

PENGEMBANGAN GAME FIRST PERSON SHOOTER BERBASIS WEB MENGUNAKAN UNITY DENGAN METODE GAME DEVELOPMENT LIFE CYCLE

Mirza Muhamad Alghivary, Aries Suharso, Sofi Defiyanti

Teknik Informatika, Universitas Singaperbangsa Karawang

Jalan HS. Ronggo Waluyo, Karawang, Indonesia

2010361170086@student.unsika.ac.id

ABSTRAK

Pengembangan game *First Person Shooter* (FPS) berbasis web menjadi tren dalam industri game karena memberikan akses lebih luas tanpa memerlukan instalasi software tambahan. Namun, pengembang game lokal di Indonesia menghadapi kendala signifikan, seperti keterbatasan modal, akses teknologi terbaru, serta pengalaman dalam manajemen proyek dan pemasaran, yang menghambat kemampuan bersaing secara global dan memenuhi standar internasional. Penelitian ini bertujuan mengembangkan game FPS berbasis web menggunakan *Unity*, *game engine multiplatform*, dengan menerapkan metode *Game Development Life Cycle* (GDLC) sebagai solusi atas tantangan tersebut. GDLC menyediakan kerangka kerja terstruktur yang mencakup tahapan *concept and planning*, *pre-production*, *production*, dan *testing*. Pada tahap *concept and planning*, dilakukan identifikasi kebutuhan pengguna dan penentuan fitur utama game, seperti mekanisme menembak, *respawn*, dan antarmuka pengguna. Tahap *pre-production* berfokus pada desain mekanika permainan, karakter, dan lingkungan, sedangkan tahap *production* melibatkan pengembangan fitur menggunakan *Microsoft Visual Studio*. Akhirnya, tahap *testing* dilakukan dengan metode *blackbox testing* untuk memastikan game bebas bug dan berfungsi optimal. Hasil penelitian menunjukkan bahwa penerapan GDLC dapat membantu pengembang mengatasi kendala teknis dan menghasilkan game berbasis web yang sesuai dengan kebutuhan pengguna.

Kata kunci : *First Person Shooter, Unity, Web, Game Development Life Cycle, Pengembangan Game, Microsoft Visual Studio.*

1. PENDAHULUAN

Pengembangan teknologi game telah berkembang pesat dalam beberapa dekade terakhir, dengan game berbasis web menjadi salah satu tren utama. Game berbasis web memungkinkan pengguna untuk mengakses hiburan dan pendidikan tanpa memerlukan instalasi perangkat lunak tambahan, memberikan kemudahan serta fleksibilitas yang tinggi. *Unity*, sebagai salah satu engine game terkemuka, mendukung pembuatan game lintas platform dengan berbagai fitur seperti rendering grafis berkualitas tinggi dan alat pengembangan berbasis skrip. Dalam konteks ini, metode *Game Development Life Cycle* (GDLC) menjadi sangat relevan, karena menyediakan pendekatan sistematis untuk merancang, mengembangkan, menguji, dan memelihara game berbasis web. [1].

Penggunaan *Game Development Life Cycle* (GDLC) oleh pengembang game *First-Person Shooter* (FPS) di Indonesia memiliki beberapa manfaat dan tantangan yang penting untuk diperhatikan. Di Indonesia, industri game terus berkembang, dan GDLC menawarkan pendekatan terstruktur yang dapat membantu pengembang mengelola proyek mereka secara lebih efisien dan efektif. Bagi pengembang game FPS di Indonesia, GDLC memberikan kerangka kerja yang jelas dengan tahapan yang terstruktur mulai dari konsep hingga pasca-peluncuran. Dalam lingkungan industri game yang terus berkembang, memiliki proses yang terorganisir sangat penting untuk memastikan bahwa setiap aspek pengembangan game diperhatikan. Ini membantu pengembang untuk

tetap fokus pada tujuan utama dan menghindari keterlambatan yang disebabkan oleh perencanaan yang kurang matang. Dengan mengikuti GDLC, pengembang game FPS di Indonesia dapat meningkatkan kapasitas mereka dalam menghadapi tantangan teknis dan bisnis. GDLC juga membantu mereka memenuhi standar industri global, yang penting untuk bersaing di pasar internasional. Penerapan GDLC memungkinkan pengembang Indonesia untuk menghasilkan produk yang berkualitas tinggi, yang pada akhirnya dapat membuka peluang untuk kolaborasi dan ekspansi di pasar global [2].

Pengembangan game berbasis web menggunakan *Unity* juga memiliki keunggulan tertentu, seperti aksesibilitas yang lebih besar bagi para pemain tanpa perlu mengunduh atau menginstal game secara terpisah. Hal ini sejalan dengan tren game online yang terus meningkat.

