

**IMPLEMENTASI DAN ANALISIS PERFORMA *HIGH*
AVAILABILITY WEB SERVER PADA *CLUSTER KUBERNETES***

TUGAS AKHIR

**Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer**



OLEH:

Muhammad Hafizh Al Adil

09011281924066

**JURUSAN SISTEM KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA**

2024

HALAMAN PENGESAHAN

HALAMAN PENGESAHAN
IMPLEMENTASI DAN ANALISIS PERFORMA *HIGH AVAILABILITY*
WEB SERVER PADA *CLUSTER KUBERNETES*

TUGAS AKHIR

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer

Oleh

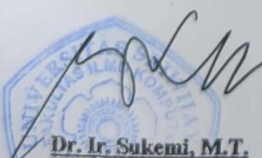
Muhammad Hafizh Al Adil
09011281924066

Palembang, 14 Juli 2024

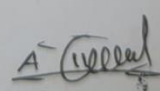
Mengetahui,

Ketua Jurusan Sistem Komputer

Pembimbing Tugas Akhir


Dr. Ir. Sukemi, M.T.

NIP. 196612032006041001


Ahmad Hervanto, S.Kom., M.T.

NIP. 198701222015041002

AUTHENTICATION PAGE

AUTHENTICATION PAGE
IMPLEMENTATION AND PERFORMANCE ANALYSIS OF HIGH
AVAILABILITY WEB SERVER ON KUBERNETES CLUSTER

FINAL TASK

*Submitted To Fulfill One Of The Requirements
To Obtain A Bachelor's Degree In Computer Science*


By

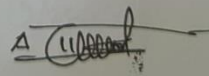
Muhammad Hafizh Al Adil
09011281924066

Palembang, 16 July 2024

Acknowledge,

Head of Computer System Department Supervisor


Dr. Ir. Sukemi, M.T.
NIP. 196612032006041001


Ahmad Hervanto, S.Kom., M.T.
NIP. 198701222015041002

HALAMAN PERSETUJUAN

HALAMAN PERSETUJUAN

Telah diuji dan lulus pada

Hari : Jumat

Tanggal : 12 Juli 2024

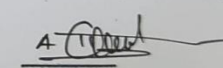
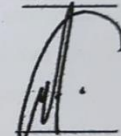
Tim Penguji

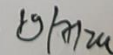
1. Ketua : Kemahyanto Exaudi, M.T.

2. Sekretaris : Muhammad Ali Buchari, M.T.

3. Penguji : Dr. Ahmad Zarkasi, M.T.

4. Pembimbing : Ahmad Heryanto, M.T.



Mengetahui, 

Ketua Jurusan Sistem Komputer



Dr. Ir. Sukemi, M.T.
NIP. 196612032006041001

HALAMAN PERNYATAAN

HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama: Muhammad Hafizh Al Adil

NIM: 09011281924066

Judul: Implementasi dan Analisis Performa High Availability Web Server Pada Cluster Kubernetes

Hasil Pengecekan Plagiat/Turnitin: 3%

Menyatakan bahwa skripsi ini merupakan hasil karya sendiri dan bukan penjiplakan atau plagiat. Apabila terbukti adanya unsur penjiplakan atau plagiat, saya bersedia menerima sanksi akademik dari Universitas Sriwijaya.

Demikian persyaratan ini saya buat dalam keadaan sadar dan tanpa adanya paksaan dari pihak manapun.



Palembang, Juli 2024



Muhammad Hafizh Al Adil

NIM: 09011281924066

IMPLEMENTASI DAN ANALISIS PERFORMA *HIGH AVAILABILITY* *WEB SERVER* PADA *CLUSTER KUBERNETES*

Muhammad Hafizh Al Adil (09011281924066)

