

## **QUALITY OF SERVICE (QOS) CLOUD GAMINGANYWHERE PADA GAME DENGAN SPEECH RECOGNITION SYSTEM SEBAGAI COMMAND INPUT**

### **THE QUALITY OF SERVICES (QOS) OF GAMINGANYWHERE ON A GAME WITH SPEECH RECOGNITION SYSTEM AS COMMAND INPUT**

Sussi, S.Si., M.T. <sup>1</sup>, Dr. Ir. Rendy Munadi M.T.<sup>2</sup>, Nurwulan Fitriyanti, S.Pd, M.PFis<sup>3</sup>, Indra Perdana Putra Sutejo, S.T.<sup>4</sup>

<sup>1,2,3,4</sup>Fakultas Teknik Elektro, Universitas Telkom

<sup>1</sup>[sussiss@telkomuniversity.ac.id](mailto:sussiss@telkomuniversity.ac.id), <sup>2</sup>[rendymunadi@telkomuniversity.ac.id](mailto:rendymunadi@telkomuniversity.ac.id), <sup>3</sup>[nurwulanf@telkomuniversity.ac.id](mailto:nurwulanf@telkomuniversity.ac.id), <sup>4</sup>[indraperdanaps@student.telkomuniversity.ac.id](mailto:indraperdanaps@student.telkomuniversity.ac.id)

#### **Abstrak**

Perkembangan dunia industri *game* semakin semarak dengan munculnya teknologi jaringan dibidang *Cloud Gaming*. Platform *Cloud Gaming* yang sering digunakan dan sifatnya *open source* yaitu *GamingAnywhere*. Dengan menggunakan *cloud gaming* *GamingAnywhere* dan *platform speech recognition system* *FreeePIE*, *client* dapat memainkan *game* berspesifikasi tinggi pada perangkat miliknya yang berspesifikasi lebih rendah dengan sistem *input* menggunakan perintah suara. Pencinta *game* yang memiliki kerusakan motoris tangan masih bisa menikmati *game* dengan masukan suara.

Penelitian tentang *Cloud Gaming* yang ada masih terbatas, karena teknologi *Cloud Gaming* merupakan teknologi baru (2013). Penelitian ini ditujukan untuk memberikan informasi mengenai *Quality of Service* (QoS) dari *GamingAnywhere*. Dari hasil pengukuran, untuk meraih QoS yang optimal dalam menjalankan *game* dengan *cloud gaming* *GamingAnywhere*, dibutuhkan minimal *bandwidth* sebesar 3 Mbps. Bila *bandwidth* yang diberikan kurang dari 3 Mbps, sistem akan mengalami *delay* yang massif bernilai  $\pm 0.5$  detik pada *game* *NEVERBALL* dan bernilai  $\pm 1.9$  detik pada *game* *7 Days to Die* dan *packet loss* yang dihasilkan pun berkisar 45-90%.

**Kata kunci : GamingAnywhere, Quality of Service, Cloud Gaming**

#### **Abstract**

The development of the game industry is growing faster with the emergence of network technology on cloud gaming. the most used of open source cloud gaming platform is *GamingAnywhere*. By using cloud gaming *GamingAnywhere* and *FreeePIE* speech recognition platform system, the client can play hi-spec games on their unsupported device with voice commands as system input. Game lovers who have hand motor disorder can still enjoy games with voice as an input.

Because of the cloud gaming is a new technology (2013), therefore research on cloud gaming are still few. this research is intended to investigate the *Quality of Service* (QoS) of *GamingAnywhere*. After the measurement process, we conclude that to achieve optimum QoS for running games using *GamingAnywhere*, it require 3 Mbps minimum of bandwidth. If the given bandwidth is less than 3 Mbps, the system will experience a massive delay of  $\pm 0.5$  seconds in the *NEVERBALL* game and  $\pm 1.9$  seconds in the *7 Days to Die* game then the *packet loss* will be 45-90%, in which out of standard quality.

**Keywords: GamingAnywhere, Quality of Service, Cloud Gaming**

## 1. PENDAHULUAN

*Game* (permainan) pada saat ini sudah menjadi salah satu sarana hiburan bagi masyarakat luas di berbagai belahan dunia. Karena kebutuhan yang besar ini, industri *gaming* telah masuk menuju fase *Next Generation Gaming*, dimana kualitas *game* secara keseluruhan sangatlah bagus, diantaranya adalah dari segi grafis dengan tingkat realisme yang tinggi dan pemakaian efek partikel yang kompleks. Tak hanya menyediakan hiburan, terdapat juga *game* yang dibuat sebagai sarana edukasi dan pendidikan sosial, atau dengan *genre* yang disebut dengan *Games With A Purpose* (GWAP)[9].

