

SERVICE AVAILABILITY DAN PERFORMA SUMBER DAYA PROSESSOR PADA INFRASTRUKTUR SERVER VIRTUAL

Dadan Irwan
Program Studi Teknik Komputer Universitas Islam "45"
Jln. Cut Meutia No. 83 Bekasi, Indonesia
Email: dadanirwan@gmail.com

ABSTRACT

Server that have high processor capacity and virtualization technology capabilities are generally not fully utilized optimally as they are only used to provide one application service. Virtualization is a technology that has the ability to empower the physical resources of a server to be built into multiple server machines virtually. In addition, the provision of the server is still using a single server, so that when a service failure caused by damage to its hardware or software components, the service availability to users will be disrupted. In this research will be developed the ability of server-based virtual server service and application of server failover as a technique in providing service availability. The method used in this research is through experimental approach and simulation is done by generating a number of request rate to the server. The process of testing and retrieving data is done by using several applications such as *httperf* generator, *sysstat*, and *free*. The result of this research is the percentage of processor utilization on native server, using average resource equal to 67.99%, while processor utilization on virtual server with 2 VM is about 43.73% average, so the processor level on the virtual server is lower than native server.

Keyword: *failback, failover, native server, service availability, virtual server*

ABSTRAK

Server yang memiliki kapasitas *processor* tinggi dan kemampuan teknologi virtualisasi umumnya belum sepenuhnya dimanfaatkan secara optimal karena hanya digunakan untuk menyediakan satu layanan aplikasi. Virtualisasi merupakan sebuah teknologi yang memiliki kemampuan dalam memberdayakan sumber daya fisik sebuah server untuk dibangun menjadi beberapa mesin server secara *virtual*. Selain itu, penyediaan server masih menggunakan server tunggal, sehingga ketika terjadi kegagalan layanan yang disebabkan adanya kerusakan pada komponen *hardware* atau *software*-nya, maka layanan ketersediaan (*service availability*) kepada *users* akan terganggu. Pada penelitian ini akan dikembangkan kemampuan layanan server berbasis server *virtual* dan penerapan *failover server* sebagai teknik dalam menyediakan ketersediaan layanan (*service availability*). Metode yang digunakan pada penelitian ini yaitu melalui pendekatan eksperimen dan dilakukan simulasi dengan membangkitkan sejumlah *request rate* ke server. Proses pengujian dan pengambilan data dilakukan dengan menggunakan beberapa aplikasi diantaranya *httperf* generator, *sysstat*, dan *free*. Hasil dari penelitian ini yaitu presentase utilisasi *processor* pada *native server*, menggunakan *resources* rata-rata sebesar 67.99%, sementara utilisasi *processor* pada *virtual server* dengan 2 VM sebesar rata-rata sekitar 43.73%, sehingga tingkat prosesor pada server virtual lebih rendah dari *native server*.

Kata kunci: *failback, failover, native server, service availability, virtual server*

1. PENDAHULUAN

Pertumbuhan teknologi informasi dan infrastrukturnya saat ini tumbuh sangat cepat. Penyebaran informasi dan data disajikan dalam layanan berbasis *web* sehingga dapat diakses secara *online*. Sementara itu, penyediaan informasi dan

data memerlukan berbagai aspek diantaranya perangkat keras, perangkat lunak, serta infrastruktur jaringan. Perangkat keras yang digunakan sebagai penyedia layanan utama adalah *server*. *Server* memiliki fungsi dalam menyediakan layanan seperti *web server*, *mail server*, *file server*, *database server* dan

layanan lainnya. Untuk membangun sebuah *server* diperlukan kapasitas mesin yang tinggi dan dukungan teknologi *processor multicore*, sehingga memiliki kemampuan layanan secara maksimal (Ardianto *et al.* 2012). *Server* dirancang untuk dapat bekerja secara terus-menerus tanpa mengalami gangguan (Calzolari *et al.* 2010). Menurut Lwin dan Thein (2009), perusahaan dan pelaku bisnis sangat membutuhkan ketersediaan layanan aplikasi dan informasi setiap hari yang mampu memproteksi data dari berbagai gangguan.

