

# LIVE MIGRATION PADA CLOUD COMPUTING BERBASIS PROXMOX DENGAN METODE PRE-COPY

## LIVE MIGRATION IN CLOUD COMPUTING USING PRE-COPY METHOD ON PROXMOX

Nur Muthiah<sup>1</sup>, Andrew Brian Osmond, S.T., M.T.<sup>2</sup>, Roswan Latuconsina, S.T., M.T.<sup>3</sup>

<sup>1,2,3</sup>Prodi S1 Sistem Komputer, Fakultas Teknik Elektro, Universitas Telkom

<sup>1</sup>[nurmuthiah@student.telkomuniversity.ac.id](mailto:nurmuthiah@student.telkomuniversity.ac.id), <sup>2</sup>[abosmond@telkomuniversity.ac.id](mailto:abosmond@telkomuniversity.ac.id), <sup>3</sup>[Roswan@telkomuniversity.ac.id](mailto:Roswan@telkomuniversity.ac.id)

---

### Abstrak

*Cloud computing* merupakan salah satu teknologi yang memungkinkan pengguna untuk mengisi penyimpanan data informasi. Data informasi secara permanen tersimpan di *server* melalui internet dan tersimpan secara sementara di komputer pengguna, seperti tersimpan di dalam desktop, tablet android ataupun ipad, dan *notebook*. Sumber daya yang dipakai pada *cloud computing* ini, dapat dimanfaatkan sesuai kebutuhan pengguna dan membayar sumber daya yang di ambil atau terpakai pada *cloud computing* sesuai yang digunakan.

Manfaat *resource* yang tersimpan pada *cloud computing* dengan sumber daya komputasi tersebut dapat dipenuhi oleh layanan *cloud Infrastructure as a Service (IaaS)*. Sistem yang memberi pengguna kemampuan untuk menjalankan dan mengendalikan seluruh akses, seperti contohnya mesin virtual yang digunakan untuk melintasi berbagai sumber fisik, seperti halnya *switch*, *processor*, RAM atau memori, dan penyimpanan *disk* yang sesuai untuk kebutuhan pengguna.

Konsep *live migration* pada *cloud computing* dibangun dengan mengacu pada arsitektur proxmox. Proxmox mempunyai arsitektur infrastruktur berbasis *website* dan bersifat *private server*. Salah satu teknologi yang berhubungan dengan *live migration* dan menerapkan konsep tersebut adalah *web service oriented architecture*. *Live migration* mendukung penyimpanan status mesin virtual pada saat itu dan dimulai lagi pada lain waktu, sehingga membuat *downtime* lebih besar dengan ukuran data transfer menjadi besar sehingga waktu migrasi menjadi semakin lama dengan pengaruh *migration speed (bandwidth)* yang diukur besar.

**Kata kunci:** *Cloud Computing, Live Migration, Proxmox.*

---

### Abstract

Cloud computing is one of the technologies that allows users to fill information storage data. Data information is permanently stored on the server over the internet and stored temporarily on the user's computer, such as stored in the desktop, android tablet or ipad, and notebook. Resources used in this cloud computing, can be utilized according to user needs and pay for resources that are taken or used in cloud computing as used.

The benefits of resources stored in cloud computing with these computing resources can be met by cloud Infrastructure as a Service (IaaS). Systems that give users the ability to run and control all access, such as virtual machines used to traverse multiple physical sources, such as switches, processors, RAM or memory, and disk storage that is suitable for user needs.

The concept of live migration in cloud computing is built by referring to the proxmox architecture. Proxmox has a website-based infrastructure architecture and is a private server. One of the technologies related to live migration and applying the concept is a web service oriented architecture. Live migration supports storing virtual machine status at that time and starts again at a later time, making larger downtime with large data transfer sizes so that the migration time becomes longer with the influence of large measured migration speed (bandwidth).

**Keywords:** *Cloud Computing, Live Migration, Proxmox.*

---

### 1. Pendahuluan

Seiring dengan perkembangan teknologi dan informasi yang serba canggih seperti sekarang saat ini, perubahan teknologi informasi mengalami perkembangan yang lebih efisien dengan kemudahan akses, sehingga dalam kegiatan sehari-hari yang dianggap tidak mungkin dapat dikerjakan dalam waktu singkat, maka saat ini dapat dilakukan. Pengembangan teknologi komputasi berbasis internet pada masa sekarang lebih diarahkan pada proses aplikasi sistem yang mudah dan tidak memerlukan banyak waktu atau tenaga.

Teknologi *cloud computing* atau biasa dikenal dengan komputasi awan sekarang, sudah banyak digunakan dan bukan merupakan hal asing lagi. Teknologi *cloud computing* dapat lebih menghemat pengeluaran dengan dibandingkan harus membangun sendiri infrastruktur jaringan dalam jangka pendek. Kebutuhan untuk teknologi *cloud computing* sebagian besar hanya pada biaya koneksi dan data *processing* yang sesuai dengan kebutuhan. Pada sebagian perusahaan atau institusi, penggunaan media penyimpanan data dan keamanan menjadi faktor utama yang harus diperhatikan. Biaya yang harus dikeluarkan untuk mengembangkan media penyimpanan data dan keamanan sangatlah besar terlebih lagi untuk perusahaan-perusahaan yang berskala besar.

