

PENGUJIAN EFEKTIFITAS SISTEM *CLUSTER* DENGAN PENERAPAN *HIGH AVAILABILITY* PADA SERVER VIRTUAL (Studi Kasus di BPS Provinsi Sumatera Utara)

*Testing the Effectiveness of Cluster Systems by Implementing High Availability on
Virtual Servers
(Case Study at BPS-Statistics of Sumatera Utara Province)*

Didik Darmadi*, Rafles Susandi**

*BPS Provinsi Sumatera Utara, *E-mail*: didik@bps.go.id

** Dinas Komunikasi dan Informatika Kota Payakumbuh

ABSTRAK

Penelitian ini bertujuan untuk mengetahui apakah pemanfaatan server akan lebih efektif jika dilakukan virtualisasi dalam sistem *cluster* daripada masing-masing server berjalan secara sendiri-sendiri. *Network Development Lifecycle (NDLC)* digunakan sebagai metode dalam penelitian ini. Dari hasil analisis kondisi berjalan dan sistem yang diinginkan terdapat kesenjangan yang kemudian dijadikan dasar dalam mendesain ulang sistem jaringan menuju kondisi yang sesuai keinginan. Hasil desain disimulasikan dengan menerapkan sistem *cluster* pada server-server fisik yang tersedia dengan menggunakan perangkat lunak Proxmox VE 6.2 kemudian dilakukan uji coba untuk menghitung efektifitasnya. Hasil dari kajian ini membuktikan bahwa sistem *cluster* yang dibangun dengan Proxmox VE 6.2 dapat menjalankan layanan *high availability* pada lingkungan server virtual. Sistem *cluster* dengan server virtual juga meningkatkan efisiensi sumber daya server yang tersedia dimana banyak server virtual dapat dibuat pada satu mesin server fisik. Penerapan *high availability* mampu meningkatkan efektifitas server dalam menjalankan layanan. *Downtime* saat terjadinya gangguan server relatif sangat singkat, namun tergantung pada jenis tipe virtualisasi dan sistem operasi server virtual yang dipilih.

Kata kunci: clustering, container, Proxmox VE, server, VE, virtualisasi,

ABSTRACT

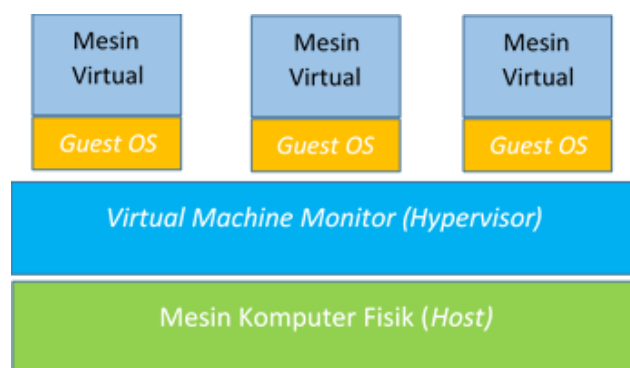
This study aims to investigate whether server utilization will be more effective if virtualization is carried out in an effective cluster system rather than each server running independently. Network Development Lifecycle (NDLC) is used as a method in this study. From analysis results from the current conditions and the desired system, there is a gap obtained which is then used as the basis for redesigning the network system to the desired condition. The design results are simulated by implementing a cluster system on available physical servers using Proxmox VE 6.5 software and then a test is carried out to calculate its effectiveness. The results of this study prove that the Cluster System built with Proxmox VE 6.2 can run high availability services in a virtual server environment. Cluster systems with virtual servers also increase the efficiency of available server resources where many virtual servers can be created on one physical server machine. The implementation of high availability is able to increase the effectiveness of the server in running the service. Downtime, when a server outage occurs, is relatively short, but depends on the type of virtualization and the virtual server operating system selected.

Keywords: clustering, container, Proxmox VE, server, virtualization

I. PENDAHULUAN

Layanan berbasis komputer sudah menjadi kebutuhan dasar pada pelaksanaan pekerjaan di Badan Pusat Statistik (BPS) Provinsi Sumatera Utara. Tanpa layanan ini, beberapa pekerjaan khususnya yang terkait dengan pengolahan data dan penyediaan informasi dapat terganggu bahkan terhenti. Server menjadi komponen penting dalam pelaksanaan layanan berbasis komputer karena data dan layanan disimpan dalam mesin server. Kebutuhan akan server meningkat seiring dengan peningkatan layanan yang diperlukan dalam proses bisnis di BPS Provinsi Sumatera Utara. Server harus selalu tersedia dalam memberikan layanan sehingga pekerjaan dapat berjalan dengan lancar.

Keterbatasan jumlah mesin server yang tersedia membuat tidak semua layanan dapat dijalankan pada mesin server. pada sisi lain, sumber daya yang tersedia pada mesin server tidak terpakai secara optimal dan hanya sedikit sumber daya yang digunakan. Virtualisasi memungkinkan sebuah mesin server dapat menjalankan beberapa sistem operasi dan layanan pada saat bersamaan. Virtualisasi memberikan efisiensi dalam penggunaan sumber daya mesin server dan penggunaan infrastruktur yang optimal (Ali et al., 2015; Scroggins, 2017; Suryono & Afif, 2012). Mesin server fisik yang cukup mahal memiliki sumber daya yang sangat besar dan sedikit terpakai jika hanya menjalankan satu sistem operasi dan layanan. Sisa sumber daya ini dapat dimanfaatkan untuk membuat mesin-mesin server baru secara virtual (Agustian et al., 2018). Berbagai layanan yang seharusnya membutuhkan banyak mesin server fisik dapat dijalankan pada satu atau dua mesin server fisik.



