Teknologi Virtual pada Infrastruktur Komputasi Awan Privat dengan MiniPC berbasis OpenSource untuk Pembelajaran

Pujianto Yugopuspito, I Made Murwantara, Frans Panduwinata
Teknik Informatika, Fakultas Ilmu Komputer, Universitas Pelita Harapan
Jl. M.H. Thamrin Boulevard 1100 Lippo Village, Tangerang, Banten 15811
e-mail: {yugopuspito, made.murwantara, frans.panduwinata}@uph.edu

Abstrak

Infrastruktur komputasi awan memiliki kemampuan untuk mengadopsi arsitektur di dalam suatu lingkungan virtual. Seluruh sistem komputasi di dalam infrastruktur terisolasi dan tidak mempengaruhi satu dengan yang lain, sehingga keamanan dan kinerja masing-masing obyek virtual tidak mempengaruhi satu dengan yang lain. Tulisan ini melaporkan hasil eksplorasi implementasi pada lingkungan komputasi awan privat (private cloud) mempergunakan beberapa metode deployment, Docker dan Virtual Machine, pada arsitektur komputasi awan berbasis OpenSource ala OpenStack, yaitu dengan menerapkannya pada beberapa MiniPC Next Unit Computing. Hasil implementasi memperlihatkan kemampuan infrastruktur komputasi awan untuk mengoperasikan berbeda jenis metode virtualisasi dalam satu arsitektur yaitu mesin virtual (virtual machine) dan Docker. Secara umum mesin virtual lebih mudah dikerjakan dan menggunakan CPU load lebih rendah. Lebih lanjut akan membuka wawasan dalam penggunaan teknologi ini sebagai perangkat untuk melakukan akselerasi penelitian dalam berbagai bidang yang berbeda dan tentunya dimaksudkan sebagai sarana penunjang pembelajaran teknologi awan.

Kata kunci: *Komputasi Awan, Teknologi Virtual, Virtual Machine, Docker*

1. Pendahuluan

Satu dekade belakangan ini kebutuhan atas komputasi awan (*cloud computing*) [1] telah meningkat dengan sangat cepat. Beberapa penyedia jasa komputasi awan telah mengantisipasi dengan meningkatkan fleksibilitas interkoneksi antara infrastruktur awan publik (*public cloud*), seperti Amazon EC2, IBM Cloud dan GoogleCloud, dengan awan privat (*private cloud*), serta OpenStack dan Eucalyptus, mempergunakan berbagai teknik komunikasi. Kebutuhan awan privat juga telah meningkat, terutama pada lingkungan pendidikan dan riset, sehingga implementasinya perlu disederhanakan agar dapat diadopsi oleh kalangan luas.

Beberapa penelitian [2], [3], [4] pada komputasi awan memperlihatkan bahwa konfigurasi perangkat keras sangat menentukan model operasional yang dapat dilakukan, terutama guna mendukung berbagai macam penelitian yang mempergunakan komputasi secara virtual. Hal lain yang juga menjadi perhatian adalah dukungan untuk melaksanakan pembelajaran dalam merancang, merangkai, mengimplementasikan dan mengoperasikan komputasi awan bagi mahasiswa dan peneliti di universitas. Kami membangun komputasi awan ini dengan mempertimbangkan kebutuhan-kebutuhan tersebut. Lebih lanjut, infrastruktur awan privat yang dibangun juga harus mudah untuk dilakukan dan diulang prosesnya, sehingga mahasiswa dan peneliti dapat melakukannya tanpa kesulitan, dan dengan biaya yang terjangkau, serta hemat energi. Berdasarkan kebutuhan itu, kami mempergunakan sumber dari OpenSource, dengan suatu arsitektur awan yang telah banyak digunakan oleh kalangan organisasi penelitian dan komersial.

