

# RANCANG BANGUN LABORATORIUM VIRTUAL BERBASIS CLOUD COMPUTING DI STMIK STIKOM INDONESIA

I Nyoman Buda Hartawan<sup>1</sup> dan I Kadek Susila Satwika<sup>2</sup>

<sup>1</sup> STMIK STIKOM Indonesia  
Denpasar, Indonesia  
buda.hartawan@gmail.com

<sup>2</sup> STMIK STIKOM Indonesia  
Denpasar, Indonesia  
susila.satwika@gmail.com

## **Abstract:**

Praktikum merupakan kegiatan pembelajaran yang dilaksanakan di STMIK STIKOM Indonesia dalam ruang laboratorium. Kendala yang dihadapi dalam pelaksanaan praktikum khususnya Praktikum Manajemen Jaringan dan Server adalah keterbatasan waktu penggunaan laboratorium. Selain itu penggunaan laboratorium secara bersama-sama menyebabkan adanya perubahan konfigurasi yang sudah dilakukan oleh praktikan, sehingga harus mengulang kembali pada pertemuan selanjutnya. Pada penelitian ini dirancang laboratorium virtual menggunakan proxmox yang memungkinkan praktikan memiliki mesin server secara virtual dan dapat diakses tidak hanya dari laboratorium. Setelah implementasi selanjutnya dilakukan pengujian dengan berpedoman pada modul Praktikum Manajemen Jaringan dan Server. Hasil pengujian menunjukkan saat dilakukan instalasi aplikasi, penggunaan CPU dan memory pada masing-masing mesin virtual yang melakukan aktifitas, meningkat hingga 98%. Penggunaan CPU dan memory pada server utama mencapai 20%. Dengan jumlah total memory sebesar 16GB, server hanya mampu mengaktifkan mesin sebanyak 10 buah secara bersamaan.

**Keywords:** *Laboratorium Virtual, cloud computing, proxmox, manajemen jaringan dan server*

## **1. Pendahuluan**

STMIK STIKOM Indonesia adalah sebuah Sekolah Tinggi Manajemen Teknik Informatika yang berlokasi di Denpasar Bali. Sebagai sebuah lembaga pendidikan di bidang teknologi informasi, STMIK STIKOM Indonesia tidak hanya membekali mahasiswa secara teori, namun juga skill yang diperoleh melalui praktikum.

Praktikum Manajemen Jaringan dan *Server* merupakan salah satu mata kuliah yang ditawarkan oleh Program Studi Sistem Komputer. Mata kuliah ini membahas cara merancang, membangun, dan mengelola sebuah *server*. Pelaksanaan praktikum selama ini dilakukan dimana laboratorium dengan

memanfaatkan aplikasi *virtual* mesin yang terpasang disetiap komputer. Hal ini tidak berjalan maksimal karena *server* yang sebelumnya sudah dikonfigurasi oleh mahasiswa, pada praktikum berikutnya konfigurasi tidak bisa dilanjutkan karena *server virtual* yang sudah dibuat tidak dapat dijalankan kembali atau bahkan ada yang terhapus. Hal ini disebabkan karena *server* yang dirancang tidak dilengkapi sistem keamanan, sehingga siapa saja dapat melakukan modifikasi secara penuh. Hal ini mengingat laboratorium digunakan secara bersama-sama. Mahasiswa harus melakukan instalasi *server* baru untuk dapat melanjutkan konfigurasi sesuai dengan modul yang sudah tersedia. Hal ini menyebabkan lambatnya proses pembelajaran karena harus menunggu untuk melakukan instalasi sistem operasi *server* terlebih dahulu. Pada penelitian ini diusulkan adanya laboratorium *virtual* yang dapat mengatasi permasalahan tersebut.

Pada penelitian ini akan dibangun sebuah laboratorium *virtual* dengan memanfaatkan sebuah *server* yang didalamnya akan dibangun beberapa *server virtual*. Laboratorium yang akan dibangun dapat diakses secara lokal di lingkungan STMIK STIKOM Indonesia. Hal ini mengingat setiap ruangan di STMIK STIKOM Indonesia sudah terhubung jaringan lokal maupun internet. Mahasiswa dapat mengakses laboratorium *virtual* dari ruangan ataupun dari luar ruangan seperti ruang berkumpul mahasiswa ataupun kantin yang terhubung jaringan LAN (Local Area Network) secara kabel maupun wireless. Mahasiswa melakukan akses ke *server* secara remote. Komputer yang digunakan oleh mahasiswa untuk mengakses laboratorium *virtual* tidak perlu terhubung internet, karena laboratorium *virtual* sudah terhubung internet secara langsung.

