

Penerapan Fuzzy Sugeno untuk Deteksi Overload Host pada Dynamic VM Consolidation

M Naufal Adrian Pratama Putra ¹, Chaerul Umam ²

^{1,2} Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro

111202214188@mhs.dinus.ac.id, chaerul@dsn.dinus.ac.id

Info Artikel

Riwayat Artikel:

Received 2026-01-08

Revised 2026-06-13

Accepted 2026-06-20

Abstract – High energy consumption in cloud data centers has a direct impact on operational costs and environmental efficiency, so a more adaptive and energy-efficient resource management strategy is needed through Dynamic Virtual Machine (VM) consolidation. This research proposes a novelty in the form of applying the Zero Order Fuzzy Sugeno method as a more responsive and lightweight host overload detection mechanism. The proposed method was tested through simulations using CloudSim version 7 with PlanetLab workloads. Performance evaluation was conducted through simulations using the CloudSim 7G framework with a standard PlanetLab benchmark workload. Tests were executed in 12 cross-combination scenarios to compare the performance of Fuzzy Sugeno with the system's built-in benchmarking methods, such as Inter Quartile Range (IQR) and Median Absolute Deviation (MAD). The test results show that the Fuzzy Sugeno method is able to provide significant energy consumption savings of up to 27.5%, with the lowest energy consumption achievement of 144.07 kWh, much more efficient than the IQR-MU comparison method which reached 198.94 kWh. Although there is a trade-off in the form of an increase in the frequency of VM migration which results in a violation of the Service Level Agreement (SLA) of 0.70%, the Fuzzy Sugeno method has proven to be very effective and is recommended for green data center scenarios that prioritize maximum power efficiency.

Keywords: Cloud Computing; CloudSim; Dynamic VM Consolidation; Energy Efficiency; Fuzzy Sugeno; Host Overload Detection.

Corresponding Author:

M Naufal Adrian Pratama Putra

Email:

111202214188@mhs.dinus.ac.id



This is an open access article under the [CC BY 4.0](https://creativecommons.org/licenses/by/4.0/) license.

Abstrak – Konsumsi energi yang tinggi pada data center cloud berdampak langsung pada biaya operasional dan efisiensi lingkungan, sehingga diperlukan strategi manajemen sumber daya yang lebih adaptif dan hemat energi melalui Dynamic Virtual Machine (VM) consolidation. Penelitian ini mengusulkan kebaruan berupa penerapan metode Fuzzy Sugeno Orde Nol sebagai mekanisme host overload detection yang lebih responsif dan ringan. Evaluasi kinerja dilakukan melalui simulasi menggunakan framework CloudSim 7G dengan workload benchmark standar dari PlanetLab. Pengujian dieksekusi dalam 12 skenario kombinasi silang untuk membandingkan kinerja Fuzzy Sugeno dengan metode pembandingan bawaan sistem seperti Inter Quartile Range (IQR) dan Median Absolute Deviation (MAD). Hasil pengujian menunjukkan bahwa metode Fuzzy Sugeno mampu memberikan penghematan konsumsi energi secara signifikan hingga 27,5%, dengan pencapaian konsumsi energi terendah sebesar 144,07 kWh, jauh lebih efisien dibandingkan dengan metode pembandingan IQR-MU yang mencapai 198,94 kWh. Meskipun terdapat trade-off berupa peningkatan frekuensi migrasi VM yang berdampak pada pelanggaran Service Level Agreement (SLA) sebesar 0,70%, metode Fuzzy Sugeno terbukti sangat efektif dan direkomendasikan untuk skenario green data center yang memprioritaskan efisiensi daya secara maksimal.

Kata Kunci: Cloud Computing, CloudSim, Dynamic VM Consolidation, Efisiensi Energi, Fuzzy Sugeno, Host Overload Detection

I. PENDAHULUAN

Cloud computing telah menjadi infrastruktur utama dalam penyediaan layanan digital berskala besar. Salah satu kunci untuk mencapai efisiensi operasional dalam pusat data cloud adalah melalui proses *Virtual Machine (VM) Consolidation*, yaitu strategi untuk mengoptimalkan penggunaan sumber daya dan efisiensi energi dengan memindahkan VM secara dinamis. Salah satu kunci untuk mencapai efisiensi operasional dalam pusat data cloud adalah melalui proses *Virtual Machine (VM) Consolidation*, yaitu strategi untuk mengoptimalkan penggunaan sumber daya dan efisiensi energi dengan memindahkan VM secara dinamis, sebuah area yang terus dikembangkan dalam penelitian terkini[1]. Dynamic VM consolidation menggunakan migrasi langsung (live migration) dan mengubah node yang tidak aktif ke mode tidur (sleep mode) memungkinkan penyedia cloud untuk mengoptimalkan penggunaan sumber daya dan mengurangi konsumsi energi[2]. Dalam konteks ini, host overload detection menjadi aspek krusial agar perpindahan VM tidak menyebabkan degradasi layanan[3]. Upaya untuk merancang algoritma yang hemat energi dalam konsolidasi VM merupakan area penelitian yang sangat aktif dan penting, dengan berbagai perbandingan dan evaluasi metode yang terus dilakukan untuk menemukan solusi optimal[4].

Konsumsi energi pada pusat data *cloud* merupakan isu krusial yang terus meningkat seiring dengan pertumbuhan pesat layanan digital. Studi menunjukkan bahwa konsumsi energi *global data center* diproyeksikan meningkat dari sekitar 200 TWh pada tahun 2016 menjadi hampir 2967 TWh pada tahun 2030[5]. Peningkatan ini berdampak langsung pada biaya operasional yang tinggi serta emisi karbon yang signifikan, sehingga efisiensi energi menjadi fokus utama dalam pengelolaan infrastruktur *cloud*. Selain itu, tingkat utilisasi rata-rata *server* di pusat data umumnya relatif rendah, yaitu sekitar 30%, yang menyebabkan banyak *host* berada dalam kondisi *idle* namun tetap mengonsumsi daya[5]. Dengan mematikan atau mengalihkan *host* yang tidak aktif ke mode tidur, penyedia layanan *cloud* dapat mengurangi konsumsi energi secara signifikan[6]. Oleh karena itu, penerapan strategi *VM consolidation* yang efektif menjadi salah satu pendekatan penting untuk meningkatkan efisiensi energi sekaligus mendukung inisiatif *green data center* dengan menekan biaya operasional dan dampak lingkungan[7].

