

KATA PENGANTAR

Dengan nama Allah SWT Yang Maha Pengasih dan Penyayang. Segala puji bagi Allah SWT karena atas rahmat dan hidayahNya-lah penulis dapat menyelesaikan Skripsi yang berjudul “Rancang Bangun Permainan Karapan Sapi 3D”. Shalawat dan salam atas junjungan besar kita Nabi Muhammad S.A.W. beserta keluarga dan para sahabat sekalian. Skripsi ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer di Program Studi Teknik Informatika Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya Malang.

Melalui kesempatan ini, penulis ingin menyampaikan rasa hormat dan terima kasih penulis yang sebesar-besarnya kepada semua pihak yang telah memberikan bantuan – bantuan baik lahir maupun batin selama penulisan tugas akhir ini. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa hormat dan terima kasih penulis kepada :

1. Ayahanda Eko Kamari, Ibunda April Kristiwati, S.Pd, dan seluruh keluarga besar atas segala nasehat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikannya skripsi 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, M.T dan Bapak Issa Arwani, S.Kom, M.Sc selaku Ketua dan Sekretaris Program Studi Teknik Informatika Universitas Brawijaya.
4. Bapak Eriq Muhammad Adams, S.T, M.Kom dan Bapak Wibisono Sukmo Wardhono, S.T, M.T selaku dosen pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi.
5. Bapak Suprpto, S.T, M.T selaku dosen penasehat akademik yang selalu memberikan nasehat kepada penulis selama menempuh masa studi.
6. Seluruh Dosen Teknik Informatika Universitas Brawijaya atas kesediaan membagi ilmunya kepada penulis.

7. Seluruh Civitas Akademika Teknik Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Teknik Informatika Universitas Brawijaya dan selama penyelesaian skripsi.
8. Sahabat-sahabat *founder* Exolion Game Studio, Aldian Lodewyk Manuputty, Kukuh Heru Irawan, dan Wisnu Aditya yang selalu bekerja bersama dalam menyelesaikan skripsi.
9. Sahabat-sahabatku Angkatan 2007 Teknik Informatika, The Legendary TPL, terimakasih atas segala bantuannya selama menempuh studi di Teknik Informatika Universitas Brawijaya.
10. Adik Ari Agustina dan Keluarga yang telah menyediakan tempat selama proses pengerjaan skripsi.
11. Semua sahabatku di kota Batu dan Pasuruan atas doa, dukungan dan semangat yang selalu diberikan selama penulis menempuh studi di Teknik Informatika Universitas Brawijaya.
12. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun yang tidak langsung demi terselesaikannya skripsi ini.

Hanya doa yang bisa penulis berikan semoga Allah SWT memberikan pahala serta balasan kebaikan yang berlipat. Penulis menyadari bahwa skripsi ini masih banyak kekurangan dan masih jauh dari sempurna. Untuk itu, saran dan kritik yang membangun sangat penulis harapkan. Semoga skripsi ini membawa manfaat bagi penyusun maupun pihak lain yang menggunakannya.

Malang, 09 Januari 2013

Penulis

ABSTRAK

Febrian Aris Pradika. 2013. : Rancang Bangun Permainan Karapan Sapi 3D. Skripsi Program Studi Teknik Informatika, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.

Dosen Pembimbing : Eriq Muhammad Adams, S.T, M.Kom dan Wibisono Sukmo Wardhono, S.T, M.T.

Game dengan tema kebudayaan sangat jarang ditemukan. Indonesia memiliki berbagai macam jenis budaya, salah satunya adalah perlombaan karapan sapi. Budaya tersebut tentunya menarik untuk diangkat menjadi tema sebuah *game*. Karapan Sapi 3D merupakan sebuah *game* yang dibuat dengan latar belakang kebudayaan dan bertujuan selain mendapatkan hiburan, juga memberikan pendidikan tentang kebudayaan yang ada di Indonesia. *Game* ini dijalankan pada *platform* PC (*Personal Computer*) yang menggunakan Sistem Operasi Windows. *Game* ini dirancang berdasarkan *gameplay* yang dibuat pada *Game Design* dan *Technical Design* sebagai bahasa pemodelan (UML). *Game Design* bertujuan untuk membuat dokumentasi pembuatan *game* secara formal dan sesuai format agar tidak keluar batasan *gameplay* yang telah dirancang. *Game* ini diimplementasikan dengan bahasa pemrograman JAVA dan menggunakan jMonkeyPlatform *game engine*. Pada pengujian unit, disimpulkan bahwa unit modul dari *game* sudah memenuhi kebutuhan fungsional yang telah dirancang pada tahap perancangan. Pada pengujian validasi dapat disimpulkan bahwa implementasi dan fungsionalitas *game* Karapan Sapi 3D telah memenuhi kebutuhan yang telah dijabarkan pada tahap analisis kebutuhan. Pada hasil *play testing* didapatkan *feedback* dari *user* bahwa *game* telah dianggap menyenangkan untuk dimainkan. Hasil pengujian performa menunjukkan bahwa kualitas terbaik *frame-per-second* dalam *game* dapat diperoleh dengan menggunakan komputer atau *laptop* yang memenuhi atau di atas spesifikasi minimum.

Kata Kunci : *Game*, Karapan Sapi 3D, *Game Design*, *Technical Design*, *Frame-per-Second*.

ABSTRACT

Febrian Aris Pradika. 2013. : *Developing Karapan Sapi 3D Game.*

Advisor : Eriq Muhammad Adams, S.T, M.Kom dan Wibisono Sukmo Wardhono, S.T, M.T.

Games with the theme of culture are rare. Indonesia has many kind of culture, one of which is Karapan Sapi (Bull Race). That Culture is certainly exciting to be appointed as a gaming theme. Karapan Sapi 3D is a game created by the cultural background and aims not only for entertainment, but also provides education about the culture in Indonesia. The game is run on a PC platform (Personal Computer) using Windows Operating System. This game is designed based on gameplay that made on the Game Design and Technical Design as a modeling language (UML). Game Design aims to make game development a formal documentation and appropriate format to keep it out of gameplay restrictions that have been designed. The game is implemented with JAVA programming language and uses the game engine jMonkeyPlatform. In unit testing, it was concluded that the unit module of the game is to meet the functional requirements that have been designed at the design stage. In validation testing can be concluded that the Karapan Sapi 3D implementation and game functionality meets the needs that have been described in the requirement analysis stage. In play testing result obtained feedback from user that game is considered fun to play. The performance test results indicate that the frame-per-second best quality in the game can be obtained by using a computer or laptop that meets or above the minimum specs.

Keywords : Game, Karapan Sapi 3D, Game Design, Technical Design, Frame-per-Second.

DAFTAR ISI

KATA PENGANTAR	i
ABSTRAK	iii
ABSTRACT	iv
DAFTAR ISI	v
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan	3
1.5 Manfaat	3
1.6 Sistematika Penulisan	4
BAB II DASAR TEORI	5
2.1 <i>Game</i> (Permainan)	5
2.1.1 <i>Genre</i> (Klasifikasi) <i>Game</i>	6
2.1.2 <i>Game Platform</i>	8
2.1.3 <i>Game Rating Categories</i>	8
2.2 <i>Game Engine</i>	9
2.2.1 <i>jMonkeyEngine</i> (<i>jMonkeyPlatform Game Engine</i>)	10
2.2.2 <i>jMonkeyPlatform Structure</i>	11
2.3 Karapan Sapi	12
2.4 Teori Pengujian	13
2.4.1 Pengujian Terstruktur (<i>Whitebox Testing</i>)	13
2.4.2 Pengujian Fungsionalitas (<i>Blackbox Testing</i>)	14
2.4.3 <i>Play Testing</i>	15
2.4.4 Pengujian Performa (<i>Performance Testing</i>)	15
BAB III METODOLOGI PENELITIAN	17
3.1 Studi Literatur	18
3.2 Perancangan <i>Game</i> Karapan Sapi 3D	18

3.2.1	<i>Game Design</i>	18
3.2.1.1	<i>One-Sheet Document</i>	19
3.2.1.2	<i>Ten-Pager Document</i>	20
3.2.2	<i>Technical Design</i>	23
3.3	<i>Implementasi Game (Game Implementation)</i>	24
3.4	<i>Pengujian Game (Game Testing)</i>	24
3.4.1	<i>Whitebox Testing</i>	24
3.4.1.1	<i>Unit Testing</i>	24
3.4.1.2	<i>Integration Testing</i>	24
3.4.2	<i>Blackbox Testing</i>	24
3.4.3	<i>Play Testing</i>	25
3.4.4	<i>Performance Testing</i>	25
BAB IV PERANCANGAN		26
4.1	<i>Game Design</i>	26
4.1.1	<i>Game Title</i>	26
4.1.2	<i>Game Goal</i>	27
4.1.3	<i>Game Outline</i>	27
4.1.4	<i>Karakter</i>	28
4.1.4.1	<i>Profil Karakter</i>	28
4.1.5	<i>Gerakan Karakter</i>	30
4.1.6	<i>Klasifikasi Karakter</i>	31
4.1.6.1	<i>Karakter Utama (Main Character)</i>	31
4.1.6.2	<i>Karakter Figuran (Side Character)</i>	32
4.1.7	<i>Environment</i>	32
4.1.8	<i>Kontrol Game</i>	33
4.1.8.1	<i>Maju</i>	33
4.1.8.2	<i>Samping Kanan dan Kiri</i>	34
4.1.8.3	<i>Kontrol Gerakan Brake (Mengerem)</i>	34
4.1.9	<i>Interface Game</i>	34
4.1.9.1	<i>Menu Utama</i>	34
4.1.9.2	<i>Choose Character (Pilih Karakter)</i>	35
4.1.9.3	<i>Choose Level (Pilih Level)</i>	36

4.1.9.4	<i>Interface Game</i>	36
4.1.10	<i>Level Game</i>	37
4.1.10.1	<i>Deskripsi Level</i>	37
4.1.10.2	<i>Level Background</i>	37
4.1.11	<i>Sistem Pada Game</i>	39
4.1.11.1	<i>Health Points dan Speed Indicator</i>	39
4.1.11.2	<i>Rintang dan Mystery Box Collision</i>	39
4.1.11.3	<i>Rekor dan Scoring System</i>	39
4.1.11.4	<i>Sudut Pandang Kamera</i>	40
4.1.11.5	<i>Musik dan Sound Effect (Audio)</i>	40
4.2	<i>Technical Design</i>	40
4.2.1	<i>Use Case Diagram</i>	41
4.2.1.1	<i>Identifikasi Aktor</i>	41
4.2.1.2	<i>Daftar Kebutuhan</i>	41
4.2.2	<i>Class Diagram</i>	43
4.2.3	<i>Activity Diagram</i>	44
4.2.3.1	<i>Activity Diagram Start</i>	46
4.2.3.2	<i>Activity Diagram Kontrol Karakter</i>	47
4.2.3.3	<i>Activity Diagram Mencatat High Score</i>	48
4.2.3.4	<i>Activity Diagram Lihat High Score</i>	48
4.2.3.5	<i>Activity Diagram Lihat Credit</i>	49
4.2.3.6	<i>Activity Diagram Exit</i>	50
BAB V	IMPLEMENTASI	51
5.1	<i>Spesifikasi Sistem</i>	51
5.1.1	<i>Spesifikasi Perangkat Keras</i>	51
5.1.2	<i>Spesifikasi Perangkat Lunak</i>	51
5.2	<i>Batasan-Batasan Implementasi</i>	52
5.3	<i>Implementasi Artwork</i>	52
5.3.1	<i>Implementasi Game Cover</i>	52
5.3.2	<i>Implementasi Karakter</i>	52
5.3.3	<i>Implementasi Environment</i>	54
5.4	<i>Implementasi Level</i>	54

5.4.1	Implementasi Level 1	54
5.4.2	Implementasi Level 2	55
5.4.3	Implementasi Level 3	56
5.5	Implementasi <i>Procedure</i>	56
5.5.1	Implementasi <i>Procedure</i> Scoring()	56
5.5.2	Implementasi <i>Procedure</i> setSuperCow()	57
5.5.3	Implementasi <i>Procedure</i> bonusVelocity()	58
5.5.4	Implementasi <i>Procedure</i> updateHealth()	58
5.5.5	Implementasi <i>Procedure</i> updateGameCollision()	59
5.6	Implementasi Antarmuka	60
5.6.1	Implementasi <i>Main Menu</i>	61
5.6.2	Implementasi Sub-menu <i>Choose Character</i>	61
5.6.3	Implementasi Sub-menu <i>Choose Level</i>	61
5.6.4	Implementasi <i>In-Game Screen</i>	61
BAB VI PENGUJIAN DAN ANALISIS		64
6.1	Pengujian	64
6.1.1	Pengujian <i>Whitebox</i>	64
6.1.1.1	Pengujian Unit <i>Procedure</i> Scoring()	64
6.1.1.2	Pengujian Unit <i>Procedure</i> setSuperCow()	67
6.1.1.3	Pengujian Unit <i>Procedure</i> bonusVelocity()	69
6.1.1.4	Pengujian Unit <i>Procedure</i> updateHealth()	71
6.1.1.5	Pengujian Integrasi <i>Procedure</i> updateGameCollision()	72
6.1.2	Pengujian <i>Blackbox</i>	75
6.1.2.1	Kasus Uji Mulai <i>Game</i>	76
6.1.2.2	Kasus Uji Pilih Karakter	76
6.1.2.3	Kasus Uji Pilih Level	77
6.1.2.4	Kasus Uji Mulai Bermain	78
6.1.2.5	Kasus Uji Bergerak	78
6.1.2.6	Kasus Uji <i>Game State</i>	79
6.1.2.7	Kasus Uji Melihat <i>High Score</i>	80
6.1.2.8	Kasus Uji Mulai Melihat <i>Credit</i>	80
6.1.2.9	Kasus Uji <i>Exit Game</i>	81

6.1.3	<i>Play Testing</i>	81
6.1.4	Pengujian Performa.....	82
6.2	Analisis.....	84
6.2.1	Analisis Pengujian <i>Whitebox</i>	84
6.2.1.1	Analisis Pengujian Unit.....	85
6.2.1.2	Analisis Pengujian Intergrasi.....	85
6.2.2	Analisis Pengujian <i>Blackbox</i>	85
6.2.3	Analisis <i>Play Testing</i>	85
6.2.4	Analisis Pengujian Performa.....	86
BAB VI PENUTUP		87
7.1	Kesimpulan.....	87
7.2	Saran.....	87
DAFTAR PUSTAKA		89



DAFTAR GAMBAR

Gambar 2.1 Tampilan <i>Game</i> 3D Battlezone, Akari.....	5
Gambar 2.2 <i>Game</i> , <i>Publisher</i> dan <i>Genre</i>	7
Gambar 2.3 jMonkeyPlatform Alpha-4.	10
Gambar 2.4 jMonkeyPlatform <i>Game Engine Structure</i>	11
Gambar 2.5 Perlombaan Karapan Sapi di Pulau Madura.	12
Gambar 2.6 <i>Flow Graph</i>	14
Gambar 2.7 <i>Blackbox Testing</i>	15
Gambar 3.1 Diagram Alir Runtutan Pengerjaan Penelitian.....	17
Gambar 3.2 <i>Use Case Diagram</i> Karapan Sapi 3D.....	23
Gambar 4.1 Perancangan <i>Game Title</i>	26
Gambar 4.2 Prototype <i>Game</i> Karapan Sapi 3D	28
Gambar 4.3 Rancangan Menu Utama	34
Gambar 4.4 Rancangan Sub-menu <i>Choose Character</i>	35
Gambar 4.5 Rancangan Sub-menu <i>Choose Level</i>	36
Gambar 4.6 Rancangan <i>Interface Game</i>	37
Gambar 4.7 <i>Class Diagram</i> <i>Game</i> Karapan Sapi 3D	45
Gambar 4.8 <i>Use Case Game</i> Karapan Sapi 3D.....	46
Gambar 4.9 <i>Activity Diagram Start</i>	46
Gambar 4.10 <i>Activity Diagram</i> Kontrol Karakter.....	47
Gambar 4.11 <i>Activity Diagram</i> Mencatat <i>High Score</i>	48
Gambar 4.12 <i>Activity Diagram</i> Lihat <i>High Score</i>	49
Gambar 4.13 <i>Activity Diagram</i> Lihat <i>Credit</i>	49
Gambar 4.14 <i>Activity Diagram Exit</i>	50
Gambar 5.1 <i>Game Cover</i> Karapan Sapi 3D	53
Gambar 5.2 Implementasi Karakter <i>Game</i> Karapan Sapi 3D.....	53
Gambar 5.3 <i>Environtment Level</i> Karapan Sapi 3D	54
Gambar 5.4 Implementasi Level 1	55
Gambar 5.5 Implementasi Level 2	55
Gambar 5.6 Implementasi Level 3	56
Gambar 5.7 Menu Utama <i>Game</i> Karapan Sapi 3D.....	62
Gambar 5.8 Menu <i>Choose Character</i>	62
Gambar 5.9 Menu <i>Choose Level</i>	63

Gambar 5.10 <i>Interface In-Game Karapan Sapi 3D</i>	63
Gambar 6.1 <i>Flowgraph Procedure Scoring()</i>	66
Gambar 6.2 <i>Flowgraph Procedure setSuperCow()</i>	68
Gambar 6.3 <i>Flowgraph Procedure bonusVelocity()</i>	70
Gambar 6.4 <i>Flowgraph Procedure updateHealth()</i>	71
Gambar 6.5 <i>Flowgraph Procedure updateGameCollision()</i>	74



DAFTAR TABEL

Tabel 2.1 <i>Game Rating Categories</i>	8
Tabel 3.1 <i>One-Sheet Document</i>	19
Tabel 4.1 Profil Karakter Brown.....	28
Tabel 4.2 Profil Karakter White.....	29
Tabel 4.3 Profil karakter Baldy.....	29
Tabel 4.4 Gerakan Lari	30
Tabel 4.5 Gerakan Lompat.....	30
Tabel 4.6 Gerakan <i>Stand (idle)</i>	31
Tabel 4.7 Gerakan Supercow!.....	31
Tabel 4.8 Keterangan Objek	32
Tabel 4.9 Kontrol <i>Game</i>	33
Tabel 4.10 Level <i>Background</i>	38
Tabel 4.11 Sistem Penilaian.....	40
Tabel 4.12 Identifikasi Aktor.....	41
Tabel 4.13 Daftar kebutuhan.....	41
Tabel 4.14 Deskripsi <i>class diagram</i> Karapan Sapi 3D	43
Tabel 5.1 <i>Procedure Scoring()</i>	57
Tabel 5.2 <i>Procedure setSuperCow()</i>	58
Tabel 5.3 <i>Procedure bonusVelocity()</i>	59
Tabel 5.4 <i>Procedure updateHealth()</i>	59
Tabel 5.5 <i>Procedure updateGameCollision()</i>	60
Tabel 6.1 Pengujian <i>Procedure Scoring()</i>	65
Tabel 6.2 Test Case Pengujian <i>Procedure Scoring(gameTime, minute, second, curHealth, level)</i>	66
Tabel 6.3 Pengujian <i>Procedure setSuperCow()</i>	67
Tabel 6.4 Test Case Pengujian <i>Procedure setSuperCow(boolean supercow)</i>	68
Tabel 6.5 Pengujian <i>Procedure bonusVelocity()</i>	69

Tabel 6.6 Test Case Pengujian <i>Procedure</i> bonusVelocity(float bVelocity).....	70
Tabel 6.7 Pengujian <i>Procedure</i> updateHealth()	71
Tabel 6.8 Test Case Pengujian <i>Procedure</i> updateHealth(float updateHealth)	72
Tabel 6.9 Pengujian <i>Procedure</i> updateGameCollision().....	73
Tabel 6.10 Test Case Pengujian <i>Procedure</i> updateGameCollision(PhysicsCollisionEvent event)	74
Tabel 6.11 Test Case Mulai <i>Game</i>	76
Tabel 6.12 Test Case Pilih Karakter	76
Tabel 6.13 Test Case Untuk Pilih Level	77
Tabel 6.14 Test Case Untuk Mulai Bermain.....	78
Tabel 6.15 Test Case Untuk Bergerak	78
Tabel 6.16 Test Case <i>Game State</i>	79
Tabel 6.17 Test Case Untuk Melihat <i>High Score</i>	80
Tabel 6.18 Test Case Melihat <i>Credit</i>	80
Tabel 6.19 Test Case Untuk <i>Exit Game</i>	81
Tabel 6.20 Hasil Kuisisioner <i>Play Testing</i>	82
Tabel 6.21 Standar <i>Frame-per-Second</i>	83
Tabel 6.22 Pengujian Performa FPS Karapan Sapi 3D.....	83

BAB I

PENDAHULUAN

1.1 Latar Belakang

Game, atau dalam bahasa Indonesia disebut permainan, adalah setiap kegiatan yang dilakukan hanya untuk kesenangan [KWG-00]. Tujuan dari *game* adalah untuk mendapatkan hiburan, pendidikan, simulasi, olah raga dan masih banyak tujuan lainnya yang dapat diperoleh. Dalam *game*, pemain dapat berinteraksi dengan sebuah sistem yang dibuat dalam *game* atau biasa disebut dengan istilah *gameplay*. *Gameplay* adalah strategi yang diperlukan untuk mendapatkan atau mencapai sebuah kondisi akhir (*end-point*) *game*. Strategi yang digunakan untuk mencapai kondisi akhir harus memiliki efek positif dan negatif. Jika hanya ada salah satu, maka akan tercipta sebuah kondisi *imbalance* dalam *game* [ROL-04].

Game berkembang sejalan dengan perkembangan teknologi dan memasuki era *digital* dari *game* 2D (dua dimensi) sampai *game* 3D (tiga dimensi), mulai dari *computer*, *console* bahkan pada *mobile device*. Akan tetapi, tujuan dari dibuatnya sebuah *game* hanya sebagai sarana hiburan sehingga jarang ditemukan sebuah *game* yang mengangkat tema kebudayaan atau pendidikan. *Game* yang dibuat dan dikembangkan saat ini kebanyakan merupakan *game* yang bertema *sci-fi* (*science-fiction*) atau imajinasi.

Game 3D dapat dikembangkan dengan menggunakan *game engine* dengan tujuan untuk memudahkan pengembangan sebuah *game*. *Game engine* yang telah ada salah satunya adalah jMonkeyEngine. jMonkeyEngine adalah *open source* 3D *game framework* yang berbasis bahasa pemrograman java [JME-11]. *Developer game* yang menggunakan jMonkeyEngine sebagai *game engine* mereka untuk mengembangkan sebuah *game* 3D masih sangat sedikit meskipun jMonkeyEngine memberikan fitur-fitur untuk pengembangan sebuah *game* 3D yang tidak kalah dengan *game engine* berlisensi.

Indonesia merupakan negara dengan berbagai macam bentuk dan jenis budaya, salah satu contoh kebudayaan yang dapat diangkat menjadi tema sebuah *game* adalah Karapan Sapi (*Bull Race*). Sampai saat ini, masih belum terdapat *game* kebudayaan yang mengangkat perlombaan karapan sapi sebagai tema sebuah *game* 3D menggunakan *platform* PC. Sedangkan *game* karapan sapi pada *platform* selain PC, kebanyakan mensimulasikan perlombaan karapan sapi sebenarnya tanpa adanya fitur-fitur tambahan, sehingga *game* terkesan monoton dan membosankan.

