Rancang Bangun Game "MeuEndBude" Multiplayer Online 3D Menggunakan Algoritma *Line Trace (Raycast)*

Faiz Aiman Fajrirrahman[1], Atthariq[2*], Guntur Syahputra[3]

[1,2,3] *Jurusan Teknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe*
Jln. B.Aceh Medan Km.280 Buketrata 24301 INDONESIA

[1]faizaiman0106@gmail.com

[2]*atthariq.huzaifah@pnl.ac.id (penulis koresponden)

[3]guntur@pnl.ac.id

Abstrak— **Permainan video game telah menjadi salah satu hiburan populer di era modern ini. Game "Meu end Bude" adalah game *multiplayer online* 3D yang menawarkan pengalaman bermain yang menarik dan seru bagi para pemainnya. Untuk membangun perkembangan game online multiplayer yang lebih menarik dan lebih realistis pada khususnya game peperangan developer dituntut untuk dapat menciptakan game yang memiliki akurasi atau ketepatan disaat menembak. Contoh nya yaitu disaat memainkan game dalam fase tembak menembak yang di luncurkan tidak tepat mengenai sasaran atau arah tembakan yang dihasilkan berbeda, maka dengan itu dibutuhkan algoritma Line Trace (Raycast) Penelitian ini bertujuan untuk merancang dan membangun game "Meu end Bude" dengan menggunakan algoritma *Line Trace* atau yang lebih dikenal sebagai *RayCast* untuk mengoptimalkan deteksi dan interaksi objek dalam lingkungan permainan. Hasil penelitian menunjukkan bahwa algoritma Line Trace berhasil meningkatkan responsivitas interaksi pemain dengan objek dalam permainan. Ketepatan dan efisiensi deteksi benturan serta tindakan pemain menjadi lebih unggul, Fitur multiplayer online juga memungkinkan pemain untuk berinteraksi dan bersaing secara *real-time*, meningkatkan tingkat keterlibatan pemain dalam permainan. Penggunaan algoritma *Line Trace* dalam lingkungan 3D *multiplayer online* telah berhasil menciptakan lingkungan permainan yang menarik, realistis, dan memuaskan bagi pemain.**

Kata kunci— Game, *Multiplayer Online*, Algoritma *Line Trace (RayCast)*

Abstract— ***Video games have become one of the popular forms of entertainment in this modern era. The game 'Meu end Bude' is a 3D online multiplayer game that offers an engaging and exciting playing experience for its players. In order to enhance the development of more captivating and realistic online multiplayer games, especially warfare games, developers are required to create games that have accuracy and precision when shooting. For example, when playing a game during the shooting phase, if the launched shots do not accurately hit the target or the resulting shot directions differ, an algorithm called Line Trace (Raycast) is needed. This research aims to design and develop the game 'Meu end Bude' using the Line Trace algorithm, also known as RayCast, to optimize object detection and interaction within the game environment. The research results indicate that the Line Trace algorithm successfully enhances the responsiveness of player interactions with objects within the game. Accuracy and efficiency in collision detection as well as player actions are improved. The online multiplayer feature also enables players to interact and compete in real-time, thereby increasing the level of player engagement in the game. The utilization of the Line Trace algorithm in a 3D online multiplayer environment has successfully created an engaging, realistic, and satisfying gaming environment for players.***

Keywords— *Game, Multiplayer Online, Line Trace Algorithm (RayCast)*

I. Pendahuluan

Perkembangan video game online multiplayer dengan grafis yang semakin realisits telah menjadi salah satu revolusi terbesar dalam industri hiburan digital. Permainan *Video* (bahasa Inggris: *video game*) adalah permainan yang menggunakan interaksi dengan antarmuka pengguna melalui gambar yang dihasilkan oleh piranti *video.* Permainan *video* umumnya menyediakan sistem penghargaan misalnya skor yang dihitung berdasarkan tingkat keberhasilan yang dicapai dalam menyelesaikan tugas-tugas yang ada didalam permainan [1].