Jika kita fokus pada lingkungan akademik, kebutuhan komputasi awan sebagai perangkat untuk membantu peningkatan produktivitas dalam melaksanakan penelitian telah menjadi suatu hal yang tidak dapat ditinggalkan. Tulisan ini membagikan pengalaman kami mengimplementasikan berbasis OpenSource, yang lebih dikenal dengan nama produk OpenStack, untuk mendukung operasional penelitian yang sedang dilakukan. Kami akan menampilkan, pada tulisan, pengalaman dalam melakukan deployment dan pengujian serta arsitektur dari sistem awan yang kami pergunakan. Lebih lanjut, kami juga akan menjabarkan karakteristik dasar dari arsitektur awan guna penelitian penggunaan energi listrik.

Pada tulisan ini, pertama kami membekali pembaca dengan metodologi penelitian dan pengetahuan dasar mengenai komputer awan, virtualisasi serta implementasi. Kemudian, arsitektur dan

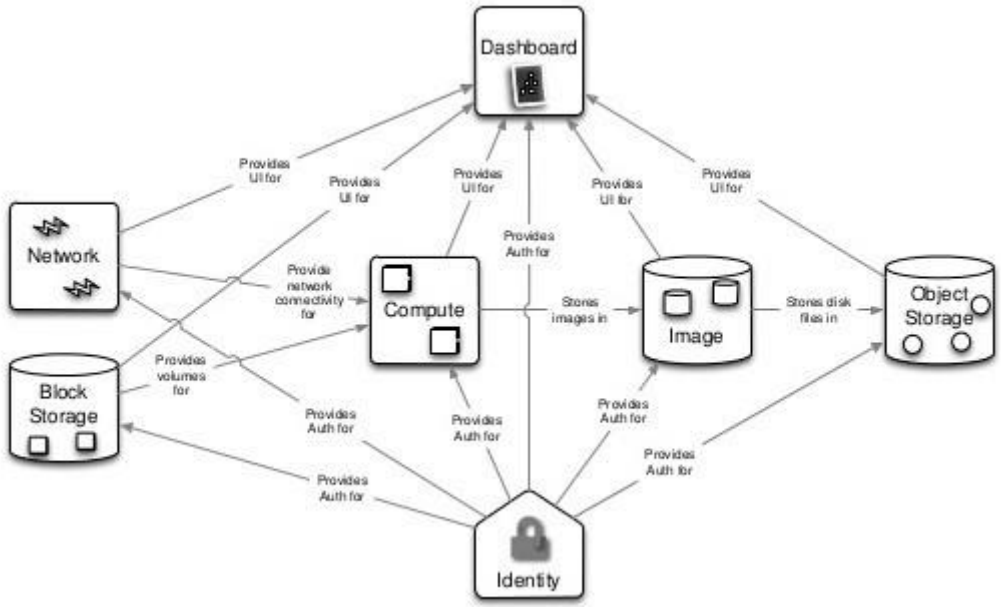
konfigurasi akan dijelaskan secara lebih detail. Setelah itu, teknik pengujian dan evaluasi akan dideskripsikan. Terakhir, kami akan menampilkan penutup dan penelitian lanjutan yang mungkin bisa dilakukan mempergunakan hasil penelitian ini, dan dukungan dalam pembelajaran teknologi awan.

2. Metode Penelitian

Metode penelitian yang digunakan adalah eksperimental. Suatu bentuk penelitian eksplorasi terhadap model dari sistem atau arsitektur, dalam hal ini adalah komputasi awan yang dibangun dengan mengunakan MiniPC. Kriteria dari keberhasilan dari penelitian ini adalah suatu sistem virtual yang berhasil dijalankan pada komputasi awan tersebut. Ada dua jenis pengujian yaitu pengujian operasional instalasi dan pengujian mempergunakan beban kerja. Pada pengujian pertama, menguji keberhasilan mengakses setiap perangkat virtual dan manajemen komputasi awan, memastikan tidak mengalami keterlambatan yang diketahui. Pengujian kedua, mempergunakan suatu alat ukur berupa piranti lunak untuk mengukur penggunaan energi listrik dalam perangkat virtual tersebut.