Proxmox merupakan sebuah *platform* perangkat lunak yang mengelola infrastruktur dengan menerapkan gaya IaaS yang terukur pada sifat *privat server*. Platform proxmox menjadi sebuah solusi manajemen virtualisasi *server open source* berbasis Qemu/KVM dan LXC yang lengkap, ringan dan mudah. Proxmox menawarkan kemampuan untuk mengelola *server virtual* (VPS) dengan teknologi Linux OpenVZ dan teknologi KVM. Proxmox termasuk distro Linux virtualisasi berbasis Debian (64 bit), dengan KVM kita tidak hanya bisa menginstal linux saja akan tetapi, *operating system windows* pun bisa kita instal, namun, yang membuat istimewa dari proxmox adalah kemudahan dalam instalasi dan administrasi berbasis *web*. Dengan menginstal proxmox, maka *cloud* kita dapat memiliki kemampuan *live migration* dari *virtual machine* dengan melalui *interface web*. *Live Migration* merupakan kemampuan untuk memindahkan mesin virtual yang sedang berjalan kepada *host* fisik lain tanpa mengurangi interupsi kepada layanan *hypervisor*. *Live migration* transparan kepada pengguna akhir, mesin virtual tetap menyala, koneksi jaringan tetap aktif, dan aplikasi pengguna tetap berjalan saat mesin virtual dipindahkan ke *host* fisik baru. *Live migration* juga mendukung penyimpanan status mesin virtual pada saat itu dan dimulai lagi pada lain waktu. Pada *live migration* memungkinkan konten layanan tetap dapat diakses selama proses *live migration* terjadi [8].

## 2. Dasar Teori

### 2.1 Cloud Computing

*Cloud computing* merupakan salah satu teknologi yang memungkinkan pengguna untuk mengisip penyimpanan data informasi yang dimana, data informasi secara permanen tersimpan di *server* melalui internet dan tersimpan secara sementara di komputer pengguna, seperti tersimpan di dalam desktop, tablet android ataupun ipad, dan *notebook*. Sumber daya yang dipakai pada *cloud computing* ini, dapat dimanfaatkan sesuai kebutuhan pengguna dan membayar sumber daya yang di ambil atau terpakai pada *cloud computing* sesuai yang digunakan. Konsumen tidak mengelola atau mengendalikan infrastruktur *cloud computing* yang mendasari tetapi memiliki kontrol atas sistem operasi, penyimpanan, aplikasi yang dijalankan, dan mungkin kontrol terbatas komponen jaringannya, seperti misalnya *firewall host*. Model bisnisnya mirip dengan penyedia pusat data yang menyewakan ruangan untuk *colocation*, tapi ini lebih ke level mikronya. Penyewa tidak perlu tau dengan mesin apa dan bagaimana caranya penyedia layanan menyediakan layanan IaaS yang terpenting permintaan mereka atas sumber daya dasar teknologi informasi itu dapat dipenuhi [4].

### 2.2 Hypervisor

*Hypervisor* adalah sebuah teknik virtualisasi yang memungkinkan beberapa *operating system* untuk berjalan bersamaan pada sebuah *host*. Dikatakan teknik virtualisasi, karena OS yang ada bukanlah sebuah OS yang sesungguhnya tetapi hanya sebuah *virtual machine* saja. Tugas dari *hypervisor* adalah untuk mengatur setiap *operating system* tersebut sesuai dengan gilirannya agar tidak mengganggu satu dengan yang lainnya. Terkadang, *hypervisor* juga disebut sebagai *Virtual Machine Management (VMM)* yang sesuai dengan tugasnya dalam mengatur beberapa *virtual machine* [5].

### 2.3 Kernel Based Virtual Machine

KVM (*Kernel Based Virtual Machine*) merupakan bagian integral linux sejak tahun 2007 yang merupakan sebuah solusi untuk melakukan virtualisasi penuh pada Linux dengan perangkat keras x86 yang mengandung ekstensi virtualisasi (Intel VT atau AMD-V). Dengan menggunakan KVM, seseorang dapat menjalankan beberapa mesin virtual pada Linux atau Windows. Setiap mesin virtual memiliki perangkat keras virtual pribadi yang berupa kartu jaringan, *disk*, dan adapter grafis. KVM termasuk perangkat lunak *open source* yang di dalamnya terdapat *mainline Linux*, pada 2.6.20. Komponen *userspace* dari KVM termasuk dalam *mainline QEMU*. KVM diimplementasikan sebagai modul kernel *loadable* yang mengubah kernel Linux menjadi *baremetal hypervisor* [12].

### 2.4 Proxmox

Proxmox VE (*Virtual Environment*) adalah sebuah solusi manajemen virtualisasi *server open source* berbasis Qemu/KVM dan LXC yang lengkap, ringan dan mudah. Proxmox menawarkan kemampuan untuk mengelola *server virtual* (VPS) dengan teknologi Linux OpenVZ dan teknologi KVM. Proxmox termasuk distro Linux virtualisasi berbasis Debian (64 bit), dengan KVM kita tidak hanya bisa menginstal linux saja akan tetapi, *operating system windows* pun bisa kita instal namun, yang membuat istimewa dari proxmox adalah kemudahan dalam instalasi dan administrasi berbasis *web*. Proxmox dikembangkan oleh Proxmox *Server Solutions* di Austria di bawah Yayasan Internet Austria dan dilepaskan di bawah GPL (*General Public License*), karena ini merupakan solusi *open source* yang dapat di sesuaikan sesuai dengan kebutuhan.

