

IMPLEMENTASI DAN ANALISIS PERFORMANSI SERVER APLIKASI MOBICENTS DENGAN WEBRTC IMPLEMENTATION AND PERFORMANCE ANALYSIS OF MOBICENTS APPLICATION SERVER WITH SUPPORT WEBRTC

Ahmad Abdillah Siddiq¹, Dr. Ir. Rendy Munadi, M.T.², Ratna Mayasari, S.T., M.T.³

^{1,2,3}Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

¹siddiq.abdhy@gmail.com, ²rendymunadi@telkomuniversity.ac.id, ³ratnamayasari@telkomuniversity.ac.id

Abstrak

Teknologi telekomunikasi semakin berkembang pesat. Perkembangan pesat ini ditandai dengan lahirnya layanan komunikasi yang semakin beragam. Layanan yang beragam ini sangat dibutuhkan oleh user. Layanan komunikasi yang dibutuhkan sudah sampai pada layanan suara dan *video*. Selain keberagaman, tentulah juga layanan diharapkan memiliki kualitas sesuai standar dan dengan biaya yang murah. Layanan komunikasi suara dan *video* juga harus bersifat *realtime* dan *reliable*. Untuk memenuhi layanan komunikasi yang diinginkan tersebut, maka dari itu layanan perlu dikembangkan pada jaringan berbasis IP (*Internet Protocol*), yang kemudian layanan ini akan dilihat kualitasnya terhadap beberapa parameter.

Dalam penelitian ini, mengimplementasikan sever aplikasi Mobicents yang mendukung layanan WebRTC. *Server Mobicents* sebagai *server* aplikasi yang berdiri sendiri, mampu mendukung layanan WebRTC. Komunikasi yang dilakukan memanfaatkan *browser* sebagai medianya. Kemudian dalam implementasinya, akan dilakukan pengiriman suara dan *video* antar *client*. Dari hasil tersebut, akan dianalisis performansi dari layanan WebRTC tersebut.

Pada penelitian ini dilihat performansinya meliputi parameter QoS yaitu *delay*, *jitter*, *throughput*, dan *packet loss* dan parameter uji server seperti *CPU usage* dan *memory usage*. Dari hasil pengukuran didapat bahwa dengan variasi *background traffic* dari 0 Mbps sampai 80 Mbps yang diberikan, nilai rata-rata *delay* sebesar 5,3753254 ms, *jitter* sebesar 0,4697894 ms, *throughput* sebesar 275.567,2277 Bps dan *packet loss* sebesar 0,020305 %. Hasil yang didapat masih memenuhi standar ITU-T. Kemudian, hasil *CPU usage* tertinggi adalah 30,7 % pada saat proses *start server* dan hasil *memory usage* meningkat dari 17,88 % ke 30,7 %. Pada penelitian ini, juga dilakukan implementasi pada jaringan berbeda. Nilai QoS pada jaringan *wired* dan jaringan *wireless* tetap memenuhi standar ITU-T.

Kata Kunci: WebRTC, *Internet Protocol*, Mobicents, QoS

Abstract

Telecommunication technology is growing rapidly. This rapid development was followed by diverse communications services. These diverse services are needed by the user. Required communication services has reached the voice and video services. In addition to diversity, surely also expected to have the quality of service according to the standard and have low cost. Voice and video communication services must also be real-time and reliable. To meet the desired communication services, so that's why need to be developed in the network services based on IP (Internet Protocol), then the service will be evaluated the quality of some parameters.

In this research, implementing Mobicents application servers that support WebRTC. Mobicents server as a stand-alone application server can support WebRTC services. Communication is done utilizing the browser as a medium. Later in its implementation, will be the delivery of voice and video between clients. From these results, it will be analyzed the performance of the WebRTC services.

In this research seen its performance QoS parameters that include delay, jitter, throughput and packet loss and parameters for server performance such as CPU usage and memory usage. From the measurement results obtained that the background traffic variations from 0 Mbps to 80 Mbps given, the average value of delay is 5.3753254 ms, jitter is 0.4697894 ms, throughput is 275,567.2277 Bps and packet loss is 0.020305%. The results still meet the standard of ITU-T. The highest result of CPU usage is 30.7 % when start server process and the results of memory usage increase from 17.88 % to 30.7 %. In this research, also implemented on different networks. QoS value in the wired network still meet the standard of ITU-T.

Keyword: WebRTC, *Internet Protocol*, Mobicents, QoS

1. Pendahuluan

Seiring majunya peradaban dunia, maka semakin maju juga aspek dalam kehidupan. Salah satu aspeknya, yaitu teknologi komunikasi dan informasi yang kini menjadi bagian penting dalam kehidupan manusia terutama dalam kehidupan metropolitan. Teknologi komunikasi dan informasi sangat dibutuhkan di bidang ilmu pengetahuan, bisnis, maupun ekonomi.

Kemajuan teknologi kini sudah berada di era digital, berbagai layanan berbasis teknologi terus diciptakan. Pengguna pun semakin kreatif dalam hal kebutuhan dan kepuasan akan teknologi komunikasi dan informasi. Teknologi komunikasi yang selama ini sangat bergantung pada *operator* telekomunikasi perlahan memiliki pilihan lain. Sekarang komunikasi sudah bisa dilakukan via *web* baik *voice*, *video* maupun *data*. Dimana komunikasi *real-time communication* (RTC) dapat dilakukan dalam *web* antar *browser* seperti Chrome, Firefox, dan Opera yang telah mendukung fitur tersebut.

