

IMPLEMENTASI GAME SEBAGAI LAYANAN (GAAS) MENGGUNAKAN OPEN-SOURCE CLOUD GAMING SERVER GAMINGANYWHERE

Altodia Herwindita U¹, Tody Ariefianto W, ST., MT.², M. Iqbal, ST., MT.³

^{1,2,3} Fakultas Teknik Elektro, Telkom University, Bandung

Jalan Telekomunikasi, Dayeuh Kolot Bandung 40257 Indonesia

¹altodia.h@gmail.com ²tody.wibowo@gmail.com ³iqbal@tans.ac.id

Abstrak

Cloud computing sangat berkembang akhir-akhir ini sebagai *software* sebagai layanan (SaaS), *platform* sebagai layanan (PaaS) maupun infrastruktur sebagai layanan (IaaS). Menonton video, mendengarkan music, maupun bermain *game* dapat kita lakukan tanpa kita harus memiliki infrastruktur yang memenuhi untuk melakukan semua itu berkat adanya *cloud computing*. *Game* sebagai layanan (GaaS) adalah sebuah layanan *cloud computing* yang merupakan pengembangan dari Infrastruktur sebagai layanan pada model layanan *cloud computing*.

Pada Tugas Akhir ini diimplementasikan sebuah game sebagai layanan cloud gaming server menggunakan open-source GamingAnywhere yang diakses oleh sebuah atau beberapa client melalui jaringan lokal kabel (Ethernet) atau jaringan lokal 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 mendekodekan frame-frame yang diterima sehingga user bisa bermain dan melakukan input control. Control dari *client* yang merupakan *input* dari keyboard, mouse, maupun joystick dari user akan di transmisikan ke *server cloud*. Untuk mendapatkan performa dari sistem cloud game yang diimplementasikan ini, dilakukan pengujian dengan tiga jenis game dan beberapa skenario pengujian serta dilakukan pengujian Quality of Experience (QoE).

Bermain game dengan menggunakan server render terbukti lebih efisien dalam penggunaan resource seperti CPU, RAM dan GPU. Untuk penggunaan CPU hanya digunakan maksimal 25%, untuk RAM maksimal 150 MB, GPU Memory maksimal sekitar 65 MB, GPU Usage dibawah 20% dan didapatkan frame rate diatas 25 fps.

Keyword : GamingAnywhere, GaaS, Cloud computing

Abstract

Cloud computing had been developing lately as software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS). Watching videos, listening to music, even playing game can be done without having a required infrastructure thanks to cloud computing. Game as a service (GaaS) is an developed cloud computing model based on infrastructure as a service model.

On this final project, game as a service cloud gaming server implemented using open-source GamingAnywhere accessed by single or multi user by wire local network (Ethernet) or wireless local network (WLAN). Server capture the audio and video frame from the game then encode those frame and deliver those frame to the client. After the client receive the A/V frame, client will decode received frame so the user could play the game and input the control. Control from client is some inputs from keyboard, mouse and joystick will be transmitted to the server cloud. To obtain implemented cloud game system performance, some experiment conducted using three kinds of game and some scenarios along with Quality of Experience experiment to some respondent.

Playing game using server render proved more efficient on resource usage compared playing game using self-render as CPU, RAM and GPU. On CPU Usage maximum 25%, on RAM maximum 150 MBs, GPU Memory maximum 65 MBs, GPU Usage below 20% and 25 fps on frame rate.

Keyword : GamingAnywhere, GaaS, Cloud computing

1. Pendahuluan

Bermain *game* adalah salah satu aktivitas favorit sebagian orang, mulai dari anak-anak sampai dewasa, laki-laki dan perempuan. Seiring berkembangnya teknologi, berkembang pula kecanggihan *game*. Semakin canggih *game* maka kebutuhan minimum perangkat yang dibutuhkan untuk bermain *game* pun semakin tinggi. Untuk memenuhinya maka harus dilakukan *upgrade* pada *Central Processor Unit*, *Graphic Processor Unit*, RAM, dan menambah media penyimpanan. Tentu tidak murah biaya untuk melakukan *upgrade* tersebut. Untuk mengatasi hal tersebut, maka dilakukan terobosan yaitu dibangun sebuah *Cloud Gaming* berdasarkan *cloud computing* menggunakan model *Infrastructure as a Service* (IaaS).

2. Cloud Computing

2.1 Model Layanan Cloud Computing

Terdapat tiga jenis model layanan cloud computing dasar, yaitu :

a. Infrastructure as a Service (IaaS)

Layanan ini menawarkan resource teknologi dasar, meliputi media penyimpanan, pemrosesan grafik, memory, sistem operasi, kapasitas jaringan dan lain-lain, yang dapat digunakan oleh pengguna layanan cloud untuk menjalankan aplikasi dari user maupun dari penyedia layanan. Untuk menjalankan aplikasi tersebut, pengguna cloud menginstall aplikasi perangkat lunak ke dalam infrastruktur cloud. Salah satu penyedia IaaS ini adalah Amazon EC2 (Elastic Computing Cloud). Layanan Amazon EC2 ini menyediakan berbagai pilihan peripheral seperti CPU, GPU, media storage dan sistem operasi.

b. Platform as a Service (PaaS)

PaaS adalah layanan yang menyediakan modul atau *library* siap pakai yang dapat digunakan untuk mengembangkan sebuah aplikasi yang hanya dapat digunakan untuk mengembangkan sebuah aplikasi yang hanya dapat berjalan di atas platform tersebut. PaaS tidak memungkinkan pengguna layanan untuk melakukan konfigurasi atau kendali terhadap resource memory, storage, processor dan lain-lain. Salah satu penyedia PaaS adalah Google AppEngine yang menyediakan pengembangan

aplikasi diatas platform Google menggunakan bahasa pemrograman Java dan Phython.

c. Software as a Service (SaaS)

Pada jenis layanan ini, penyedia layanan cloud menyediakan aplikasi perangkat lunak pada cloud dan pengguna cloud dapat mengakses aplikasi dari cloud client. Pengguna tidak dapat mengatur infrastruktur dan platform cloud dimana aplikasi yang diakses pengguna berjalan. Pengguna dapat menggunakan SaaS dengan cara berlangganan atau pay-per-use sehingga pengguna tidak perlu melakukan instalasi dan pembaruan aplikasi perangkat lunak. Salah satu penyedia SaaS adalah IBM smart cloud, pada IBM smart cloud ini pengguna dapat melakukan collaborative meeting, mengedit document, berkirim email.

