

RANCANG BANGUN GAME 3D CZEL BERBASIS ANDROID DENGAN METODE LINEAR CONGRUENTIAL GENERATOR

Hansen¹, Tony Darmanto², Kartono³

¹²³Teknik Informatika STMIK Widya Dharma, Pontianak

e-mail: ¹leonardushansen90@gmail.com, ² tony.darmanto@yahoo.co.id , ³kartono1102@gmail.com

Abstract

First Person Shooter Game is one type of game (genre) that is favored because it uses the first person view in which the player seems to be the main character in the game that surrounded by weapons and combat bullet. Combat bullets in FPS games are usually used as ammunition to fire with weapons used by players. This combat bullet is designed according to the needs of the programmer in its design. The Linear Congruential Generator (LCG) method is one of the Pseudo Random Number Generator (PRNG) that is widely used in computer programs to generate random numbers and has many advantages such as speed, ease of implementation, portable code, parameter and test results. Therefore, this study intends to find out how to apply the Linear Congruential Generator (LCG) method in the Android platform FPS game designed so that the random numbers generated can be used to randomize the type of ammunition and the location of ammunition dropped.

Keywords: First Person Shooter, Linear Congruential Generator, Android.

Abstrak

Game First Person Shooter (FPS) adalah salah satu jenis game (genre) yang digemari karena menggunakan pandangan orang pertama di mana pemain seolah-olah menjadi karakter utama dalam game yang berpusat pada permainan di sekitar senjata-senjata dan peluru tempur. Peluru tempur pada permainan FPS biasanya digunakan sebagai amunisi untuk melakukan tembakan dengan senjata-senjata yang digunakan oleh pemain. Peluru tempur ini biasanya ditempatkan sesuai dengan keinginan programmer dalam perancangannya. Metode Linear Congruential Generator (LCG) merupakan salah satu dari pembangkit bilangan acak Semu atau Pseudo Random Number Generator (PRNG) yang banyak digunakan dalam program komputer untuk menghasilkan bilangan acak dan memiliki banyak keuntungan seperti kecepatan, kemudahan implementasi, ketersediaan kode portabel, parameter dan hasil tes. Oleh karena itu, penelitian ini bermaksud mencari tahu bagaimana cara menerapkan metode Linear Congruential Generator (LCG) pada game FPS ber-platform Android yang dirancang sehingga bilangan acak yang dihasilkan dapat digunakan untuk mengacak jenis amunisi dan lokasi amunisi dijatuhkan.

Kata Kunci: First Person Shooter, Linear Congruential Generator, Android.

1. PENDAHULUAN

Game telah menjadi satu hal yang ada di dalam keseharian masyarakat. Dahulu, game hanya dijadikan sarana hiburan semata namun sekarang game telah menjadi luas fungsinya, misalnya game dapat dijadikan sarana pembelajaran, lahan bisnis, dan dipertandingkan sebagai salah satu dari cabang olahraga oleh para profesional. Perkembangan game platform juga dapat dilihat secara langsung oleh masyarakat, pada mulanya game hanya dimainkan di komputer dan console tetapi sekarang sudah memasuki era mobile game.

Mobile game adalah sebuah game yang didesain dan dimainkan oleh mobile devices, seperti PDA, smartphone, tablet PCs, dan portable media player. Dan sekarang ini, mobile game telah dibuat di berbagai macam platform seperti Symbian, Apple IOS, Android serta Windows Phone. Keuntungan tersendiri memainkan mobile game adalah portabilitas, yaitu player dapat bermain game dimana saja selama masyarakat mempunyai mobile devices yang mampu menjalankan mobile games. Perencanaan dan pembuatan sebuah game membutuhkan imajinasi atau wawasan yang luas. Apa yang terjadi bila seorang programmer tidak mengetahui seluk beluk tentang game yang akan dibangun. Untuk itu perlu adanya penjelasan atau sosialisasi tentang game yang nantinya akan membuka wawasan seorang programmer untuk membuat game.

Saat ini perkembangan *game* sangat pesat. *Game* yang pertama hanya berupa permainan *tenis for two* pada Osiloskop (alat ukur elektronika yang berfungsi memproyeksikan bentuk sinyal listrik agar dapat dilihat dan dipelajari). Kemudian berkembang hingga menjadi *game* 3 Dimensi. Seiring berkembangnya sistem operasi berbasis Android, pembuatan *game* dalam *platform* Android diharapkan dapat meningkatkan minat para *programmer* untuk membuat aplikasi-aplikasi yang ber-*platform independent*.

Dengan munculnya sistem operasi berbasis Android, *programmer* berlomba-lomba untuk membuat *game smartphone* yang menarik, menantang, dan tentunya dapat menghibur *user*. Salah satu contoh *game* yang dapat dikatakan menarik, menantang, dan tentunya menghibur *user* adalah *game First Person Shooter* (FPS). *Game First Person Shooter* (FPS) adalah salah satu jenis *game* (*genre*) yang digemari karena menggunakan pandangan orang pertama dimana pemain seolah-olah menjadi karakter utama dalam *game* yang berpusat pada permainan di sekitar senjata-senjata dan peluru tempur. Peluru tempur pada permainan FPS biasanya digunakan sebagai amunisi untuk melakukan tembakan dengan senjata-senjata yang digunakan oleh pemain. Peluru tempur ini biasanya ditempatkan sesuai dengan keinginan *programmer* dalam perancangannya.

Metode *Linear Congruential Generator* (LCG) merupakan salah satu dari pembangkit bilangan acak Semu atau *Pseudo Random Number Generator* (PRNG). Metode ini paling banyak digunakan dalam *program* komputer untuk menghasilkan bilangan acak. Keuntungan praktisnya adalah kecepatan, kemudahan implementasi, dan ketersediaan kode portable, parameter dan hasil tes.

Dengan demikian, penulis bermaksud untuk membuat sebuah *Game* 3D Czel dalam *platform* Android dengan konsep unik yang menantang dan dapat menghibur. Dalam penerapannya penulis menggunakan metode *Linear Congruential Generator* (LCG) untuk menghasilkan bilangan acak yang digunakan untuk mengacak jenis amunisi dan mengacak lokasi amunisi dijatuhkan.

2. METODE PENELITIAN

2.1 Metode Penelitian

Penulis menggunakan beberapa metode penelitian dalam menyusun skripsi ini yaitu: Rancangan Penelitian, Metode Pengumpulan Data, Teknik Analisis Aplikasi, Teknik Perancangan Aplikasi.

