

# IMPLEMENTASI DAN ANALISIS MOBILE CLOUD GAMES PADA CLIENT ANDROID MENGGUNAKAN OPEN-SOURCE COUD GAMING SERVER GAMINGANYWHERE

## IMPLEMENTATION AND ANALYSIS MOBILE CLOUD GAMES FOR ANDROID CLIENT USING OPEN-SOURCE CLOUD GAMING SERVER GAMINGANYWHERE

Dimas Adibuana<sup>1</sup>, Dr. Ir. Rendy Munadi M.T.<sup>2</sup>, Hafidudin, S.T., M.T.<sup>3</sup>  
<sup>1,2,3</sup> Fakultas Teknik Elektro, Telkom University, Bandung

Jalan Telekomunikasi, Dayeuh Kolot Bandung 40257 Indonesia

<sup>1</sup>[dimasadibuana@yahoo.com](mailto:dimasadibuana@yahoo.com) <sup>2</sup>[rendy\\_munadi@yahoo.co.id](mailto:rendy_munadi@yahoo.co.id) <sup>3</sup>[hafidudin@telkomuniversity.ac.id](mailto:hafidudin@telkomuniversity.ac.id)

---

### ABSTRAK

*Mobile cloud games* merupakan sebuah layanan *cloud computing* yang merupakan pengembangan dari Infrastruktur sebagai layanan pada model layanan *cloud computing*. Perangkat *mobile phone* dengan *operation sytem* android yang bekerja sebagai *client* mampu mengakses *server* yang menjalankan suatu *game* melalui jaringan nirkabel (WLAN). *Server* melakukan capture terhadap frame *audio* dan frame *video* kemudian dikodekan dan dikirimkan kepada *client*. Setelah *client* mendapatkan frame A/V, *client* akan mengkodekan *frame-frame* yang diterima sehingga *user* bisa bermain dan melakukan *input control*. dilakukan pengujian dengan tiga parameter yaitu kebutuhan *resource*, *Quality of Service*, dan *Quality of Experience*.

Hasil pengujian dihasilkan kebutuhan *resource client* yang hanya memakai 10% *cpu usage*, 20 MB RAM untuk *game* Lego Batman dan 3% *cpu usage*, 12 MB RAM untuk *game* Deadpool. Nilai FPS *client* yaitu 27 untuk Lego Batman dan 14 untuk deadpool, nilai ini terbilang kecil dikarenakan kemampuan *rendering server* yang kurang maksimal. Untuk *delay respon* sistem cukup memuaskan yaitu bernilai 0,78 detik untuk Lego Batman dan 0,26 detik untuk Deadpool pada *bandwidth* 3 Mbps dengan topologi jaringan lokal.

**Keyword :** Keyword : cloud computing, mobile cloud Games, gamingsnywhere

---

### ABSTRACT

*Mobile cloud games* is a cloud computing service which is the development of infrastructure as a service cloud computing service model. Mobile phone devices with android operating system that works as a client is able to access a server running a game via wireless network ( WLAN ). Server performs against the frame capture audio and video frame is then encoded and sent to the client. Once the client gets frame A / V, the client will encode frames received so the user can play and perform input control. Tested with three parameters: resource usage, Quality of Service and Quality of Experience.

Measurement result is the resource needs of the client that only uses 10 % of *cpu usage*, 20 MB of RAM for the game Lego Batman and 3% of *cpu usage*, 12 MB RAM for Deadpool game . FPS value for the client about 27 for Lego Batman and 14 for Deadpool, value for Deadpool is smaller than Lego Batman because the rendering capabilities are less than the maximum server . Responsive delay was satisfactory system that is worth to Lego Batman 0.78 seconds and 0.26 seconds for Deadpool at 3 Mbps bandwidth with the local network topology.

**Keyword :** Keyword : cloud computing, mobile cloud games, gaminganywhere

---

### 1. Pendahuluan

Dewasa ini, Teknologi sebuah telepon seluler atau *mobile phone* berkembang dengan pesat. Dengan hadirnya *Operating System* Android pada telepon seluler, pengguna *mobile phone* mampu mengakses dan mengatur layanan serta fitur-fitur pada *Operating System* dengan bebas. *Mobile phone* juga berfungsi tidak hanya sebagai alat komunikasi saja, namun dengan adanya berbagai layanan multimedia, pengguna *mobile phone* dapat bermain game dengan pengalaman yang lebih menyenangkan.

