

Implementasi Virtual Private Server Menggunakan Xen Hypervisor

Krisnawan Hartanto¹⁾, Adian Fatchur Rochim²⁾, Kodrat Iman Satoto²⁾

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro,

Jln. Prof. Sudharto, Tembalang, Semarang, Indonesia

ABSTRACT

Using a dedicated server for one operating system that only requires resources that are not too large is a waste. The remaining resources may be greater than the resources used, so that remaining resources will be wasted. On the other hand, a server with one operating system will preclude the use by the different operating system.

The author suggests a method for the problem. The method offered is the use of a virtual machine that is also called virtual private servers (VPS). With virtual machine, a server machine can be equipped with multiple operating systems, so that existing resources can be optimized. VPS bridges the gap between shared web hosting services from dedicated server service, giving customers the freedom from other VPS services in the realm of software, but with a lower expenditure than dedicated servers.

Using server virtualization makes more efficient use of resources. Resources remaining unused by a single operating system can be used by other operating systems. Virtualization also allows backup mechanisms for virtual machine.

Keyword : *virtual, virtualization, virtual machine, server, xen hypervisor, hypervm*

PENDAHULUAN

Latar Belakang

Virtualisasi merupakan strategi untuk mengurangi konsumsi daya pusat data. Dengan virtualisasi, satu host server fisik memiliki banyak server virtual. Virtualisasi memungkinkan pusat data untuk mengkonsolidasikan infrastruktur server fisik dengan menempatkan server virtual pada sejumlah kecil server fisik yang lebih kuat, sehingga menggunakan energy listrik yang sedikit dan menyederhanakan pusat data. Selain mendapatkan penggunaan hardware yang lebih baik, virtualisasi mengurangi penggunaan ruang pusat data, membuat komputasi dengan penggunaan daya yang lebih baik, dan mengurangi kebutuhan energi pada pusat data. Banyak perusahaan menggunakan virtualisasi untuk membatasi konsumsi energi dari pusat data.

Perusahaan TI terkemuka bergabung pada Februari 2007, untuk membentuk sebuah kelompok nirlaba yang disebut Grid Green^[6] yang bertujuan untuk mengatasi masalah konsumsi daya pusat data yang besar. Kelompok ini berusaha untuk menentukan dan menyebarkan praktek hemat energi terbaik dalam operasi pusat data, konstruksi, desain, dan mendorong metrik *user-centric* baru dan standar teknologi.

Penggunaan sebuah piranti *server* terdedikasi kurang efisien apabila hanya digunakan untuk sistem operasi tunggal dengan kebutuhan sumberdaya kecil. Mesin *server* yang telah terinstal dengan sebuah sistem operasi tidak dapat berjalan berdampingan dengan sistem operasi lain secara bersamaan. Hal ini akan sangat merepotkan apabila *server* fisik yang dimiliki hanya satu tetapi membutuhkan *server* dengan jumlah lebih dari satu. Merupakan pemborosan yang sangat besar apabila diharuskan untuk membeli *server* hanya untuk kebutuhan yang tidak terlalu besar. Alangkah baiknya apabila sisa sumber daya yang tidak terpakai tersebut dapat dimanfaatkan untuk kepentingan yang lain.

Tujuan Penelitian

- Tujuan dari penelitian tugas akhir ini adalah untuk menerapkan teknik pembuatan virtual private server dengan menggunakan perangkat lunak Xen hypervisor.
- Menerapkan *backup* mesin virtual sebagai skenario *disaster recovery*.

Batasan Masalah

Dalam penulisan tugas akhir ini pembahasan masalah hanya dibatasi pada permasalahan berikut :

- Penelitian dilakukan untuk mengimplementasikan virtualisasi *server*.
- Penelitian dilakukan dengan menggunakan system operasi centos 5.3 sebagai system operasi *host*.
- Virtualisasi dilakukan dengan menggunakan Xen hypervisor.
- Penelitian tidak membahas modifikasi sistem operasi
- Peneliti menggunakan hypervm sebagai *control panel*.
- Peneliti menggunakan fitur *backup* yang dimiliki oleh HyperVM sebagai metode *backup*.
- Implementasi aplikasi hanya sebatas pada pembuktian bahwa aplikasi dapat berjalan di atas sistem yang dibangun.
- Peneliti tidak membahas perancangan maupun pembuatan aplikasi.

