane
  - worker
  - etcd
  hostname_override: gnricp7appl2h
  user: centos
  docker_socket: /var/run/docker.sock
  ssh_key_path: ~/.ssh/id_rsa
- address: 192.168.101.68
  port: "22"
  role:
  - controlplane
  - worker
  - etcd
  hostname_override: gnricp7appl2i
  user: centos
  docker_socket: /var/run/docker.sock
  ssh_key_path: ~/.ssh/id_rsa
- address: 192.168.101.69
  port: "22"
  role:
  - controlplane
  - worker
  - etcd
  hostname_override: gnricp7appl2j
  user: centos
  docker_socket: /var/run/docker.sock
  ssh_key_path: ~/.ssh/id_rsa
  
```

Gambar 5. Cluster Config Yaml

```

static_resources:
  listeners:
  - name: listener_http
    address:
      socket_address: { address: 0.0.0.0, port_value: 80 }
    filter_chains:
    - filters:
      - name: envoy_http_connection_manager
        config:
          code_type: auto
          stat_prefix: ingress_http
          route_config:
            name: local_route
            virtual_hosts:
            - name: backend
              domains:
              - "rancher03.general.co.id"
            routes:
            - match:
                prefix: "/"
              redirect:
                path_redirect: "/"
                http_redirect: true
            http_filters:
            - name: envoy_router
              config: {}
  - name: listener_https
    address:
      socket_address: { address: 0.0.0.0, port_value: 443 }
    filter_chains:
    - filters:
      - name: envoy_http_connection_manager
        config:
          code_type: auto
          stat_prefix: ingress_https
          route_config:
            name: local_route
            virtual_hosts:
            - name: backend
              domains:
              - "rancher03.general.co.id"
            routes:
            - match:
                prefix: "/"
              route:
                cluster: rancher_cluster
            http_filters:
            - name: envoy_router
              config: {}
  
```

Gambar 6. Envoy Config

3.4.2. Instalasi Rancher

Penulis melakukan instalasi Rancher dengan cara mempersiapkan helm *chart* dan sistem *image* rancher ke *private repository* *image* terlebih dahulu, Kemudian instalasi rancher dengan helm *chart* tersebut pada sistem kubernetes.

```

[roo@rancher-cluster rancher-ns]$ helm template rancher ./rancher-2.5.3.tgz --output-dir .
$ --namespace cattle-system \
$ --set hostname=rancher03.general.co.id \
$ --set certmanager.version=cert-manager-v1.9.4 \
$ --set rancherimage=registry3.general.co.id/rancher/rancher \
$ --set systemdefaultregistry=registry3.general.co.id \
$ --set useBundledSystemChart=true
NOTE: ./rancher/templates/serviceaccount.yaml
NOTE: ./rancher/templates/clusterrolebinding.yaml
NOTE: ./rancher/templates/service.yaml
NOTE: ./rancher/templates/deployment.yaml
NOTE: ./rancher/templates/ingress.yaml
NOTE: ./rancher/templates/issuer-rancher.yaml
  
```

Gambar 7. Helm Chart Rancher

```
[root@rancher-cluster rancher-new]# kubectl -n cattle-system get
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
rancher   3/3     3             3           464d
```

Gambar 8. Status Deployment Rancher

3.5. Pengujian dan Analisis

Pengujian sistem dilakukan setelah implementasi telah dilaksanakan. Tujuan dari uji coba sistem ini adalah memastikan hasil dari implementasi sistem yang telah dibuat sesuai dengan perancangan yang telah dilakukan. Apabila terjadi kesalahan yang menyebabkan kegagalan dalam ujicoba maka akan dilakukan kembali perancangan serta mengulang implementasi.

3.6. Kesimpulan dan Saran

Penarikan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi dan pengujian metode sudah dilakukan. Kesimpulan ini dari hasil pengujian dan analisis metode. Tahap akhir dari penulisan ini adalah saran yang dimaksudkan guna memperbaiki kesalahan yang terjadi serta memberikan pertimbangan untuk pengembangan selanjutnya.