Walaupun demikian, hampir semua *game* dibuat untuk dimainkan dengan semacam pengendali manual seperti tetikus, *keyboard* *joy stick*. Bagi kebanyakan orang, hal ini tidak menjadi masalah, tetapi untuk sebagian orang dengan kerusakan motorik (*motor impairment*) dan tidak bisa menggunakan secara efektif pengendali manual tersebut menjadikan penghambat utama tidak dapat merasakan keuntungan dari *game* tersebut.

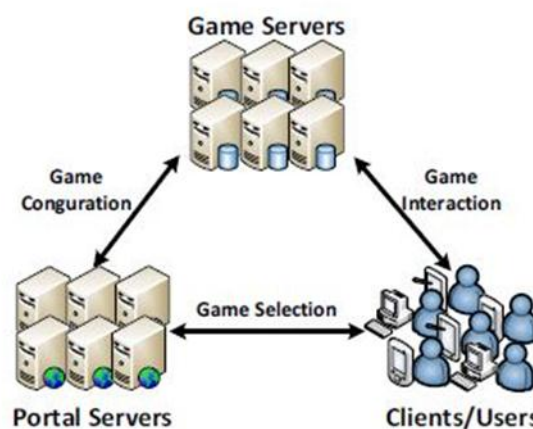
Terdapat beberapa cara untuk membuat *game* komputer menjadi lebih mudah diakses untuk semua kalangan adalah dengan menerapkan *hands-free access* dan konsep *cloud gaming*. *Hands-free access* ini akan membantu orang-orang dengan kerusakan motorik agar dapat memainkan *game* dan konsep *cloud gaming* akan membuat *game* dapat dimainkan walaupun pada perangkat komputer yang *low-end*.

Adapun tujuan penelitian yaitu mengimplementasikan *speech recognition system* pada sebuah *game* agar dapat dikendalikan dengan menggunakan perintah suara, memainkan *game* berspesifikasi tinggi dengan perangkat komputer *low-end* melalui *cloud server* serta melakukan analisis QoS.

## 2. DASAR TEORI DAN PERANCANGAN

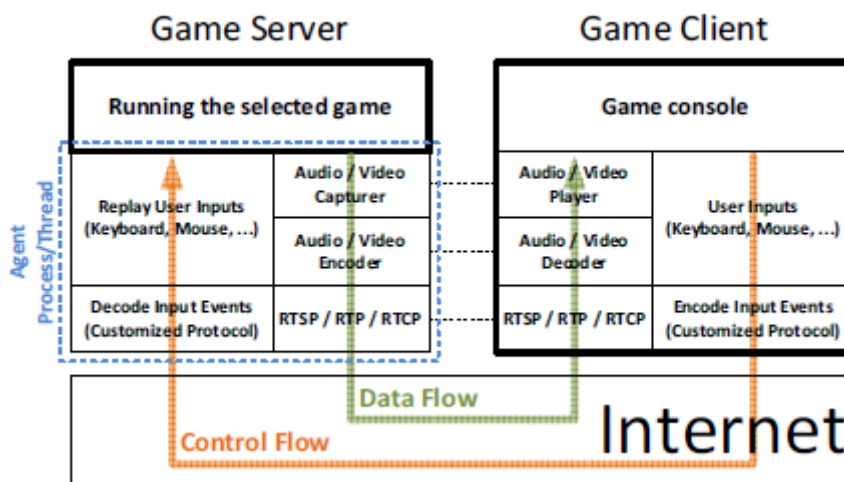
### 2.1 Cloud Gaming

*Cloud gaming* adalah proses menjalankan sebuah aplikasi *gaming* interaktif secara *remote* di dalam *cloud* dan mengeluarkan keluaran dalam bentuk *video* ke perangkat pengguna melalui internet [8]. Pengguna melakukan interaksi dengan aplikasi melalui sebuah *thin client*, yang bertanggung jawab untuk menampilkan video dari *cloud rendering server* dan juga mengumpulkan perintah-perintah dari pengguna serta mengirimkannya kembali ke *cloud*[8].



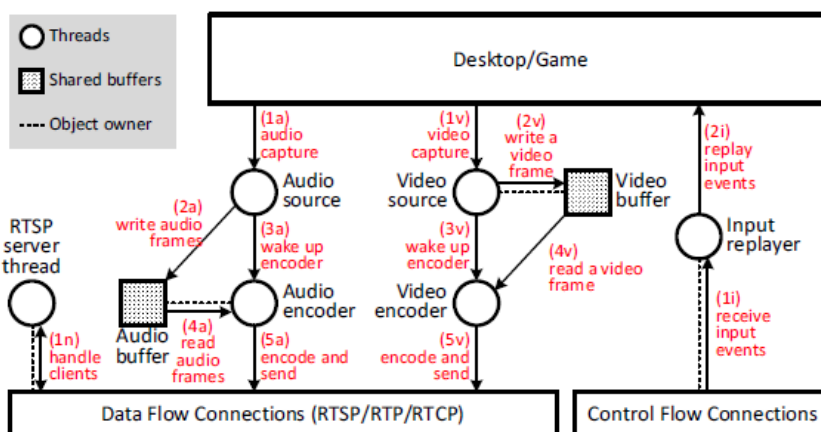
Gambar 1. Arsitektur Cloud Gaming [3]

*GamingAnywhere* merupakan salah satu *platform cloud gaming* yang bersifat *open source* dan *cross platform*, di mana *game* yang dijalankan pada perangkat PC dapat dimainkan melalui perangkat lainnya seperti *smartphone*.



