

Perancangan Game Puzzle Labirin Menggunakan Metode *Game Development Life Cycle* (GDLC) Berbasis *Unreal Engine*

Suwahyudi Syarif^{a,1,*}, Tasrif Hasanuddin^{a,2}, Mardiyah Hasnawi^{a,3}

^a Universitas Muslim Indonesia, Jalan Urip Sumoharjo, Makassar, 90231

¹ suwahyudisyarif1998@gmail.com; ² tasrif.hasanuddin@umi.ac.id; ³ mardiyah.hasnawi@umi.ac.id;

*corresponding author

INFORMASI ARTIKEL	ABSTRAK
Diterima : 07 – 02 – 2022 Direvisi : 21 – 02 – 2022 Diterbitkan : 28 – 02 – 2022	Perancangan sebuah game memerlukan alur perancangan layaknya perancangan sebuah sistem atau yang sering disebut dengan <i>System Development Life Cycle</i> (SDLC). Maka dari itu perancangan game membutuhkan panduan yang lebih spesifik yang khusus dalam merancang game. Maka muncullah istilah <i>Game Development Life Cycle</i> (GDLC). Perancangan sebuah game juga membutuhkan sebuah engine untuk mendukung pengembangan game. <i>Unreal engine</i> merupakan game engine yang di kembangkan oleh <i>Epic Games</i> yang memiliki banyak fitur salah satunya ialah adanya template awal yang akan mempermudah pengembang untuk membuat game. Adapun keluaran dari penelitian ini ialah sebuah aplikasi game yang dibuat dengan metode <i>Game Development Life Cycle</i> (GDLC) dan menggunakan engine <i>Unreal</i> .
Kata Kunci: Metode <i>Game Development Life Cycle</i> <i>Unreal Engine</i> Perancangan Game Puzzle	
	This is an open access article under the CC-BY-SA license



I. Pendahuluan

Teknologi tidak pernah berhenti untuk berkembang dengan pesat dan membuat terobosan-terobosan baru dalam segi kehidupan. Hal ini ditandai dengan adanya pengembangan teknologi disegala bidang. Bukan hanya teknologi untuk membantu dan mempermudah pekerjaan manusia saja, namun teknologi untuk hiburan juga berkembang. Salah satu teknologi di bidang hiburan ialah video game. Perkembangan video game sangatlah pesat, dibuktikan dengan pendapatan video game global yang terus meningkat sampai 2021 [1]. Peningkatan ini terjadi di semua jenis video game baik yang berbasis dekstop, *mobile* maupun *console*. Perancangan sebuah game para *developer* selaku pengembang memerlukan alur perancangan layaknya perancangan sebuah sistem atau yang sering di sebut dengan *System Development Life Cycle* (SDLC) [1], [2]. Untuk perancangan game yang sebelumnya menggunakan SDLC sekarang mulai mengalami beberapa perubahan. Maka dari itu perancangan game membutuhkan panduan yang lebih spesifik yang khusus dalam merancang game. Maka muncullah istilah *Game Development Life Cycle* (GDLC).

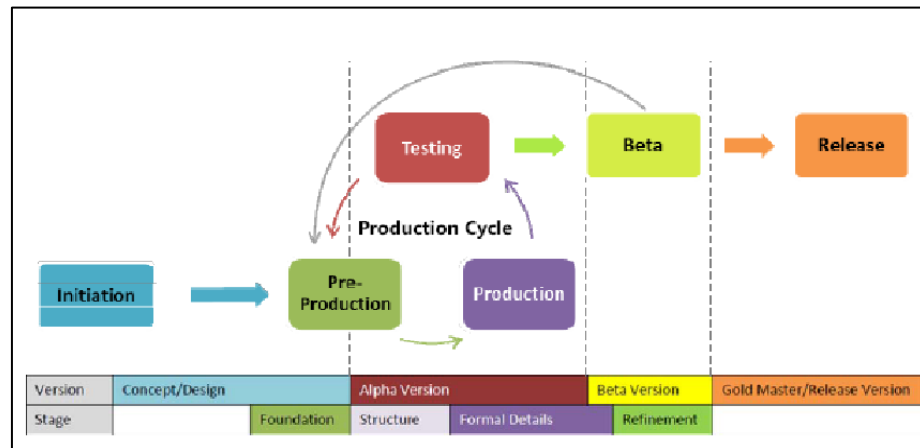
Perancangan sebuah game juga membutuhkan sebuah engine untuk mendukung pengembangan game. *Unreal engine* merupakan game engine yang dikembangkan oleh *Epic Games* yang memiliki banyak fitur salah satunya ialah adanya template awal yang akan mempermudah pengembang untuk membuat game. Game puzzle terdiri dari beberapa jenis, salah satu di antaranya ialah maze atau labirin. Maze merupakan game puzzle di mana jalan dari permainannya ialah dengan mencari jalan yang benar dari banyaknya jalur yang ada sampai ke lokasi tempat yang di tentukan. Oleh karena itu, maka akan di buat sebuah perancangan game bergenre puzzle maze menggunakan engine *Unreal* dengan metode *Game Development Life Cycle* (GDLC).

