



Implementasi Cloud Computing Cluster Menggunakan Microstack Untuk Layanan Web

Muhammad Arif Fadhly Ridha¹, Khairul Gunawan*²

¹Teknik Informatika, Politeknik Caltex Riau, Pekanbaru, Indonesia

²Teknologi Rekayasa Komputer, Politeknik Caltex Riau, Pekanbaru, Indonesia

¹fadhly@pcr.ac.id, ²*khairul22trk@mahasiswa.pcr.ac.id

*Corresponding Author

Diserahkan: 19 Juni 2023

Diterima: 14 Juli 2023

Diterbitkan: 30 November 2023

ABSTRAK

Cloud Computing adalah sistem komputerisasi berbasis jaringan/internet, dimana suatu sumber daya, software, informasi dan aplikasi disediakan untuk digunakan oleh komputer lain yang membutuhkan. Cloud Computing sering digunakan oleh perusahaan untuk menunjang layanan web perusahaan tersebut. Permasalahan yang sering dialami dalam cloud computing adalah downtime. Dimana permasalahan ini terjadi karena file system atau sever web tersebut error dan tidak memiliki backup. Permasalahan ini dapat di atasi dengan cara teknik cluster cloud computing yang dapat menyediakan layanan web yang memiliki high availability. Cluster terdiri dari server yang berbeda bekerja sama untuk memastikan bahwa downtime dari sumber daya kritis dikurangi seminimal mungkin dengan tujuan yaitu memastikan bahwa sumber daya kritis mencapai ketersediaan maksimum yang disebut sebagai high availability. Untuk penerapan cluster cloud computing kita membutuhkan platform yang memiliki tools yang dapat menunjang kebutuhan tersebut. Salah satu platform yang dapat digunakan adalah OpenStack, OpenStack adalah perangkat lunak bebas dan open-source software platform untuk cloud computing, sebagian besar digunakan sebagai Infrastructure as a Service (IaaS). Salah satu package yang dirilis OpenStack adalah MicroStack yang berisi tools dari OpenStack untuk membangun cluster pada cloud computing. Hasil dari penelitian ini adalah tercapainya high availability pada layanan web server dimana layanan web masih dapat diakses saat ada dua node dimatikan dan saat salah satu server dimatikan. Penggunaan CPU meningkat ketika ada beberapa jumlah permintaan http request saat beberapa node dimatikan dan juga saat salah satu server fisik dimatikan. Penggunaan memori juga meningkat ketika ada beberapa node dimatikan dan juga saat salah satu server fisik dimatikan.

Kata kunci: *Cloud Computing, Cluster, Microstack*

ABSTRACT

Cloud Computing is a network/internet-based computerized system, where resources, software, information and applications are provided for use by other computers in need. Cloud Computing is often used by companies to support the company's web services. The problem that is often experienced in cloud computing is downtime. Where this problem occurs because the file system or web server is an error and does not have a backup. This problem can be overcome by means of cloud computing cluster techniques that can provide web services that have high availability. Clusters consist of different servers working together to ensure that downtime of critical resources is reduced to a minimum with the aim of ensuring that critical resources reach maximum availability which is known as high availability. For

the implementation of cloud computing clusters, we need a platform that has tools that can support these needs. One of the platforms that can be used is OpenStack, OpenStack is a free software and open-source software platform for cloud computing, mostly used as Infrastructure as a Service (IaaS) . One of the packages released by OpenStack is MicroStack which contains tools from OpenStack to build clusters in cloud computing. The result of this research is the achievement of high availability on web server services where web services can still be accessed when two nodes are turned off and when one of the servers is turned off. CPU usage increases when there are multiple number of http requests when multiple nodes are shut down and also when one of the physical servers is shutting down. Memory usage also increases when multiple nodes are shut down and also when one of the physical servers is shutting down.

Keywords: *Cloud Computing, Cluster, Microstack*

1. PENDAHULUAN

Dengan perkembangan teknologi yang sangat pesat, setiap orang atau otoritas pasti membutuhkan informasi dan komunikasi untuk berbagai kebutuhan sehari-hari saat ini. Tentu saja, berbagai informasi yang ingin diambil membutuhkan koneksi internet. Berkat teknologi internet, informasi yang dibutuhkan semua orang tersedia dengan mudah dan cepat. Bahkan dengan koneksi internet, setiap orang lebih mudah untuk berkomunikasi dengan cepat dan akurat . (H & Dkk, 2016)

Penggunaan internet juga telah memberikan kemudahan bagi pengguna untuk membuat website, blog, sistem informasi, dan lain-lain yang menggunakan web server. Biasanya, situs web dipasang di server fisik yang juga menyimpan situs web lain, jadi tidak ada bagian khusus untuk situs web tersebut. Sebuah situs web berbagi sumber daya dengan situs web lain. Jadi, ketika pengguna mengakses beberapa halaman pada saat yang sama, sumber daya, layanan, dan perangkat keras yang tersedia secara otomatis digunakan pada waktu yang sama. Sumber daya ini adalah prosesor, memori, RAM, dan perangkat keras .

