Jurnal Pseudocode, Volume IV Nomor 2, September 2017, ISSN 2355-5920
www.ejournal.unib.ac.id/index.php/pseudocode

ANALISIS DAN PERANCANGAN INFRASTRUKTUR PRIVATE CLOUD DENGAN OPENSTACK

Muhammad Fauzan[1], Andrew Fiade, Fenty Eka M. A.[3]
[1]Mahasiswa Program Studi Teknik Informatika Fakultas Sains dan Teknologi
[23]Staff Pengajar Program Studi Teknik Informatika Fakultas Sains dan Teknologi
Universitas Islam Negeri Syarif Hidayatullah Jakarta

[1]mufahaza@gmail.com
[2]myresearch@uinjkt.ac.id
[3]fentyema@gmail.com

Abstrak: Pusat TIK Nasional menggunakan infrastuktur IT konvensional dimana penggunaan *resource* masih bersifat boros, dan manajemen serta monitoring yang rumit. Oleh karena itu, perlu dirancang sebuah infrastruktur *private cloud* menggunakan Openstack yang dapat melakukan *resource sharing.* Tujuan perancangan *privat cloud* adalah melakukan efisiensi dan skalabilitas yang tinggi serta memudahkan akses, monitoring dan manajemen secara terpusat. Berdasarkan hasil penelitian yang dilakukan, sistem infrastruktur *private cloud* yang dirancang dan dibangun dengan Openstack versi Kilo menggunakan satu node *controller* dan satu node *compute* mampu melakukan efisiensi sumber daya infrastruktur IT sebesar 1 CPU dan 638.296 bytes RAM dan mampu melakukan *resource sharing* infrastruktur IT.
Kata Kunci: Cloud Computing, Private Cloud, Infrastructure as a Service, IaaS, Openstack

***Abstract:** The National ICT Center uses conventional IT infrastructure where resource use is still wasteful, and management and monitoring are complex. Therefore, it is necessary to construct a private cloud infrastructure by using Openstack that can do resource sharing. The purpose of private cloud design is to perform high efficiency and scalability and facilitate centralized access, monitoring and management. Based on the results of research conducted, using one node controller and one node compute efficiency and capacity can be done. CPU and 638,296 bytes of RAM and can do resource sharing of IT infrastructure.*
Keywords:** Cloud Computing, Private Cloud, Infrastructure As Service, IaaS, Openstack

1. PENDAHULUAN

Cloud computing adalah salah satu model komputasi yang belakangan ini banyak diterapkan, tidak hanya perusahaan yang bergerak di bidang teknologi, tapi perusahaan yang bergerak di bidang lain pun saat ini sudah banyak yang menggunakannya untuk menggerakkan perusahaannya. Penggunaan teknologi *cloud computing* pun tidak hanya mempermudah pekerjaan manusia, tetapi juga dapat mengurangi *cost* karena teknologi ini dapat menghemat *resource* yang tidak terpakai karena infrastruktur tidak dapat dibagi-bagi dan digunakan oleh pengguna lain, karena itulah infrastruktur *cloud* dikembangkan dan saat ini sudah memasuki masa pengembangan yang lebih jauh lagi. Untuk membangun infrastruktur *cloud,* diperlukan pemahaman yang jelas tentang kekuatan dan juga keterbatasan *cloud computing* oleh arsitek *cloud* tersebut, karena perbedaan kebutuhan infrastruktur yang dibangun akan juga akan membuat pengembangan dan pembangunan disetiap kasus berbeda-beda.

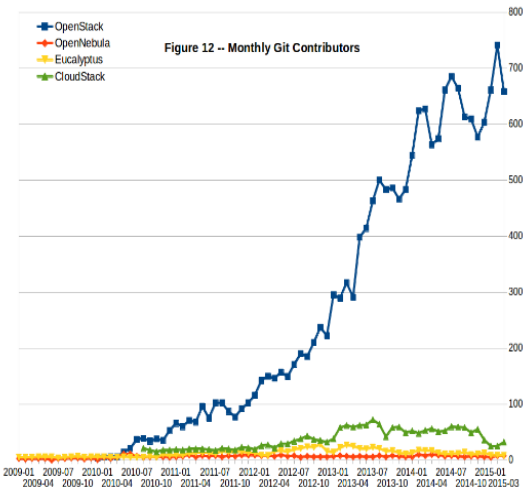
Pusat TIK Nasional adalah salah satu instansi pemerintahan di bawah pengawasan Kementerian Komunikasi dan Informatika yang berfokus pada pelatihan dan riset di bidang TIK. Menurut hasil