Sebuah mesin *server* atau *native server* yang memiliki kapasitas tinggi dan kemampuan teknologi virtualisasi dapat dimanfaatkan untuk memberikan lebih dari satu layanan. Teknologi virtualisasi yang ditanamkan pada setiap mesin *server* dapat digunakan dan dimanfaatkan untuk dibangun beberapa mesin *server* secara *virtual* yang memberikan layanan berbeda. Virtualisasi merupakan sebuah teknologi yang bertujuan memberdayakan sumber daya fisik sebuah *server* untuk dibangun menjadi beberapa mesin *server* secara *virtual* (Afriandi 2012). Menurut Tseng *at al.* (2013), kemampuan kinerja mesin *server* secara *virtual* pada sebuah *native server* tergantung dari kapasitas mesin fisik *server*-nya. Oleh karena itu, perlu dilakukan pengukuran performa utilisasi *processor* dan *memory* pada mesin *server* tersebut. Pengukuran sangat penting dilakukan untuk mengetahui performansi dari *server* agar layanan kepada *users* tetap stabil. Pengukuran dilakukan ketika mesin *server* digunakan secara *native*

dan setelah dibangun beberapa mesin *server* secara *virtual*.

Beberapa penelitian terkait penerapan teknologi virtualisasi telah dilakukan dengan beberapa pendekatan. Pendekatan yang dilakukan oleh Rahma (2012), yaitu membangun sistem *cloud computing* menggunakan teknologi virtualisasi dengan memanfaatkan empat unit mesin *server* berbasis *clustering*. Namun belum diterapkan teknologi *failover server* sebagai solusi *service availability* ketika terjadi kegagalan layanan pada mesin-mesin *server* tersebut. Pendekatan lainnya dilakukan oleh Mardiyanto (2013), untuk mengatasi kegagalan layanan sistem informasi menggunakan virtualisasi yang memanfaatkan teknologi *clustering server* dan *load balancing*. Namun belum diterapkan pemanfaatan teknologi *failover server* untuk mengatasi kegagalan layanan pada *server* utama (*main server*).

Sementara itu, teknologi virtualisasi umumnya diterapkan terpusat pada satu mesin *server*, sehingga ketika terjadi kerusakan pada *server* tersebut, maka layanan kepada *users* akan terganggu (Ngilu *et al.* 2013). Gangguan layanan pada mesin *server* dapat disebabkan oleh kerusakan *hardware* atau *software*-nya. Untuk mengatasi kegagalan ketersediaan layanan (*service availability*) dari sebuah *server* perlu dibangun sebuah mesin *server* lain yang bertindak sebagai cadangan (*backup server*), sehingga layanan kepada *user* tetap berjalan. Teknologi *failover server* adalah sebuah teknologi yang digunakan untuk

mengatasi kegagalan *service availability* pada sebuah *server* dengan cara menggantikan peran *server* utama dengan sebuah *server* cadangan (Mughtar, 2012). Menurut Hirt (2009) teknologi *failover server* bertujuan untuk membantu menjaga akses *users* ke aplikasi dan sumber daya *server* ketika terjadi kegagalan fungsi sebuah *server* yang mengakibatkan *server* berhenti bekerja.

Berdasarkan paparan di atas, dapat disimpulkan bahwa mesin *server* yang memiliki kapasitas tinggi dengan dukungan teknologi virtualisasi dapat dikembangkan menjadi mesin *server* yang mampu memberikan beberapa layanan berbeda dengan perfomansi yang tetap stabil. Sementara itu, dengan kemampuan teknologi *failover server*, ketika *main server* mengalami kegagalan layanan maka *backup server* akan mengambil alih fungsi *main server*. Pada penelitian ini akan dilakukan pengujian kemampuan *service availability* dengan teknologi *failover server* serta utilisasi *procesor* pada *native server* dan

server virtual. Skenario yang digunakan untuk mengetahui kemampuan *service availability* yaitu dengan melakukan sejumlah *request rate* pada *native server* dan *backup server* dan dilakukan pengambilan data secara bertahap dan terpisah. Jumlah *request rate* akan di-generate dengan jumlah yang bervariasi menggunakan aplikasi tertentu.

