ISSN : 2085-4218

Implementasi dan Analisis Performansi Server Aplikasi Mobicents pada Cloud Computing dengan WebRTC

Al Asyari Pratama[1,*], Rendy Munadi[1], Ratna Mayasari[1]
1 Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

Abstrak. Teknologi WebRTC *(Web Real-Time Communication)* merupakan teknologi yang memungkinkan komunikasi *real-time* menggunakan protokol Websockets dan *web browser* sebagai klien. Pengembangan teknologi WebRTC harus didukung oleh infrastruktur jaringan yang bagus, baik bandwidth maupun keandalan sistem dalam melakukan pelayanan. Selain itu, infrastruktur juga harus bersifat fleksibel untuk kemudahan dalam akses server WebRTC. Salah satu teknologi yang menawarkan fitur tersebut yaitu *Cloud Computing*.

Dalam penelitian ini diimplementasikan suatu server yang mendukung WebRTC pada *Cloud Computing*. Proxmox VE digunakan sebagai *platform* untuk membangun infrastruktur *Cloud Computing* dan Mobicents *Application Server* sebagai server VoIP yang mendukung WebRTC. QoS *(Quality of Service)*, CPU *usage*, *memory usage*, serta keandalan sistem dalam menangani ribuan panggilan menjadi parameter yang digunakan untuk melihat performansi dari Mobicents *Application Server* pada *Cloud Computing*.

Dari hasil pengukuran parameter QoS, server dengan *background traffic* 80 Mbps memiliki nilai *one way delay* dan *jitter* terbesar dengan nilai *one way delay* 11.541736 ms dan *jitter* 1.115947 ms. Besarnya nilai *throughput* berada dalam *interval* 1.605 Mbps dan 1.730 Mbps. Persentase panggilan sukses terbesar yaitu 99.81%, diperoleh saat kondisi server dengan beban *traffic* panggilan 2000 panggilan/detik. Pada kondisi ini, *instance* memiliki spesifikasi 2CPUs 1GB dengan nilai CPU usage 62.82% dan nilai memory usage 410 MB.

Kata Kunci: *Cloud Computing*, Mobicents, VoIP, WebRTC, WebSocket

1. Pendahuluan

Teknologi jaringan dan multimedia mengalami perkembangan yang sangat pesat. Perkembangan teknologi yang sangat pesat ini mendorong teknologi *web* dan internet tidak hanya menjadi media informasi, namun juga menjadi media komunikasi yang banyak digunakan. Dengan adanya teknologi WebRTC, diharapkan teknologi ini dapat menjadi salah satu media komunikasi berbasis layanan VoIP yang universal.

Perkembangan teknologi web sendiri didesain dengan pendekatan aplikasi dekstop, mengedepankan kecepatan akses, kemudahan bagi pengguna dan keinteraktifan[HYPERLINK \l "RMu12" 1]. Teknologi WebRTC juga harus didukung oleh infrastruktur jaringan yang bagus, baik *bandwidth* serta keandalan sistem dalam melakukan pelayanan. Selain itu, infrastruktur juga harus bersifat fleksibel untuk kemudahan dalam akses server WebRTC seperti *deployment*, *maintenance*, pengelolaan, dll. *Cloud Computing* adalah teknologi yang memiliki fitur tersebut.

Dalam penelitian diimplementasikan Mobicents *Application Server* yang mendukung WebRTC pada *Cloud Computing*. Implementasi menggunakan Proxmox VE sebagai *platform* untuk membangun infrastruktur *Cloud Computing*. Setelah itu dilakukan pengukuran dan analisis performansi Mobicents terhadap parameter QoS seperti *one way delay*, *jitter* dan *throughput*, CPU *usage*, *memory usage* serta kemampuan server dalam menangani ribuan panggilan yang masuk ke dalam sistem.

ISSN : 2085-4218

2. Dasar Teori

2.1 Voice over Internet Protocol (VoIP)

Voice over Internet Protocol adalah teknologi yang memungkinkan komunikasi suara, video dan data secara *real-time* melalui jaringan *Internet Protocol* (IP)2]}.

