

RANCANG BANGUN GAME EDUKASI FISIKA SMA PENGENALAN GERAK PARABOLIK

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Deddi Ega Bramantiya

NIM : 0910963008



INFORMATIKA / ILMU KOMPUTER

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2015

PENGESAHAN

RANCANG BANGUN GAME EDUKASI FISIKA SMA PENGENALAN GERAK
PARABOLIK

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Deddi Ega Bramantiya
NIM:0910963008

Skripsi ini telah diuji dan dinyatakan lulus pada
7 Januari 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Erig Muhammad Adam J., S.T., M.Kom
NIP:19850410 2012121001

Wibisono Sukmo Wardhono, S.T., M. T.
NIK:201008 820404 1 001

Mengetahui
Ketua Program Studi Informatika

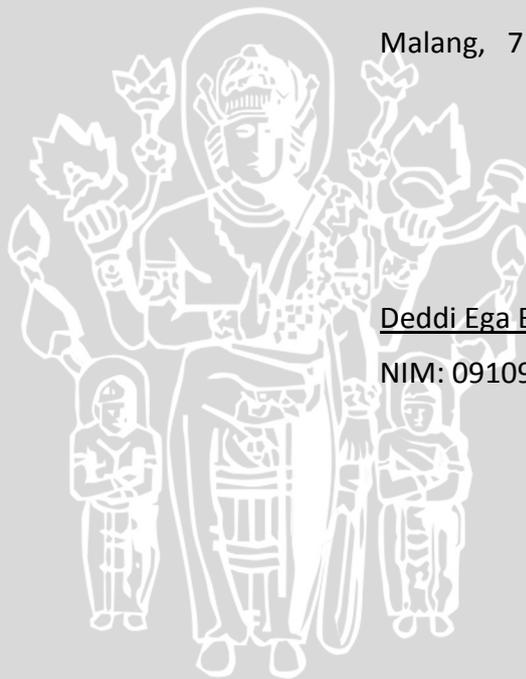
Marji, Drs., M.T.
NIP: 19670801 199203 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 7 Januari 2016



Deddi Ega Bramantiya

NIM: 0910963008

KATA PENGANTAR

Dengan nama Allah SWT Yang Maha Pengasih dan Maha Penyayang. Segala puji bagi Allah SWT karena atas rahmat dan hidayah-Na-lah penulis dapat menyelesaikan Tugas Akhir yang berjudul “Rancang Bangun Game Edukasi Fisika SMA Pengenalan Gerak Parabolik”. Shawatil serta salam atas junjungan besar kita Nabi Muhammad S.A.W. beserta keluarga dan para sahabat sekalian. Tugas Akhir ini disusun untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer di Program Studi Informatika Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya Malang. Melalui kesempatan ini, penulis menyampaikan rasa hormat dan terimakasih yang sebesar-besarnya kepada pihak yang telah memberikan bantuan lahir maupun batin selama penulisan tugas akhir ini. Oleh karena itu pada kesempatan ini penulis ingin menyampaikan rasa hormat dan termakasih penulis kepada :

1. Kepada orang tua penulis dan keluarga yang senantiasa tiada henti hentinya memberikan do'a demi terselesaikannya tugas akhir ini.
2. Bapak Ir. Sutrisno, M.T, Bapak Ir. Heru Nurwasito, M.Kom, Bapak Himawat Aryadita, S.T, M.Sc, dan Bapak Eddy Santoso, S.Kom selaku Ketua, Wakil Ketua 1, Wakil Ketua 2 dan Wakil Ketua 3 Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
3. Bapak Drs. Marji, MT dan Bapak Issa Arwani, S.Kom., M.Sc selaku Ketua dan Sekretaris Program Studi Informatika serta segenap Bapak / Ibu Dosen, Staff Administrasi dan Perpustakaan Program Studi Informatika Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
4. Bapak Eriq Muh. Adams Jonemaro, S.T., M.Kom selaku dan Bapak Wibisono Sukmo Wardhono selaku S.T., M.T. dosen pembimbing tugas akhir penulis.
5. Seluruh dosen Program Studi Informatika atas kesediaan membagi ilmunya kepada penulis.
6. Semua Asisten Ka. Lab serta Laboran dari Laboratorium Program Studi Informatika yang telah memberikan banyak bantuan dan dukungan dalam menyelesaikan tugas akhir ini.
7. Saudara Diti Ramashami serta Dhio Alfanda Fanani yang telah banyak mebantuk penulis dalam proses pengembangan *game* pletokan ini.
8. Teman-teman seperjuangan atas dukungan dan semangat yang diberikan.
9. Seluruh pihak yang telah membantu kelancaran dan penulisan tugas akhir yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa tugas akhir ini masih banyak kekurangan dan masih jauh dari sempurna. Untuk itu, saran dan kritik yang membangun, sangat penulis harapkan. Semoga tugas akhir ini membawa manfaat bagi penyusun maupun pihak lain yang menggunakannya.

Malang, 7 Januari 2016

Deddi Ega Bramantiya
prominense@gmail.com



ABSTRAK

Benda dikatakan bergerak jika benda tersebut mengalami perpindahan posisi terhadap benda lain sebagai acuan, baik mendekat maupun menjauh. Perubahan letak benda dilihat dengan membandingkan letak benda tersebut terhadap suatu titik yang dianggap tidak bergerak, sehingga gerak memiliki pengertian yang relatif . Gerak parabolik adalah gerak yang membentuk sudut tertentu pada gerak horizontal. Gerak ini merupakan perpaduan antara gerak lurus beraturan pada arah horizontal dengan gerak lurus berubah beraturan pada arah vertikal. Edukasi merupakan kegiatan pembelajaran pengetahuan dan keterampilan melalui proses tanya jawab , diskusi, penelitian atau pelatihan. Pada umumnya edukasi terjadi ketika adanya interaksi antara pengajar dengan bimbingannya, namun dengan menggunakan media yang ada sekarang seseorang dapat belajar secara otodidak. *Edutainment* atau edukasi melalui media *entertainment* merupakan proses pendidikan melalui media hiburan. Meskipun *edutainment* tidak terbatas pada *video games* dan mengacu pada semua bentuk edukasi yang juga bertujuan untuk menghibur, *edutainment* lebih banyak mengacu pada *video games* dengan tujuan *edukasi* yang jelas.



ABSTRACT

Moving Object is an object that experiencing displacement of position of using other object as a reference, either closer or away. Changes in the location of objects seen by comparing the location of the object to a point that was considered not moving, so that the motion has a relative definition. Parabolic motion is a motion that forms a certain angle to the horizontal motion. movement is a mix of rectilinear motion without acceleration in the horizontal direction with a rectilinear motion with constant acceleration in the vertical direction, Ignoring air friction so that object only influenced by the gravitation acceleration. Education is a learning activity through a process of discussion, research or training. In general, education occurs when the interaction between teachers with its students, but by using existing media one can learn by itself. Edutainment or education through entertainment media is the process of education through entertainment media. Although edutainment not limited to video games and refers to all forms of education also aims to entertain, edutainment mostly refer to video games with a clear educational purpose.



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan masalah	2
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Gerak.....	4
2.1.1 Gerak Lurus Beraturan	4
2.1.2 Gerak Lurus Berubah Beraturan	4
2.1.3 Gerak Jatuh Bebas.....	6
2.1.4 Gerak Parabolik	6
2.2 Edukasi	7
2.3 Game.....	8
2.3.1 Game Genre	8
2.3.2 Game Edukasi.....	9
2.4 UnityGame Engine	9
BAB 3 METODOLOGI	11
3.1 Metodologi Penelitian	11
3.1.1 Studi Literatur	11

3.1.2 Perancangan.....	12
3.1.3 Implementasi	13
3.1.4 Pengujian.....	14
3.2 Perancangan	14
3.2.1 Deskripsi <i>Game</i>	14
3.2.2 <i>Rapid Iterative Prototyping</i>	15
3.2.3 Perancangan Level	21
3.2.4 <i>Game Screen Flow</i>	24
3.2.5 Perancangan <i>Navigator</i>	29
3.2.6 Perancangan AI	31
BAB 4 Implementasi	33
4.1 Pemilihan Teknologi.....	33
4.1.1 Spesifikasi Perangkat Keras.....	33
4.1.2 Spesifikasi Perangkat Lunak.....	33
4.2 Batasan Implementasi	33
4.3 Implementasi Gameplay.....	34
4.3.1 Input Permainan.....	34
4.3.2 Implementasi Gerak Peluru	34
4.3.3 Implementasi Perhitungan Kecepatan.....	35
4.3.4 Implementasi Navigator.....	36
4.3.5 Implementasi AI	37
4.4 Implementasi Level.....	39
4.4.1 Implementasi Area 1	39
4.4.2 Implementasi Area 2	39
4.4.3 Implementasi Area 3	40
4.4.4 Implementasi Area 4	40
4.4.5 Implementasi Area 5	41
4.4.6 Implementasi Area 6	41
4.4.7 Implementasi Area 7	42
4.5 Implementasi Art	42
4.5.1 Implementasi Logo.....	42
4.5.2 Implementasi Karakter.....	43

4.5.3 Implementasi PowerUp	43
4.6 Implementasi User Interface	44
4.6.1 Implementasi Menu utama.....	44
4.6.2 Implementasi Pengaturan Karakter	44
4.6.3 Implementasi Tutorial.....	45
4.6.4 Implementasi Pengaturan Area	45
4.6.5 Implementasi Pengaturan tim	46
4.6.6 Implementasi HUD <i>Game</i>	46
4.6.7 Implementasi Pengaturan 1.....	49
4.6.8 Implementasi Pengaturan 2.....	49
BAB 5 Pengujian dan Evaluasi	50
5.1 Pengujian White Box.....	50
5.2 Pengujian Black Box	55
5.2.1 Pengujian <i>Functionality</i>	55
5.2.2 <i>Combinatorial Testing</i>	55
5.2.3 TFD	63
5.3 Analisa Pengujian.....	70
BAB 6 Penutup	71
6.1 Kesimpulan.....	71
6.2 Saran	71
DAFTAR PUSTAKA.....	72
LAMPIRAN A How to Play.....	73



DAFTAR TABEL

Tabel 2.1 Percepatan Gravitasi di Berbagai Lokasi (Sumarno, 2009)	6
Tabel 3.1 Implementasi Pembuatan Game.....	13
Tabel 4.1 Spesifikasi Perangkat Keras (laptop)	33
Tabel 4.2 Spesifikasi Perangkat Lunak (laptop).....	33
Tabel 4.3 <i>Source code</i> gerak peluru	34
Tabel 4.4 <i>Source code</i> perhitungan kecepatan	36
Tabel 4.5 <i>Source code Navigator</i>	36
Tabel 4.6 <i>Source code</i> pencarian target.....	38
Tabel 4.7 <i>Source code</i> perhitungan jarak.....	38
Tabel 4.8 <i>Source code</i> pengurutan prioritas	38
Tabel 5.1 <i>Source code</i> gerak peluru	50
Tabel 5.2 Kasus uji untuk pergerakan peluru.....	50
Tabel 5.3 <i>Source code</i> perhitungan kecepatan.....	51
Tabel 5.4 Kasus uji untuk perhitungan kecepatan.....	52
Tabel 5.5 <i>Source code Navigator</i>	52
Tabel 5.6 Kasus Uji untuk <i>Navigator</i>	54
Tabel 5.7 Tabel Uji <i>Functionality</i>	55
Tabel 5.8 Tabel <i>Combinatorial Testing</i> pengaturan permainan 1 parameter	56
Tabel 5.9 Tabel <i>Combinatorial Testing</i> pengaturan permainan 2 parameter	56
Tabel 5.10 Tabel <i>Combinatorial Testing</i> pengaturan permainan 3 parameter	56
Tabel 5.11 Tabel <i>Combinatorial Testing</i> pengaturan permainan 4 parameter	56
Tabel 5.12 Tabel <i>Combinatorial Testing</i> pengaturan permainan 5 parameter	57
Tabel 5.13 Tabel hasil Uji <i>Combinatorial Testing</i> pengaturan permainan	57
Tabel 5.14 Tabel <i>Combinatorial Testing</i> pengaturan area 1 parameter.....	57
Tabel 5.15 Tabel <i>Combinatorial Testing</i> pengaturan area 2 parameter.....	57
Tabel 5.16 Tabel <i>Combinatorial Testing</i> pengaturan area 3 parameter.....	58
Tabel 5.17 Tabel Hasil Uji <i>Combinatorial Testing</i> pengaturan area.....	58
Tabel 5.18 Tabel <i>Combinatorial Testing</i> pengaturan karakter 1 parameter	59
Tabel 5.19 Tabel <i>Combinatorial Testing</i> pengaturan karakter 2 parameter	59
Tabel 5.20 Tabel <i>Combinatorial Testing</i> pengaturan karakter 3 parameter	60

Tabel 5.21 Tabel *Combinatorial Testing* pengaturan karakter 4 parameter 61

Tabel 5.22 Tabel Hasil Uji *Combinatorial Testing* pengaturan karakter 62

Tabel 5.23 Tabel *Data Dictionary* TFD Bermain *Game* 64

Tabel 5.24 Tabel kasus Uji Jalur 1 TFD Pergerakan Karakter 65

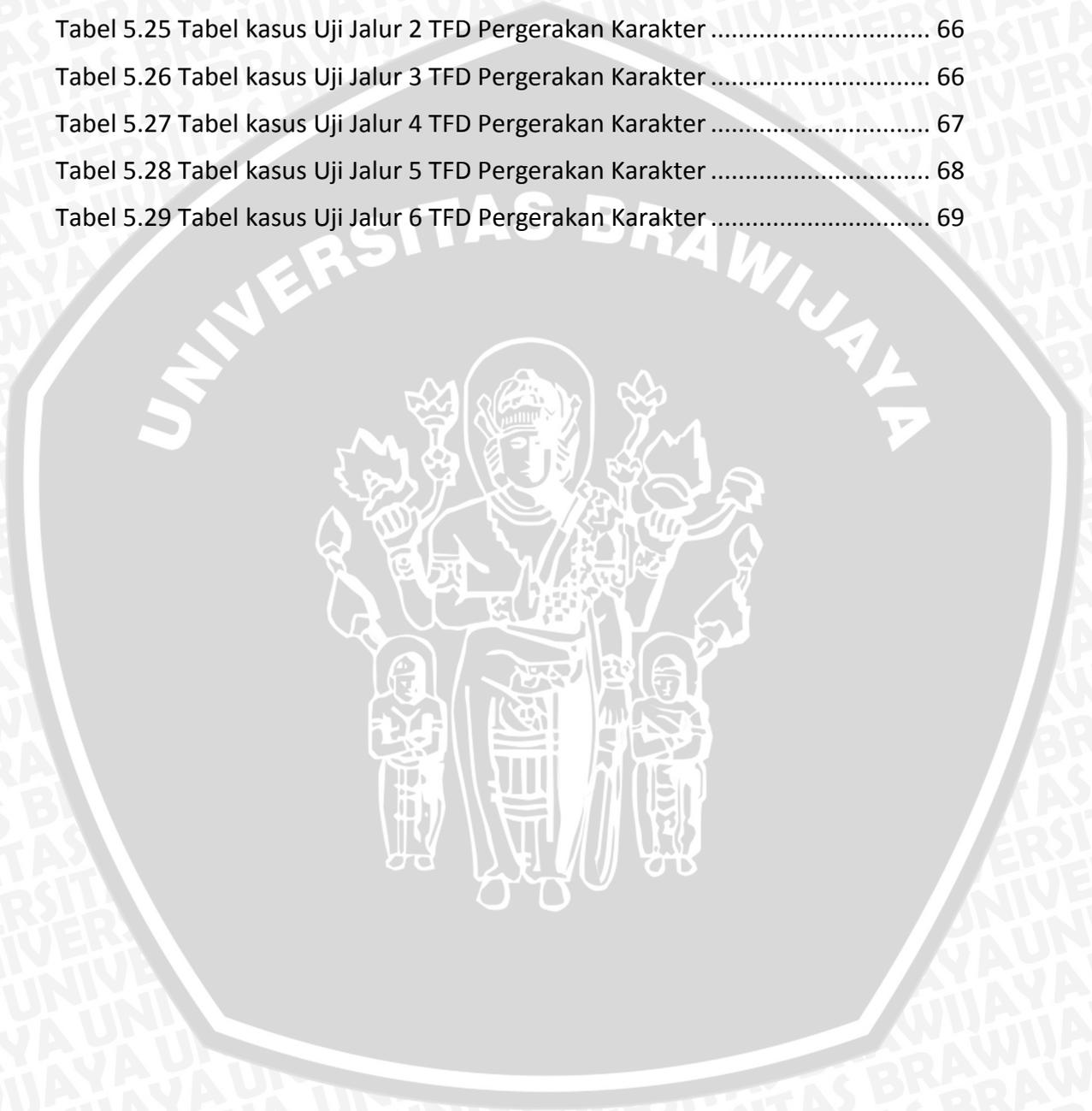
Tabel 5.25 Tabel kasus Uji Jalur 2 TFD Pergerakan Karakter 66

Tabel 5.26 Tabel kasus Uji Jalur 3 TFD Pergerakan Karakter 66

Tabel 5.27 Tabel kasus Uji Jalur 4 TFD Pergerakan Karakter 67

Tabel 5.28 Tabel kasus Uji Jalur 5 TFD Pergerakan Karakter 68

Tabel 5.29 Tabel kasus Uji Jalur 6 TFD Pergerakan Karakter 69



DAFTAR GAMBAR

Gambar 2.1 Lintasan Gerak Parabolik (Haryadi, 2009).....	6
Gambar 2.2 <i>User Interface</i> “War 3” oleh M. E. Lyon dan Brian West (1977) (Ahl, 1979)	9
Gambar 2.3 <i>User Interface</i> pada Unity	10
Gambar 3.1 Diagram Blok Metodologi	11
Gambar 3.2 <i>Paper Prototype</i> Iterasi 1	17
Gambar 3.3 Karakter dan <i>Terrain</i> pada Prototype	17
Gambar 3.4 Proses penembakan pada prototype.....	18
Gambar 3.5 <i>Paper Prototype</i> Iterasi 2	18
Gambar 3.6 <i>Paper Prototype</i> Iterasi 3	19
Gambar 3.7 Simbol dan Kartu <i>Powerup</i>	19
Gambar 3.8 Kartu Senjata dan Kartu Perlengkapan	20
Gambar 3.9 Kartu Event.....	21
Gambar 3.10 Konsep Area 1	21
Gambar 3.11 Konsep Area 2	22
Gambar 3.12 Konsep Area 3	22
Gambar 3.13 Konsep Area 4	23
Gambar 3.14 Konsep Area 5	23
Gambar 3.15 Konsep Area 6	24
Gambar 3.16 konsep Area 7.....	24
Gambar 3.17 <i>Game Screen Flow</i> Keseluruhan.....	25
Gambar 3.18 Konsep Tampilan Menu Utama.....	25
Gambar 3.19 Konsep Tampilan Pengaturan Karakter	26
Gambar 3.20 Konsep Tampilan Tutorial	26
Gambar 3.21 Konsep Tampilan Pengaturan Area.....	27
Gambar 3.22 Konsep Tampilan Pengaturan Tim	27
Gambar 3.23 Konsep Tampilan Permainan	28
Gambar 3.24 Konsep Tampilan Pengaturan 1	28
Gambar 3.25 Konsep Tampilan Pengaturan 2	29
Gambar 3.26 <i>Flowchart</i> Fitur <i>Navigator</i>	30

Gambar 3.27 <i>Flowchart</i> AI untuk karakter.....	32
Gambar 4.1 Input pada Keyboard.....	34
Gambar 4.2 Implementasi Navigator.....	36
Gambar 4.3 Implementasi Area 1.....	39
Gambar 4.4 Implementasi Area 2.....	40
Gambar 4.5 Implementasi Area 3.....	40
Gambar 4.6 Implementasi Area 4.....	41
Gambar 4.7 Implementasi Area 5.....	41
Gambar 4.8 Implementasi Area 6.....	42
Gambar 4.9 Implementasi Area 7.....	42
Gambar 4.10 Logo <i>Game</i> Pletokan.....	43
Gambar 4.11 Implementasi Karakter.....	43
Gambar 4.12 Implementasi Bagian dari Karakter.....	43
Gambar 4.13 Implementasi <i>Powerup</i>	44
Gambar 4.14 Implementasi Menu Utama.....	44
Gambar 4.15 Implementasi Pengaturan Karakter.....	45
Gambar 4.16 Implementasi Tutorial.....	45
Gambar 4.17 Implementasi Pengaturan Area.....	46
Gambar 4.18 Implementasi Pengaturan tim.....	46
Gambar 4.19 Implementasi <i>Game</i> HUD.....	47
Gambar 4.20 Implementasi Log HUD.....	47
Gambar 4.21 Implementasi Turn Order HUD.....	47
Gambar 4.22 Implementasi <i>Timer</i> HUD.....	48
Gambar 4.23 Implementasi Properti Karakter HUD.....	48
Gambar 4.24 Implementasi Avatar Karakter HUD.....	48
Gambar 4.25 Implementasi Target HUD.....	48
Gambar 4.26 Implementasi Pengaturan Permainan.....	49
Gambar 4.27 Implementasi Pengaturan Permainan 2.....	49
Gambar 5.1 <i>Flow graph</i> dari gerak peluru.....	50
Gambar 5.2 <i>Flow graph</i> dari perhitungan kecepatan.....	51
Gambar 5.3 <i>Flow graph</i> Navigator.....	53
Gambar 5.4 <i>Test Flow Diagram</i> Pergerakan Karakter.....	64

DAFTAR LAMPIRAN

LAMPIRAN A How to Play..... 73



BAB 1 PENDAHULUAN

1.1 Latar belakang

Fisika merupakan salah satu cabang dari ilmu pengetahuan alam (IPA) yang mempelajari gejala alam secara keseluruhan untuk dapat memahami apa yang mengendalikan atau menentukan gejala tersebut. Pada dasarnya fisika adalah ilmu dasar sehingga mampu dikaitkan dengan disiplin ilmu lain untuk membentuk ilmu baru, misalnya dengan ilmu astronomi akan membentuk ilmu astrofisika, dengan biologi akan membentuk biofisika, dengan ilmu kesehatan akan membentuk fisika medis, dengan ilmu bahan membentuk fisika material, dengan geologi membentuk geofisika, dan lain sebagainya. Pembelajaran fisika meliputi materi, energi, dan fenomena atau kejadian alam salah satu diantaranya yaitu gerak.