wawancara dengan staf IT Pusat TIK Nasional, diketahui bahwa pada infrastruktur IT di Pusat TIK Nasional, ketika kebutuhan sumber daya tidak mencukupi, staff IT harus secara menambahkan *hardware* secara manual. Selain itu, proses monitoring dan manajemen pun rumit, karena belum dapat dilakukan secara terpusat. Oleh karena itu, riset ini dilakukan untuk menyelesaikan masalah tersebut dengan membangun sebuah infrastruktur *cloud* di Pusat TIK Nasional.

Keuntungan dari *private cloud* dibanding *public cloud* (menyewa layanan *cloud* ke vendor), yaitu keamanan yang lebih tinggi karena *cloud* layanan ini hanya bergerak dalam jaringan internal Pusat TIK Nasional. Selain itu, Pusat TIK Nasional dapat menghemat *bandwidth* dan tidak perlu terus-menerus menyewa layanan *cloud.* Tetapi kebalikannya, pengembangan *private cloud* ini pun tidak lepas dari modal yang cukup besar, dibanding dengan menggunakan layanan *public cloud*, yang infrastrukturnya diurus oleh penyedia tersebut, *private cloud* membutuhkan departemen IT khusus untuk mengurus infrastrukturnya.

Untuk membangun sebuah infrastruktur *cloud* di Pusat TIK Nasional, dibutuhkan sebuah *software* atau *platform*, beberapa diantaranya yaitu Openstack, VMware VCloud, VMWare VSphere, Opennebula dan CloudStack. Untuk produk VMWare, dalam penggunaanya dikenakan biaya lisensi sementara produk *opensource* tidak dikenakan biaya dalam penggunaannya. Untuk menghemat pengeluaran Pihak IT Pusat TIK Nasional, dibangun infrastruktur *cloud* dengan *software / platform Opensource*.

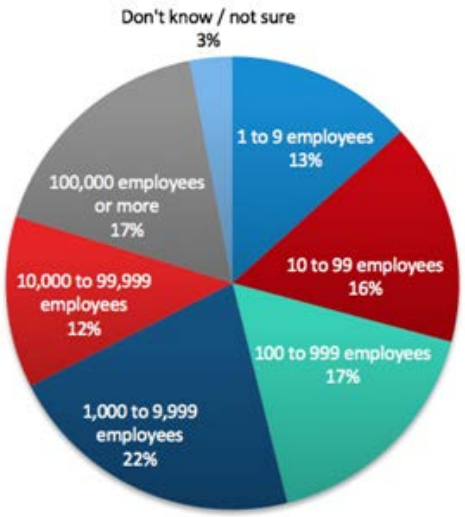
Qiengye Jiang, dalam jurnal publikasinya [1] menjelaskan perbandingan *platform cloud* yang bersifat *Opensource* yang dikembangkan oleh komunitas dari tahin 2009 sampai dengan 2015 seperti yang dapat dilihat pada gambar 1.3.



Gambar 1.1 Kontributor Git bulanan [1]

Pada gambar 1.1 dapat dilihat kontributor git pada *platform Openstack* jauh mengungguli *platform* komunitas lain. Statistik tersebut menunjukan bahwa, *Openstack* baru diciptakan pada tahun 2010, dimana jumlah kontributornya masih dibawah 100 orang, lalu. Dengan perkembangan yang pesat, pada tahun 2015 *Openstack* telah memiliki sebanyak 600 lebih contributor [2]. Pengertian Git itu sendiri adalah git adalah sebuah *system versioning control* (sistem kontrol versi) terdistribusi, atau biasa disebut dengan *source code management software* (aplikasi manajemen kode sumber), artinya *soure code* dan pengembangan *Openstack* disup*port* oleh banyak *developer*. perangkat lunak *Openstack* akan selalu berkembang cepat mengikuti kebutuhan yang ada.

Sementara itu, berdasarkan hasil survei dari OpenStack *User Committee* dibantu oleh *Foundation staff* dan *independent data scientist* yang melibatkan 1.325 *users* dan 352 *deployment*, Openstack digunakan perusahaan / organisasi berskala kecil, menengah, hingga yang berskala besar.