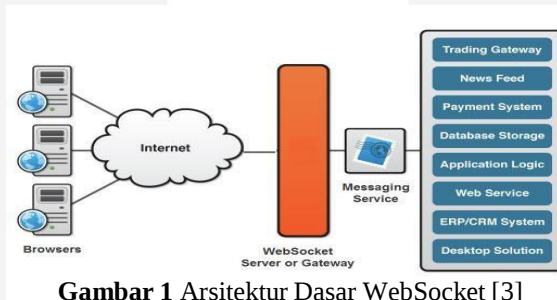
Perkembangan yang pesat ini jelas harus didukung oleh infrastruktur jaringan yang sangat kuat, baik itu kebutuhan *bandwidth* untuk pengiriman suara dan gambar maupun kehandalan sistem dalam melakukan pelayanannya. Perkembangan teknologi jelas harus didukung oleh perkembangan kualitas yang baik juga. Untuk itulah diharapkan dengan menggunakan *server* aplikasi *Mobicents*, dapat melayani layanan yang diminta *client* dengan performansi lebih baik yaitu mampu mengurangi waktu yang dibutuhkan untuk eksekusi layanan tersebut (Tsietsi;Terzoli;Wells, 2006). Aplikasi *Mobicents* dikenalkan untuk menjadi *application server* di era *Next Generation Intelligent Network* (NGIN).

Pada penelitian ini, penulis merancang jaringan *webRTC* menggunakan *Mobicents* sebagai *server* aplikasinya yang berdiri sendiri. Parameter yang akan diukur adalah parameter *Quality of Service* (QoS). Dan parameter uji *server*. Diharapkan dengan menggunakan *Mobicents* sebagai *server* aplikasinya, hasil analisa QoS yang didapatkan memenuhi rekomendasi dari ITU-T.

2. Dasar Teori

2.1 HTML5 WebSocket [23]

WebSocket sebagai bagian pengembangan dari *HTML5 initiative*, yaitu memperkenalkan *Interface WebSocket JavaScript*, yang mendefinisikan satu koneksi *socket full-duplex* dimana pesan dapat dikirim antara klien dan *server*. Standar *WebSocket* menyederhanakan banyak kompleksitas sekitar komunikasi *web bi-directional* dan manajemen koneksi.



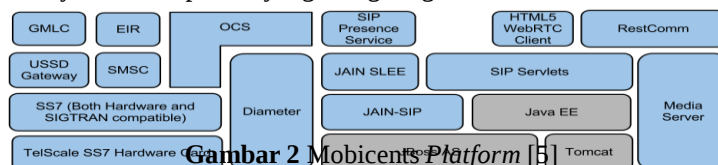
Gambar 1 Arsitektur Dasar WebSocket [3]

2.2 WebRTC [7]

WebRTC adalah sekumpulan standar dari *World Wide Web Consortium* (WC3) dan *Internet Engineering Task Force* (IETF) yang dimulai sejak awal 2011, dimana ia dapat melakukan *real-time communication* (RTC) dalam *web* antar browser yang telah mendukung fitur tersebut.

2.3 Mobicents [5]

Mobicents adalah sebuah *Open source platform* VoIP yang ditulis di *Java* untuk membantu menciptakan, menyebarkan, mengelola layanan dan aplikasi yang mengintegrasikan suara, *video* dan data pada jaringan IP.



Gambar 2 Mobicents Platform [5]

2.4 Quality of Service (QoS) [18].

2.4.1 Delay

Secara teknis, *delay* merupakan banyaknya waktu yang diperlukan sebuah paket untuk melakukan perjalanan dari sumber ke tujuan. Bersama dengan *bandwidth*, *delay* mendefinisikan kecepatan dan kapasitas dalam jaringan. *Delay* merupakan masalah biasa dalam slow speed links.

2.4.2 Jitter

Jitter dapat didefinisikan sebagai variasi kedatangan paket di sisi penerima (*delay*). *Jitter* merupakan masalah khas dari connectionless network atau packet switched network serta slow speed links. Komunikasi real time, contohnya seperti VoIP, biasanya memiliki masalah kualitas dari efek ini. Besarnya *jitter* diantara titik awal dan akhir komunikasi seharusnya kurang dari 150 ms. Sedangkan, besar *jitter* untuk wireless kurang dari 5 ms (ITU G.107).

2.4.3 Packet loss

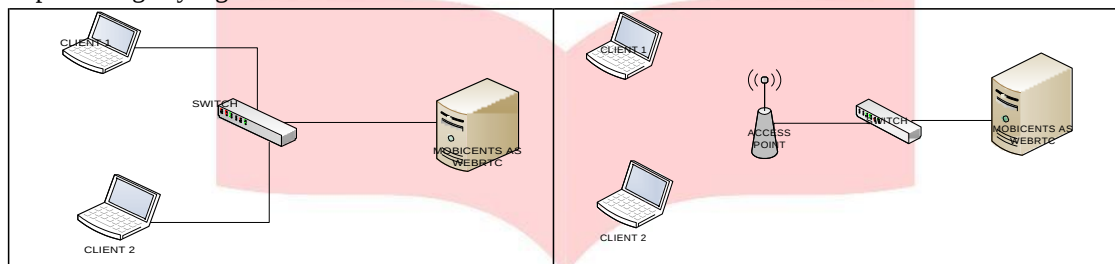
Packet loss merupakan banyaknya paket yang gagal mencapai tempat tujuan saat pengiriman paket. Protokol UDP mendasari komunikasi real time, dimana protokol ini bersifat *connectionless*.