Proses *VM consolidation* ini secara umum terdiri dari empat tahap utama yang saling terkait. Tahap pertama adalah *host overload detection*, di mana sistem secara terus-menerus memantau penggunaan sumber daya untuk menentukan kapan sebuah *host* dianggap mengalami kelebihan beban sehingga satu atau lebih VM harus dimigrasikan. Tahap kedua adalah *host underload detection*, di mana sistem mengidentifikasi *host* dengan beban rendah yang siap dialihkan ke modus tidur. Setelah *host* yang perlu ditindaklanjuti terdeteksi, tahap ketiga adalah *VM selection*, yaitu menentukan VM mana dari *host* tersebut yang akan dimigrasikan. Tahap keempat adalah *VM placement*, yaitu proses menemukan *host* tujuan yang paling sesuai untuk dimigrasikan[8]. Penelitian ini berfokus pada penerapan pendekatan Fuzzy Sugeno untuk mendeteksi kondisi *overload* pada *host* dalam lingkungan *dynamic VM consolidation*.

Berbagai metode telah digunakan pada proses *VM consolidation*, seperti *Exponential Weighted Moving Average (EWMA)* [9], *Multi-Dimensional Regression Host Utilization (MDRHU)* [10], dan pendekatan *heuristic adaptif* [2]. Selain itu, pendekatan berbasis logika *fuzzy* juga telah mulai banyak diterapkan dalam proses *VM consolidation*, salah satunya dengan kombinasi *fuzzy* dan *threshold* [11],[12]. Berbagai metode ini seringkali diimplementasikan dan diuji menggunakan platform simulasi seperti CloudSim [13],[14].

Metode statistik seperti EWMA memiliki keunggulan dalam kesederhanaan dan kecepatan komputasi, namun rentan menghasilkan prediksi yang buruk ketika terjadi perubahan data secara tiba-tiba (*sudden change*) pada beban kerja yang fluktuatif[9]. Metode MDRHU yang berbasis regresi multi-dimensi menawarkan akurasi lebih tinggi karena mempertimbangkan banyak parameter sekaligus, seperti CPU, memori, dan bandwidth jaringan[10]. Akan tetapi, kalkulasi multi-dimensi ini membuat MDRHU memiliki waktu eksekusi (*execution time*) yang jauh lebih tinggi dan kompleksitas komputasi yang berat sehingga berpotensi menambah *overhead* komputasi pada pengelola *data center*[10]. Sementara itu, metode *heuristic adaptif* lebih fleksibel terhadap perubahan beban, namun tidak cukup presisi jika tidak disesuaikan dengan karakteristik *workload* tertentu [2]. Metode berbasis *fuzzy* memiliki kelebihan dalam menangani ketidakpastian dan data kabur, tetapi sebagian masih terbatas pada pendekatan *rule based* sederhana [15],[16].

Pendekatan berbasis logika *fuzzy* telah digunakan dalam berbagai penelitian sebagai solusi untuk deteksi *host overload* karena kemampuannya menangani ketidakpastian dan variasi beban kerja [12],[15],[16]. Pada penelitian sebelumnya, metode Fuzzy Mamdani telah diandalkan untuk mendeteksi *host overload*[8]. Meskipun dinilai efektif, pendekatan Fuzzy Mamdani memiliki kekurangan utama pada tahap defuzzifikasinya yang menggunakan metode Centroid[8]. Metode ini membutuhkan perhitungan matematis yang rumit untuk membagi dan mencari luas wilayah (*area*) di bawah kurva fungsi keanggotaan[8], sehingga memakan waktu pemrosesan yang lebih lama dan kurang responsif jika diterapkan pada skala data center masif yang membutuhkan keputusan seketika. Tantangan utama dari metode *fuzzy* terletak pada proses perancangan sistem inferensi yang efektif. Penentuan fungsi keanggotaan dan aturan *fuzzy* yang tepat sangat krusial, karena jika tidak dirancang secara akurat, dapat menurunkan performa deteksi dan menyebabkan hasil yang tidak stabil atau tidak konsisten dalam kondisi beban dinamis. Selain itu, sebagian pendekatan *fuzzy* masih menggunakan parameter dan aturan yang ditentukan secara manual atau statis, sehingga kurang adaptif terhadap perubahan beban kerja secara *real-time*.

Penelitian ini mengusulkan penggunaan Fuzzy Sugeno sebagai pendekatan alternatif yang lebih presisi dan adaptif untuk deteksi *overload host*. Perbedaan utama sekaligus keunggulan Fuzzy Sugeno dibandingkan Fuzzy Mamdani terletak pada model output-nya. Fuzzy Sugeno menggunakan nilai konstanta tegas (*crisp*) berupa *singleton* pada bagian konsekuen (aturan *IF-THEN*)[17]. Dengan demikian, proses defuzzifikasinya cukup diselesaikan melalui perhitungan matematika dasar berupa rata-rata terbobot (*weighted average*)[17][18]. Fuzzy Sugeno mampu menggabungkan kekuatan *fuzzy logic* dalam menangani ketidakpastian dengan model output yang dapat dihitung langsung, sehingga menghasilkan sistem deteksi yang lebih responsif dan efisien [12],[19]. Pendekatan ini juga memungkinkan pengembangan sistem otomatisasi *VM consolidation* yang lebih akurat berdasarkan kondisi sumber daya secara *real time*.

CloudSim merupakan salah satu platform simulasi yang paling banyak digunakan dalam penelitian *cloud computing* untuk mengevaluasi algoritma manajemen sumber daya, termasuk *Dynamic Virtual Machine (VM) Consolidation*. Platform ini menyediakan lingkungan simulasi yang fleksibel dan terkontrol untuk memodelkan infrastruktur *data center*, *host*, *virtual machine*, serta kebijakan penjadwalan dan migrasi VM tanpa memerlukan perangkat keras fisik berskala besar. Dengan dukungan terhadap simulasi konsumsi energi, SLA, dan performa sistem,

CloudSim memungkinkan peneliti untuk melakukan analisis komprehensif terhadap dampak suatu metode deteksi *overload* dan kebijakan *VM consolidation* secara sistematis dan terukur[13][14].