*Live migration* pada proxmox mempunyai kemampuan memindahkan sebuah VM dari *node* satu ke *node* lainnya. Hal ini dibutuhkan bila ada peristiwa perawatan mesin *server (node)* dan layanan yang ada dalam *server* dipaksa tidak boleh terganggu, maka dari itu kita dapat memindahkan *virtual machine* yang ada dalam *node* ke *node* lain. Manajemen proxmox dilakukan melalui antarmuka *website* berdasarkan kerangka kerja *javascript* yang memungkinkan *administrator* untuk mengontrol semua fitur. Proxmox menawarkan model penyimpanan yang fleksibel. Gambar mesin virtual dapat disimpan pada satu atau beberapa penyimpanan lokal pada penyimpanan bersama seperti NFS dan SAN. Proxmox mendukung VLAN, ikatan dan agregasi jaringan yang memungkinkan untuk membangun jaringan virtual fleksibel yang kompleks untuk dihubungkan ke adapter jaringan fisik untuk *server host* yang diberi konfigurasi jaringan (TCP/IP) dengan memanfaatkan kekuatan penuh dari tumpukan jaringan Linux. Beberapa keuntungan dengan menggunakan proxmox, di antaranya:

- a. Kinerja terbaik,
- b. Instalasi yang telah dioptimalkan, sehingga lebih cepat,
- c. Mudah dalam manajemen, dan
- d. Cocok untuk kelas *enterprise* [11].

## 2.5 Data Transfer

Data transfer adalah total ukuran data yang ditransfer selama proses migrasi, meskipun lebih rendah dibatasi oleh ukuran VM dan itu tergantung pada teknik migrasi yang digunakan [14].

## 2.6 Downtime

*Downtime* adalah waktu henti yang berdurasi fase ketika berhenti dan menyalin. VM tidak aktif pada sumber atau di tempat tujuan selama *downtime* dengan rumus  $Tdt \propto WWS(i-1)$  [14].

## 2.7 Migration Time

Waktu migrasi adalah lama waktu migrasi dari suatu *host* ke *host* yang lain dengan waktu yang dibutuhkan untuk menyelesaikan sepenuhnya semua tiga fase migrasi, di antaranya yaitu *push phases*, hentikan dan salin dan *pull phases*[14].

## 2.8 Live Migration

*Live migration* merupakan perpindahan mesin virtual dari satu mesin *worker node* ke mesin *worker node* lainnya yang dalam keadaan terus menyala. Ketika dilakukan dengan benar, maka proses ini berlangsung tanpa ada efek yang nyata dari sudut pandang pengguna akhir, contohnya pengaplikasian *live migration* yang memungkinkan seorang *administrator* untuk memindahkan mesin virtual untuk melakukan pemeliharaan atau *upgrade* tanpa mematikan sistem pengguna untuk *downtime*. Salah satu keuntungan yang paling signifikan dari *live migration* adalah kenyataan bahwa kemampuan ini memfasilitasi pemeliharaan yang proaktif. Jika, sebuah kegagalan segera dicurigai, maka potensi masalah dapat diselesaikan sebelum terjadi gangguan layanan. *Live migration* dapat digunakan untuk *load balancing* yang dimana, beban kerja dibagi bersama antara komputer dalam rangka mengoptimalkan pemanfaatan sumber daya CPU yang tersedia. Cara kerja dari *live migration* agar mampu memindahkan mesin virtual dari satu mesin fisik ke mesin fisik lainnya dengan melalui 5 tahapan, di antaranya:

- a. Inisiasi perintah untuk melakukan *live migration*.
- b. Halaman memori mesin virtual ditransfer dari *host* fisik sumber ke *host* fisik tujuan. Pada fase ini, dapat dimodifikasi pada halaman memori mesin virtual yang akan dilacak.
- c. Penanganan penyimpanan mesin virtual dipindahkan dari *host* fisik sumber setelah halaman memori tersalin semua.
- d. Mesin virtual dipindahkan dan dinyalakan di *host fisik tujuan*.
- e. Perbaiki informasi jaringan setelah mesin virtual dipindahkan [2].

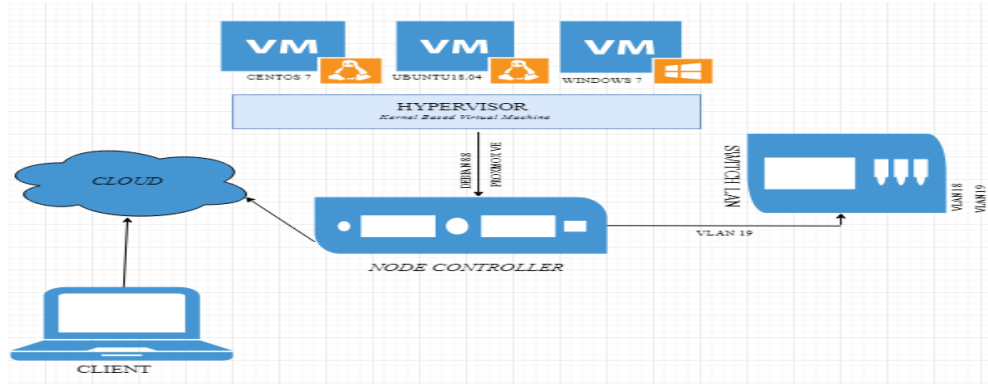
## 2.9 Algoritma Pre-Copy

*Pre-copy* adalah salah satu virtual yang populer dalam memetodekan migrasi langsung sebelum VM mesin dimigrasikan secara iteratif pada transfer halaman memori yang diperbarui dengan memigrasikan citra mesin virtual yang konsisten, sehingga dapat lebih efisien dan termasuk salah satu fitur defaultnya *qemu*. *Pre-copy* dalam pendekatan utama untuk *live VM migration* dan juga termasuk pendekatan berbasis *hypervisor* dan *VMware*, *Xen* sedangkan *KVM* dengan pendekatan OS level yang tidak menggunakan *hypervisors* dan *OpenVZ*. Pada gambar mekanisme umum algoritma *pre-copy*, *pre-copy* bekerja dengan melakukan penyalinan *memory state* beberapa kali hingga yang terakhir adalah penyalinan *processor state*. Tujuan utama penggunaan *pre-copy* adalah untuk menjaga *downtime* tetap kecil dengan meminimalkan jumlah VM *state* yang ditransfer selama *downtime*, sehingga total *migration time* dapat diminimalisasi. Keuntungan menggunakan metode *pre-copy*, yaitu salah satunya *downtime* < 1 second pada saat pemrosesan dibatalkannya migrasi, maka sistem tidak akan terjadi kegagalan. Kerugiannya, yaitu terjadinya *overhead* dari halaman duplikat transmisi data yang mengakibatkan akan terjadinya *corrupt* data. Apabila ada data yang *corrupt* pada saat pemrosesan, maka proses *pre-copy* pun akan secara langsung berhenti [10].

