

# IMPLEMENTASI DAN ANALISIS *PLATFORM HORIZON* DALAM *SERVICE DASHBOARD* BERBASIS OPENSTACK

## IMPLEMENTATION AND ANALYSIS OF *HORIZON PLATFORM* IN *DASHBOARD SERVICE* BASED ON OPENSTACK

Devy Febri Fetria<sup>1</sup>, Tody Ariefianto Wibowo<sup>2</sup>, Joko Purnomo<sup>3</sup>

<sup>1,2</sup>Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

<sup>3</sup>Senior Engineering, Ericsson Corp.

[devfetria@gmail.com](mailto:devfetria@gmail.com), [tody.wibowo@gmail.com](mailto:tody.wibowo@gmail.com), [joko.purnomo@ericsson.com](mailto:joko.purnomo@ericsson.com)

---

**Abstrak** - *Openstack* merupakan sebuah satuan kontrol dalam SDN (*Software Define Network*) yang dapat menyatukan beberapa blok layanan ke dalam satu kendali. Beberapa blok layanan tersebut antara lain *Service Application*, *Dashboard*, *Compute*, *Storage* dan *Network*. Untuk menunjang layanan – layanan tersebut dibutuhkan suatu sistem visualisasi dalam memudahkan *user* dalam melakukan pengkonfigurasi jaringan agar lebih mudah. Sistem ini dinamakan *Dashboard* yang berada dalam *platform Horizon* yang merupakan sebuah *web interface* bagi *user* untuk membuat jaringan baru, melakukan konfigurasi, melakukan pemeliharaan dan pengelolaan perangkat jaringan atau dapat dikatakan sebagai *cloud administrators*.

Dari hasil analisis servis *Dashboard* pada tugas akhir ini dapat diambil kesimpulan bahwa kinerja dari servis *Dashboard* dapat dikatakan cukup baik. Dari hasil pengujiannya didapatkan kesimpulan dengan pengujian *Click bandwidth* maksimum pada server adalah 1100 Kbps dan pada *user* adalah 1500 Kbps, untuk pengujian *Time bandwidth* maksimum pada server adalah 2200 Kbps dan pada *user* adalah 1100 Kbps, dan untuk pengujian *Ramp bandwidth* maksimum pada server adalah 2200 Kbps dan pada *user* adalah 1100 Kbps.

**Kata kunci** : *OpenStack*, *Dashboard*, *Horizon*, *Cloud Computing*.

---

**Abstract** – *OpenStack* is a main control of SDN (*Software Define Network*) which able to combining several blocks into one control. There are *Application Service*, *Dashboard Service*, *Compute Service*, *Storage Service*, and *Service Network*. For supporting the services, it needs a visualization system for facilitate user to do the networking configuration more easier. This system is named *Dashboard* in *Horizon platform* which is a user web interface for creating a new network, do some maintenance and managing the network devices or its called *cloud administrators*. From the result of *Dashboard* service analysis in this final project, it could be deduced that the performace from its service is good enough. From the testing obtained with *Click* testing the maximum bandwidth in server is 300 kbps and in user side is 1200 kbps, for *Time* testing the maximum bandwidth in server is 650 kbps and in user side is 2200 kbps, and for *Ramp* testing the maximum bandwidth in server is 2400 Kbps and in user side is 1200 Kbps.