Untuk membangun perkembangan game online multiplayer yang lebih menarik dan lebih realistis pada khususnya game peperangan developer dituntut untuk dapat menciptakan game yang memiliki akurasi atau ketepatan disaat menembak. Contoh nya disaat memainkan game dalam fase tembak menembak yang di luncurkan tidak tepat mengenai sasaran atau arah tembakan yang dihasilkan berbeda, maka dengan itu dibutuhkan algoritma *Line Trace (Raycast).*

Ray-Casting Metode dimana gambar dari seluruh permukaan objek yang terlihat diperoleh dengan cara memancarkan garis sinar dari kamera/*viewer* menuju objek dan proses dilaksanakan dalam tiap *pixel* dari layar monitor [2].

Di masa lampau, banyak permainan bergantung pada teknik yang disebut raycasting untuk merender lingkungan 3D

menjadi gambar 2D (layar Anda). Raycasting juga memungkinkan mesin untuk menentukan objek pertama yang dilintasi oleh sinar. Para pengembang kemudian mulai bertanya, "Bagaimana jika sinar tersebut berasal dari moncong senjata untuk meniru tembakan peluru?" Dari gagasan ini lahirlah hitscan.

Pada kebanyakan implementasi senjata hitscan, ketika pemain melepaskan tembakan, mesin fisika akan:

a. Mencari tahu arah yang diindikasikan oleh senjata.
b. Memancarkan sinar dari moncong senjata sampai jarak tertentu.
c. Menerapkan raycasting untuk menentukan apakah sinar mengenai suatu objek.

Apabila ray menetapkan bahwa sebuah objek berada dalam lintasan tembakan, mesin akan memberikan pemberitahuan melalui pesan bahwa objek tersebut "terkena" oleh tembakan peluru. Target kemudian dapat melakukan segala perhitungan yang diperlukan untuk mencatat kerusakan.

Penelitian ini berkaitan dengan penelitian sebelumnya dengan judul "Penerapan *Dynamic Lighting* Pada 2D *Endless Runner Game* Menggunakan *Visibility Polygon Computation*". Penelitian sebelumnya menggunakan *Dynamic Lighting* yaitu sinar *Line Trace* untuk memberikan Cahaya pada suatu ruangan sedankan pada penelitian ini menggunakan *Line Trace* untuk menciptakan kerusakan *ray* terhadap benturan yang terjadi dan menghitung kerusakan yang dihasilkan oleh *ray* tersebut [3].

Penelitian ini berkaitan dengan penelitian sebelumnya dengan judul "Penerapan Algoritma *Naïve Bayes* Untuk Menentukan Perilaku NPC Pada Game *Action Multiplayer*". Penelitian sebelumnya menggunakan algoritma *Naïve Bayes* untuk menentukan perilaku NPC pada game *action multiplayer*, sedangkan penelitian menggunakan algoritma *Line Trace* untuk menentukan kerusakan tembakan dari hasil pemain disaat menembak [4].

Berdasarkan latar belakang di atas, dapat disimpulkan bahwa dengan menerapkan algoritma line Trace pada pembuatan game multiplayer online 3d dapat memperoleh tembekan dengan akurasi dan arah tembakan yang tepat mengenai target yang dituju serta memberikan kerusakan yang dihasilkan oleh tembakan tersebut. oleh karena itu, penelitian ini menggunakan algoritma line trace menjadi hal yang sangat penting dalam lingkup game peperangan sehingga dapat memberikan pengalaman bermain secara realistis dan realtime.

I. Metodologi Penelitian

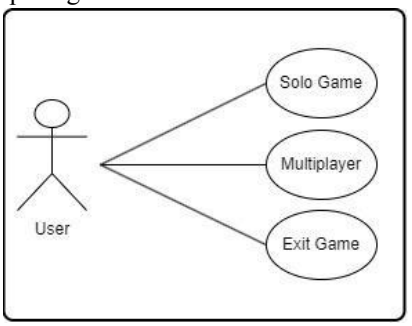
Penelitian ini menggunakan algoritma *Raycast* untuk menentukan apakah sinar yang ditembakkan dari senjata pemain mengenai target yang dimaksud, sehingga memungkinkan sistem penguncian target atau penembakan yang akurat dan Raycast digunakan untuk mendeteksi apakah sinar yang ditembakkan oleh musuh mengenai karakter pemain, yang memungkinkan sistem kerusakan atau mekanika gameplay lainnya untuk menghitung jumlah kerusakan yang diterima oleh karakter pemain dan multiplayer online untuk dapat merasakan suasana bermain melawan Permian yang ada diseluruh dunia.