## 3. Gambaran Umum Sistem

### 3.1 Diagram Blok Umum

Pada gambar 3.1 menjelaskan tentang arsitektur diagram blok *live migration* langsung pada mesin virtual yang dimana mesin virtual ditransfer dari satu *host* fisik ke *host* yang lain, yaitu berupa *server* fisik yang menjadi *node controller* dan satu *laptop* yang berfungsi sebagai *client*.



Gambar 3.1 Diagram Blok Umum Live Migration

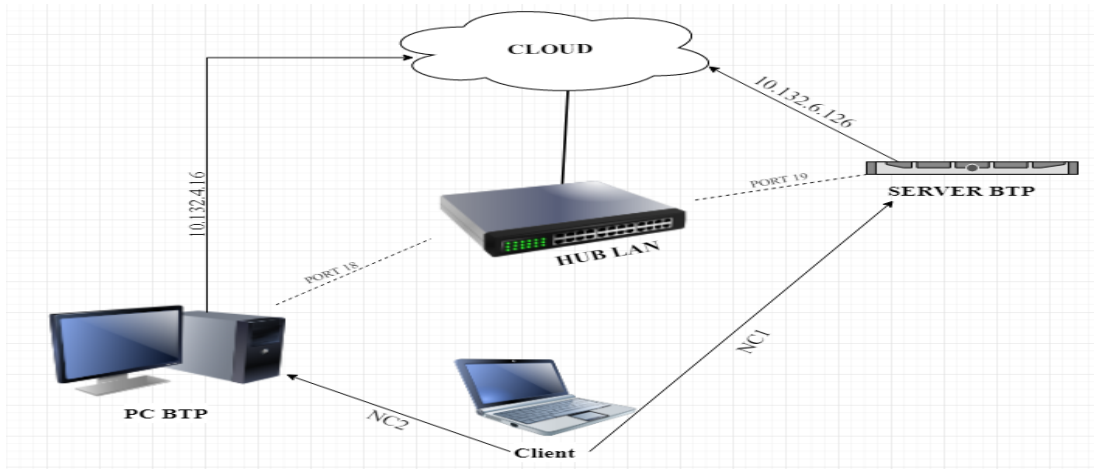
Hypervisor bertindak sebagai antarmuka antara perangkat keras yang mendasarinya virtual mesin di server fisik dengan client. Hypervisor dengan melakukan semua tugas manajemen mesin virtual tanpa mengganggu layanan mesin virtual pada berbagai alat virtualisasi komersial dengan open source yang tersedia. Pada hypervisor ini, untuk memvirtualisikannya digunakan proxmox VE (Virtual Environment). Proxmox VE adalah salah satu platform yang menjadikan sebuah solusi manajemen virtualisasi server open source dengan dukungan teknologi KVM (Kernel Based Virtual Machine). Mesin virtual yang akan digunakan ada tiga, yaitu VM 100 sebagai centos, VM 101 sebagai ubuntu, dan VM 103 sebagai windows. Dari ketiga VM tersebut terinstal di server yang sebagai NC sumber. NC sumber ini yang terhubung pada switch LAN dengan menggunakan VLAN 19. Virtual mesin tersebut dapat dijelaskan pada tabel 3.1. Berikut tabel spesifikasi detail mesin virtual.

Tabel 3.1 Spesifikasi Detail Mesin Virtual

Komponen	Mesin Virtual		
	VM 100	VM 101	VM 103
Hypervisor	KVM	KVM	KVM
Name	Centos 7-x86_64 bit	Ubuntu Dekstop 18.04	Win7 Ultimate
HDD size	32 G	32 G	32 G
Storage	SCSI	IDE	IDE
CPU	1 Core	1 Core	1 Core
Memory	4 GiB	4 GiB	4 GiB
NIC	Intel E1000	Intel E1000	Intel E1000