Sehingga mahasiswa dapat melakukan instalasi aplikasi *server* melalui internet. Dosen dapat memonitor kegiatan mahasiswa selama proses konfigurasi, serta mengawasi penggunaan sumber daya *server*.

## 2. State of The Art

Sebelumnya pernah dilakukan penelitian oleh Stephen D. Burd dkk mengenai perbandingan antara lab komputer *virtual* dengan metode komputasi cloud dan lab komputer fisik. Penelitian didasarkan pada survey terhadap mahasiswa yang menggunakan laboratorium fisik dan *virtual*. Hasil menunjukkan bahwa laboratorium *virtual* lebih baik dalam hal ketersediaan keamanan fisik.[1]

Ahmad Hamim Thohari, dkk melakukan penelitian mengenai rancang bangun laboratorium *virtual*. Dalam penelitiannya laboratorium *virtual* digunakan untuk mata kuliah Pemrograman Berbasis Objek. Aplikasi *virtual* yang digunakan adalah proxmox. Pengujian dilakukan dengan memberi beban pada *server*. Beban menjalankan komputer *virtual* untuk startup, membuka aplikasi dan compile secara bersamaan. Hasil penelitian menunjukkan bahwa laboratorium *virtual* dapat berjalan dengan baik ketika terdapat 15 komputer. *Virtual* aktif secara bersamaan.[2]

Sumarto, dkk melakukan penelitian rancang bangun lab komputer *virtual* berbasis cloud computing menggunakan platform Ovirt pada jaringan terpusat. Hasil pengujian terhadap lab komputer *virtual* dengan studi kasus mata kuliah Pemrograman Berbasis Objek menunjukkan, *server* yang digunakan selama pengujian dapat menjalankan lab komputer *virtual* dengan 10 komputer dengan baik.[3]

Dari penelitian yang telah ada menunjukkan bahwa laboratorium *virtual* merupakan topik yang menarik untuk diangkat karena dianggap sebagai solusi untuk mengatasi kekurangan laboratorium fisik. Mahasiswa mampu melakukan praktikum seperti layaknya pada laboratorium fisik. Pada penelitian ini dilakukan rancang bangun laboratorium *virtual* untuk praktikum manajemen jaringan dan *server*. Pada sebuah *server* akan dibuat beberapa *server virtual* yang nantinya dapat diakses oleh mahasiswa untuk melakukan konfigurasi *server*. Pengujian dilakukan terhadap aktivitas CPU (Central Processing Unit), *memory*, dan akses jaringan.

## 3. Cloud Computing

*Cloud computing* dapat didefinisikan sebagai kemudahan berbagi dan penggunaan aplikasi serta sumber daya dalam jaringan untuk pekerjaan yang dilakukan tanpa kekhawatiran tentang kepemilikan dan pengelolaan sumber daya jaringan dan aplikasi. Pada *cloud computing* data tidak disimpan pada computer pribadi, tetapi pada host lain untuk dapat diakses kapan saja dan dimana saja.[4] *Cloud computing* menjadi teknologi yang dapat diadaptasi oleh banyak organisasi dengan kemudahan skalabilitas dan penggunaan sumber daya virtual melalui internet.[5]

### 3.1 Karakteristik *Cloud Computing* [6]

Adapun karakteristik *cloud computing* adalah sebagai berikut.