Jurusan Sistem Komputer, Fakultas Ilmu Komputer, Universitas Sriwijaya

Email: 09011281924066@student.unsri.ac.id

ABSTRAK

Pada masa kini semakin meningkat pengguna internet yang mengakses website pada waktu tertentu yang mana dapat menyebabkan kinerja web server menjadi buruk hingga mengalami downtime. Maka dari itu diperlukan sebuah sistem yang dapat aktif menjalankan web server secara terus menerus tanpa gangguan, yaitu *Kubernetes*. *Kubernetes* menerapkan *cluster based* yang terdiri dari satu *node master* dan beberapa *node worker*, bila *node master* mengalami kegagalan maka *Kubernetes* tidak dapat mengontrol lingkungan *cluster*, untuk mengatasi masalah tersebut dapat menggunakan *High Available Cluster Kubernetes*. Untuk mengetahui apakah *High Available Cluster Kubernetes* efektif dalam menangani beban kinerja yang tinggi akan dilakukan perbandingan dengan *Regular Cluster Kubernetes* dengan melakukan *load testing*, *failover simulation* dan pengukuran *availability*. Berdasarkan hasil pengujian diketahui pada *load testing*, *High Available Cluster Kubernetes* unggul pada parameter *response time*, *packet loss*, *throughput*, *CPU usage cluster*, *CPU usage node worker* dan *memory usage*, yang mana menunjukkan bahwa dengan menambahkan *pod* web server dan *node worker* dapat meningkatkan ketersediaan website, sedangkan *Regular Cluster Kubernetes* unggul pada parameter *CPU usage node master*, hal tersebut dikarenakan *Regular Cluster Kubernetes* lebih sedikit mengontrol komponen pada cluster dibandingkan *High Available Cluster Kubernetes* dan *node master* tidak menggunakan banyak CPU pada saat *load testing* sehingga menambahkan node master tidak disarankan.

Kata Kunci: Web Server, High Availability, Kubernetes, Server Cluster, Kontainer

IMPLEMENTATION AND PERFORMANCE ANALYSIS OF HIGH AVAILABILITY WEB SERVER ON KUBERNETES CLUSTER

Muhammad Hafizh Al Adil (09011281924066)

Department of Computer Systems, Faculty of Computer Science, Sriwijaya University

Email: 09011281924066@student.unsri.ac.id

ABSTRACT

Nowdays, there is a growing number of internet users accessing websites at specific intervals, which can result in degraded web server performance and even lead to downtime. Therefore, a system capable of continuously operating the web server without interruption is required, such as Kubernetes. Kubernetes implements a cluster-based architecture consisting of a master node and several worker nodes. If the master node fails, Kubernetes cannot control the cluster environment. To address this issue, a High Availability Cluster Kubernetes can be employed. To evaluate the effectiveness of a High Availability Cluster Kubernetes in handling high-performance loads, a comparison with a Regular Cluster Kubernetes will be conducted through load testing, failover simulation, and availability measurement. Based on the test results, it has been observed that in load testing, the High Availability Cluster Kubernetes outperforms in parameters such as response time, packet loss, throughput, cluster CPU usage, worker node CPU usage, and memory usage. This indicates that adding web server pods and worker nodes can enhance website availability. Conversely, the Regular Cluster Kubernetes excels in the master node CPU usage parameter, as it controls fewer components in the cluster compared to the High Availability Cluster Kubernetes, resulting in lower CPU usage by the master node during load testing. Consequently, adding a master node is not recommended.

Keywords: *Web Server, High Availability, Kubernetes, Server Cluster, Container*

KATA PENGANTAR

Asslamu'alaikum Wr.Wb

Puji dan syukur Allhamdulillah penulis panjatkan atas kehadiran Allah SWT, karena berkat rahmat dan karunia-nya lah sehingga penulis dapat menyelesaikan penyusunan Tugas Akhir ini dengan judul “**Implementasi dan Analisis Performa *High Availability Web Server* Pada *Cluster Kubernetes*”.**

Pada kesempatan ini, penulis ingin mengucapkan terima kasih kepada pihak yang telah memberikan bantuan, dorongan, motivasi, semangat, dan bimbingan dalam penyusunan proposal skripsi ini, yakni:

1. Allah SWT, yang telah melimpahkan Berkat dan Rahmat-Nya sehingga penulis dapat menyelesaikan Tugas Akhir dengan baik dan lancar.
2. Kedua orang tua saya yang telah membesarkan saya, tetap mendorong untuk dapat mencapai impian saya dan selalu mendoakan saya.
3. Bapak Prof. Erwin, S.Si.,M.Si., selaku Dekan Fakultas Ilmu Komputer Universitas Sriwijaya.
4. Bapak Dr. Ir. Sukemi, M.T., selaku Ketua Jurusan Sistem Komputer Universitas Sriwijaya.
5. Ibu Prof. Dr. Ir. Siti Nurmaini, M.T., selaku Dosen Pembimbing Akademik.
6. Bapak Ahmad Heryanto,S.Kom, M.T., selaku Dosen Pembimbing Tugas Akhir yang telah bersedia membimbing penulis dan membantu menyelesaikan Tugas Akhir penulis.
7. Mbak Sari selaku admin Jurusan Sistem Komputer Unggulan yang telah membantu pemberkasan Tugas Akhir ini.
8. Khalifaturahman Al Hazri Repaldi, selaku teman seperjuangan saya yang telah membantu dan menemani saya dalam pengerjaan Tugas Akhir
9. Mega Aulia Putri, yang telah memberikan motivasi dan semangat selama proses pengerjaan Tugas Akhir saya.
10. Dan semua pihak yang telah membantu.

Dalam penyusunan Skripsi ini saya menyadari sepenuhnya bahwa laporan ini masih jauh dari kata sempurna, oleh karena itu saya mengharapkan saran dan kritik dari semua pihak yang berkenan demi laporan yang lebih baik lagi.

Akhir kata, saya harap semoga Skripsi ini dapat bermanfaat serta dapat memberikan pengetahuan dan wawasan bagi semua pihak yang membutuhkannya.

Palembang, Juli 2024

Muhammad Hafizh Al Adil

NIM. 09011281924066

DAFTAR ISI

HALAMAN PENGESAHAN	ii
<i>AUTHENTICATION PAGE</i>	iii
HALAMAN PERSETUJUAN	iv
HALAMAN PERNYATAAN	v
ABSTRAK	vi
<i>ABSTRACT</i>	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xiv
DAFTAR TABEL	xv
DAFTAR GRAFIK	xvi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan	3
1.4. Manfaat	4
1.5. Batasan Masalah	4
1.6. Metodologi Penelitian	4
1.7. Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA	7
2.1. Kajian Pustaka	7
2.2. Web Server	15
2.3. Server Cluster	17

2.4.	Virtualisasi	17
2.5.	Kontainer	17
2.6.	Kubernetes	20
2.6.1.	Node Master	21
2.6.1.1.	Kube-apiserver	22
2.6.1.2.	Kube-Controller-Manager	22
2.6.1.3.	Kube-Scheduler	22
2.6.1.4.	Etc	22
2.6.2.	Node Worker	22
2.6.2.1.	Kube-Proxy	22
2.6.2.2.	Kubelet	23
2.6.2.3.	Container Runtime	23
2.6.3.	Pods	23
2.6.4.	Workloads Resource	24
2.6.4.1.	Jobs	24
2.6.4.2.	StatefulSet	24
2.6.4.3.	DaemonSet	24
2.6.4.4.	Deployment	24
2.6.5.	Network Service	25
2.6.5.1.	ClusterIP	25
2.6.5.2.	NodePort	25
2.6.5.3.	Load Balancer	25
2.7.	High Availability	25
2.8.	Program Pendukung Penelitian	28
2.8.1.	Apache HTTP Server	28

2.8.2.	<i>Docker</i>	28
2.8.3.	<i>Keepalived</i>	28
2.8.4.	<i>Haproxy</i>	29
2.8.5.	<i>Kubeadm</i>	29
2.8.6.	<i>Kubectl</i>	29
2.8.7.	<i>Flannel</i>	29
2.8.8.	<i>Metallb</i>	30
2.8.9.	<i>Metric Server</i>	30
2.8.10.	<i>Apache Benchmark</i>	31
BAB III METODOLOGI PENELITIAN		32
3.1.	Pendahuluan	32
3.2.	Kerangka Kerja Penelitian	33
3.3.	Topologi Penelitian	35
3.4.	Perancangan Sistem	36
3.4.1.	Kebutuhan Perangkat Keras	36
3.4.2.	Kebutuhan Perangkat Lunak	37
3.5.	Membuat <i>Image Docker</i>	38
3.6.	Perancangan <i>Cluster</i>	40
3.6.1.	Perancangan <i>Load Balancer</i>	40
3.6.1.1.	Perancangan <i>Keepalived</i>	40
3.6.1.2.	Perancangan <i>HAProxy</i>	43
3.6.2.	Perancangan <i>Kubernetes</i> Untuk <i>Node Master</i> dan <i>Worker</i>	44
3.6.3.	Inisiasi <i>Cluster</i> Menggunakan <i>Kubeadm</i>	46
3.6.4.	Konfigurasi <i>Kubectl</i>	47
3.6.5.	Instalasi <i>Flannel</i>	48