**Keywords** : *Openstack*, *Dashboard*, *Horizon*, *Cloud Computing*.

---

### 1. PENDAHULUAN

Dewasa ini semua tidak lepas dari teknologi yang membuat kebutuhan manusia akan dunia teknologi dan informasi semakin meningkat, terutama dalam hal perkembangan dunia jaringan (*Network*) telekomunikasi yang berbasis *Internet Protocol* (IP). Dengan semakin banyaknya pengaplikasian teknologi itu, maka semakin dibutuhkan banyak perangkat dengan pengkonfigurasi yang berbeda-beda untuk mendukung terealisasinya infrastruktur jaringan yang sangat luas tersebut, hal tersebut dikarenakan untuk melakukan pengaturan perangkat tersebut kita harus melakukannya satu per satu dan akan memakan waktu yang lama untuk infrastruktur yang begitu besar dan luas. *Openstack* merupakan sebuah satuan kontrol dalam SDN (*Software Define Network*) yang dapat menyatukan beberapa blok layanan ke dalam satu kendali. Beberapa blok layanan tersebut antara lain *Service Application*, *Dashboard*, *Compute*, *Storage* dan *Network*. Untuk menunjang layanan – layanan tersebut dibutuhkan suatu sistem visualisasi dalam memudahkan *user* dalam melakukan pengkonfigurasi jaringan agar lebih mudah. Sistem ini dinamakan *Dashboard* yang berada dalam *platform Horizon* yang

merupakan sebuah *web interface* bagi *user* untuk membuat jaringan baru, melakukan konfigurasi, melakukan pemeliharaan dan pengelolaan perangkat jaringan atau dapat dikatakan sebagai *cloud administrators*.

## 2. DASAR TEORI

### 2.1 Cloud Computing

Secara umum, definisi *cloud computing* (komputasi awan) merupakan gabungan pemanfaatan teknologi komputer (komputasi) dalam suatu jaringan dengan pengembangan berbasis internet (awan) yang mempunyai fungsi untuk menjalankan program atau aplikasi melalui komputer – komputer yang terkoneksi pada waktu yang sama, tetapi tak semua yang terkoneksi melalui internet menggunakan *cloud computing*. Teknologi komputer berbasis sistem *cloud* ini merupakan sebuah teknologi yang menjadikan internet sebagai pusat server untuk mengelola data dan juga aplikasi pengguna. Teknologi ini mengizinkan para pengguna untuk menjalankan program tanpa instalasi dan mengizinkan pengguna untuk mengakses data pribadi mereka melalui komputer dengan akses internet<sup>[1]</sup>. *cloud computing* mempunyai 3 tingkatan layanan yang diberikan kepada pengguna, yaitu :

- SaaS (*Software as a Service*) : Layanannya berupa *software* siap pakai.
- PaaS (*Platform as a Service*) : Menawarkan layanan berupa platform-platform untuk membuat aplikasi
- IaaS (*Infrastructure as a Service*) : Menyediakan layanan sumber daya komputasi seperti prosesor, memori, dan *storage* yang sudah tervirtualisasi.

### 2.2 Openstack

*Openstack* adalah sebuah *platform* awan yang terdiri dari *software open source* untuk menyediakan basis menjalankan *cloud* IaaS (*Infrastructure as a Service*), baik pribadi maupun yaitu berupa sumber daya untuk komputasi dan penyimpanan data dalam bentuk mesin virtual. *Openstack* mempunyai kemampuan skalabilitas yang lebih besar dibandingkan kerangka kerja awan lainnya. Tujuan *Openstack* adalah untuk memungkinkan setiap organisasi atau perusahaan membuat dan menyediakan layanan komputasi awal dengan menggunakan perangkat lunak *opensource* yang berjalan di atas perangkat keras yang standar. Di dalam *Openstack* terdapat beberapa komponen yang memudahkan pengguna dalam melakukan konfigurasi infrastruktur *cloud computing* ini, di antaranya adalah: *Openstack Compute*, *Openstack Storage*, dan *Openstack Image Service*. *Openstack* memiliki keunggulan dengan banyaknya jumlah *instance* atau *virtual machine* yang dapat dibuat. Diperlukan alokasi sumber daya (*bandwidth*, memori, CPU (*Central Processing Unit*), *storage*) di awal pembuatan sebuah server *instance* yang disesuaikan dengan kebutuhan. Tetapi alokasi sumber daya seperti itu dirasa kurang maksimal, karena komputasi dari suatu *instance* bisa saja berubah-ubah sesuai tugas yang diberikan secara *real-time*.<sup>[6]</sup>

