Host Overloading Detection pada Dynamic VM Consolidation Menggunakan Fuzzy Mamdani

Host Overloading Detection on Dynamic VM Consolidation Using Fuzzy Mamdani

Chaerul Umam*[1], Guruh Fajar Shidik[2]
[1,2] Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
E-mail: ***[1]chaerul@dsn.dinus.ac.id, [2]guruh.shidik@dsn.dinus.ac.id**

Abstrak

Perkembangan Cloud Computing telah mengakibatkan pembangunan data center skala besar di seluruh dunia yang berisi ribuan node. Data Center Cloud mengkonsumsi energi listrik yang besar yang tentunya mengakibatkan biaya operasi yang tinggi. Konsumsi energi di Data Center akan terus tumbuh pesat kecuali dengan mengembangkan dan menerapkan manajemen resource yang hemat energi. Dynamic VM consolidation bisa menjadi strategi efektif untuk mengatasi masalah pemborosan energi pada data center cloud. Strategi ini dapat diuraikan ke dalam empat tugas pengambilan keputusan, yaitu Host overloading detection (memutuskan kapan host harus dianggap sebagai kelebihan beban), Host underloading detection (memutuskan kapan host harus dianggap underloaded / kekurangan beban), VM selection (memutuskan VMs mana yang harus pindah dari host yang kelebihan beban), dan VM placement (memutuskan tentang host mana yang harus dipilih untuk menerima migrasi VM). Penelitian ini mengusulkan metode fuzzy logic dalam proses host overloading detection. Dataset untuk menguji metode menggunakan data workload dari PlanetLab. Hasil dari pengujian metode yang diusulkan menunjukkan hasil yang menjanjikan dengan peningkatan efisiensi energi 2,24%.

Kata Kunci *— Efisiensi Energi, Cloud Computing, Resource Manajemen, Host Overload Detection, Dynamic VM Consolidation*

Abstract

Cloud Computing developments have resulted in the construction of large-scale data centers around the world which contains thousands of nodes. Cloud Data Center consume large electrical energy which would result in high operating costs. Energy Consumption in the Data Center will continue to grow rapidly unless by developing and implementing energy-efficient resource management. Dynamic VM consolidation can be an effective strategy to overcome the problem of energy waste in the Cloud Data Center. This strategy can be decomposed into four decision-making tasks, namely overloading detection Host (decide when the host must be considered as overloaded), Host underloading detection (decide when the host must be considered underloaded), VM selection (decide which VMs should move from host overloaded), and VM placement (decide hosts which should be chosen to receive VM migration). This study proposes a method of fuzzy logic in the process of host overloading detection. Dataset to test the proposed method using workload data from PlanetLab. The results of measurements the proposed method shows promising results with up to 2.24 % of energy efficient improvement

Keywords *— Energy Efficiency, Cloud Computing, Resource Manajemen, Host Overload Detection, Dynamic VM Consolidation*

Dikirim: 01 Maret 2017; Direvisi: 23 Maret 2017; Diterima: 12 Juli 2017

1. PENDAHULUAN

Cloud Computing telah merevolusi industri teknologi informasi dan komunikasi dengan memungkinkan penyediaan *resource* komputasi yang elastis sesuai kebutuhan. *Resource* komputasi yang didapatkan dapat disesuaikan dengan dana yang ada. Semakin berkembangnya teknologi *cloud* serta semakin banyaknya kebutuhan akan layanan *cloud* maka semakin banyak pula dibangun *data center* penyedia layanan *cloud* [1].

Perkembangan *Cloud Computing* telah mengakibatkan pembangunan *data center* skala besar di seluruh dunia yang berisi ribuan node. *Data Center Cloud* mengkonsumsi energi listrik yang besar yang tentunya mengakibatkan biaya operasi yang tinggi serta menghasilkan emisi karbon dioksida (CO2) ke lingkungan. Konsumsi energi *data center* di seluruh dunia telah meningkat sebesar 56% dari tahun 2005 sampai 2010, dan pada tahun 2010 dicatat menjadi antara 1,1% dan 1,5% dari penggunaan listrik global [2]. Selain itu, emisi karbon dioksida dari industri ICT saat ini diperkirakan 2% dari emisi global, yang setara dengan emisi dari industri penerbangan [3] dan secara signifikan berkontribusi pada efek rumah kaca. Seperti yang diproyeksikan oleh Koomey, konsumsi energi di *data center* akan terus tumbuh pesat kecuali dengan mengembangkan dan menerapkan manajemen resource yang hemat energi.

Untuk mengatasi tingginya penggunaan energi, maka perlu untuk menghilangkan inefisiensi pengiriman listrik ke resource komputasi. Hal ini dapat dilakukan dengan meningkatkan infrastruktur fisik *data center*, alokasi resource, serta pengembangan algoritma manajemen. Kemajuan terbaru dalam desain *data center* telah menghasilkan peningkatan yang signifikan dari efisiensi infrastruktur. Seperti dilansir proyek Open Compute, Facebook Oregon *data center* mencapai *Power Usage Effectiveness* (PUE) dari 1,08, yang berarti bahwa sekitar 91% dari konsumsi energi data center dikonsumsi oleh *resource* komputasi [4]. Sumber pemborosan energi terletak pada penggunaan resource komputasi yang tidak efisien. Data yang dikumpulkan lebih dari 5000 server yang beroperasi selama 6 bulan, menunjukkan bahwa pemanfaatan resource jarang mendekati 100 %. Server lebih sering beroperasi menggunakan resource 10-50% dari total kapasitas resource yang tersedia [5]. Oleh karena itu, membiarkan server yang pemanfaatannya rendah sangat tidak efisien dari perspektif penggunaan energi.