Unity adalah salah satu game engine yang paling populer digunakan dalam pengembangan game. Salah satunya developer game besar yaitu *Innersloth* pembuat game *Among Us* dan *Thatgamecompany* pembuat game *Sky: Children of the Light*. Faktor utama yang mendorong perkembangan penggunaan *Unity* [3] meliputi keterjangkauan dan kemudahan akses dalam *Unity* karena menyediakan versi gratis yang bisa digunakan oleh pengembang indie atau pemula, serta paket berbayar dengan fitur lebih lengkap untuk pengembang profesional. Ini membuat *Unity* menjadi pilihan utama bagi banyak pengembang game di seluruh dunia. Dapat dimainkan secara cross-

platform, Salah satu keunggulan utama Unity adalah kemampuannya untuk mengembangkan game lintas platform. Dengan Unity, game yang dikembangkan dapat dengan mudah di-porting ke berbagai platform seperti iOS, Android, Windows, MacOS, konsol, dan bahkan perangkat VR/AR. Unity juga memiliki komunitas dan dukungan yang kuat, Unity memiliki komunitas pengembang yang sangat aktif, dengan forum, dokumentasi, dan berbagai tutorial yang tersedia secara online. Selain itu, Unity Asset Store menyediakan berbagai aset (seperti model 3D, skrip, plugin) yang dapat mempercepat proses pengembangan game. penggunaannya dalam bidang lain selain game, Unity juga banyak digunakan dalam pengembangan aplikasi realitas virtual (VR), augmented reality (AR), simulasi, visualisasi arsitektur, dan berbagai aplikasi interaktif lainnya.

Namun di Indonesia, khususnya para developer game lokal memiliki kekurangan yang sama secara signifikan, kekurangan-kekurangan yang dihadapi oleh pengembang game di Indonesia yang bersumber dari [4] meliputi keterbatasan sumber daya dan modal diaman banyak developer game di Indonesia menghadapi kendala dalam hal modal dan akses ke teknologi terbaru. Hal ini membatasi kemampuan mereka untuk bersaing dengan developer internasional. Lalu kurangnya pengalaman dan kompetensi bagi kebanyakan pengembang yang masih kurang berpengalaman, terutama dalam hal manajemen proyek dan pemasaran. Dan juga pasar yang terbatas di beberapa tempat, pasar lokal yang relatif kecil dan belum sepenuhnya mendukung produk lokal menjadi tantangan lain.

Dalam skripsi yang berjudul “Pengembangan Game *First Person Shooter* Berbasis Web Menggunakan Unity dengan Metode GDLC (*Game Development Life Cycle*)”, penulis akan melakukan perkembangan terhadap pengalaman penulis dalam memainkan game yang dikembangkan. Dengan demikian, skripsi ini dapat memberikan kontribusi dalam pengembangan game berbasis web di Indonesia dan pengalaman bagi penulis, khususnya dalam genre *First Person Shooter*.

2. TINJAUAN PUSTAKA

2.1. Rancang Bangun

Rancang bangun adalah fase setelah analisis siklus pengembangan sistem, dimana persyaratan fungsional ditentukan dan bagaimana sistem akan terbentuk, yang dapat berupa menggambar, merencanakan dan membuat sketsa atau mengorganisasikan beberapa elemen terpisah menjadi satu. unit dan operasi terintegrasi termasuk konfigurasi komponen perangkat lunak sistem [5].

Rancang Bangun merupakan uraian, perencanaan dan garis besar atau penataan beberapa elemen yang terpisah menjadi satu kesatuan yang utuh dan fungsional. Pengertian desain dengan demikian adalah kegiatan mengubah hasil analisis ke dalam bentuk paket perangkat lunak dan kemudian membuat

sistem atau menyempurnakan sistem yang sudah ada [6]

2.2. Unity

Unity adalah sebuah *game engine* yang memungkinkan pengguna untuk membuat *game* 3D dan 2D dengan mudah dan cepat. Selain itu, *Unity* juga dapat digunakan untuk pengembangan aplikasi *Android* dan *Augmented Reality* (AR). [7].

2.3. API (Application Programming Interface)

Application Programming Interface (API) adalah seperangkat aturan dan protokol yang memungkinkan berbagai perangkat lunak untuk berkomunikasi satu sama lain. Dalam konteks pengembangan *game*, API sering digunakan untuk mengintegrasikan fungsionalitas eksternal, seperti sistem manajemen pengguna, sistem pembayaran, atau fungsionalitas jaringan. Pada penelitian ini API digunakan untuk berbagai tujuan, seperti mengelola data pengguna, menyediakan fitur jejaring sosial, atau mengintegrasikan mekanisme permainan multi pemain.

2.4. C#

Bahasa pemrograman C# adalah bahasa pemrograman yang dikembangkan oleh Microsoft dan digunakan untuk membuat aplikasi desktop, web, dan mobile. C# merupakan bahasa pemrograman yang berorientasi objek dan memiliki sintaks yang mirip dengan bahasa pemrograman Java. Beberapa fitur unggulan dari C# adalah *garbage collection*, delegasi, dan *event handling*.

