

**RANCANG BANGUN PERMAINAN GOBAK SODOR 3D PADA
PLATFORM PC**

Konsentrasi : Rekayasa Perangkat Lunak

SKRIPSI

Untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer



Disusun Oleh :

Aldian Lodewyk Manuputty

NIM. 0710683014

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
PROGRAM STUDI TEKNIK INFORMATIKA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2013**

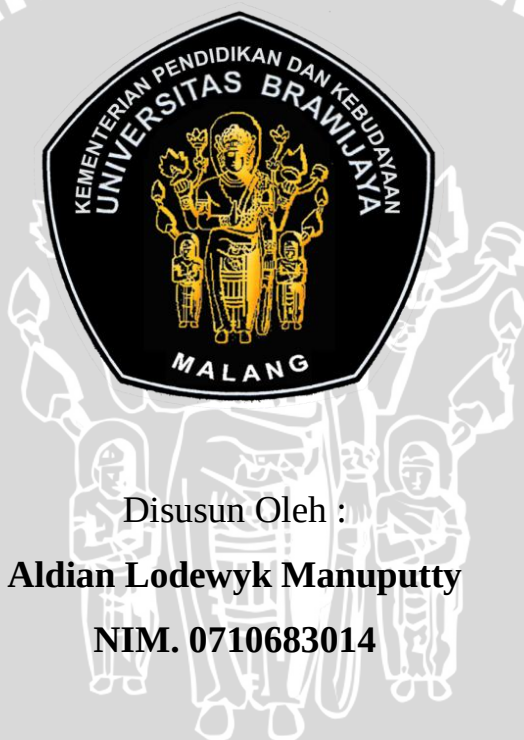
LEMBAR PERSETUJUAN

RANCANG BANGUN PERMAINAN GOBAK SODOR 3D PADA PLATFORM PC

Konsentrasi : Rekayasa Perangkat Lunak

SKRIPSI

Untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer



Disusun Oleh :

Aldian Lodewyk Manuputty

NIM. 0710683014

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

Suprpto, ST.,MT.

NIP. 19710727 196603 1 001

Eriq M. Adams J, ST., .Kom.

NIK. 850410 061 1 0027

KATA PENGANTAR

Dengan nama Tuhan Yang Maha Pengasih dan Penyayang. Segala puji syukur kepada Tuhan Yesus Kristus karena atas kemurahan dan kasih setianya penulis dapat menyelesaikan Skripsi yang berjudul “Rancang Bangun Permainan Gobak Sodor 3D”. 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. Papa Jonias Manuputty, dan mama Ernie Indrawatie atas kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta adik terkasih Ditha Arianti Manuputty yang senantiasa tiada henti – hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini.
2. Terkasih Elsa Priskila Yonathan dan keluarga, atas segala dukungan dan doa nya selama penulis menyelesaikan skripsi ini.
3. 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.
4. Bapak Drs. Marji, M.T dan Bapak Issa Arwani, S.Kom, M.Sc selaku Ketua dan Sekretaris Program Studi Teknik Informatika Universitas Brawijaya.
5. Bapak Suprpto, S.T., MT, dan Bapak Eriq M. Adams J, ST., M.Kom selaku dosen pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
6. Bapak Himawat Aryadita, S.T, M.Sc selaku dosen penasehat akademik yang selalu memberikan nasehat kepada penulis selama menempuh masa studi.

7. Seluruh Dosen Teknik Informatika Universitas Brawijaya atas kesediaan membagi ilmunya kepada penulis.
8. Kukuh Heru Irawan, Wisnu Aditya, Febrian Aris Pradika atas segala bantuan dan kerjasamanya selama penulis menyelesaikan skripsi ini.
9. Anza Ansori, Deta Pratama, Adam Hendra Brata dan Sativandi Putra yang telah banyak memberikan masukan dalam pembuatan skripsi ini dan seluruh Sahabat - sahabatku Angkatan 2007 Teknik Perangkat Lunak (sampai kapanpun kita tetap Legendary TPL07) terimakasih atas segala bantuannya dan perjuangan selama menempuh studi di Teknik Informatika Universitas Brawijaya.
10. Ari Agustina dan keluarga yang telah menyediakan tempat untuk proses pengerjaan skripsi ini.
11. Seluruh anggota NOOBS FC atas semua dukungan kepada penulis selama pembuatan skripsi.
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 Tuhan Yesus memberikan kemurahan serta kasih yang berlimpah. 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

Aldian Lodewyk Manuputty. 2013. : Rancang Bangun Permainan Gobak Sodor 3D Pada Platform PC. Skripsi Program Studi Teknik Informatika, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.
Dosen Pembimbing: Suprpto, ST.,MT dan Eriq M. Adams J, ST., .Kom.

ABSTRAK

Gobak sodor merupakan salah satu permainan tradisional yang mulai dilupakan akibat perkembangan teknologi. Saat ini anak-anak lebih tertarik untuk memainkan permainan yang berbasis multimedia atau yang lebih dikenal dengan *video game*. Oleh karena itu penulis mengembangkan *Game Gobak Sodor 3D* yang dibuat untuk kembali memperkenalkan anak-anak pada permainan tradisional, dengan harapan permainan gobak sodor dapat kembali dikenal dan dimainkan. Perancangan *game* terdiri dari pembuatan *Game Design Document* dan *Technical Design* kemudian diimplementasikan dan dikembangkan menggunakan *game engine* jMonkey yang berbasis Java, sedangkan untuk pemodelan karakter dan lingkungan permainan menggunakan Blender 3D. Dalam pengembangannya *game* ini memakai *library* yang sudah tersedia pada *game engine* jMonkey. Pengujian *Game Gobak Sodor 3D* meliputi pengujian perangkat lunak menggunakan metode *white-box testing*, *black-box testing* serta pengujian performa melalui analisis nilai *frame rate*. Berdasarkan hasil pengujian unit dapat disimpulkan bahwa unit modul dari program *game* sudah memenuhi kebutuhan fungsional yang telah dirancang pada tahap perancangan. Pada pengujian validasi dapat disimpulkan bahwa implementasi dan fungsionalitas *game* Gobak Sodor 3D telah memenuhi kebutuhan yang telah dijabarkan pada tahap analisa kebutuhan. Hasil pengujian performa menunjukkan hasil yang bervariasi berdasarkan spesifikasi komputer yang menjalankan *game* Gobak Sodor 3D.

Kata Kunci : Gobak Sodor, *video game*, *frame rate*, *game engine*

Aldian Lodewyk Manuputty. 2013. : 3D Gobak Sodor Game Development.

Advisor : Suprpto, ST.,MT and Eriq M. Adams J, ST., .Kom.

ABSTRACT

Gobak sodor is one of traditional games that is started to be forgotten because of the technology expansion. Nowadays, children are more interested to play multimedia-based games or known as video games. Therefore, the author developed a 3D Gobak Sodor game to re-introduce the traditional game to the teenagers with hope Gobak Sodor is started to be known and played again. The design of the game consists of makin g Game Design and Technical Design that is implemented by using jMonkey Java based, whereas the model of characters and game environment used 3D Blender. On its development, the researcher used the library on this game that is already available on the jMonkey engine. The Gobak Sodor 3D game has been tested by software testing using white-box testing and performance testing through the frame rate analysis. Based on the test result, it can be conclude that the unit module of the game program has met the functional requirements that have been designed in drafting stage. In validation testing, it can be concluded that the implementation and functionality of Sodor Gobak 3D game is already required the needs in analysis phase. The results show the vary performance based on the specifications of the computer running the Gobak Sodor 3D game.

Keywords : Gobak Sodor, video game, frame rate, game engine

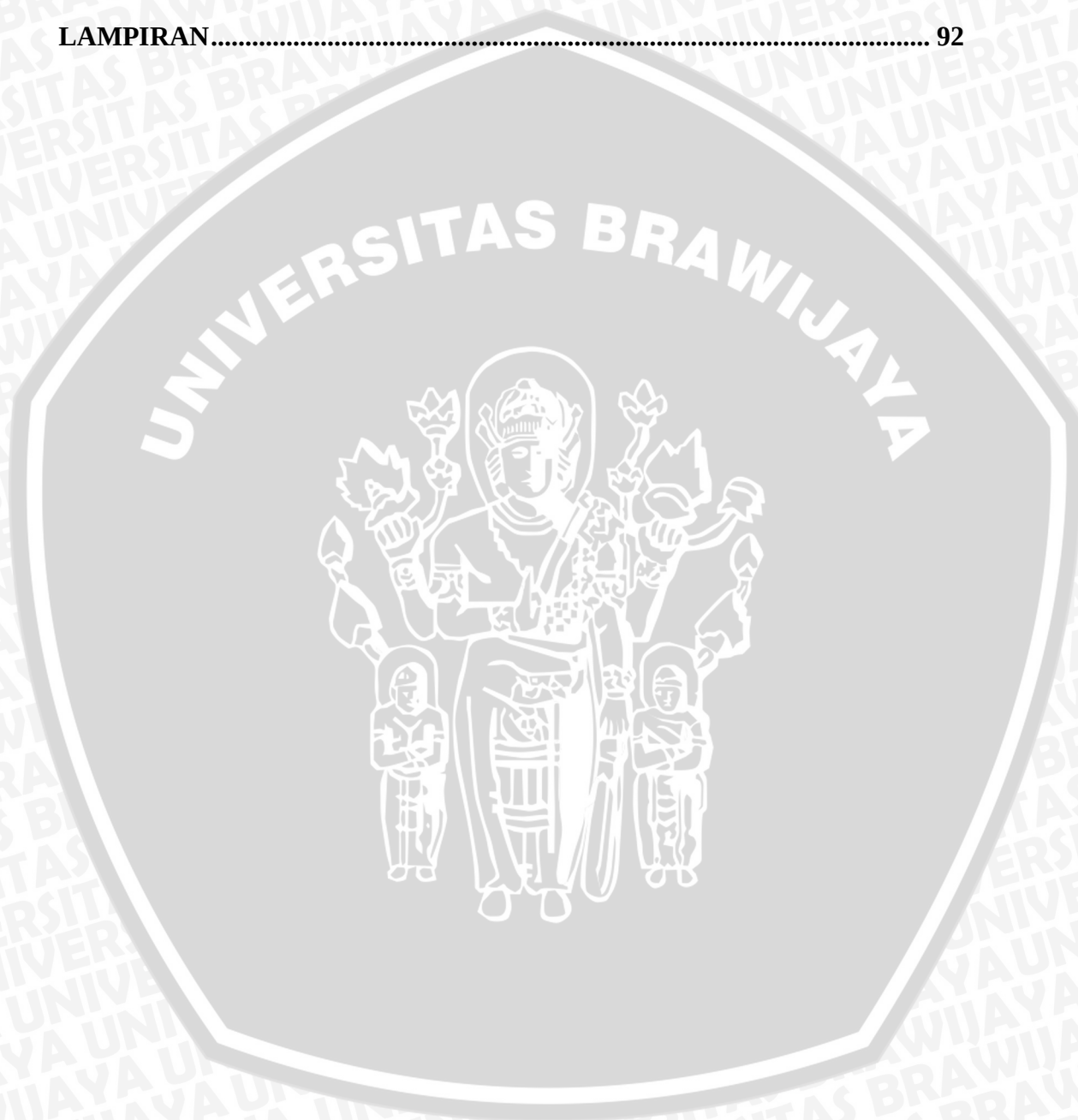
DAFTAR ISI

KATA PENGANTAR.....	ii
ABSTRAK	iv
ABSTRACT	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	ix
DAFTAR TABEL	xi
 BAB 1 PENDAHULUAN	 1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Manfaat.....	3
1.6 Sistematika Penulisan.....	3
 BAB II DASAR TEORI.....	 5
2.1 Permainan	5
2.1.1 <i>Hybrid Genre (Persuasive – Sport Game)</i>	5
2.1.2 Klasifikasi Rating <i>Game</i> Berdasarkan Umur	6
2.1.3 <i>Game Engine</i>	7
2.1.4 <i>jMonkey Engine (jMonkey Platform Game Engine)</i>	8
2.1.5 <i>jMonkey Platform Structure</i>	8
2.2 Gobak Sodor (Galasin)	10
2.3 Teori Pengujian	10
2.3.1 Pengujian Terstruktur (<i>White box testing</i>)	11
2.3.2 Pengujian Fungsionalitas (<i>Blackbox Testing</i>)	12
2.3.3 Pengujian Performa.....	12
2.3.4 Playtesting.....	13
 BAB III METODOLOGI PENELITIAN	 14

3.1	Studi Literatur.....	15
3.2	Perancangan <i>Game</i>	15
3.2.1	<i>Game design</i>	15
3.2.2	<i>Technical Design</i>	22
3.3	Implementasi	23
3.4	Pengujian	24
3.4.1	Pengujian Terstruktur (Whitebox Testing)	24
3.4.2	Pengujian Fungsional (Blackbox Testing)	24
3.4.3	Pengujian Performa (Frame Per Second)	24
3.4.4	Playtesting.....	25
BAB IV	PERANCANGAN.....	26
4.1	<i>Game Design</i>	26
4.1.1	<i>Game Goal</i>	27
4.1.2	<i>Game Outline</i>	27
4.1.3	Karakter.....	27
4.1.4	Gerakan Karakter.....	31
4.1.5	Klasifikasi Karakter	32
4.1.6	Environment.....	33
4.1.7	Kontrol Pada <i>Game</i>	34
4.1.8	Antarmuka <i>Game</i>	38
4.1.9	Level <i>Game</i>	41
4.1.10	Sistem Pada <i>Game</i>	43
4.2	<i>Technical Design</i>	44
4.2.1	<i>Use Case</i>	44
4.2.2	<i>Class Diagram</i>	47
4.2.3	<i>Activity Diagram</i>	49
BAB V	IMPLEMENTASI.....	55
5.1	Spesifikasi Sistem.....	55
5.1.1	Spesifikasi Perangkat Keras	55
5.1.2	Spesifikasi Perangkat Lunak.....	55
5.2	Batasan-Batasan Implementasi.....	56

5.3	Implementasi <i>Artwork</i>	56
5.3.1	Implementasi <i>Game Cover</i>	56
5.3.2	Implementasi Karakter	58
5.3.3	Implementasi <i>Environment</i>	58
5.4	Implementasi Level	59
5.4.1	Level 1	59
5.4.2	Level 2	60
5.4.3	Level 3	60
5.5	Implementasi Algoritma	60
5.5.1	Implementasi Algoritma <i>onAction()</i>	61
5.5.2	Implementasi Algoritma <i>decreaseLife()</i>	61
5.5.3	Implementasi Algoritma <i>finish()</i>	62
	Sumber: Implementasi	62
5.5.4	Implementasi Algoritma <i>collision()</i>	63
5.6	Implementasi Antarmuka	64
5.6.1	Antarmuka Menu	64
5.6.2	Antarmuka Pilih Karakter	64
5.6.3	Antarmuka Permainan	65
BAB VI	PENGUJIAN DAN ANALISIS	66
6.1	Pengujian	66
6.1.1	Pengujian Unit	66
6.1.2	Pengujian Integrasi	72
6.1.3	Pengujian Validasi	75
6.1.4	Pengujian Performa	82
6.1.5	Playtesting	83
6.2	Analisis	84
6.2.1	Analisis Pengujian Unit	84
6.2.2	Analisis Pengujian Integrasi	84
6.2.3	Analisis Pengujian Validasi	84
6.2.4	Analisis Pengujian Performa	85
6.2.5	Analisis Hasil <i>Playtesting</i>	85

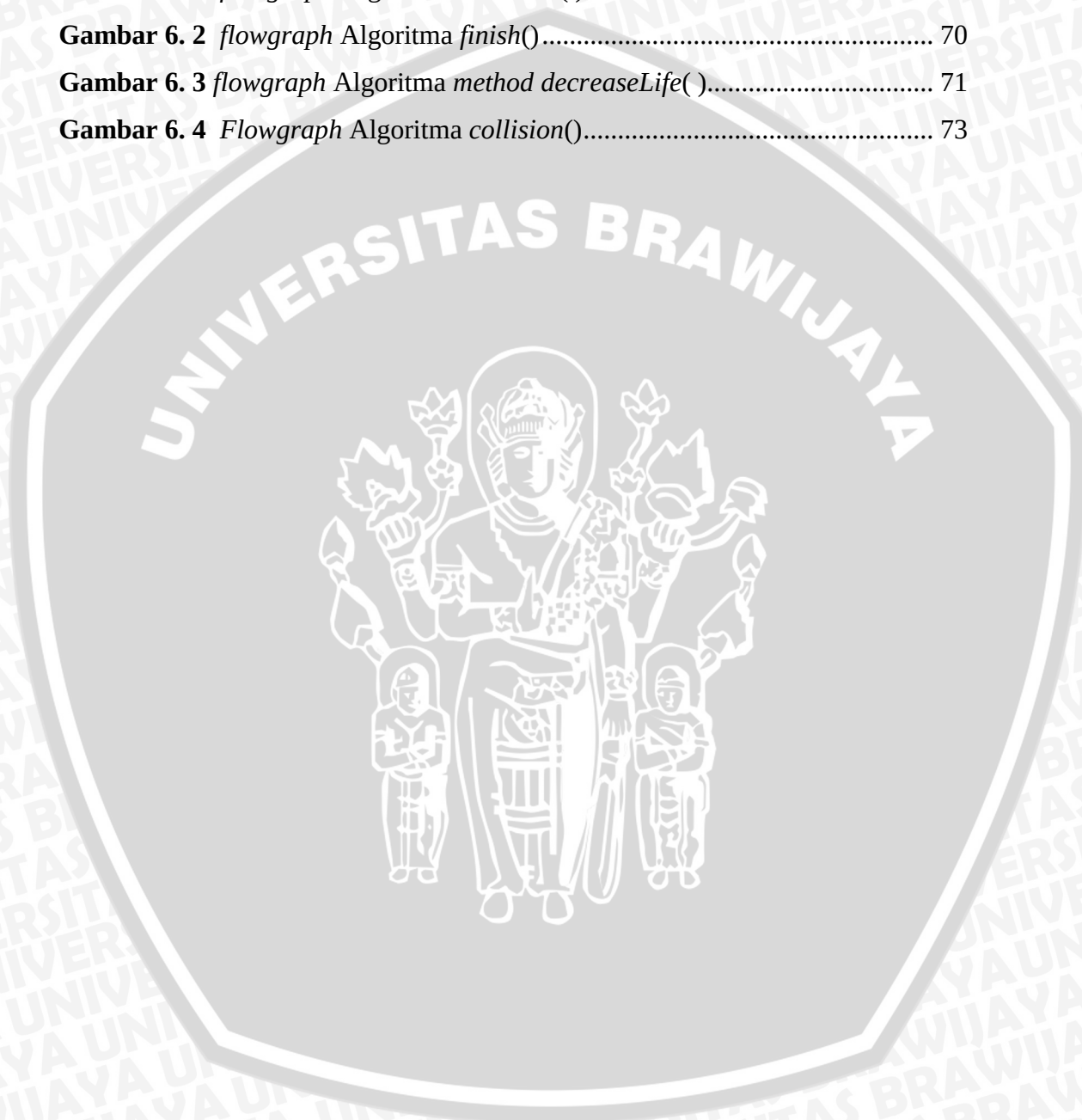
BAB VII PENUTUP.....	87
7.1 Kesimpulan.....	87
7.2 Saran	87
DAFTAR PUSTAKA.....	89
LAMPIRAN.....	92



DAFTAR GAMBAR

Gambar 3. 1 Diagram Alir Runtutan Pengerjaan Tugas Akhir.	14
Gambar 3. 2 <i>Gameflow</i> Gobak Sodor 3D.....	18
Gambar 3. 3 Rancangan karakter Gobak Sodor 3D	19
Gambar 3. 4 Kontrol pada <i>game</i> Gobak Sodor 3D	20
Gambar 3. 5 Rancangan <i>Interface game</i> Gobak Sodor 3D	21
Gambar 3. 6 Rancangan <i>environment game</i> Gobak Sodor 3D.....	22
Gambar 3. 7 <i>flowchart</i> implementasi <i>game</i> gobak sodor 3D	23
Gambar 4. 1 Perancangan <i>Cover game</i> Gobak Sodor 3D	26
Gambar 4. 2 Rancangan antarmuka menu.....	39
Gambar 4. 3 Rancangan antarmuka pilih karakter	40
Gambar 4. 4 Rancangan antarmuka permainan.....	41
Gambar 4. 5 <i>Use Case diagram</i> Permainan Gobak Sodor 3D	47
Gambar 4. 6 <i>Class Diagram</i> Permainan Gobak Sodor 3D.....	48
Gambar 4. 7 <i>Activity Diagram Start</i>	50
Gambar 4. 8 <i>Activity Diagram</i> Pilih Karakter	50
Gambar 4. 9 <i>Activity Diagram</i> Mula Permainan	51
Gambar 4. 10 <i>Activity Diagram</i> Bergerak ke Kiri	51
Gambar 4. 11 <i>Activity Diagram</i> Bergerak ke Kanan	52
Gambar 4. 12 <i>Activity Diagram</i> Bergerak Maju	52
Gambar 4. 13 <i>Activity Diagram</i> Bergerak Mundur.....	53
Gambar 4. 14 <i>Activity Diagram Best Score</i>	53
Gambar 4. 15 <i>Activity Diagram Credit</i>	54
Gambar 4. 16 <i>Activity Diagram Exit</i>	54
Gambar 5. 1 Implementasi Halaman <i>cover game</i> Gobak Sodor 3D	57
Gambar 5. 2 Implementasi karakter <i>game</i> Gobak Sodor 3D.....	58
Gambar 5. 3 Implementasi <i>Environtment game</i> Gobak Sodor 3D	59
Gambar 5. 4 Implementasi level 1.....	59
Gambar 5. 5 Implementasi level 2.....	60
Gambar 5. 6 Implementasi level 3.....	60

Gambar 5. 7 Implementasi antarmuka menu.....	64
Gambar 5. 8 Implementasi antarmuka pilih karakter	65
Gambar 5. 9 Implementasi antarmuka permainan.....	65
Gambar 6. 1 <i>flowgraph</i> Algoritma <i>onAction()</i>	67
Gambar 6. 2 <i>flowgraph</i> Algoritma <i>finish()</i>	70
Gambar 6. 3 <i>flowgraph</i> Algoritma <i>method decreaseLife()</i>	71
Gambar 6. 4 <i>Flowgraph</i> Algoritma <i>collision()</i>	73



DAFTAR TABEL

Tabel 2. 1 Klasifikasi rating <i>game</i> berdasarkan umur	6
Tabel 3. 1 <i>One sheet document</i>	15
Tabel 3. 3 Tabel standar FPS.....	25
Tabel 4. 1 Profil Karakter Baldi	28
Tabel 4. 2 Profil karakter Doni.....	28
Tabel 4. 3 Profil karakter Jacob.....	29
Tabel 4. 4 Profil karakter Koko.....	29
Tabel 4. 5 Profil karakter Rini.....	30
Tabel 4. 6 Profil karakter Rocky	30
Tabel 4. 7 Profil karakter Spiky	31
Tabel 4. 8 Gerakan Lari.....	31
Tabel 4. 9 Gerakan <i>stand (idle)</i>	32
Tabel 4. 10 Gerakan Jaga	32
Tabel 4. 11 Keterangan objek.....	33
Tabel 4. 12 Kontrol gerakan maju.....	34
Tabel 4. 13 Kontrol gerakan belok kanan dan kiri	35
Tabel 4. 14 Kontrol gerakan maju menyamping kiri dan kanan	36
Tabel 4. 15 Kontrol gerakan mundur	37
Tabel 4. 16 Kontrol gerakan mundur menyamping kiri dan kanan.....	38
Tabel 4. 17 Latar belakang level	42
Tabel 4. 18 Identifikasi aktor.....	44
Tabel 4. 19 Daftar kebutuhan	45
Tabel 4. 20 Deskripsi <i>class diagram</i> Gobak Sodor 3D	48
Tabel 5. 1 Algoritma <i>onAction()</i>	61
Tabel 5. 2 Algoritma <i>decreaseLife()</i>	62
Tabel 5. 3 Algoritma <i>finish()</i>	62
Tabel 5. 4 Algoritma <i>collision()</i>	63
Tabel 6. 1 Pengujian Algoritma <i>onAction()</i>	66
Tabel 6. 2 Kasus uji algoritma <i>onAction()</i>	68

Tabel 6. 3 Pengujian unit algoritma <i>finish()</i>	69
Tabel 6. 4 Kasus uji algoritma <i>finish()</i>	70
Tabel 6. 5 Pengujian unit algoritma <i>decreaseLife()</i>	71
Tabel 6. 6 Kasus uji algoritma <i>decreaseLife()</i>	72
Tabel 6. 7 Kasus uji algoritma <i>collision()</i>	72
Tabel 6. 8 Kasus uji algoritma <i>Collision()</i>	74
Tabel 6. 9 Kasus uji untuk pengujian validasi start <i>game</i>	76
Tabel 6. 10 Kasus uji untuk pengujian validasi start <i>game</i>	76
Tabel 6. 11 Kasus uji untuk pengujian memulai permainan	77
Tabel 6. 12 Kasus uji untuk pengujian kontrol gerak ke kiri	77
Tabel 6. 13 Kasus uji untuk pengujian kontrol gerak ke kanan	78
Tabel 6. 14 Kasus uji untuk pengujian kontrol gerak maju.....	78
Tabel 6. 15 Kasus uji untuk pengujian kontrol gerak maju.....	79
Tabel 6. 16 Kasus uji untuk pengujian pause <i>game</i>	80
Tabel 6. 17 Kasus uji untuk pengujian melihat <i>Best Score</i>	80
Tabel 6. 18 Kasus uji untuk pengujian melihat <i>Credit</i>	81
Tabel 6. 19 Kasus uji untuk pengujian Exit <i>game</i>	81
Tabel 6. 20 Pengujian performa analisa <i>frame rate</i>	82
Tabel 6. 21 Hasil kuisioner <i>playtesting</i>	83
Tabel 6. 22 Umpan balik	84

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Permainan, atau dalam bahasa Inggris disebut dengan *game* adalah aktivitas yang dimainkan untuk olahraga atau hiburan menurut aturan [LXP-96]. Sebuah *game* memiliki kriteria tersendiri yang membuatnya menjadi menyenangkan, yaitu selalu memiliki aturan dan tujuan, ada ketidakpastian disetiap perulangannya, sehingga hasil yang terjadi pada tiap *game* akan berbeda-beda. Dalam sebuah *game* selalu ada kompetisi, baik itu kompetisi antar pemain, atau kompetisi seorang pemain terhadap *game* itu sendiri [KWG-11].

