

## DESAIN DAN ANALISIS KINERJA VIRTUALISASI SERVER MENGGUNAKAN PROXMOX VIRTUAL ENVIRONMENT

Budi Harijanto<sup>1</sup>, Yuri Ariyanto<sup>2</sup>  
[budi.hijet@gmail.com](mailto:budi.hijet@gmail.com), [yuri.bjn@gmail.com](mailto:yuri.bjn@gmail.com)

<sup>1</sup>Program Studi Manajemen Informatika Jurusan Teknologi Informasi Politeknik Negeri Malang

<sup>2</sup>Program Studi Manajemen Informatika Jurusan Teknologi Informasi Politeknik Negeri Malang

### Abstrak

Teknologi virtualisasi merupakan topik yang mulai ramai diteliti oleh para peneliti. Hal tersebut tidak terlepas dari kemampuan teknologi tersebut dalam memangkas biaya penyediaan infrastruktur dan operasional secara mandiri bagi setiap servis yang akan dilayani. Melalui teknologi ini, sebuah layanan dapat dikonfigurasi tanpa mempengaruhi konfigurasi dari layanan lainnya meskipun dalam satu mesin fisik yang sama. Dengan teknologi ini satu mesin fisik dijadikan sebuah sumber daya bersama yang dapat dibagi dan dipakai oleh beberapa layanan sekaligus. Setiap layanan tersebut ditempatkan dalam sebuah wadah atau kontainer. Setiap kontainer dari mesin virtual memiliki sumber daya masing-masing. Sumber daya tersebut dialokasikan dan diatur oleh hypervisor-nya. Hypervisor diklasifikasikan menjadi 2 jenis yaitu Native/Bare-metal Hypervisor dan Hosted Hypervisor. Proxmox Virtual Environment (ProxmoxVE) merupakan sistem operasi mesin virtual yang dibangun dari sistem operasi Linux Debian dengan kernel RHEL yang telah dimodifikasi. Dalam penelitian ini, akan diteliti performa dari virtualisasi server menggunakan ProxmoxVE. Penelitian ini dimaksudkan untuk mengetahui ketergantungan antar mesin virtual yang berjalan di atas ProxmoxVE. Pengujian dilakukan dengan memberi pembebanan sesuai skenario pengujian yang ada. Dari hasil penelitian dapat disimpulkan bahwa ProxmoxVE dapat menjalankan mesin virtual secara penuh layaknya baremetal hypervisor. Hal ini ditunjukkan dengan tidak adanya saling keterkaitan antar mesin virtual yang diujikan.

**Kata kunci:** virtualisasi, hypervisor, ProxmoxVE.

### Abstract

Virtualization technology is a topic that recently studied by researchers today. It because of the technology's are be able to cut the cost of providing infrastructure and operations independently for each service to be served. Through this technology, a service can be configured without affecting the other configuration of services even in the same physical machine. With this technology, one physical machine are used as a shared resource that can be shared and used by multiple services at once. Each service is placed in a container. Each container of the virtual machine has the resources respectively. These resources are allocated and managed by the hypervisor. Hypervisor are classified into two types, Native / Bare-metal Hypervisor and Hosted Hypervisor. Proxmox Virtual Environment (ProxmoxVE) is a virtual machine operating system that built from Linux Debian operating system with modified RHEL kernel. In this study, the performance of virtualized servers using ProxmoxVE are being examined. This study aimed to determine dependencies between virtual machines running on the top of ProxmoxVE. Load testing scenarios are used in this testbed. The result of this study concluded that ProxmoxVE can run a virtual machine same as a baremetal hypervisor. It's indicated by there is on dependencies between the virtual Machines.