2.5. Website

Website adalah kumpulan halaman *web* dari nama *domain* yang berisi informasi. Sebuah *website* biasanya terdiri dari beberapa halaman *web* yang saling terhubung. Tautan antara banyak halaman *web* dengan banyak halaman *web* lainnya, bagian dari halaman *web* yang sama, disebut *hyperlink*, sedangkan teks yang digunakan sebagai sarana penghubung disebut *hypertext*. *Website* dapat diakses melalui *web browser* yaitu software untuk mengakses halaman *web* seperti *Internet Explorer*, *Mozilla Firefox*, *Opera*, *Safari*, *Chrome*, dan lainnya. [8].

Website merupakan hasil penggabungan halaman-halaman yang dapat menampilkan berbagai informasi seperti teks, video, gambar, suara, dan animasi. Kumpulan ini bisa berwujud situs yang statis atau dinamis, dirancang sedemikian rupa sehingga halaman-halaman tersebut dapat terkoneksi satu sama lain. [9]

2.6. Game Development Life Cycle

Game Development Life Cycle (GDLC) adalah kerangka kerja yang digunakan untuk mengembangkan permainan video secara sistematis. GDLC mirip dengan *Software Development Life*

Cycle (SDLC), tetapi disesuaikan dengan kebutuhan khusus pengembangan permainan [10].

2.7. Penelitian Terdahulu

Dalam penyusunan penelitian ini, peneliti melakukan kajian terlebih dahulu dari jurnal atau peneliti terdahulu untuk dijadikan referensi

- a. Nurfi Nopriansyah dan Tumini (2021).
Judul: Penerapan Metode Game Development Life Cycle (GDLC) dan Algoritma Fisher Yates pada Game 2D "Mari Menjadi Pintar" di Unity.
Metode: Penelitian ini menerapkan metode GDLC yang terdiri dari enam tahapan: inisiasi, pra-produksi, produksi, pengujian, beta, dan rilis, serta algoritma Fisher-Yates untuk pengacakan elemen.
Hasil : Pengembangan game edukasi 2D ini berhasil membantu anak-anak memahami alfabet dan angka.
- b. Mustofa, Jordy Lasmana Putra, dan Chandra Kesuma (2022).
Judul: Penerapan Game Development Life Cycle untuk Video Game dengan Model Role Playing Game.
Metode: GDLC diterapkan dengan enam tahapan dari inisiasi hingga rilis.
Hasil: Game RPG berhasil dikembangkan dan berjalan sesuai tujuan, membuktikan efektivitas GDLC.
- c. Rido Ramadan dan Yani Widayani (2019).
Judul: Game Development Life Cycle Guidelines.
Metode: Mengusulkan model GDLC baru yang lebih terstruktur dengan fokus pada kualitas pada setiap fase.
Hasil: Pedoman GDLC ini membantu pengembang menciptakan game berkualitas tinggi.
- d. Rendy Adiwikarta dan Harya Bima Dirgantara (2020).
Judul: Pengembangan Permainan Video Endless Running Berbasis Android Menggunakan Framework Game Development Life Cycle.
Metode: GDLC diterapkan untuk membangun game endless running dengan enam tahapan utama.
Hasil: Game berhasil dikembangkan dan memberikan pengalaman bermain yang menarik.
- e. Renna Yanwastika Ariyana, Erma Susanti, Muhammad Rizqy Ath-Thaariq, dan Riki Apriadi (2022).
Judul: Penerapan Metode Game Development Life Cycle (GDLC) pada Pengembangan Game Motif Batik Khas Yogyakarta.
Metode: GDLC diterapkan dalam pembuatan game pengenalan motif batik khas Yogyakarta.
Hasil: Game edukasi ini berhasil membantu pelestarian budaya dengan pendekatan interaktif.
- f. Jayanto Ardi Laksono dan Cindy Taurusta (2020).
Judul: Design and Development Desktop-based 2D "ASKARA" Game.
Metode: Menggunakan GDLC dengan enam tahapan mulai dari inisiasi hingga rilis.

Hasil: Game edukasi berbasis desktop berhasil dikembangkan dengan respons positif dari pengguna.

- g. Nurfi Nopriansyah dan Tumini (2021).
Judul: Penerapan Metode Game Development Life Cycle (GDLC) dan Algoritma Fisher Yates pada Game 2D "Mari Menjadi Pintar" di Unity.
Metode: GDLC dan algoritma Fisher-Yates digunakan untuk pengembangan game edukasi.
Hasil: Game ini berhasil membantu anak-anak belajar dengan cara menyenangkan.
- h. Sauda (2020).
Judul: Rancang dan Bangun Game Edukasi Anak-Anak Berbasis Android dengan Unity Menggunakan Metode Game Development Life Cycle.
Metode: GDLC diterapkan dalam pengembangan game edukasi berbasis Android.
Hasil: Game edukasi ini memberikan media pembelajaran interaktif bagi anak-anak.
- i. Mustofa, Jordy Lasmana Putra, dan Chandra Kesuma (2022).
Judul: Penerapan Game Development Life Cycle untuk Video Game dengan Model Role Playing Game.
Metode: GDLC diterapkan dengan struktur enam tahap dalam pengembangan game RPG.
Hasil: Game RPG ini berhasil berjalan dengan baik dan sesuai harapan.
- j. Nurfi Nopriansyah dan Tumini (2021).
Judul: Penerapan Metode Game Development Life Cycle (GDLC) dan Algoritma Fisher Yates pada Game 2D "Mari Menjadi Pintar" di Unity.
Metode: GDLC dan algoritma Fisher-Yates digunakan untuk mengembangkan game edukasi.
Hasil: Game edukasi berhasil mendukung pembelajaran dasar anak-anak secara interaktif