2.2 Session Initiation Protocol (SIP)

SIP merupakan *signaling protocol* yang digunakan pada VoIP. SIP beroperasi pada lapisan aplikasi yang berfungsi untuk memulai sesi pengguna untuk transmisi multimedia seperti suara dan video[HYPERLINK \l "Ren11" 3].

2.3 Web Real-Time Communication (WebRTC)

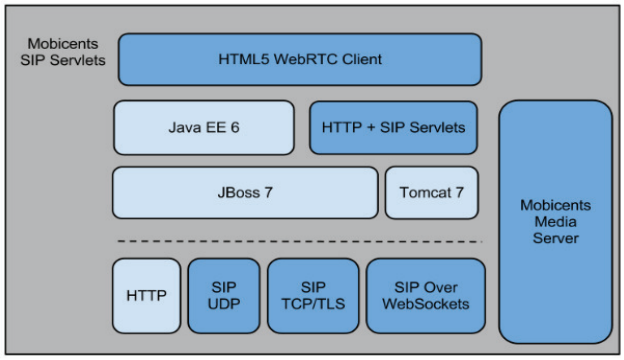
WebRTC merupakan sebuah teknologi yang memungkinkan aplikasi web untuk melakukan komunikasi *real-time* tanpa menggunakan *plugin*4]*}*. Perkembangan dari WebRTC dapat dilihat dari semakin banyaknya layanan web yang mendukung teknologi ini. Beberapa web browser yang mendukung WebRTC adalah Chrome, Firefox dan Opera.

2.4 WebSocket

WebSocket adalah protocol jaringan yang mendefinisikan bagaimana server dan klien dapat berkomunikasi melalui *web*[HYPERLINK \l "Wan13" 5]. Websocket membuat komunikasi *real-time* jauh lebih efisien karena menghemat *bandwidth*, CPU *power* dan *latency*. WebSocket mengurangi *latency* karena sekali koneksi WebSocket didirikan, server dapat mengirim pesan yang telah tersedia.

2.5 Mobicents SIP Servlets

Mobicents SIP Servlets adalah server aplikasi yang berjalan di atas JBoss AS (*Application Server*) dan *platform* ini bertujuan untuk membantu pengguna untuk membuat, menyebarkan, dan mengelola layanan dan aplikasi yang mengintegrasikan suara, video dan data secara *real-time* melalui jaringan IP (*Internet Protocol*)6]}.



Gambar 1. Arsitektur Mobicents SIP Servlets[HYPERLINK \l "Mob25" 7]

2.6 Cloud Computing

Cloud Computing merupakan sebuah teknologi untuk memindahkan layanan seperti *processor*, *storage*, *network* dan *software* ke dalam jaringan atau internet dan dapat diakses menggunakan pola akses *remote*8]}.

2.7 Quality of Sevice (QoS)

Quality of Service merupakan parameter yang menunjukkan kemampuan suatu jaringan untuk menyediakan layanan pada berbagai jenis *platform* teknologi.

2.7.1 One way delay

One way delay merupakan waktu yang diperlukan sebuah paket dari pengirim ke penerima. *One way delay* yang distandarkan oleh ITU-T G.114 untuk komunikasi suara dan video adalah kurang dari 150 ms[HYPERLINK \l "Nov07" 9].

ISSN : 2085-4218

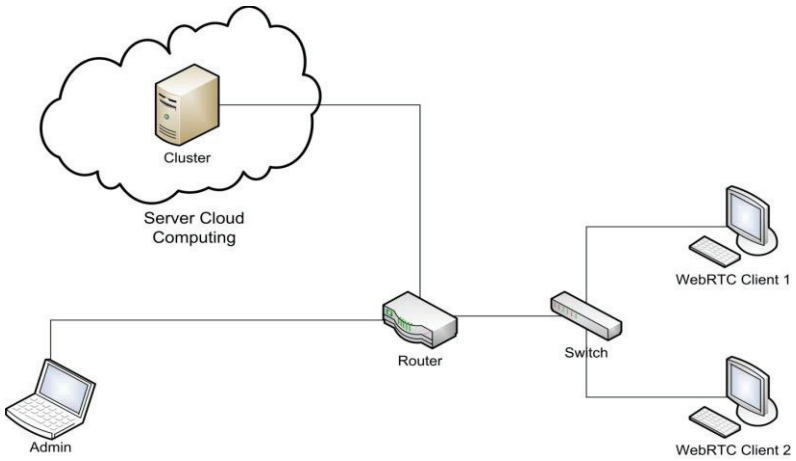
2.7.2 Jitter

Jitter didefinisikan sebagai variasi *delay* yang diakibatkan oleh panjang antrian dalam suatu pengolahan data dan *reassemble* paket-paket data di akhir pengiriman akibat kegagalan sebelumnya. Cisco menetapkan bahwa jitter untuk komunikasi *real-time* seperti video dan suara tidak boleh melebihi 100 ms10]}.