2.1.1 Rancangan Penelitian

Penulis menggunakan rancangan penelitian deskriptif dan perancangan eksperimen dalam penelitian ini. Desain penelitian ini akan memberikan gambaran mengenai bagaimana penerapan metode *Linear Congruential Generator* pada *platform* Android. Eksperimen dilakukan dengan mengimplementasikan aplikasi *game* ke dalam *smartphone* dengan *platform* Android.

2.1.2 Metode Pengumpulan Data

Metode pengumpulan data yang dilakukan adalah dengan menelusuri literatur dan artikel yang dapat digunakan sebagai bahan yang dapat digunakan untuk menyusun penelitian.

2.1.3 Teknik Analisis Aplikasi

Penulis menggunakan *Unified Modeling Language* (UML) sebagai teknik analisis sistem yang digunakan untuk menggambarkan, dan mendokumentasikan perancangan aplikasi *game*.

2.1.4 Teknik Perancangan Aplikasi

Perancangan sistem menggunakan *game engine* *Unity 2018.2.8f1*, bahasa pemrograman *Microsoft Visual Studio 2017* untuk melakukan *code program* dan menerapkan metode *Linear Congruential Generator*.

2.2 Landasan Teori

2.2.1 Game

Sebuah *game* adalah kegiatan pemecahan masalah, dilakukan dengan tindakan bermain. [1]. *Game* adalah jenis kegiatan bermain, dilakukan dalam konteks pura-pura nyata, dimana peserta mencoba untuk mencapai setidaknya satu wewenang, tujuan yang tidak sepele dengan bertindak sesuai dengan aturan. [2].

2.2.2 First Person Shooter (FPS)

First-person shooter adalah sebuah tipe permainan video dimana pemain mengasumsikan cara pandang pemeran utama, sehingga kamera permainan termasuk senjata karakter, akan tetapi sisa model karakter tidak terlihat. [3]. *First-person shooter* adalah jenis permainan video yang permainannya melibatkan penembakan terhadap musuh dan target lain dan dimana pemain melihat aksi seolah-olah melalui mata karakter yang mereka kontrol. [4].

2.2.3 Linear Congruential Generator (LCG)

Linear Congruential Generator adalah sebuah algoritma yang baik untuk mengacak sejumlah angka, disingkat dengan LCG. Pada algoritma ini, kita mulai dengan sebuah angka awal x_0 , sebuah pengali a , penambah c , dan sebuah modulo m .

$$x_n = (a x_{n-1} + c) \bmod m, 0 < x_n < m. [5].$$

Linear Congruent Method (LCM) merupakan pembangkit bilangan acak yang banyak digunakan dalam program komputer. LCM membangkitkan bilangan acak yang didefinisikan dengan:

$$Z_i = (a Z_{i-1} + c) \bmod m$$

Dimana :

Z_i = bilangan acak ke i

Z_{i-1} = bilangan acak sebelumnya

a = faktor pengali

c = *increment*

m = *modulus*

Syarat-syarat dalam menentukan LCM adalah sebagai berikut:

- Konstanta a harus lebih besar dari \sqrt{m}
- Untuk konstanta c harus berangka ganjil apabila m bernilai pangkat dua. Tidak boleh nilai dari kelipatan m
- Untuk m harus bilangan prima
- Nilai pertama z_0 harus merupakan angka *integer* dan juga ganjil cukup besar. [6].

2.2.4 Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. [7]. Dalam membuat program biasanya yang banyak dipakai untuk menentukan target versi Android adalah nomor *API Level*. [8].

Tabel 1. API LEVEL ANDROID

Code name	Version Number	Initial release date	API level	Support status
Alpha	1.0	September 23, 2008	1	Discontinued
Beta	1.1	February 9, 2009	2	Discontinued
Cupcake	1.5	April 27, 2009	3	Discontinued
Donut	1.6	September 15, 2009	4	Discontinued
Eclair	2.0-2.1	October 26, 2009	5-7	Discontinued
Froyo	2.2-2.2.3	May 20, 2010	8	Discontinued
Gingerbread	2.3-2.3.7	December 6, 2010	9-10	Discontinued
Honeycomb	3.0-3.2.6	February 22, 2011	11-13	Discontinued
Icre Cream Sandwich	4.0-4.0.4	October 18,2011	14-15	Discontinued
Jelly Bean	4.1-4.3.1	July 9, 2012	16-18	Discontinued
KitKat	4.4-4.4.4	October 31, 2013	19	Discontinued
Lollipop	5.0-5.1.1	November 12, 2014	21-22	Supported
Marshmallow	6.0-6.0.1	October 5, 2015	23	Supported
Nougat	7.0-7.1.1	August 22, 2016	24-26	Supported

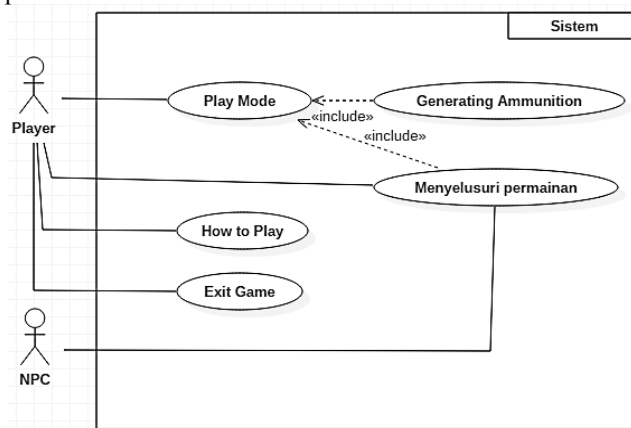
2.2.5 Kecerdasan Buatan

Kecerdasan Buatan (AI) adalah biasanya didefinisikan sebagai ilmu pengetahuan dan teknik meniru, memperluas dan menambah kecerdasan manusia melalui cara dan teknik buatan untuk membuat mesin cerdas. [9]. Kecerdasan buatan (AI) adalah bidang ilmu komputer yang melihat logika di balik kecerdasan manusia. Bidang ini mencari cara untuk memahami bagaimana kami berpikir dan menciptakan kembali kecerdasan ini dalam mesin. Karena sifatnya, AI meluas di seluruh aktivitas manusia, membuatnya relevan dengan cara yang berbeda untuk setiap industri. [10].