Video *game* sendiri tidak hanya digemari oleh anak-anak, namun juga orang dewasa. setelah ditelaah ada beberapa manfaat positif dari bermain video *game*, antara lain, dapat memberikan rasa rileks, melatih kemampuan memusatkan perhatian dan konsentrasi, melatih memecahkan masalah, melatih mengembangkan kecepatan reaksi, dll [SUR-05].

Seiring dengan perkembangan zaman, sarana untuk bermain pun mengalami perkembangan. Dahulu, mungkin anak-anak bisa bermain petak umpet, kucing-kucingan, atau bermain layang-layang dan berbagai macam permainan tradisional lainnya di lapangan terbuka yang luas. Akan tetapi, saat ini dengan lahan yang tersedia semakin sempit dan adanya perkembangan zaman, jenis permainan pun berubah [SUP-08].

Dampak dari perkembangan zaman adalah banyak generasi baru yang tidak pernah memainkan bahkan mengetahui permainan-permainan tradisional yang merupakan sarana untuk membangun kerjasama. Karena didukung oleh pertumbuhan teknologi, banyak permainan-permainan virtual yang semakin membuat permainan-permainan tradisional tersisihkan.

Salah satu permainan tradisional yang dilupakan adalah gobak sodor (galah asin). Permainan gobak sodor sendiri merupakan salah satu permainan favorit pada zaman dahulu, dikarenakan permainan ini merupakan permainan tim, yang membutuhkan kekompakan dan kerjasama sebagai anggota tim.

Berdasarkan uraian dan permasalahan diatas maka penulis mengambil judul tugas akhir **“Rancang Bangun Permainan Gobak Sodor 3D”**, yaitu sebuah *game* 3D yang merupakan simulasi dari permainan Gobak sodor, dan *game* ini hanya dimainkan oleh seorang pemain. Dengan harapan dapat melestarikan dan memperkenalkan kembali permainan tradisional lewat media interaksi digital. Sehingga generasi baru bangsa ini dapat tetap mengetahui juga belajar dari permainan tradisional sehingga tetap dapat menjaga kelestariannya, meskipun ditengah perubahan zaman dan teknologi.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas, maka dapat dirumuskan permasalahannya yaitu:

1. Bagaimana merancang permainan tradisional Gobak Sodor beserta dengan aturan-aturan permainan ke dalam perancangan *gameplay* Gobak Sodor 3D?
2. Bagaimana mensimulasikan dan mengadopsi peraturan-peraturan permainan tradisional Gobak Sodor ke dalam *game* Gobak Sodor 3D?
3. Bagaimana menguji permainan komputer Gobak Sodor 3D?

1.3 Batasan Masalah

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

1. *Genre* permainan adalah *Hybrid Genre (Persuasive-Sport Game)*.
2. Gobak Sodor 3D merupakan *game single player*.
3. Sudut pandang kamera permainan ini adalah *orang ketiga*.
4. Preferensi karakter tidak mempengaruhi *gameplay*.

1.4 Tujuan

Tujuan penulisan tugas akhir ini adalah untuk dapat merancang, mengimplementasi, dan menguji *game* gobak sodor 3D.

1.5 Manfaat

Manfaat penelitian ini antara lain:

- a. Bagi penulis
 2. Mengaplikasikan ilmu yang didapat selama mengikuti perkuliahan di Teknik Informatika Universitas Brawijaya.
 3. Mendapatkan pemahaman tentang perancangan dan pengembangan permainan pada *platform* PC dengan menggunakan *game engine* jMonkey Engine.
- b. Bagi pengguna
 1. Menyediakan sarana hiburan berupa *game* tradisional yang bersifat virtual 3D dan dapat dimainkan dengan media interaksi digital.
 2. Menyediakan, melestarikan, sekaligus memperkenalkan permainan tradisional dengan media visual 3D yang dapat dimainkan pada *platform* PC.

1.6 Sistematika Penulisan

Sistematika penulisan laporan tugas akhir 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 *software* maupun *hardware* yang diperlukan untuk perancangan *game* Gobak Sodor 3D.

Bab III Metode Penelitian

Berisi tentang langkah-langkah dalam merancang *game* Gobak Sodor 3D yang dijalankan pada *platform* PC. Pada bab ini dijelaskan langkah-langkah yang akan dilakukan yaitu perumusan masalah, studi literatur, perancangan aplikasi permainan *game* Gobak Sodor 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 *game*, perancangan *game* Gobak Sodor 3D pada *platform* PC.

Bab V Implementasi

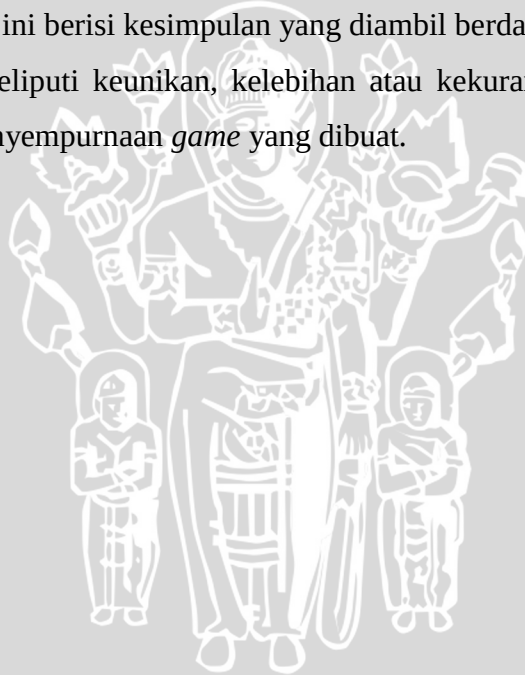
Bab ini berisi tentang implementasi *game* yang dibangun, meliputi pembuatan aplikasi *game* pada *platform* PC menggunakan *game engine*.

Bab VI Pengujian dan Analisis

Berisi mengenai analisa hasil perancangan *game*, hasil analisa output *game*, pembahasan terjadinya 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 *game* yang dibuat.



BAB II

DASAR TEORI

2.1 Permainan

Permainan, atau dalam bahasa Inggris disebut dengan *game* adalah aktivitas yang dimainkan untuk olahraga atau hiburan menurut aturan [LXP-96]. Sebuah *game* memiliki kriteria tersendiri yang membuatnya menjadi menyenangkan, yaitu selalu memiliki aturan dan tujuan, ada ketidakpastian disetiap perulangannya, sehingga hasil yang terjadi pada tiap *game* akan berbeda-beda. Dalam sebuah *game* selalu ada kompetisi, baik itu kompetisi antar pemain, atau kompetisi seorang pemain terhadap *game* itu sendiri [KWG-11].

Game sendiri terbagi atas beberapa klasifikasi (*genre*) yang didasari interaksi pemainnya. Visualisasi juga menjadi ukuran klasifikasi *genre*, namun untuk beberapa kasus pengembang *game* membuat kompilasi antar berbagai *genre*. Tentu saja menghasilkan variasi format *game* lebih banyak [SIB-09].

Pada Tahun 1981 *game* 3D untuk komputer rumahan pertama diperkenalkan, yaitu *game* 3D *Monster Maze* (Jk. Greye Software). *Game* 3D ini menggunakan sudut pandang orang pertama (*first person shooter*). *Game* ini dijalankan pada komputer rumahan dengan tipe Sinclair ZX81. *Game* ini menampilkan seseorang yang berusaha menghindari kejaran dinosaurus T-rex, melalui labirin-labirin hitam dan putih [MAR-03].

2.1.1 *Hybrid Genre (Persuasive – Sport Game)*

Genre game adalah klasifikasi *game* yang didasari interaksi pemainnya. Visualisasi juga menjadi ukuran klasifikasi *genre* ini. Namun untuk beberapa kasus pengembang *game* membuat kompilasi antar berbagai *genre* ini. Tentu saja menghasilkan variasi format *game* lebih banyak [SIB-09].

Beberapa *game* lintas *genre* (*Hybrid Genre*), menggabungkan fitur yang tidak biasanya ditemukan bersama-sama, kemudian menggabungkan tantangan dari setiap *genre*. Lintas *genre* di gunakan dengan tujuan untuk mencakup

audience yang lebih luas, dengan memasukan unsur-unsur *audience* dari kedua *genre*.

Persuasive game genre adalah , bentuk lain dari budaya. Yang berada pada persimpangan antara fenomena seperti budaya suatu daerah, teknologi *mobile*, jaringan komunikasi, fiksi realitas, dan seni pertunjukan, dengan menggabungkan potongan-potongan dari berbagai konteks untuk menghasilkan pengalaman bermain yang baru [MON-09].

Sport game genre biasanya terbagi dalam dua sudut pandang. Yaitu sudut pandang pemain dan sudut pandang manajer. Biasanya sudut pandang pemain dirancang untuk anak muda, dimana ketangkasan jari lebih penting untuk *gameplay*. Sedangkan sudut pandang manajer lebih peduli dengan perencanaan, realisasi, statistik, dan strategi [ROG-09]

2.1.2 Klasifikasi Rating Game Berdasarkan Umur

Berdasarkan studi yang dilakukan APA.org (*American Psychological Association*), di ketahui bahwa karena karakter di *game* sangat interaktif, maka *game* kekerasan dapat memperburuk pikiran dan tindakan agresif pada beberapa orang, bahkan lebih dari pada saat menonton televisi. Klasifikasi game berdasarkan umur dapat dilihat pada tabel 2.1.

Tabel 2. 1 Klasifikasi rating *game* berdasarkan umur

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+ (Everyone 10 and older): 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 kontent yang bersisi gambar atau video telanjang.
	RP (RATING PENDING): Judul terdaftar sebagai RP (Rating Pending) telah disampaikan kepada ESRB dan sedang menunggu penilaian akhir. (Simbol ini hanya akan muncul dalam iklan sebelum dirilis <i>game</i> .)

Sumber: [ESR-12]

2.1.3 *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].

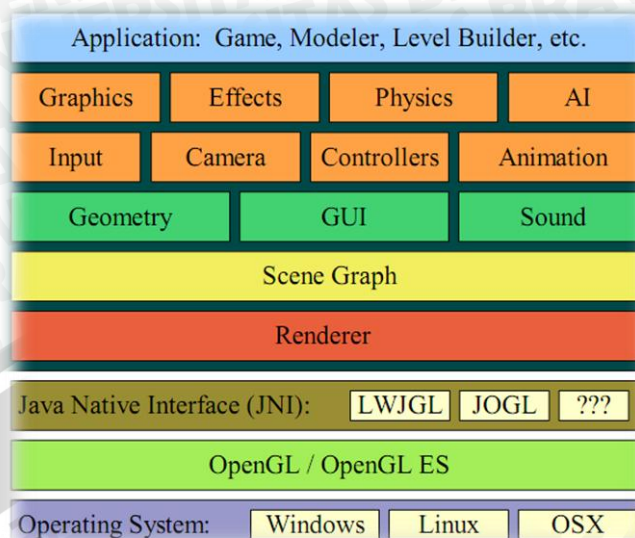
2.1.4 jMonkey Engine (jMonkey Platform Game Engine)

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-12]. 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 2.2.
5. *Modularity* lengkap untuk menjaga kesesuaian pengembangan *game* dari awal sampai akhir dan mendapatkan hal penting dalam pengembangan *game* langsung dari modul secara bersamaan.

2.1.5 jMonkey Platform Structure

jMonkeyPlatform terdiri dari tiga bagian utama dan ditunjukkan pada Gambar 2.2, dibawah ini:



Gambar 2. 1 jMonkeyPlatform *game engine* structure

Sumber: [JME-11]

Gambar akan dijelaskan di bawah ini:

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*.

2.2 Gobak Sodor (Galasin)

Permainan ini dimainkan oleh dua kelompok minimal per kelompoknya terdiri dari tiga anak dan lebih tepatnya empat orang per kelompok. Kedua kelompok ini saling menjadi lawan main. Kelompok permainan gobak sodor (galasin) ini, dibedakan menjadi kelompok bermain dan kelompok penjaga. Untuk memainkan permainan gobak sodor (galasin) membutuhkan lapangan yang agak luas, seperti lapangan bulutangkis. Lapangan permainan Gobak sodor (galasin) ini, terdiri dari 6 petak atau kamar ukuran $\pm 3 \times 4$ meter untuk empat pemain setiap kelompok. Keenam petak ini saling berhadapan, seperti persegi panjang dibagi enam. Keenam petak itu dipisahkan oleh jalur utama (vertical) dan tiga jalur horizontal (menyilang) dengan Lebar jalur ± 30 centimeter. Kelompok penjaga memiliki tugas untuk menjaga areanya melalui jalur-jalur yang tersedia agar tidak dilewati kelompok pemain [SUR-06]. Gambar permainan gobak sodor dapat dilihat pada Gambar 2.2.



Gambar 2. 2 Permainan Gobak Sodor (Galasin)

Sumber: [PUB-10]

2.3 Teori Pengujian

Pada bagian ini akan dibahas tentang teori pengujian yang digunakan pada pembuatan *game* Gobak Sodor 3D.

2.3.1 Pengujian Terstruktur (*White box testing*)

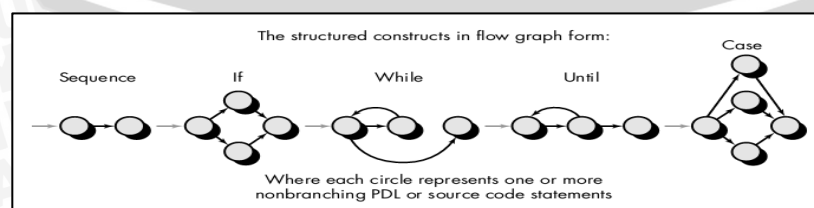
Pengujian terstruktur yang juga dikenal sebagai *whitebox testing*, diimplementasikan untuk menemukan *bug* pada operasi tingkat rendah, seperti yang terjadi pada baris kode, basis data, dan *interface*. Pengujian ini didasarkan pada bagaimana sebuah sistem beroperasi [REX-02].

Dengan *white-box* 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 tugas akhir 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 statemen 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 flowgraph dapat dilihat pada Gambar 2.3.



Gambar 2. 3 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 *statemen* proses baru atau kondisi baru. *Complexity* dapat 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.

2.3.2 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.

2.3.3 Pengujian Performa

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].

2.3.4 Playtesting

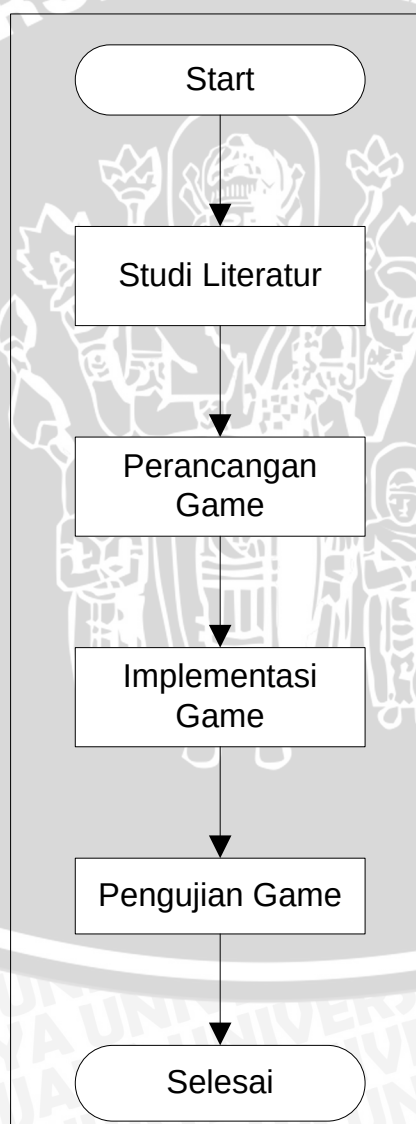
Playtesting adalah sesuatu yang dilakukan oleh *game designer* melalui proses desain untuk mengetahui apakah *game* dapat memberikan pengalaman bagi pemain sesuai dengan *goal* yang telah ditentukan [FUL-08]. Terdapat banyak cara untuk melakukan *playtesting*, sebagian ada yang menggunakan cara informal dan kualitatif, dan sebagian yang menggunakan cara lebih struktural dan kuantitatif. Sebuah *game garage* bisa memiliki 10 – 20 *playtester* untuk melakukan *playtesting* dan semua merupakan pengujian yang penting dan berharga. Tetapi, satu hal yang sama dari semua bentuk *playtesting* adalah memiliki satu tujuan yaitu mendapatkan umpan balik 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 *playtesting*, 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 *playtesting* dapat dilakukan oleh orang terdekat seperti keluarga dan teman-teman di luar team desain [FUL-250].

BAB III

METODOLOGI PENELITIAN

Pada bab ini dijelaskan langkah-langkah yang akan dilakukan dalam pengerjaan tugas akhir, yaitu studi literatur, perancangan *game*, implementasi dan pengujian dari aplikasi perangkat lunak yang akan dibuat. Kesimpulan dan saran disertakan sebagai catatan atas aplikasi dan kemungkinan arah pengembangan aplikasi selanjutnya. Berikut diagram alir runtutan pengerjaan tugas akhir seperti ditunjukkan pada Gambar 3.1.



Gambar 3. 1 Diagram Alir Runtutan Pengerjaan Tugas Akhir.

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. *GameEngine*
 - *jMonkeyEngine* (*jMonkeyPlatform GameEngine*)
 - *jMonkeyPlatform Structure*
- c. Gobak Sodor (Galasin)

3.2 Perancangan *Game*

Tahap perancangan *game* terdiri dari dua bagian yaitu *game design* dan *technical design*.

3.2.1 *Game design*

Game design adalah tahap perancangan *game* yang menjelaskan mengenai keseluruhan *game*, mulai dari *gameplay*, perancangan karakter, perancangan level, perancangan *interface* dan kontrol pada *game*. *Game design* pada bab ini akan menjelaskan garis besar dari *game* Gobak Sodor 3D yang di susun dalam *one sheet document* dan *ten pager document*.

3.2.1.1 *One Sheet Document*

One sheet document adalah gambaran sederhana dari permainan Gobak Sodor 3D. Untuk lebih jelasnya dapat dilihat pada Tabel 3.1.

Tabel 3. 1 *One sheet document*

NO	Elemen	Keterangan
1.	Ringkasan <i>Game</i>	<i>Game</i> Gobak Sodor 3D, merupakan <i>game</i> yang berjalan pada <i>platform</i> PC, dengan sudut pandang orang ketiga, dengan target usia pemain 5-13 tahun.
2.	Sekilas Cerita	Baldi dan teman-teman adalah anak berusia sekitar

		8-10 tahun yang berasal dari berbagai daerah di Indonesia bahkan ada yang berasal dari Australia. Mereka adalah siswa salah satu Sekolah dasar di kota Malang. Mereka merasa heran karena permainan tradisional sudah tidak diminati lagi, mereka sepakat untuk, memperkenalkan salah satu permainan tradisional yaitu gobak sodor dengan harapan mereka dapat melestarikan budaya nasional, melalui permainan tradisional.
3.	Cara bermain	Pemain memilih salah satu karakter dari tujuh karakter yang disediakan, antara lain Baldi, Doni, Jacob, Koko, Rini, Rocky, dan Spiky. Setelah memilih karakter, pemain harus mengendalikan karakter untuk menyebrangi petak-petak dengan penjaga (NPC) yang berjaga di garis. Pemain harus berhasil menyeberang, dan kembali lagi ke lokasi <i>start</i> tanpa tersentuh oleh penjaga (NPC) yang bertugas menjaga daerah masing masing.
4.	USP (Unique Selling Point)	• Grafis yang bagus. • Gerakan yang lucu dari setiap karakter nya. • Menantang dengan peningkatan kecepatan penjaga di setiap level nya. • Mudah di kendalikan.
5.	ESRB Rating	

Sumber: Perancangan

3.2.1.2 *Ten Pager Document*

Ten pager document adalah dokumen yang berisi penjabaran garis besar dari perancangan *game* Gobak Sodor 3D. Tujuan nya adalah agar pembaca dapat segera memahami konsep utama *game* Gobak Sodor 3D tanpa harus melihat secara detail. Berikut merupakan isi dari *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. *Game Story*

Baldi dan teman-teman adalah anak berusia sekitar 8-10 tahun yang berasal dari berbagai daerah di Indonesia bahkan ada yang berasal dari Australia. Mereka adalah siswa salah satu Sekolah dasar di kota Malang. Pada suatu hari salah seorang guru mereka mengajarkan mereka tentang permainan-permainan tradisional yang ada di Indonesia. Lalu mereka merasa tertarik dengan satu permainan yaitu gobak sodor. Merasa heran karena permainan tradisional sudah tidak diminati lagi, mereka sepakat untuk, memperkenalkan salah satu permainan tradisional yaitu gobak sodor, dengan bermain gobak sodor di beberapa daerah di Indonesia. Mereka harus bermain dengan mematuhi setiap peraturan yang ada dalam permainan gobak sodor, dengan harapan mereka dapat melestarikan budaya nasional, melalui permainan tradisional.