A. Perancangan Use Case Diagram

Diagram Use Case merupakan salah satu jenis diagram yang umum digunakan dalam pemodelan perangkat lunak. Diagram ini digunakan untuk menggambarkan interaksi antara sistem perangkat lunak yang sedang dikembangkan dan berbagai aktor eksternal yang berinteraksi dengan sistem tersebut.

1. Use Case Diagram

Use Case Diagram merupakan gambaran dari user saat menjalankan system atau aplikasi. Adapun use case diagram dapat dilihat pada gambar 1.



Gambar 1. Use Case Diagram

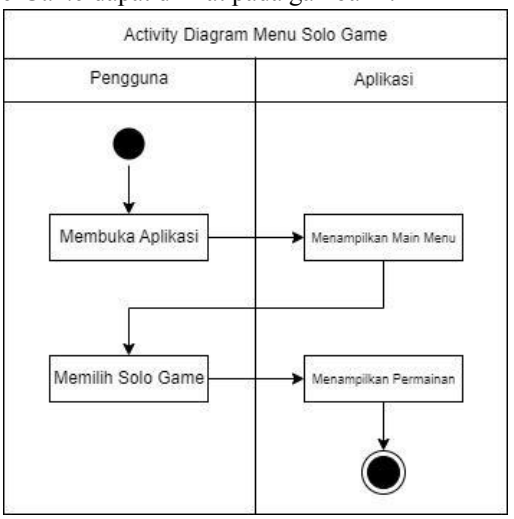
Dijelaskan dari gambar 3.1 *Use Case Diagram* adalah tampilan yang ada didalam menu awal aplikasi game.

2. Activity Diagram

Activity Diagram adalah gambaran suatu proses atau bagaimana jalan nya aplikasi pada system.

a) *Activity Diagram Solo Game*

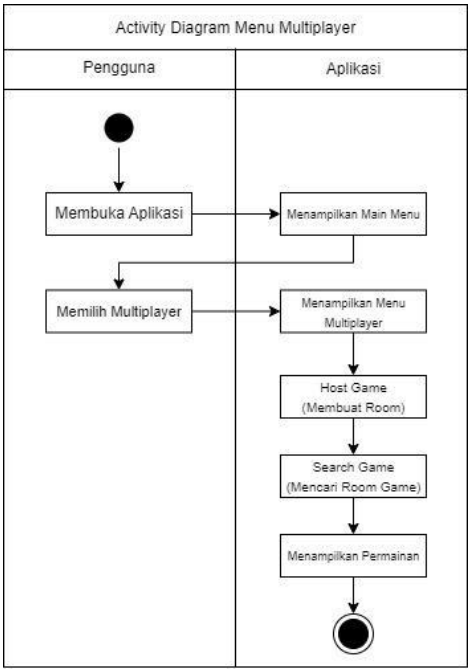
Diagram solo game user membuka aplikasi, kemudian akan tampil menu awal. Pada Menu menu wal aplikasi *user* menekan tombol "*Solo Game*" untuk memulai permainan. Adapun *Activity Diagram Solo Game* dapat dilihat pada gambar 2.



Gambar 2. *Activity Diagram Solo Game*

b) *Activity Diagram Multiplayer*

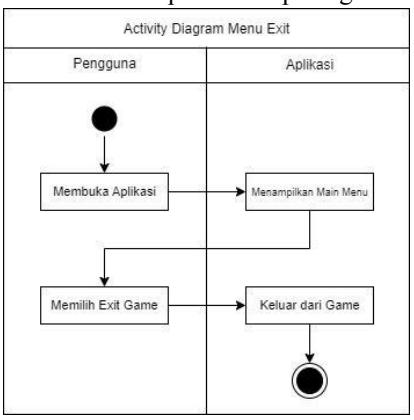
Diagram *multiplayer user* membuka aplikasi, kemudian akan tampil menu awal. Pada Menu menu awal aplikasi *user* menekan tombol "*Multiplayer*" maka akan menampilkan menu multiplayer. Adapun *Activity Diagram Multiplayer* dapat dilihat pada gambar 3.