Distributed dynamic VM consolidation bisa menjadi strategi efektif untuk mengatasi masalah pemborosan energi pada *data center cloud*. Pada penelitian dijelaskan prosedur dari strategi ini dapat diuraikan ke dalam empat tugas pengambilan keputusan, yaitu: (1) *Host overloading detection*, memutuskan kapan *host* harus dianggap sebagai kelebihan beban. Dalam situasi ini, satu atau lebih VMs harus bermigrasi dari *host* ini. (2) *Host underloading detection*, memutuskan kapan *host* harus dianggap *underloaded* (kekurangan beban). Dalam situasi ini, *host* siap untuk beralih ke modus tidur dan semua VMs harus bermigrasi dari *host* ini. (3) *VM selection*, memutuskan VMs mana yang harus pindah dari host yang kelebihan beban dan (4) *VM placement*, memutuskan tentang *host* mana yang harus dipilih untuk menerima migrasi VM [6].

Berdasarkan latar belakang yang telah dipaparkan diatas serta penelitian-penelitian yang telah dilakukan sebelumnya maka penulis termotifasi untuk melakukan penelitian tentang efisiensi energi pada *cloud computing* menggunakan Fuzzy Mamdani pada *host overloading detection* untuk mengoptimalkan penggunaan energi. Penulis tertarik untuk menggunakan Fuzzy Mamdani agar konsumsi energi dapat lebih efisien dibanding metode sebelumnya.

Dari beberapa penelitian yang terkait yang berhubungan dengan efisiensi penggunaan energi pada *data center cloud*, antara lain sebagai berikut:

Pada Penelitian [6] yang dilakukan oleh Anton Beloglazov dan Rajkumar Buyya, menjelaskan bahwa *Dynamic consolidation* pada *virtual machines* (VMs) adalah cara yang efektif untuk meningkatkan pemanfaatan *resource* dan efisiensi energi di *data center cloud*. Penentuan kapankah *host* dianggap *overload* untuk mengalokasikan pemindahan VMs dari host tersebut merupakan aspek *Dynamic VM consolidation* yang secara langsung mempengaruhi pemanfaatan *resource* dan kualitas layanan (QoS) yang disampaikan oleh sistem. Diusulkan heuristik adaptif untuk *Dynamic VM consolidation* berdasarkan analisis data historis dari penggunaan *resource* oleh VMs. Algoritma yang diusulkan secara signifikan mengurangi konsumsi energi , serta tetap

terjaganya *Service Level Agreements* (SLA). Dalam penelitiannya juga menyediakan Model simulasi berbasis Java yang mampu menghitung energi yang efisien dan proses evaluasi untuk menunjukkan kinerja algoritma.

Selanjutnya pada paper [1] Oleh Anton Beloglazov dan Rajkumar Buyya lagi, Solusi yang ada sebelumnya untuk masalah *host overloading detection* umumnya bergantung pada analisis statistik data historis seperti dijelaskan pada paper sebelumnya[6]. Keterbatasan pendekatan ini adalah menyebabkan hasil yang kurang optimal dan tidak diijinkan spesifikasi eksplisit dari tujuan QoS. Diusulkan pendekatan baru bahwa untuk memecahkan masalah *host overloading detection* dengan memaksimalkan rata-rata waktu *intermigration* di bawah QoS yang ditentukan berdasarkan model rantai Markov.

Masoumzadeh SS dan Hlavacs Helmut [7], membahas tentang efisiensi energi pada *data center cloud* dengan menggunakan Fuzzy Q-Learning sebagai *intelligent decission making* pada *host overloading* detection untuk mengoptimalkan penggunaan energi dan performa. Dari hasil percobaan penelitian ini menunjukkan keunggulan FQL sebagai *intelligent decission making* dibandingkan dengan menggunakan Median Absolute Deviation (MAD) dan Inter Quartil Range (IQR).

Pengelola sistem *cloud* perlu menggunakan strategi untuk mengalokasikan sumberdaya ke VM (*Virtual Machine*) dengan *live migration* VM yang ada di dalam server-server fisik secara dinamis. Itulah yang dimaksud strategi *Distributed dynamic VM consolidation*. Pada penelitian sebelumnya dijelaskan prosedur dari strategi ini dapat diuraikan ke dalam empat tugas pengambilan keputusan [1]: (1) *Host overloading detection*, disini memutuskan kapan host harus dianggap sebagai kelebihan beban. Dalam situasi ini, satu atau lebih VMs harus bermigrasi dari *host* ini; (2) *Host underloading detection*, disini memutuskan kapan *host* harus dianggap *underloaded* (kekurangan beban). Dalam situasi ini, *host* siap untuk beralih ke modus tidur dan semua VMs harus bermigrasi dari *host* ini; (3) *VM selection*, disini memutuskan VMs mana yang harus pindah dari *host* yang kelebihan beban; (4) *VM placement*, disini memutuskan tentang *host* mana yang harus dipilih untuk menerima migrasi VM.

