



## Model Infrastruktur dan Manajemen *Platform Server* Berbasis *Cloud Computing*

Mulki Indana Zulfa<sup>1</sup>, Ari Fadli<sup>2</sup>, Yogi Ramadhani<sup>3</sup>

<sup>1,2,3</sup>Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jenderal Soedirman

Jl. Mayjen Sungkono KM No.5, Blater, Purbalingga 53371, Indonesia

Email korespondensi : mulki\_indanazulfa@unsoed.ac.id

Dikirim 17 Agustus 2017, Direvisi - , Diterima 15 September 2017

Abstrak - *Cloud computing* adalah teknologi baru yang masih sangat berkembang pesat. Teknologi ini menjadikan internet sebagai media utama untuk mengelola data dan aplikasi secara *remote*. Teknologi ini juga memungkinkan para pengguna untuk menjalankan suatu aplikasi tanpa harus memikirkan aspek infrastruktur dan *platform*-nya. Aspek teknis lain seperti *memory*, *storage*, *backup* and *restore*, dapat dilakukan dengan sangat mudah karena semuanya sudah disediakan oleh penyedia jasa *cloud*. Penelitian ini bertujuan untuk merancang dan memodelkan infrastruktur serta manajemen kebutuhan *platform server* di jaringan komputer Fakultas Teknik Universitas Jenderal Soedirman. Tahap pertama dalam penelitian ini adalah studi literatur, dengan mencari tahu model implementasi pada penelitian sebelumnya. Kemudian hasilnya digabungkan dengan pendekatan baru yang disesuaikan dengan *resource* yang tersedia dan mencoba mengimplementasikannya langsung di jaringan server yang ada. Hasil percobaan menunjukkan, implementasi teknologi *cloud computing* mampu menggantikan infrastruktur jaringan yang lama tanpa mengubah kondisi server yang sudah ada sebelumnya.

Kata kunci – *cloud computing*, jaringan komputer, *platform server*, model infrastruktur.

Abstract - *Cloud computing* is a new technology that is still very rapidly growing. This technology makes the Internet as the main media for the management of data and applications remotely. *Cloud computing* allows users to run an application without having to think about infrastructure and its platforms. Other technical aspects such as *memory*, *storage*, *backup* and *restore*, can be done very easily. This research is intended to modeling the infrastructure and management of computer platform in computer network of Faculty of Engineering, University of Jenderal Soedirman. The first stage in this research is literature study, by finding out the implementation model in previous research. Then the result will be combined with a new approach to existing resources and try to implement directly on the existing server network. The results showed that the implementation of *cloud computing* technology is able to replace the existing platform network.

Keywords – component, *cloud computing*, computer network, server platform, infrastructure model.

### I. PENDAHULUAN

Saat ini masalah utama dalam manajemen jaringan komputer terutama yang berhubungan dengan komputasi adalah *resource sharing*. Pada server fisik konvensional, konfigurasi *hardware* seperti prosesor, *memory*, dan *harddisk* harus dilakukan di awal instalasi. Kebutuhan untuk mengubah konfigurasi *hardware* secara cepat tidak mungkin dilakukan pada kondisi server seperti itu. Terlebih jika pertumbuhan data semakin cepat maka permintaan untuk menambah alokasi server baru akan semakin bertambah pula. *Cloud computing* adalah solusi yang tepat untuk masalah tersebut. Menurut Onno W Purbo, *cloud*

*computing* merupakan sebuah model komputasi dimana sumber daya seperti prosesor, *storage*, *network*, dan *software* menjadi abstrak dan hanya diberikan sebagai layanan di jaringan internet menggunakan pola akses *remote* [1]. Selain itu, *Cloud computing* merupakan sebuah model komputasi, yang sumber daya yang digunakan seperti *processor*, *memory*, *storage*, *network*, dan *software* menjadi abstrak dan diberikan sebagai layanan di jaringan dengan menggunakan pola akses *remote* [2]. Dengan penerapan teknologi *cloud computing* ini, permintaan alokasi sebuah server untuk kepentingan tertentu, misalnya untuk kepentingan riset, dapat diadakan dengan cepat dan mudah. Selain itu dengan menerapkan *cloud computing*, maka proses

*network scalability* dapat lebih dinamis dan mudah dikontrol [3]. Efektivitas biaya merupakan salah satu faktor yang signifikan pada pengambilan keputusan implementasi teknologi baru, terutama *Cloud Computing* [4].

Permasalahan berbeda ditunjukkan oleh hasil observasi penelitian [5], dimana *resource server* yang terpakai sangat tidak efisien. Hasil penelitian tersebut menunjukkan spesifikasi server Intel Pentium Core i5 dengan RAM 4GB, yang selama ini digunakan sebagai webserver, hanya mengonsumsi CPU sebanyak 1% dan RAM 26%. Dalam penelitian [6], untuk mengoptimalkan sumber daya / *resource server* yang ada, maka dibangunlah sebuah teknologi *cloud* berbasis virtualisasi dengan memanfaatkan sistem operasi Proxmox VE dan FreeNAS. Dalam penelitian tersebut, FreeNAS dikonfigurasi sebagai media *backup*.

Kemudian hasil analisis kinerja virtualisasi server tersebut ditulis di dalam penelitian [7], dimana beban kerja *workstation* yang ada di dalam server virtualisasi tidak saling memengaruhi *resource workstation* lainnya.