3.6.6.	<i>Instalasi Metric Server</i>	49
3.6.7.	<i>Instalasi Metallb</i>	50
3.6.8.	<i>Menyebarkan Web Server Apache</i>	51
3.7.	<i>Load Testing</i>	54
3.8.	<i>Failover Simulation</i>	57
3.9.	<i>Pengukuran Availability</i>	59
BAB IV HASIL DAN PEMBAHASAN		60
4.1.	Pendahuluan	60
4.2.	<i>Load Testing</i>	60
4.2.1.	<i>Response Time</i>	60
4.2.2.	<i>Packet Loss</i>	63
4.2.3.	<i>Throughput</i>	65
4.2.4.	<i>CPU Usage</i>	67
4.2.5.	<i>Memory Usage</i>	72
4.3.	<i>Failover Simulation</i>	77
4.4.	<i>Pengukuran Availability</i>	80
BAB V KESIMPULAN DAN SARAN		83
5.1.	Kesimpulan	83
5.2.	Saran	84
DAFTAR PUSTAKA		85
LAMPIRAN		91

DAFTAR GAMBAR

Gambar 2. 1	Cara Kerja HTTP.....	16
Gambar 2. 2	Perbedaan Virtualisasi Tradisional dan Kontainer.	18
Gambar 2. 3	Arsitektur Kubernetes.....	21
Gambar 2. 4	Komponen Pada Pod.....	23
Gambar 2. 5	Cara Kerja Metric Server.....	31
Gambar 3. 1	Kerangka Kerja Penelitian.....	33
Gambar 3. 2	Topologi High Available Cluster Kubernetes.	35
Gambar 3. 3	Topologi Regular Cluster Kubernetes.	36
Gambar 3. 4	Buat Repositori di Docker Hub.	39
Gambar 3. 5	Status Node.....	48
Gambar 3. 6	Status Pod Flannel.	49
Gambar 3. 7	Status Pod Metics Server.....	49
Gambar 3. 8	Status Pod Metallb.....	51
Gambar 3. 9	Status Pod Apache Web Server.....	53
Gambar 3. 10	Website Halaman Index.....	53
Gambar 3. 11	Website Halaman Profile.....	54
Gambar 3. 12	Keluaran Load Testing Menggunakan Apache Benchmark.....	56
Gambar 3. 13	Keluaran Metric Server.....	57
Gambar 3. 14	Keluaran Kubectl Get Nodes.	58
Gambar 3. 15	Website Saat Web Server Nonaktif.	58
Gambar 4. 1	Hasil Log Controller Manager.....	79
Gambar 4. 2	Hasil Log Controller Manager.....	79
Gambar 4. 3	Hasil Log Controller Manager.....	80

DAFTAR TABEL

Tabel 2. 1 Refrensi Penelitian.	7
Tabel 2. 2 Five Nine Availability.	27
Tabel 3. 1 Kebutuhan Perangkat Keras.	36
Tabel 3. 2 Kebutuhan Perangkat Lunak.	37
Tabel 3. 3 Skenario Load Testing.	54
Tabel 3. 4 Skenario Failover Simulation Pada High Available Cluster Kubernetes.	57
Tabel 3. 5 Skenario Failover Simulation Pada High Regular Cluster Kubernetes.	57
Tabel 4. 1 Hasil Response Time.	61
Tabel 4. 2 Hasil Packet Loss.	63
Tabel 4. 3 Hasil Throughput.	65
Tabel 4. 4. Hasil CPU Usage Cluster.	67
Tabel 4. 5 Hasil CPU Usage Node Master.	69
Tabel 4. 6 Hasil CPU Usage Node Worker.	70
Tabel 4. 7 Hasil Memory Usage.	72
Tabel 4. 8 Rekap Data Load Testing Metode Permintaan GET.	74
Tabel 4. 9 Rekap Data Hasil Load Testing Metode Permintaan POST.	75
Tabel 4. 10 Hasil Failover Simulation Pada High Available Cluster Kubernetes.	78
Tabel 4. 11 Hasil Failover Simulation Pada Regular Cluster Kubernetes.	78
Tabel 4. 12 Hasil Pengukuran Availability Pada High Available Cluster Kubernetes.	81
Tabel 4. 13 Hasil Pengukuran Availability Pada Regular Cluster Kubernetes.	81