CloudSim mendukung penggunaan *workload* nyata seperti PlanetLab, sehingga hasil simulasi dapat merepresentasikan pola beban kerja yang dinamis dan fluktuatif sebagaimana terjadi pada lingkungan *cloud* nyata. Versi terbaru CloudSim, yaitu CloudSim 7, juga menghadirkan peningkatan pada modul simulasi energi dan skalabilitas sistem, yang menjadikannya semakin relevan untuk penelitian terkait efisiensi energi dan manajemen sumber daya *cloud*. Oleh karena itu, penggunaan CloudSim dalam penelitian ini dinilai tepat untuk mengevaluasi kinerja metode Fuzzy Sugeno secara objektif dan konsisten, serta memungkinkan perbandingan yang adil dengan metode deteksi *overload* lain yang telah ada.

Penelitian ini bertujuan untuk menerapkan dan mengevaluasi kinerja sistem deteksi *host overload* berbasis Fuzzy Sugeno pada scenario *dynamic VM consolidation* di lingkungan *cloud computing*. Sistem ini dirancang untuk meningkatkan akurasi deteksi *overload host* serta mengoptimalkan efisiensi energi. Melalui simulasi menggunakan CloudSim 7, penelitian ini juga akan membandingkan performa pendekatan ini dengan metode deteksi lain yang telah ada.

II. METODE

A. Data

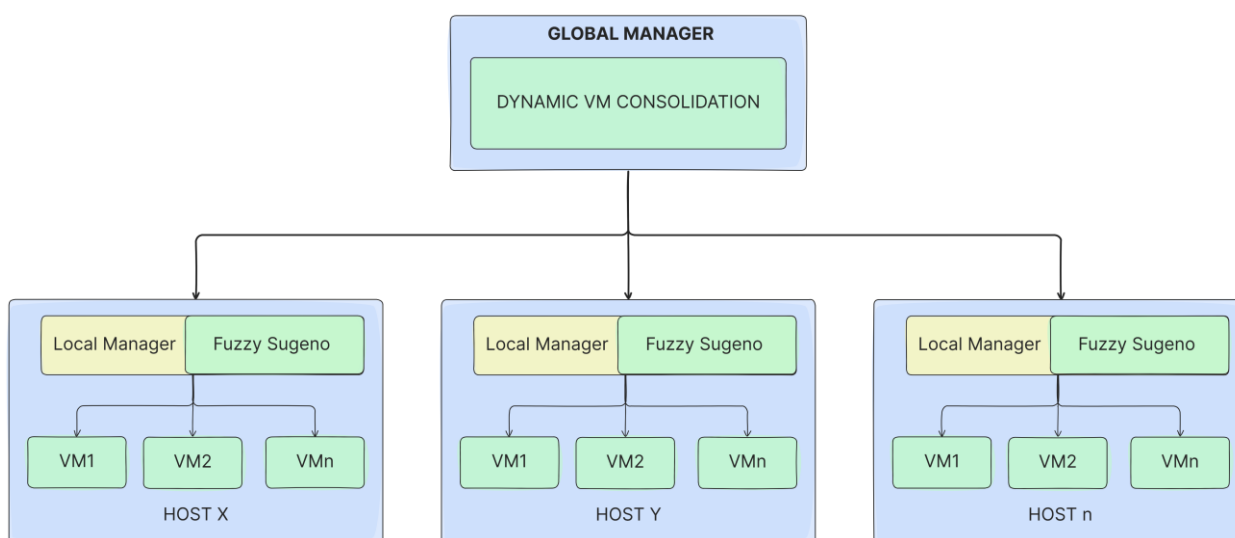
Penelitian ini menggunakan data *workload* dari PlanetLab. Planet lab yang dipakai merupakan *workload* yang telah disediakan pada Cloudsim. Secara spesifik, dataset yang digunakan dalam simulasi ini adalah jejak *workload* PlanetLab yang diambil pada tanggal 3 Maret 2011 (20110303). Meskipun direkam pada tahun 2011, dataset PlanetLab tetap dipertahankan karena secara luas diakui dan digunakan sebagai benchmark standar dalam evaluasi *Dynamic VM Consolidation* pada platform CloudSim. Dataset ini berisi jejak utilisasi CPU historis dari ribuan Virtual Machine (VM) yang didistribusikan pada lebih dari 500 server di seluruh dunia[16].

B. Cloudsim

Seluruh eksperimen dalam penelitian ini dilakukan menggunakan platform simulasi CloudSim. CloudSim adalah sebuah *framework* simulasi berbasis Java yang dirancang khusus untuk pemodelan dan simulasi infrastruktur *cloud computing*. Platform ini memungkinkan untuk melakukan simulasi algoritma manajemen sumber daya seperti *VM consolidation* tanpa memerlukan perangkat keras fisik berskala besar.

Simulasi dilakukan dengan CloudSim versi 7 dan dijalankan di atas lingkungan Java OpenJDK 17.0.16. Setiap simulasi dilakukan dengan jumlah 800 host. Konfigurasi yang digunakan untuk simulasi menggunakan HP ProLiant ML110 G5 (1 x [Xeon 3075 2660 MHz, 2 cores], 4GB) sebagai *host*. *Bandwidth* jaringan untuk setiap *host* diatur sebesar 1.000.000 Kbit/s (setara dengan 1 Gbit/s).

C. Model



Gambar 1. Model Penelitian

Arsitektur model penelitian ini seperti yang ditunjukkan Gambar 1 mengadopsi struktur hierarkis yang terdiri dari *global manager* dan *local manager*. Sistem deteksi Fuzzy Sugeno ditempatkan di dalam *local manager* pada

setiap host (Host X, Y, n). Fuzzy Sugeno ditempatkan pada *local manager* untuk mendeteksi apakah *host* tersebut mengalami kondisi *overload*. Hasil deteksi dari setiap *local manager* ini kemudian didistribusikan ke *global manager*, yang bertindak sebagai pusat kendali untuk mengelola keseluruhan proses *Dynamic VM Consolidation*.