Game selain digunakan untuk mendapatkan hiburan, juga dapat digunakan sebagai sarana pembelajaran dan pendidikan. Berdasarkan kebutuhan akan *game* bertema kebudayaan dan pendidikan yang dapat digunakan sebagai media pembelajaran akan kekayaan budaya sehingga kebudayaan tetap terlestarikan dan belum adanya *game* yang mengangkat perlombaan karapan sapi menjadi sebuah *game* 3D menggunakan *platform* PC, maka penulis mengambil judul penelitian **“Rancang Bangun Permainan Karapan Sapi 3D”** dengan tujuan *game* yang bertemakan kebudayaan ini dapat dijadikan sebuah sarana pendidikan sekaligus hiburan dan dapat digunakan oleh masyarakat luas sehingga dapat melestarikan salah satu dari banyak kebudayaan yang terdapat di Indonesia dan memberi tahu dunia bahwa kebudayaan tersebut asli Indonesia.

1.2 Rumusan Masalah

Berdasarkan uraian dan permasalahan yang diangkat pada bagian latar belakang di atas, maka dapat dirumuskan permasalahannya yaitu:

1. Penjelasan alur cerita (*storyline*) dan *gameplay* pada *game* Karapan Sapi 3D.
2. Pembahasan perancangan *game* Karapan Sapi 3D pada *platform* PC berdasarkan *storyline* dan *gameplay*.
3. Pembahasan implementasi *game* Karapan Sapi 3D pada *platform* PC dengan *game engine* jMonkeyEngine yang menggunakan bahasa pemrograman Java.

4. Pembahasan skenario, proses, dan hasil pengujian berdasarkan struktur, fungsionalitas, dan performa pada *game* Karapan Sapi 3D pada *platform* PC.

1.3 Batasan Masalah

Agar permasalahan yang dirumuskan lebih terfokus, maka penelitian ini dibatasi dalam hal:

1. *Genre game* adalah *arcade-racing game*.
2. *Game-mode* Karapan Sapi 3D adalah *time trial single-player*.
3. *Point of View* (PoV) *game* ini adalah *third-person*.
4. *Frame rate testing* dengan metode *benchmark* digunakan pada pengujian performa.

1.4 Tujuan

Tujuan penelitian ini sesuai dengan latar belakang dan rumusan masalah adalah untuk membangun *game* Karapan Sapi 3D.

1.5 Manfaat

Manfaat dari penelitian ini antara lain:

- a. Bagi penulis
 1. Mengaplikasikan ilmu yang didapat selama mengikuti perkuliahan di Teknik Informatika Universitas Brawijaya.
 2. Mendapatkan pemahaman tentang perancangan dan pengembangan *game* pada *platform* PC dengan menggunakan *game engine* *jMonkeyEngine*.
- b. Bagi pengguna
 1. Menyediakan sarana hiburan dan pendidikan secara bersamaan pada *game* ber-*platform* PC.
 2. Menyediakan *game* yang menggunakan objek dari kebudayaan Indonesia.

1.6 Sistematika Penulisan

Sistematika penulisan laporan penelitian ini adalah:

Bab I Pendahuluan

Berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

Bab II Dasar Teori

Berisi tentang dasar teori secara luas mengenai *game*, *software* maupun *hardware* yang diperlukan untuk perancangan *game* Karapan Sapi 3D.

Bab III Metode Penelitian

Berisi tentang langkah-langkah dalam merancang *game* dan *gameplay* Karapan Sapi 3D yang dapat dijalankan pada *platform Personal Computer* (PC). Pada bab ini dijelaskan langkah-langkah yang akan dilakukan yaitu perumusan masalah, studi literatur, perancangan aplikasi *game* Karapan Sapi 3D, implementasi, dan pengujian. Kesimpulan dan saran disertakan sebagai catatan atas aplikasi dan kemungkinan arah pengembangan aplikasi selanjutnya.

Bab IV Perancangan dan Realisasi Sistem

Bab ini berisi tentang perencanaan *game* yang dibuat, meliputi deskripsi *game*, spesifikasi kebutuhan, perancangan *game*, perancangan aplikasi *game* dengan *platform* PC.

Bab V Implementasi

Bab ini berisi tentang implementasi *game* yang dibangun, meliputi pembuatan aplikasi *game* pada *platform* PC menggunakan *game engine* jMonkeyEngine.

Bab VI Pengujian dan Analisis

Berisi mengenai analisa hasil pengujian dari perancangan *game*, hasil analisa output *game*, pembahasan terjadinya *bug* dan kegagalan (apabila terjadi kegagalan).

Bab VII Kesimpulan dan Saran

Pada bab ini berisi kesimpulan yang diambil berdasarkan analisa hal-hal penting, meliputi keunikan, kelebihan atau kekurangan, serta saran-saran untuk penyempurnaan dan pengembangan dari *game* yang dibuat.

BAB II

DASAR TEORI

2.1 *Game* (Permainan)

Game, atau dalam bahasa Indonesia disebut permainan, adalah setiap kegiatan yang dilakukan hanya untuk kesenangan yang memiliki tujuan untuk mendapatkan hiburan, pendidikan, simulasi, olah raga, dan lain-lain. *Video Game* pertama kali diciptakan oleh Will Higinbotham, seorang fisikawan yang bekerja untuk Brookhaven National Laboratory, yang diberi nama “Tennis for Two” pada tahun 1958. *Game* tersebut merupakan *game* tenis meja yang dijalankan pada sebuah *oscilloscope* [EDG-06].

Game berkembang secara drastis pada tahun 1980. Perusahaan *game* Atari berhasil menciptakan *video game* 3D pertama yang memberikan tampilan objek gunung, bulan, dan berbagai rintangan yang digunakan sebagai *background* dan *environment* yang diberi nama “Battlezone”. Battlezone adalah *game* bertema militer dimana pemain dapat mengontrol tank untuk menghancurkan tank musuh. Battlezone menggunakan *Point of View* (sudut pandang) *1st-person* dan memiliki *cross-hair* seperti teropong agar seolah-olah pemain berada di dalam tank tersebut seperti ditunjukkan pada Gambar 2.1[EDG-06].



Gambar 2.1 Tampilan *Game* 3D Battlezone, Akari
Sumber: [EDG-06].

2.1.1 Genre (Klasifikasi) Game

Genre merupakan kata dari bahasa Perancis yang memiliki arti *type* dalam bahasa Inggris atau dalam bahasa Indonesia adalah tipe dari *game*. *Genre* dapat diklasifikasikan menjadi beberapa jenis *genre* inti (*singular core genre*), dapat digabungkan antara satu *genre* inti dengan yang lain, dan pada *genre* inti masih dapat dibagi menjadi *sub-genre* [KEV-04]. Menurut Kevin Oxland, berikut ini adalah klasifikasi *genre* inti beserta penjelasannya dan contoh *game* beserta *genre* yang telah ada dipasaran pada Gambar 2.2:

1. *Sports* : *Game* dengan tema olah raga seperti sepak bola, tenis, bola basket, dan lain-lain.
2. *Adventure* : *Game* dengan tema petualangan, sering memiliki cerita interaktif yang akan terpecahkan selama petualangan berlangsung untuk mengatasi rintangan, penuh dengan misteri, dan memiliki banyak plot.
3. *Action* : *Game* dengan banyak aksi, membutuhkan sedikit proses berfikir, lebih mengandalkan reflek dan reaksi yang cepat, dan pemecahan masalah yang sederhana.
4. *Simulation* : *Game* yang mensimulasikan dan dibuat semirip mungkin dengan kegiatan nyata kedalam *game*, biasanya digunakan sebagai proses pembelajaran atau proses pengetesan.
5. *Strategy* : *Game* yang pada umumnya memberikan tantangan untuk mendapatkan sumber daya, membangun unit, kemudian mengatur siasat dalam bertahan atau menyerang untuk menghancurkan dan mengalahkan musuh.
6. *Puzzle* : *Game* yang didesain untuk memberikan tantangan kepada pemain untuk menemukan solusi agar masalah dapat terpecahkan.
7. *Role-play* : *Game* yang memiliki kebebasan dalam menentukan peran yang akan diambil dalam menyelesaikan *game*. Bahkan memberikan kebebasan dalam mengatur karakter seperti tampilan fisik, *gender*, ras, *job*, dan sebagainya. *Genre* ini disebut MMORPG bila dimainkan secara bersamaan dengan pemain lain.
8. *Management* : *Game* dengan *genre* management memungkinkan pemain untuk membangun sebuah kota atau rumah, kemudian mengisinya dengan

penduduk. *Game* semacam Football Manager juga dapat dianggap *genre* ini.

9. *Uncategorized*: Jika sebuah *game* tidak dapat dimasukkan kedalam *genre* yang terdapat sebelumnya dan sulit ditentukan *genre game* tersebut, maka dapat dikategorikan sebagai *Uncategorized*. Tetapi desainer *game* lebih cenderung untuk memberikan *genre hybrid* pada *game* yang masuk kedalam *genre Uncategorized*.

Game	Publisher	Genre
<i>Age of Empires III</i>	Microsoft	Strategy
<i>Battlefield 2</i>	Electronic Arts	Action-FPS
<i>Call of Duty 2</i>	Activision	Action-FPS
<i>Civilization IV</i>	2K Games	Strategy
<i>Command & Conquer (various)</i>	Electronic Arts	RTS
<i>Delta Force Xtreme</i>	Novalogic	Action-FPS
<i>Diablo III</i>	Blizzard	RPG
<i>Doom 3</i>	Activision	Action-FPS
<i>Duke Nukem Forever</i>	Gathering of Developers	Action-FPS
<i>Dungeon Siege</i>	Microsoft	Action-RPG
<i>Enter the Matrix</i>	Infogrames	Action-FPS
<i>Everquest II</i>	Sony	RPG
<i>Far Cry</i>	Ubisoft	Action-FPS
<i>Ghost Recon: Advanced Warfighter</i>	Ubisoft	Action-FPS
<i>Grand Theft Auto: San Andreas</i>	Rockstar Games	Action-Sim
<i>Half-Life 2</i>	Sierra	Action-FPS
<i>Homeworld 2</i>	Vivendi Universal	RTS
<i>Medal of Honor: Allied Assault</i>	Electronic Arts	Action-FPS
<i>Myst III: Exile</i>	Ubisoft	Adventure
<i>PlanetSide</i>	Sony	Action-FPS
<i>Rainbow Six 3: Raven Shield</i>	Ubisoft	Action-FPS
<i>Return to Castle Wolfenstein</i>	Activision	Action-FPS
<i>Rome: Total War</i>	Activision	RTS
<i>SimCity 4</i>	Electronic Arts	Strategy-Sim
<i>Star Wars: Knights of the Old Republic 2</i>	LucasArts	Action-RPG
<i>Syberia 2</i>	Microids	Adventure
<i>Tom Clancy's Splinter Cell: Chaos Theory</i>	Ubisoft	Action-FPS
<i>Unreal II: The Awakening</i>	Infogrames	Action-FPS
<i>Unreal Tournament 2004</i>	Infogrames	Action-FPS
<i>WarCraft III: Reign of Chaos</i>	Blizzard	RTS
<i>World War II Online: Battlefield Europe</i>	Playnet/Cornered Rat Software	FPS-RPG-Sim-Strategy

Gambar 2.2 *Game, Publisher dan Genre.*

Sumber: [KEN-07].

2.1.2 Game Platform

Platform merupakan sistem dimana sebuah *game* dijalankan [GUY-07]. Komputer memiliki tiga pilihan OS (*Operating System*) paling dominan yang dapat digunakan sebagai *platform* sebuah *game* yaitu Microsoft Windows, Linux dan Macintosh. Tipe *platform* lainnya adalah *console*, macam-macam *console* yang terkenal seperti Sony PlayStation, Nintendo Wii, Nintendo GameCube, Sega DreamCast dan Microsoft X-BOX. *Platform* bahkan sudah memasuki dunia *Gadget* seperti PDA (*Personal Digital Assistants*), *Smartphones* (Android), BlackBerry, dan lain-lain [KEN-07].

2.1.3 Game Rating Categories

Game rating categories menunjukkan kesesuaian usia dengan *game* yang dikembangkan [ESR-13]. Penjelasan tentang kategori *rating* dengan mengacu pada organisasi *rating* ESRB dapat dilihat pada Tabel 2.1.

Tabel 2.1 *Game Rating Categories*

Logo	Keterangan
	EC (<i>Early Childhood</i>): Klasifikasi <i>game</i> ini memiliki konten yang mungkin cocok untuk anak usia 3 keatas. Dan tidak memiliki konten yang dianggap tidak pantas dilihat oleh anak dengan usia tersebut.
	E (<i>Everyone</i>): Klasifikasi <i>game</i> ini memiliki konten yang mungkin cocok untuk anak dengan usia 6 atau lebih tua. Judul dalam kategori ini mungkin berisi kartun, fantasi atau kekerasan ringan dan / atau menggunakan bahasa ringan (tidak kasar).
	E10+ (<i>Everyone 10 and older</i>): Klasifikasi <i>game</i> ini memiliki konten yang mungkin cocok untuk usia 10 atau lebih tua. Judul dalam kategori ini mungkin berisi kartun, fantasi atau kekerasan ringan, bahasa ringan dan / atau dengan tema sugestif minimal.

	T (Teen): Klasifikasi <i>game</i> ini memiliki konten yang mungkin cocok untuk usia 13 atau lebih tua. Judul dalam kategori <i>game</i> ini mungkin berisi kekerasan, tema sugestif, humor kasar, minimal darah, simulasi perjudian, dan / atau menggunakan bahasa kuat yang minim.
	M (Mature): Klasifikasi <i>game</i> ini memiliki konten yang mungkin cocok untuk orang usia 17 atau lebih tua. Judul dalam kategori ini mungkin berisi kekerasan intens, darah dan , konten seksual dan bahasa kuat.
	AO (Adults Only): Klasifikasi <i>game</i> ini memiliki konten yang hanya dan harus dimainkan oleh orang 18 tahun keatas. Judul dalam kategori ini mungkin termasuk adegan kekerasan berkepanjangan intens dan konten seksual grafis dengan konten yang bersisi gambar atau video telanjang.
	RP (RATING PENDING): Judul terdaftar sebagai RP (<i>Rating Pending</i>) telah disampaikan kepada ESRB dan sedang menunggu penilaian akhir. (Simbol ini hanya akan muncul dalam iklan sebelum merilis <i>game</i>.)

Sumber: [ESR-13].

2.2 Game Engine

Game engine menyediakan fitur penting pada lingkungan *game* yang dapat digunakan untuk membantu mempermudah proses pengembangan *game*. Banyak definisi tentang *game engine*, namun pada intinya *game engine* adalah sebuah kode untuk mengerjakan hal-hal dasar (*low-level*), seperti berkomunikasi dengan *graphic adapters*, mengatur render, transformasi model, dan menangani fungsi dan perhitungan yang susah dipahami, yang digunakan untuk menjalankan *game* [SZO-04].

Game engine memungkinkan *game* berkembang secara signifikan dan masuk akal dengan membuat lingkungan grafik yang konsisten dan menerapkannya pada objek yang mematuhi hukum fisika dan *requirements* dalam *game* seperti *collision*, *gravity*, dan sebagainya [KEN-07].

2.2.1 jMonkeyEngine (jMonkeyPlatform Game Engine)

jMonkeyEngine adalah *open source* 3D *game framework* berbasis java, *game engine* yang dibuat khusus untuk pengembang *game* yang ingin membuat *game* 3D dengan standar teknologi modern. jMonkeyEngine dibuat dan dijalankan dengan menggunakan bahasa pemrograman Java secara keseluruhan. *Engine* ini dapat digunakan sebagai *stand-alone* SDK atau sebagai *libraries* untuk SDK lain seperti Eclipse dan NetBeans [JME-11]. Fitur yang diberikan oleh jMonkeyEngine antara lain:

1. *Open source software*, jMonkeyEngine dapat digunakan dan dikembangkan sesuai keinginan sendiri tanpa harus mendapatkan lisensi.
2. Sedikit adaptasi untuk kompatibilitas silang yang dapat berjalan pada basis OpenGL dan JVM (*Java Virtual Machine*).
3. Dibangun dengan arsitektur berbasis shader yang memastikan standar grafik sesuai dengan generasi grafik saat ini ataupun masa datang.
4. jMonkeyPlatform merupakan *stand-alone* SDK dengan IDE (*Integrated Development Environment*) lengkap, GUI (*Graphics User Interface*) editor, dan *plugin* yang ditunjukkan pada Gambar 2.3.
5. *Modularity* lengkap untuk menjaga kesesuaian pengembangan *game* dari awal sampai akhir dan mendapatkan hal penting dalam pengembangan *game* langsung dari modul secara bersamaan.



Gambar 2.3 jMonkeyPlatform Alpha-4.

Sumber: Perancangan

2.2.2 jMonkeyPlatform Structure

jMonkeyPlatform terdiri dari tiga bagian utama dan ditunjukkan pada Gambar 2.4, yaitu:

1. Application

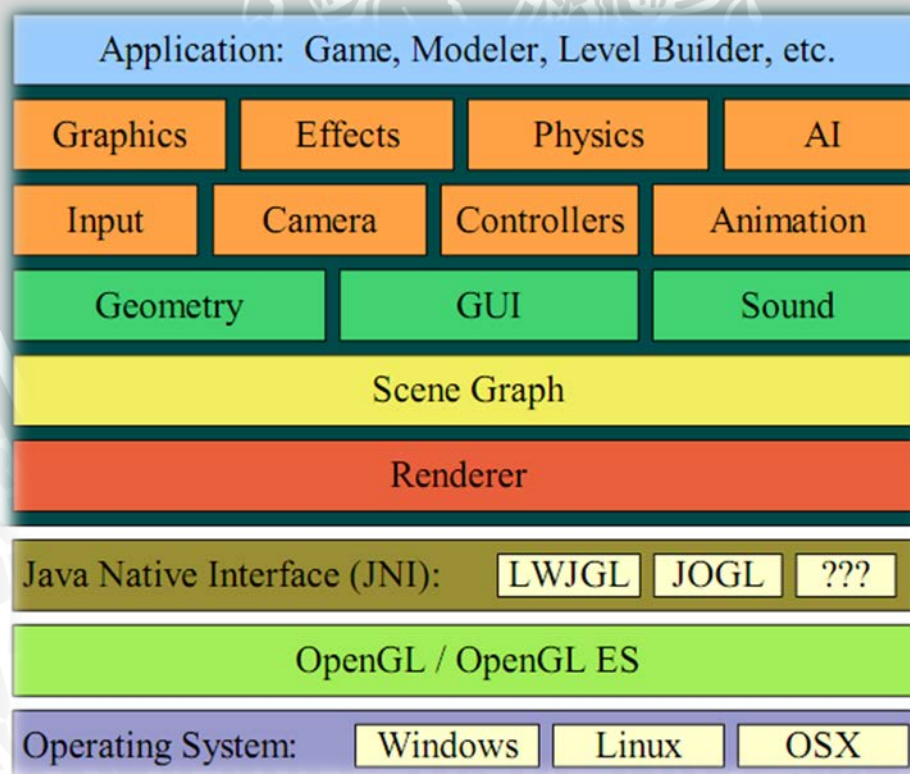
jMonkeyEngine memiliki aplikasi utama *Game*, *Modeler*, *Level Builder*, dan lain-lain yang mengerjakan hal dasar (*low-level*) seperti berkomunikasi dengan *adapter framework adapter* yaitu *objects*, *interface*, *scene graph*, dan *renderer*.

2. JNI (Java Native Interface)

Java Native Interface merupakan sebuah *framework* yang memungkinkan kode Java berjalan di JVM (*Java Virtual Machine*) untuk digunakan pada aplikasi *Native* yang dalam hal ini adalah JOGL (*Java OpenGL*).

3. OS (Operating System)

Operating System merupakan *platform* dimana *game engine* berjalan dan digunakan untuk pengembangan *game*.



Gambar 2.4 jMonkeyPlatform Game Engine Structure.

Sumber: [JME-11]

2.3 Karapan Sapi

Karapan Sapi merupakan atraksi paling terkenal dari Pulau Madura. Sebuah *game* dimana dua orang penunggang yang masing-masing berdiri diatas “keleles”, masing-masing keleles ditarik oleh dua ekor sapi dan saling berlomba untuk mencapai garis *finish*. Sapi yang digunakan untuk berlomba minimal harus berumur dua tahun dan memiliki tinggi 120 sentimeter. Panjang lintasan yang digunakan untuk lomba tergantung dimana lomba diadakan, ada tiga kategori yaitu 120 meter untuk tingkat kecamatan, 130 meter untuk tingkat kabupaten, dan 140 meter untuk tingkat wilayah [BTS-11].

Pembuat kebudayaan Karapan Sapi adalah Pangeran Katandur pada abad ke-13, tradisi ini merupakan simbol rasa terima kasih terhadap Tuhan YME. yang dilakukan oleh peternak sapi di Madura [BTS-11]. Contoh perlombaan Karapan Sapi dapat dilihat pada Gambar 2.5.



Gambar 2.5 Perlombaan Karapan Sapi di Pulau Madura.

Sumber: [BTS-11]

2.4 Teori Pengujian

Teori pengujian yang digunakan pada pembuatan *game* Karapan Sapi 3D dijelaskan pada bagian ini.

2.4.1 Pengujian Terstruktur (*Whitebox Testing*)

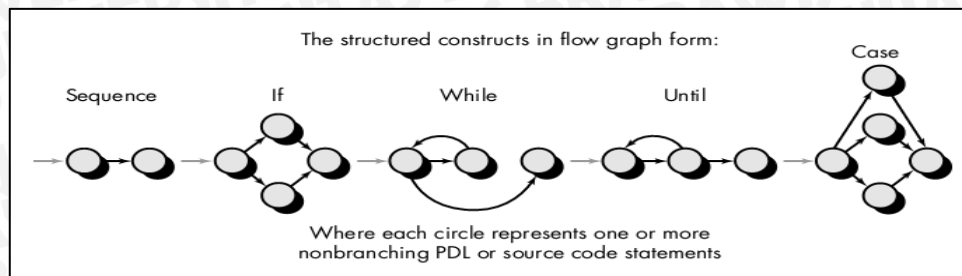
Pengujian terstruktur yang juga dikenal sebagai *whitebox testing*, diimplementasikan untuk menemukan *bug* pada operasi tingkat rendah, seperti yang terjadi pada baris kode, basis data, dan *interface*. Pengujian ini didasarkan pada bagaimana sebuah sistem beroperasi [REX-02].

Dengan *whitebox* didapat kasus uji yang diperlukan untuk:

1. Menjamin bahwa semua alur independen di dalam modul telah dijalankan minimal sebanyak satu kali.
2. Menguji semua keputusan logis pada setiap percabangan.
3. Menguji semua perulangan pada setiap batas masing-masing perulangan.
4. Menguji struktur data internal untuk memastikan validitasnya.

