

VISUALISASI DATA PENGGUNA LABORATORIUM VIRTUAL PADA KOMPONEN TELEMETERING OPENSTACK BERBASIS WEB MONITORING

Prajaka Brian Lizadi, Hidayat Nur Isnianto
Departemen Teknik Elektro dan Informatika
Universitas Gadjah Mada

prajaka.brian.l@mail.ugm.ac.id, hnisnianto@mail.ugm.ac.id

Abstract – Currently the use of Cloud computing to meet human necessity was become more common. Cloud computing more comfort provide given because of his model is placed in a container and allows easily accessible according to necessity in various locations. One of the Cloud computing platform that is widely started using are Openstack , because of the ability to manage and regulate the entire system through a web-based dashboard, the command line and through RESTful API. One of the supporting components computing system on a platform Openstack is a project that serves as a Ceilometer telemetering data to present the results of the monitoring, but it is still the raw data that require interface for communication data and human beings. This research created a system of data visualization as an interface to use by admin and users in monitoring data usage per instance on a virtual laboratoritium. The result from this research is web monitoring interface with differents graphic type and specification to engage admin requirement for analisys in virtual laboratory activity.

Keywords : Cloud computing, Openstack, Ceilometer, data visualization.

Intisari – Saat ini penggunaan komputasi awan untuk memenuhi kebutuhan manusia sudah menjadi hal yang sangat umum. Komputasi awan dapat memberikan kenyamanan yang lebih karena modelnya yang ditempatkan dalam satu wadah dan memungkinkan untuk dapat diakses sesuai kebutuhan di berbagai lokasi. Salah satu platform komputasi awan yang mulai banyak digunakan adalah Openstack, karena memiliki kelebihan dalam mengelola dan mengatur keseluruhan sistemnya melalui dashboard berbasis web, perangkat perintah baris dan melalui RESTful API. Salah satu komponen pendukung sistem komputasi pada platform Openstack adalah Ceilometer yang berfungsi sebagai proyek telemetering untuk menyajikan data hasil monitoring, tetapi masih merupakan data mentah sehingga memerlukan interface untuk komunikasi data dan manusia. Pada penelitian ini dibuat suatu sistem visualisasi data sebagai interface untuk digunakan admin dan pengguna dalam memantau data penggunaan setiap instance pada laboratoritium virtual. Hasil yang diperoleh berupa tampilan halaman web monitoring dengan tipe grafik dan spesifikasi data berbeda – beda untuk memenuhi kebutuhan analisis admin dalam kegiatan laboratorium virtual

Kata kunci : Komputasi awan, Openstack, Ceilometer, Visualisasi data.

I. PENDAHULUAN

Perkembangan teknologi yang sangat pesat menyebabkan perubahan signifikan pada kebiasaan manusia. Kebiasaan yang dilakukan secara manual, secara bertahap dikembangkan agar dapat menjadi otomatis. Perwujudan tersebut adalah sesuatu yang diharapkan agar dapat meringankan kegiatan dalam lingkungan kerja. Penggambaran data dengan bentuk monitoring dalam suatu sistem kerja adalah salah satu otomatisasi yang dibentuk oleh manusia untuk memudahkan pengaturan dan analisa kegiatan dalam lingkungan kerja.

Dalam lingkungan kampus D-IV Teknologi Rekayasa Internet, kebutuhan terhadap infrastruktur jaringan dalam kegiatan praktikum, adalah sesuatu yang seharusnya wajib untuk disediakan. Hal ini menunjukkan, bahwa kebutuhan terhadap ketersediaan komputasi awan pada lingkungan kampus semakin diperlukan karena memberikan kemudahan dalam mengelola infrastruktur jaringan secara menyeluruh. Salah satu layanan komputasi awan yang baru – baru ini populer digunakan adalah Openstack. Layanan Openstack menjadi platform yang mulai banyak digunakan karena memberikan fasilitas layanan yang aman, handal, fleksibel, high-efficiency, dan dapat dikembangkan [1]. Pada lingkungan kampus, salah satu keuntungan menggunakan platform Openstack adalah pengguna lebih fleksibel dalam membangun infrastruktur terkait kebutuhan

praktikum komputer jaringan [2]. Platform Openstack pada kegiatan praktikum bekerja dengan membentuk suatu project yang digunakan untuk membangun instance virtual, sehingga kegiatan praktikum dapat disebut sebagai laboratorium virtual karena instance virtual dapat di akses dari semua jaringan internet UGM.

Sewajarnya, suatu perangkat yang dikembangkan oleh manusia pasti memiliki suatu kekurangan. Pada platform Openstack, kekurangan terdapat pada aplikasi metering yang belum memiliki kemampuan visualisasi data untuk keperluan analisis dan monitoring. Komponen Ceilometer pada platform Openstack dijelaskan telah menyajikan data – data monitoring [3], namun pada kenyataannya Ceilometer hanya bekerja sebatas pada metering data.

Dalam penelitian ini penulis memberikan solusi yaitu pengembangan komponen Ceilometer dengan mengubah data mentah pada aplikasi metering menjadi informasi berbentuk visual dalam sebuah halaman web monitoring. Halaman web monitoring akan dibangun dengan menggunakan bahasa markup web HTML, bahasa pemrograman PHP, dan Javascript. Dengan adanya pengembangan sistem visualisasi data dalam halaman web monitoring ini, admin dapat dengan mudah untuk menganalisis dan memonitor data dalam kegiatan laboratorium virtual. Selain hal tersebut, pengembangan ini diharapkan dapat mejadi solusi pada aplikasi Ceilometer

yang belum menyajikan *monitoring* dari sistem kerja Openstack [4].

II. KAJIAN PUSTAKA

2.1. Komputasi Awan (*Cloud computing*)

Definisi sederhana dari *cloud computing* adalah suatu model yang memungkinkan jaringan dapat diakses dengan mudah sesuai kebutuhan di berbagai lokasi, dimana model ini dapat mengumpulkan sumber daya komputasi seperti *network*, *server*, *storage*, aplikasi dan *services* dalam satu wadah.