Selain itu, permasalahan yang sering dihadapi oleh instansi adalah kurangnya high availability, sehingga setiap kali layanan web server tiba-tiba terhenti maka seluruh proses bisnis yang berjalan pada web server juga ikut berhenti, perkembangan infrastruktur IT juga memaksa masing-masing instansi tersebut untuk menggunakan inovasi terbaru dan banyaknya jaringan yang terkoneksi. melaksanakan. perangkat tentu saja juga meningkat. Oleh karena itu, diperlukan suatu sistem yang dapat berbagi sumber daya yang ada, sehingga setiap situs memiliki area sumber daya khusus dan semua layanan tersedia secara real time dan kapan saja, yang juga menjamin ketersediaan layanan yang tinggi, karena semua data sesuai di server data yang ketersediaannya tinggi .

Masalah ini dapat diatasi dengan teknologi cloud computing cluster yang dapat menyediakan layanan jaringan dengan ketersediaan tinggi. Komputasi awan adalah proses singkat pemrosesan energi melalui Internet untuk menjalankan program melalui komputer yang terhubung satu sama lain dalam waktu yang sama. Komputasi awan juga merupakan teknologi yang mengubah Internet menjadi server terpusat untuk mengelola data dan aplikasi pengguna. Tentunya dengan mengimplementasikan sistem cloud computing, pembagian ini dapat diwujudkan, resource yang tersedia dapat dibagi antara beberapa mesin virtual. Cluster terdiri dari berbagai server yang bekerja sama untuk memastikan downtime untuk sumber daya kritis diminimalkan. Tujuannya adalah untuk memastikan bahwa sumber daya kritis mencapai ketersediaan maksimum, yang dikenal sebagai ketersediaan tinggi. Untuk mengimplementasikan cluster cloud computing, diperlukan sebuah platform dengan tools yang dapat mendukung kebutuhan tersebut .

Salah satu platform cloud computing yang mayoritas di gunakan oleh perusahaan yaitu OpenStack. OpenStack adalah perangkat lunak bebas dan open- source software platform untuk cloud computing, sebagian besar digunakan sebagai Infrastructure as a Service (IaaS), dimana server virtual dan sumber daya lain tersedia untuk pelanggan . Platform perangkat lunak terdiri dari komponen- komponen yang

saling terkait yang mengendalikan beragam perangkat keras multi- vendor yang meliputi pengolahan, penyimpanan dan sumber daya jaringan di seluruh data center. Pengguna dapat mengelolanya melalui dasbor berbasis web, hingga command-line tools, atau REST layanan web.

Salah satu package yang dirilis OpenStack adalah MicroStack dengan Ubuntu sebagai sistem operasinya. MicroStack dapat memberikan kemudahan bagi para pengguna dalam mengakses berbagai informasi yang dibutuhkan melalui pemanfaatan internet. Perbedaan MicroStack dengan package OpenStack lainnya adalah instalasi yang lebih cepat, tingkat keamanannya mutakhir, memiliki komponen OpenStack yang lengkap, dan ukurannya lebih kecil. Untuk mengatasi kelemahan dalam mengakses beberapa halaman web yang dilakukan sekaligus, umumnya sebuah instansi akan membeli perangkat server yang baru. Namun hal tersebut tidaklah efisien karena akan membuat ruangan pada instansi tersebut menjadi full dengan perangkat server. Oleh karena itu, dibangunlah web server pada cloud computing cluster dengan menggunakan MicroStack.

2. TINJAUAN PUSTAKA

Pada penelitian ini terdapat beberapa penelitian terdahulu yang menjadi rujukan dan landasn teori. Pada topik penelitian pertama dengan judul “Implementasi Cluster Pada Web Server Berbasis Cloud Computing”. Menurut penelitian yang telah dilakukan menyatakan bahwa Rancangan sistem Cluster pada web server berbasis Cloud Computing ini berhasil dilakukan dengan tahapan penelitian Perancangan jaringan, Implementasi rancangan sistem dan pengujian sistem. Pada implementasinya sistem ini mampu meminimalisir masalah downtime dengan nilai 10 detik pada saat terjadi proses failover. Sistem Cluster pada web server juga telah berhasil dibangun diatas platform virtual berbasis Proxmox yang didalamnya terpasang sistem operasi Debian Squeeze pada platform KVM. Hasil yang muncul cukup mumpuni untuk skala mesin virtual dalam hal pelayanan yang mendekati mesin server real[1].