2. *Gameplay*

Permainan Gobak Sodor 3D merupakan permainan *single player*, dimana terdapat beberapa karakter pada halaman permainan yang terdiri dari :

- Karakter yang dikendalikan oleh pemain (*player* utama)
- Karakter Penjaga (NPC) yang berperan sebagai *enemies*

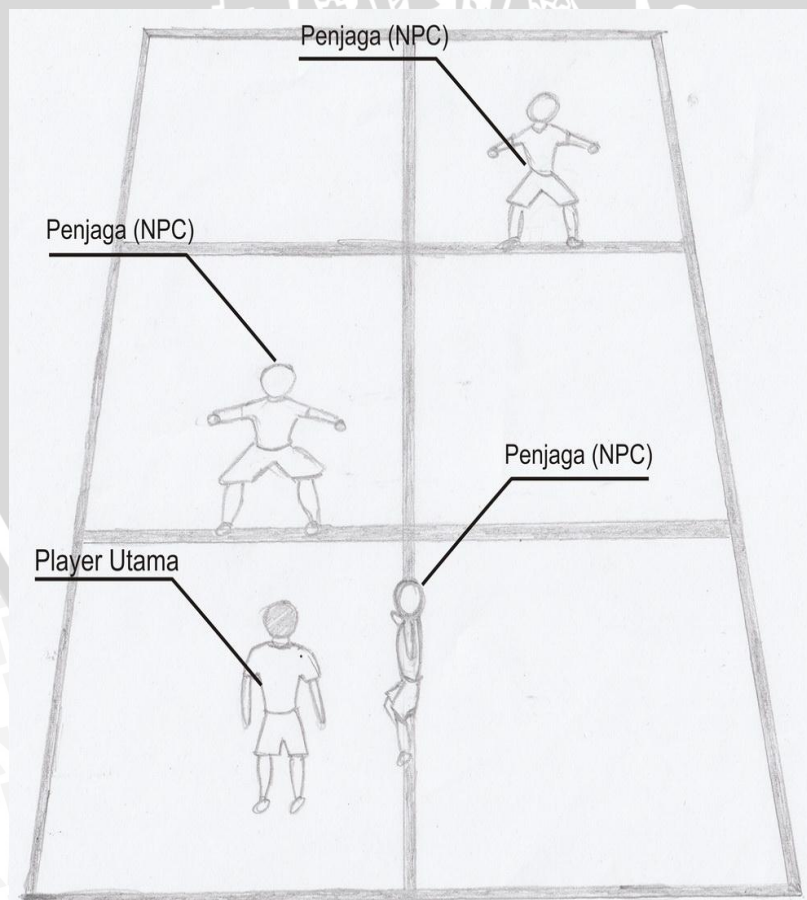
Player utama harus berusaha untuk menyeberang, dan kembali ke lokasi awal pada area yang sudah ditentukan. Area tersebut akan dijaga oleh karakter penjaga (NPC), yang bertugas untuk menghalangi *player* utama menyeberang bahkan kembali ke lokasi awal. *Player* utama tidak boleh tersentuh oleh karakter penjaga (NPC), karena ketika tersentuh maka *player* utama akan

kehilangan *life* (nyawa) yang disediakan sebanyak *default* 3. Saat *life player* utama habis, maka *Game* akan berakhir. Namun jika *player* utama dapat menyeberang dan kembali ke lokasi awal tanpa tersentuh oleh karakter penjaga, maka *player* utama dapat melanjutkan permainan ke level selanjutnya, dan berusaha untuk mencatat *Best Score*.

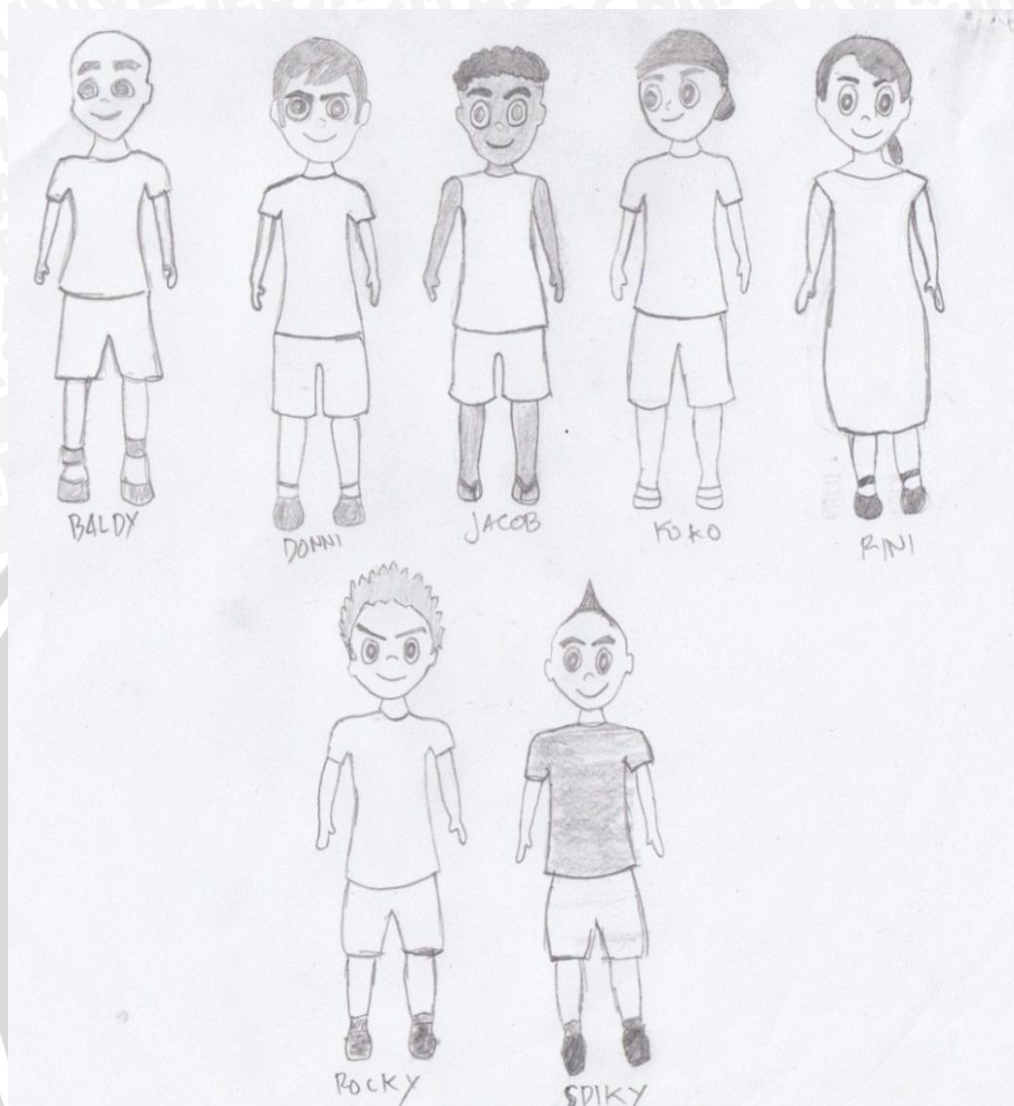
3. *Gameflow*

Gameflow menjelaskan proses interaksi di dalam *game* Gobak Sodor 3D. Untuk keterangan lebih lengkap dapat dilihat pada Gambar 3.2

Karakter *player* utama sedang berusaha melewati hadangan karakter penjaga (NPC), yang berjaga pada garis-garis yang dibuat. Tugas karakter *player* utama adalah menerobos penjaga (NPC), tanpa boleh tersentuh, untuk berhasil menyeberang, dan kembali lagi ke lokasi awal.



Gambar 3. 2 *Gameflow* Gobak Sodor 3D
Sumber : Perancangan



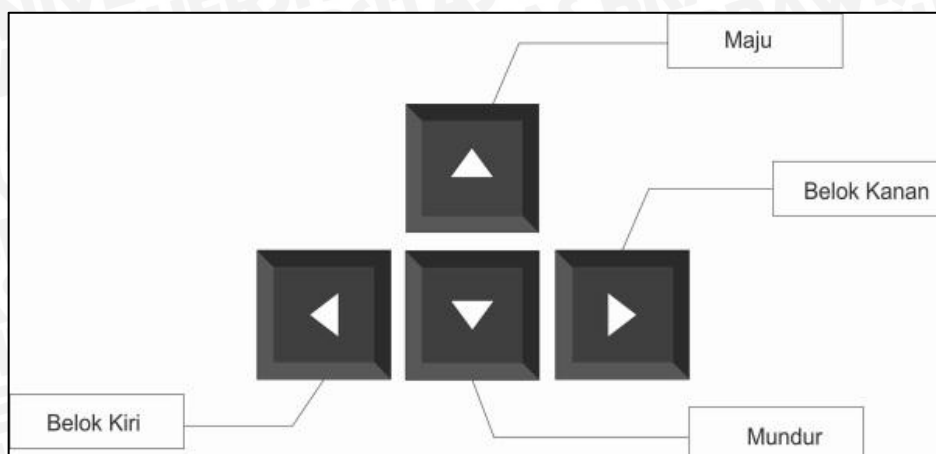
Gambar 3. 3 Rancangan karakter Gobak Sodor 3D
Sumber : Perancangan

4. Karakter

Game Gobak Sodor 3D memiliki 7 karakter yang dapat dimainkan. Untuk lebih jelasnya dapat dilihat pada Gambar 3.3.

5. Kontrol

Game Gobak Sodor 3D dikendalikan menggunakan *keyboard*. Untuk lebih jelasnya dapat dilihat pada Gambar 3.4.



Gambar 3. 4 Kontrol pada *game* Gobak Sodor 3D
Sumber : Perancangan

6. Konsep Utama *Gameplay* dan Fitur

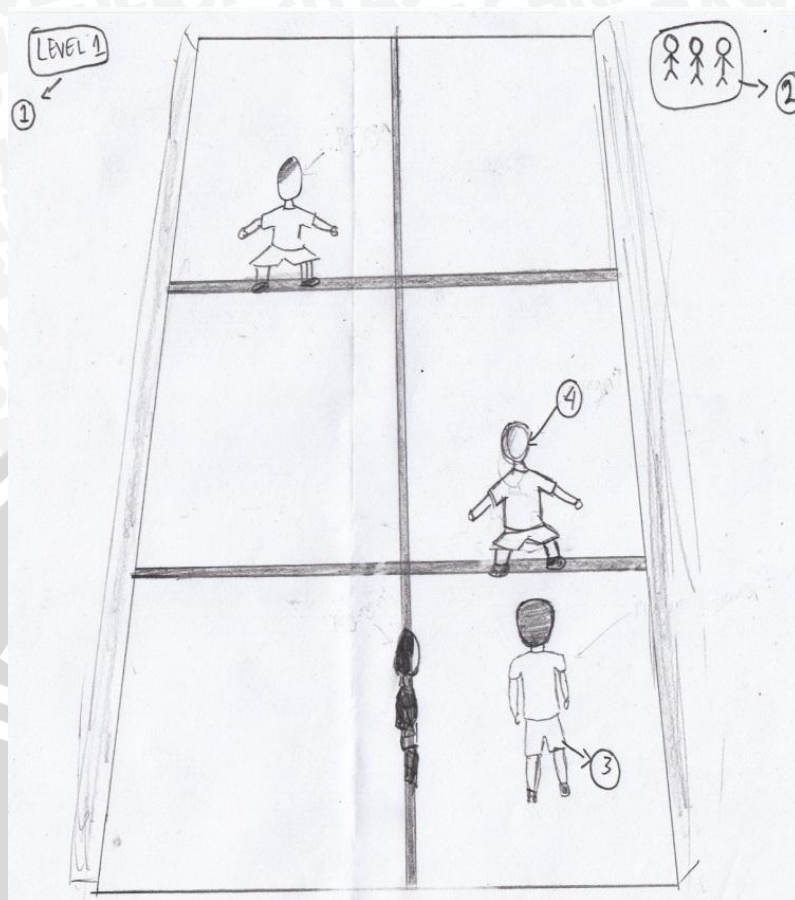
Konsep utama *gameplay* permainan Gobak Sodor 3D adalah untuk bisa memenangi setiap level permainan sehingga dapat melanjutkan ke level berikutnya. Untuk memenangi permainan, setiap pemain harus bisa mengendalikan karakter nya untuk menyeberangi area yang ditentukan, dan kembali ke titik semula tanpa boleh tersentuh penjaga (NPC).

7. *Game World*

Game world pada permainan Gobak Sodor 3D menggunakan latar belakang ikon berbagai daerah di Indonesia, seperti Candi Borobudur, Tugu Monas, dll.

8. *Interface*

Tampilan *interface game* Gobak Sodor 3D terdiri dari halaman yang akan menampilkan informasi level, nyawa, karakter *player* dan karakter penjaga (NPC). Rancangan tampilan *interface game* Gobak Sodor 3D dapat dilihat pada Gambar 3.5.



Gambar 3. 5 Rancangan *Interface* game Gobak Sodor 3D

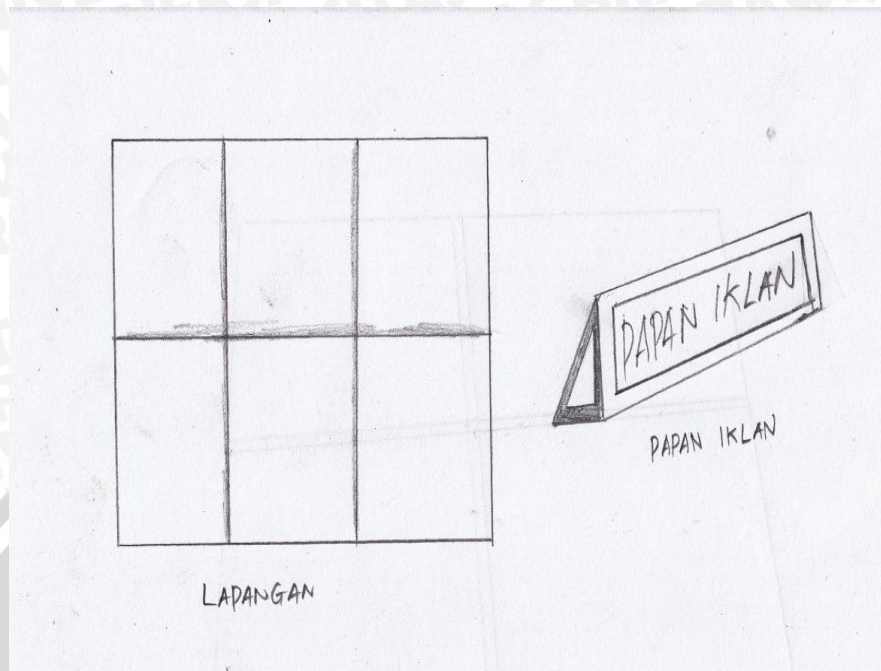
Sumber : Perancangan

- **Informasi:**

1. Informasi level permainan
2. *Life* (nyawa) pemain, akan berkurang tiap kali karakter *player* utama tersentuh penjaga.
3. Karakter *player* utama, yang dikendalikan oleh pemain.
4. Karakter penjaga (NPC), yang bertugas menjaga daerahnya, agar tidak dilewati oleh karakter *player* utama.

9. Environment

Environment adalah objek yang ada pada *game*. *Game* Gobak Sodor 3D memiliki *environment* yang akan menjadi bagian dalam *game*. Rancangan *environment* game Gobak Sodor 3D dapat dilihat pada Gambar 3.6.



Gambar 3. 6 Rancangan *environment game* Gobak Sodor 3D
Sumber : Perancangan

10. Penjaga (NPC)

Penjaga pada permainan Gobak Sodor 3D ini adalah hasil *random* dari karakter-karakter lain yang tidak terpilih sebagai *player* utama yang berperan sebagai penjaga. Tugas utama musuh adalah menghalangi karakter *player* utama untuk dapat menyeberang, dan kembali ke titik awal.

3.2.2 Technical Design

Technical design adalah tahap pemodelan menggunakan bahasa pemodelan UML untuk menentukan kebutuhan *user* menggunakan *use case diagram*, *class diagram* dan *activity diagram*.

1. Use Case

Use Case digunakan untuk memodelkan fungsionalitas dari *game* berdasarkan kebutuhan.

2. Class Diagram

Class diagram menggambarkan *class* yang ada dalam *game* dan bagaimana relasi antara *class* yang ada.

3. Activity Diagram

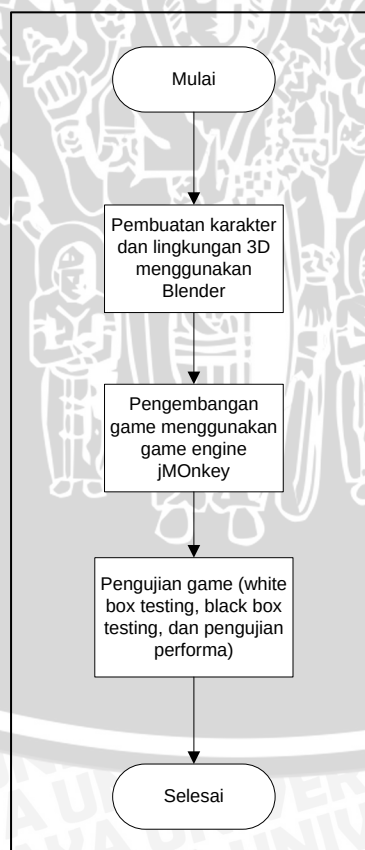
Activity diagram menggambarkan aktifitas yang dilakukan oleh user kepada sistem dan respon dari sistem.

3.3 Implementasi

Perancangan permainan gobak sodor 3D, akan dibagi menjadi beberapa tahapan, yaitu:

1. Pembuatan karakter dan lingkungan 3D menggunakan *software* blender.
2. Pengembangan *game* menggunakan *game engine* jMonkey.
3. Pengujian *game* Gobak Sodor 3D, menggunakan metode whitebox testing, blackbox testing dan pengujian performa (frame rate testing).

Tahap perancangan *game* Gobak Sodor 3D, lebih jelas dapat dilihat pada Gambar 3.7.



Gambar 3.7 flowchart implementasi *game* gobak sodor 3D

Sumber : Perancangan

3.4 Pengujian

Pengujian *game* gobak sodor 3d akan dilakukan dalam beberapa tahapan antara lain :

3.4.1 Pengujian Terstruktur (Whitebox Testing)

Pengujian struktural ini dilakukan pada unit program yang lebih kecil dari rancang bangun permainan gobak sodor 3d pada *platform* PC, yaitu pengujian yang dilakukan pada subrutin atau operasi yang terkait dengan objek. Dengan pengujian ini diharapkan agar dapat meminimalisir kesalahan-kesalahan yang mungkin terjadi pada tingkat dasar.

3.4.2 Pengujian Fungsional (Blackbox Testing)

Pengujian fungsional ini dilakukan pada spesifikasi program atau komponen pada rancang bangun permainan gobak sodor 3d pada *platform* PC. Pengujian akan dilakukan dengan cara menguji *output* gerakan karakter, setelah di beri *input* gerak berdasarkan *keyboard*. Apabila gerakan tidak sesuai dengan yang di ramalkan, maka pengujian ini telah berhasil mendeteksi masalah pada rancang bangun permainan gobak sodor 3D pada *platform* PC. Dengan pengujian ini, diharapkan agar kesalahan-kesalahan yang tidak semestinya terjadi pada permainan gobak sodor 3d dapat di deteksi.

3.4.3 Pengujian Performa (Frame Per Second)

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 permainan gobak sodor 3D. Dengan pengujian ini diharapkan performa *game* Permainan gobak sodor 3D khususnya FPS (frame per second) dapat diketahui, dan memenuhi standart yaitu minimal 60 FPS. Untuk dapat menghasilkan performa yang maksimal dalam sebuah *game* dituntut untuk dapat memenuhi standar fps yang telah ditetapkan. Standar FPS dapat dilihat pada Tabel 3.2.

Tabel 3. 2 Tabel standar FPS

Nilai FPS	Efek
< 30 FPS	Kualitas dibawah rata-rata dan gambar terlihat patah-patah.
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]

3.2.5.1 Skenario Pengujian Performa

Skenario pengujian frame rate akan dilakukan dengan menggunakan “n” komputer untuk mendapatkan hasil rata-rata frame per second.

3.4.4 Playtesting

Playtesting yang dilakukan pada *game* Gobak Sodor 3D adalah dengan meminta 10 orang *playtester* untuk memainkan *game*. Skenario *playtesting* dimulai dengan penyusunan kuisioner untuk mendapatkan umpan balik dari *playtester*. *Playtester* diberi waktu 15 menit untuk memainkan *game* kemudian penulis akan bertanya tentang pendapat *playtester* sesuai dengan kuisioner yang telah dibuat.

BAB IV

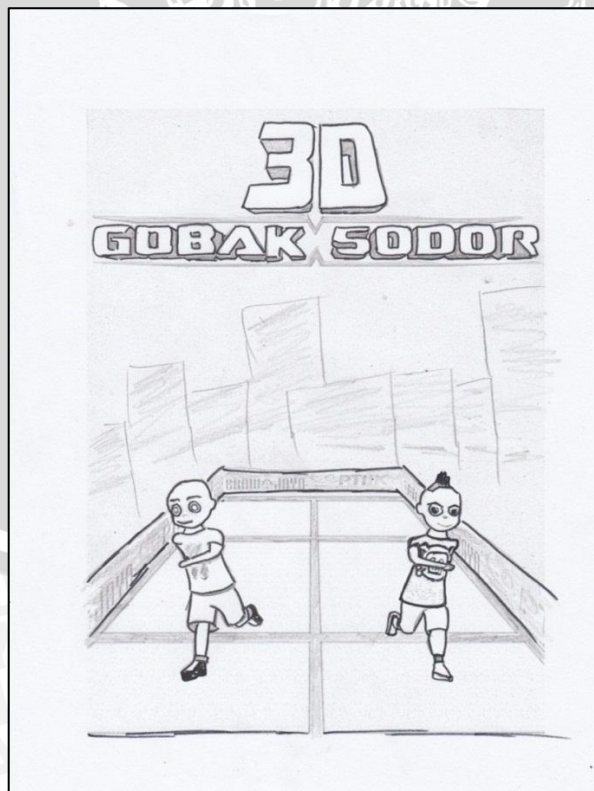
PERANCANGAN

Bab ini membahas mengenai perancangan permainan Gobak Sodor 3D yang dirancang melalui tahapan *Game Design*, dan *Technical Design*.

4.1 *Game Design*

Bagian ini akan membahas perancangan *game* Gobak Sodor 3D yang terdiri dari *Game Design* yang terdiri dari perancangan model, perancangan sistem dan perancangan *user interface*.

Perancangan *cover game* Gobak Sodor 3D adalah dengan menampilkan dua karakter yang ada dalam *game* Gobak Sodor 3D, yang sedang berlari pada arena permainan, dengan latar belakang *silhouette* perkotan. *Cover* menggambarkan bahwa permainan tradisional masih bisa tetap dikenal ditengah kemajuan zaman. Rancangan *cover game* Gobak Sodor 3D dapat dilihat pada gambar 4.1.



Gambar 4. 1 Perancangan *Cover game* Gobak Sodor 3D
Sumber : Perancangan

4.1.1 *Game Goal*

Tujuan akhir dari *game* Gobak sodor 3D, adalah untuk mencatat rekor dan waktu terbaik yang pernah di catat oleh sistem. Akumulasi *score* dan waktu akan terus berubah ketika ada pemain yang mencatat *score* dan waktu yang lebih baik.