Terdapat lima karakteristik penting dari *cloud computing* [5];

1. *On-demand self-service*. Pengguna dapat memesan dan mengelola layanan tanpa interaksi manusia dengan penyedia layanan, misalnya dengan menggunakan, sebuah portal *web* dan manajemen antarmuka. Pengadaan dan perlengkapan layanan serta sumberdaya yang terkait terjadi secara otomatis pada penyedia.
2. *Broad network access*. Kemampuan yang tersedia melalui jaringan dan diakses melalui mekanisme standar, yang mengenalkan penggunaan berbagai *platform* (misalnya, telepon selular, laptop, dan PDA).
3. *Resource pooling*. Penyatuan sumberdaya komputasi yang dimiliki penyedia untuk melayani beberapa konsumen menggunakan model multi-penyewa, dengan sumberdaya fisik dan *virtual* yang berbeda, ditetapkan secara dinamis dan ditugaskan sesuai dengan permintaan konsumen.
4. *Rapid elasticity*. Kemampuan dapat dengan cepat dan elastis ditetapkan.
5. *Measured Service*. Sistem komputasi awan secara otomatis mengawasi dan mengoptimalkan penggunaan sumberdaya dengan memanfaatkan kemampuan pengukuran (*metering*) pada beberapa tingkat yang sesuai dengan jenis layanan (misalnya, penyimpanan, pemrosesan, *bandwidth*, dan *account* pengguna aktif). Penggunaan sumberdaya dapat dipantau, dikendalikan, dan dilaporkan sebagai upaya memberikan transparansi bagi penyedia dan konsumen dari layanan yang digunakan.

2.2. Layanan *Cloud computing*

Pada *Cloud computing* terdapat tiga jenis model layanan dijelaskan sebagai berikut [5];

1. *Cloud Software as a Service (SaaS)*. Kemampuan yang diberikan kepada konsumen untuk hanya menggunakan aplikasi yang disediakan pada layanan *cloud*. Aplikasi tersebut dapat diakses dari berbagai perangkat client melalui antarmuka seperti *web browser* (misalnya, email berbasis *web*).
2. *Cloud Platform as a Service (PaaS)*. Kemampuan yang diberikan kepada konsumen untuk menjalankan aplikasi yang dibuat konsumen atau memperbolehkan konsumen untuk mengolah menggunakan bahasa pemrograman dan peralatan yang didukung oleh provider pada infrastruktur komputasi awan.
3. *Cloud Infrastructure as a Service (IaaS)*. Kemampuan yang diberikan kepada konsumen untuk memproses, menyimpan, dan melakukan komputasi sumberdaya pada layanan komputasi awan, dimana konsumen dapat menjalankan perangkat lunak secara bebas, dapat mencakup

sistem operasi dan aplikasi.

2.2.1. *Infrastructure as a Service (IaaS)*

Secara definisi *IaaS* adalah layanan komputasi awan yang umumnya berupa seperangkat hardware komputer yang berupa “*virtualisasi*”, bersama dengan jaringan internet, *bandwidth*, dukungan alamat IP, keseimbangan beban, jaminan online secara realtime (terus menerus koneksi ke internet) dan keamanan dalam “ruang lingkup” satu unit layanan *IaaS* [6].

Penerapan *Infrastructure as a Service (IaaS)* memiliki beberapa keuntungan, diantaranya sebagai berikut [7]:

1. Mudah dan cepat tersedia, dengan ketersediaan infrastruktur yang dapat dipesan sesuai kebutuhan.
2. Tidak perlu beli perangkat TI yang biasanya menguras biaya modal cukup besar, dan memakan waktu untuk instalasi serta konfigurasi.
3. Pengelolaan infrastruktur dilakukan oleh penyedia *IaaS*, sehingga hemat biaya profesional TI yang tergolong semakin mahal.
4. Dengan layanan infrastruktur (*IaaS*) anda dapat memperbesar dan mengurangi kapasitas *server* atau *storage* lebih cepat.
5. Hemat investasi awal bagi perusahaan startup ataupun bagi para developer aplikasi sebelum aplikasi tersebut dioperasikan secara normal atau dapat memperkecil biaya pra-operasional.

2.3. Openstack

Openstack adalah suatu proyek open source yang digunakan untuk *platform cloud computing*. Secara sederhana, Openstack merupakan *platform cloud computing* yang berasal dari berbagai macam tipe *cloud*. Teknologi Openstack terdiri dari serangkaian aplikasi yang saling terkait dan setiap bagian dari aplikasi tersebut berfungsi mengontrol pengolahan, penyimpanan, serta jaringan sumber daya lintas pusat data. Openstack memiliki kemampuan dalam mengelola dan mengatur keseluruhan sistemnya melalui dashboard berbasis *web*, perangkat perintah baris dan melalui *RESTful API*.

2.3.1. Komponen Openstack

Terdapat komponen-komponen penting untuk mengimplementasikan Openstack agar terbentuk sebuah *cloud computing*;

1. Horizon (*Dashboard*)
2. Keystone (*Identity*)
3. Neutron (*Networking*)
4. Cinder (*Block Storage*)
5. Nova (*Compute*)
6. Glance (*Image*)
7. Swift (*Object Storage*)
8. Ceilometer (*metering*)
9. Heat (*Orchestration*)

2.3.2. Ceilometer

Proyek *Ceilometer* adalah layanan pengumpulan data yang menyediakan kemampuan untuk menormalisasi dan mengubah data di seluruh komponen inti Openstack. *Ceilometer* adalah komponen proyek *telemetry*. Datanya dapat digunakan untuk menyediakan pelanggan

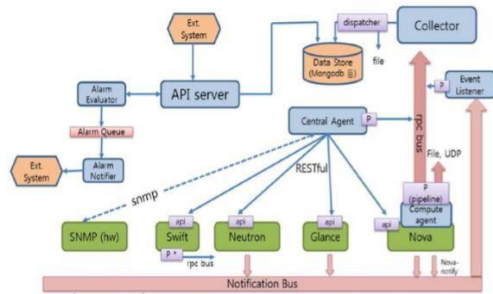
billing, resource tracking, benchmarking, scalability, dan statistical di seluruh semua platform Openstack.