Penelitian yang akan dilakukan fokus pada prosedur *Host Overloading Detection*. *Host overloading detection* merupakan proses untuk memutuskan kapan *host/server* harus dianggap sebagai kelebihan beban. Parameter yang dapat digunakan untuk mengukur apakah host overload atau tidak, dapat dilihat dari *CPU utilization , Memory utilization, Storage utilization*, maupun *Bandwidth utilization* pada *host/server* tersebut. Jika *host* terdeteksi overload, maka dalam situasi ini, satu atau lebih VMs harus bermigrasi dari *host* ini[1].

Metode Mamdani sering juga dikenal dengan nama Metode Max-Min[8]. Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Untuk mendapatkan *output*, diperlukan 4 tahapan yaitu : tahapan pertama pembentukan himpunan fuzzy, tahapan kedua aplikasi fungsi implikasi, tahapan ketiga komposisi aturan, dan tahapan yang terakhir adalah Penegasan (*defuzzy*)[9].

Untuk membuat sebuah simulasi yang dapat dievaluasi, sangat penting untuk melakukan eksperimen dengan menggunakan *workload* atau beban kerja yang diambil dari *real* sistem. Data yang digunakan untuk percobaan adalah data insfrastruktur monitoring dari PlanetLab. *Dataset* PlanetLab berisi data penggunaan CPU *Virtual Machine* dari server-server yang berlokasi lebih dari 500 *data center* di seluruh dunia. *Workload* dari PlanetLab dapat mewakili lingkungan *IaaS Cloud*, seperti *Amazon EC2*, artinya bahwa VMs dibuat dan dikelola oleh banyak pengguna independen, dan penyedia infrastruktur tidak mengetahui aplikasi apa saja yang berjalan di VM[10].

CloudSim adalah framework untuk pemodelan dan simulasi infrastruktur *cloud computing*. Dengan menggunakan simulator akan mempermudah penelitian karena sangat tidak mungkin untuk membangun sendiri sebuah lingkungan *IaaS Cloud* secara nyata. CloudSim mendukung pemodelan dan simulasi *data center cloud* skala besar, *host server virtual* dengan *resource* yang bisa di kustomisasi, *resurce* energi komputasi, topologi jaringan pada data center, penyisipan dinamis elemen simulasi, berhenti dan melanjutkan simulasi, serta kebijakan yang ditetapkan pengguna untuk alokasi *host* mesin virtual dan kebijakan untuk alokasi *resource host*

ke mesin virtual. CloudSim merupakan simulator Cloud berbasis pemrograman JAVA yang telah dikembangkan oleh Cloud Computing dan Distributed Sistem Laboratorium Melbourne University[11].

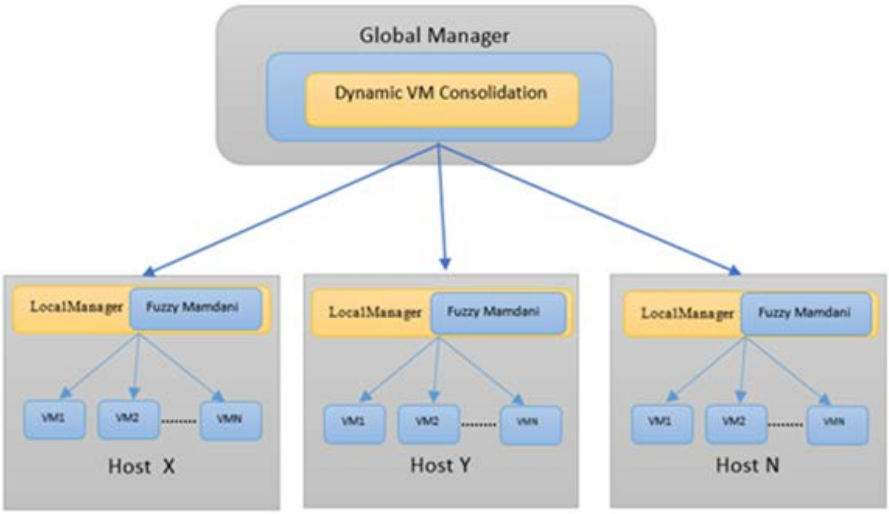
2. METODE PENELITIAN

2.1. Data

Penelitian ini memakai data yang berasal dari PlanetLab. Dataset berupa workload dari server-server pada data center Amazone Enterprise Cloud 2 yang sudah dimonitor oleh PlanetLab. Data yang tersedia merupakan data workload perhari pada data center. Terdapat 800 Host yang akan disimulasikan dari workload tersebut.