**Keywords:** virtualization, hypervisor, ProxmoxVE

---

## 1 Pendahuluan

Pada masa sekarang ini, teknologi virtualisasi merupakan topik yang mulai ramai diteliti oleh para peneliti. Hal tersebut tidak terlepas dari kemampuan teknologi tersebut dalam

memangkas biaya penyediaan infrastruktur dan operasional secara mandiri bagi setiap servis yang akan dilayani. Melalui teknologi ini, sebuah layanan dapat dikonfigurasi tanpa mempengaruhi konfigurasi dari layanan lainnya meskipun dalam satu mesin fisik yang sama. Dengan teknologi ini satu mesin fisik dijadikan sebuah sumber daya bersama yang dapat dibagi dan dipakai oleh beberapa layanan sekaligus. Masing-masing layanan dapat memiliki sistem operasi sendiri-sendiri. Setiap layanan tersebut ditempatkan dalam sebuah wadah atau kontainer. Karena memiliki sistem operasi sendiri pada masing-masing layanannya, konfigurasi dari masing-masing layanan pun tidak saling mempengaruhi. Selain itu, masing-masing mesin virtual di dalam masing-masing kontainer dapat dimatikan/dihidupkan sesuai dengan kebutuhan (perawatan misalnya) tanpa harus mengganggu layanan yang lain. Sehingga dapat dikatakan ketersediaan sebuah layanan dapat lebih terjamin meskipun ada beberapa layanan yang sedang mengalami masalah atau perbaikan.

Setiap kontainer dari mesin virtual memiliki sumber daya masing-masing. Sumber daya tersebut dialokasikan dan diatur oleh *hypervisor*-nya. *Hypervisor* adalah sebuah *firmware* yang bertugas membuat, mengatur, menjalankan, dan memonitor sebuah mesin virtual. *Hypervisor* diklasifikasikan menjadi 2 jenis tergantung tempat di mana dia berdiri. Yang pertama, *Native/Bare-metal Hypervisor*. *Hypervisor* ini dapat mengakses langsung perangkat keras pada suatu mesin fisik tanpa melalui sebuah sistem operasi. Contoh dari *hypervisor* jenis ini adalah *VMware ESX/ESXi*, *Microsoft Hyper-V 2008/2012* dan lain sebagainya. Yang kedua, *hosted hypervisor*. *Hypervisor* ini hanya dapat mengakses perangkat keras pada suatu mesin fisik melalui sebuah sistem operasi di bawahnya. Contoh dari *hypervisor* jenis ini adalah *VMware Workstation*, *VMware Player*, *VMware Fusion*, *Microsoft Virtual PC*, *Oracle VirtualBox* dan lain sebagainya. Perbedaan tempat sebuah *hypervisor* itu berjalan merupakan faktor yang cukup menentukan performa dari sebuah mesin virtual yang dijalankan. Performa dari sebuah mesin virtual yang terdapat dalam *baremetal* terbukti lebih baik dari pada *hosted hypervisor*. Hal ini dikarenakan mesin virtual tersebut memiliki akses langsung terhadap perangkat keras dari sebuah mesin fisik. Mesin virtual tersebut dapat menggunakan sumber daya lebih bebas karena tidak terpotong oleh sebuah sistem operasi yang berjalan di bawahnya.

Selain tempat dari sebuah *hypervisor* itu berada, faktor lain yang cukup menentukan adalah dukungan dari *hardware* dari mesin fisik itu sendiri. Agar dapat mendukung virtualisasi secara penuh, sebuah *processor* dari mesin fisik haruslah memiliki dukungan terhadap virtualisasi. Pada *processor* buatan *Intel*, dukungan ini ditandai oleh adanya fitur *Intel Virtualization Technology (VT-x)* pada tabel spesifikasi *processor*-nya. Sedangkan pada *processor* buatan *AMD* dukungan ini ditandai oleh adanya fitur *AMD-Virtualization* pada tabel spesifikasi *processor*-nya. Dan untuk mendukung akses I/O secara penuh, fitur *Intel Virtualization Technology for direct I/O (VT-d)* pun ditambahkan pada *processor* buatan intel. Berbeda dengan *processor* buatan Intel, *processor* buatan *AMD* mendukung akses I/O secara penuh melalui satu fitur lengkap yang telah dirangkum dalam fitur *AMD-Virtualization* yang dimilikinya. Selain *processor* yang harus mendukung teknologi virtualisasi, *chipset* dari *motherboard* yang dipakai pun menentukan dukungan terhadap teknologi tersebut. Hal ini biasa terjadi pada *motherboard* yang akan dipasang *processor* buatan *Intel*. Sedangkan untuk *chipset* dari *motherboard* yang akan dipasang *processor* buatan *AMD*, sudah semua serinya mendukung teknologi virtualisasi.