1) Mahasiswa Teknik Elektro UNDIP

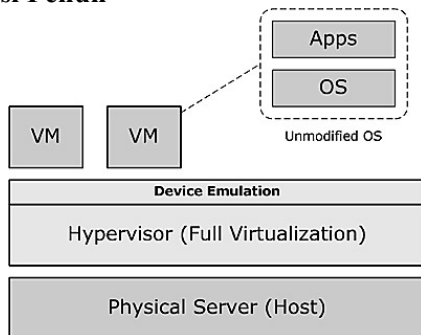
2) Dosen Teknik Elektro UNDIP

LANDASAN TEORI

Pengertian Virtualisasi

Virtualisasi merupakan suatu aplikasi perangkat lunak untuk mensimulasikan sumber daya perangkat keras. Menurut Sundarranjan^[7], virtualisasi adalah sebuah teknik agar perangkat keras pada sebuah mesin dapat dibagi melalui pembagian perangkat keras atau lunak, berbagi waktu dan simulasi menjadi banyak lingkungan eksekusi, tiap bagian dapat berperan sebagai sistem yang lengkap. Sumber daya perangkat keras dibagikan di antara klien-klien yang berpikir bahwa mereka berjalan di atas perangkat keras asli.

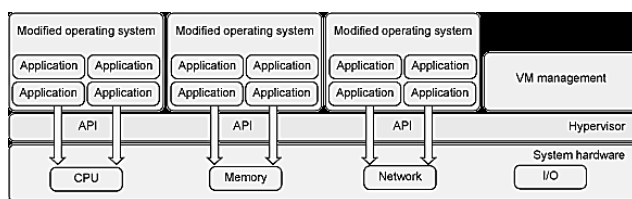
Virtualisasi Penuh



Gambar 1 Teknik virtualisasi penuh (Anonim, 2011)

Mesin virtual mengabstraksi perangkat keras, mengijinkan sebuah sistem operasi taktermodifikasi untuk dapat berjalan^[4]. Sistem operasi yang berjalan pada mesin virtual memberikan instruksi kepada perangkat keras dengan cara melalui mesin virtual. Virtualisasi penuh memberikan pemodelan lengkap dari perangkat keras. Keuntungan utama dari sistem ini menurut Sundarranjan^[7] adalah sistem operasi yang berjalan pada mesin virtual tidak butuh untuk dimodifikasi.

Paravirtualisasi



Gambar 2 Skema teknik paravirtualisasi (Anonim, 2011)

Paravirtualisasi adalah sebuah teknik yang mana kumpulan instruksi dari perangkat keras (yang tidak mendukung virtualisasi) dimodifikasi menjadi sebuah kumpulan instruksi yang dapat divirtualisasi secara penuh. Memodifikasi kumpulan instruksi perangkat keras, berarti sistem operasi juga butuh untuk diarahkan ke kumpulan instruksi baru^[7]. Mesin virtual tidak mensimulasikan sistem operasi secara menyeluruh, tetapi menawarkan *API* yang membutuhkan modifikasi sistem operasi (merupakan teknik yang digunakan oleh XEN)^[4].

Pendekatan paravirtualisasi memodifikasi sistem operasi tamu.

Nilai utama dari paravirtualisasi terletak pada pengeluaran yang rendah, tetapi keuntungan unjuk kerja paravirtualisasi dibanding virtualisasi penuh bergantung pada beban kerja. Kekurangan dari paravirtualisasi terletak pada kemampuan yang tidak mendukung sistem operasi yang tidak dapat dimodifikasi (misal Windows *server*).

Hardware-Assisted Virtualization

Hardware-Assisted Virtualization mengacu pada penambahan perangkat keras pada arsitektur sistem untuk mengurangi banyak pengeluaran *hypervisor* sehubungan dengan memerangkap dan mengemulasi operasi *I/O* dan instruksi status yang dijalankan dalam sebuah sistem operasi tamu. William menyebutkan pada bukunya yang berjudul *Virtualization with Xen*^[10], prosesor *Hardware-assist* memberikan *SO* tamu otoritas yang dibutuhkan untuk mendapatkan akses langsung ke sumberdaya tanpa harus membagi kendali dari perangkat keras. Sebelumnya, *VMM* harus mengemulasi perangkat lunak untuk *SO* tamu saat mengendalikan *platform* fisik. Prosesor baru tersebut memberikan *VMM* dan *SO* tamu otoritas yang dibutuhkan tanpa emulasi perangkat keras dan perubahan *SO*.