2.2. Usulan Model

Fuzzy Mamdani ditempatkan pada local manager masing-masing Host Virtual Machine. Fuzzy Mamdani ini bertugas mendeteksi apakah host tersebut kelebihan beban atau tidak. Seperti yang ditunjukkan pada Gambar 1, hasil dari Fuzzy local manager akan didistribusikan ke global manager dimana global manager ini merupakan pusat dari pengelolaan host fisik pada Data Center Cloud.



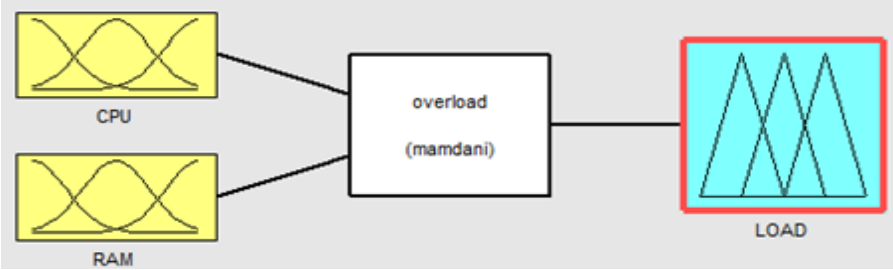
Gambar 1. Usulan Model

2.3. Fuzzy Mamdani Pada Host Overload Detection

2.3.1. Fuzzifikasi

Pada proses fuzzifikasi langkah yang pertama adalah menentukan variabel fuzzy dan himpunan fuzzinya. Kemudian tentukan derajat kesepadanan (degree of match) antara data masukan fuzzy dengan himpunan fuzzy yang telah didefenisikan untuk setiap variabel masukan sistem dari setiap aturan fuzzy. Pada metode mamdani, baik variabel input maupun variabel output dibagi menjadi satu atau lebih himpunan fuzzy.

Pada Gambar 2, dapat dilihat bahwa input pada fuzzy adalah Resource RequestCPU dan Resource RequestRAM, sedangkan output pada fuzzy adalah Load Server.



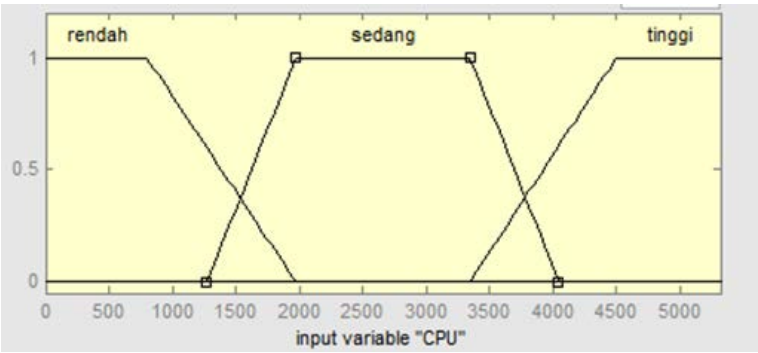
Gambar 2. Fuzzifikasi

1. Indikator input Nilai requestCPU
   Rendah = 0 – 1973
   Sedang = 1273 – 4046
   Tinggi = 3346 – 5320

2. Indikator input Nilai requestRAM
   Rendah = 0 – 1565
   Sedang = 1065 – 3030
   Tinggi = 2530 – 4096

3. Indikator Output Nilai LOAD SERVER
   Normal = 0 – 70
   Overload = 70 – 100

RequestCPU terdiri dari 3 bagian (0-1973, 1273-4046, 3346-5320), seperti terlihat pada Gambar 3.



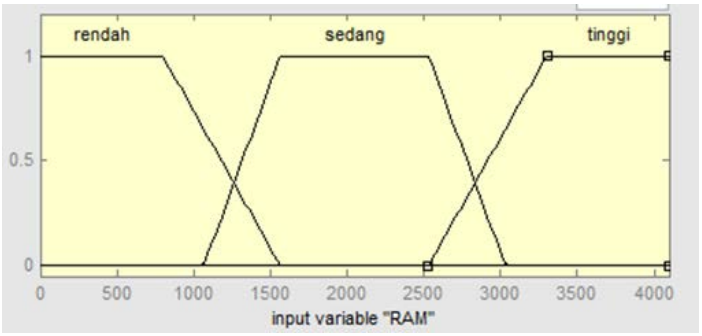
Gambar 3. Variabel Input RequestCPU

$$\mu\,\mathrm{CPURendah}\,(x) = \begin{cases} 1 & x \le 800 \\ \dfrac{1973 - x}{1973 - 800} & 800 \le x \le 1973 \\ 0 & x \ge 1973 \end{cases} \quad (1)$$

$$\mu\,\mathrm{CPUSedang}\,(x) = \begin{cases} 0 & x < 1273 \ OR \ x > 4046 \\ \dfrac{x - 1273}{1973 - 1273} & 1273 \le x \le 1973 \\ 1 & 1973 \le x \le 3346 \\ \dfrac{4046 - x}{4046 - 3346} & 3346 \le x \le 4046 \end{cases} \quad (2)$$

$$\mu\,\text{CPUTinggi}\,(x) = \begin{cases} 0 & x \le 3346 \\ \dfrac{x - 3346}{4500 - 3346} & 3346 \le x \le 4500 \\ 1 & x \ge 4500 \end{cases} \quad (3)$$

RequestRAM terdiri dari 3 bagian (0-1565, 1065-3030, 2530-4096), seperti terlihat pada Gambar 4.