Pada bagian ini, kami perlu menjabarkan secara singkat mengenai teknologi komputasi awan dan virtualisasi yang telah dipergunakan dalam penelitian ini. Komputasi awan umumnya dilayani oleh suatu pusat data (*data center*) dan merupakan kumpulan dari banyak perangkat computer server yang saling terhubung mempergunakan teknologi jaringan komputer dan komunikasi data kecepatan tinggi. Salah satu contoh komputasi awan adalah OpenStack, yang dapat dipergunakan secara bebas, atau bersifat OpenSource, dan dapat dikembangkan bagi keperluan penelitian. OpenStack juga merupakan basis yang dipergunakan untuk membangun teknologi komputasi awan komersial seperti IBM Cloud [5] dan GoogleCloud [6].

Arsitektur komputasi awan Openstack, seperti diperlihatkan pada Gambar 1, terdiri dari beberapa bagian yang saling berhubungan: *compute node, object storage, block storage, network, image, identity* dan *dashboard.*

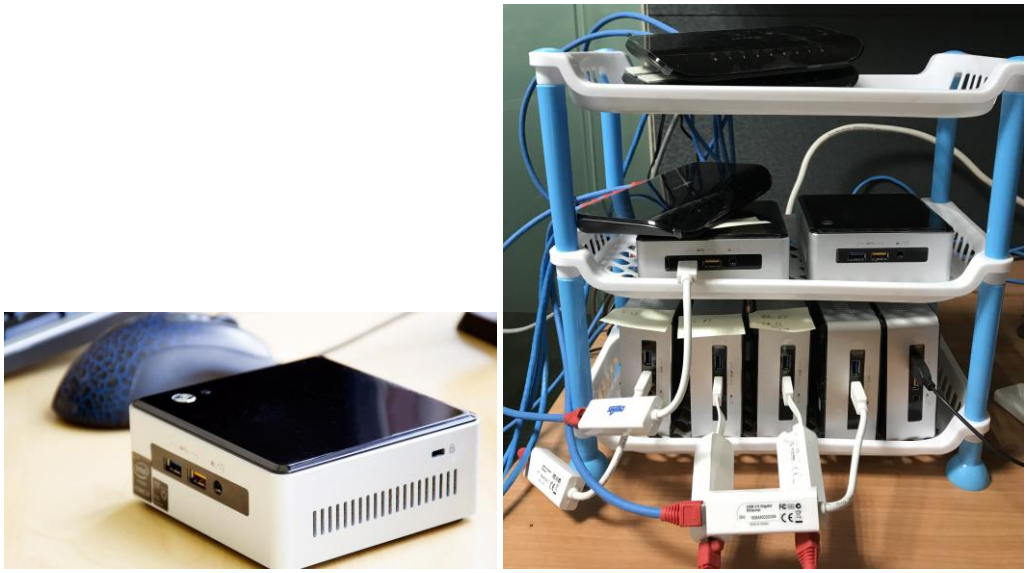


Gambar 1. Arsitektur Dasar Komputasi Awan OpenStack [7]

1. *Compute Node*, perangkat komputer server yang bekerja sebagai penyedia komputasi dan umumnya melakukan sharing resources seperti CPU, *Memory* dan *Storage*. *Compute node* umumnya berjumlah banyak dan menjadi bagian terpenting untuk melakukan aktifitas Cloud.
2. Object Storage, tempat menyimpan seluruh file yang umumnya berbentuk *image* untuk virtualisasi, misalkan ISO atau Kernel-based Virtual Machine (KVM).
3. *Block Storage* untuk menangani penyimpanan file untuk operasional.
4. Network sebagai media komunikasi
5. *Image* adalah bentuk dari system operasi berbentuk virtual.
6. *Identity* untuk menangani keamanan dan isolasi dari setiap obyek di dalam Cloud
7. *Dashboard* merupakan antarmuka berbentuk *Graphical User Interface* (GUI) dan terminal bagi pengguna dan administrator.