Gambar 1.2 Statistik organisasi / perusahaan pengguna Openstack berdasarkan skalanya [2]

Dari gambar 1.2 bisa dilihat bahwa pengguna Openstack terbanyak dengan raihan 22% adalah organisasi / perusahaan yang memiliki 1.000 sampai 9.999 karyawan, lalu peringkat kedua dengan 17% suara adalah organisasi / perusahaan yang memiliki 100.000 karyawan lebih. Dari data ini dapat diketahui bahwa sudah banyak perusahaan / organisasi berskala besar yang mempercayakan departemen *cloud*-nya pada Openstack.

Berdasarkan data tersebut, masalah dapat dirumuskan sebagai berikut:

(a) Bagaimana merancang infrastruktur *Private Cloud* dengan *Openstack* yang mampu melakukan efisiensi sumber daya infrastruktur IT di Pusat TIK Nasional;

(b) Bagaimana merancang infrastruktur *Private Cloud* dengan *Openstack* yang mampu melakukan *resource sharing* IT di Pusat TIK Nasional.

Untuk menyelesaikan masalah diatas, dilakukan penelitian dengan tujuan sebagai berikut:

(a) Merancang dan membangun infrastruktur *private cloud* dengan *Openstack*;

(b) Menguji fungsionalitas dan kapabilitas infrastruktur *private cloud* yang dibangun di Pusat TIK Nasional.

Penelitian yang dilakukan dibatasi pada:

(a) *User* harus terhubung dengan jaringan internal Pusat TIK Nasional untuk mengakses infrastruktur *cloud*;

(b) Infrastruktur *cloud* dibangun dengan arsitektur dua node, node *controller* untuk mengatur semua akses dan servis, dan node *compute* untuk melakukan komputasi dari servis-servis *cloud*, dan dengan satu *network interface card* pada masing-masing node;

(c) Tidak membahas sisi keamanan pada infrastruktur *private cloud* yang dibangun.

2. LANDASAN TEORI

2.1. Cloud Computing

Kemunculan *Cloud Computing* dilatarbelakangi oleh kebutuhan dunia industri dan komputerisasi akan pemanfaatan bersama sumber daya komputasi yang tersebar namun dapat digunakan sesuai keperluan (*on demand*). Hal lainnya yang mendukung munculnya teknologi *Cloud Computing* adalah teknologi web 2.0, teknologi *Web Service*, serta kemampuan komputasi otomatis yang dilakukan oleh komputer (*automatic computation*) terkait dengan manajemen sumber daya yang dimilikinya. Tentu saja, perkembangan perangkat keras (termasuk juga perangkat di jaringan komputer dan perangkat keras komputer), perangkat lunak (termasuk juga sistem operasi, aplikasi, metode pengembangan perangkat lunak, *library*), serta perkembangan internet (dengan kemunculan 4G dan 5G yang menandai kemajuan kecepatan internet di dunia), turut mendukung lahirnya *Cloud Computing*

sebagai sebuah teknologi dan layanan komputasi dan sumber daya komputasi [3].

Proses implementasi dari *cloud computing* membutuhkan pemahaman yang lebih baik. Pengetahuan tentang *cloud computing* dapat memberikan informasi yang lebih tentang definisi, arsitektur dan model-model *delivery service* yang dimilikinya. Isu dari *cloud computing* saat ini adalah efisiensi dan kelincahan dari sistem yang dimilikinya. Sistem ini juga secara sekaligus dapat meningkatkan kehandalan, keamanan dan pengendalian yang lebih baik, baik bagi pengguna maupun penyedia jasa. Pada setiap fase pengembangan desain dari *cloud computing* perlu dipertimbangkan penggunaan istilah-istilah yang benar, sehingga penggunanya dapat merasa nyaman dan tidak ragu-ragu dalam mengadopsi layanan-layanan yang akan dikirimkan kepada mereka oleh para penyedia jasa. Oleh karena itu, diperlukan cara yang efektif untuk mendistribusikan *cloud computing* kepada penggunanya [4].

2.1.1. Layanan Cloud Computing (IaaS)

Infrastructure as a Service adalah sebuah layanan yang menyewakan sumber daya teknologi informasi dasar, yang meliputi media penyimpanan, *processing power*, *memory*, sistem operasi, kapasitas jaringan, dan lain-lain yang dapat digunakan oleh pengguna untuk menjalankan aplikasi yang dimilikinya. IaaS memungkinkan perusahaan untuk memindahkan program yang ada kedalam *cloud*, dan menutup *server* lokal, dan *data center.*

Keuntungan dari IaaS ini adalah pengguna tidak perlu membeli komputer fisik dan konfigurasi komputer virtual tersebut sudah kelebihan beban, pengguna pun bisa menambahkan CPU, RAM dan *storage* dengan segera [5].