### 2.3 OpenStack Dashboard

*Openstack Dashboard* merupakan suatu layanan *user interface* dalam infrastruktur *Openstack* yang memberikan akses visualisasi bagi *user* dalam menciptakan *cloud*. Akan tetapi, agar sistem ini dapat berfungsi sepenuhnya dengan baik, beberapa platform pendukung sistem *cloud Openstack* seperti *Nova*, *Swift*, *Glance*, *Keystone*, *Neutron* dan *Cinder* ini harus sudah terinstal sepenuhnya terlebih dahulu dalam server. *Openstack Dashboard* ini didesain dengan sangat fleksibel sehingga mudah untuk disesuaikan dengan produk atau jenis layanan lainnya, seperti bisa ditambahkannya layanan *monitoring*, *maintenance* dll. Pada layanan *Dashboard* ini *user* dapat mendapatkan beberapa kemudahan dalam menggunakan *Openstack* ini, diantaranya :

1. Informasi pemakaian dan quota *cloud*.
2. Dapat dengan mudah mengoperasikan *cloud virtual machine*.
3. Mengelola volume untuk control pembuatan, penghapusan, dan konektivitas ke Block Storage.
4. Menjamin akses dan keamanan sistem *cloud*.<sup>[9]</sup>

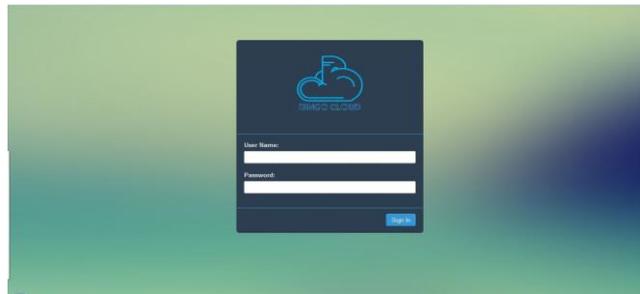
## 3. PERANCANGAN SISTEM

### 3.1 Blok Diagram Sistem

Dibawah ini merupakan gambar arsitektur secara keseluruhan dari *Openstack* yang terdiri dari beberapa blok layanan dengan fungsi kerja masing-masing, yaitu :

- *Dashboard* : Sebagai *user interface*.
- *Network* : Jaringan yang berfungsi sebagai jembatan semua layanan pada *Openstack*.
- *Compute* : Sistem yang mengatur dan mengontrol layanan-layanan pada *Openstack*.
- *Storage* : Menyediakan media penyimpanan.
- *Identify* : Untuk keamanan data bagi *user*.<sup>[10]</sup>



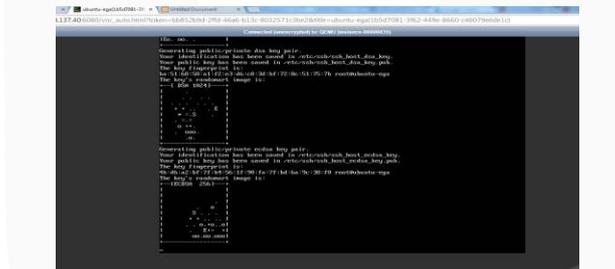


**Gambar 4** Halaman *Login* Dashboard Openstack



**Gambar 5** Halaman utama Dashboard Openstack

Dari desain pada halaman utama admin tersebut, diharapkan proses monitoring keseluruhan sistem dapat dipahami dan dilakukan dengan mudah oleh *user admin*. Selain itu fitur lain dari *Openstack* Dashboard ini adalah dengan penggunaan konsol VNC (*Virtual Network Computing*) yang digunakan untuk melakukan pengaksesan pada *cloud computing* Openstack ini secara langsung oleh setiap *user* secara manual. Berikut adalah tampilan konsol VNC yang berada didalam menu *Instance Dashboard*.