- a. *On-demand self-service*. Pengguna dapat memesan layanan, dan sumberdaya server tanpa adanya interaksi dengan penyedia layanan.
- b. *Broad network access*. Kemampuan ketersediaan melalui jaringan dan dapat diakses dengan mekanisme standar oleh berbagai platform (seperti telepon seluler, tablet, laptop, dan PDA).
- c. *Resource pooling*. Penyedia sumberdaya komputasi dikumpulkan untuk melayani beberapa konsumen menggunakan model multi-tenant, dengan sumber daya fisik dan virtual yang berbeda yang ditetapkan secara dinamis dan dipindahkan sesuai dengan permintaan konsumen. Contoh sumber daya termasuk penyimpanan, pemrosesan, memori, dan *bandwidth* jaringan.
- d. *Rapid elasticity*. Kemampuan dapat ditingkatkan ataupun diturunkan secara elastis oleh konsumen.
- e. *Measured service*. Sistem *cloud* dapat melakukan kontrol secara otomatis dan mengoptimalkan penggunaan sumberdaya dengan memanfaatkan kemampuan pengukuran pada beberapa *level* yang sesuai dengan jenis layanan (seperti penyimpanan, pemrosesan, *bandwidth*, dan akun pengguna aktif)

### 3.2 Layanan *cloud computing* [6].

Adapun layanan *cloud computing* adalah sebagai berikut.

- a. *Cloud Software as a Service (SaaS)*. Konsumen tidak mengelola atau melakukan

kontrol terhadap infrastruktur *cloud* termasuk jaringan, server, sistem operasi, penyimpanan, atau bahkan kemampuan aplikasi individu, dengan kemungkinan pengecualian terbatas terhadap pengaturan konfigurasi aplikasi pengguna tertentu.

- b. *Cloud Platform as a Service* (PaaS). Konsumen tidak mengelola atau melakukan kontrol terhadap infrastruktur *cloud* termasuk jaringan, server, sistem operasi, atau penyimpanan, namun memiliki kontrol atas aplikasi dan memungkinkan melakukan konfigurasi *hosting*.
- c. *Cloud Infrastructure as a Service* (IaaS). Konsumen tidak mengelola atau melakukan kontrol terhadap infrastruktur *cloud* tetapi memiliki kontrol atas sistem operasi, penyimpanan, aplikasi yang disebarkan, dan mungkin kontrol terbatas terhadap komponen jaringan yang dipilih (misalnya, *firewall host*)

#### 4. Virtualisasi

*Virtualization* / Virtualisasi merupakan sebuah teknik yang berfungsi untuk menciptakan / membuat sesuatu dalam bentuk maya dari sesuatu yang berbentuk nyata. Virtualisasi dapat membuat sesuatu yang ada menjadi berlipat ganda, seperti contohnya dalam Virtualisasi *user* dapat membuat sebuah computer di dalam sebuah computer, jadi dalam 1 komputer dapat dibuat lebih dari 1 sistem computer dengan spesifikasi yang berbeda-beda. [1]

*Virtual Machine* (VM) adalah sebuah mesin yang mempunyai dasar logika yang menggunakan pendekatan lapisan-lapisan (layers) dari sistem komputer. Sehingga sistem komputer dengan tersendiri dibangun atas lapisan-lapisan tersebut, dengan urutan lapisannya mulai dari lapisan terendah sampai lapisan teratas adalah sebagai berikut:

- Perangkat keras (semua bagian fisik komputer)
- Kernel (program untuk mengontrol disk dan sistem file, multi-tasking, load-balancing, *networking* dan security)
- Sistem program (program yang membantu general *user*)

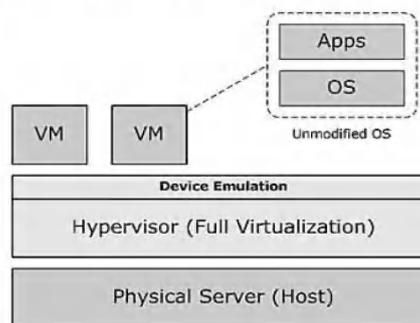
Kernel yang berada pada lapisan kedua ini, menggunakan instruksi perangkat keras untuk menciptakan seperangkat system call yang dapat digunakan oleh komponen-komponen pada level sistem program. Sistem program kemudian dapat menggunakan system call dan perangkat keras

lainnya seolah-olah pada level yang sama. Meskipun sistem program berada di level tertinggi, namun program aplikasi bisa melihat segala sesuatu pada tingkatan dibawahnya seakan-akan mereka adalah bagian dari mesin. Pendekatan dengan lapisan-lapisan inilah yang kemudian menjadi kesimpulan logis pada konsep *Virtual Machine* (VM). Berikut adalah beberapa jenis dari pendekatan dari Virtualisasi.