Sementara virtualisasi merupakan teknik untuk mempergunakan perangkat komputasi, seperti komputer server, secara terdistribusi sehingga pengguna dapat menggunakan aplikasi yang dibutuhkan secara bersama (*sharing*) dan terisolasi dengan menyembunyikan karakteristik perangkat yang menyediakan komputasi tersebut. Virtualisasi pada perangkat komputer dapat dibagi menjadi dua, yaitu pertama virtualisasi secara penuh dengan melakukan simulasi perangkat keras untuk mengoperasikan piranti lunak, umumnya terdiri dari system operasi yang ditumpangkan diatasnya. Kedua, paravirtualisasi melakukan isolasi untuk mengoperasikan piranti lunak yang ditumpangkan ke atasnya, akan tetapi perangkat keras yang menjalankannya tidak melakukan simulasi, sehingga seolah-olah ada system operasi terpisah yang berjalan diatasnya.

MiniPC adalah perangkat keras *miniature personal computer* (PC) dengan kebutuhan daya listrik yang rendah. Berukuran kecil 4 x 4 inch, dengan harga sepersepuluh dari *real server*, IntelNUC. Gambar 2 mengilustrasi MiniPC yang digunakan.



Gambar 2. MiniPC

3. Hasil dan Pembahasan

Pada bagian ini, kami akan berdiskusi mengenai proses implementasi, hasil dan pengujian. Evaluasi dilakukan dalam bentuk penggunaan awal arsitektur sebagai tempat melakukan penelitian untuk identifikasi konsumsi energi pada lingkungan infrastruktur virtual. Pada penelitian ini *deployment* arsitektur komputasi awan mempergunakan *tool* otomasi Fuel untuk OpenStack [8] yang membantu melakukan implementasi mulai dari konfigurasi hingga installasi sistem operasi dan aplikasi yang dibutuhkan melalui koneksi jaringan internet.