4.1.2 *Game Outline*

Game Gobak Sodor 3D, merupakan *game single player*, yang menggunakan sudut pandang orang ke tiga, yang dikontrol menggunakan *keyboard*. Peraturan dalam permainan Gobak Sodor adalah kelompok yang mendapat giliran sebagai pemain harus bisa menyeberangi arena yang sudah ditandai dengan garis, dan dijaga oleh kelompok yang berperan sebagai penjaga. kelompok pemain harus menghindari kelompok penjaga, agar tidak tersentuh. Apabila tersentuh oleh kelompok penjaga, maka kelompok pemain harus berganti posisi dengan kelompok penjaga. Dalam *game* Gobak Sodor 3D ini, aturan tersebut di adopsi dan di aplikasikan, namun tidak seluruh peraturan. antara lain:

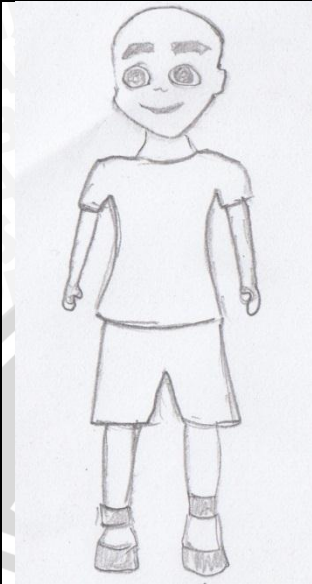
1. Dalam *game* Gobak Sodor 3D, kelompok pemain hanya diwakili oleh satu karakter *player* utama.
2. Dalam *Game* Gobak Sodor 3D, apabila karakter *player* utama berhasil tersentuh oleh penjaga, maka karakter *player* utama akan kembali ke titik awal untuk memulai kembali permainan, namun kesempatan untuk memulai kembali permainan dibatasi sebanyak 3 kali. apabila lebih dari 3 kali, maka permainan akan berakhir.

4.1.3 *Karakter*

Pada tahap ini, akan menampilkan tentang informasi karakter dan latar belakang setiap karakter yang ada dalam *game* Gobak Sodor 3D. Informasi karakter baldy dapat dilihat pada Tabel 4.1, informasi karakter Doni dapat dilihat pada Tabel 4.2, informasi karakter Jacob dapat dilihat pada Tabel 4.3, informasi karakter Koko dapat dilihat pada Tabel 4.4, informasi karakter Rini dapat dilihat pada Tabel 4.5, informasi karakter Rocky dapat dilihat pada Tabel 4.6, informasi karakter Spiky dapat dilihat pada Tabel 4.7.

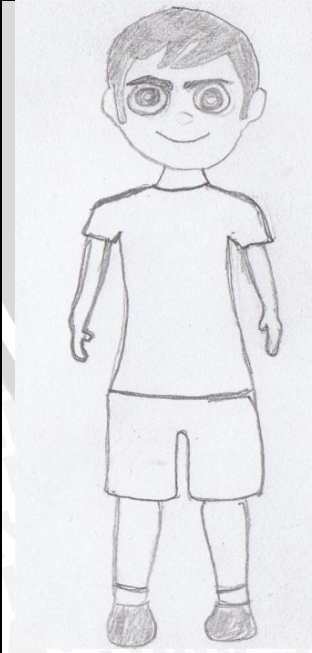
4.1.3.1 Profil Karakter

Tabel 4. 1 Profil Karakter Baldi

Karakter	Profil
	Nama : Baldy
	Jenis Kelamin : Laki-Laki
	Usia : 10 Tahun
	Latar belakang: Baldy adalah seorang anak yang berasal dari Malang, orang tuanya berwira usaha dengan membuka warung makan di kompleks tempat tinggal nya. Baldy merupakan seorang Siswa SD kelas 5. Dia suka bermain musik dan bersepeda.

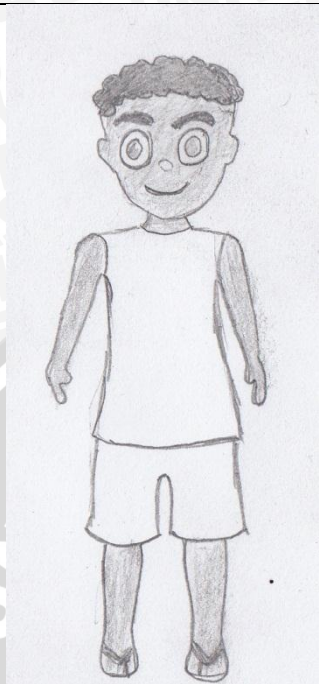
Sumber: Perancangan

Tabel 4. 2 Profil karakter Doni

Karakter	Profil
	Nama : Doni
	Jenis Kelamin : Laki-Laki
	Usia : 10 Tahun
	Latar belakang: Donni adalah seorang anak yang berasal dari Kalimantan, namun saat ini sedang mengikuti orang tua nya yang bekerja sebagai PNS di salah satu instansi DI kota Malang. sekarang kelas 5 SD sekelas dengan Baldi. Dia suka bermain sepak bola dan bermain kasti.

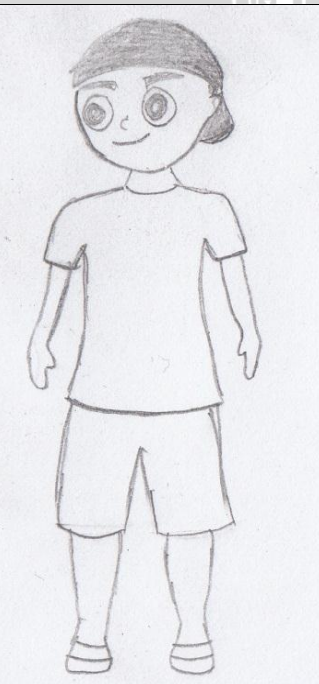
Sumber : Perancangan

Tabel 4. 3 Profil karakter Jacob

Karakter	Profil
	Nama : Jacob
	Jenis Kelamin : Laki-Laki
	Usia : 10 Tahun
	Latar belakang: Jacob adalah seorang anak yang berasal dari papua. Bersekolah di Malang, karea mengikuti orang tuanya yang membuka usaha di Malang. dia satu sekolah dengan Baldi, dan Doni, namun tidak sekelas. Jacob suka sekali olahraga, dia suka bermain Sepak bola bersama Doni.

Sumber : Perancangan

Tabel 4. 4 Profil karakter Koko

Karakter	Profil
	Nama : Koko
	Jenis Kelamin : Laki-Laki
	Usia : 10 Tahun
	Latar belakang: Koko adalah anak keturunan Tionghoa, namun kedua orang tua bahkan kakek neneknya sudah merupakan warga negara Indonesia. Koko adalah teman sekelas Jacob, dan juga merupakan sahabat Baldy, dan Doni. Koko suka bermain Video game, dan basket.

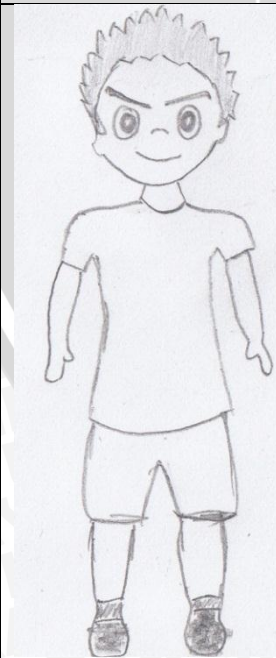
Sumber : Perancangan

Tabel 4. 5 Profil karakter Rini

Karakter	Profil
	Nama : Rini
	Jenis Kelamin : Perempuan
	Usia : 10 Tahun
	Latar belakang: Rini adalah saudara sepupu Baldy, yang kebetulan tinggal di kompleks yang sama dengan Baldy. Orang tua Rini bekerja di sebuah perusahaan swasta di kota Malang. Rini juga sekelas dengan Jacob dan koko. Rini suka bermain musik, dan dia sangat menyukai kucing.

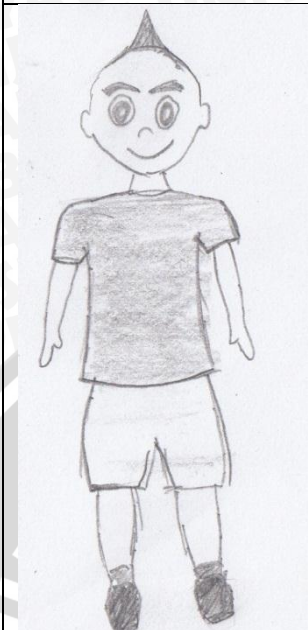
Sumber : Perancangan

Tabel 4. 6 Profil karakter Rocky

Karakter	Profil
	Nama : Rocky
	Jenis Kelamin : Laki-Laki
	Usia : 10 Tahun
	Latar belakang: Rocky adalah seorang anak dari orang tua yang berkebangsaan Australia. Sudah 5 tahun Rocky dan keluarga nya tinggal di Indonesia, kebetulan orang tuanya mendirikan sebuah LSM di Indonesia. Rocky sudah fasih berbahasa Indonesia. Rocky suka olah raga, dan membaca buku tentang budaya Indonesia.

Sumber : Perancangan

Tabel 4. 7 Profil karakter Spiky

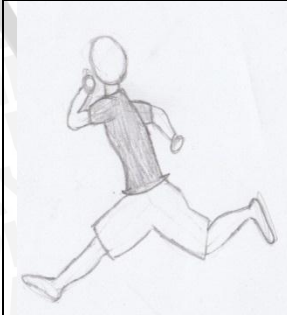
Karakter	Profil
	Nama : Spiky
	Jenis Kelamin : Laki-Laki
	Usia : 12 Tahun
	Latar belakang: Spiky adalah siswa kelas 6 SD. Dia tinggal bersama kakek dan nenek nya di Malang, sedangkan orang tuanya ada di Sumatera untuk bekerja. Spiky pernah tdak naik kelas saat di kelas 4 SD. Spiky suka berkumpul dan bermain bersama Baldy, Doni, Jacob, Koko, Rini dan Rocky.

Sumber : Perancangan

4.1.4 Gerakan Karakter

Bagian ini akan menjelaskan tentang gerakan-gerakan yang tersedia untuk setiap karakter dalam permainan gobak sodor 3D. Gerakan-gerakan yang akan digunakan oleh karakter pemain utama adalah gerakan lari, dan gerakan stand, sedangkan untuk karakter penjaga menggunakan gerakan jaga. gerakan lari dapat dilihat pada Tabel 4.8, gerakan jaga dapat dilihat pada Tabel 4.9, gerakan *stand* (*idle*) dapat dilihat pada Tabel 4.10

Tabel 4. 8 Gerakan Lari

Gambar	Keterangan
	Jenis Gerakan : Lari
	Gerakan lari, merupakan gerakan utama bagi karakter <i>Player</i> . Karena, dengan gerakan lari, karakter <i>player</i> dapat berpindah tempat dan menghindari penjaga, untuk menyelesaikan level.

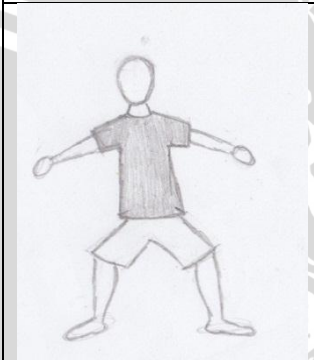
Sumber : Perancangan

Tabel 4. 9 Gerakan *stand (idle)*

Gambar	Keterangan
	Jenis Gerakan : idle
	Gerakan <i>idle</i> , atau gerakan yang dilakukan karakter <i>player</i> ketika sedang berhenti berlari. Gerakan ini tidak mempengaruhi jalan nya permainan.

Sumber : Perancangan

Tabel 4. 10 Gerakan Jaga

Gambar	Keterangan
	Jenis Gerakan : Jaga
	Gerakan Jaga merupakan gerakan utama karakter Penjaga (NPC). Karena, dengan gerakan jaga, karakter penjaga dapat menghalangi karakter <i>player</i> untuk melewati daerah yang dijaga nya.

Sumber : Perancangan

4.1.5 Klasifikasi Karakter

Pada bagian ini akan menjelaskan klasifikasi karakter yang terdapat pada *game* Gobak Sodor 3D, yang terdiri dari karakter pemain (*player* utama), dan karakter *enemy* (NPC), juga objek yang tersedia di dalam *game* Gobak Sodor 3D.

4.1.5.1 Karakter Pemain (*player* utama)

Karakter *Player* utama adalah karakter yang dikendalikan oleh pemain. Gerakan *player* utama ditentukan oleh perintah yang diberikan oleh pemain melalui kontrol yang tersedia. Karakter utama merupakan karakter yang dipilih oleh pemain pada halaman Pilih Karakter. Pada halaman pilih karakter terdapat 7 karakter yang dapat dipilih oleh pemain yaitu karakter Baldy, Doni, Jacob, Koko,

Rini, Rocky, dan Spiky. Untuk dapat menyelesaikan setiap level, pemain harus bisa menggerakkan karakter *player* utama melewati hadangan dari penjaga (NPC).

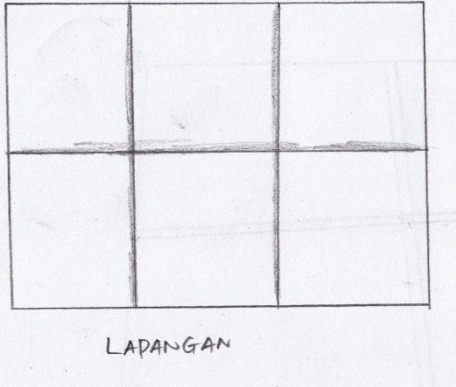
4.1.5.2 Karakter *Enemy* (NPC)

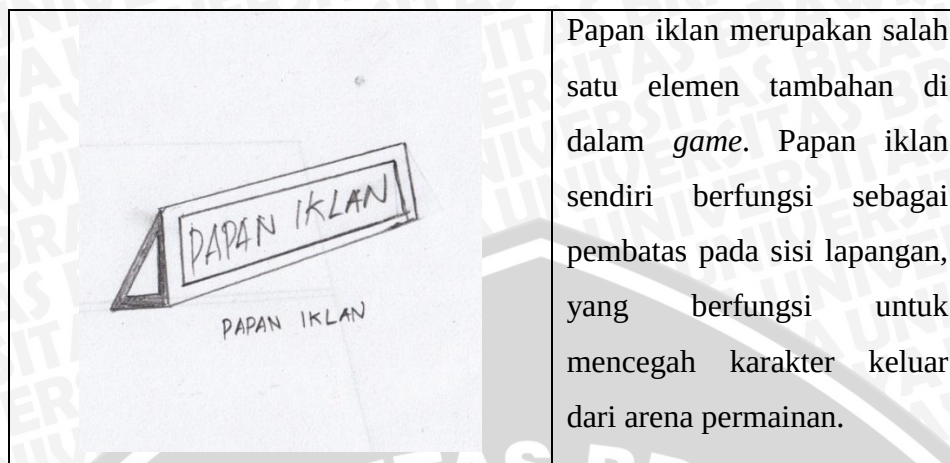
Karakter *enemy* adalah karakter yang berperan sebagai penjaga dalam permainan Gobak Sodor 3D. Karakter *enemy* merupakan karakter *random* dari ke enam karakter yang tidak terpilih sebagai karakter *player* utama. Dalam menjalankan tugas nya untuk menjaga daerah nya, karakter *enemy* digerakan menggunakan algoritma AI sederhana. Untuk setiap level, kecepatan gerak dari karakter *enemy* akan semakin bertambah, sehingga menambah tantangan dalam permainan ini.

4.1.6 Environment

Pada bagian ini akan dijelaskan tentang objek-objek yang terdapat pada lingkungan sekitar *game* Gobak Sodor 3D. Keterangan dapat dilihat pada Tabel 4.11 berikut ini:

Tabel 4. 11 Keterangan objek

Jenis <i>Environment</i>	Keterangan
	Lapangan merupakan elemen terpenting dalam permainan gobak sodor. Hal ini dikarenakan lapangan merupakan inti dari permainan. Lapangan digunakan untuk menandai daerah yang harus dilewati pemain, dan juga merupakan basis gerak dari setiap penjaga.



Papan iklan merupakan salah satu elemen tambahan di dalam *game*. Papan iklan sendiri berfungsi sebagai pembatas pada sisi lapangan, yang berfungsi untuk mencegah karakter keluar dari arena permainan.

Sumber : Perancangan

4.1.7 Kontrol Pada Game

Pada bagian ini akan dijelaskan tentang kontrol yang digunakan dalam *game*, yaitu *keyboard*. *Game* akan sepenuhnya dikontrol menggunakan *keyboard* yaitu tombol arah yang terdapat pada setiap *keyboard*. Untuk lebih jelasnya dapat dilihat pada tabel dibawah ini.

4.1.7.1 Maju

Gerakan maju dihasilkan oleh inputan arah seperti yang ditunjukkan pada Tabel 4.12 dibawah ini. Ketika arah maju di tekan, maka karakter *player* utama akan bergerak lurus maju ke depan sesuai dengan arah panah yang ditunjukkan gambar pada tabel. Karakter akan berhenti bergerak ketika tombol dilepaskan.

Tabel 4. 12 Kontrol gerakan maju

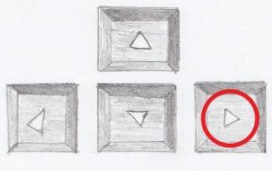
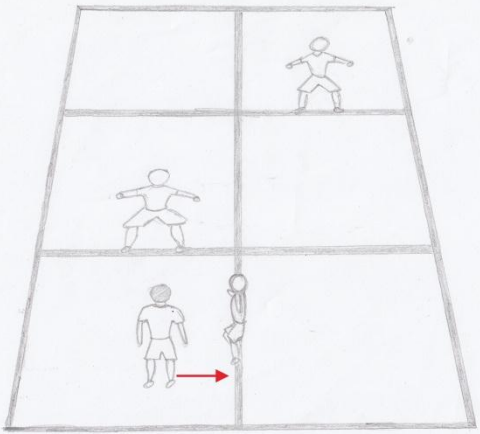
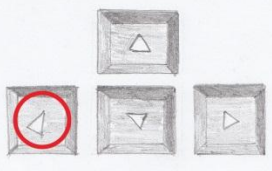
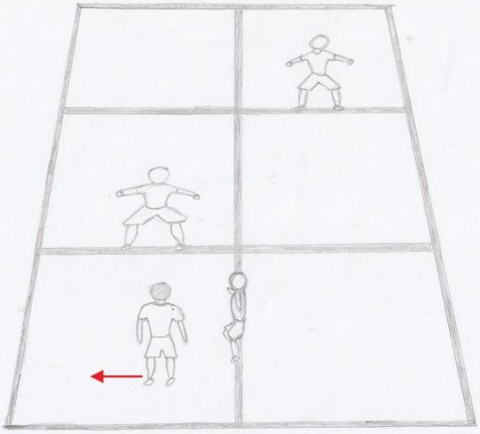
Kontrol	Gerakan yang dihasilkan

Sumber: Perancangan

4.1.7.2 Belok

Gerakan belok dihasilkan oleh inputan arah seperti yang ditunjukkan pada Tabel 4.13 dibawah ini. Ketika arah kanan di tekan, maka karakter *player* utama akan bergerak ke arah kanan sesuai dengan arah panah yang ditunjukkan gambar pada tabel. Begitu pula ketika arah kiri di tekan, maka karakter akan berbelok ke arah kiri sesuai dengan arah panah yang ditunjukkan gambar pada tabel. karakter akan berhenti bergerak ketika tombol arah kiri maupun kanan berhenti di tekan.

Tabel 4. 13 Kontrol gerakan belok kanan dan kiri

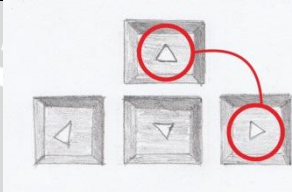
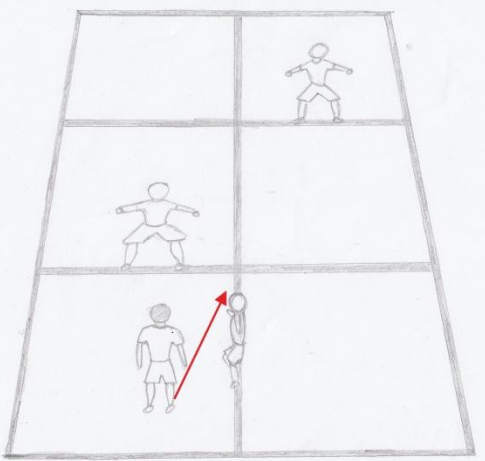
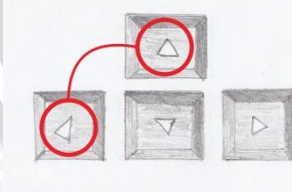
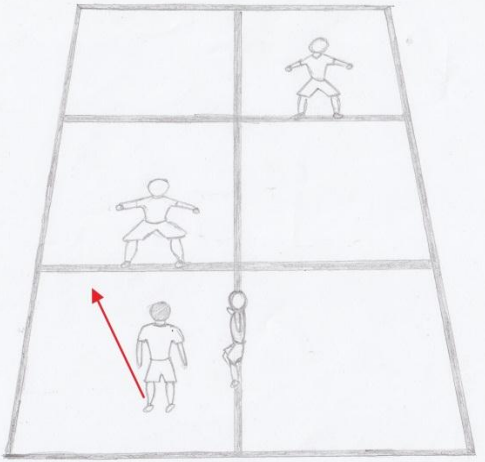
Kontrol	Gerakan yang dihasilkan
	
	

Sumber : Perancangan

4.1.7.3 Maju Menyamping

Gerakan maju menyamping dihasilkan oleh inputan arah seperti yang ditunjukkan pada Tabel 4.14 dibawah ini. Ketika arah maju dan arah kanan di tekan bersamaan, maka karakter *player* utama akan bergerak maju menyamping ke arah kanan sesuai dengan arah panah yang ditunjukkan gambar pada tabel. Begitu pula ketika arah maju dan arah kiri ditekan bersamaan, maka karakter *player* utama akan bergerak maju menyamping ke arah kiri sesuai dengan arah panah yang ditunjukkan gambar pada tabel. Karakter akan berhenti bergerak ketika tombol dilepaskan.

Tabel 4. 14 Kontrol gerakan maju menyamping kiri dan kanan

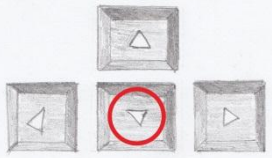
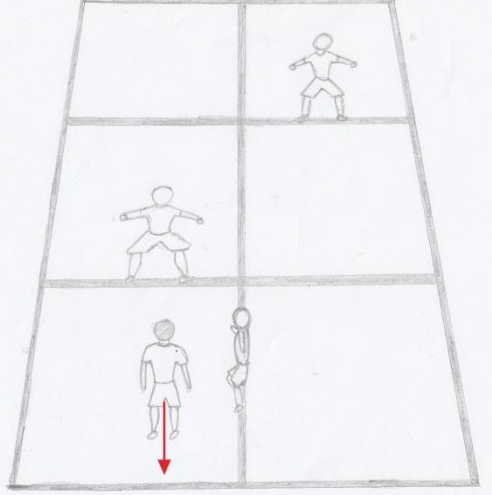
Kontrol	Gerakan yang dihasilkan
	
	

Sumber : Perancangan

4.1.7.4 Mundur

Gerakan mundur dihasilkan oleh inputan arah seperti yang ditunjukkan pada Tabel 4.15 dibawah ini. Ketika arah mundur di tekan, maka karakter *player* utama akan bergerak lurus mundur ke belakang sesuai dengan arah panah yang ditunjukkan gambar pada tabel. Karakter akan berhenti bergerak ketika tombol dilepaskan.