##### a) Full Virtualization

Sistem operasi yang berjalan pada mesin *virtual* memberikan instruksi kepada perangkat keras dengan cara melalui mesin *virtual*.

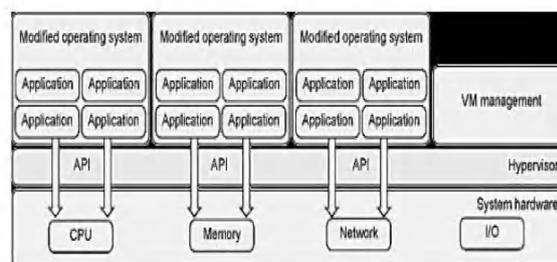
Full *Virtualization* memberikan pemodelan lengkap dari perangkat keras. Jadi *Full Virtualization* dapat membuat para *user* percaya bahwa terdapat system operasi berbeda pada suatu *computer*. Berikut adalah gambaran dari *Full Virtualization*.



Gambar 1 Skema Full Virtualisasi

##### b) Para Virtualization

Para *Virtualization* 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. Mesin *virtual* tidak mensimulasikan sistem operasi secara menyeluruh, tetapi menawarkan API yang membutuhkan modifikasi sistem operasi (merupakan teknik yang digunakan oleh XEN).



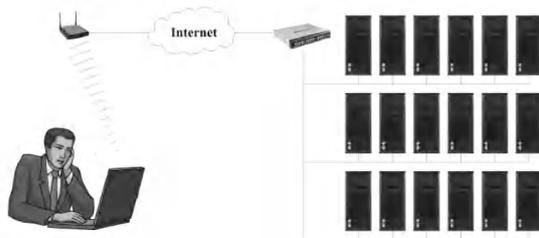
Gambar 1 Skema Para Virtualization

c) *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*, 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.

**5. Lab Virtual**

Lab virtual memungkinkan user dapat melakukan akses dari berbagai area yang terjangkau jaringan. Gambar 3 merupakan scenario akses yang dilakukan oleh user. Seorang user secara fisik berinteraksi dengan laptopnya untuk mengakses salah satu mesin virtual melalui *web-based interface*. [1]



Gambar 3 Ilustrasi akses mesin virtual oleh user

**6. Proxmox**

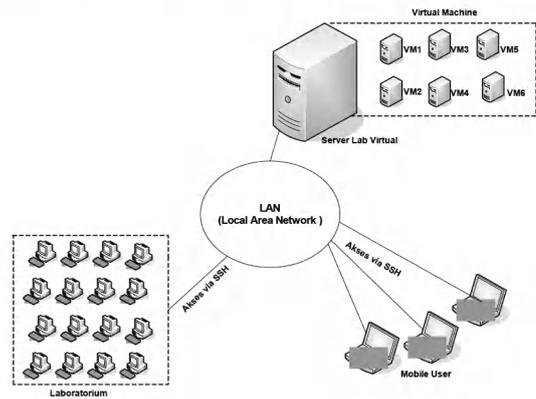
Proxmox VE (Virtual Environment) adalah salah satu distro Linux dari basis Debian yang mempunyai fungsi khusus sebagai Virtualisasi baik appliance maupun operating system. Proxmox berjalan dengan mode text, walaupun seperti itu Proxmox akan lebih mudah dikonfigurasi melalui akses remote dengan web based. Untuk sekarang Proxmox dengan release terbaru adalah Proxmox VE 2.3. [3]

**7. Pembahasan**

7.1 Perancangan Implementasi Lab Virtual

Laboratorium *virtual* dapat diakses melalui jaringan local baik melalui ruang laboratorium

ataupun diakses secara mobile melalui laptop dalam jaringan local.



Gambar 4 Desain Sistem Laboratorium *Virtual*

Desain seperti diatas memungkinkan *user* untuk mengakses *Laboratorium virtual* secara local baik menggunakan jaringan kabel maupun wireless.

Laboratorium virtual yang dibuat dalam penelitian ini terdiri dari sebuah server yang didalamnya terdapat beberapa server virtual. Server virtual inilah yang nantinya akan diakses oleh user melalui SSH (akses jarak jauh). Dengan demikian kebutuhan terhadap server fisik dapat dikurangi. Walaupun menggunakan mesin server virtual mahasiswa tetap mampu melakukan konfigurasi layaknya melakukan konfigurasi pada komputer fisik. Tabel 1 Merupakan spesifikasi server yang digunakan pada penelitian ini.