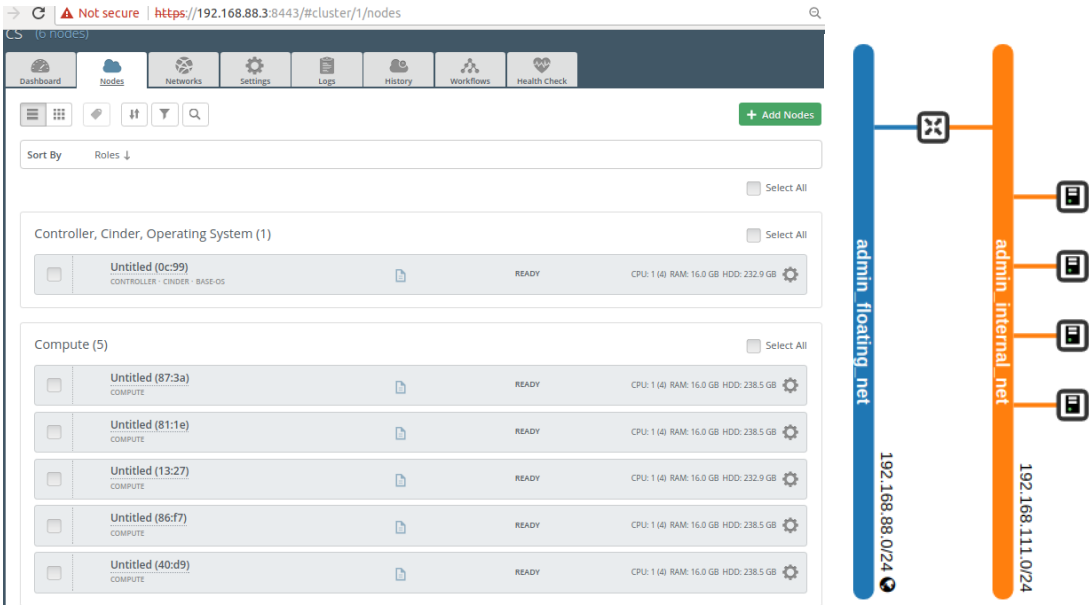
3.1. Konfigurasi Komputasi Awan

Kami mempergunakan 6 unit IntelNUC, yang mengkonsumsi daya sekitar 15 Watt untuk setiap unitnya, dengan konfigurasi, seperti diperlihatkan pada Gambar 3, satu unit sebagai *Controller*, *Storage* (Cinder) dan penyimpan *image* sistem operasi untuk mengatur akses dan penjadwalan serta menyimpan file operasional. Lima unit sebagai *compute nodes* yang juga *sharing storage* sebagai media penyimpan bagi komputasi awan privat, dimana masing-masing perangkat memiliki spesifikasi processor Intel i7 QuadCore, 16 GB memory dan minimal 240GB kapasitas penyimpanan dengan tipe *solid state drive* (SSD).

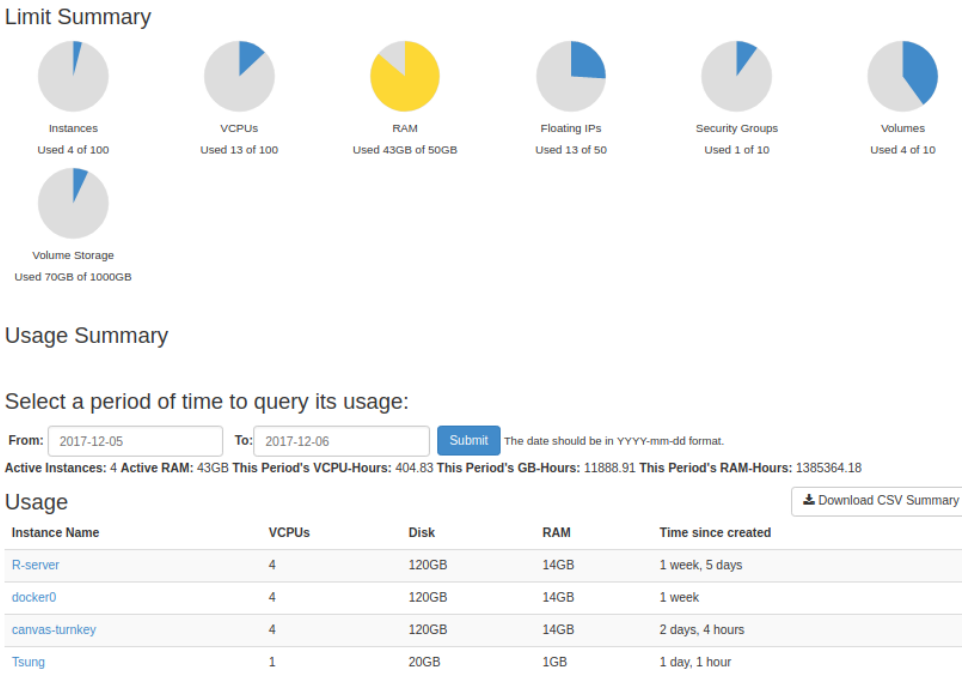
Jaringan koneksi dirancang memiliki dua bagian besar, yaitu *Internal* dan *External Network* yang diimplementasikan sebagai ***admin_floating_net*** dan ***admin_internal_net***, seperti terlihat pada Gambar 3. Pada jaringan internal dipergunakan alamat internet 192.168.111.0/24 dan jaringan eksternal 192.168.88.0/24, dimana kedua jaringan tersebut terhubung melalui sebuah Router Virtual yang juga berfungsi sebagai Firewall. Pengguna dapat mengakses aplikasi didalam Cloud melalui jaringan eksternal dan tidak dapat bersentuhan langsung dengan jaringan internal, sehingga dapat dikatakan bahwa Cloud merupakan internet-based system yang aman.