3. METODE PENELITIAN

Menggunakan Game Development Life Cycle (GDLC) untuk pembuatan game First-Person Shooter (FPS) berbasis web menggunakan Unity pada penelitian ini memiliki berbagai alasan kuat yang memastikan proses pengembangan berjalan lancar dan menghasilkan produk berkualitas tinggi. GDLC menyediakan kerangka kerja yang terstruktur dengan tahapan yang jelas mulai dari konsep hingga pasca-peluncuran. Ini membantu penulis untuk memahami apa yang harus dilakukan pada setiap tahap, memastikan semua aspek pengembangan game diperhatikan dan dikelola dengan baik. Dalam konteks game FPS berbasis web, struktur yang jelas sangat penting untuk mengelola kompleksitas pengembangan gameplay, grafis, dan integrasi web.



Gambar 1. Game Development Life Cycle

Tahap-tahap GDLC yang digunakan pada penelitian ini meliputi:

a. *Concept and Planning*

Tahap ini mencakup ideasi dan dokumentasi konsep permainan, termasuk genre, karakter, dan mekanika. Fokus utama adalah genre FPS. Penulis juga menyusun rencana proyek yang mencakup waktu dan sumber daya yang dibutuhkan.

b. *Pre-Production*

Penulis membuat dokumen desain permainan berisi mekanika dan spesifikasi teknis. Tahap ini juga mencakup pembuatan prototipe awal untuk menguji konsep, pembentukan tim pengembang, serta pembagian peran dan tanggung jawab. Namun, dalam penelitian ini, seluruh proses pengembangan dilakukan secara mandiri.

c. *Production*

Tahap inti pengembangan menggunakan *Unity*, di mana elemen permainan seperti kode, grafik, suara, dan animasi dibuat dan diintegrasikan. Penulisan kode dilakukan dengan *Microsoft Visual Studio*.

d. *Testing*

Permainan diuji menggunakan metode *blackbox testing* untuk mengidentifikasi *bug* dan memastikan permainan sesuai tujuan awal pengembangan. Hasil pengujian digunakan untuk meningkatkan kualitas permainan.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Penelitian

Hasil dari penelitian ini adalah merancang dan mengembangkan game First Person Shooter berbasis web menggunakan *Unity* dengan metode Game Development Life Cycle (GDLC).

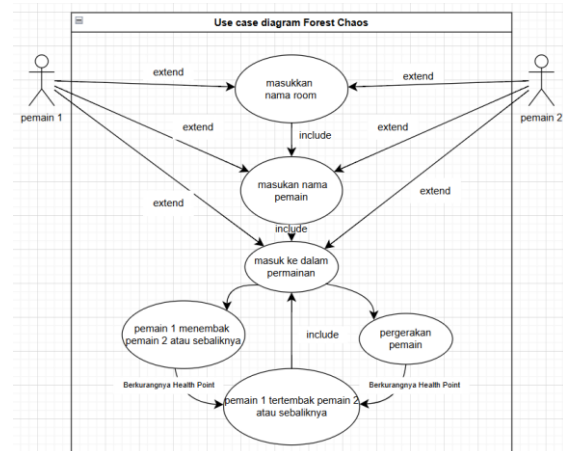
Aplikasi ini dikembangkan menggunakan bahasa pemrograman C# yang terdiri dari beberapa tahapan yaitu *concept and planning*, *pre-production*, *production*, dan *testing*.

4.1.1. Concept and Planning

Dalam proses *Concept and Planning* pada pengembangan game ini diberi nama *Forest Chaos*, game *Forest Chaos* ini diimplementasikan dan dikembangkan dengan memanfaatkan pemrograman yang memiliki spesifikasi sebagai berikut:

- Sistem Operasi yang digunakan adalah Windows 11 yang akan digunakan sebagai lingkungan pengembangan utama penulis untuk menulis dan menguji kode pada game.
- Microsoft Visual Studio dipilih sebagai editor kode untuk menulis, mengedit, dan mengelola kode aplikasi. Microsoft Visual Studio menyediakan berbagai fitur yang mendukung produktivitas pengembangan, seperti penyorotan sintaksis, debugger, dan integrasi dengan berbagai ekstensi.
- Unity digunakan sebagai *tools* utama untuk mengembangkan game *First Person Shooter* berbasis web dengan bahasa pemrograman C#, dan alat untuk merancang antarmuka pada game.

Adapun beberapa peran lingkungan yang ada di game pada penelitian ini menggunakan *Use case Diagram* seperti yang dapat dilihat pada gambar 2.



