

Komparasi Performa Metode Load Balancing Dan Framework Pada Layanan Cloud Computing

Fazza Dwi Riandy ^{1*}, Winarno Sugeng ^{1*}

¹Program Studi Informatika, Institut Teknologi Nasional

Email: fazzariandy@mhs.itenas.ac.id

Email: Winarno@itenas.ac.id

Received 27 01 2023 | Revised 03 02 2023 | Accepted 03 02 2023

ABSTRAK

Komputasi awan adalah model komputasi yang memungkinkan user untuk mengakses kumpulan sumber daya komputasi yang dapat dikonfigurasi bersama (seperti jaringan, server, penyimpanan, aplikasi, dan layanan). Sumber daya ini dapat digunakan dan dirilis dengan cepat dengan upaya manajemen minimal atau interaksi penyedia layanan. Komputasi awan bergantung pada jaringan server jarak jauh, yang dihosting di Internet, untuk menyimpan, memproses, dan mengelola data daripada server lokal atau komputer pribadi. Ini memungkinkan pengguna untuk mengakses data dan aplikasi yang sama dari mana saja, kapan saja, di perangkat apa saja, tanpa perlu instalasi atau pemeliharaan. Manfaat komputasi awan meliputi penghematan biaya, skalabilitas, fleksibilitas, keamanan, dan keandalan. Namun, ini juga menghadirkan tantangan baru, seperti vendor lock-in, privasi dan kedaulatan data, serta kepatuhan.

Kata kunci: *Komputasi Awan, Layanan, Jaringan.*

ABSTRACT

Cloud computing is a model of computing that enables on-demand access to a shared pool of configurable computing resources (such as networks, servers, storage, applications, and services). These resources can be rapidly deployed and released with minimal management effort or service provider interaction. Cloud computing relies on a network of remote servers, hosted on the Internet, to store, process, and manage data rather than a local server or a personal computer. This allows users to access the same data and applications from anywhere, at any time, on any device, without the need for installation or maintenance. The benefits of cloud computing include cost savings, scalability, flexibility, security, and reliability. However, it also introduces new challenges, such as vendor lock-in, data privacy and sovereignty, and compliance. The methods commonly used are Virtual Network Computing and Load Balancing.

Keywords: *Cloud Computing, Service, Network.*

1. PENDAHULUAN

Cloud computing mengacu pada penggunaan layanan komputasi—termasuk *server*, penyimpanan, basis data, jaringan, perangkat lunak, analitik, dan kecerdasan—melalui Internet (awan). Perusahaan yang menawarkan layanan komputasi ini disebut penyedia *cloud* dan biasanya membebaskan biaya kepada pengguna berdasarkan penggunaan, bukan tarif tetap. Hal ini memungkinkan pengguna untuk mengakses sumber daya teknologi sesuai kebutuhan, daripada membangun dan memelihara infrastruktur internal.

Cloud computing telah membuat terobosan besar dalam sektor IT. Dengan kemunculannya telah benar-benar merevolusi sektor IT. *Cloud computing* telah memainkan peran penting dalam melayani permintaan untuk penyimpanan dan infrastruktur (Rashid & Chaturvedi, 2019). Salah satu kemampuan *cloud computing* adalah kemampuannya untuk menyediakan sumber daya seperti perangkat keras dan perangkat lunak melalui jaringan yang dapat diakses dimana saja dan kapan saja. Secara garis besar kita dapat membagi *cloud computing* sebagai (Rashid & Chaturvedi, 2019) :

- a. *Private cloud*
Jenis *cloud computing* ini digunakan untuk organisasi atau bisnis, misalnya *cloud* untuk kebutuhan organisasi atau perusahaan tertentu.
- b. *Public cloud*
Penyedia *public cloud* seperti Google, Amazon, Microsoft, dll. *Public cloud* menyediakan infrastruktur dan layanan kepada publik atau organisasi apa pun.
- c. *Community cloud*
Dalam *community cloud* layanan dan infrastruktur disediakan untuk organisasi dengan minat yang serupa.
- d. *Hybrid cloud*
Jenis *cloud* ini merupakan campuran dari *private* dan *public cloud*. Meski fungsinya bercampur, tetapi tetap saja masing-masing memiliki identitas individu masing – masing, maka informasi yang di dalamnya tidak akan tercampur satu dengan yang lainnya.