II. Metode

A. *Game Development Life Cycle* (GDLC)

GDLC adalah suatu proses pengembangan sebuah game yang menerapkan pendekatan iteratif yang terdiri dari 6 fase pengembangan, di mulai dari fase inialisasi/pembuatan konsep, *preproduction*, *production*, *testing*, *beta* dan *realease* [3]. Dari 6 fase tersebut dapat di kelompokkan menjadi 3 proses utama yaitu:

- 1) Instalasi *The Dude* Proses Inialisasi yang terdiri dari konsep dan *design*.
- 2) Proses produksi terdiri dari pra-produksi, produksi dan pengujian.
- 3) *Release*

Gambar 1. *Ease Dan Proses GDLC*

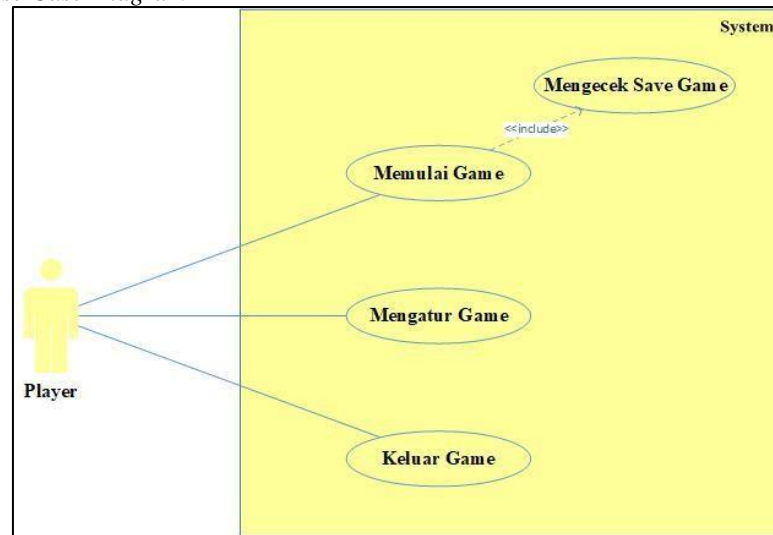
- 1) *Initiation* adalah proses awal yang berupa pembuatan konsep kasar dari game, mulai dari menentukan game seperti apa yang akan dibuat, mulai dari indentifikasi dari trending, topik, target user dari game yang akan dibuat. Output dari tahap initiation adalah konsep game dan deskripsi permainan yang sangat sederhana [3].
- 2) *Pra-production* adalah salah satu fase yang penting dalam siklus produksi. Pra-produksi melibatkan penciptaan dan revisi desain *game* dan pembuatan prototipe permainan. Desain *game* berfokus untuk menjelaskan *gameplay*, karakter, alur sistem, *storyboard*, tampilan antarmuka dan *asset* yang digunakan.
- 3) *Production* adalah proses inti yang berputar di sekitar penciptaan aset, pembuatan kode sumber, dan integrasi kedua elemen. Prototipe terkait dalam fase ini adalah perincian dan penyempurnaan formal. Rincian Formal adalah struktur yang di sempurnakan dengan mekanika dan aset yang lebih lengkap. Kegiatan produksi yang terkait dengan penciptaan dan penyempurnaan detail formal adalah menyeimbangkan permainan (terkait dengan kriteria kualitas yang seimbang), menambahkan fitur baru, meningkatkan kinerja secara keseluruhan, dan memperbaiki bug (terkait dengan kriteria kualitas fungsional dan internal yang lengkap). Penyeimbangan permainan yaitu penyesuaian yang terkait dengan kesulitan permainan untuk membuat kesulitan game yang tepat (*Leveling*). *Refinement* adalah prototipe lengkap yang merupakan subjek dari permainan. Kriteria kualitas terkait game fun dan dapat di akses. Kegiatan selama penyempurnaan di arahkan untuk membuat permainan lebih menyenangkan, menantang, dan lebih mudah di pahami. Hanya perubahan kecil yang di izinkan dalam fase ini.
- 4) *Testing* dalam konteks ini berarti pengujian internal di lakukan untuk menguji kegunaan permainan dan pemutaran. Metode pengujian khusus untuk setiap tahap prototipe. Perincian Formal Pengujian di lakukan menggunakan *playtest* untuk menilai fungsionalitas fitur dan kesulitan permainan (terkait dengan keseimbangan). Metode untuk menguji kriteria kualitas fungsional adalah melalui fitur *play testing*. Untuk