

## IMPLEMENTASI DAN ANALISIS PERFORMANSI MOBICENTS SEBAGAI SERVER APLIKASI UNTUK LAYANAN CLICK TO CALL

### IMPLEMENTATION DAN PERFORMANCE ANALYSIS OF MOBICENTS AS APPLICATION SERVER FOR CLICK TO CALL SERVICE

<sup>1</sup>Rizka Darmawan Dwi Putra    <sup>2</sup>Dr. Rendy Munadi, Ir.,MT    <sup>3</sup>Ridha Muldina Negara, ST. MT.

<sup>1 2 3</sup> Fakultas Teknik Elektro, Universitas Telkom, Bandung

<sup>1</sup>[rizkadarmawandp@gmail.com](mailto:rizkadarmawandp@gmail.com), <sup>2</sup>[rnd@telkomuniversity.ac.id](mailto:rnd@telkomuniversity.ac.id), <sup>3</sup>[rmg@telkomuniversity.ac.id](mailto:rmg@telkomuniversity.ac.id)

#### Abstrak

Teknologi komunikasi merupakan salah satu faktor terpenting dalam aspek kehidupan baik di bidang ilmu pengetahuan, bisnis, maupun ekonomi. Salah satu perkembangan teknologi saat ini adalah *Voice over Internet Protocol (VoIP)* dimana kita dapat berkomunikasi melalui media *internet*. Selain itu kita membutuhkan akses informasi yang cepat agar efisien dari segi waktu. Di sisi lain, perkembangan ini menuntut kebutuhan infrastruktur jaringan yang sangat penting baik itu kebutuhan bandwidth untuk pengiriman suara maupun kehandalan sistem dalam melakukan pelayanannya. Komunikasi berbasis *IP* sangat dipengaruhi oleh *delay*, *throughput*, dan parameter lainnya. Disini akan diuji kelayakan dari layanan *click to call* yang dilewatkan pada jaringan berbasis *IP*. Penelitian ini mengimplementasikan layanan *click to call* pada server aplikasi *Mobicents* yang berdiri sendiri. Dari implementasi ini selanjutnya dianalisis *call* pada server dari aspek *quality of service*-nya dengan parameter *delay*, *jitter*, *packet loss* dan *throughput*. Selain itu akan dianalisis dari parameter lainnya yaitu *post dial delay*, *CPU usage* dan *Memory usage*. Analisis dilakukan berdasarkan hasil uji coba proses pengiriman perintah pada sistem untuk selanjutnya melakukan pengiriman suara dari server menuju client dengan client lainnya dan *background traffic*. Pada penelitian ini juga diuji tentang performansi server dengan media *wired* dan *wireless*. Dari pengujian dan analisis diperoleh nilai *one way delay* sebesar 19.99319104 ms, *jitter* 0.307859479 ms, *throughput* 10732,4619 bytes/s dan total PDD 0.277226437 s. Kemudian untuk *background traffic* maksimal sebesar 80 Mbps nilai *delay* 19.9940779 ms, *jitter* 0.502872421 ms, *throughput* 10727.92687 bytes/s dan total PDD 0.309896433 s. Kemudian untuk nilai *CPU usage* dan *Memory usage* tertinggi berturut-turut yaitu 11.34 % dan 29.4 %. Hasil yang diperoleh masih dibawah batas maksimum yang distandarkan ITU-T, Cisco dan IETF, maka disimpulkan sistem ini dapat berfungsi dengan baik. Serta menggunakan media *wired* akan memberikan hasil performansi lebih baik daripada menggunakan media *wireless*.

**Kata Kunci:** *Click to call, SIP Servlets, Mobicents*

#### Abstract

*Communication technology is one of the most important factors in aspects of life in the fields of science, business, and economics. One of the current technological developments is Voice over Internet Protocol (VoIP) where we can communicate via the Internet. In addition we need instant access to get information to be efficient in terms of time. On the other hand, this development requires a network infrastructure needs are very important both bandwidth requirements for the delivery of voice as well as system reliability in doing ministry. IP-based communication is strongly influenced by the delay, throughput, and other parameters. Here will be tested the feasibility of a click to call service are passed to IP-based networks. This final project implements the click to call services on an application server Mobicents that stand-alone. Of this implementation further analyzed call on the server from the aspect of its quality of service parameters delay, jitter, packet loss and throughput. Additionally, it will be analyzed from other parameters, namely the post dial delay, CPU usage and memory usage. The analysis was performed based on the test results the process of sending the next command on the system for sending sound from client to client server towards the other and the background traffic. In this final project also tested on the performance of servers with wired and wireless media. From the testing and analysis of one-way delay values obtained by 19.99319104 ms, jitter 0.307859479 ms, throughput 10732.4619 bytes / s and total PDD 0.277226437 s. Then for background traffic at 80 Mbps maximum value delay 19.9940779 ms, jitter 0.502872421 ms, throughput 10727.92687 bytes / s and total PDD 0.309896433 s. Then for the value of CPU usage and memory usage is highest row 11.34% and 29.4%. The results obtained are still below the maximum limit of standardized ITU-T, Cisco and the IETF, it is concluded the system can function properly. As well as using wired media will result in better performance than using wireless media.*