Gambar 2. Sistem Kerja GamingAnywhere [3]

Terdapat dua jenis *network flow* pada arsitektur *GamingAnywhere*, yakni *data flow* dan *control flow*. *Data flow* digunakan untuk men-stream-kan *frames* audio dan video dari *server* ke *client*, sedangkan *control flow* berjalan dalam arah yang berlawanan, yakni digunakan untuk mengirimkan gerakan pengguna dari sisi *client* ke *server*.



Gambar 3. Hubungan antar modul server, shared buffer dan koneksi jaringan[3].

Pada *GamingAnywhere*, *RTSP server thread* merupakan hal pertama yang dijalankan pada *server*. *RTSP server thread* menerima *RTSP commands* dari sebuah *client*, menjalankan encoder dan mengatur *data flow* untuk mengirimkan *frames* yang telah diencode[3]. *Data flows* dapat dibawa/disampaikan melalui koneksi jaringan tunggal atau banyak (*multiple*) berdasarkan protokol layer *transport* yang diinginkan, seperti TCP atau UDP. Pada TCP, *frames* yang terencode dikirimkan sebagai data biner dalam RTSP. Kedua *RTSP command* dan *RTP/RTCP packets* dikirimkan melalui koneksi *RTSP over TCP* yang terbangun dengan *client*. Pada UDP, *frames* yang terencode dikirimkan melalui *RTP over UDP*.

## 2.2 Speech recognition system

*Speech recognition system* adalah sebuah proses konversi suara yang dikeluarkan menjadi sebuah teks (*speech-to-text*). *Speech recognition* memungkinkan pengguna untuk memberikan perintah atau masukan (*input*) kepada sebuah aplikasi tertentu dengan menggunakan suaranya. Proses tersebut dilakukan oleh sebuah komponen piranti lunak yang disebut *speech recognition engine* [2]. Aplikasi tersebut dapat mengerjakan salah satu dari dua hal, aplikasi dapat menerjemahkan hasil dari *recognition* sebagai sebuah perintah dan aplikasi menangani teks yang dikenali (*recognized text*) hanya sebagai teks saja, maka aplikasi bertindak sebagai aplikasi dikte (*dictation*). Dalam kasus penelitian, aplikasi bertindak sebagai sebuah *command and control*.

Terdapat banyak faktor yang mempengaruhi kuat (*robustness*) dari sebuah *speech recognition*, diantaranya yaitu kualitas *microphone* yang digunakan dan tingkat kebisingan keadaan sekitar. Kinerja dari *speech recognition system* ini biasanya dievaluasi berdasarkan tingkat akurasi dan kecepatan prosesnya. Tingkat akurasi dinilai dari *word error rate* (WER), dan kecepatan proses diukur dengan faktor *real time*. Tingkat akurasi dari sebuah sistem *speech recognition* dipengaruhi oleh beberapa hal, diantaranya adalah besar kosa kata, cara pengucapan (*pronunciation*), artikulasi, aksen, volume suara, kecepatan pengucapan.

## 2.3 QoS (Quality of Service)

QoS adalah kemampuan suatu jaringan untuk menyediakan layanan yang baik dengan menyediakan bandwidth, mengatasi jitter dan delay [7]. Parameter QoS adalah *latency*, *jitter*, *packet loss*, *throughput*, *MOS*, *echo cancellation* dan PDD. QoS sangat ditentukan oleh kualitas jaringan yang digunakan.

### 2.3.1 Packet Loss

Parameter yang menunjukkan jumlah total paket yang hilang, dapat terjadi karena *collision* dan *congestion* pada jaringan dan berpengaruh pada semua aplikasi karena *retransmisi* akan mengurangi efisiensi jaringan secara keseluruhan meskipun jumlah bandwidth cukup tersedia [9].

Tabel 1. Kategori Paket Loss [Telecommunications and Internet Protocol Harmonization Over Network (TIPHON)]

Kategori Degradasi	Paket Loss	Indeks
Indeks Sangat Bagus	0 %	4
Bagus	3 %	3
Sedang	15 %	2
Jelek	25 %	1

### 2.3.2 Delay

*Delay* Adalah waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. Delay dapat dipengaruhi oleh jarak, media fisik, kongesti atau juga waktu proses yang lama [9].