menguji kriteria kualitas internal yang lengkap, dapat di lakukan melalui *playtesting* bersamaan dengan uji fungsi. Ketika tester menemukan bug, celah, atau kegagalan selama *play testing*, penyebab dan skenario untuk mereproduksi kesalahan perlu di dokumentasikan dan di analisis. Untuk menguji kriteria kualitas yang seimbang, bermain dengan beberapa perawatan yang berbeda di gunakan untuk mengkategorikan apakah perawatan terlalu sulit, terlalu mudah, atau baik-baik saja. Perbaikan Pengujian terkait dengan menyenangkan dan kriteria kualitas aksesibilitas. Dalam penyempurnaan pengujian, kesenangan di uji melalui *playtest* dan umpan balik langsung dari sesama pengembang, apakah itu membosankan, membuat frustrasi, menantang, dll. Aksesibilitas dapat di uji melalui pengamatan perilaku penguji. Jika tester merasa sulit untuk bermain dan memahami permainan, itu berarti bahwa game tersebut tidak cukup dapat di akses. Output dari pengujian adalah laporan bug, permintaan perubahan, dan keputusan pengembangan. Hasilnya akan memutuskan apakah sudah waktunya untuk maju ke fase berikutnya (Beta) atau mengulangi siklus produksi.
- 5) *Beta* adalah fase untuk melakukan pengujian pihak ketiga atau eksternal yang di sebut pengujian beta. Pengujian beta masih menggunakan metode pengujian yang sama dengan metode pengujian sebelumnya, karena prototipe terkait dalam pengujian beta adalah perincian dan penyempurnaan formal. Metode pemilihan tester datang dalam dua jenis: beta tertutup dan beta terbuka. Di tahap beta hanya memungkinkan individu yang di undang untuk menjadi peserta, sementara beta terbuka memungkinkan siapa saja yang mendaftar menjadi peserta. Kriteria kualitas dalam beta terkait erat dengan tahap prototipe saat ini. Dalam pengujian detail resmi, penguji di minta untuk menemukan bug (terkait dengan kriteria kualitas fungsional dan internal yang lengkap) [4]. Dalam penyempurnaan pengujian, penguji di beri lebih banyak kebebasan untuk menikmati permainan, karena sasaran lebih di arahkan untuk mendapatkan umpan balik (terkait dengan kriteria kualitas

aksesibilitas dan menyenangkan). Output dari pengujian beta adalah laporan bug dan masukan pengguna. Sesi Beta di tutup terutama karena 2 alasan, baik jangka beta berakhir atau jumlah penguji beta yang di tentukan telah memberikan laporan uji mereka. Dari sini, dapat menyebabkan siklus produksi lagi untuk memperbaiki produk atau terus merilis game jika hasilnya memuaskan.

- 6) Release adalah Sudah saatnya *build game* telah mencapai tahap akhir dan siap untuk di rilis ke publik. Rilis melibatkan peluncuran produk, dokumentasi proyek, berbagi pengetahuan, *post mortems*, dan perencanaan untuk pemeliharaan dan ekspansi permainan [3].