Tabel 1 Spesifikasi Server

No	Parameter	Spesifikasi
1	Prosesor	Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz
2	Memory	16 GB RAM DDR3
3	Hard disk	500

7.2 Pengujian Laboratorium Virtual

Setelah melakukan tahapan implementasi sistem, kemudian dilanjutkan dengan tahapan pengujian sistem. Pengujian sistem dilakukan untuk mengukur kinerja laboratorium *virtual* yang telah dibangun. Pengujian dilakukan dengan mengacu pada kebutuhan sistem yang sesuai dengan modul praktikum manajemen jaringan dan *server* yang berlaku di STMIK STIKOM Indonesia.

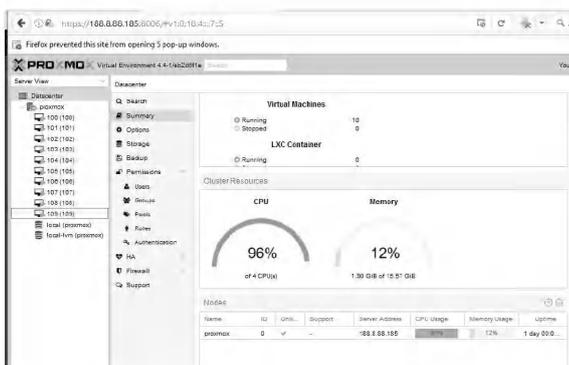
Pengujian laboratorium *virtual* yang dibangun dimulai dengan melakukan sinkronisasi waktu dari mesin-mesin *virtual* dan proxmox dengan menggunakan *network time protocol*. Seluruh mesin memiliki penunjuk waktu yang sama, hal ini

dilakukan agar waktu pengujian untuk seluruh mesin menjadi seragam. Keseragaman waktu diperlukan karena pengujian dilakukan dengan mengaktifkan 1, 2, 4, 6, 8, dan 10 mesin berturut-turut secara bersamaan. Dengan adanya keseragaman waktu, maka dapat dibangun script agar mesin-mesin *virtual* dapat diakses dalam satu waktu yang sama. Disamping itu, keseragaman waktu memungkinkan untuk membuat skenario agar beberapa mesin melakukan proses yang sama dalam satu waktu. Hal ini penting untuk mengetahui kinerja laboratorium *virtual* yang dibangun ketika terdapat 1, 2, 4, 6, 8, dan 10 mesin yang aktif secara bersamaan.

Berdasarkan modul praktikum manajemen jaringan dan *server* yang berlaku di STMIK STIKOM Indonesia, maka pengujian dikategorikan menjadi 4 (empat) bagian, yaitu:

1. Penggunaan *resources* ketika 1, 2, 4, 6, 8, dan 10 mesin yang aktif secara bersamaan (*startup*).

*Startup* merupakan kondisi mesin dari keadaan non-aktif sampai aktif untuk digunakan. Pengujian ini bertujuan untuk mengetahui kondisi penggunaan *resources* ketika terdapat beberapa mesin yang aktif secara bersamaan.

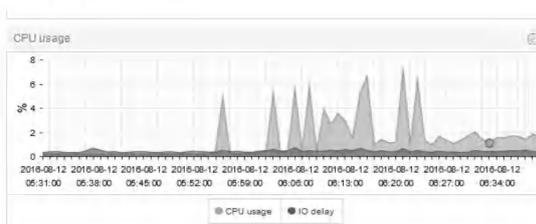


Gambar 5 CPU dan *memory* burst ketika 10 mesin *startup* bersamaan

Gambar 5 menunjukkan adanya CPU burst yang tinggi ketika 10 mesin diaktifkan secara bersamaan, CPU burst yang terjadi mencapai 96%, sedangkan *memory* masing tergolong rendah yaitu 12%. Hal ini menunjukkan bahwa, ketika mesin mulai diaktifkan maka dibutuhkan *resources* yang tinggi. Semakin banyak mesin yang dijalankan bersamaan maka semakin lama waktu yang dibutuhkan mesin untuk *startup*. Dengan 10 mesin yang diaktifkan secara bersamaan, menghasilkan CPU burst 96%. Hal ini menunjukkan penggunaan CPU yang sangat tinggi, karena mempengaruhi lamanya waktu *startup*. Berdasarkan data ini maka maksimal mesin yang dapat dijalankan bersamaan dengan rata-rata waktu *startup* 3 menit adalah sebanyak 10 mesin.