Ada beberapa urutan pemahaman data Ceilometer pada Openstack yaitu;

1. Meter. Hanya mengukur aspek tertentu dari penggunaan sumber daya atau kinerja yang sedang berlangsung. Meter memiliki tiga tipe berbeda yaitu *cumulative*, *delta* and *gauge*. Jenis *cumulative* secara monoton meningkatkan nilai seperti nilai integrasi. *Delta* adalah perubahan nilai dari sebelumnya seperti diferensiasi nilai. *Gauge* adalah nilai yang berdiri sendiri yang hanya berkaitan dengan durasi saat ini.
2. Sample. Bekerja dengan menyederhanakan poin data individu yang berkaitan dengan pengukuran khusus. Sample memiliki atribut yang sama dengan Meter, juga ditambahkan timestamp dan pengukur nilai (disebut Volume).
3. Statistics. Nilai statistik agregasi selama durasi waktu yang ditentukan dan yang bukan seperti Sample adalah satu poin data. Terdapat 5 perbedaan fungsi agregasi (*count*, *max*, *min*, *avg*, *sum*). Artinya adalah jumlah Sample, nilai maksimal, nilai minimal, nilai rata-rata, jumlah total setiap periode.
4. Event. Memberikan informasi kepada pengguna kapan sesuatu objek berubah dalam Openstack. Sebagai contoh, ketika pengguna mengunduh *image*, pengguna membuat *Virtual Machine* dan ketika pengguna mengeksekusi perintah CRUD (*Create*, *Read*, *Update*, *Delete*) REST.

2.3.3. Struktur dan Komponen Ceilometer

Struktur dan komponen Ceilometer dapat kita lihat pada Gambar 2.1. terdapat 5 komponen dasar yang mendukung Ceilometer yaitu;



Gambar 2. 1 Struktur Komponen Ceilometer

1. *Collector*. Komponen penting yang terdapat pada server utama. *Notification messages* diproses dan di ubah menjadi data *metering message* yang kemudian dikirim kembali ke *message bus* dengan menggunakan topik yang sesuai. *metering message* secara langsung telah tercatat pada *Data Store* tanpa modifikasi. Pada awalnya, fungsi *pipeline* terletak pada *collector*, namun sekarang telah dipisah.
2. *Central agent*. Komponen yang terdapat pada server utama. Bekerja dengan memicu point dari polling. Terutama yang tidak berhubungan dengan fungsi komponen Nova yang telah digunakan seperti Object Storage (Swift), *image* (Glance). Akuisisi data ditransfer ke *collector* melalui RPC Bus.
3. *Compute agent*. Merupakan modul instalasi tambahan. Meskipun telah tercatat sebelum *push agent compute*, *agent* hanya digunakan untuk

billing. VM unit penghitungan *monitoring* (CPU number, accumulative CPU time, CPU utilization, virtual NIC packet rate, disk and memory statistics characteristic value) sangatlah penting. Akan tetapi, *Compute agent* tidak mendukung dimana metode *push agent* digunakan. Libvirt (*default*), hyper-V, dan Vmware tersedia pada hypervisor.

4. *Data Store*. Penyimpanan menggunakan *default* MongoDB. PostgreSQL, HBase, IBM DB2 sedang dalam proses pengembangan.
5. *API server*. *Web server* yang terletak pada server pusat. Untuk balasan dari yang dibutuhkan oleh *API*, digunakan mekanisme *RESTful*. *Pecan server* sangat ringan digunakan sebagai *web framework* bahasa pemrograman python. *Pecan server* mendukung *REST-style controller*, penggunaan bahasa dan mendukung JSON.

2.6. Visualisasi Data

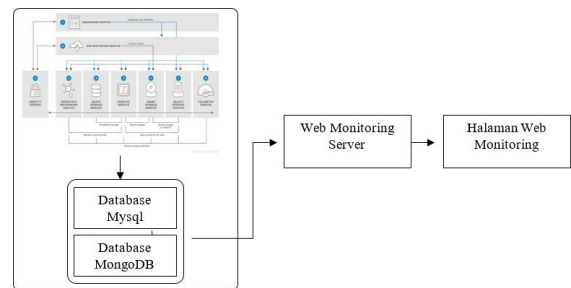
Tujuan utama dari visualisasi data adalah untuk menjembatani informasi secara jelas dan efisien kepada pengguna lewat grafik informasi yang dipilih, seperti tabel dan grafik. Visualisasi yang efektif membantu pengguna dalam menganalisa dan penalaran tentang data dan bukti. Ia membuat data yang kompleks bisa di akses, dipahami dan berguna. Pengguna bisa melakukan pekerjaan analisis tertentu, seperti melakukan perbandingan atau memahami kausalitas, dan prinsip perancangan dari grafik (contohnya, memperlihatkan perbandingan atau kausalitas) mengikuti pekerjaan tersebut. Tabel pada umumnya digunakan saat pengguna akan melihat ukuran tertentu dari sebuah variabel, sementara grafik dari berbagai tipe digunakan untuk melihat pola atau keterkaitan dalam data untuk satu atau lebih variabel [7].

III. METODOLOGI DAN PERCOBAAN

Tahap awal pembuatan sistem ini adalah dengan merancang topologi. Topologi yang diterapkan pada sistem ini adalah seperti Gambar 3.1 dan Gambar 3.2.



Gambar 3. 1 Topologi Perangkat

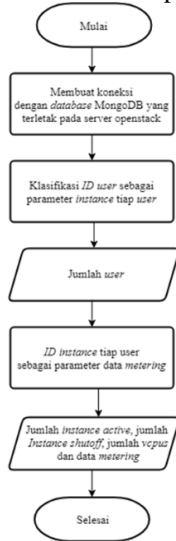


Gambar 3. 2 Topologi Proses Sistem

Platform Openstack memiliki aplikasi pendukung untuk *metering* data pengguna. Data hasil *telemetry* akan dikumpulkan dan dicatat kedalam *database* MongoDB. Kumpulan dari data tersebut kemudian akan di visualisasikan kedalam *web monitoring* berbentuk grafik.

Web monitoring yang akan digunakan menggunakan sistem operasi CentOS dan aplikasi pendukung (Apache, PHP, Javascript, PHP MongoDB driver). Ditempatkan pada alamat jaringan yang sama.

Gambar 3.3. merupakan alur dari sistem untuk menghasilkan visualisasasi data berupa grafik.