Pada topik penelitian kedua dengan judul “ Analisis dan pembangunan infrastruktur cloud computing”. Infrastruktur yang dibangunnya adalah server as a service (SaaS) yang ditujukan untuk pemenuhan kebutuhan kegiatan praktikum siswa khususnya mata pelajaran jaringan komputer siswa kelas sebelas. Hasil dari penelitian ini yaitu prototype IaaS yang dimana prototype IaaS tersebut memiliki kinerja yang lebih baik dibandingkan dengan sistem yang sedang berjalan. Peneliti menggunakan Ubuntu Enterprise Cloud (UEC) sebagai aplikasi untuk membangun IaaS[2].

Pada topik penelitian ke tiga yang berjudul “Implementasi dan analisis kinerja private cloud computing di lingkungan GDLN universitas udayana” yang melakukan penelitian implementasi dan analisis kinerja private cloud computing menggunakan platform openstack[3].

Pada topik penelitian terakhir yang berjudul “Implementasi web server pada cloud computing dengan OpenVZ “yang menyatakan web server dapat dibangun pada jaringan cloud computing dengan virtualisasi OpenVZ . Web server berbasis cloud computing layanannya tetap bisa diakses oleh client walaupun ada beberapa node yang mati. Dengan saran untuk penelitian selanjutnya bisa dengan menggabungkan dua server fisik atau clustering[4].

2.1 Landasan Teori

2.1.1 *Cloud Computing*

Cloud Computing adalah sistem komputerisasi berbasis jaringan/*internet*, dimana suatu sumber daya, *software*, informasi dan aplikasi disediakan untuk digunakan oleh komputer lain yang membutuhkan. Model *billing* dari layanan ini umumnya mirip dengan modem layanan publik. Ketersediaan sesuai kebutuhan, mudah untuk di kontrol, dinamik dan skalabilitas yang hampir tanpa limit adalah beberapa atribut penting dari *cloud computing*. [5]

2.1.2 Node

Sistem komputer terdiri dari beberapa *node*, paling sederhananya terdiri atas dua *node*. *Node* adalah computer yang mandiri dimana artinya mampu untuk memproses tugas komputasi tanpa *node* lain[6].

2.1.3 Cluster

Cluster terdiri dari set komputer terhubung dengan kerja sama sehingga banyak hal tetapi dilihat sistemnya tunggal. Tidak seperti komputer jaringan, komputer *cluster* komputer memiliki setiap *node* set untuk melakukan tugas yang sama, dikendalikan dan dijadwal oleh perangkat lunak. Komponen *cluster* biasanya terhubung satu dengan yang lain melalui jaringan Local Area Network (LAN) dengan setiap *node* komputer digunakan sebagai *server* berjalan sendiri dari sebuah sistem operasi[7].

2.1.4 OpenStack

OpenStack adalah *platform* perangkat lunak komputasi awan sumber terbuka gratis yang digunakan terutama sebagai layanan Infrastruktur sebagai Layanan (IaaS), di mana *server* virtual dan sumber daya lainnya disediakan untuk pelanggan. Platform perangkat lunak terdiri dari komponen yang saling berhubungan yang mengontrol beberapa perangkat keras dari vendor yang berbeda yang mencakup pemrosesan, penyimpanan, dan sumber daya jaringan di seluruh pusat data. Pengguna dapat mengelolanya melalui dasbor berbasis *web*, alat baris perintah, atau layanan *web* REST[8].

2.1.5 High Availability

High Availability adalah karakteristik sistem untuk melindungi atau memulihkan dari gangguan ringan dalam kerangka waktu singkat dengan cara yang sebagian besar otomatis [9].

2.1.6 Uptime and Downtime

Uptime merupakan lamanya waktu ketika *service* atau aplikasi dalam keadaan tersedia. Downtime merupakan lamanya waktu ketika aplikasi atau *service* tidak tersedia. Biasanya dihitung dari outage terjadi sampai waktu ketika *service* atau aplikasi tersedia kembali.

2.1.7 Web Server

Web server merupakan *server* di *internet* yang melayani koneksi transfer lewat protocol HTTP. *Web server* merupakan salah satu hal yang terpenting dari *server* di *internet* dibandingkan *server* lainnya seperti *ftp server*, *e-mail server*, ataupun *news server*[10].

2.1.8 Virtualisasi

Virtualisasi adalah suatu aplikasi perangkat lunak yang dapat mensimulasikan sumber daya perangkat keras[11].

2.1.9 JMeter

JMeter atau The Apache JMeter™ adalah aplikasi open source berbasis Java yang dapat dipergunakan untuk performance test. JMeter bisa digunakan untuk melakukan load/stress testing *Web Application*, FTP Application dan *Database server* test [12].