Gambar 1. Konsep Virtualisasi Komputer

Virtualisasi komputer menggambarkan bagaimana mesin komputer fisik direpresentasikan dalam objek virtual. Dengan direpresentasi secara virtual maka akan didapat sumberdaya yang lebih banyak untuk dapat digunakan. Sebuah software yang disebut *sebagai Virtual Machine Monitor* atau *hypervisor* bertugas mengelola sumberdaya mesin fisik atau *host* untuk menjalankan mesin-mesin virtual yang ada didalamnya. Setiap mesin virtual

(*guest host*) memiliki sumberdaya dan sistem operasi (*guest operating system/OS*) masing-masing. Gambar 1 memperlihatkan konsep virtualisasi komputer (Portnoy, 2016).

Sistem *cluster* menggabungkan beberapa mesin server fisik dalam satu antar muka pengelolaan. Setiap mesin server yang masuk dalam sistem cluster dihubungkan melalui jaringan LAN (Local Area Network) dengan kecepatan tinggi sehingga menjadi seolah-olah satu mesin server tunggal (Adi et al., 2016). Sistem *cluster* mampu memberikan jaminan tinggi ketersediaan layanan (*high availability*). Jika ada satu mesin server fisik dalam sistem *cluster* (*node*) mengalami kegagalan (*down*) maka layanan yang ada pada *node* tersebut dapat langsung dialihkan pada *node* yang lain.

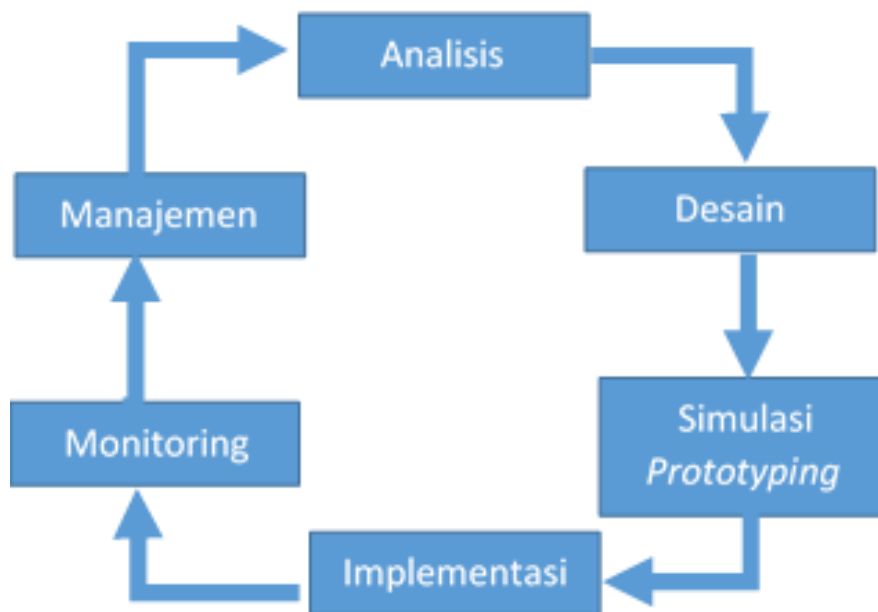
Fitur *High Availability (HA)* secara umum sudah disediakan oleh berbagai aplikasi sistem *cluster*. HA memanfaatkan media penyimpanan yang dipakai bersama (*shared disk*) sehingga saat satu *node* gagal berjalan (*failure*) maka *node* yang lain akan mengambil alih dengan menjalankan virtual server yang semula berjalan pada *node* yang mengalami kegagalan (Ljubojevi & Baji, 2019). Pada lingkungan server virtual yang berjalan di sistem *cluster*, pengelolaan HA relatif mudah melalui halaman manajemen sistem cluster. Tanpa virtualisasi dan sistem *cluster*, penerapan HA membutuhkan penarikan data yang tersimpan pada berbagai sumber media penyimpanan (Harper et al., 2009).

Kumar dan Charu menyatakan bahwa virtualisasi mampu meningkatkan utilitas dari komputer dengan meminimalkan biaya, konsumsi daya serta penggunaan ruang (Kumar & Charu, 2015). Virtualisasi mempermudah manajemen dan pemeliharaan mesin server karena hanya fokus pada satu *hardware* saja yang menjalankan berbagai mesin virtual (Suryono & Afif, 2012). Selain efisiensi dalam penggunaan mesin server, penerapan efektifitas pemanfaatan server dengan meminimalkan *downtime* layanan sudah menjadi praktik yang umum dalam menjalankan proses bisnis organisasi (Al-Aomar et al., 2016). Penelitian ini dilakukan untuk melihat efektifitas pemanfaatan mesin server dengan mengukur *downtime* pada server virtual yang berjalan dalam sistem *cluster*.

II. METODE PENELITIAN

Metode yang digunakan dalam penelitian ini mengacu pada metode pengembangan dalam bidang jaringan komputer yakni *Network Development Lyfe Cycle (NDLC)*. Sebagaimana dalam pengembangan sistem, NDLC memiliki tahapan analisis, desain, simulasi, implementasi, monitoring dan manajemen yang dilaksanakan secara berkesinambungan sebagaimana pada Gambar 2 (Goldman & Rawles, 2004; Rianafirin & Kurniawan, 2018).