Berdasarkan hasil penelitian tersebut, maka dalam penelitian ini diusulkan model infrastruktur *cloud computing* dengan memanfaatkan Proxmox VE sebagai server virtualisasi dan 2 server FreeNAS sebagai *backup data center*.

## II. METODE PENELITIAN

### A. Analisis Kebutuhan Platform Server

Sebelumnya ada beberapa platform server yang sudah berjalan di jaringan komputer Fakultas Teknik Unsoed. Kebutuhan yang paling banyak adalah menyediakan *platform* server untuk menyediakan informasi jurusan dalam bentuk portal website. Di Fakultas Teknik Unsoed, ada 5 jurusan yang telah memiliki portal website dan 1 portal website fakultas. Semua portal website jurusan harus berada di bawah subdomain fakultas teknik, yaitu ft.unsoed.ac.id.

Tahap menentukan kebutuhan sistem merupakan bagian paling penting dalam membangun sebuah sistem. Kebutuhan sistem ini mencakup tentang apa yang harus dikerjakan dan karakteristik apa yang harus dimiliki sistem.

Tabel 1 menjelaskan kebutuhan platform server seperti portal website jurusan, Fakultas Teknik Unsoed mempunyai server jurnal elektronik, beberapa Sistem Informasi (SI) seperti legalisir *online*, pengajuan tugas akhir *online* dan laboratorium pengujian teknik sipil (iLab Sipil).

Tabel 1. Kebutuhan Platform Server

No	Jenis Server	Platform	Domain
1	Portal website fakultas	Drupal	ft.unsoed.ac.id
2	Portal website jurusan	Wordpress	Contoh : elektro.ft.unsoed.ac.id
3	Jurnal <i>online</i>	OJS	dinarek.unsoed.ac.id

No	Jenis Server	Platform	Domain
4	SI legalisir <i>online</i>	Play framework	sillo.ft.unsoed.ac.id
5	SI pengajuan tugas akhir	Lavarel framework	sikap.ft.unsoed.ac.id
6	iLab Sipil	Play framework	ilab.unsoed.ac.id

Dari keenam jenis kebutuhan server tersebut, hanya server nomor 1, 2, dan 3 yang sudah *running* pada jaringan lama. Untuk mengembangkan jaringan tersebut menjadi jaringan server dengan teknologi *cloud*, Fakultas Teknik telah menyediakan 1 server jenis blade dan 2 server jenis *desktop*. Hasil dari penelitian ini menjadi rekomendasi konfigurasi jaringan server berbasis *cloud* di Fakultas Teknik, sehingga keenam jenis kebutuhan server tersebut dapat dipenuhi.

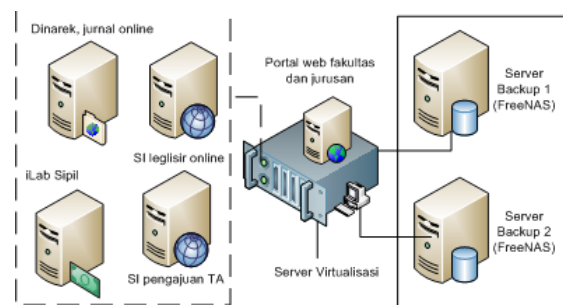
### B. Konfigurasi Infrastruktur Jaringan Cloud

Tabel 2 memuat spesifikasi 3 server yang disediakan oleh fakultas untuk membangun jaringan server berbasis *cloud*.

Tabel 2. Spesifikasi Server

No	Spesifikasi Server	Target konfigurasi
1	2 Prosesor Xeon E5-2620v4 RAM 32 GB DDR4 4x Gb Ethernet Network card Harddisk 1 TB SAS, 320 GB SAS dan 1 TB SATA, masing-masing berjumlah 2	Server virtualisasi dengan Proxmox VE (Virtual Environment)
2	Intel Pentium Core i5 RAM 4 GB DDR3 Harddisk 500 GB	Server backup 1 FreeNAS v9.10.2
3	Intel Pentium Core i5 RAM 4 GB DDR3 Harddisk 500 GB	Server backup 2 FreeNAS v9.10.2

Strategi yang digunakan dalam membangun jaringan server berbasis *cloud* adalah menggunakan teknologi virtualisasi dengan sistem operasi Proxmox versi 4.3 yang dijalankan pada mesin server dengan spesifikasi terbaik (lihat Tabel 2, nomor 1). Proxmox adalah sistem operasi hasil modifikasi kernel Linux Debian. Proxmox hanya tersedia untuk arsitektur komputer 64bit. Dengan konsep ini, maka di dalam server nomor 1 dapat diciptakan sumber daya baru untuk memenuhi semua kebutuhan server yang telah dijelaskan pada Tabel 1.



Gambar 1. Desain Topologi Jaringan Cloud

Gambar 1 menjelaskan tentang desain topologi jaringan server berbasis *cloud*. Garis putus-putus menunjukkan bahwa server di dalam garis tersebut adalah *host server* yang berada di dalam server utama virtualisasi. Kemudian dari hasil penelitian [4] yang hanya menerapkan 1 FreeNAS sebagai server *backup*, maka dalam penelitian ini kebutuhan server *backup* ditambah menjadi 2 FreeNAS. Versi FreeNAS yang digunakan adalah 9.10.2.