Tabel 2. Kategori Paket Delay [TIPHON]

Kategori Latency	Paket Delay	Indeks
Indeks Sangat Bagus	<150 ms	4
Bagus	150 ms s/d. 300 ms	3
Sedang	300 ms s/d. 450 ms	2
Jelek	>450 ms	1

### 2.3.3 Throughput

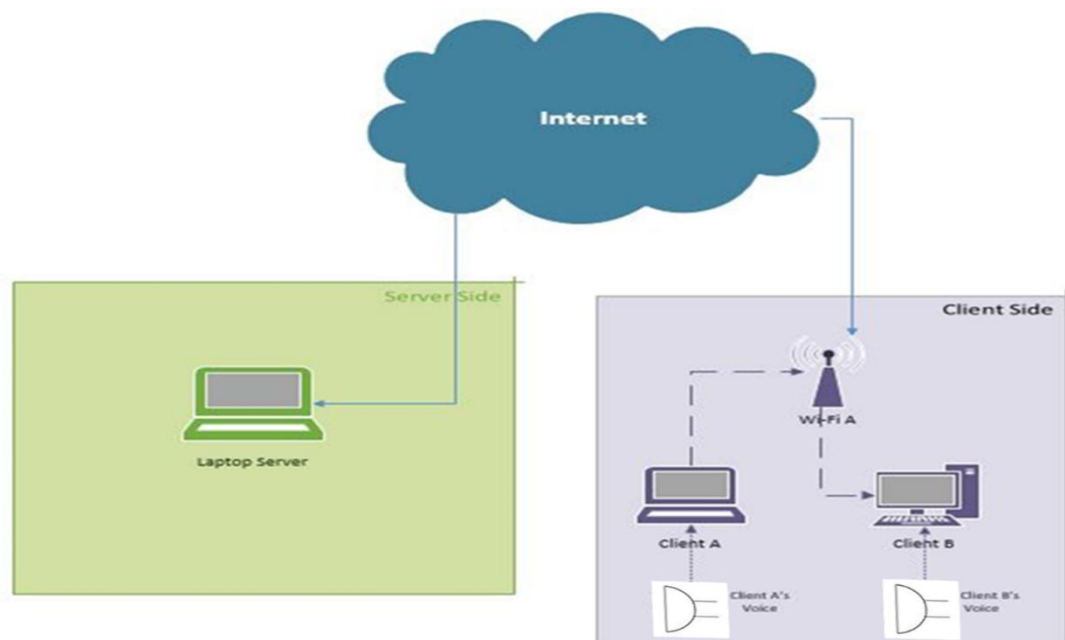
*Throughput* adalah kecepatan transfer data efektif, yang diukur dalam bps. *Throughput* merupakan jumlah total kedatangan paket yang sukses yang diamati pada tujuan selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut [9].

Tabel 3. Kategori Paket *Throughput* [TIPHON]

Kategori <i>Throughput</i>	Paket <i>Throughput</i>	Indeks
Indeks Sangat Bagus	100%	4
Bagus	75%	3
Sedang	50%	2
Jelek	<25%	1

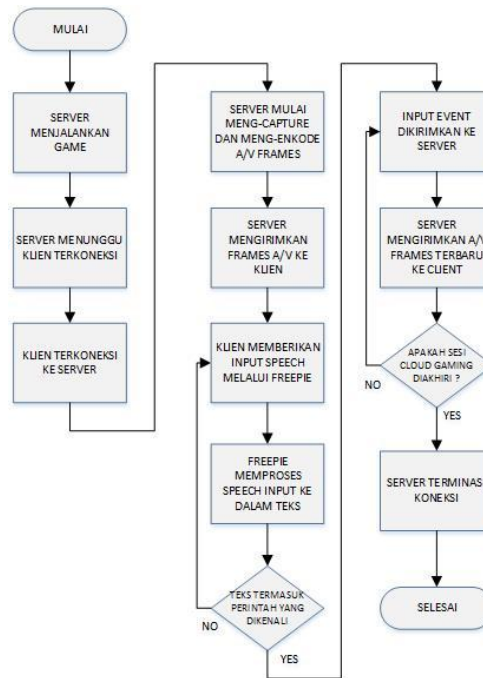
### 2.4 Model Sistem

Sistem dalam penelitian menghasilkan sebuah interaksi *game* di sisi *client* dan *server*. Dengan memanfaatkan teknologi *cloud computing* dalam *cloud gaming*, pengguna akan dapat memainkan *game* tertentu yang sebelumnya telah dipasang dalam *server* mengendalikannya melalui perintah suara. Satu laptop dijadikan sebagai *server* tunggal beserta judul *game* tertentu untuk melayani dua *client*. *Server* akan mengirimkan *stream* video dan audio kepada pengguna melalui internet beriringan dengan interaksi yang dilakukan. Di sisi pengguna, pengguna melakukan interaksi dengan *game* melalui perintah yang dihasilkan oleh perintah suara melalui *FreeePIE*.