Output setiap tahapan dalam NDLC akan menjadi input pada tahapan setelahnya. Pengembangan NDLC dimulai dengan menganalisis kondisi yang ada saat ini dan sistem yang diinginkan. Dalam tahapan analisis juga dilakukan identifikasi kebutuhan perangkat keras dan perangkat lunak. Tahap desain meliputi desain jaringan dan desain sistem sesuai dengan kebutuhan yang diinginkan. Tahap simulasi atau *prototyping* adalah melakukan percobaan sesuai dengan desain yang telah dibuat. Setelah ujicoba bisa berjalan lancar maka dapat dilakukan implementasi pada lingkungan yang sebenarnya. Tahapan monitoring dan manajemen memastikan sistem dapat berjalan dengan baik. Semua tahapan ini berkesinambungan sehingga dapat dilakukan perbaikan dan peningkatan dari waktu ke waktu.



Gambar 2. *Network Development Lyfe Cycle*

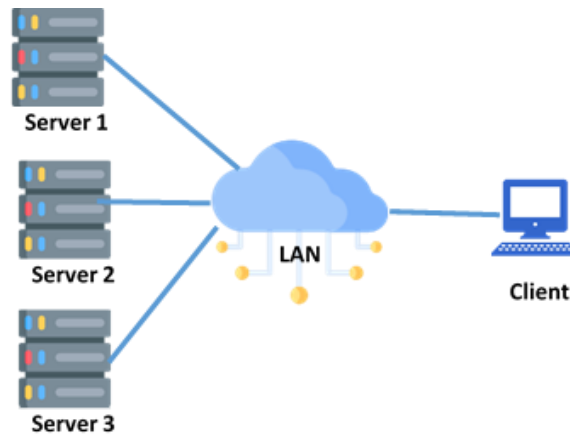
A. Kondisi Sistem yang Berjalan dan Sistem yang Diinginkan

BPS Provinsi Sumatera Utara memiliki tiga mesin server fisik Fujitsu Primergy yang berjalan sendiri-sendiri. Mesin server tersebut digunakan sebagai server pengolahan, server *file sharing* dan web server. Secara rinci mesin-mesin server yang dimiliki BPS Provinsi Sumatera Utara dapat dilihat pada Tabel 1.

Tabel 1. Mesin server fisik yang dimiliki BPS Provinsi Sumatera Utara

Merk/tipe	Spesifikasi	Penggunaan
Fujitsu Primergy RX300-S5	CPU(s)16 x Intel(R) Xeon(R) CPU E5520 @ 2.27GHz (2 Sockets), RAM 12 GB	Server Pengolahan dengan sistem operasi Windows Server 2008R2
Fujitsu Primergy RX300-S5	CPU(s)16 x Intel(R) Xeon(R) CPU E5520 @ 2.27GHz (2 Sockets), RAM 12 GB	Server <i>file sharing</i> dengan sistem operasi Windows Server 2008R2
Fujitsu Primergy RX300-S5	CPU(s)16 x Intel(R) Xeon(R) CPU E5520 @ 2.27GHz (2 Sockets), RAM 28 GB	Web Server dengan sistem operasi Linux Ubuntu 20.04

Meskipun ketiga server tersebut terhubung dengan jaringan LAN, namun dalam manajemennya masih dilakukan secara terpisah untuk masing-masing server. Jika salah satu server mengalami *down* maka layanan yang memanfaatkan server tersebut akan terhenti. Topologi jaringan pada sistem berjalan dapat dilihat pada Gambar 3.



Gambar 3. Topologi sistem sebelum penelitian

Berdasarkan kondisi yang berjalan sebelum penelitian dimulai, pengembangan sistem dilakukan dengan memanfaatkan tiga server yang ada secara terpisah. Pada penelitian ini ketiga mesin server tersebut dijadikan satu kesatuan manajemen sehingga memudahkan dalam pengelolaan dan mendukung ketersediaan layanan. Dengan menjadikan setiap mesin server sebagai node dalam satu manajemen memungkinkan pembuatan server virtual dengan jumlah yang lebih banyak dari mesin server fisik yang tersedia. Server-server yang dibutuhkan dalam menjalankan layanan di BPS Provinsi Sumatera Utara antara lain Server Pengolahan, Server Robot Kapow, *Web Server Production* dan *Web Server Development*.

B. Kebutuhan Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan sistem *cluster* ini adalah:

- Tiga unit mesin server yang akan digunakan sebagai node dalam sistem *cluster*. Tiga unit merupakan jumlah minimal *node* agar fitur HA pada proxmox dapat berjalan. Dengan adanya tiga buah *node* maka sistem *quorum* dapat berjalan jika ada salah satu *node* mengalami kerusakan atau gangguan yang *down* (Gmbh, 2020).
- Switch sebagai perangkat yang menghubungkan semua node serta menghubungkan sistem cluster dengan jaringan LAN.
- *Network Attached Storage* (NAS) Server sebagai media penyimpanan file yang dapat digunakan bersama-sama oleh semua *node* dalam sistem *cluster*. Dengan adanya penyimpanan yang digunakan bersama (*sharing*) memungkinkan sistem *cluster* menerapkan HA (Gmbh, 2020). *Protocol* yang digunakan dalam *sharing* sumber daya NAS Server adalah *Network File System* (NFS). NAS Server juga digunakan sebagai pengganti Server *File Sharing* yang sebelumnya menggunakan satu mesin server tersendiri.
- Komputer PC digunakan sebagai *client* untuk mengelola sistem *cluster* dan melakukan uji coba. Manajemen sistem cluster menggunakan antar muka web sehingga hanya membutuhkan browser untuk mengakses sistem *cluster* dari *client*.