2. BAHAN DAN METODE

2.1 Bahan

Pada tahapan ini dilakukan identifikasi dan analisis kebutuhan yang terkait dengan penggunaan kebutuhan perangkat keras yang akan digunakan untuk membangun server baik *native server* maupun *virtual server*. Berdasarkan identifikasi dan analisis yang dilakukan, maka pada penelitian ini diperlukan beberapa perangkat keras dan perangkat lunak seperti disajikan pada Tabel 1 dan Tabel 2.

2.2 Metode

Tahapan penelitian yang dilakukan disajikan pada Gambar 1.

Tabel 1 Kebutuhan perangkat keras

| No | Jenis perangkat keras | Jumlah | Keterangan |
|----|------------------------------|--------|--------------------------------|
| 1 | Server | 2 unit | Tidak identik |
| 2 | Switch | 2 unit | - |
| 3 | Harddisk Drive (HDD) | 4 unit | HDD kedua memiliki ukuran sama |
| 4 | Network Interface Card (NIC) | 4 unit | - |
| 5 | Komputer client | 2 unit | - |
| 6 | Router | 1 unit | - |
| 7 | Modem ADSL | 1 unit | - |

Tabel 2 Kebutuhan perangkat lunak

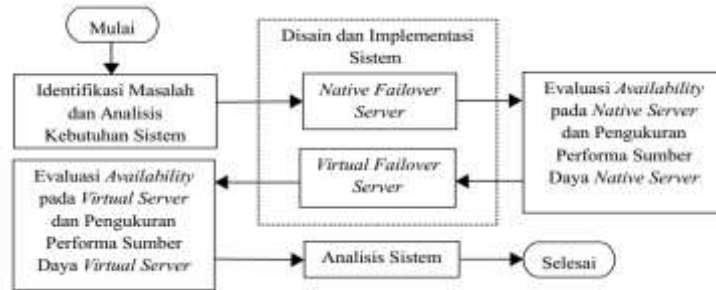
| No | Jenis perangkat lunak | Nama perangkat lunak | Keterangan |
|----|----------------------------|-------------------------|------------------|
| 1 | Sistem Operasi | Turnkey Lamp | |
| 2 | Aplikasi Web Server | Apache2, PHP, Mysql | |
| 3 | Aplikasi Hypervisor | ProxmoxVE versi 3.4 | Free open source |
| 4 | Aplikasi Clustering | Heartbeat | |
| 5 | Aplikasi sinkronisasi data | DRBDVersi 8.4.5 | |
| 6 | Tools Benchmarking | Httpperf, Sysstat, Free | |