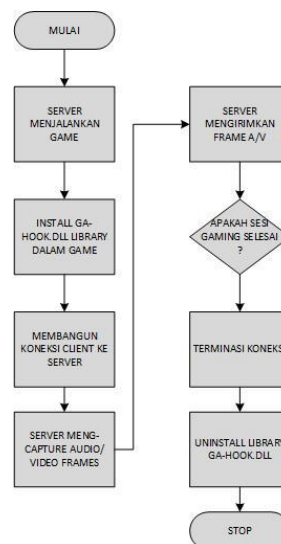
Gambar 4. Blok sistem cloud gaming dengan implementasi speech recognition system

Gambar 4 menunjukkan disisi *pengguna*, suara yang dihasilkan diproses oleh *speech recognition engine*, *FreeePIE* untuk diubah menjadi sebuah teks atau *stroke keyboard*. Bila teks atau *stroke keyboard* tersebut terdapat dalam daftar perintah yang telah ditetapkan sebelumnya, maka perangkat akan mengirimkan data tersebut ke dalam *server* untuk dieksekusi. Lalu di sisi *server*, data yang diterima dari pengguna dieksekusi menjadi sebuah *input command* yang dimaksud dan menjalankan interaksi dalam *game*. Hasil interaksi tersebut dikirimkan kembali ke sisi pengguna dalam bentuk video dan audio *stream*.



Gambar 5. Diagram alir model penelitian secara keseluruhan

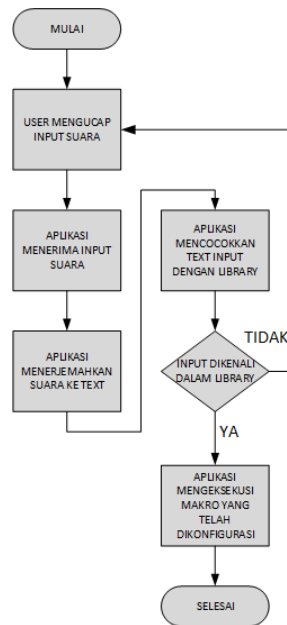
Pada gambar 5, ditunjukkan diagram alir secara keseluruhan dari *cloud gaming* yang dikendalikan oleh suara pengguna. Input suara pengguna akan diolah menggunakan *platform FreePie*. Skenario pengukuran parameter akan dilakukan ketika *client* terhubung dengan *server* dan sesi *gaming* berlangsung. Pengecekan parameter dilakukan dengan menggunakan aplikasi Wireshark untuk mengukur QoS (*delay, troughput, packet loss*).



Gambar 6. Diagram alir cloud gaming GamingAnywhere

Pada gambar 6, ditunjukkan diagram alir dari sistem *cloud gaming* menggunakan *platform GamingAnywhere*. Sebelum sesi *cloud gaming* dimulai, *server* menjalankan *game* terlebih dahulu dan dilakukan proses instalasi *ga-hook.dll* ke dalam *game*. *Server* mulai meng-*capture* audio/video

frames dan siap mengirimkan ke *client*. Ketika *client* telah terhubung, *server* akan mengirimkan *frame* audio/video yang sebelumnya telah di-encode menggunakan *library* yang disediakan.



Gambar 4. Diagram alir speech recognition system

Pada gambar 7, ditunjukkan diagram alir dari proses *speech recognition system*. Input dari pengguna yang berupa suara akan diteruskan oleh *microphone* ke dalam aplikasi. Kemudian aplikasi mentranslasikan masukan yang berupa suara tersebut ke dalam sebuah teks. Bila input tersebut terdapat di dalam *library* dan makro telah dikonfigurasi, maka aplikasi akan menjalankan perintah yang ditentukan. Makro yang dikonfigurasi dapat berupa *keystroke* tunggal ataupun kombinasi beberapa *keystroke*.

### 2.5 Spesifikasi Game Uji dan PC

Pengukuran sistem akan dilakukan dengan menggunakan dua jenis *pc game* dengan *genre* dan spesifikasi yang berbeda. Dua buah game yang digunakan untuk menguji system dengan spesifikasi sebagai berikut:

a) **NEVERBALL**

Tabel 4. System requirement *game NEVERBALL*

OS	Linux/Windows
Processor	500 MHz Processor
Graphics	OpenGL 1.2.1 or greater
Memory	128 MB
Storage	200 MB

b) **7 Days to Die**

Tabel 5. System requirement *game 7 Days to Die*

OS	Windows XP (SP3)
Processor	2.4 Ghz Dual Core
Graphics	512 MB dedicated memory
Memory	4GB RAM
Storage	3GB

c) **PC Server**

Tabel 6. Spesifikasi PC Server

<b>OS</b>	Windows 10
<b>Processor</b>	AMD FX-7500 up to 3.3 Ghz
<b>Graphics</b>	AMD Radeon R7 2GB
<b>Memory</b>	8GB RAM
<b>Storage</b>	500GB

d) **PC Client**

Tabel 7. Spesifikasi PC Client

<b>OS</b>	Windows 7
<b>Processor</b>	AMD E-450 1.65Ghz
<b>Graphics</b>	356MB
<b>Memory</b>	2GB RAM
<b>Storage</b>	300GB