Gambar 2. Use Case Diagram

a. Skenario *Use Case* Masuk

Berikut adalah proses skenario *use case* pengguna untuk masuk ke dalam permainan, seperti yang dapat dilihat pada tabel 1

Tabel 1. Skenario *Use case* masuk

Identifikasi use case	
Nama	Masuk
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain belum memasuki permainan
Kondisi akhir	Pemain memasuki permainan
Aksi Aktor	Reaksi Sistem
1. Memasukkan nama pemain dan nama ruangan	2. Nama ruangan valid
	3. Memasuki permainan
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem
	1. Nama ruangan tidak valid
	2. Tidak dapat memasuki permainan

b. Skenario *Use case* menembak

Berikut adalah skenario *use case* pengguna untuk menembak di dalam permainan, seperti yang dapat dilihat pada tabel 2

Tabel 2. Skenario *Use case* menembak

Identifikasi use case	
Nama	In-game screen
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain 1 mengarahkan senjata kearah pemain 2 atau sebaliknya
Kondisi akhir	Pemain 1 menembak pemain 2 atau sebaliknya
Aksi Aktor	Reaksi Sistem

1. Pemain 1 menekan mouse kiri saat mengarah ke arah pemain 2 atau sebaliknya	
	2. Pemain berhasil menembak mengenai pemain lain
	3. Pemain lain tumbang
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem
1. Pemain menekan mouse kiri saat mengarah ke arah pemain lain	
	2. Pemain tidak berhasil menembak mengenai pemain lain
	3. Pemain lain tidak tumbang

c. Skenario *Use case* bergerak

Berikut adalah skenario *use case* pengguna untuk bergerak di dalam permainan, seperti yang dapat dilihat pada tabel 3

Tabel 3 Skenario *Use case* bergerak

Identifikasi use case	
Nama	<i>In-game screen</i>
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain diam di tempat
Kondisi akhir	Pemain bergerak ke berbagai arah
Aksi Aktor	Reaksi Sistem
1. Pemain menekan keyboard W, A, S, D, dan Spasi	
	2. Pemain berhasil bergerak ke arah depan, kiri, belakang, kanan, dan melompat

d. Skenario *Use case* *Respawn* (hidup kembali di titik yang ditentukan)

Berikut adalah skenario *use case* pengguna saat *Respawn* di dalam permainan, seperti yang dapat dilihat pada tabel 4

Tabel 4. Skenario *Use case* *Respawn*

Identifikasi use case	
Nama	<i>In-game screen</i>
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain <i>spawn</i> pada <i>spawn point</i> yang sudah di tentukan
Kondisi akhir	Pemain mati dan <i>respawn</i> di <i>spawn point</i>
Aksi Aktor	Reaksi Sistem
1. Pemain 1 tertembak oleh pemain 2 atau sebaliknya	
	2. Pemain mati dan <i>respawn</i> di <i>spawn point</i>
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem

1. Pemain 1 tertembak oleh pemain 2	
	2. Pemain tidak mati dan tidak <i>respawn</i>

e. Skenario *Use case* *Health Point*

Berikut adalah skenario *use case* *Health Point* pengguna di dalam permainan, seperti yang dapat dilihat pada tabel 5

Tabel 5. Skenario *Use case* *Health Point*

Identifikasi use case	
Nama	<i>In-game screen</i>
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain memiliki <i>health point</i> penuh
Kondisi akhir	Pemain tidak memiliki <i>health point</i> penuh
Aksi Aktor	Reaksi Sistem
1. Pemain 1 tertembak oleh pemain 2 atau sebaliknya	
	2. Poin darah milik pemain 2 berkurang atau sebaliknya
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem
1. Pemain 1 tidak berhasil menembak pemain 2 atau sebaliknya	
	2. Pemain 2 masih memiliki poin darah atau sebaliknya

4.1.2. Pre-Production

Tahap *Pre-Production* ini dimana pengembang mengembangkan desain permainan yang mendetail, yaitu mekanika permainan, karakter, dan desain lingkungan yang akan menjadi acuan utama selama proses pengembangan menggunakan Unity.

a. Mekanika Permainan

Mekanika permainan dalam game *First-Person Shooter* (FPS) ini menggunakan aturan dan sistem yang mengatur bagaimana pemain menggerakkan karakter.

b. Karakter

Pada penelitian ini, karakter dibuat menjadi 2 yakni polisi dan robot. Pemain mendapat salah satu karakter secara acak. Masing-masing karakter tidak memiliki perbedaan kelebihan apapun.

c. Lingkungan

Desain lingkungan pada penelitian ini juga menjadi sorotan yang penting, juga menjadi obstacle (rintangan) bagi pemain seperti gedung tinggi, container, dan pintu otomatis. Beberapa dari *obstacle* di ambil dari *assetstore* yang di sediakan oleh Unity.