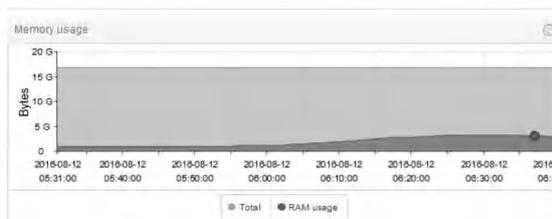
2. Penggunaan *resources* ketika 1, 2, 4, 6, 8, dan 10 mesin aktif melakukan *update repository*.

*Update repository* merupakan aktifitas pertama kali yang harus dilakukan ketika melakukan instalasi aplikasi pada *server*. *Repository* merupakan kumpulan aplikasi yang disediakan untuk sebuah sistem operasi yang umumnya istilah ini digunakan pada sistem operasi berbasis Linux. *Update repository* bertujuan untuk memperbaharui sumber aplikasi yang akan digunakan untuk melakukan instalasi aplikasi selanjutnya. Sehingga pada penelitian ini dilakukan pengujian dengan variasi jumlah mesin yang aktif bersamaan melakukan *update repository*. Pada Gambar 6 dan Gambar 7 menunjukkan penggunaan CPU dan *memory* pada *server*.



Gambar 6 Penggunaan CPU ketika mesin *update repository*

Pada gambar diatas menunjukkan grafik penggunaan CPU pada *server*. Pada waktu 05.31 – 05.54 belum menunjukkan penggunaan CPU, kemudian selanjutnya peneliti mengaktifkan 1 mesin untuk melakukan *update repository*. Pada waktu 05.54 – 05.56 terlihat penggunaan CPU yang signifikan dari sebelumnya, namun masing tergolong rendah. Penggunaan CPU pada waktu-waktu selanjutnya menunjukkan bahwa ketika mesin aktif melakukan proses, maka mesin mulai menggunakan CPU. Jumlah penggunaan CPU mengalami fluktuasi, namun pada gambar dapat dilihat bahwa penggunaan CPU cenderung meningkat seiring dengan meningkatnya jumlah mesin yang melakukan aktifitas *update repository*.



Gambar 7 Penggunaan *Memory* ketika *update repository*

Pada Gambar 7 menunjukkan penggunaan *memory* ketika mesin melakukan *update repository*. Penggunaan *memory* oleh mesin saat melakukan *update repository* tidak mengalami peningkatan

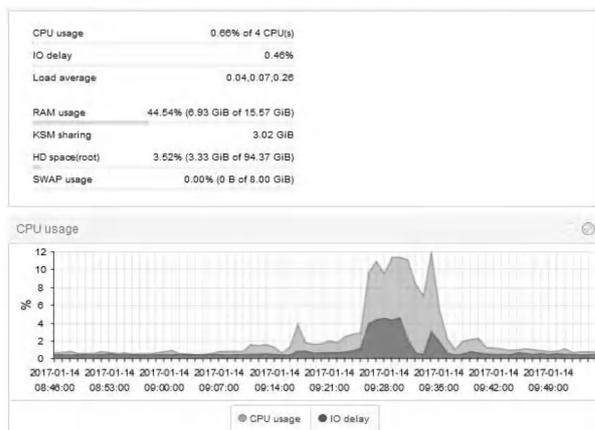
yang signifikan. Artinya *memory* yang dimiliki masih dapat digunakan untuk melakukan proses *update repository*.

3. Penggunaan *resources* ketika 1, 2, 4, 6, 8, dan 10 mesin aktif melakukan instalasi aplikasi dan konfigurasi.

Setelah tahapan *update repository*, selanjutnya dilakukan tahapan instalasi dan konfigurasi aplikasi *server*. Aplikasi *server* merupakan aplikasi yang diinstal pada *server* sesuai dengan fungsi *server* yang diinginkan. Pada penelitian ini aplikasi *server* yang diinstal adalah *web server*. *Web server* berfungsi sebagai penyedia halaman web yang nantinya dapat diakses oleh *user*. Aplikasi *web server* dipilih berdasarkan materi dari modul yang menunjukkan bahwa materi *web server* memiliki lebih banyak aplikasi yang harus diinstall dibandingkan dengan materi yang lainnya. Disamping banyaknya jumlah aplikasi yang harus diinstall, *web server* membutuhkan konfigurasi yang lebih kompleks dibandingkan dengan materi yang lain. Adapun aplikasi yang diinstall untuk membangun sebuah *web server* dapat dilihat pada Gambar 8.