2.4.4 Throughput

Throughput dalam jaringan telekomunikasi merupakan rata-rata pengiriman sukses dalam suatu pengiriman (satuan bps). Sedangkan, sistem throughput atau jumlah throughput merupakan jumlah rata-rata packet data yang sukses dikirimkan oleh semua terminal pada sebuah jaringan.

3. Perancangan dan Implementasi

Dalam proses perancangan sebuah sistem, diperlukan sebuah skenario yang terstruktur dengan baik. Untuk memudahkan proses perancangan implementasi diperlukan topologi yang membantu dalam memahami proses perancangan yang akan dibuat.



Gambar 3 Topologi Jaringan

Gambar 3 merupakan gambar konfigurasi jaringan dalam tugas akhir ini, akan dibangun konfigurasi jaringan untuk implementasi layanan WebRTC. Server yang digunakan adalah server aplikasi Mobicents. Server Mobicents akan menyediakan layanan WebRTC pada paket WebSocket. Pada implementasi, untuk *client* menggunakan dua buah PC yang semuanya sudah terhubung ke jaringan yang sama dengan PC server. Kedua *client* akan masuk pada sistem Mobicents *via browser* dengan memasukkan IP server.

4. Pengujian dan Analisis

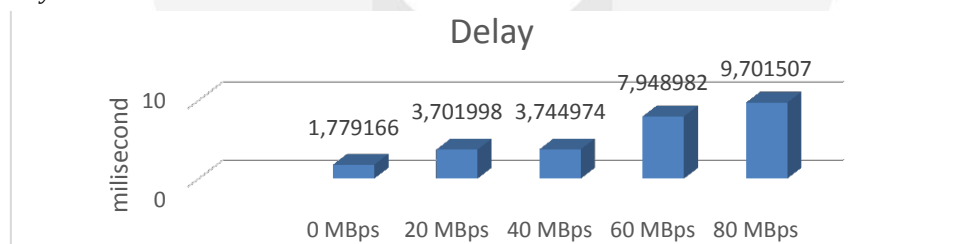
4.1 Pengujian Quality of Service (QoS)

4.1.1 Sistematika Pengukuran

1. Pengukuran performansi server aplikasi Mobicents dengan *background traffic*. Variasi *background traffic* yang dipakai adalah sebesar 0 MBps, 20 MBps, 40 MBps, 60 MBps dan 80.
2. Pengukuran performansi server aplikasi Mobicents dengan media *wired* dan *wireless*.

4.1.2 Hasil Pengukuran dengan Background Traffic

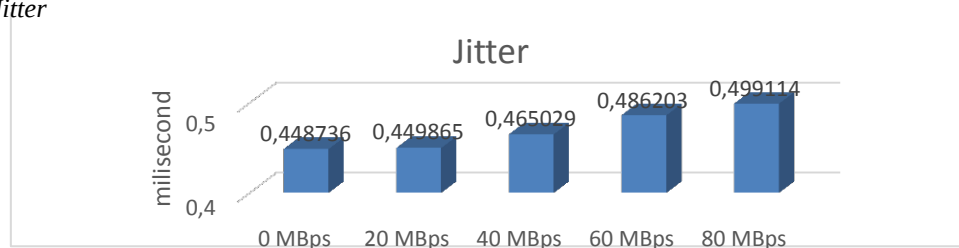
a. Delay



Gambar 4 Grafik Delay pada Skenario Pertama

Dari hasil pengukuran bahwa besar nilai *delay* sebanding dengan besarnya *background traffic* yang digunakan. Hal ini dikarenakan dengan semakin besar nilai *background traffic* yang diberikan maka lalu lintas dari pengirim ke penerima semakin padat, maka proses pengiriman data pun akan memakan waktu yang lebih lama, sesuai dengan konsepnya bahwa *delay* adalah waktu yang dibutuhkan saat proses pengiriman paket dari pengirim dan penerima. Nilai *delay* tertinggi yaitu 9,701507 ms dengan *background traffic* 80 Mbps yang besarnya mendekati kapasitas maksimal. Dari seluruh hasil perhitungan *delay* menunjukkan bahwa nilai *delay* layanan WebRTC pada server aplikasi Mobicents masih memenuhi standar ITU-T yaitu sebesar 150 ms.

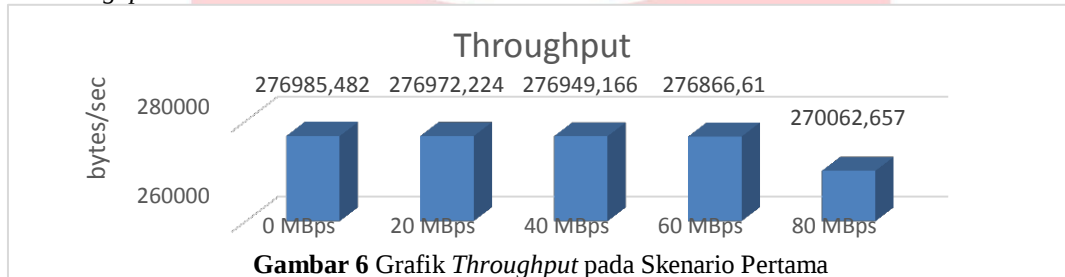
b. Jitter



Gambar 5 Grafik Jitter pada Skenario Pertama

Dari hasil pengukuran, bahwa besar *jitter* sebanding dengan besarnya *background traffic* yang digunakan. Hal ini terjadi karena kepadatan trafik yang telah ditambahkan *background traffic*, semakin besar nilai *background traffic* maka sistem semakin tidak stabil dalam pengiriman paket dan menyebabkan variasi kedatangan antar paket semakin beragam. Nilai *jitter* tertinggi yaitu 0,499114 ms dengan *background traffic* 80 Mbps, penambahan *background traffic* yang mendekati batas maksimal membuat laju kedatangan paket semakin lama dan bervariasi. Dari seluruh hasil perhitungan *jitter* menunjukkan bahwa nilai *jitter* layanan WebRTC pada server aplikasi Mobicents dengan berbagai variasi *background traffic* memenuhi standar ITU-T G.1010 yaitu sebesar 1 ms.