Domain-domain pada Xen

Domain kontrol, *dom0*, berfungsi seperti instalasi Linux standar, sehingga dapat dijalankan aplikasi modus pengguna, seperti yang digunakan untuk mengelola lingkungan Xen, serta dipasang driver pengguna yang diperlukan untuk mendukung platform perangkat keras. Karena kemampuan untuk mengkompilasi dan menjalankan hampir semua perangkat keras dengan driver Linux yang tersedia, Xen memiliki beragam dukungan perangkat keras. Hal ini memberikan fleksibilitas yang lebih besar kepada organisasi TI dengan pilihan mereka akan jaringan fisik dan perangkat penyimpanan maupun mengijinkan Xen untuk diterapkan pada hampir setiap lingkungan x86.

API perangkat keras virtual mencakup kontrol antarmuka yang mengelola pemaparan perangkat fisik, baik pembuatan maupun penciptaan, melalui perangkat *I/O* berikut:

1. *Virtual Network Interfaces (VIFs)*
2. *Virtual Firewall and Routers (VFRs)*
3. *Virtual Block Devices (VBDs)*

Setiap perangkat *I/O* virtual memiliki sebuah *Access Control List (ACL)* yang terkait dengannya. Mirip dengan *ACL* untuk file system atau jaringan, *ACL* ini berisi informasi tentang *domU* yang memiliki

akses ke perangkat, serta batasan dan tipe akses yang diijinkan.

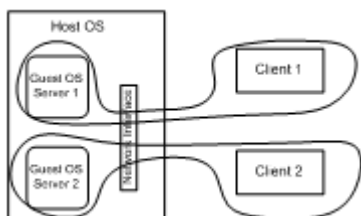
Antarmuka kontrol serta kumpulan statistik yang memprofilkan komponen sistem Xen, diekspor ke seperangkat alat manajemen modus pengguna yang berjalan di alat *dom0*. Alat-alat ini dapat digunakan untuk:

1. Membuat dan menghapus *DomU*
2. Membuat dan menghapus perangkat I/O virtual
3. Memantau aktivitas jaringan
4. Migrasi langsung mesin virtual dari satu host Xen ke yang lain
5. Memantau kinerja pada tingkat *systemwide* atau *per-domain*

PERANCANGAN SISTEM

Perancangan Sistem Virtualisasi

Sistem ini dirancang dengan menggunakan sebuah mesin server yang divirtualisasi sehingga dapat menjalankan beberapa sistem operasi. Virtualisasi dijalankan memakai perangkat lunak paravirtualisasi Xen Hypervisor, yang terinstal di atas sistem operasi centos 5.3. Untuk memudahkan manajemen mesin-mesin virtual, penulis menggunakan hypervm sebagai *control panel*. Perancangan secara umum dapat dilihat pada gambar di bawah ini.



Gambar 3 Skema mesin tervirtualisasi

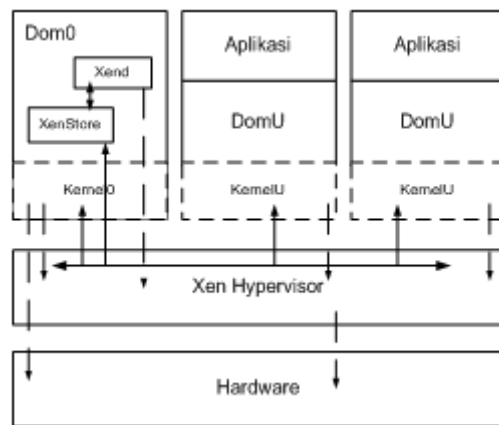
Perancangan Host

Instalasi centos sebagai lapisan sistem operasi dasar harus membuat partisi *LVM (Logical Volume Manager)* karena xen akan memakai partisi *LVM* untuk virtualisasi. Instalasi *host* dalam *GUI* diperlukan apabila guest yang akan diinstal berbasis *GUI*, seperti sistem operasi Windows. Pada penelitian ini, penulis akan menggunakan sistem operasi linux dengan *distro* Centos 5.3. Linux dipilih karena xen hypervisor berjalan pada kernel Linux.