Ada dua metode pengujian yang termasuk pengujian *white-box* yaitu pengujian *basis path* dan pengujian *control structure*. Dalam penelitian ini, metode pengujian *whitebox* yang digunakan adalah pengujian *basis path*.

Metode *basis path* memungkinkan perancang kasus uji untuk memperoleh ukuran kompleksitas logis dari sebuah rancangan prosedural dan menggunakan ukuran ini sebagai pedoman untuk mendefinisikan *basis set* dari jalur eksekusi. Kasus uji dibuat untuk menjamin *basis set* mengeksekusi setiap baris *statement* di dalam program minimal satu kali selama pengujian. Metode *basis path* dilakukan dengan membuat notasi yang merepresentasikan aliran kontrol yang disebut dengan *flow graph*. Sebenarnya metode *basis path* dapat dilakukan tanpa menggunakan *flow graph*, namun *flow graph* dapat dijadikan sebagai alat untuk memudahkan pemahaman aliran kontrol dan mengilustrasikan pendekatan metode ini [PRE-01]. Contoh *flow graph* dapat dilihat pada Gambar 2.6.



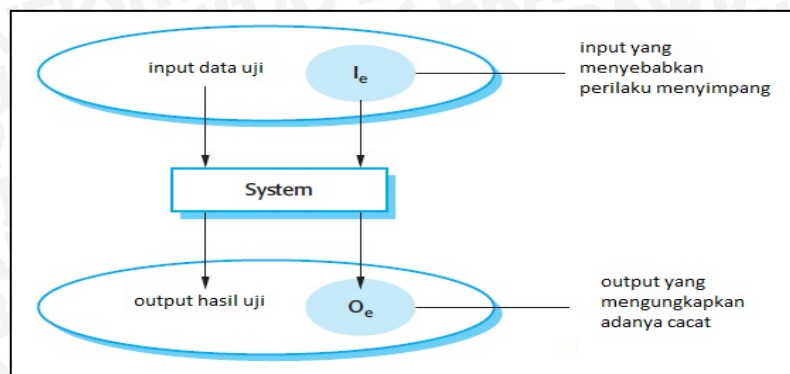
Gambar 2.6 Flow Graph.
Sumber: [PRE-01]

Setelah *flow graph* didefinisikan maka langkah selanjutnya adalah menentukan ukuran kompleksitas (*cyclomatic complexity*). Saat digunakan dalam konteks pengujian *basis path*, nilai untuk *cyclomatic complexity* mendefinisikan jumlah *independent path* dalam *basis set* program dan memberikan batas atas jumlah pengujian yang harus dilakukan untuk memastikan semua *statemen* telah dieksekusi minimal satu kali. *Independent path* adalah sembarang *path* di dalam program yang memberikan paling sedikit satu set *statement* proses baru atau kondisi baru. *Complexity* dihitung dengan beberapa cara, diantaranya [PRE-01]:

1. Jumlah *region* pada *flowgraph* sesuai dengan *cyclomatic complexity*.
2. *Cyclomatic complexity* $V(G)$, untuk grafik G adalah $V(G) = E - N + 2$, dimana E adalah jumlah *edge*, dan N adalah jumlah *node*.
3. $V(G) = P + 1$, dimana P adalah jumlah *predicate node* yaitu *node* yang merupakan kondisi.

2.4.2 Pengujian Fungsionalitas (*Blackbox Testing*)

Blackbox testing sering digunakan untuk menemukan *bug* pada operasi tingkat tinggi, pada level fitur, profil, operasional, dan skenario pelanggan [REX-02]. *Blackbox testing* dilakukan dengan memberi beberapa masukan pada sistem baik masukan yang bersifat normal maupun masukan yang menyebabkan perilaku yang menyimpang pada sistem, hal ini ditujukan agar dapat mendeteksi berbagai macam kemungkinan *bug* yang terdapat pada sistem. Contoh dapat dilihat pada Gambar 2.7.



Gambar 2.7 Blackbox Testing
Sumber: [SOM-11]

2.4.3 Play Testing

Play testing adalah sesuatu yang dilakukan oleh *game designer* melalui proses desain untuk mengetahui apakah *game* dapat memberikan pengalaman bagi pemain sesuai dengan *goal* yang telah ditentukan [FUL-08]. Terdapat banyak cara untuk melakukan *play testing*, sebagian ada yang menggunakan cara informal dan kualitatif, dan sebagian yang menggunakan cara lebih struktural dan kuantitatif. Sebuah *game garage* bisa memiliki 10 – 20 *playtester* untuk melakukan *play testing* dan semua merupakan pengujian yang penting dan berharga. Tetapi, satu hal yang sama dari semua bentuk *play testing* adalah memiliki satu tujuan yaitu mendapatkan *feedback* yang berguna dari pemain untuk meningkatkan keseluruhan pengalaman yang dimiliki *game* [FUL-08].

Sebagai *game designer*, *goal* utama dalam mendesain sebuah *game* adalah memastikan bahwa *game* berfungsi sesuai dengan harapan, komplit secara internal, seimbang, dan menarik untuk dimainkan [FUL-08]. Dalam melakukan *play testing*, sejak *prototype game* selesai dibuat, *tester* terbaik untuk melakukan pengujian pertama kali adalah *game designer*. Jika *prototype game* siap untuk dimainkan oleh orang lain, maka *play testing* dapat dilakukan oleh orang terdekat seperti keluarga dan teman-teman di luar team desain [FUL-08:250].

2.4.4 Pengujian Performa (Performance Testing)

Frame rate (Frame-per-Second) adalah jumlah *frame* animasi yang ditampilkan dalam setiap detik. *Frame rate* terlalu lambat membuat animasi

terputus-putus, sedangkan *frame rate* terlalu cepat membuat *blur detail* animasi [RAM-05].

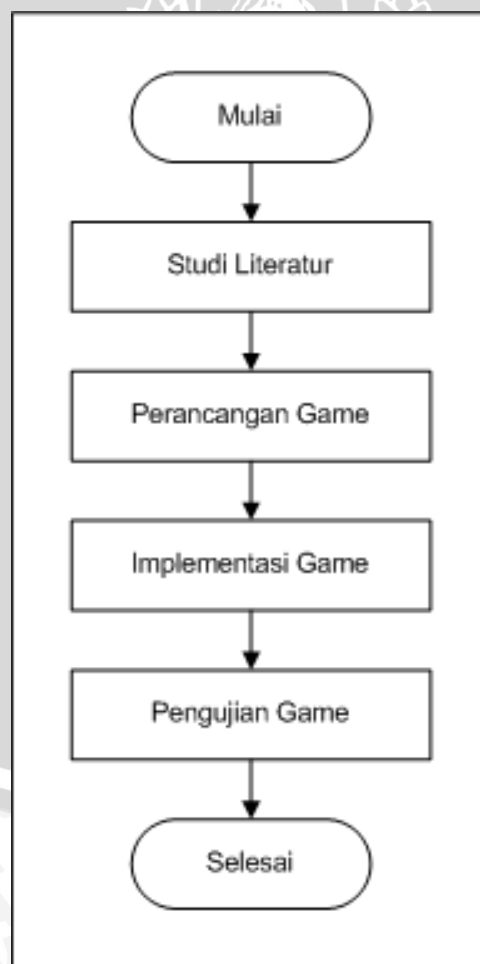
Pengujian kinerja ini akan dipantau dengan menggunakan *benchmarking*. Kegiatan yang memungkinkan kita memantau bagaimana performa dari *game*. Ini merupakan pengukuran seberapa baik performa *game* dan memungkinkan kita untuk melihat berapa banyak *frame* animasi sedang ditampilkan per detik saat *game* sedang berjalan [ADA-10].



BAB III

METODOLOGI PENELITIAN

Langkah-langkah yang akan dilakukan dalam pengerjaan penelitian yaitu studi literatur, tahap perancangan yang terdiri dari *Game Design* dan *Technical Design*, tahap implementasi dengan menggunakan jMonkeyEngine, dan tahap pengujian dan analisis yang menggunakan *whitebox*, *blackbox* dan *performance testing* dari aplikasi *game* yang akan dibuat. Kesimpulan dan saran disertakan sebagai catatan atas aplikasi dan kemungkinan arah pengembangan aplikasi selanjutnya. Berikut diagram alir runtutan pengerjaan penelitian seperti ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram Alir Runtutan Pengerjaan Penelitian.
Sumber: Perancangan.

3.1 Studi Literatur

Studi literatur menjelaskan dasar teori yang digunakan untuk menunjang penulisan skripsi. Teori-teori pendukung tersebut meliputi:

- a. *Game* (Permainan)
 - *Genre* (Klasifikasi) *Game*
 - *Game Platform*
- b. *Game Engine*
 - *jMonkeyEngine* (*jMonkeyPlatform Game Engine*)
 - *jMonkeyPlatform Structure*
- c. Karapan Sapi

3.2 Perancangan *Game Karapan Sapi 3D*

Perancangan *game* dilakukan setelah semua dasar teori didapatkan melalui tahap studi literatur. Perancangan *game* dilakukan berdasarkan *game design* [SCR-10] dan *technical design*. *Game design* meliputi *one-sheet document* yang berisi tentang ringkasan *game* dan *ten-pager document* yang berisi tentang rincian *game*. *Technical design* meliputi OOA (*Object Oriented Analysis*) dan OOD (*Object Oriented Design*) menggunakan pemodelan UML (*Unified Modeling Language*). Perancangan subsistem *game* dilakukan dengan mengidentifikasi kelas dan *interface* yang dibutuhkan yang dimodelkan dalam *class diagram*. Hubungan interaksi antar elemen (objek) yang telah diidentifikasi dimodelkan dalam *sequence diagram* yang menggambarkan interaksi antar objek yang disusun dalam urutan waktu.

3.2.1 *Game Design*

Game design adalah tahap perancangan *game* yang menjelaskan keseluruhan *game*, mulai dari *gameplay*, perancangan karakter, perancangan level, perancangan *interface* dan kontrol pada *game*. *Game design* disusun dalam *one-sheet document* dan *ten-pager document*.

3.2.1.1 One-Sheet Document

One-sheet document adalah gambaran ringkas dari *game* Karapan Sapi 3D. Untuk lebih jelasnya dapat dilihat pada Tabel 3.1.

Tabel 3.1 *One-Sheet Document*

No	Elemen	Keterangan
1.	Ringkasan <i>Game</i>	Karapan Sapi 3D adalah sebuah <i>pervasive game</i> , yang berjalan di <i>platform</i> PC dengan sistem operasi Windows. Target usia pemain antara 5 sampai 13 tahun.
2.	ESRB Rating	
2.	Sekilas Cerita	<i>Game</i> ini diangkat dari kebudayaan asli Indonesia yang berasal dari Madura yaitu Karapan Sapi. <i>Game</i> ini mengimplementasikan <i>game</i> Karapan Sapi yang sesungguhnya, dimana pemain akan mengendalikan kereta sapi untuk mencapai <i>finish</i> dan mencatatkan waktu tercepat.
3.	Cara bermain	Pemain memilih salah satu karakter dari dua karakter yang disediakan, yaitu White and Brown. Setelah memilih karakter, dapat memilih level dari level 1 sampai dengan 3. Setelah pemain memilih karakter dan level, pemain harus mengendalikan karakter untuk mencapai garis <i>finish</i> dengan menghindari rintangan dan mendapatkan <i>mystery box</i> yang berisikan power up untuk membantu mencapai garis <i>finish</i> lebih cepat.
4.	USP (<i>Unique Selling Point</i>)	Untuk menambah daya tarik <i>game</i> akan ditambahkan fitur kedalam <i>game</i> seperti karakter sapi dengan bentuk lucu. Selama <i>game</i> pemain harus dapat menghindari rintangan dan mendapatkan kotak (<i>mystery box</i>) yang

	berisi <i>power up</i> , pada setiap level rintangan akan semakin sulit dan <i>mystery box</i> semakin sedikit.
--	---

Sumber: Perancangan

3.2.1.2 Ten-Pager Document

Ten-pager document adalah dokumen yang berisi penjabaran garis besar dari perancangan *game* Karapan Sapi 3D. Tujuan nya adalah agar pembaca dapat segera memahami konsep utama *game* tanpa harus melihat secara detail. Isi *ten-pager document* antara lain *game story*, penjelasan tentang *gameplay*, alur permainan atau *gameflow*, perancangan karakter, kontrol dalam *game*, konsep utama *gameplay* dan fitur, *game world*, *interface game*, perancangan *environment*, dan karakter NPC.

1. Cerita dan Gameplay

Game Karapan Sapi 3D bercerita tentang dua ekor sapi, Brown dan White, yang merupakan sapi peliharaan dari Baldi, seorang anak Madura berkepala gundul yang berusia 10 tahun, yang berencana akan ikut dalam perlombaan Karapan Sapi yang diadakan secara rutin oleh pemerintah setempat. Lomba Karapan Sapi tersebut akan diselenggarakan pada bulan depan, karena sapi yang hanya boleh didaftarkan hanya 1 ekor, Baldy kesulitan dalam memilih apakah Brown atau White yang nantinya akan didaftarkan dalam lomba tersebut. Melihat pemilik mereka dilanda kegalauan dalam memilih, Brown dan White sepakat melakukan sparring untuk menentukan siapa yang lebih layak didaftarkan sebagai peserta lomba Karapan Sapi tersebut sekaligus sebagai ajang latihan untuk lomba sebenarnya dan berjanji tidak akan ada dendam bila salah satu mereka tidak terpilih. Dengan menggunakan *setting* sesuai dengan lomba sebenarnya, Brown dan White mengikuti Baldy berkeliling Indonesia untuk melakukan *training camp* dan berusaha semaksimal mungkin untuk mencapai garis *finish* dengan waktu tercepat sekaligus melewati rintangan untuk melatih kemampuan sampai melewati batas maksimum.

Game Karapan Sapi 3D merupakan *game single player*, dimana pemain bisa memilih 2 karakter yang akan ditunggangi oleh Baldy yaitu Brown atau White.

Pemain harus berusaha untuk mencapai garis *finish* untuk mendapatkan catatan waktu tercepat. Pemain harus mengendalikan karakter untuk mencapai garis *finish* dengan menghindari rintangan yang terdapat di jalur dan mendapatkan *mystery box* yang berisikan power up untuk membantu mencapai garis *finish* lebih cepat.

2. **Gameflow**

Gameflow dari *game* Karapan Sapi 3D dirancang agar pemain dapat memainkan *game* dari awal mulai *game* sampai mencapai kondisi akhir yang berupa *game over* atau *finish* sesuai dengan *gameplay* yang dirancang.

3. **Karakter**

Karakter disesuaikan dengan tema dari *game* untuk mengimplementasikan kondisi perlombaan Karapan Sapi sebenarnya. Sehingga karakter yang dibuat merupakan aktor dari perlombaan Karapan Sapi, Sapi dan penunggangnya, yang akan digunakan sebagai karakter utama dalam *story line game* Karapan Sapi 3D. Karakter dibuat dengan model yang lucu agar *game* tidak terkesan terlalu serius dan membosankan.

4. **Kontrol**

Kontrol pada *game* dibuat semudah mungkin sehingga tidak menyulitkan pemain saat akan memulai *game*, karena *game* Karapan Sapi 3D merupakan *game* dengan tujuan hiburan dan pendidikan tentang kebudayaan asli Indonesia.

5. **Konsep Utama *Gameplay* dan Fitur**

Konsep utama *gameplay game* Karapan Sapi 3D adalah untuk bisa mencapai garis *finish* dengan catatan waktu tercepat. Semakin bertambah level, tingkat kesulitan penyelesaian akan semakin bertambah dengan semakin bertambahnya rintangan dan semakin berkurangnya *mystery box*.

Fitur yang diberikan tidak terbatas hanya pada perlombaan Karapan Sapi yang sebenarnya, tetapi juga diberikan fitur tambahan sebagai variasi dan pembeda dengan *game* Karapan Sapi 3D.

6. **Game World**

Game world pada *game* Karapan Sapi 3D menggunakan latar belakang ikon berbagai daerah di Indonesia, dengan tujuan selain memperkenalkan kebudayaan juga memperkenalkan tempat eksotis di Indonesia.

7. **Interface**

Interface merupakan tampilan yang dimulai dari tampilan menu sampai dengan tampilan pada saat *game* dan kondisi akhir, *interface* dirancang dengan tampilan minimalis namun elegan dan mencerminkan kebudayaan daerah Madura sehingga tidak terlalu mengganggu dan membingungkan pemain untuk memulai *game*.

8. **Environment**

Lingkungan atau *enviromtment* pada *game* akan disesuaikan dengan kondisi lingkungan nyata dari setiap level dan sedikit ditambahkan papan reklame atau advertisement untuk memberitahukan pemain bahwa *game* dibuat oleh mahasiswa PTIIK UB.

9. **Game Experience**

Karapan Sapi 3D menambahkan fitur yang tidak ada pada perlombaan Karapan Sapi sesungguhnya, sehingga dapat memberikan sebuah *game experience* tersendiri bagi pemain.

10. **Kompetitor**

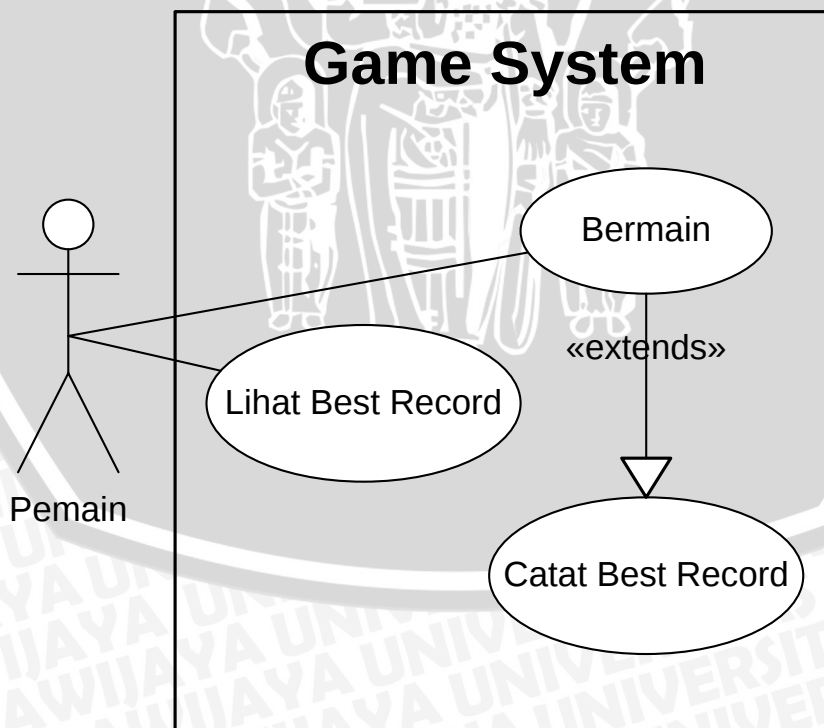
Game Karapan Sapi 3D memiliki kompetitor yang memiliki genre *game* hampir sama yaitu *game* PepsiMan. Namun *game* Karapan Sapi 3D merupakan *game* yang bertema kebudayaan dan bertujuan untuk melestarikan kebudayaan dan sarana pendidikan, berbeda dengan *game* PepsiMan yang bertujuan sebagai

sarana pemasaran, sehingga diantara kedua *game* tersebut masih terdapat batas pembeda.

3.2.2 Technical Design

Technical design berfungsi sebagai analisis kebutuhan yang bertujuan untuk mendapatkan semua kebutuhan yang diperlukan dari *game* yang akan dibangun sesuai dengan *game design* yang telah dirancang. *Technical design* dilakukan dengan mengidentifikasi semua kebutuhan (*requirements*) sistem dan siapa saja yang terlibat di dalamnya. Analisis juga dilakukan untuk mengetahui kondisi lapangan yang ada sehingga dapat diketahui implementasi perangkat lunak yang akan digunakan.

Metode analisis yang digunakan adalah OOA (*Object Oriented Analysis*) dengan menggunakan UML (*Unified Modeling Language*) sebagai bahasa pemodelan. *Use case diagram* digunakan untuk mendeskripsikan kebutuhan dan fungsionalitas sistem dari perspektif *end-user*. *Technical design* dilakukan dengan mengidentifikasi semua kebutuhan (*requirements*) *game* pada *game design* yang kemudian akan dimodelkan dalam *use case diagram* seperti pada Gambar 3.2.



Gambar 3.2 Use Case Diagram Karapan Sapi 3D.

Sumber: Perancangan.

3.3 Implementasi *Game* (*Game Implementation*)

Implementasi aplikasi dilakukan dengan mengacu kepada perancangan aplikasi. Implementasi *game* dilakukan dengan menggunakan bahasa pemrograman berbasis objek yaitu menggunakan bahasa pemrograman Java dan platform jMonkeyPlatform sebagai IDE (*Integrated Developing Environment*).

3.4 Pengujian *Game* (*Game Testing*)

Pengujian perangkat lunak pada skripsi ini dilakukan agar dapat menunjukkan bahwa perangkat lunak telah mampu bekerja sesuai dengan spesifikasi dari kebutuhan yang melandasinya. Pengujian dilakukan menggunakan metode *whitebox testing*, *blackbox testing* dan *performance testing*.

3.4.1 *Whitebox Testing*

Whitebox testing adalah teknik verifikasi yang digunakan untuk mencari tahu apakah kode telah bekerja sesuai dengan harapan. Pada pengujian aplikasi, hanya digunakan *Unit Testing* dan *Integration Testing*.

3.4.1.1 *Unit Testing*

Unit testing merupakan pengujian yang dilakukan pada masing-masing komponen secara terpisah untuk mengetahui apakah masing-masing komponen telah bekerja sesuai harapan secara terpisah.

3.4.1.2 *Integration Testing*

Integration testing merupakan pengujian yang dilakukan terhadap sebuah sistem secara keseluruhan untuk mengetahui apakah masing-masing komponen dapat berinteraksi dan telah terintegrasi sesuai harapan.

3.4.2 *Blackbox Testing*

Blackbox testing adalah *testing* yang terfokus pada penentuan apakah suatu program telah melakukan apa yang seharusnya dilakukan berdasarkan *requirement*. *Blackbox* dilakukan untuk mencari tahu apakah output yang

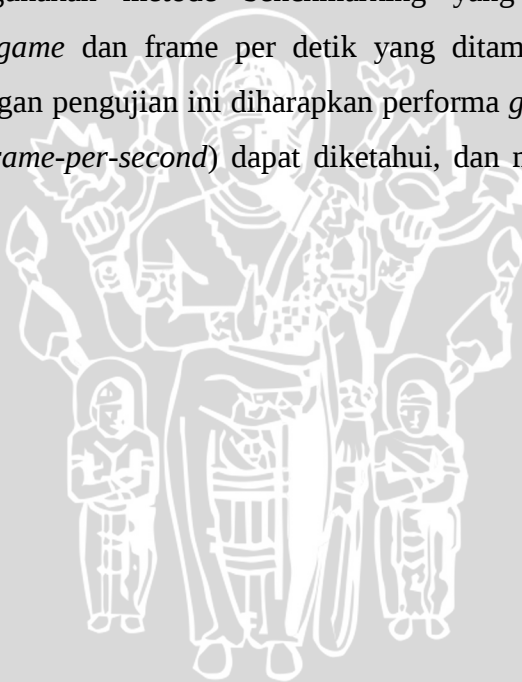
dihasilkan sistem sesuai harapan bila diberikan input dan tidak memperdulikan bagaimana kode dan sistemnya bekerja.