Selain 3 jenis layanan cloud diatas masih ada jenis lain yang merupakan pengembangan dari jenis-jenis diatas, seperti Network as a Service (NaaS), Communication as a Service (CaaS), Gaming as a Service (GaaS), dan akan lebih banyak pengembangannya di masa mendatang.

2.2 Sistem Cloud Gaming

Sistem cloud gaming atau secara umum real-time remote rendering sistem dibagi dalam tiga kategori, yaitu :

a. 3D Graphic Streaming

Pada 3D graphic streaming, server cloud mengintersep *graphic command*, mengkompres *command*, kemudian stream *command* tersebut kepada client. Client merender grafik game menggunakan chip grafik berdasar grafik *command* seperti OpenGL dan Direct3D. Grafik chip milik client harus kompatibel dengan *command* grafik yang di stream dan cukup kuat untuk merender grafik game pada kualitas tinggi dan real-time. 3D graphic streaming tidak menggunakan GPU pada server cloud untuk melakukan render grafik game. Sistem ini tidak cocok digunakan pada client yang memiliki resource yang terbatas, seperti mobile device.

b. Video Streaming

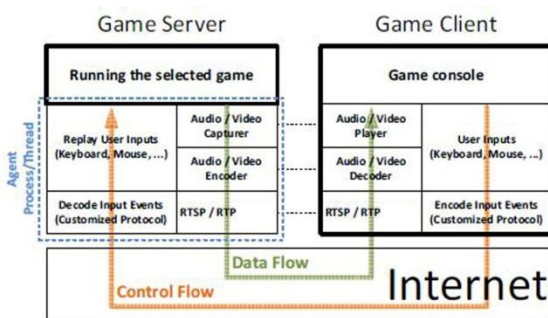
Dengan pendekatan video streaming, server cloud merender grafik 3D menjadi video 2D,

mengompres video kemudian stream video tersebut kepada client. Client mendekodekan dan menampilkan video stream. Pendekodean dapat dilakukan menggunakan video dekoder yang umum diproduksi. Sistem ini tidak memberatkan client untuk melakukan rendering grafik 3D secara intensif dan cocok untuk client yang memiliki keterbatasan resource.

c. Video Streaming dengan operasi post-rendering

Sistem video streaming dengan operasi post-rendering berada diantara 3D graphic streaming dengan video streaming, pada saat 3D graphic stream merender pada server cloud, operasi post-rendering bekerja secara opsional pada thin client untuk mengaugmentasi gerakan, pencahayaan dan tekstur. Operasi post-rendering ini mempunyai kompleksitas komputasi yang rendah dan dapat berjalan pada real-time tanpa GPU.

2.3 Arsitektur Sistem GamingAnywhere



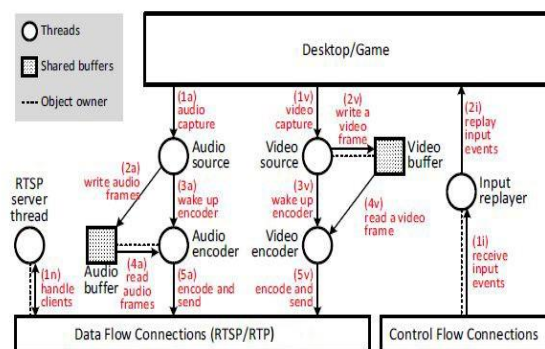
Gambar 1. Hubungan antara Server dan Client

Pada sistem cloud game GamingAnywhere terdapat 2 jenis network flow yaitu data flow dan control flow. Data flow digunakan untuk stream video dan audio dari server ke *client*, sedangkan control flow digunakan untuk mengirimkan perintah dari *client* ke server. 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.

2.4 Cloud Game Server

Pada saat agen dijalankan, keempat modul dari agen tersebut (RTSP Server, Video Source, Audio Source dan Input Replayer) ikut dijalankan. RTSP Server dan Input Replayer menunggu client yang akan mengakses, sedangkan Video dan Audio Source berada dalam posisi idle setelah inisiasi. Saat client terhubung ke RTSP Server, modul enkoder baik Video Encoder maupun Audio Encoder dijalankan dan masing enkoder memberikan notifikasi pada modul yang berkaitan untuk memulai meng-capture frame audio maupun video. Sistem kerja enkoder bersifat real-time, artinya enkoder melakukan pengkodean frame yang di-capture secara real-time.



Gambar 2. Hubungan antar modul pada Cloud Gaming Server

2.4.1 RTSP and RTP Server

RTSP server adalah thread yang pertama kali dijalankan pada agen RTSP server menerima perintah RTSP dari client kemudian menjalankan enkoder dan men-setup data flow untuk pengiriman frame audio maupun video yang sudah terkodekan. Perintah RTSP yang dikirimkan oleh client diterima RTSP server melalui protokol TCP. Sedangkan frame audio dan video yang terkodekan dikirimkan menggunakan protokol RTP melalui UDP.

RTSP server mengekspos sebuah antarmuka bagi enkoder untuk mengirim frame yang terkodekan sehingga pada saat enkoder membangkitkan frame yang

sudah dikodekan, maka langsung dapat dikirim kepada client.

2.4.2 Video Source

Untuk meng-capture tampilan game, dapat dilakukan dengan dua cara. Pertama, disebut *desktop capture module*, dimana seluruh tampilan *desktop* di-capture pada *rate* yang ditentukan. Kedua, disebut *API intercept module*, dimana dilakukan intersep pada *graphic drawing function* dari sebuah game. *Desktop capture module* aktif mengambil *screenshot* pada *desktop* dengan frekuensi yang ditentukan. Contohnya, jika *frame rate* yang dikehendaki adalah 30 fps, maka interval untuk meng-capture adalah 1/30 detik kurang lebih 33.33 ms. Sedangkan cara kerja *API intercept module* adalah pada saat sebuah game menyelesaikan render dari sebuah tampilan di *back buffer* sebuah *game*^[1], *API intercept module* mempunyai kesempatan untuk meng-capture tampilan game. Contohnya, jika dikehendaki *frame rate* sebesar 50 fps, sedangkan tampilan game diupdate sekitar 100 kali per detik, maka *API intercept module* akan meng-capture satu tampilan game tiap dua detik perubahan tampilan.