4.1.3. Production

Pada tahap *Production* penulis akan menggunakan software Unity yang dimana tahapan ini

adalah inti dari pengembangan permainan, di mana semua elemen dibuat dan diintegrasikan. Proses ini melibatkan penulisan kode menggunakan Microsoft Visual Studio untuk pembuatan aset menggunakan algoritma fuzzy seperti batas jarak menembak, pergerakan pandangan yang dilihat karakter, dan pergerakan karakter.

Berikut fungsi dari kode program yang dibuat pada penelitian :

a. Batas jarak menembak

Kode program batasan pada jarak menembak dibuat bertujuan untuk keseimbangan permainan, hingga pemain mengetahui jarak tembakan yang dibutuhkan agar dapat menembak pemain lainnya dengan tepat sasaran.

Berikut adalah kode program untuk batas jarak menembak:

```
[SerializeField]
```

```
private float weaponRange = 100.0f;
```

b. Pergerakan pandangan

Kode program pergerakan pandangan dibuat supaya pemain dapat melihat area atau pemain lain di sekitar karakter.

Berikut adalah kode program untuk menggerakkan pandangan pada karakter:

```
111 void Look()
112 {
113     transform.Rotate(Vector3.up * Input.GetAxisRaw("Mouse X") * mouseSensitivity);
114
115     verticalLookRotation += Input.GetAxisRaw("Mouse Y") * mouseSensitivity;
116     verticalLookRotation = Mathf.Clamp(verticalLookRotation, -90f, 90f);
117
118     cameraHolder.transform.localEulerAngles = Vector3.left * verticalLookRotation;
119 }
```

Gambar 3. Kode program pergerakan pandangan karakter

c. Pergerakan karakter

Pergerakan karakter adalah hal terpenting dalam sebuah permainan *First Person Shooter* (FPS), dimana pemain dapat menggerakkan karakter kedepan, belakang, kanan, kiri, dan lompat.

Berikut adalah kode program untuk pergerakan karakter:

```
121 void Move()
122 {
123     Vector3 moveDir = new Vector3(Input.GetAxisRaw("Horizontal"), 0, Input.GetAxisRaw("Vertical")).normalized;
124
125     moveAmount = Vector3.SmoothDamp(moveAmount, moveDir * (Input.GetKey(KeyCode.LeftShift) ? sprintSpeed : walkSpeed),
126     ref smoothMoveVelocity, smoothTime);
127
128
129     void Jump()
130     {
131         if (Input.GetKeyDown(KeyCode.Space) && grounded)
132         {
133             rb.AddForce(transform.up * jumpForce);
134         }
135     }
136 }
```

Gambar 4. Kode program pergerakan pada karakter

4.1.4. Testing

Pada tahap *Testing*, perkembangan permainan memasuki fase blackbox testing, di mana permainan akan diuji dengan tujuan untuk memverifikasi bahwa fitur dapat digunakan sesuai dengan harapan, serta untuk mengidentifikasi dan memperbaiki potensi masalah yang mungkin timbul.

Proses pengujian ditujukan pada pengujian fungsional fitur yang dibuat dengan fokus pada pengujian pada perangkat nyata yaitu pada ASUS TUF Gaming F15 dengan spesifikasi processor intel i7 generasi 11, ram 16gb, dan versi DirectX 12. Akan tetapi karena game ini adalah berbasis web maka tidak

ada minimum spesifikasi. Berikut adalah skenario-skenario pengujian yang telah dilakukan pada *game*:

a. Masuk ke dalam *game*

Pengujian fitur saat masuk ke dalam *game* dilakukan untuk memastikan sistem berfungsi sesuai dengan skenario yang dirancang. Dalam skenario pertama, pemain mengisi nama pemain yaitu *Elzaine* dan nama *room DefaultRoom* yang belum tersedia. Hasil yang diharapkan adalah sistem membuat *room* baru dengan nama *DefaultRoom* dan pemain berhasil masuk ke dalam *game* di *room* tersebut. Hasil pengujian menunjukkan sistem berfungsi sesuai harapan, sehingga dinyatakan valid. Pada skenario kedua, pemain mengisi nama pemain yaitu *Araqt* dan nama *room DefaultRoom* yang sudah tersedia. Sistem diharapkan memasukkan pemain ke dalam *game* di *room* yang telah ada, dan hasil pengujian membuktikan bahwa sistem bekerja sesuai harapan, sehingga juga dinyatakan valid. Skenario terakhir menguji kondisi ketika pemain tidak mengisi nama pemain dan nama *room*. Dalam kasus ini, sistem diharapkan tidak mengizinkan pemain masuk ke dalam *game* dan tidak melakukan aksi lebih lanjut. Hasil pengujian menunjukkan sistem bekerja seperti yang diharapkan, sehingga dinyatakan valid. Keseluruhan pengujian menunjukkan bahwa fitur "Masuk ke dalam *game*" telah berfungsi sesuai spesifikasi.

b. Menembak dengan jarak yang tepat dan mengurangi *health point* karakter pemain lain.