Model	Ruang lingkup	Pengelola	Level Keamanan
Public Model	Masyarakat umum dan industri	<i>Cloud services provider</i>	Rendah
Private Model	Organisasi tunggal	Organisasi tunggal	Tinggi
Community Model	Organisasi memiliki kebijakan dan keamanan serupa	Banyak organisasi atau <i>cloud services providers</i>	Tinggi
Hybrid Model	Publik dan organisasi	Publik dan organisasi	Menengah

Table I Perbedaan Model *Cloud Computing*

Sumber : Rashid & Chaturvedi, 2019

Cloud computing memberikan sejumlah manfaat lain, termasuk kemampuan untuk meningkatkan atau menurunkan skala sumber daya sesuai kebutuhan, mengurangi biaya, serta meningkatkan keandalan dan keamanan. Ini juga memungkinkan fleksibilitas yang lebih besar, karena pengguna dapat mengakses aplikasi dan data mereka dari perangkat apa pun dengan koneksi internet.

Load balancer as a service adalah sebuah layanan untuk mendistribusikan permintaan user secara merata pada dua atau lebih komputer. Untuk mengoptimalkan kinerja *load balancer* perlu menggunakan metode tambahan yaitu *round robin*. *round robin* adalah algoritma yang digunakan dalam *load balancer* pada *cloud computing*, merupakan algoritma dengan metode sederhana dan mudah untuk diterapkan (G. Ramadhan, 2019), metode *least connections* adalah sebuah algoritma yang bekerja dengan cara menjadwalkan koneksi pada suatu jaringan saat akan mengakses server dengan cara memilih server manakah yang memiliki koneksi paling sedikit pada waktu tertentu (R. Adenan, 2018), dan metode *source ip* adalah algoritma penjadwalan dengan cara mengatur koneksi jaringan berdasarkan alamat ip yang terkoneksi secara konsisten untuk diarahkan ke server yang sama (H. K. Cakrawerdya, 2017). Metode metode tersebut merupakan metode *load balancer* yang biasa digunakan saat merancang sebuah *cloud*.

Virtual Network Computer merupakan salah satu framework yang dapat digunakan dalam implementasi *cloud gaming* pada layanan *cloud computing*. Cloud Gaming berbasis *web browser* ini memiliki kelebihan tersendiri karena semua komputer pasti memiliki software web browser yang sudah terinstal dan mudah serta familiar dalam penggunaannya (Sadewo et al., 2019).

Skyegrid adalah sebuah framework *cloud gaming* (GaaS) yang memungkinkan user dapat memainkan game secara remote.

Cloud gaming adalah jenis layanan yang memungkinkan pengguna bermain *game* melalui layanan *cloud*, bukan secara lokal. Artinya, game dijalankan di server jarak jauh dan video serta audio didistribusikan melalui internet ke perangkat pengguna (Asri et al., 2020). Cloud gaming dapat diakses melalui browser web, aplikasi khusus, atau konsol game. Keuntungan utama cloud gaming adalah memungkinkan pengguna bermain game di berbagai perangkat, termasuk perangkat yang mungkin tidak memiliki persyaratan perangkat keras atau perangkat lunak untuk menjalankan game secara lokal.

2. TINJAUAN TEORITIS

2.1 *Cloud Computing*

Cloud Computing adalah teknologi yang menghadirkan perangkat lunak komputasi serta sumber daya perangkat keras sebagai layanan melalui internet. *Cloud computing* menyediakan akses ke server, *storage*, database, dan serangkaian layanan aplikasi yang luas melalui internet (Almaarif et al., 2020). Platform layanan cloud seperti Amazon Web Services memiliki dan melakukan pemeliharaan terhadap *hardware* yang terhubung dengan jaringan yang dibutuhkan untuk layanan aplikasi ini, sementara pengguna menggunakan apa yang dibutuhkan melalui aplikasi web.

