

PENERAPAN MODEL PENGEMBANGAN GAME GDLC (GAME DEVELOPMENT LIFE CYCLE) DALAM MEMBANGUN GAME PLATFORM BERBASIS MOBILE

Rio Andriyat Krisdiawan¹, Darsanto²

¹Fakultas Ilmu Komputer Universitas Kuningan

²Teknik Komputer Universitas Wiralodra

¹rioandriyat@uniku.com / rioandriyat@uniku.ac.id

²aziz.faziz310105@gmail.com / shanto.ft@unwir.ac.id

ABSTRAK

Permainan atau yang sering disebut *games* merupakan salah satu aktivitas yang sering dilakukan oleh kebanyakan orang dalam mencari hiburan dan kesenangan. Perkembangan games dari masa kemasa semakin beragam jenis dan aktivitasnya. Mulai dari games tradisional yang dilakukan secara konvensional sampai dengan games yang dilakukan dengan teknologi. Dilihat dari perkembangan teknologi, games bukan lagi sebagai aktivitas yang sulit untuk dimainkan, games dapat dimainkan dimana saja melalui perangkat teknologi apapun. Perkembangan games selain dari teknologi, juga sangat berkembang dari segi jenis/*genre* dari games itu sendiri, yang menyesuaikan dengan kebutuhan pengguna/*player*. Berbagai *genre* game pun muncul, seperti *genre action, adventure, RPG, simulasi, strategi* dan lainnya. Pada penelitian kali ini peneliti berfokus kepada *subgenre games action* yaitu *platform games*. Beberapa platform games yang terkenal yaitu Mario Bros, *Jumping Flash, Donkey Kong*. Games yang ringan tetapi memiliki unsur kesenangan dan menantang bagi yang memainkannya. Peneliti merasa tertarik dalam meneliti bagaimana membangun sebuah game platform dengan teknik pengembangan game GDLC (*Game Development Life Cycle*), dengan model perancangan game menggunakan model UML (*Unified Modeling Language*). Game yang akan dikembangkan yaitu *game platform* yang merupakan *subgenre* dari *genre action games* yang dapat dijalankan pada perangkat *mobile phone* berbasis *android*.

Kata Kunci : *Games Platform, GDLC, UML.*

ABSTRACT

Games or often called games are one of the activities that are often carried out by most people in seeking entertainment and pleasure. The development of games from time to time increasingly diverse types and activities. Starting from traditional games carried out in a conventional manner to games made with technology. Viewed from technological developments, games are no longer an activity that is difficult to play, games can be played anywhere through any technological device. The development of games apart from technology, is also very developed in terms of types / genres from the games themselves, which adjust to the needs of users / players. Various game genres also appear, such as the genre of action, adventure, RPG, simulation, strategy and others. In this study the researchers focused on action games subgenre, platform games. Some well-known platform games are Mario Bros, Jumping Flash, Donkey Kong. Games are lightweight but have elements of fun and challenge for those who play them. Researchers feel interested in researching how to build a game platform with GDLC game development techniques (Game Development Life Cycle), with the game design model using the UML (Unified Modeling Language) model. The game that will be developed is a platform game which is a subgenre of the action games genre that can be run on an Android-based mobile phone.

Keywords: *Games Platform, GDLC, UML*

I. PENDAHULUAN

Permainan atau yang sering disebut *games* merupakan salah satu aktivitas yang sering dilakukan oleh kebanyakan orang dalam mencari hiburan dan kesenangan. Perkembangan *games* dari masa kemasa semakin beragam jenis dan aktivitasnya. Mulai dari *games* tradisional yang dilakukan secara konvensional sampai dengan *games* yang dilakukan dengan teknologi. Teknologi dalam permainan sangatlah beragam, mulai dengan teknologi sederhana, sampai teknologi yang sangat kompleks.

Dilihat dari perkembangan teknologi, *games* bukan lagi sebagai aktivitas yang sulit untuk dimainkan, *games* dapat dimainkan dimana saja melalui perangkat teknologi apapun. Hal tersebut tidak terlepas dari perkembangan teknologi yang memungkinkan berbagai macam *game* dapat dimainkan dalam berbagai platform system operasi. Mulai dari *games* yang dapat dimainkan di perangkat Komputer, *Handphone*, *MobilePhone*, *Xbox*, *Console* dan lainnya.