**Gambar 6** Tampilan konsol VNC pada Dashboard Openstack

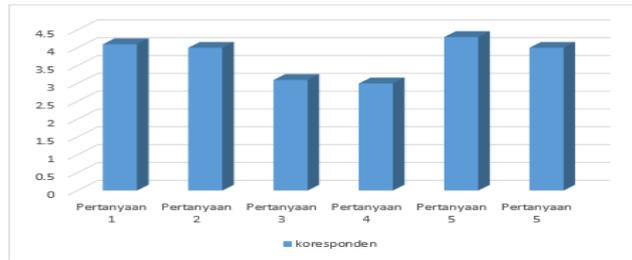
Pada gambar 6 diatas menunjukkan bahwa konsol tersebut sedang melakukan *generating* IP untuk pengaksesan ke servernya, dimana ketika pengaksesan tersebut berhasil maka *user* dapat melakukan konfigurasi virtual pada server Openstack dengan hak akun yang dimiliki oleh *user* tersebut.

#### 4. PENGUJIAN DAN ANALISIS

Pengujian dan analisis pada *service* dashboard dilakukan dengan tujuan untuk mengetahui performansi *service* tersebut dengan *platform* horizon. Dengan dilakukan pengujian pada *platform* tersebut dapat diketahui apakah *platform* dapat bekerja dengan baik atau benar. Untuk pengujian, yang menjadi acuan dasarnya adalah pengujian pada bagian *user interface* dari layanan Openstack ini, yaitu pada Desain UI, UX, lalu kinerja *website* seperti : kecepatan pengaksesan tiap halaman, beban penggunaan *website* oleh banyak *user*.

##### 4.1 Pengujian Desain UI (*User Interface*)

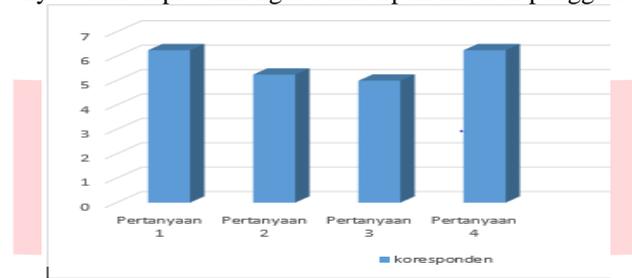
Pada pengujian UI (*User Interface*) ini yang menjadi acuan pengujian adalah menitik beratkan kepada desain yang ditampilkan pada layanan *website* Openstack ini. Karena UI (*User Interface*) merupakan mekanisme komunikasi antara pengguna dengan sistem.



Gambar 7 Grafik hasil penilaian UI

#### 4.2 Pengujian Desain UX (User Experience)

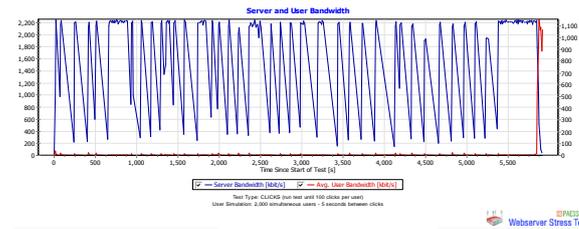
Pada pengujian UX (*User Experience*) ini yang menjadi acuan dalam pengujiannya adalah menitik beratkan pada pengalaman pengguna layanan *website* Openstack dalam menggunakannya, apakah setiap menu dan eksekusi ke aksi berikutnya sudah dapat dimengerti dan dipahami oleh pengguna atau tidak.