Pada penelitian ini, sejumlah *instance*, obyek virtual yang dapat beroperasi mandiri, berbasis sistem operasi OpenSource Linux: distribusi Ubuntu dan Debian. Masing-maing *instance* diinstal dengan aplikasi berbeda jenis, misalkan untuk Ubuntu versi Xenial (16.04) melakukan layanan Big Data dan Machine Learning mempergunakan aplikasi R [9] yang melekat dengan Shiny webserver, sehingga mampu

melakukan analisa data yang bersumber dari Twitter secara real-time guna melakukan pengujian Sentiment Analysis. Dan distro Linux Debian (versi Jessie) teraplikasi sebuah sistem *online learning*, Canvas [10], berbasis Bahasa pemrograman Ruby dan Rails yang dipasang mempergunakan virtual machine image siap pakai dari Turnkey. Sedangkan, Server Ubuntu lainnya terinstall aplikasi online-learning yang sama dengan versi Debian, akan tetapi berjalan dalam model virtualisasi Docker [11], [12]. Adapun bentuk implementasi ini dapat dilihat pada Gambar 4.



Gambar 3. Konfigurasi Komputer Server dan Jaringan Data pada Cloud Arsitektur Openstack



Gambar 4. Overview Seluruh Instance pada Implementasi OpenStack

3.2. Evaluasi

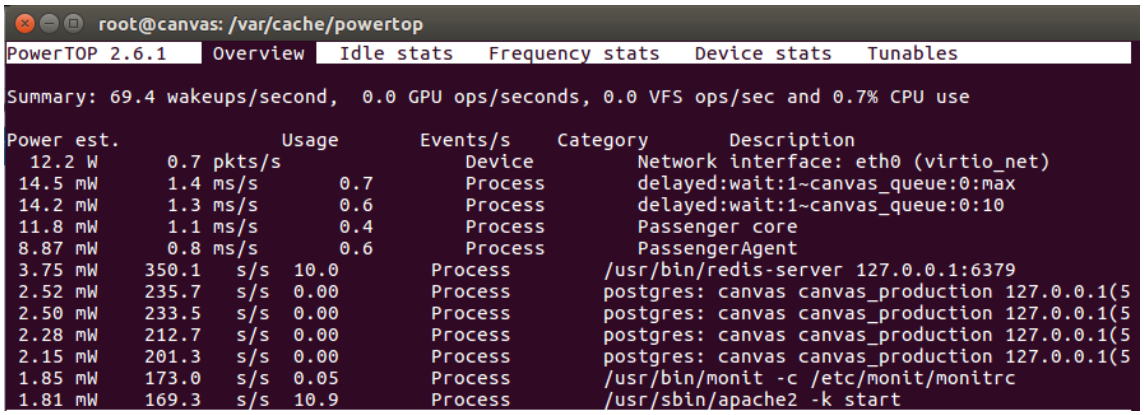
Kami melakukan evaluasi terhadap kinerja dan operasional arsitektur dengan menguji operasional *instance* yang terdapat didalam sistem. Pengujian dibagi menjadi dua (2), yaitu pengujian operasional instalasi dan pengujian mempergunakan beban kerja (*workload*). Pada pengujian pertama, kami mengakses alamat internet dari website setiap instance dan manajemen komputasi awan, dan ini berhasil karena seluruh website *instance* dan manajemen dapat diakses dengan mudah dan tidak mengalami keterlambatan yang diketahui (*delay*). Pengujian kedua, kami mempergunakan suatu aplikasi PowerTop yang umumnya

dilakukan untuk mengukur penggunaan energi listrik dalam perangkat virtual [13], seperti terlihat pada Gambar 5.