2.1.2. Komponen Cloud Computing

Di bawah ini akan dijelaskan pengertian dan fungsi dari *Controller* Node dan *Compute* node.

(a) Node *Compute*

Node *Compute* merupakan komponen pada *Cloud Computing* yang memiliki fungsi utama untuk melakukan kontrol terhadap node (komputer) pada sistem *Cloud Computing*.

(b) Node *Controller*

Node *Controller* merupakan komponen yang berhubungan langsung dengan pengguna layanan berbasis *Cloud Computing*. Para pengguna biasa maupun pengguna tertinggi (administrator). Sehingga posisi *Cloud Controller* tepat berada di antara pengguna layanan *Cloud Computing* dan *Cluster Controller*, yang tentu saja memiliki sejumlah Node *Controller* di belakangnya.

2.2. Private Cloud

Private cloud dimaksudkan sebagai model *deployment Cloud Computing* yang ditujukan untuk penggunaan yang terbatas pada kalangan tertentu saja (*private*), Model *deployment* ini umumnya banyak diterapkan untuk lingkungan laboratorium riset, sekolah, perpustakaan, gedung/bangunan (kantor/perusahaan), dan lain-lain [3].

2.3. 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 perusahaan 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 [6].

Ada beberapa istilah yang sering digunakan oleh infrastruktur *cloud* berbasis Openstack, diantaranya yaitu:

(a) *Tenant*: *Tenant* adalah istilah yang digunakan Keystone dan *equivalent* dengan *project* dalam Horizon (web-ui). *Tenant* atau *project* adalah *group* items yang terdiri dari *user*s, *image*s, *instances*, *network*s, dan *volume*.

(b) *Ephemeral Disk*: Ephemeral *Disk* adalah sebuah *temporary disk* yang digunakan oleh *instances* (*virtual machine*).

(c) *Instances*: *Instances* adalah istilah di Openstack yang menandakan sebuah *Virtual machine*s.

(d) Flavor: Flavor adalah *hardware* yang diasosiasikan ke *instances* (*virtual machine*) yang akan dibuat. *Hardware* tersebut adalah RAM, CPUs, dan *Disks* .

Openstack tersusun dari beberapa komponen. Adapun komponen-komponen tersebut adalah sebagai berikut:

1. Nova (*Compute Service).* Semua kegiatan yang diperlukan untuk mendukung siklus hidup dari *instance* dalam OpenStack *cloud* yang ditangani oleh Nova. Hal ini membuat Nova sebagai Platform Manajemen yang mengelola sumber daya komputasi, jaringan, otorisasi, dan kebutuhan skalabilitas dari OpenStack *cloud*.
2. Glance (*Image Service) OpenStack Imaging Service* adalah salah satu produk dari *OpenStack* yang digunakan untuk layanan *virtual disk images*.
3. Keystone (*Identity Service)* menyediakan layanan identitas dan akses kebijakan untuk semua komponen dalam keluarga *OpenStack.* Keystone menerapkan itu di *REST*-nya sendiri yang berbasis *API* (*Identity API*). Keystone menyediakan otentikasi dan otorisasi untuk semua komponen *OpenStack*. Otorisasi akan memverifikasi apakah pengguna yang terotentikasi memiliki akses ke layanannya yang dia minta atau tidak.
4. Neutron (*Neutworking Service*) adalah salah satu komponen openstack yang menyediakan layanan *cloud Network as a Service.* Neutron menyediakan API yang memungkinkan Anda menentukan konektivitas jaringan dan menangani di awan.
5. Cinder (*Block Storage Service*) adalah komponen penyimpanan blok yang lebih analog dengan gagasan tradisional komputer yang dapat mengakses lokasi tertentu pada *disk* drive serta menyediakan perangkat penyimpanan untuk digunakan dengan *instances* pada OpenStack.
6. Horizon (*User Interface Service*) merupakan suatu layanan *user interface* dalam infrastruktur *Openstack* yang memberikan akses visusalisasi bagi *user* dalam menciptakan *cloud*.

3. METODE PENELITIAN

Dengan metode pengembangan sistem NDLC, dilakukan enam tahapan dasar dalam membangun/mengembangkan sistem.