Dengan perkembangan teknologi yang pesat, kebutuhan akan *resource* sebuah *game* pun lebih ditingkatkan seiring dengan kecanggihannya game tersebut. Untuk memenuhi spesifikasi yang dibutuhkan suatu game, maka perlu dilakukan *upgrade* pada *Central Processor Unit*, *Graphic Processor Unit*, RAM, dan menambah media penyimpanan (*storage*). Tentu tidak murah biaya untuk melakukan *upgrade* tersebut. Untuk mengatasi hal tersebut, maka dilakukan terobosan yaitu dibangun sebuah *Cloud Gaming* berdasarkan *cloud computing*.

Pada Tugas Akhir ini akan dilakukan implementasi sebuah infrastruktur *mobile cloud games* berbasis *cloud* menggunakan *server open-source* GamingAnywhere. GamingAnywhere adalah sebuah *open-source cloud gaming system* yang bisa dikonfigurasi dan dimodifikasi sesuai dengan keinginan pengguna.

## 2. Cloud Computing

*Cloud Computing* sebagai suatu layanan teknologi informasi yang dapat dimanfaatkan oleh pengguna dengan berbasis jaringan. Dimana suatu sumber daya, perangkat lunak, informasi dan aplikasi disediakan pada komputer yang disebut *server* dan untuk digunakan oleh komputer lain yang membutuhkan. *Cloud computing* mempunyai dua kata yaitu “*Cloud*” dan “*Computing*”. *Cloud* yang berarti awan atau *internet* itu sendiri dan *computing* yaitu proses komputasi. Konsep *Cloud computing* biasanya dianggap sebagai *internet* dimana *internet* sendiri digambarkan sebagai awan besar yang berisi sekumpulan komputer yang saling terhubung..

### 2.1 Mobile Cloud Gaming

*Mobile cloud gaming* didefinisikan sebagai suatu perangkat *mobile* yang mampu mengakses *cloud* sebagai eksternal *resource* untuk pengolahan skenario dan interaksi untuk menjalankan suatu *game*. Ada dua tipe *mobile cloud games*[8], yaitu:

#### 1. Mobile cloud video gaming

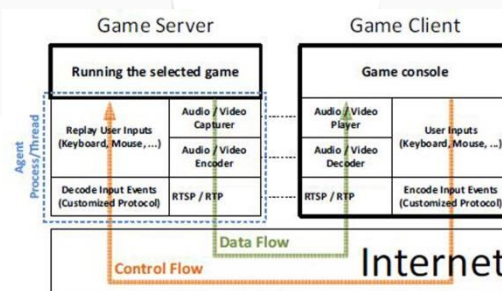
Pemain mengakses *server game* dari portal yang tersedia di *mobile phone*, kemudian client melakukan inisiasi pada *server machine* dan menjalankan *game* yang telah di pilih di *mobile phone*. Saat *game* sedang berjalan di *server machine* atau *virtual machine*, sisi *client* lalu melakukan *game streaming* dimana *game* berjalan bersamaan pada *client* dan *server machine* atau *virtual machine* di *cloud*.

#### 2. Mobile browser gaming

*Browser games* merupakan jenis dari *games* komputer, banyak dari *game* tersebut berjalan di situs jejaring sosial dengan pengguna yang cukup banyak. Seperti yang di gambarkan pada gambar 2.2, *server mobile browser games* berjalan di *cloud*. Sementara *web browser* pada ponsel berfungsi sebagai *interface* untuk pemain. Komponen penting dalam *mobile browser games* meliputi:

- a) *Game server* : aplikasi pada sisi *server*, yang terdiri dari *server-side scripting* dan *user-interface scripting*.
- b) *Communication protocol* : untuk mengaktifkan prosedur sebuah *game*, komunikasi antara *server* dengan *mobile browser*.
- c) Aplikasi *runtime* : pada ponsel, aplikasi *runtime* harus ter-*install* pada *web browser* sebagai *plugin*. Beberapa contoh aplikasi runtime diantaranya Adobe Flash player, Adobe Shockwave player, Microsoft Silverlight.