3. HASIL DAN PEMBAHASAN

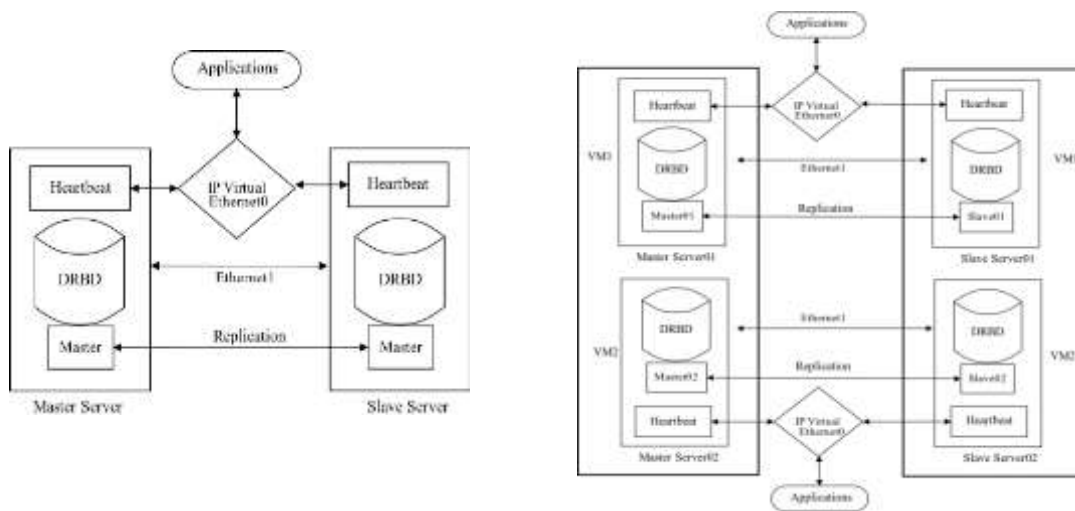
3.1 Arsitektur Sistem

Pada tahapan ini didisain dua jenis arsitektur yang berbeda, arsitektur pertama

yaitu disain sistem pada *native server* dan tahapan kedua yaitu arsitektur ke-2 pada *virtual server*. Pada tahap pertama dibangun arsitektur sistem seperti pada Gambar 2.



Gambar 1 Tahapan Penelitian

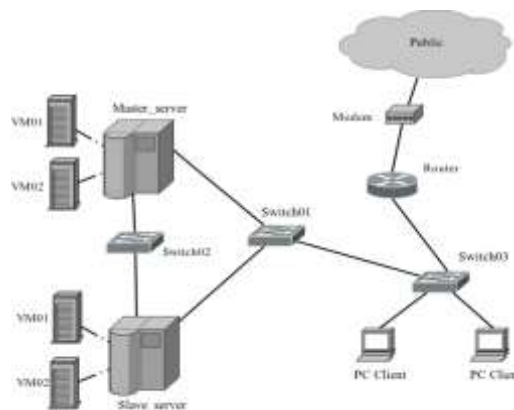


Gambar 2 Arsitektur Sistem

3.2 Disain Topologi

Disain topologi diperlukan untuk mengetahui kebutuhan perangkat jaringan yang akan digunakan dan agar sistem dapat diakses oleh *client* baik pada jaringan

internal maupun jaringan publik. Disain topologi yang dibangun untuk *virtual server cluster* dengan masing-masing terdapat dua *virtual machine* (VM) ditampilkan pada Gambar 3.



Gambar 3 Disain Topologi Jaringan

3.3 Rancangan dan Implementasi Sistem

Pada tahapan ini dilakukan rancangan dan implementasi kedua model server yaitu model server pertama adalah server yang *non virtual* dan server kedua adalah dengan menggunakan virtualisasi penuh (*full virtualization*). Tabel 3 dan Tabel 4 merupakan rancangan sistem pada aspek *hardware* dan *software*

3.4 Pengujian Sistem

Proses pengujian dilakukan untuk memperoleh pembuktian serta solusi dari permasalahan yang sudah ditetapkan

sebelumnya melalui metodologi dan tahapan-tahapan yang dilakukan. Dalam proses ini diperlukan beberapa skenario pengujian dan pembebanan berupa simulasi secara terencana (*planned scenario*) dan dalam implementasinya digunakan aplikasi *benchmark tools* berbasis *open source* seperti ditunjukkan pada Tabel 5. Skenario aktivitas *request users* terhadap layanan server di-generate menggunakan aplikasi *Httpperf generator* dengan jumlah koneksi dan *request rate* tertentu. .

Tabel 3 Rancangan logik *virtual machine* untuk kedua server

| No | Spesifikasi logik perangkat keras | Server utama | | Server cadangan | |
|----|------------------------------------|--------------|--------|-----------------|--------|
| | | VM01 | VM02 | VM01 | VM02 |
| 1 | <i>Processor</i> | 1 core | 1 core | 1 core | 1 core |
| 2 | <i>Memory (RAM)</i> | 512 Mb | 512 Mb | 512 Mb | 512 Mb |
| 3 | <i>Harddisk Drive 01</i> | 50 Gb | 50 Gb | 50 Gb | 50 Gb |
| 4 | <i>Harddisk Drive 02</i> | 10 Gb | 10 Gb | 10 Gb | 10 Gb |
| 5 | <i>Network Interface Card(NIC)</i> | Eth0 | Eth1 | Eth0 | Eth1 |