3. HASIL DAN PEMBAHASAN

3.1 Perancangan Unified Modeling Language (UML)

3.1.1 Use Case Aplikasi Game



Gambar 1. Use Case Aplikasi Game

Pada diagram *Use Case Aplikasi Game* akan dijelaskan proses-proses yang berjalan di dalam aplikasi *game*. *Actor* yang digunakan adalah *player* dan NPC. Berdasarkan *use case* yang ada pada Gambar 1. Diketahui aplikasi *game* ini memiliki tiga tombol utama yaitu tombol Play, How to Play dan Exit Game. Untuk memainkan *game*, *player* dapat memilih tombol Play. Setelah tombol Play dipilih, *player* dapat menelusuri permainan dan berinteraksi dengan NPC, selain itu proses *generating ammunition* yang menggunakan metode *Linear Congruential Generator* dilakukan secara berkala oleh sistem. Untuk mengetahui cara bermain, dengan memilih tombol How to Play pada menu utama, sistem akan mengarahkan *player* menuju panel How to Play yang berisi panduan dan petunjuk permainan. Exit Game merupakan salah satu tombol pada menu utama yang dapat dipilih untuk keluar dari permainan.

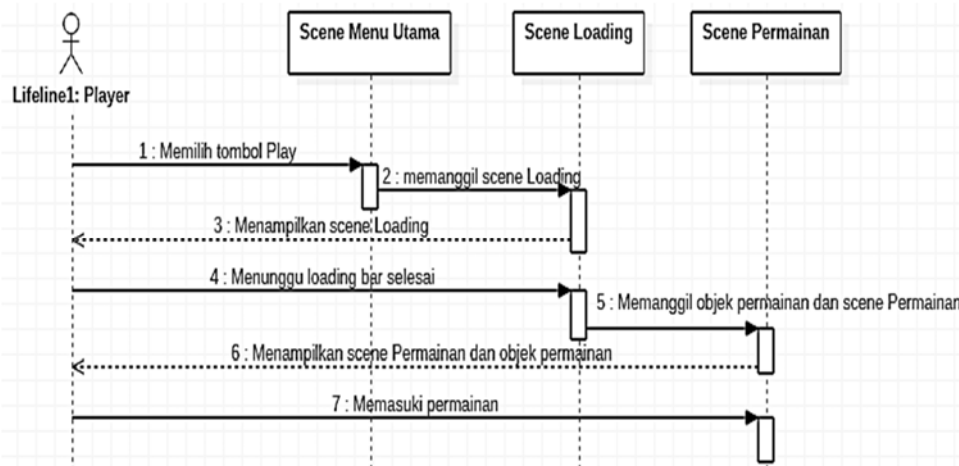
3.1.2 Diagram Sekuensial (Sequence Diagram)

Diagram sekuensial adalah diagram yang digunakan untuk menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Diagram sekuensial untuk perancangan *game* ini terdiri atas lima diagram sebagai berikut:

3.1.2.1 Sequence Diagram Play Mode

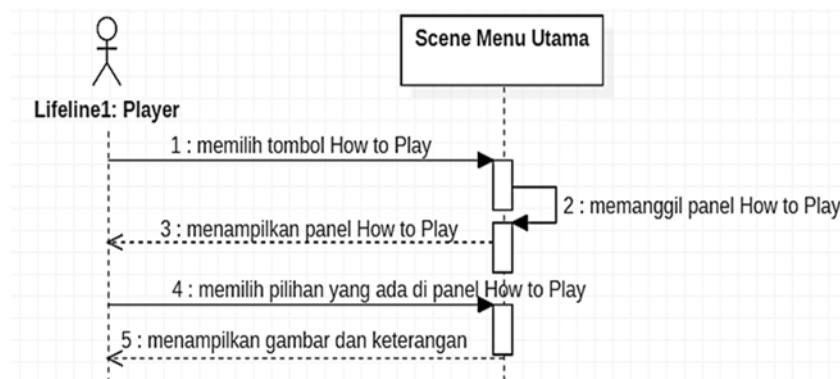
Diagram sekuensial yang ada pada Gambar 2 dapat dijelaskan sebagai berikut:

- Player* memilih menu *play mode* pada menu utama, kemudian sistem akan menampilkan *scene Loading* sebelum sistem menampilkan *scene Permainan*.
- Kemudian *player* harus menunggu *scene Loading* sampai selesai.
- Setelah *scene Loading* selesai, sistem akan memanggil objek-objek yang ada di dalam permainan serta memanggil *scene Permainan*, setelah itu barulah sistem akan menampilkan *scene Permainan* dan objek-objek permainan yang sebelumnya sudah dipanggil pada *player*.
- Setelah *scene Permainan* dan objek permainan sudah ditampilkan untuk *player*, *player* dapat memasuki permainan.



Gambar 2. Sequence Diagram Play Mode

3.1.2.2 Sequence Diagram How to Play



Gambar 3. Sequence Diagram How to Play

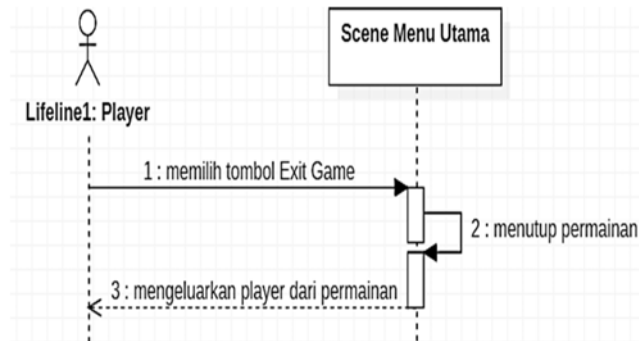
Diagram sekuensial yang ada pada Gambar 3, dapat dijelaskan sebagai berikut:

- Ketika *player* memilih pilihan tombol How to Play pada menu utama, sistem akan memanggil panel How to Play dan menampilkannya pada *player*.

b. Setelah panel How to Play muncul, *player* dapat memilih pilihan-pilihan yang ada dalam panel How to Play tersebut, ketika pilihan tersebut dipilih, gambar dan keterangan akan ditampilkan pada *player*.