**Keywords:** *Click to call, SIP Servlets, Mobicents*

#### 1 Pendahuluan

##### 1.1 Latar Belakang

Teknologi komunikasi merupakan salah satu faktor terpenting dalam aspek kehidupan. Salah satu perkembangan teknologi saat ini adalah *Voice over Internet Protocol (VoIP)* dimana kita dapat berkomunikasi melalui media internet [1]. Di sini kita dapat melihat bahwa pelanggan mengharapkan dengan adanya

perkembangan teknologi, kualitas layanan juga ikut meningkat. 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 [2]. Pada penelitian sebelumnya telah terbukti bahwa Mobicents dapat diterapkan pada arsitektur IMS untuk menjalankan layanan *video conference* [3]. Namun penelitian tersebut lebih terfokus pada arsitektur IMS nya, bukan pada server aplikasi Mobicents itu sendiri. Sedangkan Mobicents yang merupakan *platform* VoIP yang dapat menyediakan banyak layanan seharusnya bisa menjadi *server* yang berdiri sendiri tanpa bantuan arsitektur IMS. Maka dari itu penulis mencoba untuk menganalisis performansi dari server aplikasi Mobicents ini untuk dapat berdiri sendiri tanpa arsitektur lain yang mendukungnya, dengan mencoba pada layanan *click to call*. Selain itu, pengguna juga mengharapkan dapat melakukan komunikasi *real time* ini dengan gratis tanpa harus terhubung dengan *internet*, melainkan *intranet*. Diharapkan dengan server aplikasi Mobicents, hasil analisa QoS yang didapatkan memenuhi rekomendasi dari ITU-T.

## 2 Dasar Teori

### 2.1 Click to call

*Click to call*, juga dikenal sebagai *click to talk*, *click to chat*, *click to text*, adalah suatu bentuk komunikasi berbasis web dimana seseorang mengklik suatu objek (misalnya tombol, gambar atau teks) untuk meminta koneksi langsung dengan orang lain secara *real time* baik melalui telepon, *Voice over Internet protocol (VoIP)*, atau teks. Sebuah implementasi dari apa yang kemudian disebut teknologi *click to call* yang awalnya dikenal juga sebagai *auto-dial*. Penamaan tersebut sebenarnya keliru, karena hal ini membuat user dapat menekan suatu tombol dari komputer untuk menelepon suatu nomor yang disediakan oleh web.

### 2.2 Voice over Internet Protocol

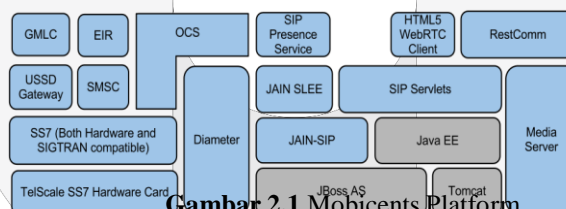
*Voice over Internet Protocol (VoIP)* adalah teknologi yang memungkinkan komunikasi suara, video dan data menggunakan jaringan berbasis IP (internet protokol) untuk dijalankan di atas infrastruktur packet network. Suara kita berbentuk analog agar dapat dilewatkan pada jalur packet Switch dengan baik maka harus dikonversikan ke bentuk digital melalui proses coder- decoder [1].

### 2.3 Application Server

Server Aplikasi adalah aplikasi pada sistem komputer yang berfungsi melayani permintaan akses dari komputer pengguna atau klien.

### 2.4 Mobicents

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 2.1 Mobicents Platform

### 2.5 Apache Tomcat 7

Apache Tomcat merupakan *software open source* implementasi dari teknologi Java Servlet dan Java Server Pages. Spesifikasi Java Servlet dan JSP dikembangkan di bawah Java Community Process. Apache Tomcat dikembangkan di lingkungan terbuka dan partisipatif dan dirilis di bawah Apache License versi 2. Apache Tomcat dimaksudkan untuk menjadi sebuah kolaborasi yang terbaik-of-breed pengembang dari seluruh dunia Apache Tomcat 7.0 adalah titik masuk top-level dari bundel dokumentasi untuk kontainer Apache Tomcat Servlet / JSP. Apache Tomcat versi 7.0 mengimplementasikan Servlet 3.0 dan JSP 2.2 spesifikasi dari Java Community Process, dan mencakup banyak fitur tambahan yang membuatnya menjadi platform yang berguna untuk mengembangkan dan menggunakan aplikasi web dan layanan web. [8]

### 2.6 Quality of Service

Parameter QoS menggolongkan kualitas transfer yang diberikan oleh suatu koneksi yang diperoleh dengan embandingkan unit data pada sisi masukan dan keluaran interface. Parameter QoS adalah :

#### 2.6.1 Delay

Delay adalah waktu tunda suatu paket yang diakibatkan oleh proses transmisi dari satu titik ke titik lain yang menjadi tujuannya.