Gambar 3. 3 Diagram Alir Sistem

IV. ANALISIS DAN PEMBAHASAN

4.1. Informasi Data pada Dashboard

Pada halaman dashboard digambarkan jumlah total data dari user, core CPU fisik, instance, VCPU, VCPU yang digunakan, dan VCPU yang bisa digunakan. Pada Gambar 4.1 terlihat informasi pada total user berisi 12 user yang terdaftar. Informasi tersebut dapat dibuktikan dengan melihat pada halaman web Openstack yang menampilkan informasi data project atau user.



Gambar 4. 1 Informasi Total Data Instance

Pada Gambar 4.2, urutan daftar yang ada dalam halaman web Openstack menunjukkan jumlah total user sebanyak 12. Pengujian ini membuktikan informasi jumlah Total User sesuai dengan informasi pada Gambar 4.1.

Group	Name	Deskripsi	ID Project	Resiko Default	Dibuat	Aktif
Users (users)	admin	admin project	274470e57564689505c0576a7c56	Default	Ya	Manajemen Memberi akses anggota...
	PrinsipKebudayaan		3af5e78c736708a6284173c57929176	Default	Ya	Manajemen Memberi akses anggota...
	repositori2020		45c73043d6d48138a616891235fbc	Default	Ya	Manajemen Memberi akses anggota...
	ProjectLuh		02b15a2644648955736a2a227894	Default	Ya	Manajemen Memberi akses anggota...
	admin	admin tenant	548895a0838481Acu4a0877162892	Default	Ya	Manajemen Memberi akses anggota...
	Manajemen		78929193848d71822819864786617	Default	Ya	Manajemen Memberi akses anggota...
	ProjectRian		9d78a1974e4484385434a13252196	Default	Ya	Manajemen Memberi akses anggota...
	server		059481082749221242846877722c5	Default	Ya	Manajemen Memberi akses anggota...
	win-server		4746819702c08045a001038036a04	Default	Ya	Manajemen Memberi akses anggota...
	project1		4e180a7294216240f8540e108d379104	Default	Ya	Manajemen Memberi akses anggota...
	repositori	Tenant for the openstack services	912137a42248a0a051495365754c	Default	Ya	Manajemen Memberi akses anggota...
	gnc1		626c1282394627a175164410587	Default	Ya	Manajemen Memberi akses anggota...

Gambar 4. 2 Halaman Dashboard Openstack

Total VM active dan Shutdown adalah informasi berbentuk teks dinamis yang selanjutnya akan dijelaskan. Informasi teks dinamis akan berubah sesuai dengan kondisi instance. Sesuai dengan Gambar 4.1, terdapat instance yang menunjukkan informasi keadaan active dan shutdown.

Pada Gambar 4.3, dapat dilihat bahwa data yang sedang active dan shutdown. Pada bagian ini di ambil contoh pada instance dengan nama windows dan win - server yang sedang dalam keadaan shutdown. Data tersebut terhitung dalam

jumlah total data instance VM shutdown pada Gambar 4.1.

Instance Name	Type	VCPU	Last Update	Status
instansi1	x1.medium	2	2018-05-10 09:23:34	active
inst-51	x1.medium	2	2018-05-10 09:23:34	active
inst-53	x1.medium	2	2018-05-10 09:23:34	active
publsherv1	x1.small	1	2018-05-10 09:23:34	active
inst	x1.medium	2	2018-05-10 09:23:34	active
server-centos	x1.medium	2	2018-05-10 09:23:33	shutdown
inst-44	x1.medium	2	2018-05-10 09:23:34	active
inst-45	x1.medium	2	2018-05-10 09:23:34	active
inst10	x1.large	4	2018-05-10 09:23:33	shutdown
inst-41	x1.medium	2	2018-05-10 09:23:34	active
windows_sabu	x1.large	4	2018-05-10 09:23:34	shutdown
inst-52	x1.medium	2	2018-05-10 09:23:34	active
windows	x1.large	4	2018-05-10 09:23:33	shutdown
win-server	x1.large	4	2018-05-10 09:23:33	shutdown
inst52	x1.large	4	2018-05-10 09:23:34	active

Gambar 4. 3 Daftar Keadaan Instance

Instance pada Gambar 4.3, dapat dibuktikan dalam keadaan shutdown dengan melihat Gambar 4.4 pada halaman Openstack berikut.

Gambar 4. 4 Halaman Daftar Instance Pada Dashboard Openstack

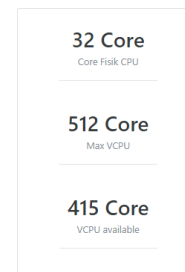
Informasi Total VCPU adalah jumlah dari core CPU yang digunakan oleh tiap instance. Data tersebut dibagi menjadi 4 tipe, yaitu adalah x1.small, x1.medium, x1.large dan x1.xlarge. Tipe - tipe tersebut mewakili dari penggunaan core yang dapat dilihat pada Tabel 4.1.

Tabel 4. 1 Daftar Tipe Instance

Tipe	Core VCPU	Disk	Memory
x1.small	1	20 GB	2048 MB
x1.medium	2	40 GB	4096 MB
x1.large	4	80 GB	8192 MB
x1.xlarge	8	160 GB	16384 MB

Total VCPU pada Gambar 4.2 adalah hasil dari penjumlahan semua jumlah core dari setiap instance ID.

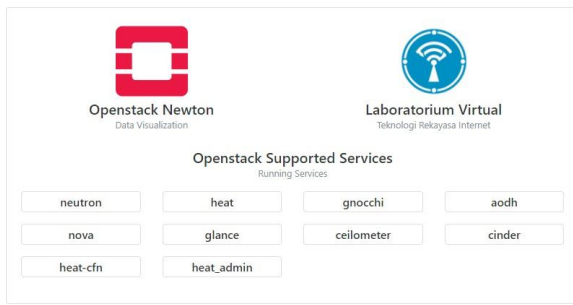
Server yang digunakan oleh platform Openstack pada penelitian proyek akhir ini memiliki spesifikasi core sebesar 32 core. Data tersebut, dapat digunakan sebagai parameter untuk menentukan jumlah banyak nya VCPU yang masih dapat digunakan. Informasi akan ditampilkan dengan data teks dinamis seperti Gambar 4.5.