2.2 Klasifikasi *Cloud Computing*

Berdasarkan penelitian yang dilakukan oleh Saputra et al., 2020 *cloud computing* mempunyai tiga tingkatan layanan, yaitu:

a. *Software as a Service (SaaS)*

Layanan yang dibangun untuk menyediakan perangkat lunak maupun aplikasi yang dapat diakses oleh pengguna.

b. *Platform as a Service (PaaS)*

merupakan layanan yang diberikan kepada konsumen untuk membagikan aplikasi yang dibuat konsumen atau diperoleh ke infrastruktur *cloud computing* menggunakan bahasa pemrograman dan peralatan yang didukung oleh provider

c. *Infrastructure as a Service (IaaS)*

layanan yang menyewakan *resource* seperti, media penyimpanan, *processing power*, *memory*, sistem operasi, kapasitas jaringan dan lainnya, yang dapat digunakan oleh *user* untuk menjalankan aplikasi yang dimilikinya.

d. *Gaming as a Service (GaaS)*

"Gaming as a Service" (GaaS) mengacu pada penggunaan video game sebagai layanan Dengan GaaS, pengguna membayar biaya berlangganan atau sejumlah kecil untuk setiap sesi. Model ini mirip dengan model *cloud computing* lainnya seperti *Software as a Service (SaaS)* atau *Platform as a Service (PaaS)*. Ada beberapa manfaat model GaaS untuk pengembang dan pemain game. Untuk pengembang, GaaS memungkinkan mereka menghasilkan pendapatan berkelanjutan dari game mereka. Itu juga memungkinkan mereka untuk terus memperbarui dan meningkatkan permainan, menambahkan konten dan fitur baru untuk membuat pemain tetap terlibat (Zaiets, 2020). Untuk pemain, GaaS menawarkan cara yang lebih nyaman dan terjangkau untuk mengakses banyak game. Daripada membeli setiap game satu per satu, pemain dapat membayar satu biaya berlangganan untuk mengakses game. Ini juga memungkinkan pemain untuk mencoba game baru tanpa melakukan pembelian penuh, yang dapat membantu saat memutuskan apakah akan berinvestasi dalam game yang lebih mahal.

2.3 *Load Balancing*

Load balancing merupakan teknik untuk membagi beban jaringan pada dua atau lebih jalur koneksi secara seimbang, dengan tujuan untuk meringankan beban trafik, memaksimalkan nilai *throughput*, memperkecil *response time*, dan menghindari *overload* pada pada jalur koneksi (Lema, 2019). *Load balancing* digunakan apabila suatu server maupun jaringan memiliki user yang telah melebihi kapasitas dari server tersebut. Terdapat algoritma – algoritma *load balancing* diantaranya (Lema, 2019) :

a. Round robin

Algoritma *round robin* bekerja dengan cara membagi beban trafik jaringan bergiliran dan berurutan dari satu server ke server lain sehingga membentuk sebuah *loop*

b. Least connection

Algoritma *least connection* bekerja dengan cara membagi beban trafik berdasarkan berapa banyak koneksi yang sedang dilayani oleh server. Server dengan pelayanan koneksi paling sedikit akan diberikan beban yang berikutnya masuk.

c. Source IP

algoritma *source IP* adalah algoritma penjadwalan dengan cara mengatur *request* dari alamat IP secara acak untuk diarahkan ke server yang sama.

3. HASIL PENELITIAN DAN PEMBAHASAN

3.1 Pengujian *Load Balancing*

Berdasarkan penelitian yang dilakukan oleh Saputra et al pada tahun 2020 dengan judul "Implementasi Infrastructure as a Service pada Cloud Computing Menggunakan Metode Load Balancing", didapatkan hasil pengujian dengan metode *load balancing* sebagai berikut

Parameter	Server	Jaringan		
		Lokal	Sepi	Sibuk
Throughput (Mbps)	Round Robin	0.14	0.14	0.10
	Least Connection	0.13	0.14	0.10
	Source IP	0.13	0.13	0.10
Delay (ms)	Round Robin	17.21	17.19	23.70
	Least Connection	18.04	17.90	23.19
	Source IP	19.01	19.01	24.80
Jitter (ms)	Round Robin	17.16	17.14	23.65
	Least Connection	17.99	17.86	23.13
	Source IP	18.96	18.96	24.83
Packet Loss (%)	Round Robin	0.00	0.00	0.46
	Least Connection	0.00	0.00	0.37
	Source IP	0.04	0.02	0.84

Pengujian diatas dilakukan dengan menggunakan tiga jenis keadaan untuk pengukurannya, keadaan lokal, *web server* diakses secara LAN. Keadaan sepi yaitu keadaan dimana *web server* diakses secara remote dengan keadaan trafik yang sepi (*request* terhadap server rendah), sedangkan keadaan terakhir yaitu sibuk, kondisi ini terjadi dimana trafik pada server padat (*request* terhadap server tinggi).