Perkembangan *games* selain dari teknologi, juga sangat berkembang dari segi jenis/*genre* dari *games* itu sendiri, yang menyesuaikan dengan kebutuhan pengguna/*player*. Berbagai *genre* *game* pun muncul, seperti *genre action*, *adventure*, *RPG*, simulasi, strategi dan lainnya. Dari *genre* tersebut masih dapat dikelompokkan ke berbagai jenis *subgenre* seperti halnya *game action*; yaitu *game* yang menampilkan aksi laga yang penuh dengan adegan baku tembak, menghindari rintangan yang memerlukan konsentrasi untuk mencapai point tertinggi. *Subgenre* dari *game action* seperti *platform games*, *shooter games*, *fighting games*, *survival games*.

Pada penelitian kali ini peneliti berfokus kepada *subgenre games action* yaitu *platform games*. Beberapa *platform games* yang terkenal yaitu *Mario bros*, *Jumping Flash*, *Donkey Kong*. *Games* yang ringan tetapi memiliki unsur kesenangan dan menantang bagi yang memainkannya. Menghindari rintangan, melawan musuh, dan mencapai skor tertinggi dari permainan tersebut.

Dalam membangun *game* yang sangat menarik seperti itu pastinya pengembang *game/game development* memiliki tahapan sendiri dalam membuat dan mengembangkannya, sehingga *player* dapat

terus menerus tertarik dan selalu memainkan *game* tersebut. Tidak bisa dipungkiri, *games platform* sampai dengan saat ini masih banyak digemari dan dimainkan oleh berbagai kalangan, karena unsur *game* tersebut tidak mengenal batasa usia untuk dimainkan.

Peneliti merasa tertarik dalam meneliti bagaimana membangun sebuah *game platform* dengan teknik pengembangan *game GDLC (Game Development Life Cycle)*. *GDLC* adalah suatu proses pengembangan sebuah *game* yang menerapkan pendekatan iteratif yang terdiri dari 6 fase pengembangan, dimulai dari fase inialisasi/pembuatan konsep, *preproduction*, *production*, *testing*, *beta* dan *realease*. Dari 6 fase tersebut dapat dikelompokkan menjadi 3 proses utama yaitu; Proses Inialisasi yang terdiri dari konsep dan design, Proses produksi terdiri dari Pra Produksi, Produksi, dan Pengujian (*Alpha* dan *Beta*), dan *Realease*.

Untuk model perancangan *game* pada penelitian ini penulis menggunakan model *UML(Unifield Modeling Language)*; yaitu suatu pemodelan desain yang menerapkan pemrograman berorientasi objek (*OOP*) didalamnya. *Game* yang akan dikembangkan yaitu *game platform* yang merupakan *subgenre* dari *genre action games* yang dapat dijalankan pada perangkat *mobile phone* berbasis *android*.

Dari latarbelakang diatas, peneliti merumuskan permasalahan dalam penelitian ini sebagai berikut:

1. Bagaimana Menerapkan model pengembangan *game GDLC* dalam membangun *Game Platform*.
2. Bagaimana tahapan *GDLC* dalam menentukan pembuatan *game* yang menarik dan menyenangkan.
3. Bagaimana merancang sebuah *game platform* dengan model perancangan sistem berbasis *UML*.

Maksud dari penelitian ini adalah untuk menerapkan proses pengembangan *game platform* dengan menggunakan model pengembangan yang sesuai, yaitu *Game Development Life Cycle (GDLC)*.

Tujuan dari penelitian ini adalah untuk merancang dan membangun sebuah *game platform 2D* yang menarik dan menyenangkan bagi pengguna.

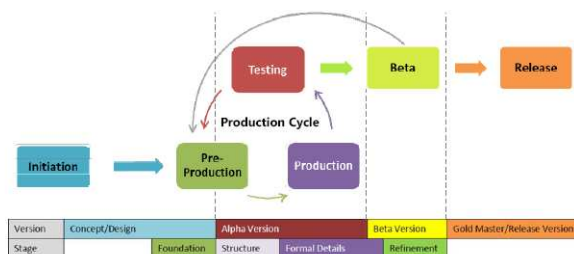
II. METODELOGI PENELITIAN

A. Metode Pengembangan Game

Metodologi pengembangan game yang digunakan oleh penulis dalam melakukan penelitian ini adalah *Game Development Life Cycle (GDLC)*.