D. Fuzzy Sugeno

Untuk meningkatkan dan memastikan reproduksibilitas penelitian, implementasi mekanisme host overload detection berbasis Fuzzy Sugeno yang ditanamkan pada arsitektur CloudSim direpresentasikan melalui pseudocode pada Algoritma 1.

TABLE 1
ALGORITMA OVERLOAD DETECTION

Algoritma Fuzzy Sugeno Overload Detection
1: Initialize the utilization threshold
2: Fuzzify the total requested CPU MIPS into cpuUsages
3: Fuzzify the total requested RAM into ramUsages
4: If cpuUsages or ramUsages is empty
5: Return the fallback policy decision
6: Initialize totalWeightedOutput = 0 and totalWeight = 0
7: For each cpu in cpuUsages
8: For each ram in ramUsages
9: Calculate the rule output z for the current cpu and ram category
10: Calculate the firing strength w = min(cpu membership, ram membership)
11: Update totalWeightedOutput by adding (w * z)
12: Update totalWeight by adding w
13: Endfor
14: Endfor
15: Calculate the utilization = totalWeightedOutput / totalWeight
16: Normalize the utilization by dividing it by 100
17: If normalized utilization > threshold
18: Return True
19: Else
20: Return False

E. Fuzzifikasi

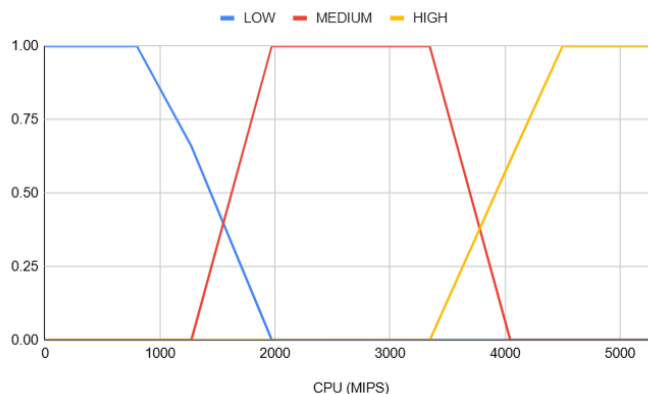
Berdasarkan kapasitas maksimal dari *host* fisik yang disimulasikan (Kapasitas CPU 5320 MIPS dan RAM 4096 MB), himpunan *fuzzy* untuk masing-masing parameter masukan dibagi menjadi tiga kategori linguistik: RENDAH (*LOW*), SEDANG (*MEDIUM*), dan TINGGI (*HIGH*). Secara matematis, derajat keanggotaan $\mu(x)$ untuk utilisasi CPU didefinisikan pada Persamaan 1 hingga 3.

$$\mu_{CPU_Rendah}(x) = \begin{cases} 1 & x \leq 800 \\ \frac{1973-x}{1173} & 800 < x \leq 1973 \\ 0 & x > 1973 \end{cases} \quad (1)$$

$$\mu_{CPU_Sedang}(x) = \begin{cases} 0 & x \leq 1273 \text{ atau } x \geq 4046 \\ \frac{x-1273}{1973-1273} & 1273 < x < 1973 \\ 1 & 1973 \leq x \leq 3346 \\ \frac{4046-x}{4046-3346} & 3346 < x < 4046 \end{cases} \quad (2)$$

$$\mu_{CPU_Tinggi}(x) = \begin{cases} 0 & x \leq 3346 \\ \frac{x-3346}{4500-3346} & 3346 < x < 4500 \\ 1 & x \geq 4500 \end{cases} \quad (3)$$

Secara visual, representasi dari fungsi keanggotaan utilisasi CPU di atas dapat dilihat pada Gambar 2.



Gambar 2. Utilisasi CPU

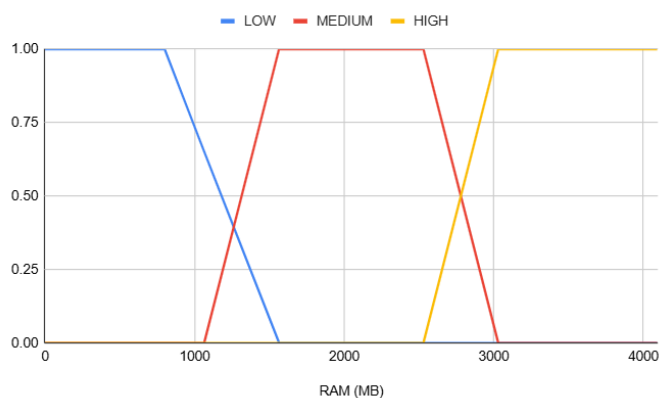
Selanjutnya, derajat keanggotaan $\mu(x)$ untuk utilisasi RAM pada host didefinisikan menggunakan Persamaan 4 hingga 6.

$$\mu_{RAM_Rendah}(x) = \begin{cases} 1 & x \leq 800 \\ \frac{1565-x}{1565-800} & 800 < x < 1565 \\ 0 & x \geq 1565 \end{cases} \quad (4)$$

$$\mu_{RAM_Sedang}(x) = \begin{cases} 0 & x \leq 1065 \text{ atau } x \geq 3030 \\ \frac{x-1065}{1565-1065} & 1065 < x < 1565 \\ 1 & 1565 \leq x \leq 2530 \\ \frac{3030-x}{3030-2530} & 2530 < x < 3030 \end{cases} \quad (5)$$

$$\mu_{RAM_Tinggi}(x) = \begin{cases} 0 & x \leq 2530 \\ \frac{x-2530}{3030-2530} & 2530 < x < 3030 \\ 1 & x \geq 3030 \end{cases} \quad (6)$$

Secara visual, representasi dari fungsi keanggotaan utilisasi RAM di atas ditunjukkan pada Gambar 3.



