JURNAL INFORMATIKA, Vol.9 No.1 April 2022, Halaman 32-42
ISSN: 2355-6579 | E-ISSN**:** 2528-2247

Analisis Latensi Pada Aplikasi *Virtual Reality* Untuk Anak Dengan *Autism Syndrome Disorder*

Muhammad Ariq Rafli[1], Meredita Susanty[2]

[1,2] Universitas Pertamina
Jl Teuku Nyak Arief, Simprug, Kebayoran Lama, Jakarta Selatan 12220, Indonesia

e-mail: [1]arqrfly@gmail.com, [2]meredita.susanty@universitaspertamina.ac.id

Informasi Artikel	Diterima: 29-12-2021	Direvisi: 18-01-2022	Disetujui: 21-01-2022

Abstrak

Virtual Reality (VR) atau realitas maya adalah sebuah cabang media interaktif yang terus berkembang. VR merupakan teknologi yang dapat membuat pengguna dapat berinteraksi dengan lingkungan dalam dunia maya yang disimulasikan oleh komputer. VR membuat pengguna merasa berada di dalam lingkungan tersebut. Dalam mengembangkan VR kita harus memiliki pengetahuan tentang teknologi sensing and tracking, stereoscopic display, multimodal interaksi dan recognition (contoh : metafora dan menu tiga dimensi, suara, dan gerakan), grafis komputer, simulasi fisika dan lainya. Dengan teknologi-teknologi tersebut, VR dapat memanipulasi otak penggunanya seakan-akan berada pada dunia lain. Karena membutuhkan tingkat grafis visual yang tinggi untuk memberikan pengalaman yang realistis, salah satu tantangan utama dalam mengembangkan aplikasi permainan VR adalah latensi yang tinggi. Latensi yang tinggi mengakibatkan menurunnya kualitas pengalaman pengguna, karena ada waktu tunda antara aksi dari pengguna dan respon pada aplikasi. Latensi dalam penggunaan VR bahkan dapat menyebabkan sensasi mabuk. Penelitian ini bertujuan mengembangkan aplikasi VR dengan latensi rendah dengan studi kasus aplikasi permainan untuk anak dengan Autism Spectrum Disorder (ASD) dengan tetap mempertahankan kualitas grafik visual yang baik. Pada penelitian ini, nilai latensi per komponen kualitas visual dan nilai kualitas performa aplikasi diukur menggunakan tools profiler dari Unity. Hasil pengujian menunjukkan bahwa penurunan nilai masing-masing komponen visual seperti anti-aliasing, shadow resolution, cascades dan soft shadow dapat mengurangi latensi. Diantara komponen visual tersebut, anti-aliasing yang berkontribusi paling besar terhadap latensi. Penelitian ini berhasil mendapatkan kombinasi optimum diantara komponen-komponen visual tersebut untuk mendapatkan latensi yang rendah dan grafik visual yang baik.

Kata Kunci: realitas maya, latensi, unity, grafik visual

Abstract

Virtual Reality (VR) is a branch of interactive media that constantly evolves. VR is a technology that allows users to interact with the environment in a virtual world simulated by a computer. VR makes users feel like they are in the environment. In developing VR, we must have knowledge of sensing and tracking, stereoscopic display, multimodal interactions and recognition (example: metaphors and three-dimensional menus, sound, and motion), computer graphics, physics simulation and others. With these technologies, VR can manipulate the user's brain as if they were in a different world. Because it requires a high level of visual graphics to provide a realistic experience, one of the main challenges in developing VR gaming applications is high latency. High latency results in a decreased quality of the user experience, as there is a time delay between the user's action and the application's response. Latency in VR use can even lead to a headache. This study aims to develop a low-latency VR application with a case study of a game application for children with Autism Spectrum Disorder (ASD) while maintaining good visual graphic quality. This study measured the value of latency per visual quality component and application performance quality using a profiler from Unity. The test results show that decreasing the value of each visual component, such as anti-aliasing, shadow resolution, cascades, and soft shadow, will reduce latency. Among these visual components, anti-aliasing contributes the most to latency. This study obtained the optimum combination of these visual components to obtain low latency and good visual graphics.

Keywords*: virtual reality, latency, Unity, visual graphic*

1. Pendahuluan

Virtual Reality (VR) atau realitas maya adalah sebuah cabang media interaktif yang terus berkembang . VR merupakan teknologi yang dapat membuat pengguna dapat berinteraksi dengan lingkungan dalam dunia maya yang disimulasikan oleh komputer. VR membuat pengguna merasa berada di dalam lingkungan tersebut. Dalam mengembangkan VR kita harus memiliki pengetahuan tentang teknologi *sensing and tracking, stereoscopic display*, interaksi multimodal dan *recognition* (contoh : metafora dan menu tiga dimensi, suara, dan gerakan), grafis komputer, simulasi fisika dan lainya (Capilla & Martínez, 2004). Dengan teknologi-teknologi tersebut, VR dapat memanipulasi otak penggunanya seakan-akan berada pada dunia lain.

Dalam pengembangan aplikasi VR, terdapat banyak tantangan. Menurut pengembang pada Vive Team, permasalahan pada pengembangan VR adalah kurangnya alat-alat untuk pengembangan seperti server masih belum kuat, *game engine* komersial yang masih memiliki keterbatasan seperti sedikitnya opsi untuk *user interface* (UI) yang bagus. Lalu menurut Gamasutra (Christian Nutt, 2013), tantangan utamanya yaitu membuat desain UI yang bagus, penyakit simulator, dan latensi. Penyakit simulator adalah sensasi mabuk yang dikarenakan apa yang dilihat mata tidak sesuai dengan gerakan pada tubuh. Hal tersebut terjadi karena di dalam VR, badan pengguna bergerak sedangkan aslinya badan tidak bergerak sama sekali. Ketidaksesuaian antara apa yang dilihat dengan gerakan tubuh menjadi penyebab penyakit simulator maupun efek negatif seperti pusing dan mata tegang. Ketidaksesuaian ini disebabkan karena adanya latensi pada VR. Latensi adalah penundaan antara aksi dan reaksi yang merupakan musuh terbesar pada VR. Latensi menggambarkan waktu antara pergerakan objek yang dilacak dan pergerakannya yang sesuai dengan yang digambarkan di komputer melalui output grafikal.