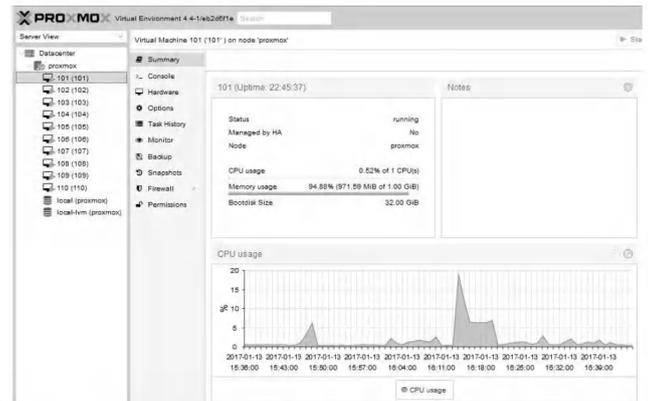
```
# aptitude install apache2
# aptitude install php5
# aptitude install php5-mysql
# aptitude install mysql-server
# aptitude install libapache2-mod-auth-mysql
# aptitude install phpmyadmin
```

Gambar 8 Aplikasi untuk membangun Web Server

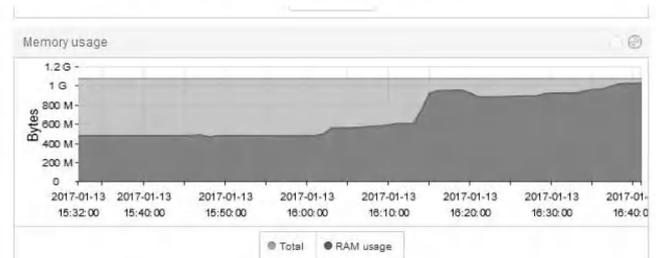


Gambar 9 Penggunaan CPU *server* utama saat instalasi aplikasi dan konfigurasi

Saat dilakukan instalasi dan konfigurasi aplikasi, kondisi *resources* pada *server* utama masih normal yaitu penggunaan CPU sebesar 0,66% dan *memory* 44,54%. Jumlah ini meningkat seiring dengan meningkatnya jumlah mesin yang melakukan instalasi dan konfigurasi aplikasi. Namun demikian penggunaan CPU masih berada dibawah 10%, dan dikategorikan CPU masih mampu menangani 10 mesin. Saat mesin melakukan instalasi dan konfigurasi aplikasi, penggunaan *memory* meningkat signifikan.



Gambar 10 Penggunaan CPU pada mesin *virtual* saat instalasi aplikasi dan konfigurasi



Gambar 11 Penggunaan *memory* pada mesin *virtual* saat instalasi aplikasi dan konfigurasi

Gambar 11 menunjukkan penggunaan CPU pada mesin *virtual* yang sedang aktif melakukan instalasi dan konfigurasi aplikasi. Penggunaan CPU pada saat instalasi dan konfigurasi aplikasi tidak mengalami peningkatan signifikan. Sedangkan penggunaan *memory* saat instalasi dan konfigurasi aplikasi mengalami peningkatan yang signifikan. Penggunaan *memory* mencapai 94,88% dari total *memory* 1 GB untuk masing-masing mesin *virtual*. Penggunaan *memory* dapat dilihat pada Gambar 11.

4. Penggunaan *resources* ketika 1, 2, 4, 6, 8, dan 10 mesin aktif diakses oleh *user* secara bersamaan.

Ketika *web server* sudah selesai diinstallasi dan konfigurasi, selanjutnya dilakukan pengujian akses *website* melalui *browser* oleh *user*.