2.1.10 MicroStack

MicroStack menyediakan instalasi OpenStack *single-node* dan *multi-node* yang dapat digunakan langsung di *workstation*. Meskipun dirancang untuk pengembang untuk membuat prototipe dan menguji, ini juga cocok untuk perangkat Edge, IoT, dan rumah. MicroStack adalah OpenStack on the fly, artinya semua layanan OpenStack dan pustaka pendukungnya dikemas ke dalam satu paket yang dapat diinstal, ditingkatkan, atau dihapus dengan mudah. MicroStack mencakup semua komponen OpenStack utama seperti Keystone, Nova, Neutron, Glance, dan Cinder . [13]

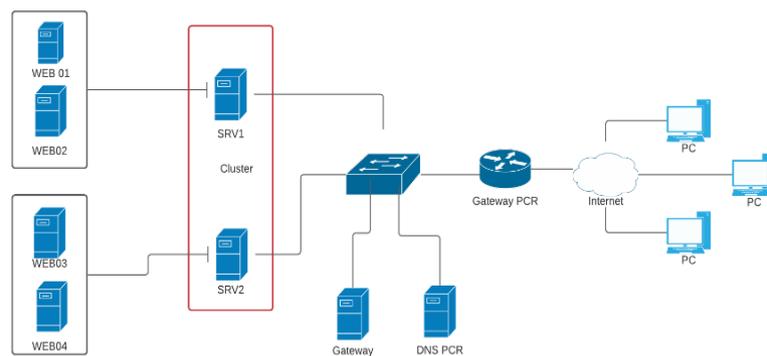
2.1.11 Ubuntu

Pertama kali dirilis pada tahun 2004, Ubuntu adalah sistem operasi dan distribusi Linux berbasis Debian yang gratis dan *open source*. Sistem operasi ini didasarkan pada infrastruktur Debian dan terdiri dari *server*, desktop, dan sistem operasi Linux [14].

3. METODE PENELITIAN

3.1 Topologi

Berikut adalah perancangan topologi jaringan yang akan di buat, beserta tabel pengalamatan yang akan diimplementasikan dalam penelitian ini.

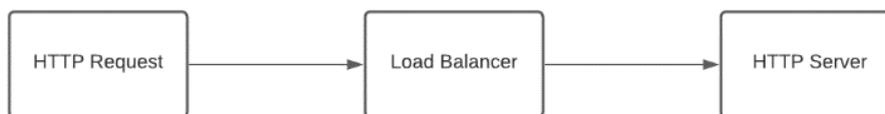


Gambar 1 Topologi

Rancangan topologi diatas ini menjelaskan, yaitu ada 2 server fisik yang masing masing memiliki Virtual Machine web server terhubung ke jaringan publik (internet) milik Politeknik Caltex Riau dan setiap PC Client terhubung ke internet sebagai pengujian terhadap high availability pada SRV1 dan SRV2. Pada topologi ini, SRV1 akan terhubung ke SRV2 dengan menggunakan teknik Cluster menggunakan MicroStack.

3.2 Architecture Logic

Berikut adalah architecture logic yang akan di diimplementasikan dalam penelitian ini.



Gambar 2 Arsitektur Logis

3.3 Skenario Pengujian

Pada penelitian ini skenario yang terjadi adalah *client* dengan jumlah besar dapat mengakses layanan *web* yang tersedia dengan hasil yang diharapkan adalah terciptanya *High Availability*. Pertama beberapa *client* akan mengakses layanan *web*, lalu permintaan tersebut akan diteruskan ke DNS PCR lalu diteruskan ke Gateway yang berada diluar *cluster*. Setelah permintaan diterima oleh *balancer*, lalu *balancer* akan membagi rata permintaan tersebut dan diteruskan ke *web server* yang berada di dalam masing-masing *server* fisik. Pada kasus penelitian ini apabila salah satu *web server* atau *server* fisik mengalami *downtime* atau *error* maka layanan *web* masi tetap bisa berjalan dengan cara *client* akan dibagi rata ke *server* yang masi berfungsi.

3.4 Tabel IP

Berikut adalah Skenario IP yang akan digunakan dalam penelitian ini:

Tabel 1 Skenario IP

Perangkat	Interface	IP Adres	Subnet Mask	Software
SRV1	Eth0		/25	Ubuntu + Microstack
SRV2	Eth0		/25	Ubuntu + Microstack
WEB01	Eth0		/25	Ubuntu + Nginx
WEB02	Eth0		/25	Ubuntu + Nginx
WEB03	Eth0		/25	Ubuntu + Nginx
WEB04	Eth0		/25	Ubuntu + Nginx
Gateway	Eth0	IP Publik	/25	Ubuntu + Nginx
PC-Client	Eth0	IP Private	/24	Widows + Apache JMeter

3.5 Kebutuhan Perangkat

Proses penelitian pada penelitian ini membutuhkan beberapa perangkat dalam proses pengerjaan nya, baik perangkat keras maupun perangkat lunak sebagai berikut:

i) SRV1 dan SRV2

Spesifikasi yang digunakan pada server dapat dilihat pada Tabel 2.