Setiap *frame* yang di-capture diberikan *timestamp* disimpan pada *shared buffer* yang dimiliki oleh modul *video source* dengan *video encoder*. *Video source* sebagai *writer* pada *buffer* dan *video encoder* sebagai *reader*. Digunakan ekstensi GDI (Graphics Device Interface) untuk meng-capture *desktop* pada Windows. GDI adalah sebuah API dari Microsoft Windows yang mempunyai fungsi untuk menggambarkan obyek graphical dan mengirimkan ke periperal output seperti monitor. Untuk *API intercept interface module*, mendukung game DirectDraw, Direct3D dan SDL dengan *hooking* DirectX API pada Window. Kedua modul ini mendukung pixel berformat RGBA, BGRA dan YUV420p.

2.4.3 Audio Source

Digunakan library Windows audio session API (WASAPI) untuk meng-capture suara pada Windows. Audio source module rutin meng-capture frame audio dari perangkat audio (waveform output device). Frame yang dicapture dikopikan ke buffer shared dengan encoder oleh audio source module. Encoder akan aktif jika setiap frame audio dicapture oleh audio source module.

2.4.4 Frame Encoding

Frame audio dan video dikodekan oleh modul enkoder yang berbeda pada saat ada client yang terhubung. Modul enkoder audio dan video diimplementasikan menggunakan library libavcodec yang merupakan project dari ffmpeg.

Untuk enkoder audio, GamingAnywhere menggunakan enkoder libmp3lame. Libmp3lame adalah enkoder audio mengkodekan raw audio menjadi MP3 (Motion Picture Expert Group 1 atau 2 Audio Layer III). MP3 adalah sebuah format koding audio untuk audio digital yang menggunakan lossy data compression. Libmp3lame tersedia pada libavcodec dengan konfigurasi yang bisa disesuaikan oleh user. Konfigurasi enkoder yang digunakan untuk Tugas Akhir ini adalah sebagai berikut :

Audio Bit Rate	128 Kbps
Audio Sample Rate	44.1 KHz
Audio Channel	2 (Stereo)

Tabel 1. Konfigurasi Enkoder Audio.

Untuk enkoder video, GamingAnywhere menggunakan enkoder x264. X264 merupakan library untuk mengkodekan raw video menjadi H.264/MPEG-4 AVC. H.264/MPEG-4 AVC (Advanced Video Coding) adalah sebuah standar format kompresi video yang dipublikasi oleh ITU-T Video Coding Expert Group berkolaborasi dengan ISO/IEC JTC1 Moving Picture Expert Group (MPEG). Enkoder x264 pun termasuk dalam library libavcodec dan bisa dikonfigurasi sesuai keinginan user. Konfigurasi yang digunakan dalam Tugas Akhir ini adalah sebagai berikut :

Video Bit Rate	3000 Kbps
Video-specific method	Diamond
Video-specific GoP Size	48
Video-specific thread	4
Video-specific profile	main
Video-specific preset	veryfast
Video-specific tune	zerolatency

Tabel 2. Konfigurasi Enkoder Video

2.4.5 Input Replayer

Modul ini mempunyai dua tugas yaitu, meng-*capture input event* pada *client* dan mengulangi input event pada *client* sebagai input event pada server. Tidak seperti frame audio dan video, input event control dikirim melalui koneksi yang terpisah melalui UDP meskipun dapat digunakan koneksi RTSP untuk mengirim input events dari *client* ke server, tidak dilakukan karena :

- 1) Pengiriman input event dapat mengalami delay karena paket lain, seperti paket RTCP yang menggunakan koneksi RTSP.
- 2) Pengiriman data melalui RTSP mengalami delay panjang karena RTSP adalah text-based yang notabene memakan waktu yang cukup lama.
- 3) Tidak ada standar untuk meng-*embed* input event pada koneksi RTSP.

Setelah menerima input event, input handling module menrubah event yang iterima menjadi format yang sesuai dengan server dan mengirim event-structure ke server. GamingAnywhere mengulangi input event menggunakan fungsi SendInput pada Windows.

2.5 Cloud Gaming Client

Pada dasarnya, client adalah sebuah *remote desktop client* yang menampilkan tampilan game secara real-time yang dicapture pada server kemudian dikirimkan dengan bentuk frame audio dan video yang terkodekan oleh encoder pada server. Agar client dapat menerima frame audio serta video dari server, maka pada client harus terdapat GamingAnywhere. Pada client terdapat dua buah modul, tugas modul pertama adalah untuk mengangani *input event* dari user (jalur i). Tugas modul kedua adalah untuk melakukan render audio dan video (jalur r).

2.5.1 RTSP dan RTP Client

Pada client GamingAnywhere digunakan *library live555* untuk menangani komunikasi jaringan. GamingAnywhere memanfaatkan *class framework* RTSPClient dan MediaSink dari *library live555*. Setelah RTSPClient berhasil membangun sesi audio dan video, MediaSink menangani frame-frame audio dan video tersebut yang dikirim dari server.

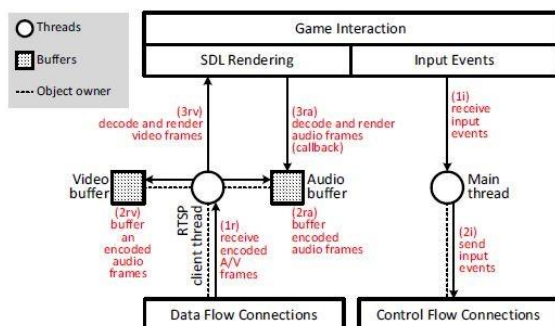
2.5.2 Frame Buffering dan Decoding

Dekoder video tidak melakukan buffer terhadap frame video sama sekali. Komponen video buffer digunakan untuk menampung frame video yang sama dengan frame video terbaru yang diterima dari server.