2.7.3 Throughput

Throughput pada jaringan telekomunikasi merupakan rata-rata pengiriman sukses melalui saluran telekomunikasi dalam suatu pengiriman. *Throughput* diukur dalam satuan *bits per second* (bps atau bit/s).

3. Perancangan dan Implementasi



Gambar 2. Topologi Jaringan

Dalam pengujian sistem, dilakukan dua skenario pengukuran. Skenario pertama, pengukuran QoS dengan variasi *background traffic*. Skenario kedua, pengukuran CPU *usage*, *memory usage*, serta persentase panggilan sukses yang mampu ditangani sistem dengan variasi beban *traffic* panggilan *(emulate call)*.

4. Pengujian dan Analisis

Pada bab ini akan dibahas mengenai hasil dari pengujian sistem yang telah diimplementasikan. Parameter yang diukur yaitu QoS dari WebRTC *video call*, persentase panggilan sukses dari *emulate call* yang masuk ke dalam sistem, serta CPU *usage* dan *memory usage* dari *instance*.

4.1 Pengujian QoS

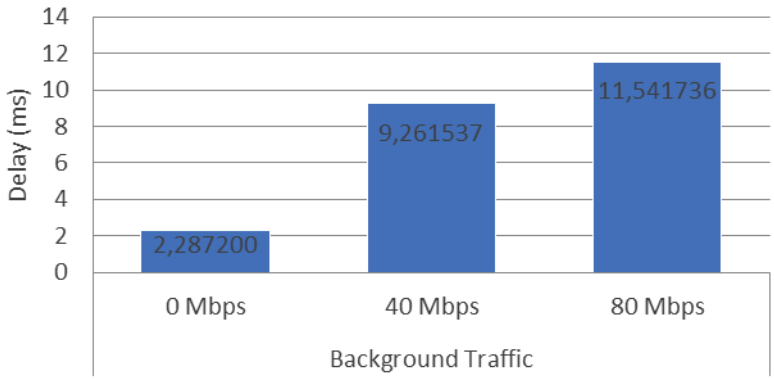
Pada skenario ini dilakukan pengukuran QoS dari WebRTC *video call* menggunakan server WebRTC yaitu Mobicents SIP Servlets Jboss AS 7. Server diinstal dan dikonfigurasi di *instance* yang menggunakan sistem operasi Ubuntu Server 14.04 64-bit. Server mempunyai spesifikasi 4CPUs 2GB. Parameter yang diuji meliputi *one way delay*, *jitter* dan *throughput*.

4.1.1 Tujuan Pengukuran

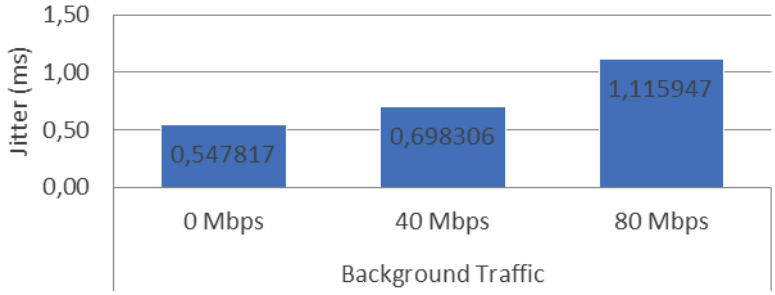
Pengukuran Pengukuran ini bertujuan untuk mengetahui nilai *one way delay, jitter* dan *throughput* dari layanan WebRTC *video call* mengunakan Mobicents *Application Server*. Sistematika penggukuran dari layanan yang digunakan menggunakan variasi background traffic 0 Mbps, 40 Mbps dan 80 Mbps. Hal ini dilakukan untuk melihat kualitas dari layanan ketika terjadi penyempitan pada *bandwidth* saluran transmisi dari jaringan yang diimplementasikan.