Gambar 3. *Activity Diagram Multiplayer*

c) *Activity Diagram Exit Game*

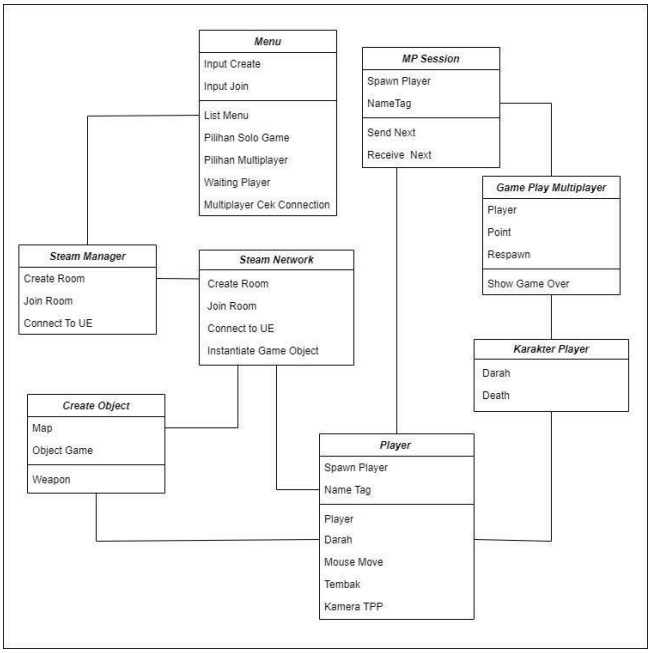
Diagram *exit game user* membuka aplikasi, kemudian akan tampil menu awal. Pada Menu menu awal aplikasi *user* menekan tombol "*Exit Game*" untuk keluar dari permainan. Adapun *Activity Diagram Exit Game* dapat dilihat pada gambar 4.



Gambar 4. *Activity Diagram Exit Game*

3. Class Diagram

Berikut perencanaan yang didalamnya terdapat inti dari pembuatan *game* 3D. Adapun *Class Diagram* dapat dilihat pada gambar 5.

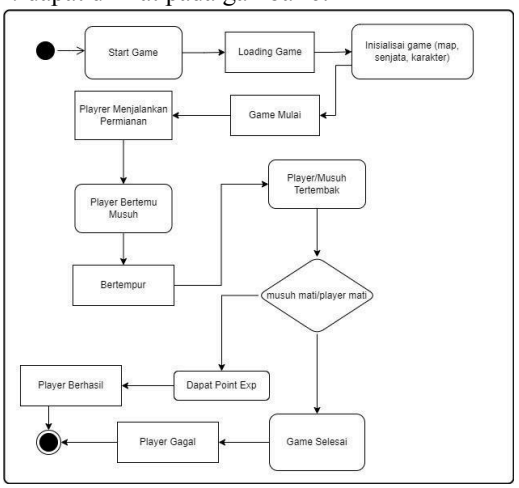


Gambar 5. *Class Diagram*

Dapat dilihat pada gambar 5. digambarkan alur kerja atau aktivitas sebuah sistem game yang dimulai start game untuk memulai permainan kemudian menunggu untuk memasuki game, setelah game muncul maka akan menginisiasi game berupa map, senjata item dan point. Untuk memulai permainan *player* menjalankan karakter dan membuka portal untuk bertemu musuh yang akan di tempur, jika player atau musuh tertembak maka musuh akan mati. *Player* yang berhasil membunuh musuh akan mendapatkan *point* dan berhasil menyelesaikan permainan, apabila *player* mati maka game akan selesai dan *player* tersebut gagal.

a. *Activity Class Diagram*

Berikut adalah activity class diagram yang dimana merupakan alur game MeuEndBude. Adapun *Activity Class Diagram* dapat dilihat pada gambar 6.