Metode pengembangan sistem yang digunakan dalam perancangan aplikasi perangkat lunak ini menggunakan metodologi kerja GDLC (*Game Development Life Cycle*). GDLC adalah suatu proses pengembangan sebuah game yang menerapkan pendekatan iteratif yang terdiri dari 6 fase pengembangan, dimulai dari fase inialisasi/pembuatan konsep, preproduction, production, testing, beta dan release. Dari 6 fase tersebut dapat dikelompokkan menjadi 3 proses utama yaitu:

- Proses Inialisasi yang terdiri dari konsep dan design.
- Proses produksi terdiri dari Pra Produksi, Produksi, dan Pengujian (Alpha dan Beta).
- Realease



Gambar 1. Fase dan Proses GDLC

B. Penjelasan Fase dan Proses GDLC:

1. Initiation (Inisiasi)

“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”.

Pada tahap ini peneliti akan membuat sebuah konsep game yang akan dibangun, berupa analisa akan seperti apa game itu

dibangun, initiation akan menghasilkan konsep game dan deskripsi sederhana game. Pada initiation akan menjabarkan tentang scenario game, karakter, cerita dalam game, target pemain, platform yang digunakan dan game engine.

- Jenis permainan yang dibuat
- Game skenario
- Game karakter
- Story game
- Target user/pengguna
- Platform game yang digunakan

2. Pra-Production

“Pra-produksi adalah salah satu fase yang penting dalam siklus produksi. Pra-produksi melibatkan penciptaan dan revisi desain game dan pembuatan prototipe permainan. Desain game berfokus pada mendefinisikan genre permainan, gameplay, game mekanik/konvensional, alur cerita, karakter, tantangan, faktor kesenangan, aspek teknis, dan dokumentasi elemennya dalam *Dokumen Desain Game (GDD)*.”

Pra-produksi berakhir ketika revisi atau perubahan desain game telah disetujui dan didokumentasikan di *GDD*”

Pada tahap *Pra-production* peneliti akan melakukan pengumpulan data dengan cara mencari referensi, yang berkaitan dengan game yang akan dibuat, seperti:

- Genre yang akan di pakai pada *game* ini yaitu *game platform*.
- Skenario game berupa pegunungan, karakter pada game yaitu pokemon, enemy berupa semut, landak dan pokemon ball sebagai peluru untuk weapon pokemon.
- Scenario game player berupa pokemon berpetualangan menghindari enemy, dan mengumpulkan pokemon ball sebagai senjata untuk melawan enemy. Player menjelajahi rintangan sambil mengumpulkan koin gold sebagai point.
- Target pemain pada game ini yaitu masyarakat umum.
- Peneliti menggunakan game engine *Construct 2D* dan menggunakan *Corel* dan *Photoshop* untuk membuat aset yang akan di terapkan pada *game*.

3. Production

“Produksi 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”.

Gamedesign dan *prototype* dan point lainnya yang ada pada *pre-production* yang disempurnakan pada tahap ini. Artinya pada tahap ini penulis memiliki fokus pada menerjemahkan rancangan *mengumpulkan data, desain game, membuat skenario permainan, membuat aset untuk karakter player maupun enemy, pepohonan, makanan player dalam game*, dan aspek – aspek lainnya menjadi unsur penyusun *game*. Tahap ini mencakup *assetcreation, programming* dan *integration* antara *asset* dan *sourcecode*.

4. Pengujian

“Pengujian dalam konteks ini berarti pengujian *internal* dan *eksternal* dilakukan untuk menguji kegunaan permainan. Metode pengujian khusus untuk setiap tahap *prototype*”.

a) Alpha Testing

Setelah melakukan produksi, Pengujian ini dilakukan *peneliti* untuk melakukan pengecekan apakah masih ada bug atau tidak dan akan ada kemungkinan pengurangan atau penambahan *feature*. Jika ada bug/eror dan penambahan *feature* maka peneliti akan memperbaikinya.

b) Beta Testing

Setelah *game* selesai dibuat, belum berarti *game* tersebut akan diterima oleh masyarakat. Peneliti akan melakukan *Eksternaltesting*, dikenal dengan istilah *betatesting* akan dilakukan oleh peneliti 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 eror maka peneliti akan mengulangi *productioncycle*.

5. Release

Jika *Game* yang sudah selesai dibuat dan lulus *betatesting* menandakan *game* tersebut maka peneliti sudah siap untuk dirilis.

C. Model Perancangan UML

“*Unified Modelling Language (UML)* adalah keluarga notasi grafis yang didukung oleh model-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek”.(Martin Fowler,2004).