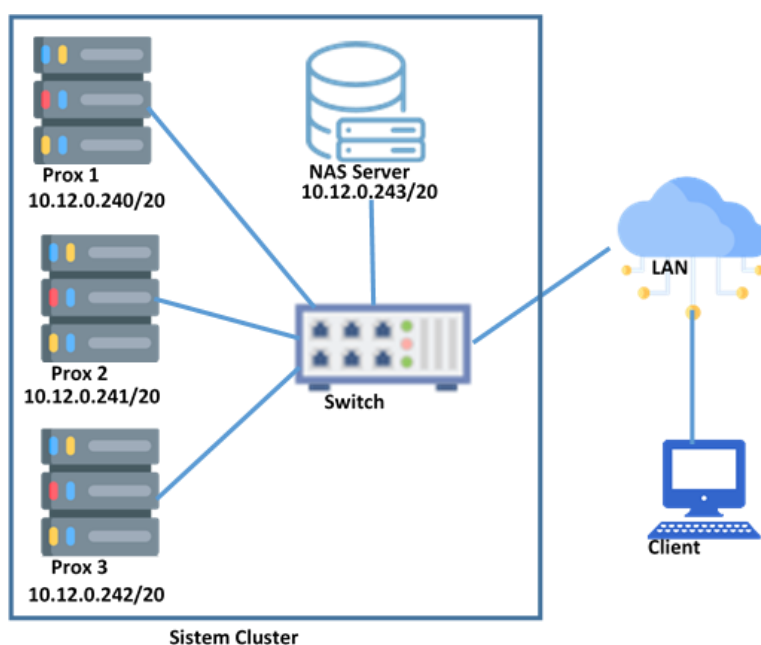
C. Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan dalam pengembangan sistem *cluster* ini adalah:

- Proxmox Virtual Environment atau Proxmox VE yang saat penelitian ini dilaksanakan sudah versi 6.2. Proxmox VE 6.2 digunakan untuk mengimplementasikan sistem *cluster* dan mengelola server virtual yang berjalan didalamnya. Proxmox VE 6.2 merupakan perangkat lunak *open source* berbasis linux debian yang memiliki kemampuan untuk membuat sistem *cluster* dengan fitur virtualisasi virtual machine (qemu) dan container (lxc), HA, *live migration*, *backup* serta fitur-fitur lainnya. Dengan manajemen yang mudah berbasis web dan berbagai fitur yang dimiliki, Proxmox menjadi solusi siap pakai dalam pembuatan sistem cluster (Kovari & Dukan, 2012).
- Ubuntu Server 20.04 yang dijalankan pada server virtual sebagai sistem operasi untuk layanan web server baik production maupun development.
- Microsoft Windows Server 2016 digunakan sebagai sistem operasi pada server virtual untuk layanan server pengolahan dan server robot Kapow.

D. Desain Sistem Cluster dan Jaringan

Tiga unit mesin server fisik yang dimiliki BPS Provinsi Sumatera Utara saling terhubung melalui switch berkecepatan tinggi membentuk sistem *cluster*. Sistem *cluster* juga dilengkapi dengan NAS Server sebagai tempat penyimpanan yang dapat digunakan bersama (*share*) oleh ketiga mesin server fisik yang disebut sebagai *node*. Gambar 3 menjelaskan topologi sistem *cluster* dengan tiga buah *node*, masing-masing *node* diberi nama Prox 1, Prox 2 dan Prox 3. Pengaturan alamat IP pada masing-masing *node* dan NAS Server harus disesuaikan sehingga setiap *node* dan NAS Server dapat berkomunikasi melalui jaringan.



Gambar 4. Topologi sistem *cluster* yang diusulkan

Sistem *cluster* terhubung dengan jaringan LAN kantor BPS Provinsi Sumatera Utara. Manajemen dan monitoring sistem *cluster* dilakukan melalui PC *client* yang terhubung jaringan LAN dengan menggunakan web browser.

E. Simulasi dan Uji Coba

Simulasi dilakukan dengan membuat empat buah server virtual yang dijalankan di atas sistem *cluster*. Empat buah server virtual yang dibuat sesuai dengan kebutuhan server BPS Provinsi Sumatera Utara yakni: Server Pengolahan dan Server Robot-Kapow yang menggunakan sistem operasi Windows Server 2016 serta *Web Server Production* dan *Web Server Development* yang menggunakan sistem operasi Ubuntu Server 20.04. *Linux Container (lxc)* pada proxmox digunakan untuk membuat server virtual yang menggunakan sistem operasi Ubuntu Server 20.04. Pada virtual server yang menggunakan sistem operasi Windows

Server 2016, *Linux Container (lxc)* tidak dapat digunakan sehingga harus membuat virtual machine yang berfungsi selayaknya mesin server fisik. Server virtual yang dibuat secara rinci dapat dilihat pada Table 2.