### 2.6.2 Jitter

Jitter merupakan variasi delay antar paket yang dikirim oleh sumber ke tujuan di dalam media jaringan.

### 2.6.3 Throughput

Dalam komunikasi jaringan, seperti ethernet atau packet radio, atau jaringan, throughput adalah kecepatan (rate) transfer data efektif.

### 2.6.4 Packet Loss

*Packet loss* merupakan banyaknya paket yang gagal mencapai tempat tujuan saat pengiriman paket

## 2.7 Post Dial Delay

*Delay call setup* atau *delay* yang terjadi saat pembangunan hubungan disebut juga dengan *Post Dial Delay* (PDD). PDD berdasar rekomendasi IETF adalah periode yang dimulai setelah penelepon mendial digit terakhir nomor tujuannya hingga memperoleh bit terakhir yang berisi pesan status tujuan, apakah *ringing* atau *busy*. Berdasarkan rekomendasi ITU-T, PDD adalah interval antara digit terakhir yang telah di-*dial* hingga menerima ringback. PDD pada VoIP dapat diukur mulai dari kondisi INVITE hingga memperoleh tanda dari nomor tujuan, seperti kode TRYING atau RINGING [10].

## 2.8 Mean Opinion Score (MOS)

Metode ini merupakan metode yang digunakan untuk menentukan kualitas suara dalam jaringan IP berdasar kepada standart ITU-T P.800. Metode ini bersifat subjektif, karena berdasarkan pendapat pendengar. Untuk menentukan nilai MOS terdapat dua cara yaitu, *conversation opinion test* dan *listening test*. Rekomendasi nilai ITU-T P.800 untuk nilai MOS adalah sebagai berikut:

**Tabel 2.1** Rekomendasi ITU-T P.800 untuk nilai kualitas berdasarkan MOS

Nilai MOS	Opini
5	sangat baik
4	baik
3	cukup baik
2	tidak baik
1	buruk

## 3 Perancangan dan Implementasi

Untuk memberikan suatu gambaran bagaimana topologi jaringan yang akan diimplementasikan dalam pengerjaan penelitian ini, maka dibuat suatu desain topologi yang akan digunakan. Topologi jaringan ini terdiri dari beberapa alat seperti PC server, PC client, switch, dan access point. Gambar 3.3. merupakan bentuk topologi jaringan yang diimplementasikan



**Gambar 3.1** Topologi Jaringan *click to call*

Gambar diatas merupakan gambar konfigurasi jaringan untuk implementasi layanan *Click to Call* menggunakan server aplikasi Mobicents. Dalam penelitian ini server aplikasi Mobicents berada dalam satu personal komputer. Server aplikasi Mobicents diinstal dalam satu personal komputer dengan *operating system* linux ubuntu 12.04 dan memiliki satu alamat IP serta konfigurasi server dilakukan melalui *web console server* serta *GUI server*. Untuk *client* menggunakan empat buah PC/Notebook yang sudah terinstal *softphone X-Lite* yang dua diantaranya terhubung ke switch menggunakan media kabel (*wired*) dan yang lainnya terhubung ke access point menggunakan *wireless*. Media kabel yang digunakan adalah kabel UTP cat 5

## 4 Pengujian dan Analisis Hasil Implementasi

Pada bab ini akan dibahas mengenai analisis dari hasil implementasi yang telah dilakukan. Ada pun parameter data yang akan diukur adalah yang termasuk sebagai parameter QoS yaitu *throughput*, *one way delay*, *packet loss* dan *jitter*. Selain itu parameter lain untuk menunjang analisis performansi server Mobicents yang diukur yaitu *post dial delay* (PDD), *CPU usage* dan *Memory usage*. Untuk memperoleh data parameter QoS dan nilai PDD digunakan *software network analyzer* yaitu wireshark-1.0.4 yang diinstal di sisi *client*. Wireshark

digunakan untuk meng-*capture* paket-paket data serta protocol yang ada pada jaringan. Sedangkan untuk memperoleh data CPU *usage* dan *Memory usage* digunakan *software* Glances yang ter-*instal* pada PC server. Glances digunakan untuk melihat kinerja PC server, sistem atau aplikasi apa yang sedang berjalan di PC server tersebut. Pengujian dilakukan pada dua skenario yang berbeda, yaitu sebagai berikut

#### 1. Pengukuran *Background Traffic*

Pengukuran ini dilakukan pada server Mobicents dengan *background traffic* sebesar 20 Mbps, 40 Mbps, 60 Mbps, dan 80 Mbps. Sehingga didapat parameter uji *delay*, *jitter*, *packet loss*, *throughput*, PDD, CPU *usage* dan *Memory usage*.

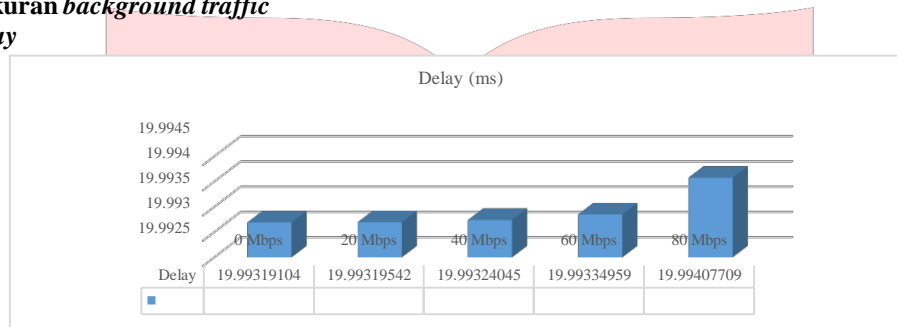
#### 2. Pengukuran *Wired* dan *Wireless*

Pengukuran ini dilakukan pada server Mobicents menggunakan dua media berbeda, yaitu *Wired* dan *Wireless*. Sehingga didapat parameter uji *delay*, *jitter*, *packet loss*, *throughput*, dan PDD.

### 4.1 Performansi QOS

#### 4.1.1 Pengukuran *background traffic*

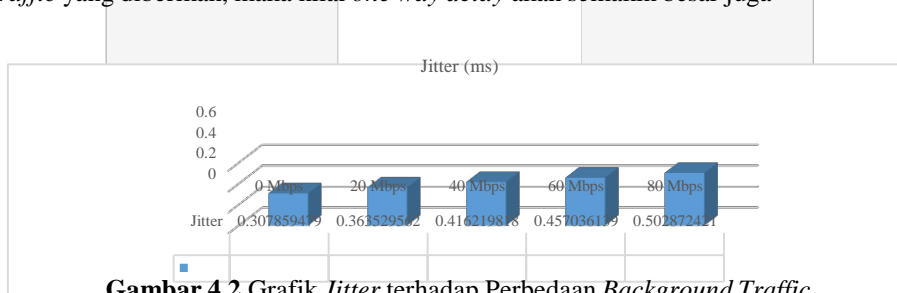
##### 4.1.1.1 *Delay*



**Gambar 4.1** Grafik *Delay* terhadap Perbedaan *Background Traffic*

Dari hasil pengukuran yang dilakukan pada layanan *click to call* dapat dilihat bahwa ada pengaruh perubahan besar nilai *background traffic* pada nilai *one way delay* terhadap server Mobicents. Hal ini dikarenakan semakin besar nilai *background traffic* maka lebar *bandwidth* yang tersedia untuk paket *click to call* akan semakin sempit sehingga mempengaruhi nilai *one way delay*. Dapat dilihat dari Gambar 4.1 bahwa semakin besar nilai *background traffic* yang diberikan, maka nilai *one way delay* akan semakin besar juga

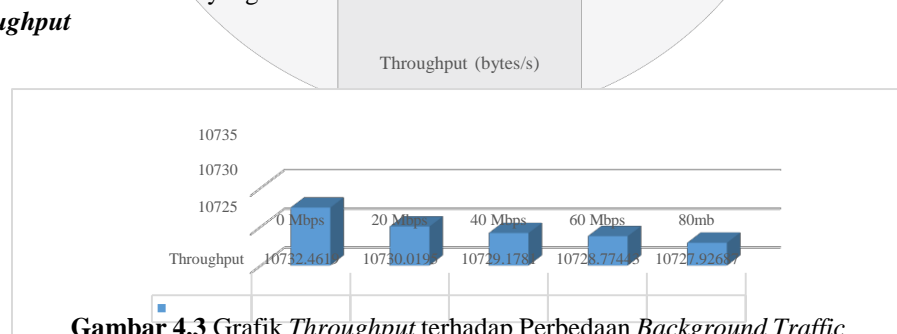
##### 4.1.1.2 *Jitter*



**Gambar 4.2** Grafik *Jitter* terhadap Perbedaan *Background Traffic*

Dari Gambar 4.2 dapat diamati semakin besar nilai *bandwidth*, maka semakin besar nilai *jitter*-nya. Hal ini dikarenakan semakin besar nilai *background traffic* maka lebar *bandwidth* yang tersedia untuk paket *click to call* akan semakin sempit sehingga mempengaruhi nilai *jitter*. Semakin besar variasi *delay* terjadi maka kedatangan paket semakin besar yang berubah-ubah.