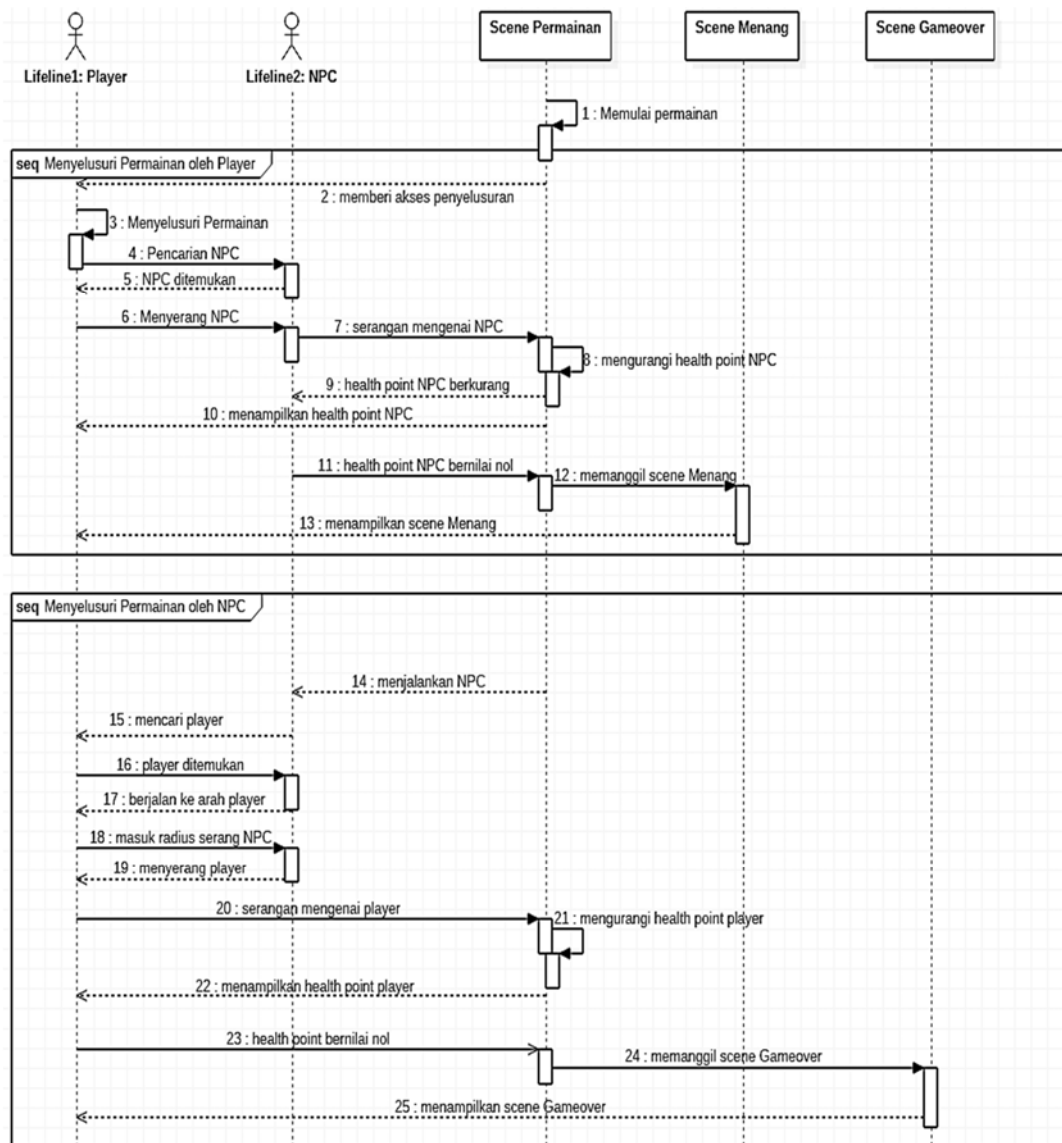
3.1.2.3 Sequence Diagram Exit Game

Diagram sekuensial yang ada pada Gambar 4 dapat dijelaskan bahwa ketika *player* memilih tombol Exit Game, maka sistem akan menutup permainan dan mengeluarkan *player* dari permainan.



Gambar 4. Sequence Diagram Exit Game

3.1.2.4 Sequence Diagram Menyelusuri Permainan

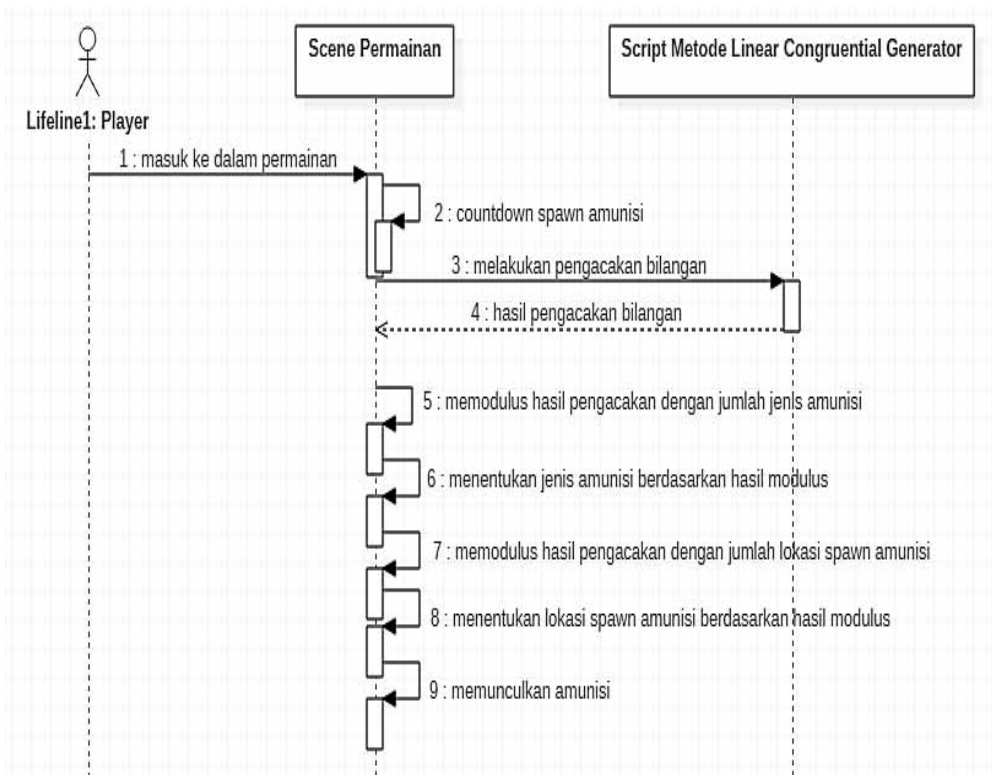


Gambar 5. Sequence Diagram Menyelusuri Permainan

Berdasarkan diagram sekuensial yang ada pada Gambar 5, setelah permainan dimulai, terdapat dua buah *sequence* yang dilakukan disaat yang bersamaan dan dapat dituliskan sebagai berikut:

- a. Sequence Diagram Menyelusuri Permainan oleh Player
 - 1) Permainan dimulai oleh sistem dan sistem akan memberikan akses penyelusuran agar *player* dapat menyelusuri permainan.
 - 2) *Player* memiliki misi untuk mengalahkan NPC untuk memenangkan permainan, oleh karena itu *player* diharuskan untuk mencari NPC.
 - 3) Setelah NPC ditemukan *player* dapat menyerang NPC tersebut, apabila serangan mengenai NPC maka sistem akan mengurangi *health point* NPC dan menampilkan *health point* yang dimiliki NPC kepada *player*.
 - 4) Apabila *health point* NPC habis atau bernilai nol, maka sistem akan secara otomatis memanggil *scene* Menang dan menampilkannya pada *player*.
- b. Sequence Diagram Menyelusuri Permainan oleh NPC
 - 1) Setelah permainan dimulai, sistem kemudian mengaktifkan NPC agar NPC dapat beroperasi di dalam permainan.
 - 2) NPC di dalam permainan ini memiliki misi untuk mengalahkan *player* untuk itu NPC harus dapat menemukan *player* dengan mencarinya terlebih dahulu.
 - 3) Setelah NPC menemukan *player*, maka NPC akan berjalan ke arah *player*. Apabila *player* berada dalam radius serang NPC maka NPC akan menyerang NPC tersebut.
 - 4) Ketika serangan dari NPC mengenai *player*, maka sistem akan mengurangi *health point* dari *player* dan menampilkan *health point* yang telah berkurang tersebut kepada *player*.
 - 5) Ketika *health point* yang dimiliki oleh *player* bernilai nol, maka sistem akan secara otomatis memanggil *scene* Gameover dan menampilkannya pada *player*.

3.1.2.5 Sequence Diagram Generating Ammunition



Gambar 6. Sequence Diagram Generating Ammunition

Diagram sekuensial yang ada pada Gambar 6 dapat dijelaskan sebagai berikut:

- a. Pertama-tama, ketika *player* sudah memasuki permainan, sistem akan melakukan *countdown* dalam permainan untuk memunculkan amunisi-amunisi yang dapat digunakan *player* untuk mengalahkan NPC.
- b. Setelah *countdown* terpenuhi, maka sistem akan melakukan pengacakan bilangan dengan metode *Linear Congruential Generator*. Hasil dari pengacakan bilangan tersebut akan dimodulus dengan jumlah jenis amunisi. Dari hasil modulus tersebut dapat ditentukan jenis amunisi yang akan dimunculkan.
- c. Hasil pengacakan bilangan yang dihasilkan juga akan digunakan untuk menentukan lokasi *spawn* amunisi, bilangan acak yang dihasilkan di modulus dengan jumlah lokasi *spawn*. Dari hasil modulus tersebut, dapat ditentukan dimana lokasi *spawn* amunisi akan dimunculkan.

d. Setelah jenis amunisi dan lokasi tempat *spawn* ditentukan, barulah amunisi dapat dimunculkan di dalam permainan.

3.2 Implementasi Algoritma Pada Permainan

3.2.1 Gambaran Algoritma Pada Permainan

Dalam perancangan permainan 3D Czel ini penulis menggunakan metode *Linear Congruential Generator* sebagai dasar dalam membuat permainan ini sehingga permainan dapat menjadi lebih baik dan menarik untuk dimainkan oleh pengguna. Metode *Linear Congruential Generator* ini sendiri merupakan salah satu cara untuk menghasilkan bilangan acak. Penulis menggunakan Metode *Linear Congruential Generator* ini karena dalam implementasinya metode ini lebih cepat dan praktis. Metode *Linear Congruential Generator* dalam penelitian ini digunakan sebagai cara untuk mengacak amunisi dan lokasi munculnya amunisi-amunisi yang sudah ditentukan tempatnya. Metode ini memiliki rumus sebagai berikut:

$$Z_i = (a \cdot Z_{i-1} + c) \bmod m$$

Dimana :

Z_i = bilangan acak ke – i

Z_{i-1} = bilangan acak sebelum nya

a = faktor pengali

c = *increment*

m = *modulus*

Syarat-syarat dalam menentukan LCG adalah sebagai berikut:

- a harus lebih besar dari \sqrt{m}
- Nilai a, c, Z_0 tidak boleh lebih dari m dan kurang dari nol
- Untuk c harus relatif prima dengan m
- a - 1 dapat dibagi dengan semua faktor prima dari m

Penulis menggunakan metode *Linear Congruential Generator* ini berdasarkan syarat-syarat di atas sehingga menghasilkan deretan bilangan acak yang baik. Deretan bilangan acak dapat dikatakan baik apabila terjadi pengulangan setelah kesekian kali, misalnya setelah 15 kali muncul bilangan acak yang berbeda kemudian terjadi pengulangan pada pembangkitan bilangan acak ke 16.

Untuk pengujiannya, digunakan bilangan di antara nilai 0 sampai 50. Nilai m yang digunakan adalah 50, besar nilai modulus mempengaruhi variasi munculnya angka yang berbeda sehingga penulis menggunakan nilai terbesar dari proses pengujian, yaitu 50. Untuk nilai c, harus relatif prima dengan 50. Dilakukan proses pencarian bilangan yang relatif prima dan didapatkan beberapa angka yang kemudian dipilih secara acak. Nilai yang relatif prima dengan 50 yang dipilih adalah 7, sehingga nilai c adalah 7. Nilai a yang digunakan harus lebih besar dari \sqrt{m} yaitu 7,07. Nilai a - 1 juga harus dapat dibagi dengan semua faktor prima dari m. Faktor prima dari m adalah 1, 2, dan 5. Nilai 10 dapat dibagi dengan semua faktor prima dari m, sehingga didapatkan nilai a adalah 11 apabila a - 1 = 10.

Berikut merupakan contoh dari hasil pengacakan dengan menggunakan kunci a, c, dan Z_0 serta m berdasarkan nilai yang sudah ditentukan sebelumnya sehingga dapat menghasilkan deretan bilangan acak yang baik karena mengikuti syarat-syarat yang sudah ditentukan:

$$Z_i = (a \cdot Z_{i-1} + c) \bmod m$$

$$a = 11$$

$$c = 7$$

$$m = 50$$

Z_0 yang digunakan untuk mengawali pengujian adalah 1.

