

Manajemen Adaptif Komputasi Awan untuk Efisiensi Energi

I Made Murwantara

*Teknik Informatika, Fakultas Ilmu Komputer, Universitas Pelita Harapan
Jl. M. H. Thamrin Boulevard 1100 Lippo Village Tangerang
Email: made.murwantara@uph.edu*

Abstrak. *Komputasi Awan (selanjutnya disebut Cloud Computing) melakukan distribusi pekerjaan pada setiap anggota cluster komputer secara terjadwal dan terorganisir. Akan tetapi disetiap Data Centre yang menyediakan layanan Cloud Computing umumnya mengoperasikan ribuan, bahkan lebih, komputer server tanpa memperhatikan apakah layanan sedang digunakan atau tidak. Hal tersebut mengakibatkan konsumsi listrik yang sangat besar dan kemungkinan tidak digunakan secara efisien. Tulisan ini menampilkan bagaimana efisiensi energi pada Cloud Computing dapat dilakukan secara adaptif, berdasarkan perubahan beban kerja, dengan memperhatikan konfigurasi piranti lunak. Pendekatan yang dilakukan mengadopsi dari metode Software Product Line Engineering dan pemodelan run-time. Disain sisi adaptif mempergunakan kemampuan model run-time untuk memberikan kesempatan rekonfigurasi dengan memperhatikan perubahan beban kerja. Perubahan yang dilakukan berupa rekonfigurasi piranti lunak pada Virtual Machine yang sedang berjalan secara terjadwal. Tingkat adaptif dikerjakan dengan pemodelan transisi yang mengadopsi teknik Rule-Based dimana referensi perubahan didapatkan dari suatu pelatihan data mempergunakan salah satu metode dari Machine Learning. Disain awal ini dapat di implementasikan pada lingkungan Virtual dan Cloud Computing.*

Kata kunci: *Cloud Computing, Software Product Line Engineering, Energi, Adaptif, Model Run-time.*

1. Pendahuluan

Konsumsi energi pada Data Centre yang memberikan layanan Komputasi Awan (selanjutnya disebut Cloud Computing) meningkat setiap tahunnya [1]. Secara global, konsumsi energi untuk Data Centre telah mencapai 416 Terra Watt Hour (TWh) pada tahun 2016 [2]. Untuk USA sendiri, per tahun rata-rata konsumsi daya listrik untuk operasional Data Centre mencapai 90.000 Mega Watt hour (MWh). Diprediksi, pada tahun 2025 konsumsi energi listrik untuk seluruh Data Centre didunia akan mencapai 1/5 dari konsumsi listrik global. Hal ini terjadi karena kebutuhan layanan komputasi baik virtual maupun on-premise meningkat. Salah satu pemicunya adalah layanan kantor virtual, seperti aplikasi perkantoran dan layanan perangkat bergerak seperti telepon genggam.