### 2.2 GamingAnywhere

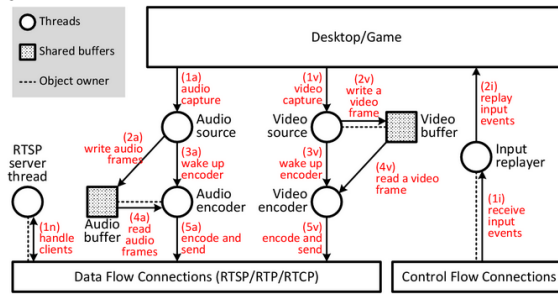


Gambar 2.1: Hubungan antara Server dan Client[3]

Gambar 2.1 menunjukkan arsitektur *game server* dan *game client* pada GamingAnywhere. Pada arsitektur tersebut, ada 2 tipe *flow network* yaitu *data flow* dan *control flow*. *Data flow* berfungsi sebagai *stream audio* dan *video (A/V) frame* dari *server* ke *client*, sedangkan *control flow* digunakan untuk mengirimkan perintah dari *client* ke *server*. Sistem GamingAnywhere mendukung beberapa tipe sebuah *game*, termasuk *game komputer* dan *game* yang terintegrasi pada sebuah *web* atau *Web-based games*. Terdapat sebuah agen yang berjalan bersama *game* yang dimainkan dapat berupa *stand-alone proses* maupun *thread* yang di-*inject* pada *game*. Agen ini memiliki dua tugas utama yaitu, pertama mengambil *frame audio* dan *video* dari sebuah *game*, mengkodekan *frame*, kemudian mengirimkan *frame* yang sudah dikodekan ke *client* via *data flow*. Kedua berinteraksi dengan *game* dengan bertindak sebagai user. Agen menerima perintah dari user yang merupakan input dari keyboard, mouse, maupun joystick dari user kemudian setelah dikodekan dikirimkan melalui protokol yang dikostumisasi

melalui *control flow*, diterima oleh *server* dan didekodekan hasilnya berupa *input* yang dimasukkan oleh user dan akan memainkan *game* dengan input dari user.

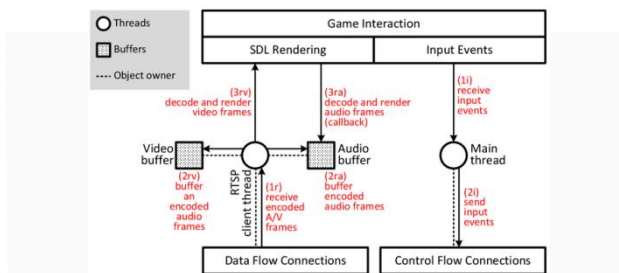
### 2.3 Server GamingAnywhere



Gambar 2.2 antara modul *server*, *shared buffer*, koneksi Hubungan jaringan.[3]

Hubungan antara *server* modul ditunjukkan pada gambar 2.2 dimana beberapa modul di implementasikan pada bagian terpisah. Ketika agen di *launch*, empat modul yaitu *server RTSP*, *audio source*, *video source*, dan *input replayer* ikut ter-*launch* juga. *RTSP server* dan *input replayer* segera menunggu *client* yang masuk (dimulai dari 1n dan 1i pada gambar). Modul *Audio source* dan *Video source* didiamkan setelah proses inisialisasi. Ketika *client* terhubung ke *RTSP server*, bagian *encoder* akan dijalankan dan *encoder* akan memberitahu modul sumber sesuai yang siap untuk mengkodekan *frames* yang ditangkap. Modul sumber kemudian memulai untuk menangkap *audio* dan *video frame* ketika satu atau lebih *encoders* sudah siap. *Audio* dan *video* dikedokan bersamaan secara *real time*. *Audio* dan *video* digambarkan di jalur 1a ke 5a dan 1v ke 5v.