Fenomena gerak sering kita alami maupun kita saksikan, bahkan hampir setiap saat kita melakukan hal tersebut, mulai dari bangun kita hingga tidur kembali kita tidak pernah lepas dari gerak. Mobil bergerak, buah kelapa jatuh dari tangkainya, pesawat terbang di langit, merupakan contoh dari benda gerak pada kehidupan sehari-hari. Setiap benda bergerak pasti membentuk lintasan, benda jatuh bebas lintasan lurus kebawah, kincir angin membentuk lintasan melingkar, lemparan bola basket membentuk lintasan melengkung, lintasan melengkung inilah yang disebut dengan gerak parabolik. (Sumarno, 2009)

Di bangku SMA siswa menerima materi pembelajaran gerak parabolik. Materi diberikan melalui metode konvensional dimana seorang guru memberikan ceramah, diskusi, latihan soal, serta tugas agar siswa mampu memahami materi yang dipelajari. Namun pada kenyataannya, siswa memandang materi ini sebagai materi yang sulit sehingga keminatan siswa untuk mempelajari tersebut menjadi berkurang. Hal tersebut dapat diatasi dengan mengubah metode pembelajaran, salah satunya menggunakan *game* sebagai media edukasi. (Aeni, 2009)

Game edukasi merupakan metode pembelajaran dimana pemain menerima ilmu dengan pola *Learning by doing* sehingga pemain dituntut belajar untuk menyelesaikan permasalahan yang tersedia pada game tersebut. Beberapa kelebihan dari game edukasi yaitu merupakan visualisasi dari kehidupan yang nyata. Soal-soal materi disajikan sebagai kegiatan menyenangkan sehingga mereka tidak sadar bahwa mereka sedang belajar dan mampu menerima materi dengan lebih mudah. (Michael, 2006)

Dengan permasalahan di atas, muncul ide untuk membuat Rancang Bangun Game Edukasi SMA Pengenalan Gerak Parabolik yang diharapkan dapat membantu siswa dalam memahami tentang gerak parabolik. Dengan adanya game edukasi ini diharapkan mampu meningkatkan keminatan serta memberikan alternatif lain dalam sarana edukasi.

1.2 Rumusan masalah

Berdasarkan latar belakang diatas, dapat diambil rumusan masalah sebagai berikut :

1. Bagaimana kebutuhan rancang bangun *game* edukasi pengenalan gerak parabolik?
2. Bagaimana implementasi *game* edukasi materi gerak parabolik?
3. Bagaimana mengetahui fungsionalitas *game* edukasi pengenalan gerak parabolik?

1.3 Tujuan

Tujuan dari pembuatan aplikasi ini ialah :

1. Merancang *game* edukasi pengenalan gerak parabolik
2. Mengimplementasikan *game* edukasi berdasarkan materi gerak parabolik
3. Menguji *game* edukasi pengenalan gerak parabolik

1.4 Manfaat

1. Bagi Penulis
 - a. Mengaplikasikan ilmu yang didapat selama mengikuti perkuliahan di PTIIK.
 - b. Mendapat pemahaman mengenai perancangan dan pengembangan *game* menggunakan bantuan *game engine*.
2. Bagi Pengguna
 - a. Memberikan sarana hiburan serta memberikan pemahaman mengenai gerak parabolik
 - b. Memberikan sarana alternatif pembelajaran selain dengan membaca dan mengerjakan soal

1.5 Batasan masalah

Supaya pembahasan fokus pada tujuan yang diharapkan, maka diberi batasan permasalahan sebagai berikut :

1. Objek di dalam *game* merupakan objek 2 dimensi
2. *Game* dapat dimainkan oleh 1 maupun 2 pemain
3. *Game* tersedia secara *offline* atau tidak menggunakan koneksi internet
4. Sudut pandang pemain dalam *game* ini adalah sudut pandang orang ketiga (*third person*)
5. *Game* bergenre *Artillery*

1.6 Sistematika pembahasan

Berikut adalah sistematika penulisan yang digunakan dalam skripsi ini :

1. BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang masalah, rumusan masalah, tujuan pembuatan, manfaat, batasan permasalahan serta sistematika penulisan yang digunakan dalam penulisan skripsi ini.

2. BAB II LANDASAN KEPUSTAKAAN

Bab ini berisi tentang pembahasan konsep dasar serta teori yang dijadikan pedoman dalam perancangan serta pembuatan aplikasi

3. BAB III METODOLOGI DAN PERANCANGAN

Bab ini berisi tentang tahapan yang dilakukan pada saat membangun aplikasi, sekaligus membuat perancangan aplikasi.

4. BAB IV IMPLEMENTASI

Bab ini berisi mengenai implementasi dari hasil perancangan dan batasan implementasi

5. BAB V PENGUJIAN DAN EVALUASI

Bab ini berisi mengenai pengujian yang dilakukan pada *game* serta evaluasi dari hasil pengujian yang dilakukan

6. BAB VI PENUTUP

Bab ini berisi mengenai kesimpulan berdasarkan hasil pengujian dan analisis pada proses pembuatan beserta saran untuk pengembangan dan penyempurnaan aplikasi *game* edukasi dimasa mendatang

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Gerak

Gerak merupakan perubahan posisi suatu benda terhadap sebuah acuan tertentu. Benda dikatakan bergerak jika benda tersebut mengalami perpindahan posisi pada benda lain baik mendekat maupun menjauh. Perubahan letak benda dilihat dengan membandingkan letak benda tersebut terhadap suatu titik yang dianggap tidak bergerak, sehingga gerak memiliki pengertian yang relatif (Sumarno, 2009)

2.1.1 Gerak Lurus Beraturan

Suatu Benda dikatakan mengalami gerak lurus beraturan jika lintasan yang ditempuh oleh benda tersebut berupa garis lurus dan kecepatannya selalu tetap setiap saat. Sebuah benda yang bergerak lurus menempuh yang sama untuk selang yang sama pula. Secara matematis, persamaan gerak lurus beraturan adalah : (Sumarno, 2009)

$$S = v.t \quad (2.1)$$

Keterangan :

s = jarak yang ditempuh (m)

v = kecepatan benda (m/s)

t = waktu yang diperlukan (s)

2.1.2 Gerak Lurus Berubah Beraturan

Gerak lurus berubah beraturan didefinisikan sebagai gerak suatu benda pada lintasan lurus dengan kecepatan yang berubah secara teratur. Hal ini disebabkan karena adanya percepatan secara konstan yang dialami oleh benda tersebut. Secara sistematis, persamaan gerak lurus berubah dapat ditulis pada persamaan 2.2 : (Widodo, 2009)

$$v = v_0 + a.t \quad (2.2)$$

Keterangan :

v = kecepatan akhir(m)

v₀ = kecepatan awal benda (m/s)

t = waktu yang diperlukan (s)

a = percepatan yang dialami benda (m/s²)

selanjutnya, kita lihat bagaimana menghitung posisi benda setelah waktu t ketika benda mengalami percepatan konstanta. Dari definisi kecepatan rata-rata :

$$\bar{v} = \frac{x + x_0}{t}$$

Persamaan ini bisa kita tuliskan sebagai :

$$x = x_0 + \bar{v}t$$

Karena kecepatan bertambah secara beraturan, kecepatan rata-rata \bar{v} akan beraturan di tengah-tengah antara kecepatan awal dan kecepatan akhir, yang dirumuskan :

$$\bar{v} = \frac{v_0 + v}{2} \quad (2.3)$$

Keterangan :

\bar{v} = kecepatan rata-rata (m)

v_0 = kecepatan awal benda (m/s)

v = kecepatan akhir benda (m/s)

Dengan menggabungkan persamaan (2.2) dan (2.3) maka didapatkan

$$\begin{aligned} x &= x_0 + \bar{v}t \\ &= x_0 + \left(\frac{v_0 + v}{2}\right)t \\ &= x_0 + \left(\frac{v_0 + v_0 + at}{2}\right)t \\ &= x_0 + v_0 \cdot t + \frac{1}{2} a t^2 \end{aligned} \quad (2.4)$$

Keterangan :

\bar{v} = kecepatan rata-rata (m)

v_0 = kecepatan awal benda (m/s)

v = kecepatan akhir benda (m/s)

a = percepatan yang dialami benda (m/s²)

t = waktu (s)

x_0 = posisi awal benda (m)

x = posisi benda (m)

2.1.3 Gerak Jatuh Bebas

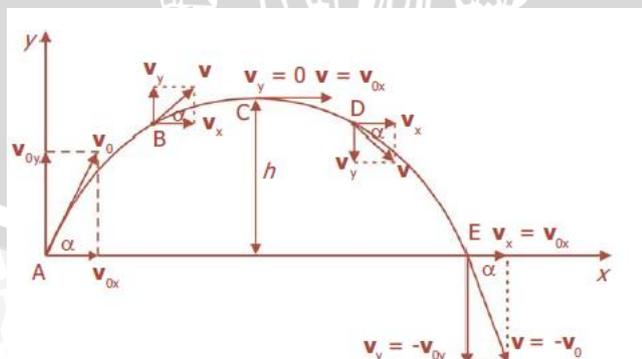
Gerak jatuh bebas merupakan contoh gerak lurus beraturan dimana benda mengalami percepatan secara konstan. Galileo menemukan bahwa semua benda akan jatuh dengan percepatan konstan yang sama jika tidak ada udara atau hambatan lainnya. Semua benda, berat atau ringan jatuh dengan percepatan yang sama jika tidak ada udara. Besar percepatan gravitasi sedikit bervariasi menurut garis lintang dan ketinggian. Tetapi variasi ini begitu kecil sehingga kita bisa mengabaikannya untuk sebagian besar kasus. Efek hambatan udara sering kali kecil, dan akan sering diabaikan. Tabel 2.1 menunjukkan besarnya percepatan gravitasi diberbagai tempat dengan ketinggian yang berbeda. (Sumarno, 2009)

Tabel 2.1 Percepatan Gravitasi di Berbagai Lokasi (Sumarno, 2009)

Lokasi	Ketinggian(m)	$g(m/s^2)$
New York	0	9,803
San Fransisco	100	9,800
Denver	1.650	9,796
Pikes Peak	4.300	9,789
Khatulistiwa	0	9,780
Kutub utara	0	9,832

2.1.4 Gerak Parabolik

Gerak parabolik adalah gerak yang membentuk sudut tertentu pada gerak horizontal. Gerak ini merupakan perpaduan antara gerak lurus beraturan pada arah horizontal dengan gerak lurus berubah beraturan pada arah vertikal. Gesekan terhadap udara diabaikan sehingga benda hanya dipengaruhi oleh percepatan gravitasi. Gambar 2.1 menunjukkan bentuk lintasan gerak parabolik (Haryadi, 2009)



Gambar 2.1 Lintasan Gerak Parabolik (Haryadi, 2009)

Berdasarkan gambar 2.1, garis horizontal merupakan sumbu x dan garis vertikal merupakan sumbu y sebagai titik acuan benda. Jika kecepatan awal benda adalah v_0 dengan sudut elevasi α , maka kecepatan awal peluru dapat diuraikan dalam komponen vertikal dan horizontal sebagai berikut :

$$v_{0x} = v_0 \cdot \cos \alpha$$

$$v_{0y} = v_0 \cdot \sin \alpha$$

Keterangan :

v_0 = kecepatan awal benda (m/s)

v_{0x} = kecepatan akhir benda pada sumbu x (m/s)

v_{0y} = kecepatan akhir benda pada sumbu y (m/s)

α = sudut yang dibentuk oleh benda terhadap bidang datar

Sedangkan, untuk posisi benda pada titik tertentu dapat dirumuskan pada persamaan 2.5 dan 2.6 :

$$x = v_{0x} \cdot t$$

$$x = v_0 \cos \alpha \cdot t \quad (2.5)$$

$$y = v_{0y} \cdot t - \frac{1}{2} g t^2$$

$$y = v_0 \sin \alpha \cdot t - \frac{1}{2} g t^2 \quad (2.6)$$

Keterangan :

x = posisi benda pada sumbu x (m)

y = posisi benda pada sumbu y (m)

t = waktu yang ditempuh (s)

v_0 = kecepatan awal benda (m/s)

v_{0x} = kecepatan akhir benda pada sumbu x (m/s)

v_{0y} = kecepatan akhir benda pada sumbu y (m/s)

α = sudut yang dibentuk oleh benda terhadap bidang datar

g = percepatan gravitasi (m/s^2)

2.2 Edukasi

Edukasi merupakan kegiatan pembelajaran pengetahuan dan keterampilan melalui proses tanya jawab, diskusi, penelitian atau pelatihan. Pada umumnya edukasi terjadi ketika adanya interaksi antara pengajar dengan bimbingannya, namun dengan menggunakan media yang ada sekarang seseorang dapat belajar secara otodidak. (Michael, 2006)

Edutainment atau edukasi melalui media *entertainment* merupakan istilah yang muncul pada tahun 1990 dengan adanya “*multi media*” komputer personal. Meskipun *edutainment* tidak terbatas pada *video games* dan mengacu pada semua bentuk edukasi yang juga bertujuan untuk menghibur, *edutainment* lebih banyak mengacu pada *video games* dengan tujuan *edukasi* yang jelas, terutama untuk taman kanak-kanak dan pengguna baru. (Michael, 2006)

2.3 Game

Game merupakan sebuah aktivitas yang memerlukan setidaknya satu orang yang berpartisipasi. Sebuah *game* memiliki kondisi kemenangan yang jelas, dan untuk menyempainya pemain dibatasi oleh beberapa aturan. Sementara itu, *video game* yang sudah marak pada saat ini merupakan permainan atau *game* yang dimainkan dengan media layar *video*. (Rogers, 2010)

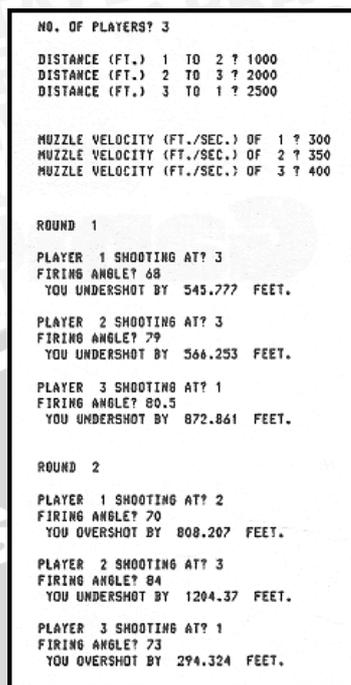
Game merupakan kegiatan yang sengaja dilakukan dengan memisahkan antara kehidupan nyata, membuat dunia imajinasi yang mungkin mempunyai hubungan dengan kehidupan nyata dan mampu menarik perhatian pemain seluruhnya. *Game* dimainkan dalam waktu dan tempat tertentu, sesuai dengan peraturan yang ada dan membentuk kelompok sosial berdasarkan pemainnya. (Michael, 2006)

Game mengikut sertakan pemain dengan melibatkan mereka dalam sebuah konflik yang terstruktur. Pemain diberi tantangan untuk menyelesaikan tujuan dengan mengikuti peraturan dan prosedur yang membuat *game* semakin sulit. Dalam *singleplayer game*, tantangan berasal dari sistem itu sendiri, namun pada *multiplayer game* tantangan berasal dari pemain lain sendiri atau bahkan keduanya. (Fullerton, 2008)

2.3.1 Game Genre

Genre merupakan sebuah pengklasifikasian dari suatu *game* berdasarkan tata cara bermain yang tersedia. *Game* yang akan dikembangkan merupakan *game* bergenre *Artillery*, dimana dua atau lebih pemain bergantian menembak satu sama lain dengan memperkirakan lintasan peluru berdasarkan perhitungan untuk dapat mengenai pemain lain. *Genre game* ini sering dideskripsikan layaknya *shooting game* karena berisi tentang kedua pemain yang saling menembak, walaupun sebenarnya genre ini lebih cocok diklasifikasikan sebagai *game* strategi. Penganalisaan jarak antar pemain, kecepatan lontaran peluru, serta pembedaan sudut antar pemain dan lontaran peluru merupakan kunci kemenangan pada *game* ini. Contoh lain dari *game* bergenre ini ialah “*worm*”, “*gunbound*”, “*scorched earth*” dan “*space tanks*”.

Artillery game untuk komputer dikembangkan pertama kali oleh Mike Forman dengan judul “*Artillery*” pada tahun 1976. Selanjutnya pada tahun 1977 M. E. Lyon dan Brian west memunculkan *game* bergenre sama dengan judul “*War 3*”. Kedua *game* ini masih berbasis teks sehingga segala tampilan, warna dan suara sangat bergantung pada imajinasi pemain. Gambar 2.3 menunjukkan *user interface* dari *game* “*War 3*”. (Ahl, 1979)



Gambar 2.2 User Interface “War 3” oleh M. E. Lyon dan Brian West (1977) (Ahl, 1979)

2.3.2 Game Edukasi

Game dengan konten pendidikan lebih dikenal dengan istilah *game* edukasi. *Game* ini bertujuan untuk memancing minat belajar anak dalam mempelajari materi dengan bermain, sehingga dengan perasaan senang yang dialami pemain diharapkan dapat mempermudah memahami materi yang disajikan. (Aeni, 2009)

2.4 UnityGame Engine

Game engine merupakan *software framework* yang didesain untuk pembuatan dan pengembangan *video games*, salah satunya adalah Unity. Unity merupakan *game engine multi-platform* yang didevelop oleh *Unity Technologies* dengan menargetkan berbagai platforms seperti windows, Mac, Linux, iOS, Andorid, Xbox 360, Wii U, PS3 maupun web-based. Fokus Unity terletak pada *game* 3D, namun pada pengembangannya Unity dapat bekerja dengan baik pada *game* 2D. (Hocking, 2015)

Antarmuka pada Unity dibagi menjadi beberapa bagian , antara lain : *scene*, *game*, *toolbar*, *hieararchy*, *inspector*, *project*, dan *console*. Setiap bagian memiliki fungsi yang berbeda dalam pembuatan *game*. Contoh tampilan antarmuka pada Unity dapat dilihat pada gambar 2.3.



Gambar 2.3 User Interface pada Unity

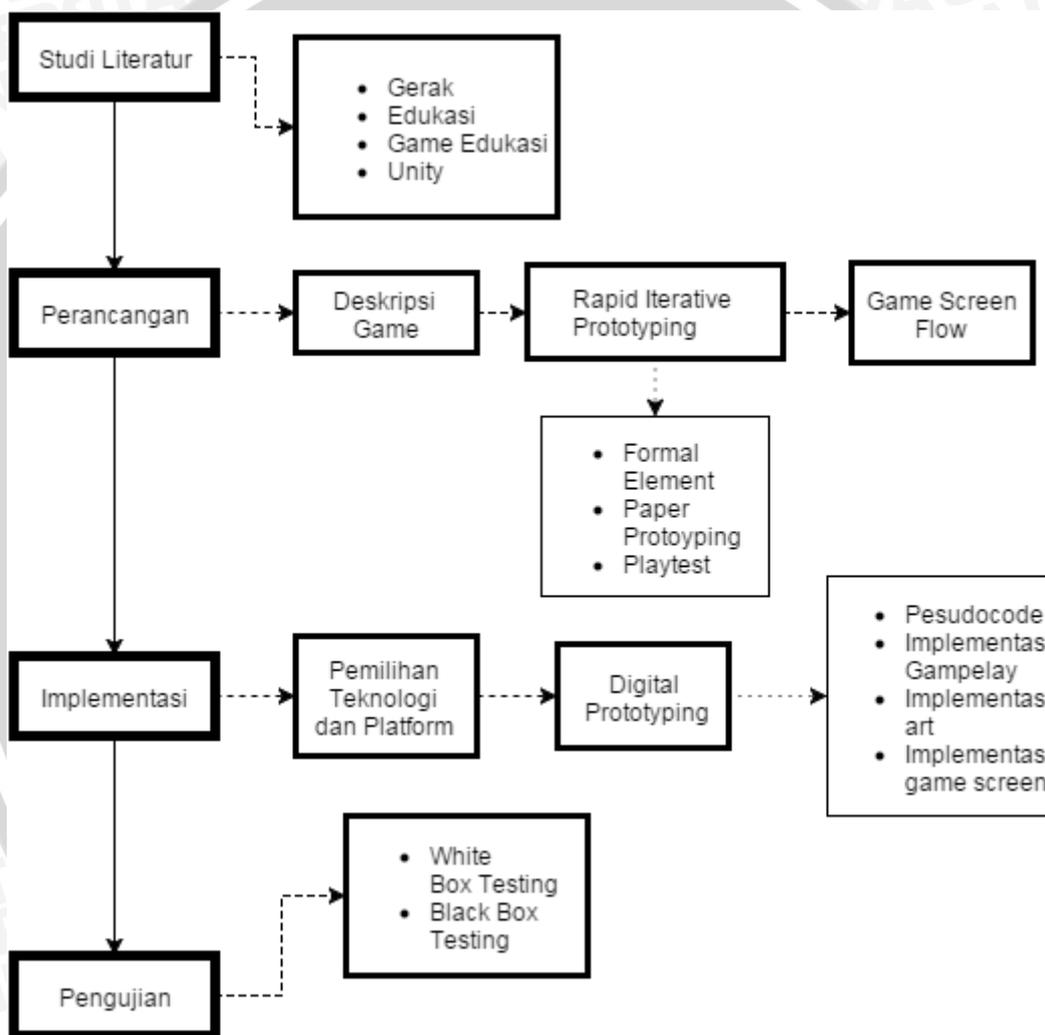
Adapun fungsi dari bagian-bagian pada Unity yaitu :

- Scene** : Tampilan ini digunakan untuk berinteraksi dengan objek pada *game* secara langsung. Dalam tampilan ini, pengguna dapat mengubah posisi, ukuran, rotasi serta skala dari objek tersebut.
- Game** : Tampilan ini digunakan untuk melihat tampilan serta menguji *game* yang sudah dibuat.
- Toolbar** : Tampilan yang memiliki *scene navigator* untuk menekan tombol *play*, *pause*, dan *stop* yang memiliki pengaruh langsung pada tampilan *game*. Pada tampilan ini juga terdapat menu untuk mengubah *layouts* dan *layers*.
- Hierarchy** : Tampilan yang berisi daftar objek yang terdapat pada *scene*. Pembuatan objek baru dapat dilakukan pada tampilan ini.
- Inspector** : Tampilan yang berisi komponen-komponen dan konfigurasi pada objek(dari tampilan *hierarchy* dan *scene*) maupun aset (dari tampilan *project*) yang sedang dipilih.
- Project** : Tampilan yang berisi daftar aset yang tersedia untuk digunakan dalam *game*. Penambahan aset dapat dilakukan pada tampilan ini.
- Console** : Tampilan yang berisi teks mengenai hal-hal yang terjadi pada *game*, baik pesan kesalahan, maupun peringatan yang ada. Pada tampilan ini dapat digunakan untuk memeriksa apakah *game* sudah berjalan dengan benar.