### 3.2 Perancangan Topologi Jaringan Cloud

Pada gambar 3.2 ini menggambarkan dua mesin fisik, yaitu berupa satu server BTP dan satu PC BTP bermodelkan topologi mesh. Server fisik sebagai node controller pertama dan PC BTP sebagai node controller kedua. Pada tiap kedua node controller yang digunakan pada penelitian ini berbeda prosesor. Pada NC pertama menggunakan prosesor Server Intel® Xeon® inside-Dell PowerEdge R610 dan sedangkan pada NC kedua menggunakan prosesor Intel® Core i5-6600 CPU @ 3.30 GHz. Node controller (NC), yaitu node yang di jalankan di setiap node proxmox yang akan mengatur siklus kehidupan dari setiap VM (Virtual Machine) yang sedang berjalan di node. Di satu sisi, NC berinteraksi dengan sistem operasi dan hypervisor yang berjalan di node. NC akan menanyakan sistem operasi yang sedang berjalan di node untuk mengetahui sumber daya fisik yang digunakan pada node, seperti jumlah core, besar memory, dan ketersediaan disk. Node controller juga berfungsi sebagai pusat data yang terkait dengan ketersediaan dan penggunaan sumber daya di node. Switch atau HUB LAN yang digunakan pada penelitian ini berjumlah satu unit dengan tujuan untuk menjaga konektivitas jaringan, agar server dapat terhubung ke jaringan lain. Kemudian, pada kedua mesin fisik ini, akan di instal debian yang sebagai OS (Operating System), lalu didalam debian akan di instal manual proxmox VE dan di dalam proxmox tersebut sudah ter-include QEMU/KVM yang sebagai hypervisor default. Untuk menginstal semua VM beserta hypervisor, maka di butuhkan koneksi internet baik yang terhubung pada HUB LAN. Setelah, semua terinstal sesuai dengan yang di butuhkan, maka di dapatkan IP yang sudah di setting pada HUB LAN. IP pada server BTP, yaitu 10.132.6.126 yang sebagai NC pertama dengan port LAN 19 dan IP pada PC BTP, yaitu 10.132.4.16 yang sebagai NC kedua dengan port LAN 18 dengan kedua NC memiliki gateway, DNS, dan subnetmask yang sama, yaitu gateway 10.132.4.1, DNS server 8.8.8.8 dan subnetmask 255.255.252.0. Kemudian selanjutnya, akan di uji migrasi data disk local VM dan file berupa video streaming melalui client yang diakses lewat router dan tidak langsung dari PC ataupun server. Pengujian akan dilakukan dari sumber node, yaitu dari server BTP ke sumber tujuan, yaitu PC BTP. Kemudian, file tersebut di copy beserta disk local VM dari NC pertama ke NC kedua dengan mengikuti algoritma pre-copy. Untuk pengujian video streaming dilakukan saat video dalam keadaan di putar di satu VM.



Gambar 3.2 Topologi Cloud Fisik

Ketika VM *online*, maka dapat dilakukan sebuah migrasi dengan menggunakan perintah (*command*) di terminal, sebagai berikut “qm migrate VMID targetnode --online --with-local-disks”. Lalu dari pengcopian tersebut, akan terjadinya pengukuran *downtime* ketika saat *stop and copy*, *migration time* dan ukuran data transfer. Selanjutnya, setelah semua selesai migrasi, maka di dapatkan hasil data pengujian dengan berupa tabel dan grafik.

Untuk mengimplementasikan *cloud* pada server, maka dibutuhkan spesifikasi perangkat keras (*hardware*) dan perangkat lunak (*software*) di perancangan tugas akhir kali ini, berikut di antaranya:

Tabel 4.1 Spesifikasi Virtualisasi

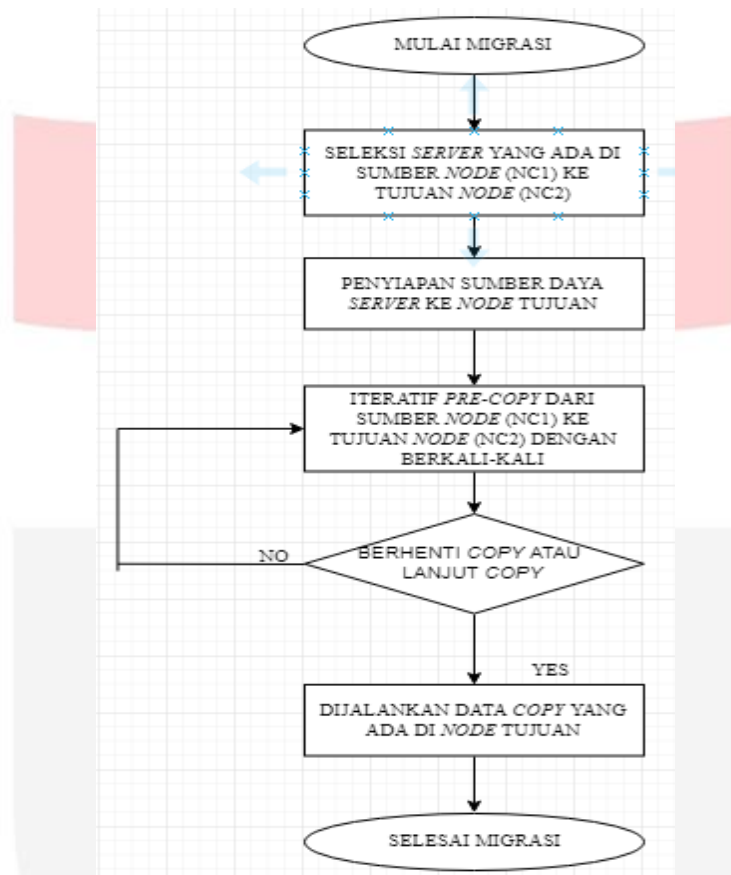
Node Controller	Hardware	Software	Network
NC1	Server Intel® Xeon® inside- Dell PowerEdge R610	Hypervisor: KVM	IP: 10.132.6.126
	RAM: 16 GB	Operating System: Debian 8.8-64 bit	Gateway: 10.132.4.1
	HDD: 600 GB	HDD size: 572.5 GB	DNS server: 8.8.8.8
	NIC: Bridged mode	RAM: 8 GiB	Subnetmask: 255.255.252.0
		CPU: E5620 @2.40 GHz	
		NIC: Intel E1000	
NC2	Intel ® Core i5-6600 CPU @ 3.30 GHz	Hypervisor: KVM	IP: 10.132.4.16
	RAM: 8 GB	Operating System: Debian 8.8-64 bit	Gateway: 10.132.4.1
	HDD: 1 Tera	HDD size: 967.5 GB	DNS server: 8.8.8.8
		RAM: 8 GiB	Subnetmask: 255.255.252.0
	NIC: NAT mode	CPU: @ 3.30 GHz	
		NIC: Intel E1000	

Mesin virtual yang digunakan dalam penelitian ini, ada tiga macam, yaitu VM 100, VM 101, dan VM 103. Mesin virtual tersebut, terinstal dan tersimpan pada NC pertama yaitu di *server* Xeon.