Container memanfaatkan *kernel* dan *library* pada sistem operasi *host* server untuk menjalankan sebuah lingkungan server yang independen. Hasil dari virtualisasi dengan *container* membutuhkan sumber daya yang lebih kecil sehingga dapat dibuat lebih banyak server virtual pada *host* server dibandingkan dengan membuat virtual machine. *Booting* atau *restarting container* lebih cepat karena menggunakan *kernel* dari *host* server yang sudah berjalan lebih dahulu (Bernstein, 2014).

Tabel 2. Spesifikasi detail server virtual yang dikembangkan pada penelitian

Komponen	Server Virtual			
	CT 101	CT 102	VM 103	VM 104
Nama	Production	Development	Pengolahan	Robot-Kapow
Tipe	<i>Container (lxc)</i>	<i>Container (lxc)</i>	<i>Virtual Machine (qemu)</i>	<i>Virtual Machine (qemu)</i>
<i>node</i>	Prox 3	Prox 3	Prox 1	Prox 2
Sistem Operasi	Ubuntu Server 20.04	Ubuntu Server 20.04	Windows Server 2016	Windows Server 2016
<i>Processor</i>	2 Core	2 Core	8 Core	8 Core
<i>Memory</i>	2 GB	2 GB	4 GB	4 GB
<i>Storage</i>	100 GB	100 GB	100 GB	100 GB
<i>IP Address</i>	10.12.0.101/20	10.12.0.102/20	10.12.0.103/20	10.12.0.104/20

Fitur HA pada sistem *cluster* diaktifkan dengan membuat HA Group bernama Server BPS dan memasukkan *node* Prox 1, Prox 2 dan Prox 3 sebagai anggota dari grup tersebut. Setiap server virtual yang sudah dibuat dimasukkan sebagai sumber daya HA dengan *max restart* 1 dan *max relocate* 2. *Max restart* 1 menginstruksikan kepada HA manager untuk mencoba sekali restart saat terjadi kegagalan sebelum dipindah ke *node* lain. *Max relocate* 2 menginstruksikan kepada HA manager untuk mencoba memindahkan layanan ke *node* lain sebanyak dua kali saat terjadi kegagalan.

Setelah sistem *cluster* disimulasikan dan berjalan secara normal dilakukan ujicoba untuk mengetahui apakah fitur HA bekerja pada sistem *cluster* tersebut. Uji coba dilakukan dengan skenario *live migration* dan *node failure*. Pada skenario *live migration*, server virtual yang

sedang berjalan pada masing-masing *node* dipindahkan ke *node* lainnya. Skenario *node failure* dilakukan dengan memutus salah satu *node* dari sistem *cluster* untuk mensimulasikan terjadinya *failure* pada *node* tersebut. Kegagalan atau *failure* pada sistem *cluster* dapat disebabkan antara lain oleh kegagalan *hardware*, jaringan, power/kelistrikan, *reboot* dan *software* (Birke et al., 2014). Skenario *node failure* dapat merepresentasikan apa yang akan terjadi jika salah satu *node* atau lebih dalam sistem *cluster* mengalami kegagalan.

Setiap perpindahan server virtual dari satu *node* ke *node* lainnya akan menyebabkan terhentinya layanan oleh server virtual tersebut. Lama waktu terhentinya layanan (*downtime*) pada masing-masing server virtual saat perpindahan *node* diukur dengan menggunakan aplikasi Fping. Aplikasi Fping dijalankan untuk melakukan ping ke virtual server dengan interval 100 milidetik. Ukuran paket yang dikirimkan pada perintah ping adalah 1 *byte* yang merupakan ukuran terkecil yang dapat digunakan. Dengan menggunakan ukuran terkecil diharapkan tidak ada kegagalan pengiriman paket karena server virtual tidak bisa menjawab permintaan yang besar (Adi et al., 2016). *Downtime* didapatkan dengan melihat jumlah paket yang gagal kemudian dikali dengan 100 milidetik.

III. HASIL DAN PEMBAHASAN

A. Implementasi Sistem *Cluster*

Implementasi sistem *cluster* dilakukan dengan menginstall Proxmox VE 6.2 pada setiap mesin server fisik. Pengaturan jaringan pada setiap mesin server dilakukan agar ketiga mesin server berada dalam jaringan yang sama sehingga dapat berkomunikasi satu sama lain. Ketiga mesin server fisik yang sudah terinstall Proxmox VE 6.2 dan terhubung dengan switch kemudian digabungkan menjadi satu sistem cluster dimana setiap mesin fisik menjadi *node* dalam sistem cluster. NAS Server dihubungkan melalui switch yang sama kemudian dilakukan pengaturan untuk dapat digunakan sebagai media penyimpanan bersama oleh masing-masing *node*. Manajemen sistem *cluster* dapat dilakukan melalui satu halaman administrasi berbasis web yang dapat diakses oleh client menggunakan *web browser*.