3.4.3 Play Testing

Play testing dilakukan untuk mengetahui *feedback* dari *user* tentang *game* Karapan Sapi 3D. Pada pengujian ini memberikan sebuah kuisioner kepada 15 pemain/*user* tentang kondisi *gameplay*, saran dan kesan pada *game* Karapan Sapi 3D.

3.4.4 Performance Testing

Pengujian performa yang akan dilakukan disini adalah *frame rate testing*. Dimana akan menggunakan metode benchmarking yang akan membantu mengamati performa *game* dan frame per detik yang ditampilkan oleh *game* Karapan Sapi 3D. Dengan pengujian ini diharapkan performa *game* Karapan Sapi 3D khususnya FPS (*frame-per-second*) dapat diketahui, dan memenuhi standart yaitu minimal 60 FPS.



BAB IV

PERANCANGAN

Perancangan *game* Karapan Sapi 3D terdiri atas dua tahap. Pada tahap pertama melakukan perancangan *game* berdasarkan *gameplay* Karapan Sapi 3D menggunakan *Game Design* dan pada tahap kedua melakukan proses analisis kebutuhan dengan menggunakan *Technical Design*.

4.1 *Game Design*

Game Design digunakan untuk perancangan sebuah *game* [SCR-10]. Tahap perancangan dilakukan dengan cara menjabarkan secara rinci *gameplay* dengan menggunakan *one-sheet document* dan *ten-pager document*.

4.1.1 *Game Title*

Judul *game* yang dirancang adalah Karapan Sapi 3D, sebuah *game* yang bertemakan kebudayaan daerah asli Madura, dan ditempatkan pada menu utama. Perancangan *game title* dapat dilihat pada Gambar 4.1.



Gambar 4.1 Perancangan *Game Title*
Sumber: Perancangan

4.1.2 Game Goal

Karakter pemain harus dapat melewati rintangan dan mendapatkan *mystery box* untuk membantu mencapai garis *finish* dengan waktu tercepat setiap *level*-nya dan *Health Points* (HP) tidak boleh mencapai nilai 0 untuk setiap levelnya. Tingkat kesulitan penyelesaian *game* akan semakin bertambah sejalan dengan meningkatnya level, hal tersebut ditandai dengan semakin bertambahnya rintangan dan semakin berkurangnya *mystery box*.

4.1.3 Game Outline

Game Karapan Sapi 3D, merupakan *game* single player, yang menggunakan sudut pandang orang ke tiga, yang dikontrol menggunakan keyboard. Gambaran *interface prototype game* dapat dilihat pada Gambar 4.2 berikut penjelasan:

1. Karakter : Karakter yang nantinya dimainkan adalah seorang penunggang karapan beserta dengan sapi yang menyeret kereta.
2. Tanjakan : Tanjakan dapat digunakan untuk melewati rintangan ataupun dapat berfungsi sebagai rintangan.
3. Finish Line : Garis *finish* dimana pemain harus dapat melintasinya untuk mengakhiri *game*.
4. Background : Latar belakang *game* yang nantinya dapat berupa penonton atau bangunan.
5. Rintangan : Rintangan yang berupa palang atau parit dimaksudkan untuk menghambat pemain untuk mencapai garis *finish*.
6. Track : Lintasan yang digunakan dalam *game*.
7. Item Box : Digunakan sebagai penambah status player.
8. Sky : Langit yang ada dalam *game*.
9. Time : Menunjukan waktu yang dilewati.



Gambar 4.2 Prototype Game Karapan Sapi 3D
Sumber: Perancangan

4.1.4 Karakter

Karakter dalam *game* disesuaikan dengan *storyline* dan *gameplay*. Karakter memiliki informasi profil dan background yang berbeda untuk menambah keunikan dalam *game*. Informasi karakter brown dapat dilihat pada Tabel 4.1, informasi karakter White dapat dilihat pada Tabel 4.2, dan informasi penunggang sekaligus pemilik kedua sapi, Baldy, dapat dilihat pada Tabel 4.3.

4.1.4.1 Profil Karakter

Tabel 4.1 Profil Karakter Brown

Karakter	Profil
	Nama : Brown
	Kecepatan : 20 Kmph
	Health : 75 <i>Health Points</i>
	Latar belakang: Brown adalah seekor sapi yang berasal dari Madura. Pekerjaannya sehari-hari membantu Baldy, sang pemilik, dalam membajak sawah.

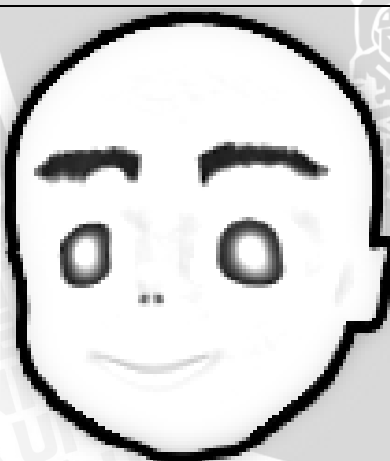
Sumber: Perancangan

Tabel 4.2 Profil Karakter White

Karakter	Profil
	Nama : White
	Kecepatan : 15 Kmph
	Health : 100 <i>Health Points</i>
	Latar belakang: White merupakan keturunan dari sapi perah milik Baldy. Selain membantu Baldy membajak sawah, White juga sering menemani Baldy jalan-jalan di padang rumput dekat rumahnya.

Sumber: Perancangan

Tabel 4.3 Profil karakter Baldy

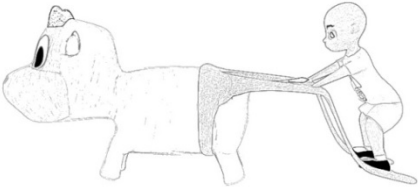
Karakter	Profil
	Nama : Baldy
	Jenis Kelamin : Laki-Laki
	Usia : 10 Tahun
	Latar belakang: Baldy merupakan anak keturunan asli Madura. Selain menuntut ilmu di bangku Sekolah Dasar, dia juga sering membantu orang tuanya membajak sawah dan berjalan-jalan bersama Brown dan White.

Sumber: Perancangan

4.1.5 Gerakan Karakter

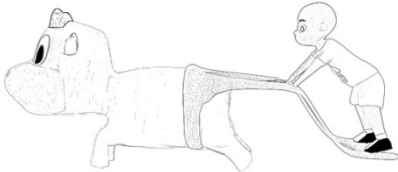
Pada bagian ini akan dijelaskan tentang gerakan-gerakan yang tersedia untuk setiap karakter dalam *game* Karapan Sapi 3D. Gerakan-gerakan yang akan digunakan oleh karakter pemain utama adalah gerakan lari, gerakan *idle*, gerakan lompat dan gerakan supercow!. Gerakan lari dapat dilihat pada Tabel 4.4, gerakan jaga dapat dilihat pada Tabel 4.5, gerakan *idle* dapat dilihat pada Tabel 4.6, gerakan, dan gerakan supercow! dapat dilihat pada Tabel 4.7.

Tabel 4.4 Gerakan Lari

Gambar	Keterangan
	Jenis Gerakan : Lari
	Gerakan lari merupakan gerakan utama bagi karakter <i>Player</i> . Karena gerakan lari menunjukkan bahwa karakter <i>player</i> berpindah tempat dan menghindari rintangan dan untuk mencapai garis <i>finish</i> .

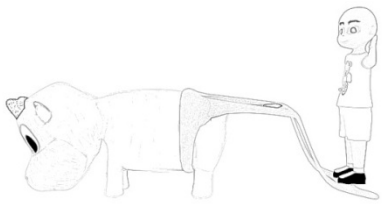
Sumber: Perancangan

Tabel 4.5 Gerakan Lompat

Gambar	Keterangan
	Jenis Gerakan : Lompat
	Gerakan Jaga merupakan gerakan yang digunakan untuk menghindari rintangan dengan mengangkat bagian tubuh karakter sapi dan karakter manusia.

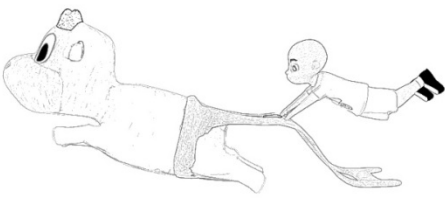
Sumber: Perancangan

Tabel 4.6 Gerakan *Stand (idle)*

Gambar	Keterangan
	Jenis Gerakan : Idle
	Gerakan <i>idle</i> , atau gerakan yang dilakukan karakter <i>player</i> ketika sedang berhenti berlari dan tidak melompat.

Sumber: Perancangan

Tabel 4.7 Gerakan *Supercow!*

Gambar	Keterangan
	Jenis Gerakan : SuperCow!
	Gerakan <i>Supercow!</i> , adalah gerakan khusus yang bisa didapatkan dari <i>mystery box</i> . Gerakan ini membuat karakter sapi dan karakter manusia terbang.

Sumber: Perancangan

4.1.6 Klasifikasi Karakter

Karakter pada *game* Karapan Sapi 3D diklasifikasikan menjadi dua jenis, yang pertama adalah karakter utama, dan yang kedua adalah karakter figuran.

4.1.6.1 Karakter Utama (*Main Character*)

Karakter utama adalah karakter yang dikendalikan oleh pemain. Gerakan karakter utama ditentukan oleh perintah atau *input* yang diberikan oleh pemain melalui kontrol yang tersedia yang dalam *game* ini adalah *keyboard*. Karakter utama merupakan karakter yang dipilih oleh pemain pada sub-menu *Choose Char*

(Pilih Karakter). Pada sub-menu pilih karakter pemain dapat memilih dua karakter yaitu Brown dan White.

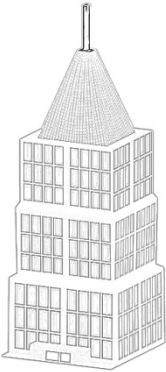
4.1.6.2 Karakter Figuran (*Side Character*)

Karakter figuran adalah karakter yang berperan sebagai pembantu dalam *game* Karapan Sapi 3D. Karakter pembantu dalam *game* ini adalah Baldy, dimana dia akan menunggangi sapi yang merupakan karakter utama dalam cerita (*storyline*) *game* Karapan Sapi 3D.

4.1.7 *Environment*

Pada bagian ini akan dijelaskan tentang objek yang terdapat pada lingkungan dan sekitar *game* Karapan Sapi 3D. Keterangan dapat dilihat pada Tabel 4.8.

Tabel 4.8 Keterangan Objek

Jenis <i>Environment</i>	Keterangan
	<i>Mystery box</i> merupakan objek yang terdapat pada <i>game</i> untuk memberikan <i>buff</i> spesial kepada karakter.
	Gedung rektorat merupakan salah satu <i>environment</i> unik. <i>Environment</i> unik digunakan sebagai latar belakang utama <i>game</i>. Pada level yang berbeda akan memiliki <i>environment</i> unik yang berbeda.

Sumber: Perancangan

4.1.8 Kontrol Game

Pada bagian kontrol *game* akan dijelaskan tentang kontrol yang digunakan dalam *game*, yaitu *keyboard*. *Game* akan sepenuhnya dikontrol menggunakan *keyboard* yaitu tombol W, A, S, D, dan spasi untuk gerakan karakter, dan tombol P untuk *game state pause*. yang terdapat pada setiap *keyboard*. Untuk lebih jelasnya dapat dilihat pada Tabel 4.9.

Tabel 4.9 Kontrol Game

Kontrol	
Gerakan yang dihasilkan	• W untuk Maju • A bergerak ke kiri • D bergerak ke kanan • S untuk mengerem • Spasi untuk melompat • P untuk pause/unpause

Sumber: Perancangan

4.1.8.1 Maju

Gerakan maju dihasilkan oleh *input* W seperti yang ditunjukkan pada Tabel 4.9. Ketika *input* W di tekan, karakter *player* akan bergerak lurus maju ke depan. Karakter akan mengurangi kecepatan sampai berhenti bergerak ketika tombol dilepaskan.

4.1.8.2 Samping Kanan dan Kiri

Gerakan samping dihasilkan oleh *input* A dan D seperti yang ditunjukkan pada Tabel 4.9. Ketika *input* D di tekan, maka karakter *player* akan bergerak ke arah kanan. Begitu pula ketika *input* A di tekan, maka karakter akan berbelok ke arah kiri. karakter akan berhenti bergerak ketika tombol A maupun D dilepaskan.

4.1.8.3 Kontrol Gerakan *Brake* (Mengerem)

Gerakan mengerem dihasilkan oleh *input* S seperti yang ditunjukkan pada Tabel 4.9. Ketika *input* S di tekan, maka karakter *player* akan mengurangi kecepatan sampai berhenti total apabila tombol S tidak dilepaskan.

4.1.9 *Interface Game*

Interface game Karapan Sapi 3D dari rancangan Menu Utama dapat dilihat pada Gambar 4.3, sub-menu *Choose Char* pada Gambar 4.4, sub-menu *Choose Level* dan tampilan *game* pada Gambar 4.5 dan 4.6.

4.1.9.1 Menu Utama



Gambar 4.3 Rancangan Menu Utama
Sumber: Perancangan

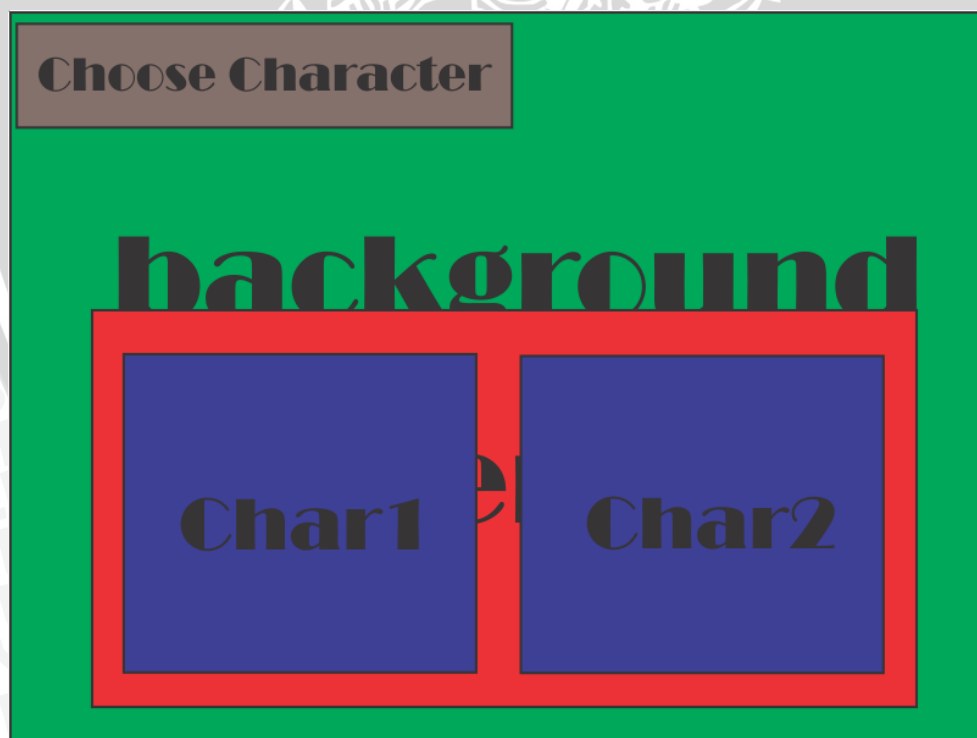
Pada Gambar 4.3 ditunjukkan bahwa pada menu utama terdapat terdapat tiga kolom yang akan dijelaskan di bawah:

1. Kolom judul *game*.
2. Kolom menu.
3. Kolom pilihan menu.
4. Kolom *background*.

4.1.9.2 Choose Character (Pilih Karakter)

Dari Gambar 4.4 ditunjukkan bahwa pada sub-menu pilih karakter terdapat terdapat 4 kolom yang akan dijelaskan di bawah:

1. Kolom *background*.
2. Kolom menu karakter.
3. Kolom pilihan karakter.
4. Kolom judul tampilan.



Gambar 4.4 Rancangan Sub-menu *Choose Character*
Sumber: Perancangan

4.1.9.3 Choose Level (Pilih Level)



Gambar 4.5 Rancangan Sub-menu *Choose Level*

Sumber: Perancangan

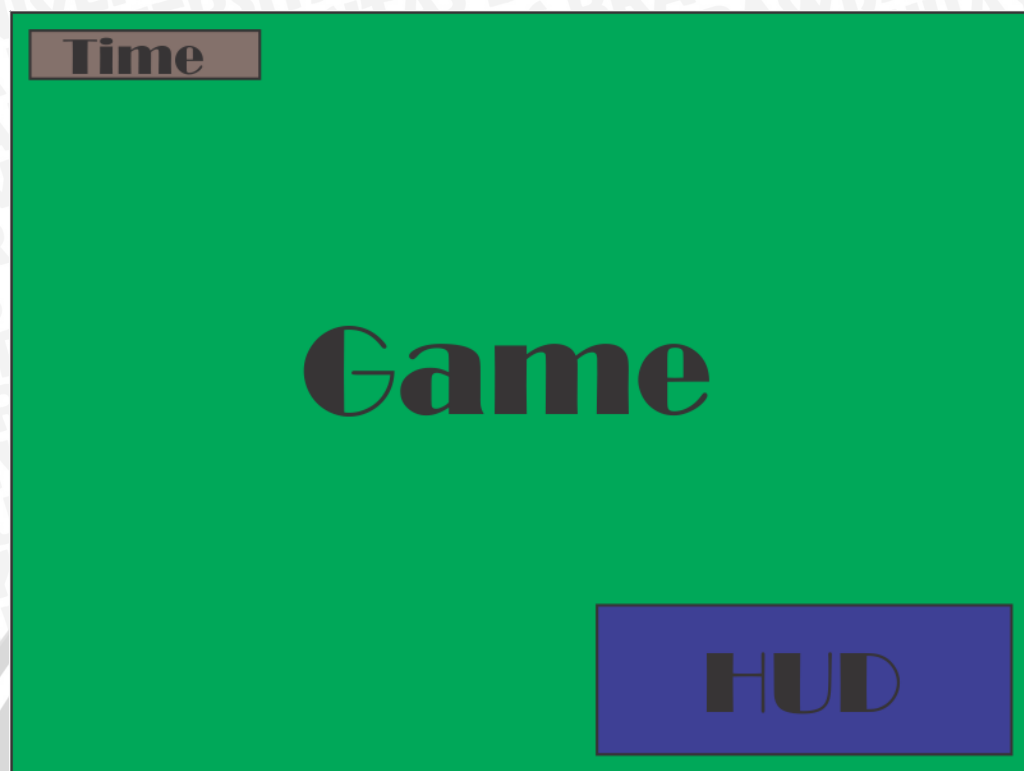
Dari Gambar 4.5 ditunjukkan bahwa pada sub-menu pilih karakter terdapat terdapat 3 kolom yang akan dijelaskan di bawah:

1. Kolom *background*.
2. Kolom menu level.
3. Kolom pilihan level.
4. Kolom judul tampilan.

4.1.9.4 Interface Game

Dari Gambar 4.6, rancangan halaman *game*, dapat dilihat bahwa pada *interface game* akan terdapat :

1. *Game Time*.
2. *HUD (Heads-Up Display)*.
3. *In-game display*.



Gambar 4.6 Rancangan *Interface Game*
Sumber: Perancangan

4.1.10 Level Game

Penjelasan tentang level dan latar belakang dari setiap level akan dijelaskan dengan detail sebagai berikut.

4.1.10.1 Deskripsi Level

Level dalam *game* Karapan Sapi 3D dirancang dengan karakteristik yang berbeda di setiap tingkatannya untuk menunjukkan dan mengenalkan daerah dan budaya yang ada di Indonesia, sebagai contoh Gedung Rektorat dan PTIIK UB di Malang, Tugu Monas dan Pantai Ancol di Jakarta dan Candi Borobudur di Jogjakarta. Tingkat kesulitan akan meningkat pada setiap level yaitu dengan bertambahnya objek rintangan dan berkurangnya objek *mystery box*.

4.1.10.2 Level Background

Background pada *game* Karapan Sapi 3D, akan menggunakan *icon* yang terdapat di setiap daerah. Pada level satu, menggunakan *background* Universitas

Brawijaya dan gedung PTIHK yang berada di kota Malang, level dua menggunakan *background* Tugu Monas dan Pantai Ancol yang berada di kota Jakarta dan level 3 menggunakan *background* Candi Borobudur di daerah Jogjakarta. Tabel 4.10 akan menjelaskan detail level *background*.

Tabel 4.10 Level *Background*

Level	Background	Keterangan
1 (satu)		Mengambil latar belakang halaman rektorat Universitas Brawijaya sebagai <i>background</i> level 1. Universitas Brawijaya digunakan agar pemain mengetahui bahwa pembuatan <i>game</i> dilakukan di UB.
2 (dua)		Mengambil latar belakang Tugu Monas, sebagai <i>background</i> level 2.
3 (tiga)		Mengambil latar belakang Candi Borobudur sebagai latar belakang level 3.

Sumber: Perancangan

4.1.11 Sistem Pada *Game*

Sistem pada *game* digunakan sebagai rincian sistem dari *gameplay* yang dirancang agar sesuai dengan tujuan perancangan *game*.

4.1.11.1 *Health Points dan Speed Indicator*

Setiap karakter memiliki life point dan kecepatan maksimum yang berbeda sebagai perbedaan status antara Brown dan White. Apabila *life point* karakter mencapai nilai nol, maka salah satu *end-point* (kondisi akhir) akan tercapai, yaitu *game over*. Karakter diharuskan mencapai *finish* dengan kondisi *Health points* tidak boleh nol untuk mencapai *end-point* yang diharapkan. Sistem penilaian dapat dilihat pada Tabel 4.11.

4.1.11.2 *Rintangan dan Mystery Box Collision*

Rintangan dan *mystery box* merupakan fitur variasi dari perlombaan karapan sapi yang sebenarnya agar *game* tidak terkesan membosankan. Tabrakan dengan objek rintangan, akan mengurangi *life point* dan kecepatan secara berkala. Sedangkan untuk tabrakan dengan *mystery box*, akan memberikan bonus efek (*debuff status*) bervariasi yaitu bonus *movement speed*, penambahan *life point*, dan kekuatan Supercow!. Rintangan akan semakin banyak sesuai dengan bertambahnya level, sedangkan *mystery box* akan semakin sedikit sesuai dengan bertambahnya level sehingga tingkat kesulitan level akan semakin bertambah. Sistem penilaian dapat dilihat pada Tabel 4.11.