Tabel 2 Spersivikasi SRV1 dan SRV2	
Spesifikasi	
Processore	4 Core
Hard Disk	100 GB
Memory	8 GB
Operating System	Ubuntu 20.04
Software	MicroStack

ii) WEB1-4

Adapun spesifikasi web yang dibutuhkan pada penelitian ini dapat dilihat seperti Tabel 3 berikut

Tabel 3 Spesifikasi Web Server	
Spesifikasi	
Processore	1 Core
Hard Disk	20 GB
Memory	2 GB
Operating System	Ubuntu 20.04
Software	NginX

iii) Gateway

Adapun spesifikasi gateway yang diperlukan selama pengerjaan penelitian dapat dilihat seperti Tabel 4 berikut

Tabel 4 Spesifikasi Gateway	
Spesifikasi	
Processore	2 Core
Hard Disk	100 GB
Memory	8 GB
Operating System	Ubuntu 20.04
Software	NginX

iv) PC-Client

Adapun spesifikasi PC-Client yang dibutuhkan pada penelitian ini dapat dilihat seperti Tabel 5 berikut

Tabel 5 Spesifikasi PC-Client

Spesifikasi	
Processore	4 Core
Hard Disk	1 TB
Memory	10 GB
Operating System	Windows 11 Home
Software	Apache JMeter, Web Browser

3.6 Metode Pengujian

3.6.1 Pengujian High Availability

Pengujian ini dilakukan untuk melihat bagaimana kemampuan dari *web server* dalam menangani permintaan dari *client* saat seluruh *node* yang berada pada *server* fisik *running* dan saat hanya satu *node* yang *running* pada *server* fisik. Pertama, dua puluh empat *client* mengakses *website* yang berada pada empat *node* dimana *node* tersebut berada pada masing-masing *server* fisik lalu dihitung berapa waktu yang dibutuhkan untuk mengakses layanan *web*. Setelah itu, dua *node* dimatikan dan dilihat apakah *client* masih bisa melakukan akses *website* tersebut lalu dihitung berapa waktu yang dibutuhkan untuk mengakses layanan *web* dan perpindahan layanan terhadap *server* yang mati. Kemudian salah satu *server* fisik dimatikan dan dilihat apakah *client* masih bisa mengakses *website* tersebut lalu dihitung berapa waktu yang dibutuhkan untuk mengakses layanan *web* dan perpindahan layanan terhadap *server* yang mati

3.6.2 Pengujian Performance

i) CPU

Pada penggunaan prosesor yang dilihat adalah persentase penggunaan prosesor saat *service* dijalankan dan diakses oleh *client*. Pertama *web server* diakses oleh dua puluh *client* dan dilihat persentase penggunaan prosesor. Kemudian, jumlah *client* ditambahkan sepuluh *client* menjadi tiga puluh *client* dengan tetap melihat persentase penggunaan prosesor. Setelah itu, tiga puluh *client* yang terhubung ditambahkan dua puluh *client* lagi menjadi lima puluh *client* dan tetap melihat persentase penggunaan prosesor. Pengujian ini akan dilakukan dua kali dalam sehari dan dilaksanakan selama satu minggu. Tool yang akan digunakan adalah *jmeter*

ii) RAM

Pada penggunaan RAM yang dilihat adalah penggunaan RAM saat *web server* dinyalakan. Pada hari pertama semua *server* akan dinyalakan sepanjang hari dan akan dilihat penggunaan memorinya. Pada hari ke dua dua *web server* dimatikan sementara yang lainnya tetap berjalan sepanjang hari dan akan dilihat penggunaan memorinya. Pada hari ketiga satu *server* fisik akan dimatikan dan yang lainnya akan tetap berjalan sepanjang hari. Semakin lama layanan *web* dinyalakan maka semakin naik penggunaan RAM nya. Pengujian ini akan dilakukan selama tiga hari. Untuk melihat penggunaan RAM gunakan perintah *htop* pada *linux*.

4. HASIL DAN PEMBAHASAN

4.1 Pengujian High Availability

i) Semua server dijalankan

Pada pengujian *high availability* yang pertama kita akan melakukan uji coba ketika seluruh *web server* dijalankan. *Web server* masih bisa diakses dengan lancar karena seluruh *server* masih berjalan.

