

IMPLEMENTASI GYROSCOPE SENSOR DALAM SISTEM KONTROL PADA GAME YANG BERBASISKAN CLOUD GAMING

IMPLEMENTATION OF GYROSCOPE SENSOR IN CONTROL SYSTEM ON GAME BASED ON CLOUD GAMING

Gilang Cahya GumiLang¹, Dr. Ir. Rendy Munadi., M.T.², Sussi, S.Si., M.T.³

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

¹gilanggumilang@student.telkomuniversity.ac.id, ²rendimunadi@telkomuniversity.co.id,

³sussiss@telkomuniversity.ac.id

Abstrak

Cloud Gaming merupakan bagian dari layanan *cloud computing* yang memproses sebuah aplikasi gaming interaktif secara *remote* dan mengeluarkan output video ke komputer *client* melalui perantara jaringan. Dengan adanya *Cloud Gaming*, pengguna dapat menjalankan aplikasi *game* yang spesifikasinya lebih tinggi dibandingkan dengan komputer penggunanya.

Namun sebagian pemain tidak dapat merasakan bagaimana serunya bermain *game* dikarenakan keterbatasan fisiknya, terutama pada bagian tangan. Untuk mengatasi masalah tersebut, penulis membuat dan menganalisis sensor *gyroscope* dengan menggunakan Mikrokontroller Arduino sebagai kontroller pada game yang berbasiskan dengan *cloud gaming* untuk mengetahui hasil dari performa kontroller tersebut.

Dari hasil pengujian, client membutuhkan 0% GPU usage, 24-27.2% CPU usage dan 684.8-702.2 MB RAM untuk memainkan NEVERBALL. Client membutuhkan 0% GPU usage, 27.4-37.8% CPU usage dan 893.8-1008.2 MB RAM untuk memainkan DRAGONS DOGMA. Framerate yang dicapai oleh client pada game NEVERBALL bernilai 44-45 FPS dan pada game DRAGON'S DOGMA: DARK ARISEN, Framerate yang dicapai bernilai 43-49 FPS.. Lama delay yang dicapai sebesar 0.003374 detik pada game NEVERBALL dan bernilai 0.003545 detik pada game DRAGONS DOGMA.

Kata kunci : *Cloud gaming, Arduino, Gyroscope Sensor*

Abstract

Cloud Gaming is part of a cloud computing service that processes an interactive gaming application remotely and outputs video to the client computer through a network intermediary. With the existence of Cloud Gaming, users can run game applications with higher specifications than the user's computer.

However, some players cannot feel the excitement of playing games due to physical limitations, especially in the hands. To overcome this problem, the authors create and analyze gyroscope sensors using Arduino Microcontroller as a controller in a game based on cloud gaming to find out the results of the controller performance.

From the test results, the client requires 0% GPU usage, 24-27.2% CPU usage and 684.8-702.2 MB RAM to play NEVERBALL. The client requires 0% GPU usage, 27.4-37.8% CPU usage and 893.8-1008.2MB RAM to play DOGMA DRAGONS. Framerate achieved by the client in the NEVERBALL game is worth 44-45 FPS and in the game DRAGON'S DOGMA: DARK ARISEN, the Framerate achieved is 43-49 FPS.

Keywords: *Cloud gaming, Arduino, Gyroscope Sensor*

1. Pendahuluan

Industri game setiap tahun semakin meningkat [1]. Pemain dalam memainkan game terbaru membutuhkan perangkat dengan spesifikasi yang tinggi ditandai dengan RAM, CPU dan GPU yang besar. Pemain yang memiliki perangkat yang rendah tidak mampu memainkan game dengan tingkat komputasi dan grafis yang kompleks. Industri game membutuhkan sebuah teknologi yang mampu mensolusikan permasalahan pemain yang memiliki perangkat rendah sehingga tetap mampu menikmati permainan seperti bermain dengan perangkat berspesifikasi tinggi [2].