Tabel 4. 15 Kontrol gerakan mundur

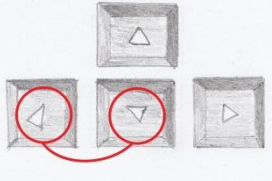
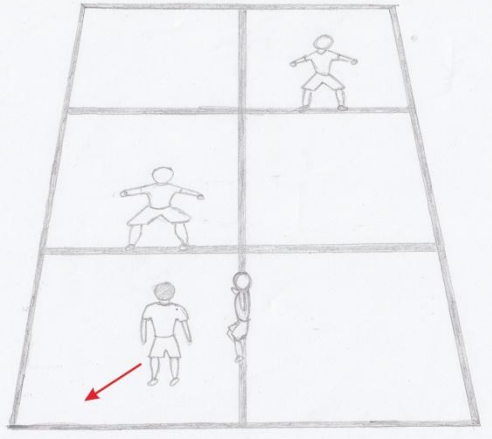
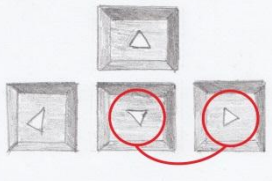
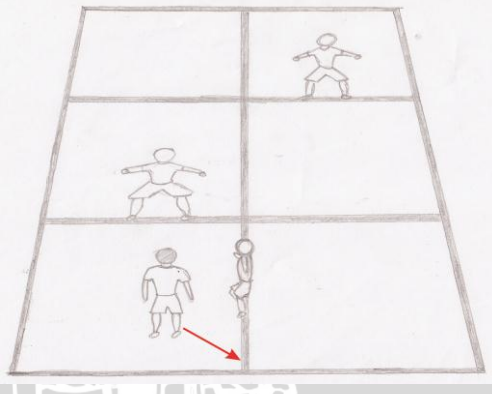
Kontrol	Gerakan yang dihasilkan
	

Sumber : Perancangan

4.1.7.5 Mundur Menyamping

Gerakan mundur menyamping dihasilkan oleh inputan arah seperti yang ditunjukkan pada Tabel 4.16 dibawah ini. Ketika arah mundur dan arah kanan di tekan bersamaan, maka karakter *player* utama akan bergerak mundur menyamping ke arah kanan sesuai dengan arah panah yang ditunjukkan gambar pada tabel. Begitu pula ketika arah mundur dan arah kiri ditekan bersamaan, maka karakter *player* utama akan bergerak mundur menyamping ke arah kiri sesuai dengan arah panah yang ditunjukkan gambar pada tabel. Karakter akan berhenti bergerak ketika tombol dilepaskan.

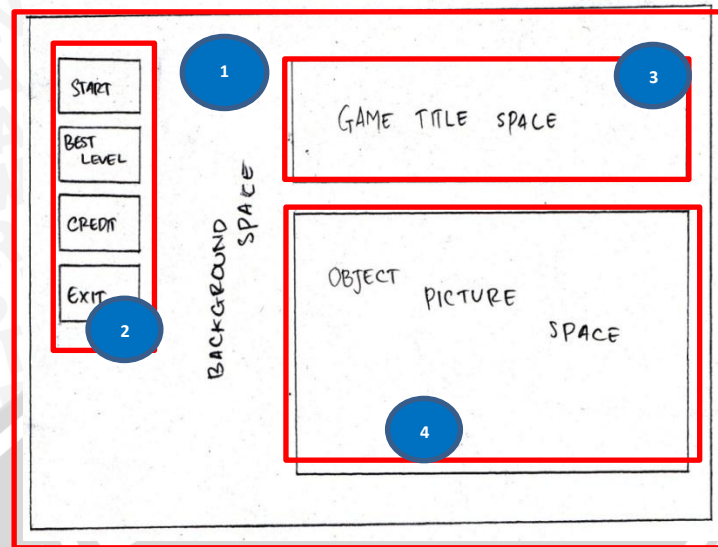
Tabel 4. 16 Kontrol gerakan mundur menyamping kiri dan kanan

Kontrol	Gerakan yang dihasilkan
	
	

Sumber : Perancangan

4.1.8 Antarmuka Game

Pada bagian ini akan menjelaskan tampilan *game* gobak sodor 3D. Dimulai dari halaman menu, halaman pilih karakter dan halaman permainan. untuk lebih jelasnya dapat dilihat Rancangan halaman menu pada Gambar 4.2 , rancangan halaman pilih karakter pada Gambar 4.3 dan rancangan halaman permainan pada Gambar 4.4.



Gambar 4. 2 Rancangan antarmuka menu
Sumber : Perancangan

4.1.8.1 Antarmuka Menu

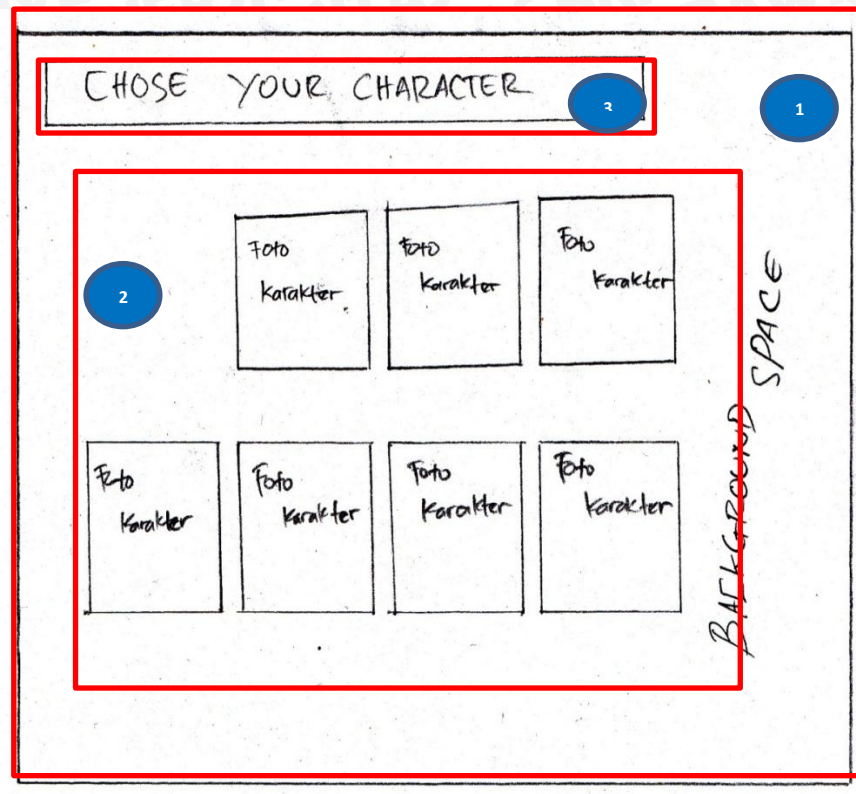
Halaman menu adalah halaman yang memuat menu-menu utama di dalam *game* Gobak Sodor 3D, antara lain *Start*, *Best Score*, *Credit*, dan *Exit*. Perancangan halaman menu ditunjukkan pada Gambar 4.2.

Berdasarkan Gambar 4.2 ditunjukkan bahwa pada halaman menu terdapat terdapat 4 kolom yang akan dijelaskan di bawah:

1. Kolom background
2. Kolom pilihan menu
3. Kolom judul *game*
4. Kolom objek gambar (karakter/logo)

4.1.8.2 Antarmuka Pilih Karakter

Halaman pilih karakter merupakan halaman yang menampilkan pilihan karakter yang terdapat pada *game* Gobak Sodor 3D. Pada halaman ini akan menampilkan icon dan foto setiap karakter antara lain karakter Baldi, karakter Doni, karakter Jacob, karakter Koko, karakter Rini, karakter Rocky, dan karakter Spiky. Tampilan perancangan halaman pilih karakter dapat dilihat pada Gambar 4.3.



Gambar 4. 3 Rancangan antarmuka pilih karakter

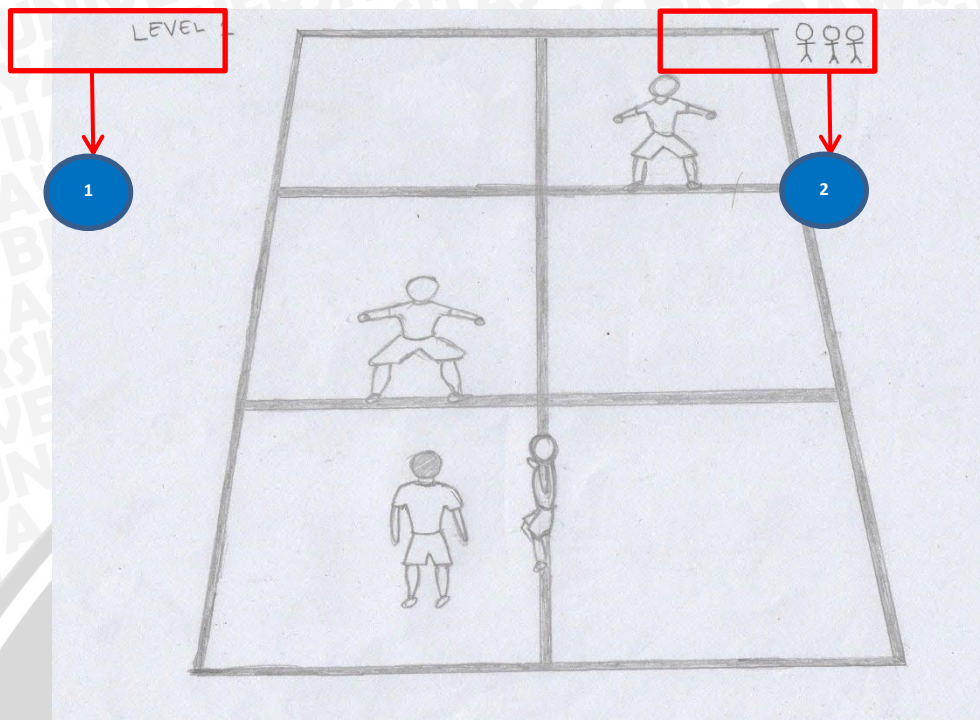
Sumber : Perancangan

Dari gambar diatas ditunjukan bahwa pada halaman pilih karakter terdapat terdapat 3 kolom yang akan dijelaskan di bawah:

1. Kolom background
2. Kolom karakter yang dapat dipilih
3. Kolom judul halaman

4.1.8.3 Antarmuka Permainan

Halaman permainan merupakan halaman yang menampilkan secara keseluruhan *gameplay*, aktifitas, dan interaksi yang terjadi di dalam *game* Gobak Sodor 3D. Rancangan halaman permainan dapat dilihat pada Gambar 4.4.



Gambar 4. 4 Rancangan antarmuka permainan
Sumber : Perancangan

Dari gambar rancangan halaman permainan di atas, dapat dilihat bahwa pada halaman permainan akan terdapat :

1. *Level status*
2. *Life Status*

4.1.9 Level Game

Pada bagian ini akan menjelaskan tentang level yang terdapat pada *game* Gobak Sodor 3D. Bagian ini akan menjelaskan deskripsi level dan *background* level.

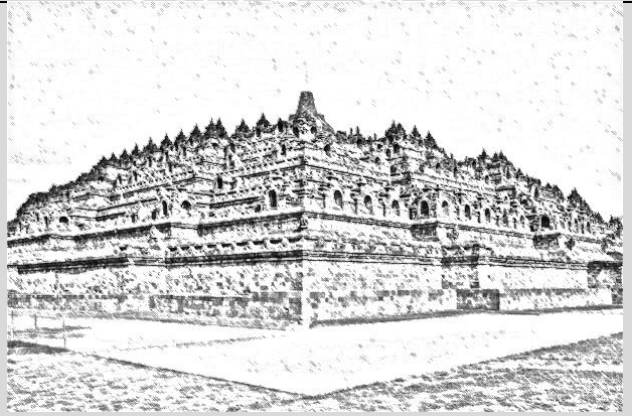
4.1.9.1 Deskripsi Level

Level dalam permainan Gobak Sodor 3D memiliki karakteristik yang berbeda-beda di setiap tingkatannya. Baik dari segi tampilan maupun kesulitan. Untuk tampilan, akan dibuat *background* yang mengangkat tema budaya suatu daerah, maupun ikon-ikon tertentu suatu daerah. contoh tugu Monas di Jakarta dan Candi Borobudur di Yogyakarta.

4.1.9.2 Latar Belakang Level

Pada bagian ini akan dibahas mengenai latar belakang yang ada pada tiap level, dimulai dari level 1 yang memiliki latar belakang halaman rektorat Universitas brawijaya, level dua yang memiliki latar belakang tugu Monas dan level 3 yang memiliki latar belakang Candi Borobudur. untuk lebih jelas nya dapat dilihat pada Tabel 4.17.

Tabel 4. 17 Latar belakang level

Level	Latar Belakang	Keterangan
1 (satu)		Mengambil latar belakang halaman rektorat Universitas Brawijaya sebagai latar belakang level 1
2 (dua)		Mengambil latar belakang Candi Borobudur sebagai latar belakang level 2.

3 (tiga)		Mengambil latar belakang Tugu Monas sebagai latar belakang level 3.
----------	--	---

Sumber : Perancangan

4.1.10 Sistem Pada Game

Pada bagian ini akan menjelaskan tentang sistem-sistem yang terdapat pada *game*, antara lain NPC, *life status*, sudut pandang kamera, dan *audio (music dan sound effect)* yang terdapat pada *game* Gobak Sodor 3D.

4.1.11.1 NPC

Karakter NPC (*Non Player Control*) digerakan menggunakan algoritma AI sederhana. Untuk setiap level, kecepatan gerak dari karakter NPC akan semakin bertambah, sehingga menambah tantangan dalam permainan ini.

4.1.11.2 Life Status

Secara *default* karakter *player* utama memiliki tiga *life status*, atau tiga kesempatan untuk tersentuh oleh karakter penjaga. Setiap kali karakter *player* utama tersentuh oleh karakter penjaga, maka karakter *player* utama secara otomatis akan kembali ke titik *start* untuk memulai kembali permainan. Namun *life status* karakter *player* utama sudah berkurang satu. Apabila setelah tiga kali karakter *player* utama tersentuh oleh penjaga, maka *game* akan berakhir.

4.1.11.3 Sudut Pandang Kamera

Sudut pandang kamera pada permainan Gobak Sodor 3D adalah sudut pandang orang ketiga. Sehingga *player* dapat melihat secara utuh karakter yang sedang dikendalikan.

4.1.11.4 Audio (Music dan Sound effect)

Permainan Gobak Sodor 3D akan memiliki *audio* berupa musik dan efek suara. Setiap file *audio* akan berformat *.wav, dan merupakan *file* yang tidak memiliki lisensi, atau *free file*.

4.2 Technical Design

Technical design adalah proses pemodelan menggunakan bahasa pemodelan UML untuk menentukan kebutuhan *user* dalam *use case diagram*, *class diagram*, dan *activity diagram*.

4.2.1 Use Case

Diagram *use case* adalah salah satu diagram untuk memodelkan aspek perilaku sistem. Diagram *use case* menunjukkan sekumpulan *use case*, aktor, dan hubungannya. *Use case* merupakan fungsionalitas dari sistem yang diinisialisasi oleh aktor. Sebelum membuat *use case diagram*, sebaiknya kita lebih dulu mengetahui siapa aktor yang terlibat di dalam *use case*, dan apa saja kebutuhan yang harus disiapkan oleh sistem.

4.2.1.1 Identifikasi Aktor

Tahap ini adalah tahap untuk melakukan identifikasi terhadap aktor yang akan berinteraksi dengan *game* Gobak Sodor 3D. Tabel 4.18 memperlihatkan aktor yang terlibat beserta penjelasannya yang merupakan hasil dari proses identifikasi aktor.

Tabel 4. 18 Identifikasi aktor

Aktor	Deskripsi
<i>Player</i>	<i>Player</i> adalah pengguna yang dapat

	memainkan <i>game</i> Gobak Sodor 3D, dan memiliki akses untuk melihat best score, dan credit.
--	--

Sumber : Perancangan

4.2.1.2 Daftar Kebutuhan

Daftar kebutuhan adalah daftar yang menjabarkan kebutuhan fungsional yang ada di dalam *game* Gobak Sodor 3D. Daftar kebutuhan *game* Gobak Sodor 3D dapat dilihat pada tabel 4.19.

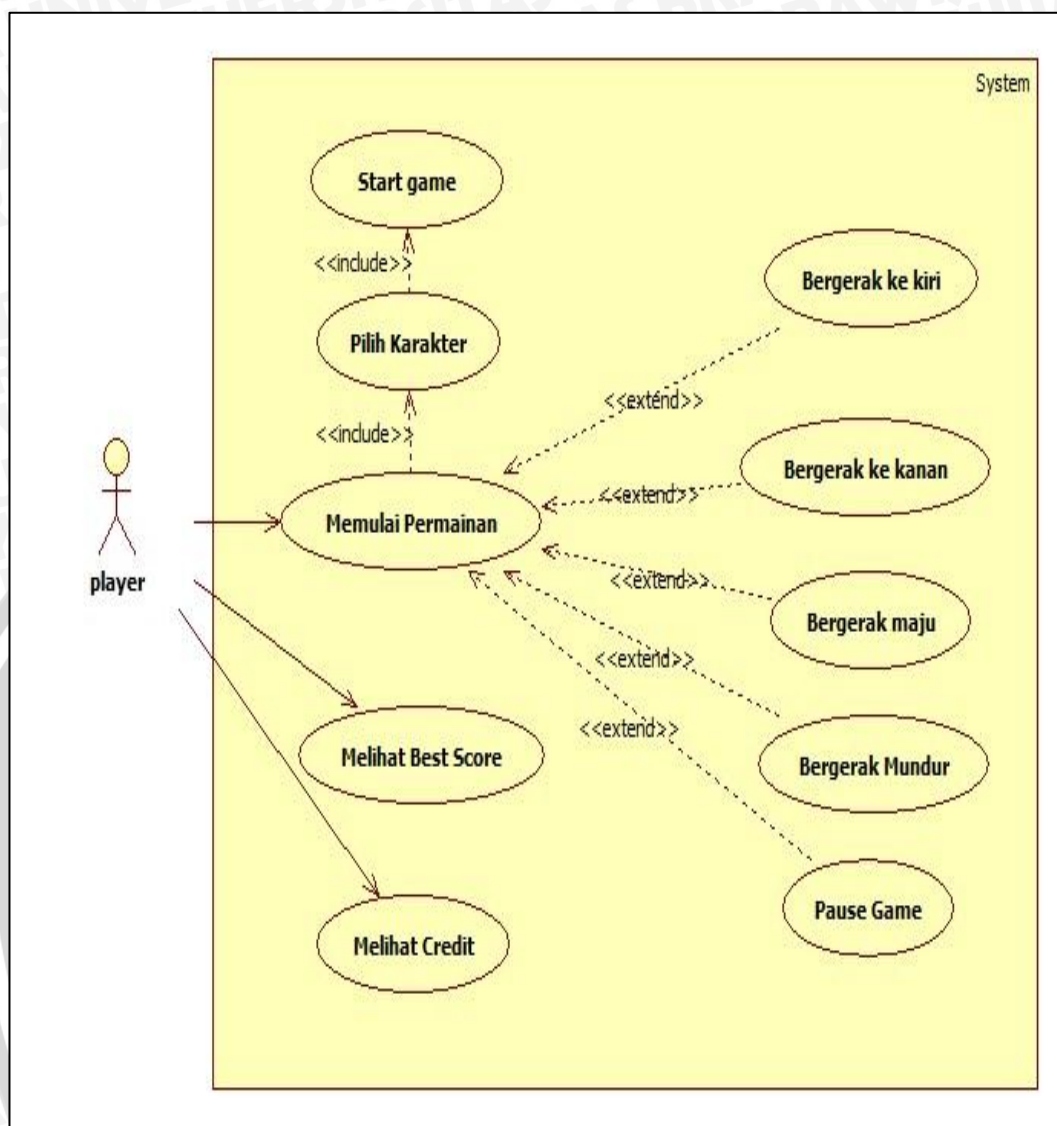
Tabel 4. 19 Daftar kebutuhan

Kode	Kebutuhan	Use Case
TDD_001	<i>Game</i> harus menyediakan fasilitas untuk memulai permainan.	Start Game
TDD_002	<i>Game</i> harus menyediakan fasilitas untuk memilih karakter. Sehingga pemain dapat memilih karakter yang akan menjadi karakter <i>player</i> utama.	Pilih Karakter
TDD_003	<i>Game</i> harus menyediakan fasilitas untuk memulai permainan. Sehingga pemain dapat memainkan permainan.	Memulai Permainan
TDD_004	<i>Game</i> harus menyediakan fasilitas untuk menggerakan karakter. sehingga pemain dapat menggerakan karakter <i>player</i> utama. Gerakan-gerakan yang dihasilkan antara lain: • Bergerak maju • Bergerak mundur • Bergerak ke kiri	Bergerak ke Kiri, Bergerak ke Kanan, Bergerak Maju dan Bergerak Mundur

	Bergerak ke kanan	
TDD_005	Game harus menyediakan fasilitas untuk <i>pause game</i> . Sehingga pemain dapat menghentikan <i>game</i> untuk sementara waktu.	Pause Game
TDD_006	Game harus menyediakan fasilitas untuk melihat <i>best score</i> yang pernah dicatat oleh sistem. Sehingga pemain dapat melihat catatan <i>best score</i> .	Best Score
TDD_007	Game harus menyediakan fasilitas untuk melihat <i>credit</i> . Sehingga pemain dapat melihat siapa saja yang terlibat dalam pembuatan <i>game</i> .	Credit
TDD_008	Game harus menyediakan fasilitas untuk keluar dari <i>game</i> . Sehingga pemain dapat mengakhiri permainan dan keluar dari <i>game</i> .	Exit

Sumber : Perancangan

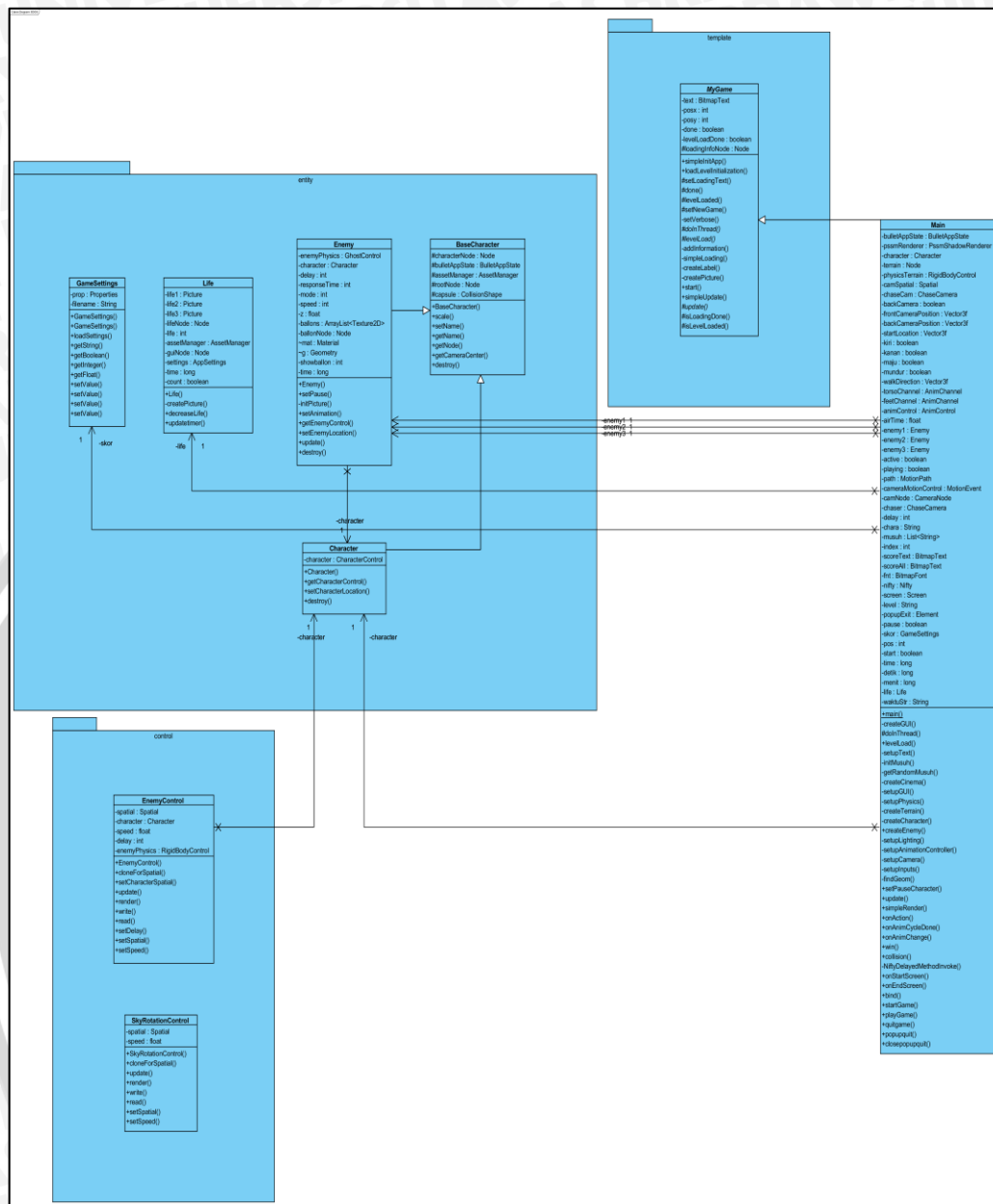
Use case diagram dalam perancangan *game* Gobak Sodor 3D digunakan untuk memodelkan fungsionalitas dari *game*. Diagram *use case* ini melibatkan *player* sebagai aktor dan beberapa *use case*. Setiap *use case* memiliki activity diagram untuk menjelaskan aktivitas yang terjadi pada masing-masing *use case*. *Use case diagram game* Gobak Sodor 3D ditunjukkan pada Gambar 4.5.