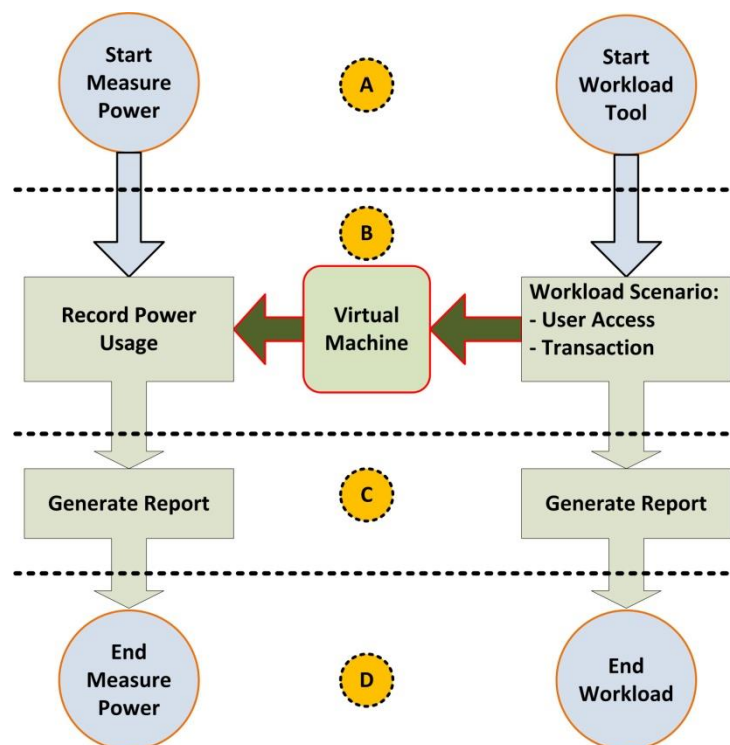
Cloud Computing dikenal sebagai bagian layanan dari suatu Data Centre yang menggabungkan komputer dalam jumlah kecil hingga ekstra besar. Umumnya kelompok komputer yang digabungkan pada suatu sistem Cloud dapat terdiri dari berbagai jenis perangkat keras dan piranti lunak. Beberapa tipe layanan Cloud antara lain, Infrastructure as a Service (IaaS) yang memberikan layanan perangkat keras, Software as a Service (SaaS) untuk layanan hanya piranti lunak dan Platform as a Service (Paas) untuk pelanggan yang membutuhkan sedikit manajemen perangkat keras dan memiliki layanan piranti lunak secara utuh. Elemen infrastruktur terkecil adalah Virtual Machine, yaitu operasional sistem yang bekerja selayaknya perangkat server secara virtual dan melekat pada perangkat keras dengan dijembatani oleh *hypervisor*.

Sebagian besar Data Centre mengoperasikan Cloud Computing sebagai layanan utama karena dapat memberikan keleluasaan bagi pengguna untuk membayar sesuai yang dengan kapasitas yang digunakan, atau lebih dikenal dengan layanan Pay-As-You-Go. Layanan lain yang juga sama menariknya adalah elastisitas, yaitu perubahan strategi dengan menambahkan atau mengurangi jumlah Virtual Machine yang dioperasikan berdasarkan beban kerja yang masuk kedalam aplikasi. Elastisitas pada Cloud ditentukan dengan melakukan konfigurasi sebelum beban kerja diterima oleh sistem atau pada saat belum beroperasi menerima beban.

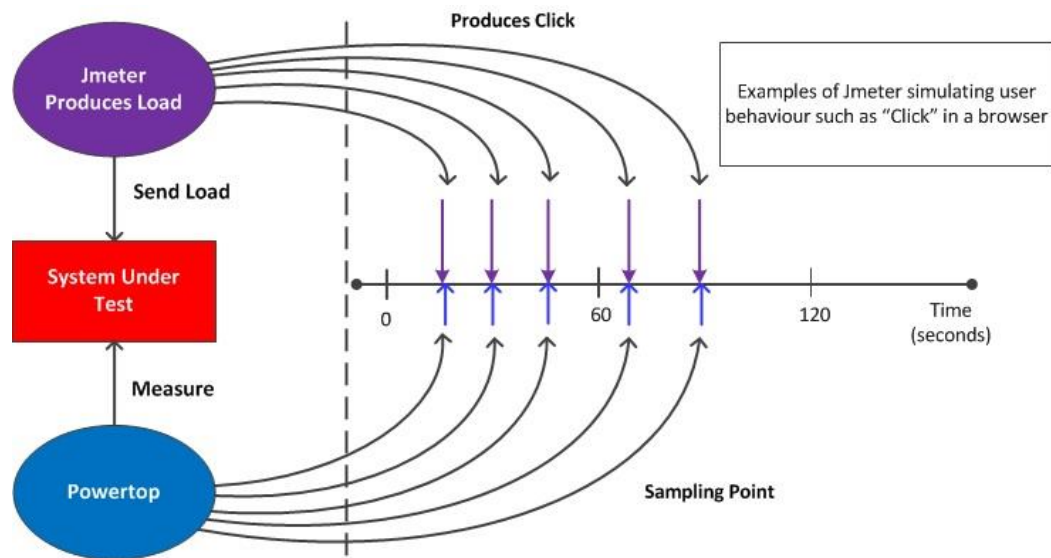
Elastisitas memberikan manajemen adaptif pada Cloud Computing secara perangkat keras, dimana penggunaan energi listrik tidak diperhatikan sebagai suatu karakteristik, melainkan sebagai pola yang membebani aplikasi sehingga diperlukan intervensi secara otomatis untuk menambahkan kemampuan infrastruktur [3]. Berbeda dengan pandangan diatas, penulis melihat bahwa kombinasi piranti lunak turut berperan aktif dalam memberikan utilisasi berupa proses pada sistem yang sedang bekerja [4]. Sebagai contoh, kombinasi web server dengan konfigurasi Ubuntu-Apache-MySQL