*Website* dibangun menggunakan CMS (*Content Management Systems*). Tampilan *website* yang diakses oleh *user* dapat dilihat pada Gambar 12



Gambar 12 Contoh Tampilan *website*

Sesuai dengan tujuan Praktikum Manajemen Jaringan dan *Server*, mahasiswa harus mampu membangun sebuah *server*. Pada materi Web *Server*, mahasiswa diwajibkan untuk mampu membangun sebuah web *server* dan dapat diakses melalui jaringan lokal. Gambar 12 merupakan contoh tampilan *website* yang dikonfigurasi pada web *server* yang telah dibangun. Web *server* dibangun pada masing-masing mesin, yang selanjutnya diakses melalui *user*. Akses *website* yang dilakukan oleh *user* tidak memberikan pengaruh peningkatan *resources* baik CPU dan *memory* secara signifikan. Hal ini dilakukan karena hanya terdapat 2 *user* saja yang melakukan akses ke mesin. Walaupun demikian dengan mengacu pada tujuan praktikum manajemen jaringan dan *server*, spesifikasi yang diberikan untuk mesin *virtual* memadai untuk melakukan praktikum.

## 8. Kesimpulan

1. Perancangan dan pembangunan laboratorium *virtual* untuk Praktikum Manajemen Jaringan dan *Server* di STMIK STIKOM Indonesia dilakukan melalui instalasi dan konfigurasi proxmox, analisis kebutuhan spesifikasi mesin *virtual* yang mengacu pada materi modul Praktikum Manajemen Jaringan dan *Server*, serta pengujian dan evaluasi sistem.
2. Penggunaan *resources server* meningkat seiring dengan peningkatan jumlah mesin *virtual* yang melakukan aktifitas baik instalasi aplikasi, konfigurasi, dan akses oleh *user*. Namun ketika dilakukan instalasi aplikasi, penggunaan CPU dan *memory* pada masing-masing mesin *virtual*

yang melakukan aktifitas, meningkat hingga 98%. Penggunaan CPU dan *memory* pada *server* utama mencapai 20%. Dengan jumlah total *memory* sebesar 16GB, *server* hanya mampu mengaktifkan mesin sebanyak 10 buah secara bersamaan.

3. Alokasi 1GB *memory* untuk masing-masing mesin *virtual*, merupakan jumlah minimum yang harus diberikan untuk memenuhi kebutuhan dalam penggunaan Modul Praktikum Manajemen Jaringan dan *Server*. Dengan 4 core CPU yang disediakan, mampu digunakan oleh 10 mesin *virtual* yang aktif secara bersamaan. Jika ingin membangun 20 mesin *virtual*, maka diperlukan 2 buah *server* dengan spesifikasi yang sama.

## 9. Daftar Pustaka

- [1] Burd, Stephen D, et al, 2009, "Virtual Computing Laboratories: A Case Study with Comparisons to Physical Computing Laboratories" Journal of Information Technology Education: Innovations in Practice.
- [2] Thohari, A. Hamim, et al, 2014, "Rancang Bangun Lab Komputer Virtual Berbasis Cloud Computing Menggunakan Proxmox Pada Jaringan Terpusat" Politeknik Negeri Batam
- [3] Sumarto, et al, 2015, "Rancang Bangun Lab Komputer Virtual Berbasis Cloud Computing Menggunakan Ovirt Pada Jaringan Terpusat", Politeknik Negeri Batam
- [4] Mark-Shane E. Scale (2009) Cloud Computing and Collaboration. Library Hi Tech News, Vol. 26 Iss: 9, pp.10 - 13).
- [5] Ercana, Tuncay (2010) Effective Use of Cloud Computing in Educational Institutions. Procedia Social and Behavioral Sciences 2 (2010) : p. 938–942
- [6] Mell, P and Grance T (2011). The NIST Definition of Cloud Computing. Special Publication 800-145. tersedia online pada : <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>

**First Author:** Menempuh pendidikan S1 di Universitas Udayana tahun 2006-2010 dan Pendidikan S2 di Instittut Teknologi Sepuluh Nopember Surabaya tahun 2011-2013. Saat ini berstatus tenaga pendidik di STMIK STIKOM Indonesia.

**Second Author:** Menempuh pendidikan S1 di Universitas Udayana tahun 2008-2012 dan Pendidikan S2 di Universitas Udayana tahun 2014-2016. Saat ini berstatus tenaga pendidik di STMIK STIKOM Indonesia..