ISSN : 2085-4218

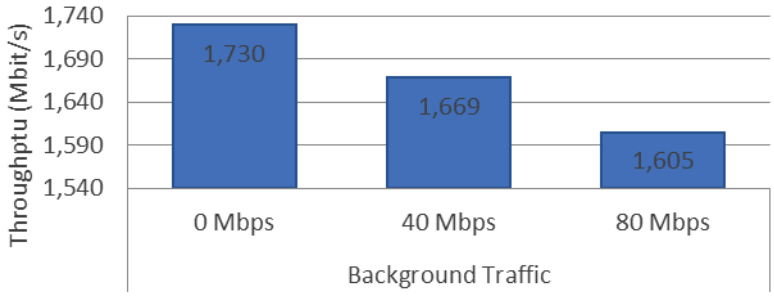
4.1.2 Hasil Pengukuran



Gambar 3. Hasil Pengukuran *One Way Delay*



Gambar 4. Hasil Pengukuran *Jitter*



Gambar 5. Hasil Pengukuran *Throughput*

4.1.3 Analisis Pengujian

Dari hasil pengukuran parameter QoS yang dilakukan, kondisi server dengan *background traffic* 0 Mbps memiliki nilai *one way delay* dan *jitter* terkecil dengan nilai *one way delay* 2.287200 ms dan *jitter* 0.547817 ms. Sedangkan kondisi server dengan *background traffic* 80 Mbps memiliki nilai *one way delay* dan *jitter* terbesar dengan nilai *one way delay* 11.541736 ms dan *jitter* 1.115947 ms. Besarnya nilai *throughput* berada dalam *interval* 1.605 Mbps dan 1.730 Mbps. Hasil pengujian yang diperoleh memenuhi standar yang ditetapkan dan tergolong sebagai kualitas layanan yang sangat baik.

Hasil pengukuran menunjukkan bahwa besar nilai *one way delay* dan *jitter* berbanding lurus dengan besarnya *background traffic* yang digunakan. Hal ini dikarenakan dengan semakin besarnya nilai *background traffic*, maka menyebabkan terjadinya penyempitan *bandwidth* dari saluran transmisi yang digunakan. Penyempitan *bandwidth* akan menyebabkan *traffic* pengiriman data akan padat sehingga waktu yang dibutuhkan paket dari pengirim ke penerima akan memakan waktu yang lebih lama dan juga jaringan mejadi semakin tidak stabil yang menyebabkan nilai variasi *delay* menjadi semakin bervariasi. Berbeda dengan *delay* dan *jitter,* nilai *throughput* berbanding terbalik dengan besarnya *background traffic* yang digunakan. Naiknya nilai *background traffic* menyebabkan kecepatan pengiriman paket dari pengirim ke penerima pun menurun.