Tabel 4 Sistem layanan pada server

| No | Layanan server | Server utama | | Server cadangan | |
|----|------------------------|---------------------|------|---------------------|------|
| | | VM01 | VM02 | VM01 | VM02 |
| 1 | Sistem Operasi | <i>Turnkey Lamp</i> | | <i>Turnkey Lamp</i> | |
| 2 | <i>Failover</i> | <i>Heartbeat</i> | | <i>Heartbeat</i> | |
| 3 | <i>Clustering</i> | DRBD | | DRBD | |
| 4 | <i>Web server</i> | <i>Apache2</i> | | <i>Apache2</i> | |
| 5 | <i>Database server</i> | <i>Mysql</i> | | <i>Mysql</i> | |
| 6 | Konten Website | Ya | | Tidak | |

Tabel 5 Jenis pengujian dan skenario yang dilakukan

| No | Jenis pengujian | Skenario pengujian | Server yang diuji |
|----|--|---|---|
| 1 | Pengujian <i>failover</i> dan <i>failback</i> | Melakukan penonaktifan dan pengaktifan layanan <i>Heartbeat</i> | <i>Native server dan virtual server</i> |
| 2 | Pengujian utilisasi sumber daya <i>processor</i> dan <i>memory</i> . | Melakukan pembebanan pada server dengan mengirim jumlah koneksi maksimal 4000 dan <i>request rate</i> antara 100-1000 dengan interval 100 koneksi | |

3.5 Pengukuran dan Pengambilan Data

Proses pengukuran dan pengambilan data baik pada proses *failover*, *failback*, serta untuk mengetahui utilisasi sumber daya

processor dan *memory* dilakukan dengan melakukan uji pembebanan seperti pada Tabel 5. Langkah yang dilakukan untuk pengambilan data pada proses *failover* dan

failback dapat dilihat pada Tabel 6. Nilai akhir diperoleh dengan menggunakan Persamaan 1 untuk perhitungan nilai rata-rata pada setiap proses pengambilan data

tersebut. Data yang dihasilkan dari masing-masing proses pengujian disimpan dalam sebuah *file* yang memiliki ekstensi *.csv dan *.txt.

Tabel 6 Pengambilan data pada proses *failover* dan *failback*

| No | Jenis pengukuran dan pengambilan data | Skenario pembebanan |
|----|--|---|
| 1 | Proses <i>Failover</i> dan <i>failback</i> | Melakukan pembebanan pada server dengan mengirim jumlah koneksi maksimal 2000 dan <i>request rate</i> antara 10-50 dengan interval 10 koneksi, setiap pengujian dilakukan pengulangan sebanyak 5 kali (lihat Lampiran 3 dan 4) |
| 2 | Utilisasi <i>processor</i> dan <i>memory</i> | Dilakukan proses pengujian dengan skenario pada Tabel 11 yang dilakukan dari <i>client</i> , pada waktu yang sama dilakukan pengukuran utilisasi <i>processor</i> menggunakan aplikasi Sysstat dan pengukuran <i>memory</i> menggunakan aplikasi Free yang dijalankan pada server (lihat Lampiran 5 dan 6). |

3.7 Analisis hasil Pengujian

Processor merupakan komponen utama pada *server* yang berfungsi untuk memproses data. Pada komputer *server*, komponen tersebut memberikan pengaruh dalam proses kecepatan layanan pada *client*, semakin tinggi spesifikasi *processor* maka semakin cepat juga proses yang dieksekusinya. Secara spesifik penggunaan *processor* oleh sistem pada komputer selain digunakan untuk aplikasi berbasis *user* juga digunakan untuk *system* sehingga perlu diketahui berapa total utilisasi *processor* pada saat memberikan layanan kepada *client*.