Setiap kebutuhan server pada Tabel 1 dikonfigurasi secara terpisah baik sistem operasi maupun versi platformnya. Pemilihan versi *platform* disesuaikan dengan tahap pengembangan awalnya. Kebutuhan portal website fakultas dan jurusan dijadikan satu di dalam satu server yang dibantu dengan *tools* pengelola *hosting* yaitu Sentora. Sentora dipilih karena kemudahan instalasi dan konfigurasinya.

Ada 5 alokasi host server yang dibuat pada server virtualisasi. Tabel 3 menunjukkan konfigurasi server dan versi *platform* yang diimplementasikan pada jaringan baru berbasis teknologi *cloud* di Fakultas Teknik Unsoed.

Tabel 3. Kebutuhan Platform Server

No	Jenis Server	Platform	Sistem Operasi
1	Portal website fakultas dan 5 jurusan	Drupal dan wordpress	Ubuntu 14.04 64bit, dengan <i>tools</i> pendukung Sentora Panel
2	Dinarek, Jurnal online	OJS versi 2.4.7	Centos 6, 64bit
3	SI legalisir online	Play Framework 1.2.4	Ubuntu 14.04, 64bit
4	SI pengajuan tugas akhir	Framework Lavarel 5.2	Ubuntu 14.04, 64bit
5	iLab Sipil	Play Framework 1.2.4	Ubuntu 14.04, 64bit

### III. HASIL PENELITIAN

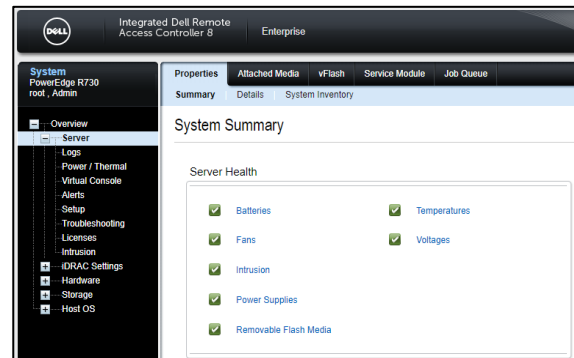
Desain topologi jaringan server baru berbasis *cloud* di Fakultas Teknik menggantikan jaringan server lama yang selama ini dikelola terpusat oleh Lembaga Pengembang Teknologi dan Sistem Informasi (LPTSI) Unsoed.

#### A. Konfigurasi Server Virtualisasi

Server utama virtualisasi menggunakan vendor dari Dell yang menyediakan akses remote BIOS server melalui *integrated Dell Remote Access Controller* (iDRAC). iDRAC adalah antarmuka khusus (lihat Gambar 2) yang dapat diakses melalui browser. Melalui iDRAC ini semua status server dapat diketahui dengan cepat, bahkan ketika server dalam keadaan mati.

Sumber daya *harddisk* yang dimiliki oleh server utama virtualisasi ini ada 6, detailnya dapat dilihat kembali pada Tabel 2. Dengan sumber daya seperti itu memungkinkan untuk dikonfigurasi RAID (*Redundant*

*Array Independent Disk*) level 1. RAID level 1 adalah teknik penulisan data secara identik (*mirror*) pada 2 *harddisk* atau lebih. Sehingga jika salah satu *harddisk* terdeteksi rusak, sistem akan tetap berjalan dengan mengambil data pada *harddisk* lainnya.



Gambar 2. Antarmuka Web iDRAC

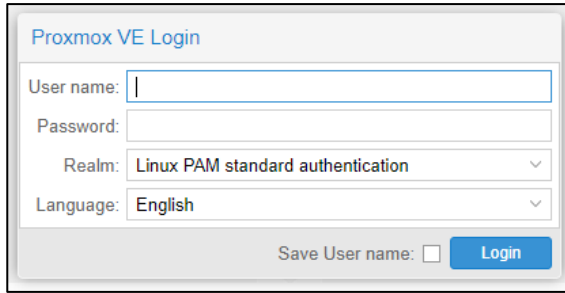
Setelah konfigurasi BIOS dan RAID telah selesai, langkah berikutnya adalah melakukan instalasi Proxmox VE 4.3. Ada 2 pilihan cara instalasi proxmox yaitu dengan menggunakan *live USB* dan CD. Pada penelitian ini, instalasi proxmox yang dipilih adalah menggunakan *live USB*.



Gambar 3. Konfigurasi IP Address

Jika pada saat instalasi proxmox, server yang digunakan telah terhubung ke jaringan internet maka pada tahap instalasi Gambar 3, textbox *IP address* akan otomatis terisi alamat IP *public* (lihat Gambar 3).

Ketika instalasi telah masuk tahap akhir, komputer server diminta untuk melakukan *restart*. Setelah proses *restart* selesai, server virtualisasi proxmox siap untuk digunakan. Proxmox tidak memiliki *Graphical User Interface* (GUI), untuk memulai melakukan manajemen *host server*, dapat dilakukan secara *remote* dengan menggunakan browser. Dengan mengakses alamat IP server virtualisasi via https, maka tampilan login proxmox akan muncul seperti yang ditunjukkan Gambar 4. Data login yang digunakan untuk masuk ke dalam sistem utama proxmox, dimasukkan ketika tahap instalasi proxmox.