Gambar 4. 5 Informasi Total Core Fisik dan VCPU

4.2. Informasi Aplikasi Pendukung

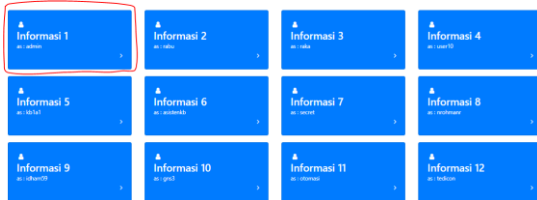
Halaman ini memberikan informasi tentang aplikasi yang mendukung berjalanya sistem dalam platform Openstack. Data running service pada Gambar 4.6, adalah teks dinamis yang sesuai dengan data yang terdapat pada database keystone.



Gambar 4. 6 Informasi *Running Service*

4.3. Informasi Pada Halaman Pengguna

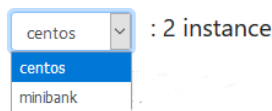
Jumlah total *user* seperti yang ditampilkan oleh Gambar 4.1, pada halaman ini akan ditampilkan secara deskriptif dengan menampilkan nama dan urutan. Pengguna tersebut ditampilkan menggunakan *card-body* yang diberikan fungsi untuk menghubungkan data dengan halaman data *instance*. Sebagai contoh, pada bagian ini digunakan *user* admin untuk mengakses halaman informasi data *instance* yang terdaftar pada *user* admin. Halaman *user* menampilkan informasi seperti pada Gambar 4.7. Informasi 1 seperti yang dilingkari pada gambar adalah perwakilan dari *user* admin.



Gambar 4. 7 Informasi Data *User*

4.4. Informasi Data Instance

Halaman data *instance* digunakan untuk menampilkan informasi dari data tiap *instance* yang telah dibuat pada masing – masing *user*. Seperti yang sudah dijelaskan sebelumnya, informasi yang terdapat pada halaman *user* digunakan untuk menghubungkan data kehalaman data *instance*. Dapat disebut juga bahwa data ID *user* digunakan sebagai parameter untuk mengakses data tiap *instance*. Setiap *user* dapat membangun *instance* lebih dari satu sampai batas core VCPU habis digunakan. Oleh karena itu, dibangunlah sebuah fungsi on change dengan dropdown pada halaman *web* untuk memilih *instance* dan melihat informasi data *instance* yang dipilih. Tampilan tabel dropdown, pada bagian ini mengambil contoh dari *user* dengan nama raka. Informasi tersebut berbentuk seperti Gambar 4.8.

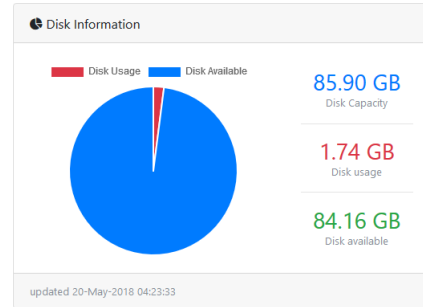


Gambar 4. 8 Informasi Dropdown *Instance*

Data *instance* dalam dropdown, ketika dipilih akan menampilkan informasi data yang sudah diklasifikasikan sesuai daftar *instance*. Daftar data yang ditampilkan memiliki daftar seperti pada Lampiran 1. Pada proyek akhir ini, data yang akan ditampilkan pada halaman *web* hanya mencakup data yang tertera pada batasan masalah. Pemilihan data yang akan ditampilkan berdasarkan perbandingan penggunaan data untuk kebutuhan analisis dan *diskusi* terkait data yang lebih penting untuk

ditampilkan. Data tersebut ditampilkan dalam beberapa grafik berebeda.

Data *disk* pada halaman ini akan di bagi menjadi 3 bentuk informasi dalam sebuah grafik pie. Sebagai pengujian, digunakan data *instance* centos dari data salah satu *user* bernama raka. *Instance* centos memiliki spesifikasi dengan tipe x1.large. Pada Gambar 4.9, bahwa warna merah mewakili jumlah data yang digunakan oleh *instance* centos, kemudian warna biru pada grafik mewakili jumlah *disk* yang masih tersedia.



Gambar 4. 9 Grafik *Disk Information*

Data yang terdapat pada grafik ditampilkan pada fungsi pop up dalam persen. Penampilan data tersebut seperti pada Gambar 4.10 dan 4.11.



Gambar 4. 10 Pop Up *Disk Available*

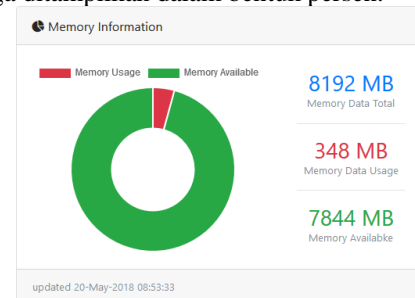


Gambar 4. 11 Pop Up *Disk Usage*

Informasi teks yang terletak disebelah grafik adalah teks dinamis yang menunjukan data dalam bentuk satuan GB. Data teks tersebut, adalah informasi yang ditampilkan berdasarkan satuan yang digunakan oleh data persen pada grafik. Informasi yang ditampilkan dari data tersebut akan menyesuaikan dengan parameter nama dan ID pada teks.

Kesimpulan dari informasi diatas, dapat dikatakan bahwa *instance* centos pada *user* raka, pada 20 Mei 2018 pukul 04:23:33 sudah menggunakan *disk* 2.03% atau sebesar 1.74 GB dan masih menyediakan *disk* 97.97% atau sebesar 85.90 GB.

Informasi data *memory*, pada dasarnya memiliki kesamaan tujuan informasi dengan data *disk*. Perbedaannya terletak pada informasi data yang ditampilkan. Pada Gambar 4.12, terlihat pada grafik bahwa warna merah mewakili dari data *memory* usage, kemudian warna hijau mewakili data *memory* available. Informasi data pada grafik juga ditampilkan dalam bentuk persen.



Gambar 4. 12 Grafik *Memory Information*

Data persen dari grafik dapat dilihat seperti pada Gambar 4.13 dan Gambar 4.14.