BAB 3 METODOLOGI

3.1 Metodologi Penelitian

Bab ini menjelaskan mengenai langkah-langkah yang dilakukan dalam perancangan, implementasi serta pengujian dari *game* yang akan dibuat. Metodologi yang diambil dalam pembuatan *game* ini terdiri dari studi literatur, perancangan, implementasi, dan pengujian dengan menggunakan metode *rapid Iterative Prototyping* sebagai metode perancangan. Gambar 2.1 menunjukkan diagram blok dari pembuatan *game*.



Gambar 3.1 Diagram Blok Metodologi

3.1.1 Studi Literatur

Studi literatur menjelaskan tentang referensi beserta teori dasar yang digunakan dalam pembuatan *game*. Pengambilan data berasal dari website serta jurnal yang tersedia. Literatur yang diambil meliputi :

1. Gerak
2. Edukasi
3. *Game*
4. Unity

3.1.2 Perancangan

Setelah melakukan pengumpulan data yang diperlukan dalam implementasi, dilakukan fase perancangan. Pada fase perancangan dilakukan tahapan-tahapan awal pengembangan *game* sehingga diharapkan dapat mempermudah dalam pengeimplementasian *game*. Secara teknis perancangan meliputi penentuan deskripsi *game*, *rapid iterative prototyping*, dan penentuan *Game Screen Flow*.

1. Deskripsi *game*

Penentuan judul *game*, topik, ESRB rating, *platform game*, serta *genre game*.

2. *Rapid Iterative Prototyping*

Metode pembuatan *game* dengan basis pembuatan *prototype* secara berulang-ulang sesuai dengan kebutuhan. Langkah ini terdiri dari penentuan formal element, *paper prototyping* dan *playtest*.

a. *Formal Element*

Pembentukan struktur *game* dengan menentukan *formal element* yang ada. *Players, objectives, procedures, rules, resources, conflict, boundaries, dan outcomes* merupakan *element* yang dibentuk pada tahap ini.

b. *Paper Prototyping*

Pembentukan *prototype* menggunakan media kertas dengan menfokuskan pada *gameplay* dari *game* itu sendiri. Perubahan komponen berdasarkan masukan dari pemain dapat dilakukan dengan mudah dan cepat menggunakan media kertas.

c. *Playtest*

Menguji *prototype* dari *game* yang sudah dirancang dengan dimainkan oleh beberapa *playtester* untuk mendapatkan saran dalam pengembangan *game*. *Prototype* diubah berdasarkan saran yang masuk dan dilakukan pengujian ulang. Perulangan dilakukan beberapa kali dan berhenti ketika *game* dirasa menarik.

3. *Game Screen Flow*

Perancangan tampilan yang digunakan pada *game* serta bagaimana hubungan antara tampilan.

3.1.3 Implementasi

Implementasi aplikasi ini dilakukan dengan menggunakan bahasa pemrograman C# dan menggunakan *game engine* Unity dan *mono develop* sebagai *Software Development Kit* (SDK) . Hasil dari implementasi berupa aplikasi yang siap dimainkan di PC. Implementasi meliputi pemilihan teknologi dan platform , pseudocode dan pembuatan digital prototyping.

1. Pemilihan Teknologi dan *Platform*

Menentukan teknologi yang digunakan serta penentuan *platform* dimana game dapat dimainkan.

2. *Pseudocode*

Pembuatan *pseudocode* mengenai komponen komponen penting pada game yang dirancang.

3. *Digital Prototyping*

Pengimplementasian hasil perancangan dan *pseudocode* menggunakan teknologi yang sudah ditentukan dengan pembuatan *digital prototype*. Pengimplementasian meliputi implementasi *gameplay*, implementasi level, implementasi art, dan implementasi *game screen*.

- a. Implementasi *Gameplay*

Mengimplementasi *gameplay* game sesuai hasil perancangan

- b. Implementasi Level

Mengimplementasikan desain level pada *game*

- c. Implementasi *Art*

Mengumpulkan asset baik membuat sendiri maupun mencari sumber gratis serta mengimplementasi pada *game*

- d. Implementasi *Game Screen*

Implementasi *screen flow* berdasarkan perancangan.

Pada tabel 3.1 akan dijelaskan lebih detail mengenai program yang digunakan untuk masing-masing implementasi dari program tersebut.

Tabel 3.1 Implementasi Pembuatan Game

Jenis Implementasi	Nama Program
Desain 2D	Adobe Photoshop
Game Engine	Unity 5.1.0f3
Software Development Kit	Mono Develop
Bahasa Pemrograman	C#

3.1.4 Pengujian

Membahas mengenai tahapan pengujian dan analisa *game* yang telah dibangun. Pengujian berfungsi untuk mengetahui seberapa banyak progress permainan berjalan serta mengetahui kesalah-kesalah yang ada pada permainan. Tahap pengujian pada *game* ini terdiri dari pengujian fungsionalitas, pengujian *blackbox*, dan pengujian *whitebox*.

1. Pengujian *Fungsionalitas*

Pengujian untuk memastikan apakah fitur pada *game* telah terimplementasi dengan baik.

2. Pengujian *Blackbox*

Pengujian tanpa memperhatikan cara kerja algoritma pada *source code*, namun lebih pada pengamatan hasil dari data uji serta *fungsional* dari perangkat lunak

3. Pengujian *Whitebox*

Pengujian yang didasarkan terhadap detail perancangan serta kode (*source code*) dari program serta menggunakan struktur control dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus uji.

3.2 Perancangan

Perancangan dari *game* berisikan tentang konsep *game* yang dikembangkan seperti apa yang tertera pada metodologi dimulai dari konsep *game*, judul dan sebagainya.

3.2.1 Deskripsi *Game*

Game ini dimainkan dapat dimainkan lebih dari satu pemain secara bergantian pada satu perangkat dengan menggunakan mouse dan keyboard sebagai input untuk menggerakkan karakter pada *game* ini.

1. Judul *Game*

Judul yang akan diambil pada *game* yang akan dikembangkan ini adalah pletokan, diambil dari permainan khas masyarakat betawi yang menggunakan senjata mainan terbuat dari bambu dan pelurunya terdiri dari kertas yang dibasahkan atau biji jambu. Permainan ini merupakan permainan saling tembak yang dapat dimainkan antar kelompok, sehingga pada *game* ini tiap pemain menggerakkan sekelompok karakter.

2. *Genre* dan Topik

Genre dari *game* ini adalah *game artillery*. *Game* bergenre ini merupakan *game* dimana pemain mengendalikan sekelompok karakter dan bergantian untuk mengembak satu sama lain dengan memperhitungkan jarak antar

karakter, ketinggian, sudut yang dibentuk peluru dan kecepatan tembakan. Topik pada game ini sendiri adalah strategi.

3. *Goal* (Tujuan)

Tujuan dari *game* Pletokan ini cukup sederhana, yaitu pemain harus mengalahkan pemain lain dengan mengalahkan semua karakter pemain lawan atau menghancurkan objek yang ada.

4. *Platform*

Game ini dibuat menggunakan *gameengine* unity yang mampu mengembangkan *game* untuk beberapa *platform* system operasi, namun pada pengembangan ini dikhususkan pada PC.

5. ESRB Rating

Game pletokan ini termasuk dalam kategori *Everyone* 10+ karena didalamnya tidak terdapat unsur sara maupun seksual namun terdapat sedikit unsur kekerasan sehingga *game* pletokan ini dapat dikatakan layak untuk semua pemain dengan umur lebih dari 10.

3.2.2 Rapid Iterative Prototyping

1. *Formal Element*

a. *Players*

Game didesain untuk dimainkan oleh 2 pemain dimana masing-masing pemain menggarakan sekelompok karakter untuk melawan kelompok lain. *Game* dapat dimainkan sendiri dengan AI sebagai lawannya.

b. *Objectives*

Tujuan dari *game* ini adalah menggunakan karakter yang tersisa untuk mengalahkan seluruh karakter lawan selama karakter pemain masih ada.

c. *Procedures*

Adapun *procedures* pada permainan ini, pertama-tama sistem menentukan karakter mana yang terlebih dahulu bergerak. Setiap giliran pemain menentukan karakter target kemudian menghitung berapa sudut dan kecepatan yang diperlukan berdasarkan jarak antar karakter tersebut. Giliran selesai ketika pemain menembakkan peluru berdasarkan sudut dan kecepatan yang telah dihitung sebelumnya. *Game* berakhir ketika karakter dalam salah satu kelompok telah habis.

d. *Rules*

Adapun peraturan dalam *game* ini, yaitu sudut yang dibentuk hanyalah sudut-sudut tertentu, karakter tidak dapat bergerak ketika

mencapai jarak tertentu, kecepatan peluru tidak bias melebihi kecepatan yang sudah ditentukan.

e. *Resources*

Resources merupakan asset yang digunakan oleh pemain untuk mencapai tujuan. Asset tersebut ialah *health*, peluru, *timer*, dan *powerup*. Ketika *health* karakter habis karakter tersebut tidak dapat digunakan, ketika peluru senjata habis maka senjata tersebut tidak dapat digunakan, ketika *timer* habis sebelum pemain menembak maka giliran selesai, ketika karakter mendapatkan *powerup* karakter tersebut mendapatkan bonus.

f. *Conflict*

Dengan adanya *powerup* dan karakter musuh, pemain harus menentukan prioritas antara *powerup* dan musuh, antara musuh yang dekat atau yang jauh, antara musuh dengan *health* sedikit atau musuh dengan *health* banyak, bergerak mendekati maupun menjauhi musuh.

g. *Boundaries*

Boundaries merupakan pembatas antara komponen yang masuk dalam game maupun yang tidak termasuk. Dalam *game* ini *boundaries* merupakan batas area masing-masing area yang dapat dilewati oleh pemain. Area diluar batas tidak dapat dilewati oleh pemain.

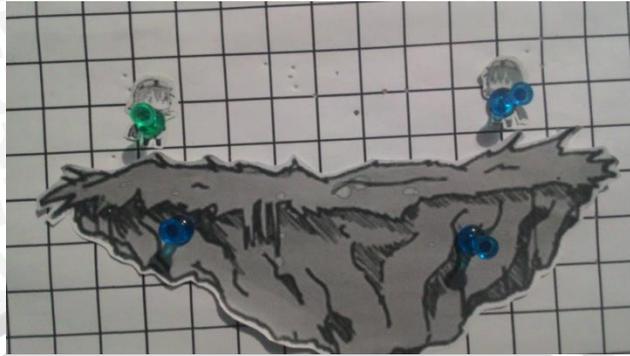
h. *Outcomes*

Outcomes atau hasil akhir pada game ini ada 3 macam, menang ketika karakter pemain masih tersisa dan karakter lawan tidak tersisa, kalah ketika karakter pemain sudah tidak tersisa dan karakter lawan masih tersisa dan seri ketika karakter pemain dan karakter lawan sama-sama tidak tersisa

2. *Paper Prototyping*

Pembuatan bentuk fisik dari *game* menggunakan media kertas untuk menguji *gameplay* dari *game* tersebut. Media kertas digunakan untuk mempermudah pengubahan dari *gameplay* tersebut. *Playtest* dilakukan untuk menguji *gameplay* tersebut untuk mencapai *gameplay* yang *fun* dan *balance*. Dalam pembuatan *game* pletokan, *playtest* dilakukan dengan beberapa *game* developer lain dan dilakukan berulang-ulang.

a. Iterasi 1



Gambar 3.2 Paper Prototype Iterasi 1

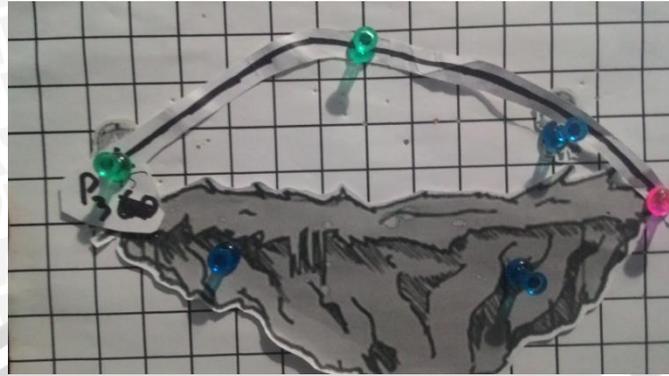
Rule :

- Pemain menggerakkan karakter secara bergantian
- Karakter dapat bergerak kemanapun selama dalam 1 terrain
- Perbedaan karakter dibedakan dengan warna pin. Bentuk *Terrain* dan karakter digambarkan pada gambar 3.3
- Karakter memiliki health point 100
- Game berakhir ketika karakter lawan habis



Gambar 3.3 Karakter dan *Terrain* pada Prototype

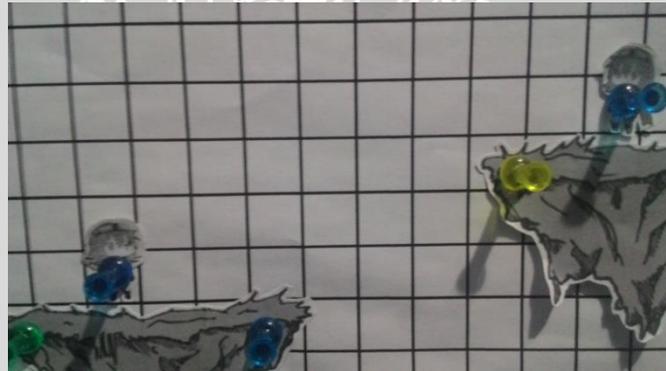
- Sebelum menembak, pemain dapat menggerakkan karakter
- Pemain menentukan sudut penembakan antara 30, 45 , dan 60. Sudut penembakan menentukan bentuk lintasan peluru.
- Pemain menentukan dadu untuk menentukan kecepatan awal dari peluru.
- Objek pertama yang mengenai Lintasan Peluru akan terkena damage, dan peluru berhenti di lokasi tersebut.
- Ketika perhitungan damage selesai, maka giliran pemain tersebut telah selesai.



Gambar 3.4 Proses penembakan pada prototype

Hasil Evaluasi *playtest* :

- Pergerakan pemain yang tidak terbatas sehingga sulit bagi pemain untuk menentukan target
 - Waktu yang tidak terbatas membuat perhitungan terlalu mudah
- b. Iterasi 2



Gambar 3.5 Paper Prototype Iterasi 2

Perubahan *rule* :

- Pemain memiliki batas waktu 60 detik untuk menggerakkan dan menembak, ketika waktu habis maka giliran pemain tersebut selesai.
- Pemain hanya dapat memindahkan karakter maksimal 2 kotak dari posisi awal.

Hasil Evaluasi *playtest* :

- Kurangnya opsi pemain dalam menggerakkan karakter.
- Tidak ada perbedaan karakteristik pada karakter
- Kurangnya variasi senjata

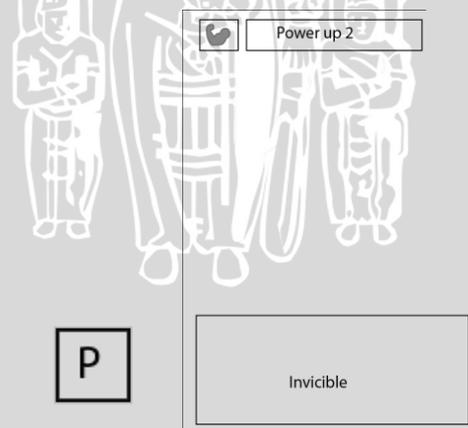
c. Iterasi 3



Gambar 3.6 Paper Prototype Iterasi 3

Perubahan *rule* :

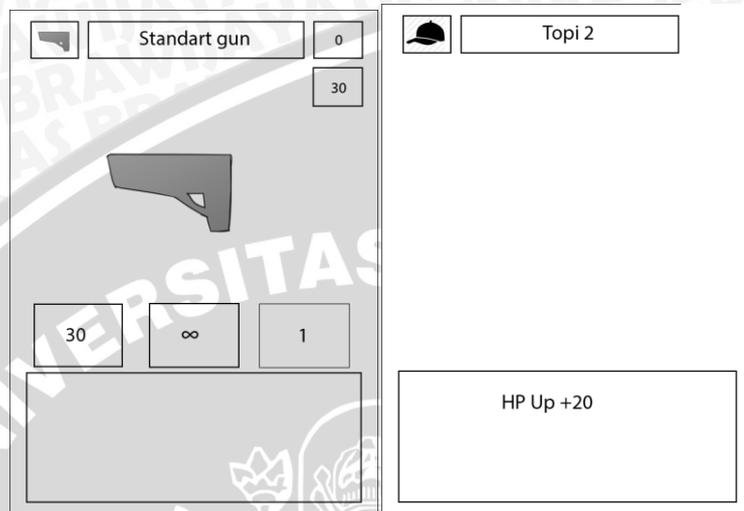
- Terdapat powerup yang tersebar di area permainan. Kartu dan simbol *powerup* digambarkan pada gambar 3.7.
- Powerup dapat diperoleh dengan bergerak ke posisi powerup tersebut atau dengan ditembak
- Powerup tidak menghentikan gerak peluru.
- Ketika pemain mendapatkan powerup, pemain tersebut mengambil 1 kartu dari tumpukan kartu powerup untuk mengetahui keterangan dari *powerup* yang didapat.



Gambar 3.7 Simbol dan Kartu Powerup

- Pada awal permainan pemain dapat memilih senjata dan perlengkapan untuk karakter.
- Senjata dan perlengkapan mempengaruhi karakteristik dari karakter. Contoh perlengkapan dan senjata digambarkan pada gambar 3.8

- Perlengkapan dibagi menjadi 4 bagian, kepala, badan tangan dan kaki.
- Masing-masing senjata memiliki tingkat kerusakan berbeda dan amunisi yang akan berkurang setiap kali dipakai.
- Senjata yang tidak memiliki amunisi tidak dapat digunakan.



Gambar 3.8 Kartu Senjata dan Kartu Perlengkapan

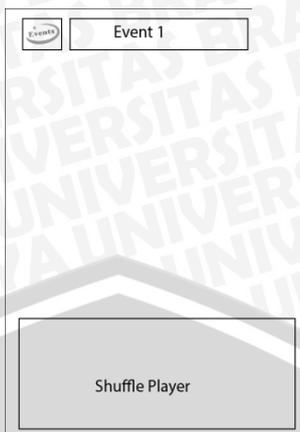
Hasil Evaluasi *playtest* :

- Waktu masing-masing pemain diganti dari 60 detik menjadi pilihan 30 detik, 45 detik dan 60 detik.
- Penambahan senjata pada masing-masing karakter untuk mengatasi masalah karakter tidak dapat menembak dikarenakan amunisi semua senjata telah habis
- Penambahan event yang dilakukan secara periodik.

d. Iterasi 4

Perubahan *rule* :

- Setiap karakter memiliki senjata dengan amunisi tidak terbatas dengan karakteristik yang lebih rendah dibandingkan dengan senjata beramunisi
- Terdapat event setiap akhir pada giliran genap.
- Ketika event dilakukan pemain mengambil kartu event untuk melihat keterangan dari event tersebut. Kartu event digambarkan pada gambar 3.9



Gambar 3.9 Kartu Event

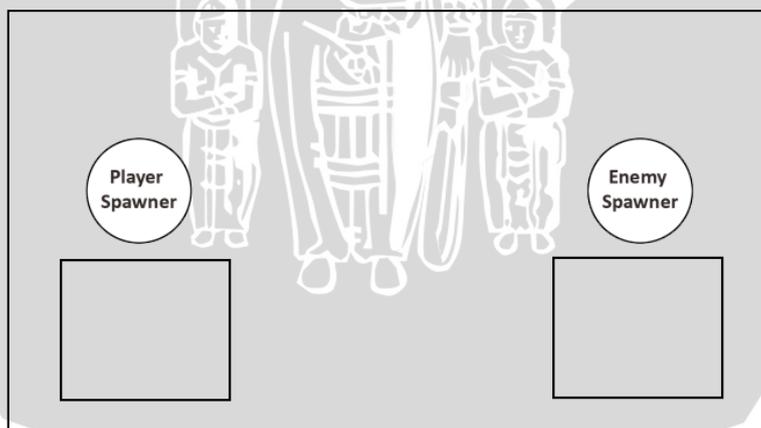
Hasil Evaluasi *playtest* :

- Tidak ada tambahan ide dari *playtest* ini, hal ini mengindikasikan bahwa permainan dirasa cukup dan dapat diimplementasikan.