Gambar 3. Utilisasi RAM

F. Aturan Fuzzy

Penelitian ini mengadopsi model Fuzzy Sugeno Orde Nol. Ciri khas model ini adalah bagian output dari setiap aturan yang berupa sebuah nilai konstanta, bukan himpunan *fuzzy*. Aturan-aturan ini didefinisikan dalam basis aturan mengikuti format: IF (utilisasi CPU adalah X) AND (utilisasi RAM adalah Y) THEN (output_load = z). Nilai z adalah nilai konstanta yang telah ditetapkan yang merepresentasikan tingkat *load* untuk kombinasi input tersebut. Sistem inferensi Fuzzy Sugeno Orde Nol ini menggunakan serangkaian aturan 'if-then' sebagai berikut:

- [R1] IF CPU adalah Rendah AND RAM Rendah THEN Load = 25.

- [R2] IF CPU adalah Rendah AND RAM Sedang THEN Load = 25.
- [R3] IF CPU adalah Rendah AND RAM Tinggi THEN Load = 50.
- [R4] IF CPU adalah Sedang AND RAM Rendah THEN Load = 50.
- [R5] IF CPU adalah Sedang AND RAM Sedang THEN Load = 65.
- [R6] IF CPU adalah Sedang AND RAM Tinggi THEN Load = 95.
- [R7] IF CPU adalah Tinggi AND RAM Rendah THEN Load = 75.
- [R8] IF CPU adalah Tinggi AND RAM Sedang THEN Load = 95.
- [R9] IF CPU adalah Tinggi AND RAM Tinggi THEN Load = 95.

Penyusunan aturan ini dirancang berdasarkan prinsip prioritas asimetris, di mana parameter utilisasi CPU diberikan bobot urgensi yang lebih tinggi dibandingkan RAM. Pendekatan asimetris ini didasarkan pada landasan literatur bahwa utilisasi CPU yang mendekati atau menyentuh kapasitas maksimum merupakan faktor utama penyebab degradasi performa aplikasi, yang secara langsung berujung pada tingginya pelanggaran Service Level Agreement (SLA)[20]. Di sisi lain, meskipun RAM merupakan parameter yang krusial, utilisasi RAM lebih berdampak pada durasi waktu migrasi VM itu sendiri (migration time) dibandingkan penyebab instan dari berhentinya layanan[12].

G. Sistem Inferensi

Proses inferensi dalam sistem ini bertugas untuk mengevaluasi setiap aturan *fuzzy* yang ada dalam *rule base* terhadap input *fuzzy* dari tahap fuzzifikasi. Mesin inferensi menggunakan fungsi minimum (MIN).

$$W_i = \min(\mu A_i(x), \mu B_i(y)) \quad (7)$$

Keterangan :

- W_i adalah *firing strength* (bobot) dari aturan ke- i .
- $\mu A_i(x)$ adalah derajat keanggotaan input CPU (x) pada himpunan *fuzzy* A
- $\mu B_i(y)$ adalah derajat keanggotaan input RAM (y) pada himpunan *fuzzy* B

Fungsi ini mengambil nilai derajat keanggotaan terkecil dari kedua kondisi *input* CPU dan RAM. *Output* dari operasi MIN ini adalah *firing strength* atau bobot aturan, yang merepresentasikan seberapa kuat aturan tersebut berkontribusi pada *output* akhir.

H. Defuzzifikasi

Tahap defuzzifikasi bertujuan mengambil *output* dari semua aturan yang aktif dan mengubahnya menjadi satu nilai akhir yang tegas (*crisp*) menggunakan metode rata-rata terbobot (*weighted average*).

$$Z = \frac{\sum_{i=1}^n w_i \times z_i}{\sum_{i=1}^n w_i} \quad (8)$$

Keterangan :

- Z adalah nilai *output crisp* akhir (hasil defuzzifikasi) yang merepresentasikan tingkat utilisasi *host*
- n adalah jumlah total aturan *fuzzy*
- w_i adalah *firing strength* dari aturan ke- i
- z_i adalah nilai *output* konstanta dari aturan ke- i

I. Pengujian

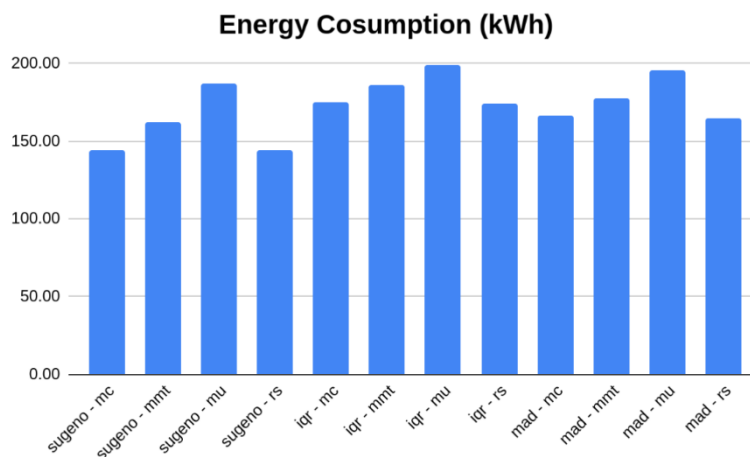
Simulasi pada penelitian ini bersifat deterministik karena menggunakan dataset PlanetLab yang sama, konfigurasi *host* yang identik, serta parameter simulasi yang tetap pada setiap skenario pengujian. Dengan karakteristik tersebut, setiap eksekusi menghasilkan keluaran yang konsisten untuk kombinasi algoritma yang sama. Oleh karena itu, pengujian tidak dilakukan secara berulang untuk menghitung simpangan baku (*standard deviation*) maupun *confidence interval* karena tidak terdapat variasi hasil yang signifikan antar eksekusi pada kondisi simulasi yang identik.