Cloud gaming merupakan sebuah layanan *cloud computing* yang menjalankan aplikasi *gaming* interaktif secara *remote* dalam struktur *cloud* dan mengirimkan sebuah *output* berupa video ke *client*. Pemain akan berinteraksi dengan *client* dan mengirimkan input perintah kemudian *client* mengirimkan input tersebut ke *server cloud*. Dengan adanya *cloud gaming*, pemain dapat merasakan serunya bermain game PC terbaru tanpa harus memikirkan spesifikasi perangkatnya [2] [3] [4].

Pemain yang memiliki keterbatasan fisik pada tangan tentunya akan mengalami kesulitan dalam memberikan input perintah kepada *client* ketika memainkan *game*. Penelitian yang dilakukan oleh Indra Perdana Putera Sutejo dimana memainkan *game* menggunakan *cloud GamingAnywhere* dengan *speech recognition system* sebagai *command input* memungkinkan pemain dengan kebutuhan khusus tetap bisa memainkan *game* pada berbagai perangkat PC yang dimilikinya [3] [4].

Sensor Gyroscope merupakan sensor yang digunakan sebagai *gesture recognition* untuk mengkontrol gestur tangan atau bagian tubuh lainnya sehingga dapat dijadikan *command input* dalam memainkan game [5]. Penelitian yang dilakukan dalam Tugas Akhir adalah mengimplementasi sensor gyroscope sebagai *controller* atau inputan pada game yang berbasiskan *cloud gaming server* menggunakan *open-source GamingAnywhere*.

2. Dasar Teori

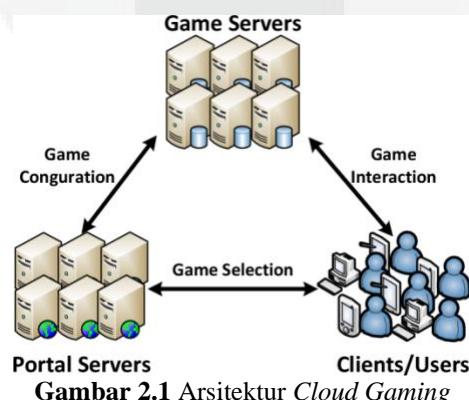
2.1 Sensor *Gyroscope*

Sensor *Gyroscope* merupakan sebuah sensor pendeksi rotasi atau perputaran pada suatu perangkat dan bekerjasama dengan *Accelerometer* untuk memiringkan atau memutarkan suatu perangkat. Sensor ini dapat mendekripsi rotasi sampai 360 derajat. Salah satu contoh tipe sensor *gyroscope* tersebut adalah MPU6050. MPU6050 merupakan perangkat sensor yang terdiri dari *3-axis accelerometer* (sensor percepatan) dan *3-axis gyroscope* (sensor keseimbangan) atau yang kita kenal dengan *6DOF (Degrees Of Freedom)* dimana gerakan yang dapat diakses oleh modul sensor ini melalui tiga sumbu yaitu x, y, dan z. Sensor ini mampu membaca kemiringan sudut berdasarkan data sensor yang termuat pada modul *MPU6050*.[15]

2.2 Cloud Gaming

Cloud Gaming merupakan bagian dari layanan *cloud computing* yang memproses sebuah aplikasi *gaming* interaktif secara *remote* dan mengeluarkan *output* video ke komputer *client* melalui perantara internet. Dengan adanya *Cloud Gaming*, pengguna dapat menjalankan aplikasi *game* yang spesifikasinya lebih tinggi dibandingkan dengan komputer penggunanya.

Gambar 2.1 menunjukkan arsitektur *cloud gaming*, dimana *client* melakukan *login* ke sistem melalui portal server, *client* mencari *game* dan melakukan permintaan ke server portal untuk memainkan *game*. Lalu server portal melakukan pencarian dan menemukan game server. Kemudian *game server* menjalankan *game* yang dipilih *client* dan mengembalikan *URL game server* ke *client*. *Client* terhubung ke *game server* dan *client* mulai bermain.