Tabel 6 Data pengujian client asli ketika berjalan normal

PC	Server yang diakses	Waktu
1	WEB01	0.3s
2	WEB01	0.3s
3	WEB01	0.3s

PC	Server yang diakses	Waktu
4	WEB01	0.3s
5	WEB01	0.3s
6	WEB01	0.3s
7	WEB02	0.3s
8	WEB02	0.3s
9	WEB02	0.3s
10	WEB02	0.3s
11	WEB02	0.3s
12	WEB02	0.3s
13	WEB03	0.3s
14	WEB03	0.3s
15	WEB03	0.3s
16	WEB03	0.3s
17	WEB03	0.3s
18	WEB03	0.3s
19	WEB04	0.3s
20	WEB04	0.3s
21	WEB04	0.3s
22	WEB04	0.3s
23	WEB04	0.3s
24	WEB04	0.3s

Berdasarkan data pada Tabel 6 layanan *web* dapat diakses oleh *client* asli dengan rata – rata kecepatan yang di butuh kan oleh *client* untuk mengakses layanan *web* tersebut adalah 0.3s

ii) Dua *web server* dimatikan

Status dari *web server* WEB03 dan WEB04 adalah *shut down* atau dimatikan. *Web server* masih bisa di akses dengan beberapa *delay web browser* saat dimuat ulang karena *server load balancer* mengalihkan permintaan ke *server* yang tersedia.

Tabel 7 Data pengujian client asli ketika 2 webserver dimatikan

PC	Server yang diakses	Waktu	Delay perpindahan
1	WEB01	0.3s	-
2	WEB01	0.3s	-
3	WEB01	0.3s	-
4	WEB01	0.3s	-
5	WEB01	0.3s	-
6	WEB01	0.3s	-
7	WEB02	0.3s	2s
8	WEB02	0.3s	2s
9	WEB02	0.3s	2s
10	WEB02	0.3s	2s
11	WEB02	0.3s	2s
12	WEB02	0.3s	2s
13	WEB03	0.3s	1s
14	WEB03	0.3s	1s
15	WEB03	0.3s	1s
16	WEB03	0.3s	1s
17	WEB03	0.3s	1s
18	WEB03	0.3s	1s
19	WEB04	0.3s	-
20	WEB04	0.3s	-
21	WEB04	0.3s	-

PC	Server yang diakses	Waktu	Delay perpindahan
22	WEB04	0.3s	-
23	WEB04	0.3s	-
24	WEB04	0.3s	-

Berdasarkan data pada Tabel 7 layanan *web* dapat diakses oleh *client* asli dengan rata – rata kecepatan yang di butuh kan oleh *client* untuk mengakses layanan *web* tersebut adalah 0.3s . Namun *client* yang mengakses layanan *web* yang berada pada WEB02 dan WEB03 akan mengalami *delay* saat layanan akan dipindahkan ke server yang tersedia. Untuk *client* yang megakses layanan pada WEB02 akan mengalami rata – rata *delay* selama 2 detik dan *client* yang megakses layanan WEB03 akan mengalami rata – rata *delay* selama 1 detik. Hal tersebut terjadi karena untuk perpindahan , *load balancer* akan memeriksa layanan *web* mana yang masi aktif dan dilakukan secara beurutan.

iii) SRV2 dimatikan

Status SRV2 adalah *down* atau dimatikan. *Web server* masih bisa di akses dengan beberapa delay *web browser* saat dimuat ulang karena *server load balancer* mengalihkan permintaan ke *server* yang tersedia

Tabel 8 Data pengujian client asli ketika SRV2 dimatikan

PC	Server yang diakses	Waktu	Delay perpindahan
1	WEB01	0.3s	-
2	WEB01	0.3s	-
3	WEB01	0.3s	-
4	WEB01	0.3s	-
5	WEB01	0.3s	-
6	WEB01	0.3s	-
7	WEB02	0.3s	-
8	WEB02	0.3s	-
9	WEB02	0.3s	-
10	WEB02	0.3s	-
11	WEB02	0.3s	-
12	WEB02	0.3s	-
13	WEB03	0.3s	1,09s
14	WEB03	0.3s	1,09s
15	WEB03	0.3s	1,09s
16	WEB03	0.3s	1,09s
17	WEB03	0.3s	1,09s
18	WEB03	0.3s	1,09s
19	WEB04	0.3s	1,06s
20	WEB04	0.3s	1,06s
21	WEB04	0.3s	1,06s
22	WEB04	0.3s	1,06s
23	WEB04	0.3s	1,06s
24	WEB04	0.3s	1,06s