Berbeda dengan frame video, pada saat RTSP Client menerima frame audio, modul RTSP Client tidak melakukan decoding melainkan menempatkan semua frame audio yang diterima pada sebuah buffer. Hal ini dikarenakan render audio SDL diimplementasikan menggunakan *on-demand approach* yaitu untuk memutar frame audio pada SDL sebuah fungsi *callback* harus diregistrasikan dan dipanggil. Pada saat modul RTSP Client menerima frame audio baru, modul tersebut akan menambahkan frame pada buffer dan memicu fungsi *callback* untuk membaca frame audio.

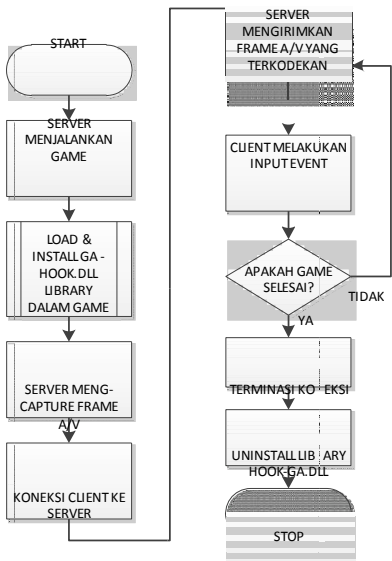
2.5.3 Input Handling

Modul Input Handling pada client mempunyai dua tugas utama, yaitu meng-*capture input event* yang dilakukan oleh pemain dan meneruskannya ke server. GamingAnywhere mendukung mekanisme untuk SDL (Simple Direct Layer) *event capturing*. SDL adalah sebuah *library cross-platform* yang didesain untuk menyediakan akses *low-level* pada audio, keyboard, mouse dan joystick serta hardware grafik via OpenGL dan Direct3D.

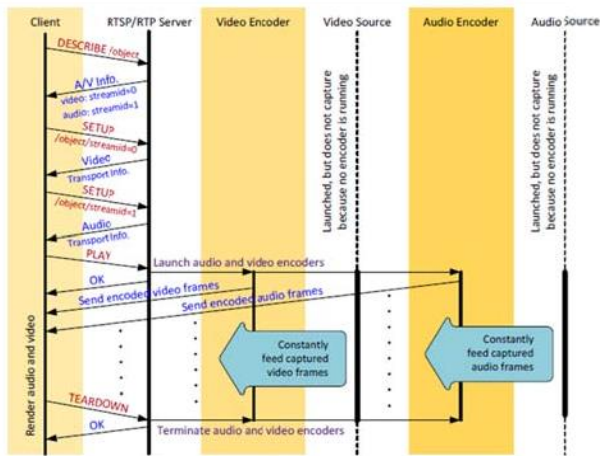


Gambar 3. Hubungan antar Modul pada Client

2.6 Diagram Alir



Gambar 4. Flowchart Sistem Cloud Gaming

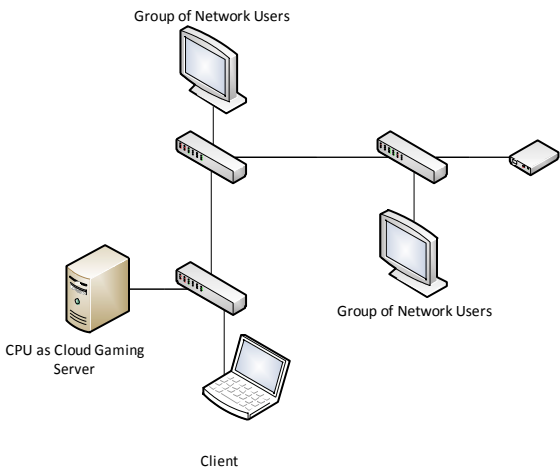


Gambar 5. UML Diagram RTSP/RTP Protokol Pada GamingAnywhere

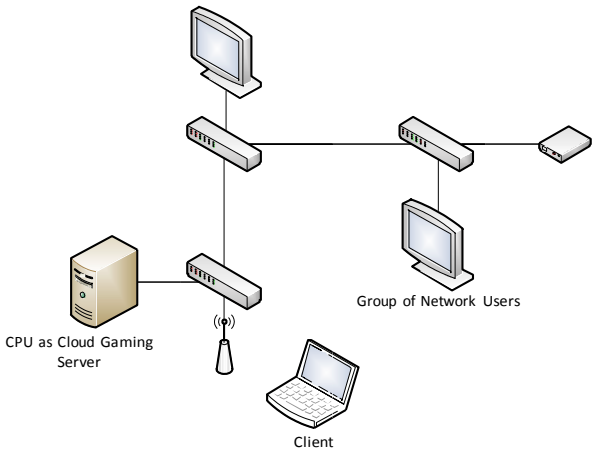
3. Perancangan Sistem

3.1 Arsitektur Sistem

Sistem Cloud Gaming pada Tugas Akhir ini diimplemntasikan pada rumah kos Anerfa dengan topologi sebagai berikut



Gambar 6 Arsitektur implementasi sistem pada jaringan lokal berkabel



Gambar 7. Arsitektur implemetasi sistem pada jaringan lokal nirkabel

3.2 Jenis Game

Terdapat tiga buah game yang digunakan untuk menguji sistem pada gaming cloud ini, antara lain :

1. LEGO Batman
LEGO Batman dirilis pada September 2008 dikembangkan oleh Traveller’s Tales dan dipublikasikan oleh Warner Bros. Game ini bergenre Action-Adventure. Pada game ini, pemain beraksi sebagai Batman atau Robin melawan kriminal dan penjahat di Kota Gotham.

Spefikasi minimum yang dibutuhkan untuk

menjalankan game ini adalah^[8] :

Tabel 3. Spesifikasi Minimum LEGO Batman

2. Deadpool

Deadpool dirilis pada Juni 2013 dikembangkan oleh High Moon Studios dan dipublikasikan oleh Activision. Game ini mempunyai genre Third Person Shooter/Adventure. Pada game ini, pemain bertindak sebagai Deadpool. Deadpool adalah karakter di X-Men yang merupakan ciptaan Marvel. Pada game ini disajikan fitur *hack n slash* dimana pemain dituntut untuk bertindak dengan cepat menghabisi lawan.