Gambar 2.1 Arsitektur *Cloud Gaming*

2.3 GamingAnyWhere



Gambar 2.3 Logo GamingAnyWhere

GamingAnywhere adalah sebuah aplikasi *cloud gaming* yang bersifat *open-source*. GamingAnywhere ini dapat dimodifikasi dan diatur sesuai dengan keinginan pengguna. GamingAnyWhere saat ini mendukung Windows dan Linux dan dapat *cross platform* ke OS lain termasuk Android.

GamingAnyWhere memiliki tiga *external libraries* yang digunakan untuk menjalankan sistem, yaitu *libavcodec/libavformat*, *live555*, dan *SDL Library*.[8]

- a. *Libavcodec/libavformat* berfungsi untuk merekam, mengubah, dan men-stream audio maupun video. *Libavcodec/libavformat* digunakan juga untuk menghandle protokol RTP di sisi server.
- b. *Live555 library* berfungsi untuk multimedia streaming dengan menggunakan protokol *open standard* (RTP, RTCP, RTSP, dan SIP). Pada sistem GamingAnyWhere, *live555 library* digunakan untuk meng-handle protokol RTSP/RTP di sisi *Client*.
- c. *Simple DirectMedia Layer (SDL) library* merupakan sebuah *cross platform library* yang dirancang untuk menyediakan akses ke audio, keyboard, mouse, joystick, *SDL library* pada sistem GamingAnyWhere digunakan untuk me-render audio dan video di sisi .

GamingAnyWhere dapat di *download* untuk keperluan riset atau untuk dikembangkan lebih lanjut. Untuk mendownload aplikasi tersebut, dapat dilihat pada laman <http://gaminganywhere.org/download.html>.

3. Perancangan Sistem

Desain sistem dalam penelitian tugas akhir menghasilkan interaksi game antara sisi *client* dengan sisi server menggunakan teknologi *cloud gaming*. Pemain dapat memainkan *game* yang berada di dalam server dengan menggunakan *command input* sensor *gyroscope*. Satu laptop dijadikan server untuk melayani *client*. Server akan mengirimkan video dan audio *stream* ke *client*. Kemudian *client* melakukan interaksi dengan *game* melalui *command input* yang dihasilkan oleh *sensor gyroscope*.

3.1. Diagram Blok

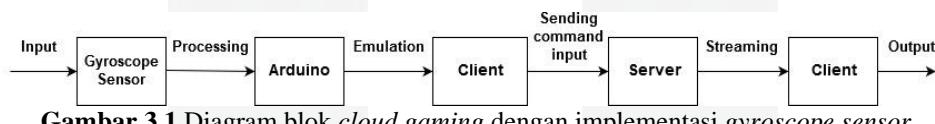
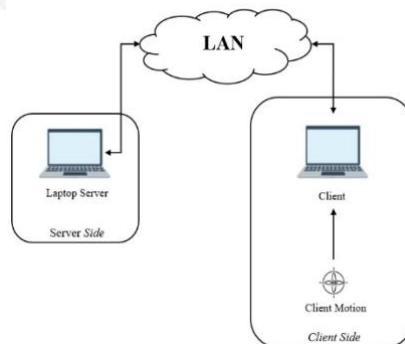
**Gambar 3.1** Diagram blok *cloud gaming* dengan implementasi *gyroscope sensor*

Diagram blok pada gambar 3.1 menunjukkan bahwa *command input* diperoleh dari sensor *gyroscope* lalu diproses oleh Arduino dan diemulasikan sehingga *client* mengenali *command input* tersebut sebagai *input gerak mouse*. *Client* mengirimkan *command input* menuju server. Kemudian server mengirimkan A/V frame ke *client*.

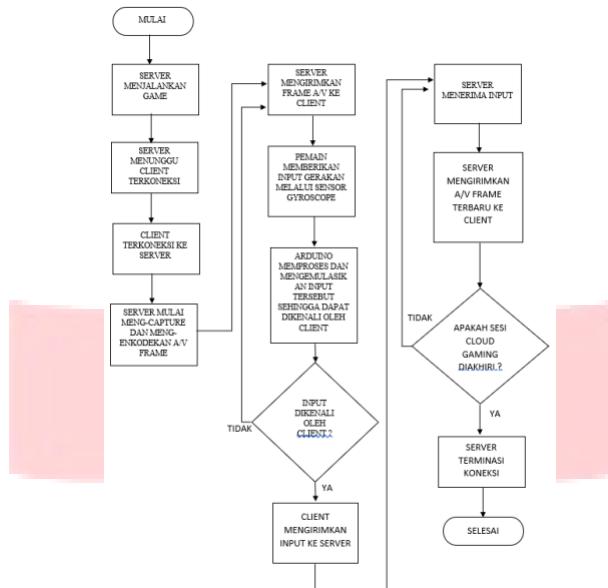
3.2 Topologi Jaringan

**Gambar 3.2** Topologi jaringan *cloud gaming* dengan implementasi *gyroscope sensor*

Topologi jaringan *cloud gaming* dengan implementasi *gyroscope sensor* menunjukkan bahwa sistem terdiri dari 2 sisi, yaitu sisi server dan sisi *client*. Pada sisi *client*, sensor gerak yang dihasilkan akan diproses dan diemulasikan oleh Arduino sehingga *command input* tersebut dikenali oleh laptop *client*, kemudian *command input* tersebut dikirim ke server. Pada sisi server *command input* yang

diterima dari *client* akan dieksekusi sehingga terjadi interaksi dalam *game*. Hasil interaksi tersebut dikirimkan kembali ke sisi *client* kedalam bentuk video dan audio *stream*.

3.3 Flowchart



Gambar 3.3 Flowchart model secara keseluruhan

Gambar 3.3 menunjukkan keseluruhan alir diagram dari *cloud gaming* yang memiliki *command input* dari sensor *gyroscope*, *command input* tersebut diproses oleh *Arduino*. Untuk memulai sesi *cloud gaming*, server menjalankan *game* terlebih dahulu. Setelah itu server menunggu client untuk saling terkoneksi, kemudian server mulai meng-*capture* dan meng-enkodekan A/V *frame* dan mengirimkan A/V tersebut ke *client*. *Client* mendapatkan *command input* melalui sensor *gyroscope* untuk diproses dan diemulasikan oleh *Arduino*. Lalu *command input* tersebut dikenali oleh laptop *client* sebagai perintah dan mengirimkan hasil proses tersebut ke server. Kemudian server mengirimkan A/V *frame* terbaru ke *client*.

4. Data dan Analisis

Pada Bab IV ini membahas analisis dari hasil pengujian dengan menggunakan skenario dan parameter yang telah ditentukan pada Bab III. Pengujian dilakukan dengan dengan cara mengimplementasikan sensor *gyroscope* dan GamingAnyWhere pada server yang berjalan di atas Windows 10 dan client yang berjalan di atas Windows 7. Untuk dapat melihat performansi game yang berjalan, maka dilakukan pengambilan data menggunakan software MSI Afterburner, HWiNFO, dan FPS counter pada aplikasi GeForce Experience.

4.1 Resource Usage pada Server

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

Tabel 4.1 Resource Usage saat GA OFF pada Game Neverball

Game	Resolusi	GPU Average (%)	CPU Average (%)	Memory Usage (%)	FPS
Neverball	800x600	15.8	16.5	3363.6	59.9
	1366x768	30.1	16	3611.4	59.9

Tabel 4.2 Resource Usage saat GA ON pada Game Neverball

Game	Resolusi	GPU Average (%)	CPU Average (%)	Memory Usage (%)	FPS
Neverball	800x600	26.5	17.7	3509.1	59.9
	1366x768	30.5	16.9	3580.6	59.9

Tabel 4.3 Resource Usage saat GA OFF pada Game Dragons Dogma

Game	Resolusi	GPU Average (%)	CPU Average (%)	Memory Usage (%)	FPS
Dragons Dogma	800x600	38.6	50	3523.2	114.4
	1366x768	50.1	50.1	3854.9	85.9

Tabel 4.4 Resource Usage saat GA ON pada Game Dragons Dogma

Game	Resolusi	GPU Average (%)	CPU Average (%)	Memory Usage (%)	FPS
Dragons Dogma	800x600	40.5	47.4	3835.1	117.1
Dragons Dogma	1366x768	50.6	51.2	4393.9	82.6