Artikel ini memberikan arahan dalam bentuk pemodelan adaptif untuk Cloud Computing. Bentuk adaptif yang dilakukan adalah dengan melakukan rekonfigurasi piranti lunak pada sistem Cluster yang sedang beroperasi dengan tidak menghentikan keseluruhan operasi yang sedang bekerja. Hal lain yang dikerjakan adalah dengan melakukan penjadwalan perubahan dengan melakukan otomasi pada Load-Balancer dan Manajer piranti lunak tanpa intervensi dari operator, secara autonomous. Rekonfigurasi piranti lunak dimodelkan oleh Software Product Line Engineering dalam bentuk Feature Model yang dijadikan referensi oleh Manajer piranti lunak.

Efisiensi energi tidak dapat dilakukan tanpa mengetahui konsumsi energi pada sistem. Untuk pengukuran energi secara fisik seperti komputer server dapat dilakukan secara langsung, akan tetapi untuk layanan virtual dibutuhkan teknik yang cukup berbeda. Kami telah berhasil melakukan pengukuran energi tanpa mempergunakan peralatan khusus pada beberapa penelitian kami terdahulu, yang dapat dibaca pada [3, 4, 5, 6]. Berdasarkan pengukuran yang telah kami dapatkan, dilakukan prediksi dengan pendekatan Machine Learning [4, 7] untuk memperkirakan jarak-antara kinerja dan kebutuhan energi pada kombinasi piranti lunak. Suatu piranti lunak Open Source, powertop, buatan Intel dipergunakan untuk mengukur energi pada virtual CPU (vCPU)[8]. Bentuk pengukuran energi diperlihatkan pada gambar 1. Adapun pengukuran energi memerlukan beban kerja yang menyerupai pengguna mengakses aplikasi. Kami telah mengembangkan teknik simulasi beban kerja guna memberikan beban pada sistem, seperti diperlihatkan pada Gambar 2.