UML adalah sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek. UML juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan. Aplikasi atau sistem yang tidak terdokumentasi biasanya dapat menghambat pengembangan karena *developer* harus melakukan penelusuran dan mempelajari kode program. UML juga dapat menjadi alat bantu untuk *transfer* ilmu tentang sistem atau aplikasi yang akan dikembangkan dari satu *developer* ke *developer* lainnya. Tidak hanya antar *developer* terhadap orang bisnis dan siapapun dapat memahami sebuah sistem dengan adanya UML.

UML diciptakan oleh Object Management Group yang diawali dengan versi 1.0 pada Januari 1997. Dalam pengembangan berorientasi objek ada beberapa prinsip yang harus dikenal:

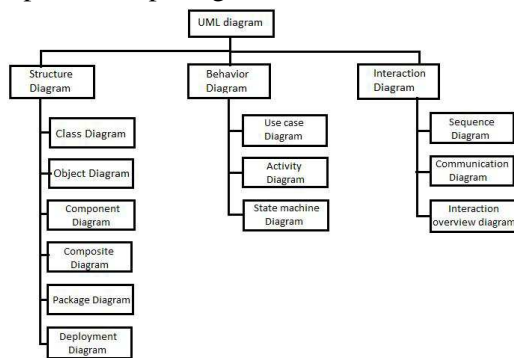
- a. Object
- b. Class
- c. Abstraction
- d. Encapsulation
- e. Inheritance
- f. Polymorphism.

Dengan menggunakan *UML* kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena *UML* juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, *UML* tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, *UML* mendefinisikan notasi dan *syntax/semantik*. Notasi *UML* merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai

diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

Menurut Rosa A.S dan Salahuddin, UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam macam diagram tersebut dapat dilihat pada gambar di bawah ini.

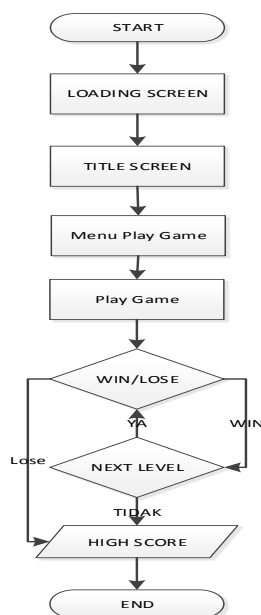


Gambar 2. Macam-macam Diagram UML

Pemodelan yang akan digunakan oleh peneliti pada penelitian ini yaitu; *Use case diagram*, *Activity Diagram*, *class diagram* dan *sequence diagram*.

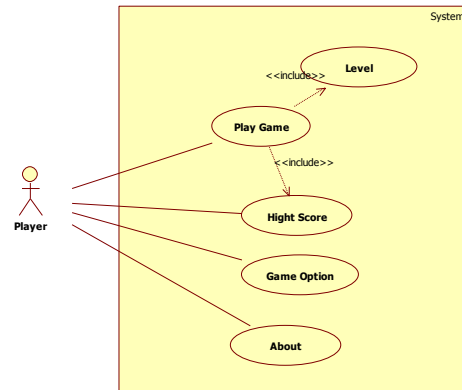
III. HASIL DAN PEMBAHASAN

A. Gameplay



Gambar 3. Gameplay Pika Adventure

B. Perancangan Use Case Diagram



Gambar 4. Use case diagram Game

Pada gambar 4 use case diagram akan di deskripsikan sebagai berikut :

a. Skenario Use Case Play Game

Tabel 1. Deskripsi Use Case Play Game

Nama Use Case	Play Game	
Aktor	Player	
Deskripsi	Proses Ini adalah player akan memainkan game .	
Kondisi	Player	Game
	1. Memilih menu Play.	2. Mengecek Lock Level
	3. Memilih level	4. Menampilkan level
	5. Memainkan Game sesuai level	6. Menampilkan layout game sesuai level yang dipilih
Pre-Condition	Player Memainkan Permainan	
Post-Condition	Player Memilih tipe Permainan dan Level sebelum bermain	

b. Skenario Use Case High Score

Tabel 2. Deskripsi Use Case High Score

Nama Use Case	High Score	
Aktor	Player	
Deskripsi	Melihat High Score	
Kondisi	Player	Game
	Player memilih High Score	Menampilkan High Score
Pre-Condition	Player Memilih menu High Score	
Post-Condition	Player dapat melihat High Score	