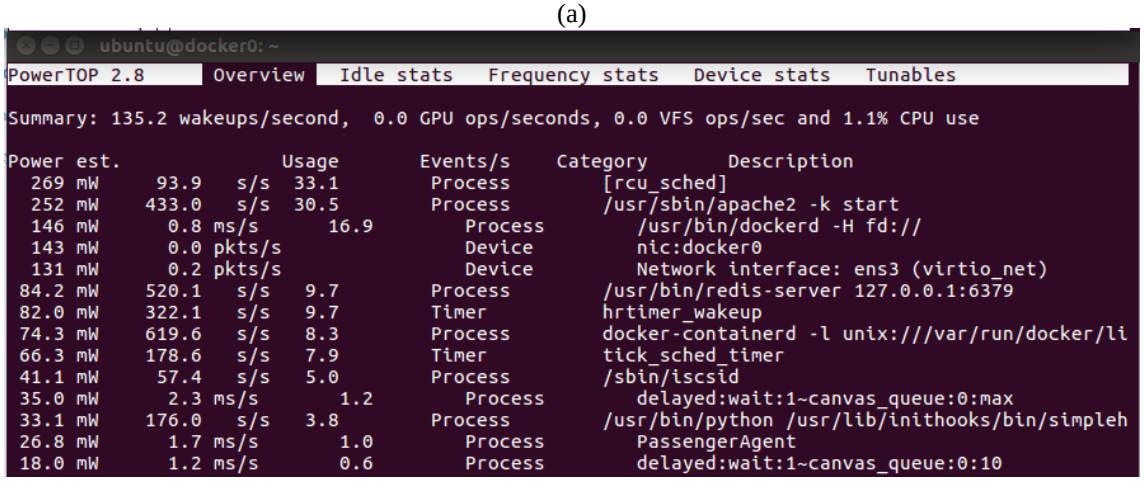
Gambar 5(a) memperlihatkan bagaimana sebuah instance dengan hanya Virtual Machine yang berjalan didalamnya memiliki banyak PID (process ID) dan mempergunakan CPU hanya 0.7% untuk memanggil seluruh proses didalam sistem dengan kemampuan memanggil fungsional sistem sebanyak 69.4 panggilan per detik. sementara pada Gambar 5(b) teknologi Docker mempergunakan 1.1% dari seluruh kapasitas CPU. Untuk memperlihatkan bahwa instance bekerja dengan model instalasi yang sama dapat dilihat pada detail proses Gambar 5, terlihat aplikasi *online-learning* Canvas berjalan dalam koneksi layanan web dan database. Pada Gambar 5(a) koneksi antara aplikasi dan pengguna langsung terhubung dengan sistem karena ditangani oleh VM dan tidak melalui tambahan model virtualisasi tambahan. Sementara pada Gambar 5(b), setiap koneksi yang terjadi akan ditangani, pertama oleh *instance* sebagai induk (*host*) kemudian diteruskan ke dalam sistem virtualisasi Docker sebelum mendapatkan layanan online-learning.

Perbedaan mendasar dari implementasi *virtual machine* (VM) dan Docker di dalam VM adalah sebagai berikut:

- Pada VM tidak akan dapat melakukan perubahan arsitektur di dalamnya, dan hanya melayani aplikasi sesuai yang terdapat di dalamnya.
- Pada Docker di dalam VM, pada intinya Docker merupakan virtualisasi yang melakukan *sharing resources* dengan *commonality sharing host*, sehingga di dalamnya bisa dikembangkan arsitektur yang mandiri dan terisolasi dari host VM, dalam hal ini OpenStack.
- Pada Docker dibutuhkan usaha lebih untuk melakukan instalasi dan konfigurasi virtual di dalamnya, serta dibutuhkan pengetahuan mengenai operasional sistem virtual pada komputasi awan.



(a)



(b)

Gambar 5. Hasil Pengujian Beban Kerja untuk Virtualisasi pada (a) Virtual Machine berbasis Kernel-based Virtual Machine (KVM) dan (b) KVM dengan Docker.