4.1.11.3 *Rekor dan Scoring System*

Game Karapan Sapi 3D menggunakan sistem penilaian berdasarkan waktu tempuh saat mencapai *finish* dan sisa *health points*. Variabel tersebut digunakan sebagai variabel perhitungan dengan rumus yang dibuat untuk mendapatkan skor total. Skor total akan dibandingkan apakah lebih baik dari rekor yang pernah dicatat sebelumnya, apabila kondisi tercapai, maka *player* akan mencatatkan rekornya pada sistem sebagai rekor terbaik dan akan coba dipecahkan oleh *player* lain. Sistem penilaian dapat dilihat pada Tabel 4.11.

Tabel 4.11 Sistem Penilaian.

Jenis	State	Penilaian
<i>Speed</i>	Pengurangan	Velocity += (-0.025)
	Standar	Velocity += 0.01
	Bonus	Velocity += 0.1
<i>Health points</i>	Pengurangan	curHealth += (-1)
	Bonus	curHealth += 12.5
<i>Score</i>	Total	$100 * 60 / (\text{minute} * 60 + \text{second}) + \text{curHealth}$

Sumber: Perancangan

4.1.11.4 Sudut Pandang Kamera

Sudut pandang kamera pada *game* Karapan Sapi 3D adalah sudut pandang orang ketiga. Sehingga *user* atau pemain dapat melihat secara utuh karakter yang sedang dikendalikan.

4.1.11.5 Musik dan Sound Effect (Audio)

Game Karapan Sapi 3D menggunakan *audio* berupa musik dan efek suara. Setiap *file audio* akan menggunakan format ekstensi Waveform Audio File Format (WAV) karena memiliki *rendering audio* dengan kualitas paling bagus pada *game engine* jMonkeyPlatform, dan file yang digunakan merupakan *file* yang tidak membutuhkan lisensi, atau *free license*.

4.2 Technical Design

Technical Design, yang berfungsi sebagai analisis kebutuhan, pada tahap perancangan ini dilakukan untuk membangun sebuah *game* berdasarkan hasil pengumpulan, pemahaman dan penetapan dari kebutuhan sesuai dengan *gameplay* pada *Game Design* agar dapat memenuhi keinginan pengguna. Proses *Technical Design* diawali dengan analisis data yang akan digunakan, memodelkan kedalam diagram *use case*, membuat *class diagram* untuk perancangan struktur *class*, dan membuat *activity diagram* untuk menggambarkan kegiatan yang dapat dilakukan pada sistem berdasarkan *use case diagram*.

4.2.1 Use Case Diagram

Use case diagram dimulai dengan tahap identifikasi aktor, kemudian membuat daftar kebutuhan, dan menggambarkan *use case diagram* berdasarkan tahap identifikasi aktor dan daftar kebutuhan.

4.2.1.1 Identifikasi Aktor

Tahap identifikasi aktor merupakan tahap untuk melakukan identifikasi terhadap aktor yang akan berinteraksi dengan *game* Karapan Sapi 3D. Tabel 4.12 memperlihatkan aktor yang terlibat beserta penjelasan yang merupakan hasil dari proses identifikasi aktor.

Tabel 4.12 Identifikasi Aktor

Aktor	Deskripsi
User	User adalah pemain (<i>player</i>) yang bisa memainkan karakter utama pada <i>game</i> Karapan Sapi 3D dan memiliki akses untuk melihat <i>highscore</i> dan <i>credit</i> .

Sumber: Perancangan

4.2.1.2 Daftar Kebutuhan

Daftar kebutuhan yang terdapat pada *game* Karapan Sapi 3D dapat dilihat pada Tabel 4.13.

Tabel 4.13 Daftar kebutuhan

N0	Kode	Kebutuhan	Use Case
1.	TD_001	Game harus menyediakan fasilitas untuk memulai permainan.	Start game
2.	TD_002	Game harus menyediakan fasilitas untuk memilih karakter. Sehingga pemain dapat memilih karakter yang akan menjadi karakter <i>player</i> utama.	Pilih karakter

3.	TD_003	Game harus menyediakan fasilitas memilih level, sehingga pemain dapat memilih level.	Pilih level
4.	TD_004	Game harus menyediakan fasilitas untuk bermain.	Bermain
5.	TD_005	Game harus menyediakan fasilitas untuk menggerakkan karakter. sehingga pemain dapat menggerakkan karakter <i>player</i> utama.	Maju, brake, gerak kanan, gerak kiri, lompat
6.	TD_006	Game harus menyediakan fasilitas untuk melakukan catat <i>highscore</i> .	Mencatat <i>high score</i>
7.	TD_007	Game harus menyediakan fasilitas untuk <i>pause game</i> . Sehingga pemain dapat menghentikan <i>game</i> untuk sementara waktu.	<i>Pause game</i>
8.	TD_008	Game harus menyediakan fasilitas untuk melihat <i>best score</i> yang pernah dicatat oleh sistem. Sehingga pemain dapat melihat catatan <i>best score</i> .	<i>Lihat high score</i>
9.	TD_009	Game harus menyediakan fasilitas untuk melihat <i>credit</i> . Sehingga pemain dapat melihat siapa saja yang terlibat dalam pembuatan <i>game</i> .	<i>Credit</i>
10.	TD_010	Game harus menyediakan fasilitas untuk keluar dari <i>game</i> . Sehingga pemain dapat mengakhiri permainan dan keluar dari <i>game</i> .	<i>Exit</i>

Sumber : Perancangan

Diagram *use case* dalam perancangan *game* Karapan Sapi 3D digunakan untuk memodelkan fungsionalitas dari *game*. Diagram *use case* ini melibatkan *player* sebagai aktor dan beberapa *use case*. *Use case* merupakan fungsionalitas dari sistem yang diinisialisasi oleh aktor. Setiap *use case* memiliki aktifitas yang berbeda untuk menjelaskan interaksi antara *player* dan sistem yang akan digambarkan pada *activity diagram*. Diagram *use case* untuk *game* Karapan Sapi 3D dibuat dengan mengacu pada tabel identifikasi aktor dan daftar kebutuhan ditunjukkan pada Gambar 4.8.

4.2.2 Class Diagram

Class diagram memberikan gambaran pemodelan elemen *class* dan struktur yang membentuk sebuah perangkat lunak yang dalam hal ini adalah sebuah *game*. *Class* bisa didapatkan dengan menganalisis secara detail terhadap *use case* yang dimodelkan dan menyesuaikan komponen pada *game engine*. Gambar 4.7 dan Tabel 4.14 menunjukkan rancangan *class diagram* beserta penjelasan tiap *class game* Karapan Sapi 3D.

Tabel 4.14 Deskripsi *class diagram* Karapan Sapi 3D

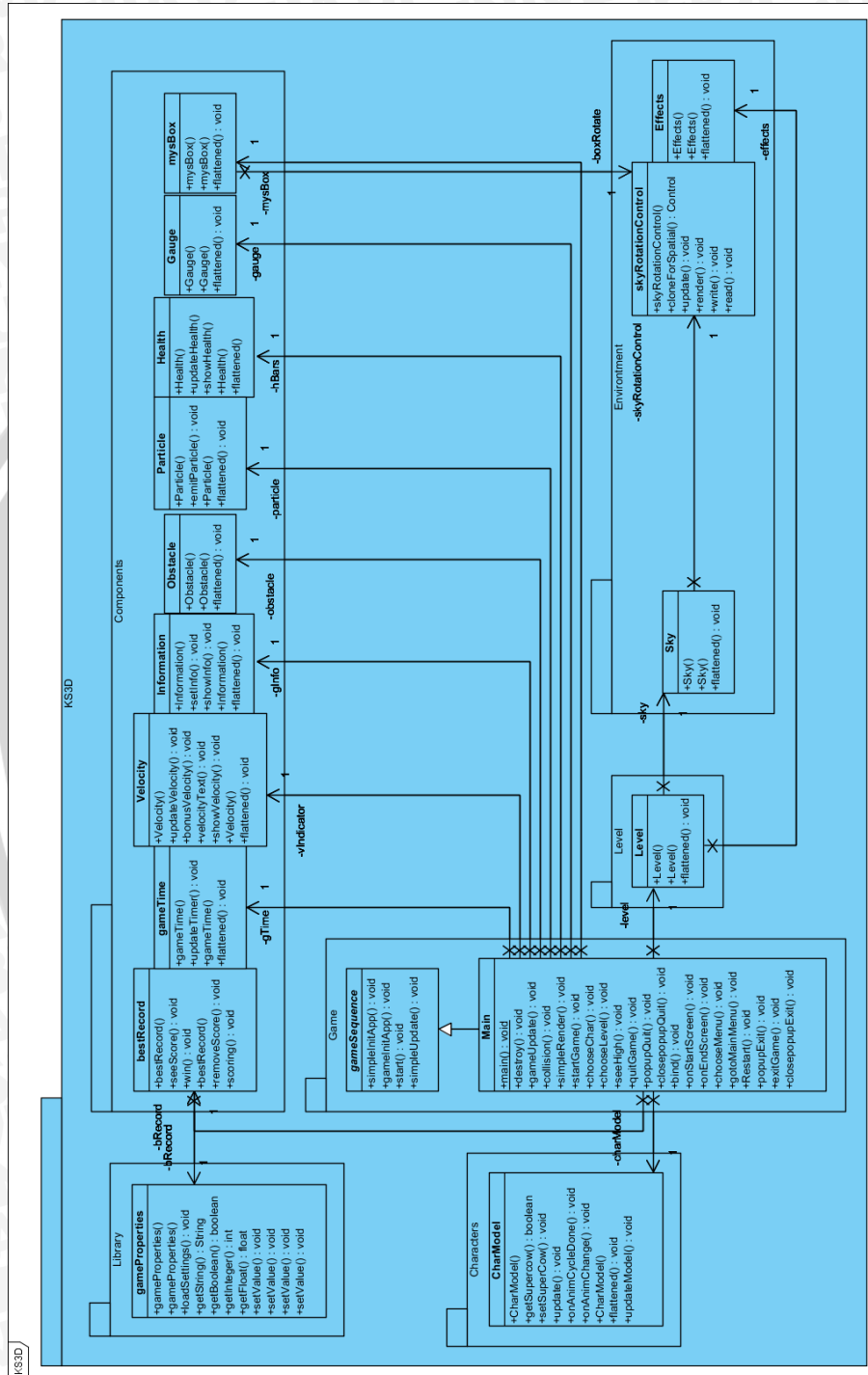
No	Class	Keterangan
1.	<i>Main</i>	<i>Class</i> yang pertama kali di eksekusi saat program dijalankan.
2.	<i>GameSequence</i>	<i>Class</i> yang berfungsi membagi <i>thread</i> yg akan di <i>load</i> pada saat <i>game</i> dijalankan.
3.	<i>charModel</i>	<i>Class</i> yang mengatur fisik setiap karakter yang ada di dalam <i>game</i> .
4.	<i>Level</i>	<i>Class</i> yang mengatur fisik setiap level yang ada di dalam <i>game</i> .
5.	<i>Sky</i>	<i>Class</i> yang membuat tampilan langit pada <i>game</i> .
6.	<i>skyRotationControl</i>	<i>Class</i> yang mengatur putaran (rotasi) langit, dapat juga digunakan untuk mengatur putaran objek lain.
7.	<i>Effects</i>	<i>Class</i> yang membuat tampilan efek seperti cahaya dan bayangan.

8.	<i>Library</i>	<i>Class abstract yang mendeklarasikan akses asset dan rootnode pada game.</i>
9.	<i>GUI</i>	<i>Class abstract yang mendeklarasikan akses setting dan guinode.</i>
10.	<i>gameProperties</i>	<i>Class yang berfungsi untuk melakukan akses data highscore pada file .properties.</i>
11.	<i>Information</i>	<i>Class yang mengatur tampilan informasi buff yang didapat pada game.</i>
12.	<i>Health</i>	<i>Class yang berfungsi untuk memberikan health points pada karakter.</i>
13.	<i>Particle</i>	<i>Class yang mengatur tampilan partikel untuk ledakan mysterybox.</i>
14.	<i>gameTime</i>	<i>Class yang mengatur tampilan waktu game.</i>
15.	<i>Gauge</i>	<i>Class yang mengatur antarmuka pengukur kecepatan dan health points.</i>
16.	<i>bestRecord</i>	<i>Class yang mengatur high score.</i>
17.	<i>mysBox</i>	<i>Class yang mengatur fisik dari mystery box.</i>
18.	<i>Obstacle</i>	<i>Class yang mengatur fisik dari rintangan.</i>
19.	<i>Velocity</i>	<i>Class yang berfungsi untuk memberikan kecepatan pada karakter.</i>

Sumber : Perancangan

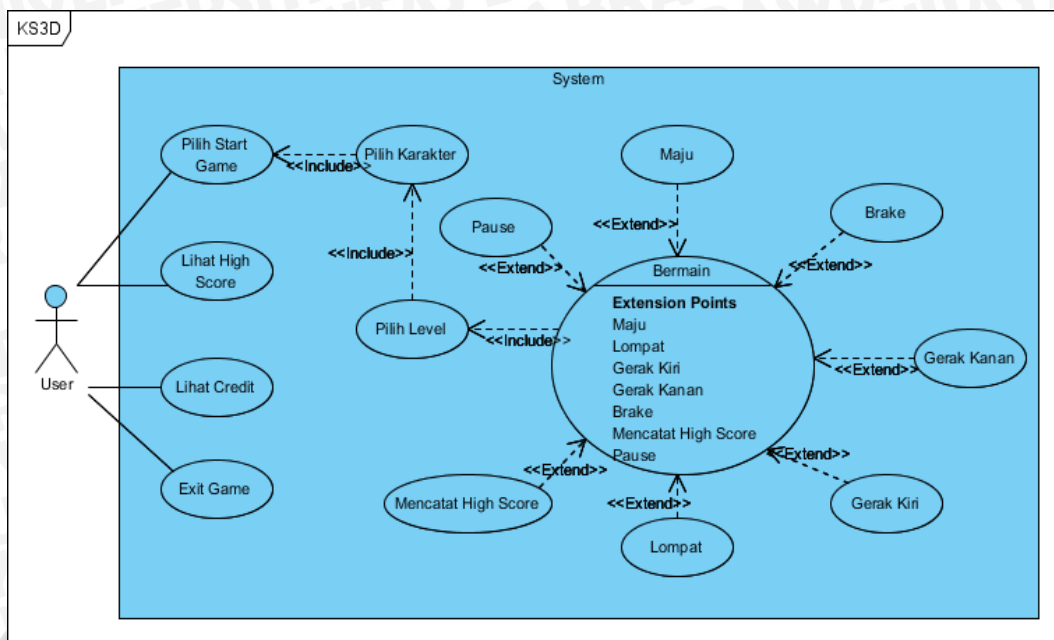
4.2.3 Activity Diagram

Activity diagram dirancang untuk menggambarkan kegiatan yang dapat dilakukan pada sistem sesuai dengan rancangan *use case*. Kegiatan yang dapat dilakukan oleh aktor yang dalam hal ini adalah *player* antara lain memilih menu *start* (bermain), mengontrol karakter, mencatat *high score*, melihat *high score*, melihat *credit*, dan memilih menu *exit*.



Gambar 4.7 Class Diagram Game Karapan Sapi 3D

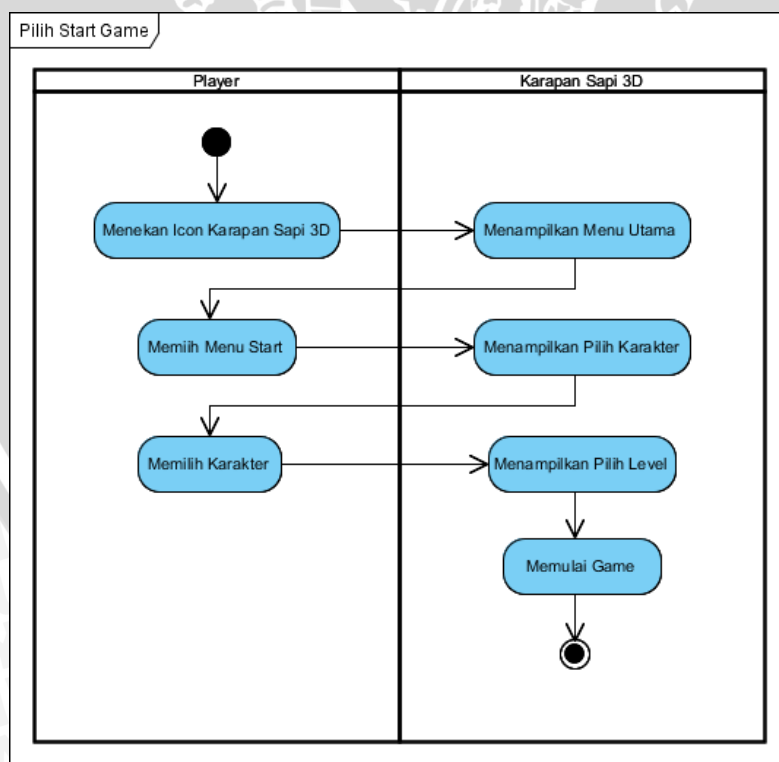
Sumber: Perancangan



Gambar 4.8 Use Case Game Karapan Sapi 3D

Sumber: Perancangan

4.2.3.1 Activity Diagram Start



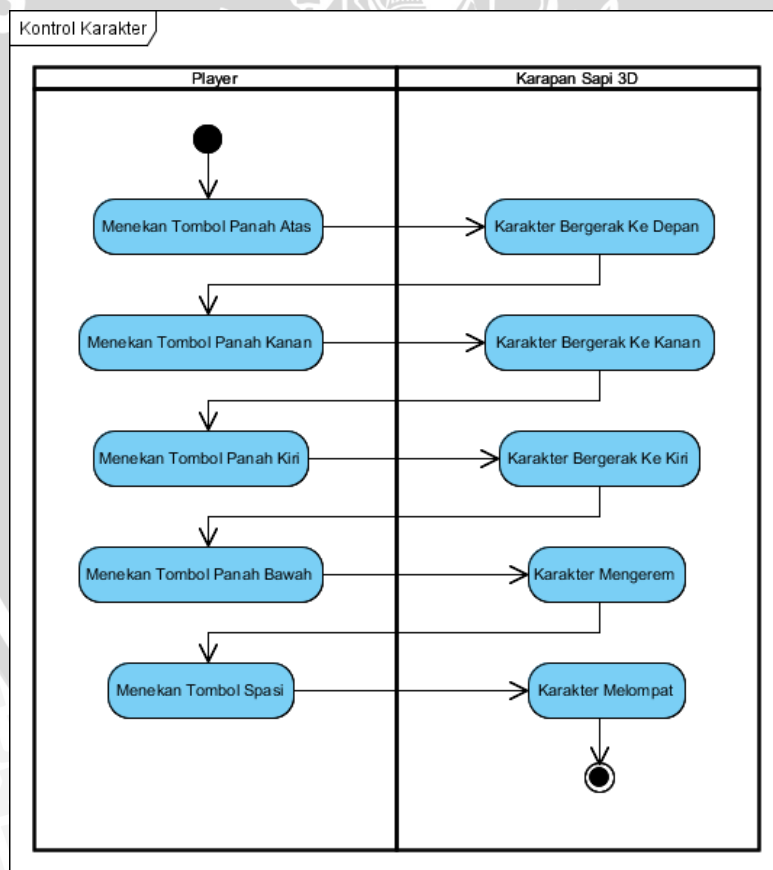
Gambar 4.9 Activity Diagram Start

Sumber: Perancangan

Berdasarkan *activity diagram* pada Gambar 4.9, dapat dilihat bahwa ketika *user* memilih *icon* aplikasi *Game* *Karapan Sapi 3D*, maka sistem akan menampilkan halaman menu, kemudian *user* memilih menu *start* yang terdapat pada menu utama. Sistem memberikan tanggapan dengan menampilkan halaman pilih karakter yang terdapat dua karakter yang dapat dipilih oleh pemain. *User* dapat memilih salah satu karakter, dan sistem akan memberikan tanggapan dengan menampilkan pilihan level. *User* akan dapat bermain setelah memilih level.

4.2.3.2 Activity Diagram Kontrol Karakter

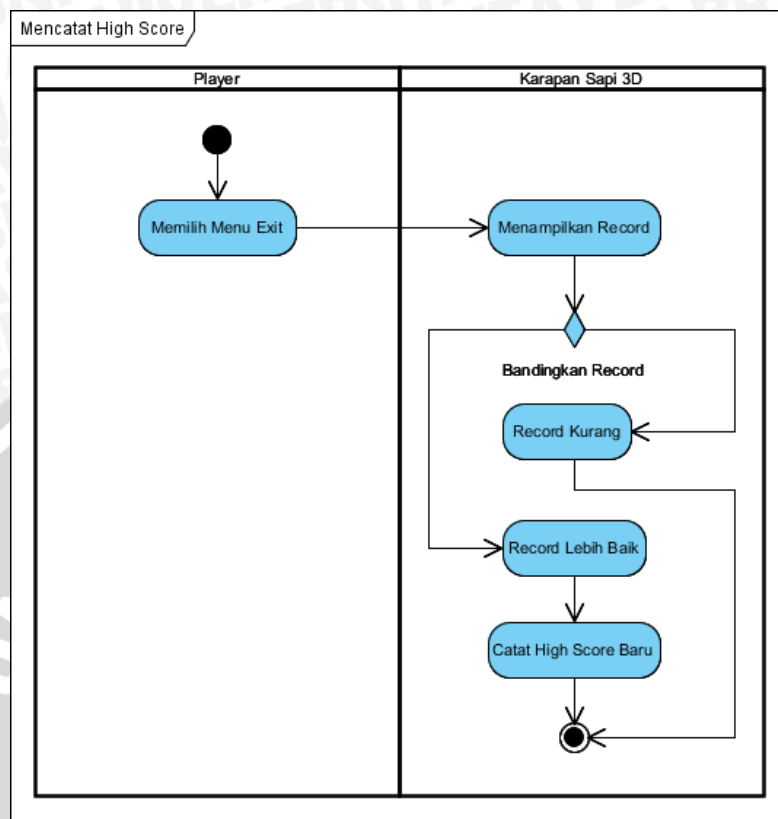
Berdasarkan *activity diagram* pada Gambar 4.10, *user* yang sedang bermain *game* *Karapan Sapi 3D* dapat menekan tombol W, D, A, S, dan spasi. Sistem akan merespon sesuai dengan inputan dari *user*, bila *user* menekan tombol atas, maka karakter akan bergerak ke depan, dan seterusnya.