### 3.3 Perancangan Mekanisme Migrasi

Pada gambar flowchart 3.3 dijelaskan alur algoritma pengujian pre-copy dengan menjalankan mesin virtual *host* dengan 6 tahapan, di antaranya:

1. Proses migrasi dimulai dengan menyeleksi server yang ada di sumber *node* ke *node* tujuan dengan adanya gangguan layanan yang lebih sedikit untuk diperbaiki dalam pengurangan biaya *overhead* yang di berikan ke metode migrasi dengan kebutuhan dalam pembuatan mesin virtual yang efisien.
2. Dilakukannya proses penyiapan sumber daya *server* ke *node* tujuan, seperti RAM ataupun *harddisk* yang kosong.
3. Kemudian, dilakukannya proses iteratif *pre-copy* dari sumber *node* ke *node* tujuan dengan berkali-kali proses. Pada saat proses iteratif *pre-copy*, data yang di iteratif berupa *migration time*, ukuran data transfer dan *downtime*. *Downtime* akan terjadi selama menjalankan virtual mesin yang dihentikan dengan berisi pengiriman status prosesor.
4. Lalu, dilakukannya tahapan pengecekan. Apakah berhenti atau melanjutkan *pre-copy* data? Apabila saat migrasi data berhenti, maka akan mengulang ke tahapan sebelumnya, yaitu mengiteratif *pre-copian* tetapi jika, tidak berhenti, maka akan tetap melanjutkan *pre-copy*, dan akan masuk ke tahapan berikutnya.
5. Tahapan berikutnya, melakukan proses jalannya data *copy* yang ada di *node* tujuan dan akan diambil data akhir sebagai kesimpulan.
6. Terakhir, tahapan proses migrasi selesai.



Gambar 3.3 Flowchart Metode Pre-Copy

#### 4. Hasil dan Analisa

##### 4.1 Skenario Uji Migrasi

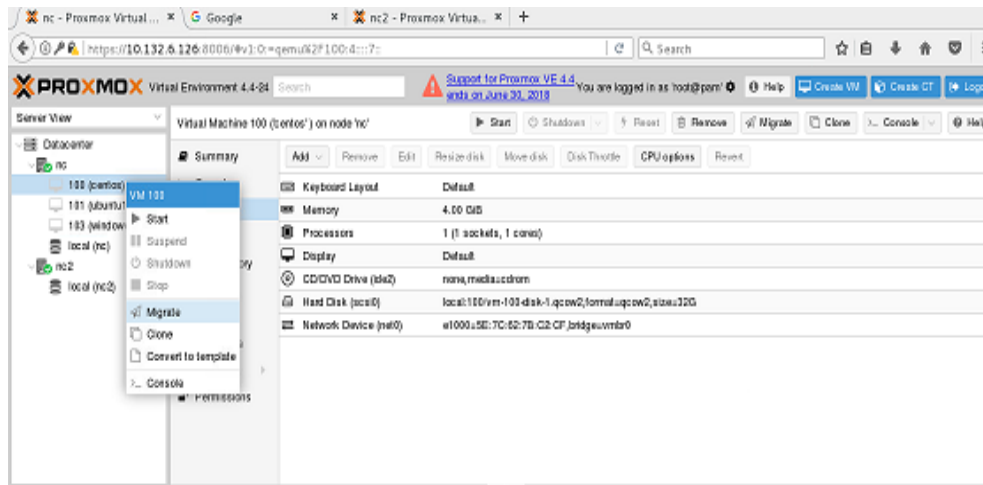
Pada tahapan ini, akan dibahas tentang skenario pengujian *Live Migration* dengan beberapa konten yang akan di uji, seperti di antaranya:

1. Skenario uji VM
2. Skenario uji video *streaming*

##### 4.1.2 Skenario Uji Migrasi VM

1. Cara Pengujian

Pengujian ini, dilakukan dengan cara fitur *migrate* yang ada pada proxmox VE seperti pada gambar 4.19 dibawah ini. Pengujian VM dilakukan ketika VM sudah dibuat dan VM dalam keadaan mati. Ketika VM ingin melakukan uji migrasi, maka di ubah terlebih dahulu "CD/DVD *drive.iso*" yang ada pada spesifikasi *hardware* ke "don't use any media". Setelah itu, dilakukan migrasi dengan fitur *migrate* dengan klik kanan dari VM. Pengujian ini, dilakukan dari *node* sumber (NC1) dengan IP 10.132.6.126 ke *node* tujuan (NC2) dengan IP 10.132.4.16, kemudian VM yang telah di migrasi akan pindah ke *node* tujuan (NC2).



Gambar 4.19 Cara Uji VM

2. Alat Uji
  - 1.) Satu buah *server* fisik
  - 2.) Satu buah perangkat computer
  - 3.) Satu buah laptop yang sebagai *client*
3. Tujuan Pengujian  
 Pengujian ini dilakukan untuk mengetahui seberapa besar file yang telah di migrasikan dari *node* sumber (NC1) ke *node* tujuan (NC2) dan untuk mengetahui migrasi dapat berjalan dengan sukses atau gagal saat dilakukannya migrasi.
4. Jumlah Pengujian  
 Pengujian VM ini, hanya dilakukan sebanyak tiga puluh kali uji coba pada setiap VM nya.