III. Hasil dan Pembahasan

A. Perancangan Use Case Diagram



Gambar 2. Use Case Diagram Game

Pada gambar 2 use case diagram akan dideskripsikan sebagai berikut :

1) Skenario Use Case Melalui Game

Tabel 1. Deskripsi Use Case Melalui

Nama Use Case	Memulai Game
Aktor	Player
Deskripsi	Proses ini adalah player akan memainkan game
Kondisi	Player Game
	<ol style="list-style-type: none"> 1. Memilih Menu Mulai 2. Mengecek Save Game 3. Memainkan Game sesuai Checkpoint Save Game 4. Mentransfer karakter pemain
Pre-Condition	Player Memainkan Permainan
Pos-Condition	Player memasuki permainan

2) Skenario Use Case Mengatur Game

Tabel 2. Deskripsi Use Case Mengatur Game

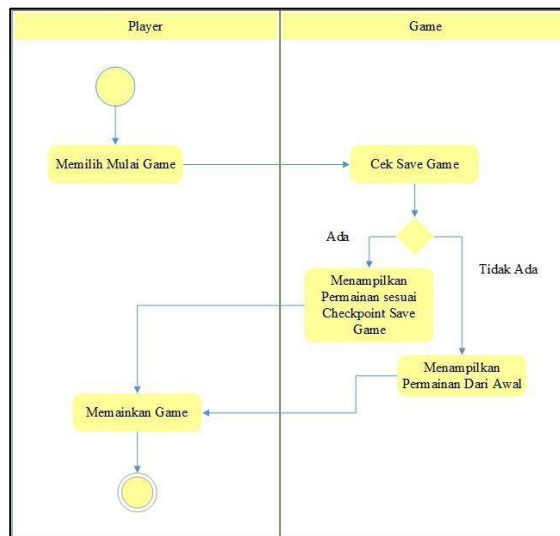
Nama Use Case	Mengatur Game
Aktor	Player
Deskripsi	Melihat menu pengaturan game

Kondisi	Player	Game
	Player memilih Pengaturan Game	Menampilkan Pengaturan Game
Pre-Condition	Player Memilih menu Pengaturan Game	
Pos-Condition	Player dapat melihat Pengaturan Game	

B. Activity Diagram

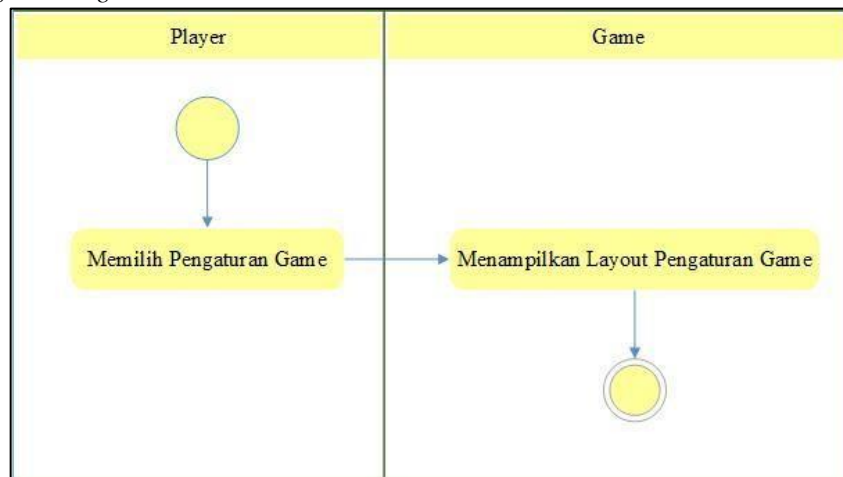
Diagram activity akan menampilkan aliran kerja atau aktivitas dari sebuah sistem yang ada pada aplikasi game. Di mana aktivitas ini akan menggambarkan aktivitas game yang akan menampilkan aktivitas-aktivitas yang dilakukan oleh system [5].