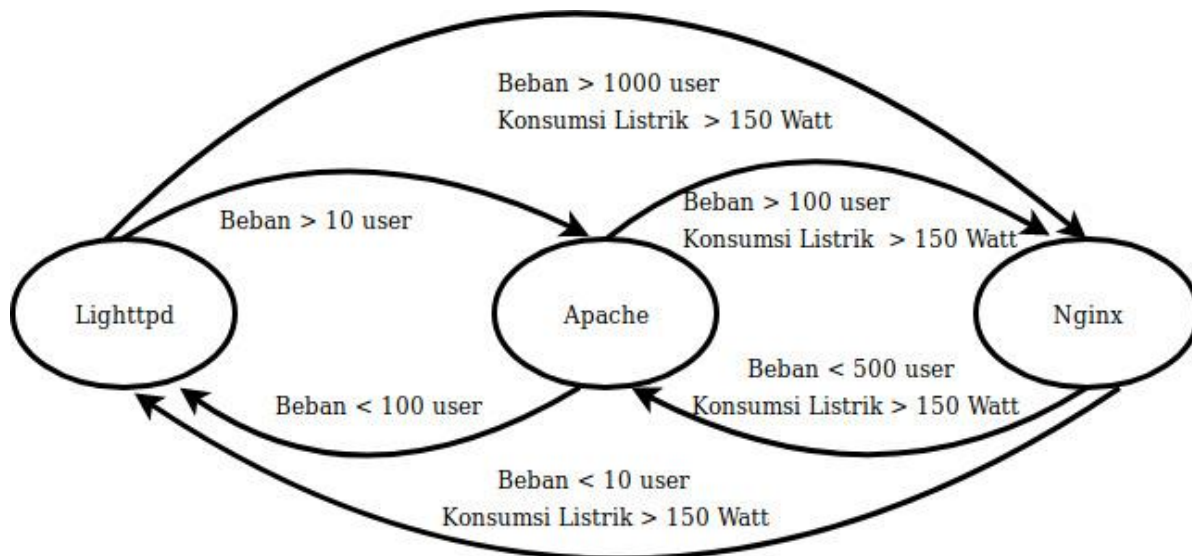
Prosedur pengukuran dilakukan dengan [A] beban kerja mulai dijalankan dan piranti lunak pengukuran energi juga diaktifkan bersamaan. Kemudian, [B] beban kerja mengakses Virtual Machine dengan skenario pengguna mengakses halaman web, login, memberikan komentar pada suatu halaman, dan keluar, pada saat bersamaan, energi yang dipergunakan di ukur dan disimpan. Setelah skenario berakhir, laporan hasil dibentuk ke dalam file [C] sebelum akhirnya pengukuran dihentikan [D].



Gambar 2. Beban Kerja untuk Pengukuran Sistem [3]

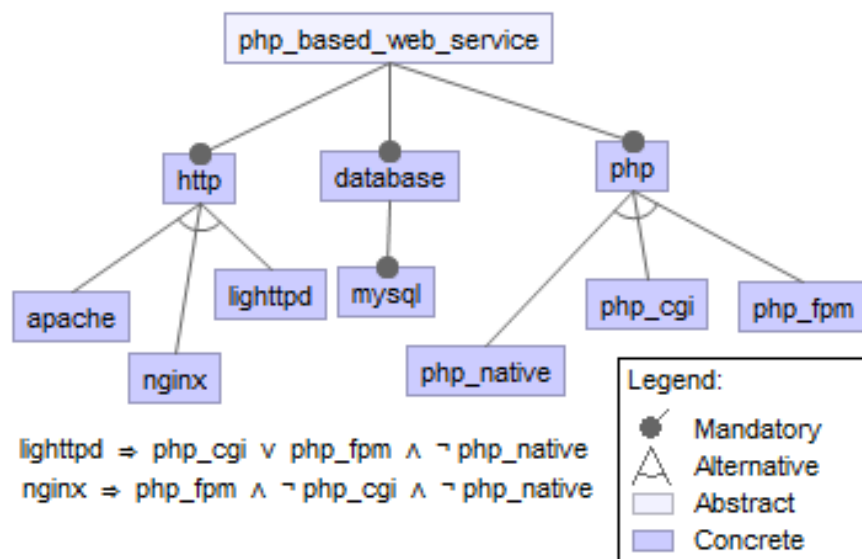
Umumnya, penyedia jasa Cloud Computing memiliki Load-Balancer untuk memudahkan manajemen Virtual Machine pengguna. Kami meningkatkan kegunaan Load-Balancer tidak hanya untuk mengatur Virtual Machine tetapi juga melakukan otomatisasi berdasarkan hasil pengukuran real-time. Hal ini memungkinkan digunakannya elemen dari Load-Balancer sebagai sensor yang terus melihat beban kerja yang berpengaruh pada kinerja layanan aplikasi. Sementara itu, pengukuran secara sampling pada Virtual Machine memperlihatkan konsumsi energi yang terjadi. Kemudian, kami melakukan kombinasi data yang dihasilkan pada Load-Balancer dan Virtual Machine untuk menentukan kapan suatu konfigurasi piranti lunak harus dilakukan, sehingga kemungkinan untuk melakukan elastisitas perangkat keras virtual dapat dikurangi. Yang selanjutnya dapat mengurangi konsumsi energi pada Data Centre.