Gambar 4. Tampilan Login Server Proxmox

**B. Konfigurasi Server Backup**

Instalasi FreeNAS akan memformat semua *harddisk* yang terpasang. Sehingga dalam penelitian ini, sistem operasi FreeNAS akan disimpan di dalam flashdrive 64GB, sedangkan sejumlah *harddisk* yang terdapat di dalam server akan diformat semua untuk kepentingan *backup* server utama virtualisasi. Dengan konfigurasi seperti ini, semua file *host server* yang dibuat di server utama virtualisasi sebenarnya disimpan di dalam *harddisk* yang terdapat di *server backup* FreeNAS. Jumlah *server backup* FreeNAS yang disediakan ada dua.

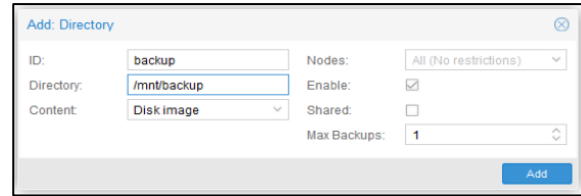
Setelah instalasi FreeNAS selesai, untuk memulai konfigurasi FreeNAS dapat dilakukan secara *remote* melalui *browser* (lihat Gambar 5). *Password* login ke dalam sistem FreeNAS ditentukan pada saat tahap instalasi.



Gambar 5. Tampilan Login Server Freenas

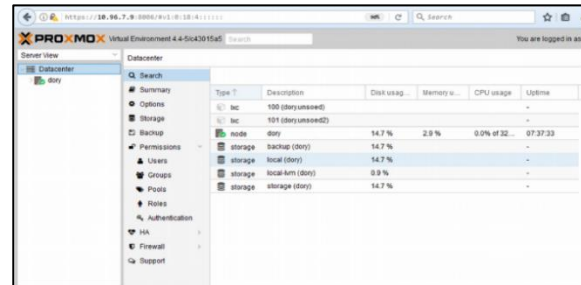
Setelah kedua FreeNAS selesai diinstalasi, tahap berikutnya adalah mengkonfigurasi server virtualisasi proxmox untuk menambahkan *directory backup* terhadap dua server FreeNAS.

Langkah yang dapat dilakukan untuk menambahkan *directory backup* adalah masuk ke dalam sistem utama proxmox, kemudian pilih menu *Data Center*. Setelah itu pilih menu *storage* → *add* → *directory* seperti yang ditunjukkan pada Gambar 6.



Gambar 6. Menu Add Storage Direcotry

Jika proses penambahan *directory* berhasil, maka beberapa *list storage* akan muncul di menu *datacenter* proxmox (lihat Gambar 7).



Gambar 7. Datacenter Yang Berhasil Ditambahkan

**C. Konfigurasi Host Server**

Tipe server virtual yang dapat dibuat oleh proxmox ada dua yaitu VM (*Virtual Machine*) dan CT (*Container*). *Container* di proxmox menggunakan LXC. LXC atau *Linux Container* adalah virtualisasi pada level sistem operasi yang dapat digunakan untuk menjalankan banyak sistem operasi yang terisolasi pada sebuah komputer. LXC tidak menyediakan *virtual machine*, tapi hanya menyediakan lingkungan virtual yang terisolasi dan memiliki proses dan jaringan sendiri. VM membutuhkan lebih banyak *resource* komputer induknya (server utama virtualisasi), karena kernel OS-nya lengkap dan semua program yang digunakan harus diinstalasi terlebih dahulu. sedangkan CT lebih ringan, karena aplikasi yang ada menggunakan *kernel* yang sama dengan komputer induknya. Mengingat hal tersebut maka dalam penelitian ini, semua *host server* yang dibuat oleh mesin utama virtualisasi menggunakan jenis CT.

Tabel 4 menjelaskan kebutuhan *host server* yang dibuat dalam server utama virtualisasi. Perbedaan spesifikasi sumber daya *host server* ditentukan oleh prediksi beban kerja dan pertumbuhan data dalam *storage*, selama server tersebut *running* dalam melayani *request* dari *client*.

Tabel 4. Konfigurasi Sumber Daya Host Server

No	Jenis Server	Tipe Host	Resource
1	Portal website fakultas dan jurusan	Container (CT)	Harddisk 200 GB RAM 4 GB CPU Core 2 Memory swap 1GB
2	Dinarek, Jurnal online	Container (CT)	Harddisk 200 GB RAM 4 GB CPU Core 2 Memory swap 1GB

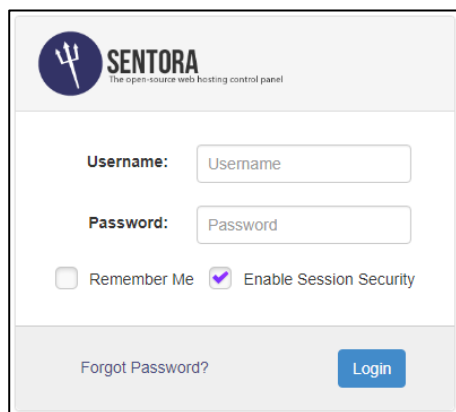
No	Jenis Server	Tipe Host	Resource
3	SI legalisir online	Container (CT)	Harddisk 50 GB RAM 2 GB CPU Core 2 Memory swap 1GB
4	SI pengajuan tugas akhir	Container (CT)	Harddisk 100 GB RAM 4 GB CPU Core 2 Memory swap 1GB
5	iLab Sipil	Container (CT)	Harddisk 200 GB RAM 4 GB CPU Core 2 Memory swap 1GB