4. HASIL DAN PEMBAHASAN

Berikut ini adalah hasil dari penelitian yang telah dilakukan:

4.1. Deskripsi Sistem

Pada penelitian ini dibangun sebuah sistem Rancher Kubernetes cluster dengan fitur *High Availability*. Metode *high availability* menggunakan Envoy Proxy dengan algoritma *round-robin* untuk membagi traffic dari *user* ke node rancher yang sudah di deploy.



Gambar 9. Rancher High Availability

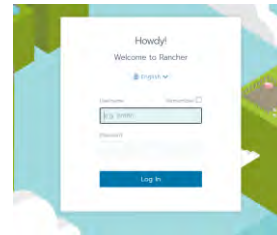
Jika salah satu *node worker* rancher mati, *user* masih dapat mengakses rancher lewat

node yang lain, hal itu membuat *user* mendapatkan layanan *failover* dari Rancher *High Availability* tersebut.

4.2. High Availability Rancher

1. Rancher Login

User harus memasukkan *username* dan *password* terlebih dahulu untuk masuk ke dalam rancher web interface.



Gambar 10. Login Screen

2. Rancher Cluster Dashboard

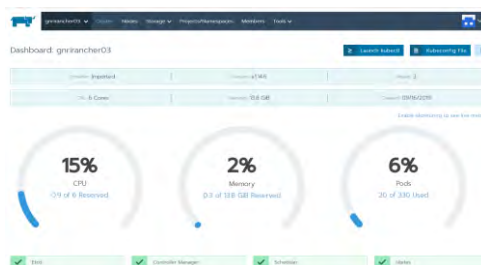
Cluster Dashboard merupakan tampilan awal halaman setelah *login*. Seperti pada gambar dibawah ini, terlihat ada 2 *cluster* yaitu *cluster rancher sistem* dan *cluster rancher untuk production*.



Gambar 11. Rancher Cluster Dashboard

3. Rancher Cluster

Pada halaman berikut ini, *user* mampu melihat halaman *dashboard cluster* untuk melihat *metric* penggunaan *cluster*.



Gambar 12. Rancher Cluster

4. Rancher Node

Halaman ini berisikan informasi tipe *node* yang ada pada rancher *cluster*, serta memberi informasi batasan *pod* yang bisa pada di *handle* oleh *node* tersebut.



Gambar 13. Rancher Node

5. Rancher Storage

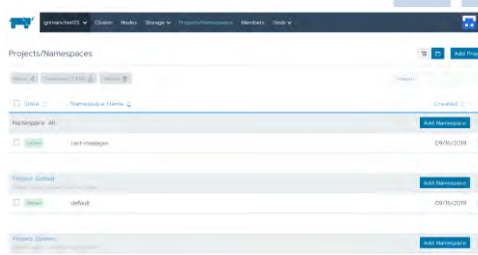
Halaman berikut berisikan informasi tentang media penyimpanan yang dapat digunakan pada *cluster*. mulai dari NFS share, *Google Persistent Disk* hingga *Gluster Volume*.



Gambar 14. Rancher Storage

6. Rancher Project

Untuk manajemen proyek aplikasi agar sesuai dengan *role based access user* masing-masing, *user* maupun administrator harus melakukan pembuatan terlebih dahulu di halaman tersebut.



Gambar 15. Rancher Storage

7. Rancher Members

User yang belum memiliki akun, harus mendaftar terlebih dahulu untuk mengelola terhadap proyek yang ada di dalam *cluster*. Nantinya administrator menyesuaikan *user* dengan proyek terkait. Jadi *user* sudah ditentukan proyek mana yang bisa di lihat dan dikelola pada saat *login* ke dalam *cluster*.