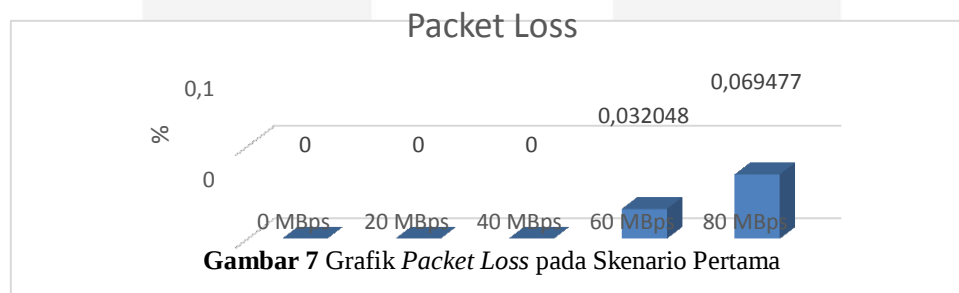
c. *Throughput*.



Gambar 6 Grafik *Throughput* pada Skenario Pertama

Dari hasil pengukuran bahwa besar nilai *throughput* berbanding terbalik dengan besarnya *background traffic* yang digunakan, hal ini sesuai dengan konsep *throughput* sebagai rata-rata kesuksesan paket terkirim dalam satuan waktu, dikarenakan semakin padatnya trafik saat pengiriman mengurangi keberhasilan pengiriman paket. Nilai *throughput* tertinggi yaitu 276985,482 Bps dengan *background traffic* 0 Mbps, karena trafik dari pengirim ke penerima yang kosong tanpa penambahan *background traffic*. Semua hasil *throughput* tidak lebih besar daripada nilai *rate* kirim yaitu 313.750 Bps dan semua hasil *throughput* juga berada diantara nilai *throughput* minimum (BW min) yaitu 63.250 Bps dan *throughput* maximal (BW max) yaitu 313.750 Bps, jadi semua hasil nilai *throughput* masih bernilai baik.

d. *Packet Loss*

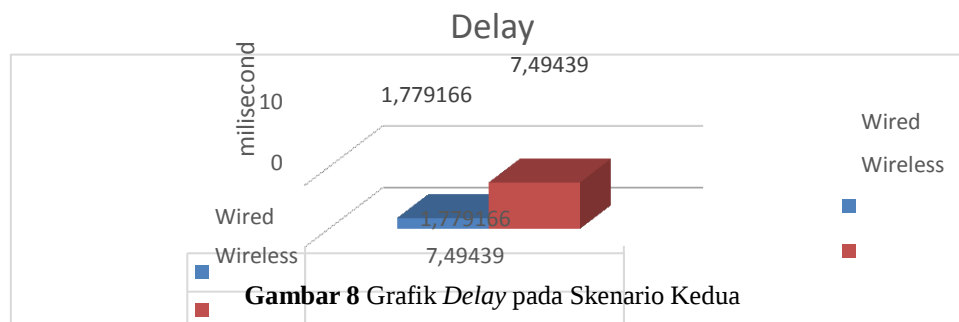


Gambar 7 Grafik *Packet Loss* pada Skenario Pertama

Dari hasil pengukuran bahwa besar nilai *packet loss* sebanding dengan besarnya *background traffic* yang digunakan, karena semakin padat trafik dari pengirim ke penerima membuat pengiriman paket semakin banyak kemungkinan paket hilang. Nilai *packet loss* tertinggi yaitu 0,069477 % dengan *background traffic* 80 Mbps. Dari seluruh hasil perhitungan *packet loss* menunjukkan bahwa nilai *packet loss* layanan WebRTC pada server aplikasi Mobicents masih memenuhi standar ITU-T yaitu sebesar 1 %.

4.1.3 Hasil Pengukuran Media Wired dan Wireless

a. *Delay*

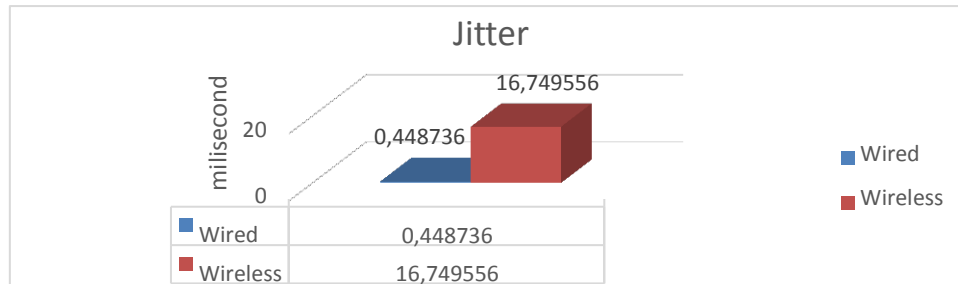


Gambar 8 Grafik *Delay* pada Skenario Kedua

Dari gambar dapat terlihat bahwa hasil nilai *delay* pada jaringan *wired* lebih kecil dibanding jaringan *wireless*. Hal ini menunjukkan bahwa perbedaan media memiliki pengaruh terhadap kualitas dan kecepatan sistem dalam pengiriman paket ke penerima, hal ini dikarenakan kondisi jarak dan lingkungan pada jaringan *wireless* dengan penjelasan lengkap di subbab 3.2.1. Nilai *delay* terbesar adalah 7,49439 ms dengan media