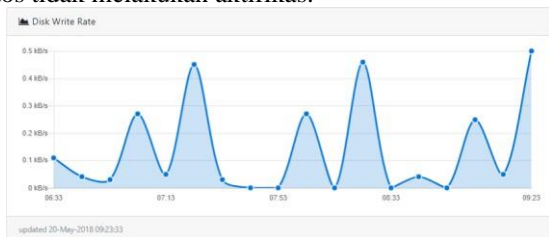
Gambar 4. 13 Pop Up Memory Usage



Gambar 4. 14 Pop Up Memory Available

Kesimpulan yang dapat diambil dari informasi yang ditampilkan adalah pada 20 Mei 2018 pukul 08:53:33 data *memory usage* menunjukkan 4.25% atau penggunaan data *memory* sebesar 348 MB dan *memory available* menunjukkan 95.75% atau jumlah *memory* yang tersedia sebesar 7844 MB.

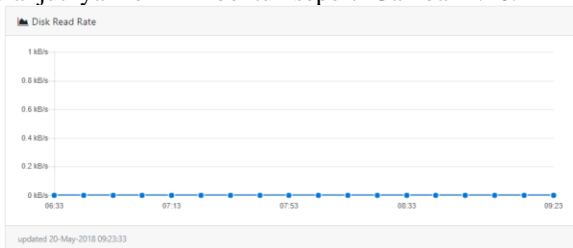
Informasi *disk write bytes rate* dan *disk read bytes rate* pada halaman ini ditampilkan dalam grafik dengan tipe line. Data tersebut adalah hasil mengukur tingkat rata – rata kegiatan membaca dan menulis *disk* dalam rentang waktu 10 menit. Waktu 10 menit ditentukan oleh aplikasi *telemetering* pada *platform* Opestack. Gambar 4.15 menunjukkan informasi dari *instance* centos dalam *user* dengan nama raka. Grafik line di atas menjelaskan informasi tentang *disk write rate bytes*. Data tersebut diperbaharui terakhir pada 20 Mei 2018 pukul 09.23.33. Grafik tersebut memiliki rentang waktu pada label selama 3 jam. Data yang ditampilkan adalah pada saat *instance* centos tidak melakukan aktifitas.



Gambar 4. 15 Informasi Grafik Disk Write Bytes Rate

Kuota yang terdapat disamping grafik, secara dinamis akan mengikuti data terbesar yang ditampilkan dalam grafik. Pada grafik diatas, data terbesar menunjukkan 0.5 kB/s. Jadi data dari rentang waktu 09.13 sampai dengan 09.23, rata-rata *disk* menulis data sebesar 0.5 kB/s.

Grafik *disk read bytes rate* yang akan dijelaskan selanjutnya memiliki bentuk seperti Gambar 4.16.



Gambar 4. 16 Informasi Grafik Disk Read Bytes Rate

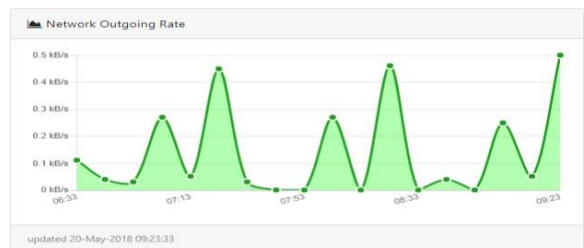
Tidak berbeda dari penjelasan pada *disk write bytes rate*. Pada grafik *disk read bytes rate* data yang tampak memiliki rata – rata sebesar 0 kB/s, kejadian tersebut dikarenakan tidak adanya proses membaca data dalam kurun waktu 3 jam terakhir. Data terakhir diperbaharui pada 20 Mei 2018 pukul 09.23.33.

Dalam grafik ditentukan rentang waktu grafik selama 3 jam. Hal ini dikarenakan dalam kebutuhan analisis kerja efektif pada laboratorium *virtual* adalah selama 2-3 jam.

Informasi yang ditunjukkan *network incoming* dan *outgoing bytes rate* memiliki pendekatan yang sama dengan penjelasan *disk write / read bytes rate*. Tidak ada perbedaan cara menampilkan data, hanya saja fungsi dari data tersebut memiliki tujuan berbeda. Informasi dari *network incoming / outgoing rate* berbentuk seperti Gambar 4.17 dan Gambar 4.18.



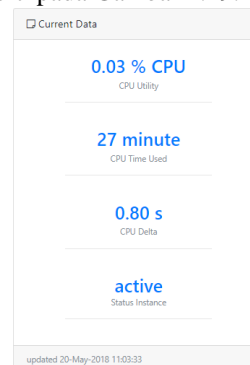
Gambar 4. 17 Informasi Grafik Network Incoming Rate



Gambar 4. 18 Informasi Grafik Network Incoming Rate

Pengujian menggunakan data *instance* yang sama yaitu centos. Pada saat *instance* tidak melakukan aktifitas, data dari *network incoming* dan *outgoing* menampilkan informasi data yang sama. Dari grafik tersebut, dapat disimpulkan pada *instance* centos, rata – rata data *network* yang masuk terbesar dalam waktu 3 jam terdapat pada rentang waktu 09.13 sampai 09.23 dengan menunjukkan informasi sebesar 0.5 kB/s . Kemudian rata – rata data yang keluar adalah sebesar 0.5 kB/s pada rentang waktu yang sama.

Informasi yang disajikan teks dinamis pada halaman ini ditampilkan seperti pada Gambar 4.19.

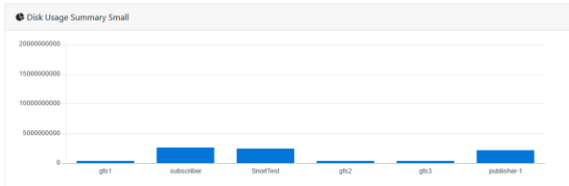


Gambar 4. 19 Informasi Data Teks Dinamis

Setiap data yang ditampilkan pada Gambar 4.20 memiliki fungsi tujuan penjelasan sesuai dengan nama yang tertera. Data yang ditampilkan terakhir diperbaharui pada 20 Mei 2018 pukul 11.03.33. Pengujian pada bagian ini dilakukan pada *instance* centos. Informasi CPU Utility menunjukkan waktu penggunaan CPU sebesar 0.03%, kemudian CPU

Time *user* sebesar 27 minute, lalu CPU *Delta* sebesar 0.80 second dan Status pada *instance* adalah *active*.