c. Skenario Use Case Game Option

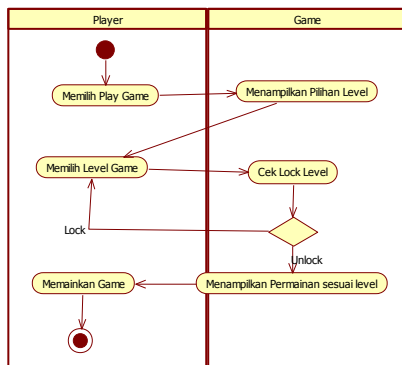
Tabel 3. Deskripsi Use Case Game Option

Nama Use Case	Game Option	
Aktor	Player	
Deskripsi	Melihat menu game option.	
Kondisi	Player	Game
	Player memilih Game Option	Menampilkan Game Option
Pre-Condition	Player Memilih menu Game Option	
Post-Condition	Player dapat melihat option game control	

C. Activity Diagram

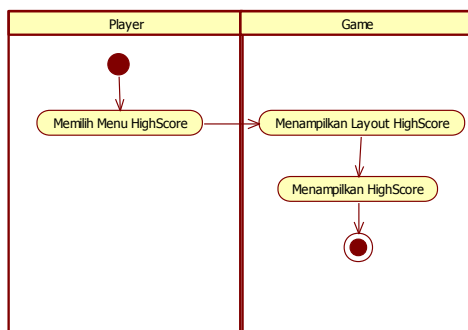
Diagram *activity* akan menampilkan aliran kerja atau aktivitas dari sebuah sistem yang ada pada aplikasi *game*. Di mana aktivitas ini akan menggambarkan aktivitas *game* dan bukan yang di lakukan oleh aktor terhadap aplikasi, jadi *activity* diagram ini akan menampilkan aktivitas-aktivitas yang dilakukan oleh *system*.