3.2.3 Perancangan Level

Pada perancangan level ini, akan dibahas mengenai rancangan level yang disediakan pada *game* plotokan ini. *Game* ini terdiri dari 7 level dengan 3 diantaranya ada level tutorial untuk membiasakan pemain mengenai cara bermain. Setiap level pada *game* ini disebut dengan area dengan tujuan dari masing-masing level sama yaitu mengalahkan musuh yang ada. Dibawah ini akan dijelaskan mengenai level-level yang tersedia

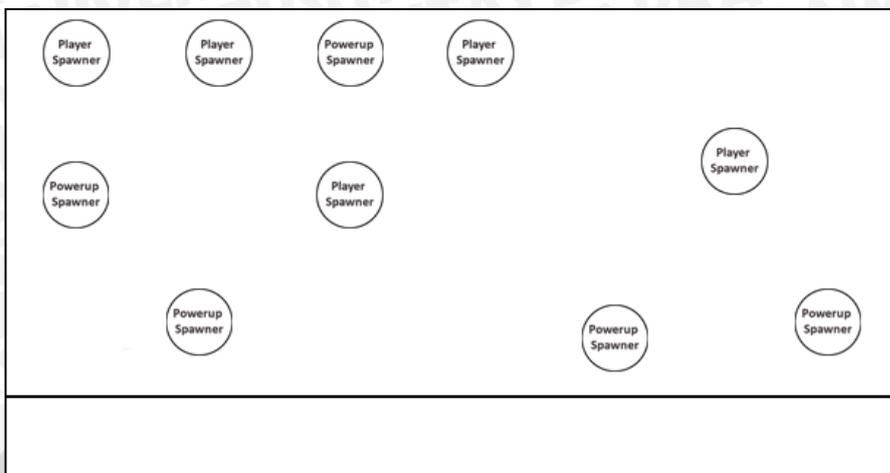
1. Area 1



Gambar 3.10 Konsep Area 1

Gambar 3.10 menggambarkan area 1 dimana merupakan level tutorial awal dengan desain dua pulau yang terpisah dimana pemain dan musuh berada pada pulau yang berbeda. Pada area ini pemain cukup menyesuaikan kecepatan tembakan untuk mengenai musuh tanpa perlu menggerakkan karakter. Pada area ini pemain diharapkan memahami pengaruh kecepatan terhadap lintasan peluru.

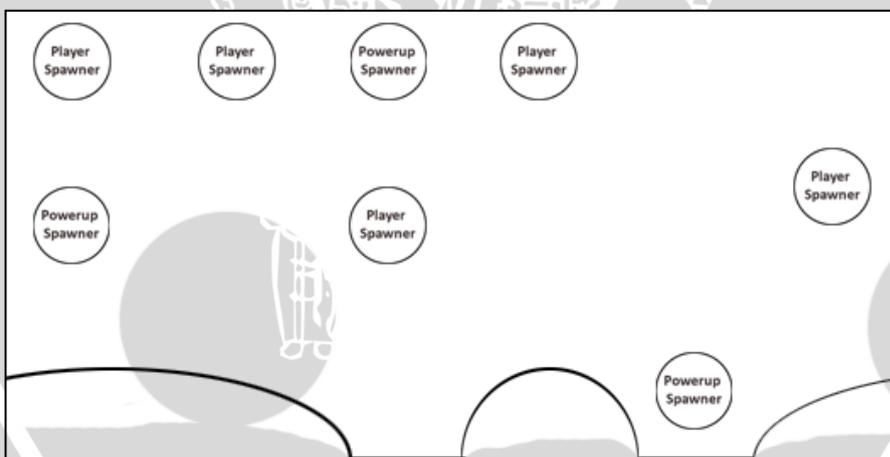
4. Area 4



Gambar 3.13 Konsep Area 4

Gambar 3.13 Menggambarkan area 4 dimana merupakan area non-tutorial pertama dengan terrain datar. Terrain dirancang mendatar untuk memudahkan pemain dalam melakukan perhitungan. *Spawn* karakter diacak sesuai dengan lokasi *spawn*er yang ada, sehingga memungkinkan karakter memiliki tempat yang berbeda-beda di awal permainan. Area ini menggunakan tema perbukitan.

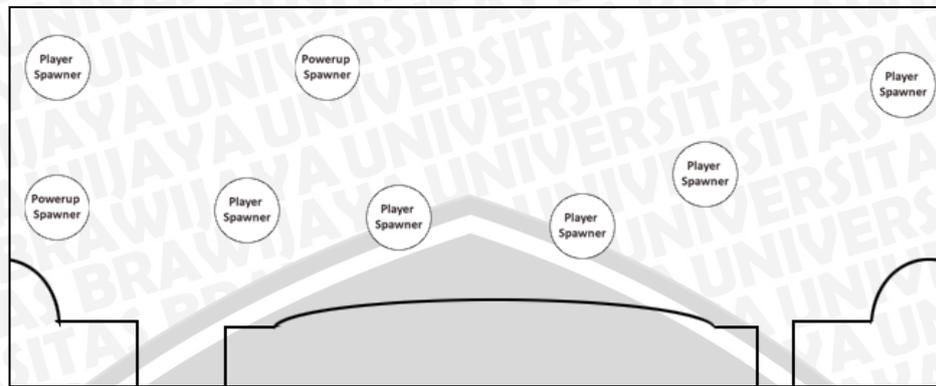
5. Area 5



Gambar 3.14 Konsep Area 5

Gambar 3.14 menggambarkan area 5 dimana merupakan area yang memiliki ketinggian yang berbeda. Perbedaan tinggi terrain tidak terlalu jauh sehingga perhitungan peluru tidak terlalu sulit. Lokasi *spawn* pemain menggunakan konsep seperti area 4 dimana terdapat beberapa lokasi sekaligus. Area ini menggunakan tema laut.

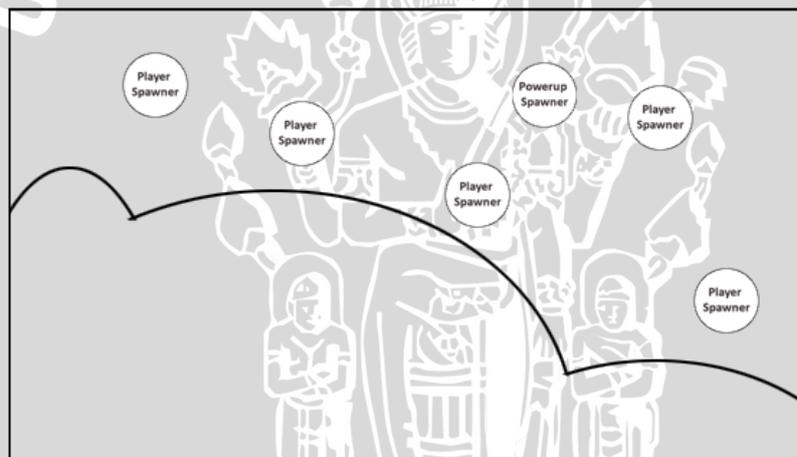
6. Area 6



Gambar 3.15 Konsep Area 6

Gambar 3.15 menggambarkan area 6 dimana merupakan modifikasi dari area 6 dimana terdiri dari 3 pulau yang dipisahkan dengan laut. Dengan dipisahkannya pulau gerak karakter menjadi terbatas dan bergantung pada lokasi awal karakter. Tema pada area ini adalah laut.

7. Area 7

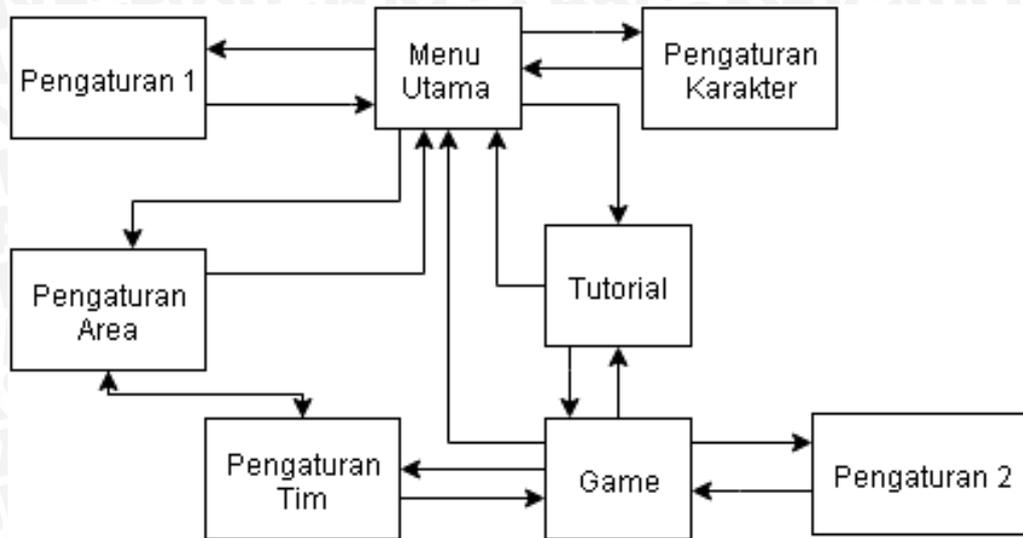


Gambar 3.16 konsep Area 7

Gambar 3.16 menggambarkan area 7 dengan bertemakan hutan, dimana terrain pada area ini tidak rata. Perbedaan terrain yang cukup besar membuat perhitungan kecepatan peluru menjadi sulit untuk dilakukan.

3.2.4 Game Screen Flow

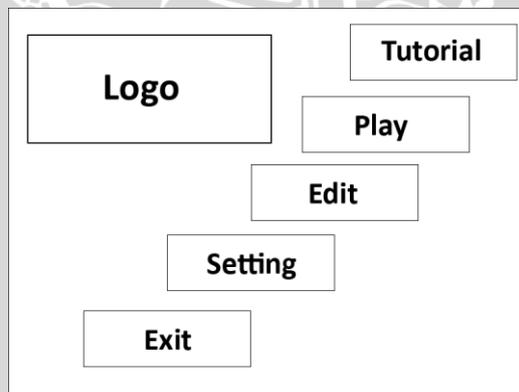
Game pletokan ini memiliki beberapa scene yang saling berhubungan satu sama lain. Gambar 3.17 menggambarkan hubungan antara scene yang ada pada game pletokan.



Gambar 3.17 Game Screen Flow Keseluruhan

1. Menu Utama

Merupakan tampilan awal pada saat *game* dimainkan. Pada menu ini ada beberapa tombol, yaitu *Setting*, *Edit*, *Play*, *Tutorial* dan *Exit*. Ketika tombol *Setting* ditekan akan muncul opsi untuk merubah pengaturan *game*, ketika tombol *edit* ditekan akan muncul *scene* pengaturan karakter, ketika tombol *play* ditekan akan muncul *scene* pengaturan area, ketika tombol *tutorial* ditekan akan muncul *scene* tutorial, dan ketika tombol *exit* ditekan akan muncul konfirmasi untuk menutup *game*. Gambar 3.18 menggambarkan konsep dari menu utama.

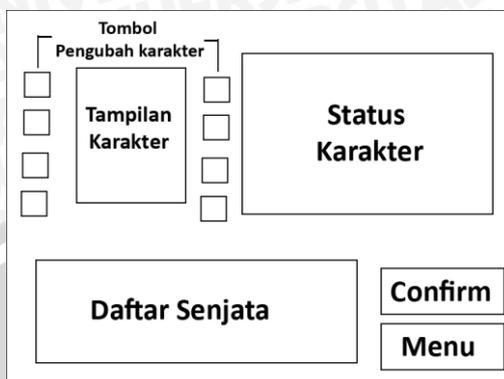


Gambar 3.18 Konsep Tampilan Menu Utama

2. Pengaturan Karakter

Merupakan tampilan untuk membuat karakter baru atau mengubah perlengkapan karakter termasuk senjata. Terdapat beberapa tombol pad tampilan ini yaitu *New*, *Load*, *Save*, *Confirm*, dan *Back*. Ketika tombol *New* ditekan akan membuat karakter baru dengan senjata kosong, ketika tombol *load* ditekan akan membuka data karakter yang telah disimpan terlebih dahulu, ketika tombol *save* ditekan akan menyimpan data karakter yang sekarang diubah, ketika tombol *confirm* ditekan akan

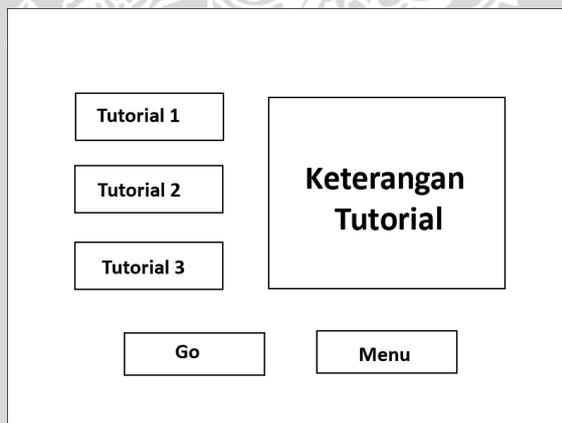
kembali ke menu awal dengan perubahan karakter, ketika tombol *Back* ditekan akan kembali ke menu awal tanpa perubahan karakter. Gambar 3.19 menggambarkan konsep dari pengaturan karakter.



Gambar 3.19 Konsep Tampilan Pengaturan Karakter

3. Tutorial

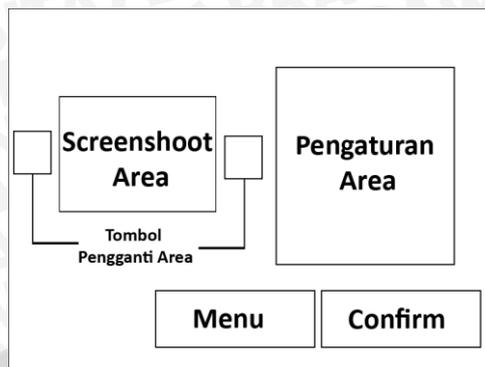
Tampilan untuk memilih area tutorial. Ada 2 macam tombol pada tampilan ini, yaitu tombol area dan tombol *Back*. Ketika tombol *back* ditekan akan kembali ke menu awal, ketika tombol *area* ditekan akan menuju *Game scene* sesuai dengan area yang terpilih. Gambar 3.20 menggambarkan konsep dari tutorial.



Gambar 3.20 Konsep Tampilan Tutorial

4. Pengaturan Area

Tampilan untuk memilih level selain level tutorial. Di tampilan ini ada beberapa bagian dari area yang bisa diubah yaitu jenis area, *timer*, jenis permainan, jenis *powerup*, dan jenis *event*. Jenis permainan disini ialah player melawan player, player melawan Ai, dan Ai melawan Ai. Difficulty dari Ai terdiri dari 3 jenis, *easy*, *medium*, dan *hard*. Gambar 3.21 menggambarkan konsep dari pengaturan area.



Gambar 3.21 Konsep Tampilan Pengaturan Area

5. Pengaturan tim

Tampilan untuk menambah atau mengubah karakter yang akan digunakan dalam tim. Dalam tampilan ini ada beberapa tombol, yaitu tombol *add*, *delete*, *play*, dan *back*. Ketika tombol *add* ditekan akan menambah karakter pada kelompok yang dipilih, ketika tombol *delete* ditekan akan menghapus karakter pada kelompok yang dipilih, ketika tombol *back* dipilih akan kembali ke *scene* pengaturan area dan ketika tombol *play* ditekan akan membua *scene* baru sesuai dengan area dan karakter yang dipilih. Gambar 3.22 menggambarkan konsep dari pengaturan tim.

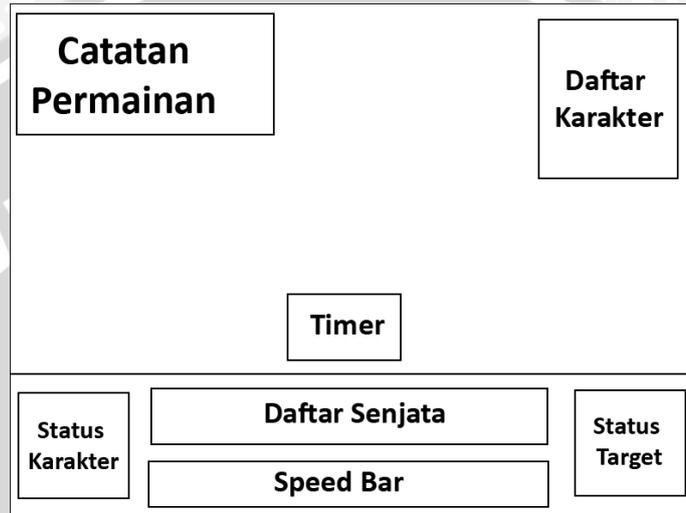


Gambar 3.22 Konsep Tampilan Pengaturan Tim

6. Game HUD

Tampilan yang berisi inti gameplay dari game pletakan. Tampilan ini berisi catatan permainan, daftar karakter, *timer*, catatan sudut, catatan jarak, penunjuk karakter, *navigator* dan 3 tombol lainnya. Catatan permainan berisi kejadian yang terjadi pada game, mulai dari *damage* yang diterima karakter, *event* yang terjadi serta posisi ledakan peluru, daftar karakter berisi tentang daftar karakter yang masih hidup serta urutan giliran karakter, *timer* berisi waktu karakter yang tersisa, penunjuk menunjukkan karakter yang sedang berjalan dan karakter

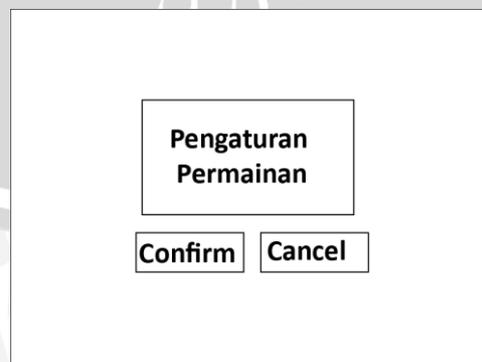
target, catatan sudut menunjukkan sudut yang akan dibentuk oleh peluru dan karakter, catatan jarak merupakan jarak antara karakter dengan penunjuk atau target, *navigator* merupakan tampilan pembantu yang memberitahukan apakah kecepatan atau sudut yang dibentuk sudah benar. Tombol *skip* jika ditekan akan melewati giliran karakter yang ada, tombol senjata jika ditekan akan mengubah senjata karakter yang sedang berjalan, tombol pengaturan jika ditekan akan membuka tampilan pengaturan 2. Gambar 3.23 menggambarkan konsep dari tampilan permainan.



Gambar 3.23 Konsep Tampilan Permainan

7. Pengaturan 1

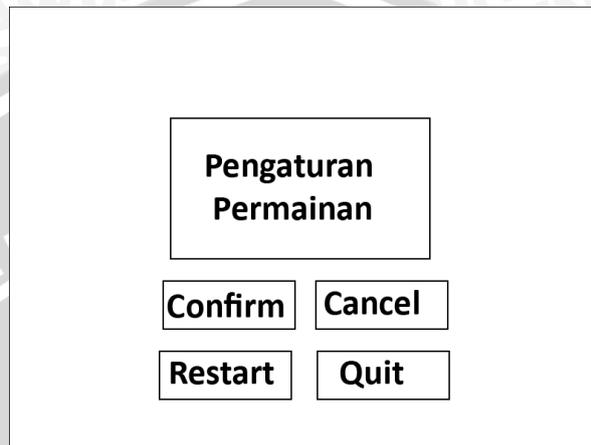
Tampilan untuk mengubah pengaturan permainan. Dalam tampilan ini terdapat 2 slide bar dan 2 tombol. Slide bar BGM untuk mengubah besar kecilnya musik *background*, *slide bar Sounds* untuk mengubah besar kecilnya efek suara, tombol *confirm* jika ditekan akan kembali ke menu awal dengan perubahan yang ada, jika tombol *cancel* ditekan akan kembali ke menu awal tanpa perubahan pengaturan. Gambar 3.24 menggambarkan konsep dari pengaturan 1.



Gambar 3.24 Konsep Tampilan Pengaturan 1

8. Pengaturan 2

Tampilan untuk mengubah pengaturan permainan. Tampilan ini mirip dengan tampilan pengaturan 1 hanya saja terdapat 2 tombol tambahan yaitu tombol *restart* dan tombol *quit*. Tombol *restart* jika ditekan akan memuat ulang game dan tombol *quit* jika ditekan akan menutup tampilan yang ada dan kembali ke menu awal. Gambar 3.25 menggambarkan konsep dari pengaturan 2.