D. Manajemen Platform

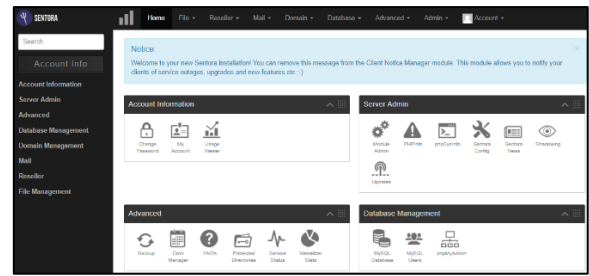
Berdasarkan informasi dari Tabel 3, portal website fakultas teknik dan kelima jurusan yang ada, digabung ke dalam 1 *host server*. Dari Tabel 3 juga dapat diketahui bahwa portal website fakultas dan jurusan menggunakan *platform* yang berbeda yaitu drupal dan wordpress. Oleh karena itu dibutuhkan tools khusus untuk menangani kebutuhan perbedaan *platform* tersebut. Penelitian ini *tools* yang digunakan adalah Sentora Control Panel. Sentora Control Panel adalah *open source tools* yang dapat digunakan untuk mengelola kebutuhan *web hosting*. Sentora dipilih karena kemudahan instalasi dan penggunaannya.

Instalasi sentora dilakukan secara *online* dengan melakukan perintah *wget* terhadap *official website* sentora. Hal-hal yang perlu diperhatikan selama proses instalasi sentora adalah *IP address* atau target nama domain yang akan digunakan. Dalam penelitian ini, nama domain yang digunakan adalah *teknik.unsoed.ac.id*. Kemudian semua alamat website jurusan menjadi subdomainnya, contohnya *te.teknik.unsoed.ac.id* (lihat Gambar 10).

Setelah sentora berhasil diinstalasi, untuk masuk ke dalam menu utama sentora cukup mengakses alamat *IP address* sentora atau nama domainnya via browser. Jika berhasil, maka *form login* akan tampil (lihat Gambar 8) dan meminta user untuk melakukan otentifikasi sebelum masuk ke dalam *dashboard* utama (lihat Gambar 9).



Gambar 8. Form Login Sentora



Gambar 9. Dashboard Sentora

Sub-domain	Home directory	Status
te.teknik.unsoed.ac.id	/te_teknik_unsoed_ac_id	Live
tg.teknik.unsoed.ac.id	/tg_teknik_unsoed_ac_id	Live
tl.teknik.unsoed.ac.id	/tl_teknik_unsoed_ac_id	Live
ts.teknik.unsoed.ac.id	/ts_teknik_unsoed_ac_id	Live

Gambar 10. Manajemen Subdomain di Sentora

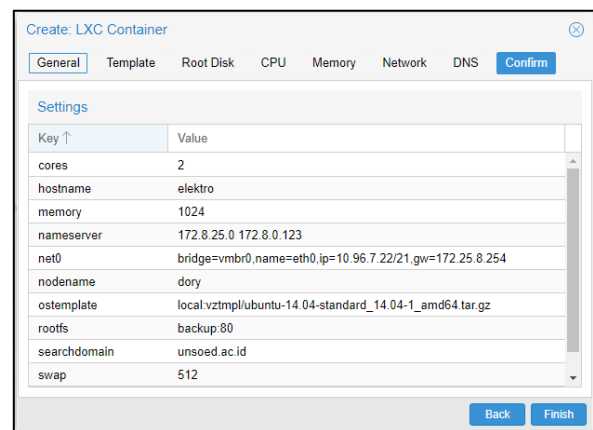
IV. PEMBAHASAN

A. Membuat host server dengan CT

Semua kebutuhan *platform host server* di mesin utama virtualisasi menggunakan *container* (CT). Terdapat berbagai macam jenis pilihan sistem operasi yang dapat dibuat melalui CT ini seperti debian, ubuntu, fedora, centos dan lain sebagainya. Sistem operasi tersebut dapat dibuat melalui file *template* berekstensi *.tar.gz* yang dapat diunduh di <https://download.openvz.org/template/precreated/>.

Semua *template* tersebut harus diupload ke mesin utama virtualisasi, tepatnya di *directory* */var/lib/vz/template/cache*. Metode upload yang dapat digunakan yaitu perintah *scp* via SSH, transfer langsung dari windows explorer via *winscp* atau *filezilla*.

Beberapa paramater penting untuk membuat CT baru adalah alokasi *disk*, *memory*, *swap*, *CPU core*, *IP address*, *gateway*, *DNS*, dan *password root* yang akan digunakan. Gambar 11 menunjukkan *parameter summary* sebelum CT akan dibuat oleh proxmox.



Gambar 11. Spesifikasi Membuat CT Baru

Host server dengan CT ini terdapat kelemahan yaitu tidak dapat memodifikasi kernel yang digunakan [8]. Hal ini terjadi karena kernel host server dengan skema CT masih menyatu dengan kernel induknya (mesin utama virtualisasi). Tetapi skema CT ini juga mempunyai kelebihan yaitu penggunaan memori yang lebih efisien.

**B. Penggunaan Resource Server Utama Virtualisasi**

Berdasarkan Tabel 2 di atas, jumlah harddisk yang dimiliki oleh server utama virtualisasi ada 6 yaitu 2 buah disk 320 GB SAS, 2 buah disk 1 TB SAS, dan 2 buah disk 1 TB SATA, dimana 3 disknya diset menjadi RAID level 1, sehingga secara sistem, harddisk yang dikenal oleh proxmox hanya 3 (lihat Tabel 5).