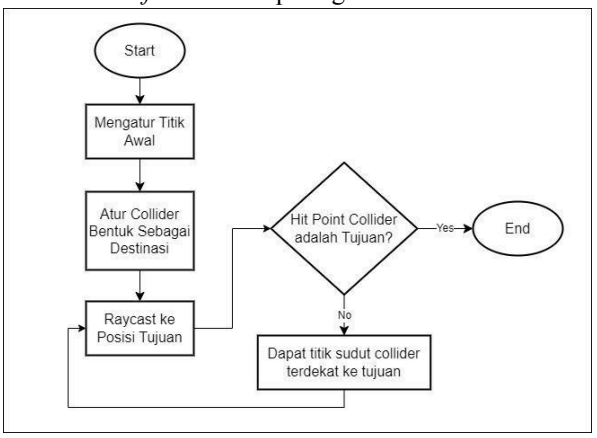
Gambar 6. *Activity Class Diagram*

Dapat dilihat pada gambar 6. digambarkan alur kerja atau aktivitas sebuah sistem game yang dimulai start game untuk memulai permainan kemudian menunggu untuk memasuki game, setelah game muncul maka akan menginisiasi game berupa map, senjata item dan point. Untuk memulai permainan *player* menjalankan karakter dan membuka portal untuk bertemu musuh yang akan di tempur,

jika player atau musuh tertembak maka musuh akan mati. *Player* yang berhasil membunuh musuh akan mendapatkan *point* dan berhasil menyelesaikan permainan, apabila *player* mati maka game akan selesai dan *player* tersebut gagal.

B. Flowchart Line Trace (Raycast)

Metode penelitian yang akan dilakukan dapat digambarkan ke dalam bentuk *flowchart* seperti gambar 7.



Gambar 7. Flowchart Algoritma *RayCast*

Dapat dilihat gambar 7. dimulai dengan mengatur titik awal untuk penempatan algoritma *raycast* tersebut, setelah mengatur titik awal algoritma mengatur *collider* bentuk yang berguna sebagai destinasi dari algoritma dan mengatur posisi tujuan *raycast* untuk *collider* bentuk sebagai destinas, *Raycast hit point* yaitu *Raycast* mengenai *point* yang sudah di tentukan, jika *Hit Point collider* adalah tujuan yang di tuju maka selesai jika tidak maka dapatkan atau temukan titik sudut *collider* terdekat ke tujuan kemudian bergerak ke arah *collider*.

III. Hasil dan Pembahasan

A. Implementasi Perangkat Lunak

1) Tampilan Menu Utama

Tampilan Menu Utama adalah tampilan awal disaat game dibuka, pada Menu Utama terdapat beberapa *button* yaitu "*Single Player*" untuk bermain sendiri, "*Multiplayer*" untuk bermain bersama, "*Profiles*" untuk memilih akun, dan "*Quit Game*" untuk keluar dari game. Tampilan menu utama dapat dilihat pada gambar 8.

Gambar 8. Tampilan Menu Utama

2) *LobbyMenu*

LobbyMenu merupakan letak dimana player berkumpul didalam satu *room* sebelum memulai permainan, pada tampilan "*LobbyMenu*" memiliki beberapa button yaitu "*Gamemode Option*" untuk mengatur durasi bermain dan jumlah score, "*Select Weapon*" untuk memilih senjata yang dipakai, dan "*Select Character*" untuk memilih karakter. Tampilan *LobbyMenu* dapat dilihat pada gambar 9.

Gambar 4.2 Tampilan *LobbyMenu*

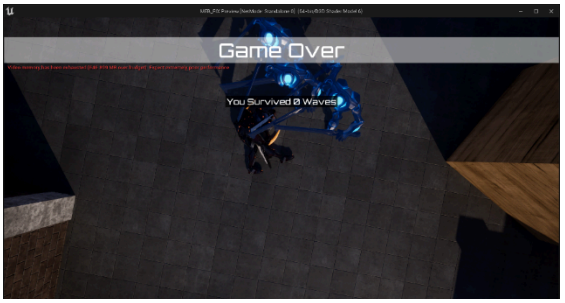
3) GamePlay

Tampilan gameplay berisi permainan yang dipilih dan merupakan bagian area gameplay, pada tampilan gameplay ini menampilkan *Head up Display* (HUD) seperti tampilan *Health*, Stamina, *Ammo* dan *Wave*. Tampilan *Gameplay* dapat dilihat pada gambar 10.