Spesifikasi minimum yang dibutuhkan untuk menjalankan game ini adalah^[9] :

CPU	Intel Core2Duo E8200 2.66 GHz, AMD Phenom 8750 Triple Core
GPU	Nvidia GeForce 8800 GTS, AMD Radeon HD 4850
RAM	2 GB
HDD	7 GB

Tabel 4. Spesifikasi Minimum Deadpool

3. Total War : Shogun 2

Total War : Shogun 2 dirilis pada Maret 2011 dikembangkan oleh Creative Assembly dan dipublikasikan oleh Sega. Game ini bergenre Real-Time Strategy. Game ini mengambil setting pada jaman abad ke enam belas di Jepang. Pemain diharuskan mengatur sumber daya untuk kelangsungan klan, membuat pasukan, memerintahkan pasukan untuk bertahan maupun menyerang. Game ini tidak dibutuhkan aksi yang cepat dari pemain.

Spesifikasi minimum yang dibutuhkan untuk menjalankan game ini adalah^[10] :

CPU	Intel Core2Duo E6700 2.66 GHz, AMD Athlon 64 X2 Dual Core 5800+
GPU	Nvidia GeForce GT530, AMD Radeon HD 2900 XT 512MB
RAM	4 GB
HDD	20 GB

Tabel 5. Spesifikasi minimum Total War : Shogun 2

3.3 Spesifikasi Sistem

CPU	Intel Pentium 4 3.0GHz, AMD Athlon 64 3200+
GPU	Nvidia GeForce 6600, AMD Radeon X600
RAM	1GB
HDD Space	4.2 GB

Sistem Cloud Gaming menggunakan GamingAnywhere yang merupakan open-source cloud gaming server diimplementasikan pada sebuah CPU yang mempunyai spesifikasi sebagai berikut :

CPU	AMD Phenom II X4 945 3.30 GHz
RAM	6GB
GPU	AMD Radeon XFX HD6770 DDR5 1GB
Sistem Operasi	Microsoft Windows 7 64-bit

Tabel 6. Spesifikasi Cloud Game Server

Untuk client digunakan pada laptop yang mempunyai spesifikasi sebagai berikut :

Laptop	Samsung Notebook Series 3
CPU	Intel Celeron CPU B820 1.70 GHz
RAM	2 GB (1.78 GB Usable)
GPU	Intel HD Graphics / Nvidia GT610M
Sistem Operasi	Microsoft Windows 7 32-bit

Tabel 7. Spesifikasi Cloud Game Client

3.4 Skenario Pengujian

Skenario A : pengujian terhadap sistem menggunakan tiga jenis game (game tidak berjalan bersamaan) untuk mengetahui performa *client* pada saat client bermain game dengan self-render dibandingkan dengan saat client bermain menggunakan sistem cloud game (server render). Prosedur pengujian ini adalah sebagai berikut :

1. *Client* melakukan self-render dengan menggunakan tiga jenis game (game tidak berjalan bersamaan) dengan variable resolusi tampilan game yaitu 1024x768, 1280x720 dan 1360x768.
2. Dilakukan pengujian pada saat client melakukan server render menggunakan tiga jenis game (game tidak berjalan bersamaan) dengan variable resolusi tampilan game yaitu 1024x768, 1280x720 dan 1360x768 dengan bandwidth unlimited pada jaringan kabel (Ethernet) dan pada jaringan nirkabel (WLAN).

3. Dilakukan pengujian saat client melakukan server render menggunakan tiga jenis game (game tidak berjalan bersamaan) dengan variable bandwidth jaringan yaitu 1.5 Mbps, 3 Mbps, 5 Mbps dan Unlimited pada resolusi 1280x720 pada jaringan kabel (Ethernet) dan pada jaringan nirkabel (WLAN).

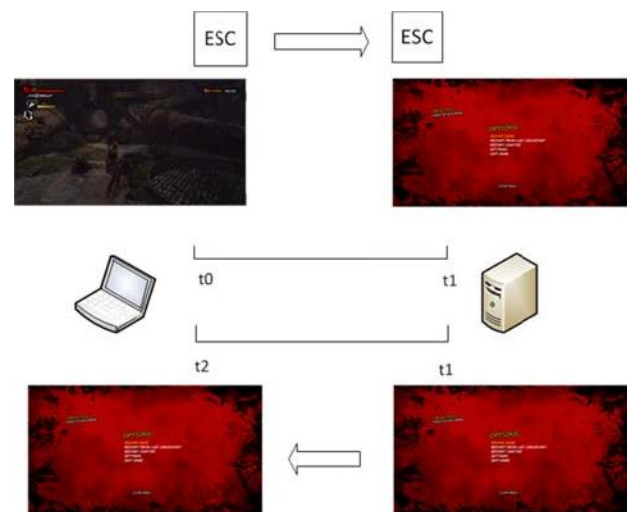
Skenario B : pengujian terhadap sistem

menggunakan tiga jenis game (game tidak berjalan bersamaan) untuk mengetahui performa server pada saat server bermain game dengan GamingAnywhere tidak aktif dan saat aktif, perbandingan performa apabila server diakses oleh single-user dan multi-user (2 & 3 client (client ke-2 dan ke-3 hanya sebagai pengamat). Prosedur pengujian ini adalah sebagai berikut :

1. Server menjalankan tiga jenis game (game tidak berjalan bersamaan), dengan GamingAnywhere tidak aktif pada resolusi 1280x720
2. Server menjalankan tiga jenis game (game tidak berjalan bersamaan), dengan GamingAnywhere aktif dengan n client terhubung (n = 0,1,2,3) dengan resolusi 1280x720 dan bandwidth Unlimited pada jaringan kabel (Ethernet)
3. Server menjalankan tiga jenis game (game tidak berjalan bersamaan), dengan GamingAnywhere aktif dengan n client terhubung (n = 0,1,2,3) dengan resolusi 1280x720 dan bandwidth Unlimited pada jaringan nirkabel (WLAN)

Skenario C : pengujian terhadap sistem menggunakan menggunakan tiga jenis game (game tidak berjalan bersamaan) untuk mengetahui delay respon sistem dan throughput pada jaringan kabel dan jaringan nirkabel. Prosedur pengujian ini adalah sebagai berikut :