Tabel 5. Perbandingan Resource Disk

Jumlah volume harddisk server utama	Jumlah volume harddisk host server
<b>Total disk :</b>	<b>Alokasi disk :</b>
2 x 320 GB SAS	Server 1 : 200 GB
2 x 1 TB SAS	Server 2 : 200 GB
2 x 1 TB SATA	Server 3 : 50 GB
<b>RAID level 1:</b>	Server 4 : 100 GB
1 x 320 GB SAS	Server 5 : 200 GB
1 x 1 TB SAS	
1 x 1 TB SATA	Total disk terpakai 750 GB
<b>Total disk terbaca :</b>	
1 x 320 GB SAS (sistem operasi)	
1 x 1 TB SAS (local disk)	
1 x 1 TB SATA (not used)	

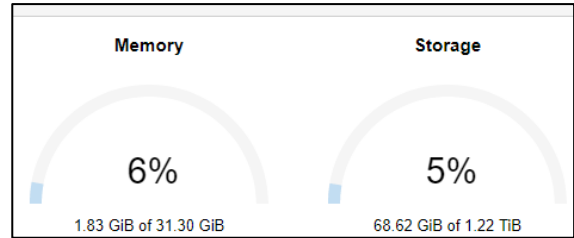
Kemudian berdasarkan Tabel 2 juga, pada server utama virtualisasi terpasang memori 32 GB. Sedangkan total alokasi memori pada host server berjumlah 18 GB. Sehingga ada sekitar 14 GB free memory pada server utama virtualisasi (lihat Tabel 6).

Tabel 6. Perbandingan Resource Memori

Jumlah volume memori server utama	Jumlah volume memori host server
<b>Total RAM :</b>	<b>Alokasi disk :</b>
32 GB DDR4	Server 1 : 4 GB
	Server 2 : 4 GB
	Server 3 : 4 GB
	Server 4 : 4 GB
	Server 5 : 2 GB
	Total memori terpakai 18 GB

Perbandingan kedua resource tersebut ditampilkan juga melalui menu summary.

Pada Gambar 12, statistik yang ditunjukkan langsung dari proxmox tidak sama dengan perhitungan yang ada pada Tabel 5 dan 6, hal ini disebabkan karena alokasi resource (disk dan memori) yang diset pada semua host server tersebut belum dalam kondisi penuh (sesuai dengan alokasi yang diset pada saat pembuatan CT baru).



Gambar 12. Penggunaan Resource Server Utama

**C. Servis yang berjalan pada host server**

Berdasarkan Tabel 4 di atas, semua kebutuhan platform server yang ada di host server, berjenis container (CT). Skema CT ini mempunyai kelebihan dalam penggunaan memori yang lebih ringan. Kelebihan lain dari CT adalah terisolasinya semua platform aplikasi yang telah terinstalasi di dalamnya, artinya semua service yang berjalan di satu CT, tidak berpengaruh terhadap CT lainnya [9].

Gambar 13 menunjukkan servis yang berjalan di host server yang mengimplementasikan Sistem Informasi Legalisir Online. Servis yang berjalan tersebut didapatkan melalui perintah ps aux yang dieksekusi secara remote menggunakan ssh.

```

root@ilab1:~# ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0  33276  3828 ?        Ss   Aug11  0:00 /sbin/init
root      651  0.0  0.0  34856   176 ?        Ss   Aug11  0:00 upstart-udev-bridge --daemon
root      822  0.0  0.0  49544  3276 ?        Ss   Aug11  0:00 /lib/systemd/systemd-udev --da
root     1066  0.0  0.0  15652  1844 ?        Ss   Aug11  0:00 upstart-socket-bridge --daemon
root     1098  0.0  0.0  15200  192 ?        Ss   Aug11  0:00 upstart-file-bridge --daemon
message+ 1634  0.0  0.0  39228  2264 ?        Ss   Aug11  0:00 dbus-daemon --system --fork
syslog    1652  0.0  0.0  255848  3444 ?       Ssl  Aug11  0:17 rsyslogd
root     1687  0.0  0.0  43456  3152 ?        Ss   Aug11  0:00 /lib/systemd/systemd-logind
root     1704  0.0  0.1  43372  3386 ?        Ss   Aug11  0:00 /usr/sbin/sshd -D
root     1730  0.0  0.0  23660  2096 ?        Ss   Aug11  0:00 cron
root     1751  0.0  0.0  19304  2224 ?        Ss   Aug11  1:46 /usr/sbin/irqbalance
mysqld    1811  0.0  1.2 696152 53808 ?       Ssl  Aug11  3:10 /usr/sbin/mysqld
root     1867  0.0  0.0  25340  2916 ?        Ss   Aug11  0:01 /usr/lib/postfix/master
postfix   1881  0.0  0.0  27464  2828 ?        Ss   Aug11  0:00 qmgr -l -t unix -u
root     1902  0.0  0.3  48004 14908 ?        Ss   Aug11  0:00 python /opt/play-1.2.4/play run
root     1904  0.0  0.0  12792  1316 lxc/consol Ss   Aug11  0:05 /sbin/getty -8 38400 console
root     1906  0.0  0.0  12792  1898 lxc/tty2 Ss+  Aug11  0:00 /sbin/getty -8 38400 tty2
root     1907  0.0  0.0  12792  1968 lxc/tty1 Ss+  Aug11  0:00 /sbin/getty -8 38400 tty1
root     1923  0.0  8.7 481262 363068 ?       Sl   Aug11  5:13 java -javaagent:/opt/play-1.2.4
postfix   3840  0.0  0.0  27452  2920 ?        Ss   07:54  0:00 pickup -l -t unix -u -c
root     3860  0.0  0.1  105640  6604 ?       Ss   08:48  0:00 shhd: root@pts/3
root     3879  0.0  0.1  19204  4408 pts/3 Ss   08:48  0:00 -bash
root     3985  0.0  0.0  15572  2044 pts/3 R+   08:50  0:00 ps aux
    
```