ISSN : 2085-4218

4.2 Pengujian dengan Emulate Call

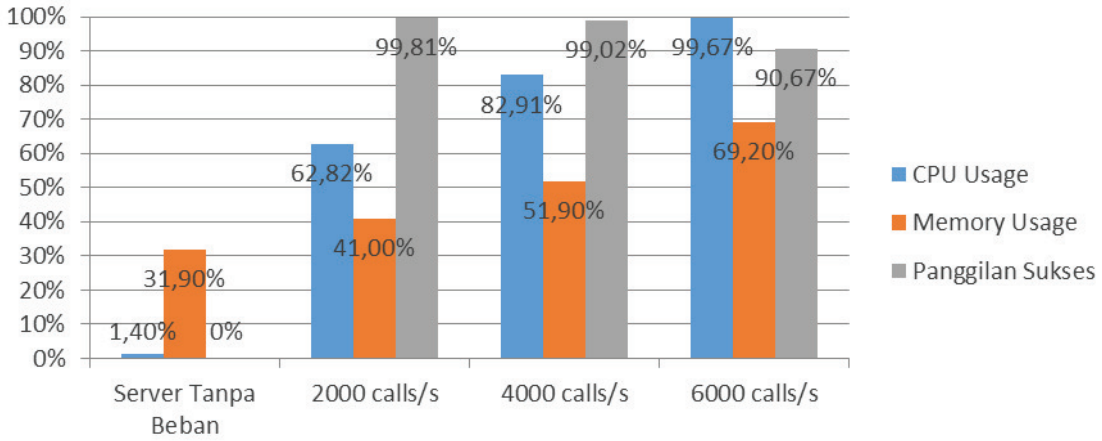
Pada skenario ini, dilakukan pengukuran ketika dilakukan *emulate call* ke server WebRTC yaitu Mobicents SIP Servlets Jboss AS 7. Parameter yang diuji yaitu persentase panggilan sukses, *CPU usage* dan *memory usage*. Pengukuran dilakukan dengan variasi *emulate call* 2000 calls/s, 4000 calls/s, 6000 calls/s, 8000 calls/s dan 10000 calls/s. Pengujian keandalan sistem dilakukan pada dua sistem yang memiliki spesifikasi yang berbeda yaitu 2CPUs 1GB dan 3CPUs 1GB. Berikut adalah konfigurasi *emulate call* pada dua *instance* berbeda:

1. Instance 2CPUs 1GB
   - Jumlah panggilan simultan maksimal yang berada di dalam sistem yaitu 10000 panggilan.
   - Skenario akan dihentikan apabila sistem sudah menangani panggilan sebanyak 1.2 juta panggilan.
2. Instance 3CPUs 1GB
   - Jumlah panggilan simultan maksimal yang berada di dalam sistem yaitu 100000 panggilan.
   - Skenario akan dihentikan apabila sistem sudah menangani panggilan sebanyak 4.8 juta panggilan.

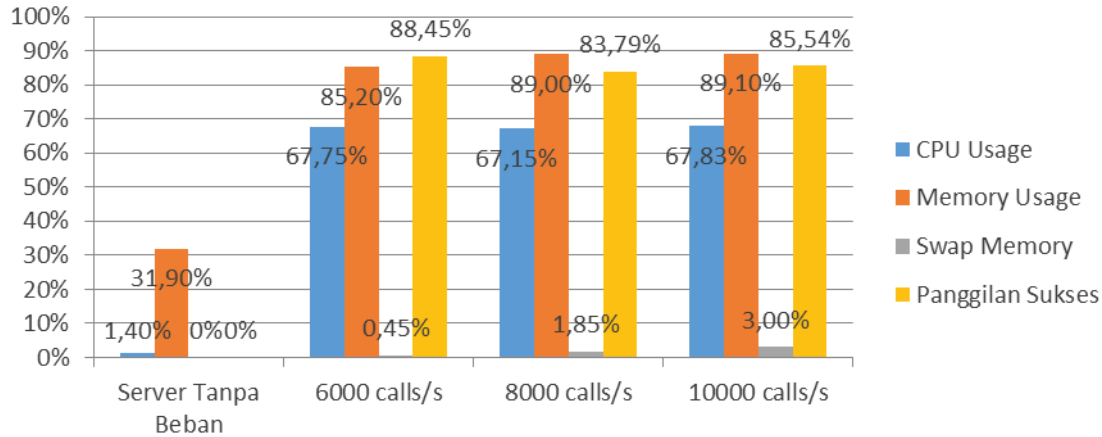
4.2.1 Tujuan Pengukuran

Pengukuran ini bertujuan mengetahui keandalan sistem dalam menangani ribuan panggilan, serta hubungannya dengan CPU *usage* dan *memory usage* dari *instance* ketika dilakukan beban *traffic* panggilan pada sistem.

4.2.2 Hasil Pengukuran



Gambar 6. Hasil Pengukuran dengan *Emulate Call* pada *instance* 2CPUs 1GB



Gambar 7. Hasil Pengukuran dengan *Emulate Call* pada *instance* 3CPUs 1GB

ISSN : 2085-4218

4.2.3 Analisis Pengujian

Berdasarkan hasil pengukuran *emulate call* yang dilakukan yaitu 2000 calls/s, 4000 calls/s dan 6000calls/s, didapatkan bahwa server Mobicents pada instance 2CPUs 1GB mampu menangani ribuan panggilan hingga 4000 calls/s dengan persentase panggilan sukses 99.02%. Dilain hal, pada saat server dengan kondisi beban *traffic* panggilan 6000 calls/s, terjadi *overload* pada CPU *usage*. *Overload* pada CPU menyebabkan meningkatnya persentase panggilan gagal dari seluruh total panggilan yang masuk ke sistem. Persentase panggilan gagal meningkat dari 0.98% menjadi 9.33%.