Perancangan Virtualisasi dengan Xen Hypervisor

Xen Hypervisor menerapkan teknologi paravirtualisasi, yang berarti sistem operasi yang akan

diimplementasikan pada xen harus dimodifikasi secara khusus agar dapat berjalan secara baik. Banyak penyedia layanan *web hosting* telah menyediakan sistem operasi yang dimodifikasi sehingga proses modifikasi tidak perlu dilakukan sendiri.



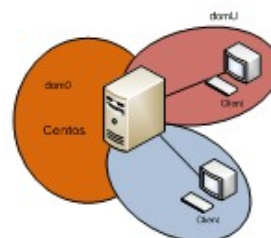
Gambar 4 Implementasi sistem operasi termodifikasi

Desain Antarmuka Manajemen Mesin Virtual

Penggunaan hypervm sebagai *control panel* memudahkan pengguna dalam mengelola mesin-mesin virtual. Hypervm memiliki fitur-fitur untuk membuat mesin virtual secara *graphic user interface (GUI)*, menampilkan penggunaan sumber daya, sehingga seorang admin vps mengetahui secara jelas alokasi penggunaan sumber daya mesin. Hypervm bukan merupakan bagian dari lapisan virtualisasi. Hypervm hanya mempermudah manajemen mesin virtual secara *GUI*.

Akses Client Dalam Satu Jaringan

Pada pengujian akses dalam satu jaringan, *client* melakukan telnet terhadap mesin virtual menggunakan alamat ip yang masih berada pada jaringan yang sama. Dalam hal ini *client* dan *server* sama-sama menggunakan alamat ip *aaa.bbb.ccc.xxx*.

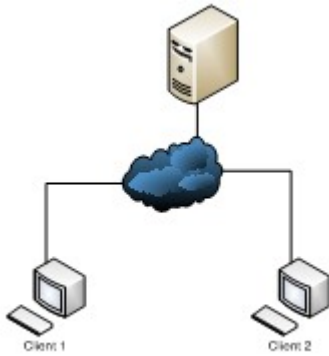


Gambar 5 Pengaksesan mesin virtual dalam satu jaringan

Akses Client Dalam Jaringan Berbeda

Pada pengujian akses dalam jaringan berbeda, *client* melakukan telnet terhadap mesin virtual dari jaringan yang berbeda, dalam hal ini menggunakan alamat ip *ddd.eee.fff.xxx* dan server menggunakan

alamat ip aaa.bbb.ccc.xxx sehingga akses berada pada jaringan yang berbeda.



Gambar 6 Pengaksesan mesin virtual dalam jaringan berbeda

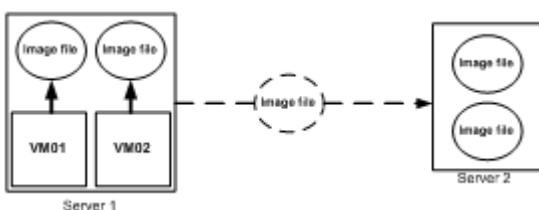
Desain Environment Sistem

Berdasarkan kebutuhan minimal yang diperlukan untuk implementasi host, sistem operasi *guest*, dan layanan aplikasi, maka mesin server membutuhkan memori minimal sebesar 2GB. Kebutuhan memori tersebut disesuaikan dengan jumlah dan kebutuhan memori sistem operasi *guest* yang akan diimplementasikan sebesar 640, keperluan total minimal dari aplikasi, dan kebutuhan memori untuk sistem operasi *host* sebesar 512.

Sistem memerlukan prosesor dengan dukungan *Virtualization Technology (VT)* karena sistem operasi taktermomodifikasi akan diimplementasikan sebagai mesin virtual. Dukungan tersebut dapat diberikan oleh prosesor dengan fitur *VT* seperti yang dimiliki oleh Intel-VT dan AMD-V.

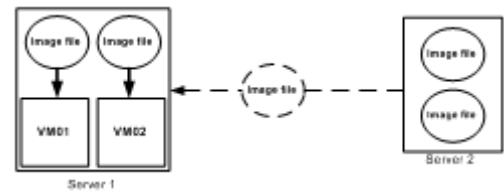
Perancangan Disaster Recovery

Proses *backup* dapat dilihat pada gambar di bawah ini. Skenario *backup* yang penulis lakukan cukup sederhana, hanya meliputi dua langkah. Langkah pertama, keadaan mesin virtual yang ada saat itu ditangkap dalam bentuk *image* dan disimpan pada direktori yang terdapat pada *server 1*. Langkah kedua, *image* mesin virtual yang telah disimpan pada *server 1* diunduh ke *server 2* melalui koneksi jaringan.