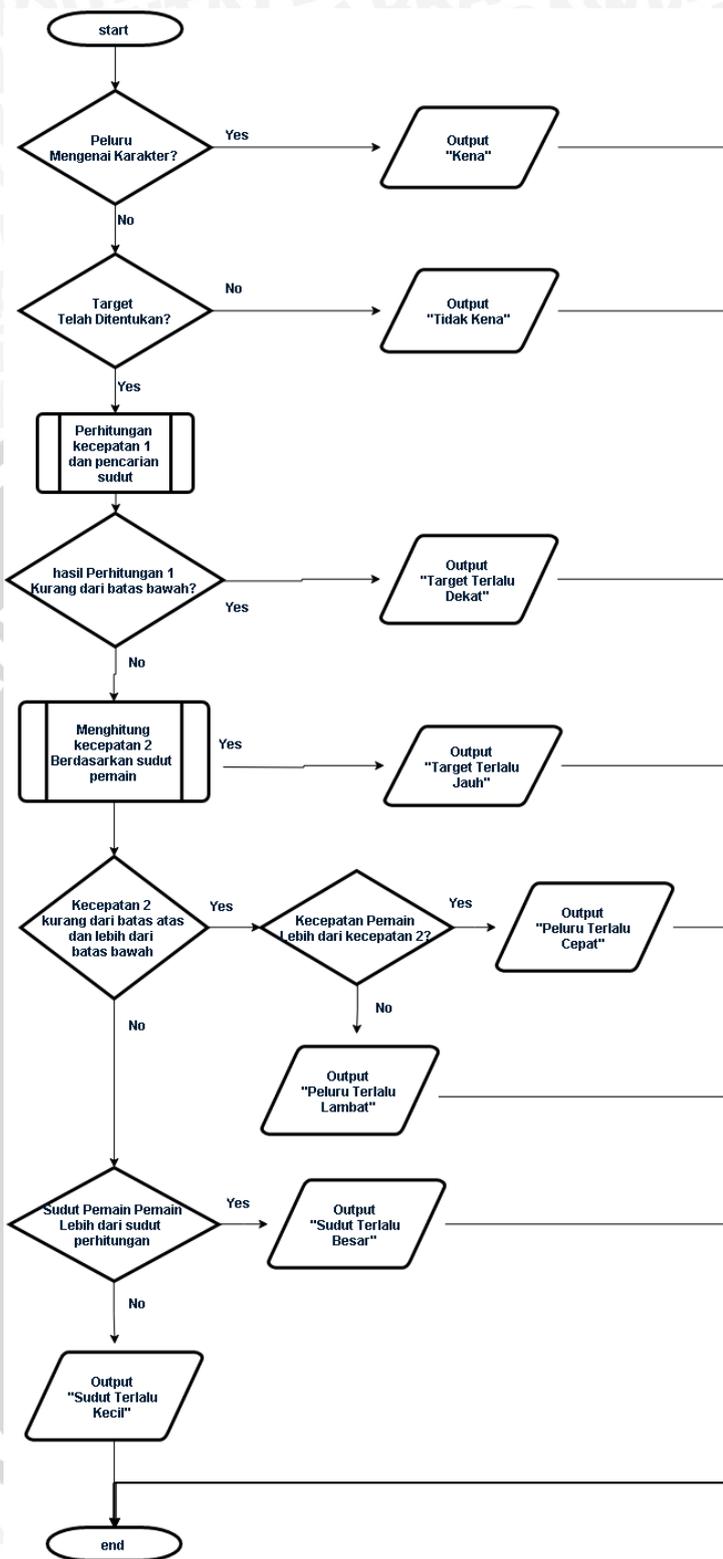


Gambar 3.25 Konsep Tampilan Pengaturan 2

3.2.5 Perancangan Navigator

Pada proses edukasi khususnya ketika peserta atau murid mengerjakan soal, kesalahan merupakan hal yang wajar terjadi. Oleh karena itu seorang guru harus memberikan saran, petunjuk atau arahan sehingga memungkinkan peserta untuk tidak melakukan kesalahan yang sama. Peranan guru dalam pemberi saran ini diwujudkan dalam fitur *Navigator*.

Navigator merupakan fitur merupakan pemberi sara, petunjuk atau arahan kepada pemain ketika seorang pemain melakukan kesalahan dalam menemuk. Fitur ini memberikan petunjuk pada pemain mengenai bagaimana seharusnya peluru ditembakkan ketika peluru pemain tidak mengenai target. *Navigator* menghitung kecepatan peluru yang diperlukan untuk mengenai target berdasarkan sudut yang sebelumnya ditentukan oleh pemain dan membandingkan dengan kecepatan yang diambil oleh pemain. Apabila hasil perhitungan kecepatan lebih besar dari kecepatan peluru pemain, maka navigator memberi saran untuk mempercepat laju peluru, apabila hasil perhitungan lebih kecil dari kecepatan peluru pemain *navigator* memberi saran untuk memperlambat laju peluru, dan jika hasil perhitungan berada diluar batas kecepatan yang diberikan maka navigator akan memberi saran untuk menggerakkan karakter. *Flowchart* dari navigator ditunjukkan pada gambar 3.26



Gambar 3.26 Flowchart Fitur Navigator

3.2.6 Perancangan AI

Selain dimainkan bersama orang lain, pletokan dapat dimainkan sendiri dengan melawan musuh yang digerakkan oleh kecerdasan buatan atau biasa disebut dengan AI. Skripsi ini memfokuskan kepada gameplay dari permainan itu sendiri, sehingga AI yang digunakan pada game ini menggunakan AI sederhana. Perancangan AI dapat diuraikan sebagai berikut :

1. Analisa *Behavior*

a. Pencarian Target

Pada langkah awal AI mencari seluruh objek karakter yang ada pada permainan yang nantinya diidentifikasi sebagai kawan ataupun lawan.

b. Pengurutan Prioritas Target

Pengurutan prioritas target diambil berdasarkan jarak antara AI berada dan objek karakter lawan yang ditemukan. Karakter yang memiliki jarak terdekat menjadi prioritas target penembakan.

c. Pencarian sudut dan kecepatan

Ketika target telah ditentukan AI menghitung berapa kecepatan dan sudut untuk dapat mengenai target. Apabila hasil perhitungan berada diluar limitasi kecepatan dan sudut yang diberikan AI berusaha untuk menempatkan posisinya berdasarkan hasil yang ditemukan.

d. Penempatan posisi

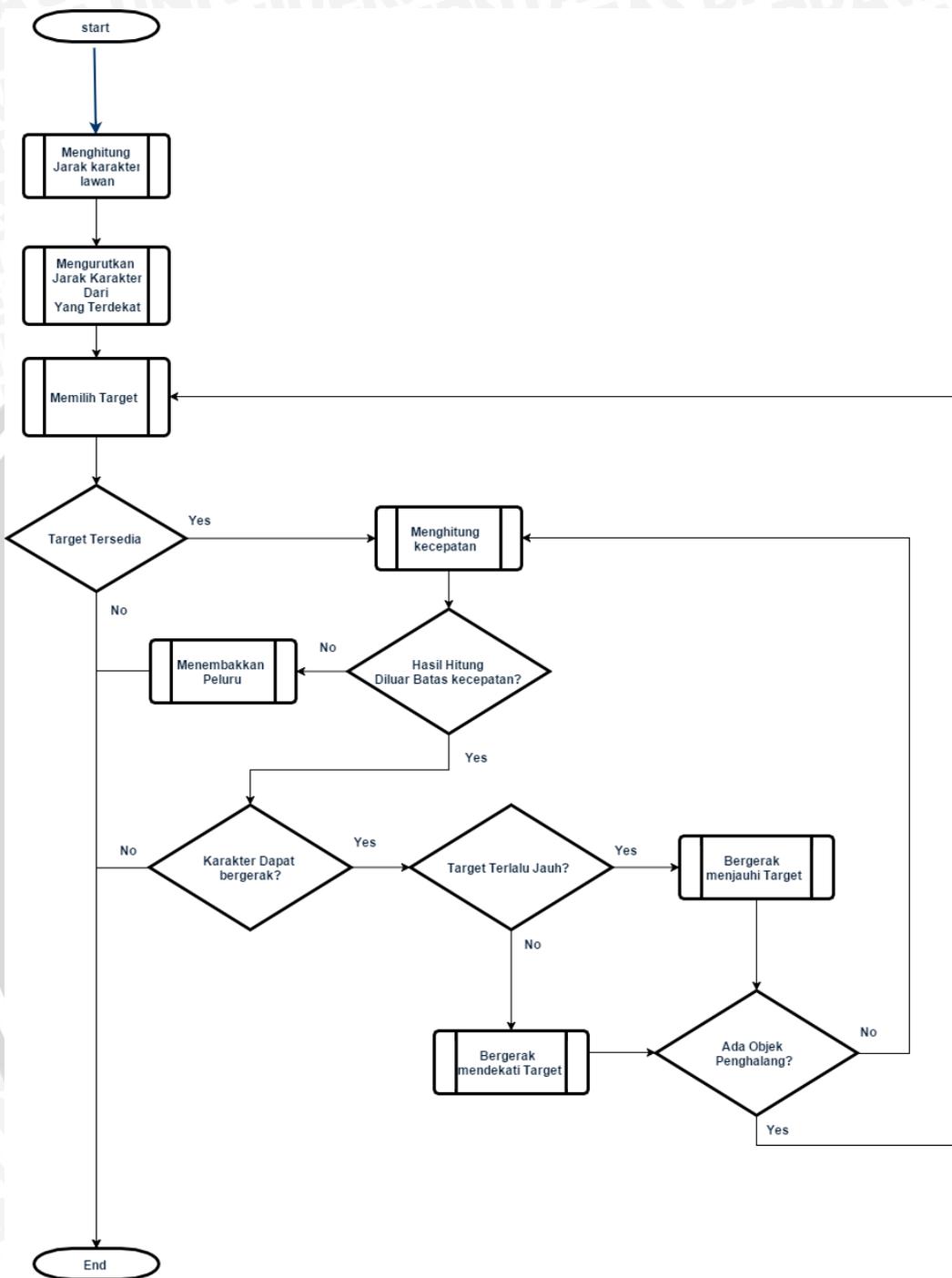
AI dapat bergerak mendekat ketika kecepatan yang diperlukan melebihi batas yang diberikan dan bergerak menjauh ketika hasil perhitungan kurang dari batas. Apabila kesempatan untuk bergerak bagi AI telah habis, maka AI akan mengakhiri gilirannya, apabila AI terhalang objek maka AI akan mengubah target serangan ke prioritas dibawahnya.

e. Menembak

Perbedaan tingkat kesulitan AI berada pada proses penembakan. Ketika mencapai posisi yang tepat AI menembakkan peluru berdasarkan hasil perhitungan yang sebelumnya dengan pemberian *error* dimana semakin tinggi tingkat kesulitan AI maka rentang *error* yang diberikan semakin kecil. Ketika AI selesai menembak maka AI akan mengakhiri gilirannya.

Flow chart dari jalannya AI ditunjukkan pada gambar 3.27

2. Flowchart



Gambar 3.27 Flowchart AI untuk karakter



BAB 4 IMPLEMENTASI

Bab ini membahas mengenai pengimplementasian pada game pletokan berdasarkan pada rancangan yang dibahas pada bab sebelumnya.

4.1 Pemilihan Teknologi

4.1.1 Spesifikasi Perangkat Keras

Pengimplementasian game pletokan dibuat di dalam lingkungan yang terdiri dari perangkat keras beserta perangkat lunak. Perangkat keras pada pengembangan pletokan berupa sebuah computer dengan spesifikasi seperti pada tabel 4.1

Tabel 4.1 Spesifikasi Perangkat Keras (laptop)

Laptop	
Processor	Genuine Intel(R) CPU T2130 @ 1.86GHz (2 CPUs), ~1.9GHz
Memory (RAM)	1024MB RAM
Harddisk	128 GB
Graphic Card	ATI RADEON XPRESS 200M Series

4.1.2 Spesifikasi Perangkat Lunak

Berbagai macam teknologi perangkat lunak dalam pengembangan *game* ini dijelaskan pada tabel 4.2

Tabel 4.2 Spesifikasi Perangkat Lunak (laptop)

Software	
Operating System	Windows 7 Ultimate 32-bit
Bahasa Pemrograman	C#
2D Graphic Editor	Adobe Phototoshop
Game Engine	Unity 5.1.0f3
Software Development Kit	Mono Develop

4.2 Batasan Implementasi

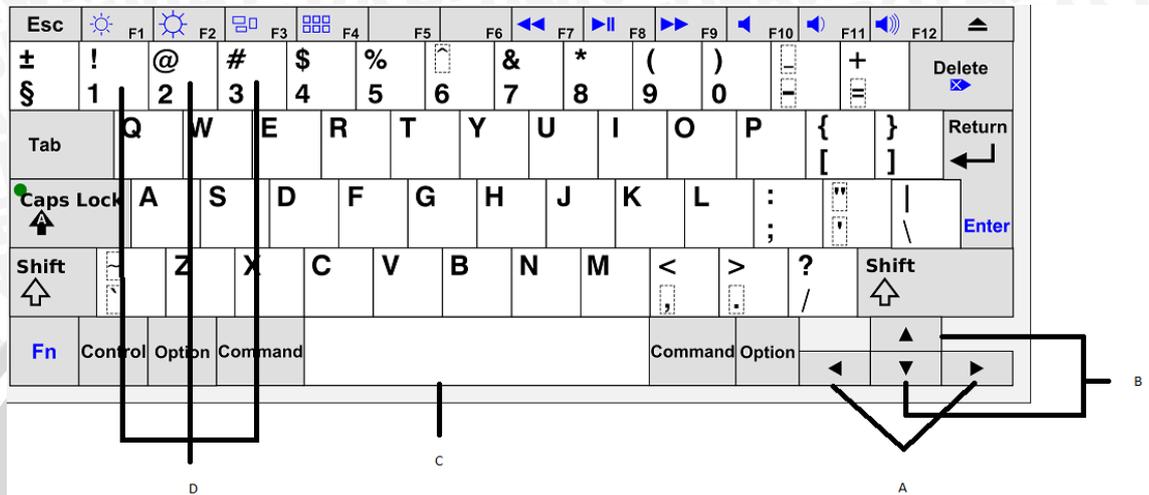
Beberapa batasan dalam mengimplementasikan *game* ini adalah sebagai berikut :

1. *Game* ini dibuat untuk platform desktop
2. *Game* lebih memfokuskan pada gameplay dengan AI sebagai pendukung

4.3 Implementasi Gameplay

4.3.1 Input Permainan

Input dari *game* ini menggunakan keyboard dengan penjelasan sebagai serperti pada gambar 4.1 :



Gambar 4.1 Input pada Keyboard

Keterangan :

- A : Mengubah posisi karakter ke kiri dan kekanan
- B : Mengubah besar sudut peluru yang akan ditembak
- C : Tombol untuk menembakkan peluru
- D : Tombol untuk mengubah senjata

4.3.2 Implementasi Gerak Peluru

Dalam *game* pletokan, peluru yang ditembakkan pemain membentuk lintasan gerak parabolik dengan pengaruh kecepatan, sudut, serta gravitasi. Dengan menggunakan rumus 2.5 gerak peluru dapat diimplementasikan seperti pada tabel 4.3

Tabel 4.3 Source code gerak peluru

SourceCode gerak peluru	
1	public static Vector2 calculatepointat(Vector2 speed, float gravity, float
2	time, int direction){
3	float x = speed.x * time * direction;
4	float y = speed.y * time - (gravity * time * time) / 2;
5	return new Vector2(x,y); }

Pada tabel 4.3, dapat dilihat bahwa perhitungan posisi peluru membutuhkan kecepatan, gravitasi, waktu serta arah. Arah disini menentukan kearah manakah peluru tersebut, positif untuk mengarah kekanan dan sebaliknya. Hasil dari perhitungan ini berupa Vector2.

4.3.3 Implementasi Perhitungan Kecepatan

Apabila posisi benda sudah diketahui dan ingin menentukan berapa kecepatan yang diperlukan untuk mencapai posisi tersebut dapat memasukkan persamaan 2.5 kedalam persamaan 2.6

$$\begin{aligned}
 x &= v_0 \cos \alpha \cdot t \\
 t &= \frac{x}{v_0 \cos \alpha} \\
 y &= v_0 \sin \alpha \cdot t - \frac{1}{2} g t^2 \\
 y &= v_0 \sin \alpha \cdot \frac{x}{v_0 \cos \alpha} - \frac{1}{2} g \left(\frac{x}{v_0 \cos \alpha} \right)^2 \\
 y &= \frac{x \cdot v_0 \sin \alpha}{v_0 \cos \alpha} - \frac{g x^2}{2 v_0^2 \cos \alpha^2} \\
 y &= \frac{2 x \cdot v_0^2 \sin \alpha \cdot v_0 \cos \alpha - g x^2}{2 v_0^2 \cos \alpha^2} \\
 2 v_0^2 \cos \alpha^2 \cdot y &= 2 x \cdot v_0^2 \sin \alpha \cdot \cos \alpha - g x^2 \\
 2 x \cdot v_0^2 \sin \alpha \cdot \cos \alpha - 2 v_0^2 \cos \alpha^2 \cdot y &= g x^2 \\
 x \cdot v_0^2 \sin 2\alpha - 2 v_0^2 \cos \alpha^2 \cdot y &= g x^2 \\
 v_0^2 (x \sin 2\alpha - 2 \cdot y \cdot \cos \alpha^2) &= g x^2 \\
 v_0^2 &= \frac{g x^2}{(x \sin 2\alpha - 2 \cdot y \cdot \cos \alpha^2)} \\
 v_0 &= \sqrt{\frac{g x^2}{(x \sin 2\alpha - 2 \cdot y \cdot \cos \alpha^2)}} \tag{4.1}
 \end{aligned}$$

Keterangan :

- x = posisi benda pada sumbu x (m)
- y = posisi benda pada sumbu y (m)
- v₀ = kecepatan awal benda (m/s)
- α = sudut yang dibentuk oleh benda terhadap bidang datar
- g = percepatan gravitasi (m/s²)



Dengan menggunakan hasil penjabaran rumus pada persamaan 4.1 maka perhitungan kecepatan dapat diimplementasikan seperti pada tabel 4.4

Tabel 4.4 Source code perhitungan kecepatan

SourceCode perhitungan kecepatan	
1	public float calculatepower(Vector2 distance, float angle, float gravity){
2	float power = -1;
3	float pembagi = (Mathf.Sin (Mathf.Deg2Rad * (2 * angle)) * distance.x - distance.y * 2 * Mathf.Cos (Mathf.Deg2Rad * (angle)) * Mathf.Cos (Mathf.Deg2Rad * (angle)));
4	if(pembagi>0)
5	power = Mathf.Sqrt(gravity * distance.x
6	* distance.x / pembagi);
7	return power; }
8	

Method ini menghasilkan kecepatan peluru dengan tipe data float. Nilai negatif berarti hasil perhitungan tidak dapat terdefinisi karena angka negatif tidak dapat dicari akarnya. Hal ini menandakan posisi penembak dengan target terlalu dekat peluru tidak dapat mengenai target.