Gambar 4. 5 Use Case diagram Permainan Gobak Sodor 3D
Sumber : Perancangan

4.2.2 Class Diagram

Diagram *class* memberikan gambaran pemodelan elemen – elemen *class* yang membentuk sebuah perangkat lunak. *Class* bisa didapatkan dengan menganalisis secara detail terhadap *use case* yang dimodelkan. Selain itu, *class* dibuat dengan menyesuaikan komponen-komponen *game engine* yang digunakan. Berikut adalah rancangan diagram *class game* Gobak sodor 3D, yang ditunjukkan pada Gambar 4.6.



Gambar 4. 6 *Class Diagram* Permainan Gobak Sodor 3D

Sumber : Perancangan

Pada diagram *class* di atas digambarkan *interface* dan beberapa *class* utama yang menyusun *game* Gobak Sodor 3D. Deskripsi dari *interface* dan beberapa *class* utama akan dijelaskan pada Tabel 4.20.

Tabel 4. 20 Deskripsi *class diagram* Gobak Sodor 3D

NO	Class	Keterangan
1.	Main	Class yang pertama kali di eksekusi saat program

		dijalankan.
2.	<i>MyGame</i>	<i>Class</i> yang berfungsi membagi <i>thread</i> yg akan di <i>load</i> pada saat <i>game</i> dijalankan
3.	<i>BaseCharacter</i>	<i>Class</i> yang mengatur ukuran dan fisik setiap karakter yang ada di dalam <i>game</i>
4.	<i>Enemy</i>	<i>Class</i> yang memanggil, menentukan lokasi dan animasi dari karakter <i>enemy</i> (NPC).
5.	<i>Life</i>	<i>Class</i> yang mengatur nyawa karakter <i>player</i> utama.
6.	<i>GameSetting</i>	<i>Class</i> yang menghubungkan ke file properties yang berisi data record.
7.	<i>Character</i>	<i>Class</i> yang mengatur lokasi dan kontrol karakter.
8.	<i>EnemyControl</i>	<i>Class</i> yang mengatur kecepatan, dan fisik dari karakter <i>enemy</i> (NPC).
9.	<i>SkyRotationControl</i>	<i>Class</i> yang mengatur gerakan rotasi <i>skybox</i> .

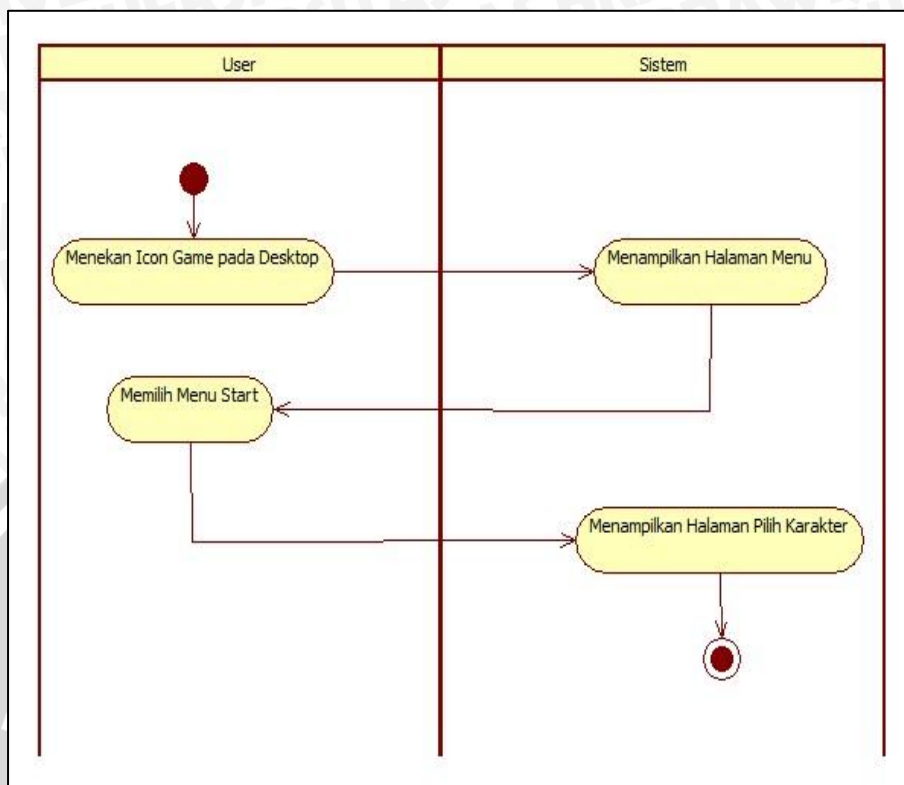
Sumber : Perancangan

4.2.3 Activity Diagram

Activity diagram menunjukkan proses-proses yang terjadi ketika *player* melakukan tindakan pada sistem sesuai dengan *use case*.

4.2.3.1 Activity Diagram Start

Activity diagram start, menunjukan aktifitas *player* ketika akan memulai permainan. *Player* memilih icon aplikasi *Game Gobak Sodor 3D*, yang telah diinstal pada PC, maka sistem akan menampilkan halaman menu, kemudian *player* memilih menu *start* yang ada pada halaman menu. Sistem kemudian menanggapi dengan menampilkan halaman pilih karakter, yang terdapat tujuh karakter yang dapat dipilih oleh pemain. *Activity diagram start* ditunjukan pada Gambar 4.7

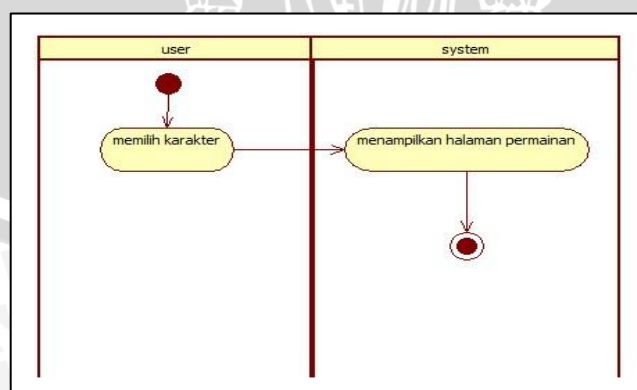


Gambar 4. 7 Activity Diagram Start

Sumber : Perancangan

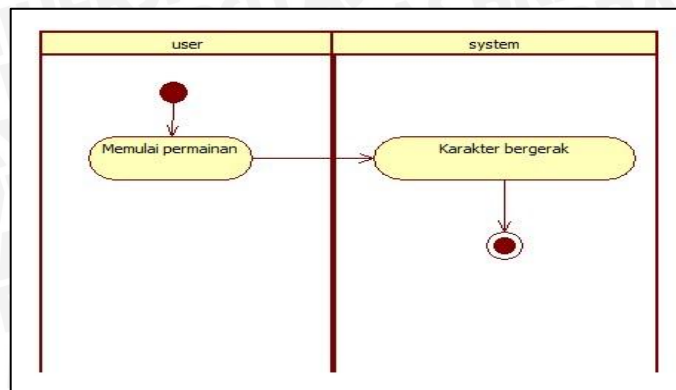
4.2.3.2 Activity Diagram Pilih Karakter

Activity diagram pilih karakter menunjukkan aktifitas *player* ketika berada pada halaman pilih karakter. *Player* kemudian memilih salah satu karakter, kemudian sistem menanggapi dengan menampilkan halaman permainan. Activity Diagram Pilih Karakter ditunjukkan pada Gambar 4.8.



Gambar 4. 8 Activity Diagram Pilih Karakter

Sumber : Perancangan



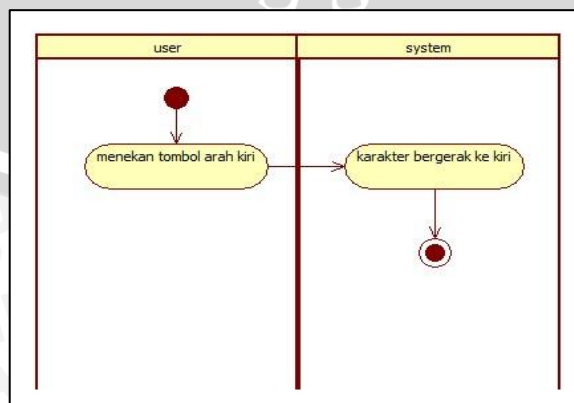
Gambar 4. 9 Activity Diagram Mula Permainan
Sumber : Perancangan

4.2.3.3 Activity Diagram Mulai Permainan

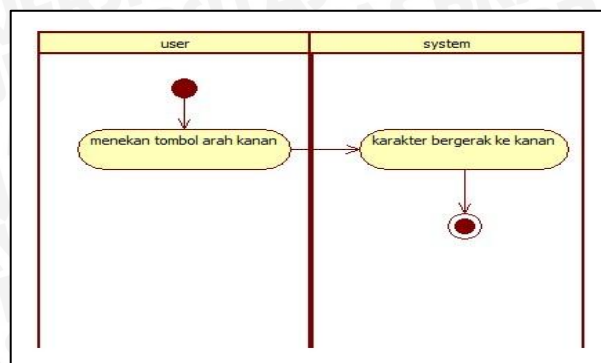
Activity diagram mulai permainan menunjukkan aktifitas *player* ketika berada pada halaman permainan. *Player* dapat memulai permainan, yang kemudian akan ditanggapi oleh sistem dengan menggerakkan karakter yang ada di dalam halaman permainan, baik karakter *player* utama, maupun karakter NPC. Activity Diagram Mulai Permainan ditunjukkan pada Gambar 4.9.

4.2.3.4 Activity Diagram Bergerak ke Kiri

Activity diagram bergerak ke kiri menunjukkan aktifitas *player* ketika berada pada halaman permainan. *Player* menggerakkan karakter menggunakan kontroler yang disediakan dengan menekan tombol arah kiri, yang kemudian ditanggapi oleh sistem yang menghasilkan karakter yang bergerak ke arah kiri. Activity Diagram Bergerak ke Kiri ditunjukkan pada Gambar 4.10.



Gambar 4. 10 Activity Diagram Bergerak ke Kiri
Sumber : Perancangan



Gambar 4. 11 Activity Diagram Bergerak ke Kanan

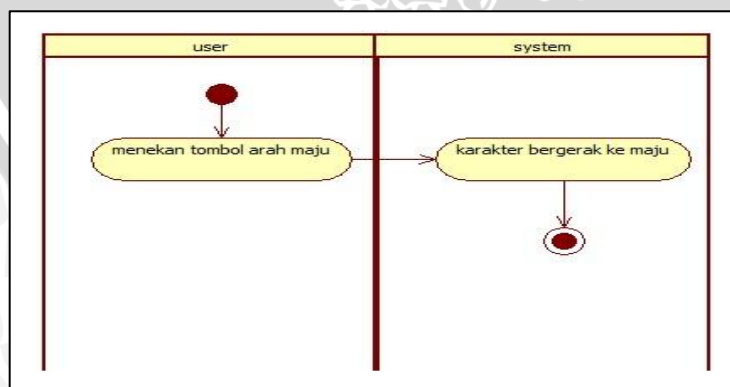
Sumber : Perancangan

4.2.3.5 Activity Diagram Bergerak ke Kanan

Activity diagram bergerak ke kanan menunjukkan aktifitas *player* ketika berada pada halaman permainan. *Player* menggerakkan karakter menggunakan kontroler yang disediakan dengan menekan tombol arah kanan, yang kemudian ditanggapi oleh sistem yang menghasilkan karakter yang bergerak ke arah kanan. Activity diagram bergerak ke kanan ditunjukkan pada Gambar 4.11.

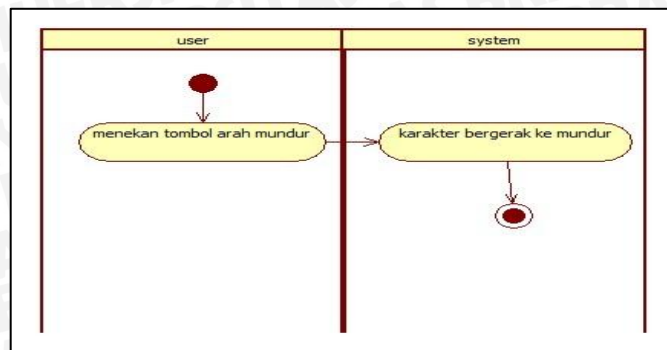
4.2.3.6 Activity Diagram Bergerak Maju

Activity diagram bergerak maju menunjukkan aktifitas *player* ketika berada pada halaman permainan. *Player* menggerakkan karakter menggunakan kontroler yang disediakan dengan menekan tombol arah depan (maju), yang kemudian ditanggapi oleh sistem yang menghasilkan karakter yang bergerak maju. Activity diagram bergerak maju ditunjukkan pada Gambar 4.12.



Gambar 4. 12 Activity Diagram Bergerak Maju

Sumber : Perancangan



Gambar 4. 13 Activity Diagram Bergerak Mundur

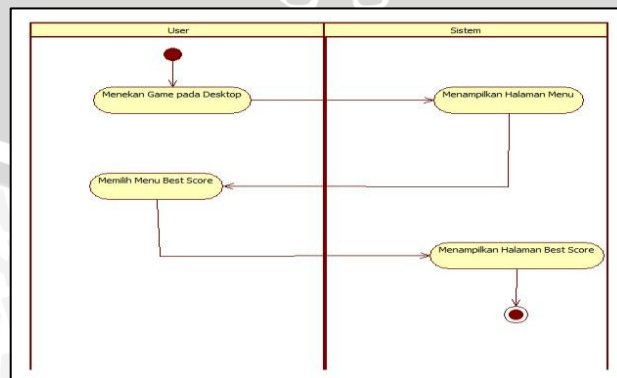
Sumber : Perancangan

4.2.3.7 Activity Diagram Bergerak Mundur

Activity diagram bergerak mundur menunjukkan aktifitas *player* ketika berada pada halaman permainan. *Player* menggerakkan karakter menggunakan kontroler yang disediakan dengan menekan tombol arah belakang (mundur), yang kemudian ditanggapi oleh sistem yang menghasilkan karakter yang bergerak mundur. Activity diagram bergerak mundur ditunjukkan pada Gambar 4.13.

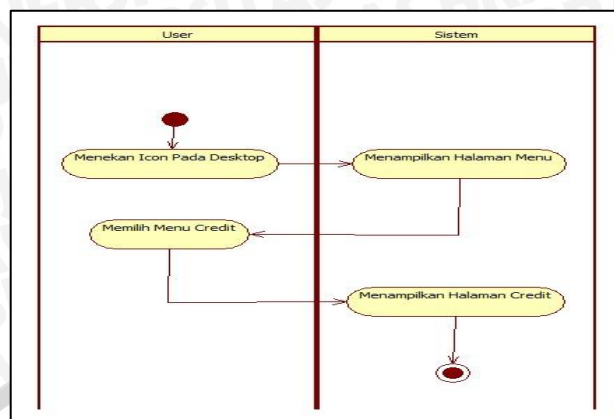
4.2.3.8 Activity Diagram Best Score

Activity diagram *Best Score* menunjukkan aktifitas *player* ketika berada pada halaman menu, *player* memilih menu *Best Score*, yang kemudian ditanggapi oleh sistem dengan menampilkan halaman *Best Score*, yaitu halaman yang menampilkan data *record score* terbaik yang pernah dicatat oleh sistem. Activity diagram *Best Score* ditunjukkan pada Gambar 4.14.



Gambar 4. 14 Activity Diagram *Best Score*

Sumber : Perancangan



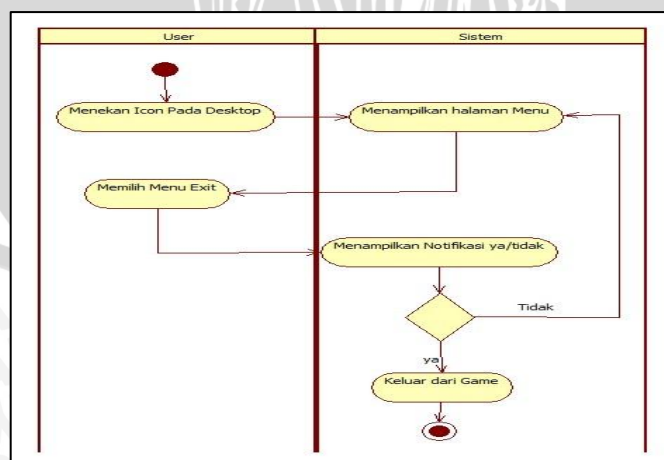
Gambar 4. 15 Activity Diagram Credit
Sumber : Perancangan

4.2.3.9 Activity Diagram Credit

Activity diagram Credit menunjukkan aktifitas *player* ketika berada pada halaman menu, *player* memilih menu *Credit*, yang kemudian ditanggapi oleh sistem dengan menampilkan halaman *Credit*. Activity diagram Credit ditunjukkan pada Gambar 4.15.

4.2.3.10 Activity Diagram Exit

Activity diagram Exit menunjukkan aktifitas *player* ketika berada pada halaman menu, *player* memilih menu *Exit*, yang kemudian ditanggapi oleh sistem dengan keluar dari permainan. Activity diagram Exit ditunjukkan pada Gambar 4.16.



Gambar 4. 16 Activity Diagram Exit
Sumber : Perancangan

BAB V

IMPLEMENTASI

Pada bab ini dibahas mengenai implementasi perangkat lunak berdasarkan hasil yang telah didapatkan dari analisis kebutuhan dan proses perancangan perangkat lunak dibuat. Pembahasan terdiri dari penjelasan tentang spesifikasi sistem, batasan-batasan dalam implementasi, implementasi algoritma, dan implementasi *artwork game* Gobak Sodor 3D dan implementasi antarmuka *game* Gobak Sodor 3D.

5.1 Spesifikasi Sistem

Perangkat lunak *game* Gobak Sodor 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 perangkat lunak permainan Perang Kemerdekaan Indonesia adalah sebagai berikut.

1. Intel(R) Core(TM) i5-2430M CPU @ 2.40GHz (4 CPUs), ~2.4GHz
2. Memory storage 750 GB
3. RAM memory 8 GB
4. Graphic Card :
 - Intel(R) HD Graphics Family (on board)
 - Nvidia GeForce GT 540M (External)
5. Monitor Generic PnP Monitor Widescreen Display
6. Sound Device Speakers (Realtek High Definition Audio)

5.1.2 Spesifikasi Perangkat Lunak

Pengembangan aplikasi permainan Gobak Sodor 3D ini, menggunakan komputer dengan spesifikasi perangkat lunak sebagai berikut :

1. Sistem operasi Microsoft Windows 7 64 bit, home edition.

2. *Game engine* jMonkeyEngine SDK 3.0 Beta
3. Blender 3D versi 2.63
4. Ogre exporter versi 2.49

5.2 Batasan-Batasan Implementasi

Beberapa batasan dalam mengimplementasikan sistem adalah sebagai berikut:

1. Pengembangan perangkat lunak menggunakan *game* jMonkey SDK 3.0 beta.
2. *Library* yang digunakan merupakan *library* yang sudah tersedia pada jMonkey *game engine*.
3. Pembuatan karakter, animasi, *environment* dan level menggunakan Blender 3D.
4. File 3D yang digunakan untuk *game* Gobak Sodor 3D menggunakan format *.j3o.
5. Konfigurasi *general user interface* menggunakan xml.

5.3 Implementasi Artwork

Pada sub bab ini akan di tampilkan hasil implementasi *artwork* dari *game* Gobak Sodor 3D, yang sudah di rancang pada bab sebelum nya. Hasil implementasi *game cover* dapat dilihat pada Gambar 5.1, hasil implementasi karakter dapat dilihat pada Gambar 5.2, dan implementasi *environment* dapat dilihat pada Gambar 5.3.

5.3.1 Implementasi Game Cover

Pada bagian ini akan menjelaskan tentang hasil implementasi *cover game* Gobak Sodor 3D, yang telah di rancang pada bab sebelumnya. Hasil implementasi dapat dilihat pada Gambar 5.1.



Gambar 5. 1 Implementasi Halaman cover game Gobak Sodor 3D

Sumber : Implementasi



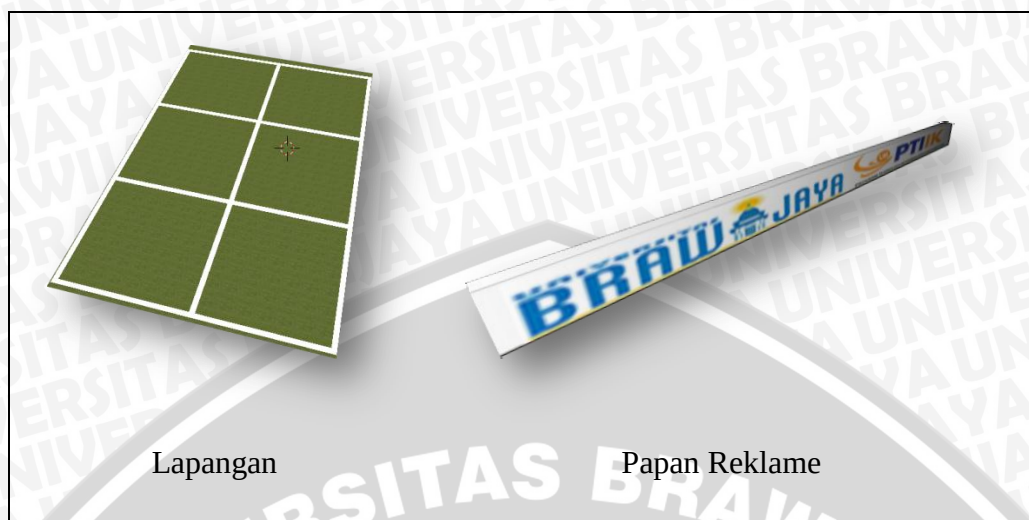
Gambar 5. 2 Implementasi karakter *game* Gobak Sodor 3D
Sumber : Implementasi

5.3.2 Implementasi Karakter

Pada bagian ini akan menampilkan hasil implementasi karakter yang sudah di rancang pada bab sebelumnya, antara lain karakter Baldy, Doni, Jacob, Koko, Rini, Rocky dan Spiky. Hasil implementasi dapat dilihat pada Gambar 5.2.

5.3.3 Implementasi *Environment*

Pada bagian ini akan menampilkan hasil implementasi rancangan *environment* yang ada pada *game* Gobak Sodor 3D, yang telah di rancang pada bab sebelumnya. hasil implementasi dapat dilihat pada Gambar 5.3.