Gambar 10. Tampilan *GamePlay*

4) Tampilan *PlayerLose*

Tampilan *PlayerLose* ini akna muncul ketika player terbunuh oleh NPC yang mengerjarnya, setelah terbunuh game akan berakhir dan kembali kebagian *LobbyMenu*. Tampilan *Playerlose* Dapat dilihat pada gambar 11.



Gambar 11. Tampilan *PlayerLose*

B. Implementasi *Line Trace (Raycast)*

Algoritma *RayCasting* Metode dimana gambar dari seluruh permukaan objek yang terlihat diperoleh dengan cara memancarkan garis sinar dari kamera/*viewer* menuju objek dan proses dilaksanakan dalam tiap *pixel* dari layar monitor. Pada gambar 12. merupakan gambaran sebelum menggunakan algoritma.

Gambar 12. Sebelum Algortima

Dari hasil gambar 12. dapat dilihat karakter melancarkan tembakan kepada target yang ada didepannya. Tembakan yang diluncurkan tersebut tidak tau kemana arah yang tuju dikarenakan tidak ada bentuk peluru atau cahaya yang dapat dilihat oleh player. Maka dengan itu dibutuhkan nya algoritma *Line Trace* untuk mengetahui kemana arah peluru tersebut bergerak apakah mengenai target yang dituju atau tidak.

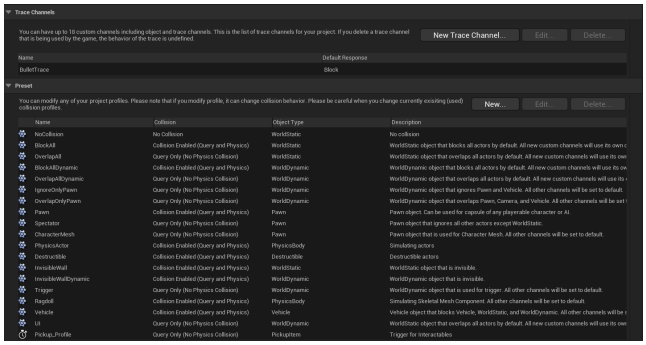
a. Algoritma *LineTrace*

Peluru yang diluncurkan terlihat oleh player dengan tanda yang berbentu cahaya leser berwarna merah, cahaya merah tersebut dihasilkan dari tembakan player yang keluar dari *muzzle* senjata, namun *ray* tersebut belum memberikan efek apapun. Untuk memberikan kerusakan akibat dari cahaya *ray* tersebut dibutuhkan nya *Bullet Trace* dan *Physical Asset.* Dapat dilihat pada gambar 13. merupakan gambaran algoritma *Linetrace* telah aktif.

Gambar 13. Algortima *Line Trace*

b. Variabel *BulletTrace*

Bullet Trace merupakan suatu variable untuk mendeteksi tabrakan antara ray dengan collider yang dituju, dengan adanya variable bullet trace ini cahaya ray yang dihasilkan dari tembakan pemain mampu mendeteksi kerusakan yang diakibatkan cahaya *ray* tersebut. Dapat dilihat pada gambar 14. merupakan tampilan *bulletrace.*



Gambar 14. *Bullet Trace*

Setelah *bullet trace* telah aktif maka *ray* tersebut akan kerusakan dan memberikan damage yang sudah diatur didalam program. Setelah memberikan *damage*, *damage* yang masuk atau mengenai *collider* akan dihitung kerusakan nya dan mengurangi darah player tersebut.

c. Variabel *Physical Asset*

Physical asset merupakan variabel yang berguna untuk memberikan informasi atau memberikan efek angka kerusakan yang dihasilkan oleh benturan cahaya *ray* sehingga pemain dapat mengetahui kerusakan yang diterima target atau *collider.* Dapat dilihat pada gambar 15. merupakan efek dari *Physical Asset.*

Gambar 15. *Physical Asset*

Untuk semakin menarik disaat menembak dipadukan asset-aset yang ditambahkan pada *physical asset* supaya hasil tembakan memberikan efek yang menarik saat pemain berada didalam fase peperangan.

d. Perhitungan Kerusakan

Kerusakan yang dihasilkan oleh *ray* tersebut berdasarkan dari senjata yang digunakan, masing-masing senjata memiliki tingkat kerusakan yang berbeda. Pada penelitian ini terdapat enam senjata dengan perhitungan, dapat dilihat pada tabel 1. berikut ini.