4.3.4 Implementasi Navigator

Fitur navigator diimplementasikan sebagai maskot dengan dialog box berisi keterangan mengenai tembakan pemain. Isi dialog berubah setiap kali pemain selesai menembak dan isi dari dialog menyesuaikan dengan lokasi terakhir peluru. Implementasi fitur ini ditunjukkan pada gambar 4.2



Gambar 4.2 Implementasi Navigator

Dialog box memiliki 8 variasi dengan kondisi yang berbeda-beda. Kondisi tersebut ialah “Kena”, “Miss”, “Target Terlalu Jauh”, “Target Terlalu Dekat”, “Peluru Terlalu Cepat”, “Peluru Terlalu Lambat”, “Sudut Terlalu Besar”, dan “Sudut Terlalu kecil” yang diimplementasikan sebagai pada tabel 4.5 :

Tabel 4.5 Source code Navigator

SourceCode navigator	
1	public string getnav(){
2	string nav="";

```
3      if(hit){
4          nav = "kena";}
5      else{
6          if(target!=null){
7              float power = uim.manager .
8                  getcurrentcharscript(). speedbefore;
9              int angle = uim.manager.
10                 getcurrentcharscript().anglebefore;
11                 float temp = calculatepower
12                 (distance,angle,GameManager.gravity);
13                 if(powercalculate < 2.5){
14                     nav ="Target Terlalu Dekat";
15                 } else if(powercalculate > 15) {
16                     nav ="Target Terlalu Jauh";
17                 } else if(temp > 2.5 && temp<15){
18                     if(power < temp){
19                         nav ="perluru terlalu lambat";
20                     } else{
21                         nav ="peluru terlalu cepat";}
22                 } else if( angle < anglecalculate){
23                     nav = "sudut kurang";
24                 } else{
25                     nav = "sudut berlelebih;}}
26             else{
27                 nav = "miss";}}
28         return nav;}
```

4.3.5 Implementasi AI

Pada langkah awal, AI mencari seluruh objek karakter yang ada pada permainan. Objek karakter dibedakan berdasarkan *tag* yang diberikan, dalam hal ini *tag* tersebut adalah "Player". Implementasi pencarian pemain ditunjukkan pada tabel 4.6

Tabel 4.6 Source code pencarian target

SourceCode Pencarian Target	
1	public void seektarget(){
2	Target = GameObject.FindGameObjectsWithTag("Player");}

Ketika seluruh karakter ditemukan AI mulai menghitung jarak antara posisi dimana AI berada dengan posisi karakter yang ada. Untuk karakter dengan tim yang sama atau karakter yang sudah tidak dapat bergerak diberikan nilai yang cukup besar untuk mempermudah penentuan prioritas target dimana dalam hal ini besarnya nilai tersebut adalah 9999999. Implementasi perhitungan jarak ditunjukkan pada tabel 4.7

Tabel 4.7 Source code perhitungan jarak

SourceCode Perhitungan Jarak	
1	public float[] calculatedistance(GameObject[] charalist){
2	float[] distance = new float[charalist.Length];
3	for(int i=0;i<distance.Length;i++){
4	if(charalist[i].tag.Equals("Player")){
5	distance[i] = (transform. position - charalist[i]. transform.
6	position). magnitude;
7	if(distance[i] == 0 charalist[i]. GetComponent
8	<CharacterProperties>(). dead issameteam(charalist[i]))
9	distance[i] = 9999999;
10	}else
11	distance[i] = 9999999;
	}return distance;}

Tahap berikutnya merupakan tahapan penentuan prioritas target. Penentuan piroritas diambil berdasarkan jarak, semakin dekat jarak karakter lawan dengan jarak karakter yang sedang aktif, semakin tinggi prioritasnya. Implementasi pengurutan prioritas ditunjukkan pada tabel 4.8

Tabel 4.8 Source code pengurutan prioritas

SourceCode Pengurutan prioritas	
1	public int[] getsortedindex(float[] distance){
2	int[] temp = new int[distance.Length];
3	for(int i=0;i<distance.Length;i++){
4	temp[i] = i;
5	for(int i=0;i<(distance.Length-1);i++){

6	for(int d=(i+1);d<distance.Length;d++){
7	if(distance[temp[i]]>distance[temp[d]]){
8	int tempswap = temp[i];
9	temp[i] = temp[d];
10	temp[d] = tempswap;}}
11	}return temp; }

4.4 Implementasi Level

4.4.1 Implementasi Area 1

Area 1 diimplementasikan sebagai area yang memiliki dua pulau secara terpisah dimana karakter pemain dan karakter musuh terletak pada pulau yang berbeda. Permainan berakhir ketika karakter musuh kalah atau ketika amunisi pemain habis. Pada area ini tidak terdapat batas waktu dalam menembak. Implementasi Area 1 ditunjukkan pada gambar 4.3



Gambar 4.3 Implementasi Area 1

4.4.2 Implementasi Area 2

Area 2 diimplementasikan sebagai area yang memiliki dua pulau secara terpisah dimana karakter pemain dan karakter musuh terletak pada pulau yang berbeda. Pulau tempat karakter berada merupakan deretan pulau-pulau kecil untuk memberi ruang gerak pada pemain. Terdapat 2 peti pada area ini yang dapat digunakan untuk membantu menyelesaikan permainan. Permainan berakhir ketika karakter musuh kalah atau ketika amunisi pemain habis. Pada area ini tidak terdapat batas waktu dalam menembak. Implementasi area 2 ditunjukkan pada gambar 4.4



Gambar 4.4 Implementasi Area 2

4.4.3 Implementasi Area 3

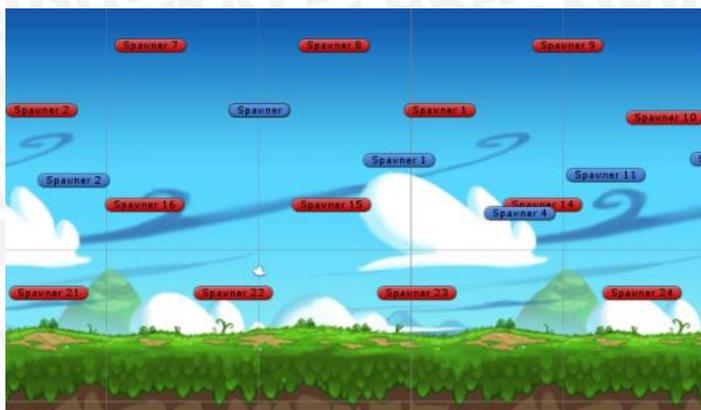
Area 3 diimplementasikan sebagai area yang memiliki tiga pulau secara terpisah dimana karakter pemain dan karakter musuh terletak pada pulau yang berbeda. Pulau tempat pemain berada dan pulau tempat musuh berada dipisahkan oleh pulau besar.. Permainan berakhir ketika karakter musuh kalah atau ketika amunisi pemain habis. Pada area ini tidak terdapat batas waktu dalam menembak. Implementasi area 3 ditunjukkan pada gambar 4.5



Gambar 4.5 Implementasi Area 3

4.4.4 Implementasi Area 4

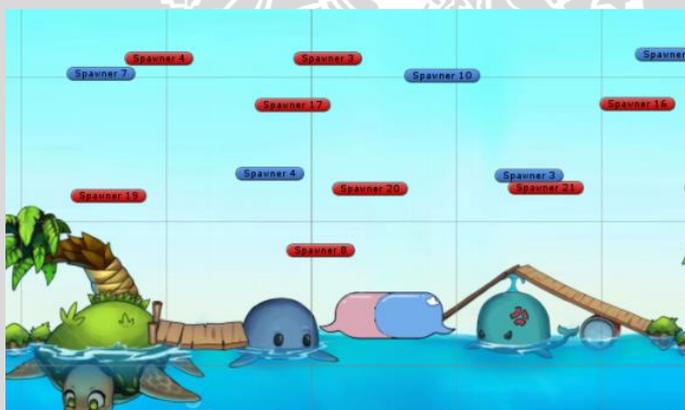
Area 4 diimplementasikan sebagai area yang relatif datar sehingga area ini merupakan kategori area mudah. Area ini memiliki beberapa lokasi munculnya karakter sehingga memungkinkan karakter muncul pada lokasi yang berbeda di setiap permainan. Permainan berakhir ketika salah satu atau kedua tim tidak lagi memiliki karakter yang masih bisa bergerak. Implementasi area 4 ditunjukkan pada gambar 4.6



Gambar 4.6 Implementasi Area 4

4.4.5 Implementasi Area 5

Area 5 diimplementasikan sebagai area dengan ketinggian beragam sehingga area ini merupakan kategori area menengah. Pemain dapat berjalan melewati punggung kura-kura, jembatan, kepala paus, dan bahkan balon kata. Area ini memiliki beberapa lokasi munculnya karakter sehingga memungkinkan karakter muncul pada lokasi yang berbeda di setiap permainan. Permainan berakhir ketika salah satu atau kedua tim tidak lagi memiliki karakter yang masih bisa bergerak. Implementasi area 5 ditunjukkan pada gambar 4.7



Gambar 4.7 Implementasi Area 5

4.4.6 Implementasi Area 6

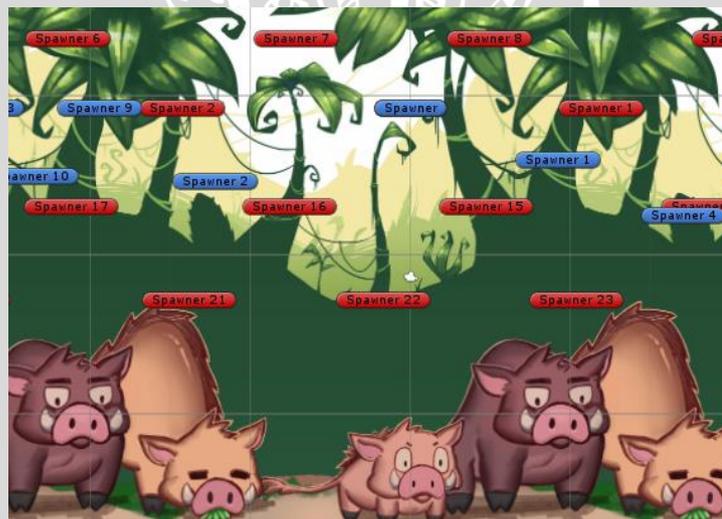
Area 6 diimplementasikan sebagai area yang dengan ketinggian beragam dan terpisah sehingga area ini merupakan kategori area menengah. Walaupun pulau dipisahkan dengan air, pemain tidak dapat berjalan diatas air sehingga gerakan karakter menjadi terbatas. Area ini memiliki beberapa lokasi munculnya karakter sehingga memungkinkan karakter muncul pada lokasi yang berbeda di setiap permainan. Permainan berakhir ketika salah satu atau kedua tim tidak lagi memiliki karakter yang masih bisa bergerak. Implementasi Area 6 ditunjukkan pada gambar 4.8



Gambar 4.8 Implementasi Area 6

4.4.7 Implementasi Area 7

Area 7 diimplementasikan sebagai area yang memiliki ketinggian beragam. Perbedaan ketinggian yang cukup besar membuat area ini lebih sulit untuk dimainkan. Pada area ini pemain dapat berjalan melewati punggung babi hutan. Area ini memiliki beberapa lokasi munculnya karakter sehingga memungkinkan karakter muncul pada lokasi yang berbeda di setiap permainan. Permainan berakhir ketika salah satu atau kedua tim tidak lagi memiliki karakter yang masih bisa bergerak. Implementasi area 7 ditunjukkan pada gambar 4.9



Gambar 4.9 Implementasi Area 7

4.5 Implementasi Art

4.5.1 Implementasi Logo

Logo dari *game* plotkan diimplementasikan menggunakan siluet maskot sebagai model dan dibuat menggunakan clip paint. Hasil Implementasi dari logo dapat dilihat pada gambar 4.10.



Gambar 4.10 Logo Game Pletokan

4.5.2 Implementasi Karakter

Sesuai dengan hasil perancangan *game* pletokan ini hanya terdapat 1 karakter dimana pemain dapat mengganti bagian bagiannya sesuai dengan keinginan sehingga dapat membedakan antara karakter 1 dengan karakter yang lain. Implementasi karakter ditunjukkan pada gambar 4.11



Gambar 4.11 Implementasi Karakter

Bagian tersebut ialah kepala, tangan, badan dan kaki. Penggantian bagian tersebut berpengaruh terhadap kemampuan dari karakter tersebut. Penggantian hanya dapat dilakukan diluar permainan. Bagian bagian dari karakter yang dapat diganti ditunjukkan pada gambar 4.12



Gambar 4.12 Implementasi Bagian dari Karakter

4.5.3 Implementasi PowerUp

Pada *game* pletokan ini terdapat 3 peti yang berfungsi sebagai powerup. Perbedaan dari ketiga ini terletak pada simbol yang memiliki arti tertentu, simbol sayap berarti bersisi hal yang positif, simbol tengkorak berarti bersisi hal yang negatif dan simbol tandatanya berisi keduanya. Implementasi powerup ditunjukkan pada gambar 4.13



Gambar 4.13 Implementasi Powerup

4.6 Implementasi User Interface

Bagian ini berisi penjelasan tentang *User Interface* yang diimplementasikan pada game pletokan. Implementasi *User interface* ini meliputi menu utama, pengaturan karakter, tutorial, pengaturan area, pengaturan tim, game HUD, pengaturan 1, dan pengaturan 2. Setiap perubahan *User Interface* diberi jeda waktu untuk menjalankan animasi.

4.6.1 Implementasi Menu utama

User Interface ini merupakan *user interface* yang pertama dijumpai oleh pemain. Terdapat 5 tombol dalam *interface* ini antara lain *story mode* untuk berpindah ke *interface* tutorial, *versus mode* untuk berpindah ke *interface* pengaturan area, *character config* untuk berpindah ke *interface* pengaturan karakter, *option* untuk berpindah ke *interface* pengaturan 1, dan *quit* untuk keluar dari permainan. Implementasi menu utama ditunjukkan pada gambar 4.14



Gambar 4.14 Implementasi Menu Utama

4.6.2 Implementasi Pengaturan Karakter

User interface ini berfungsi untuk mengubah atau membuat karakter baru. *Interface* ini memiliki 2 bagian penting yaitu bagian karakter dan bagian senjata. Bagian Karakter berisi gambar karakter yang menunjukkan tampilan karakter saat ini, arah panah disekitar gambar karakter berfungsi untuk mengubah bagian tertentu, *statistics* menunjukkan kemampuan karakter, serta tombol *new* untuk membuat karakter baru, *save* untuk menyimpan karakter, *load* untuk memuat ulang data karakter, serta *reset* untuk mengembalikan karakter ke pengaturan awal. Bagian senjata berisi *cost* yang menunjukkan limitasi senjata yang dapat dibawa, daftar senjata untuk mengubah senjata apa saja yang dibawa, tombol *new* untuk membuat daftar senjata baru, *save* untuk menyimpan daftar senjata, *load* untuk memuat ulang daftar senjata yang sebelumnya tersimpan, serta *reset*

untuk mengembalikan daftar senjata ke pengaturan awal. Tombol *save* untuk menerapkan perubahan karakter maupun senjata pada dan kembali ke menu utama sedangkan tombol *title* berfungsi untuk kembali ke menu utama tanpa menerapkan perubahan perubahan karakter maupun senjata. Implementasi pengaturan karakter ditunjukkan pada gambar 4.15



Gambar 4.15 Implementasi Pengaturan Karakter

4.6.3 Implementasi Tutorial

User interface ini berfungsi untuk memilih 1 dari 3 tutorial yang ingin dijalankan. Interface ini berisi tombol tutorial untuk memilih jenis tutorial, gambar area untuk menunjukkan gambaran area yang dipilih, tombol menu untuk kembali ke menu awal, dan tombol go untuk menuju ke area tutorial sesuai dengan pilihan. Implementasi tutorial ditunjukkan pada gambar 4.16



Gambar 4.16 Implementasi Tutorial

4.6.4 Implementasi Pengaturan Area

User interface ini berfungsi untuk memainkan area selain area tutorial. Pada *Interface* ini terdapat pengaturan tipe permainan yang berupa *player vs player*, *player vs AI*, dan *AI vs AI*. Selain itu terdapat pengaturan *timer* untuk mengatur standar waktu pada permainan, pengaturan *hazard* untuk mengubah *event* apa saja yang akan terjadi, pengaturan *powerup* untuk mengubah apa saja *powerup* yang akan muncul, serta gambar area untuk menunjukkan gambaran area yang dipilih. Tombol *go* digunakan untuk membuka tampilan pengaturan tim dan tombol menu untuk kembali ke menu awal. Implementasi pengaturan area ditunjukkan pada gambar 4.17



Gambar 4.17 Implementasi Pengaturan Area

4.6.5 Implementasi Pengaturan tim

User interface ini berfungsi untuk mengatur karakter yang ada pada tim. *Game* baru bisa dimainkan jika kedua tim terdapat minimal 1 karakter. Terdapat 2 komponen utama pada *interface* ini yaitu komponen tim dan komponen karakter. Komponen tim berisi daftar karakter pada suatu tim serta dapat mengambahkan karakter kedalam tim. Komponen karakter berisi penjelasan karakter yang dipilih mengenai daftar senjata yang dibawa serta kemampuan karakter tersebut. *Interface* ini dapat juga digunakan untuk menghapus karakter yang ada pada tim. Tombol *play* berfungsi untuk menuju ke area permainan sesuai dengan pengaturan area dan pengaturan tim yang sudah ditentukan serta tombol *back* berfungsi untuk kembali ke pengaturan area. Implementasi pengaturan tim ditunjukkan pada gambar 4.18



Gambar 4.18 Implementasi Pengaturan tim

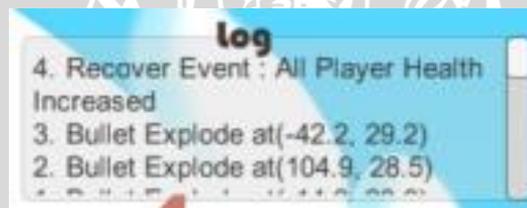
4.6.6 Implementasi HUD Game

Implementasi HUD dibagi menjadi dua, antara lain HUD bagian atas dan bagian bawah. Bagian atas terdiri dari tombol bergambar gerigi yang berfungsi untuk membuka pengaturan 2, tombol arah panah berfungsi untuk melewati giliran karakter yang sedang berjalan, *log* berisi catatan permainan, *turn order* berisi mengenai urutan giliran karakter, serta *navigator* yang berfungsi sebagai pemberi saran. HUD bawah berisi mengenai timer, daftar senjata untuk mengubah senjata yang digunakan karakter, gambar player untuk menunjukkan gambar dari karakter yang sedang mendapat giliran, gambar target untuk

menampilkan target yang dipilih, angle untuk menunjukkan sudut yang sedang dipilih, power untuk menunjukkan kecepatan peluru sekarang dan sebelumnya, distance untuk menunjukkan jarak antara karakter dengan target, serta gravitasi untuk menunjukkan besarnya gravitasi. Implementasi *game* HUD ditunjukkan pada gambar 4.19



Gambar 4.19 Implementasi *Game* HUD



Gambar 4.20 Implementasi Log HUD

Gambar 4.20 menunjukkan log HUD yang berisi tentang catatan permainan. Catatan ini berisi Event yang terjadi, posisi peluru meledak, tingkat kerusakan yang dialami karakter, serta powerup yang didapat oleh karakter.



Gambar 4.21 Implementasi Turn Order HUD

Gambar 4.21 menunjukkan Turn Order HUD yang berisi mengenai daftar karakter pada game yang masih dapat digerakkan. Berisi mengenai speed poin untuk menentukan urutan giliran dari karakter. Pemain dapat memilih target melalui HUD ini.



Gambar 4.22 Implementasi *Timer* HUD

Gambar 4.22 menunjukkan *Timer* HUD yang menunjukkan waktu yang tersisa pada pemain untuk menggerakkan salah satu karakter. Ketika *timer* ini menunjukkan angka 0, giliran pemain tersebut berakhir.



Gambar 4.23 Implementasi Properti Karakter HUD

Gambar 4.23 menunjukkan Properti karakter yang berisi keterangan mengenai karakter yang aktif baik itu sudut yang diambil pada penembakan sebelumnya dan sudut sekarang, daftar senjata, dan kecepatan yang diambil pada penembakan sebelumnya. Pemilihan senjata dapat dilakukan pada HUD ini.



Gambar 4.24 Implementasi Avatar Karakter HUD

Gambar 4.24 menunjukkan Avatar karakter HUD menunjukkan *Icon* dan karakter yang sedang aktif.



Gambar 4.25 Implementasi Target HUD

Gambar 4.25 menunjukkan target HUD yang berisi mengenai keterangan target dan keterangan gravitasi pada permainan. Apabila target tidak ditentukan jarak berisi mengenai jarak antara karakter yang sedang aktif dan pointer.

4.6.7 Implementasi Pengaturan 1

User interface ini berfungsi untuk mengubah pengaturan permainan. Interface ini berisi mengenai BGM untuk mengubah besar kecilnya *volume* musik *background*, Sounds untuk mengubah besar kecilnya *sounds effect*, serta *properties* untuk menghilangkan atau menampilkan keterangan karakter. Tombol *Save* berfungsi untuk kembali ke menu utama dengan menerepakan pengubahan pengaturan, serta tombol *quit* untuk kembali ke menu awal tanpa menerapkan pengubahan pengaturan. Implementasi pengaturan 1 ditunjukkan pada gambar 4.26



Gambar 4.26 Implementasi Pengaturan Permainan

4.6.8 Implementasi Pengaturan 2

User interface yang memiliki fungsi mirip dengan pengaturan 1 dengan penambahan tombol *restart* dan *quit*. Ketika tampilan ini muncul, kondisi permainan akan berhenti sampai tampilan ini ditutup. Tombol *confirm* berfungsi untuk kembali ke menu permainan dengan menerepakan pengubahan pengaturan, serta tombol *cancel* untuk kembali ke menu permainan tanpa menerapkan pengubahan pengaturan. Tombol *restart* berfungsi untuk memuat ulang permainan dan tombol *quit* berfungsi untuk kembali ke menu awal. Implementasi pengaturan 2 ditunjukkan pada gambar 4.27



Gambar 4.27 Implementasi Pengaturan Permainan 2

BAB 5 PENGUJIAN DAN EVALUASI

Bab ini berisi mengenai tahapan pengujian dari implementas game pletokan menggunakan pengujian *whitebox* dan *blackbox*. Pengujian dilakukan pada perangkat dengan speisifikasi sesuai pada tabel 4.1.

5.1 Pengujian White Box

Pengujian *whitebox* merupakan pengujian aplikasi berdasarkan detail perancangan serta *source code* dari program dengan membaginya kedalam beberapa proses uji. Pengujian *whitebox* yang digunakan menggunakan metode *basis path testing* sedangkan yang diuji adalah gerak peluru, pencarian kecepatan dan fitur *navigator*.

Tabel 5.1 *Source code* gerak peluru

SourceCode gerak peluru	
1	public static Vector2 calculatepointat(Vector2 speed, float gravity, float
2	time, int direction){
3	float x = speed.x * time * direction;
4	1 float y = speed.y * time - (gravity * time * time) / 2;
5	return new Vector2(x,y); }

1

Gambar 5.1 *Flow graph* dari gerak peluru

Flow graph gerak peluru pada gambar 5.1 menggambarkan *path* yang terdapat dalam tabel 5.1. Perhitungan gerak peluru tidak memiliki cabang sehingga flograph hanya memiliki 1 *node* dan 0 *edge*. Nilai *cyclometrix complexity* dapat dihitung pada persamaan 5.1.

$$V(G) = E - N + 2$$

$$V(G) = 0 - 1 + 2$$

$$V(G) = 1 \tag{5.1}$$

Pada persamaan 5.1 $V(G)$ merupakan nilai *cyclometrix complexity* dengan E sebagai *Edge* dan N adalah *Node*. Nilai *cyclometrix complexity* 1 merupakan nilai dengan kategori kompleksitas rendah.

Tabel 5.2 Kasus uji untuk pergerakan peluru

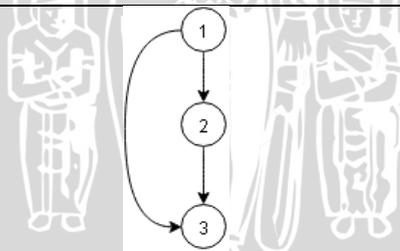
No.	Parameter Kasus Uji	Hasil yang diharapkan	Hasil yang didapat	Status Valid
1.	Speed = (18,24) gravity = 10	(72,16)	(72,16)	Valid

	time = 4 direction = 1			
1.	Speed = (16,12) gravity = 10 time = 2 direction = -1	(-32,4)	(-32,4)	Valid

Dari Tabel 5.2, didapatkan hasil yang valid dari parameter kasus uji yang diberikan. Maka dapat disimpulkan pergerakan peluru bekerja sesuai harapan.