Pengujian dilakukan dengan mengamati dan mencatat hasil metode Fuzzy Sugeno dengan beberapa metode bawaan dari CloudSim. Pengujian pada penelitian ini menggunakan *energy consumption*, *VM migration*, *SLA violation*, dan *performance degradation* sebagai parameter. Untuk menangkap fluktuasi beban kerja yang dinamis, simulasi dijalankan selama periode siklus waktu 24 jam penuh dengan interval pemantauan utilisasi CPU setiap 5 menit berdasarkan data PlanetLab. Pengujian dieksekusi sebanyak 12 kali yang mewakili 12 skenario kombinasi pasangan algoritma. Skenario ini merupakan kombinasi silang antara 3 metode Host Overload Detection (Fuzzy

Sugeno, IQR, dan MAD) dengan 4 kebijakan VM Selection (MC, MMT, MU, dan RS). Setiap pasangan kombinasi dieksekusi satu kali secara deterministik terhadap dataset yang sama, sehingga hasil kinerja dari metode Fuzzy Sugeno dapat dibandingkan secara langsung dengan metode statistik bawaan CloudSim pada kondisi lingkungan yang identik.

Sebelum melakukan pengujian, metode Fuzzy Sugeno ditambahkan pada RunnerAbstract.java lalu dipanggil pada file Play.java untuk menjalankan simulasi CloudSim.

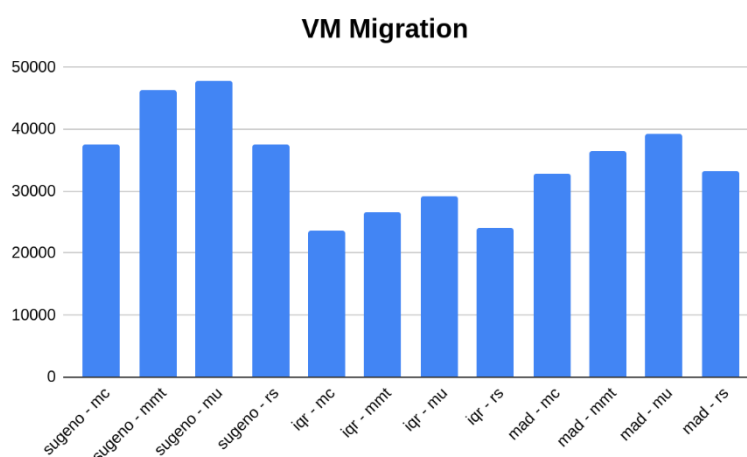
III. HASIL DAN PEMBAHASAN



Gambar 4. Diagram Energy Consumption

Hasil pengujian konsumsi energi menunjukkan bahwa metode deteksi *overload* berbasis Fuzzy Sugeno mampu memberikan peningkatan efisiensi energi yang signifikan dibandingkan metode pembanding bawaan CloudSim. Kombinasi metode Sugeno dengan kebijakan seleksi VM *Random Selection* (RS) dan *Maximum Correlation* (MC) menghasilkan konsumsi energi terendah, masing-masing sebesar 144.07 kWh dan 144.16 kWh. Nilai ini jauh lebih rendah dibandingkan metode statistik seperti IQR-MU yang mencatatkan konsumsi energi tertinggi sebesar 198.94 kWh. Penurunan konsumsi energi ini mengindikasikan bahwa metode Fuzzy Sugeno lebih efektif dalam mengidentifikasi kondisi *overload* secara dini, sehingga proses *VM consolidation* dapat dilakukan lebih cepat dan jumlah host aktif dapat dikurangi secara optimal.

Efektivitas Fuzzy Sugeno dalam menekan konsumsi energi tidak terlepas dari karakteristik *output crisp* yang dihasilkan oleh model Sugeno orde nol. *Output* ini memungkinkan sistem untuk mengambil keputusan migrasi VM secara tegas terhadap perubahan kecil pada utilisasi CPU dan RAM. Dengan demikian, *host* yang berpotensi mengalami *overload* dapat segera ditangani, sementara *host underload* dapat dialihkan ke mode tidur. Pendekatan ini secara langsung berkontribusi pada pengurangan jumlah *host* aktif dan konsumsi daya total data center.

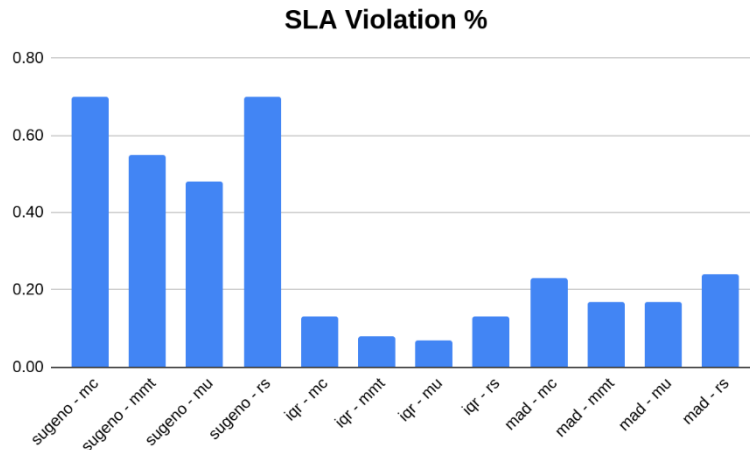


Gambar 5. Diagram VM Migration

Metode Fuzzy Sugeno menunjukkan jumlah migrasi VM yang lebih tinggi dibandingkan metode pembanding. Kombinasi Sugeno-MMT dan Sugeno-MU mencatatkan jumlah migrasi tertinggi, masing-masing sebesar 46.420 dan 47.840 migrasi. Sebaliknya, metode IQR menghasilkan jumlah migrasi yang jauh lebih rendah, misalnya IQR-MC

dengan 23.552 migrasi. Tingginya frekuensi migrasi pada metode Sugeno menunjukkan bahwa sistem deteksi yang diusulkan memiliki sensitivitas yang lebih tinggi dalam mendeteksi kondisi *overload host*.