wireless, sedangkan nilai *delay* pada jaringan *wired* sebesar 1,779166 ms. Hasil perhitungan *delay* pada kedua jaringan bernilai baik dan masih dalam standar ITU-T sebesar 150 ms.

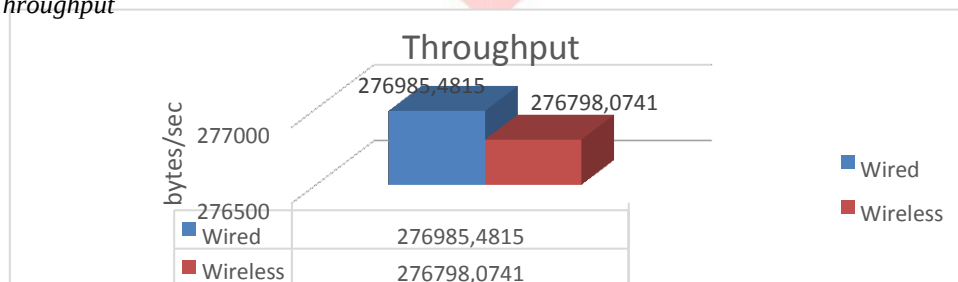
b. *Jitter*



Gambar 9 Grafik *Jitter* pada Skenario Kedua

Dari gambar dapat terlihat bahwa hasil *jitter* pada jaringan *wired* lebih kecil dibanding jaringan *wireless*. Hal ini menunjukkan bahwa perbedaan media memiliki pengaruh terhadap banyaknya variasi kedatangan paket ataupun kestabilan jaringan dalam mengirim paket dari pengirim ke penerima. Nilai *jitter* terbesar adalah 16,749556 ms dengan media *wireless*, sedangkan nilai *jitter* pada jaringan *wired* sebesar 0,448736 ms. Hal ini dikarenakan kondisi jarak dan lingkungan pada jaringan *wireless* dengan penjelasan lengkap di subbab 3.2.1.

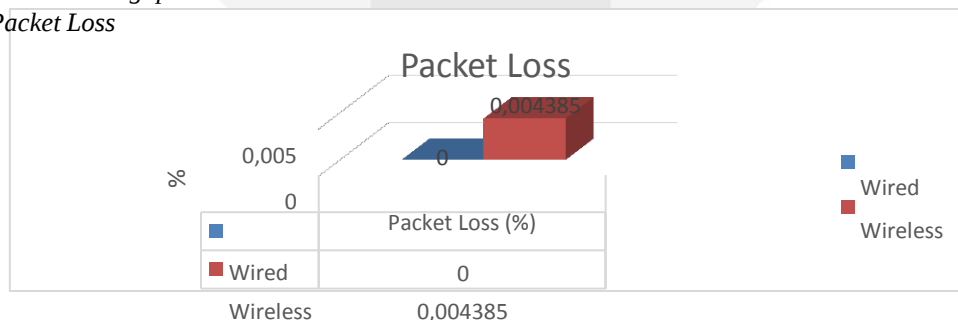
c. *Throughput*



Gambar 10 Grafik *Throughput* pada Skenario Kedua

Dari gambar dapat terlihat bahwa hasil *throughput* pada jaringan *wired* lebih besar dibanding jaringan *wireless*. Hal ini menunjukkan bahwa perbedaan media memiliki pengaruh terhadap kualitas dan keberhasilan sistem dalam pengiriman paket ke penerima, hal ini dikarenakan kondisi jarak dan lingkungan pada jaringan *wireless* dengan penjelasan lengkap di subbab 3.2.1. Nilai *throughput* terbesar adalah 276985,4815 Bps dengan media *wired*, sedangkan *throughput* pada jaringan *wireless* adalah 276798,0741 Bps. Semua hasil *throughput* tidak lebih besar daripada nilai *rate* kirim yaitu 313.750 Bps dan semua hasil *throughput* juga berada diantara nilai *throughput* minimum (BW min) yaitu 63.250 Bps dan *throughput* maximal (BW max) yaitu 313.750 Bps, jadi semua hasil nilai *throughput* masih bernilai baik.

e. *Packet Loss*



Gambar 11 Grafik *Packet Loss* pada Skenario Kedua

Dari gambar dapat terlihat bahwa hasil *packet loss* pada jaringan *wired* lebih kecil dibanding jaringan *wireless*. Hal ini menunjukkan bahwa perbedaan media memiliki pengaruh terhadap kualitas dalam pengiriman paket ke penerima. Nilai *packet loss* terbesar adalah 0,004385 % dengan media *wireless* dan pada jaringan *wired* bernilai 0 %. Hal ini dikarenakan pada jaringan *wired* dengan trafik kosong tanpa beban trafik membuat tidak adanya paket yang hilang saat pengiriman, sedangkan pada *wireless* adanya pengaruh kondisi jarak dan lingkungan dengan penjelasan lengkap di subbab 3.2.1. Hasil perhitungan *packet loss* pada kedua jaringan bernilai baik dan masih dalam standar ITU-T sebesar 1 %.