Tabel 5.3 Source code perhitungan kecepatan

SourceCode perhitungan kecepatan	
1	public float calculatepower(Vector2 distance, float angle, float gravity){
2	float power = -1;
3	float pembagi = (Mathf.Sin (Mathf.Deg2Rad *
4	(2 * angle)) * distance.x - distance.y * 2 *
5	Mathf.Cos (Mathf.Deg2Rad * (angle)) *
6	Mathf.Cos (Mathf.Deg2Rad * (angle)));
7	if(pembagi>0)
8	2 power = Mathf.Sqrt(gravity * distance.x * distance.x
9	/ pembagi);
20	3 return power; }



Gambar 5.2 Flow graph dari perhitungan kecepatan

Pada gambar 5.2 didapatkan :

Path 1 : 1 – 3

Path 2 : 1 – 2 – 3

Flow graph perhitungan kecepatan pada gambar 5.2 menggambarkan path yang terdapat dalam tabel 5.3. yang terdiri dari 3 node dan 3 edge. Nilai cyclometrix complexity dapat dihitung sebagai pada persamaan 5.2.

$$V(G) = E - N + 2$$

$$V(G) = 3 - 3 + 2 \tag{5.2}$$



$$V(G) = 2$$

Tabel 5.4 Kasus uji untuk perhitungan kecepatan

No.	Parameter Kasus Uji	Hasil yang diharapkan	Hasil yang didapat	Status Valid
1.	distance = (2,1) gravity = 10 angle = 36.8699	62.49999	62.49998	Valid
2.	distance = (2,30) gravity = 10 angle = 45	-1	-1	Valid

Dari Tabel 5.4 , didapatkan hasil yang valid dari parameter kasus uji yang diberikan. Maka dapat disimpulkan perhitungan kecepatan bekerja sesuai harapan.

Tabel 5.5 Source code Navigator

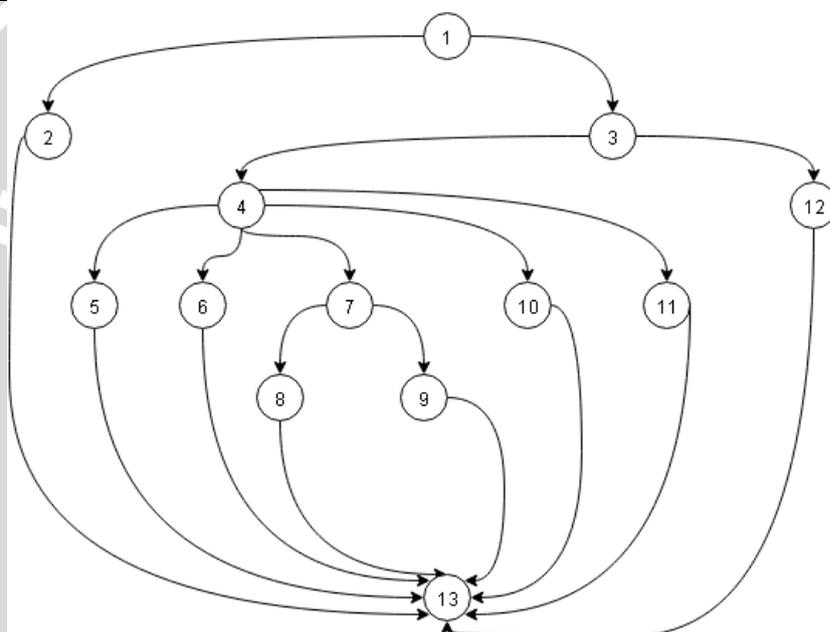
```

SourceCode navigator
1 public string getnav(bool hit){
2     1 [ string nav="";
3     if(hit){
4     2 [ [ nav = "kena";
5     }
6     3 [ else{
7     [
8         float power = uim.manager .
9     4 [ getcurrentcharscript(). speedbefore;
10        int angle = uim.manager.
11        getcurrentcharscript().anglebefore;
12        float temp = calculatepower
13        (distance,angle,GameManager.gravity);
14        if(powercalculate < 2.5){
15        5 [ [ nav ="Target Terlalu Dekat";
16        } else if(powercalculate > 15) {
17        6 [ [ nav ="Target Terlalu Jauh";
18        7 [ [ } else if(temp > 2.5 && temp<15){
19        if(power < temp){
20        8 [ [ nav ="perluru terlalu lambat";
    
```



```

21 } else{
22     9      [   [   nav ="peluru terlalu cepat";}
23           [   [   } else if( angle < anglecalculate){
24     10     [   [   nav = "sudut kurang";
25           [   [   } else{
26     11     [   [   nav = "sudut berlelebi";
27           [   [   }   }
28           else{
29     12     [   [   nav = "miss";}]
30     13 [   [   return nav;}
    
```



Gambar 5.3 Flow graph Navigator

Pada gambar 5.3 didapatkan :

Path 1 : 1 – 2 – 13

Path 2 : 1 – 3 – 4 – 5 – 13

Path 3 : 1 – 3 – 4 – 6 – 13

Path 4 : 1 – 3 – 4 – 7 – 8 – 13

Path 5 : 1 – 3 – 4 – 7 – 9 – 13

Path 6 : 1 – 3 – 4 – 10 – 13

Path 7 : 1 – 3 – 4 – 11 – 13

Path 8 : 1 – 3 – 12 – 13



Flow graph perhitungan kecepatan pada gambar 5.3 menggambarkan *path* yang terdapat dalam tabel 5.5 yang terdiri dari 19 *edge* dan 13 *node*. Nilai *cyclometrix complexity* dapat dihitung pada persamaan 5.3.

$$V(G) = E - N + 2$$

$$V(G) = 19 - 13 + 2$$

$$V(G) = 8 \quad (5.3)$$

Tabel 5.6 Kasus Uji untuk Navigator

No.	Parameter Kasus Uji	Hasil yang diharapkan	Hasil yang didapat	Status Valid
1.	Peluru mengenai objek	Kena	Kena	Valid
2.	Peluru tidak mengenai objek Target ditentukan Hasil perhitungan kecepatan kurang dari batas bawah	Target Terlalu Dekat	Target Terlalu Dekat	Valid
3.	Peluru tidak mengenai objek Target ditentukan Hasil perhitungan kecepatan lebih dari batas atas	Target Terlalu Jauh	Target Terlalu Jauh	Valid
4.	Peluru tidak mengenai objek Target ditentukan Kecepatan peluru lbih besar dari hasil hitung	Peluru Terlalu Cepat	Peluru Terlalu Cepat	Valid
5.	Peluru tidak mengenai objek Target ditentukan Kecepatan peluru lbih besar dari hasil hitung	Peluru Terlalu Lambat	Peluru Terlalu Lambat	Valid
6.	Peluru tidak mengenai objek Target ditentukan sudut peluru lbih besar dari hasil hitung	Sudut terlalu Besar	Sudut Terlalu Besar	Valid
7.	Peluru tidak mengenai objek Target ditentukan sudut peluru lbih kecil dari hasil hitung	Sudut Terlalu Kecil	Sudut Terlalu kecil	Valid

8.	Peluru tidak mengenai objek Target ditentukan	Miss	Miss	Valid
----	--	------	------	-------

Dari Tabel 5.6, didapatkan hasil yang valid dari parameter kasus uji yang diberikan. Maka dapat disimpulkan *navigator* bekerja sesuai harapan.

5.2 Pengujian Black Box

Pengujian *blackbox* merupakan pengujian tanpa memperhatikan algoritma pada source code, meualikan lebih pada hasil pengamatan dari data uji serta fungsional dari perangkat lunak. Pada pengujian kali ini menggunakan metode pengujian fungsionalitas, pengujian kombinatorial, dan *Test Flow Diagram* (TFD).

5.2.1 Pengujian *Functionality*

Pengujian *fiunctionality* merupakan pengujian yang menguji apakah fitur pada game sudah terimplementasikan dengan baik.

Tabel 5.7 Tabel Uji *Functionality*

No.	List Fitur	Status
1.	Main Menu	Valid
2.	Pengaturan karakter	Valid
3.	Tutorial	Valid
4.	Pengaturan Area	Valid
5.	Pengaturan Tim	Valid
6.	Timer HUD	Valid
7.	Speed HUD	Valid
8.	Character HUD	Valid
9.	Log HUD	Valid
10.	Turn Order HUD	Valid
11.	Pengaturan Permainan 1	Valid
12.	Pengaturan Permainan 2	Valid

Pada table 5.7 dilakukan 12 pengujian pada masing-masing fitur yang menghasilkan 12 nilai valid. Hal ini menunjukkan bahwa *game* sudah terimplementasikan dengan baik.

5.2.2 *Combinatorial Testing*

Combinatorial testing merupakan metode pengujian untuk menemukan kesalahan dan mendapatkan kepercayaan diri dalam perangkat lunak. Pengujian kombinatorial pada game ini dilakukan pada pengaturan permainan, pengaturan area, dan pengaturan karakter.

Pada pengaturan permainan terdapat 5 parameter yang akan diuji, yaitu status bgm, *volume* bgm, status *sound effect*, volume sound effect, dan status properti. Pada tabel 5.8 berisi mengenai tabel kombinatorial dengan parameter status BGM yang berisi aktif dan tidak aktif.

Tabel 5.8 Tabel *Combinatorial Testing* pengaturan permainan 1 parameter

Status BGM
Aktif
Tidak Aktif

Pada tabel 5.9 ditambahkan parameter kedua yaitu *Volume* BGM yang berisi 0.5 dan 1.

Tabel 5.9 Tabel *Combinatorial Testing* pengaturan permainan 2 parameter

Status BGM	Volume BGM
Aktif	0.5
Tidak Aktif	1
Aktif	1
Tidak Aktif	0.5

Pada tabel 5.10 ditambahkan parameter ketiga yaitu status *sound effect* yang berisi aktif dan tidak aktif.

Tabel 5.10 Tabel *Combinatorial Testing* pengaturan permainan 3 parameter

Status BGM	Volume BGM	Status Sound Effect
Aktif	0.5	Aktif
Tidak Aktif	1	Tidak Aktif
Aktif	1	Aktif
Tidak Aktif	0.5	Tidak Aktif

Pada tabel 5.11 ditambahkan parameter keempat yaitu *Volume Sound Effect* yang berisi 0.5 dan 1.

Tabel 5.11 Tabel *Combinatorial Testing* pengaturan permainan 4 parameter

Status BGM	Volume BGM	Status Sound Effect	Volume Sound Effect
Aktif	0.5	Aktif	0.5
Tidak Aktif	1	Tidak Aktif	1
Aktif	1	Aktif	1
Tidak Aktif	0.5	Tidak Aktif	0.5

Pada tabel 5.12 ditambahkan parameter kelima yaitu status properti yang berisi aktif dan tidak aktif.

Tabel 5.12 Tabel *Combinatorial Testing* pengaturan permainan 5 parameter

Status BGM	Volume BGM	Status Sound Effect	Volume Sound Effect	Status properti
Aktif	0.5	Aktif	0.5	Aktif
Tidak Aktif	1	Tidak Aktif	1	Tidak Aktif
Aktif	1	Aktif	1	Aktif
Tidak Aktif	0.5	Tidak Aktif	0.5	Tidak Aktif

Pada tabel 5.13 dilakukan pengujian dari hasil tabel pariwise yang ada pada tabel 5.12

Tabel 5.13 Tabel hasil Uji *Combinatorial Testing* pengaturan permainan

Status BGM	Volume BGM	Status Sound Effect	Volume Sound Effect	Status properti	Hasil Uji
Aktif	0.5	Aktif	0.5	Aktif	Valid
Tidak Aktif	1	Tidak Aktif	1	Tidak Aktif	Valid
Aktif	1	Aktif	1	Aktif	Valid
Tidak Aktif	0.5	Tidak Aktif	0.5	Tidak Aktif	Valid

Dari Tabel 5.13, didapatkan hasil yang valid dari parameter kasus uji yang diberikan. Maka dapat disimpulkan pengaturan permainan bekerja dengan baik.

Pada pengaturan area terdapat 3 parameter yang akan diuji, yaitu nama area, tipe permainan, dan *timer*. Pada tabel 5.14 berisi mengenai tabel kombinatorial dengan parameter nama area yang berisi area 4, area 5, area 6 dan area 7.

Tabel 5.14 Tabel *Combinatorial Testing* pengaturan area 1 parameter

Nama Area
Area 4
Area 5
Area 6
Area 7

Pada tabel 5.15 ditambahkan parameter kedua yaitu tipe permainan yang berisi player vs player, player vs ai dan ai vs ai.

Tabel 5.15 Tabel *Combinatorial Testing* pengaturan area 2 parameter

Nama Area	Tipe Permainan
Area 4	Player vs Player

Area 5	Player vs Ai
Area 6	Ai vs Ai
Area 7	Player vs Player
Area 4	Player vs Ai
Area 5	Ai vs Ai
Area 6	Player vs Player
Area 7	Player vs Ai
Area 4	Ai vs Ai
Area 5	Player vs Player
Area 6	Player vs AI
Area 7	AI vs AI

Pada tabel 5.16 ditambahkan parameter ketiga yaitu *timer* yang berisi 30,50, dan 99

Tabel 5.16 Tabel *Combinatorial Testing* pengaturan area 3 parameter

Nama Area	Tipe Permainan	Timer
Area 4	Player vs Player	30
Area 5	Player vs Ai	50
Area 6	Ai vs Ai	99
Area 7	Player vs Player	50
Area 4	Player vs Ai	99
Area 5	Ai vs Ai	30
Area 6	Player vs Player	99
Area 7	Player vs Ai	30
Area 4	Ai vs Ai	50
Area 5	Player vs Player	99
Area 6	Player vs AI	30
Area 7	AI vs AI	99

Pada tabel 5.17 dilakukan pengujian dari hasil tabel pariwise yang ada pada tabel 5.18

Tabel 5.17 Tabel Hasil Uji *Combinatorial Testing* pengaturan area

Nama Area	Tipe Permainan	Timer	Hasil Uji
Area 4	Player vs Player	30	Valid

Area 5	Player vs Ai	50	Valid
Area 6	Ai vs Ai	99	Valid
Area 7	Player vs Player	50	Valid
Area 4	Player vs Ai	99	Valid
Area 5	Ai vs Ai	30	Valid
Area 6	Player vs Player	99	Valid
Area 7	Player vs Ai	30	Valid
Area 4	Ai vs Ai	50	Valid
Area 5	Player vs Player	99	Valid
Area 6	Player vs Ai	30	Valid
Area 7	AI vs AI	99	Valid

Dari Tabel 5.17, didapatkan hasil yang valid dari parameter kasus uji yang diberikan. Maka dapat disimpulkan pengaturan area bekerja dengan baik.

Pada pengaturan karakter terdapat 4 parameter yang akan diuji, yaitu kepala, badan, tangan, dan kaki. Pada tabel 5.18 berisi mengenai tabel kombinatorial dengan parameter kepala yang berisi kepala 1, kepala 2, kepala 3, kepala 4 dan kepala 5.

Tabel 5.18 Tabel Combinatorial Testing pengaturan karakter 1 parameter

Kepala
Kepala 1
Kepala 2
Kepala 3
Kepala 4
Kepala 5

Pada tabel 5.19 ditambahkan parameter kedua yaitu badan yang berisi badan 1, badan 2, badan 3, badan 4, dan badan 5.

Tabel 5.19 Tabel Combinatorial Testing pengaturan karakter 2 parameter

Kepala	Badan
Kepala 1	Badan 1
Kepala 2	Badan 2
Kepala 3	Badan 3
Kepala 4	Badan 4
Kepala 5	Badan 5

Kepala 1	Badan 2
Kepala 2	Badan 3
Kepala 3	Badan 4
Kepala 4	Badan 5
Kepala 5	Badan 1
Kepala 1	Badan 3
Kepala 2	Badan 4
Kepala 3	Badan 5
Kepala 4	Badan 1
Kepala 5	Badan 2
Kepala 1	Badan 4
Kepala 2	Badan 5
Kepala 3	Badan 1
Kepala 4	Badan 2
Kepala 5	Badan 3
Kepala 1	Badan 5
Kepala 2	Badan 1
Kepala 3	Badan 2
Kepala 4	Badan 3
Kepala 5	Badan 4

Pada tabel 5.20 ditambahkan parameter ketiga yaitu tangan yang berisi tangan 1, tangan 2, tangan 3, tangan 4, dan tangan 5.

Tabel 5.20 Tabel *Combinatorial Testing* pengaturan karakter 3 parameter

Kepala	Badan	Tangan
Kepala 1	Badan 1	Tangan 1
Kepala 2	Badan 2	Tangan 2
Kepala 3	Badan 3	Tangan 3
Kepala 4	Badan 4	Tangan 4
Kepala 5	Badan 5	Tangan 5
Kepala 1	Badan 2	Tangan 3
Kepala 2	Badan 3	Tangan 1
Kepala 3	Badan 4	Tangan 5

Kepala 4	Badan 5	Tangan 2
Kepala 5	Badan 1	Tangan 4
Kepala 1	Badan 3	Tangan 2
Kepala 2	Badan 4	Tangan 3
Kepala 3	Badan 5	Tangan 1
Kepala 4	Badan 1	Tangan 5
Kepala 5	Badan 2	Tangan 5
Kepala 1	Badan 4	Tangan 4
Kepala 2	Badan 5	Tangan 4
Kepala 3	Badan 1	Tangan 2
Kepala 4	Badan 2	Tangan 1
Kepala 5	Badan 3	Tangan 3
Kepala 1	Badan 5	Tangan 3
Kepala 2	Badan 1	Tangan 5
Kepala 3	Badan 2	Tangan 4
Kepala 4	Badan 3	Tangan 5
Kepala 5	Badan 4	Tangan 1

Pada tabel 5.21 ditambahkan parameter keempat yaitu kaki yang berisi kaki 1, kaki 2, kaki 3, kaki 4, dan kaki 5.

Tabel 5.21 Tabel *Combinatorial Testing* pengaturan karakter 4 parameter

Kepala	Badan	Tangan	Kaki
Kepala 1	Badan 1	Tangan 1	Kaki 1
Kepala 2	Badan 2	Tangan 2	Kaki 2
Kepala 3	Badan 3	Tangan 3	Kaki 3
Kepala 4	Badan 4	Tangan 4	Kaki 4
Kepala 5	Badan 5	Tangan 5	Kaki 5
Kepala 1	Badan 2	Tangan 3	Kaki 4
Kepala 2	Badan 3	Tangan 1	Kaki 5
Kepala 3	Badan 4	Tangan 5	Kaki 1
Kepala 4	Badan 5	Tangan 2	Kaki 3
Kepala 5	Badan 1	Tangan 4	Kaki 2
Kepala 1	Badan 3	Tangan 2	Kaki 1

Kepala 2	Badan 4	Tangan 3	Kaki 2
Kepala 3	Badan 5	Tangan 1	Kaki 4
Kepala 4	Badan 1	Tangan 5	Kaki 5
Kepala 5	Badan 2	Tangan 5	Kaki 3
Kepala 1	Badan 4	Tangan 4	Kaki 3
Kepala 2	Badan 5	Tangan 4	Kaki 1
Kepala 3	Badan 1	Tangan 2	Kaki 5
Kepala 4	Badan 2	Tangan 1	Kaki 2
Kepala 5	Badan 3	Tangan 3	Kaki 4
Kepala 1	Badan 5	Tangan 3	Kaki 2
Kepala 2	Badan 1	Tangan 5	Kaki 4
Kepala 3	Badan 2	Tangan 4	Kaki 5
Kepala 4	Badan 3	Tangan 5	Kaki 2
Kepala 5	Badan 4	Tangan 1	Kaki 1

Pada tabel 5.22 dilakukan pengujian dari hasil tabel pairwise yang ada pada tabel 5.21

Tabel 5.22 Tabel Hasil Uji *Combinatorial Testing* pengaturan karakter

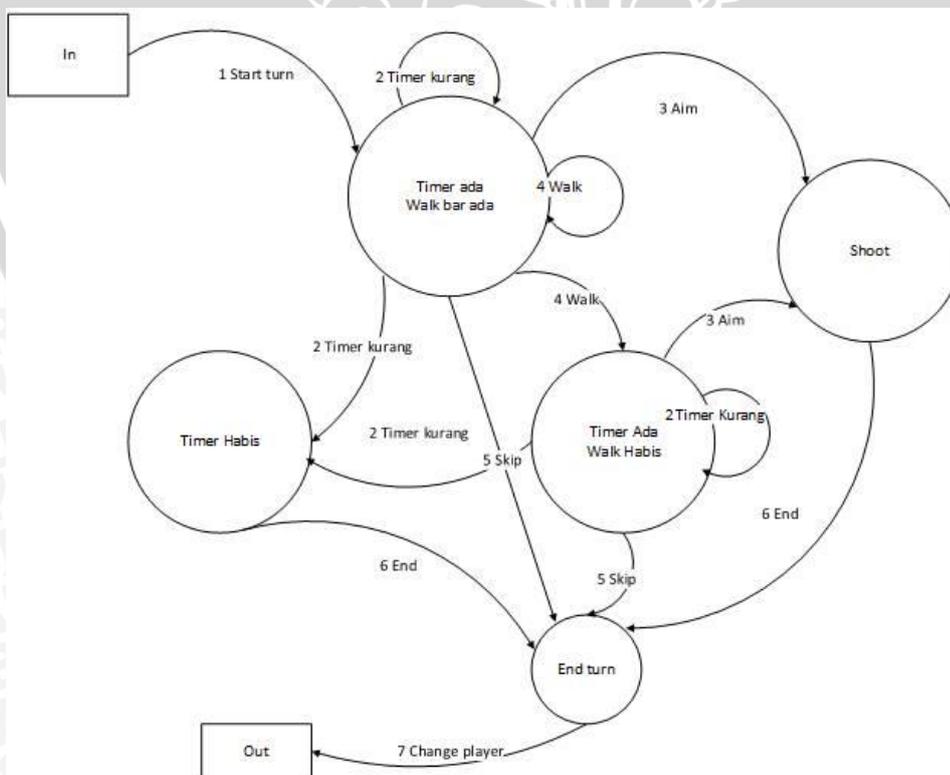
Kepala	Badan	Tangan	Kaki	Hasil Uji
Kepala 1	Badan 1	Tangan 1	Kaki 1	Valid
Kepala 2	Badan 2	Tangan 2	Kaki 2	Valid
Kepala 3	Badan 3	Tangan 3	Kaki 3	Valid
Kepala 4	Badan 4	Tangan 4	Kaki 4	Valid
Kepala 5	Badan 5	Tangan 5	Kaki 5	Valid
Kepala 1	Badan 2	Tangan 3	Kaki 4	Valid
Kepala 2	Badan 3	Tangan 1	Kaki 5	Valid
Kepala 3	Badan 4	Tangan 5	Kaki 1	Valid
Kepala 4	Badan 5	Tangan 2	Kaki 3	Valid
Kepala 5	Badan 1	Tangan 4	Kaki 2	Valid
Kepala 1	Badan 3	Tangan 2	Kaki 1	Valid
Kepala 2	Badan 4	Tangan 3	Kaki 2	Valid
Kepala 3	Badan 5	Tangan 1	Kaki 4	Valid
Kepala 4	Badan 1	Tangan 5	Kaki 5	Valid

Kepala 5	Badan 2	Tangan 5	Kaki 3	Valid
Kepala 1	Badan 4	Tangan 4	Kaki 3	Valid
Kepala 2	Badan 5	Tangan 4	Kaki 1	Valid
Kepala 3	Badan 1	Tangan 2	Kaki 5	Valid
Kepala 4	Badan 2	Tangan 1	Kaki 2	Valid
Kepala 5	Badan 3	Tangan 3	Kaki 4	Valid
Kepala 1	Badan 5	Tangan 3	Kaki 2	Valid
Kepala 2	Badan 1	Tangan 5	Kaki 4	Valid
Kepala 3	Badan 2	Tangan 4	Kaki 5	Valid
Kepala 4	Badan 3	Tangan 5	Kaki 2	Valid
Kepala 5	Badan 4	Tangan 1	Kaki 1	Valid

Dari Tabel pada tabel 5.22, didapatkan hasil yang valid dari parameter kasus uji yang diberikan. Maka dapat disimpulkan pengaturan karakter bekerja dengan baik.