(a) Analisis. Melakukan identifikasi dan menemukan permasalahan yang ada, dipahami mengapa permasalahan itu terjadi di Pusat TIK Nasional, dan hal apa saja yang ditimbulkan dari permasalahan itu.

(b) Design. Pada tahap ini, diperbaiki sistem berjalan atau dirancang sistem baru. Pada riset ini ditambahkan sebuah infrastruktur *cloud* diterapkan di lokasi penelitian.

(c) Simulasi Prototipe. Sebelum mengimplementasikan pada sistem berjalan, dilakukan riset dan percobaan terlebih dahulu agar diketahui masalah apa yang akan dihadapi pada saat implementasi.

(d) Implementasi. Berbekal hasil riset dan percobaan simulasi, dilakukan implementasi infrastruktur *cloud* yang sudah di *design* sebelumnya.

(e) Monitoring. Setelah berhasil mengimplementasikan sistem infrastruktur cloud, sistem baru akan diuji apa berjalan semestinya atau tidak.

(f) Manajemen. Agar sistem infrastrutur *cloud* yang dibangun dapat digunakan dengan baik dan bertahan lama, perlu dilakukan aktivitas perawatan, pemeliharaan, serta pengololaah dari sistem yang telah dibangun. Tahapan ini juga menentukan siapa saja yang dapat mengakses dan mengelola sistem infrastruktur *cloud*.

4. PERANCANGAN SISTEM

4.1. Analisis

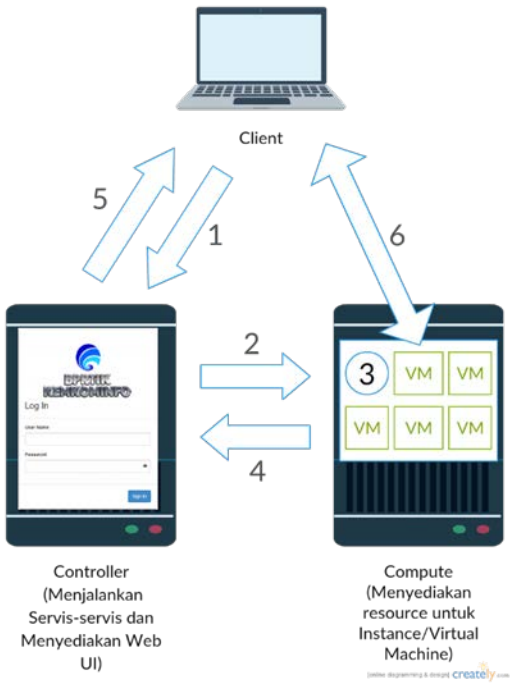
Tabel 4.1Kebutuhan Hardware

NO	*HARDWARE*	KETERANGAN
1	*Server* 2 *pcs*	Sebagai penyedia *resource cloud*
2	NIC 2 *pcs*	Sebagai penghubung ke jaringan intranet Pustiknas, masing-masing node menggunakan satu buah NIC
3	Laptop	Sebagai *client*

Kebutuhan *software*:

1. Ubuntu Server 14.04. Sebagai sistem operasi untuk *server*. Dipilih versi 14.04 dibanding versi 15.10 (terbaru) karena versi ini adalah versi *Long Time Support* (LTS)
2. Open SSH Server. Digunakan untuk melakukan *remote* dari *client* ke *cloud* node
3. Git. Digunakan untuk mengambil *source code* Openstack terbaru langsung dari Github
4. Openstack. Versi yang digunakan adalah versi “Kilo” dimana versi ini sudah stabil dan sudah banyak digunakan.
5. NTP. NTP akan di-install di *controller* node dan *compute* node agar waktu dapat tersingkronisasi.
6. MySQL. Dipilih *database* ini karena dukungan dari komunitas yang tinggi Selain itu, diketahui bagaimana menggunakannya dan command-command MySQL.
7. Python-mysqldb. dibutuhkan *package* ini karena sebagian besar *source code* openstack tertulis dengan phython *database* yang digunakan adalah MySQL.
8. RabbitMQ. Sebagai protokol untuk saling berkirim pesan antar node dalam infrasturtukr *cloud*.
9. Qemu. Sebagai *tools* untuk menjalankan virtualisasi. Dipilih Qemu dibanding KVM karena KVM membutuhkan teknologi *hardware assisted* dari Intel-VT / AMD-VT dimana spesifikasi server yang digunakan belum memiliki fitur ini, sementara Qemu tidak membutuhkan teknologi *hardware assisted.*
10. LVM. Sebagai protokol *block storage driver*.