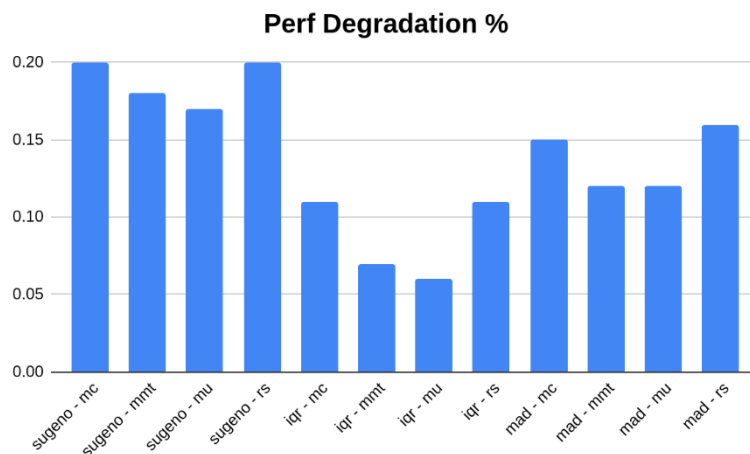
Peningkatan jumlah migrasi ini mencerminkan sifat agresif dari metode Fuzzy Sugeno dalam merespons fluktuasi beban kerja. Ketika utilisasi CPU atau RAM mendekati nilai ambang *fuzzy* “tinggi”, sistem dengan cepat mengklasifikasikan kondisi *host* sebagai *overload* dan memicu proses migrasi VM. Meskipun pendekatan ini efektif untuk mencegah *overload* berkepanjangan dan menurunkan konsumsi energi, konsekuensinya adalah meningkatnya migrasi yang harus ditangani oleh sistem.



Gambar 6. Diagram SLA Violation

Berdasarkan grafik persentase pelanggaran SLA terdapat *trade off* antara efisiensi energi dan kualitas layanan (QoS). Metode sugeno yang agresif, meskipun unggul dalam menghemat energi, menunjukkan dampak negatif yang signifikan terhadap QoS. Kombinasi Sugeno-MC dan Sugeno-RS, yang sebelumnya terbukti paling efisien secara energi, justru mencatatkan persentase pelanggaran SLA tertinggi, yaitu sebesar 0.70%. Sebagai perbandingan, metode IQR-MU hanya mencatatkan pelanggaran SLA sebesar 0.07%, sedangkan MAD-MU sebesar 0.17%.

Tingginya pelanggaran SLA pada metode Sugeno dapat dijelaskan oleh dampak langsung dari frekuensi migrasi VM yang tinggi. Proses migrasi VM menimbulkan *overhead* jaringan dan komputasi yang dapat mempengaruhi kinerja VM sementara waktu. Ketika migrasi terjadi terlalu sering, terutama pada kondisi beban yang fluktuatif, maka risiko terjadinya penurunan kualitas layanan dan pelanggaran SLA menjadi lebih besar. Hal ini menunjukkan bahwa peningkatan efisiensi energi yang dicapai oleh metode Sugeno harus dibayar dengan penurunan kualitas layanan.



Gambar 7. Diagram Perf Degradation

Selaras dengan hasil pelanggaran SLA, metode Fuzzy Sugeno juga menghasilkan tingkat degradasi performa yang lebih tinggi. Kombinasi Sugeno-MC dan Sugeno-RS mencatatkan degradasi performa tertinggi sebesar 0.20%, sedangkan metode IQR-MU berhasil menekan degradasi performa hingga 0.06%. Pola ini menunjukkan korelasi yang kuat antara jumlah migrasi VM, pelanggaran SLA, dan degradasi performa.

Degradasi performa yang lebih tinggi pada metode Sugeno disebabkan oleh meningkatnya aktivitas migrasi VM yang memerlukan alokasi ulang sumber daya secara dinamis. Meskipun dampak degradasi performa ini relatif kecil secara absolut, akumulasinya dapat menjadi signifikan pada skenario data center berskala besar dengan jumlah VM yang tinggi. Oleh karena itu, hasil ini menegaskan bahwa pemilihan metode deteksi *overload* harus mempertimbangkan kebutuhan dan prioritas operasional *data center*, apakah lebih menekankan efisiensi energi atau kualitas layanan.

Hasil pengujian menunjukkan bahwa metode Fuzzy Sugeno menghasilkan konsumsi energi terendah sebesar 144,07 kWh pada kombinasi Sugeno-RS. Nilai tersebut lebih rendah sekitar 27,5% dibandingkan kombinasi IQR-MU yang menghasilkan konsumsi energi sebesar 198,94 kWh. Hasil ini menunjukkan bahwa pendekatan Fuzzy Sugeno mampu melakukan deteksi *host overload* secara lebih adaptif.

Temuan ini konsisten dengan penelitian sebelumnya yang menunjukkan bahwa pendekatan *fuzzy* mampu meningkatkan efisiensi energi pada lingkungan *Dynamic VM Consolidation*[11], [12]. Dibandingkan pendekatan Fuzzy Mamdani pada penelitian[8], metode yang diusulkan menggunakan Fuzzy Sugeno Orde Nol dengan mekanisme defuzzifikasi *weighted average* sehingga proses inferensi dapat dilakukan dengan lebih sederhana dan efisien secara komputasi. Selain itu, penelitian ini mengevaluasi performa metode menggunakan empat metrik utama, yaitu *energy consumption*, *VM migration*, *SLA violation*, dan *performance degradation*, sehingga memberikan gambaran yang lebih komprehensif mengenai *trade-off* antara efisiensi energi dan kualitas layanan.

IV. SIMPULAN

Penelitian ini berhasil menerapkan metode Fuzzy Sugeno Orde Nol untuk deteksi *host overload* pada mekanisme *Dynamic VM Consolidation* di lingkungan *cloud computing*. Metode yang diusulkan memanfaatkan dua parameter utama, yaitu utilisasi CPU dan RAM, sebagai masukan sistem inferensi *fuzzy* untuk menentukan tingkat beban host secara adaptif. Hasil pengujian menunjukkan bahwa kombinasi Sugeno-RS menghasilkan konsumsi energi terendah sebesar 144,07 kWh, lebih rendah sekitar 27,5% dibandingkan kombinasi IQR-MU yang menghasilkan konsumsi energi sebesar 198,94 kWh. Hasil tersebut menunjukkan bahwa pendekatan Fuzzy Sugeno mampu meningkatkan efisiensi energi melalui pengurangan jumlah *host* aktif secara lebih optimal dibandingkan metode *overload detection* bawaan CloudSim. Namun demikian, peningkatan efisiensi energi tersebut diikuti oleh kenaikan jumlah *VM migration*, *SLA violation*, dan *performance degradation* sehingga menunjukkan adanya *trade-off* antara efisiensi energi dan kualitas layanan. Berdasarkan hasil tersebut, metode Fuzzy Sugeno dapat menjadi alternatif yang efektif untuk *host overload detection* pada lingkungan *cloud* yang berorientasi pada efisiensi energi. Untuk pengembangan selanjutnya, penelitian dapat diperluas dengan menerapkan fungsi keanggotaan adaptif, menambahkan parameter lain seperti *bandwidth* dan *disk I/O*, serta melakukan pengujian pada *workload* yang lebih beragam maupun lingkungan *cloud* nyata guna meningkatkan akurasi dan generalisasi model.