Gambar 7 Prosedur Backup

Proses pengembalian *image file* menjadi mesin virtual kembali seperti sedia kala dilakukan dengan cara *image file* yang berada pada *server 2* diunggah kembali ke *server 1* kemudian dilakukan prosedur pemulihan yang akan memulihkan data yang ada di mesin virtual. *Image file* dikembalikan ke *server 1* dengan asumsi *server backup* memiliki lingkungan yang identik dengan *server 1* sehingga apabila *server 1* mengalami kerusakan, *restore image* dapat dilakukan di *server backup* dan mesin virtual dapat berjalan pada *server backup*. Namun dalam penelitian ini, *server 2* tidak identik dengan *server 1*, sehingga *restore image* dilakukan pada *server 1*.



Gambar 8 Prosedur pemulihan

IMPLEMENTASI DAN PENGUJIAN

Implementasi Host

Mesin server yang dipakai harus memiliki minimum memori sebesar 2 Gb dan memiliki prosesor dengan dukungan terhadap virtualisasi *hardware-assisted*. Dapat menggunakan prosesor keluaran intel dengan seri Intel-VT atau dapat pula AMD-V. Pada proses instalasi, partisi yang dibuat harus menyertakan *LVM (Logical Volume Manager)*, hal ini disebabkan sistem operasi tamu yang diinstal akan tersimpan pada *LVM* tersebut.

Implementasi Xen Hypervisor

Sebelum melakukan instalasi, perlu diperhatikan bahwa kita harus login terlebih dahulu sebagai *super user*, atau disebut juga dengan *root*. untuk instalasi Xen hypervisor, pada mode *super user*, diketik perintah `#Yum install kernel-xen xen.`

Implementasi Hypervm

HyperVM merupakan *control panel* untuk melakukan pengelolaan akun vps (*guest / domU*) yang saat ini sangat digemari karena memiliki harga yang tergolong murah. HyperVM dapat digunakan dengan kernel Xen atau openVZ, walaupun masih terus dikembangkan, namun sudah sangat layak digunakan pada lingkungan produksi. Implementasi Hypervm meliputi beberapa tahapan sebagai berikut:

1. Mengunduh *OS template*
2. Mengunduh File Instalasi
3. Menjalankan File Instalasi

Implementasi Mesin Virtual

Implementasi Sistem Operasi Linux

Mesin virtual pada xen berada pada domain dengan hak akses yang terbatas atau domU. Cara implementasi mesin virtual adalah sebagai berikut.

1. Klik *tab* Add Xen
2. Mengisikan *vm name* dan *password*
3. Memilih *os template* untuk sistem operasi yang akan diinstal
4. Menentukan *quota* sumber daya perangkat keras mesin
5. Menentukan hak-hak yang akan dimiliki oleh virtual mesin
6. Klik *add* untuk mejalankan proses instalasi.

Cara untuk melihat mesin virtual yang telah terinstal dapat dilakukan pada *console dom0*.

Implementasi Sistem Operasi Windows

Implementasi sistem operasi windows sebagai *guest OS* memerlukan host yang menggunakan *Graphic User Interface*. Persyaratan utama implementasi sistem operasi windows sebagai *guest os* adalah ketersediaan ruang disk. Ruang disk yang dimaksud dapat berupa disk drive fisik yang melekat pada sistem *host* atau *disk image*. Untuk membuat *disk image* dapat digunakan perintah *dd*. Sebagai contoh, baris perintah di bawah ini membuat 20Gb *disk image* dengan file bernama *win2k3*. `dd if=/dev/zero of=/var/xen/win2k3.img bs=1 count=1 seek=20G.`