4.2. Design



Gambar 4.1 Alur pembuatan Instance / Virtual Machine

Seperti yang dapat dilihat pada gambar 4.3, sistem dapat dijelaskan dengan alur sebagai berikut:

1. *User / client* melakukan login kepada web UI yang tersedia di node c*ontroller*, setelah berhasil login, user dapat melakukan perintah pembuatan Instance pada web UI tersebut.
2. Setelah itu, *service* Openstack di *controller* node akan melakukan *request* pembuatan Instance pada *compute* node.
3. Node *compute* akan melakukan pembuatan *instance* / VM.
4. Setelah berhasil, node *compute* akan mengirimkan *endpoint* pada node *controller* sebagai identitas dari *instance* / VM yang dibuat.
5. Setelah itu, *instance* / VM yang dibuat akan menunjukan status active pada web UI Openstack.
6. *User / client* pun dapat mengakses *instance* / VM via SSH / VNC.

4.3. Simulasi Prototipe

Dengan membuat *simulation prototype* pada *software* Oracle VM Virtualbox, dapat dibuat *snapshot* (*system restore)* sehingga ketika terjadi masalah, hanya perlu di-*restore snapshot* yang ada, tanpa perlu mengulang dari awal.

4.4. Implementasi

Pada tahap ini dilakukan beberapa konfigurasi berikut ini;

(a) Pada controller node dilakukan konfigurasi pada file keystone.conf, glance-api.conf, glance-registry.conf, nova.conf, neutron.conf,, cinder.conf dan local_setting.py.

(b) Pada compute node dilakukan konfigurasi pada file nova.conf dan neutron.conf.

Setelah melakukan konfigurasi tersebut, hal selanjutnya yang dilakukan adalah Melakukan singkronisasi *database* komponen-komponen Openstack. Lalu me*restart service-service* Openstack.

5. HASIL DAN PEMBAHASAN

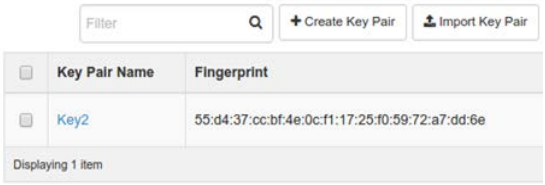
Setelah fase implementasi berhasil dilakukan, tahap selanjutnya yang dilakukan adalah melakukan monitoring. Pada tahap monitoring, dilakukan pengecekan terhadap implementasi yang telah dilakukan. Beberapa hal yang dilakukan adalah sebagai berikut:

(a) Membuat flavor. Setelah login berhasil. Pilih menu Admin -> Sistem lalu pilih menu “Flavor”, lalu pilih “Create Flavor”.

Flavor Name	VCPUs	RAM	Root Disk
m1.nano	1	64MB	0GB
m1.micro	1	128MB	0GB
m1.tiny	1	512MB	1GB
mini flavor	1	512MB	20GB

Gambar 5.1 Flavor Berhasil Dibuat

(b) Membuat keypair. Untuk membuat *keypair*, masuk menu *project* -> *compute* -> *access* & *security*, pilih tab "Key Pairs" lalu klik tombol "Create Key Pair". Lalu akan muncul sebuah *pop up* untuk mengisi nama *keypair* yang akan dibuat. Setelah memberi nama *keypair*, klik "Create Key Pair" yang ada pada *pop up* tersebut.



Gambar 5.2 Keypair Berhasil Dibuat

(c) Menambah *image*. Untuk menambah sebuah *image* pada sistem infrastruktur *cloud* openstack, pilih menu *project* -> *compute* -> *image*, klik tombol "create *image*".

Image Name	Type	Status
Debian Cloud	Image	Active
cirros-0.3.4-x86_64-uec	Image	Active
cirros-0.3.4-x86_64-uec-ramdisk	Image	Active
cirros-0.3.4-x86_64-uec-kernel	Image	Active

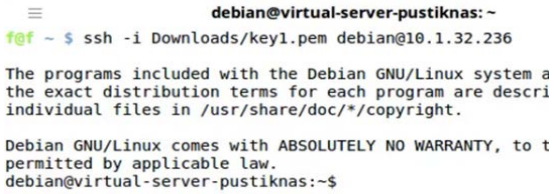
Gambar 5.3 *Image* Berhasil Ditambah

(d) Membangun dan Mengakses *Instance*. Untuk membangun sebuah *instance*, pilih menu *project* -> *compute* -> *instances*, lalu klik tombol "Launch *Instance*" dan akan muncul sebuah *pop up* pembuatan *instance*. Setelah muncul *pop up* tersebut, isi kolom-kolom yang ada dengan pada *tab details* tersebut. Setelah tab *details* sudah terisi semua, pindahkan tab aktif pada tab "*access* & *security*" lalu pilih keypair yang sebelumnya telah dibuat. Setelah itu, klik tombol "*Launch*" agar pembuatan *instance* dimulai.