*Proxmox Virtual Environment (ProxmoxVE)* merupakan sistem operasi mesin virtual yang mulai banyak dipakai oleh para pengguna teknologi virtualisasi. Hal ini dikarenakan

sistem operasi tersebut bersifat *Open source* sehingga lebih mudah untuk dimodifikasi dan dikembangkan. *ProxmoxVE* merupakan sistem operasi turunan Linux Debian dengan *kernel* RHEL yang telah dimodifikasi agar dapat membuat, menjalankan, dan mengatur mesin virtual. *ProxmoxVE* mendukung dua tipe virtualisasi yaitu virtualisasi yang berbasis kontainer *OpenVZ* dan virtualisasi penuh dengan *KVM*.

Dalam penelitian ini, akan diteliti performa dari virtualisasi *server* menggunakan *ProxmoxVE*. Penelitian ini dimaksudkan untuk mengetahui ketergantungan antar mesin virtual yang berjalan di atas *ProxmoxVE*. Penelitian ini perlu dilakukan untuk membuktikan isu-isu yang mengatakan bahwa *ProxmoxVE* dapat menjalankan mesin virtual secara penuh layaknya *baremetal hypervisor* meskipun *ProxmoxVE* bukan merupakan *baremetal hypervisor*. Dalam penelitian ini empat buah mesin virtual akan dijalankan dan akan diberikan beban secara manual untuk mensimulasikan beban yang ditanggung oleh masing-masing mesin virtual tersebut. Pembebanan akan dilakukan dengan menggunakan aplikasi yang sudah tersedia pada *distro linux* yaitu *stress* dan *cpulimit*. Setelah diberi beban, masing-masing mesin virtual akan diukur tingkat bebannya menggunakan sebuah aplikasi *monitoring* buatan sendiri (berbasis *Python*) yang dibantu dengan *library psutil*. Hasil dari pengukuran beban dari masing-masing mesin virtual yang telah dibebani akan dibandingkan dan disimpulkan ada tidaknya ketergantungan antar mesin virtual yang berjalan. Kesimpulan tersebut akan dijadikan saran atau rekomendasi dalam penggunaan *ProxmoxVE* sebagai sistem operasi yang akan menangani virtualisasi.

## 2 Kajian Teori

### 2.1 Teknologi Virtualisasi

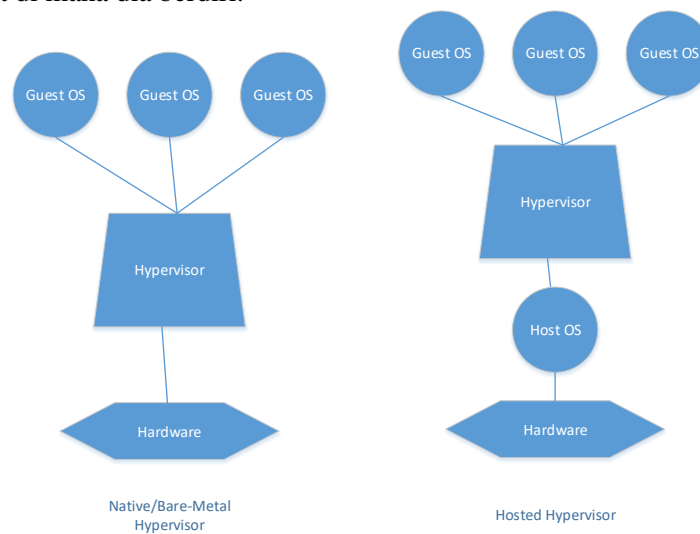
Teknologi virtualisasi merupakan sebuah teknologi yang memungkinkan sebuah mesin fisik dijadikan sebuah sumber daya bersama yang dapat dibagi dan dipakai oleh beberapa layanan sekaligus. Layanan-layanan tersebut dapat dikonfigurasi sendiri tanpa mempengaruhi konfigurasi dari layanan lainnya meskipun dalam satu mesin fisik yang sama. Masing-masing layanan dapat memiliki sistem operasi sendiri-sendiri. Setiap layanan tersebut ditempatkan dalam sebuah wadah atau kontainer. Karena memiliki sistem operasi sendiri pada masing-masing layanannya, konfigurasi dari masing-masing layanan pun tidak saling mempengaruhi. Selain itu, masing-masing mesin virtual di dalam masing-masing kontainer dapat dimatikan/dihidupkan sesuai dengan kebutuhan (pada masa perawatan misalnya) tanpa harus mengganggu layanan yang lain. Sehingga dapat dikatakan ketersediaan sebuah layanan dapat lebih terjamin meskipun ada beberapa layanan yang sedang mengalami masalah atau perbaikan. Dengan menerapkan teknologi ini, biaya penyediaan infrastruktur dan operasional secara mandiri bagi setiap servis yang akan dilayani dapat dipangkas.