Gambar 13. Host Server Legalisir Online

Berdasarkan Tabel 1, Sistem Informasi Legalisir Online dibangun menggunakan Play Framework dengan bahasa java. Sehingga dalam dalam host server tersebut berjalan servis java.

```

root@teknik1:~# ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0  33596  4072 ?        Ss   Aug11  0:00 /sbin/init
root      307  0.0  0.0  12492  124 ?        Ss   Aug11  0:00 upstart-udev-bridge --daemon
root      368  0.0  0.0  49284  3208 ?        Ss   Aug11  0:00 /lib/systemd/s
root      737  0.0  0.0  23436  2312 ?        Ss   Aug11  0:00 rpcbind
root      872  0.0  0.0  15420  1780 ?        Ss   Aug11  0:00 upstart-file-b
root      877  0.0  0.0  15540  2012 ?        Ss   Aug11  0:00 upstart-socket
statd    1008  0.0  0.0  21556  2536 ?        Ss   Aug11  0:00 rpc.statd -L
syslog    1446  0.0  0.2 256052 12356 ?       Ssl  Aug11  0:04 rsyslogd
root     1838  0.0  0.1  61396  5352 ?        Ss   Aug11  0:00 /usr/sbin/sshd
root     1880  0.0  0.1  17788  2624 ?        Ss   Aug11  0:00 /usr/sbin/dove
daemon   1882  0.0  0.0  19152  164 ?        Ss   Aug11  0:00 atd
root     1883  0.0  0.0  23668  2360 ?        Ss   Aug11  0:01 cron
root     1886  0.0  0.0  19320  2276 ?        Ss   Aug11  1:38 /usr/sbin/irqb
bind     1903  0.0  2.0 2487852 85272 ?       Ssl  Aug11  0:02 /usr/sbin/name
dovecot   1943  0.0  0.0  9300  2000 ?        Ss   Aug11  0:00 dovecot/anvil
root     1948  0.0  0.0  2428  100 ?        Ss   Aug11  0:00 dovecot/loq
root     1946  0.0  0.0  23004  3664 ?        Ss   Aug11  0:00 dovecot/connfig
mysqld    1954  0.0  2.6 887396 110448 ?       Ssl  Aug11  3:33 /usr/sbin/mysq
root     2095  0.0  0.0  25356  2968 ?        Ss   Aug11  0:02 /usr/lib/postf
postfix   2108  0.0  0.0  27600  2856 ?        Ss   Aug11  0:01 qmgr -l -t fif
nobody   2301  0.0  0.0  67992  3960 ?        Ss   Aug11  0:07 proftpd: (acce
root     2339  0.0  0.7 438224 31208 ?       Ss   Aug11  0:12 /usr/sbin/apac
root     2368  0.0  0.1  43280  1324 lxc/consol Ss   Aug11  0:10 /sbin/getty
root     2370  0.0  0.0  12800  1896 lxc/tty2 Ss+  Aug11  0:00 /sbin/getty -8
root     2371  0.0  0.0  12800  1988 lxc/tty1 Ss+  Aug11  0:00 /sbin/getty -8
www-data 29512  0.0  0.2 438256 10780 ?        Ss   06:10  0:00 /usr/sbin/apac
www-data 29513  0.0  0.2 438256 10780 ?        Ss   06:10  0:00 /usr/sbin/apac
www-data 29514  0.0  0.2 438288 10780 ?        Ss   06:10  0:00 /usr/sbin/apac
www-data 29556  0.0  0.2 438256 10780 ?        Ss   06:10  0:00 /usr/sbin/apac
root     2957  0.0  0.2 438256 10780 ?        Ss   06:10  0:00 /usr/sbin/apac
postfix   30597  0.0  0.0  27420  2852 ?        Ss   14:43  0:00 pickup -l -t -f
root     30729  0.0  0.1  95116  6104 ?        Ss   15:53  0:00 shhd: root@pts
root     30748  0.0  0.1  19236  4372 pts/2 Ss   15:53  0:00 -bash
root     30769  0.0  0.0  15580  2112 pts/2 R+   15:53  0:00 ps aux
    
```

Gambar 14. Host Server Portal Website Fakultas

Berbeda dengan Gambar 13, Pada Gambar 14 ditunjukkan servis yang berjalan di *host server* yang menjalankan Sentora Control Panel untuk mengelola kebutuhan portal website fakultas dan website jurusan. Sehingga pada *host server* tersebut tidak ada servis java yang berjalan. Melalui kedua gambar tersebut dapat disimpulkan bahwa pembuatan *host server* dengan pilihan *container* (CT) tidak saling mempengaruhi CT lainnya. Hal ini menjadi isu penting ketika akan menjadikan sebuah server menjadi server utama virtualisasi [10]. Sehingga pemilihan CT pada semua *host server* telah sesuai dengan hasil penelitian [9][10].