Instance Name	Image Name	IP Address	Size	Key Pair
Virtual Server Pustiknas	Debian Cloud	10.1.32.236	mini flavor	key1
Displaying 1 item				

Gambar 5.4 *Instance* Berhasil Dibangun

Setelah *Instance* berstatus *running*, *instances* dapat diakses melalui SSH dengan menggunakan keypair yang dipilih sebelummnya.



Gambar 5.5 Mengakses *Instance* via SSH

Setelah fase monitoring, fase yang selanjutnya dilakukan adalah fase manajemen. Berikut ini adalah beberapa aktivitas yang akan dilakukan dalam fase ini:

(a) Manajemen *User*. Untuk memiliki sistem yang aman, *user* biasa diberi *role user* untuk membatasi akses terhadap sistem. Sebalikanya, seorang admin membutuhkan *role* administrator untuk mengurus dan memelihara sistem infrastruktur *cloud* yang telah dibuat sehingga administrator dapat melihat keseluruhan *user* yang ada.

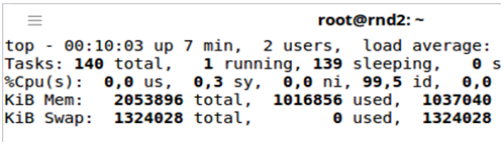
(b) Manajemen *Resource*. Uuntuk mengawasi penggunaan *resource*, seorang administrator

harus seringkali melihat *resource* usage yang digunakan.

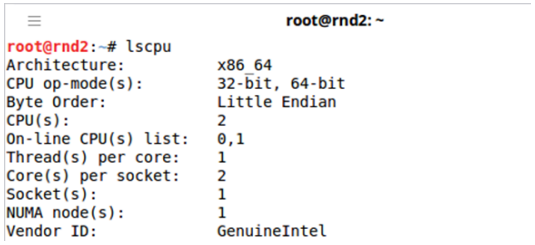
(c) Manajemen *System*. Sebuah sistem yang baru diterapkan biasanya memiliki masalah-masalah yang baru terlihat pada saat menggunakannya. Oleh karena itu, dibutuhkan aturan dan langkah-langkah yang harus dilakukan agar *user* tidak terganggu dengan masalah yang ada. Adapun aturan dan langkah-langkah tersebut adalah sebagai berkut:

1. *User* yang menemui masalah pada sistem melaporkan pada administrator mengenai masalah yang dirasakannya.
2. Admin yang menemui masalah atau mendapat laporan dari *user* mengenai masalah memperbaiki sistem secepat mungkin.
3. Administrator mengikuti perkembangan sistem dan versi Openstack terbaru yang dirilis setiap enam bulan sekali, dan melakukan *update* dalam jangka waktu agar dapat terus mendapat sup*port* pada sistem openstack yang digunakan.

Berikut ini adalah spesifikasi dan penggunaan *resource* RAM dan CPU pada server Pusat TIK Nasional di ruang RnD 2 (*Research and Development*).

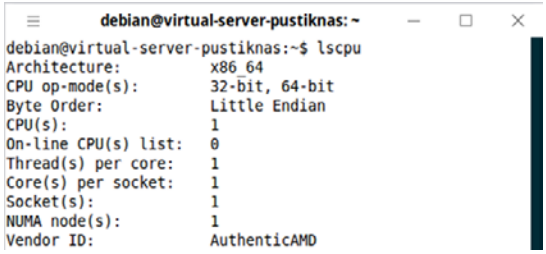


Gambar 5.6 Penggunaan RAM *server* RnD 2 tanpa *cloud*



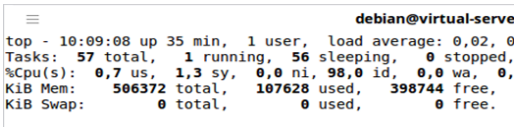
Gambar 5.7 Spesifikasi CPU *server* RnD 2 tanpa *cloud*

Berikut ini adalah tampilan spesifikasi instance setelah dibangun dengan menjalankan perintah "lscpu".