Setiap kontainer dari mesin virtual memiliki sumber daya masing-masing. Sumber daya tersebut dialokasikan dan diatur oleh *hypervisor*-nya. Salah satu faktor yang cukup menentukan sebuah virtualisasi adalah dukungan dari *hardware* dari mesin fisik itu sendiri. Agar dapat mendukung virtualisasi secara penuh, sebuah *processor* dari mesin fisik haruslah memiliki dukungan terhadap virtualisasi. Pada *processor* buatan *Intel* [1], dukungan ini ditandai oleh adanya fitur *Intel Virtualization Technology (VT-x)* pada tabel spesifikasi *processor*-nya. Sedangkan pada *processor* buatan *AMD* [2] dukungan ini ditandai oleh adanya fitur *AMD-Virtualization* pada tabel spesifikasi *processor*-nya. Dan untuk mendukung akses I/O secara penuh, fitur *Intel Virtualization Technology for direct I/O (VT-d)* pun ditambahkan pada *processor* buatan *intel* [1]. Berbeda dengan *processor* buatan *Intel*, *processor* buatan *AMD* [2]

mendukung akses I/O secara penuh melalui satu fitur lengkap yang telah dirangkum dalam fitur *AMD-Virtualization* yang dimilikinya. Selain *processor* yang harus mendukung teknologi virtualisasi, *chipset* dari *motherboard* yang dipakai pun menentukan dukungan terhadap teknologi tersebut. Hal ini biasa terjadi pada *motherboard* yang akan dipasang *processor* buatan *Intel*. Sedangkan untuk *chipset* dari *motherboard* yang akan dipasang *processor* buatan *AMD*, sudah semua serinya mendukung teknologi virtualisasi.

## 2.2 Hypervisor

*Hypervisor* [3] adalah sebuah *firmware* yang bertugas membuat, mengatur, menjalankan, dan memonitor sebuah mesin virtual. *Hypervisor* diklasifikasikan menjadi 2 jenis tergantung tempat di mana dia berdiri.



**Gambar 1** Jenis-Jenis *Hypervisor*

Yang pertama, *Native/Bare-metal Hypervisor*. *Hypervisor* ini dapat mengakses langsung perangkat keras pada suatu mesin fisik tanpa melalui sebuah sistem operasi. Contoh dari *hypervisor* jenis ini adalah *VMware ESX/ESXi*, *Microsoft Hyper-V 2008/2012* dan lain sebagainya. Yang kedua, *hosted hypervisor*. *Hypervisor* ini hanya dapat mengakses perangkat keras pada suatu mesin fisik melalui sebuah sistem operasi di bawahnya. Contoh dari *hypervisor* jenis ini adalah *VMware Workstation*, *VMware Player*, *VMware Fusion*, *Microsoft Virtual PC*, *Oracle VirtualBox* dan lain sebagainya. Perbedaan tempat sebuah *hypervisor* itu berjalan merupakan faktor yang cukup menentukan performa dari sebuah mesin virtual yang dijalankan. Performa dari sebuah mesin virtual yang terdapat dalam *baremetal* terbukti lebih baik dari pada *hosted hypervisor*. Hal ini dikarenakan mesin virtual tersebut memiliki akses langsung terhadap perangkat keras dari sebuah mesin fisik. Mesin virtual tersebut dapat menggunakan sumber daya lebih bebas karena tidak terpotong oleh sebuah sistem operasi yang berjalan di bawahnya.