Kemudian penelitian lainnya membandingkan teknologi virtualisasi Xen dan LXC (salah satu bentuk *container*) yang digunakan untuk menjalankan beberapa komponen aplikasi. Hasilnya kedua teknologi virtualisasi tersebut mampu menyediakan portabilitas, isolasi dan optimisasi sumber daya *hardware*, namun LXC membutuhkan sumber daya yang jauh lebih kecil [11][12].

## V. PENUTUP

### A. Kesimpulan

Teknologi virtualisasi server dengan menggunakan Sistem Operasi Proxmox 4.3 dapat digunakan untuk membangun jaringan server berbasis *cloud*. Untuk mencegah gagalnya proses *backup* dari server utama virtualisasi maka dibutuhkan lebih dari satu server FreeNAS sebagai cadangan bagi server FreeNAS lain. Pembuatan *host server* dengan pilihan *container* (CT) tidak saling mempengaruhi CT lainnya. Pilihan *container* (CT) sebagai *host server* dapat menjadi solusi yang cepat, hemat memori, dan handal untuk membangun kebutuhan server website fakultas dan jurusan. Kebutuhan server website fakultas dan jurusan yang masih dibawah penamaan subdomain yang sama (ft.unsoed.ac.id) dapat menggunakan *tools* pengelola *hosting* untuk mempermudah pengelolaannya.

### B. Saran

Berdasarkan hasil penelitian ini disarankan semua tatakelola server di Fakultas Teknik sudah dilaksanakan secara mandiri, tidak lagi terpusat di LPTSI, dengan menerapkan teknologi jaringan berbasis *cloud*. Dalam penelitian ini kinerja *host server* dengan menggunakan *container* (CT) belum sempat dibandingkan dengan kinerja *host server* yang menggunakan *virtual machine* (VM). Untuk penelitian yang akan datang, disarankan untuk mencari hasil pengujian yang lebih detail mengenai dasar pemilihan CT atau VM yang dibandingkan dengan kebutuhan *platform* aplikasi yang akan dijelankannya.

## UCAPAN TERIMA KASIH

Terimakasih kepada Universitas Jenderal Soedirman melalui Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) atas skim penelitian dosen pemula dana BLU 2017, sehingga penelitian ini dapat diselesaikan dan menghasilkan output sesuai dengan yang diharapkan.

## DAFTAR PUSTAKA

- [1]. Onno W Purbo. *Membuat Sendiri Cloud Computing Server Menggunakan Open Source*. Yogyakarta: Andi Publisher.
- [2]. Alfatih, H. dan R. Marco. *Analisis pengembangan dan perancangan sistem informasi akademik smart berbasis cloud computing pada sekolah menengah umum negeri (smun) di daerah istimewa yogyakarta*. Jurnal Telematika. 8(2):63–91.
- [3]. Onno W Purbo. *Petunjuk Praktis Cloud Computing Menggunakan Open Source*. <http://kambing.ui.ac.id/onnopurbo/ebook/ebookvoip/O-WP-20110701-petunjuk-praktis-cloudcomputing-menggunakan-opensource.pdf>. Diakses tanggal 1 Februari 2017.
- [4]. Asniar dan Siska Komala Sari. *Pemanfaatan Cloud Computing untuk Enterprise Resources Planning di Indonesia*. Jurnal Infotel Vol 7 No 1 (2015): Mei 2015.
- [5]. Iwan Sofana. *Cloud Computing Teori dan Praktik (OpenNebula, VMware, dan Amazon AWS)*. Bandung: Informatika.
- [6]. Ricky Chandra S, Arie S.M Lumenta, Arthur M. Rumagit. *Perancangan Servercloud Computing Menggunakan Proxmox*. E-journal Teknik Elektro dan Komputer (2014), ISSN : 2301-8402. Unsrat Manado.
- [7]. Edwar Ali, Didik Sudyana. *Virtualization Technology for Optimizing Server Resource Usage*. 3rd International Conference on Engineering and Technology Development. 2014. ISSN 2301-6590. Bandar Lampung University.
- [8]. Firmansyah Adiputra. "Container Dan Docker". *Teknik Virtualisasi Dalam Pengelolaan Banyak Aplikasi Web*. Jurnal Simantec, Vol. 4 No. 3 Juni 2015, ISSN 2088-2130.
- [9]. Budi Harijanto, Yuri Ariyanto. *Desain dan Analisis Kinerja Virtualisasi Server Menggunakan Proxmox VE*. Jurnal Simantec Vol.5 No.1. 2015. ISSN 2088-2130. Politeknik Negeri Malang.
- [10]. Dua, R., Raja, A. R., & Kakadia, D. *Virtualization vs Containerization to support PaaS*. International Conference on Cloud Engineering (pp. 610–614). 2014.
- [11]. Scheepers, M. J. *Virtualization and Containerization of Application Infrastructure: A Comparison*. 21st Twente Student Conference on IT (pp. 1–7). Enschede, The Netherlands: University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science.
- [12]. Sudha, M., Harish, G. M., & Usha, J. *Performance Analysis of Linux Containers - An Alternative Approach to Virtual Machines*. *Software Engineering*. 4(1), 820–824. International Journal of Advanced Research in Computer Science and Software Engineering, 4(1), 820–824.