DAFTAR GRAFIK

Grafik 4. 1 Hasil Response Time.....	62
Grafik 4. 2 Hasil Packet Loss.....	64
Grafik 4. 3 Hasil Throughput.....	66
Grafik 4. 4 Hasil CPU Usage Cluster	68
Grafik 4. 5 Hasil CPU Usage Node Master.	70
Grafik 4. 6 Hasil CPU Usage Node Worker.	71
Grafik 4. 7 Hasil Memory Usage.....	73
Grafik 4. 8 Hasil Rekap Data Load Testing Metode Permintaan GET.....	75
Grafik 4. 9 Hasil Rekap Data Load Testing Metode Permintaan POST.....	76

BAB I PENDAHULUAN

1.1. Latar Belakang

Website menjadi kebutuhan individu untuk mendapatkan berbagai macam informasi dan memudahkan berbagai macam pekerjaan. Untuk dapat mengakses sebuah *website*, pengguna dapat menggunakan *web browser* melalui komputer pengguna. *Web browser* akan berkomunikasi dengan *server* yang menyimpan *website* tersebut melalui *web server* menggunakan protokol HTTP (*Hypertext Transfer Protocol*) dengan cara *web browser* mengirimkan permintaan HTTP ke *web server*, *web server* akan menerima permintaan HTTP tersebut dan mengirimkan balasan yang diminta pengguna berupa *website* [1], [2]. Agar *website* dapat diakses oleh siapa saja dan kapan saja, *web server* harus dapat aktif menangani permintaan HTTP dan meresponsnya tanpa gangguan.

Hingga saat ini pengguna internet semakin meningkat, sehingga semakin banyak yang mengakses *website* pada waktu tertentu menyebabkan kinerja *web server* menjadi buruk hingga menyebabkan kegagalan pada *web server*. Oleh karena itu diperlukan sistem yang dapat menjalankan aplikasi *web server* dalam tingkat tinggi, secara terus menerus tanpa gangguan pada jangka waktu tertentu. Salah satu cara untuk mengatasi masalah tersebut dapat menggunakan *server cluster*, yaitu sekumpulan *server* yang saling terhubung dan bekerja secara bersama-sama, dengan menggunakan *server cluster* dapat meningkatkan ketersediaan layanan *web server* [2], [3]. *Server cluster* dapat diterapkan secara virtualisasi yaitu sebuah teknologi yang dapat mengabstraksikan perangkat keras menjadi beberapa mesin virtual yang disebut VM dengan cara membagi sumber daya komputasi seperti CPU dan *memory* dari komputer fisik dan dapat dijalankan secara bersamaan [4], [5].

Karena VM mengabstraksikan lingkungan komputasi fisik menjadi virtual, maka memerlukan sistem operasi yang berjalan pada VM tersebut sehingga memakan sumber daya yang lebih banyak. Agar aplikasi *web server* dapat dijalankan sesuai dengan sumber daya yang dibutuhkan dapat menggunakan

teknik virtualisasi secara kontainer. Kontainer adalah sebuah paket perangkat lunak ringan yang mengemas kode dan semua dependensinya serta menjalankannya dalam lingkungan yang terisolasi dan mandiri [6], [7]. Dengan menggunakan kontainer aplikasi dapat berjalan lebih ringan, efisien dan fleksibel dibandingkan virtualisasi tradisional, selain itu lebih mudah dibuat dan didistribusikan.

Kontainer bersifat ringan dan mudah dijalankan, agar dapat menangani beban yang tinggi maka diperlukan sebuah mekanisme untuk mengolah kontainer dalam jumlah yang banyak secara otomatis dengan cara mengorkestrasikan kontainer [7]. Selain itu dengan mengorkestrasi kontainer, dapat menerapkan aplikasi yang sama dalam lingkungan berbeda tanpa mendesain ulang aplikasi tersebut serta mekanisme orkestrasi kontainer menggunakan layanan mikro (*microservice*) sehingga dapat mempermudah pengaturan layanan seperti penyimpanan, jaringan dan keamanan [7], [8].