3.2 Pengujian *Virtual Network Computing*

Berdasarkan penelitian yang dilakukan oleh Sadewo et al pada tahun 2019 dengan judul "Implementasi Dan Analisis Cloud Gaming Pada Layanan Gaas Berbasis Web Browser Menggunakan Metode Virtual Network Computing", didapatkan hasil sebagai berikut :

Parameter	Resolusi	Bandwith (Mbps)		
		1.5	3	6
Throughput (Kbps)	Low	0.85 Mbps	0.14 Mbps	0.19 Kbps
	Medium	0.85 Kbps	0.00082 Kbps	0.162 Kbps
	High	0.80 Kbps	0.81 Kbps	0.85 Kbps
Delay (ms)	Low	1660 ms	750 ms	700 ms

	Medium	1660 ms	770 ms	700 ms
	High	1660 ms	16330 ms	830 ms
Jitter (ms)	Low	1660 ms	750 ms	700 ms
	Medium	1660 ms	770 ms	700 ms
	High	1660 ms	16330 ms	830 ms
Packet Loss (%)	Low	4.2%	3.0%	0.9%
	Medium	5.4%	3.6%	1.5%
	High	6.3%	4.5%	2.4%

Pengujian diatas dilakukan dengan cara memainkan sebuah game dengan layanan *Game as a Service* (GaaS) pada *browser* Google Chrome. Pengujian dilakukan pada tiga nilai *bandwith* yang berbeda – beda, terdapat tiga pengujian *bandwith* yaitu pengujian pada *bandwith* 1.5 Mbps, 3 Mbps, dan 6 Mbps. Pada pengujian diatas peneliti tidak menguji parameter *jitter*, maka dari itu diasumsikan nilai dari *jitter* sama dengan nilai *delay*, karena nilai *jitter* merupakan bagian dari *delay*.

3.3 Pengujian Skyegrid

Berdasarkan penelitian yang dilakukan oleh Asri et al pada tahun 2020 dengan judul “Implementasi Dan Analisis Cloud Gaming Skyegrid Pada Perangkat Android”, didapatkan hasil sebagai berikut :

Pengujian ke -	Parameter	Bandwith	
		20 Mbps	30 Mbps
1	Delay (ms)	510 ms	510 ms
	Throughput	3.12 Mbps	3.3 Mbps
	Packet Loss	4.1 %	3.55 %
	Jitter	510 ms	510 ms
2	Delay (ms)	630 ms	630 ms
	Throughput	3.42 Mbps	5.65 Mbps
	Packet Loss	6.06 %	5.35 %
	Jitter	630 ms	610 ms

Pengujian diatas dilakukan dengan cara memainkan dua game yang berbeda dengan layanan *Game as a Service* (GaaS) pada *framework* Skyegrid. Pengujian dilakukan pada dua nilai *bandwith* yang berbeda, terdapat dua pengujian *bandwith* yaitu pengujian pada *bandwith* 20 Mbps, dan 30 Mbps. Pada pengujian diatas peneliti tidak menguji parameter *jitter*, maka dari itu diasumsikan nilai dari *jitter* sama dengan nilai *delay*, karena nilai *jitter* merupakan bagian dari *delay*.

4. KESIMPULAN & SARAN

4.1 Kesimpulan dan Rekomendasi

Pada penelitian metode *load balancing* dapat disimpulkan bahwa algoritma *round robin* lebih unggul dari algoritma yang lainnya pada kondisi jaringan yang stabil (lokal dan sepi), sedangkan pada kondisi jaringan sibuk algoritma *least connection* lebih unggul daripada algoritma lainnya. Maka dapat diambil kesimpulan, untuk perancangan *cloud computing* yang berbasis infastruktur (IaaS) maka dapat menggunakan dua metode berikut, *round robin*, dapat digunakan apabila layanan *cloud computing* memiliki jaringan yang relatif stabil (request terhadap server rendah). Untuk algoritma *least connection*, dapat diterapkan pada server *cloud* yang trafik pada server relatif sibuk (request terhadap server tinggi).