1) Activity Diagram Mulai Game



Gambar 3. Activity Diagram Mulai Game

2) Activity Diagram Pengaturan Game



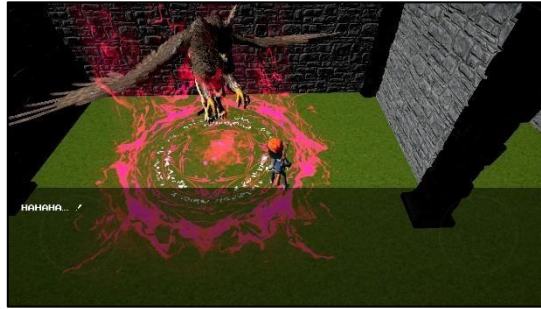
Gambar 4. Activity Diagram Pengaturan Game

C. Production

Produksi adalah proses inti yang berputar disekitar penciptaan aset, pembuatan kode sumber dan integrasi kedua elemen [6]. Prototipe terkait dalam fase ini adalah perincian dan penyempurnaan formal. Hasil dari game yang telah dirancang dapat dilihat pada tampilan gambar berikut:

1) Tampilan Menu Game

Gambar 5. Tampilan Menu *Game*2) Tampilan Pengaturan *Game*Gambar 6. Tampilan Pengaturan *Game*3) Tampilan *Gameplay*Gambar 7. Tampilan *Gameplay*4) Tampilan *Pause Game*Gambar 8. Tampilan *Pause Game*5) Tampilan *Game Over*Gambar 9. Tampilan *Game Over*6) Tampilan *Finish Labirin*

Gambar 10. Tampilan *Finish* Labirin7) *Tampilan Game Selesai*Gambar 11. Tampilan *Game Selesai*8) *Tampilan Credits Game*Gambar 12. Tampilan *Credits Game*D. *Pengujian*

Pengujian pada konteks ini berarti pengujian internal dan eksternal yang dilakukan untuk menguji kegunaan *game*. Metode pengujian khusus untuk setiap tahap *prototype*.

1) *Alpha Testing*

Setelah melakukan produksi, pengujian dilakukan menggunakan pengujian *Black Box* untuk mengecek apakah masih ada bug atau tidak dan akan ada kemungkinan pengurangan atau penambahan *feature*.

Tabel 3. Deskripsi *Use Case* Mengatur *Game*

No	Fungsi yang di uji	Cara Menguji	Hasil yang diharapkan	Hasil yang keluar
1	Mulai <i>Game</i>	Memilih menu mulai <i>game</i>	Mulai <i>game</i> dapat diakses dan dapat memasuki permainan	Sesuai dengan harapan <i>Valid</i>
2	Cek Menu	Memilih semua menu pada <i>game</i>	Semua layout dapat diakses	Sesuai dengan harapan <i>Valid</i>
3	<i>Save Game</i>	Pergi ke <i>Checkpoint</i>	Data permainan tersimpan	Sesuai dengan harapan <i>Valid</i>

2) Beta Testing

Setelah *game* selesai dibuat, belum berarti *game* tersebut akan diterima oleh masyarakat. Maka dari itu akan dilakukan *Eksternal testing*, dikenal dengan istilah *beta testing* dilakukan untuk menguji keberterimaan *game* dan untuk mendeteksi berbagai error dan keluhan yang dilemparkan oleh *thirdpartytester*. Beta berada diluar *productioncycle*, tetapi jika hasil dari testing ini berpotensi masih ada *error*, maka *productioncycle* akan diulangi lagi.

a) Pengujian UAT (User Acceptance Test)

Pengujian UAT yang dilakukan berlangsung di lingkungan user dan melibatkan beberapa pengujian ekstensif oleh sekelompok *user* [3]. Untuk mengetahui tanggapan *user* terhadap *game* yang dirancang, maka dilakukan pengujian dengan memberikan 5 pertanyaan kepada 20 user dimana jawaban dari pertanyaan tersebut terdiri dari tingkatan yang dapat dipilih sebagai berikut:

Tabel 4. Komponen Pilihan Jawaban Responden

A	Sangat : Mudah/Bagus/Sesuai/Bisa
B	Mudah/Bagus/Sesuai/Bisa
C	Netral
D	Cukup : Sulit/Bagus/Sesuai/Bisa
E	Sangat : Sulit/Jelek/Tidak Sesuai/Tidak Bisa

Tabel 5. Komponen Bobot Nilai Responden

Jawaban	Bobot
Sangat : Mudah/Bagus/Sesuai/Bisa	5
Mudah/Bagus/Sesuai/Bisa	4
Netral	3
Cukup : Sulit/Bagus/Sesuai/Bisa	2
Sangat : Sulit/Jelek/Tidak Sesuai/Tidak Bisa	1

Tabel 6. Pertanyaan Quesioner

No	Pertanyaan	A	B	C	D	E
1	Apakah Tampilan UI (User Interface) game ini menarik?					
2	Apakah game sangat menantang untuk dimainkan?					
3	Apakah objek dalam game tersebut sesuai dengan tema game?					
4	Apakah game dapat dijalankan di platform android?					
5	Apakah control game mudah untuk dikendalikan?					