Gambar 4.10 Activity Diagram Kontrol Karakter

Sumber: Perancangan

4.2.3.3 Activity Diagram Mencatat High Score



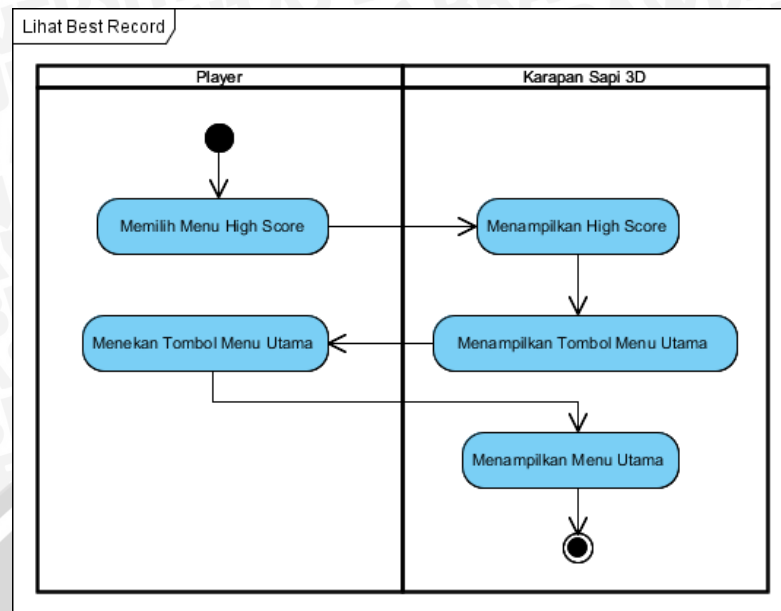
Gambar 4.11 Activity Diagram Mencatat High Score

Sumber: Perancangan

Activity diagram pada Gambar 4.11 menunjukkan setelah *user* mencapai *finish*, sistem akan menampilkan *record* dan melakukan perbandingan dengan *record high score*, apabila *record* lebih baik, maka sistem akan mencatatkan *high score* terbaru, namun apabila *record* kurang dari *record high score*, maka sistem tidak mencatat *high score*.

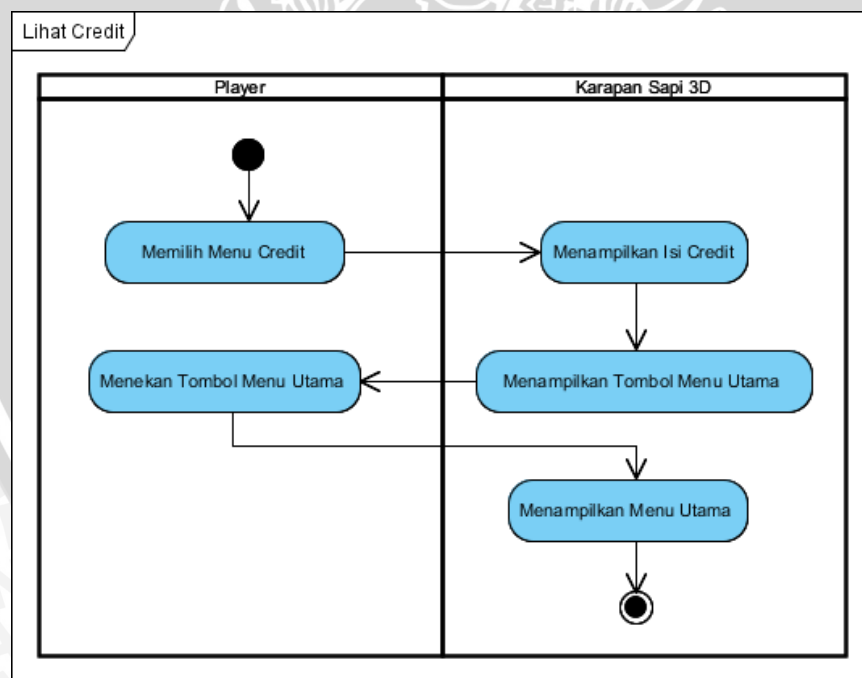
4.2.3.4 Activity Diagram Lihat High Score

Berdasarkan *activity diagram* pada Gambar 4.12, apabila *user* menekan tombol *high score*, maka sistem akan memberikan tanggapan berupa tampilan nilai *high score*. Pada tampilan *high score* terdapat tombol untuk kembali ke menu utama, setelah *user* menekan tombol menu utama maka akan direspon oleh sistem dengan menampilkan kembali menu utama.



Gambar 4.12 Activity Diagram Lihat High Score
Sumber: Perancangan

4.2.3.5 Activity Diagram Lihat Credit

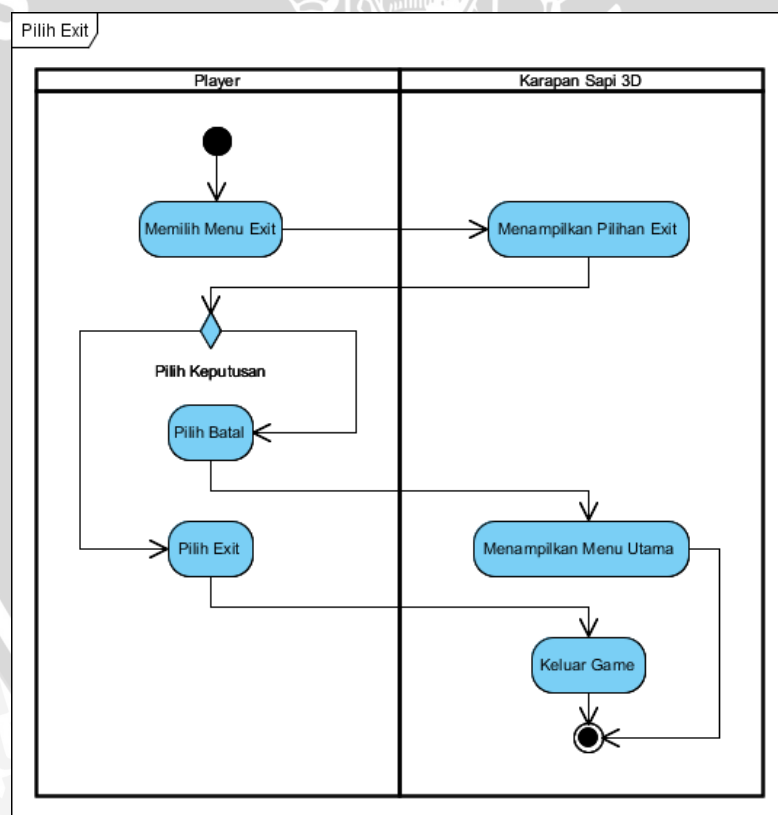


Gambar 4.13 Activity Diagram Lihat Credit
Sumber: Perancangan

Activity diagram pada Gambar 4.13 menunjukkan apabila *user* menekan menu *Credit*, maka sistem akan memberikan tanggapan dengan menampilkan pihak-pihak atau para tersangka yang terlibat dalam pembuatan *game*. Seperti pada tampilan lihat *high score*, pada tampilan lihat *credit* juga terdapat tombol untuk kembali ke menu utama, yang apabila ditekan, akan direspon oleh sistem dengan mengembalikan *user* ke tampilan menu utama.

4.2.3.6 Activity Diagram Exit

Activity diagram pada Gambar 4.14 menunjukkan apabila *user* menekan menu *Exit*, maka sistem akan memberikan tanggapan dengan menampilkan pilihan apakah *user* yakin ingin keluar dari *game*. Apabila *user* menekan ya, maka sistem akan berhenti atau keluar dari *game*. Tetapi apabila *user* memilih batal, maka sistem akan merespon dengan mengembalikan tampilan menu utama.



Gambar 4.14 Activity Diagram Exit
Sumber: Perancangan

BAB V

IMPLEMENTASI

Implementasi membahas tentang hasil yang telah didapatkan dari analisis kebutuhan dan proses perancangan *game* yang menggunakan *Game Design*. Pembahasan terdiri dari penjelasan tentang spesifikasi sistem, batasan-batasan dalam implementasi, implementasi *procedure*, dan implementasi antarmuka aplikasi.

5.1 Spesifikasi Sistem

Game Karapan Sapi 3D dikembangkan dalam lingkungan implementasi yang terdiri dari perangkat keras dan perangkat lunak.

5.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang dipakai dalam proses pengembangan *game* Karapan Sapi 3D adalah sebagai berikut.

1. Intel(R) Core(TM) i5-2410M CPU @ 2.30GHz (4 CPUs), ~2.7GHz
2. HDD 640 GB
3. RAM 6 GB
4. *Graphic Card* :
 - Intel(R) HD Graphics Family (on-board)
 - Nvidia GeForce GT 540M CUDA 2GB (External)
5. *Monitor Generic PnP Monitor Widescreen Display*
6. *Sound Device Speakers (Realtek High Definition Audio)*

5.1.2 Spesifikasi Perangkat Lunak

Pengembangan aplikasi *game* Karapan Sapi 3D ini, menggunakan komputer dengan spesifikasi perangkat lunak sebagai berikut :

1. Sistem operasi Microsoft Windows 7 64 bit, *Ultimate Edition*.
2. *Game engine* jMonkeyEngine SDK 3.0 Beta.
3. Blender 3D versi 2.63.

4. OgreExporter versi 2.49.

5.2 Batasan-Batasan Implementasi

Beberapa batasan dalam mengimplementasikan sistem adalah sebagai berikut:

1. Pengembangan *game* menggunakan *game engine* jMonkey SDK 3.0 beta.
2. *Library* yang digunakan merupakan *library* yang sudah tersedia pada jMonkeyPlatform *game engine*.
3. Pembuatan karakter, animasi, *enviromtent* dan level menggunakan Blender 3D.
4. File 3D yang digunakan untuk *game* Karapan Sapi 3D menggunakan format *.j3o.
5. Konfigurasi *user interface* menggunakan *library* NiftyGUI dan berekstensi .xml.

5.3 Implementasi Artwork

Pada sub bab ini akan di tampilkan hasil implementasi artwork dari *game* Karapan Sapi 3D. *Artwork* yang diimplementasikan berdasarkan pada tahap perancangan yang dibuat.

5.3.1 Implementasi Game Cover

Game cover dapat digunakan untuk tampilan *cover* apabila dilakukan pengepakan atau pada *splash screen* saat memulai *game*. *Game cover* dapat dilihat pada Gambar 5.1.

5.3.2 Implementasi Karakter

Bagian ini menampilkan hasil implementasi karakter menjadi sebuah objek 3D dari tahap perancangan, yaitu Brown dan White. Model 3D dapat dilihat pada Gambar 5.2.

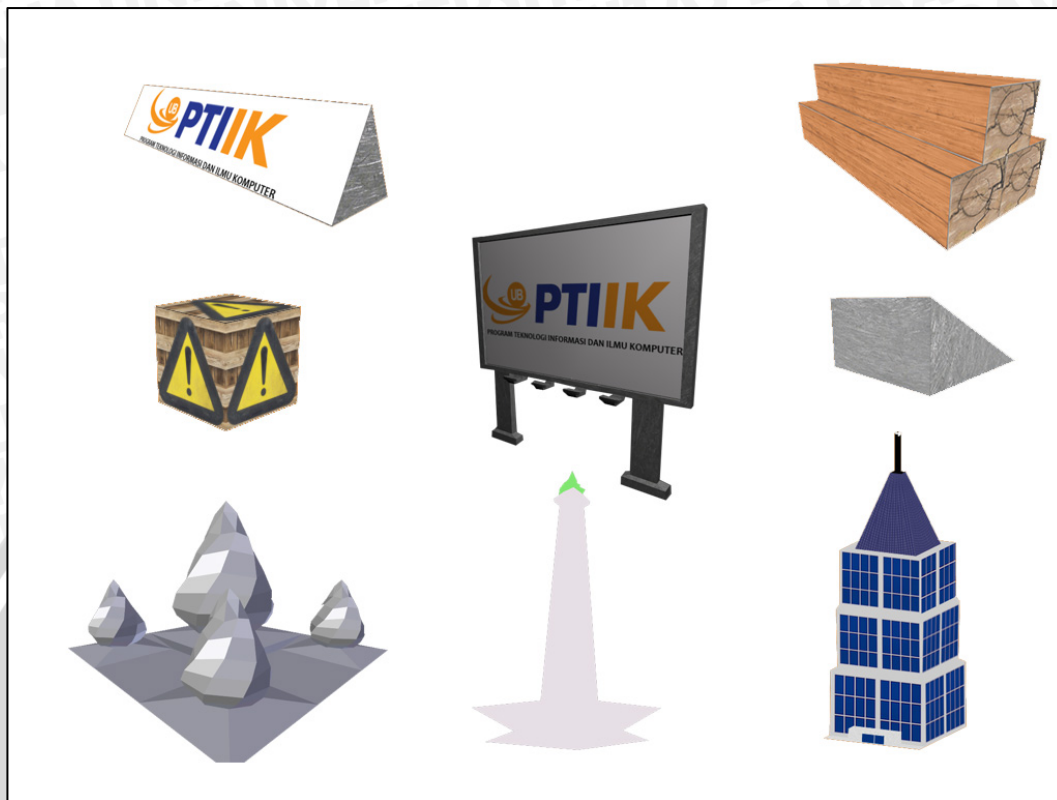


Gambar 5.1 Game Cover Karapan Sapi 3D
Sumber: Implementasi



Gambar 5.2 Implementasi Karakter Game Karapan Sapi 3D
Sumber: Implementasi

5.3.3 Implementasi *Environment*



Gambar 5.3 *Environment* Level Karapan Sapi 3D

Sumber: Implementasi

Environment pada *game* digunakan sebagai objek latar belakang atau objek *in-game*. Pada tahap implementasi, *environment* pada tahap perancangan diimplementasikan menjadi objek 3D. Gambar model 3D dari *environment* dapat dilihat pada Gambar 5.3.

5.4 Implementasi Level

Level diimplementasikan dengan menggabungkan objek 3D pada *environment* menjadi satu *scene*.

5.4.1 Implementasi Level 1

Level 1 pada *game* Karapan Sapi 3D menggunakan gedung Rektorat UB dan gedung PTIIK sebagai latar belakang (objek unik yang tidak ada pada level lainnya). Pada level 1, hanya terdapat sedikit rintangan dan banyak *mystery box*. Hasil implementasi level 1 dapat dilihat pada Gambar 5.4.

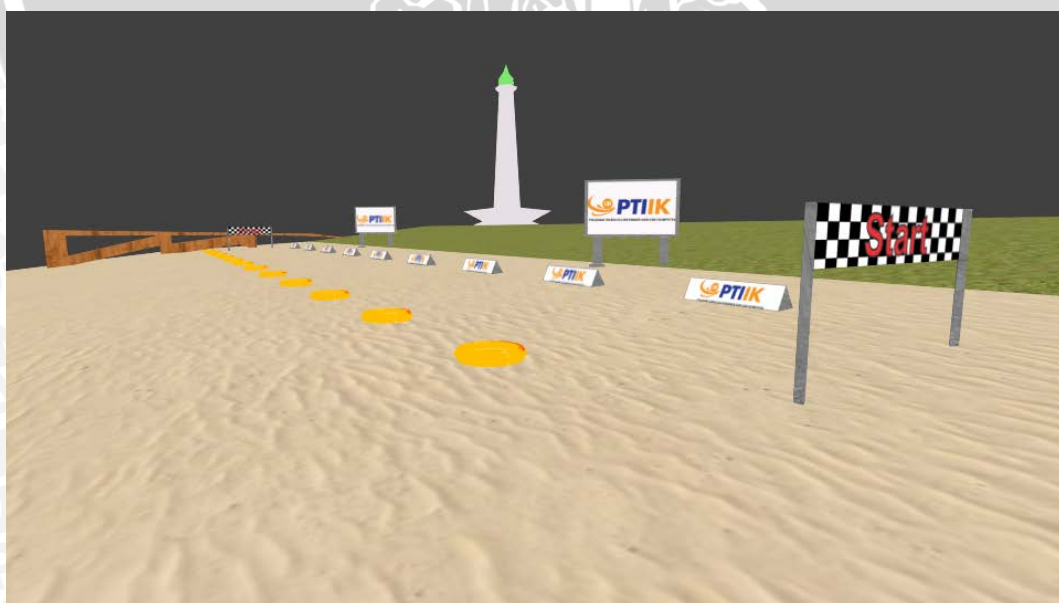


Gambar 5.4 Implementasi Level 1

Sumber: Implementasi

5.4.2 Implementasi Level 2

Level 2 pada *game* Karapan Sapi 3D menggunakan Tugu Monas sebagai latar belakang (objek unik yang tidak ada pada level lainnya). Pada level 2, terdapat rintangan dan *mystery box* dengan jumlah menengah untuk menambah tingkat kesulitan. Hasil implementasi level 2 dapat dilihat pada Gambar 5.5.



Gambar 5.5 Implementasi Level 2

Sumber: Implementasi

5.4.3 Implementasi Level 3



Gambar 5.6 Implementasi Level 3

Sumber: Implementasi

Level 3 pada *game* Karapan Sapi 3D menggunakan Candi Borobudur sebagai latar belakang (objek unik yang tidak ada pada level lainnya). Pada level 3, terdapat banyak rintangan dan sedikit *mystery box* agar tingkat kesulitan *game* paling sulit. Hasil implementasi level 3 dapat dilihat pada Gambar 5.6.

5.5 Implementasi Procedure

Game Karapan Sapi 3D memiliki beberapa fungsi atau *method* yang terdapat pada beberapa *class*. *Procedure* yang dicantumkan pada penulisan skripsi ini hanya *method procedure* dari beberapa proses (operasi) saja sehingga tidak semua *method* akan dicantumkan. Implementasi *procedure* akan direpresentasikan dalam bentuk *pseudo-code* dari *procedure* tersebut.

5.5.1 Implementasi Procedure Scoring()

Scoring procedure digunakan untuk melakukan penghitungan *score* dan perbandingan antara *score* terbaik dengan *score* yang didapatkan. Penghitungan *score* didapatkan dengan memberikan rumus perhitungan dengan beberapa variabel sebagai variabel hitung. Rincian *procedure* dapat dilihat pada Tabel 5.1.

Tabel 5.1 Procedure Scoring()

Elemen	Keterangan
Nama <i>Procedure</i>	Scoring
Masukan	gameTime, minute, second, curHealth, level
Proses	1. Constructor memanggil method <code>setText()</code>. 2. Instansiasi class <code>gameProperties</code> untuk load assets <code>bestRecord.properties</code>. 3. Method <code>setText()</code> membuat text untuk tampilan huruf score a. Ambil jenis font pada asset. b. Instansiasi library game engine <code>BitmapText</code> c. Setting instance <code>BitmapText</code>. d. Masukan <code>BitmapText</code> ke dalam Node. 4. Class main memanggil method <code>win(gameTime, minute, second, curHealth, level)</code> pada class <code>bestRecord</code>. a. Ambil data best record pada file <code>bestRecord.properties</code>. b. Membuat String data best record. 5. Ambil date pada system. 6. Buat String <code>textMsg</code>. 7. Membuat variabel <code>totalScore = 100 * 60/(minute*60+second) + curHealth</code>. 8. Jika <code>bestScore < totalScore</code>. a. Set data best record dengan parameter method <code>scoring()</code>. b. Set <code>textMsg = "YAY! Kamu Baru Saja Memecahkan Rekor"</code>. 9. Jika tidak. a. Set <code>textMsg = "Maaf Kamu Gagal Memecahkan Rekor"</code>. 10. Tampilkan data best record dalam file <code>bestRecord.properties</code>. 11. Tampilkan data current record. 12. Lepas semua node anak dari <code>guiNode</code>. 13. Masukan <code>BitmapText</code> pada <code>guiNode</code>.

Sumber: Implementasi

5.5.2 Implementasi *Procedure* `setSuperCow()`

Procedure `setSuperCow` dilakukan untuk memberikan *buff* supercow saat masukan bernilai benar. *Buff* memberikan efek pada karakter berupa perbesaran ukuran karakter dan mengubah posisi tinggi karakter dan animasi yang unik sehingga karakter terlihat melayang. Rincian *procedure* `setSuperCow` dapat dilihat pada Tabel 5.2.

Tabel 5.2 *Procedure* setSuperCow()

Elemen	Keterangan
Nama <i>Procedure</i>	setSuperCow
Masukan	boolean supercow
Proses	1. This.supercow = false. 2. This.supercow = supercow. 3. Ambil milisecond pada supercowTime. 4. Panggil method setLocalScale(1). 5. Panggil method setGravity(0) pada characterControl. 6. Panggil method setPhysicsLocation 7. Update method updateModel(). a. Jika supercow. i. Ambil milisecond pada current. ii. Panggil method setLocalScale(1). iii. Panggil method setGravity pada characterControl. iv. Panggil method setPhysicsLocation(vector3f(charPosisiX, 5, charPosisiY). v. Jika current - supercowTime > 2000. 1. supercowTime = 0. 2. This.supercow = false. 3. Panggil method setLocalScale(0.5). 4. Panggil method setGravity(25) pada characterControl. vi. Jika tidak. 1. Jika nama animasi != "supercows". a. Set animasi = supercows.

Sumber: Implementasi

5.5.3 Implementasi *Procedure* bonusVelocity()

Procedure bonusVelocity digunakan untuk memberikan *buff* kepada karakter berupa penambahan kecepatan selama beberapa detik. Implementasi *procedure* bonusVelocity secara rinci dapat dilihat pada Tabel 5.3.

5.5.4 Implementasi *Procedure* updateHealth()

Procedure updateHealth dilakukan untuk memberikan tambahan *health points* pada karakter. Masukan updateHealth dapat bernilai positif atau negatif tergantung pada objek *collision* yang ditabrak player. Rincian *procedure* updateHealth dapat dilihat pada Tabel 5.4.

Tabel 5.3 *Procedure* bonusVelocity()

Elemen	Keterangan
Nama <i>Procedure</i>	bonusVelocity
Masukan	bVelocity
Proses	- Boolean bonused = false. - Jika bonused = false. - Velocity += bVelocity. - maxVelocity += bVelocity. - Bonused = true. - Ambil milisecond pada bonusTime - gotBonus = true. - Update method updateVelocity(). - Jika gotBonus = true. - Update milisecond pada current. - Jika current - bonusTime > 2000. - gotBonus = false. - Bonused = false. - Velocity -= bVelocity. - maxVelocity -= bVelocity. - bonusTime = 0.

Sumber: Implementasi

Tabel 5.4 *Procedure* updateHealth()

Elemen	Keterangan
Nama <i>Procedure</i>	updateHealth
Masukan	float updateHealth
Proses	- curHealth += updateHealth. - Jika curHealth > maxHealth. - curHealth = maxHealth. - Namun jika curHealth < minHealth. - curHealth = minHealth.bonusTime = 0.

Sumber: Implementasi

5.5.5 Implementasi *Procedure* updateGameCollision()

Procedure updateGameCollision digunakan untuk mendeteksi tabrakan antara player dengan objek lain dengan *game*. Setelah tabrakan terdeteksi, dapat diberikan statement berupa *game logic* sesuai dengan *gameplay* yang dibuat. Implementasi *procedure* updateGameCollision secara rinci dapat dilihat pada Tabel 5.5.

Tabel 5.5 Procedure updateGameCollision()

Elemen	Keterangan
Nama Procedure	updateGameCollision
Masukan	nodeA, NodeB
Proses	1. Buat nodeA = event.getNodeA(). 2. Buat nodeB = event.getNodeB(). 3. Jika tidak finish. a. Jika nama nodeA = "player". i. Jika nama nodeB = "finishLine". 1. Tampilkan layar finish. 2. Panggil method win(gameTime, minute, second, curHealth, level). 3. Finish = true. b. Jika nama nodeB = "player". i. Jika nama nodeA = "obs". 1. Mengurangi health point. 2. Mengurangi kecepatan. 3. Jika curHealth = 0. a. Tampilkan layar gameOver. b. Lepas semua node anak pada guiNode. c. Masukan scoreAll pada guiNode. d. Set scoreAll = "". e. Finish = true. ii. Jika nama nodeA = "mysBox" 1. Lepas nodeA dari node parent. 2. Lepas bulletState dari nodeA. 3. Memancarkan particle. 4. Ambil tipe data box. a. Jika tipe = 0. i. Menampilkan text HP+25. ii. Panggil updateHealth(12.5). b. Namun jika tipe = 1. i. Menampilkan text speed+10. ii. Panggil bonusVelocity(10). c. Jika tidak. i. Menampilkan text SUPERCOW! ii. Set supercow = true. 4. Selesai semua jika.