Memastikan latensi yang rendah merupakan hal yang krusial dalam sebuah pengembangan aplikasi permainan VR. Untuk memberikan pengalaman yang realistis, VR harus memiliki tingkat grafis visual yang tinggi. Di sisi lain, tingkat grafis visual yang tinggi dapat menimbulkan latensi yang tinggi pula. Latensi yang direkomendasikan adalah 20ms (Raaen & Kjellmo, 2015). Untuk dapat mencapai nilai ini pendekatan yang umumnya dilakukan adalah menurunkan tingkat grafis visual. Solusi tersebut dapat dibilang berhasil jika pengguna tidak merasakan latensi atau bahkan mendapatkan penyakit simulator dari aplikasi permainan yang dikembangkan.

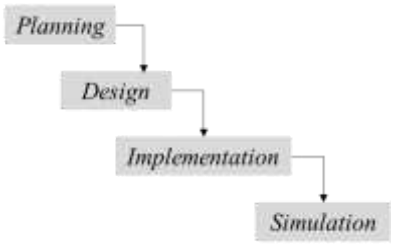
Karena mampu mensimulasikan dunia nyata, VR memiliki potensi besar sebagai media pembelajaran interaktif. Penelitian ini menggunakan studi kasus aplikasi VR untuk melatih kemampuan sosial anak dengan ASD. Penggunaan VR untuk terapi anak dengan ASD sudah dilakukan sejak pertengahan tahun 1990. VR dapat membantu anak dengan ASD dengan cara menyediakan sebuah *virtual environment* (VE) agar anak tersebut dapat mempersiapkan diri pada situasi yang dapat membuat stress (George Musser, 2018). VR mampu memberikan latihan skenario umum sosial sehari-hari yang aman dan terkontrol dengan jumlah yang tidak terbatas. VR juga sudah digunakan untuk menggantikan metode-metode yang digunakan oleh tenaga ahli psikologi. Salah satu metode terapi tersebut bernama Sally-Anne di mana metode tersebut digunakan untuk memahami keyakinan, maksud, dan emosi orang tersebut. Pada metode ini partisipan dapat melihat sebuah percakapan antara dua boneka. Lalu, partisipan akan disuruh untuk memprediksi perilaku si boneka tersebut (Bekele et al., 2013; Nathan Caruana & Jon Brock, 2017). Beberapa ilmuwan juga menggunakan VR sebagai alat untuk role-playing dalam melatih kemampuan sosial (Ke et al., 2020). Banyak kemampuan yang bisa dipelajari oleh anak dengan ASD menggunakan VR, seperti *public speaking*, wawancara, dan aktivitas sosial lainnya (Bellani et al., 2011; Karami et al., 2021).

Penelitian ini menggunakan aplikasi permainan VR sebagai studi kasus untuk melakukan analisis latensi. Aplikasi permainan ditujukan untuk melatih kemampuan sosial anak dengan ASD. Karena itu visual yang ditampilkan harus semirip mungkin dengan keadaan aslinya. Faktor latensi juga menjadi hal yang sangat penting karena kecenderungan anak ASD yang lebih peka terhadap ketidaksesuaian apa yang dilihat dan dirasakan tersebut (Cindy Hatch-Rasmussen, n.d.). Meskipun anak dengan ASD cenderung lebih peka, penelitian sebelumnya (Sahin et al., 2018) menunjukkan hanya tiga dari enam belas partisipan yang merasakan efek negatif seperti pusing dan mata tegang saat menggunakan VR, Karena itu, dalam penelitian ini latensi yang ingin dicapai sama dengan latensi yang digunakan pada umumnya berkisar 20ms. Permasalahan utama yang akan diteliti dalam penelitian ini adalah bagaimana memastikan kedua faktor tersebut memiliki kualitas yang baik. Lebih spesifik, bagaimana mendapatkan latensi yang rendah dan tingkat grafis visual yang tinggi dan mencari tahu bagaimana pengaruh komponen-komponen kualitas visual terhadap latensi aplikasi permainan pada komputer dengan spesifikasi

minimum penggunaan *virtual reality headset* Oculus Rift S.

2. Metode Penelitian

Alur kerja penelitian akan dilakukan sesuai dengan diagram alir pada Gambar 1. Fase pertama yaitu *planning* yang bertujuan untuk melakukan perencanaan proses pengembangan. Kemudian, fase *design* yang bertujuan untuk melakukan perancangan model dan kelas untuk implementasi. Selanjutnya adalah tahap *implementation* di mana pengembang melakukan implementasi dari rancangan yang telah dibuat. Dan terakhir adalah fase *simulation* di mana dilakukan simulasi aplikasi untuk mengukur latensi, menentukan komponen kualitas visual mana yang berpengaruh dan mengukur kualitas performa dari aplikasi permainan.



Gambar 1. Diagram Alir Proses Pengembangan

2.1 *Planning*

Pada fase ini pengembang mengumpulkan beberapa set tugas yang akan dilakukan berdasarkan dokumen fungsional dan non-fungsional. Sebuah tugas dapat terdiri dari beberapa tugas-tugas yang kecil dan dikategorikan. Aktivitas pada proses *planning* ini terdiri dari melakukan perkiraan waktu pengerjaan *user-stories*, penentuan prioritas *user-story*, dan perencanaan pengembangan untuk tiap iterasi. Proses penentuan prioritas ditentukan dengan *story points*. *Story points* ditentukan berdasarkan *values* dan *risk*. *Values* terbagi menjadi tiga yaitu *critical*, *significant business value*, dan *nice to have*. Sedangkan *risk* ditentukan berdasarkan *risk index* yang ditentukan berdasarkan faktor *completeness*, *volatility*, dan *complexity*. Indeks tersebut kemudian ditentukan dari rendah yang dimulai dari 0 hingga 6 sebagai *high*. Sebuah iterasi ditentukan dari jumlah enam *story point*. Dimana satu point dikerjakan selama dua hari.