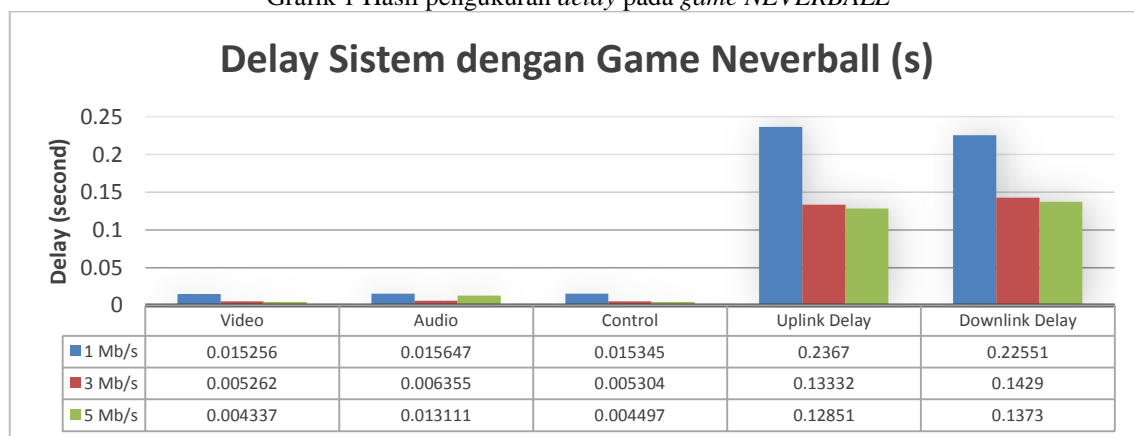
**2.6 Skenario Pengukuran QoS**

Terdapat tiga parameter yang diuji dalam pengukuran QoS, diantaranya yaitu *delay*, *packet loss* dan *throughput* yang diuji dalam kondisi *bandwidth* berbeda-beda. *Bandwidth* yang digunakan dalam penelitian yaitu 1 Mbps, 3 Mbps dan 5 Mbps. Pembatasan *bandwidth* dilakukan dengan menggunakan *software* NEWT. Pengukuran dilakukan pada *game NEVERBALL* dan *7 Days to Die* dalam kondisi satu *server* menjalankan satu *game* dengan satu *client* yang saling terhubung dengan *bandwidth* berbeda-beda. Percobaan dilakukan lima kali dengan durasi waktu 120 detik dengan asumsi rata-rata *game* yang layak dimainkan memiliki 60 frame perdetik. Sehingga jumlah frame yang diujikan dalam setiap melakukan eksperimen adalah 7200 frame.

**3. PEMBAHASAN**

Hasil pengukuran *delay*, *packet loss* dan *throughput* yang dilakukan menggunakan Wireshark pada *bandwidth* 1 Mbps, 3 Mbps dan 5 Mbps dengan waktu uji coba masing masing *bandwidth* selama 120 detik ketika diakses oleh satu *client* terlihat pada Grafik 1-4. Pengukuran *delay*, *paket loss* dan *throughput* pada *game 7 Days to Die* dan *NEVERBALL* dalam keadaan terhubung dengan *server* GamingAnywhere dengan satu *client*.