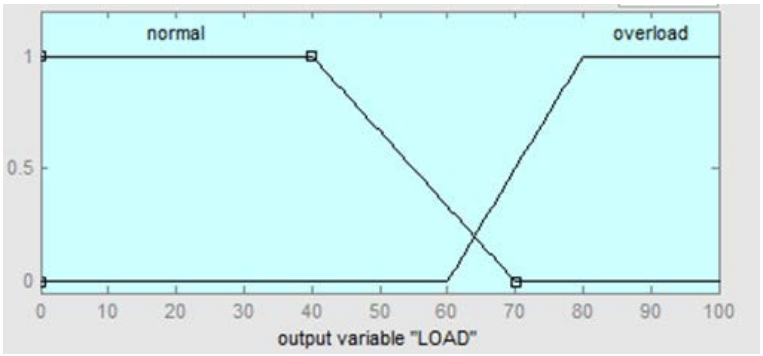
Gambar 4. Variabel Input RequestRAM

$$\mu\,\text{RAMRendah}\,(x) = \begin{cases} 1 & x \le 800 \\ \dfrac{1565 - x}{1565 - 800} & 800 \le x \le 1565 \\ 0 & x \ge 1565 \end{cases} \quad (4)$$

$$\mu\,\text{RAMSedang}\,(x) = \begin{cases} 0 & x \le 1065\ OR\ x \ge 3030 \\ \dfrac{x - 1065}{1565 - 1065} & 1065 \le x \le 1565 \\ 1 & 1565 \le x \le 2530 \\ \dfrac{3030 - x}{3030 - 2530} & x > 2530 \le x \le 3030 \end{cases} \quad (5)$$

$$\mu\,\text{RAMTinggi}\,(x) = \begin{cases} 0 & x \le 2530 \\ \dfrac{x - 2530}{3300 - 2530} & 2530 \le x \le 3300 \\ 1 & x \ge 3300 \end{cases} \quad (6)$$

Nilai Output Keputusan LOAD SERVER terdiri dari 2 bagian (0-70, 60-100), seperti terlihat pada Gambar 5.



Gambar 5. Nilai Output Keputusan Load Server

$$\mu \text{LOADNormal}(x) = \begin{cases} 1 & x \leq 40 \\ \frac{70-x}{70-40} & 40 \leq x \leq 70 \\ 0 & x \geq 70 \end{cases} \quad (7)$$

$$\mu \text{LOADOverload}(x) = \begin{cases} 0 & x \leq 60 \\ \frac{x-60}{80-60} & 60 \leq x \leq 80 \\ 1 & x \geq 80 \end{cases} \quad (8)$$

2.3.2. Fuzzy Rule

[RI] Jika CPU Rendah maka LOAD Normal
[R2] Jika CPU Sedang dan RAM Rendah maka LOAD Normal
[R3] Jika CPU Sedang dan RAM Sedang maka LOAD Normal
[R4] Jika CPU Sedang dan RAM Tinggi maka LOAD Overload
[R5] Jika CPU Tinggi dan RAM Rendah maka LOAD Normal
[R6] Jika CPU Tinggi dan RAM Sedang maka LOAD Overload
[R7] Jika CPU Tinggi dan RAM Tinggi maka LOAD Overload

2.3.3. Mesin Inferensi

Pada mesin inferensi, fungsi implikasi yang digunakan adalah fungsi implikasi MIN, sedangkan komposisi aturan yang digunakan menggunakan metode MAX.

2.3.4. Defuzzifikasi

Defuzzifikasi menggunakan metode centroid, nilai *crisp z* dihitung dengan membagi wilayah menjadi 2 bagian (W1 dan W2) yang luasnya masing-masing A1, dan A2. Momen terhadap nilai keanggotaan masing-masing adalah M1 dan M2.

2.4. Pengujian

Pengujian dilakukan dengan menggunakan CloudSim yang berbasis pemrograman java. CloudSim dimodifikasi dengan menambahkan model yang diusulkan. Tahapan eksperimen pada penelitian ini yaitu:

1. Menambahkan metode Fuzzy Logic Mamdani pada proses Host Overload Detection di vmAllocationPolicy.
2. Pengujian dilakukan dengan cara mengkombinasikan metode Host Overloading Detection dengan metode-metode selection yang sudah ada pada vmSelectionPolicy.
3. Apabila pengujian Host Overload Detection yang dikombinasikan dengan VM Selection diatas telah selesai, selanjutnya beberapa workload yang lain diuji secara bergantian.
4. Hasil dari pengujian dari tiap workload kemudian dibandingkan.

2.4.1. Konfigurasi CloudSim

Konfigurasi simulasi cloudsim terdiri dari 800 host dengan seri server HP ProLiant ML 110 G5 ($Intel\,Xeon$ 3075, 2 $cores$ × 2660 MHz, 4 GB) yang memiliki frekuensi 2660 MIPS pada masing-masing core-nya. Setiap server juga memiliki bandwidth jaringan 1GB / s. Karakteristik model server ini sama dengan layanan cloud computing Amazon Enterprise Cloud 2.