TABEL I
DAMAGE WEAPON

No.	*Weapon*	*Damage Ray*	Keterangan
1.	Pistol	1 *ray* = 15	8 *ray* (target mati)
2.	Auto Pistol	1 *ray* = 12	9 *ray* (target mati)
3.	Assault Rifle	1 *ray* = 25	4 *ray* (target mati)
4.	SMG	1 *ray* = 20	4 *ray* (target mati)
5.	LMG	1 *ray* = 18	6 *ray* (target mati)
6.	Shotgun	1 *ray* = 25	4 *ray* (target mati)

Dari tabel 1. Dapat disimpulkan masing-masing senjata memiliki tingkat kerusakan yang berbeda, Pistol memiliki kerusakan dengan 1 *ray* sebesar 15, maka dibutuhkan 8 benturan *ray* untuk mengalahkan musuh atau target. Auto Pistol memiliki kerusakan dengan 1 *ray* sebesar 12, maka dibutuhkan 9 benturan *ray* untuk mengalahkan musuh atau target. Assault Rifle memiliki kerusakan dengan 1 *ray* sebesar 25, maka dibutuhkan 4 benturan *ray* untuk mengalahkan musuh atau target. SMG memiliki kerusakan dengan 1 *ray* sebesar 20, maka dibutuhkan benturan *ray* untuk mengalahkan musuh atau target. LMG memiliki kerusakan dengan 1 *ray*

sebesar 18, maka dibutuhkan 6 benturan *ray* untuk mengalahkan musuh atau target. Shotgun memiliki kerusakan dengan 1 *ray* sebesar 25 maka dibutuhkan 4 benturan *ray* untuk mengalahkan musuh atau target

C. Hasil Pengujian Jaringan

Proses perancangan dan pembuatan skripsi Implementasi Rancang Bangun Game "MeuEndBude" Multiplayer Online 3D Menggunakan Algoritma Line Trace (Raycast), dalam tahap pengujiannya yaitu menggunakan *QOS* yang dimana merupakan sebuah metode yang digunakan untuk mengukur suatu kualitas jaringan dan hasil pengujian akan dimasukan ke dalam bentuk tabel. Pengujian tersebut akan dilakukan dengan urutan sebagai berikut.

1. Packet Loss

Packet loss merupakan banyaknya paket yang gagal mencapai tempat tujuan paket tersebut dikirim. Ketika packet loss besar maka dapat diketahui bahwa jaringan sedang sibuk atau terjadi overload. Berdasarkan teori dan hasil pengamatan disaat melakukan pengukuran. Pengukuran packet loss memiliki tiga pengujian yaitu bermain single player, multiplayer dengan jaringan yang sama dan multiplayer dengan jaringan yang berbeda. maka hasil yang didapatkan bisa dilihat pada tabel 2. berikut ini.

TABEL II
HASIL PENGUKURAN *PACKET LOSS*

No.	Hari/Tanggal	Pengujian	*Packet Loss*	Hasil
1.	Jumat 26 Agustus 2023	1	2.3%	Bagus
2.	Sabtu 27 Agustus 2023	2	0.2%	Sangat Bagus
3.	Sabtu 27 Agustus 2023	2	0.0%	Sangat Bagus
4.	Sabtu 27 Agustus 2023	3	0.0%	Sangat Bagus
5.	Sabtu 27 Agustus 2023	3	0.0%	Sangat Bagus

Dari hasil pengukuran packet loss rata-rata packet loss 0 maka masuk kedalam kategori "Sangat Bagus" hanya saja pada pengujian pertama yaitu bermain single player memperoleh nilai 2.3% dengan hasil "Bagus".