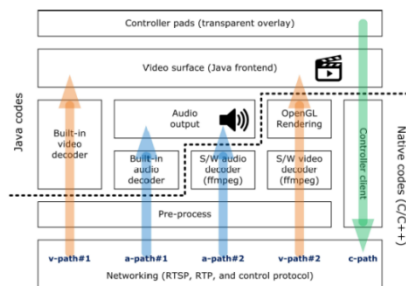
### 2.4 Mobile Client GamingAnywhere



Gambar 2.3: Hubungan antara modul *client*, *shared buffer*, koneksi jaringan[4]

*Client* pada dasarnya adalah *remote desktop* yang menampilkan layar permainan secara *real-time* yang di *capture* di *server* dan disampaikan dalam bentuk *video* dan *audio* yang dikodekan. Hubungan antara modul *client* ditunjukkan pada gambar 2.3. *GamingAnywhere client* berisi dua bagian, yang pertama yaitu untuk menangani *input handle* (dimulai pada jalur 1i) dan yang kedua adalah *render audio* dan *video frame* (dimulai pada jalur 1r). Untuk menggunakan *GamingAnywhere*, pengguna terlebih dahulu masuk ke portal dan memilih game yang akan dimainkan. Portal kemudian menjalankan game tersebut dan *server* juga akan menjalankannya di cloud. Portal lalu memberitahu pengguna untuk membangun koneksi pada *server*, dimana game tersebut sudah siap untuk dijalankan.

### 2.6 Porting Client ke Android



Gambar 2.4: Arsitektur *mobile client*[4]

Gambar 2.4 menunjukkan arsitektur perangkat lunak pada *client* android. Kerangka pada *mobile client* ditulis pada bahasa pemrograman java, tetapi beberapa komponen diimplementasikan sebagai *loadable shared object* dalam *native C* dan *C++ codes*. *Native codes* bertujuan untuk menjembatani kode antara java dan GamingAnyWhere library dan meminimalkan *maintenance overhead* dengan memakai kode yang sudah ada di desktop.

### 2.6.1 Decoding

*Mobile client* GamingAnyWhere mendukung dua jenis *codecs*, yaitu *software codecs* dan *Adnroid built-in codecs*. *Audio* dan *video codecs* dikonfigurasi secara independen. Oleh karena itu, pada gambar 2.4 ada empat jalur, yaitu *a-path#1* dan *a-path#2* untuk *audio* dan *v-path#1* dan *v-path#2* untuk *video*. *Software codecs* sama jika *client* yang digunakan adalah desktop. Sebaliknya, *built-in codecs* telah disediakan oleh Android MediaCodec framework, yang tersedia pada Android 4.1 atau lebih. MediaCodec framework hanya dapat diakses dari sisi java. Oleh karena itu, pada penerimaan paket *audio* dan *video* di jaringan, paket dipecah dalam *native codes* dan kemudian diteruskan ke java untuk dikodekan. Untuk paket *audio*, *pengkodean frame audio* yang masih mentah dapat diambil dari framework dan digunakan untuk *playback*. Untuk paket *video*, *pengkodean video frame* secara dilakukan langsung ke *user*, artinya *frame video* mentah tidak dapat diakses oleh aplikasi dan secara otomatis dirubah ukurannya agar sesuai dengan resolusi layar *mobile phone*.

### 2.6.2 Controller

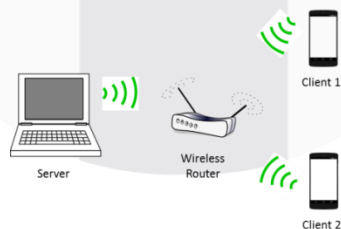


Gambar 2.5: Screenshot *mobile client*

*Controller pad* yaitu tombol transparan pada permukaan layar *video*. Dirancang untuk mendukung berbagai jenis game. Dalam *software* GAClient, telah dirancang beberapa *controller* yaitu Basic, Empty, DualPad, Limbo, N64, NDS, dan PSP seperti pada gambar 2.8

## PERANCANGAN SISTEM

### 3.1 Topologi Sistem



Gambar 3.1: Topologi sistem *Mobile Cloud Games* dengan jaringan nirkabel

Pengujian dilakukan dengan Wireless Router TP-Link Wireless N ADSL2+ Modem Router dengan kecepatan *up to* 300 Mbps. Jaringan yang digunakan adalah jaringan dengan *low traffic* dan secara *direct*. Laptop bertindak sebagai *server cloud gaming*. Dua *mobile phones* bertindak sebagai *client*.