Pengujian dilakukan untuk memverifikasi mekanisme menembak ke arah karakter pemain lain serta pengurangan *health point* sesuai fungsinya. Pada skenario pertama, pemain menembak ke arah karakter pemain lain dengan jarak yang berada dalam batas yang ditentukan. Sistem berhasil mengenali aksi pemain menggunakan tombol mouse kiri untuk menembak, dan tembakan tersebut mengenai karakter pemain lain sehingga *health point*-nya berkurang. Hasil pengujian dinyatakan valid dan sesuai dengan harapan. Pada skenario kedua, pemain menembak ke arah karakter pemain lain tetapi dengan jarak yang berada di luar batas tembakan. Sistem tetap memproses aksi tombol mouse kiri untuk menembak, tetapi tembakan tidak mengenai karakter pemain lain, sehingga *health point*-nya tetap penuh. Hasil pengujian ini juga valid dan sesuai dengan harapan. Pengujian ini menunjukkan bahwa mekanisme menembak dan efeknya terhadap *health point* telah berfungsi sebagaimana mestinya.

c. Menggerakkan karakter di dalam *game*.

Pengujian dilakukan untuk memastikan fungsi kontrol karakter berjalan sesuai dengan input dari pemain. Pada skenario pertama, pemain menekan tombol W, A, S, D, dan spasi satu per satu untuk menggerakkan karakter maju, mundur, ke kanan, ke kiri, serta melompat. Hasil pengujian

menunjukkan bahwa karakter dapat bergerak sesuai keinginan pemain, sehingga dinyatakan valid dan sesuai dengan harapan. Pada skenario kedua, pemain tidak menekan tombol *W*, *A*, *S*, *D*, maupun *spasi*. Hasilnya, karakter tidak bergerak sama sekali, sesuai dengan keinginan pemain. Pengujian ini berhasil menunjukkan bahwa mekanisme kontrol pergerakan karakter berfungsi dengan baik sesuai skenario yang dirancang.

- d. Karakter *respawn* (hidup kembali ke titik yang sudah ditentukan) saat kehabisan *health point*. Pengujian dilakukan untuk memverifikasi mekanisme *respawn* pada karakter pemain. Pada skenario pertama, pemain kehabisan *health point* setelah tertembak oleh pemain lawan. Hasil pengujian menunjukkan bahwa karakter pemain berhasil *respawn* ke titik yang telah ditentukan, sehingga dinyatakan valid dan sesuai dengan harapan. Pada skenario kedua, pemain masih memiliki sisa *health point* meskipun telah tertembak oleh pemain lawan. Dalam kondisi ini, karakter tidak melakukan *respawn*, sesuai dengan fungsi yang diharapkan. Pengujian ini berhasil memastikan bahwa mekanisme *respawn* berfungsi dengan baik berdasarkan kondisi yang telah ditetapkan.

Setelah berhasil menyelesaikan tahap *testing*, berikut adalah gambar hasil dari fitur yang sudah diuji dan sudah dinyatakan valid dengan blackbox testing :

a. Menembak

Berikut adalah gambar hasil dari pengujian menembak saat sebelum dan sesudah menembak:



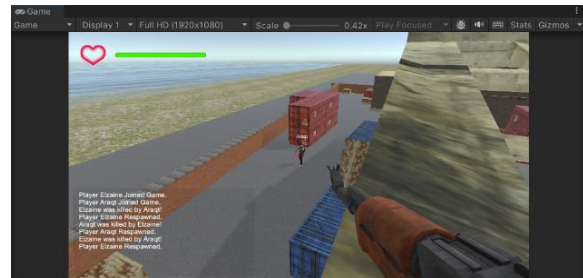
Gambar 5. Sebelum terkena tembakan



Gambar 6. Setelah terkena tembakan

b. Pergerakan Karakter

Berikut adalah gambar hasil pengujian pergerakan karakter saat karakter berpindah tempat:



Gambar 7. Posisi awal

Perubahan posisi dapat dilihat pada gambar 2.

c. *Respawn*

Berikut adalah gambar hasil pengujian *respawn* terhadap karakter saat karakter kehabisan *health point*:



Gambar 8. Ketika kehabisan *health point*



Gambar 9. Posisi saat *respawn*

d. *Health point*

Hasil dari pengujian *health point* dapat dilihat pada gambar 4 yang dimana hal itu terjadi saat pengujian menembak dilakukan.