a. Activity Diagram Play Game



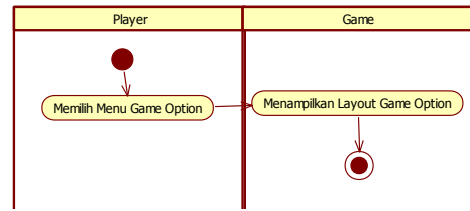
Gambar 5. Activity diagram Play Game

b. Activity Diagram High Score



Gambar 6. Activity diagram High Score

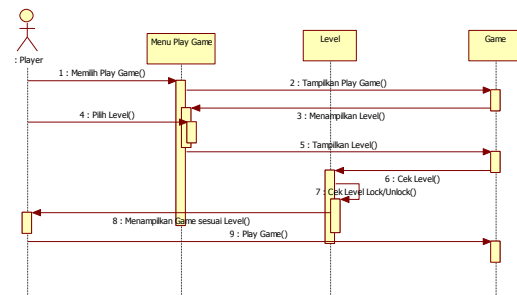
c. Activity Diagram Game Option



Gambar 7. Activity diagram Game Option

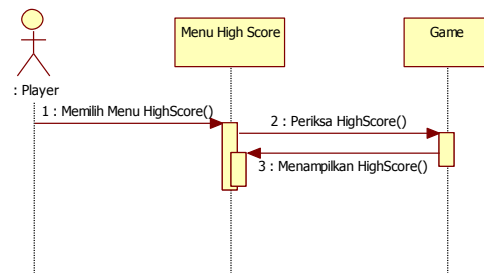
D. Sequence Diagram

a. Sequence Diagram Play Game



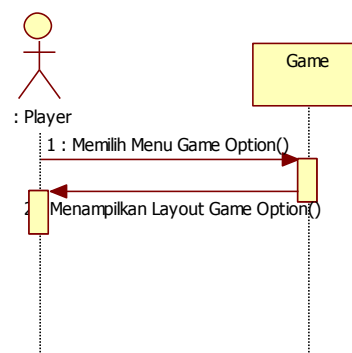
Gambar 8. sequence diagram Edukasi

b. Sequence Diagram High Score



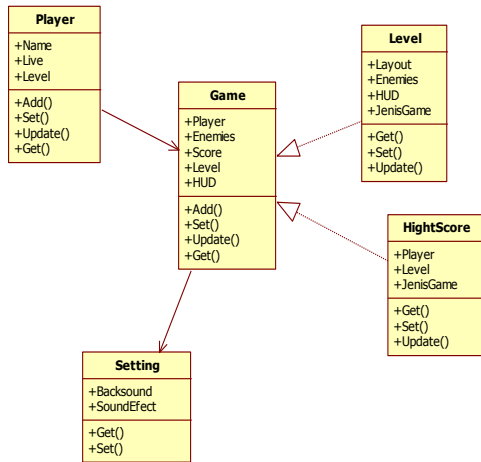
Gambar 9. Sequence diagram High Score

c. Sequence Diagram Game Option



Gambar 10. Sequence diagram Game Option

E. Class Diagram

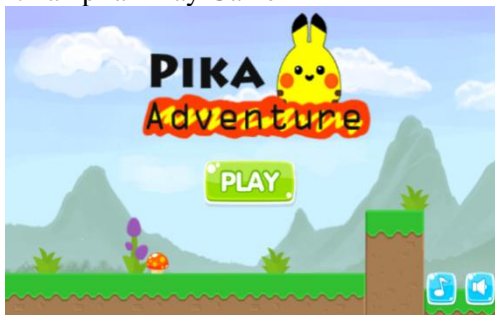


Gambar 11. Class diagram Game

F. Production

Produksi 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. Hasil dari game yang sudah dibuat dapat dilihat pada tampilan Gambar berikut :

1. Tampilan Play Game



Gambar 12. Tampilan Play Game

2. Tampilan Level



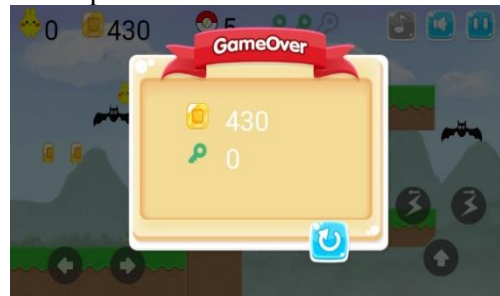
Gambar 13. Tampilan Level.

3. TampilanPlay Game



Gambar 14. Halaman Play Game

4. Tampilan Game Over



Gambar 15. Halaman Game Over

5. Tampilan Victory Condition



Gambar 16. Halaman Victory Condition

G. Pengujian

Pengujian dalam konteks ini berarti pengujian internal dan eksternal dilakukan untuk menguji kegunaan permainan. Metode pengujian khusus untuk setiap tahap prototipe

1. Alpha Testing

Setelah melakukan produksi, Pengujian ini dilakukan *peneliti* untuk melakukan pengecekan apakah masih ada bug atau tidak dan akan ada kemungkinan pengurangan atau penambahan *feature*. Jika ada bug/eror dan penambahan *feature* maka peneliti akan memperbaikinya.

a. Pengujian *Black Box*

Tabel 4. Hasil Pengujian *Black Box*

No	Fungsi yang di uji	Cara Menguji	Hasil yang diharapkan	Hasil yang keluar
1	Play Game	Memilih menu playgame	Playgame dapat diakses dan layot playgame dapat diakses	Sesuai dengan harapan <i>Valid</i>
2	Cek Menu	Memilih semua menu pada game	Semua layout dapat diakses	Sesuai dengan harapan <i>Valid</i>
3.	Level	Mengakses Level	Level hanya bias diakses jika terbuka	Sesuai dengan harapan <i>Valid</i>

2. *Beta Testing*

Setelah game selesai dibuat, belum berarti game tersebut akan diterima oleh masyarakat. Peneliti akan melakukan *Eksternaltesting*, dikenal dengan istilah *betatesting* akan dilakukan oleh peneliti 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 eror maka peneliti akan mengulangi *productioncycle* lagi.

a. Pengujian *UAT (User Acceptance Test)*

Pengujian *UAT* yang dilakukan penulis berlangsung di lingkungan user dan melibatkan beberapa pengujian ekstensif oleh sekelompok user. Untuk mengetahui tanggapan user terhadap game yang dibangun, maka dilakukan pengujian dengan memberikan 4 pertanyaan kepada 20 user dimana jawaban dari pertanyaan tersebut terdiri dari tingkatan yang dapat dipilih sebagai berikut:

Tabel 5 Komponen Pilihan Jawaban responden

A	Sangat : Mudah/Bagus/Sesuai/Jelas
B	Mudah/Bagus/Sesuai/Jelas

C	Netral
D	Cukup : Sulit/Bagus/Sesuai/Jelas
E	Sangat : Sulit/Jelek/Tidak Sesuai/Tidak Jelas