DAFTAR PUSTAKA

- [1] R. Dhaya *et al.*, "Energy-Efficient Resource Allocation and Migration in Private Cloud Data Centre," *Wirel. Commun. Mob. Comput.*, vol. 2022, 2022, doi: 10.1155/2022/3174716.
- [2] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," *Futur. Gener. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, 2012, doi: 10.1016/j.future.2011.04.017.
- [3] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers," *Concurr. Comput. Pract. Exp.*, vol. 24, no. 13, pp. 1397–1420, 2012, doi: 10.1002/cpe.1867.
- [4] Q. Zhou *et al.*, "Energy Efficient Algorithms based on VM Consolidation for Cloud Computing: Comparisons and Evaluations," *Proc. - 20th IEEE/ACM Int. Symp. Clust. Cloud Internet Comput. CCGRID 2020*, pp. 489–498, 2020, doi: 10.1109/CCGrid49817.2020.00-44.
- [5] A. Katal, S. Dahiyah, and T. Choudhury, *Energy efficiency in cloud computing data centers: a survey on software technologies*, vol. 26, no. 3. Springer US, 2023. doi: 10.1007/s10586-022-03713-0.
- [6] A. Fadil, "Strategi Efisiensi Energi dan Penyeimbangan Beban Kerja Layanan Cloud Computing Melalui Konsolidasi Mesin Virtual Dinamis," *Appl. Technol. Comput. Sci. J.*, vol. 3, no. 1, pp. 1–12, 2020, doi: 10.33086/atcsj.v3i1.1680.
- [7] S. Manikandan, E. Elakiya, K. C. Rajheshwari, and K. Sivakumar, "Efficient energy consumption in hybrid cloud environment using adaptive backtracking virtual machine consolidation," *Sci. Rep.*, vol. 14, no. 1, pp. 1–8, 2024, doi: 10.1038/s41598-024-72459-z.
- [8] C. Umam and G. F. Shidik, "Host Overloading Detection pada Dynamic VM Consolidation Menggunakan Fuzzy Mamdani," *Creat. Inf. Technol. J.*, vol. 4, no. 2, p. 94, 2018, doi: 10.24076/citec.2017v4i2.101.
- [9] S. Kulshrestha and S. Patel, "An efficient host overload detection algorithm for cloud data center based on exponential weighted moving average," *Int. J. Commun. Syst.*, vol. 34, no. 4, pp. 1–30, 2021, doi: 10.1002/dac.4708.
- [10] A. A. El-Moursy, A. Abdelsamea, R. Kamran, and M. Saad, "Multi-Dimensional Regression Host Utilization algorithm (MDRHU) for Host Overload Detection in Cloud Computing," *J. Cloud Comput.*, vol. 8, no. 1, 2019, doi: 10.1186/s13677-019-0130-2.
- [11] N. Baskaran and R. Eswari, "An efficient threshold-fuzzy-based algorithm for VM consolidation in cloud datacenter," *Int. J. Grid High Perform. Comput.*, vol. 13, no. 1, pp. 18–46, 2021, doi: 10.4018/IJGHPC.2021010102.
- [12] A. Choudhary, M. C. Govil, G. Singh, L. K. Awasthi, and E. S. Pilli, "Energy-efficient fuzzy-based approach for dynamic virtual machine consolidation," *Int. J. Grid Util. Comput.*, vol. 10, no. 4, pp. 308–325, 2019, doi: 10.1504/IJGUC.2019.100863.
- [13] R. Andreoli, J. Zhao, T. Cucinotta, and R. Buyya, "CloudSim 7G: An Integrated Toolkit for Modeling and Simulation of Future Generation Cloud Computing Environments," pp. 1–18, 2024, doi: 10.1002/spe.3413.
- [14] J. Byrne *et al.*, "A review of cloud computing simulation platforms & related environments," *CLOSER 2017 - Proc. 7th Int. Conf. Cloud*

- Comput. Serv. Sci.*, no. Closer, pp. 651–663, 2017, doi: 10.5220/0006373006790691.
- [15] C. Anglano, M. Canonico, and M. Guazzone, “FCMS: A fuzzy controller for CPU and memory consolidation under SLA constraints,” *Concurr. Comput. Pract. Exp.*, vol. 29, no. 5, pp. 1–17, 2017, doi: 10.1002/cpe.3968.
- [16] M. A. H. Monil and R. M. Rahman, “VM consolidation approach based on heuristics fuzzy logic, and migration control,” *J. Cloud Comput.*, vol. 5, no. 1, 2016, doi: 10.1186/s13677-016-0059-7.
- [17] L. Salimian, F. Safi Esfahani, and M. H. Nadimi-Shahraki, “An adaptive fuzzy threshold-based approach for energy and performance efficient consolidation of virtual machines,” *Computing*, vol. 98, no. 6, pp. 641–660, 2016, doi: 10.1007/s00607-015-0474-5.
- [18] D. A. Diartono, J. S. Informasi, and K. Semarang, “Analisis Perbandingan Metode Fuzzy Logic Dan Metode SAW Dalam Pemilihan Keluarga Penerima Bantuan Sosial,” vol. 8, no. 3, pp. 193–198, 2023.
- [19] C. Vijaya and P. Srinivasan, “A Hybrid Technique for Server Consolidation in Cloud Computing Environment,” *Cybern. Inf. Technol.*, vol. 20, no. 1, pp. 36–52, 2020, doi: 10.2478/cait-2020-0003.
- [20] A. Beloglazov and R. Buyya, “Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 7, pp. 1366–1379, 2013, doi: 10.1109/TPDS.2012.240.