4.2 Pengukuran CPU Usage

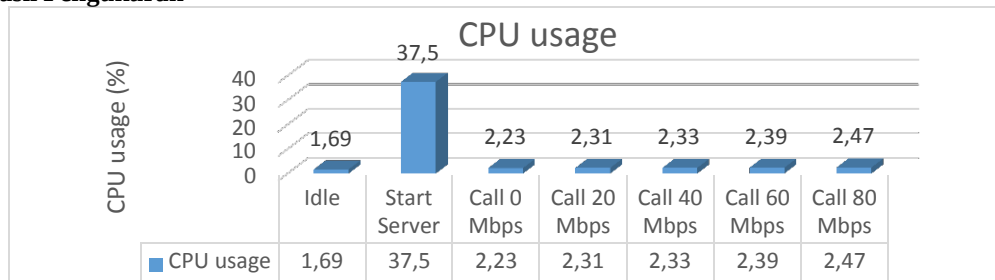
4.2.1 Tujuan Pengukuran

Tujuan pengukuran adalah untuk mengetahui bagaimana proses digunakan server aplikasi Mobicents dengan layana webRTC, dan bagaimana pengaruhnya terhadap penggunaan CPU.

4.2.2 Sistematika Pengukuran

CPU usage diukur pada saat server belum dijalankan (*idle*), saat server dijalankan (*start server*), dan saat komunikasi dijalankan melibatkan *background traffic* sebesar 0 Mbps, 20 Mbps, 40 Mbps, 60 Mbps, 80 Mbps.

4.2.3 Hasil Pengukuran



Gambar 12 Grafik CPU Usage

4.2.4 Analisa Hasil Pengukuran CPU Usage

Dapat dilihat dari Gambar 12 bahwa semakin besar nilai *bandwidth* yang diberikan, maka CPU usage akan semakin besar juga. Hal ini dikarenakan semakin besar nilai *background traffic* maka lebar *bandwidth* yang tersedia untuk paket webRTC akan semakin sempit sehingga mempengaruhi kinerja server yaitu nilai CPU usage. Dari skenario diatas dapat dikatakan bahwa nilai CPU usage yang tertinggi adalah ketika server mulai dijalankan (*start server*), yaitu 37.5%. Bisa disimpulkan berarti server paling berat bekerja adalah saat server mulai dijalankan. Saat dilakukan panggilan (*call*) nilai CPU usage cenderung kecil, walaupun terjadi peningkatan nilai CPU usage namun peningkatan tersebut tidak signifikan.

4.3 Pengukuran Memory Usage

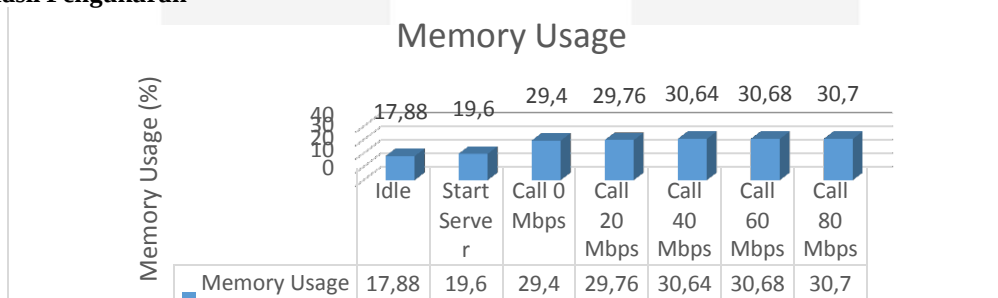
4.3.1 Tujuan Pengukuran

Tujuan pengukuran adalah untuk mengetahui bagaimana proses digunakan server aplikasi Mobicents dengan layana webRTC, dan bagaimana pengaruhnya terhadap penggunaan Memory.

4.3.2 Sistematika Pengukuran

Memory usage diukur pada saat server belum dijalankan (*idle*), saat server dijalankan (*start server*), dan saat komunikasi dijalankan melibatkan *background traffic* sebesar 0 Mbps, 20 Mbps, 40 Mbps, 60 Mbps, 80 Mbps.

4.3.3 Hasil Pengukuran



Gambar 13 Grafik Memory Usage

4.3.4 Analisa Hasil Pengukuran Memory Usage

Dapat dilihat dari Gambar 13 bahwa semakin besar nilai *bandwidth* yang diberikan, maka memory usage akan semakin besar juga. Hal ini dikarenakan semakin besar nilai *background traffic* maka lebar *bandwidth* yang tersedia untuk paket webRTC akan semakin sempit sehingga mempengaruhi kinerja server yaitu nilai memory usage. Dari skenario diatas dapat dikatakan bahwa terjadi peningkatan nilai memory usage dari posisi idle sampai proses panggilan dengan *background traffic* 80 Mbps, yaitu 17.88% ke 30,7 %. Semakin besar nilai memory usage berarti memori yang digunakan PC server dalam mengolah proses yang terjadi di server semakin besar juga. Namun nilai yang didapat terhitung masih dapat di tolerir karena tidak terlalu memberatkan kerja PC server atau tidak melebihi 50%.