Gambar 8 Grafik hasil penilaian UX

#### 4.3 Pengujian Kecepatan Pengaksesan Halaman (Skema Click)

- Pengujian (2000 users 100 kali klik), click delay → 5 detik

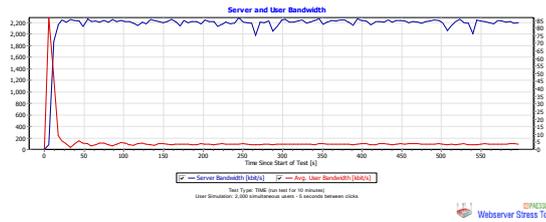


Gambar 9 Grafik bandwidth server dan user 2000 user

Pada pengujian 2000 user yang melakukan 100 kali klik dengan delay maksimal 5 detik, sehingga pada pengujian ini didapat kan suatu analisis bahwa *bandwith user* didapat 90 sampai 100 Kbps dan *bandwith server* rata – rata yang digunakan 1598,52 Kbps. *Bandwith* yang digunakan server rata – rata sebesar 1598,52 Kbps dengan pengambilan sampel tiap 125 detik dan pada pengujian dengan 2000 user ini terdapat error dengan nilai rata – rata 414,708 error sehingga penyebab terjadinya *ripple* pada grafik bandwidth server ini adalah karena jumlah pengakses dan error tersebut. Contoh sampel terjadinya error yang menyebabkan *ripple* pada grafik diatas adalah pada detik 249 sampai 259 (interval 10 detik) terjadi error sebanyak 991 error dengan presentase 54,90 % sehingga terjadi penurunan *bandwidth* server mendekati 200 kbps. Error tersebut disebabkan oleh adanya paket yang error atau kegagalan pada saat mengakses halaman URL yang sudah didaftarkan. *Ripple* pada *bandwidth user* dalam percobaan 2000 pengakses tidak tampak dikarenakan hasil akumulasi dari jumlah rata-rata waktu akses per user menghasilkan kecepatan pengaksesan yang sangat kecil dan memiliki rentang yang tidak terlalu berbeda.

#### 4.4 Pengujian Beban Penggunaan layanan Website Dashboard (Skema Time)

- Pengujian (2000 user dengan click delay maksimum 5 detik) per user



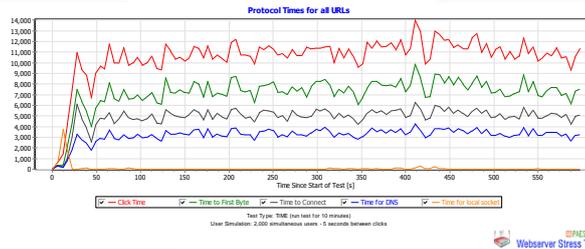
**Gambar 9** Grafik bandwidth server dan user 200 user

Dari grafik pada gambar 9 diatas dapat diamati bahwa :

1. *Bandwidth user* yang ditunjukkan pada grafik diatas bahwa *bandwidth* maksimal yang diperoleh 85 Kbps dengan rata – rata *bandwidth user* adalah 1,3510 kbps.
2. *Bandwidth* yang digunakan oleh server berdasarkan dari grafik diatas berkisar antara 2200 sampai 2400 Kbps.

Hal ini dikarenakan *bandwith user* didapat dari *packet outgoing* yang bukan hanya *request* tetapi diberi beban oleh aplikasi web stress tool, dan di rekam bahwa *packet outgoing* yang memenuhi jaringan dan yang didapat sebesar 85 Kbps. Sedangkan pada *Bandwith server* didapat 2200 Kbps dikarenakan *packet incoming* dan beban dari aplikasi web stress tool memadati trafik di server sehingga terekam 2200 Kbps untuk *bandwidth* maksimumnya.