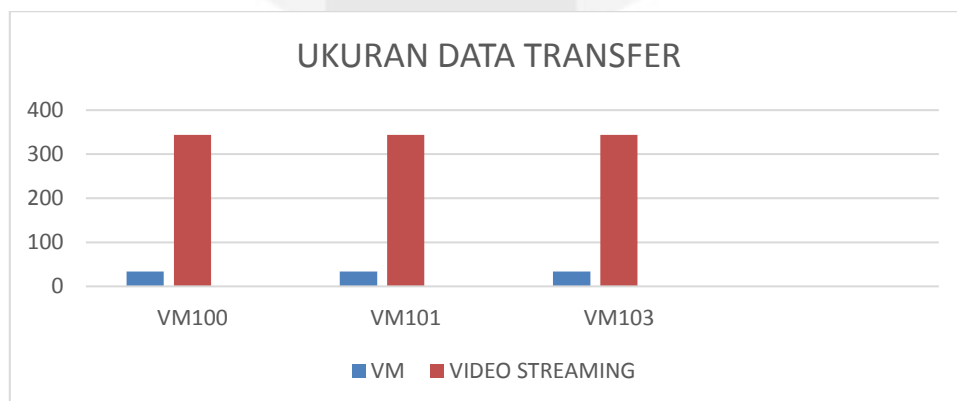
**4.1.3 Skenario Uji Migrasi Video Streaming**

1. Cara Pengujian  
 Pengujian ini, dilakukan dengan cara ketika VM sedang di akses, lalu pada terminal OS dilakukannya *command* migrasi VM “qm migrate VMID targetnode --online --with-local-disks”. Kemudian saat kondisi VM nyala, video atau youtube yang sedang di akses pada VM tersebut, dilakukannya migrasi pada terminal. Pengujian ini, dilakukan dari *node* sumber (NC1) dengan IP 10.132.6.126 ke *node* tujuan (NC2) dengan IP 10.132.4.16, setelah itu, VM yang telah di migrasi akan pindah ke *node* tujuan (NC2).
2. Alat Uji
  - 1.) Satu buah *server* fisik
  - 2.) Satu buah perangkat computer
  - 3.) Satu buah laptop yang sebagai *client*
3. Tujuan Pengujian  
 Pengujian ini dilakukan untuk mengetahui seberapa besar file video yang sedang di akses pada ketiga VM yang dapat di migrasikan dengan sukses atau gagal dari NC1 ke NC2.
4. Jumlah Pengujian  
 Pengujian VM ini, hanya dilakukan sebanyak tiga puluh kali uji coba pada setiap VM nya.

**4.2 Hasil Pengujian Migrasi**

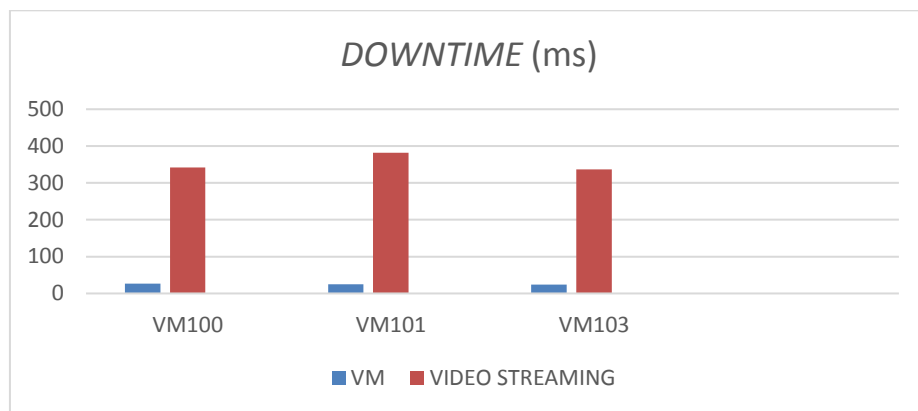
Berikut grafik batang dari hasil pengujian migrasi, yaitu:

A. Berikut grafik batang dari hasil pengujian ukuran data transfer



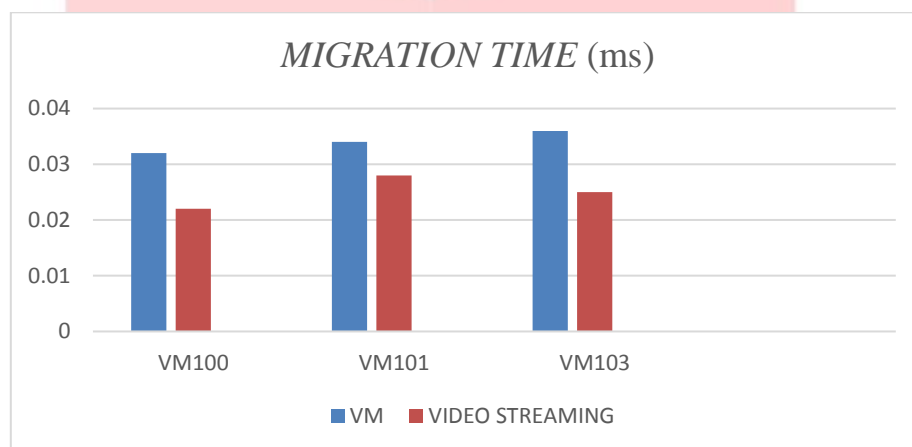
Gambar 4.2.1 Grafik Ukuran Data Transfer