Untuk melakukan otomatisasi rekonfigurasi diperlukan suatu mekanisme yang menyerupai model umpan-balik pada sistem kendali. Pemodelan otomatisasi ini dilakukan dengan bantuan diagram transisi, seperti diperlihatkan pada Gambar 3. Rekonfigurasi dilakukan dengan menggunakan dua parameter, yaitu beban kerja yang berhubungan dengan kinerja aplikasi dan daya listrik yang berelasi dengan kerja dari virtual CPU (vCPU). Pada parameter beban, jika pengguna yang mengakses sistem melebihi 10 pengguna per detik maka dilakukan transisi dengan melakukan konfigurasi web server mempergunakan Apache, jika melebihi 1000 pengguna per detik dan konsumsi listrik melebihi 150 Watt per Virtual Machine maka dilakukan rekonfigurasi mempergunakan web server Nginx. Sebaliknya, jika pengguna berubah hanya menjadi 10 dan konsumsi listrik 150 Watt, sementara Nginx beroperasi maka dilakukan rekonfigurasi menjadi Lighttpd.



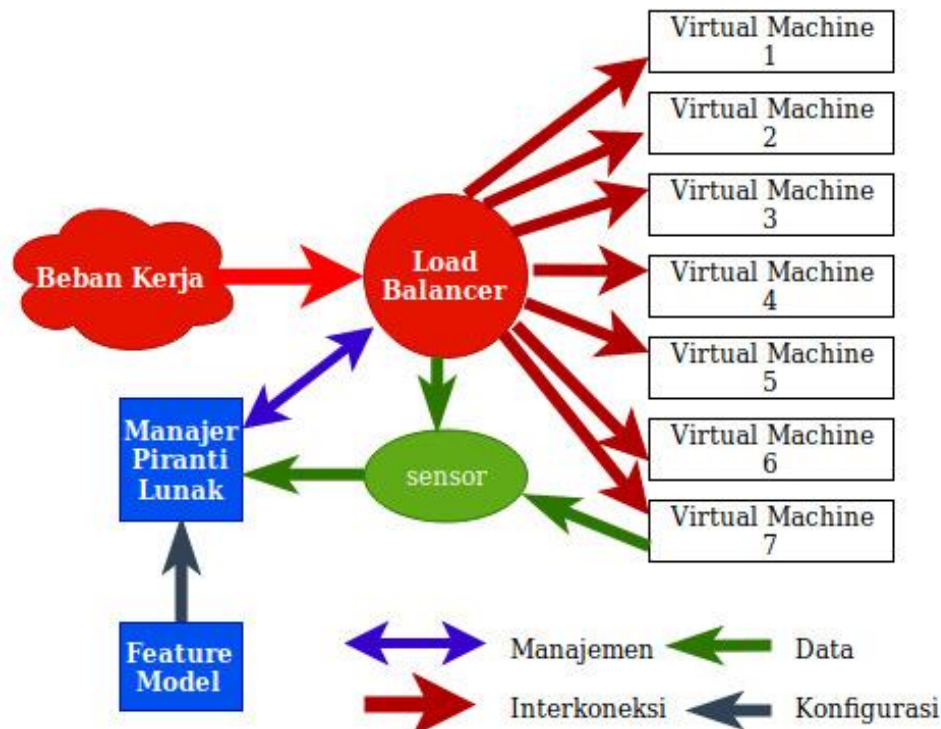
Gambar 3. Diagram Transisi untuk Manajemen Otomasi

Rekonfigurasi dikendalikan oleh Manajer piranti lunak dalam bentuk Feature Model (FM), diperlihatkan pada gambar 4. Dibawah gambar diperlihatkan constraint yang menjadi batasan dalam konfigurasi. Lighttpd tidak dapat mempergunakan feature `php_native`, begitu juga Nginx. Feature Model di implementasikan dalam bentuk package Ubuntu pada repository piranti lunak.



Gambar 4. Feature Model sebagai Referensi Arsitektur untuk Rekonfigurasi.