Salah satu alat yang dapat digunakan untuk orkestrasi kontainer pada skala besar yaitu *Kubernetes*. *Kubernetes* adalah sebuah platform orkestrasi kontainer yang bersifat *open source* yang dapat digunakan untuk mengolah penyebaran aplikasi bersifat kontainer dan menyediakan otomatisasi dan konfigurasi yang bersifat deklaratif [8]–[10]. *Kubernetes* bersifat *cluster based*, yang mana sekumpulan *node* atau *server* yang membentuk sebuah *cluster* yang terdiri dari *node master (control plane)* yang bertugas untuk mengatur seluruh lingkungan *cluster* dan *node worker* yang bertugas untuk menjalankan aplikasi yang disebarkan [8]–[10].

Secara umum *cluster Kubernetes* terdiri dari satu *node master* dan beberapa *node worker* [11]. Bila terjadi permintaan jaringan yang tinggi pada waktu tertentu yang dapat membuat *node master* mati, maka *node master* tidak dapat mengatur *cluster* secara otomatis dan dapat membuat *cluster* tidak berjalan dengan baik bila terjadi kegagalan. Oleh karena itu untuk meningkatkan ketersediaan *cluster* dapat mengimplementasikan *High Available Cluster Kubernetes*, yaitu dengan cara menambah *master node* dan *node worker* sebagai redundansi serta memasang *load*

balancer pada *cluster* agar lalu lintas jaringan dapat didistribusikan ke setiap *node* [12].

Berdasarkan latar belakang penelitian tersebut, mengetahui kualitas kinerja *High Available Cluster Kubernetes* dengan dilakukan dengan *load testing* dengan jumlah permintaan yang tinggi. Selain itu untuk mengetahui apakah web server masih dapat memberikan layanan ketika salah satu *node* nonaktif dan waktu yang dibutuhkan *Kubernetes* untuk menangani *failover* jika web server mengalami *downtime* dapat dilakukan dengan melakukan *failover simulation*. Sehingga pada penelitian yang diajukan pada proposal tugas akhir ini berjudul “IMPLEMENTASI DAN ANALISIS PERFORMA *HIGH AVAILABILITY WEB SERVER* PADA *CLUSTER KUBERNETES*”.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan sebelumnya, maka didapatkan perumusan masalah yaitu:

1. Apakah menerapkan *web server* pada *High Available Cluster Kubernetes* lebih efektif dalam tetap aktif menangani beban jaringan secara terus menerus pada waktu tertentu tanpa mengalami kegagalan dibandingkan *Regular Cluster Kubernetes*.
2. Apakah dengan mengimplementasikan *web server* pada *High Available Cluster Kubernetes* dapat meningkatkan ketersediaan dan mengurangi *downtime* pada web server.

1.3. Tujuan

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

1. Menganalisis kinerja *web server* pada *High Available Cluster Kubernetes* dengan membandingkannya dengan kinerja *web server* pada *Regular Cluster Kubernetes*.
2. Mengetahui apakah mengimplementasikan *web server* pada *High Available Cluster Kubernetes* dapat meningkatkan ketersediaan dan mengurangi *downtime* dengan melakukan simulasi *failover*.

1.4. Manfaat

Adapun manfaat yang dapat diambil dari penelitian ini adalah sebagai berikut:

1. Sebagai solusi untuk menerapkan *web server cluster* menggunakan *platform Kubernetes*.
2. Untuk mengetahui kemampuan *High Available Cluster Kubernetes* dalam menangani beban jaringan yang tinggi.

1.5. Batasan Masalah

Berikut batasan masalah pada penelitian tugas akhir ini antara lain:

1. Penelitian yang dilakukan berfokus pada analisis kinerja *web server* pada *High Available Cluster Kubernetes*.
2. *High Available Cluster Kubernetes* dirancang pada *server bare metal* dengan sumber daya komputasi yang terbatas.
3. *Web server* yang diterapkan di *cluster Kubernetes* meng-hosting *website* statis.

1.6. Metodologi Penelitian

Untuk metodologi yang digunakan pada penelitian tugas akhir memiliki beberapa tahapan meliputi:

1. Metode Studi Pustaka dan Literatur

Pada tahapan ini peneliti melakukan pengumpulan informasi dan data mengenai penelitian tugas akhir yang akan dilakukan dengan membaca artikel atau makalah penelitian-penelitian sebelumnya.