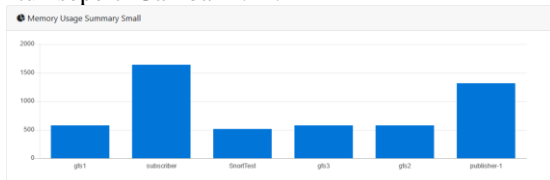
Summary data atau ringkasan data pada proyek akhir ini ditempatkan dalam satu halaman dengan nama halaman *summary* data. Halaman ini dapat di akses melalui Navigation Bar dengan nama *summary* data. Seperti namanya *summary* data adalah ringkasan semua data *instance* dalam satu halaman *web*. Ringkasan tersebut mencakup data – data yang sudah dijelaskan pada tahapan penelitian. Pada halaman ini pengujian dilakukan dengan menampilkan semua informasi dari tiap *instance* yang terdaftar. Informasi tersebut dapat dilihat pada Gambar 4.20.



Gambar 4. 20 Informasi Grafik *Summary* Data *Disk*

Gambar 4.20 adalah contoh grafik yang digunakan untuk *summary* data *disk*. *Instance* yang baru dibentuk akan secara otomatis terdaftar pada grafik sesuai dengan tipe *instance*. Grafik yang digunakan adalah grafik dengan tipe bar. Setiap tipe *instance* yang berada dalam grafik, telah dibatasi sesuai dengan kapasitas, sehingga ketika grafik bar sudah menyentuh angka akhir dari kuota yang sudah ditentukan, maka *disk* dari *instance* tersebut dapat dinyatakan penuh. Data *disk* setiap *instance* menggunakan satuan kB dengan alasan agar data dapat ditampilkan sedetail mungkin.

Informasi berikutnya adalah data *memory* yang akan dibentok seperti Gambar 4.21.

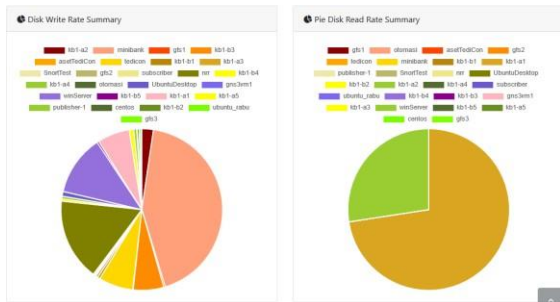


Gambar 4. 21 Informasi Grafik *Summary* Data *Memory*

Seperti halnya dengan grafik data *disk*, grafik data *memory* juga memiliki data dengan tipe *instance*, sehingga data *instance* dikelompokkan sesuai dengan tipe *instance*. Gambar 4.21 adalah contoh dari grafik *summary* data *memory* Data *memory* setiap waktunya dapat berubah – ubah sesuai dengan penggunaan *memory instance*. Grafik yang ditampilkan memiliki data bertipe MB dan total kapasitas di tulis pada kuota. Jika data mencapai batas kuota maka artinya penggunaan *memory* dapat dikatakan penuh.

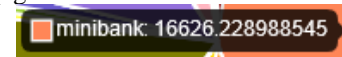
Informasi ringkasan data selanjutnya adalah pada data *disk write / read bytes rate* dan *network incoming / outgoing rate*. Ringkasan data dibuat dengan menggunakan

2 grafik yaitu grafik pie dan grafik line. Grafik pie digunakan untuk menampilkan data terbaru setiap *instance* yang memiliki kapasitas rata – rata *disk* membaca dan menulis terbesar. Bentuk grafik tersebut seperti pada Gambar 4.22.



Gambar 4. 22 Informasi Grafik *Summary* Data *Disk Write* and *Read*

Grafik pie pada Gambar 4.22 menunjukkan sejumlah *instance* ID yang diwakili oleh tiap warna. Pada kedua grafik, terlihat warna yang mendominasi grafik pie tersebut. Warna yang mendominasi tersebut adalah rata – rata *disk* yang memiliki tingkat membaca dan menulis paling besar dari semua *disk*. Data *instance* tersebut adalah data pada Gambar 4.23 dan Gambar 4.34 yang ditampilkan dalam pop up grafik.



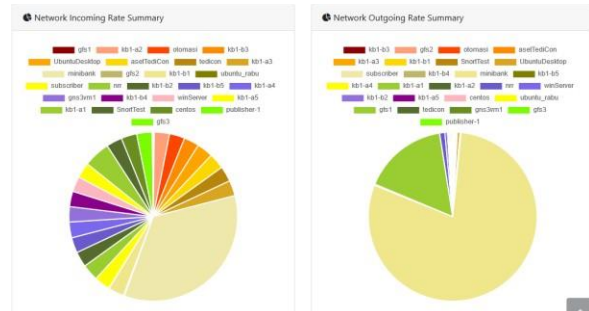
Gambar 4. 23 Pop Up Data *Disk Write* Terbesar



Gambar 4. 24 Pop UP Data *Disk Read* Terbesar

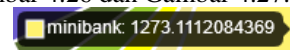
Penggunaan *resource disk write* dan *read rate bytes* terbesar adalah ditempati oleh *instance minibank* dan kb1-a1. Data yang ditampilkan menggunakan satuan B/s agar data dapat dianalisa secara detail.

Pada informasi ringkasan *network incoming / outgoing rate bytes* pada dasar nya memiliki fungsi grafik dan tujuan yang sama dengan yang dijelaskan pada ringkasan data *disk write / read rate bytes*. Grafik data tersebut dapat dilihat pada Gambar 4.25.

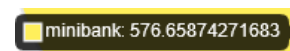


Gambar 4. 25 Grafik Informasi *Network Incoming* and *Outgoing*

Dari grafik tersebut terlihat beberapa warna yang mendominasi. Warna tersebut mewakili dari data yang ditunjukkan Gambar 4.26 dan Gambar 4.27.