Penyelesaian :

$$Z(1) = (11 \cdot 1 + 7) \bmod 50 = 18$$

$$Z(2) = (11 \cdot 18 + 7) \bmod 50 = 5$$

$$Z(3) = (11 \cdot 5 + 7) \bmod 50 = 12$$

$$Z(4) = (11 \cdot 12 + 7) \bmod 50 = 39$$

$$Z(5) = (11 \cdot 39 + 7) \bmod 50 = 36$$

$$Z(6) = (11 \cdot 36 + 7) \bmod 50 = 3$$

$$Z(7) = (11 \cdot 3 + 7) \bmod 50 = 40$$

$$Z(8) = (11 \cdot 40 + 7) \bmod 50 = 47$$

$$Z(9) = (11 \cdot 47 + 7) \bmod 50 = 24$$

$$Z(10) = (11 \cdot 24 + 7) \bmod 50 = 21$$

$$Z(11) = (11 \cdot 21 + 7) \bmod 50 = 38$$

$$Z(12) = (11 \cdot 38 + 7) \bmod 50 = 25$$

$$Z(13) = (11 \cdot 25 + 7) \bmod 50 = 32$$

$$Z(14) = (11 \cdot 32 + 7) \bmod 50 = 9$$

$$Z(15) = (11 \cdot 9 + 7) \bmod 50 = 6$$

$$Z(16) = (11 \cdot 6 + 7) \bmod 50 = 23$$

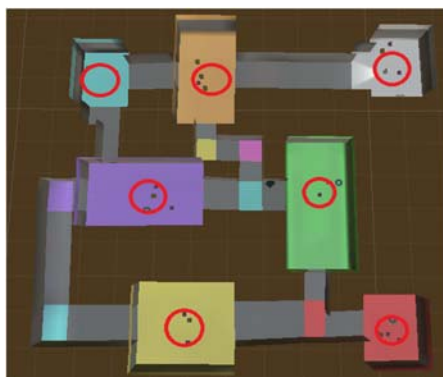
$$Z(17) = (11 \cdot 23 + 7) \bmod 50 = 10$$

$$Z(18) = (11 \cdot 10 + 7) \bmod 50 = 17$$

$Z(19) = (11 * 17 + 7) \bmod 50 = 44$
 $Z(20) = (11 * 44 + 7) \bmod 50 = 41$
 $Z(21) = (11 * 41 + 7) \bmod 50 = 8$
 $Z(22) = (11 * 8 + 7) \bmod 50 = 45$
 $Z(23) = (11 * 45 + 7) \bmod 50 = 2$
 $Z(24) = (11 * 2 + 7) \bmod 50 = 29$
 $Z(25) = (11 * 29 + 7) \bmod 50 = 26$
 $Z(26) = (11 * 26 + 7) \bmod 50 = 43$
 $Z(27) = (11 * 43 + 7) \bmod 50 = 30$
 $Z(28) = (11 * 30 + 7) \bmod 50 = 37$
 $Z(29) = (11 * 37 + 7) \bmod 50 = 14$
 $Z(30) = (11 * 14 + 7) \bmod 50 = 11$
 $Z(31) = (11 * 11 + 7) \bmod 50 = 28$
 $Z(32) = (11 * 28 + 7) \bmod 50 = 15$
 $Z(33) = (11 * 15 + 7) \bmod 50 = 22$
 $Z(34) = (11 * 22 + 7) \bmod 50 = 49$
 $Z(35) = (11 * 49 + 7) \bmod 50 = 46$
 $Z(36) = (11 * 46 + 7) \bmod 50 = 13$
 $Z(37) = (11 * 13 + 7) \bmod 50 = 0$
 $Z(38) = (11 * 0 + 7) \bmod 50 = 7$
 $Z(39) = (11 * 7 + 7) \bmod 50 = 34$
 $Z(40) = (11 * 34 + 7) \bmod 50 = 31$
 $Z(41) = (11 * 31 + 7) \bmod 50 = 48$
 $Z(42) = (11 * 48 + 7) \bmod 50 = 35$
 $Z(43) = (11 * 35 + 7) \bmod 50 = 42$
 $Z(44) = (11 * 42 + 7) \bmod 50 = 19$
 $Z(45) = (11 * 19 + 7) \bmod 50 = 16$
 $Z(46) = (11 * 16 + 7) \bmod 50 = 33$
 $Z(47) = (11 * 33 + 7) \bmod 50 = 20$
 $Z(48) = (11 * 20 + 7) \bmod 50 = 27$
 $Z(49) = (11 * 27 + 7) \bmod 50 = 4$
 $Z(50) = (11 * 4 + 7) \bmod 50 = 1$
 $Z(51) = (11 * 1 + 7) \bmod 50 = 18$
 $Z(52) = (11 * 18 + 7) \bmod 50 = 5$
 $Z(53) = (11 * 5 + 7) \bmod 50 = 12$
 $Z(54) = (11 * 12 + 7) \bmod 50 = 39$
 $Z(55) = (11 * 39 + 7) \bmod 50 = 36$

Berdasarkan pengujian pengacakan bilangan yang sudah dilakukan, terjadi pengulangan pada pengacakan bilangan ke-51. Setelah bilangan acak dihasilkan, bilangan tersebut kemudian dimodulus dengan jumlah dari jenis amunisi yang tersedia. Dari hasil modulus tersebut didapatkan bilangan yang mewakili *array* jenis amunisi yang akan dimunculkan. Hasil bilangan acak yang dihasilkan juga digunakan untuk menentukan lokasi jatuhnya amunisi. Caranya kurang lebih hampir menyerupai pengacakan jenis amunisi sebelumnya, namun tentunya jumlah lokasi munculnya amunisi yang sudah dirancang dapat berbeda dengan jumlah jenis amunisi, sehingga bedanya ada pada jumlah dari lokasi munculnya amunisi. Setelah dilakukan proses modulus pada jumlah lokasi jatuhnya amunisi, didapatkan nilai *array* yang berfungsi untuk menentukan lokasi munculnya amunisi. Selanjutnya, barulah jenis amunisi tersebut dapat dimunculkan di dalam *Dungeon* permainan ini.

3.2.2 Hasil Implementasi Metode Linear Congruential Generator Pada Game 3D Czel Berbasis Android



Gambar 7. Tampilan Dungeon dan Lokasi Spawn Amunisi

Pada tujuh titik yang dapat dilihat pada Gambar 7, di titik-titik tersebutlah dimana amunisi dimunculkan. Amunisi-amunisi akan dimunculkan secara acak dan di titik yang acak. Tampilan pemunculan amunisi pada *scene* Permainan dapat dilihat pada Gambar 8.



Gambar 8. Tampilan Munculnya Amunisi Pada Scene Permainan

Setiap jeda waktu yang diberikan, amunisi akan terus dimunculkan sampai permainan selesai. Pertama-tama akan dijalankan *function* LCG() yang berfungsi untuk mengacak suatu bilangan. Dalam pengacakannya dilakukan proses pencarian nilai-nilai variabel yang berdasarkan persyaratan-persyaratan mengenai metode tersebut. Nilai-nilai yang digunakan adalah $a = 11$, $Z_0 = 1$, $c = 7$, dan $m = 50$.