2.5. Evaluasi

Proses evaluasi data dilakukan dengan mengamati hasil dari pengujian metode yang diusulkan dengan metode host overloading detection yang sudah ada. Parameter yang dipakai untuk evaluasi dalam penelitian ini adalah Energy Consumption (EC), SLA Violation (SLAV),

Performance Degradation Due Migration (PDM), VM Migration, dan SLA Violation Time per Active Host (SLATAH). EC merupakan energi yang diperlukan dalam data center cloud.

$$E = \int_t P(u(t))dt \quad (9)$$

Untuk mengukur efisiensi dari metode yang diusulkan, kita harus mengevaluasi *power* yang dikonsumsi beban kerja data menggunakan rumus 9 . Dimana *u* adalah *CPU Utilization*, *t* adalah waktu, dan *u(t)* adalah pemanfaatan CPU yang berubah dari waktu karena beban kerja variabel.

SLA Violation Time per Active Host, setelah mengukur konsumsi daya, kita diminta untuk mengukur persentase waktu, selama host aktif menggunakan CPU sampai 100%

$$SLATAH = \frac{1}{N}\sum_{i=1}^{N}\frac{T_{si}}{T_{ai}} \quad (10)$$

Dimana N, adalah jumlah Host, Tsi total wantu sebuah Host *i* sampai menggunakan 100 % sumberdaya yang mengarah ke pelanggaran SLA. Tai adalah total Host *i* yang dalam keadaan aktif melayani VM.

PDM, Penurunan performa akibat migrasi. Dihitung dengan rumus

$$SLATAH = \frac{1}{M}\sum_{j=1}^{M}\frac{C_{dj}}{C_{rj}} \quad (11)$$

Dimana M adalah jumlah VM, Cdj adalah estimasi penurunan performa yang disebabkan oleh VM j, Crj adalah total kapasitas CPU yang diminta oleh VM j.

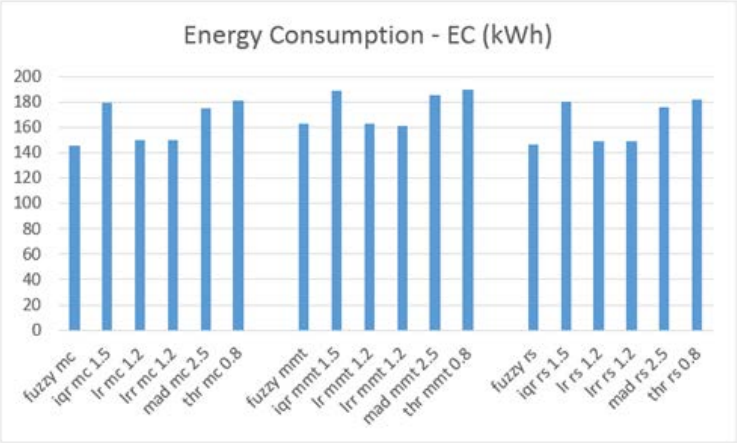
SLA Violation , pelanggaran SLA dihitung dengan rumus

$$SLATAH = SLATAH \cdot PDM \quad (12)$$

Digunakan untuk menunjukkan penuruan kinerja secara keseluruhan yang diakibatkan oleh host overload dan VM migration.

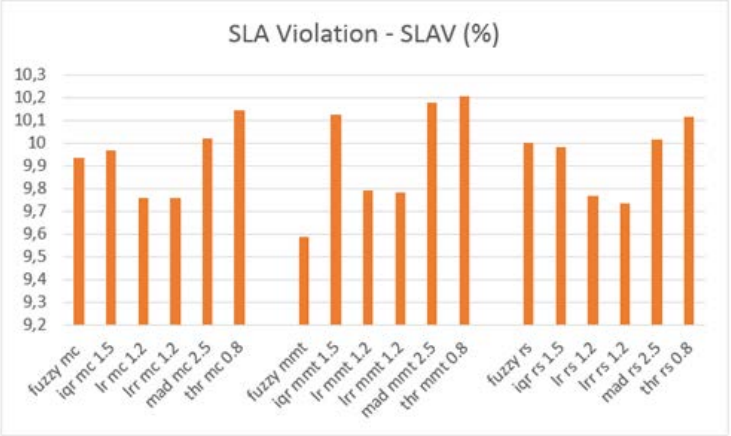
3. HASIL DAN PEMBAHASAN

Dari eksperimen yang telah dilakukan pada semua workload, konsumsi energi / Energy Consumption (EC) yang menggunakan metode Fuzzy Mamdani Sebagai Host Overloading Detection, menjadi lebih rendah dibanding metode-metode sebelumnya. Apapun VM Selection yang digunakan, selama Host Overload Detection-nya menggunakan Fuzzy Mamdani, konsumsi energi menjadi lebih rendah, artinya tujuan untuk membuat Data Center Cloud Computing yang efisien tercapai. Jika dirata-rata, konsumsi energi paling efisien terjadi ketika Host Overload Detection menggunakan Fuzzy Mamdani dikombinasikan dengan VM Selection Maximum Correlation (MC) yakni 145,818 kWh. Metode sebelumnya EC paling rendah adalah 149,16 kWh. Jadi penggunaan Fuzzy Mamdani mampu menurunkan penggunaan energi sebesar 3,342 kWh. Perbandingan rata-rata EC bisa dilihat pada Gambar 6.