5.2.3 TFD

Test Flow Diagram merupakan pengujian yang mempresentasikan mode grafis dari tingkah laku permainan dari sudut pandang pemain. TFD ditunjukkan pada gambar 5.4



Gambar 5.4 Test Flow Diagram Pergerakan Karakter

Penjelasan mengenai TFD pada gambar 5.4 dijelaskan pada data dictionary yang ada di tabel 5.23

Tabel 5.23 Tabel Data Dictionary TFD Bermain Game

Nama	Keterangan	Yang harus di cek
In	State awal pergantian karakter	
Start Turn	Persiapan awal karakter	Avatar hud menunjukkan karakter yang aktif Daftar senjata menunjukkan senjata karakter yang aktif
Walk	Karakter berjalan	
Timer Ada Walk bar ada	Timer habis Bar untuk berjalan masih ada	Timer HUD menunjukkan angka 0
Timer Berkurang	Angka pada timer berkurang	Bar berjalan masih ada
Timer Habis	Timer habis	Timer HUD menunjukkan angka 0
Timer Ada Walk bar habis	Angka pada timer berkurang Karakter tidak dapat berjalan	Bar berjalan kosong
Skip	Pemain mengakhiri gilirannya tanpa menembak dengan timer masih tersisa	Cek apakah timer HUD menunjukkan angka 0
Aim	Pemain menentukan sudut dan kecepatan sesuai dengan input	Cek apakah sudut dan kecepatan sudah berubah dengan tepat
Shoot	Karakter menembakkan peluru berdasarkan sudut dan kecepatan	Apakah peluru bergerak dengan benar
End Turn	Kondisi akhir dari game untuk menghilangkan karakter yang memiliki Health poin kurang dari 0	Cek apakah masih ada karakter yang memiliki health kurang dari 0
Change Player	Mengubah karakter aktif menjadi karakter selanjutnya	
Out	State akhir karakter	

TFD path dari pergerakan terdapat 6 jalur yaitu :

- Path 1 : 1 – 5 – 7
- Path 2 : 1 – 2 – 6 – 7
- Path 3 : 1 – 4 – 5 – 7
- Path 4 : 1 – 4 – 5 – 2 – 6 – 7
- Path 5 : 1 – 2 – 4 – 3 – 6 – 7
- Path 6 : 1 – 2 – 4 – 2 – 3 – 6 – 7

Pada tabel pada tabel 5.24, dilakukan pengujian terhadap jalur 1 untuk memeriksa apakah sudah berjalan dengan baik.

Tabel 5.24 Tabel kasus Uji Jalur 1 TFD Pergerakan Karakter

Nama	Hasil yang Diharapkan	Hasil yang Didapat
In	Kamera bergerak ke arah karakter yang sedang aktif	Kamera bergerak ke arah karakter yang sedang aktif
Start Turn	Avatar hud menunjukkan karakter yang aktif Daftar senjata menunjukkan senjata karakter yang aktif	Avatar hud menunjukkan karakter yang aktif Daftar senjata menunjukkan senjata karakter yang aktif
Timer Ada Walk Bar Ada	Timer HUD tidak menunjukkan angka 0 Bar untuk berjalan masih tersisa	Timer HUD tidak menunjukkan angka 0 Bar untuk berjalan masih tersisa
Skip	Kondisi berpindah ke kondisi end turn walaupun pemain belum menembak dan angka pada timer HUD masih tersisa	Kondisi berpindah ke kondisi end turn walaupun pemain belum menembak dan angka pada timer HUD masih tersisa
End	Memunculkan perubahan health karakter yang terkena ledakan	Memunculkan perubahan health karakter yang terkena ledakan
End Turn	Karakter dengan health poin kurang dari 0 menghilang	Karakter dengan health poin kurang dari 0 menghilang
Change Player	Karakter aktif berganti menjadi karakter lain	Karakter aktif berganti menjadi karakter lain
Out	Posisi pointer pindah ke arah karakter yang aktif sekarang	Posisi pointer pindah ke arah karakter yang aktif sekarang

Pada tabel 5.24 didapatkan hasil yang sama antara hasil yang didapat dan hasil yang diharapkan, maka dapat diambil kesimpulan bahwa jalur 1 sudah berjalan dengan baik.

Pada tabel pada tabel 5.25, dilakukan pengujian terhadap jalur 2 untuk memeriksa apakah sudah berjalan dengan baik.

Tabel 5.25 Tabel kasus Uji Jalur 2 TFD Pergerakan Karakter

Nama	Hasil yang Diharapkan	Hasil yang Didapat
In	Kamera bergerak ke arah karakter yang sedang aktif	Kamera bergerak ke arah karakter yang sedang aktif
Start Turn	Avatar hud menunjukkan karakter yang aktif Daftar senjata menunjukkan senjata karakter yang aktif	Avatar hud menunjukkan karakter yang aktif Daftar senjata menunjukkan senjata karakter yang aktif
Timer Ada Walk Bar Ada	Timer HUD tidak menunjukkan angka 0 Bar untuk berjalan masih tersisa	Timer HUD tidak menunjukkan angka 0 Bar untuk berjalan masih tersisa
Timer Berkurang	Angka pada timer HUD berkurang	Angka pada timer HUD berkurang
Timer Habis	Angka pada Timer HUD menunjukkan angka 0	Angka pada Timer HUD menunjukkan angka 0
End Turn	Karakter dengan health poin kurang dari 0 menghilang	Karakter dengan health poin kurang dari 0 menghilang
Change Player	Karakter aktif berganti menjadi karakter lain	Karakter aktif berganti menjadi karakter lain
Out	Posisi pointer pindah ke arah karakter yang aktif sekarang	Posisi pointer pindah ke arah karakter yang aktif sekarang

Pada tabel 5.25 didapatkan hasil yang sama antara hasil yang didapat dan hasil yang diharapkan, maka dapat diambil kesimpulan bahwa jalur 2 sudah berjalan dengan baik.

Pada tabel pada tabel 5.26, dilakukan pengujian terhadap jalur 3 untuk memeriksa apakah sudah berjalan dengan baik.

Tabel 5.26 Tabel kasus Uji Jalur 3 TFD Pergerakan Karakter

Nama	Hasil yang Diharapkan	Hasil yang Didapat
In	Kamera bergerak ke arah karakter yang sedang aktif	Kamera bergerak ke arah karakter yang sedang aktif
Start Turn	Avatar hud menunjukkan karakter yang aktif Daftar senjata menunjukkan	Avatar hud menunjukkan karakter yang aktif Daftar senjata menunjukkan

	senjata karakter yang aktif	senjata karakter yang aktif
Timer Ada Walk Bar Ada	Timer HUD tidak menunjukkan angka 0 Bar untuk berjalan masih tersisa	Timer HUD tidak menunjukkan angka 0 Bar untuk berjalan masih tersisa
Walk	Posisi karakter berubah	Posisi karakter berubah
Timer ada Walk habis	Timer HUD tidak menunjukkan angka 0 Bar untuk berjalan tidak tersisa Karakter tidak dapat bergerak	Timer HUD tidak menunjukkan angka 0 Bar untuk berjalan tidak tersisa Karakter tidak dapat bergerak
Skip	Kondisi berpindah ke kondisi end turn walaupun pemain belum menembak dan angka pada timer HUD masih tersisa	Kondisi berpindah ke kondisi end turn walaupun pemain belum menembak dan angka pada timer HUD masih tersisa
End	Memunculkan perubahan health karakter yang terkena ledakan	Memunculkan perubahan health karakter yang terkena ledakan
End Turn	Karakter dengan health poin kurang dari 0 menghilang	Karakter dengan health poin kurang dari 0 menghilang
Change Player	Karakter aktif berganti menjadi karakter lain	Karakter aktif berganti menjadi karakter lain
Out	Posisi pointer pindah ke arah karakter yang aktif sekarang	Posisi pointer pindah ke arah karakter yang aktif sekarang

Pada tabel 5.26 didapatkan hasil yang sama antara hasil yang didapat dan hasil yang diharapkan, maka dapat diambil kesimpulan bahwa jalur 3 sudah berjalan dengan baik.

Pada tabel pada tabel 5.27, dilakukan pengujian terhadap jalur 4 untuk memeriksa apakah sudah berjalan dengan baik.

Tabel 5.27 Tabel kasus Uji Jalur 4 TFD Pergerakan Karakter

Nama	Hasil yang Diharapkan	Hasil yang Didapat
In	Kamera bergerak ke arah karakter yang sedang aktif	Kamera bergerak ke arah karakter yang sedang aktif
Start Turn	Avatar hud menunjukkan karakter yang aktif Daftar senjata menunjukkan senjata karakter yang aktif	Avatar hud menunjukkan karakter yang aktif Daftar senjata menunjukkan senjata karakter yang aktif
Timer Ada	Timer HUD tidak menunjukkan	Timer HUD tidak menunjukkan

Walk Bar Ada	angka 0 Bar untuk berjalan masih tersisa	angka 0 Bar untuk berjalan masih tersisa
Walk	Posisi karakter berubah	Posisi karakter berubah
Timer ada Walk habis	Timer HUD tidak menunjukkan angka 0 Bar untuk berjalan tidak tersisa Karakter tidak dapat bergerak	Timer HUD tidak menunjukkan angka 0 Bar untuk berjalan tidak tersisa Karakter tidak dapat bergerak
Timer Berkurang	Angka pada timer HUD berkurang	Angka pada timer HUD berkurang
Timer Habis	Angka pada Timer HUD menunjukkan angka 0	Angka pada Timer HUD menunjukkan angka 0
End	Memunculkan perubahan health karakter yang terkena ledakan	Memunculkan perubahan health karakter yang terkena ledakan
End Turn	Karakter dengan health poin kurang dari 0 menghilang	Karakter dengan health poin kurang dari 0 menghilang
Change Player	Karakter aktif berganti menjadi karakter lain	Karakter aktif berganti menjadi karakter lain
Out	Posisi pointer pindah ke arah karakter yang aktif sekarang	Posisi pointer pindah ke arah karakter yang aktif sekarang

Pada tabel 5.27 didapatkan hasil yang sama antara hasil yang didapat dan hasil yang diharapkan, maka dapat diambil kesimpulan bahwa jalur 4 sudah berjalan dengan baik.

Pada tabel pada tabel 5.28, dilakukan pengujian terhadap jalur 5 untuk memeriksa apakah sudah berjalan dengan baik.

Tabel 5.28 Tabel kasus Uji Jalur 5 TFD Pergerakan Karakter

Nama	Hasil yang Diharapkan	Hasil yang Didapat
In	Kamera bergerak ke arah karakter yang sedang aktif	Kamera bergerak ke arah karakter yang sedang aktif
Start Turn	Avatar hud menunjukkan karakter yang aktif Daftar senjata menunjukkan senjata karakter yang aktif	Avatar hud menunjukkan karakter yang aktif Daftar senjata menunjukkan senjata karakter yang aktif
Timer Ada Walk Bar Ada	Timer HUD tidak menunjukkan angka 0 Bar untuk berjalan masih tersisa	Timer HUD tidak menunjukkan angka 0 Bar untuk berjalan masih tersisa

Timer Berkurang	Angka pada timer HUD berkurang	Angka pada timer HUD berkurang
Walk	Posisi karakter berubah	Posisi karakter berubah
Aim	Posisi sudut dan kecepatan karakter berubah sesuai dengan inputan pemain	Posisi sudut dan kecepatan karakter berubah sesuai dengan inputan pemain
Shoot	Karakter mengeluarkan projectil	Karakter mengeluarkan projectil
End	Memunculkan perubahan health karakter yang terkena ledakan	Memunculkan perubahan health karakter yang terkena ledakan
End Turn	Karakter dengan health poin kurang dari 0 menghilang	Karakter dengan health poin kurang dari 0 menghilang
Change Player	Karakter aktif berganti menjadi karakter lain	Karakter aktif berganti menjadi karakter lain
Out	Posisi pointer pindah ke arah karakter yang aktif sekarang	Posisi pointer pindah ke arah karakter yang aktif sekarang

Pada tabel 5.28 didapatkan hasil yang sama antara hasil yang didapat dan hasil yang diharapkan, maka dapat diambil kesimpulan bahwa jalur 5 sudah berjalan dengan baik.

Pada tabel pada tabel 5.29, dilakukan pengujian terhadap jalur 6 untuk memeriksa apakah sudah berjalan dengan baik.

Tabel 5.29 Tabel kasus Uji Jalur 6 TFD Pergerakan Karakter

Nama	Hasil yang Diharapkan	Hasil yang Didapat
In	Kamera bergerak ke arah karakter yang sedang aktif	Kamera bergerak ke arah karakter yang sedang aktif
Start Turn	Avatar hud menunjukkan karakter yang aktif Daftar senjata menunjukkan senjata karakter yang aktif	Avatar hud menunjukkan karakter yang aktif Daftar senjata menunjukkan senjata karakter yang aktif
Timer Ada Walk Bar Ada	Timer HUD tidak menunjukkan angka 0 Bar untuk berjalan masih tersisa	Timer HUD tidak menunjukkan angka 0 Bar untuk berjalan masih tersisa
Timer Berkurang	Angka pada timer HUD berkurang	Angka pada timer HUD berkurang
Walk	Posisi karakter berubah	Posisi karakter berubah
Timer ada	Timer HUD tidak menunjukkan	Timer HUD tidak menunjukkan angka 0

Walk Habis	angka 0 Bar untuk berjalan tidak tersisa Karakter tidak dapat bergerak	Bar untuk berjalan tidak tersisa Karakter tidak dapat bergerak
Aim	Posisi sudut dan kecepatan karakter berubah sesuai dengan inputan pemain	Posisi sudut dan kecepatan karakter berubah sesuai dengan inputan pemain
Shoot	Karakter mengeluarkan projectil	Karakter mengeluarkan projectil
End	Memunculkan perubahan health karakter yang terkena ledakan	Memunculkan perubahan health karakter yang terkena ledakan
End Turn	Karakter dengan health poin kurang dari 0 menghilang	Karakter dengan health poin kurang dari 0 menghilang
Change Player	Karakter aktif berganti menjadi karakter lain	Karakter aktif berganti menjadi karakter lain
Out	Posisi pointer pindah ke arah karakter yang aktif sekarang	Posisi pointer pindah ke arah karakter yang aktif sekarang

Pada tabel 5.29 didapatkan hasil yang sama antara hasil yang didapat dan hasil yang diharapkan, maka dapat diambil kesimpulan bahwa jalur 6 sudah berjalan dengan baik.

5.3 Analisa Pengujian

Pada pengujian *whitebox* mengenai perhitungan posisi peluru, perhitungan kecepatan peluru dan navigator menggunakan metode basis path testing, didapatkan hasil valid dari ketiga test yang dilakukan. Dari hasil pengujian ini dapat disimpulkan bahwa algoritma perhitungan dapat bekerja dengan baik.

Selanjutnya pada pengujian *blackbox* mengenai pengaturan permainan, pengaturan area, pengaturan karakter serta pergerakan karakter menggunakan metode *combinatorial testing* dan tfd didapatkan hasil valid dari keempat testing. Dari hasil pengujian ini dapat disimpulkan algoritma dari keempat fitur tersebut dapat bekerja dengan baik.

BAB 6 PENUTUP

6.1 Kesimpulan

1. *Game artillery* pletokan dirancang dengan menggunakan metode rapid iterative prototyping yang meliputi : perancangan *formal elements*, *paper prototyping* dan *playtest*.
2. *Game* pletokan merupakan permainan 2D bergenre *artillery* yang dapat dimainkan sendiri maupun 2 atau lebih pemain.
3. Pada pengujian algoritma *navigator* menggunakan metode *basis path testing* didapatkan hasil valid dari 8 kasus uji yang dilakukan sehingga fitur navigator dapat dikatakan berjalan dengan baik.
4. Berdasarkan hasil pengujian *whitebox* pada kasus uji pengaturan area, pergerakan karakter , dan pengaturan karakter dapat diambil kesimpulan bahwa fungsi dan fitur pada game pletokan berfungsi dengan baik.

6.2 Saran

1. Untuk pengembangan selanjutnya, dapat dtambahkan faktor-faktor yang berpengaruh dalam gerak parabolik seperti arah angin, berat benda dan bentuk benda.
2. *Game* dapat ditingkatkan lagi dengan membuat AI menjadi lebih kompleks dan mampu mengambil keputusan layaknya pemain manusia.
3. Gerak lain seperti gerak melingkat dan gerak jatuh bebas dapat ditambahkan kedalam game untuk memperbanyak unsure edukasi yang didapatkan oleh pemain.
4. *Game* dapat ditambahkan sistem penilaian untuk memonitor perkembangan materi yang didapat oleh pemain.

DAFTAR PUSTAKA

- Aeni, Wiwik Akhirul, 2009. *Antara Game, Pendidikan dan HP (Game Mobile Learning Sebagai Wacana Pendidikan) (Online)* (<http://m-edukasi.kemdikbud.go.id/artikel-mobile-learning-isi.php?kodenya=2009-ac>, diakses 2 September 2015)
- Ahl, David H., 1979. *More Computer Games*. New York : Workman Publishing.
- Haryadi, Bambang, 2009. *Fisika Untuk SMA / MA Kelas XI*. Jakarta: Pusat Perbukuan Departemen Pendidikan Nasional.
- Hocking, Joseph, 2015. *Unity in Action Multiplatform game development in C#*. Manning Prouduction.
- Michael, David dan Sande Chen, 2006. *Serious Games: Games that educate, Train, and Inform*. Canada: Thomson Course Technology PTR.
- Widodo, Tri, 2009. *Fisika : Untuk SMA / MA Kelas XI*. Jakarta: Pusat Perbukuan Departemen Pendidikan Nasional.
- Rogers, Scott, 2010. *Level Up! The Guide to Great Video Games*. West Sussex: John Willey & Sons, Ltd.
- Sumarno, Joko, 2009. *Fisika Untuk SMA / MA Kelas X*. Jakarta: Pusat Perbukuan Departemen Pendidikan Nasional.
- Fullerton, Tracy, 2008. *Game Design Workshop a Playecentric Approach to Creating Innovative Games*. Amsterdam: Elsevier Inc.

LAMPIRAN A How to Play

1. Tekan arah panah kanan atau kiri pada keyboard untuk menggerakkan karakter



2. Tekan dan tahan spasi pada keyboard untuk menahan tembakan, dan lepas untuk melepaskan tembakan



3. Untuk menentukan target, arahkan pointer ke objek atau nama karakter yang terletak pada daftar karakter, dan tekan klik kiri mouse
4. Untuk menghapus target, tekan klik kanan pada mouse
5. Untuk melihat area sekitar, arahkan pointer ke ujung layar