## 2.3 Proxmox

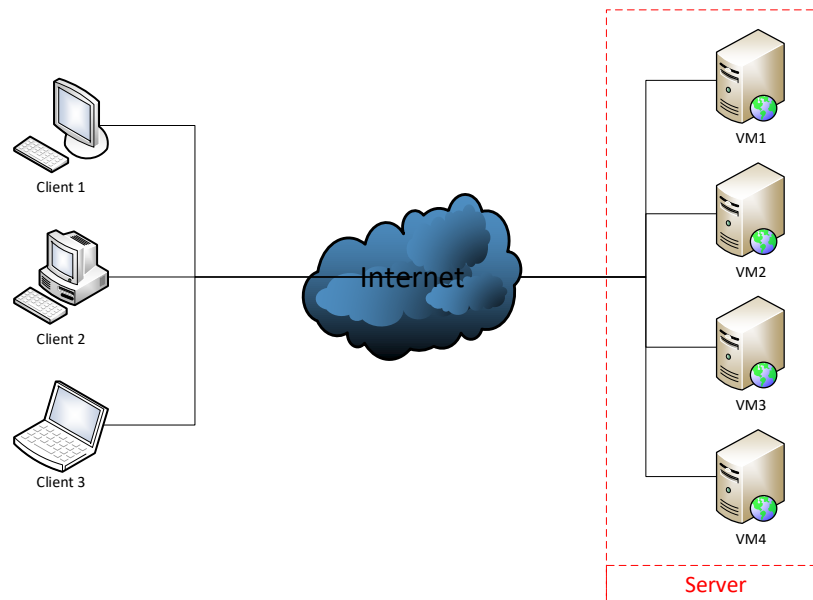
*Proxmox Virtual Environment (ProxmoxVE)* [4] merupakan sistem operasi mesin virtual yang mulai banyak dipakai oleh para pengguna teknologi virtualisasi. Hal ini dikarenakan sistem operasi tersebut bersifat *Open source* sehingga lebih mudah untuk dimodifikasi dan

dikembangkan. *ProxmoxVE* merupakan sistem operasi turunan Linux Debian dengan *kernel* RHEL yang telah dimodifikasi agar dapat membuat, menjalankan, dan mengatur mesin virtual. *ProxmoxVE* mendukung dua tipe virtualisasi yaitu virtualisasi yang berbasis kontainer *OpenVZ* dan virtualisasi penuh dengan *KVM*. *ProxmoxVE* dilengkapi dengan antarmuka pengaturan berbasis web, sehingga memudahkan untuk mengontrol kapan saja dan dari platform apapun. *ProxmoxVE* juga dilengkapi dengan alat bantu *command line* dan *REST API* untuk alat bantu pihak ketiga. Fitur-fitur yang ditawarkan oleh *ProxmoxVE* antara lain *High Availability Cluster*, *Live Migration*, *bridged networking*, *flexible storage*, *OS template building*, *scheduled backup*, dan *command line tools*.

### 3 Uji Coba

#### 3.1 Lingkungan Pengujian/Testbed

Lingkungan pengujian/*testbed* yang akan digunakan dalam penelitian ini dapat digambarkan pada Gambar 2.



**Gambar 2** Topologi Jaringan *Testbed*

Setiap *server* dalam lingkungan pengujian ini merupakan *server* virtual yang berjalan di atas sistem operasi virtualisasi *ProxmoxVE* versi 3.4. Setiap *server* dalam lingkungan pengujian ini memiliki 1 inti *processor* virtual, 1GB *memory*, 30 GB *hardisk*, dan menjalankan sistem operasi Debian 7 64-bit. *Server* virtual tersebut berjalan di atas *hardware* fisik sebuah komputer yang memiliki 6 inti *processor* dengan kecepatan 3,9GHz (AMD FX-6350), 8GB *memory*, dan 2TB *harddisk*. Untuk memaksimalkan akses sumber daya dari *hardware* fisiknya maka fitur virtualisasi *hardware* KVM akan di aktifkan pada setiap mesin virtual yang ada.

### 3.2 Skenario Pengujian

Pengujian dilakukan untuk melihat seberapa handal *ProxmoxVE* dalam menangani pembagian sumber daya antar mesin virtual yang dilayani. Masing-masing *server* dalam lingkungan pengujian ini diberi beban secara manual untuk mensimulasikan jumlah sumber daya yang terpakai di pada masing-masing komputer tersebut. Adapun beban dari masing-masing *server* ditunjukkan pada Tabel 1.

**Tabel 1** Simulasi Beban Lingkungan Pengujian

<b>Skenario</b>	<b>Komputer</b>	<b>Beban CPU (%)</b>	<b>RAM Terpakai (%)</b>
Skenario 1	Server 1	10	30
	Server 2	10	30
	Server 3	10	30
	Server 4	10	30
Skenario 2	Server 1	90	80
	Server 2	10	30
	Server 3	10	30
	Server 4	10	30
<b>Skenario</b>	<b>Komputer</b>	<b>Beban CPU (%)</b>	<b>RAM Terpakai (%)</b>
Skenario 3	Server 1	90	80
	Server 2	90	80
	Server 3	10	30
	Server 4	10	30
Skenario 4	Server 1	90	80
	Server 2	90	80
	Server 3	90	80
	Server 4	10	30
Skenario 5	Server 1	90	80
	Server 2	90	80
	Server 3	90	80
	Server 4	90	80
Skenario 6	Server 1	50	55
	Server 2	50	55
	Server 3	50	55
	Server 4	50	55
Skenario 7	Server 1	90	80
	Server 2	50	55

	Server 3	50	55
	Server 4	50	55
Skenario 8	Server 1	90	80
	Server 2	90	80
	Server 3	50	55
	Server 4	50	55
Skenario 9	Server 1	90	80
	Server 2	90	80
	Server 3	90	80
	Server 4	50	55
Skenario 10	Server 1	50	55
	Server 2	10	30
	Server 3	10	30
	Server 4	10	30
Skenario 11	Server 1	50	55
	Server 2	50	55
	Server 3	10	30
	Server 4	10	30

<b>Skenario</b>	<b>Komputer</b>	<b>Beban CPU (%)</b>	<b>RAM Terpakai (%)</b>
Skenario 12	Server 1	50	55
	Server 2	50	55
	Server 3	50	55
	Server 4	10	30
Skenario 13	Server 1	90	80
	Server 2	50	55
	Server 3	10	30
	Server 4	10	30
Skenario 14	Server 1	90	80
	Server 2	90	80
	Server 3	50	55
	Server 4	10	30
Skenario 15	Server 1	90	80
	Server 2	50	55

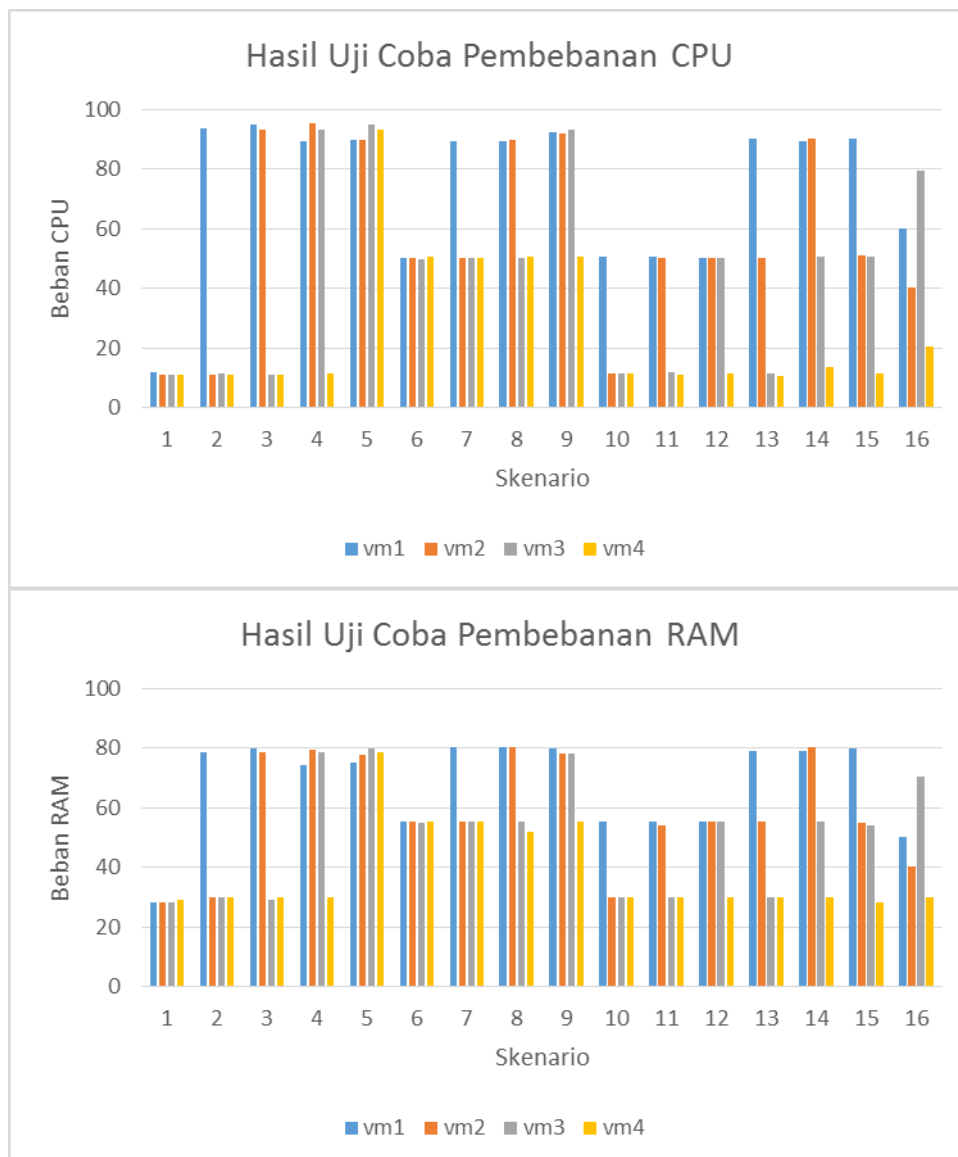
	Server 3	50	55
	Server 4	10	30
Skenario 16	Server 1	60	50
	Server 2	40	40
	Server 3	80	70
	Server 4	20	30