Tabel 7. Data Jawaban Quesioner

No	Pertanyaan	Jawaban				
		A	B	C	D	E
1	Apakah Tampilan <i>User Interface</i> (UI) game ini menarik?	9	10	1		
2	Apakah game sangat menantang untuk dimainkan?	2	5	10	3	
3	Apakah objek dalam game tersebut sesuai dengan tema game?	5	13	2		
4	Apakah game dapat dijalankan di platform android?	6	13	1		
5	Apakah control game mudah untuk dikendalikan?	11	8		1	

Tabel 8. Data Quesioner Setelah Diolah

No	Pertanyaan	Nilai					Jumlah	NR
		5	4	3	2	1		
1	Apakah Tampilan <i>User Interface</i> (UI) game ini menarik?	45	40	3			88	4,4

2	Apakah <i>game</i> sangat menantang untuk dimainkan?	10	20	30	6	66	3,3
3	Apakah objek dalam <i>game</i> tersebut sesuai dengan tema <i>game</i> ?	25	52	6		83	4,15
4	Apakah <i>game</i> dapat dijalankan di platform <i>android</i> ?	30	52	3		85	4,25
5	Apakah <i>control</i> <i>game</i> mudah untuk dikendalikan?	55	32		2	89	4,45

Ket :

NR = Nilai Rata-rata (Jumlah/Responden)

NR% = Persentase (NR/Point Tertinggi × 100)

Dari data diatas dapat disimpulkan bahwa *game* tersebut memiliki tampilan *User Interface* (UI) yang menarik, cukup menantang, sesuai dengan tema dan dapat dijalankan diplatform android dengan *control* yang mudah.

IV. Kesimpulan

Berdasarkan hasil penerapan dan pembahasan yang telah dilakukan, maka dapat diambil kesimpulan bahwa pengembangan game dapat dilakukan dengan metode *Game Development Life Cycle* (GDLC) dan menggunakan *Engine Unreal*.

Daftar Pustaka

- [1] Mustofa, V. Ma'arif, R. Wijianto, and F. Pernando, "PENERAPAN BYL's GAME DEVELOPMENT LIFE CYCLE DALAM PERANCANGAN VIDEO GAME ORACLE FOR ANGEL," *J. Ilmu Pengetah. Dan Teknol. Komput.*, vol. 4, no. 2, pp. 191–196, 2019.
- [2] M. A. Asis, Purnawansyah, and A. R. Manga, "Penerapan System Development Life Cycle pada Sistem Validasi Metode Analisis Sediaan Farmasi," *Bul. Sist. Inf. dan Teknol. Islam*, vol. 1, no. 3, pp. 145–149, 2020.
- [3] R. A. Krisdiawan, M. F. Rohmana, and A. Permana, "Pembuatan Game Runaway From Culik Dengan Algoritma Fuzzy Mamdani," *Buffer Inform.*, vol. 6, no. 1, pp. 33–40, 2020.
- [4] B. Sulistio, S. Lutfi, and R. Ridwan, "Aplikasi Kamus Bahasa Taliabu Berbasis Android Dengan Menggunakan Metode Binary Search," *JIKO (Jurnal Inform. dan Komputer)*, vol. 2, no. 2, pp. 67–72, 2019, doi: 10.33387/jiko.v2i2.1316.
- [5] M. I. Java and R. E. Putra, "Rancang Bangun Aplikasi Drone Simulator Berbasis Android Menggunakan Game Engine Unity," *J. Informatics Comput. Sci.*, vol. 1, no. 01, pp. 26–33, 2019, doi: 10.26740/jinacs.v1n01.p26-33.
- [6] S. Gumuda, "Dynamics of the process of changes in concentration of methane in the air of ventilation currents in mines.," vol. 2, no. 2, pp. 13–21, 1978.