Tabel 4.5 Resource Usage saat GA Tidak Terhubung Client pada Game Neverball

Game	Resolusi	GPU Average (%)	CPU Average (%)	Memory Usage (%)	FPS
Neverball	800x600	26	17.4	3237.4	59.9
Neverball	1366x768	30.5	17.3	3363.5	59.9

Tabel 4.6 Resource Usage saat GA Terhubung Client pada Game Neverball

Game	Resolusi	GPU Average (%)	CPU Average (%)	Memory Usage (%)	FPS
Neverball	800x600	26.9	26.3	3231.3	59.9
Neverball	1366x768	23	24.2	3860.3	59.9

Tabel 4.7 Resource Usage saat GA Tidak Terhubung Client pada Game Dragons Dogma

Game	Resolusi	GPU Average (%)	CPU Average (%)	Memory Usage (%)	FPS
Dragons Dogma	800x600	36.1	50.1	3894.8	116.1
Dragons Dogma	1366x768	43.3	52.4	4456.3	87.1

Tabel 4.8 Resource Usage saat GA Terhubung Client pada Game Dragons Dogma

Game	Resolusi	GPU Average (%)	CPU Average (%)	Memory Usage (%)	FPS
Dragons Dogma	800x600	35	56.6	3911.7	115
Dragons Dogma	1366x768	57.3	63.5	4500.6	86.2

4.2 Resource Usage pada Client

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

Tabel 4.9 Resource Usage pada client saat menjalankan game Neverball

Game	Resolusi	GPU Average (%)	CPU Average (%)	Memory Usage (%)	FPS
Neverball	800x600	0	24	684.8	44
Neverball	1366x768	0	25.2	702.2	45

Tabel 4.10 Resource Usage pada client saat menjalankan game Dragons Dogma

Game	Resolusi	GPU Average (%)	CPU Average (%)	Memory Usage (%)	FPS
Dragons Dogma	800x600	0	27.4	893.8	49
Dragons Dogma	1366x768	0	37.8	1008.2	43

4.3 Quality Of Service

Berikut ini merupakan hasil *delay* dan *throughput* pada Game Neverball dan Dragons Dogma. Hasil pengujian ini menggunakan satuan (s) untuk *delay* dan (MB) untuk *throughput*

Tabel 4.10 Hasil pengujian *delay* pada Game Neverball

Games	Resolusi	3 Mbps	5 Mbps	7 Mbps
Neverball	800x600	0.003317	0.003134	0.001860
	1366x768	0.003374	0.003140	0.003068

Tabel 4.11 Hasil pengujian *delay* pada Game Dragons Dogma

Games	Resolusi	3 Mbps	5 Mbps	7 Mbps
Dragons Dogma	800x600	0.003386	0.003070	0.002417
	1366x768	0.003545	0.003480	0.003466

Tabel 4.12 Hasil pengujian *throughput* pada Game Neverball

Games	Resolusi	3 Mbps	5 Mbps	7 Mbps
Neverball	800x600	2,759	2,948	3,006
	1366x768	2,761	2,935	2,940

Tabel 4.13 Hasil pengujian *throughput* pada Game Dragons Dogma

Games	Resolusi	3 Mbps	5 Mbps	7 Mbps
Dragons Dogma	800x600	2,743	2,974	2,994
	1366x768	2,582	2,602	2,615

4.4 Analisis Pengujian Resource Usage

Pada hasil pengujian resource usage, terlihat bahwa perubahan GPU usage meningkat secara signifikan saat GamingAnyWhere ON (aktif). Hal ini dikarenakan pada system GamingAnyWhere

terdapat library Libavcodec yang berfungsi untuk merekam, mengubah dan men-stream audio maupun video di sisi server dan client.

Peningkatan Memory Usage pada game Neverball meningkat sebesar 30.8 MB hingga 145.5 MB. Pada game Dragons Dogma, peningkatan Memory Usage meningkat sebesar 311.9 MB hingga 539 MB. Peningkatan nilai rata-rata disebabkan oleh pengaruh nilai grafis. Jika nilai grafis tersebut tinggi, maka data yang diperlukan oleh game juga tinggi.