Berdasarkan data pada Tabel 8 layanan *web* dapat diakses oleh *client* asli dengan rata – rata kecepatan yang di butuh kan oleh *client* untuk mengakses layanan *web* tersebut adalah 0.3s . Namun *client* yang mengakses layanan *web* yang berada pada WEB03 dan WEB04 akan mengalami *delay* saat layanan akan dipindahkan ke *server* yang tersedia . Untuk *client* yang megakses layanan pada WEB03 akan mengalami rata – rata *delay* selama 1.09 detik dan *client* yang megakses layanan WEB03 akan mengalami rata – rata *delay* selama 1,06 detik. Hal

tersebut terjadi karena untuk perpindahan , *load balancer* akan memeriksa layanan *web* mana yang masi aktif dan dilakukan secara beurutan

4.2 Pengujian CPU

Berdasarkan pengujian terhadap *web server* pada masing-masing *server* menggunakan Jmeter, pengujian bertujuan untuk mengukur *high availability* yang dilakukan dengan dua kali tahapan pengujian yaitu seratus *client* di pagi hari dan sore hari dengan masing-masing pengujian dilakukan tujuh kali percobaan dalam mengambil data.

Tabel 9 Hasil pengujian CPU WEB01

WEB01 Hari\Host	Pagi			Sore		
	20	30	50	20	30	50
1	1,30%	1,30%	2,00%	0,70%	0,71%	0,70%
2	0,70%	0,70%	1,30%	0,70%	0,70%	1,30%
3	0,70%	0,70%	0,70%	0,70%	0,70%	0,71%
4	0,70%	0,70%	0,70%	0,70%	1,30%	1,30%
5	0,70%	0,70%	1,30%	0,70%	0,70%	0,70%
6	0,70%	0,70%	0,70%	0,70%	1,30%	1,30%
7	0,70%	0,70%	1,30%	0,70%	1,30%	1,30%

Berdasarkan Tabel 9 diatas penggunaan CPU pada WEB01 mengalami peningkatan ketika jumlah permintaan diperbanyak , hal tersebut terjadi karena bertambahnya beban kerja pada *web server* untuk mengelola permintaan yang terus bertambah. Rata – rata yang didapatkan untuk pengujian ini adalah sebesar 0,79% untuk permintaan sebanyak tiga puluh *client* , 0,79% untuk permintaan sebanyak tiga puluh *client* , 1,14% untuk permintaan sebanyak lima puluh *client*

Tabel 10 Hasil pengujian CPU WEB02

WEB01 Hari\Host	Pagi			Sore		
	20	30	50	20	30	50
1	0,70%	1,30%	1,30%	0,70%	0,70%	0,70%
2	0,70%	0,70%	1,30%	1,30%	1,30%	1,30%
3	0,70%	1,30%	0,70%	0,70%	1,30%	1,30%
4	0,70%	0,70%	1,30%	1,30%	1,30%	1,30%
5	0,70%	0,70%	1,30%	1,30%	1,30%	1,30%
6	0,70%	0,70%	1,30%	0,70%	1,30%	1,30%
7	0,70%	0,70%	1,30%	0,70%	0,70%	1,30%

Berdasarkan Tabel 10 diatas penggunaan CPU pada WEB02 mengalami peningkatan ketika jumlah permintaan diperbanyak , hal tersebut terjadi karena bertambahnya beban kerja pada *web server* untuk mengelola permintaan yang terus bertambah. Rata – rata yang didapatkan untuk pengujian ini adalah sebesar 0,83% untuk permintaan sebanyak tiga puluh *client* , 0,87% untuk permintaan sebanyak tiga puluh *client* , 1,21% untuk permintaan sebanyak lima puluh *client*.

4.3 Pengujian RAM

- i) Pada saat semua server dijalankan

Tabel 11 Hasil pengujian RAM hari pertama

Perangkat	Penggunaan RAM
SRV1	62%
SRV2	48%
WEB01	9%
WEB02	9%
WEB03	9%

Perangkat	Penggunaan RAM
WEB04	9%
GATEWAY	8%

Pada Tabel 11 merupakan data penggunaan memori yang normal jika semua *server* fisik dan *node* dijalankan

- ii) Pada saat salah 2 *web server* dimatikan

Tabel 12 Hasil pengujian RAM hari kedua

Perangkat	Penggunaan RAM
SRV1	62%
SRV2	31%
WEB01	10%
WEB02	-
WEB03	-
WEB04	9%
GATEWAY	8%

Pada Tabel 12 merupakan data penggunaan memori ketika ada dua *node* yang tidak berjalan atau mati , berdasarkan data tersebut dapat dilihat bahwa penggunaan memori pada SRV2 menurun sebanyak 17% dari penggunaan normal. Hal tersebut terjadi karena salah satu *node* yang terletak pada SRV2 tidak berjalan atau mati.