2. Metode Perancangan Sistem

Pada tahapan ini dilakukan proses perancangan sistem dengan menggunakan metode atau pendekatan tertentu, serta menentukan perangkat keras dan perangkat lunak apa saja yang akan diperlukan untuk melakukan penelitian ini dan bagaimana cara menerapkan metode yang akan dilakukan.

3. Metode Pengujian

Pada tahapan ini dilakukan setelah perancangan sistem, di mana hasil rancangan akan dilakukan pengujian berdasarkan metodologi penelitian sehingga mendapatkan hasil data uji yang sesuai dan tepat secara konsep maupun praktisi.

4. Metode Analisis

Pada tahapan ini dilakukan analisa hasil data uji yang didapatkan pada proses pengujian rangkaian sistem berdasarkan parameter yang telah ditentukan agar mendapatkan hasil yang objektif.

5. Metode Kesimpulan dan saran

Pada tahapan ini dilakukan penarikan kesimpulan berdasarkan penelitian tugas akhir yang telah dilakukan serta memberikan saran agar dapat di kembangkan pada penelitian selanjutnya.

1.7. Sistematika Penulisan

Sistematika penulisan pada penelitian ini sebagai berikut:

BAB I. PENDAHULUAN

Pada BAB I, berisikan latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika penulisan.

BAB II. TINJAUAN PUSTAKA

Pada BAB II, berisikan pembahasan teori dasar tentang yang terkait dengan penelitian tugas akhir.

BAB III. METODOLOGI PENELITIAN

Pada BAB III, membahas tentang tahapan perancangan sistem dan penerapan metode penelitian, serta lingkungan *hardware* dan *software* yang digunakan.

BAB IV. HASIL DAN ANALISIS

Pada BAB IV. Membahas mengenai hasil dan juga Analisa dari performansi *High Availability* pada *cluster Kubernetes*.

BAB V. KESIMPULAN DAN SARAN

Pada BAB V. berisikan kesimpulan dari hasil penelitian dan memberikan saran untuk pengembangan sistem pada penelitian berikutnya.

DAFTAR PUSTAKA

- [1] D. Gourley and B. Totty, *HTTP: The Definitive Guide*, First Edit. Sebastopol: O'Reilly Media, Inc, 2002.
- [2] N. Yeager J and R. McGrath E, *Web Server Technology*. San Francisco: Morgan Kaufman Publisher, 1996.
- [3] NGINX, "What Is a Web Server? Deliver Content Fast - NGINX," *Nginx*, 2022. <https://www.nginx.com/resources/glossary/web-server/> (accessed Dec. 04, 2023).
- [4] D. Kusnetzky, *Virtualization: A Manager's Guide*, First Edit. Sebastopol: O'Reilly Media, Inc, 2011.
- [5] IBM, "What is Virtualization__ IBM," *IBM*, 2023. <https://www.ibm.com/topics/virtualization> (accessed Dec. 10, 2023).
- [6] K. Matthias and S. P. Kane, *Docker: Up & Running*, First. Sebastopol: O'Reilly Media, Inc, 2015.
- [7] K. Hermans, *Mastering Data Containerization and Orchestration: A Comprehensive Guide to Learn Data Containerization and Orchestration*. Cybellium, 2023.
- [8] G. Kuncoro, A. C. Risdianto, and O. W. Purbo, *KUBERNETES UNTUK PEMULA*. Yogyakarta: Penerbit Andi, 2023.
- [9] G. Sayfan and B. Ibryam, *Mastering Kubernetes: Dive Into Kubernetes and Learn how to Create and Operate World-class Cloud-native Systems*, Fourth Edi. Birmingham: Packt Publishing, 2023.
- [10] K. Hightower and B. Burns, *Kubernetes: Up and Running*, First Edit. Sebastopol: O'Reilly Media, Inc, 2017.
- [11] Kubernetes, "Overview | Kubernetes," *The Linux Foundation*, 2023. <https://kubernetes.io/docs/concepts/overview/> (accessed Dec. 01, 2023).
- [12] T. Critchley, "What Is High Availability?," *High Availability IT Services*, 2014. <https://www.nginx.com/resources/glossary/high-availability/> (accessed Dec. 04, 2023).