Gambar 5.8 Spesifikasi yang digunakan oleh *instance*

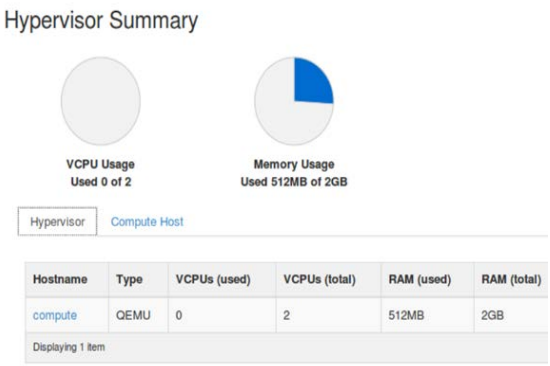
Lalu penggunaan dan spesifikasi RAM pada instance dapat dilihat pada gambar dibawah ini.



Gambar 5.9 Spesifikasi RAM yang digunakan oleh *Instance*

Seperti yang dapat dilihat pada gambar 5.19, RAM dengan infrastruktur *cloud* yang tidak digunakan sebesar 398.744 bytes. Sementara itu penggunaan RAM tanpa infrastruktur *cloud* adalah sebesar 1.037.040 bytes.

Dari pengujian diatas dapat diketahui bahwa infrastruktur IT yang dibangun dengan Openstack di Pusat TIK Nasional mampu melakukan efisiensi *resource* sebesar 1 buah CPU dan RAM sebesar 638.296 bytes.



Gambar 5.10 Resource yang di-*sharing* oleh *compute* node

Dari gambar 5.11, dapat dilihat bahwa *hostname* compute node yang menggunakan *hypervisor* qemu berhasil melakukan *sharing resource* sebesar 2 VCPU (*Virtualized* CPU) dan 2 GB *memory* yang merupakan spesifikasi dari *server compute* node.

6. PENUTUP

6.1. Kesimpulan

Berdasarkan hasil analisa, perancangan dan implementasi yang telah dilakukan, dapat disimpulkan bahwa:

1. Infrastruktur *Private Cloud* yang dirancang dengan Openstack di Pusat TIK Nasional mampu melakukan efisiensi sumber daya infrastruktur IT sebesar 1 CPU dan 638.296 bytes RAM.
2. Node *compute* pada Infrastruktur *Private Cloud* yang dirancang dengan Openstack di Pusat TIK Nasional mampu melakukan *sharing resource* infrastruktur IT.

6.2. Saran

Sebagai pengembangan dari penelitian yang telah dilakukan sebelumnya, diberikan saran sebagai berikut:

1. Karena pada penelitian kali ini digunakan spesifikasi komputer *desktop* / personal *computer* (PC). Untuk pengembangan selanjutnya, sistem diharapkan dibangun menggunakan komputer dengan spesifikasi *server*, dengan itu *performance* dari sistem infrastruktur *cloud computing* dapat berjalan lebih baik lagi.
2. Untuk pengembangan selanjutnya diharapkan untuk menambah node *compute* agar tercipta skalabilitas yang lebih besar lagi.

REFERENSI

[1] Jiang, Q. (2013). CY13-Q3 Community Analysis - OpenStack vs OpenNebula vs Eucalyptus vs *Cloud*Stack. *University of Sydney*. Sumber: http://www.qyjohn.net/?p=3373

[2] OpenStack. (2015). OPENSTACK *USER* SURVEY : A snapshot of OpenStack *user*s ' attitudes and deployments. *Openstack.org*.

[3] Pratama, I. P. A. E. (2014). *Smart City beserta Cloud Computing dan Teknologi-teknologi Pendukung Lainnya*. Bandung: Penerbit Informatika Bandung.

[4] Afdhal. (2013). *Studi Perbandingan Layanan Cloud Computing*. *Rekayasa Elektrika*.

[5] Budiyanto, A. (2012). *Pengantar Cloud Computing*. *Cloud Indonesia, Jakarta.*

[6] Fetria, D. F., Wibowo, T. A., Purnomo, J., Elektro, F. T., Telkom, U., & Engineering, S. (2015). Implementasi dan Analisis Platform Horizon Dalam *Service* Dashboard Berbasis Openstack. *Universitas Telkom*.