4.4 Hasil Pengukuran Mean Opinion Score (MOS)

4.4.1 MOS dengan E-Model

Tabel 1 Hasil Perhitungan MOS E-Model

Perhitungan MOS berdasar R Faktor									
	Delay	d-177,3	H (x)	Id	Packet Loss	30 ln (1+15e)	Ief	R faktor	MOS
Wired 0 Mbps	1,779166	-175,52	0	0,0427	0	0	7	87,1573	4,07492
Wired 20 Mbps	3,701998	-173,6	0	0,08885	0	0	7	87,1112	4,07335
Wired 40 Mbps	3,744974	-173,56	0	0,08988	0	0	7	87,1101	4,07332
Wired 60 Mbps	7,948982	-169,35	0	0,19078	0,032048	11,7758537	18,7759	75,2334	3,65958
Wired 80 Mbps	9,701507	-167,6	0	0,23284	0,069477	21,4201687	28,4202	65,547	3,30752
Wireless	7,49439	-169,81	0	0,17987	0,004385	1,91106702	8,91107	85,1091	4,00499

Dari tabel 1 kita dapat mengetahui bahwa beberapa aspek yang dinilai diatas telah cukup memenuhi standart kualitas yang diharapkan. Pada jaringan wired dengan backgorund traffic 0 Mbps hingga 40 Mbps bernilai lebih dari 4.00 yang berstandar baik, sedangkan pada jaringan wired dengan backgorund traffic 60 Mbps hingga 80 Mbps bernilai lebih dari 3.00 yang berstandar cukup. Pada jaringan wireless bernilai lebih dari 4.00 yang berarti baik. Semua hasil perhitungan MOS ini berdasarkan standar ITU-T P.800 dan ITU-T G.107.

4.4.2 MOS dengan User Experience

Mean Opinion Score adalah metode berdasar standar ITU-T P.800, yaitu penilaian kualitas layanan dalam jaringan IP yang bersifat subjektif, karena berdasar penilaian dari pengguna layanan atau berdasar *user experience*. Pada akhirnya akan diketahui kelebihan dan kekurangan dari sistem tersebut. Aspek yang dianalisa adalah:

1. Media WebRTC sebagai Interface User

- Tampilan *interface user* pada Web browser.
- Kemudahan menggunakan *interface user* untuk mengakses layanan webRTC.

2. Performansi Sistem

- Kualitas Audio.
- Kualitas Video.
- Kesesuaian Lip Synchronization.
- Efektifitas Layanan.
- Kualitas Secara Keseluruhan.

Tabel 2 Nilai Aspek Keseluruhan

Aspek Penilaian	Nilai					Total	Nilai Rata-rata
	1XN	2XN	3XN	4XN	5XN		
	Buruk	Kurang	Cukup	Baik	Sangat Baik		
Tampilan Media Softphone	0	8	63	20	0	91	3.03
Kemudahan Menggunakan	0	4	39	60	0	103	3.43
Kualitas Audio	0	8	51	36	0	95	3.17
Kualitas Video	0	6	60	28	0	94	3.13
Kesesuaian Lip Synchronization	0	12	51	28	0	91	3,03
Efektifitas Layanan	0	4	45	52	0	101	3,37
Kualitas Secara Keseluruhan	0	4	54	40	0	98	3.27
Rata-rata Total							3.20

Dari tabel 2 kita dapat mengetahui bahwa beberapa aspek yang dinilai diatas telah cukup memenuhi standart kualitas yang diharapkan. Semua komponen bernilai lebih dari 3.00 dan rata-rata dari keseluruhan nilai masing-masing aspek adalah 3.20 yang berarti “cukup” menurut standart ITU-T P.800 untuk MOS.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan dari hasil proses implementasi, pengujian, dan analisis maka dapat ditarik kesimpulan berikut.

1. Implementasi *server* aplikasi Mobicents dengan WebRTC berhasil dilakukan.
2. Hasil pengukuran QoS menunjukkan bahwa *server* aplikasi Mobicents dengan layanan webRTC yang dilakukan dengan seluruh variasi *background* trafik yang diberikan, yaitu rata-rata *delay* = 5.3753254 < 150 ms, *jitter* 0.4697894 ms, *packet loss* = 0,020305 < 1% dan *throughput* = 275567.2277 Bps masih memenuhi standar “baik” (Standar ITU-T).
3. Hasil pengukuran QoS menunjukkan bahwa *server* aplikasi Mobicents dengan layanan webRTC pada jaringan wired dan jaringan wireless, yaitu rata-rata *delay* meningkat dari 1.779166 ms pada jaringan wired menjadi 7.49439 ms pada jaringan wireless, *jitter* 0.448736 ms menjadi 16.749556 ms, *throughput*

menurun dari 276985.4815 Bps menjadi 276798.0741 Bps dan packet loss meningkat dari 0 % menjadi 0.004385 % masih memenuhi standar “baik” (Standar ITU-T).