Selanjutnya, hasil pengacakan tersebut akan digunakan untuk menentukan jenis amunisi dan lokasi *spawn* amunisi dengan memodulus hasil pengacakan dengan jumlah jenis amunisi dan lokasi jenis amunisi. Jenis amunisi yang digunakan ada empat macam sehingga jenis amunisi ditentukan dengan memodulus hasil pengacakan dengan nilai 4, sedangkan lokasi jatuhnya amunisi ada tujuh titik sehingga hasil pengacakan harus dimodulus dengan nilai 7. Berikut merupakan gambaran implementasi metode pada aplikasi:

- Dilakukan pencarian jenis amunisi. $Z(1) = 18$, maka $18 \bmod 4$. Hasil modulus yang didapatkan adalah 2, kemudian nilai tersebut dimasukkan ke dalam *array* jenis amunisi dengan nilai 1 (`ItemList[2]._itemObject`)
- Selanjutnya dilakukan pencarian lokasi jatuhnya amunisi. $Z(2) = 5$, maka $5 \bmod 7$. Hasil modulus yang didapatkan adalah 5, kemudian nilai tersebut dimasukkan ke dalam *array* lokasi jatuhnya amunisi dengan nilai 3 (`_spawnLocation[5].position`)
- Setelah didapatkan jenis amunisi dan lokasi jatuhnya amunisi, maka dilakukan proses *instantiate* yaitu proses menciptakan objek dan memunculkan objek tersebut berdasarkan *input* yang dimasukkan. Berdasarkan input sebelumnya, jenis amunisi yang akan dimunculkan adalah `ItemList[2]._itemObject` pada titik `_spawnLocation[5].position`

Terdapat probabilitas kombinasi jenis amunisi dan lokasi jatuhnya amunisi sebelum bilangan acak yang muncul berulang. Deretan bilangan acak yang dihasilkan akan berulang pada pembangkitan bilangan acak ke-51 atau $Z(51)$. Berikut merupakan tabel yang menggambarkan probabilitas kombinasi jenis amunisi dan lokasi jatuhnya amunisi yang dapat muncul:

Tabel 2. Probabilitas Kombinasi Jenis Amunisi dan Lokasi Jatuhnya Amunisi

	ItemList[0]._itemObject	ItemList[1]._itemObject	ItemList[2]._itemObject	ItemList[3]._itemObject
<code>spawnLocation[0].position</code>	2	2	2	2
<code>spawnLocation[1].position</code>	2	2	1	2
<code>spawnLocation[2].position</code>	2	2	2	1
<code>spawnLocation[3].position</code>	1	2	2	2
<code>spawnLocation[4].position</code>	2	1	2	2
<code>spawnLocation[5].position</code>	2	2	1	2
<code>_spawnLocation[6].position</code>	2	2	2	1

Berdasarkan Tabel 2, dapat dilihat bahwa dari 28 kemungkinan kombinasi yang dapat terjadi, terdapat 2:50 atau 4% kemungkinan untuk beberapa kombinasi yang bernilai 2 pada tabel dan 1:50 atau 2% kemungkinan untuk kombinasi yang bernilai 1 pada tabel.

3.3 Tampilan Game Czel

Game 3D Czel terdiri atas beberapa *scene* dan beberapa panel yang dapat diuraikan sebagai berikut:

3.3.1 Tampilan Scene Menu Utama

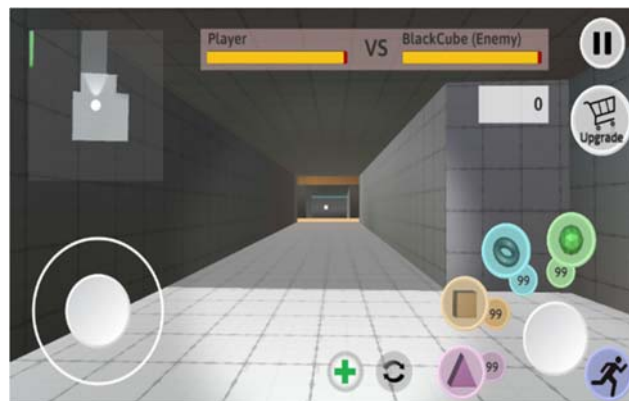
Tampilan *scene* Menu Utama dapat dilihat pada Gambar 9. Pada menu ini terdapat tiga tombol menu pilihan yang dapat dipilih *player*. Tombol-tombol tersebut adalah tombol Play, tombol How to Play, dan

tombol Exit Game. Pemain dapat memilih tombol Play untuk memulai permainan. Ketika tombol Play dipilih maka *player* akan diarahkan menuju *scene* Loading sebelum memasuki *scene* Permainan dan melakukan penelusuran permainan di dalam *Dungeon*. Pemain juga dapat memilih tombol How to Play pada *scene* Menu Utama ini untuk mengetahui cara bermain *game* Czel ini. Panel How to Play akan muncul ketika tombol How to Play dipilih. Untuk keluar dari permainan, *player* dapat memilih tombol Exit Game.



Gambar 9. Tampilan Scene Menu Utama

3.3.2 Tampilan Scene Permainan



Gambar 10. Tampilan Scene Permainan

Tampilan *scene* Permainan ditujukan pada Gambar 10. Pada *scene* Permainan ini berisikan *Dungeon* yang merupakan tempat dimana *player* dapat melakukan penelusuran dan objek-objek amunisi yang berjatuhan yang dapat digunakan *player* untuk mengalahkan NPC. NPC juga dimunculkan di area yang sudah ditentukan. Terdapat beberapa komponen yang tampak pada layar permainan seperti Analog, Minimap, dan banyak lagi yang dapat dimanfaatkan *player* untuk menyelusuri *Dungeon* dan menyelesaikan permainan.