Pada *instance* 3CPUs 1GB, saat server diberi beban *traffic* panggilan 6000 calls/s, terjadi penggunaan *swap memory* yang dimana *swap memory* sendiri merupakan penggunaan kapasitas *hardisk* untuk dijadikan memory virtual karena beban kerja RAM yang sangat berat. Besarnya *swap memory* biasanya yaitu dua kali besar RAM. *Memory usage* pada kondisi server dengan beban *traffic* panggilan 8000 calls/s dan 10000 calls/s juga mengalami hal yang sama. Namun, berbeda dengan *memory usage*, CPU *usage* berada dalam kondisi stabil dengan rata-rata 67.58%. Persentase panggilan sukses juga mengalami penurunan jika dibandingkan dengan skenario pada *instance* 2CPUs 1GB, hal ini bisa disebabkan karena dua hal yaitu naiknya jumlah panggilan simultan dan panggilan total yang masuk ke sistem serta *memory usage* yang mendekati ataupun melewati batas *threshold memory usage* server Mobicents.

5. Kesimpulan

Berdasarkan hasil pengujian, implementasi dan analisis, maka dapat diambil kesimpulan sebagai berikut:

1. Hasil pengujian *Quality of Service* yang diperoleh memenuhi standar yang ditetapkan dan tergolong sebagai kualitas layanan yang sangat baik. Namun juga perlu dilakukan pengujian seperti kemanan jaringan, performansi server apabila terjadi *failover* dan juga kualitas layanan lainnya seperti *file transfer*.
2. Dari keseluruhan hasil pengukuran dan analisis, implementasi server aplikasi Mobicents layak untuk diimplementasikan pada infrastruktur *cloud*. Hal ini dapat dilihat dari keandalan sistem yang mampu menangani hingga ribuan panggilan yang masuk ke dalam sistem. Implementasi server pada infrastruktur *cloud* juga memberikan fleksibelitas seperti pengaturan jumlah *core* CPU dan RAM yang akan digunakan oleh server WebRTC tanpa harus melakukan reinstall sistem operasi.

6. Daftar Pustaka

[1] Muhammad Iqbal C. R., Muchammad Husni, and Hudan Studiawan, "Implementasi Klien SIP Berbasis Web Menggunakan HTML5 dan Node.js," *JURNAL TEKNIK ITS ISSN: 2301-9271*, vol. 1, 2012.
[2] Rendy Munadi, *Teknik Switching*. Bandung: INFORMATIKA, 2011.
[3] Bruce Hartpence, *Packet Guide to Voice over IP*. Sebastopol: O'Reilly Media, 2013.
[4] Rob Manson, *Getting Started with WebRTC*. Birmingham: Packt Publishing, 2013.
[5] Vanessa Wang, Frank Salim, and Peter Moskovits, *The Defenitive Guide to HTML5 Websocket*. California: Apress, 2013.
[6] Manish Giri, Sachin Waghmare, Balaji Bandhu, Akshay Sawwashere, and Atul Khaire, "Migration of Mobicents SIP Servlets on Cloud Platform," *International Journal of Scientific and Research Publications ISSN 2250-3153*, vol. 2, no. 6, p. 1, 2012.
[7] Mobicents. [Online]. HYPERLINK "http://www.mobicents.org/products.html" http://www.mobicents.org/products.html
[8] Onno W. Purbo, *Petunjuk Praktis Cloud Computing Mengunakan Open Source*., 2011.
[9] Jose Ignacio Moreno Novella and Francisco Javier Gonzalez Castano, "QoS Requirement for Multimedia Services," *Resource Management in Satelite Networks*, 2007.
[10] Cisco, "Quality of Service for Voice over IP," *White Paper*, 2001.