- Pengujian (2000 user dengan *click delay* maksimum 5 detik) per URL

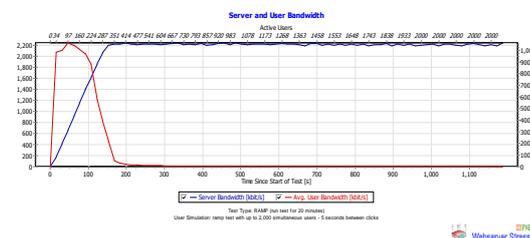


**Gambar 10** Grafik protokol pengaksesan oleh 2000 user

Pada gambar 10 dapat dianalisis bahwa, Waktu pengaksesan untuk melakukan *request* dengan inialisasi DNS setiap URL dari saat waktu awal dimulainya pengujian hingga akhir yaitu rata – rata 3 - 4 detik. Selanjutnya dari inialisasi DNS itu di si server membuka port IP bagi *user* yang melakukan *request* tersebut sehingga dapat mengakses DNS tersebut dengan waktu rata – rata 6 detik, sehingga *request* dari *user* tersebut dijawab oleh server dengan grafik berwarna hijau yaitu *Time to first byte* dengan rata – rata kecepatan waktu pengaksesan yaitu 7 - 10 detik. *Click time* adalah saat semua proses *request user* terhadap URL yang berada di server telah selesai, dengan lamanya waktu untuk melakukan proses *request* terhadap URL adalah 11 - 12 detik. terjadi peningkatan aktivitas lamanya waktu pengaksesan URL dikarenakan jumlah faktor banyaknya *user* yang mengakses berbeda dari skenario pertama dan kedua serta jumlah pengakses tiap interval waktunya berbeda karena pengaksesannya bersifat acak, sehingga dengan penambahan beban pengaksesan user secara konstan dalam 10 menit kinerja dari sistem menjadi berat dan muncul pola *ripple* yang sangat jelas dengan setiap tahapan dari alur pengaksesan pun menunjukkan peningkatan lamanya waktu pengaksesan..

#### 4.5 Pengujian Beban Penggunaan layanan Website Dashboard (Skema Ramp)

- Pengujian (2000 user dengan *click delay* maksimum 5 detik) per user



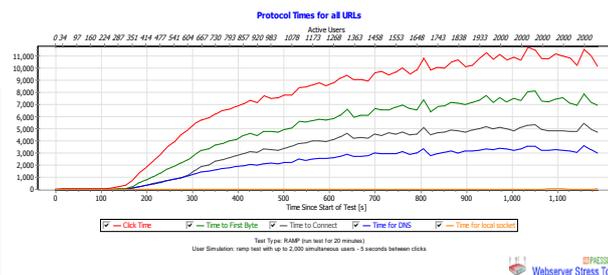
**Gambar 11** Grafik *bandwidth* server dan *user* 2000 *user*

Pada grafik 11 didapatkan suatu hasil pengukuran dengan *bandwidth* menggunakan alokasi *user* sebanyak 2000 *user* sebagai berikut :

1. *Bandwidth user* didapat 1100 Kbps untuk rata – rata *bandwidth* yang digunakan sebesar 2,25 kbps.
2. *Bandwidth* server yang maksimal digunakan adalah 2200 Kbps.

Hal ini dikarenakan oleh *Bandwidth user* yang diperoleh dari *packet outgoing* bukan hanya *request* tetapi diberi beban oleh aplikasi web stress tool, dan di rekam *packet outgoing*-nya yang memenuhi jaringan dan didapat sebesar 1100 Kbps pada awal terjadinya pengaksesan. Lalu, pada grafik menunjukkan bahwa pada saat pengaksesan sudah memasuki interval 185 – 200 detik terjadi penurunan *bandwidth* yang signifikan, hal ini dikarenakan faktor jumlah pengaksesan dimulai dari interval tersebut hingga akhir sudah mencapai suatu titik yang dapat dikatakan kinerja dari sistem ini sangat berat sehingga *bandwidth* yang dialokasikan untuk sejumlah *user* dihasilkan sangat kecil (rata – rata *bandwidth user* adalah 2,2524 Kbps). Dalam hal ini pemakaian *bandwidth user* memiliki keterkaitan dengan waktu pengaksesan halaman, sehingga semakin kecil *bandwidth* yang dihasilkan maka semakin lama pengaksesan terhadap *interface Dashboard*.