2.2 *Design*

Pada fase ini pengembang membuat rancangan model dan kelas yang akan diimplementasikan. Metode yang digunakan dalam fase ini yaitu *class, responsibilities, and collaborators* (CRC) card. CRC card berguna untuk merepresentasikan tanggung jawab sebuah kelas dan interaksi antar kelas. Sebuah kelas ditentukan dengan cara menganalisa user-stories. Kelas tersebut merepresentasikan sebuah objek. Interaksi antar kelas ditentukan dengan apa kegunaan dari kelas tersebut beserta atributnya.

2.3 *Implementation*

Fase ini adalah fase dimana code terbentuk. Pada fase ini terdapat 3 tahap yaitu *code generation*, *unit testing*, dan *code refactor*. Sebelumnya dilakukan pemilihan *technology stack* untuk dua kebutuhan; pemodelan 3D dan *game engine*. Pada pemodelan 3D terdapat beberapa pilihan aplikasi seperti Autodesk 3DS Max, Blender, dan Autodesk Maya. Dalam penelitian ini, aplikasi pemodelan 3D yang dipilih adakah Blender karena aplikasi yang bersifat open source dan memiliki komunitas yang saling membantu sehingga terdapat banyak *resource* untuk pembelajaran. Karena keterbatasan Blender untuk membuat karakter 3D dengan ekspresi wajah, untuk pemodelan 3D karakter digunakan VRoid Studio. VRoid Studio merupakan aplikasi pemodelan avatar humanoid 3D gratis yang sifat penggunaanya dapat berupa komersil dan non-komersil. Dengan VRoid studio, Karakter 3D yang bervariasi dan memiliki ekspresi dapat dibuat dengan waktu yang cepat. VRoid Studio telah menyediakan modul ekspresi wajah yang berguna untuk terapi empati anak dengan ASD.

Tabel 1. Spesifikasi *hardware* komputer

Komponen	Spesifikasi
Graphic card	NVIDIA GTX 1050Ti / AMD Radeon RX 470
Alternate graphic card	NVIDIA GTX 960 / AMD Radeon R9 290
CPU	Intel i3-6100 / AMD Ryzen 3 1200, FX4350
Memory	8GB+ RAM
Video Output	Compatible HDMI 1.3 video output
USB Ports	1x USB 3.0 port, plus 2x USB 2.0 ports
OS	Windows 10

Aplikasi permainan VR akan dikembangkan menggunakan perangkat Oculus Rift S. Diantara berbagai tools yang digunakan yakni Oculus Rift S(*Oculus Rift S and Rift Minimum Requirements and System Specifications*, n.d.), Unity (*Unity - Manual: System Requirements for Unity 2020 LTS*, n.d.), Vroid studio (*VRoid Studio*, n.d.), dan Blender (*Requirements — Blender.Org*, n.d.), yang membutuhkan spesifikasi paling tinggi adalah Oculus Rift S. Karena itu spesifikasi minimum dalam penelitian ini mengikuti

spesifikasi minimum Oculus Rift S. Tabel 1 menunjukkan spesifikasi hardware komputer minimum yang dibutuhkan.

2.3 *Simulation*

Pada tahap *simulation*, akan dilakukan tiga percobaan. Percobaan pertama bertujuan mencari tahu prasetel kualitas visual mana yang memiliki tingkat kualitas visual tertinggi dengan latensi mendekati latensi rekomendasi yaitu 20 ms. Pada percobaan pertama satu ada empat prasetel kualitas visual yang dibandingkan yaitu *High Quality* dengan *Post-Processing, High Quality, Medium Quality*, dan *Low Quality*. Komponen kualitas visual yang aktif pada masing-masing preset dapat dilihat pada Tabel 2. *Post-processing* yang digunakan pada prasetel pertama ditunjukkan pada Tabel 3. Latensi pada prasetel tersebut akan diukur dengan meletakan posisi kamera di dalam permainan ke seluruh objek didalam *scene*. Di dalam *scene* terdapat 6 karakter dengan animasi, objek statis dan dinamis, dan UI. Tujuan kamera diposisikan ke seluruh objek agar mendapatkan latensi terberat karena CPU harus mengerjakan semua *task* (*rendering, script, physics*, dan *animasi*). Nilai latensi didapatkan dengan *tools Profiler* pada unity pada saat aplikasi dijalankan. Nilai tersebut akan dianalisis dan prasetel yang terbaik akan menjadi standar untuk tahap selanjutnya.

Tabel 2. Prasetel Kualitas

Kualitas	HDR	*Anti Aliasing*	*Cast Shadow*	*Shadow Resolution*	*Shadow Distance*	*Cascades*	*Soft Shadow*
High quality + post-processing	*On*	4x	*On*	2048	100	2	*On*
High quality	*On*	2x	*On*	2048	100	2	*On*
Medium quality	*On*	2x	*On*	1024	50	2	*Off*
Low quality	*Off*	0	*Off*	0	0	0	*Off*

Tabel 3. *Post-processing* pada Prasetel Pertama (*Unity - Manual: Post-Processing*, 2020)

Proses	Deskripsi
Tone mapping	Memetakan satu set warna ke yang lain untuk memperkirakan penampilan gambar dengan rentang dinamis tinggi dalam media yang memiliki rentang dinamis lebih terbatas.
Bloom	Mereproduksi penangkapan cahaya pada kamera di dunia nyata.
Vignette	Memberikan efek gelap pada ujung-ujung layar untuk membuat aplikasi permainan menjadi lebih sinematik.
Depth of Field	Efek memburamkan background sehingga memberikan pandangan fokus kepada benda terdekat.

Percobaan kedua bertujuan mencari tahu komponen kualitas visual yang paling berpengaruh terhadap latensi. Percobaan ini dilakukan dengan mengubah nilai masing-masing komponen kualitas visual pada suatu kategori prasetel dan membandingkan nilai latensi yang didapatkan dengan prasetel yang memiliki nilai latensi dibawahnya. Tujuan masing-masing komponen kualitas visual tersebut diuji agar dapat mengetahui komponen mana yang dapat meningkatkan performa dari prasetel terbaik tetapi kualitas visual tidak lebih rendah dari prasetel di bawahnya. Metode penelitian ini dilakukan dengan cara menjalankan aplikasi dengan prasetel terbaik lalu merubah komponen kualitas visualnya diubah dengan opsi yang tersedia. Setiap perubahan opsi tersebut, nilai latensi akan dicatat untuk dianalisis. Nilai latensi didapatkan pada *tools* Profiler pada unity. Hasil penelitian ini adalah mengetahui prasetel manakah yang terbaik untuk aplikasi permainan ini, mengetahui komponen kualitas visual apa yang mempengaruhi latensi, dan menemukan cara untuk meningkatkan performa dari prasetel terbaik.

$$D = \frac{R+W}{Max} \quad (1)$$

D = Pemanfaatan Disk
R = *Read Disk per Second*
W = *Write Disk per Second*
Max = Maksimum Penggunaan Disk

Percobaan ketiga bertujuan mengukur performa aplikasi permainan untuk memastikan kualitas perangkat lunak (Fischer et al., 2010). Pengukuran performa dilakukan dengan pengaturan kualitas visual dan latensi terbaik yang didapatkan berdasarkan hasil percobaan kedua. Terdapat dua pengukuran yang akan dilakukan yaitu mengukur pemakaian CPU untuk menentukan apakah aplikasi menyebabkan *bottleneck*. Sebuah CPU disebut menjadi bottleneck jika pemakaian CPU pada saat aplikasi dijalankan mencapai 100%. Pengukuran kedua yaitu pengukuran pemakaian *disk*. Pengukuran ini menentukan apakah

aplikasi meningkatkan pemakaian *disk* yang melebih rata-rata pemakaian disk pada komputer berdasarkan persamaan 1. Untuk mendapatkan data dari kerja CPU dan *disk*, digunakan aplikasi *Performance Monitor* yang telah disediakan Windows. Untuk menentukan rata-rata kecepatan *read* dan *write disk* digunakan aplikasi AS SSD Benchmark.

3. Hasil dan Pembahasan

Berdasarkan hasil pengumpulan *requirement* dengan psikolog klinis anak, didapatkan permasalahan-permasalahan yang umumnya dihadapi oleh anak ASD yang kemudian diturunkan menjadi tujuh *user story* seperti yang dijelaskan pada Tabel 4. Permasalahan yang umum dihadapi oleh anak autis antara lain; (1) takut bertemu dengan orang baru dan berada pada lingkungan baru, (2) stress dengan perubahan kecil, (3) sulit dipengaruhi karena memiliki ketertarikan yang berbeda dengan orang lain, (4) sulit mengendalikan emosi, (5) hipo atau hiper reaktif terhadap input sensorik tertentu, (6) memiliki minat yang spesifik, (7) sulit membedakan makro dan mikro ekspresi, (8) kemampuan menyusun kalimat tergantung pada level keparahan, (9) kebingungan untuk memberikan respon terhadap emosi orang disekitar, (10) tidak merespon perilaku sosial lingkungan sekitar. Berdasarkan berbagai permasalahan ini, dilakukan prioritas untuk mengatasi permasalahan nomor (1), (7), (9) dan (10) dengan tetap memperhatikan pemasalahan nomor (5) dan (6) dalam mendesain permainan. Pada tahapan *planning* dilakukan pengelompokan dan pemeringkatan prioritas dan perencanaan pengembangan tiap iterasi dengan hasil yang ditunjukkan pada Tabel 5.

Pada tahap *design,* kelas-kelas yang dibutuhkan untuk mengakomodir semua *user stories* dimodelkan menggunakan CRC card seperti yang ditunjukkan pada Gambar 2.

Pada tahap implementasi, dibuat class diagram berdasarkan CRC diagram yang dihasilkan pada tahap desain. Class diagram yang dibuat ditunjukkan pada Gambar 3. Kode program dihasilkan dari *code generator* berdasarkan class diagram yang didefinisikan. Gambar 4 menunjukkan potongan program yang dihasilkan oleh *code generator* untuk class Lokasi.

Tabel 4. Daftar *User Stories*

Kode	*User Stories*
S1	Pengguna memilih opsi lokasi permainan, agar anak dengan ASD bisa merasakan skenario-skenario sosial kehidupan sehari-hari lainnya.
S2	*virtual enviroment* adalah lingkungan yang paling sering dihadapi anak, sehingga pemain dapat melatih kemampuan empati sehari-harinya.
S3	Pada level 1, terdapat minimal enam karakter dengan ekspresi berikut secara acak; marah, sedih, senang, kaget, netral.
S4	Sebagai pemain, saya ingin pilihan jawaban untuk pertanyaan dalam bentuk wajah tokoh tertentu dalam berbagai ekspresi.
S5	Sebagai pemain, saya ingin mendapatkan umpan balik atas jawaban yang saya pilih.
S6	Sebagai pemain, saya ingin mendapatkan *reward* jika berhasil menjawab satu pertanyaan dengan benar
S7	Sebagai pemain, saya ingin mendapatkan *reward* jika berhasil menyelesaikan satu level permainan

Tabel 5. Hasil Analisis *Planning*

Iterasi	*User Story*	*Deskripsi Pekerjaan*	*Value*	*Risk*	*Story Point*
1	S2	Membuat VE lokasi rumah	Critical	M	3
	S3	Membuat objek dan animasi karakter Ayah dengan ekspresi marah	Critical	M	0,5
	S3	Membuat objek dan animasi karakter Kakek dengan berbagai ekspresi	Critical	M	0,5
	S3	Membuat objek dan animasi karakter Ibu dengan berbagai ekspresi	Critical	M	0,5
	S3	Membuat objek dan animasi karakter Nenek dengan berbagai ekspresi	Critical	M	0,5
	S3	Membuat objek dan animasi karakter Kakak dengan berbagai ekspresi	Critical	M	0,5
	S3	Membuat objek dan animasi karakter Adik dengan berbagai ekspresi	Critical	M	0,5
2	S4, S5	Membuat UI untuk pertanyaan dan pilihan jawaban dan respon terhadap jawaban	Critical	M	3

S6, S7	Membuat *reward* berupa *confetti*	Significant Business Value	M	2
S1	Membuat pilihan lokasi permainan	Significant Business Value	M	1

Aktivitas selanjutnya pada tahap implementasi adalah melakukan *unit testing*. Terdapat dua pengujian pada aplikasi. Pengujian pertama adalah pergantian *scene* dari menu ke RuangTamu dengan menjalankan script PilihLokasi. Code dapat dilihat pada tabel 4.8. Pengujian pilih lokasi mendapatkan hasil sukses. Pengujian kedua adalah menguji apakah UI sudah dapat bekerja sesuai dengan yang diinginkan. Code dapat dilihat pada tabel 4.9. Pengujian UI mendapatkan hasil sukses. Aktivitas selanjutnya adalah code refactoring. Karena hasil dari pengujian mendapatkan sukses, maka tidak ada perlu code refactoring. Pada tahap implementasi, berhasil dibuat VE, objek 3D, dan karakter 3D menggunakan Unity, Blender dan VRoid Studio seperti yang ditunjukkan pada Gambar 5.

Selanjutnya pada tahap simulasi, dilakukan pengukuran latensi untuk tiga prasetel. *Hardware* komputer yang digunakan untuk melakukan simulasi ditunjukkan pada Tabel 6.

Tabel 6. Spesifikasi *Hardware* Komputer untuk Simulasi

Komponen	Spesifikasi
Graphic card	NVIDIA GTX 1050Ti / AMD Radeon RX 470
CPU	Intel i5-2400
Memory	8GB RAM
OS	Windows 10

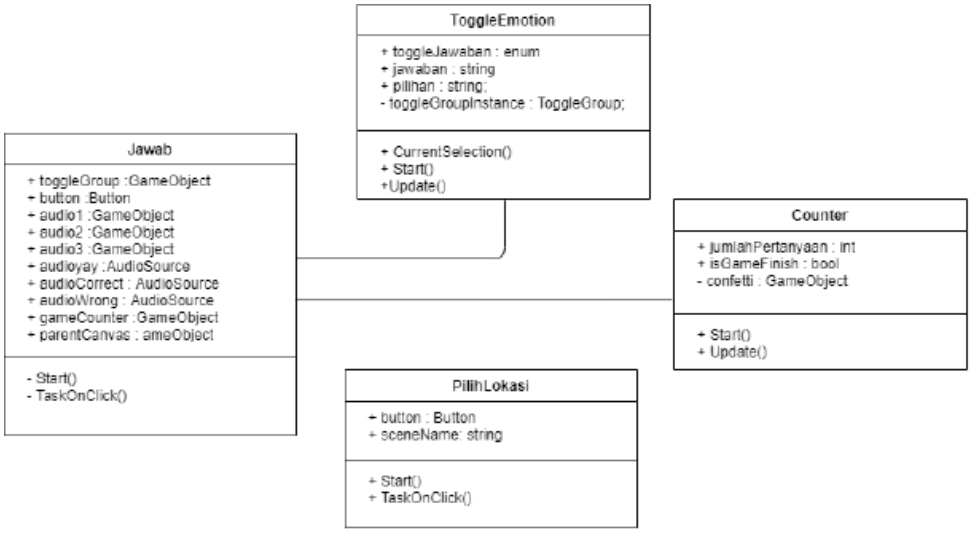
Tahap pertama pada simulasi adalah mengukur latensi menggunakan prasetel yang sudah ditentukan pada Tabel 2 menggunakan *tools* Profiler Unity. Berdasarkan hasil simulasi pada Tabel 7, didapatkan prasetel kualitas yang memiliki visual tertinggi dengan latensi rekomendasi adalah prasetel *high quality*. Prasetel *high quality* mendapatkan nilai latensi 21 ms. Prasetel tersebut dipilih karena memiliki perbedaan latensi hanya 5% dari yang direkomendasikan atau 1 ms. Sehingga prasetel *high quality* dapat dijadikan sebagai prasetel terbaik.

Tabel 7. Spesifikasi *hardware* komputer

Kualitas	Frame per Second (FPS)	Latensi (ms)
High quality + post-processing	21,5	46,5
High quality	46,9	21,3
Medium quality	93,5	10,7
Low quality	112,4	8,9

Tahap kedua pada simulasi adalah menganalisis komponen kualitas yang memengaruhi latensi dan bagaimana cara menurunkan latensi dari prasetel terbaik tanpa mengorbankan kualitas visual yang banyak. Simulasi dilakukan dengan melakukan perubahan terhadap setiap komponen. Tabel 8 adalah hasil dari simulasi per masing-masing komponen. Pada Tabel 8 baris yang diberikan highlight biru merupakan prasetel standarnya, kolom yang diberi warna abu-abu adalah komponen kualitas yang diubah pada setiap percobaan dan kolom yang tidak diberi warna adalah komponen kualitas yang nilainya tetap sama dengan nilai standar. Dari simulasi tersebut didapatkan tidak semua komponen kualitas visual memiliki pengaruh terhadap latensi seperti HDR dan *shadow distances*. Pada HDR, tidak terjadi perubahan baik latensi maupun dalam visual karena HDR hanya dapat digunakan bila menggunakan *post-processing* sedangkan pada prasetel ini *post-processing* tidak dilakukan. Pada *shadow distances* tidak terdapat perubahan latensi karena walaupun rentang jaraknya berbeda tetapi bayangan tetap di *render* dengan kapasitas CPU yang sama.

Gambar 2. CRC Card



Gambar 3. Class Diagram

Script	`PilihLokasi`

`public Button button;`
`public string sceneName;`
`void Start()`
`{`
`button = gameObject.GetComponent<Button>();`
`button.onClick.AddListener(TaskOnClick);`
`}`
`// Update is called once per frame`
`void TaskOnClick()`
`{`
`SceneManager.LoadScene(sceneName);`
`}`	

Deskripsi	• `Start()` : Mencari objek-objek dan komponen dari objek dan melakukan *casting* ke variabel lokal. • `TaskOnClick()` : Mengganti Scene berdasarkan pilihan yang di klik

Gambar 4. Contoh Potongan Program

Gambar 5. Hasil Implementasi untuk Sebuah VE

Komponen kualitas visual yang berpengaruh adalah *anti-aliasing, cast shadow, shadow resolution, cascades* dan *soft shadow*. *Anti-aliasing* memiliki pengaruh yang cukup besar terhadap latensi. *Anti-aliasing* terendah (0x) mampu memberikan latensi hingga 9 ms. Sedangkan *anti-aliasing* tertinggi (8x) mampu memberikan latensi hingga 260 ms. Perubahan

dari *anti-aliasing* dapat dilihat secara jelas. Perubahan tersebut adalah ujung-ujung objek lebih bergerigi atau lebih mulus. Pada komponen *cast shadow* perubahan latensi tidak terlalu besar. Tetapi perubahan visual terlihat dengan jelas. Perubahannya adalah tidak adanya bayangan pada VE. Pada komponen *shadow resolution* , terdapat perubahan latensi yang tidak terlalu besar. Semakin rendah *shadow resolution* semakin pecah-pecah bayangan yang dihasilkan. Tetapi merendahkan *shadow resolution* dapat meningkatkan latensi hingga 5%. Pada komponen *cascades*, terdapat perubahan jika jumlah *cascades* dinaikkan. Sedangkan jika kualitas visual *cascades* diturunkan tidak terjadi perubahan apa-apa pada latensi. Hal tersebut disebabkan tidak terdapat banyak objek dinamis. Dan terakhir pada komponen *soft shadow* terdapat perubahan yang tidak terlalu besar. Perbedaan secara visualnya adalah bayangan lebih bergerigi.

Tabel 8. Hasil Percobaan Perubahan Komponen Kualitas Visual

HDR	*Anti Aliasing*	*Cast Shadow*	*Shadow Resolution*	*Shadow Distance*	*Cascades*	*Soft Shadow*	Latensi (ms)
On	2x	On	1024	100	2	On	21,3
Off	2x	On	1024	100	2	On	21,3
On	0x	On	1024	100	2	On	9,3
On	4x	On	1024	100	2	On	21-23
On	8x	On	1024	100	2	On	260-290
On	2x	Off	1024	100	2	On	19-20
On	**2x**	**On**	**256**	**100**	**2**	**On**	**19-20**
On	2x	On	512	100	2	On	20-21
On	2x	On	2048	100	2	On	23-25
On	2x	On	4096	100	2	On	24-26
On	2x	On	1024	0	2	On	21,3
On	2x	On	1024	50	2	On	21,3
On	2x	On	1024	150	2	On	21,3
On	2x	On	1024	200	2	On	21,3
On	2x	On	1024	100	0	On	21,3
On	2x	On	1024	100	4	On	25
On	2x	On	1024	100	2	Off	19-20

Performa prasetel kualitas *High Quality* dapat ditingkatkan dengan cara merendahkan salah satu dari nilai *anti-aliasing, cast shadow, shadow resolution, cascades* dan *soft shadow*. Hasil analisis dari simulasi diatas adalah prasetel *high quality* cocok digunakan pada aplikasi permainan VR yang dikembangan dengan modifikasi *shadow resolution* diturunkan dengan pengaturan terbaik ditunjukkan dengan cetak tebal pada Tabel 8. Tujuan *shadow resolution* diturunkan adalah walaupun bayangan bergerigi tetapi terdapat komponen kualitas visual *soft shadow* sehingga bayangan tersebut tidak menjadi terlalu bergerigi.

Tabel 9. Performa Selama Aplikasi Digunakan

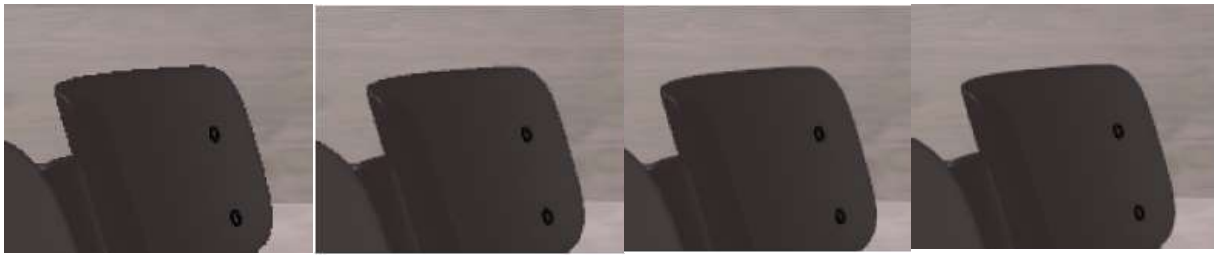
	Pemakaian CPU
Minimum	14%
Rata-rata	40%
Maksimum	60%

Tahap ketiga dalam simulasi adalah mengukur kualitas performa dari aplikasi permainan. Kualitas pertama yang diukur adalah pemakaian CPU. Data pemakaian CPU dapat dilihat pada Tabel 9. Berdasarkan data pada Tabel 9, rata-rata dan nilai maksimum memiliki nilai dibawah 100% sehingga dapat disimpulkan aplikasi permainan tidak membuat CPU menjadi *bottleneck*. Pengukuran kedua yaitu mengukur pemakaian *disk* saat permaian dijalankan. Rata-rata *read disk* adalah 512.12 MB/s dan write disk adalah 457 MB/s. Dengan data tersebut dapat dihutung maksimum pemanfaatan disk yang diasumsikan kurang lebih 484.56 MB/s. Menggunakan *performance monitor*, didapatkan pemakaian *write disk* sebesar 169.472MB/s dan *read disk* sebesar 19.382 MB/s. Sehingga nilai pemanfaatan *disk* yang didapatkan menggunakan persamaan 1 adalah 0.389 atau 38.9% dari pemakaian disk secara keseluruhan. Kualitas performa pemanfaatan disk dapat dikatakan baik karena tidak mencapai batas pemakaian maksimum *disk* atau melebihi 100%.

Pada percobaan pertama didapatkan prasetel kualitas visual terbaik adalah high quality dengan latensi 21.3 ms. Nilai latensi tersebut didapatkan pada saat aktivitas terberat permainan. Aktivitas terberat permainan adalah pada saat semua objek, cahaya, animasi, dan fisik di-render sekaligus dalam satu *frame*. Nilai latensi 21.3 ms tersebut dapat diterima karena diukur pada saat aktivitas terberat. Sehingga pada saat aktivitas lebih ringan, aplikasi permainan mampu mendapatkan latensi yang mencapai latensi rekomendasi.

Setelah melakukan percobaan kedua pada tahap simulasi, didapatkan bahwa *anti-aliasing* merupakan komponen kualitas visual yang paling berdampak terhadap latensi. Hal tersebut terjadi karena *anti-aliasing* menciptakan resolusi gambar yang tinggi (Brent Hale, 2019). Pada sebuah gambar pada komputer, terdapat detail yang tidak terbatas (Beets & Barron, 2000). Komputer pada umumnya memiliki keterbatasan perangkat keras untuk menampilkan detail tersebut. Komputer pada saat ini tidak mampu melakukan *render* pada jumlah piksel yang tidak terbatas. Maka komputer mencari cara lain untuk menyelesaikan permasalahan tersebut yaitu dengan *anti-aliasing*.

Alasan mengapa *anti-aliasing* memiliki dampak terbesar pada latensi karena terdapat banyak piksel yang harus dihaluskan agar tidak terlihat bergerigi (Beets & Barron, 2000). *Anti-aliasing* juga melakukan penggabungan antara dua warna untuk menghasilkan warna yang dapat menyatu. Semakin tinggi nilai *anti-aliasing* semakin tinggi juga kerja yang harus dilakukan oleh komputer untuk menghaluskan masing-masing piksel tersebut. Hasil dari perubahan *anti-aliasing* dapat dilihat pada Gambar 6. Jika nilai anti-aliasing diturunkan resolusi gambar terlihat bergerigi seperti ditunjukan pada gambar 6.a. Sedangkan jika nilai anti-aliasing dinaikan gambar tersebut tidak terlihat bergerigi tetapi menjadi lebih buram seperti ditunjukan pada gambar 6.d.



(a) (b) (c) (d)

Gambar 6. Perbandingan resolusi gambar antar nilai anti-aliasing (a) 0x, (b) 2x, (c) 4x, (d) 8x

(a) (b) (c)

Gambar 7. Perbandingan visual komponen *shadow resolution* (a) 256, (b) 512, (c) 1024 dengan komponen *soft shadow* Diaktifkan

Untuk mengatasi peningkatan latensi akibat *anti-aliasing*, terdapat beberapa solusi yang dapat dilakukan. Pertama yaitu dengan menurunkan nilai komponen kualitas visual yang lain seperti *shadow resolution*. Alasan *shadow resolution* diturunkan karena adanya komponen kualitas *soft shadow* yang dapat memperbaiki kualitas visual. Dengan *soft shadow* kualitas gambar dari bayangan dengan resolusi rendah terlihat tetap bersih dan masih terlihat seperti bayangan nyata. Nilai *shadow resolution* dapat diturunkan menjadi 256 dan 512. Perubahan bayangan tersebut dapat dilihat pada Gambar 7 dimana perbedaan kualitas visual tidak terlalu kentara. Jika solusi ini diterapkan pada prasetel *high quality*, latensi yang didapatkan bisa menjadi 19 ms pada resolusi 256 dan 20 ms pada resolusi 512 saat aktivitas tersibuk sesuai pada Tabel 8.

Selain menurunkan nilai komponen *shadow resolution*, solusi alternatif adalah mematikan komponen kualitas visual *soft shadow.* Mematikan *soft shadow* mampu memberikan latensi hingga 19 ms. Tetapi dengan mematikan *soft shadow* tampilan dari bayangan menjadi kurang realistis. Hasil dari mematikan *soft shadow* dapat dilihat pada Gambar 8 dimana bayangan objek menjadi sangat bergerigi.

Gambar 8. Visualisai komponen *soft shadow* Dimatikan

4. Kesimpulan

Grafik visual dan latensi merupakan dua aspek penting dalam aplikasi VR. Namun kedua aspek ini memiliki konsekuensi yang bertentangan. Kualitas grafik yang tinggi mengakibatkan latensi yang besar. Penelitian ini bertujuan mencari keseimbangan antara keduanya untuk mendapatkan latensi yang rendah dan tingkat grafis visual yang tinggi mencari tahu pengaruh komponen-komponen kualitas visual terhadap latensi dengan studi kasus aplikasi permainan berbasis VR. Karena anak ASD cenderung lebih peka, penelitian ini menetapkan nilai latensi yang ingin dicapai adalah 20 ms.

Tiga percobaan telah dilakukan dalam penelitian ini untuk mencari kombinasi optimum antara berbagai komponen visual untuk mendapatkan latensi dibawah 20 ms. Percobaan pertama mengukur latensi untuk setiap prasetel kualitas yang sudah tersedia dari Unity, percobaan kedua mencari tahu komponen kualitas visual yang paling berpengaruh terhadap latensi, dan percobaan ketiga mengukur performa dari aplikasi dengan cara mengukur performa CPU dan drive.

Dari hasil tiga percobaan tersebut didapatkan bahwa prasetel kualitas yang terbaik adalah *high quality* dengan menurunkan nilai untuk komponen *shadow resolution*. Komponen kualitas visual yang paling mempengaruhi latensi adalah *anti-aliasing* karena memberikan pekerjaan yang banyak kepada perangkat keras komputer. Kerja yang dilakukan adalah memuluskan garis-garis bergerigi agar seakan-akan terlihat seperti memiliki resolusi tinggi. *Anti-aliasing* yang tinggi mampu memberikan latensi hingga 260 ms atau tiga belas kali lipat dari latensi yang direkomendasikan.

Penelitian ini hanya melakukan percobaan terhadap pengaturan kualitas visual yang disediakan oleh *tools* unity untuk mendapatkan latensi yang rendah. Penelitian selanjutnya dapat melakukan percobaan terhadap *Depth Texture, Opaque Texture, Opaque Downsampling,Terrain Holes, Depth Bias, Normal Bias, Grading Mode. LUT sieze, SRP Batcher, Dynamic Lighting, Debug Level*, dan *Shader Variant Log Level* untuk mendapatkan latensi yang lebih rendah dengan tetap mempertahankan kualitas visual yang baik. Selain itu, penelitian selanjutnya sebaiknya juga melakukan observasi apakah latensi yang sudah dicapai benar-benar tidak mengakibatkan penyakit simulator.

Referensi

Beets, K., & Barron, D. L. (2000). *Super-sampling Anti-aliasing Analyzed*.

Bekele, E., Zheng, Z., Swanson, A., Crittendon, J., Warren, Z., & Sarkar, N. (2013). Understanding how adolescents with autism respond to facial expressions in virtual reality environments. *IEEE Transactions on Visualization and Computer Graphics*, *19*(4), 711–720. https://doi.org/10.1109/TVCG.2013.42

Bellani, M., Fornasari, L., Chittaro, L., & Brambilla, P. (2011). Virtual reality in autism: state of the art. *Epidemiology and Psychiatric Sciences*, *20*(3), 235–238. https://doi.org/10.1017/S2045796011000448

Brent Hale. (2019, January 30). *Anti-Aliasing 101: What is It & How Does it Impact Gamers?* https://techguided.com/what-is-anti-aliasing/

Capilla, R., & Martínez, M. (2004). Software Architectures for Designing Virtual Reality Applications. In F. Oquendo, B. C. Warboys, & R. Morrison (Eds.), *Software Architecture* (pp. 135–147). Springer Berlin Heidelberg.

Christian Nutt. (2013, August 20). *The biggest challenges for VR game developers, straight from Oculus*. https://www.gamedeveloper.com/programming/the-biggest-challenges-for-vr-game-developers-straight-from-oculus

Cindy Hatch-Rasmussen. (n.d.). *Sensory Integration in Autism - Autism Research Institute*. Retrieved December 29, 2021, from https://www.autism.org/sensory-integration/

Fischer, T., Böttcher, A., Coday, A., & Liebelt, H. (2010). Defining and Measuring Performance Characteristics of Current Video Games. In B. Müller-Clostermann, K. Echtle, & E. P. Rathgeb (Eds.), *Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance* (pp. 120–135). Springer Berlin Heidelberg.

George Musser. (2018, October 24). *How*

virtual reality is transforming autism studies | Spectrum | Autism Research News. https://www.spectrumnews.org/features/deep-dive/virtual-reality-transforming-autism-studies/

Karami, B., Koushki, R., Arabgol, F., Rahmani, M., & Vahabie, A. H. (2021). Effectiveness of Virtual/Augmented Reality–Based Therapeutic Interventions on Individuals With Autism Spectrum Disorder: A Comprehensive Meta-Analysis. *Frontiers in Psychiatry*, *12*, 887. https://doi.org/10.3389/FPSYT.2021.665326/BIBTEX

Ke, F., Moon, J., & Sokolikj, Z. (2020). Virtual Reality–Based Social Skills Training for Children With Autism Spectrum Disorder: *Https://Doi.Org/10.1177/01626434209456 03*. https://doi.org/10.1177/0162643420945603

Nathan Caruana, & Jon Brock. (2017, May 16). *Virtual reality yields clues to social difficulties in autism | Spectrum | Autism Research News*. https://www.spectrumnews.org/opinion/viewpoint/virtual-reality-yields-clues-social-difficulties-autism/

Oculus Rift S and Rift minimum requirements and system specifications. (n.d.). Retrieved January 18, 2022, from https://support.oculus.com/articles/getting-started/getting-started-with-rift/rift-s-minimum-requirements/

Raaen, K., & Kjellmo, I. (2015). Measuring Latency in Virtual Reality Systems. In K. Chorianopoulos, M. Divitini, J. Baalsrud Hauge, L. Jaccheri, & R. Malaka (Eds.), *Entertainment Computing - ICEC 2015* (pp. 457–462). Springer International Publishing.

Requirements — blender.org. (n.d.). Retrieved January 18, 2022, from https://www.blender.org/download/requirements/

Sahin, N. T., Keshav, N. U., Salisbury, J. P., & Vahabzadeh, A. (2018). Safety and Lack of Negative Effects of Wearable Augmented-Reality Social Communication Aid for Children and Adults with Autism. *Journal of Clinical Medicine 2018, Vol. 7, Page 188*, *7*(8), 188. https://doi.org/10.3390/JCM7080188

Unity-Manual: Post-processing. (2020). https://docs.unity3d.com/Manual/PostProcessingOverview.html

Unity-Manual: System requirements for Unity 2020 LTS. (n.d.). Retrieved January 18, 2022, from https://docs.unity3d.com/Manual/system-requirements.html#editor

VRoid Studio. (n.d.). Retrieved January 18, 2022, from https://vroid.com/en/studio