Gambar 16. Rancher Members

8. Envoy Proxy Server

Pada gambar dibawah ini , envoy proxy berjalan menunggu *request* ke *domain cluster*

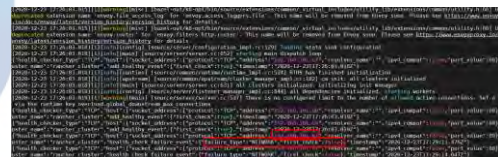
rancher dengan *service http request* di port 80 dan port 443 untuk https.



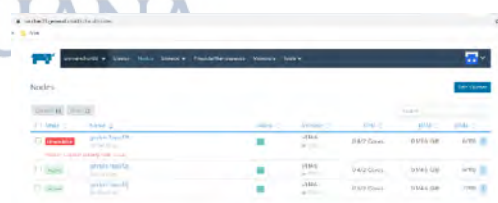
Gambar 17. Envoy Proxy Server

4.3. Testing High Availability

Untuk memastikan bahwa rancher kubernetes *cluster* sudah berhasil dikatakan *high availability*, maka percobaan dapat dilakukan dengan cara mencoba mematikan satu buah *node* rancher kemudian mencoba akses kembali rancher dengan keadaan salah satu *node* yang sudah mati. Setelah peneliti coba untuk mematikan salah satu *node* maka *log* envoy proxy pun muncul menunjukkan bahwa salah satu *node* dengan status *failure network* yang artinya *node backend* tersebut sedang dalam kondisi *down*. Selanjutnya *node* yang berstatus *failure* akan di lewati pada saat pembagian *traffic* oleh envoy proxy. Sehingga memungkinkan *user* masih bisa mengakses rancher kubernetes *cluster* menggunakan jalur *node* lainnya yang masih hidup.



Gambar 18. Envoy Proxy Health Check



Gambar 19. Rancher Testing High Availability

5. KESIMPULAN

Untuk memastikan bahwa rancher kubernetes *cluster* sudah berhasil dikatakan *high availability*, maka percobaan dapat dilakukan dengan cara mencoba mematikan satu buah *node* rancher kemudian mencoba akses kembali rancher dengan keadaan salah satu *node* yang sudah mati. Setelah peneliti coba untuk mematikan salah satu *node* maka *log* envoy proxy pun muncul menunjukkan bahwa salah satu *node* dengan status *failure*

network yang artinya *node backend* tersebut sedang dalam kondisi *down*. Selanjutnya *node* yang berstatus *failure* akan di lewati pada saat pembagian *traffic* oleh envoy proxy. Sehingga memungkinkan *user* masih bisa mengakses rancher kubernetes *cluster* menggunakan jalur *node* lainnya yang masih hidup.

DAFTAR PUSTAKA

- Adiputra, F. (2015) 'Container dan Docker: Teknik Virtualisasi dalam Pengelolaan Banyak Aplikasi Web', *Jurnal SimanteC*, 4(3).
- Arianto, Mukhammad Agus; Munir, Sirojul; Khotimah, K. (2016) 'Analisis dan Perancangan Representational State Transfer (REST) Web Service Sistem Informasi Akademik STT Terpadu Nurul Fikri Menggunakan Yii Framework', *Jurnal Teknologi Terpadu*, 2(2), pp. 1–8.
- Dattatreya Nadig, N. (2019) 'Testing Resilience of Envoy Service Proxy with Microservices', *Degree Project in Technology*.
- Dwiyatno, S., Rakhmat, E. and Gustiawan, O. (2020) 'Implementasi Virtualisasi Server Berbasis Docker Container', *Prosisko*, 7(2), pp. 165–175. Available at: <https://e-jurnal.lppmunsera.org/index.php/PROSISKO/article/view/2520/>.