Type ↑	Description	Disk us...	Memory us...	CPU usage
lxc	101 (Production)	7.4 %	79.4 %	0.1% of 2CPUs
lxc	102 (Development)	7.4 %	79.0 %	0.0% of 2CPUs
node	prox1	7.7 %	46.9 %	0.4% of 16CPUs
node	prox2	2.2 %	56.3 %	0.5% of 16CPUs
node	prox3	7.6 %	16.8 %	0.4% of 16CPUs
qemu	103 (Pengolahan)		47.3 %	0.7% of 8CPUs
qemu	104 (Robot-Kapow)		87.9 %	0.7% of 8CPUs

Gambar 5. Sistem cluster dengan tiga buah node dan empat buah server virtual

Pembuatan Server-server virtual dan aktivasi fitur HA dilakukan melalui halaman administrasi sistem *cluster*. Keempat server virtual yang telah dibuat kemudian dimasukkan sebagai sumber daya HA pada sistem *cluster*. File image server virtual disimpan pada NAS Server yang digunakan bersama oleh seluruh *node* dalam sistem *cluster*. Gambar 5 memperlihatkan halaman administrasi sistem *cluster* yang dibangun dengan Proxmox VE 6.2. Sistem *cluster* yang dibangun terdiri dari tiga buah *node*, dua buah server virtual dengan container (*lxc*) dan dua buah server virtual dengan virtual machine (*qemu*).

B. Pengujian

Sebelum dilakukan pengujian, sistem *cluster* dipastikan berjalan dengan normal. Pada skenario pengujian pertama dilakukan proses pemindahan server virtual dari satu *node* ke *node* lainnya kemudian diukur *downtime* server virtual selama proses perpindahan. Satu persatu virtual server dipindahkan ke *node* lain secara langsung dalam kondisi berjalan (*live migration*). Pada virtual server dengan tipe virtual machine (*qemu*), saat proses *live migration* server virtual tidak mengalami gangguan layanan atau *downtime* sebesar 0 detik. Server virtual dengan tipe *container (lxc)* menggunakan kernel dari sistem operasi *host* server sehingga memerlukan proses restart saat dilakukan *live migration*. Proses restart virtual server inilah yang menyebabkan terjadinya *downtime*. Lamanya *downtime* pada masing-masing server virtual dengan tipe *container (lxc)* saat perpindahan dari satu *node* ke *node* lain secara rinci dapat dilihat pada Tabel 3.

Tabel 3. Hasil pengukuran *downtime* saat *live migration*

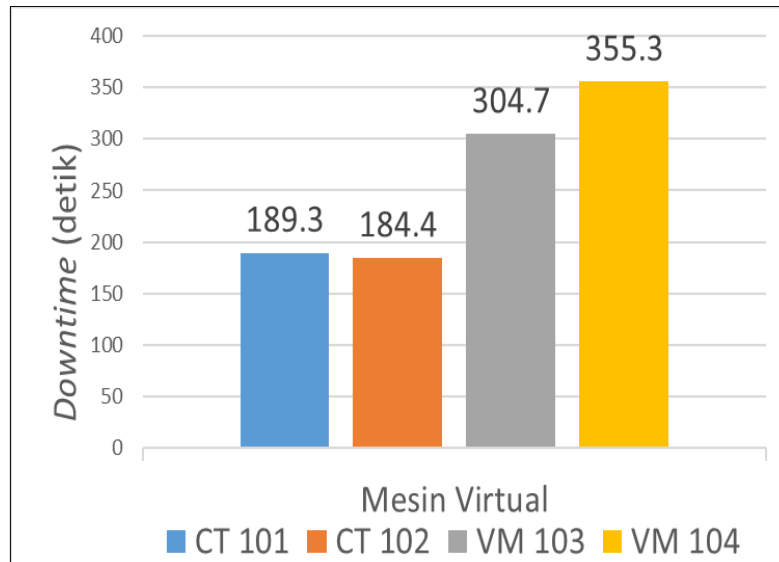
<i>Live Migration</i>	<i>Downtime (detik)</i>		
	CT 101	CT 102	Rerata
Prox 1 – Prox 2	42.7	45.5	44.10
Prox 1 – Prox 3	45.9	51.9	48.90
Prox 2 – Prox 1	30.5	51.5	41.00
Prox 2 – Prox 3	33.4	27.3	30.35
Prox 3 – Prox 1	29.6	42.6	36.10
Prox 3 – Prox 2	23.1	27.8	25.45

Skenario pengujian kedua (*node failure*) dilakukan untuk mensimulasikan kondisi dimana *node* mengalami kegagalan. Pada kondisi sistem *cluster* berjalan dengan normal, salah satu *node* diputus dari sistem *cluster* secara bergantian. Secara rinci hasil pengujian *node failure* dapat dilihat pada Tabel 4.

Tabel 4. Hasil pengujian *node failure*

<i>Node Gagal</i>	Hasil
Prox 1	VM 103 yang semula berjalan pada <i>node</i> Prox 1 berpindah secara otomatis ke <i>node</i> Prox 2. <i>Downtime</i> VM 103 saat migrasi sebesar 304.7 detik atau sekitar 5.06 menit.
Prox 2	VM 104 yang semula berjalan pada <i>node</i> Prox 2 berpindah secara otomatis ke <i>node</i> Prox 1. <i>Downtime</i> VM 103 saat migrasi sebesar 355.3 detik atau sekitar 5.92 menit.
Prox 3	CT 101 yang semula berjalan pada <i>node</i> Prox 3 berpindah secara otomatis ke <i>node</i> Prox 1 dan CT102 berpindah ke <i>node</i> Prox 2. <i>Downtime</i> CT 103 saat migrasi sebesar 189.3 detik atau sekitar 3.16 menit sedangkan <i>downtime</i> CT 102 sebesar 184.4 detik atau 3.07 menit.
Dua buah <i>node</i> gagal bersamaan	Fitur HA pada sistem <i>cluster</i> tidak dapat berjalan karena tidak memenuhi <i>quorum</i> .