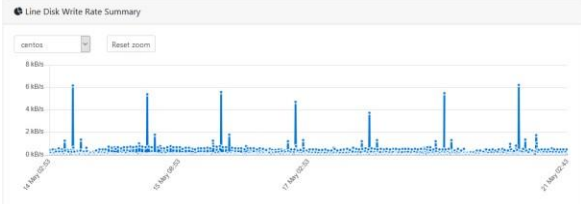
Gambar 4. 26 Pop Up Data *Network Incoming Rate* Terbesar



Gambar 4. 27 Pop Up Data *Network Outgoing Rate* Terbesar

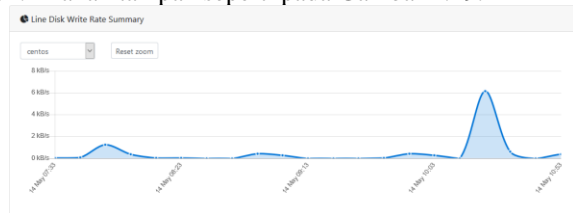
Instance minibank adalah *instance* yang memiliki penggunaan *network incoming* dan *outgoing* terbesar. Data yang ditampilkan menggunakan satuan B/s, agar data dapat dianalisa secara detail.

Informasi ringkasan data *disk write / read* dan *network incoming / outgoing* menggunakan grafik dengan tipe line yang digunakan untuk menampilkan ringkasan data dalam kurun waktu 1 minggu. Data tersebut ditampilkan sebagai record data untuk dapat memenuhi kebutuhan analisis. Grafik tersebut ditampilkankan seperti pada Gambar 4.28.



Gambar 4. 28 Informasi Grafik Line Summary Data Disk WriteRate

Grafik tersebut memiliki kegunaan yang sama dalam tujuan menampilkan data. Grafik tersebut diberikan fungsi *zoom in* dan *zoom out* untuk dapat menampilkan data secara detail. Sebagai contoh, pada bagian ini akan digunakan *instance* centos sebagai data yang dipilih untuk dilihat secara detail. Data yang di inginkan adalah data *disk write bytes rate* terletak pada tanggal 14 Mei 2018 pukul 07.33 sampai 10.53 maka pada grafik setelah dilakukan *zoom in* akan tampak seperti pada Gambar 4.29.



Gambar 4. 29 Informasi Data Summary Setelah Zoom In

Terlihat pada Gambar 4.29, data yang berada pada line chart hanya pada rentan waktu sesuai yang di inginkan. Pada grafik line tersebut juga ditambahkan beberapa fungsi *on change* pada menu dropdown untuk memilih data *instance* dan *button reset* untuk mereset grafik agar kembali seperti semula setelah proses *zoom in*. fungsi dropdown dan reset tersebut berbentuk seperti Gambar 4.30.



Gambar 4. 30 Menu Dropdown dan Menu Reset

Pada halaman ini, informasi yang disajikan berupa ringkasan data detail dalam bentuk tabel. Halaman ini digunakan untuk menampilkan masing – masing *instance* yang terdaftar dan keadaan *instance* sesuai dengan waktu diperbaharui. Informasi data yang tertera pada Gambar 4.31 diharapkan dapat digunakan untuk membantu tahap awal sebelum kegiatan analisis.

Instance Name	Type	VCPU	Last Update	Status
gfs1	m1.small	1	2018-05-21 03:03:35	active
asetfedicon	m1.large	8	2018-05-21 03:03:35	active
witServer	m1.large	4	2018-05-21 03:03:35	active
UbuntuDesktop	m1.medium	2	2018-05-21 03:03:35	active
gns3m1	m1.medium	2	2018-05-21 03:03:35	active
subscriber	m1.small	1	2018-05-21 03:03:35	active
tedicon	m1.large	8	2018-05-21 03:03:35	active
SnortTest	m1.small	1	2018-05-21 03:03:35	active
minBank	m1.large	4	2018-05-21 03:03:35	active

Gambar 4. 31 Daftar Informasi Data Halaman Tabel

Kesimpulan yang dapat diberikan adalah *instance* yang dibangun memiliki tipe spesifikasi *instance* yang berbeda sesuai dengan kebutuhan penggunaan *instance*. Data akan ditampilkan dan diperbaharui setiap 10 menit sekali, dengan tipe grafik dan fungsi analisis yang sesuai dengan data. Setiap informasi yang ditampilkan diharapkan dapat digunakan untuk analisis data dalam kegiatan laboratorium *virtual* atau penelitian yang menggunakan *platfotm* Openstack

DAFTAR PUSTAKA

- [1] C. Shao, B. Liang, F. Wang, H. Deng, W. Dai, S. Wei, X. Zhang and Z. Yuan, "Open Stack Platform and its Application in BigData Processing," 2015 8th International Conference on Intelligent Networks and Intelligent Systems, pp. 98-101, 2015.
- [2] R. Y. Faisela, PENGEMBANGAN LAYANAN INFRASTRUCTURE AS A SERVICE (IAAS) UNTUK LABORATORIUM VIRTUAL MENGGUNAKAN PLATFORM ORCHESTRATION PADA OPENSTACK, Yogyakarta: Universitas Gadjah Mada, 2017.
- [3] E. Qevani, M. Panagopoulou, C. Stampoltas, A. Tsitsipas, D. Kyriazis and M. Themistocleous, "What can OpenStack adopt from a Ganeti-based open-source IaaS?," 2014 IEEE International Conference on Cloud Computing, pp. 833-840, 2014.
- [4] B. Dongmyoung and L. Bumchul, "Analysis of Telemetry Service in OpenStack," ICTC 2015, pp. 272-274, 2015.
- [5] A. Syaikhu, "KOMPUTASI AWAN (CLOUD COMPUTING) PERPUSTAKAAN PERTANIAN," Jurnal Pustakawan Indonesia Volume 10 No. 1, pp. 1-12, 2010.
- [6] D. Cloud, "Pengertian Infrastructure As a Service (IaaS)," 6 juni 2016. [Online]. Available: <https://cloud.datacomm.co.id/blog/pengertian-iaas/>.
- [7] FlyingEagle, "Pengertian IaaS – Infrastructure as a Service," 24 Mei 2016. [Online]. Available: <http://www.mobnasesemka.com/pengertian-iaas/>.