2. Delay

Delay merupakan lamanya waktu yang dibutuhkan oleh data atau informasi untuk sampai ke tempat tujuan data atau informasi tersebut dikirim. *Delay* pada suatu jaringan akan menentukan langkah apa yang akan diambil ketika memanajemen suatu jaringan. Berdasarkan teori dan hasil pengamatan disaat melakukan pengukuran, maka hasil yang didapatkan bisa dilihat pada 3. berikut ini.

Tabel III
HASIL PENGUKURAN *DELAY*

No.	Hari/Tanggal	Pengujian	*Delay*	Hasil
1.	Jumat 26 Agustus 2023	1	49	Sangat Bagus
2.	Sabtu 27 Agustus 2023	2	130	Sangat Bagus
3.	Sabtu 27 Agustus 2023	2	45	Sangat Bagus
4.	Sabtu 27 Agustus 2023	3	498	Sangat Buruk
5.	Sabtu 27 Agustus 2023	3	47	Sangat Bagus

Dari hasil pengukuran delay untuk masing-masing pemain adalah tertinggi terdapat di pengujian ke dua pada pemain pertama dengan delay 130 dan pengujian ke tiga pada permain pertama dengan delay 498. Pada pengujian ke dua pada pemain pertama tidak mempengaruhi permainan dikarenakan delay 130 pemain tidak merasakan respon bermain yang signifikan menurun sedangkan pada pengujian ketiga pada permain pertama sangat mempengaruhi gaya bermain dikarenakan delay 498 itu terlalu besar untuk bermain game online.

VI. Kesimpulan

Berdasarkan penelitian yang telah dilakuka oleh penulis, dapat diambil kesimpulan bahwa "Rancang Bangun Game "MeuendBude" Multiplayer Online 3D Menggunakan Algoritma *LineTrace (RayCast)*" berhasil dilakukan dengan kesimpulan. Teknologi yang digunakan oleh penulis adalah Unreal Engine 5.2 dengan pemanfaatan fitur *BluePrints*. Algoritma *Raycast (LineTrace)* berhasil mendeteksi interaksi antara sinar *(ray)* yang ditembakkan dari karakter pemain ke sekitarnya dengan akurat mengenai target, sinar yang ditembakkan dari senjata mengenai target yang dimaksud akan menghasilkan kerusakan atau mekanik *gameplay* untuk menghitung jumlah kerusakan yang diterima oleh karakter pemain lainnya. Hail Pengujian *packet Loss* rata-rata packet loss 0 maka masuk kedalam kategori "Sangat Bagus" hanya saja pada pengujian pertama yaitu bermain single player memperoleh nilai 2.3% dengan hasil "Bagus". Hasil pengujian *Delay* untuk masing-masing pemain adalah tertinggi terdapat di pengujian ke dua pada pemain pertama dengan delay 130 dan pengujian ke tiga pada permain pertama dengan delay 498. Pada pengujian ke dua pada pemain pertama tidak mempengaruhi permainan dikarenakan delay 130 pemain tidak merasakan respon bermain yang signifikan menurun sedangkan pada pengujian ketiga pada permain pertama sangat mempengaruhi gaya bermain dikarenakan delay 498 itu terlalu besar untuk bermain game online.

Referensi

[1] Irsa, R. W. 2015. "Perancangan Aplikasi Game Edukasi Pembelajaran Anak Usia Dini Menggunakan Linier Congruent Method (LCM)". Jurnal Informatika Global Vol 6 No.1.

[2] Supriyadi. 2018. "Media Pembelajaran Proses Rendering Objek 3D Berbasis Multimedia," Jurnal

Teknik Komputer, no. 2, hlm. 92–98, 2018, doi: 10.31294/jtk.v4i2.3545.

[3] Atika, R. P. 2021. "Perapan Algoritma Naïve Bayes Untuk Penentuan Perilaku NPC Pada Game *Action Multiplayer*". Lhokseunawe:Politeknik Negeri Lhokseuamwe

[4] Arda, S. F., Eriq, M. A. j., & Tri, A. 2019. "Penerapan Dynamic Lighting pada 2D Endless Runner Game menggunakan Visibility Polygon Computation".