1. Client melakukan server render pada setiap game dengan resolusi 1280x720 dan bandwidth jaringan Unlimited pada jaringan kabel (Ethernet)



Gambar 8. Responsive Delay

Responsive Delay = Control Delay (t0-t1) + Video Delay (t1-t2)

3.5 Quality of Experience

Dalam pengujian ini, ada tiga parameter yang dinilai oleh responden pada saat responden bermain game di sistem ini, yaitu :

1. **Responsiveness**, tingkat kepuasan pengguna dalam hal tingkat responsif sebuah game, seberapa cepat respon sistem untuk menanggapi aksi yang diberikan oleh pengguna. *Responsiveness* ini dinilai oleh responden dalam skala 1 sampai 5, dengan skala 1 adalah Tidak Puas dan skala 5 adalah Sangat Puas.
2. **Smoothness**, tingkat kepuasan pengguna dalam hal tingkat kelancaran sebuah game, seberapa lancar game yang berjalan dalam *Cloud Gaming System* ini, apakah bersifat *laggy* (frame rate rendah) atau tidak. *Smoothness* ini dinilai oleh responden dalam skala 1 sampai 5, dengan skala 1 adalah Tidak Puas dan skala 5 adalah Sangat

2. Client melakukan server render pada setiap game dengan resolusi 1280x720 dan bandwidth jaringan Unlimited pada jaringan nirkabel (WLAN)
3. **Graphic Quality**, tingkat kepuasan pengguna dalam hal tingkat kualitas grafik yang ditampilkan pada komputer *client*, bagaimana kualitas grafik yang ditampilkan ketika responden bermain game pada *Cloud Gaming System* ini. *Graphic Quality* dinilai oleh responden dalam skala 1 sampai 5, dengan skala 1 adalah Tidak Puas dan skala 5 adalah Sangat Puas.

4. Analisis Hasil Pengujian

4.1 Pada Client

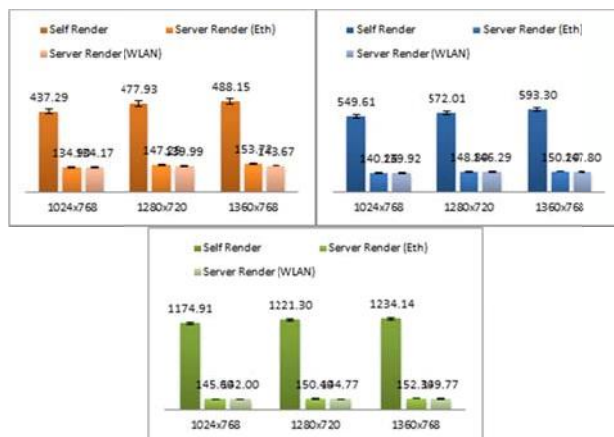
a. CPU Usage



Gambar 9. Grafik CPU Usage

Terlihat pada grafik diatas bahwa penggunaan *resource* khususnya CPU Usage pada Server Render baik pada jaringan kabel (Ethernet) maupun jaringan nirkabel (WLAN) rendah dibandingkan pada Self Render (bar pertama) sehingga user tidak perlu khawatir terjadi overheat pada prosesor karena penggunaan prosesor yang tinggi. Resolusi yang berbeda tidak mempengaruhi secara signifikan terhadap penggunaan prosesor.

b. Physical Memory



Gambar 10. Grafik Physical Memory

Pada grafik di atas dapat dilihat bahwa penggunaan Physical Memory atau RAM pada client pada saat bermain game menggunakan sistem cloud

menjalankan aplikasi lain. Perbedaan resolusi tidak mempengaruhi penggunaan RAM secara signifikan.

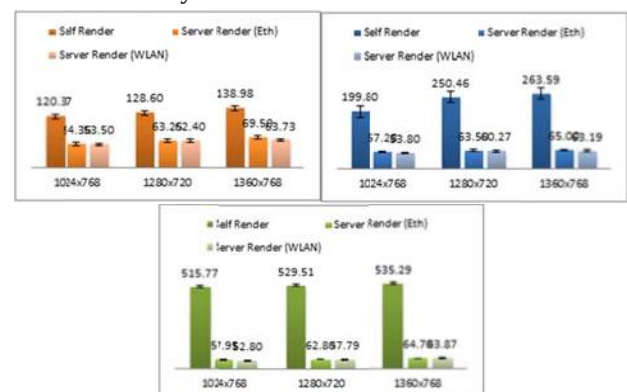
c. GPU Usage



Gambar 11. GPU Usage

Pada grafik di atas dapat dilihat bahwa perbandingan penggunaan *Graphic Processor Unit* pada Server Render baik pada jaringan kabel maupun jaringan nirkabel signifikan rendah dengan Self Render. Resolusi yang semakin besar menyebabkan penggunaan GPU semakin rendah. Hal ini berhubungan dengan *frame rate* yang diproses. Semakin rendah angka *frame rate*, semakin kecil penggunaan GPU.

d. GPU Memory



Gambar 12. Grafik GPU Memory

Pada Gambar 12 perbandingan penggunaan memori GPU pada Server Render baik jaringan kabel maupun jaringan nirkabel relatif rendah dibandingkan dengan Self Render. Angka yang dihasilkan pada Server Render relatif sama pada setiap *game*.

game terjadi perbedaan yang signifikan antara Self

Render dengan Server Render baik pada jaringan lokal maupun jaringan nirkabel. Karena penggunaan RAM yang rendah, bisa dimungkinkan pada client untuk

e. Frame Rate



Gambar 13. Grafik Frame Rate

Pada grafik berwarna orange (LEGO Batman) diatas bahwa angka frame rate pada *Self Render* tinggi, hal ini menunjukkan bahwa *client* dapat mengatasi game LEGO Batman dengan mudah di angka resolusi

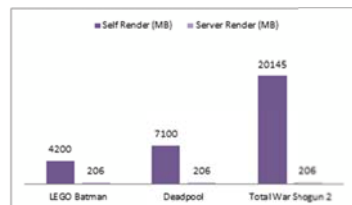
berapapun. Hal ini dikarenakan GPU pada *client* cukup mumpuni untuk mengatasi game LEGO Batman sampai

pada grafik berwarna biru (Deadpool) pada resolusi 1280x720, *frame rate client* tersaingi oleh *Server Render*

baik pada jaringan kabel maupun jaringan nirkabel. Frame rate yang rendah menyebabkan penggunaan GPU

yang rendah pula.