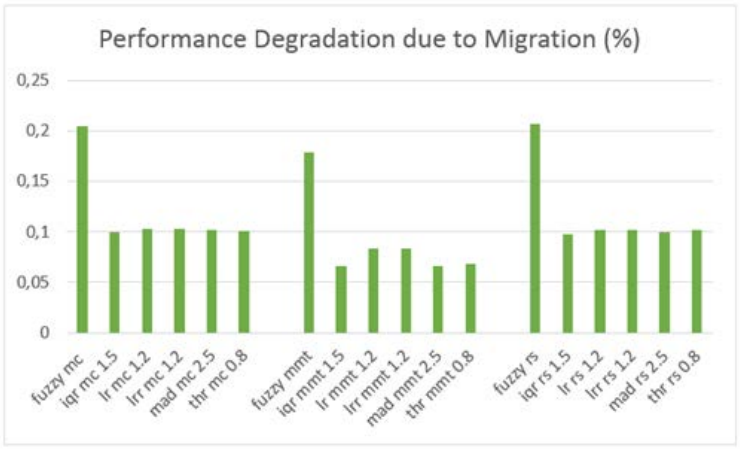
Gambar 6. Perbandingan rata-rata EC

SLAV paling rendah terjadi pada saat *Host Overload Detection* menggunakan *fuzzy Mamdani* dikombinasikan dengan *VM Selection Minimum Migration Time* (MMT) dengan nilai SLAV 9,587 % . Metode sebelumnya SLAV paling rendah adalah 9,735 %. Jadi penggunaan Fuzzy Mamdani menurunkan SLAV sebesar 0,148 %. Perbandingan rata-rata SLAV bisa dilihat pada Gambar 7.



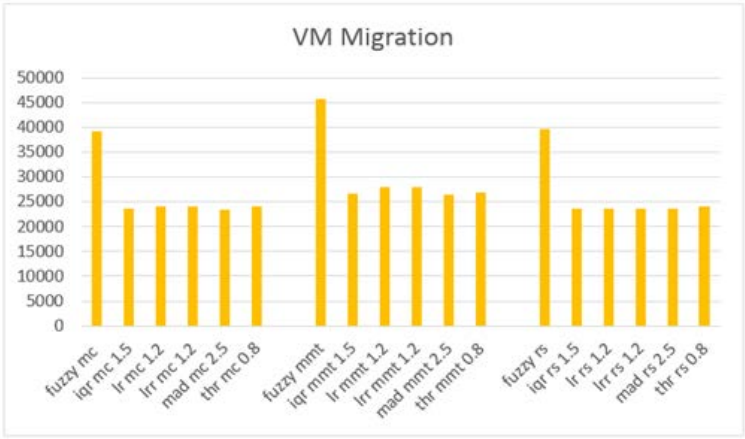
Gamabar 7. Perbandingan rata-rata SLAV

Nilai prosentase PDM saat menggunakan *Fuzzy Mamdani* pada *Host Overload Detection* lebih tinggi jika dibandingkan dengan metode-metode *Host Overload detection* yang sudah ada sebelumnya. Saat menggunakan *Fuzzy Mamdani* pada *Host Overload Detection* dikombinasikan *VM Selection MMT*, nilainya 0,179%. Nilai ini lebih tinggi dibanding dengan metode sebelumnya, yaitu 0,066%. Jadi penggunaan *Fuzzy Mamdani* justru menaikkan PDM sebesar 0,113%. Perbandingan rata-rata PDM bisa dilihat pada Gambar 8.



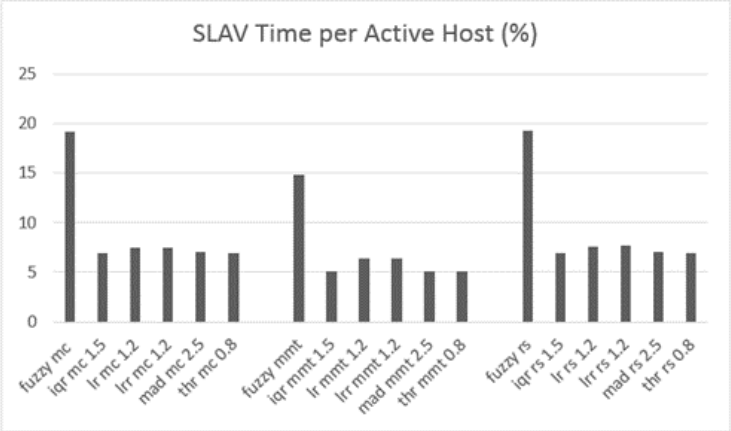
Gambar 8. Perbandingan rata-rata PDM

Nilai *VM Migration* saat menggunakan *Fuzzy Mamdani* pada *Host Overload Detection* lebih tinggi jika dibandingkan dengan metode-metode *Host Overload detection* yang sudah ada sebelumnya. Saat menggunakan *Fuzzy Mamdani* pada *Host Overload Detection* dikombinasikan *VM Selection MC*, nilainya 39234,3. Nilai ini lebih tinggi dibanding dengan metode sebelumnya, yaitu 23554,6. Jadi penggunaan *Fuzzy Mamdani* justru menaikkan VM Migration sebesar 15679,7. Perbandingan rata-rata PDM bisa dilihat pada Gambar 9.