Pada proses pengukuran utilisasi *processor* dilakukan bersamaan dengan pengukuran utilisasi pada *memory*, kedua proses tersebut dilakukan pada mesin *server* dengan menggunakan aplikasi Sysstat untuk *processor* dan aplikasi Free untuk *memory*. Pengukuran dilakukan dengan menentukan interval waktu penggunaan kedua komponen setiap detik dan langsung disimpan dalam

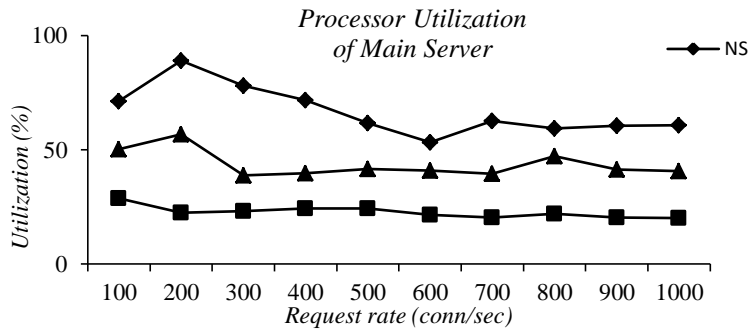
tipe *file* *.csv, kemudian data yang dihasilkan dari pengukuran dihitung nilai rata-rata untuk setiap 10 kali pengulangan. Utilisasi *processor* pada *server* utama disajikan pada Gambar 4.

Pengujian pada kedua jenis *server* diperlukan untuk mengukur seberapa besar utilisasi *processor* yang digunakan oleh sistem dalam mengeksekusi berbagai layanan aplikasi ditingkat *user* dan *system*. Skenario pengujian dilakukan mengacu pada Tabel 5 mengenai uji pembebanan kinerja sumber daya menggunakan aplikasi Httpperf dengan proses pengulangan sebanyak 10 kali.

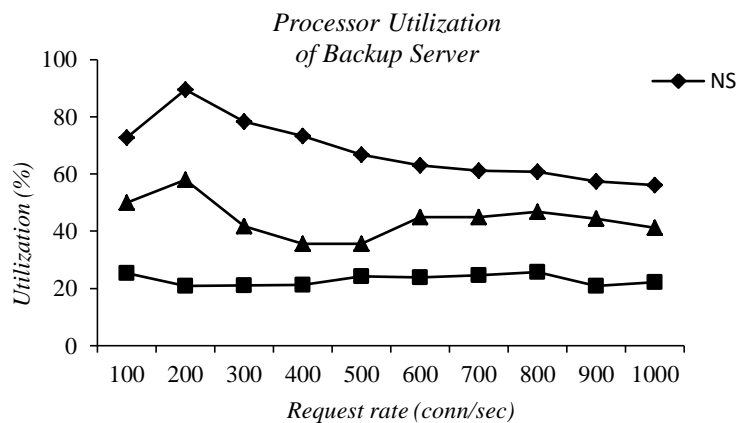
Beban tertinggi utilisasi *processor* *server* utama pada *native server* adalah 89.08%, sedangkan saat menjalankan kedua mesin *virtual* pada *virtual server* adalah 58.80%. Pada *server* cadangan beban tertinggi untuk *native server* adalah 89.05% sedangkan saat menjalankan kedua mesin *virtual* pada *virtual server* adalah 76.81%. Utilisasi *processor* pada *native server*

menggunakan *resources* rata-rata sebesar 68.08%, sedangkan utilisasi *processor* pada *virtual server* dengan beban 1 mesin virtual rata-rata 22.78% dan 2 mesin virtual 43.73%. Pada *server* cadangan utilisasi

processor untuk *native server* rata-rata 67.91% dan pada *virtual server* dengan beban 1 mesin virtual sebesar 35% dan 2 mesin virtual sebesar 51.33%.



Gambar 4 Utilisasi *processor* pada *server* utama



Gambar 5 Utilisasi *processor* pada *server* cadangan

Data utilisasi *processor* pada *native server* cenderung lebih tinggi dibandingkan dengan utilisasi pada *virtual server*. Dengan demikian, tingkat kesibukan sumber daya *processor* pada *virtual server* lebih rendah dibandingkan dengan *native server*, sehingga nilai efisiensi pemakaian *server* lebih tinggi. Salah satu penyebab tingginya utilisasi pada *native server* yaitu sistem operasi dipasang langsung di atas *hardware*, sehingga terdapat banyak aplikasi yang

berjalan dan menggunakan banyak *resources* pada *server* tersebut.