f. Storage



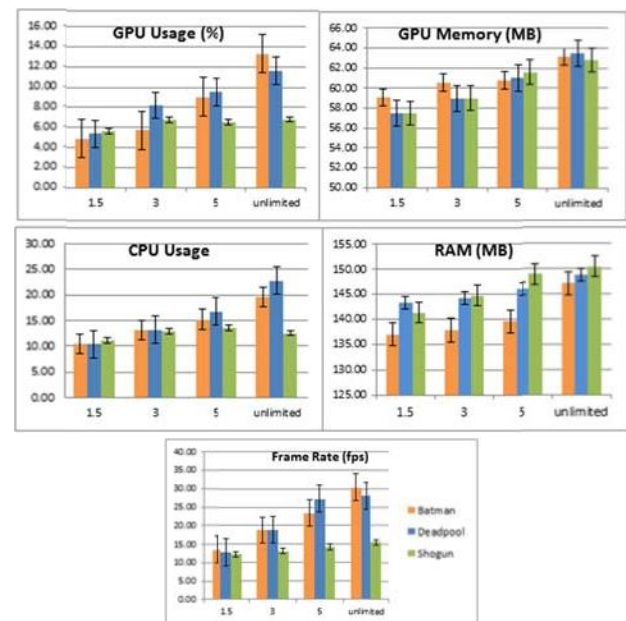
Gambar 14. Grafik Storage

Untuk bermain game, *client* harus menginstal game yang akan dimainkan sehingga dibutuhkan penyimpanan pada *harddisk* sesuai yang dibutuhkan oleh game. Pada *cloud game system*, instalasi game tidak dibutuhkan karena

semua *resource* game tersimpan pada *server cloud game*. Sebesar 206 Megabyte pada *client* dibutuhkan untuk

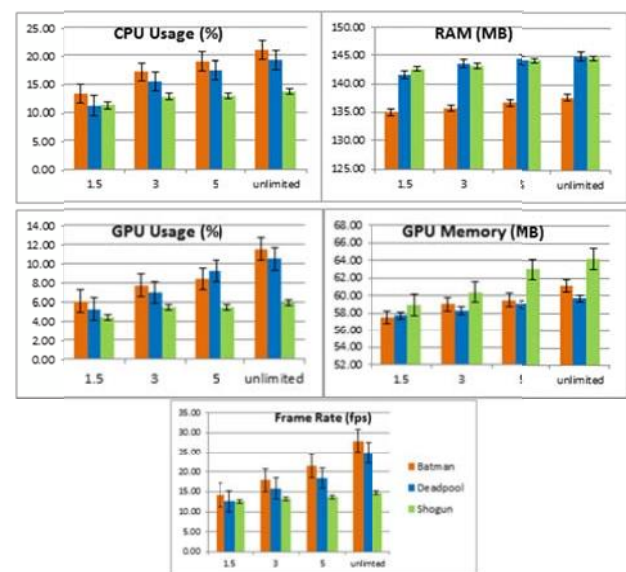
menyimpan program *client* GamingAnywhere.

g. Bandwidth Variabel



Gambar 15. Performa Client dengan Variabel Bandwidth pada Ethernet

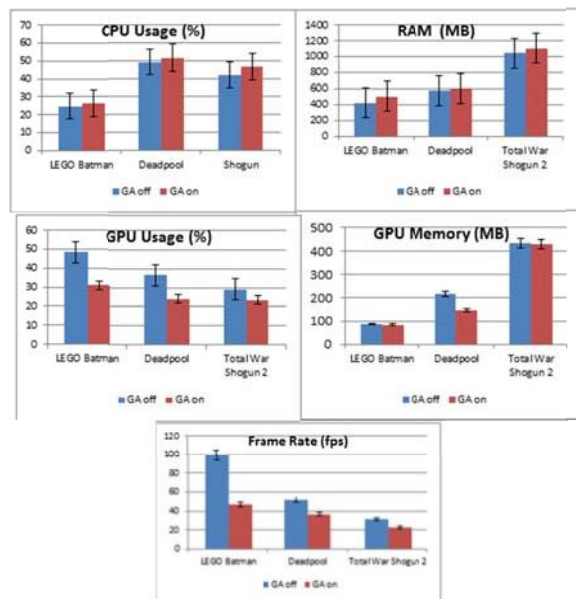
Grafik diatas adalah performa client pada saat bermain game dengan Server Render dengan variabel bandwidth. Dapat dilihat bahwa untuk setiap kenaikan bandwidth terjadi juga kenaikan grafik.



Gambar16. Performa Client dengan Variabel Bandwidth pada WLAN

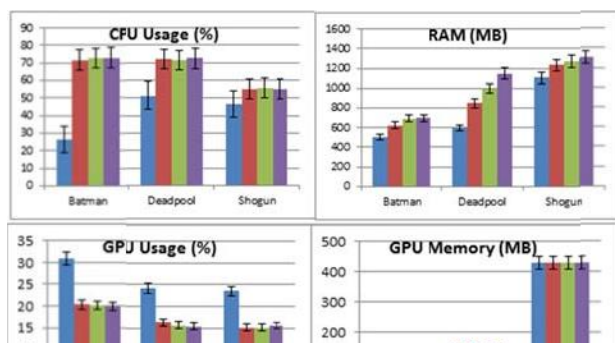
Pada jaringan nirkabel pun dengan variasi bandwidth, terjadi kenaikan grafik untuk setiap kenaikan variasi bandwidth.