Tabel 6 Komponen Bobot Nilai responden

Jawaban	Bobot
A. Sangat : Mudah/Bagus/Sesuai/Jelas	5
B. Mudah/Bagus/Sesuai/Jelas	4
C. Netral	3
D. Cukup : Sulit/Bagus/Sesuai/Jelas	2
E. Sangat : Sulit/Jelek/Tidak Sesuai/Tidak Jelas	1

Tabel 7 Pertanyaan Questioner

No	Pertanyaan	A	B	C	D	E
1	Apakah Tampilan UI (<i>User Interface</i>) 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 8 Data Jawaban Questioner

NO	Pertanyaan	Jawaban					Persentase				
		A	B	C	D	E	A	B	C	D	E
1	Apakah Tampilan UI (<i>User Interface</i>) game ini menarik?	15	3	2	1		75 %	15 %	10 %	5 %	0 %
2	Apakah game sangat menantang untuk dimainkan?	10	5	3	2	1	50 %	25 %	15 %	10 %	5 %
3	Apakah objek dalam game tersebut sesuai dengan tema game?	10	6	3	1		50 %	30 %	15 %	5 %	0 %
4	Apakah game dapat dijalankan di platform android?	8	8	2	2	1	40 %	40 %	10 %	10 %	5 %
5	Apakah control game mudah untuk dikendalikan?	10	3	3	2	2	50 %	15 %	15 %	10 %	10 %

Data yang didapat di atas diolah dengan cara mengalikan setiap poin jawaban dengan bobot yang sudah ditentukan sesuai dengan tabel bobot nilai jawaban.

Tabel 9 Data Kuesioner Setelah Diolah

NO	Pertanyaan	Nilai					jml	NR	NR %
		5	4	3	2	1			
1	Apakah Tampilan UI (User Interface) game ini menarik?	75	12	6	2	0	95	4,75	95
2	Apakah game sangat menantang untuk dimainkan?	50	20	9	4	1	84	4,2	84
3	Apakah objek dalam game tersebut sesuai dengan tema game?	50	24	9	2	0	85	4,25	85
4	Apakah game dapat dijalankan di platform android?	40	32	6	4	1	83	4,15	83
5	Apakah control game mudah untuk dikendalikan?	50	12	9	4	2	77	3,85	77

Ket :

NR = Nilai Rata-rata (Jumlah / Responden)

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

Dari data diatas dapat disimpulkan bahwa game tersebut memiliki Tampilan UI (User Interface) yang menarik, sangat menantang, sesuai dengan tema dan dapat dijalankan di platform android dengan control yang mudah.

IV. KESIMPULAN

Berdasarkan hasil penerapan dan pembahasan yang telah dilakukan, maka dapat ditarik kesimpulan sebagai berikut.

1. GDLC sangat cocok digunakan dalam mengembangkan pembuatan game.
2. GDLC berperan dalam game development team dalam membangun game.
3. Model Prancangan UML memberikan kemudahan perancangan system untuk pemrograman OOP seperti perancangan game.
4. Pengujian game sangat cocok jika menerapkan model pengujian alfa dan beta testing seperti yang dibahas.

DAFTAR PUSTAKA

- [1]. Fowler, Martin. *UML Distilled: Panduan Singkat Bahasa Pemodelan Object Standar, Edisi 3*, Penerbit Andi: Yogyakarta. 2004.
- [2]. <https://usersnap.com/blog/user-acceptance-testing-right> : Matthew Setter
- [3]. Krisdiawan, R. A. (2018). Implementasi Model Pengembangan Sistem Gdlc Dan Algoritma Linear Congruential Generator Pada Game Puzzle. *Nuansa Informatika*, 12(2).
- [4]. Pressman, Roger S. (2007). *Rekayasa Perangkat Lunak: pendekatan praktisi* (Buku1). Beizer, B. (1995). *Black-Box Testing*, Wiley. Yogyakarta: Andi
- [5]. Ramadan Rido and Yani Widyaningrum; *Game Development Life Cycle Guid; ICAC SIS 2013 ISBN: 978-979-1421-19-5; September 2013.*
- [6]. Rosa AS dan M. Shalahuddin. 2015. *Rekayasa Perangkat Lunak Terstruktur Dan Berorientasi Objek*. Bandung : INFORMATIKA

Halaman ini sengaja dikosongkan