Dari pengujian sesuai skenario di atas, akan diukur beban CPU dari masing-masing mesin virtual pada setiap skenarionya. Setelah itu akan dianalisa ketergantungan antar mesin virtual ketika sebuah beban yang lebih besar diberikan pada salah satu/beberapa mesin virtual yang ada.

#### **4 Hasil Uji Coba**

Hasil uji coba disajikan untuk mengetahui adakah ketergantungan antar mesin virtual ketika sebuah beban yang lebih besar diberikan pada salah satu/beberapa mesin virtual yang ada. Gambar 3 merupakan grafik hasil pengujian pada setiap skenario yang diujikan.





**Gambar 3** Hasil Pengukuran Beban Pada Setiap Skenario

Dari grafik pada Gambar 3 terlihat bahwa pada semua skenario yang diujikan, tidak terdapat ketergantungan antar mesin virtual yang ada. Meskipun pada salah satu/beberapa mesin virtual memiliki beban yang tinggi (>50%) seperti yang ditunjukkan pada grafik hasil uji coba skenario ke 1-3, mesin virtual yang lain tetap memiliki beban sesuai dengan beban asli yang ditanggungnya saja.

## 5 Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan, tidak terdapat ketergantungan antar mesin virtual yang ada. Meskipun pada salah satu/beberapa mesin virtual memiliki beban yang tinggi (>50%) seperti yang ditunjukkan pada grafik hasil uji coba skenario ke 1-3, mesin virtual

yang lain tetap memiliki beban sesuai dengan beban asli yang ditanggungnya saja. Sehingga dapat disimpulkan bahwa *ProxmoxVE* versi 3.4 merupakan sistem operasi yang cukup handal dalam menangani virtualisasi *server*.

## 6 Cuplikan dan Daftar Pustaka

- [1] Intel® Virtualization Technology (Intel® VT), Intel, <http://www.intel.com/content/www/us/en/virtualization/virtualization-technology/intel-virtualization-technology.html>, (13 Agustus 2015)
- [2] AMD Virtualization, AMD, <http://www.amd.com/en-us/solutions/servers/virtualization>, (13 Agustus 2015)
- [3] Strobl, Marius (2013). “*Virtualization for Reliable Embedded Systems*”. Munich: GRIN Publishing GmbH. pp. 5–6. ISBN 978-3-656-49071-5.
- [4] Simon M.C. Cheng (31 October 2014). “*Proxmox High Availability*”. Packt Publishing Ltd. pp. 41–. ISBN 978-1-78398-089-5.