B. Berikut grafik batang dari hasil pengujian *downtime*



Gambar 4.2.2 Grafik Downtime

C. Berikut grafik batang dari hasil pengujian *migration time*



Gambar 4.2.3 Grafik Migration Time

Pada tabel 4.4 dan pada gambar 4.25, gambar 4.26, dan gambar 4.27 di jelaskan hasil pengujian migrasi dari Youtube yang sedang di putar di ketiga VM tersebut di dapatkan hasil *downtime*, ukuran data transfer, dan *migration time* yang berbeda-beda. Pada saat pengujian migrasi youtube tersebut, pengujian ini sama seperti pengujian migrasi VM. Ukuran data file yang di migrasikan langsung dari youtube yang sedang di akses, sehingga mengakibatkan pengukuran yang jauh lebih besar signifikan dalam *bytes*. Pada pengujian migrasi ini, dilakukan sebanyak tiga puluh kali migrasi pada setiap VMnya yang mengakibatkan butuh waktu kurang lebih 2 jam dalam setiap migrasi VM. Pada VM 100 mendapatkan rata-rata *downtime* 342 ms dengan waktu migrasi 0.028 ms, pada VM 101 mendapatkan rata-rata *downtime* 382 ms dengan waktu migrasi 0.022 ms dan pada VM 103 mendapatkan rata-rata *downtime* 337 ms dengan waktu migrasi 0.025 ms. Hasil pengukuran bisa berbeda di ketiga VM, sehingga menjadikan *downtime* besar dengan ukuran data transfer menjadi besar dengan waktu migrasi menjadi lama. Ukuran data transfer saat di migrasikan akan semakin besar dengan pengaruh *migration speed* atau *bandwidth*.

## 5. Kesimpulan dan Saran

### 5.1 Kesimpulan

Dari hasil dan analisis pada percobaan yang telah dilakukan dapat diambil beberapa kesimpulan, yaitu:

1. Semakin besar hasil *downtime*, maka *migration time* akan semakin besar.
2. Ukuran data transfer saat dimigrasikan semakin besar sehingga mempengaruhi *migration speed* atau *bandwidth*, contoh pada VM 100 dimigrasikan secara *online* dan sedang mengakses Youtube pada saat proses migrasi, maka durasi atau *migration time* yang di dapat 0.005 ms dengan *migration speed* 2.22 MB/s dan *downtime* 522 ms dengan total ukuran data yang di transfer 34420 bytes.
3. Pengukuran pada saat melakukan migrasi ketika Youtube diakses memiliki *downtime* yang lebih besar dibandingkan dengan pengukuran saat memigrasikan VM yang tidak mengakses Youtube, sehingga menjadikan migrasi VM tanpa mengakses Youtube lebih bagus daripada memigrasikan VM pada saat mengakses Youtube dikarenakan pada saat VM mengakses Youtube maka ukuran *memory* pada VM yang dimigrasikan semakin besar disebabkan oleh adanya aplikasi tambahan yaitu *browser* yang diakses pada saat proses migrasi.



## 5.2 Saran

Saran yang diberikan untuk penelitian lebih lanjut dengan berkaitan topik ini yaitu, *cloud computing* dapat diaplikasikan dengan menggunakan jaringan WAN dan mengukur *downtime* dalam *microsecond*, sehingga menjadikan waktu pekerjaan di VM tidak mengganggu bagi *client* yang terhubung ke mesin virtual dan dapat meningkatkan popularitas sistem *cloud computing* dalam hal migrasi mesin virtual di seluruh pusat data, sehingga sumber daya akan sangat bermanfaat bagi *administrator* pusat data.

## Daftar Pustaka

- [1] Aradea. (2010). Penerapan Service Oriented Architecture Untuk Pembangunan Web Based Learning . Bandung: Aradea.
- [2] Argawal, A. (2012). Live Migration of Virtual Machine in Cloud. Jakarta: Departement of Computer, MIT Collage of Engineering.
- [3] CentOS. [Online]. <https://id.wikipedia.org/wiki/CentOS/>. [Accessed: 24 Desember 2017].
- [4] Cloud Computing. [Online]. <http://www.seagate.com/id/id/tech-insights/cloud-compute-and-cloud-storage-architecture-master-ti/>. [Accessed: 24 Desember 2017].
- [5] Effendi, I. (2016). Jurnal IT. Diambil kembali dari Apa itu Hypervisor: <https://www.it-jurnal.com/apa-itu-hypervisor/>. [Accessed: 19 September 2017].
- [6] Fadil, A. (2016). Pemilihan dan Imigrasi Virtual Machine dengan Menggunakan Multi Criteria Decision Making untuk peningkatan Kinerja. Jakarta: Fadil.
- [7] How to Install nginx with RTMP. [Online] <https://www.howtoforge.com/tutorial/how-to-install-nginx-with-rtmp-module-on-centos/>. [Accessed: 20 Desember 2018].
- [8] Isparmo. (2016). Jumlah Prnduduk Indonesia. Diambil kembali dari Internet Marketing: <http://isparmo.web.id/>. [Accessed: 19 September 2017].
- [9] Kashyap, Sanindhya. (2014). An Enhanced Approach to Live Migration of Virtual Machines.
- [10] Michae, H., & Kartik , G. (2009). Pre Copy Based Live Virtual Machine Migration Using Adiptive Pre Paging Dynamic Self Balloning. Bandung: Michae.
- [11] Musajid, Akrom. 2016. Proxmox Cloud Computing (Virtualisasi). <http://isparmo.web.id/>. [Accessed: 19 September 2017].
- [12] PT. Proweb Indonesia. 2018. KVM. <https://www.linux-kvm.org/>. [Accessed: 19 September 2017].
- [13] Sales. (2016). *Cloud Computing*. Diambil kembali dari Sales Force Tutorial: <http://www.salesforcetutorial.com/introduction-to-cloud-computing/>. [Accessed: 19 September 2017].
- [14] Sharma, S. (2016). A Three Phase Optimization Method for Pre Copy Based VM Live Migration. Bali: Sangeeta.
- [15] Wadia, Y. (2012). *Yoyo Clouds*. Diambil kembali dari Eucalyptus: <http://www.yoyoclouds.com/p/eucalyptus.html/>. [Accessed: 19 September 2017].