Sementara itu, tingginya utilisasi *processor* pada *rr* yang rendah disebabkan durasi waktu untuk memproses *rr* yang rendah memerlukan waktu yang lebih lama dibandingkan dengan *rr* yang lebih tinggi. Sebagai contoh, ketika jumlah *rr*=200 koneksi/detik dan jumlah total 4000 koneksi, maka *server* memerlukan waktu proses sekitar 20 detik. Sementara itu, ketika jumlah *rr*=1000 koneksi/detik dan jumlah

total 4000 koneksi, maka *server* memerlukan waktu proses sekitar 4 detik.

4 KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan permasalahan, tahapan-tahapan proses dan hasil dari beberapa pengujian yang dilakukan dengan menerapkan konsep dan metode yang telah ditetapkan, maka penulis mengambil beberapa simpulan sebagai berikut:

1. Utilisasi *processor* pada *native server*, menggunakan *resources* rata-rata sebesar 67.99%, sementara utilisasi *processor* pada *virtual server* dengan 2 mesin virtual sebesar rata-rata 43.73%, sehingga tingkat kesibukan *processor* pada *virtual server* lebih rendah dibandingkan dengan *native server*.
2. Berdasarkan hasil penelitian mengenai tingkat *service availability* dan utilisasi sumber daya pada *virtual server*, maka virtualisasi dengan model *hypervisor* (*full virtualization*) merupakan teknik yang dapat digunakan untuk mengoptimalkan sumber daya *server* dengan dukungan kapasitas perangkat jaringan *Gigabit ethernet*.

4.2 Saran

Perlu dilakukan proses pengujian *failover* dan *failback* pada kedua mesin virtual yang dijalankan secara bersamaan dengan penggunaan NIC yang berbeda pada setiap *server*. Sehingga dapat diketahui *service availability* pada *virtual server* dengan menjalankan 2 buah layanan yang berbeda.

Daftar Pustaka

- Afriandi A. 2012. Perancangan, Implementasi, dan Analisis Kinerja Virtualisasi *Server* Menggunakan Proxmox, VMWare ESX, dan OpenStack. [Thesis]. Magister Teknologi Informasi. Universitas Gajah Mada. Yogyakarta.
- Ardianto N, Widyawan, Sumaryono N. 2012. Pengembangan *Virtual Appliance Server* dengan Metode Virtualisasi. JNTETI. Vol 1 No. 1.
- Calzolari, Federico; Arezzini, Silvia; Ciampa, Alberto eds. *High Availability Using Virtualization*. IOP Science, 2010.
- Lwin TT. dan Thein T. 2009. High Service Availability Cluster System for Local Disaster Recovery with Markov Modeling Approach. *International Journal of Computer Science Issues* (IJCSI), Vol. 6, No. 2.
- Margaretha M, Marini, Tjiptadi R. 2012. Performa Koneksi Menggunakan Virtualisasi dengan Proxmox Pada IPTEKNET. Binus University.
- Muchtar A, Sadjad R, Niswar M. *Implementasi Failover Clustering Pada Dua Platform yang Berbeda untuk Mengatasi Kegagalan Fungsi Server*. Fakultas Teknik Universitas Hasanuddin.
- Ngilu FS, Sujoko S, Adji TB. 2013. Overhead Analysis as One Factor Scalability of Private Cloud Computing for IAAS Service. *International Journal of Scientific & Engineering Research* (IJSER), Vol. 4. No. 5. May 2013.
- Nurhaida I. 2009. Pengukuran Overhead, Linearitas, Isolasi Kinerja dan

Penggunaan Sumber Daya Perangkat Keras pada *Server* Virtual. [Thesis].Fakultas Teknik Program Studi Teknik Elektro Program Pasca Sarjana Bidang Ilmu Teknik Universitas Indonesia.Depok.

Rahma F, Adji TB, Widyawan. 2013. Scalability Analysis of KVM-Based Private Cloud For IaaS. *International Journal of Cloud Computing and Services Science (IJ-CLOSER)*. Vol. 2.No. 4. pp. 288-295.