Sumber: Implementasi

5.6 Implementasi Antarmuka

Implementasi antarmuka *game* Karapan Sapi 3D dibuat sesuai dengan tema dari *game* yaitu kebudayaan. *Background* pada interface menggunakan batik Madura. Font yang digunakan dibuat dengan tampilan yang lucu agar sesuai dengan target usia. Implementasi antarmuka berupa tampilan menu utama, tampilan menu pilih karakter, tampilan menu pilih level, tampilan *high score*, tampilan *credit* dan tampilan *in-game*.

5.6.1 Implementasi Main Menu

Halaman menu ditampilkan pertama kali ketika *user* memasuki permainan Karapan Sapi 3D. Halaman menu merupakan halaman yang menampilkan sejumlah menu utama yang dapat digunakan *user* untuk mengakses *game*, baik memulai *game*, atau melihat informasi yang terdapat pada *game*. Menu yang terdapat pada tampilan menu utama adalah menu *Start*, menu *Best level*, Menu *Credit*, dan menu *Exit*. Implementasi antarmuka halaman menu utama dapat dilihat pada Gambar 5.7.

5.6.2 Implementasi Sub-menu Choose Character

Tampilan *choose character* merupakan tampilan yang muncul setelah *user* memilih menu *start*. Tampilan *choose character* memungkinkan *user* dapat memilih karakter yang akan menjadi karakter *player* utama dalam permainan yang akan dimainkan. Implementasi tampilan *choose character* dapat dilihat pada Gambar 5.8.

5.6.3 Implementasi Sub-menu Choose Level

Tampilan menu *choose level* merupakan tampilan yang akan muncul setelah *user* memilih menu *choose character*. *User* dapat memilih level yang digunakan sebagai *scene* utama *game*. Implementasi halaman pilih karakter dapat dilihat pada Gambar 5.9.

5.6.4 Implementasi In-Game Screen

Tampilan permainan akan ditampilkan setelah *user* memilih karakter. Pada tampilan ini *user* dapat mulai memainkan permainan dengan menggerakkan karakter *player* utama menggunakan kontrol *keyboard* sebagai masukan kepada *game*. Implementasi tampilan permainan dapat dilihat pada Gambar 5.10.



Gambar 5.7 Menu Utama *Game Karapan Sapi 3D*
Sumber: Implementasi



Gambar 5.8 Menu *Choose Character*
Sumber: Implementasi



Gambar 5.9 Menu *Choose Level*
Sumber: Implementasi



Gambar 5.10 Interface *In-Game* Karapan Sapi 3D
Sumber: Implementasi

BAB VI

PENGUJIAN DAN ANALISIS

Tahapan pengujian *game* Karapan Sapi 3D dilakukan dengan metode *whitebox*, *blackbox*, dan pengujian performa untuk mendapatkan hasil pengujian. Tahapan analisis dilakukan untuk mendapatkan kesimpulan dari hasil pengujian yang didapatkan.

6.1 Pengujian

Proses pengujian yang dilakukan melalui tiga tahapan yaitu pengujian *whitebox* yang terdiri dari pengujian unit dan pengujian integrasi, pengujian *blackbox*, dan pengujian performa dilakukan dengan pengujian *frame rate* untuk mengetahui performa FPS (*frame-per-second*) pada *game* Karapan Sapi 3D.

6.1.1 Pengujian *Whitebox*

Pada pengujian *whitebox* dilakukan dengan dua pengujian yaitu pengujian unit dan pengujian testing. Pengujian unit dilakukan untuk mengetahui apakah setiap komponen bekerja secara terpisah telah bekerja sesuai harapan. Pengujian unit menggunakan metode *Whitebox Testing* dengan teknik *Basis path Testing*. Pada teknik *Basis path Testing* proses pengujian dilakukan dengan memodelkan *procedure* pada sebuah *flow graph*, menentukan *cyclometric complexity* dan melakukan uji kasus untuk setiap *path* yang ada.

6.1.1.1 Pengujian Unit *Procedure Scoring()*

Procedure scoring digunakan pada *method scoring(String gameTime, long minute, long second, float curHealth, int level)* yang merupakan salah satu *method* pada class *bestRecord* untuk mendapatkan hasil perhitungan dan perbandingan nilai *score* kemudian mencatat data *best record* apabila kondisi untuk mencatatkan *best record* terpenuhi.

Tabel 6.1 Pengujian *Procedure Scoring()*

Elemen	Keterangan	
Nama <i>Procedure</i>	Scoring	
Masukan	gameTime, minute, second, curHealth, level	
Diskripsi	Proses	Node
	1. Constructor memanggil method <code>setupText()</code> . 2. Instansiasi class <code>gameProperties</code> untuk load assets <code>bestRecord.properties</code> . 3. Method <code>setupText()</code> membuat text untuk tampilan huruf score d. Ambil jenis font pada asset. e. Instansiasi library game engine <code>BitmapText</code> f. Setting instance <code>BitmapText</code> . e. Masukan <code>BitmapText</code> ke dalam Node. 4. Class main memanggil method <code>win(gameTime, minute, second, curHealth, level)</code> pada class <code>bestRecord</code> . a. Ambil data best record pada file <code>bestRecord.properties</code> . b. Membuat String data best record. 5. Ambil date pada system. 6. Buat String <code>textMsg</code> . 7. Membuat variabel <code>totalScore = 100 *</code> 8. <code>60/(minute*60+second) + curHealth</code> .	1
	9. Jika <code>bestScore < totalScore</code> . a. Set data best record dengan parameter method <code>scoring()</code> . b. Set <code>textMsg = "YAY! Kamu Baru Saja Memecahkan Rekor"</code> .	2 3
	10. Jika tidak. a. Set <code>textMsg = "Maaf Kamu Gagal Memecahkan Rekor"</code> .	4
	11. Tampilkan data best record dalam file <code>bestRecord.properties</code> . 12. Tampilkan data current record. 13. Lepas semua node anak dari <code>guiNode</code> . 14. Masukan <code>BitmapText</code> pada <code>guiNode</code> .	5

Sumber: Pengujian dan Analisis

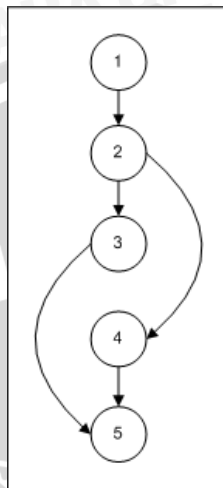
Dari *Procedure* pada Tabel 6.1, didapatkan *flowgraph* seperti pada Gambar 6.1. Pemodelan ke dalam *flowgraph* yang telah menghasilkan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 5 - 5 + 2 \\
 &= 2
 \end{aligned}$$

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu ditentukan satu buah basis set dari jalur independent yaitu:

Jalur 1 : 1 - 2 - 3 - 5

Jalur 2 : 1 - 2 - 4 - 5



Gambar 6.1 Flowgraph Procedure Scoring()

Sumber: Pengujian dan Analisis

Tabel 6.2 Test Case Pengujian Procedure Scoring(gameTime, minute, second, curHealth, level)

No	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Mendapatkan totalScore > bestScore	Mencatat <i>high score</i> dan menampilkan pesan catat <i>record</i>	Mencatat <i>high score</i> dan menampilkan pesan catat <i>record</i>
2	Mendapatkan totalScore < bestScore	Tidak mencatat <i>high score</i> dan menampilkan pesan gagal catat <i>record</i>	Tidak mencatat <i>high score</i> dan menampilkan pesan gagal catat <i>record</i>

Sumber: Pengujian dan Analisis

6.1.1.2 Pengujian Unit *Procedure* setSuperCow()

Procedure setSuperCow digunakan pada *method* setSuperCow(boolean supercow) yang merupakan salah satu *method* pada class charModel untuk mendapatkan nilai yang digunakan untuk menentukan *buff* supercow pada karakter.

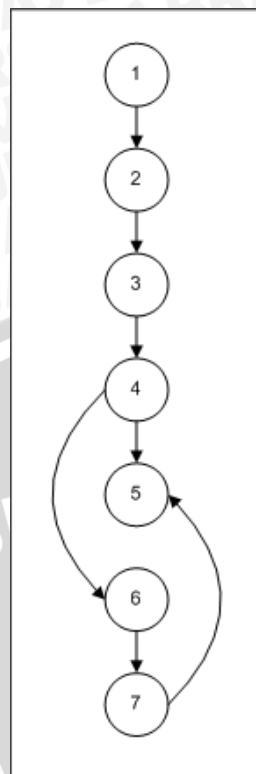
Tabel 6.3 Pengujian *Procedure* setSuperCow()

Elemen	Keterangan	
Nama <i>Procedure</i>	setSuperCow	
Masukan	boolean supercow	
Deskripsi	Proses	Node
	1. This.supercow = false. 2. This.supercow = supercow. 3. Ambil milisecond pada supercowTime. 4. Update method updateModel().	1
	a. Jika supercow.	2
	i. Ambil milisecond pada current. ii. Panggil method setLocalScale(1). iii. Panggil method setGravity pada characterControl. iv. Panggil method setPhysicsLocation(vector3f(charPosi siX, 5, charPosisiY).	3
	v. Jika current - supercowTime > 2000.	4
	1. supercowTime = 0. 2. This.supercow = false. 3. Panggil method setLocalScale(0.5). 4. Panggil method setGravity(25) pada characterControl.	5
	vi. Jika tidak. 1. Jika nama animasi != "supercows". a. Set animasi = supercows.	6
		7

Sumber: Pengujian dan Analisis

Dari *Procedure* pada Tabel 6.3, didapatkan *flowgraph* seperti pada Gambar 6.2. Pemodelan ke dalam *flowgraph* yang telah menghasilkan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 7 - 7 + 2 \\
 &= 2
 \end{aligned}$$



Gambar 6.2 Flowgraph Procedure setSuperCow()

Sumber: Perancangan

Tabel 6.4 Test Case Pengujian Procedure setSuperCow(boolean supercow)

No	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Mendapatkan buff supercow dan waktu < 2 detik	Posisi karakter melayang, ukuran membesar dan animasi supercow dimainkan	Posisi karakter melayang dan animasi supercow dimainkan
2	Mendapatkan buff supercow dan waktu > 2 detik	Posisi kembali ke tanah, ukuran normal dan animasi supercow tidak dimainkan	Posisi kembali ke tanah, ukuran normal dan animasi supercow tidak dimainkan

Sumber: Pengujian dan Analisis

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu ditentukan satu buah basis set dari jalur independent yaitu:

Jalur 1 : 1 - 2 - 3 - 4 - 6 - 7 - 5

Jalur 2 : 1 - 2 - 3 - 4 - 5

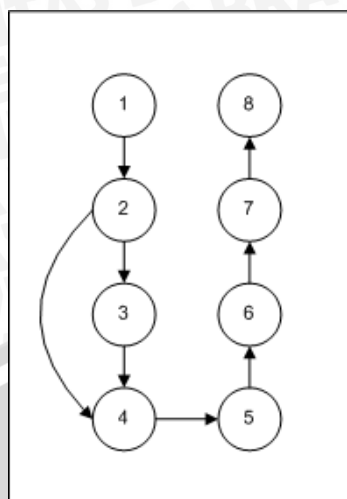
6.1.1.3 Pengujian Unit *Procedure* bonusVelocity()

Procedure bonusVelocity digunakan pada *method* bonusVelocity(float bonusVelocity) yang merupakan salah satu *method* pada class velocity untuk memberikan nilai bonus yang akan diberikan pada karakter.

Tabel 6.5 Pengujian *Procedure* bonusVelocity()

Elemen	Keterangan	
Nama <i>Procedure</i>	bonusVelocity	
Masukan	bVelocity	
Deskripsi	Proses	Node
	1. This.bVelocity = bVelocity.	1
	2. Jika bonused = false.	2
	a. Velocity += bVelocity. b. maxVelocity += bVelocity. c. Bonused = true.	3
	3. Ambil milisecond pada bonusTime 4. gotBonus = true. 5. Update method updateVelocity().	4
	a. Jika gotBonus = true.	5
	i. Update milisecond pada current.	6
	ii. Jika current - bonusTime > 2000.	7
	1. gotBonus = false. 2. Bonused = false. 3. Velocity -= bVelocity. 4. maxVelocity -= bVelocity. 5. bonusTime = 0.	8

Sumber: Pengujian dan Implementasi



Gambar 6.3 Flowgraph Procedure bonusVelocity()
Sumber: Perancangan

Tabel 6.6 Test Case Pengujian Procedure bonusVelocity(float bVelocity)

No	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Mendapatkan tambahan kecepatan dan waktu < 2 detik	Kecepatan maksimum karakter bertambah 10	Kecepatan maksimum karakter bertambah 10
2	Mendapatkan tambahan kecepatan dan waktu > 2 detik	Kecepatan maksimum karakter kembali ke awal	Kecepatan maksimum karakter kembali ke awal

Sumber: Pengujian dan Analisis

Dari Procedure pada Tabel 6.5 didapatkan flowgraph seperti pada Gambar 6.3. Pemodelan ke dalam flowgraph yang telah menghasilkan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi (garis penghubung antar node) dan N merupakan jumlah simpul (node).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 8 - 8 + 2 \\
 &= 2
 \end{aligned}$$

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu ditentukan satu buah basis set dari jalur independent yaitu:

Jalur 1 : 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8

Jalur 2 : 1 - 2 - 4 - 5 - 6 - 7 - 8

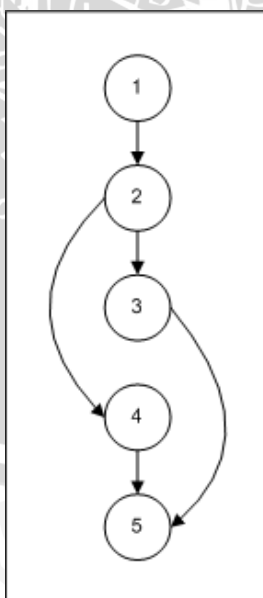
6.1.1.4 Pengujian Unit *Procedure* updateHealth()

Procedure updateHealth digunakan pada *method* updateHealth(float updateHealth) yang merupakan salah satu *method* pada class Health untuk memberikan nilai besarnya health yang akan diubah.

Tabel 6.7 Pengujian *Procedure* updateHealth()

Elemen	Keterangan	
Nama <i>Procedure</i>	updateHealth	
Masukan	float updateHealth	
Deskripsi	Proses	Node
	1. curHealth += updateHealth.	1
	2. Jika curHealth > maxHealth.	2
	a. curHealth = maxHealth.	3
	3. Namun jika curHealth < minHealth. a. curHealth = minHealth.bonusTime = 0.	4
	4. Selesai	5

Sumber: Perancangan



Gambar 6.4 Flowgraph *Procedure* updateHealth()

Sumber: Pengujian dan Implementasi

Tabel 6.8 Test Case Pengujian *Procedure* updateHealth(float updateHealth)

No	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Menambah HP pada kondisi $\text{curHealth} > \text{maxHealth}$	$\text{curHealth} = \text{maxHealth}$	$\text{curHealth} = \text{maxHealth}$
2	Mengurangi HP pada kondisi $\text{curHealth} < \text{minHealth}$	$\text{curHealth} = \text{minHealth}$	$\text{curHealth} = \text{minHealth}$

Sumber: Pengujian dan Analisis

Dari *Procedure* pada Tabel 6.7 didapatkan *flowgraph* seperti pada Gambar 6.4. Pemodelan ke dalam *flowgraph* yang telah menghasilkan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 4 - 4 + 2 \\
 &= 1
 \end{aligned}$$

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu ditentukan satu buah basis set dari jalur independent yaitu:

Jalur 1 : 1 - 2 - 4 - 5

Jalur 2 : 1 - 2 - 3 - 5

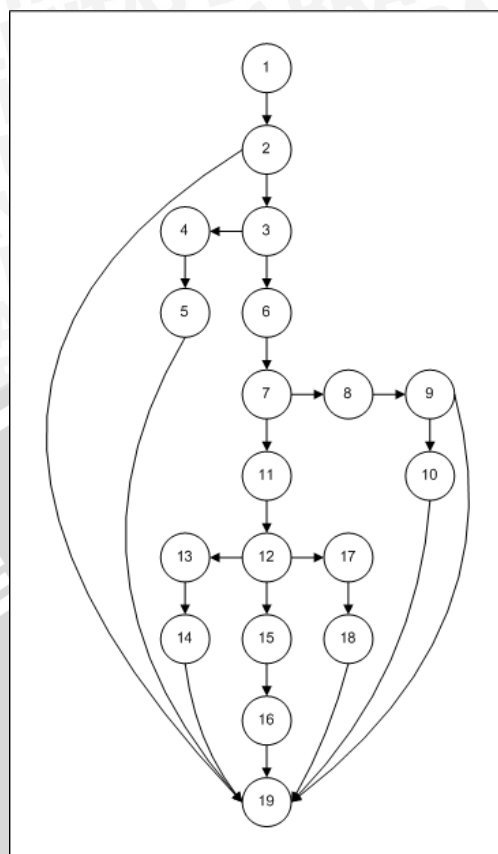
6.1.1.5 Pengujian Integrasi *Procedure* updateGameCollision()

Procedure updateGameCollision terdapat pada *method* updateGameCollision(PhysicsCollisionEvent event) yang merupakan salah satu *method* pada class Main untuk mendeteksi adanya tabrakan antar objek.

Tabel 6.9 Pengujian *Procedure* updateGameCollision()

Elemen	Keterangan	
Nama <i>Procedure</i>	updateGameCollision	
Masukan	nodeA, NodeB	
Deskripsi	Proses	Node
	1. Buat nodeA = event.getNodeA().	1
	2. Buat nodeB = event.getNodeB().	
	3. Jika tidak finish.	2
	a. Jika nama nodeA = "player".	3
	i. Jika nama nodeB = "finishLine".	4
	1. Tampilkan layar finish.	5
	2. Panggil method win(gameTime, minute, second, curHealth, level).	
	3. Finish = true.	
	b. Jika nama nodeB = "player".	6
	i. Jika nama nodeA = "obs".	7
	1. Mengurangi health point.	8
	2. Mengurangi kecepatan.	
	3. Jika curHealth = 0.	9
	a. Tampilkan layar gameOver.	10
	b. Lepas semua node anak pada guiNode.	
	c. Masukkan scoreAll pada guiNode.	
	d. Set scoreAll = "".	
	e. Finish = true.	
	ii. Jika nama nodeA = "mysBox".	11
	1. Lepas nodeA dari node parent.	12
	2. Lepas bulletState dari nodeA.	
	3. Memancarkan particle.	
	4. Ambil tipe data box.	
	a. Jika tipe = 0.	13
	i. Menampilkan text HP+25.	14
	ii. Panggil updateHealth(12.5).	
	b. Namun jika tipe = 1.	15
	i. Menampilkan text speed+10.	16
	ii. Panggil bonusVelocity(10).	
	c. Jika tidak.	17
	i. Menampilkan text SUPERCOW!	18
	ii. Set supercow = true.	
	4. Selesai semua jika.	19

Sumber: Pengujian dan Analisis



Gambar 6.5 Flowgraph Procedure updateGameCollision()

Sumber: Pengujian dan Analisis

Tabel 6.10 Test Case Pengujian Procedure

updateGameCollision(PhysicsCollisionEvent event)

No	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	<i>Finish = true</i>	Tidak ada collision	Tidak ada collision
2	Player bertabrakan dengan box dan tipe data box = 0	HP bertambah dan menampilkan text HP+25	HP bertambah dan menampilkan tex HP+25
3	Player bertabrakan dengan box dan Tipe data box = 1	Speed bertambah dan menampilkan text speed+10	Speed bertambah dan menampilkan text speed+10
4	Player bertabrakan dengan box dan Tipe data box = 2	Mendapatkan supercow dan menampilkan text SUPERCOW!	Mendapatkan supercow dan menampilkan text SUPERCOW!

5	Player bertabrakan dengan obs	HP berkurang dan speed berkurang	HP berkurang dan speed berkurang
6	player bertabrakan dengan obs dan HP = 0	Menampilkan layar <i>game over</i>	Menampilkan layar <i>game over</i>
7	Player bertabrakan dengan <i>finish</i>	Menampilkan layar <i>finish</i> dan memanggil <i>scoring procedure</i>	Menampilkan layar <i>finish</i> dan memanggil <i>scoring procedure</i>

Sumber: Pengujian dan Analisis

Dari *Procedure* pada Tabel 6.9 didapatkan *flowgraph* seperti pada Gambar 6.5. Pemodelan ke dalam *flowgraph* yang telah menghasilkan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi (garis penghubung antar *node*) dan N merupakan jumlah simpul (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 24 - 19 + 2 \\
 &= 7
 \end{aligned}$$

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu ditentukan satu buah basis set dari jalur independent yaitu:

Jalur 1 : 1 - 2 - 19

Jalur 2 : 1 - 2 - 3 - 6 - 7 - 11 - 12 - 13 - 14 - 19

Jalur 3 : 1 - 2 - 3 - 6 - 7 - 11 - 12 - 15 - 16 - 19

Jalur 4 : 1 - 2 - 3 - 6 - 7 - 11 - 12 - 17 - 18 - 19

Jalur 5 : 1 - 2 - 3 - 6 - 7 - 8 - 9 - 10 - 19

Jalur 6 : 1 - 2 - 3 - 6 - 7 - 8 - 9 - 19

Jalur 7 : 1 - 2 - 3 - 4 - 5 - 19

6.1.2 Pengujian *Blackbox*

Pengujian *blackbox* digunakan pada pengujian validasi untuk mengetahui apakah sistem yang dibangun sudah benar sesuai dengan yang dibutuhkan. Daftar kebutuhan dan hasil analisis kebutuhan akan menjadi acuan untuk melakukan pengujian validasi. Pengujian validasi menggunakan metode pengujian *Blackbox*, karena tidak diperlukan konsentrasi terhadap alur jalannya *procedure game* dan

lebih ditekankan untuk menemukan konformitas antara kinerja sistem dengan *usecase diagram*. Pada skripsi ini dilakukan pengujian validasi terhadap *game* Karapan Sapi 3D.

6.1.2.1 Kasus Uji Mulai *Game*

Pada Tabel 6.11 ditunjukkan *test case* ketika pemain menekan menu *start* untuk memulai permainan.