Implementasi membutuhkan *CD* windows server dan dapat dilakukan secara langsung melalui *CD/DVD drive* atau *CD image* diekstrak dari *CD* ke sebuah *ISO image* yang disimpan pada *filesystem* dari *host*. Setelah *CD* atau *CD image* siap, langkah selanjutnya membuat *Xen HVM configuration file* yang sesuai.

```
[screen 0: bash] root@localhost:~
File Edit View Terminal Tabs Help
import os, re
arch = os.uname()[4]
if re.search('64',arch):
    arch_libdir = 'lib64'
else:
    arch_libdir = 'lib'

kernel = "/usr/lib/xen/boot/hvmloader"
builder='hvm'
memory = 512
shadow_memory = 8
name = "win2k3"
vcpus=2
pae=1
acpi=1
apic=1

vif = ['type=ioemu, mac=00:16:3e:00:00:51, bridge=xenbr0']
disk = [ 'file:/var/xen/win2k3.img,ioemu:hda,w', 'phy:/dev/cdrom,hdc:cdrom,r']
device_model = '/usr/' + arch_libdir + '/xen/bin/qemu-dm'
boot="dca"
sd1=1
vnc=0
serial='pty'
localtime=1
usb=1
usbdevice='mouse'
keymap='ja'
~
[root@localhost ~]#
```

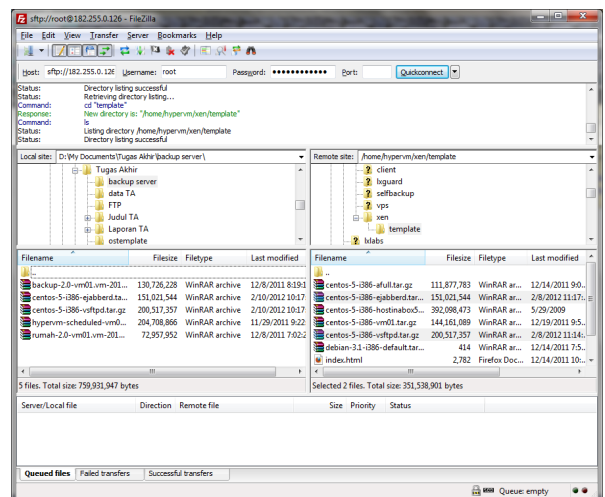
Gambar 9 Berkas konfigurasi *HVM*

Membuat Berkas *Backup* dari Mesin Virtual

Kloning mesin virtual merupakan fitur yang digunakan untuk membekukan dan menyimpan keadaan saat ini dari mesin virtual. Pembuatan klon menggunakan hypervm berbasis *GUI* dan dapat dilakukan dari jaringan luar. Klon yang dibuat akan berekstensi *tar.gz*. Untuk membuat klon dari mesin virtual, masuk ke *tab virtual machine* dan memilih mesin virtual yang akan dibuat klon. Selanjutnya memilih menu *clone vps*.

Mengunduh Berkas *Backup*

Berkas yang akan diunduh tersedia dalam ekstensi *tar.gz* dan terletak dalam folder */home/hypervm/xen/template/*. Berkas *backup* dapat disalin ke mesin lain dengan menggunakan aplikasi *FileZilla*. Berkas yang telah diunduh akan tersimpan di komputer pengakses hypervm.

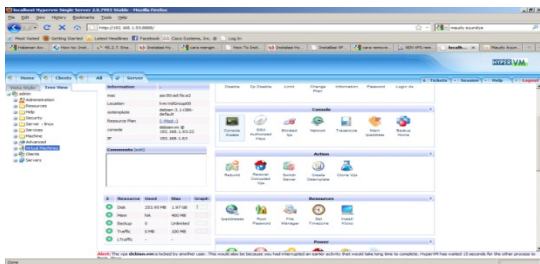


Gambar 10. Pengunduhan berkas backup

Pengujian Sistem

Pengujian Akses Mesin Virtual

Pengaksesan terhadap mesin virtual dapat dilakukan dengan dua peran utama. Pertama, akses dilakukan oleh orang yang berperan sebagai *dom0* dan memiliki hak akses tinggi. *Dom0* dapat mengakses mesin virtual dengan dua cara, melalui hypervm yang berbasis *GUI* dan melalui *ssh client* yang berbasis teks. Akses melalui hypervm dapat dilakukan dengan mengklik *console access* pada tab *console*. Akan tetapi terlebih dahulu pastikan virtual mesin telah aktif.



Gambar 11 Console access pada hypervm

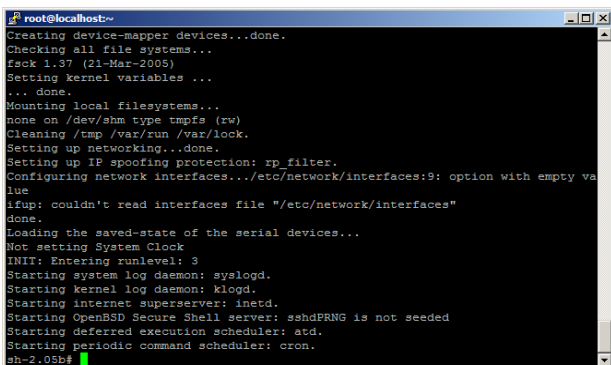
Setelah itu, muncul jendela *console* di mana mesin virtual akan diakses menggunakan *username* dan *password* yang sesuai.



Gambar 12 Akses mesin virtual melalui hypervm

Sedangkan akses mesin virtual dengan *console* mesin server atau *ssh client* bagi pengguna yang berperan sebagai *dom0* dapat dilakukan dengan mengetikkan perintah.

```
xm console vm01.v
```

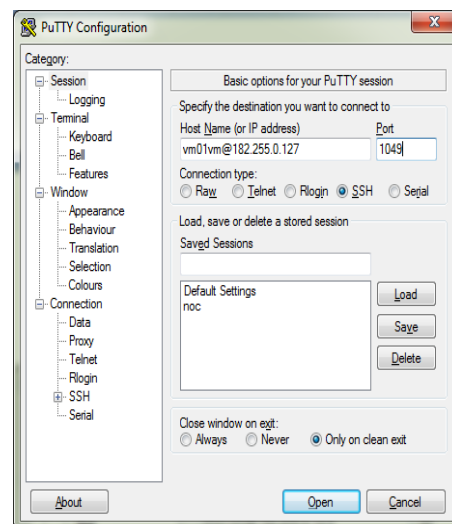


Gambar 13 Akses mesin virtual melalui dom0

Maka *console* akan beralih ke mesin virtual. Untuk mengembalikan ke *dom0* dapat dilakukan

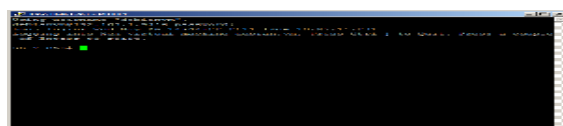
dengan menekan *ctrl+]* untuk akses dari *console* atau *ctrl+5* untuk akses dari *ssh client*.

Cara kedua adalah bagi pengguna yang tidak memiliki akses ke *dom0*. Akses dapat dilakukan menggunakan *ssh client*, semisal *putty* dengan menuliskan *hostname* sebagai berikut, nama-mesin-virtual@alamat-ip-server, sedangkan port harus ditentukan terlebih dahulu oleh *dom0*.



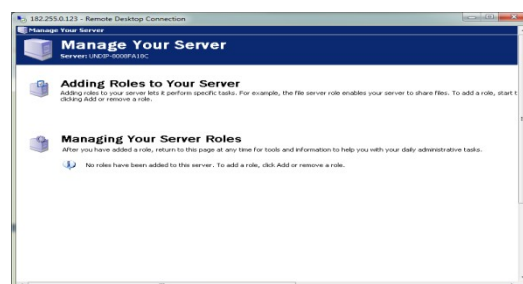
Gambar 14 Inisialisasi sesi pada putty

Apabila proses inisialisasi berhasil pengguna akan dialihkan ke jendela *console* yang menanyakan *password* untuk memastikan pengakses adalah pengguna yang berhak. Pengguna telah dapat mengakses mesin virtual sebagai *domU*.



Gambar 15 Akses mesin virtual sebagai domU

Sedangkan akses untuk sistem operasi berbasis *GUI* (*Graphic User Interface*) dapat dilakukan menggunakan *remote desktop connection*. Namun demikian *remote desktop* harus diaktifkan terlebih dahulu pada sisi *guest OS*.



Gambar 16 Akses Mesin Virtual dengan Remote Desktop



Gambar 21 Menghapus mesin virtual menggunakan hypervm

Cara kedua menggunakan perintah Linux. Perintah `xm destroy` diketikkan untuk mematikan mesin virtual. Perintah tersebut memiliki dampak sama dengan mencabut kabel power. Setelah itu setiap berkas yang terdapat pada direktori `vm01.vm` dihapus seluruhnya.