4. Simpulan

Kami telah menyampaikan sharing pengalaman dalam melakukan perencanaan, implementasi dan evaluasi penggunaan sistem komputasi awan Opensource. Penggunaan Fuel untuk OpenStack sangat memudahkan dalam melakukan penerapan komputasi awan pada perangkat komputer server hemat energi seperti IntelNUC. Berdasarkan hasil evaluasi yang telah dilakukan dapat disimpulkan bahwa arsitektur komputasi awan berbasis OpenStack yang terintegrasi sangat memudahkan dalam instalasi karena tidak banyak menghabiskan waktu dan *instance* yang dibuat cukup stabil selama dipergunakan tambahan catu daya batere yang digunakan sebagai tambahan dalam kondisi darurat jika listrik padam, guna menghindari kegagalan pada perangkat. Docker membutuhkan usaha lebih untuk melakukan instalasi dan konfigurasi virtual di dalamnya dibandingkan VM. CPU *load* Docker lebih tinggi daripada VM. Hal lain yang menarik adalah arsitektur OpenStack yang telah dibangun ini merupakan komputasi awan privat yang dapat dihubungkan dengan komputasi awan komersial atau sistem komputasi lainnya dengan melakukan konfigurasi pada API OpenStack, dimana kami merencanakan ini untuk langkah selanjutnya dari penelitian ini.

5. Ucapan Terima Kasih

Penelitian ini berhasil dikerjakan dengan bantuan pendanaan dari Lembaga Penelitian dan Pengabdian Masyarakat, Universitas Pelita Harapan dengan kontrak No. P-004-FIK/VI/2016.

Daftar Pustaka

[1] Mell P and Grance T. The NIST Definition of Cloud Computing NIST Special Publication 800-145. Available from: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf
[2] Mircea M and Adreescu AI. Using Cloud Computing in Higher Education: A Strategy to Improve Agility in the Current Financial Crisis, Communication of the IBIMA Vol. 2011, IBIMA Publishing. Available from: http://www.ibimapublishing.com/journals/CIBIMA/2011/875547/865547.pdf.
[3] Rahma N, Rochim AF dan Widianto ED. Analisis Implemntasi Infrastructure as a Service menggunakan Ubuntu Cloud Infrastruktur. Semarang [Tugas Akhir] Universitas Diponegoro; 2015. Available from: http://eprints.undip.ac.id/ 42295/1/4762-8937-1-PB.pdf.
[4] Sultan N. Cloud computing for education: A new dawn? International Journal of Information Management 2010; (30):109-116.
[5] IBM, IBM Cloud Manager with OpenStack [Internet] New York; IBM Knowledge Center [updated: 2017; cited: 10 December 2017] Available from: https://www.ibm.com/support/knowledgecenter/en/SST55W.
[6] Google, Google Cloud Platform. [Internet] Montain View; Alphabeth Inc. [updated: 2017; cited: 10 December 2017]. Available from: https://cloud.google.com.
[7] OpenStack, Open source source software for creating private and public clouds. [Internet] Austin; OpenStack Foundation [updated: 2017; cited: 10 December 2017] Available from: https://www.openstack.org.
[8] Fuel Infra, Fuel for OpenStack. [Internet] Sunnyvale; Fuel Community [updated 2017; cited: 10 December 2017] Available from: https://www.fuel-infra.org
[9] R-Project, The R Project for Statistical Computing. [Internet] Auckland; R Development Community [updated: 2017; cited: 10 December 2017] Available from: https://www.r-project.org.
[10] Canvas, Open Up with Canvas LMS. [Internet] Salt Lake City; Infrastructure Inc. [updated 2017; cited: 10 December 2017] Available from: https://www.canvaslms.com.
[11] Docker, Docker Community. [Internet] San Francisco; Docker Inc. [updated 2017; cited: 10 Dec. 2017] Available from: https://www.docker.com/docker-community
[12] Docker, Get started with Docker Machine and a local VM, [Internet] San Francisco; Docker Inc. [updated 2017; cited: 10 December 2017] Available from: https://docs.docker.com/machine/get-started/.
[13] Powertop, Intel OpenSource [Internet] Santa Clara; Intel Corporation [updated 2017; cited: 10 December 2017] Available from: https://01.org/powertop.