- Pengujian (2000 user dengan *click delay* maksimum 5 detik) per URL



**Gambar 4.12** Grafik protokol pengaksesan URL oleh 2000 user

Pada grafik diatas dapat diamati bahwa dalam pengujian pengaksesan menggunakan skema *ramp*, proses lamanya waktu pengaksesan terlihat semakin meningkat (lama). Dimulai pada saat 200 *user* ke atas waktu pemrosesan waktu terhadap URL menjadi lama yaitu diatas 1 detik. Karena berdasarkan pengujian sebelumnya (200 *user*) pada saat mencapai titik puncak terlama saat pengaksesan mencapai 90 ms. Sehingga untuk jumlah *user* dibawah 200 tidak Nampak fluktuasi waktu pengaksesan. Pada pengujian dengan 2000 *user* ini dapat amati dari data per periode waktunya bahwa semakin lama pengujian semakin banyak *user* yang mengakses disetiap intervalnya atau dapat dikatakan bahwa jumlah beban pengakses semakin lama semakin naik sehingga *ramp*-nya dapat terlihat. Selain itu pada pengaksesan dengan 2000 *user* ini juga terdapat *ripple*, hal ini disebabkan oleh adanya error pada saat pengaksesan oleh *user* pada suatu interval, sebagai contoh pada periode 443 – 458 detik error yang terjadi adalah 10,41 %.

## 5. PENUTUP

Dalam perancangan service Dashboard pada tugas akhir ini dapat didapat beberapa kesimpulan berikut :

1. Semua *platform* yang membangun Openstack (Nova, Swift, Glance, Keystone, Neutron, Cinder, dan Horizon) sudah terkonfigurasi dengan baik sehingga semua layanan Openstack tersebut dapat ditampilkan dan digunakan.
2. Dari hasil pengujian untuk desain UI (*User Interface*) dengan menggunakan kuesioner terhadap pengakses layanan Dashboard, didapatkan suatu kesimpulan bahwa rata – rata hasil penilaian kuesioner seluruhnya sebesar 3,75 dapat dikatakan *user* merasa puas.

3. Dari hasil pengujian untuk desain UX (*User Experience*) dengan menggunakan kuesioner terhadap pengakses layanan Dashboard, didapatkan suatu kesimpulan bahwa rata – rata hasil penilaian kuesioner seluruhnya sebesar 5,6.
4. Pada pengujian pengaksesan *interface Openstack Dashboard* didapatkan suatu analisis bahwa dengan pengujian sebanyak 20 *user* 100 klik dengan delay 5 detik menghasilkan *bandwidth user* = 1100 – 1200 Kbps, *bandwidth server* = 150 Kbps, pengujian 200 *user* 10 klik dengan delay 5 detik menghasilkan *bandwidth user* = 770 – 1000 Kbps, *bandwidth server* = 1500 - 1550 Kbps, dan pengujian 2000 *user* 100 klik dengan delay 5 detik menghasilkan rata - rata *bandwidth user* = 90 sampai 100 Kbps, rata – rata *bandwidth server* = 1598,52 Kbps.
5. Pada pengujian beban *interface Openstack Dashboard* dengan skema *Time* didapatkan suatu analisis bahwa dengan pengujian sebanyak 20 *user* 100 klik dengan delay 5 detik menghasilkan *bandwidth user* = 1100 Kbps, *bandwidth server* = 170 Kbps, pengujian 200 *user* 10 klik dengan delay 5 detik menghasilkan *bandwidth user* = 500 sampai 1100 Kbps, *bandwidth server* = 1500 sampai 600 Kbps, dan pengujian 2000 *user* 100 klik dengan delay 5 detik menghasilkan *bandwidth user* = 85 Kbps dengan rata – rata 1,3510 kbps, *bandwidth server* = 2200 sampai 2400 Kbps.
6. Pada pengujian beban *interface Openstack Dashboard* dengan skema *Ramp* didapatkan suatu analisis bahwa dengan pengujian sebanyak 20 *user* 100 klik dengan delay 5 detik menghasilkan *bandwidth user* = 1100 Kbps, *bandwidth server* = 150 Kbps, pengujian 200 *user* 100 klik dengan delay 5 detik menghasilkan *bandwidth user* = 1000 – 1100 Kbps, *bandwidth server* = 1500 Kbps, dan pengujian 2000 *user* 100 klik dengan delay 5 detik menghasilkan *bandwidth user* = 1100 Kbps dengan rata-rata *bandwidth* yang digunakan 2,25 kbps, *bandwidth server* = 2200 Kbps.