Grafik 1 Hasil pengukuran *delay* pada *game NEVERBALL*



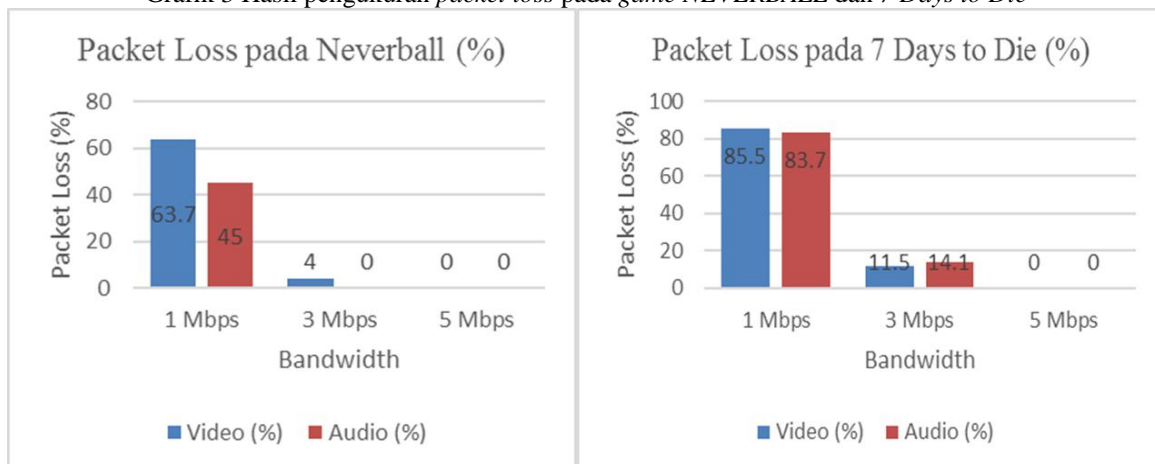


Grafik 2 Hasil pengukuran *delay* pada game *7 Days to Die*



Pada kondisi *bandwidth* 1 Mb/s kedua *game* uji menghasilkan *delay uplink* dan *downlink* yang besar bila dibandingkan dengan *delay* pada kondisi *bandwidth* 3 Mb/s dan 5 Mb/s . Khususnya pada *game 7 Days to Die*, *delay uplink* dan *delay downlink* terbilang sangat besar, mencapai nilai 1.04 detik untuk *uplink* dan 0.904 detik untuk *downlink*. Hal ini dikarenakan *bandwidth* yang kecil menyebabkan tingginya *traffic* pada jaringan. Pada kondisi *bandwidth* 3 Mb/s dan 5 Mb/s, kedua *game* uji menghasilkan *delay* yang cukup kecil karena *bandwidth* cukup besar dan *traffic* rendah. Hasil pengujian pada *game NEVERBALL* dengan *bandwidth* 1 Mb/s *delay uplink* dan *downlink* yang dihasilkan lebih tinggi dibandingkan dengan *bandwidth* 3 Mb/s dan 5 Mb/s walaupun demikian nilai *delay* tersebut terhitung kecil. Hal ini membuktikan bahwa *game NEVERBALL* tidak membutuhkan *bandwidth* yang besar untuk diakses dengan layanan *cloud gaming*.

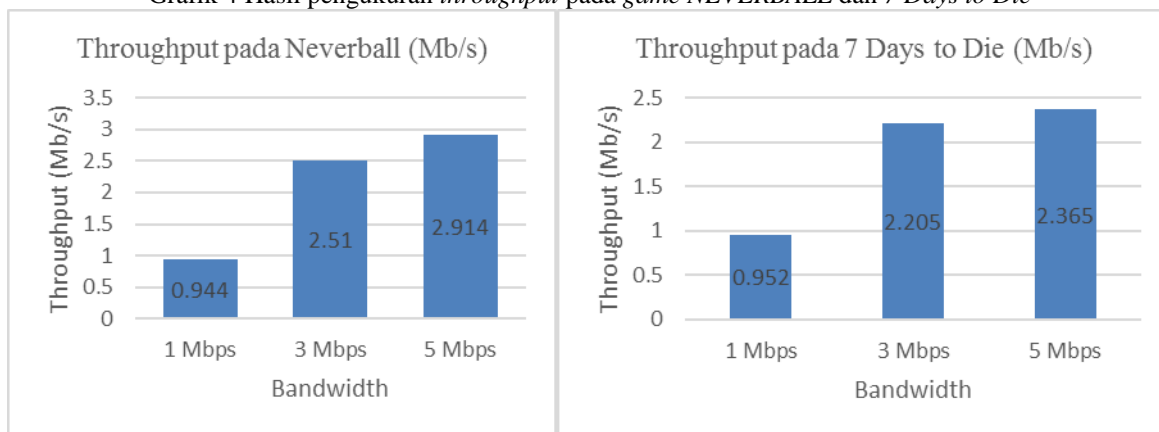
Grafik 3 Hasil pengukuran *packet loss* pada game *NEVERBALL* dan *7 Days to Die*



*Packet loss* pada kedua *game* uji dalam kondisi *bandwidth* 1 Mbps sangat tinggi. *Packet loss* video mencapai 63.7% dan audio sebesar 45% pada *game NEVERBALL* menyebabkan *video* di sisi *client* sangat tidak beraturan dan membuat *game* tidak dapat dimainkan. Pada *game 7 Days to Die*, *packet loss* video mencapai 85.5% dan audio mencapai 83.7% menyebabkan banyak *frame* yang hilang dan membuat *game* tidak dapat dimainkan sebagaimana semestinya. Dalam kondisi *bandwidth* 3 Mbps , jumlah *packet loss* terbilang rendah untuk kedua *game* uji dan sudah mendekati 0% dan *audio/video frames* yang diterima di sisi *client* sudah sesuai dengan yang

diharapkan. Walaupun masih ada *frame* yang hilang tetapi dalam frekuensi yang kecil. Hal ini disebabkan oleh besarnya ukuran *packet* video/audio dari *game 7 Days to Die* meskipun sudah di encode. Dalam kondisi *bandwidth* 5 Mbps, *packet loss* yang dihasilkan pada kedua *game* uji bernilai 0% dan *game* dapat dimainkan dengan baik.

Grafik 4 Hasil pengukuran *throughput* pada *game NEVERBALL* dan *7 Days to Die*



Dari hasil pengukuran pada kedua *game* uji diperoleh nilai *throughput* yang semakin besar dengan semakin besarnya *bandwidth* yang diberikan. Sehingga dengan *bandwidth* 5 Mbps kedua buah *game* uji dapat dimainkan dengan baik. Nilai *throughput* pada *game* Neverball lebih besar dibandingkan *game 7 Days to Die*. Nilai *throughput* pada *game 7 Days to Die* memiliki angka yang tidak terlalu jauh dengan *game* Neverball. Berdasarkan grafik 4 dapat dilihat bahwa *game* Neverball dapat dijalankan lebih baik dibandingkan *game 7 Days to Die* diperlihatkan dengan nilai *throughput* yang jauh lebih besar.

#### 4. PENUTUP

##### 4.1 Kesimpulan

Untuk meraih QoS yang optimal dalam menjalankan *game* dengan cloud gaming *GamingAnywhere*, dibutuhkan minimal *bandwidth* sebesar 3 Mbps. Bila *bandwidth* yang diberikan kurang dari 3 Mbps, sistem akan mengalami *delay* yang massif bernilai  $\pm 0.5$  detik pada *game NEVERBALL* dan bernilai  $\pm 1.9$  detik pada *game 7 Days to Die*. *Packet loss* yang dihasilkan pun akan sangat tinggi, bernilai 63.7% untuk video dan 45% untuk audio pada *game NEVERBALL* dan bernilai 85.5% untuk video dan 83.7% untuk audio pada *game 7 Days to Die*.

##### 4.2 Saran

1. Penelitian selanjutnya dapat dilakukan dengan pengukuran CPU, RAM, FPS dari setiap PC ketika terhubung dengan cloud gaming *GamingAnywhere*. Selain itu dapat dilakukan penelitian terhadap tingkat akurasi yang dihasilkan oleh sistem *speech recognition*
2. Aplikasi sistem *speech recognition* sebagai input kontrol dipasang pada *server* sehingga proses komputasi dilakukan sepenuhnya di sisi *server*.

#### DAFTAR PUSTAKA

[1] Andreas Sackl, Raimund Schatz. 2016. *QoE Management Made Uneasy: The Case of Cloud Gamig*. IEEE ICC2016-workshop W12-Workshop on Quality of Experience for future Internet Applications and Service

- [2] B. Plannerer, "An introduction to speech recognition," *Munich, Ger.*, 2005.
- [3] Chun-Ying Huang, Cheng-Hsin Hsu, Yu-Chun Chang and Kuan-Ta Chen, "GamingAnywhere : An Open *Cloud gaming* System", Department of Computer Science, National Taiwan Ocean University, 2013.
- [4] Chen Kuan-Ta. 2014. *On the Quality of Service of Cloud Gaming Sysstes*. IEEE Transactions On Multimedia, Vol 16, No.2
- [5] Herwindita. 2014. *Implementasi Game Sebagai Layanan (GAAS) Menggunakan Open-Source Cloud Gaming Server GamingAnywhere*. E-proceeding of Engineering: vol 1. No.1 Desember 2014
- [6] Ghergulescu. *Energy Consumption Analysis of Cloud-based Video Games Streaming to Mobile Devices*. MM-15-118
- [7] Grandi, Veera Venkata Santosh Surya Ganesh. *On Quality of Service of Mobile Cloud Gaming Using Gaming Anywhere*. Thesis no: MSEE-2016-53
- [8] R. Shea, J. Liu, E. Ngai, and Y. Cui, "Cloud gaming: Architecture and performance," *IEEE Netw.*, vol. 27, no. 4, pp. 16–21, 2013.
- [9] Sasmita, Patriya Wahyu. Analisis Quality of Srevice (QoS) Pada Jaringan Internet (Studi Kasus: Fakultas Kedokteran Universitas Tanjungpura)
- [10] S. Harada, J. O. Wobbrock, and J. A. Landay, "Voice games: Investigation into the use of non-speech voice input for making computer games more accessible," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6946 LNCS, no. PART 1, pp. 11–29, 2011.
- [11] Xue Zheng. 2014. *Playing High-End Video Games in the Cloud: A Measurment Study*. DOI 10.1109/TCSVT.201.2364419, IEEE Transactions on Circuits and Systems for Video Technology