Gambar 22 Menghapus mesin virtual dengan perintah Linux

Kemudian dilakukan proses untuk mengaktifkan kembali mesin virtual. Proses diawali dengan mengunggah berkas yang berisi *backup* terakhir dari mesin virtual. Berkas tersebut tersedia di folder `/home/hypervm/xen/template/`. Proses yang berlangsung merupakan kebalikan dari proses pengunduhan berkas *backup*. Pemulihan mesin virtual dilakukan dengan cara yang sama dengan proses implementasi awal mesin virtual, tetapi menggunakan berkas *backup* sebagai *OStemplate*.

PENUTUP

Kesimpulan

Dari hasil analisa dan pembahasan dapat disimpulkan bahwa :

1. Virtualisasi merupakan strategi untuk mengurangi konsumsi daya pusat data..
2. Virtualisasi dengan Xen Hypervisor memungkinkan sistem operasi berbeda platform dapat berjalan.
3. Sistem operasi tak termodifikasi dapat berjalan dengan dukungan perangkat keras *VT(Virtualization Technology)*.
4. Berbagai macam aplikasi server dapat diimplementasikan di mesin virtual.
5. Dengan hypervm pengelola dapat membuat vps, menghapus, *booting*, *shuting down*, dan mengontrol alokasi sumberdaya.
6. Berkas *backup* yang dibuat menggunakan HyperVm dapat diunduh ke mesin komputer pengakses HyperVM.

Saran

Adapun saran yang dapat diberikan sehubungan dengan pelaksanaan penelitian ini adalah :

1. Penggunaan virtualisasi dapat dikembangkan menjadi beragam kepentingan, seperti *live migration*, yaitu memindahkan sistem operasi yang sedang berjalan ke server fisik yang lain.
2. Penggunaan yang lain adalah *server consolidation*, atau mengelola banyak mesin server menjadi sebuah pengelolaan tunggal
3. Bagi instansi pendidikan, seperti sekolah dan universitas, virtualisasi dapat menjadi solusi untuk melakukan riset dengan berbagai sistem operasi tanpa harus menyediaka sebuah mesin server untuk sebuah sistem operasi

DAFTAR PUSTAKA

- [1] Barnum P., B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Nuegebauer, I. Pratt, dan A. Warfield, "Xen and the Art of Virtualization", Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles, January 2003
- [2] Benyehuda, M., "The Xen Hypervisor Virtualizing a Machine Near You", 2005, The Xen hypervisor Haifux
- [3] Fiuczynski, M. E., "Virtual Machine Monitors", 2009, Research Scholar, PrincetonUniversity.
- [4] Ray, K., "Server Virtualization and Virtual Machine Operating Systems", 2010 Colorado Springs, Colorado
- [5] Ruest, D., Ruest, N., Virtualization: A Beginner's Guide., McGraw-Hill Companies, Inc., New York, 2009
- [6] San Murugesan, "Harnessing Green IT: Principles and Practices," IEEE IT Professional, January-February 2008, pp 24-33
- [7] Sundarranjan, S., S. Bhattacharya, "Xen and Server Consolidation", 2006, Infosys Whitepaper
- [8] VMware, Inc., "A Performance Comparison of Hypervisors", VMware Performance Study.
- [9] VMware, Inc., "Understanding Full Virtualization, Paravirtualization, and Hardware Assist", VMware whitepaper.

- [10] Williams, D. E., Buytaert, K., Garcia, J. R., Rosen, R. Virtualization with Xen™: Including Xen Enterprise™, XenServer™, and XenExpress™. Syngress Publishing, Inc., Burlington, 2007

BIODATA



KRISNAWAN HARTANTO (L2F 007 044)
Dilahirkan di Yogyakarta 22 tahun yang lalu.
Menempuh Pendidikan sampai sekolah menengah atas di Semarang. Dan semenjak tahun 2007 hingga kini sedang menyelesaikan studi Strata- 1 di Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang, Konsentrasi Informatika dan Komputer.

Menyetujui,
Dosen Pembimbing I

Adian Fatchur Rochim, S.T., M.T.
NIP.19730226 1988021 001

Dosen Pembimbing II

Ir. Kodrat I. Satoto M.T.
NIP. 19631028 1993031 002