Framerate yang dihasilkan pada game Neverball mencapai 59 FPS, hal ini disebabkan karena game Neverball memiliki system requirements yang terbilang ringan. Pada game Dragons Dogma, peningkatan framerate terlihat pada pengaturan grafis low dengan resolusi 1366x768 yang menghasilkan peningkatan sebesar \pm 3 FPS. Namun peningkatan framerate tidak terlihat pada pengaturan grafis high dengan resolusi 1366x768. Hal ini disebabkan oleh game tersebut memiliki system requirements yang besar dan juga pengaturan grafis dan resolusi yang besar. Semakin tinggi pengaturan grafis dan resolusi layar, maka semakin tinggi pula proses rendering yang dilakukan sehingga menghasilkan framerate yang rendah.

Pada sisi *client*, nilai *resource usage* terlihat lebih rendah dari nilai *resource usage* pada sisi server. Hal ini dikarenakan *client* hanya melakukan proses rendering, sedangkan yang melakukan proses *encoding*, dan penyimpanan data *game* berada pada sisi server.

4.5 Analisis Pengujian Quality Of Service

Hasil pengukuran nilai *delay* yang telah dibatasi oleh masing-masing *bandwidth* pada *game* Neverball dan Dragons Dogma ditunjukkan pada tabel 4.10 dan tabel 4.11. Pada kondisi *bandwidth* 3 Mbps, nilai *delay* terlihat lebih besar bila dibandingkan dengan nilai *delay* pada kondisi *bandwidth* 5 Mbps dan 7 Mbps. Hal ini dikarenakan *bandwidth* yang kecil menyebabkan tingginya *traffic*.

Nilai *delay* menurun bersamaan dengan peningkatan *bandwidth* baik di pengaturan grafis *low* maupun di pengaturan grafis *high*. Hal ini dikarenakan pengiriman data lebih lancar jika nilai pembatasan *bandwidth* dinaikkan sehingga waktu pengiriman data lebih cepat.

Pada tabel 4.12 dan 4.13 menunjukkan nilai throughput yang dibatasi oleh masing-masing *bandwidth* pada game Neverball dan Dragons Dogma dengan pengaturan grafis low dan pengaturan grafis high. Nilai throughput pada kedua game tersebut cenderung lebih kecil dari nilai bandwidth baik di pengaturan grafis low maupun di pengaturan grafis high, hal ini dikarenakan nilai throughput menyesuaikan dengan batas nilai bandwidth yang telah ditentukan. Semakin besar throughput, maka semakin cepat pula waktu pengiriman data.

5. Kesimpulan

Dari penelitian yang telah dilakukan, maka didapatkan beberapa kesimpulan sebagai berikut :

1. Implementasi gyroscope sensor telah berhasil dilakukan dan menghasilkan sebuah controller yang berfungsi sebagai command input pada sebuah *game*.
2. GamingAnyWhere mampu menjalankan game yang mempunyai spesifikasi lebih tinggi dari komputer *client*.
3. Data Resource usage pada komputer server saat GamingAnyWhere ON lebih besar dari pada data resource usage saat GamingAnyWhere OFF. Begitu juga dengan data resource usage pada pengaturan grafis high lebih besar dari data resource usage pada pengaturan grafis low.
4. Game dengan pengaturan grafis high memiliki ukuran data yang lebih besar dibandingkan dengan pengaturan grafis low sehingga waktu yang dibutuhkan untuk mengirim data lebih lama. Nilai delay pada game Neverball dan Dragons Dogma saat bandwidth 3 Mbps terhitung lebih besar dari bandwidth 5 Mbps dan 7 Mbps sehingga throughput yang didapatkan pada bandwidth 3 Mbps lebih kecil dari bandwidth 5 Mbps dan 7 Mbps.
5. Kecepatan transfer data untuk dapat memainkan game dengan lancar yaitu minimal sekitar 5 Mbps untuk kedua *game*.