Tabel 6.11 *Test Case* Mulai *Game*

Nama Kasus Uji	Mulai <i>Game</i>
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa <i>game</i> dapat memenuhi kebutuhan fungsional untuk memulai <i>game</i> .
Prosedur Uji	Pemain menekan menu <i>start</i> pada tampilan menu utama.
Kondisi yang diharapkan	Pemain dapat masuk ke menu pilih karakter (<i>choose character</i>)
Hasil yang Didapatkan	Valid

Sumber: Pengujian dan Analisis

6.1.2.2 Kasus Uji Pilih Karakter

Pada Tabel 6.12 ditunjukkan *test case* ketika pemain telah menekan menu *start* untuk memulai permainan dan masuk tampilan pilih karakter.

Tabel 6.12 *Test Case* Pilih Karakter

Nama Kasus Uji	Pilih Karakter
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa <i>game</i> dapat memenuhi kebutuhan fungsional untuk memilih karakter yang akan dimainkan.
Prosedur Uji	1. Pemain menekan <i>icon</i> gambar karakter. 2. Pemain menekan <i>icon</i> karakter Brown. 3. Pemain menekan <i>icon</i> karakter White.
Kondisi yang	1. Masuk tampilan pilih level.

diharapkan	2. Karakter yang dimainkan sesuai dengan <i>icon</i> karakter Brown 3. Karakter yang dimainkan sesuai dengan <i>icon</i> karakter White
Hasil yang Didapatkan	Valid

Sumber: Pengujian dan Analisis

6.1.2.3 Kasus Uji Pilih Level

Pada Tabel 6.13 ditunjukkan *test case* ketika pemain telah menekan menu *start* untuk memulai permainan, telah memilih karakter dan masuk tampilan pilih level.

Tabel 6.13 Test Case Untuk Pilih Level

Nama Kasus Uji	Pilih Level
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa <i>game</i> dapat memenuhi kebutuhan fungsional untuk memilih level yang akan dipakai dalam <i>game</i> .
Prosedur Uji	1. Pemain menekan <i>icon</i> gambar level. 2. Pemain menekan <i>icon</i> level Brawijaya. 3. Pemain menekan <i>icon</i> level Monas. 4. Pemain menekan <i>icon</i> level Borobudur.
Kondisi yang diharapkan	1. Pemain masuk ke tampilan permainan. 2. Level sesuai dengan <i>icon</i> level Brawijaya. 3. Level sesuai dengan <i>icon</i> level Monas. 4. Level sesuai dengan <i>icon</i> level Borobudur.
Hasil yang Didapatkan	Valid

Sumber: Pengujian dan Analisis

6.1.2.4 Kasus Uji Mulai Bermain

Pada Tabel 6.14 ditunjukkan *test case* ketika pemain telah menekan menu *start* untuk memulai permainan, telah memilih karakter, telah memilih level, dan masuk tampilan *game*.

Tabel 6.14 *Test Case* Untuk Mulai Bermain

Nama Kasus Uji	Mulai Bermain
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa <i>game</i> dapat memenuhi kebutuhan fungsional untuk memulai permainan.
Prosedur Uji	Pemain menekan tombol <i>start</i> .
Kondisi yang diharapkan	1. Tampil informasi waktu. 2. Waktu dimulai dari 0. 3. Tampil informasi kecepatan. 4. Tampilan informasi kecepatan sesuai karakter. 5. Tampil informasi <i>health points</i>. 6. Tampilan informasi <i>health points</i> sesuai karakter.
Hasil yang Didapatkan	Valid

Sumber: Pengujian dan Analisis

6.1.2.5 Kasus Uji Bergerak

Pada Tabel 6.15 ditunjukkan *test case* ketika pemain telah menekan menu *start* untuk memulai permainan, telah memilih karakter, telah memilih level, masuk tampilan *game*, dan memulai gerakan.

Tabel 6.15 *Test Case* Untuk Bergerak

Nama Kasus Uji	Bergerak
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa <i>game</i> dapat memenuhi kebutuhan fungsional untuk pergerakan karakter.
Prosedur Uji	1. Pemain menekan tombol W pada <i>keyboard</i>. 2. Pemain menekan tombol S pada <i>keyboard</i>.

	3. Pemain menekan tombol A pada <i>keyboard</i>. 4. Pemain menekan tombol D pada <i>keyboard</i>. 5. Pemain menekan tombol spasi pada <i>keyboard</i>.
Kondisi yang diharapkan	1. Karakter bergerak ke depan. 2. Karakter melakukan pengereman. 3. Karakter bergerak ke kiri. 4. Karakter bergerak ke kanan. 5. Karakter melompat.
Hasil yang Didapatkan	Valid

Sumber: Pengujian dan Analisis

6.1.2.6 Kasus Uji *Game State*

Pada Tabel 6.16 ditunjukkan *test case* ketika pemain telah menekan menu *start* untuk memulai permainan, telah memilih karakter, telah memilih level, dan masuk tampilan *game*, mulai bermain dan berada dalam *game state*.

Tabel 6.16 *Test Case Game State*

Nama Kasus Uji	<i>Game State</i>
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa <i>game</i> dapat memenuhi kebutuhan fungsional untuk mendapatkan <i>game state</i> sesuai dengan <i>gameplay</i> yang dirancang.
Prosedur Uji	1. Karakter menabrak objek rintangan. 2. Karakter menabrak objek rintangan dan HP mencapai 0. 3. Karakter menabrak objek <i>mystery box</i>. 4. Karakter mencapai garis <i>finish</i>. 5. Menekan tombol P pada <i>keyboard</i> pada saat permainan. 6. Menekan tombol P pada <i>keyboard</i> pada saat kondisi <i>pause</i>.
Kondisi yang diharapkan	1. <i>Health points</i> dan kecepatan berkurang. 2. Menampilkan <i>game over</i>. 3. Mendapatkan random <i>buff</i> tambah kecepatan/tambah

	HP/supercow!. 4. Menampilkan rekor dan mencatat rekor baru/tidak. 5. Menampilkan <i>pause</i>. 6. <i>Game unpause</i>.
Hasil yang Didapatkan	Valid

Sumber: Pengujian dan Analisis

6.1.2.7 Kasus Uji Melihat *High Score*

Pada Tabel 6.17 ditunjukkan *test case* ketika pemain telah menekan menu *high score* untuk melihat *high score* pada *game*.

Tabel 6.17 *Test Case* Untuk Melihat *High Score*

Nama Kasus Uji	Melihat <i>high score</i>
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa <i>game</i> dapat memenuhi kebutuhan fungsional untuk melihat tampilan <i>high score</i> pada <i>game</i> .
Prosedur Uji	Pemain menekan menu <i>high score</i> pada menu utama.
Kondisi yang diharapkan	Menampilkan <i>high score</i> dari masing level.
Hasil yang Didapatkan	Valid

Sumber: Pengujian dan Analisis

6.1.2.8 Kasus Uji Mulai Melihat *Credit*

Pada Tabel 6.18 ditunjukkan *test case* ketika pemain telah menekan menu *credit* untuk melihat *credit* pada *game*.

Tabel 6.18 *Test Case* Melihat *Credit*

Nama Kasus Uji	Melihat <i>credit</i>
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa <i>game</i> dapat memenuhi kebutuhan fungsional untuk melihat tampilan

	<i>credit</i> dan melihat siapa saja pihak yang terlibat dalam pembuatan <i>game</i> .
Prosedur Uji	Pemain menekan menu <i>credit</i> pada tampilan menu utama.
Kondisi yang diharapkan	Menampilkan tampilan <i>credit</i> dan terdapat tampilan siapa saja pihak yang terlibat dalam pembuatan <i>game</i> .
Hasil yang Didapatkan	Valid

Sumber: Pengujian dan Analisis

6.1.2.9 Kasus Uji Exit Game

Pada Tabel 6.19 ditunjukkan *test case* ketika pemain menekan menu *exit* pada *game*.

Tabel 6.19 Test Case Untuk Exit Game

Nama Kasus Uji	<i>Exit game</i>
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa <i>game</i> dapat memenuhi kebutuhan fungsional untuk keluar dari permainan.
Prosedur Uji	1. Pemain menekan menu <i>Exit</i> pada tampilan menu utama. 2. Pemain memilih pilihan tidak. 3. Pemain memilih pilihan ya.
Kondisi yang diharapkan	1. Muncul tampilan konfirmasi exit <i>game</i> . 2. Kembali ke menu utama. 3. Keluar <i>game</i> .
Hasil yang Didapatkan	Valid

Sumber: Pengujian dan Analisis

6.1.3 Play Testing

Play testing dilakukan untuk mendapatkan *feedback* tentang kondisi di dalam *game*, saran dan kesan pada *game* Karapan Sapi 3D dengan memberikan

kuisisioner kepada 15 pemain berbeda yang melakukan pengujian. Hasil *play testing* dapat dilihat pada Tabel 6.20.

Tabel 6.20 Hasil Kuisisioner *Play Testing*

No.	Pertanyaan	Hasil	
		Ya	Tidak
1.	Apakah <i>game</i> memiliki tampilan menarik?	15	0
2.	Apakah <i>game</i> mudah dimainkan?	15	0
3.	Apakah <i>game</i> secara keseluruhan menyenangkan untuk dimainkan?	15	0
Saran			
1.	Ditambahkan level dan karakter		
2.	Jalur/ <i>Track</i> diperpanjang		
3.	Jenis kotak misteri ditambah		
4.	Ditambahkan jebakan-jebakan sebagai variasi rintangan		
Kesan			
1.	Karakter dan animasi gerakan lucu		
2.	Memberikan <i>icon-icon</i> daerah terkenal di Indonesia sebagai <i>background</i> level merupakan ide yang bagus		

Sumber: Pengujian dan Analisis

6.1.4 Pengujian Performa

Pengujian performa dilakukan untuk mengetahui bagaimana performa *game* Karapan Sapi 3D saat melakukan proses eksekusi dalam melakukan penampilan objek 3D. Pengujian performa dilakukan melalui analisis *frame rate* dengan menghitung rata-rata *frame-per-second* dari hasil *rendering* objek 3D dalam *game*. *Frame rate* terlalu lambat membuat animasi terputus-putus (kurang halus), sedangkan *frame rate* cepat membuat *blur detail* animasi (sangat halus). Tabel standar FPS dapat dilihat pada Tabel 6.21.

Game dijalankan pada 10 mesin komputer baik CPU maupun *notebook/laptop* dan melakukan pencatatan nilai FPS pada masing-masing mesin untuk setiap resolusi berbeda. Hasil pengujian ditunjukkan pada Tabel 6.22.

Tabel 6.21 Standar *Frame-per-Second*

Nilai FPS	Efek
< 30 FPS	Kualitas dibawah rata-rata dan gambar kurang halus.
30-40 FPS	Kualitas rata – rata dan gambar sudah mulai halus.
40-60 FPS	Kualitas bagus dan gambar sudah halus.
> 60 FPS	Kualitas terbaik dan gambar sangat halus.

Sumber: [GUR-12]

Tabel 6.22 Pengujian Performa FPS Karapan Sapi 3D

No	Spesifikasi	Resolusi	Range FPS
1.	Intel Core 2 duo, 1GB RAM, Intel HD Graphics 251MB.	1024 x 768 (window)	0-1
		1280 x 720 (window)	0-0
		1024 x 768 (<i>fullscreen</i>)	1-1
		1280 x 720 (<i>fullscreen</i>)	1-1
2.	Intel Core 2 duo, 2GB RAM, Intel HD Graphics 876MB	1024 x 768 (window)	1-1
		1280 x 720 (window)	0-1
		1024 x 768 (<i>fullscreen</i>)	1-1
		1280 x 720 (<i>fullscreen</i>)	1-1
3.	Core i7, 8GB RAM, GT 540M 2GB	1024 x 768 (window)	125-127
		1280 x 720 (window)	120-129
		1024 x 768 (<i>fullscreen</i>)	127-128
		1280 x 720 (<i>fullscreen</i>)	123-128
4.	Core i5, 8GB RAM, GT 540M 2GB	1024 x 768 (window)	103-124
		1280 x 720 (window)	103-117
		1024 x 768 (<i>fullscreen</i>)	110-123
		1280 x 720 (<i>fullscreen</i>)	103-123
5.	Core i5, 6GB RAM, GT 540M 2GB	1024 x 768 (window)	103-121
		1280 x 720 (window)	99-121
		1024 x 768 (<i>fullscreen</i>)	103-121
		1280 x 720 (<i>fullscreen</i>)	100-110
6.	Core i5, 4GB	1024 x 768 (window)	100-113

	RAM, GT 540M 2GB	1280 x 720 (window)	94-104
		1024 x 768 (<i>fullscreen</i>)	103-109
		1280 x 720 (<i>fullscreen</i>)	94-108
7.	Core i7, 8GB	1024 x 768 (window)	1-3
	RAM, Intel HD	1280 x 720 (window)	1-1
	Graphic 1.6GB	1024 x 768 (<i>fullscreen</i>)	1-3
		1280 x 720 (<i>fullscreen</i>)	1-3
8.	IntelCore2 Duo,	1024 x 768 (window)	35-37
	ATI Mobility	1280 x 720 (window)	33-37
	Radeon HD 4500	1024 x 768 (full screen)	37-40
	2 Gb, 4 Gb RAM	1280 x 720 (full screen)	36-39
9.	Core i5, 6GB	1024 x 768 (window)	1-2
	RAM, Intel HD	1280 x 720 (window)	1-1
	Graphic 1.6GB	1024 x 768 (<i>fullscreen</i>)	1-3
		1280 x 720 (<i>fullscreen</i>)	1-2
10.	Core i5, 4GB	1024 x 768 (window)	1-3
	RAM, Intel HD	1280 x 720 (window)	1-1
	Graphic 1.6GB	1024 x 768 (<i>fullscreen</i>)	1-3
		1280 x 720 (<i>fullscreen</i>)	1-3

Sumber: Pengujian dan Analisis

6.2 Analisis

Analisis bertujuan untuk mendapatkan kesimpulan dari hasil pengujian *game* Karapan Sapi 3D yang telah dilakukan. Analisis dilakukan terhadap hasil pengujian di setiap tahap pengujian. Proses analisis meliputi analisis hasil pengujian unit, analisis hasil pengujian integrasi, analisis hasil pengujian validasi, analisis hasil *play testing* dan analisis hasil pengujian performa.

6.2.1 Analisis Pengujian *Whitebox*

Analisis pengujian *whitebox* meliputi analisis pengujian unit dan pengujian integrasi.

6.2.1.1 Analisis Pengujian Unit

Analisis terhadap hasil pengujian unit dilakukan dengan melihat kesesuaian fungsi dari implementasi unit yang diuji dengan hasil perancangan *game* yang telah dirancang sebelumnya. Pengujian implementasi unit menggunakan metode basis path untuk mendapatkan kompleksitas siklomatis, dari kompleksitas siklomatis dihasilkan jalur independen untuk skenario kasus uji. Berdasarkan hal tersebut, maka dapat diambil kesimpulan bahwa unit modul dari *game* sudah memenuhi kebutuhan yang telah dirancang pada tahap perancangan.

6.2.1.2 Analisis Pengujian Integrasi

Analisis terhadap hasil pengujian integrasi dilakukan dengan melihat kesesuaian fungsi dari implementasi integrasi yang diuji dengan hasil perancangan *game* yang telah dirancang sebelumnya. Pengujian implementasi integrasi menggunakan metode basis path untuk mendapatkan kompleksitas siklomatis, dari kompleksitas siklomatis dihasilkan jalur independen untuk skenario kasus uji. Berdasarkan hal tersebut, maka dapat diambil kesimpulan bahwa integrasi dari *game* Karapan Sapi 3D sudah memenuhi kebutuhan yang telah dirancang pada tahap perancangan.

6.2.2 Analisis Pengujian *Blackbox*

Analisis terhadap hasil pengujian validasi dilakukan dengan melihat antara hasil kinerja sistem dengan daftar kebutuhan yang dimodelkan pada 9 *test case*. Berdasarkan hasil pengujian validasi dapat disimpulkan bahwa implementasi dan fungsionalitas *game* Karapan Sapi 3D telah sesuai dengan kebutuhan dan valid pada keseluruhan *test case*.

6.2.3 Analisis *Play Testing*

Analisis terhadap hasil *play testing* dengan mengacu pada tabel 6.22, maka dapat diambil kesimpulan bahwa *game* Karapan Sapi 3D sudah dianggap menarik dan menyenangkan saat dimainkan oleh pemain/user.

6.2.4 Analisis Pengujian Performa

Proses analisis terhadap hasil pengujian performa dilakukan dengan analisis hasil pengujian *frame rate*. Proses analisis terhadap hasil pengujian *frame rate* dilakukan dengan melihat performa *game* dalam menampilkan objek 3D yang ditunjukkan dengan nilai *frame rate* dan kemampuan *game* bekerja dalam kondisi lingkungan normal. Berdasarkan hal tersebut maka dapat diambil kesimpulan sebagai berikut :

1. Semakin besar resolusi yang dijalankan akan mempengaruhi nilai *frame rate* yang diperoleh menjadi semakin kecil.
2. Spesifikasi *processor*, *graphic processor* dan RAM sangat mempengaruhi nilai *frame rate* yang didapatkan pada *game* Karapan Sapi 3D. Komputer atau *notebook/laptop* dengan spesifikasi tinggi akan mendapatkan nilai *frame rate* semakin tinggi.
3. Pada kondisi mode *fullscreen*, nilai *frame rate* akan semakin besar dibandingkan dengan pada saat kondisi *window*, karena pada saat mode *fullscreen*, kerja *graphics processor* fokus terhadap proses yang sedang ditampilkan secara *fullscreen*.
4. Komputer atau *laptop* yang menggunakan VGA *eksternal* mendapatkan nilai *frame rate* lebih baik pada saat menjalankan *game* Karapan Sapi 3D.
5. Berdasarkan nilai *frame rate* yang didapat pada Tabel 6.21, spesifikasi komputer atau *notebook/laptop* yang digunakan pada pengujian dan merujuk pada Tabel 6.20 tentang standar *frame rate*, maka dapat diketahui untuk menjalankan *game* Karapan Sapi 3D agar mendapatkan nilai *frame rate* dengan kisaran 40-60 FPS diperlukan komputer atau laptop dengan spesifikasi minimum:
 1. Intel core2duo *processor* (atau sebanding).
 2. 2GB RAM
 3. VGA *eksternal* dengan memory minimum 1GB/32bit.

BAB VI

PENUTUP

7.1 Kesimpulan

Dari hasil pengamatan selama perancangan, implementasi, dan proses pengujian *game* yang digunakan, diambil kesimpulan sebagai berikut:

1. Perancangan *game* Karapan Sapi 3D terdiri dari 2 tahap yaitu tahap *Game Design* untuk perancangan dan *Technical Design* sebagai pemodelan yang menggunakan bahasa pemodelan UML dan OOAD, *game* Karapan Sapi 3D diimplementasikan menggunakan *game engine* jMonkeyEngine yang berbasis bahasa pemrograman Java.
2. Berdasarkan hasil pengujian unit dengan menggunakan metode *whitebox*, *game* Karapan Sapi 3D telah sesuai dengan rancangan *gameplay* yang dibuat pada tahap perancangan.
3. Berdasarkan hasil pengujian validasi yang menggunakan metode *blackbox testing*, didapatkan bahwa keseluruhan fungsional aplikasi permainan dapat berjalan sesuai analisis kebutuhan yang telah dibuat dan dimodelkan dalam *use case*.
4. Berdasarkan pengujian performa dengan menggunakan metode penghitungan *frame rate*, untuk mendapatkan performa minimum (layak dijalankan) dibutuhkan komputer atau laptop yang memenuhi spesifikasi minimum.

7.2 Saran

Saran yang dapat diberikan untuk pengembangan *game* Karapan Sapi 3D lebih lanjut antara lain:

1. *Game* Karapan Sapi 3D dapat ditambahkan lebih banyak variasi karakter, level dengan background tempat di Indonesia dan *environment* yang berbeda.
2. *Game* Karapan Sapi 3D dapat ditambahkan jumlah level, variasi *buff* pada *mystery box*, tingkat kesulitan penyelesaian dan fitur yang lebih bervariasi.

3. Performa *game* Karapan Sapi 3D dapat ditingkatkan dengan optimasi pada proses algoritma *view frustum culling* yaitu teknik untuk menentukan objek mana yang berada diluar *view frustum*, *view frustum* adalah bentuk piramid di depan *viewer* yang menjangkau area pandang [ASS-01].



DAFTAR PUSTAKA

- [ADA-10] Dawes, Adam. 2010. *Windows Phone 7 Game Development*. Apress.
- [ASS-01] Assarsson, Ulf. 2001. *View Frustum Culling and Animated Ray Tracing: Improvements and Methodological Considerations*. Diakses dari <http://www.cse.chalmers.se/~uffe/lic.pdf>. Tanggal Akses: 22-01-2013.
- [BTS-11] Bangkalan Tourism Site. 2011. *Bangkalan Tourism*. Diakses dari <http://www.eastjava.com/tourism/bangkalan/bullracing.html>, <http://www.eastjava.com/tourism/bangkalan/index.html>. Tanggal Akses: 16-10-2011.
- [EDG-06] Grebler, Eric D. 2006. *3D Game Programming for Teens*. Thompson Course Technology PTR.
- [FUL-08] Fullerton, Tracy. 2008. *Game Design Workshop*. Morgan Kaufmann Publishers.
- [ESR-13] ESRB Org Official Site. 2013. *ESRB Rating Guides*. Diakses dari http://www.esrb.org/ratings/ratings_guide.jsp#rating_categories. Tanggal Akses: 22-01-2013.
- [GUR-12] Guru3D Site. 2012. *VGA Charts Summer 2012*. Diakses dari http://www.guru3D.com/articles_pages/vga_charts_summer_2012,1.html. Tanggal Akses: 11-12-2012.
- [GUY-07] Lecky-Thompson, Guy W. 2007. *Video Game Design Revealed*. Cengage Learning.
- [JME-11] jMonkeyEngine Official Site. *jMonkeyEngine*. Diakses dari <http://www.jmonkeyengine.org>, <http://www.jmonkeyengine.com>. Tanggal Akses: 18-10-2011.
- [KEN-07] Finney, Kenneth C. 2007. *3D Game Programming All in One: Second Edition*. Thompson Course Technology PTR.
- [KEV-04] Oxland, Kevin. 2004. *Gameplay and Design*. Pearson Education Ltd.

- [KWG-00] Kramer, Wolfgang. 2000. *What Is a Game?*. Diakses dari : <http://www.thegamesjournal.com/articles/WhatIsaGame.shtml>. Tanggal Akses: 14-10-2011.
- [PRE-01] Pressman, Roger S. 2010. *Software Engineering: a practitioner's approach, 7th edition*. Mc Graw Hill.
- [REX-02] Black, Rex. 2002. *Managing the Testing Process : Practical Tools and Techniques for Managing Hardware and Software Testing*. Wiley Publishing, Inc., Canada.
- [ROL-04] Rollings, Andrew, Dave Morris. 2004. *Game Architecture and Design: A New Edition*. New Riders Publishing.
- [SCR-10] Rogers, Scott. 2010. *LEVEL UP! The Guide To Great Video Game Design*. John Wiley & Sons, Ltd.
- [SZO-04] Zerbst, Stefan, Oliver Duvel. 2004. *3D Game Engine Programming*. Thompson Course Technology PTR.