### 3.2 Skenario pengujian

Dalam pengujian *mobile cloud gaming* pada tugas akhir ini, ada tiga aspek dalam pengujian diantaranya yaitu kebutuhan *resource* yang meliputi *CPU Usage*, *RAM Usage*, *Frame Rate (FPS)* untuk sisi *server* dan sisi *client*, *Quality of Service (QoS)* yang meliputi *delay*, *packet loss video*, dan *throughput* pada sistem *cloud gaming* dan *Quality of experience (QOE)* yang meliputi *responsiveness*, *control*, dan *graphic quality*.

## 4.1 Pengujian Resource

Berikut ini hasil pengujian resource server dengan masing-masing rata-rata 10 kali percobaan untuk setiap resolusi yang digunakan

Tabel 4.1 : *resource game* Lego Batman saat GA tidak aktif

Game	Resolusi	CPU Usage (%)	RAM (MB)	FPS
Batman	1024x768	26,6	492,2	39,405
	1280x720	29,7	509,9	38,261
	1366x768	38	513	31,903

Tabel 4.2 : *resource game* Lego Batman saat GA aktif

Game	Resolusi	CPU Usage (%)	RAM (MB)	FPS
Batman	1024x768	32,4	773	33,254
	1280x720	35,1	808,1	28,237
	1366x768	39,1	837,3	27,592

Tabel 4.3 : *resource game* Deadpool saat GA tidak aktif

Game	Resolusi	CPU Usage (%)	RAM (MB)	FPS
Deadpool	1024x768	38,9	618,4	19,565
	1280x720	39,8	641,6	17,417
	1366x768	44,6	685,5	15,493

Tabel 4.4 : *resource game* Deadpool saat GA aktif

Game	Resolusi	CPU Usage (%)	RAM (MB)	FPS
Deadpool	1024x768	55,1	999,7	17,007
	1280x720	58,5	1070	16,685

	1366x768	66,2	1073,8	13,625
--	----------	------	--------	--------

Tabel 4.5 : *resource server game* Lego Batman saat diakses 1 client

Game	Resolusi	CPU Usage (%)	RAM (MB)	FPS
Batman	1024x768	74,2	849,1	38
	1280x720	77	936,7	39
	1366x768	79,3	954,5	39

Tabel 4.6 : *resource server game* Lego Batman saat diakses 2 client

Game	Resolusi	CPU Usage (%)	RAM (MB)	FPS
Batman	1024x768	65,9	837,2	29
	1280x720	80,8	876,6	17
	1366x768	85,7	865,9	14

Tabel 4.7 : *resource server* Deadpool saat diakses 1 client

Game	Resolusi	CPU Usage (%)	RAM (MB)	FPS
Deadpool	1024x768	69,9	967,5	15
	1280x720	75,1	995,6	13
	1366x768	81,9	954,5	12

Tabel 4.8 : *resource server* Deadpool saat diakses 2 client

Game	Resolusi	CPU Usage (%)	RAM (MB)	FPS
Deadpool	1024x768	65,9	837,2	15
	1280x720	80,8	876,6	12
	1366x768	85,7	865,9	11

Berikut ini hasil pengujian resource *client* dengan masing-masing 10 kali percobaan :

Tabel 4.9 : *resource client* Lego Batman pada *client* ke-1 dan ke-2

Game	Resolusi	CPU Usage (%)		RAM (MB)		FPS	
		Client 1	Client 2	Client 1	Client 2	Client 1	Client 2
		Batman	1024x768	11,1	7,7	14,6	16,4
1280x720	11,3		10,8	27,8	27	33	29
1366x768	8,2		6,7	16,9	19,6	34	29

Tabel 4.10 : *resource client* Deadpool pada *client* ke-1 dan ke-2

Game	Resolusi	CPU Usage (%)		RAM (MB)		FPS	
		Client 1	Client 2	Client 1	Client 2	Client 1	Client 2
		Deadpool	1024x768	3,4	5,7	11,4	15,9
1280x720	3,3		6,9	11,5	19	17	28
1366x768	2,4		10,2	12,5	15,5	19	23

Berikut ini hasil pengujian dengan *server virtual machine*:

Tabel 4.11 : *server game* Lego Batman pada *real machine* dan *virtual machine*

Game	Resolusi	CPU Usage (%)		RAM (MB)		FPS	
		Real	Virtual	Real	Virtual	Real	Virtual
		Batman	1024x768	68,4	55,8	809,2	748,8
1280x720	68,7		56,2	870,4	815,6	15,923	6,484
1366x768	69,3		59,9	874,1	817,6	13,601	5,625