Pemetaan dilakukan antara transisi diagram dan feature model untuk mendapatkan keseluruhan dari sistem yang akan diterapkan. Bentuk aplikatif dari kombinasi ini adalah sistem manajemen terintegrasi dalam bentuk otomasi piranti lunak. Implementasi yang di usulkan mempergunakan Ansible sebagai media untuk melakukan rekonfigurasi piranti lunak. Dalam hal ini, Ansible [8] bertindak sebagai scripting tool yang melakukan instalasi/reinstalasi/deinstalasi dari seluruh package piranti lunak yang terdapat pada Feature Model. Untuk melakukan integrasi secara langsung dengan Cloud, Ansible diberikan akses untuk mengubah konfigurasi Load-Balancer secara on-the-fly tanpa interupsi. Hal ini dilakukan guna mendapatkan sistem yang tetap dapat memberikan layanan ketika tengah dilakukan rekonfigurasi piranti lunak pada Virtual Machine. Bentuk nyata dari pekerjaan ini adalah menghentikan layanan pada Virtual Machine yang sedang direkonfigurasi, serta setiap permintaan layanan baru dialihkan pada anggota Cluster yang lain.



Gambar 5. Model Cloud Computing Adaptif

Gambar 5 memperlihatkan keseluruhan desain dari sistem adaptif untuk efisiensi energi pada Cloud Computing. Kategori kerja dari desain ini adalah sebagai berikut;

Beban Kerja

Beban kerja meminta layanan berupa akses halaman web kepada Load-Balancer. Bentuk akses ini adalah pengguna berusaha masuk kedalam situs web. Pada situasi nyata, beban kerja adalah pengguna yang mempergunakan browser internet untuk mengakses suatu situs web dan berinteraksi dengan sistem dalam bentuk input atau output.

Load-Balancer

Load-Balancer melakukan perhitungan awal dengan melihat konfigurasi balancer dan jumlah layanan yang sedang diberikan untuk setiap Virtual Machine. Informasi jumlah permintaan akses dan beban kerja yang masuk dikirim dalam bentuk data sensor, yang kemudian oleh sensor diteruskan pada Manajer piranti lunak. Konfigurasi pada bagian ini memiliki beberapa pilihan metode pembagian kerja. Umumnya pembagian kerja dengan metode Round-Robin dan simetris banyak digunakan pada Cloud sistem. Hal lain yang dapat meningkatkan manajemen adaptif adalah Load-Balancer dapat melakukan rekonfigurasi secara real-time, tanpa mengganggu sistem.

Virtual Machine

Virtual Machine (VM) menerima perintah dari Load-Balancer dan memberikan layanan sesuai kapasitas konfigurasi virtual, misal: vCPU 4 core, RAM 64 GB. Setiap proses yang terjadi akan memberikan utilisasi pada vCPU yang akhirnya akan meningkatkan penggunaan energi. Informasi energi yang dikonsumsi dikirimkan sebagai data sensor yang diteruskan pada Manajer Piranti Lunak.

Sensor

Sensor menerima semua informasi yang diberikan oleh Load-Balancer dan Virtual Machine. Kemudian diteruskan ke Manajer Piranti Lunak. Sensor bisa berbentuk interaksi middleware yang berkomunikasi dalam bentuk topic-subscribe. Bentuk aplikasinya adalah RabbitMQ [11].

Manajer Piranti Lunak

Manajer Piranti Lunak menerima informasi dari sensor dan memutuskan konfigurasi yang akan diterapkan berdasarkan model transisi run-time yang telah dibuat. Referensi informasi didapatkan dari Feature Model yang diterjemahkan dalam bentuk Ubuntu Package Repository. Rekonfigurasi dilakukan bekerjasama dengan Load-Balancer. Bentuk aplikasi bagian ini adalah suatu mekanisme konfigurasi Ansible yang bekerja berdasarkan trigger dari data sensor.