Perbandingan *downtime* masing-masing server virtual saat terjadi kegagalan node dapat dilihat pada Gambar 6.



Gambar 6. Downtime server virtual saat terjadi kegagalan node

C. Pembahasan

Hasil implementasi menunjukkan bahwa sistem *cluster* yang mendukung jaminan ketersediaan layanan tinggi (HA) dapat dibangun pada tiga buah mesin server fisik dengan menggunakan *software* Proxmox VE 6.2. Tiga unit mesin server fisik yang dijadikan *node* merupakan jumlah minimal agar sistem *cluster* dapat menjalankan fitur HA. Selain mesin server fisik, dalam pembangunan sistem *cluster* diperlukan media penyimpanan yang dapat digunakan bersama (*sharing*) dalam hal ini menggunakan NAS Server. Ketiga mesin server fisik dan NAS Server dihubungkan dengan switch berkecepatan tinggi.

Jumlah mesin virtual yang dapat dibangun di atas sistem *cluster* tergantung pada sumber daya (CPU, RAM, HDD) yang ada setiap *node*. Server virtual dengan tipe container memerlukan *sumber daya* yang lebih sedikit daripada server virtual dengan tipe virtual machine. Pada *node* dengan spesifikasi yang sama, dapat dibangun server virtual dengan tipe container lebih banyak daripada tipe virtual machine. Container hanya dapat digunakan pada server virtual yang menggunakan sistem operasi berbasis Linux seperti Ubuntu Server 20.04.

Pengujian *live migration* menunjukkan bahwa server virtual dengan tipe virtual machine tidak mengalami *downtime* selama proses migrasi. Proses *live migration* adalah dengan menyalin memory page server virtual pada node asal ke node tujuan. Setelah proses pemindahan selesai maka server virtual pada *node* asal akan dihentikan dan pada *node* tujuan akan dijalankan. Proses ini berjalan simultan sehingga server virtual dapat berjalan terus tanpa mengalami *downtime*. Pada server dengan tipe *container*, virtual server memanfaatkan *kernel* sistem operasi dari *node* sehingga perlu dilakukan *restart* saat terjadi perpindahan *node*. Proses *restart* inilah yang menyebabkan terjadinya *downtime* pada server virtual bertipe container.

Kegagalan atau *failure* pada sistem *cluster* dapat terjadi kapan saja khususnya pada mesin server fisik masing-masing *node*. Pengujian *node failure* menunjukkan bahwa fitur HA dapat berjalan pada sistem *cluster* yang dibangun dari tiga mesin server fisik dengan Proxmox VE 6.2. Setiap terjadi kegagalan pada satu *node* maka server virtual yang berjalan pada *node* tersebut akan dipindahkan ke *node* lainnya. Kegagalan pada dua *node* menyebabkan *quorum* tidak terpenuhi karena hanya ada satu *node* yang aktif. *Quorum* akan terpenuhi jika terdapat dua atau lebih *node* yang aktif. Fitur HA tidak bisa berjalan jika *quorum* tidak terpenuhi. Hasil ini sejalan dengan penelitian (Adi et al., 2016) yang menyatakan bahwa sistem *cluster* tetap berjalan jika jumlah server yang berjalan lebih dari setengah jumlah anggota *cluster*.

Downtime pada *node failure* tergantung pada sistem operasi yang digunakan oleh server virtual. Rata-rata *downtime* server virtual dengan sistem operasi Ubuntu 20.04 lebih kecil dibandingkan dengan mesin virtual dengan sistem operasi Windows Server 2016. Perbedaan ini terjadi karena perbedaan waktu *booting* yang dibutuhkan oleh masing-masing sistem operasi sampai dengan *service* dijalankan. Dari hasil pengukuran *downtime* ini, pemilihan sistem operasi cukup berpengaruh pada efektifitas sistem *cluster* dengan *high availability* yang diterapkan. Sistem operasi Ubuntu Server untuk menjalankan berbagai layanan menjadi pilihan yang lebih baik kecuali untuk layanan yang hanya bisa berjalan dengan sistem operasi Windows.

IV. KESIMPULAN DAN SARAN

Dari hasil implementasi dan pengujian, ada beberapa temuan penting yang dapat dihasilkan dari kajian ini. Pertama, sistem *cluster* yang menerapkan *high availability* pada virtual server dapat dibangun dengan Proxmox VE 6.2 pada tiga unit mesin server fisik dan NAS Server yang dihubungkan melalui switch berkecepatan tinggi. Virtualisasi server dengan Proxmox VE 6.2 mendukung pembuatan server virtual yang berbasis Container (lxc) maupun virtual machine (qemu). Kedua, tipe server virtual ini mendukung *High Availability* (HA) saat terjadi kegagalan pada salah satu *node* sistem *cluster*. HA pada Proxmox VE 6.2 dapat diterapkan jika terdapat minimal tiga buah *node* yang bergabung dalam sistem *cluster*. HA akan berjalan saat terjadi kegagalan pada *node* jika masih terdapat dua *node* atau lebih yang aktif.