4. Hasil pengukuran performansi *server* dengan parameter *CPU usage*, yaitu nilai tertinggi yaitu saat *start server* yaitu 37.5 %. Serta terjadi peningkatan saat terjadi *call* saat diberikan *background traffic* dari 20 Mbps ke 80 Mbps yaitu 2.23 % ke 2.47 %. Maka dari itu dapat dikatakan spesifikasi *PC server* cukup baik untuk menjalankan *server* Mobicents.
5. Hasil pengukuran performansi *server* dengan parameter *memory usage* didapatkan nilai meningkat dari posisi *idle* ke posisi *call* dengan *background traffic* 80 Mbps yaitu 17.88 % ke 30.7 %. Dengan nilai tersebut server aplikasi Mobicents dapat dikatakan stabil.
6. Hasil perhitungan metode Mean Opinion Score dengan E-Model didapat hasil, rata-rata MOS sebesar 3,83773 yang berarti berstandar “cukup” untuk jaringan wired dan 4.00499 untuk jaringan wireless yang berarti berstandar “baik” berdasarkan rekomendasi ITU-T P.800 dan ITU-T G.107.
7. Hasil perhitungan metode Mean Opinion Score dengan *user experience* pengukuran kualitas sistem layanan *video conference* dengan menggunakan metode Mean Opinion Score (MOS), didapatkan nilai rata-rata dari setiap aspek penilaian sebesar 3.20 itu “cukup” berdasarkan rekomendasi ITU-T P.800
8. Layanan webRTC pada server aplikasi Mobicents layak untuk diimplementasikan.

DAFTAR PUSTAKA

- [1] (n.d.). (Mobicents) Retrieved 2014, from <http://www.Mobicents.org>
- [2] (2011). (WEBRTC) Retrieved 05 14, 2015, from <http://www.webrtc.org>
- [3] About HTML5 WebSockets. (2015, June). Retrieved from WebSocket.org: <https://www.websocket.org/aboutwebsocket.html>
- [4] A. T. a. G. W. Mosiuoa Tsietsi, "Mobicents as a Service Creation and Deployment," Mobicents as a Service Creation and Deployment Environment for the Open IMS Core.
- [5] B. I. Priaksa, IMPLEMENTASI DAN ANALISIS PERFORMANSI MOBICENTS SEBAGAI SERVER APLIKASI PADA ARSITEKTUR IP MULTIMEDIA SUBSISTEM UNTUK LAYANAN VIDEO CONFERENCE, Bandung, 2014.
- [6] Deruelle, J. (2015, May). Mobicents SIP Servlets 2.0.0.Final with support for HTML5 WebRTC is out ! Retrieved from www.telestax.com: <http://www.telestax.com/Mobicents-sip-servlets-2-0-0-final-with-support-for-html5-webrtc-is-out/>
- [7] "Era Baru Layanan Telekomunikasi," Manajemen Telekomunikasi, 19 11 2014. [Online]. Available: <http://www.manajementelekomunikasi.org/2014/11/webrtc-dan-rcs.html>. [Accessed 16 05 2015].
- [8] Grigorik, I. (2013). WebRTC. In I. Grigorik, *High Performance Browser Networking*. O'Reilly Media. Retrieved from <http://chimera.labs.oreilly.com/books/1230000000545/ch18.html>
- [9] Hinchewa, D. (2015, May). *Live Chat and Video Call with Restcomm*. Retrieved from www.telestax.com: <http://www.telestax.com/livechat-and-video-call-with-restcomm/>
- [10] HTML5 Web Sockets: A Quantum Leap in Scalability for the Web. (2015, June). Retrieved from [websocket.org](https://www.websocket.org/quantum.html): <https://www.websocket.org/quantum.html>
- [11] "IP Telephony Voice over IP (VoIP) Introduction," cisco, [Online]. Available: <http://www.cisco.com/c/en/us/tech/voice/ip-telephony-voice-over-ip-voip/index.html>. [Accessed 10 June 2015].
- [12] M. Khan, "webrtcpedia!," WebRTC Experiments, [Online]. Available: <https://www.webrtc-experiment.com/webrtcpedia/>. [Accessed June 2015].
- [13] Munadi, R. (2009). *Teknik Switching*. Bandung: Informatika.
- [14] Mobicents - Open Source Cloud Communications. (n.d.). (Mobicents) Retrieved November 2014, from <http://www.youtube.com/watch?v=4eYM-t7EL50>
- [15] Mosiuoa Tsietsi, A. T. (n.d.). Mobicents as a Service Creation and Deployment. *Mobicents as a Service Creation and Deployment Environment for the Open IMS Core*.
- [16] N. Unuth, "Mean Opinion Score (MOS) - A Measure Of Voice Quality," About Tech, 2015. [Online]. Available: <http://voip.about.com/od/voipbasics/a/MOS.htm>. [Accessed 10 June 2015].
- [17] N. Unuth, "VoIP - What is VoIP?," About Tech, 2015. [Online]. Available: <http://voip.about.com/od/voipbasics/a/whatisvoip.htm>. [Accessed 10 June 2015].
- [18] "Quality of Service (QoS)," Cisco, [Online]. Available: <http://www.cisco.com/c/en/us/products/ios-nx-os-software/quality-of-service-qos/index.html>. [Accessed 2014].
- [19] Ray Musvibe, A. T. (n.d.). Developing a Video on Demand Service using Mobicents JSLEE.
- [20] T. Quintana, HTML5 WebRTC and SIP Over WebSockets, TeleStax, Inc, 2013.
- [21] "The E-model," International Telecommunication Union, 2005. [Online]. Available: <https://www.itu.int/ITU-T/studygroups/com12/emodelv1/tut.htm>. [Accessed 10 June 2015].
- [22] Web Standardization. (2015, June). Retrieved from [webrtc.org](http://www.webrtc.org/web-apis/standardization): <http://www.webrtc.org/web-apis/standardization>
- [23] WebRTC Architecture. (2015, June). Retrieved from [webrtc.org](http://www.webrtc.org/architecture): <http://www.webrtc.org/architecture>
- [24] What is WebSocket? (2015, June). Retrieved from www.websocket.org: <https://www.websocket.org/>