Pada penelitian *framework* Skyegrid dan VNC dapat disimpulkan bahwa, *framework* Skyegrid lebih unggul dibandingkan dengan VNC secara keseluruhan, *framework* VNC memiliki delay yang sangat tinggi dari Skyegrid yaitu 830 ms pada pengujian *bandwidth* 6 Mbps dengan pengujian pada resolusi tinggi, sedangkan pada pengujian *framework* Skyegrid didapatkan delay 510 ms dan 510 ms pada pengujian *bandwidth* 20 Mbps dan 30 Mbps di pengujian ke-1 sedangkan didapatkan delay 630 ms dan 610 ms pada pengujian ke-2 dengan *bandwidth* 20 Mbps dan 30 Mbps. Namun pada parameter *packet loss* VNC lebih unggul dari Skygrid dengan nilai 2.4 % pada pengujian *bandwidth* 6 Mbps dengan pengujian pada resolusi tinggi, sedangkan pada *framework* skygrid *packet loss* yang didapat pada pengujian *bandwidth* 20 Mbps dan 30 Mbps adalah 4.1 % dan 3.55 % pada pengujian ke-1 sedangkan didapatkan *packet loss* sebesar 6.06% dan 5.35 % pada pengujian ke-2 dengan *bandwidth* 20 Mbps dan 30 Mbps. Dari hasil pengujian dapat disimpulkan bahwa apabila *user* ingin menggunakan layanan GaaS maka direkomendasikan untuk menggunakan *framework* Skyegrid, karena secara keseluruhan *framework* Skyegrid lebih unggul dibandingkan dengan VNC, tetapi untuk mendapatkan pengalaman yang serupa *user* harus menyediakan setidaknya *bandwidth* 20 Mbps untuk merasakan pengalaman yang serupa, namun apabila *user* memiliki *bandwidth* yang terbatas maka dapat menggunakan *framework* VNC dengan minimal *bandwidth* 6 Mbps agar dapat merasakan pengalaman layanan GaaS yang stabil.

5. DAFTAR PUSTAKA

- Almaarif, A., Widyastoro, A. N., & Saedudin, R. R. (2020). Analisis Performa Network Cloud Dengan Metode Quality of Services Pada Penyedia Teknologi Layanan Cloud Computing Microsoft Azure Dan Amazon Web Services. *E-Proceeding of Engineering*, 7(2), 6965–6974.
<https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/12655>
<https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/download/12655/12386>
- Asri, F., Rizqa, A., & Maisura, M. (2020). Implementasi Dan Analisis Cloud Gaming Skyegrid Pada Perangkat Android. *Cyberspace: Jurnal Pendidikan Teknologi Informasi*, 4(1), 59.
<https://doi.org/10.22373/cj.v4i1.7092>
- Lema, D. (2019). *Load Balancing 2 Jalur Internet Menggunakan Mikrotik Round Robin*. 05(02), 137–143.
- Rashid, A., & Chaturvedi, A. (2019). Cloud Computing Characteristics and Services A Brief Review. *International Journal of Computer Sciences and Engineering*, 7(2), 421–426.
<https://doi.org/10.26438/ijcse/v7i2.421426>
- Sadewo, B. M., Munadi, R., & Sussi. (2019). Implementasi Dan Analisis Cloud Gaming Pada Layanan Gaas Berbasis Web Browser Menggunakan Metode Virtual Network Computing. *E-Proceeding of Engineering*, 6(1), 784–793.
- Saputra, A., Priyanto, H., & Safriadi, N. (2020). Implementasi Infrastructure as a Service pada Cloud Computing Menggunakan Metode Load Balancing. *Jurnal Sistem Dan Teknologi Informasi (Justin)*, 8(4), 397. <https://doi.org/10.26418/justin.v8i4.39980>
- G. Ramadhan, R. L. (2019). Analisis Performansi Load Balancing pada Cloud Computing Menggunakan Algoritma Throttled dan Greedy. *Open Library Telkom University*.
- H. K. Cakrawerdya, R. M. (2017). Implementation load balancer as a Service in Openstack Based on. *Computer Applications in Technology*, 240-245.
- Zaiets, S. (2020, 10 09). *Why AAA Studios Shift to Games-as-a-Service (GaaS) Model*. Diambil kembali dari Gridly: [https://www.gridly.com/blog/games-as-a-service/#:~:text=%2Da%2DService%3F-,Games%2Das%2Da%2DService%20\(GaaS\)%20is%20a,form%20of%20in%2Dgame%20purchases.](https://www.gridly.com/blog/games-as-a-service/#:~:text=%2Da%2DService%3F-,Games%2Das%2Da%2DService%20(GaaS)%20is%20a,form%20of%20in%2Dgame%20purchases.)