4.2. Pembahasan

Aplikasi yang berhasil dikembangkan merupakan sebuah game yang dirancang menggunakan bahasa pemrograman C# pada Microsoft Visual Studio. Berdasarkan tahapan pengembangan yang mencakup *concept and planning*, *pre-production*, *production*, *testing*, menunjukkan bahwa aplikasi ini dapat memenuhi kebutuhan pengembang. Fitur yang tersedia di dalam permainan seperti bergerak, menembak, *health point*, dan *respawn* yang telah di uji sudah di nyatakan valid dan sesuai dengan harapan untuk sebuah game yang berada di tahap pengembangan

5. KESIMPULAN DAN SARAN

Setelah melalui serangkaian tahapan pengembangan pada game FPS (*First Person Shooter*) berjudul *Forest Chaos* yang dibuat menggunakan unity berbasis web berhasil diselesaikan dan diuji. Kesimpulan yang dapat di kutip dari penelitian ini adalah Pengembangan game FPS berbasis web yang berjudul *Forest Chaos* berhasil dikembangkan dengan menggunakan Unity 3D, kode program yang dibuat dengan Microsoft Visual Code, hingga fitur yang terdapat pada permainan sudah sesuai dengan tahapan yang digunakan pada metode *Game Development Life Cycle* (GDLC). Lalu *game Forest Chaos* berhasil dibuat dengan mengintegrasikan grafis dan audio, dari segi grafis terdapat desain lingkungan dan karakter, dari segi audio terdapat suara tembakan, suara saat karakter tertembak, dan suara saat karakter kehabisan *health point*, karena game tersebut dapat dimainkan hanya dengan sekali jalan (*One-shot Game*) yang dimana pemain dapat bermain tanpa memakan banyak waktu untuk mengulanginya dari awal. Setelah merancang dan membuat game *Forest Chaos*, adapun saran pada saat membaca referensi literatur dari penelitian Rodo Maxmilano tentang pergerakan *Non-Player Character* (NPC) dalam game 2D berbasis web, dalam literatur tersebut ada hal yang menarik penulis yaitu penggunaan kecerdasan buatan untuk merancang dan mencari efektivitas pada NPC, dengan ada nya fitur tersebut pengguna dapat berinteraksi dengan NPC menjadi lebih baik dan lebih ramah pada laptop atau komputer dengan spesifikasi rendah, dan juga menghasilkan nilai yang tinggi pada saat pengujian. Dengan merancang dan mengembangkan fitur *database* pada *game* yang dimana hal ini akan memudahkan pemain untuk memasukan akun dan menyimpan *history* bermain serta poin-poin lainnya yang sudah didapatkan pemain, juga mengintegrasikan pemain dengan berbagai platform sehingga game *Forest Chaos* ini dapat dimainkan dengan berbagai sistem operasi.

DAFTAR PUSTAKA

- [1] Fauzan, S. A., 2022. Implementasi Game Development Life Cycle Model pengembangan Arnold Hendrick's dalam pembuatan game Puzzle-RPG Enigma's Dungeon. *Jurnal Ilmu Komputer Dan Informatika*, 2(2), 113–126. <https://doi.org/10.54082/jiki.26>.
- [2] Rusmana, R. A., Asriyanik, A., Setiawan, I. R. 2023. Penggunaan Metode Game Development Life Cycle (GDLC) Untuk Memudahkan Belajar Bahasa Inggris Dalam Media Game. *Jurnal of Information System Research (JOSH)*. 4(4): 1402-1412.
- [3] Xie, J. 2020. Research On Key Technologies Base Unity 3D Game Engine. *International Conference Computer Science and Education*. DOI:10.1109/ICCSE.2020.6295169
- [4] Yuwono, A. I. 2021. Eksistensi Developer Game Independen Indonesia (Studi Kasus Eksistensi Developer Game Independen Agate Studio, Creacle Studio, dan Digital Happiness Dalam Perspektif Ekonomi Politik Komunikasi). *Jurnal Media dan Komunikasi Indonesia*. 2(1).
- [5] Y. Mulyanto, F. Hamdani, and N. Hasmawati, "Rancang Bangun Sistem Informasi Penjualan Pada Toko Omg Berbasis Web Di Kecamatan Empang Kabupaten Sumbawa" *Jurnal Informatika Teknologi Dan Sains (Jinteks)*, vol. 2, no. 1, pp. 69–77, Feb. 2020, doi: 10.51401/jinteks.v2i1.560.
- [6] R. Gunawan, A. M. Yusuf, dan L. Nopitasari, "Rancang Bangun Sistem Presensi Mahasiswa Dengan Menggunakan QR Code Berbasis Android," dalam *Jurnal Teknologi Informasi dan Komputer*, vol. 14, no. 1, hlm. 47–58, 2021.
- [7] Prayoga, D., Rusdiana, L., & Yedithia, F. (2022). Pengembangan Game 2d Platformer "Virus Must Die" Berbasis Android Menggunakan Unity. *Jurnal Saintekom (Sain, Teknologi, Komputer Dan Manajemen)*, 12(2), 200–209.
- [8] R. Nasser dan S. Saputra, "Rancang Bangun Website Pada Unit Pelaksana Teknis Sekolah Menengah Pertama (SMP) Negeri 5 Baebunta," *Jurnal Ilmiah d'Computare*, pp. 44–53, 2019
- [9] A. P. Anjelita dan E. Rosiska, "Rancang Bangun Sistem Informasi E-Learning Pada SMK Negeri 3 Batam," *Comasie Journal*, 2019.
- [10] S. Asmiatun dan A. N. Putri, *Belajar Membuat Game 2D dan 3D Menggunakan Unity*, ed. pertama, Yogyakarta: Deepublish, 2019