Gambar 9. Perbandingan rata-rata VM Migration

Nilai prosentase SLATAH saat menggunakan *Fuzzy Mamdani* pada *Host Overload Detection* lebih tinggi jika dibandingkan dengan metode-metode *Host Overload detection* yang sudah ada sebelumnya. Saat menggunakan *Fuzzy Mamdani* pada *Host Overload Detection* dikombinasikan *VM Selection MMT*, nilainya 14,84 %. Nilai ini lebih tinggi dibanding dengan metode sebelumnya, yaitu 5,056 %. Jadi penggunaan *Fuzzy Mamdani* justru menaikkan SLATAH sebesar 9,784 %. Perbandingan rata-rata PDM bisa dilihat pada Gambar 10.



Gambar 10. Perbandingan SLATAH

4. KESIMPULAN

Konsumsi energi pada data center cloud computing berhasil diminimalkan dengan menerapkan metode Fuzzy Mamdani dalam proses Host Overloading Detection. Konsumsi energi paling efisien terjadi ketika Host Overload Detection menggunakan Fuzzy Mamdani dikombinasikan dengan VM Selection Maximum Correlation (MC) yakni 145,818 kWh. Metode sebelumnya konsumsi energi paling rendah adalah 149,16 kWh. Penggunaan Fuzzy Mamdani lebih efisien 3,342 kWh. Jadi, hasil pengukuran dengan metode yang diusulkan menjanjikan efisiensi energi sampai dengan 2,24%.

Fuzzy mamdani saat diterapkan menghasilkan SLA yang rendah, hal ini terlihat dari nilai performa PDM, VM Migration, dan SLATAH yang nilainya lebih tinggi dibandingkan dengan metode-metode sebelumnya. SLA ini turun disebabkan karena VM Migration yang terlalu tinggi, dimana semakin tinggi VM Migration maka semakin tinggi pula nilai Cost of VM Migration.

5. SARAN

Pada Penelitian selanjutnya disarankan untuk menambahkan variabel Bandwidth pada inputan fuzzy mamdani dalam proses Host Overloading Detection sehingga diharapkan bisa mendapatkan Efisiensi energi dan SLA yang lebih baik, serta mengurangi Cost VM Migration untuk meningkatkan SLA.

DAFTAR PUSTAKA

[1] Beloglazov, A., Buyya, R., 2013, Managing Overloaded Hosts for Dynamic Consolidation of Virtual Machines in Cloud Data Centers Under Quality of Service Constraints, *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, Vol 24, No 7, Pages 1366-1379.

[2] Koomey, J., 2011, *Growth in data center electricity use 2005 to 2010*, Analytics Press, Oakland.

[3] Gartner, Inc., 2007. *Gartner Estimates ICT Industry Accounts for 2 Percent of Global CO2 Emissions*. Gartner Press Release (April)

[4] Open Compute Project, *Energy efficiency*, http://opencompute.org/about/energy-efficiency/ diakses pada 5 April 2014.

[5] Barroso, L. A., Holzle, U., 2007. The Case for Energy-Proportional Computing, *Computer Society*, vol. 40, no. 12, Hal 33–37.

[6] Beloglazov, A., Buyya, R., 2012. Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers, *Concurrency and Computation: Practice and Experience*, vol. 24, no. 13, hal 1397-1420.

[7] Masoumzadeh, S.S., Hlavacs, H., 2013, Intelligent Decision-Making in Distributed Dynamic VM Consolidation Using Fuzzy Q-Learning, *31st International Symposium on Computer Performance, Modeling, Measurements and Evaluation 2013*, Poster 4.

[8] Ginting, R. Br., 2014, Analisis Fungsi Implikasi Max-Min dan Max-Prod Dalam Pengambilan Keputusan, *Creative Information Technology Journal*, Vol 1, No 2, hal 128-138.

[9] Sandhopi, S. N., Astuti, E. Z., 2015, Optimasi Fungsi Keanggotaan Fuzzy Menggunakan Metode Mamdani Terhadap Prediksi Perilaku Pembeli, *Techno.COM*, Vol 14, No 4

[10] Park, K. S., Pai, V. S., 2006, CoMon: a mostly-scalable monitoring system for PlanetLab, *ACM SIGOPS Operating Systems Review*, vol. 40, no. 1, hal 65–74.

[11] Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A. F., Buyya, R., 2011, Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, *Software: Practice and Experience*, vol. 41, no. 1, hal 23-50.