## REFERENSI

- [1] Anggi, Fersisilia. “**Pengertian / Definisi Cloud Computing**”. 4 Juni 2014 <http://pusatteknologi.com/pengertian-manfaat-cara-kerja-dan-contoh-cloud-computing.html> [Tanggal akses : 18 Maret 2015].
- [2] <http://www.crosstechno.com/blog/2013/apa-yang-dimaksud-dengan-flat-design/> [Tanggal akses : 10 Mei 2015].
- [3] <http://docs.openstack.org/training-guides/content/module001-ch004-openstack-architecture.html> [Tanggal akses : 18 Maret 2015].
- [4] <https://www.openstack.org/software/icehouse/> [Tanggal akses : 27 Mei 2015]
- [5] Konovalov, Nikita. 2013, **Building Dashboard with Openstack Horizon**, Mirantis. Inc.
- [6] Kumar, Rakesh. Gupta, Neha. Charu, Shilpi. Jain, Kanishk. Jangir, Sunil Kumar, 2014. **Open Source Solution for Cloud Computing Platform Using OpenStack**, Departmennt of Information Technology, JECRC, Jaipur, India.
- [7] Messerschmitt, Zwei “**Pengertian SaaS, PaaS dan IaaS**”. 24 Februari 2012. <https://zweimesserschmitt.wordpress.com/2012/02/24/pengertian-saas-paas-dan-iaas/> [diakses pada tanggal 9 Februari 2015].
- [8] Openstack Architecture Design Guide, Openstack Foundation, 2014 – 2015, [Diunduh pada tanggal : 4 Januari 2015].
- [9] Openstack Cloud Administrator Guide, Openstack Foundation, 2013 – 2014, [Diunduh pada tanggal : 4 Januari 2015].
- [10] Openstack Install Guide apt Icehouse for Ubuntu 14.04, Openstack Foundation, 2012-2013, [Diunduh pada tanggal : 4 Januari 2015].
- [11] Presekal, Alfian. “**Software Defined Network (SDN) Sebuah Paradigma Dunia Jaringan**”. 21 November 2012. <http://alfan.presekal.com/software-defined-network-sdn-paradigma-baru-dunia-jaringan/> [Tanggal Akses : 4 Februari 2015].
- [12] Rahma, Norma Fitria Puspa, 2013. **Analisis Implementasi Infrastruktur As a Service Menggunakan Ubuntu Cloud Infrastruktur**, Teknik Sistem Komputer, Fakultas Teknik, Universitas Diponegoro, Semarang, Indonesia.
- [13] Rodriguez, riyadi. “**Remote Computer Server Putty dan Winscp**”, 2012, <http://riyadi-rodriguez.blogspot.com/2012/02/remote-computer-server-putty-dan-winscp.html> [diakses pada tanggal : 14 Maret 2015].
- [14] Septian, Ridwan Fajar, 2013. **Python “Belajar Pemrograman Python Dasar”**, Bandung : POSS-UPI.
- [15] S, Girish L. Guruprasad, Dr. H S, 2014. **Building Private Cloud using OpenStack**, Dept. of ISE, Dept of CSE, BMSCE, Bangalore.
- [16] Triputra, Arifki. 2012. **Implementasi dan Analisis Performansi Infrastruktur as a Service (IaaS) Cloud Computing Berbasis Openstack**, Fakultas Elektro dan Komunikasi Institut Teknologi Telkom, Bandung, Indonesia.
- [17] Websserver Stress Tools 8 Guidance [online], Tersedia : <http://www.paessler.com/download/webstress> [Diunduh pada tanggal : 8 April 2015].