- Indrawati, N. *et al.* (2019) 'Implementasi Load Balancer Dengan Lightweight Virtualization Menggunakan Docker Untuk Layanan Video on Demand Implementation of Load Balancer in Lightweight Virtualization Using Docker for Video on Demand Service', *e-Proceeding of Engineering*, 6(1), pp. 802–809. Available at: <https://libraryproceeding.telkomuniversity.ac.id/index.php/engineering/article/viewFile/8869/8734>.
- Nugroho, M. A. (2018) 'Analisis Cluster Container Pada Kubernetes Dengan Infrastruktur Google Cloud Platform', *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 3(2), pp. 84–93. doi: 10.29100/jupi.v3i2.651.
- Nugroho, M. A. and Kartadie, R. (2016) 'Analisis Kinerja Penerapan Container untuk Load Balancing Web Server', *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 1(02), pp. 7–15. doi: 10.29100/jupi.v1i02.35.
- Pribadi, Y., Pn, A. B. and Irwansyah, M. A. (2020) 'Analysis of the Use of the Failover Clustering Method to Achieve High Availability on a Web Server (Case Study: Informatics Department Building)', *JUSTIN (Jurnal Sistem dan Teknologi Informasi)*, 8(2). doi: 10.26418/justin.v8i2.31965.
- Rahmatulloh, A. and MSN, F. (2017) 'Implementasi Load Balancing Web Server menggunakan Haproxy dan Sinkronisasi File pada Sistem Informasi Akademik Universitas Siliwangi', *Jurnal Nasional Teknologi dan Sistem Informasi*, 3(2), pp. 241–248. doi: 10.25077/teknosi.v3i2.2017.241-248.
- Rivaldy Arif Pratama1, Ratna Mayasari, S.T., M.T.2, Danu Dwi Sanjoyo, S.T., M.T.3 1,2, 3Prodi (2018) 'Implementasi Web Server Cluster Menggunakan Metode Load Balancing Pada Container Docker, Lxc, Dan Lxd', 5(3), pp. 5028–5035. Available at: <https://libraryproceeding.telkomuniversity.ac.id/index.php/engineering/article/download/7917/7809>.
- Rudiarto, S. (2017) 'Design Model System Measuring Performance of Laboratory Assistant at Faculty of Computer Science', *International Research Journal of Computer Science*, 4(8), pp. 22–29. doi: 10.26562/irjcs.2017.aucs10084.
- Rudiarto, S. (2018) 'DEVELOPMENT OF BLAST EMAIL, CHATTING, AND SMS FEATURES ON EMPLOYEE DATA APPLICATION Testing Design Coding', 5(05), pp. 255–262.
- Sumbogo, Y. T., Data, M. and Siregar, R. A. (2018) 'Implementasi Failover Dan Autoscaling Kontainer Web Server Nginx Pada Docker Menggunakan Kubernetes', *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*, 2(12), pp. 6849–6854.
- Triangga, H., Faisal, I. and Lubis, I. (2019) 'Analisis Perbandingan Algoritma Static Round-Robin dengan Least-Connection Terhadap Efisiensi Load Balancing pada Load Balancer Haproxy', *InfoTekJar (Jurnal Nasional Informatika dan Teknologi Jaringan)*, 4(1), pp. 70–75. doi: 10.30743/infotekjar.v4i1.1688.
- Yedutun, K., Noertjahyana, A. and Palit, H. N. (2019) 'Implementasi Container Kubernetes untuk Mendukung Scalability', *Jurnal Infra* 7.

KERTAS KERJA

Ringkasan

Kertas kerja ini merupakan material kelengkapan artikel jurnal dengan judul di atas. Kertas kerja berisi semua material hasil penelitian Tugas Akhir yang tidak dimuat/atau disertakan di artikel jurnal. Di dalam kertas kerja ini disajikan beberapa point seperti :

1. Literature review
2. Hasil analisa & perancangan aplikasi
3. Konfigurasi
4. Tahapan eksperimen
5. Hasil eksperimen secara keseluruhan.