##### 4.1.1.3 *Throughput*

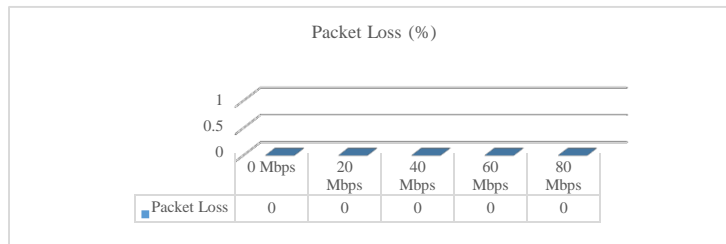


**Gambar 4.3** Grafik *Throughput* terhadap Perbedaan *Background Traffic*

Dari hasil pengukuran dapat diketahui bahwa nilai *throughput* pada server Mobicents akan semakin menurun seiring bertambahnya nilai *background traffic*. Hal ini dikarenakan semakin besar nilai *background traffic* maka lebar *bandwidth* yang tersedia untuk paket *click to call* akan semakin sempit sehingga mempengaruhi nilai *throughput*. Dapat dikatakan bahwa semakin besar *background traffic* yang diberikan, maka nilai *throughput*

akan semakin menurun yang artinya pada sisi *server* semakin tidak terbebani karena *bandwidth* system yang digunakan tidak semakin besar. Namun pada sisi *client* memiliki kecepatan pengiriman yang kecil.

#### 4.1.1.4 Packet Loss

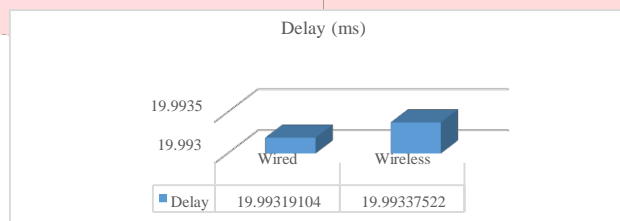


**Gambar 4. 1** Grafik *Packet Loss* terhadap Perbedaan *Background Traffic*

Dari hasil pengukuran yang dilakukan pada layanan *click to call* dapat dilihat bahwa tidak ada pengaruh perubahan besar nilai *background traffic* pada nilai *packet loss* terhadap *server* Mobicents. Nilai *packet loss* dari tanpa *background traffic* sampai dengan *background traffic* sebesar 80Mbps tidak mengalami peningkatan maupun penurunan, yaitu memiliki nilai yang sama 0%. Hal ini dikarenakan penelitian dilakukan dalam jaringan intranet Lab Switching, sehingga tidak banyak gangguan yang terjadi.

#### 4.1.2 Pengukuran Wired dan Wireless

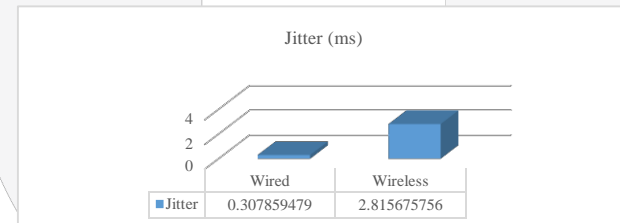
##### 4.1.2.1 Delay



**Gambar 4.5** Grafik *Delay* pada Media *Wired* dan *Wireless*

Dari hasil pengukuran yang dilakukan pada layanan *click to call* dapat dilihat bahwa ada pengaruh perubahan media dari *wired* ke *wireless* terhadap nilai *one way delay* terhadap *server* Mobicents. Dapat dilihat dari Gambar 4.5 bahwa nilai *one way delay* pada media *wired* lebih kecil dibandingkan menggunakan media *wireless*. Hal ini dikarenakan pada media *wired* jaringan lebih stabil dibandingkan pada media *wireless*. Dari skenario diatas dapat dikatakan bahwa nilai *one way delay* yang lebih baik adalah yang menggunakan media *wired*, karena ketika nilai *delay* kecil artinya kedatangan paket memiliki nilai keterlambatan kecil.

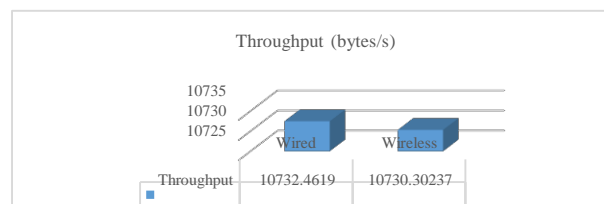
##### 4.1.2.2 Jitter



**Gambar 4.6** Grafik *Jitter* pada Media *Wired* dan *Wireless*

Dari hasil pengukuran yang dilakukan pada layanan *click to call* dapat dilihat bahwa ada pengaruh perubahan media dari *wired* ke *wireless* pada *server* Mobicents. Dari Gambar 4.6 dapat diamati nilai *jitter* pada media *wired* lebih kecil dibandingkan menggunakan media *wireless*. Hal ini dikarenakan pada media *wired* jaringan lebih stabil dibandingkan pada media *wireless*. Dari skenario diatas dapat dikatakan bahwa nilai *jitter* yang lebih baik adalah ketika menggunakan media *wired*, dikarenakan variasi keterlambatan kedatangan paket tidak banyak yang berubah-ubah sehingga nilai *jitter*nya kecil.