4.2 Pada Server



Gambar 17. Grafik Performa Server pada saat GA aktif dan nonaktif

Di atas adalah grafik perbedaan performa pada resource ketika server dengan GamingAnywhere aktif dan ketika GamingAnywhere nonaktif. Pada CPU Usage dan RAM terlihat grafik mengalami kenaikan pada saat GamingAnywhere diaktifkan, hal ini dikarenakan modul-modul pada GamingAnywhere mulai berjalan. Lainnya dengan grafik GPU Usage dan Frame Rate, grafik tersebut mengalami penurunan setelah GamingAnywhere diaktifkan, frame rate menurun dikarenakan modul peng-capture frame video dan audio mulai berjalan dengan frame rate maksimal 50 fps yang menyebabkan GPU Usage juga menurun. Sedangkan pada grafik GPU Memory terjadi penurunan nilai yang sangat kecil tergantung pada *screen game* yang di render.



Gambar 18. Grafik Performa Server dengan n Client

Grafik diatas menggambarkan performa server saat n ($n = 0, 1, 2, 3$) client terhubung pada server

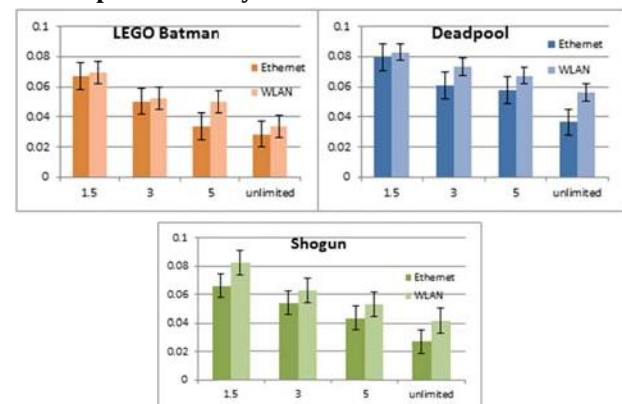
melakukan streaming game. Client ke-2 dan ke-3 hanya sebagai observer tetapi dapat melakukan input event pada server. Terjadi kenaikan penggunaan CPU pada saat

client pertama terkoneksi pada server namun relatif tidak berubah setelah client kedua dan ketiga terkoneksi. Hal ini dikarenakan GamingAnywhere menggunakan mode *one-encoder-for-all*. Pada penggunaan RAM, terjadi kenaikan setiap client yang terkoneksi. Sedangkan pada penggunaan GPU dan Frame Rate terjadi penurunan

setelah terdapat client yang terkoneksi, namun akan

cenderung konstan untuk client kedua dan client ketiga terkoneksi. Sedangkan untuk GPU Memory tidak mengalami perubahan saat berapapun client yang terkoneksi.

4.3 Responsive Delay



Gambar 19. Grafik Responsive Delay

Grafik diatas menggambarkan delay respon oleh sistem pada jaringan kabel dan jaringan nirkabel. Terlihat bahwa pada jaringan kabel nilai delay lebih kecil dibandingkan pada jaringan nirkabel karena pada

jaringan nirkabel menggunakan sistem komunikasi half-duplex dibandingkan pada jaringan kabel yang

menggunakan full-duplex. Respon delay dengan nilai dibawah 0.1 detik pada setiap game cukup memuaskan user, hal ini dibuktikan dengan penilaian dengan QoE dengan nilai 3.2 pada game LEGO Batman, 3.6 pada game Deadpool dan 3.6 pada game Total War : Shogun 2.

4.4 Hasil Quality of Experience

Game	Avg Responsiveness	Avg Smoothness	Avg. Graphic Quality
LEGO Batman	3.2	3.6	3.4
Deadpool	3.6	3	3.8
TW : Shogun 2	3.6	3.5	2.4

Tabel 8. Hasil QoE

5. Kesimpulan

Berdasarkan pengujian dan pengukuran serta analisis hasil pengujian yang dilakukan, maka didapatkan kesimpulan sebagai berikut.

1. Bermain game menggunakan system cloud gaming (Server Render) terbukti lebih efisien dalam penggunaan resource (CPU Usage, Physical Memory, GPU Usage dan GPU Memory). Untuk CPU Usage hanya digunakan sekitar 25%, untuk RAM sekitar 150 MB, untuk GPU Memory sekitar 65 MB, sedangkan untuk frame rate mencapai diatas 25 fps (LEGO Batman dan Deadpool) dan 15 fps (Total War Shogun) dan GPU Usage di bawah 20%
2. Pada saat Server diakses oleh multi-user, penggunaan CPU pada server tidak mengalami kenaikan, namun pada Physical Memory naik per user yang terhubung. Untuk frame rate dan GPU Usage, tidak mengalami perubahan setelah beberapa user terhubung.
3. Responsive Delay pada Ethernet lebih kecil daripada menggunakan WLAN namun performa client dan server tidak jauh berbeda ketika menggunakan Ethernet maupun WLAN.
4. Client hanya membutuhkan sekitar 206 MB untuk menjalankan game dengan Server Render dan tidak perlu instal game sehingga bisa menghemat storage.
5. Dengan nilai rata-rata diatas 3 per kategori dalam QoE dapat disimpulkan bahwa Game as a Service (GaaS) menggunakan GamingAnywhere cukup memuaskan pengguna.

Referensi

- [1] W. Purbo, Onno. (2012). Membuat Sendiri Cloud Computing Server Menggunakan Open Source, Yogyakarta : Penerbit Andi.
- [2] 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.

- [3] 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.

- [4] [http:// en.wikipedia.org/wiki/Cloud_computing](http://en.wikipedia.org/wiki/Cloud_computing) (Diakses pada : 28 Agustus 2013)

- [5] Introduction to Cloud Computing Architecture. White Paper. June 2009

- [6] Chun-Ying Huang, Cheng-Hsin Hsu, De-Yu Chen, and Kuan-Ta Chen, "Quantifying User Satisfaction in Mobile Cloud Games," Proceedings of ACM Multimedia 2013 (Open Source Software Competition, Nov 2013

- [7] Ryan Shea, Jiangchuan Liu, Edith C.-H. Ngal, Yong Cul, "Cloud Gaming : Architecture and Performance," IEEE Network, July 2013

- [8] http://www.game-debate.com/games/index.php?g_id=43&game=Lego:%20Batman (diakses pada 12 Juli 2014)

- [9] http://www.game-debate.com/games/index.php?g_id=4846&game=Deadpool (diakses pada 12 Juli 2014)

- [10] http://www.game-debate.com/games/index.php?g_id=1091&game=Shogun%20202:%20Total%20War (diakses pada 12 Juli 2014)