Feature Model

Bagian ini memberikan referensi arsitektur bagi Manajer Piranti Lunak untuk mengatur rekonfigurasi aplikasi. Pendekatan Software Product Line Engineering (SPLE) dipergunakan sebagai landasan untuk mengembangkan arsitektur referensi.

Pada disain yang diusulkan, Manajer Piranti Lunak memainkan peranan penting untuk menentukan kapan dan konfigurasi aplikasi yang ingin dioperasikan pada sistem yang sedang berjalan. Pada interval waktu yang telah dijadwalkan informasi beban kerja dan konsumsi energi dikirimkan pada sensor yang diteruskan pada Manajer Piranti Lunak. Dengan mempertimbangkan keputusan mengenai konfigurasi piranti lunak yang akan dioperasikan, maka Manajer Piranti Lunak memiliki What-If skenario yang menyerupai Rule-Based System. Jika suatu kondisi terpenuhi maka akan dilakukan rekonfigurasi.

Permasalahan berikutnya yang akan timbul adalah berapa lama waktu yang diberikan atau threshold untuk menentukan suatu kondisi tercapai untuk dieksekusi atau mengalami rekonfigurasi. Hal ini berkaitan dengan kestabilan sistem setelah direkonfigurasi. Tingkat kestabilan arsitektur piranti lunak pada sistem adaptif menjadi isu berikutnya yang tidak termasuk dalam lingkup pekerjaan ini dan akan dilakukan investigasi untuk mengetahui pengaruhnya pada model yang telah dibuat ini.

3. Kesimpulan

Sebuah arsitektur dan mekanisme adaptif untuk efisiensi energi pada Cloud Computing telah diusulkan. Desain dari sistem memiliki perbedaan yang mendasar dari layanan elastisitas pada Cloud Computing, yaitu rekonfigurasi piranti lunak yang mempertimbangkan kinerja dan konsumsi listrik pada layanan virtual. Serta, dipergunakannya pengukuran energi dengan piranti lunak, sehingga memudahkan manajemen secara terintegrasi. Desain yang diusulkan ini masih perlu disempurnakan dalam bentuk pengaturan transisi yang lebih sistematis dan formal. Salah satu yang menantang adalah dengan melakukan formalisasi untuk Manajer Piranti Lunak, dimana formulasi matematis akan memberikan kepastian hasil dan pengujian secara model.

Daftar Pustaka

- [1]. Joao Marques Lima, Data Centres of the World will Consume 1/5 of Earth's Power by 2025, Desember 2017, diakses 4 Januari 2019
- [2]. Radoslav Danilak, Why Energy is a Big and Rapidly Growing Problem, FORBES, diakses 4 Januari 2019.
- [3]. I M Murwantara, B Bordbar, A Simplified Method of Measurement of Energy Consumption in Cloud and Virtualized Environment, BDCLOUD, 2014. DOI:10.1109/BDCLOUD.2014.47
- [4]. I M Murwantara, B. Bordbar, L. L. Minku, Measuring Energi Consumption for Web Service Product Configuration, IIWAS 2014. DOI: 10.1145/2684200.2684314
- [5]. I M Murwantara, P. Yugospito, Studi Konsumsi Energi Listrik Komputasi Awan pada Implementasi untuk Docker dan Virtual Machine, KNSI 2018.
- [6]. I M Murwantara, P. Yugospito, Evaluating Energy Consumption in a Different Virtualization Technology, ICST 2018. DOI:10.1109/ICSTC.2018.8528695
- [7]. I M Murwantara, B. Bordbar, J. B Filho, A Self-Adaptive Architecture with Energy Management in Virtualization, ICSIIT 2017. DOI:10.1109/ICSIIT.2017.18
- [8]. Powertop, Intel, <https://01.org/powertop>, diakses 4 Januari 2019.
- [9]. D. Hall, Ansible Configuration, Packt Publishing, 2013.
- [10]. Load-Balancer, Bowman-AMuah, M.K, Google Patents, <https://www.google.com/patents/US6578068>.
- [11]. G. M. Roy, RabbitMQ in Depth, Manning Publication, 2017.