- iii) Pada saat SRV2 dimatikan

Tabel 13 Hasil pengujian RAM hari ketiga

Perangkat	Penggunaan RAM
SRV1	68%
SRV2	-
WEB01	10%
WEB02	9%
WEB03	-
WEB04	-
GATEWAY	10%

Pada Tabel 13 merupakan data penggunaan memori ketika SRV2 mati , berdasarkan data tersebut dapat dilihat bahwa penggunaan memori pada SRV1 meningkat sebanyak 6% dari penggunaan normal. Hal tersebut terjadi karenaSRV2 tidak berjalan atau mati , sehingga SRV1 harus menjalankan *service* lebih untuk menjaga ketersediaan *server*. Dapat diperhatikan pula pada *node* WEB03 dan WEB04 akan ikut mati karena kedua *node* tersebut terletak pada SRV2.

5. KESIMPULAN

Dari hasil Pengujian dapat diambil kesimpulan sebagai berikut:

- i) Layanan *web server* menggunakan *cluster cloud computing* dapat dibangun menggunakan Microstack.
- ii) *High availability* dapat tercapai , layanan *web server* tetap bisa di akses oleh *client* walaupun ada beberapa *node* dan *server* fisik yang mati.
- iii) Persentase penggunaan CPU *server* fisik apabila salah satu *server* fisik didalam *cluster* dimatikan meningkat dari 62% sampai 68% .
- iv) Persentase penggunaan CPU *web server* yang berada pada SRV1 saat diakses oleh duapuluh sampai dengan limapuluh *client* meningkat dari 0,7% sampai 1,3%.

- v) Persentase penggunaan CPU *web server* yang berada pada SRV2 saat diakses oleh duapuluh sampai dengan limapuluh *client* meningkat dari 5,10% sampai 8,40%.
- vi) Persentase penggunaan memori SRV2 saat dua *node* dimatikan menurun dari 48% sampai 31%. Sedangkan persentase *node WEB01* meningkat dari 0,9% sampai 10%.
- vii) Persentase penggunaan memori SRV1 saat SRV2 dimatikan meningkat dari 62% sampai 68%, dan *node WEB03* dan *WEB04* otomatis mati karena SRV2 tidak berjalan.

DAFTAR PUSTAKA

- [1] S. Budi, "Implementasi Cluster Pada Web Server Berbasis Cloud Computing," Universitas Islam Negeri Sunan Kalijaga Yogyakarta, 2013.
- [2] T. Ernawati, "Analisis dan pembangunan infrastruktur cloud computing ," Politeknik TEDC Bandung ,2018.
- [3] P.G.S.C. Nugraha, "Implementasi dan analisis kinerja private cloud computing di lingkungan GDLN Universitas Udayana ," Universitas Udayana ,2017.
- [4] A. Arif, "Implementasi Web Server Pada Cloud Computing Dengan OpenVZ ," Politeknik Caltex Riau ,2015.
- [5] M. V. Sugianto, "Keuntungan Teknologi Virtualisasi & Cloud Computing," 2011.
- [6] Baktikominfo, "BERKENALAN DENGAN NODE: PENGERTIAN, FUNGSI, VARIASI, DAN JENISNYA," *BAKTIKOMINFO*, 2021.
- [7] S. van Vugt, *Pro Linux High Availability Clustering*. Berkeley, CA: Apress, 2014. doi: 10.1007/978-1-4842-0079-7.
- [8] OpenStack, "Pengantar OpenStack"2023. <https://docs.openstack.org/id/security-guide/introduction/introduction-to-openstack.html> (accessed May 22, 2023).
- [9] K. Schmidt, *High Availability and Disaster Recovery*, 1st ed. Frankfurt, Germany: Springer Berlin Heidelberg, 2006. doi: 10.1007/3-540-34582-5.
- [10] A. Suryanto, "Pembangunan Aplikasi Piket Pada Sekolah Menengah Atas Satu Kudus berbasis Web," *Journal Speed – Sentra Penelitian Engineering dan Edukasi*, vol. 4, no. 4, pp. 29–33, 2012.
- [11] K. Hartanto, "Implementasi Virtual Private Server menggunakan Xen Hypervisor," Universitas Diponegoro, 2012.
- [12] Apache Software Foundation, "Apache JMeter - Apache JMeter™," *Apache JMeter™*, 2021. <https://jmeter.apache.org/>
- [13] Canonical Ltd, "Multi-node OpenStack for workstations and edge / IoT."2023. <https://microstack.run> (accessed May 22,2023).
- [14] Canonical Ltd, "Ubuntu."2023. <https://ubuntu.com> (accessed May 22,2023).