Gambar 5. 3 Implementasi *Environtment game* Gobak Sodor 3D
Sumber : Implementasi

5.4 Implementasi Level

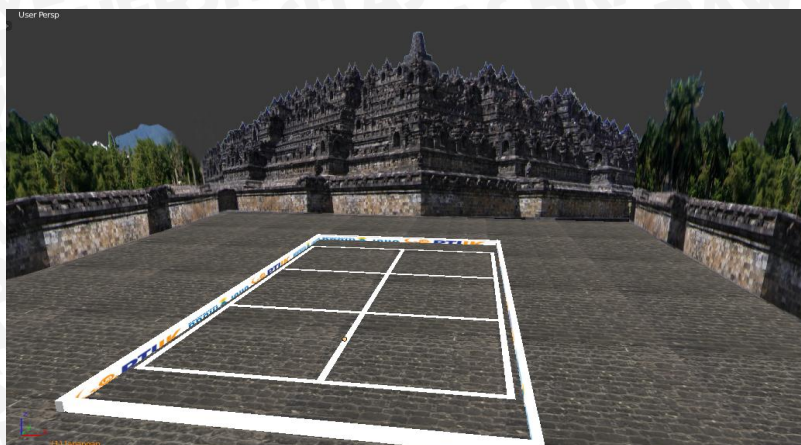
Pada bagian ini akan menampilkan hasil implementasi level yang tersedia pada *game* Gobak Sodor 3D, yang telah dirancang pada bab sebelumnya. hasil implementasi level 1 dapat dilihat pada Gambar 5.4, hasil implementasi level 2 dapat dilihat pada Gambar 5.5, dan hasil implementasi level 3 dapat dilihat pada Gambar 5.6.

5.4.1 Level 1

Implementasi level 1 *game* Gobak Sodor 3D. Level ini memiliki latar belakang gedung rektorat Universitas Brawijaya. Implementasi level 1 dapat dilihat pada Gambar 5.4



Gambar 5. 4 Implementasi level 1
Sumber : Implementasi



Gambar 5. 5 Implementasi level 2
Sumber : Implementasi



Gambar 5. 6 Implementasi level 3
Sumber : Implementasi

5.4.2 Level 2

Implementasi level 2 *game* Gobak Sodor 3D. Level ini memiliki latar belakang Candi Borobudur. Implementasi level 2 dapat dilihat pada Gambar 5.5.

5.4.3 Level 3

Implementasi level 3 *game* Gobak Sodor 3D. Level ini memiliki latar belakang Tugu Monas. Implementasi level 3 dapat dilihat pada Gambar 5.6.

5.5 Implementasi Algoritma

Perangkat lunak *game* Gobak Sodor 3D memiliki beberapa proses atau *method* yang terdapat pada beberapa *class*. Pada penulisan skripsi ini hanya

algoritma dari beberapa proses (operasi) saja sehingga tidak semua *method* akan dicantumkan. Implementasi algoritma ini akan direpresentasikan dalam bentuk *pseudocode* dari algoritma – algoritma tersebut.

5.5.1 Impelementasi Algoritma *onAction()*

Algoritma *onAction* adalah algoritma yang melakukan Proses bergerak pada *game* Gobak Sodor 3D. Proses bergerak dilakukan saat *user* berada pada halaman permainan. Ketika *user* menekan kontrol pada keyboard yaitu arah panah atas, arah panah bawah, arah panah kiri dan arah panah kanan, maka karakter bergerak sesuai masukan dari *user*. implementasi algoritma proses bergerak akan dijelaskan pada Tabel 5.1.

Tabel 5. 1 Algoritma *onAction()*

Elemen	Keterangan
Nama Algoritma:	<i>onAction()</i>
Deskripsi:	• Masukan : arah panah atas, bawah, kiri dan kanan • Proses : 1. Mapping input ke method <i>setupInputs</i> 2. Memanggil method <i>setupInputs</i> 3. Pada method <i>update</i> melakukan update arah gerak karakter. 4. jika maju (arah panah atas) a. maka arah gerak = arah depan dari kamera 5. jika mundur (arah panah bawah) a. maka arah gerak = arah belakang dari kamera 6. jika kiri (arah panah kiri) a. maka arah gerak = arah kiri dari kamera 7. jika kanan (arah panah kanan) a. maka arah gerak = arah kanan dari kamera 8. keluaran : karakter bergerak

Sumber: Implementasi

5.5.2 Impelementasi Algoritma *decreaseLife()*

Pada *game* Gobak Sodor 3D karakter *player* utama memiliki default sebanyak tiga nyawa. Setiap karakter *player* utama bertabrakan dengan karakter

NPC, maka secara otomatis nyawa karakter akan berkurang 1, dan permainan akan berakhir ketika nyawa karakter habis. Implementasi algoritma *decreaseLife* dapat dilihat pada Tabel 5.2.

Tabel 5. 2 Algoritma *decreaseLife*()

Elemen	Keterangan
Nama Algoritma:	<code>decreaseLife()</code>
Deskripsi:	• Masukan : boolean <code>decreaseLife</code> • Proses : 1. Jika (<code>count = true</code>) a) <code>life -1</code> b) <code>count = false</code> c) ambil millisecond pada current d) <code>return = true</code> 2. <code>return false</code> 3. Keluaran-

Sumber: Implementasi

5.5.3 Impelementasi Algoritma *finish*()

Algoritma *finish* berguna untuk menampilkan status permainan yang terdiri dari catatan skor, dan status *game over* ketika nyawa karakter habis. Implementasi algoritma *finsh* dapat dilihat pada Tabel 5.3.

Tabel 5. 3 Algoritma *finish*()

Elemen	Keterangan
Nama Algoritma:	<code>finish()</code>
Deskripsi:	• Proses : 1. Jika menang a) mengambil catatan skor terbaik b) mengambil waktu c) mengkonfigurasi pesan d) mengkalkulasi skor e) jika skor terbaik < skor total - menampilkan data record skor terbaik baru f) jika skor terbaik > skor total - menampilkan data record skor 2. Jika tidak menang a) menampilkan screen <i>game over</i> 3. keluaran-

Sumber: Implementasi

5.5.4 Implelementasi Algoritma *collision()*

Algoritma *collision()* adalah algoritma yang mendeteksi tabrakan antara karakter *player* utama dengan NPC maupun *checkpoint*. Pada *game* Gobak Sodor 3D terdapat dua *checkpoint* yaitu pada lokasi *start*, dan pada seberang arena. *Checkpoint* ini selain digunakan untuk merubah sudut pandang kamera ketika karakter berhasil menyeberangi arena juga digunakan untuk menentukan finish dari karakter ketika karakter kembali ke lokasi awal. Namun jika karakter bertabrakan dengan NPC maka nyawa karakter akan berkurang dan karakter kembali ke titik awal permainan. Implementasi algoritma *collision()* akan dijelaskan pada Tabel 5.4.

Tabel 5. 4 Algoritma *collision()*

Elemen	Keterangan
Nama Algoritma:	<i>collision()</i>
Deskripsi:	• Masukan : Gerak <i>player</i> utama • Proses : 1. Instansiasi class <i>Character</i> untuk mendapatkan karakter <i>player</i>. 2. Instansiasi class <i>Enemy</i> untuk mendapatkan karakter NPC 3. Jika (node B = <i>Player</i>) tabrakan dengan a. <i>frontcam</i> b. Maka kamera belakang tidak aktif c. melewati <i>checkpoint</i> 4. Namun jika (Node A = <i>backcam</i>) a. Maka kamera belakang aktif b. Jika posisi = 1 c. Maka Finish 5. Jika (node A = <i>player</i>) bertabrakan dengan a. jika (node B = <i>enemy</i>) i. jika <i>decreaseLife()</i> 1. posisi = 0 2. karakter kembali ke lokasi <i>start</i> 6. Jika <i>decreaseLife</i> benar a. Set posisi ke karakter = posisi awal 7. Jika <i>health</i> = 0 a. memanggil method <i>win</i> (<i>false</i>) 8. Keluaran-

Sumber: Implementasi

5.6 Implementasi Antarmuka

Pada bagian ini akan membahas tentang implementasi antarmuka *game* Gobak Sodor 3D. Implementasi antarmuka berupa halaman menu, halaman pilih karakter dan halaman permainan.

5.6.1 Antarmuka Menu

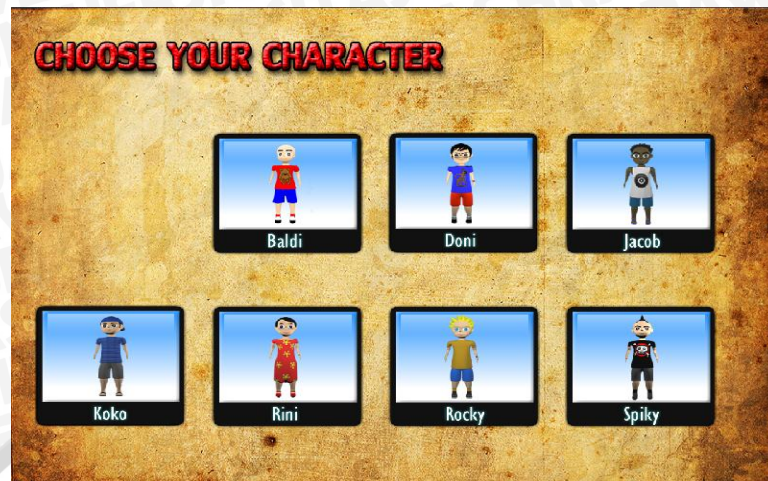
Halaman menu ditampilkan pertama kali ketika *user* memasuki permainan Gobak Sodor 3D. Halaman menu merupakan halaman yang menampilkan sejumlah menu utama untuk memudahkan *user* mengakses permainan, baik memulai permainan, maupun melihat informasi tentang permainan. menu-menu yang terdapat pada halaman menu utama adalah menu *Start*, menu *Best Score*, Menu *Credit*, dan menu *Exit*. Implementasi antarmuka halaman menu utama dapat dilihat pada Gambar 5.7.

5.6.2 Antarmuka Pilih Karakter

Halaman pilih karakter merupakan halaman yang akan muncul setelah *user* memilih menu *start*. Pada halaman ini, *user* dapat memilih karakter yang akan menjadi karakter *player* utama dalam permainan yang akan dimainkan. Implementasi halaman pilih karakter dapat dilihat pada Gambar 5.8.



Gambar 5. 7 Implementasi antarmuka menu
Sumber : Implementasi



Gambar 5. 8 Implementasi antarmuka pilih karakter
Sumber : Implementasi

5.6.3 Antarmuka Permainan

Halaman permainan adalah halaman yang akan ditampilkan setelah *user* memilih karakter. Pada halaman ini *user* dapat mulai memainkan permainan dengan menggerakkan karakter *player* utama menggunakan kontrol keyboard. Implementasi halaman permainan dapat dilihat pada Gambar 5.9.



Gambar 5. 9 Implementasi antarmuka permainan
Sumber : Implementasi

BAB VI

PENGUJIAN DAN ANALISIS

Bab ini membahas mengenai tahapan pengujian dan analisis perangkat lunak *Game Gobak Sodor 3D* yang telah dikembangkan. Proses pengujian dilakukan melalui empat tahapan yaitu pengujian unit, pengujian integrasi, pengujian validasi, dan pengujian performa. Pada pengujian unit dan integrasi, akan digunakan teknik pengujian *White-Box (White-Box Testing)*. Pada pengujian validasi akan digunakan teknik pengujian *Black-Box (Black-Box Testing)*. Pada pengujian performa akan digunakan analisis FPS (*Frame Per Second*) untuk mengetahui performa *game Gobak Sodor 3D*.

6.1 Pengujian

Proses pengujian yang dilakukan melalui empat tahapan yaitu pengujian unit, pengujian integrasi, pengujian validasi, dan pengujian performa.

6.1.1 Pengujian Unit

Pada pengujian unit digunakan metode *White-Box Testing* dengan teknik *Basis-Path Testing*. Pada teknik *Basis-Path Testing* proses pengujian dilakukan dengan memodelkan algoritma pada sebuah *flow graph*, menentukan *cyclometric complexity* dan melakukan uji kasus untuk setiap *path* yang ada.

6.1.1.1 Pengujian Unit Algoritma *onAction()*

Algoritma *onAction* adalah sebuah algoritma yang mengatur gerakan karakter *player* utama. Pengujian algoritma *onAction* dapat dilihat pada Tabel 6.1.

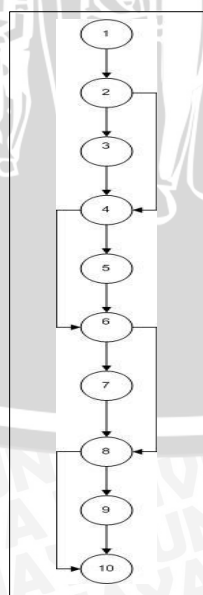
Tabel 6. 1 Pengujian Algoritma *onAction()*

Elemen	Keterangan
Nama Algoritma:	<code>onAction()</code>

Deskripsi:		• Masukan : arah panah atas, bawah, kiri dan kanan
	1	• Proses : 1. Mapping input ke method setupInputs 2. memanggil method setupInputs 3. pada method update melakukan update arah gerak karakter.
	2	4. jika maju (arah panah atas)
	3	a) maka arah gerak = arah depan dari kamera
	4	5. jika mundur (arah panah bawah)
	5	a) maka arah gerak = arah belakang dari kamera
	6	6. jika kiri (arah panah kiri)
	7	a) maka arah gerak = arah kiri dari kamera
	8	7. jika kanan (arah panah kanan)
	9	a) maka arah gerak = arah kanan dari kamera
	10	8. keluaran : karakter bergerak

Sumber : Pengujian

Dari Algoritma tersebut didapatkan *flowgraph* seperti pada Gambar 6.1 berikut.



Gambar 6. 1 *flowgraph* Algoritma *onAction()*

Sumber : Pengujian

Pemodelan ke dalam *flow graph* 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*).

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 13 - 10 + 2 \\ &= 5 \end{aligned}$$

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

Jalur 1 : 1 – 2 – 4 – 6 – 8 – 10

Jalur 2 : 1 – 2 – 4 – 5 – 6 – 7 – 8 – 9 – 10

Jalur 3 : 1 – 2 – 3 – 4 – 6 – 7 – 8 – 9 – 10

Jalur 4 : 1 – 2 – 3 – 4 – 5 – 6 – 8 – 9 – 10

Jalur 5 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10

Tabel 6.2 berisi test case yang dilakukan sesuai jalur yang telah ditentukan. Kasus uji merupakan keadaan yang dilakukan, hasil yang diharapkan merupakan hasil yang seharusnya terjadi, dan hasil yang didapatkan merupakan hasil setelah melakukan kasus uji.

Tabel 6. 2 Kasus uji algoritma *onAction()*

No	Kasus uji	Hasil yang Diharapkan	Hasil yang didapatkan
1.	Jika tidak memberikan input / menekan selain arah	Karakter tidak bergerak.	Karakter tidak bergerak
2.	Memberikan inputan arah panah atas	Karakter bergerak sesuai inputan arah panah atas (maju)	Karakter bergerak sesuai inputan arah atas (maju)
3.	Memberikan inputan arah panah bawah	Karakter bergerak sesuai inputan arah panah bawah (mundur)	Karakter bergerak sesuai inputan arah panah bawah (mundur)
4.	Memberikan inputan arah panah kiri	Karakter bergerak sesuai inputan arah panah kiri	Karakter bergerak sesuai inputan arah panah kiri
5.	Memberikan inputan arah panah kanan	Karakter bergerak sesuai inputan arah panah kanan	Karakter bergerak sesuai inputan arah panah kanan

Sumber: Pengujian

6.1.1.2 Pengujian Unit Algoritma *finish* ()

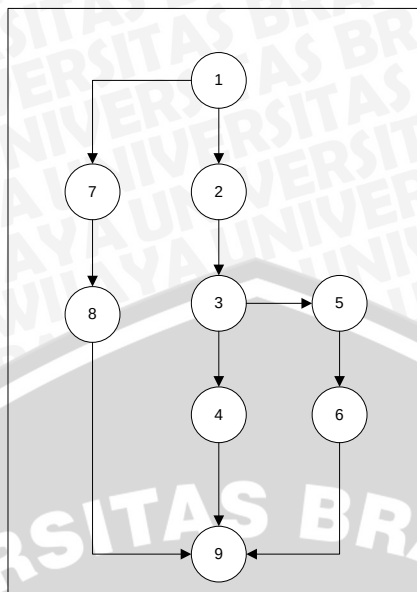
Algoritma *finish*() adalah *method* dari *class main* yang menunjukkan kondisi ketika memenangkan permainan. Pengujian algoritma *finish*() dapat dilihat pada Tabel 6.3.

Tabel 6. 3 Pengujian unit algoritma *finish*()

Elemen	Keterangan
Nama Algoritma:	method <i>finish</i> ()
Deskripsi:	• Proses :
1	1. Jika menang
2	a) mengambil catatan skor terbaik b) mengambil waktu c) mengkonfigurasi pesan d) mengkalkulasi skor
3	e) jika skor terbaik < skor total
4	- menampilkan data record skor terbaik baru
5	f) jika skor terbaik > skor total
6	- menampilkan data record skor
7	2. Jika tidak menang
8	a) menampilkan screen <i>game over</i>
9	3. Keluaran -

Sumber : Pengujian

Dari Algoritma tersebut didapatkan *flowgraph* seperti pada Gambar 6.2 berikut.



Gambar 6. 2 *flowgraph* Algoritma *finish()*
Sumber: Pengujian

Pemodelan ke dalam *flow graph* 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*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 10 - 9 + 2 \\
 &= 3
 \end{aligned}$$

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 – 9

Jalur 2 : 1 – 2 – 3 – 5 – 6 – 9

Jalur 3 : 1 – 7 – 8 – 9

Tabel 6. 4 Kasus uji algoritma *finish ()*

No	Kasus uji	Hasil yang diharapkan	Hasil yang didapatkan
1.	Jika menang dengan skor terbaik	Menampilkan data record skor terbaik baru	Menampilkan data record skor terbaik baru
2.	Jika menang	Menampilkan data record skor	Menampilkan data record skor
3.	Jika tidak menang	Menampilkan screen <i>game over</i>	Menampilkan screen <i>game over</i>

Sumber: Pengujian

6.1.1.3 Pengujian Unit Algoritma *decreaseLife* ()

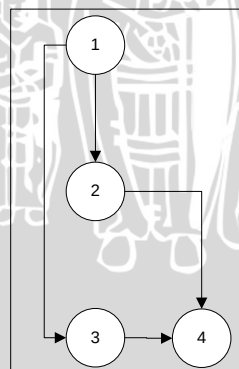
Algoritma *decreaseLife*() merupakan method pada class *Life* yang menunjukkan kondisi ketika nyawa karakter *player* berkurang setelah terjadi tabrakan dengan karakter NPC. Pengujian unit algoritma *decreaseLife*() dapat dilihat pada Tabel 6.5.

Tabel 6. 5 Pengujian unit algoritma *decreaseLife* ()

Elemen	Keterangan
Nama Algoritma:	<i>decreaseLife</i> ()
Deskripsi:	Proses :
1	1. Jika (count = true)
2	a) life -1 b) count = false c) ambil millisecond pada current d) return = true
3	2. return false
4	3. Keluaran-

Sumber: Pengujian

Dari Algoritma tersebut didapatkan *flowgraph* seperti pada Gambar 6.3 berikut.



Gambar 6. 3 *flowgraph* Algoritma method *decreaseLife*()

Sumber: Pengujian

Pemodelan ke dalam *flow graph* 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*).

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

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

Jalur 1 : 1 – 2 – 4

Jalur 2 : 1 – 3 – 4

Tabel 6. 6 Kasus uji algoritma *decreaseLife* ()

No	Kasus Uji	Hasil yang diharapkan	Hasil yang di dapatkan
1.	Jika count bernilai true	Nyawa berkurang 1	Nyawa berkurang satu
2.	Jika count bernilai false	Nyawa tetap	Nyawa tetap

Sumber: Pengujian

6.1.2 Pengujian Integrasi

Pengujian integrasi memastikan bahwa dua atau lebih unit atau integrasi lainnya dapat bekerja bersama dengan baik dan agar lebih fokus pada antarmuka yang terspesifikasi pada *low-level design* [SRJ-12].

6.1.2.1 Pengujian Integrasi Algoritma *collision*()

Algoritma *collision* terjadi ketika karakter player utama bertabrakan dengan *checkpoint*, maupun NPC. Pengujian integrasi algoritma *collision*() dapat dilihat pada Tabel 6.7

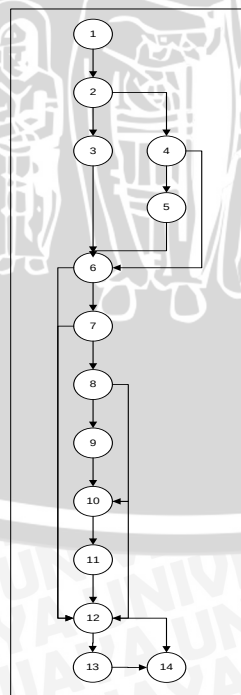
Tabel 6. 7 Kasus uji algoritma *collision* ()

Elemen	Keterangan
Nama Algoritma:	<i>collision</i> ()
Deskripsi:	Masukan : Gerak <i>player</i> utama
1	1. Instansiasi class <i>Character</i> untuk mendapatkan karakter <i>player</i> . 2. Instansiasi class <i>Enemy</i> untuk mendapatkan karakter NPC.
2	3. Jika (node B = <i>player</i>) tabrakan dengan
3	a) <i>frontcam</i> b) maka kamera belakang tidak aktif c) melewati <i>checkpoint</i>

4	4. Namun jika (Node A = backcam)
5	a) maka kamera belakang aktif b) jika posisi = 1 c) maka finish
6	5. Jika (node A = <i>player</i>) bertabrakan dengan
7	a) jika (node B = enemy)
8	i. jika decreaseLife()
9	1. posisi = 0 2. karakter kembali ke lokasi start
10	6. Jika decreaseLife benar
11	a) set posisi karakter = posisi awal
12	7. Jika health = 0
13	a) memanggil method win (false)
14	8. Keluaran

Sumber : Pengujian

Dari Algoritma tersebut didapatkan *flowgraph* seperti pada Gambar 6.4 berikut.



Gambar 6.4 *Flowgraph* Algoritma *collision()*

Sumber: Pengujian

Pemodelan ke dalam *flow graph* 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*).

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 20 - 14 + 2 \\ &= 8 \end{aligned}$$

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

Jalur 1 : 1 – 2 – 3 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 14

Jalur 2 : 1 – 2 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 14

Jalur 3 : 1 – 2 – 4 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 14

Jalur 4 : 1 – 2 – 3 – 6 – 12 – 14

Jalur 5 : 1 – 2 – 3 – 6 – 7 – 12 – 14

Jalur 6 : 1 – 2 – 3 – 6 – 7 – 8 – 10 – 11 – 12 – 13 – 14

Jalur 7 : 1 – 2 – 3 – 6 – 7 – 8 – 9 – 10 – 12 – 14

Jalur 8 : 1 – 2 – 3 – 6 – 7 – 8 – 9 – 10 – 12 – 14

Tabel 6. 8 Kasus uji algoritma *Collision()*

No	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1.	Jika karakter bertabrakan dengan frontcam atau dengan NPC	Jika bertabrakan dengan frontcam maka kamera berubah, tapi jika bertabrakan dengan NPC maka nyawa berkurang dan kembali ke start.	Jika bertabrakan dengan frontcam maka kamera berubah, tapi jika bertabrakan dengan NPC maka nyawa berkurang dan kembali ke start.
2.	Jika karakter bertabrakan dengan backcam atau dengan NPC	Jika bertabrakan dengan backcam maka kamera berubah dan jika bernilai = 1 maka memenangkan permainan, tapi jika bertabrakan dengan NPC maka nyawa berkurang dan kembali ke start.	Jika bertabrakan dengan backcam maka kamera berubah dan jika bernilai = 1 maka memenangkan permainan, tapi jika bertabrakan dengan NPC maka nyawa berkurang dan kembali ke start.
3.	Jika karakter bertabrakan dengan backcam atau dengan NPC	Jika bertabrakan dengan backcam maka kamera berubah , tapi jika bertabrakan dengan NPC maka nyawa berkurang dan	Jika bertabrakan dengan backcam maka kamera berubah , tapi jika bertabrakan dengan NPC maka nyawa berkurang dan

		kembali ke start.	kembali ke start.
4.	Jika karakter melewati checkpoint 1	Permainan berlanjut dan kamera berubah.	Permainan berlanjut dan kamera berubah.
5.	Jika karakter bertabrakan NPC setelah melewati checkpoint 1	Nyawa karakter berkurang 1 dan permainan berlanjut.	Nyawa karakter berkurang 1 dan permainan berlanjut.
6.	Jika karakter bertabrakan NPC setelah melewati checkpoint 1 dan decrease life bernilai true	Nyawa karakter berkurang 1, karakter kembali ke posisi awal dan jika nyawa = 0 maka <i>game over</i> .	Nyawa karakter berkurang 1, karakter kembali ke posisi awal dan jika nyawa = 0 maka <i>game over</i> .
7.	Jika karakter bertabrakan NPC setelah melewati checkpoint 1 dan decrease life bernilai true	Nyawa berkurang 1 dan permainan berlanjut.	Nyawa berkurang 1 dan permainan berlanjut.
8.	Jika karakter bertabrakan NPC setelah melewati checkpoint 1 dan decrease life bernilai true	Nyawa berkurang 1 dan permainan berlanjut.	Nyawa berkurang 1 dan permainan berlanjut.