Tabel 4.12 : *client game* Lego Batman pada *real machine* dan *virtual machine*

Game	Resolusi	CPU Usage (%)		RAM (MB)		FPS	
		Real	Virtual	Real	Virtual	Real	Virtual
		Batman	1024x768	4,7	4,6	13,5	14,7
1280x720	4,7		4,1	16,9	16,1	23,8	23,5
1366x768	6,1		5,6	18,9	15,8	20,9	25,5

## 4.2 Pengujian QOS

Berikut ini hasil *delay*, *packet loss* dan *throughput* pada game Lego Batman dan Deadpool. *Responsive delay* adalah jumlah dari keseluruhan delay. Hasil pengukuran merupakan rata-rata dari 10 kali percobaan.

Tabel 4.13 : masing-masing *delay game* Lego Batman

Band width	Delay				
	Video (s)	Audio (s)	Control (s)	Network Downlink (s)	Network Uplink(s)
1 Mbps	0,0133 37309	0,0788 21269	0,2229 62993	1,0981	1,1495
3 Mbps	0,0041 46451	0,0784 107	0,1968 83439	0,211	0,3339
5 Mbps	0,0038 17761	0,0783 46944	0,1984 95795	0,00122	0,00578

Tabel 4.15 : masing-masing *delay game* Deadpool

Band width	Delay				
	Video (s)	Audio (s)	Control (s)	Network Downlink (s)	Network Uplink(s)
1 Mbps	0,0166 51823	0,0786 2317	0,1677 57863	0,1362	0,2687
3 Mbps	0,0192 94387	0,0783 74299	0,1540 91012	0,00238	0,0032
5 Mbps	0,0130 34739	0,0784 69272	0,1261 21306	0,00607	0,006

Tabel 4.14 : *responsive delay game* Lego Batman

Bandwidth	Responsive Delay (s)
1 Mbps	2,562721571
3 Mbps	0,77704059
5 Mbps	0,2904405

Tabel 4.16 : *responsive delay game* Deadpool

Bandwidth	Responsive Delay (s)
1 Mbps	0,667932856
3 Mbps	0,257339698
5 Mbps	0,229695317

Berikut ini hasil pengukuran *packet loss*:

Tabel 4.17 : *Packet loss game* Lego Batman

Bandwidth	Packet Loss (%)	
	Video	Audio
1 Mbps	72,65	9,25
3 Mbps	8,9	0,11
5 Mbps	0,05	0

Tabel 4.18 : *Packet loss game* Deadpool

Bandwidth	Packet Loss (%)	
	Video	Audio
1 Mbps	10,49	0,91
3 Mbps	0,2	0,03
5 Mbps	0	0

Berikut ini hasil pengukuran *Throughput*:

Tabel 4.19 : *Throughput game* Lego Batman

Bandwidth	Throughput ( Mbit/s)		
	Video	Audio	Control
1 Mbps	0,8603	0,1217	0,0024
3 Mbps	2,6783	0,134	0,0029
5 Mbps	2,8604	0,134	0,0027

Tabel 4.20 : *Throughput game* Deadpool

Bandwidth	Throughput ( Mbit/s)		
	Video	Audio	Control
1 Mbps	0,6642	0,1335	0,003
3 Mbps	0,5568	0,1339	0,0033
5 Mbps	0,7341	0,1339	0,0033

### 4.3 Pengujian QOE

Hasil pengukuran yaitu rata-rata dari 10 orang koresponden untuk masing-masing tiap *game*.

Tabel 4.21 : hasil QOE

Game	Bandwidth	Responsiveness	Control	Graphic quality
Lego Batman	1 Mbps	1,5	1,4	1,2
	3 Mbps	3,6	3,8	2,3
	5 Mbps	3,9	3,9	5,2

Game	Bandwidth	Responsiveness	Control	Graphic quality
Deadpool	1 Mbps	1,9	1,8	1,6
	3 Mbps	2,3	2	4,1
	5 Mbps	2,9	2,4	4,4

### 4.4 Analisis kebutuhan *resource*

Dari hasil pengukuran *resource*, terlihat bahwa kebutuhan *resource* meningkat disaat GamingAnyWhere aktif. Peningkatan yang cukup signifikan yaitu pada CPU Usage. Untuk RAM mengalami peningkatan sekitar 100 MB untuk *game* Deadpool. Untuk perubahan resolusi, hal yang paling berpengaruh yaitu pada FPS, dikarenakan resolusi yang semakin berat, maka proses *rendering* semakin berat juga dan membenani kinerja CPU dan RAM, dikarenakan pengujian dilakukan dengan *server* yang tidak memiliki spesifikasi tinggi. Maka hasil *frame* per detik yang dihasilkanpun rendah. Pada saat pengujian dilakukan pada *virtual machine*, terlihat bahwa untuk CPU dan RAM terlihat lebih kecil, dikareakan OS pada *virtual machine* masih "*fresh installed*" sehingga tidak banyak program yang berjalan pada *background*. Untuk fpsnya sangat kecil dikarenakan software *virtualbox* hanya mampu memvirtuaisasikan *video card* maksimal 256 MB, sehingga hasil *renderin* pada *game* di *virtual machine* tidak masimal dibandingkan *real machine*.

Hasil *resource client* terlihat sangat kecil dikarenakan *client* mengakses *game* dengan cara *streaming* sehingga proses *rendering game* dibebankan pada *server*.

#### 4.5 Analisis QOS

Tingkat *delay* respon pada *game* Lego Batman bernilai lebih tinggi dibandingkan *game* Deadpool disebabkan paket *video* pada *game* Lego Batman lebih besar dan hal ini dibuktikan dengan lebih interaktifnya *game* Lego Batman dibandingkan *game* Deadpool. Oleh karena itu, kualitas *game* Lego Batman sangat buruk bila dijalankan dengan batas *bandwidth* 1 Mbps. Lain halnya dengan *game* *Deadpool* walaupun dijalankan dengan *bandwidth* 1 Mbps, namun *delay* respon cenderung kecil dan kualitasnya cukup baik. Untuk *bandwidth* 3 dan 5 Mbit/s, *game* dapat berjalan dengan kualitas yang cukup baik.

*Packet loss video* dengan *bandwidth* 1 Mbps terbilang cukup tinggi pada *game* Lego Batman mencapai 72,65 %. Dengan *packet loss* setinggi ini, hasil *frame video* pada layar *mobile client* sangat buruk. Untuk *packet loss audio* mencapai nilai 10,49% walaupun ada sedikit *packet loss* pada audio, namun nilai ini masih cukup untuk mendapatkan pengalaman bermain yang cukup memuaskan. Pada *game* Deadpool, nilai *packet loss video* dan *audio* kurang dari 10% dan cukup memuaskan. Hal ini membuktikan bahwa pada *game* Lego batman, paket *video* yang dikirim lebih banyak sehingga membutuhkan *bandwidth* yang lebih besar bila dibandingkan dengan *game* Deadpool. Untuk *bandwidth* 3 Mbps untuk *delay video* dan *audio* untuk kedua *game* cukup memuaskan yaitu kurang dari 10%. Bahkan untuk *game* Deadpool tidak mencapai 1% dan sangat memuaskan. Pada *bandwidth* 5 Mbps untuk kedua *game* sangat memuaskan karena hampir tidak ada paket yang hilang untuk setiap *frame game* yang dijalankan.

Nilai *throughput* pada *game* Lego Batman lebih tinggi dibanding *game* *deadpool* dan berbanding lurus dengan *packet loss* yang dihasilkan. *game* Deadpool hanya memiliki *throughput* dibawah 1 Mbps untuk semua variabel *bandwidth*, karena itu *paket loss* yang terjadi cenderung lebih sedikit dibanding *game* Lego Batman.

#### 5. Kesimpulan

Setelah melakukan pengujian dan pengukuran yang didapat serta analisis hasil pengujian. Maka dapat disimpulkan sebagai berikut.

1. Sistem *mobile cloud gaming server* mampu mengimplementasikan suatu *game* PC yang dapat dijalankan pada *mobile phone* Android serta menekan *resource* CPU *usage* dan RAM *usage* pada *client* dengan dibuktikannya hasil CPU *usage* kurang dari 15% dan RAM *usage* kurang dari 30 MB untuk kedua *game*. *Frame* yang dihasilkan tiap detik yaitu sekitar 39 *frame* untuk Lego Batman dan 13 *frame* untuk Deadpool. FPS pada Deadpool terbilang rendah dikarenakan kemampuan *server* yang tidak mampu *render game* dengan maksimal.
2. Hasil *responsive delay* yang memuaskan yaitu dengan memakai *bandwidth* lebih dari 3 Mbps. Apabila implementasi dilakukan dengan *bandwidth* kurang dari 3 Mbps, *responsive delay* bernilai tinggi mencapai 2,5 detik untuk Lego Batman dan 0,6 detik untuk Deadpool, *packet loss video* 72,65 % untuk Lego Batman dan 10,49 % untuk Deadpool.
3. Implementasi dengan *virtual machine* mampu menjalankan *game* dengan sistem *cloud game*, namun proses *rendering game* sangat rendah dengan dibuktikannya nilai FPS kurang dari 10 *frame per second* pada *game* Lego Batman. Hal ini dikarenakan software VirtualBox hanya mampu melakukan virtualisasi *video card* maksimal 256 MB.
4. Hasil *experience* oleh *user* pada *game* Lego Batman dengan nilai rata-rata puas yaitu di atas 3 untuk semua komponen. Untuk *game* Deadpool pada nilai *responsiveness* dan *control* tidak mencapai nilai 3 dikarenakan kemampuan *server* yang kurang maksimal sehingga berdampak pada performa *client*.

## Referensi

- [1] Peter Mell and Timothy Grance, "The NIST Definition of Cloud Computing," Recommendation of the National Institute of Standards and Technology, Sept, 2011.
- [2] W. Purbo, Onno. 2012. Membuat Sendiri Cloud Computing Server Menggunakan Open Source, Yogyakarta : Penerbit Andi.
- [3] Chun-Ying Huang, Cheng-Hsin Hsu, Yu-Chun Chang, and Kuan-Ta Chen, "GamingAnywhere: An Open Cloud Gaming System," Proceedings of ACM Multimedia Systems 2013, Feb, 2013.
- [4] Chung-Ying Huang, Cheng-Hsin Hsu, De-Yu Chen, and Kuan-Ta Chen, "Quantifying User Satisfaction in Mobile Cloud Games," Proceedings of ACM MoVid 2014.
- [5] Chun-Ying Huang, De-Yu Chen, Cheng-Hsin Hsu, and Kuan-Ta Chen, "GamingAnywhere: An Open-Source Cloud Gaming Testbed," Proceedings of ACM Multimedia 2013 (Open Source Software Competition Track), Oct, 2013
- [6] Sheng-Wei (kuan-ta) Chen, Yu-Chun Yeng, Po Han Tseng, Chun-Yi Huang, and Ching-Laung He,"Cloud Gaming Latency Analysis: OnLive and StreamMyGame Delay Measurement," Proceedings of ACM Multimedia Systems 2013, Apr, 2013.
- [7] Kuan-Ta Chen, Yu-Chun Chang, Hwai- Jung hsu, De-Yu Chen,Chun Yi-Huang, and Cheng-Hsin Hsu,"On the Quality of Service of Cloud Gaming Systems," IEEE Transactions on Multimedia, Vol 16, No 2, Feb, 2014.
- [8] Wei Cai, Victor C.M Leung, and Min Chen, "Next generation Mobile Cloud Gaming," IEEE Seventh International Symposium on Service-Oriented System Engineering, 2013.
- [9] Herwindita, Altodia. "Implementasi dan Analisis Game sebagai Layanan (GaaS) menggunakan Open-Source Cloud Gaming Server GamingAnyWhere," Universitas Telkom, Fakultas Teknik Elektro, 2014.
- [10] GamingANyWhere webpage, 2015. <https://www.gaminganywhere.org>
- [11] [http://www.game-debate.com/games/index.php?g\\_id=43&game=Lego:%20Batman](http://www.game-debate.com/games/index.php?g_id=43&game=Lego:%20Batman), [accessed juli 2015]
- [12] [http://www.game-debate.com/games/index.php?g\\_id=4846&game=Deadpool](http://www.game-debate.com/games/index.php?g_id=4846&game=Deadpool), [accessed juli 2015]