##### 4.1.2.3 Throughput

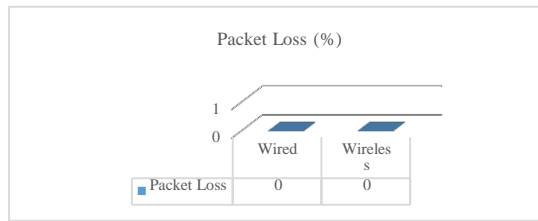


**Gambar 4.7** Grafik *Throughput* pada Media *Wired* dan *Wireless*

Dari hasil pengukuran dapat diketahui bahwa nilai *throughput* pada *server* Mobicents akan menurun jika menggunakan media *wireless* daripada media *wired*. Hal ini dikarenakan pada media *wired* jaringan lebih stabil dibandingkan pada media *wireless*. Dikatakan media *wired* lebih baik karena memiliki nilai *throughput* yang lebih

tinggi yang artinya adalah pengiriman data yang dilakukan menggunakan *bandwidth* besar sehingga akan cepat sampai tujuan.

**4.1.2.4 Packet Loss**



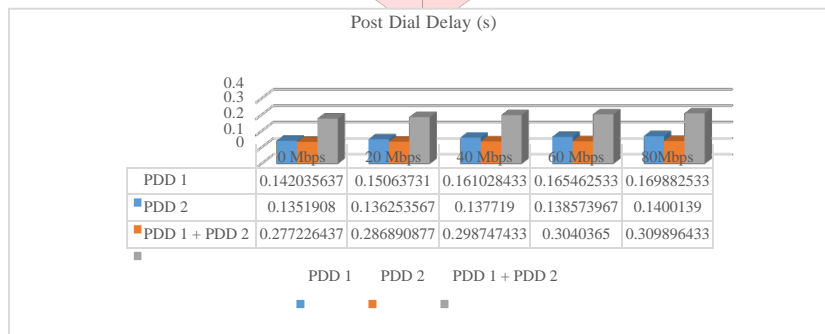
**Gambar 4.8** Grafik *Packet Loss* pada Media *Wired* dan *Wireless*

Dari hasil pengukuran yang dilakukan pada layanan *click to call* dapat dilihat bahwa tidak pengaruh perubahan media dari *wired* ke *wireless* pada *server* Mobicents. Nilai *packet loss* untuk media *wired* dan *wireless* sama-sama bernilai 0 %. Hal ini dikarenakan penelitian dilakukan dalam jaringan intranet Lab Switching, sehingga tidak banyak gangguan yang terjadi.

**4.2 Pengukuran Post Dial Delay**

Pada sub bab ini ditunjukkan hasil pengukuran *post dial delay* (PDD) dan analisis dari hasil yang telah didapatkan pada layanan *click to call* dengan menggunakan *server* Mobicents. Dalam pengukuran PDD ini pada setiap pengukuran akan didapat 2 nilai PDD, yaitu PDD 1 dan PDD 2 yang akan di jumlahkan. Dimana PDD 1, merupakan selisih INVITE dan RINGING yang di *capture* pada sisi *client* 1. Sedangkan PDD 2, pada *client* 2.

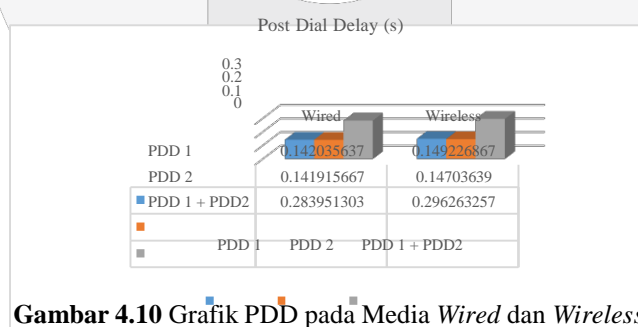
**4.2.1 Pengukuran PDD pada Background Traffic yang Berbeda**



**Gambar 4.9** Grafik PDD terhadap Perbedaan *Background Traffic*

Dari hasil pengukuran dapat diketahui bahwa nilai *throughput* pada *server* Mobicents akan menurun jika menggunakan media *wireless* daripada media *wired*. Hal ini dikarenakan pada media *wired* jaringan lebih stabil dibandingkan pada media *wireless*. Dikatakan media *wired* lebih baik karena memiliki nilai *throughput* yang lebih tinggi yang artinya adalah pengiriman data yang dilakukan menggunakan *bandwidth* besar sehingga akan cepat sampai tujuan.