Ketiga, penerapan HA pada server virtual meningkatkan efektifitas sistem *cluster* dalam menjalankan layanan berbasis komputer. Keempat, *downtime* saat terjadi kegagalan *node* sangat efisien, namun tergantung pada sistem operasi yang digunakan dalam server virtual. Server virtual dengan sistem operasi Ubuntu Server 20.04 rata-rata membutuhkan waktu 3.11 menit untuk kembali melayani saat terjadi kegagalan *node*. Sedangkan server virtual dengan sistem operasi Windows Server 2016 memiliki rata-rata *downtime* sebesar 11 menit saat terjadi

kegagalan *node*. *Downtime* yang terjadi selama *failover* akibat kegagalan *node* tergantung pada kecepatan *booting* masing-masing sistem operasi yang digunakan oleh server virtual.

Penerapan sistem *cluster* dengan *high availability* di BPS Provinsi Sumatera Utara masih perlu pengembangan lebih lanjut dengan menerapkan mitigasi *failure* pada perangkat pendukung seperti *switch* dan *power supply*. Perangkat NAS sebagai media penyimpanan *image* mesin virtual juga menjadi titik rawan kegagalan mekanisme *high availability* yang diterapkan sehingga kedepannya perlu dibuat *backup* perangkat NAS yang dapat melakukan *takeover* jika perangkat utama mengalami *failure*. Penelitian ini masih terbatas pada efektifitas *high availability* jika server fisik mengalami *failure*, peneliti kedepan diharapkan bisa mengukur efektifitas *high availability* jika perangkat-perangkat lain seperti NAS, Switcch dan LAN yang mengalami *failure*.

DAFTAR PUSTAKA

- Adi, Y. R., Nurhayati, O. D., & Widiyanto, E. D. (2016). Perancangan Sistem Cluster Server untuk Jaminan Ketersediaan Layanan Tinggi pada Lingkungan Virtual. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi (JNTETI)*, 5(2). <https://doi.org/10.22146/jnteti.v5i2.228>
- Agustian, B., Susanto, D., Informatika, T., Pamulang, U., Selatan-indonesia, T., Server, C. S., Server, F., Server, W., & Server, D. (2018). *VIRTUALIZATION OF SERVER WITH PROXMOX*. 3, 133–138.
- Al-Aomar, R., Aljeneibi, S., & Almazroui, S. (2016). Reducing operational downtime in service processes: A six sigma case study. *ICIMSA 2016 - 2016 3rd International Conference on Industrial Engineering, Management Science and Applications*, 1–5. <https://doi.org/10.1109/ICIMSA.2016.7504029>
- Ali, E., Susandri, & Rahmaddeni. (2015). Optimizing Server Resource by Using Virtualization Technology. *Procedia Computer Science*, 59(Iccsci), 320–325. <https://doi.org/10.1016/j.procs.2015.07.572>
- Bernstein, D. (2014). Containers and cloud: From LXC to docker to kubernetes. *IEEE Cloud Computing*, 1(3), 81–84. <https://doi.org/10.1109/MCC.2014.51>
- Birke, R., Giurgiu, I., Chen, L. Y., Wiesmann, D., & Engbersen, T. (2014). Failure analysis of virtual and physical machines: Patterns, causes and characteristics. *Proceedings of the International Conference on Dependable Systems and Networks*, 1–12. <https://doi.org/10.1109/DSN.2014.18>
- Gmbh, P. S. S. (2020). *Proxmox VE Administration Guide Release 6.2*. Proxmox Server Solutions Gmbh.
- Goldman, J., & Rawles, P. (2004). *Applied Data Communications: A Business-oriented Approach (4th Edition)*. 608. http://cis.msjs.edu/courses/core_courses/csis202/lessons/10/ch10.pdf
- Harper, R. E., Pelleg, D., & Spainhower, L. F. (2009). Using virtualization for high availability and disaster recovery. *IBM Journal of Research and Development*, 53(4), 1–11.
- Kovari, A., & Dukan, P. (2012). KVM & OpenVZ virtualization based IaaS open source cloud virtualization platforms: OpenNode, Proxmox VE. *2012 IEEE 10th Jubilee International Symposium on Intelligent Systems and Informatics, SISY 2012*, 335–339. <https://doi.org/10.1109/SISY.2012.6339540>
- Kumar, R., & Charu, S. (2015). An Importance of Using Virtualization Technology in Cloud Computing. *Global Journal of Computers & Technology*, 1(2), 56–60.
- Ljubojevi, M., & Bajic, A. (2019). Implementation of High-Availability Server Cluster by Using Fencing Concept. *18th International Symposium INFOTEH-JAHORINA (INFOTEH), March*, 20–22.

- Portnoy, M. (2016). *Virtualization Essentials* (Second Edi). John Wiley & Sons, Inc.
- Rianafirin, K., & Kurniawan, M. T. (2018). Design network security infrastructure cabling using network development life cycle methodology and ISO/IEC 27000 series in Yayasan Kesehatan (Yakes) Telkom Bandung. *Proceedings of the 2017 4th International Conference on Computer Applications and Information Processing Technology, CAIPT 2017, 2018-Janua*, 1–6. <https://doi.org/10.1109/CAIPT.2017.8320681>
- Scroggins, R. (2017). Emerging Virtualization Technology. *Global Journal of Computer Science and Technology*, 17(3), 11–16. <https://doi.org/10.34257/gjcssthvol17is3pg11>
- Suryono, T., & Afif, M. F. (2012). Pembuatan Prototype Virtual Server Menggunakan Proxmox Ve Untuk Optimalisasi Resource Hardware Di NOC FKIP UNS. *IJNS-Indonesian Journal on Networking and Security*, 1(November), 1–5.