3.3.3 Tampilan Panel How to Play

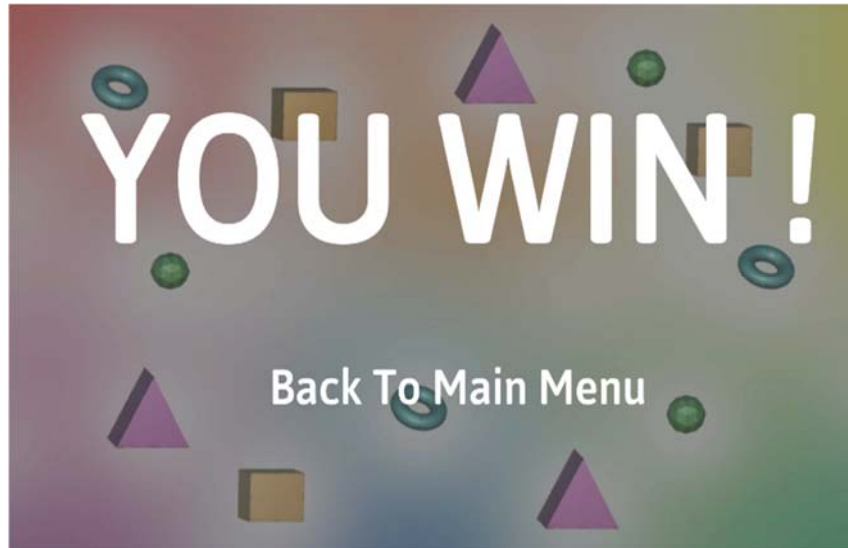


Gambar 11. Tampilan Panel How to Play

Gambaran panel How to play dapat dilihat pada Gambar 11. Pada panel ini, *player* dapat memilih tombol yang ada pada sebelah kiri layar untuk melihat bantuan yang tersedia menurut kategorinya. Saat tombol tersebut disentuh, tombol Navigasi akan muncul. Apabila hanya tersedia satu buah bantuan maka tombol Navigasi tidak akan dimunculkan. Pemain dapat menggunakan tombol Back to Main Menu yang berada di sebelah kanan atas layar untuk keluar dari panel How to Play.

3.3.4 Tampilan Scene Menang

Tampilan *scene* Menang dapat dilihat pada Gambar 12. Untuk memenangkan permainan, *player* harus mengalahkan NPC. NPC dinyatakan kalah apabila jumlah *health point* yang dimiliki oleh NPC bernilai nol. *Scene* Menang hanya dimunculkan ketika kondisi *player* memenangkan permainan terjadi.



Gambar 12. Tampilan Scene Menang

4. KESIMPULAN

Berdasarkan hasil implementasi pada perancangan *game* 3D Czel yang memiliki basis Android dan pembahasan pada bab-bab sebelumnya, maka penulis dapat menyimpulkan bahwa:

- Aplikasi *game* 3D Czel yang dirancang adalah sebuah permainan yang ditujukan untuk pengguna *smartphone* Android karena *game* tersebut memiliki basis Android.
- Game* 3D Czel ini memiliki tampilan dari sisi pandang orang pertama dan bersifat *offline*. Terdapat sebuah NPC yang berperan sebagai antagonis yang berfungsi untuk mengalahkan *player*.
- Pengacakan bilangan dengan metode *Linear Congruential Generator* (LCG) memiliki peran dalam pengacakan jenis amunisi dan lokasi *spawn* amunisi sehingga *player* dapat menyelesaikan permainan dengan mengambil amunisi-amunisi yang terjatuh dan mengalahkan NPC.
- Player* dapat memanfaatkan komponen-komponen yang ada pada *interface scene* Permainan seperti tombol-tombol untuk mempermudah penyelusuran dan membaca panduan di panel How to Play untuk mengetahui bagaimana cara bermain.
- Kondisi kemenangan dan kekalahan *player* bergantung pada jumlah *health point* yang dimiliki oleh *player* dan NPC.

5. SARAN

Penulis menyadari bahwa aplikasi *game* 3D Czel yang memiliki basis Android ini masih belum sempurna. Penulis memiliki harapan agar pembaca atau *programmer* lain dapat mengembangkan dan memperbaiki kekurangan-kekurangan yang ada pada aplikasi *game* ini. Oleh karena itu, agar penelitian ini dapat memberikan manfaat bagi pembaca, maka penulis akan memberikan beberapa saran, yaitu:

- Aplikasi *game* 3D Czel yang dirancang masih belum memiliki tampilan yang menarik. Terutama pada objek-objek yang digunakan dalam penelitian. Akan lebih baik bila objek-objek diperbaharui dengan model-model yang lebih realistis, misalnya seperti pada NPC, yang sekarang masih berbentuk dasar dapat dibuat atau diperbaharui dengan model *monster* atau model yang lebih realistis.
- Dengan menambahkan objek-objek *environment*, jumlah NPC dan *stage* serta penambahan tingkat kesulitan pada permainan, penulis berharap permainan dapat menjadi lebih menarik dan tidak membosankan.
- Penggunaan animasi pada tiap aksi yang terjadi pada permainan akan membuat permainan lebih interaktif.
- Menambahkan fitur baru seperti jenis-jenis *quest* dan teka-teki diperlukan agar permainan tidak monoton.

UCAPAN TERIMAKASIH

Penulis mengucapkan banyak terima kasih kepada ketua STIMIK Widya Dharma Pontianak, dosen pembimbing, keluarga, sahabat, dan semua pihak yang tidak dapat penulis tulis satu per satu yang telah memberikan bimbingan, arahan, saran, dan kritik dalam proses menyelesaikan penelitian ini.

DAFTAR PUSTAKA

- [1] Schell, Jesse. (2015). *The Art of Game Design : A Book of Lenses, Second Edition*. Penerbit CRC Press. Newyork.
- [2] Adams, Ernest. (2014). *Fundamentals of Game Design*. New Riders. Iowa.
- [3] Dictionaries, *First-person shooter word origin*, <https://www.dictionary.com/browse/first-person-shooter>, 8 Maret 2019, 01:15 WIB
- [4] Oxford Dictionaries, *Definition of first-person shooter in US English*, https://en.oxforddictionaries.com/definition/us/first-person_shooter, 8 Maret 2019, 01:14 WIB
- [5] Tattar, P. N., S. Ramaiah, dan B.G. Manjunath. (2016). *A Course In Statistics with R*. Penerbit John Willey & Sons, Ltd. United Kingdom.
- [6] Arizqia, Muhammad Ganda dan Anang Aris Widodo. (Januari 2017). “*Rancang bangun Aplikasi Dengan Linear Congruent Method (LCM) Sebagai Pengacakan Soal*”. Jurnal of Information Technology and Computer Science. Vol 1, no.2: hal 1.
- [7] Jubilee Enterprise. (2015). *Mengenal Dasar-Dasar Pemrograman Android*. Penerbit PT Elex Media Komputindo. Jakarta.
- [8] Supardi, Yuniar. (2017). *Koleksi Program Tugas Akhir dan Skripsi dengan Android*. Penerbit PT Elex Media Komputindo. Jakarta.
- [9] Shi, Zhongzhi. (2011). *Advanced Artificial Intelligence*. Penerbit World Scientific. Singapore.
- [10] Luce, Leanne. (2019). *Artificial Intelligence for Fashion: How AI is Revolutionizing the Fashion Industry*. Penerbit Apress. United States of America.