**4.2.2 Pengukuran PDD pada Media Wired dan Wireless**

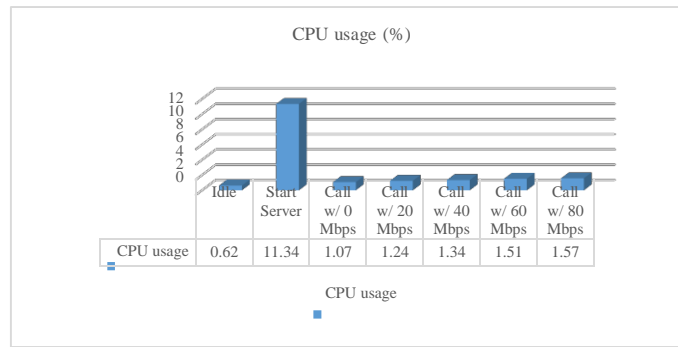


**Gambar 4.10** Grafik PDD pada Media *Wired* dan *Wireless*

Dari hasil pengukuran yang dilakukan pada layanan *click to call* dapat dilihat bahwa ada pengaruh perubahan media *wired* dan *wireless* pada nilai PDD terhadap *server* Mobicents. Dapat dilihat dari Gambar 4.10 bahwa semakin besar nilai *bandwidth* yang diberikan, maka nilai PDD akan semakin besar juga yang artinya, semakin lama proses pembentukan hubungan. Hal ini dikarenakan pada media *wired* jaringan lebih stabil dibandingkan pada media *wireless*.

**4.3 Pengukuran CPU usage**

Pada sub bab ini ditunjukkan hasil pengukuran CPU usage dan analisis dari hasil yang telah didapatkan pada layanan *click to call* dengan menggunakan *server* Mobicents.

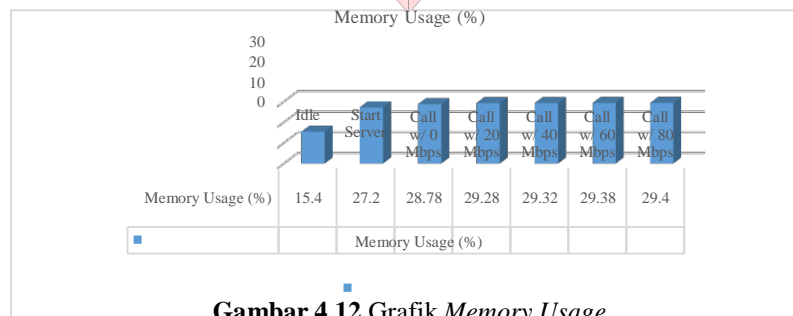


**Gambar 4.11** Grafik CPU usage

Dari hasil pengukuran yang dilakukan pada layanan *click to call* dapat dilihat bahwa ada pengaruh perubahan besar nilai *bandwidth* dari *background traffic* pada persentase CPU usage terhadap server Mobicents. Dapat dilihat dari Gambar 4.11 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 *click to call* 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 11.34%.

**4.4 Pengukuran Memory Usage**

Pada sub bab ini ditunjukkan hasil pengukuran *memory usage* dan analisis dari hasil yang telah didapatkan pada layanan *click to call* dengan menggunakan server Mobicents.



**Gambar 4.12** Grafik Memory Usage

Dari hasil pengukuran yang dilakukan pada layanan *click to call* dapat dilihat bahwa ada pengaruh perubahan besar nilai *bandwidth* dari *background traffic* pada persentase Memory usage terhadap server Mobicents. Dapat dilihat dari Gambar 4.12 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 *click to call* akan semakin sempit sehingga mempengaruhi kinerja server yaitu nilai Memory usage.

**4.5 Pengukuran dan Analisis MOS**

**4.5.1 MOS berdasarkan ITU-T P.800**

Metode ini merupakan metode yang digunakan untuk menentukan kualitas suara dalam jaringan IP berdasarkan standart ITU-T P.800. Metode ini bersifat subyektif, karena berdasarkan pendapat seseorang. Analisis performansi yang dilakukan di sini menggunakan analisis kualitatif, yaitu dengan mendatangkan responden untuk menilai performansi sistem yang telah dibuat dalam hal ini adalah server Mobicents. Cara pengukurannya setiap responden melakukan percobaan sebagai *client* dengan menjalankan layanan *click to call*. Sehingga responden dapat mengetahui kekurangan dan kelebihan dari system tersebut.

**Tabel 4.1** Nilai Masing-Masing Aspek

Aspek Penilaian	Nilai					Total	Nilai Rata-rata
	1XN	2XN	3XN	4XN	5XN		
	Buruk	Kurang	Cukup	Baik	Sangat Baik		
Tampilan Media Softphone	0	8	60	24	0	92	3.07
Kemudahan Menggunakan	0	8	63	20	0	91	3.03
Kualitas Audio	0	6	54	36	0	96	3.20

Efektivitas Layanan	0	8	66	16	0	90	3.00
Kualitas Layanan Keseluruhan	0	6	66	20	0	92	3.06
Rata-rata Total							3.07

#### 4.5.2 MOS dengan E-Model ITU-T G.107

Tabel 4. 2 Hasil Perhitungan MOS

Perhitungan MOS berdasar R Faktor									
	Delay	d-177,3	H (x)	Id	Packet Loss	30 ln (1+15e)	Ief	R faktor	MOS
Wired	19.99319	-157.307	0	0.479837	0	0	7	86.72016	4.060044
Wireless	19.99338	-157.307	0	0.479841	0	0	7	86.72016	4.060044

Dari tabel di atas kita dapat mengetahui bahwa beberapa aspek yang dinilai diatas telah cukup memenuhi standart kualitas yang diharapkan. Semua komponen bernilai lebih dari 4.00 yaitu baik 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 Mobicents sebagai *server* aplikasi yang berdiri sendiri untuk layanan *click to call* berhasil dilakukan
2. Menggunakan *background traffic* dengan nilai *bandwidth* dari 20 Mbps ke 80 Mbps, *server* Mobicents AS memiliki nilai *delay*, *jitter*, *post dial delay* meningkat, nilai *packet loss* tetap, dan nilai *throughput* menurun. Semua nilai yang didapat baik menurut standar ITU-T, Cisco dan IETF.
3. Menggunakan media *wired*, kualitas komunikasi lebih baik daripada menggunakan media *wireless*. *Statement* ini dibuktikan dengan pengukuran parameter dari *wired* ke *wireless* yaitu nilai *delay*, *jitter*, *post dial delay* meningkat, nilai *packet loss* tetap, dan nilai *throughput* menurun. Semua nilai yang didapat baik menurut standar ITU-T, Cisco dan IETF.
4. Pada pengukuran *CPU usage* dan *Memory usage*, nilai tertinggi masing-masing yaitu 11.34 % dan 29.4 %. Hasil ini membuktikan spesifikasi *PC server* cukup untuk menjalankan *server* Mobicents.
5. Pada pengukuran kualitas sistem layanan *click to call* dengan menggunakan metode Mean Opinion Score (MOS), didapatkan nilai rata-rata dari setiap aspek penilaian sebesar 3.07 itu cukup baik/fair berdasarkan rekomendasi ITU-T P.800, sedangkan menggunakan metode E-Model ITU-T G.107, didapatkan nilai untuk media *wired* dan *wireless* yaitu 4.0 dimana baik/good berdasarkan ITU-T P.800.

### 5.2 Saran

Saran yang dapat diajukan untuk penelitian lebih lanjut mengenai topik ini adalah:

1. Perlu diimplementasikan layanan *click to call* pada jaringan akses *mobile IPv6*.
2. Perlu dilakukan penelitian terhadap layanan lain yang dapat di-*deploy* oleh Mobicents seperti WebRTC, Diameter, PBX, dll.
3. Perlu dilakukan penelitian terhadap keamanan (*security*) dari *server* Mobicents.

### Daftar Pustaka :

- [1] O. O. Ayokunle, "Integrating Voice over Internet Protocol (VoIP) Technology as a Communication Tool on a Converged Network in Nigeria," *International Journal of Information and Communication Technology Research*, 2012.
- [2] A. T. G. W. Mosiuoa Tsietsi, "Mobicents as a Service Creation and Deployment Environment for the Open IMS Core," 2006.
- [3] B. I. Priaksa, "Implementasi Dan Analisis Performansi Mobicents sebagai Server Aplikasi pada jaringan IMS untuk layanan video conference," 2014.
- [4] P. SEGEČ, "SIP SERVICES USING SIP SERVLET API – THE INFOLINE SERVICE," *Journal of Information, Control and Management Systems*, 2010.
- [5] L. T. Switching, Modul Praktikum Jartel, 2014.
- [6] S. W. B. B. A. S. A. K. Mr. Manish Giri, "Migration of Mobicents SIP Servlets on Cloud Platform," *International Journal of Scientific and Research Publications*, 2012.
- [7] "Mobicents," [Online]. Available: [www.mobicents.org](http://www.mobicents.org).
- [8] "Apache Tomcat," The Apache Software Foundation , [Online]. Available: <http://tomcat.apache.org/>. [Accessed 5 May 2015].
- [9] "QoS," 2014. [Online]. Available: <http://www.voip-info.org/wiki/view/QoS>.
- [10] Yulianto, "Analisa dan Implementasi Integrasi Layanan VoIP dengan Email," in *Analisa dan Implementasi Integrasi Layanan VoIP dengan Email*, Bandung, 2009, p. 16.