DAFTAR PUSTAKA

- [1] Z. Li, H. Melvin, R. Bruzgine, P. Pocta, L. Skorin-Kapov, and A. Zgank, *Lag compensation for first-person shooter games in cloud gaming*, vol. 10768 LNCS. Springer International Publishing, 2018.
- [2] C. Y. Huang, D. Y. Chen, C. H. Hsu, and K. T. Chen, “Gaminganywhere: An open-source cloud gaming testbed,” *MM 2013 - Proc. 2013 ACM Multimed. Conf.*, pp. 827–830, 2013.
- [3] . S. ., . R. M., . N. F., and I. P. Putra Sutejo, “the the Quality of Services (Qos) of Gaminganywhere on a Game With Speech Recognition System As Command Input,” *J. Elektro dan Telekomun. Terap.*, vol. 5, no. 2, p. 682, 2019.
- [4] R. Munadi, N. Fitriyanti, I. Perdana, and P. Sutejo, “CPU USAGE DARI PENGGUNAAN CLOUD GAMINGANYWHERE PADA GAME DENGAN SPEECH RECOGNITION SYSTEM SEBAGAI COMMAND INPUT,” vol. 4, no. 1, pp. 40–45, 2019.
- [5] H. P. Gupta, H. S. Chudgar, S. Mukherjee, T. Dutta, and K. Sharma, “A Continuous Hand Gestures Recognition Technique for Human-Machine Interaction Using Accelerometer and Gyroscope Sensors,” *IEEE Sens. J.*, vol. 16, no. 16, pp. 6425–6432, 2016.
- [6] A. Herwindita, T. Ariefianto, and M. Iqbal, “Implementasi Game Sebagai Layanan (Gaas) Menggunakan Open-Source Cloud Gaming Server Gaminganywhere,” vol. 1, no. 1, pp. 201–216, 2014.
- [7] A. Budiyanto, “Pengantar Cloud Computing,” *Cloud Indones.* Jakarta, pp. 1–10, 2012.
- [8] C.-Y. Huang, C.-H. Hsu, Y.-C. Chang, and K.-T. Chen, “GamingAnywhere: An Open Cloud Gaming System,” *Proc. 4th ACM Multimed. Syst. Conf. - MMSys '13*, pp. 36–47, 2013.
- [9] B. Firman and T. A. Yogyakarta, “IMPLEMENTASI SENSOR IMU MPU6050 BERBASIS SERIAL I2C PADA SELF-,” no. August 2016, 2017.
- [10] “Neverball”[Online]. Available :
http://ftp.gunadarma.ac.id/linux/magazine/infolinux/Tahun%202005/PDF%20LINUX%201005/24_Ulasan%20Game_10.pdf [Accessed on 19 October 2018, 20:10:51 WIB].
- [11] “Game Neverball”[Online]. Available :
<https://www.mobygames.com/game/neverball/techinfo> [Accessed on 19 October 2018, 21:15:32 WIB].
- [12] “Dragon’s Dogma: Dark Arisen System Requirements”[Online]. Available :
https://www.game-debate.com/games/index.php?g_id=6502&game=Dragons%20Dogma:%20Dark%20Arisen [Accessed on 21 October 2018, 17:10:11 WIB].
- [13] “Dragon’s Dogma: Dark Arisen”[Online]. Available :
https://en.wikipedia.org/wiki/Dragon%27s_Dogma [Accessed on 20 September 2019, 12:55:50 WIB]
- [14] “Arduino Pro Micro”[Online]. Available :
https://deskthority.net/wiki/Arduino_Pro_Micro [Accessed on 20 September 2019, 13:05:07 WIB].
- [15] L. Media, F. O. R. Robotics, and P. Course, “PENGEMBANGAN HAND GLOVE CONTROL MENGGUNAKAN SENSOR MPU6050 SEBAGAI MEDIA PEMBELAJARAN MATA KULIAH PRAKTIK ROBOTIKA DEVELOPMENT OF HAND GLOVES CONTROL USING MPU6050 SENSOR AS A,” no. 3, pp. 591–600.