Sumber : Pengujian

6.1.3 Pengujian Validasi

Pengujian validasi digunakan 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 *Black Box*, karena tidak diperlukan konsentrasi terhadap alur jalannya algoritma program dan lebih ditekankan untuk menemukan konformitas antara kinerja sistem dengan *usecase*

diagram. Pada skripsi ini dilakukan pengujian validasi terhadap *game* Gobak Sodor 3D.

6.1.3.1 Kasus Uji Start Game

Pada Tabel 6.9 ditunjukkan kasus uji untuk validasi *start game* pada menu *game* Gobak Sodor 3D.

Tabel 6. 9 Kasus uji untuk pengujian validasi *start game*

Nama Kasus Uji	Start Game
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan <i>start game</i> .
Prosedur Uji	1. Sistem dijalankan saat program dieksekusi. 2. <i>Player</i> masuk ke halaman menu <i>game</i> 3. <i>Player</i> menekan tombol <i>start game</i>.
Hasil yang diharapkan	Aplikasi akan menampilkan halaman pilih karakter.
Hasil yang didapatkan	Valid

Sumber : Pengujian

6.1.3.2 Kasus Uji Pilih Karakter

Pada Tabel 6.10 ditunjukkan kasus uji untuk validasi Pilih Karakter pada menu *game* Gobak Sodor 3D.

Tabel 6. 10 Kasus uji untuk pengujian validasi *start game*

Nama Kasus Uji	Pilih karakter
Tujuan pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan pilih karakter.
Prosedur Uji	1. <i>Player</i> menekan icon start pada menu <i>game</i>. 2. <i>Player</i> masuk ke halaman Pilih karakter 3. <i>Player</i> menekan icon karakter yang dipilih
Hasil yang diharapkan	Aplikasi akan menampilkan halaman permainan dengan karakter <i>player</i> utama sesuai dengan karakter yang dipilih pada menu pilih karakter.

Hasil yang didapatkan	Valid
-----------------------	-------

Sumber : Pengujian

6.1.3.3 Kasus Uji Memulai Permainan

Pada Tabel 6.11 ditunjukkan kasus uji untuk validasi Memulai Permainan pada *game* Gobak Sodor 3D.

Tabel 6. 11 Kasus uji untuk pengujian memulai permainan

Nama kasus uji	Memulai Permainan
Tujuan pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan memulai permainan.
Prosedur Uji	1. <i>Player</i> memilih karakter yang akan dimainkan pada halaman pilih karakter. 2. <i>Player</i> masuk ke halaman pilih karakter. 3. <i>Player</i> menekan spasi untuk mengakhiri sinematik. 4. <i>Player</i> menekan kontrol arah panah pada keyboard untuk menggerakkan karakter.
Hasil yang diharapkan	Karakter Bergerak berdasarkan input masukan kontrol arah panah pada keyboard.
Hasil yang didapatkan	Valid

Sumber: Pengujian

6.1.3.4 Kasus Uji Kontrol Gerak ke Kiri

Pada Tabel 6.12 ditunjukkan kasus uji untuk validasi Kontrol bergerak ke kiri pada *game* Gobak Sodor 3D.

Tabel 6. 12 Kasus uji untuk pengujian kontrol gerak ke kiri

Nama kasus uji	Kontrol gerak ke kiri
Tujuan pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan kontrol gerak ke kiri.
Prosedur uji	1. <i>Player</i> menekan spasi untuk mengakhiri

	sinematik dan memulai permainan. 2. <i>Player</i> menekan arah panah kiri pada keyboard.
Hasil yang diharapkan	Karakter yang dikendalikan <i>Player</i> bergerak ke arah kiri.
Hasil yang didapatkan	Valid

Sumber: Pengujian

6.1.3.5 Kasus Uji Kontrol Gerak ke Kanan

Pada Tabel 6.13 ditunjukkan kasus uji untuk validasi Kontrol bergerak ke kanan pada *game* Gobak Sodor 3D.

Tabel 6. 13 Kasus uji untuk pengujian kontrol gerak ke kanan

Nama kasus uji	Kontrol Gerak ke Kanan
Tujuan pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan kontrol gerak ke kanan.
Prosedur Uji	1. <i>Player</i> menekan spasi untuk mengakhiri sinematik dan memulai permainan. 2. <i>Player</i> menekan arah panah kanan pada keyboard.
Hasil yang diharapkan	Karakter yang dikendalikan <i>Player</i> bergerak ke arah kanan.
Hasil yang didapatkan	Valid

Sumber : Pengujian

6.1.3.6 Kasus Uji Kontrol Gerak Maju

Pada Tabel 6.14 ditunjukkan kasus uji untuk validasi Kontrol bergerak maju pada *game* Gobak Sodor 3D.

Tabel 6. 14 Kasus uji untuk pengujian kontrol gerak maju

Nama kasus uji	Kontrol Gerak Maju
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa

	aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan kontrol gerak maju.
Prosedur Uji	1. <i>Player</i> menekan spasi untuk mengakhiri sinematik dan memulai permainan. 2. <i>Player</i> menekan arah panah atas pada keyboard.
Hasil yang diharapkan	Karakter yang dikendalikan <i>Player</i> bergerak maju.
Hail yang didapatkan	Valid

Sumber : Pengujian

6.1.3.7 Kasus Uji Kontrol Gerak Mundur

Pada Tabel 6.15 ditunjukkan kasus uji untuk validasi Kontrol bergerak maju pada *game* Gobak Sodor 3D.

Tabel 6. 15 Kasus uji untuk pengujian kontrol gerak maju

Nama Kasus Uji	Kontrol gerak mundur
Tujuan Peengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan kontrol gerak maju.
Prosedur Uji	1. <i>Player</i> menekan spasi untuk mengakhiri sinematik dan memulai permainan. 2. <i>Player</i> menekan arah panah bawah pada keyboard.
Hasil yang Diharapkan	Karakter yang dikendalikan <i>Player</i> bergerak mundur.
Hasil yang Didapatkan	Valid

Sumber : Pengujian

6.1.3.8 Kasus Uji *Pause Game*

Pada Tabel 6.16 ditunjukkan kasus uji untuk validasi Kontrol bergerak maju pada *game* Gobak Sodor 3D.

Tabel 6. 16 Kasus uji untuk pengujian pause *game*

Nama Kasus Uji	Pause <i>Game</i>
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan kontrol gerak maju.
Prosedur Uji	1. <i>Player</i> menekan spasi untuk mengakhiri sinematik dan memulai permainan. 2. <i>Player</i> menekan tombol “P” pada keyboard untuk pause <i>game</i>.
Hasil yang diharapkan	Permainan berhenti sementara, dan menampilkan screen pause <i>game</i> .
Hasil yang didapatkan	Valid

Sumber : Pengujian

6.1.3.9 Kasus Uji Melihat *Best Score*

Pada Tabel 6.17 ditunjukkan kasus uji untuk validasi melihat *Best Score* pada *game* Gobak Sodor 3D.

Tabel 6. 17 Kasus uji untuk pengujian melihat *Best Score*

Nama Kasus Uji	Melihat <i>Best Score</i>
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melihat <i>Best Score</i> .
Prosedur Uji	1. Sistem dijalankan saat program dieksekusi. 2. <i>Player</i> masuk ke halaman menu <i>game</i> 3. <i>Player</i> menekan tombol <i>Best Score</i>.
Hasil yang Diharapkan	Sistem menampilkan halaman <i>Best Score</i> yang pernah dicatat.
Hasil yang Didapatkan	Valid

Sumber : Pengujian

6.1.3.10 Kasus Uji Melihat *Credit*

Pada Tabel 6.18 ditunjukkan kasus uji untuk validasi melihat *Credit* pada *game* Gobak Sodor 3D.

Tabel 6. 18 Kasus uji untuk pengujian melihat *Credit*

Nama Kasus Uji	Melihat <i>Credit</i>
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melihat <i>Credit</i> .
Prosedur Uji	1. Sistem dijalankan saat program dieksekusi. 2. <i>Player</i> masuk ke halaman menu <i>game</i> 3. <i>Player</i> menekan tombol <i>Credit</i>.
Hasil yang Diharapkan	Sistem menampilkan halaman <i>Credit game</i> Gobak Sodor 3D.
Hasil yang Didapatkan	Valid

Sumber : Pengujian

6.1.3.11 Kasus Uji Exit *Game*

Pada Tabel 6.19 ditunjukkan kasus uji untuk validasi melihat *Credit* pada *game* Gobak Sodor 3D.

Tabel 6. 19 Kasus uji untuk pengujian Exit *game*

Nama Kasus Uji	Exit <i>Game</i>
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk Exit <i>game</i> .
Prosedur Uji	1. Sistem dijalankan saat program dieksekusi. 2. <i>Player</i> masuk ke halaman menu <i>game</i> 3. <i>Player</i> menekan tombol Exit.
Hasil yang Diharapkan	Sistem akan tertutup.
Hasil yang didapatkan	Valid

6.1.4 Pengujian Performa

Pengujian performa dilakukan untuk mengetahui bagaimana performa *game* Gobak Sodor 3D saat dimainkan pada beberapa spesifikasi komputer berbeda. Pengujian performa dilakukan dengan cara menganalisa *frame rate* pada saat *game* sedang dijalankan.

6.1.4.1 Pengujian Performa *frame rate*

Pengujian Performa *frame rate* dilakukan dengan cara menganalisa *frame rate* *game* Gobak Sodor 3D yang dijalankan pada 10 komputer dengan spesifikasi yang berbeda. Pengujian Performa Analisa *frame rate* dapat dilihat pada Tabel 6.20.

Tabel 6. 20 Pengujian performa analisa *frame rate*

No	Spesifikasi Komputer	FPS yang dihasilkan	
		(windowed) 1024x768	(Full screen) 1024x768
1.	Core i7, GT 540 M 2Gb, 8 Gb RAM, 750 Gb HDD	88-92	98-108
2.	Core i5, GT 540 M 2Gb, 8 Gb RAM, 750 Gb HDD	124 - 127	129 - 137
3.	Core i5, GT 540 M 2Gb, 4 Gb RAM, 750 Gb HDD	110 – 115	117 - 122
4.	Core i5, GT 540 M 2Gb, 6 Gb RAM, 750 Gb HDD	115 – 120	120 - 125
5.	Core i7, Intel HD Graphic 1,6, 8 Gb RAM, 750 Gb HDD	1-3	1-3
6.	Core i5, Intel HD Graphic 1,6 Gb, 8 Gb RAM, 750 Gb HDD	1-3	1-3
7.	Core i5, Intel HD Graphic 1,6 Gb, 4 Gb RAM, 750 Gb HDD	1-3	1-3
8.	IntelCore2 Duo CPU 2.40GHz, ATI Mobility Radeon HD 4500 2 Gb, 4 Gb RAM	35-37	34-35
9.	Intel Core 2 duo, 2 Gb RAM,	1	1

	Intel 876 Mb, 300 GB HDD		
10.	Intel Core 2 duo 2 Ghz, Mobile intel (R) 465 xpress chipset family 251 Gb, RAM 1 Gb.	0	0

Sumber: Pengujian

6.1.5 Playtesting

Playtesting dilakukan untuk mendapatkan umpan balik dari pemain dan memastikan bahwa *game* berjalan sesuai harapan, seimbang, menarik untuk dimainkan dan pemain merasa puas dengan *gameplay* dari Gobak Sodor 3D. Hasil *playtesting* ditulis ke dalam kuisioner dan dirangkum di dalam tabel yang ditunjukkan pada Tabel 6.21. Umpan balik yang didapatkan dari pemain selama *playtesting* ditunjukkan pada Tabel 6.22. Umpan balik dicatat dalam kuisioner sebagai saran yang ditulis oleh pemain.

Tabel 6. 21 Hasil kuisioner *playtesting*

NO	Komponen Penilaian	Tidak	Ya	Lumayan	Menarik	Mudah
1.	Apakah game Gobak Sodor 3D menarik untuk dimainkan?				10	
2.	Apakah game Gobak Sodor 3D mudah untuk dimainkan?			8		2
3.	Apakah tampilan game Gobak Sodor 3D menarik?			1	9	
4.	Apakah game Gobak Sodor 3D membuat anda ingin memainkan secara nyata permainan Gobak Sodor?	3	7			

Sumber: Pengujian

Tabel 6. 22 Umpan balik

NO	Umpan Balik / Saran
1.	Lebih baik jika di tambah fitur <i>multiplayer</i> .
2.	Ditambah <i>special move</i> .
3.	Ditambahkan penonton di sekitar lapangan.

Sumber: Pengujian

6.2 Analisis

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

6.2.1 Analisis Pengujian Unit

Analisis terhadap hasil pengujian unit yang dilakukan terhadap method *onAction()*, *decreaseLife()* dan *finish()*, dengan melihat kesesuaian fungsi dari implementasi unit yang diuji dengan hasil perancangan *game* yang telah dirancang sebelumnya. Berdasarkan hal tersebut, maka dapat diambil kesimpulan bahwa unit dari *game* sudah memenuhi kebutuhan yang telah dirancang pada tahap perancangan.

6.2.2 Analisis Pengujian Integrasi

Analisis terhadap hasil pengujian integrasi yang dilakukan terhadap method *collision()*, dengan melihat kesesuaian fungsi dari implementasi integrasi yang diuji dengan hasil perancangan *game* yang telah dirancang sebelumnya. Berdasarkan hal tersebut, maka dapat diambil kesimpulan bahwa integrasi dari *game* sudah memenuhi kebutuhan yang telah dirancang pada tahap perancangan.

6.2.3 Analisis Pengujian Validasi

Analisis terhadap hasil pengujian validasi dilakukan dengan melihat antara hasil kinerja sistem dengan daftar kebutuhan. Berdasarkan hasil pengujian validasi dapat disimpulkan bahwa implementasi dan fungsionalitas perangkat lunak *game* Gobak Sodor 3D telah sesuai dengan kebutuhan.

6.2.4 Analisis Pengujian Performa

Analisis terhadap hasil pengujian *Frame Rate* dilakukan dengan melihat performa *game* Gobak Sodor 3D pada saat dimainkan pada beberapa komputer dengan spesifikasi yang berbeda dalam kondisi lingkungan normal. Berdasarkan hal tersebut dapat diambil kesimpulan seperti berikut :

- a. Nilai *frame rate* akan berada diatas batas standart pada komputer atau *laptop* yang menggunakan *VGA external* dengan memory minimal 2Gb. Hal ini akan membuat *game* berjalan dengan sangat halus.
- b. Nilai *frame rate* akan berada jauh dibawah batas standart jika *game* dimainkan pada komputer atau *laptop* yang menggunakan *VGA internal* dengan memory dibawah 2 Gb. Hal ini akan membuat *game* berjalan sangat berat dan patah-patah. Oleh karena itu *game* harus dijalankan pada unit yang memenuhi minimal requirement.
- c. Ketika *game* berjalan pada unit dengan spesifikasi sesuai pada Tabel 6.20 pada nomor delapan, didapatkan hasil FPS yang cukup baik untuk memainkan *game* Gobak Sodor 3D. Dari hasil analisis tersebut dapat diketahui bahwa *minimal requirement game* Gobak Sodor 3D adalah :
 1. Intel core 2 Duo 2.0 GHz (atau yang sebanding).
 2. 2 Gb RAM.
 3. VGA dengan memory minimal 2GB , (32 Bit)

6.2.5 Analisis Hasil Playtesting

Hasil playtesting menunjukkan bahwa *game* Gobak Sodor 3D menarik bagi 10 pemain yang melakukan playtesting. 10 pemain tersebut memilih jawaban “c. menarik” untuk pertanyaan nomor satu. Pada pertanyaan apakah *game* Gobak Sodor 3D mudah untuk dimainkan?, delapan orang menjawab “b. Lumayan” dan dua orang menjawab “c. Mudah”. Pada pertanyaan apakah tampilan *game* Gobak Sodor 3D menarik? satu orang pemain menjawab “b. Lumayan” dan sembilan orang menjawab “c. Menarik”. Pada pertanyaan apakah anda ingin memainkan Gobak Sodor secara nyata?, tiga orang menjawab “a. Tidak” sedangkan tujuh orang pemain menjawab “b. Ya”.

Kesimpulannya game Gobak Sodor menarik, dan cukup mudah untuk dimainkan. Dengan tampilan yang menarik, membuat cukup banyak yang ingin memainkan permainan Gobak Sodor secara nyata.



BAB VII

PENUTUP

7.1 Kesimpulan

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

1. Tahapan Perancangan *game* Gobak Sodor 3D terdiri atas tahapan *Game Design* untuk perancangan dan tahap *technical design* yang berbasis Object Oriented Programing menggunakan bahasa pemodelan UML.
2. *Game* Gobak Sodor 3D berjalan pada *platform PC*. Implementasi dilakukan dengan pembuatan model menggunakan software Blender 3D, dan pemrograman menggunakan game engine jMonkey yang berbasis java.
3. Berdasarkan hasil pengujian validasi menggunakan metode *blackbox testing*, didapatkan keseluruhan fungsional aplikasi permainan dapat berjalan sesuai analisis kebutuhan yang telah dibuat dan dimodelkan dalam *use case*.
4. Berdasarkan hasil pengujian Validasi menggunakan metode *black-box testing*, didapatkan performa yang optimal karena *game* dapat berjalan dengan sempurna sesuai dengan perancangan.
5. Berdasarkan hasil pengujian performa *game* menggunakan metode analisis *frame rate*, didapatkan hasil bahwa Performa optimal ketika *game* berjalan pada komputer atau *laptop* yang memenuhi *minimal requirement*.

7.2 Saran

Saran yang telah diberikan untuk pengembangan *game* Gobak sodor 3D lebih lanjut antara lain:

1. *Game* Gobak Sodor 3D dapat ditambah lebih banyak variasi level dengan latar belakang yang lebih bervariasi.
2. Preferensi setiap karakter di dalam *game* Gobak Sodor 3D dibuat agar mempengaruhi gameplay.

3. *Game* Gobak Sodor 3D dibuat *multiplayer* sehingga lebih menantang.
4. Gameplay ditingkatkan dengan penambahan jumlah penjaga pada setiap peningkatan level, penambahan special move, penambahan penonton di sekitar lapangan dan reward dibuat lebih bervariasi.



DAFTAR PUSTAKA

- [ADA-10] Dawes, Adam. 2010. *Windows Phone 7 Game Development*. Apress
- [ALT-11] Thorn, Alan. 2011. *Game Engine Design and Implementation*. Jones & Bartlett Learning
- [DAV-04] Bourgh, David M, Glenn Seemann. 2004. *AI For Game Developers*. O'Reilly
- [EDE-09] Edery David, Ethan Molick . 2009. *Changing The Game*. FT Press
- [EVE-07] Everett, Gerald D, Raymond McLeod,Jr. 2007. *Software testing: testing across the entire software development life cycle*.
- [ESR-12] Entertainment Software Rating Board (official Page). 2012. *Game Ratings & Descriptor Guide*. Akses dari http://www.esrb.org/ratings/ratings_guide.jsp#rating_symbols. Tanggal akses 02 Januari 2012
- [FUL-08] Fullerton, Tracy. 2008. *Game Design Workshop*. Morgan Kaufmann Publishers.
- [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.
- [IAN-09] Millington, Ian and John Funge. 2009. *Artificial Intelligence For Games Second Edition*. Morgan Kaufmann.
- [JME-12] jMonkeyEngine Official Site. *jMonkeyEngine*. Diakses dari <http://www.jmonkeyengine.org>,<http://www.jmonkeyengine.com>. Tanggal Akses: 02 Januari 2012.
- [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-11] Kramer, Wolfgang. 2000. *What Is a Game?*. Diakses dari: <http://www.thegamesjournal.com/articles/WhatIsaGame.shtml>. Tanggal Akses: 06 Oktober 2011
- [LXP- 96] Lexicon Publication, Inc. 1996. *The New Lexicon Webster's Dictionary of The English Language*. Lexicon Publication, Inc.
- [MAR-03] Wolf J.P Mark & Bernard Perron. 2003. *The Video Game Theory Reader*. Routledge., GBR.
- [MON-09] Montola Makus, Jaakko Stenros, Annika Waern. 2009. *Theory and Design Pervasive Games Experience on the Boundary Between Life and Play*. Morgan Kaufmann.
- [PUB-10] humas1.2010. *FK Rebut Juara Pertama Gobak Sodor*. Akses dari <http://prasetya.ub.ac.id/berita/FK-Rebut-Juara-Pertama-Gobak-Sodor-192-id.html>. tanggal akses 04 januari 2012
- [RAM-05] Bangja, Ramesh. 2005. *Comprehensive Multimedia and Web Technology XII*. Laxmi Publication
- [REX-02] Black, Rex. 2002. *Managing the Testing Process : Practical Tools and Techniques for Managing Hardware and Software Testing*. Wiley Publishing, Inc., Canada
- [ROG-09] Pedersen, Roger E. *Game Design Foundations 2nd Edition*. Wordware Publishing, inc.
- [ROS-10] Roger, Scott E. 2010. *Level Up! The Guide to Great VideoGame Design*. John Wiley & Sons, Ltd.
- [SHR-10] Henry, Samuel. 2010. *Cerdas dengan Game*. PT Gramedia Pustaka Utama., Jakarta.
- [SIB-09] Sibero, Ivan C. 2009. *Langkah Mudah Membuat Game 3D*. Mediakom., Yogyakarta
- [SOM-11] Somerville, Ian. 2011. *Software engineering 9th Edition*. Pearson
- [SUP-08] Supendi Pepen, Nurhidayat. 2008. *Fun Game 50 permainan menyenangkan di Indoor dan outdoor*. Penebar Swadaya., Jakarta
- [SUR-05] Surya, Hendra. 2005. *Kiat Mengatasi Penyimpangan Perilaku*

Anak.PT. Elex Media Komputindo., Jakarta

[SUR-06] Surya, Hendra. 2006. *Kiat Membina Anak Agar Senang Berkawan*. PT. Elex Media Komputindo., Jakarta

[SZO-04] Zerbst, Stefan, Oliver Duvel. 2004. *3D GameEngine Programming*. Thompson Course Technology PTR.



LAMPIRAN

Bagian ini menunjukkan format kuisioner playtesting yang dilakukan terhadap game Gobak Sodor 3D.

Kuisioner

Feedback Game Gobak Sodor 3D

NAMA :	
USIA :	

Pertanyaan:

- Apakah game Gobak Sodor 3D menarik untuk dimainkan?
 - Tidak
 - Lumayan
 - Menarik
- Apakah game Gobak Sodor 3D mudah untuk dimainkan?
 - Tidak
 - Lumayan
 - Mudah
- Apakah tampilan game Gobak Sodor 3D menarik?
 - Tidak
 - Lumayan
 - Menarik
- Apakah game Gobak Sodor 3D membuat anda ingin memainkan secara nyata permainan Gobak Sodor?
 - Tidak
 - ya

Saran:
