

RANCANG BANGUN PERMAINAN SIMULASI UJIAN BERKENDARA 3D

SKRIPSI

Diajukan untuk memenuhi persyaratan

Memperoleh gelar Sarjana Komputer

UNIVERSITAS BRAWIJAYA



Disusun Oleh :

MURY FAJAR DEWANTORO

NIM 105060807111170

KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI

UNIVERSITAS BRAWIJAYA

PROGRAM STUDI INFORMATIKA

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2015

LEMBAR PERSETUJUAN

**RANCANG BANGUN PERMAINAN SIMULASI UJIAN
BERKENDARA 3D**

SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer



Disusun Oleh :

MURY FAJAR DEWANTORO

NIM 105060807111170

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Eriq M. Adams J., S.T., M.Kom.

NIP. 19850410 201212 1 001

Issa Arwani, S.Kom., M.Sc.

NIP. 19830922 201212 1 003



LEMBAR PENGESAHAN

**RANCANG BANGUN PERMAINAN SIMULASI UJIAN
BERKENDARA 3D**

Disusun Oleh :

MURY FAJAR DEWANTORO

NIM. 105060807111170

Skripsi ini telah diuji dan dinyatakan lulus pada tanggal 29 Januari 2015

Penguji I

Penguji II

Wibisono Sukmo Wardhono, S.T., M.T.

NIK. 820404 06 1 1 0091

Aswin Suharsono, S.T., M.T.

NIK. 840919 06 1 1 0251

Penguji III

Denny Sagita R., S.Kom., M.Kom

NIK. 85112406110250

Mengetahui

Ketua Program Studi Informatika

Drs. Marji, M.T.

NIP. 19670801 199203 1 001

PERNYATAAN**ORISINALITAS SKRIPSI**

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

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

Malang, 29 Januari 2015

Mahasiswa,

Mury Fajar Dewantoro

105060807111170

KATA PENGANTAR

Dengan nama Allah SWT Yang Maha Pengasih dan Penyayang. Segala puji bagi Allah SWT karena atas rahmat dan hidayahNya-lah penulis dapat menyelesaikan Tugas Akhir yang berjudul “Rancang Bangun Permainan Simulasi Ujian Berkendara 3D”. Shalawat serta salam atas junjungan besar kita Nabi Muhammad S.A.W. beserta keluarga dan para sahabat sekalian. Tugas Akhir ini disusun untuk memenuhi sebagai persyaratan memperoleh gelar Sarjana Komputer di Program Studi Informatika Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya Malang. Melalui kesempatan ini, penulis menyampaikan rasa hormat dan terima kasih penulis yang sebesar-besarnya kepada pihak yang telah memberikan bantuan lahir maupun batin selama penulisan tugas akhir ini. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa hormat dan terima kasih penulis kepada :

1. Kepada orang tua penulis dan seluruh keluarga yang senantiasa tiada henti hentinya memberikan do'a demi terselesainya tugas akhir ini.
2. Bapak Ir. Sutrisno, M.T, Bapak Ir. Heru Nurwasito, M.Kom, Bapak Himawat Aryadita, S.T, M.Sc, dan Bapak Eddy Santoso, S.Kom selaku Ketua, Wakil Ketua 1, Wakil Ketua 2 dan Wakil Ketua 3 Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
3. Bapak Drs. Marji, MT dan Bapak Issa Arwani, S.Kom., M.Sc selaku Ketua dan Sekretaris Program Studi Informatika serta segenap Bapak / Ibu Dosen, Staff Administrasi dan Perpustakaan Program Studi Informatika Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
4. Bapak Eriq Muh. Adams Jonemaro, ST., M.Kom dan Bapak Issa Arwani, S.Kom., M.Sc selaku dosen pembimbing tugas akhir penulis.
5. Seluruh dosen Program Studi Informatika atas kesediaan membagi ilmunya kepada penulis.
6. Semua Asisten Ka. Lab serta Laboran dari Laboratorium Program Studi Informatika yang telah memberikan banyak bantuan dan dukungan dalam menyelesaikan tugas akhir ini.

7. Saudara M. Rizky Prahesa Putra selaku artist yang membantu menyediakan keseluruhan art yang digunakan dalam tugas akhir.
8. Saudara Antonico Surya Pratama selaku promotional artist yang mempromosikan tugas akhir.
9. Mas Kurnia Prima P, Mas feбри, Mas Adit dan Mas Tian atas kesediaan waktunya untuk membantu penulis pengerjaan aplikasi dan laporan Tugas Akhir.
10. Teman seperjuangan : Yayan T, Fennia M , Maharani PSN, Silvi, Daria, Frida A, Arin A, Bernadeta L, Nurul Fitri, Alfian eka, Aditya Sheli, Mas ipul , Mas Daus dan Sandio atas dukungan dan semangat yang di berikan.
11. Seluruh pihak yang telah membantu kelancaran penulisan tugas akhir yang tidak dapat penulis sebutkan satu per satu.

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

Malang, Januari 2015

Penulis

ABSTRAK

Mury Fajar Dewantoro. 2015. : RANCANG BANGUN PERMAINAN UJIAN SIMULASI BERKENDARA 3D. Skripsi Program Studi Teknik Informatika, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya. Dosen Pembimbing : Eriq M. Adams J., S.T., M.Kom. dan Issa Arwani, S.Kom., M.Sc.

Game atau permainan adalah salah satu media hiburan yang dapat digunakan sebagai sarana pembelajaran. *Game* telah dibagi menjadi banyak *genre* dan *subgenre*. Sebuah *genre* atau jenis *game* biasanya di gunakan untuk menjelaskan gaya *gameplay game* tersebut. Dalam ujian SIM, syarat kelulusan pembuatan SIM adalah ujian teori, ujian praktik, dan ujian ketrampilan melalui simulator. Dalam ujian praktik SIM, Terdapat banyak materi yang di ujikan dalam ujian SIM, seperti uji parkir seri maju mundur , uji zigzag, uji parkir paralel, uji melewati jalan sempit, dan juga tanjakan. Untuk memenuhi kebutuhan akan media pembelajaran yang mengenal cara mengemudi kepada masyarakat sebelum mendapatkan SIM maka di buat game simulasi dengan nama *Smart Driving Simulator*. Sebelum permainan dibuat dilakukan perancangan yang terdiri dari perancangan *gameplay*, *level*, *concept arts*, dan *game screen flow*. Berdasarkan perancangan dilakukan implementasi menggunakan *Game Engine* Unity dengan bahasa pemrograman C#. Untuk menguji *game*, dilakukan pengujian yang terdiri dari *test suite* dan *playtesting*. Dari hasil pengujian yang dilakukan, *game smart driving simulation* telah berjalan dengan baik dan permainan ini dapat memberikan gambaran dalam hal menyetir mobil, serta hiburan yang menyenangkan melalui *gameplay* dan grafik yang menarik.

Kata Kunci : Unity Game Engine, game simulasi, ujian SIM

ABSTRACT

Mury Fajar Dewantoro. 2015. : *Development of 3D Driving Exam Simulation Game. Undergraduate Thesis of Informatic Study Program, Information technology and computer science Program, Brawijaya University. Advisor : Eriq M. Adams J., S.T., M.Kom. dan Issa Arwani, S.Kom., M.Sc.*

Games is one of the entertainment media that can be used as a learning tool. Gaming has splintered off into many genres and subgenres. A game genre is used to describe the style of gameplay. Requirements for driving's license test are theory test, practical test, and examination through the simulator. There are driving's material test such as perpendicular parking, zig zag, parallel parking, passing through the narrow streets, and incline start. Simulation game called *Smart Driving Simulator* is made to meets the need of driving learning to the public. Before the game was made, to design consist of designing gameplay, level, concept arts, dan game screen flow. The implementation is done by game engine unity with c# language based on design implementation. From the results of tests performed, *smart driving simulation* game has gone well and this game can give you an idea in terms of driving, as well as exciting entertainment through exciting gameplay and graphics.

Keyword : *game . driving simulation, Unity Game Engine, driving license test*

DAFTAR ISI

LEMBAR PERSETUJUAN	I
LEMBAR PENGESAHAN	II
PERNYATAAN	III
KATA PENGANTAR	IV
ABSTRAK	VI
ABSTRACT	VII
DAFTAR ISI	VIII
DAFTAR TABEL	XI
DAFTAR GAMBAR	XII
BAB I	1
1.1. LATAR BELAKANG	1
1.2. PERUMUSAN MASALAH.....	2
1.3. RUANG LINGKUP MASALAH.....	2
1.4. TUJUAN	3
1.5. MANFAAT.....	3
1.6. SISTEMATIKA PENULISAN	3
BAB II	5
2.1. GAME	5
2.2. FORCE FEEDBACK TOOLKIT	7
2.3. CAR TUTORIAL.....	9
2.4. UNITY3D GAME ENGINE.....	10
2.4.1. <i>Memahami dan Mengoptimisasi Antarmuka Pengguna</i>	10
2.4.2. <i>Saving Asset</i>	12
2.4.3. <i>Import Konten</i>	13
2.5. BLENDER.....	14
2.6. UJIAN BERKENDARA.....	16
BAB III	20
3.1. METODE PENELITIAN	20
3.1.1. <i>Studi Literatur</i>	20
3.1.2. <i>Perancangan Game</i>	21
3.1.3. <i>Implementasi</i>	21
3.1.4. <i>Pengujian</i>	22
3.2. PERANCANGAN GAME.....	23
3.2.1. <i>Deskripsi game</i>	23



3.2.2.	<i>Gameplay</i>	24
3.2.3.	<i>Rancangan Level</i>	28
3.2.4.	<i>Konsep Art</i>	31
3.2.5.	<i>Game Screen Flow</i>	35
BAB IV		42
4.1.	PEMILIHAN TEKNOLOGI	42
4.2.	IMPLEMENTASI GAMEPLAY	43
4.2.1.	<i>Implementasi karakter</i>	43
4.2.2.	<i>Implementasi level</i>	47
4.2.3.	<i>Implementasi player dan data</i>	48
4.3.	REALISASI LEVEL	50
4.3.1.	<i>Challenge 1</i>	51
4.3.2.	<i>Challenge 2</i>	51
4.3.3.	<i>Challenge 3</i>	52
4.3.4.	<i>Challenge 4</i>	52
4.3.5.	<i>Challenge 5</i>	53
4.3.6.	<i>Challenge 6</i>	54
4.3.7.	<i>Challenge 7</i>	54
4.3.8.	<i>Challenge 8</i>	55
4.3.9.	<i>Challenge 9</i>	56
4.3.10.	<i>Challenge 10</i>	57
4.4.	IMPLEMENTASI ARTS.....	57
4.4.1.	<i>Implemeantasi Logo</i>	57
4.4.2.	<i>Implemeantasi Head Up Display (HUD)</i>	58
4.4.3.	<i>Implementasi art koin</i>	61
4.4.4.	<i>Implemeantasi Art Pendukung</i>	61
4.4.5.	<i>Implementasi dunia dalam game</i>	62
4.5.	IMPLEMENTASI GAME SCREEN	63
4.5.1.	<i>Screen Menu Utama</i>	63
4.5.2.	<i>Screen Game</i>	64
4.5.3.	<i>Screen Garage</i>	64
4.5.4.	<i>Screen level</i>	65
4.5.5.	<i>Screen result</i>	65
4.5.6.	<i>Screen Allcompleted</i>	67
BAB V		68
5.1.	DESIGN TEST (<i>TEST SUITE</i>)	68
5.2.	PLAY TESTING.....	72
5.3.	HASIL ANALISA.....	74
BAB VI.....		76

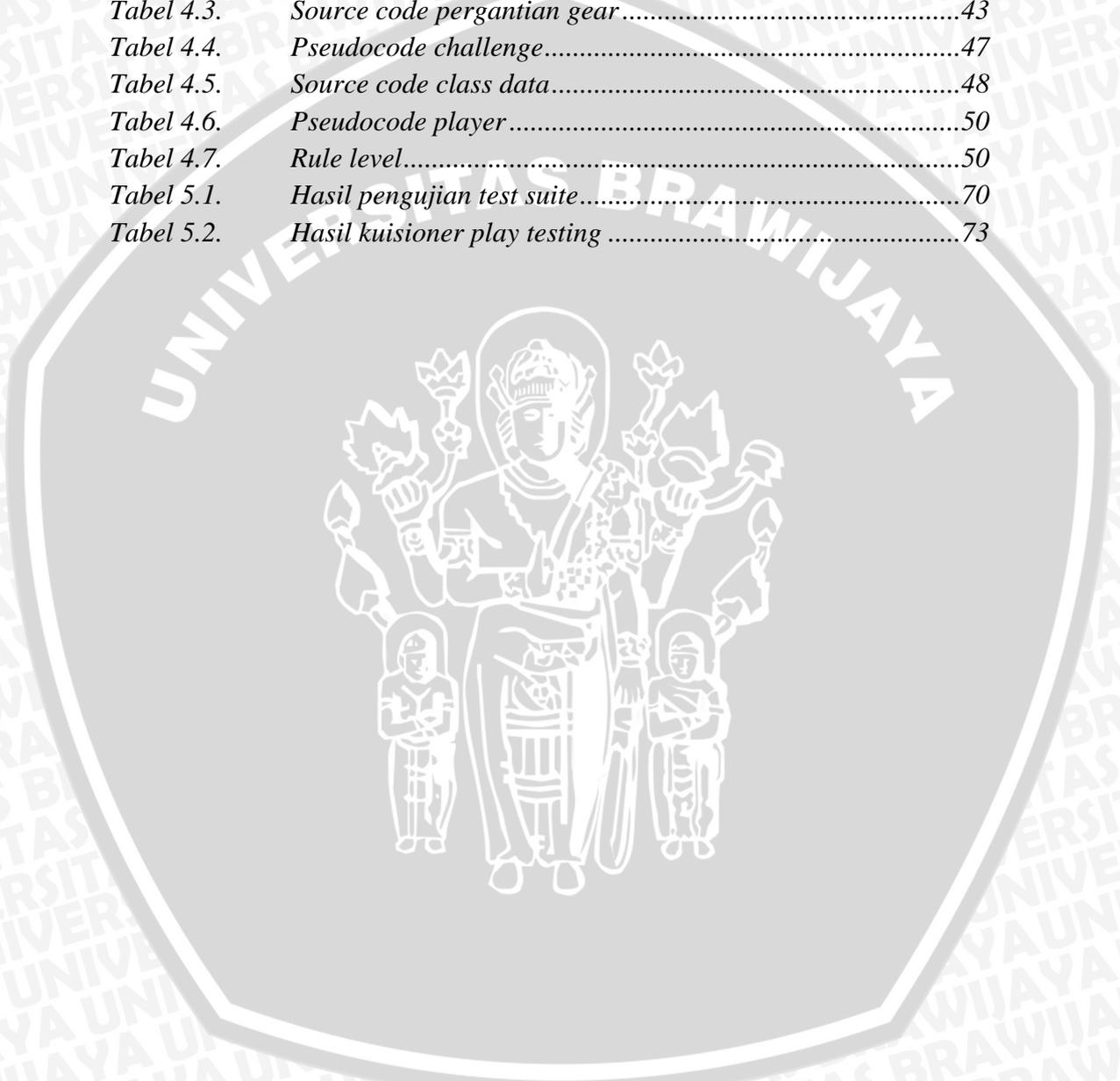


6.1. KESIMPULAN.....	76
6.2. SARAN.....	77
DAFTAR PUSTAKA.....	78
LAMPIRAN.....	79



DAFTAR TABEL

Tabel 3.1	<i>Implementasi Pembuatan Permainan</i>	22
Tabel 3.2	<i>Rancangan challenge</i>	29
Tabel 4.1.	<i>Spesifikasi perangkat keras</i>	42
Tabel 4.2.	<i>Spesifikasi perangkat lunak</i>	42
Tabel 4.3.	<i>Source code pergantian gear</i>	43
Tabel 4.4.	<i>Pseudocode challenge</i>	47
Tabel 4.5.	<i>Source code class data</i>	48
Tabel 4.6.	<i>Pseudocode player</i>	50
Tabel 4.7.	<i>Rule level</i>	50
Tabel 5.1.	<i>Hasil pengujian test suite</i>	70
Tabel 5.2.	<i>Hasil kuisioner play testing</i>	73



DAFTAR GAMBAR

Gambar 2.1	<i>Car tutorial pada unity asset store</i>	9
Gambar 2.2	<i>Interface pada unity</i>	11
Gambar 2.3	<i>Specular Color</i>	12
Gambar 2.4	<i>Cueball Prefab</i>	13
Gambar 2.5	<i>Aset pada project</i>	14
Gambar 2.6	<i>Blender 2.68a</i>	15
Gambar 2.7	<i>uji maju dan mundur</i>	16
Gambar 2.8	<i>uji zigzag</i>	17
Gambar 2.9	<i>uji parkir seri</i>	17
Gambar 2.10	<i>uji parkir paralel</i>	18
Gambar 2.11	<i>uji tanjakan</i>	18
Gambar 3.1	<i>Metodologi penelitian</i>	20
Gambar 3.2	<i>Logitech G27 Racing Wheel</i>	26
Gambar 3.3	<i>Peta kontrol steering wheel</i>	27
Gambar 3.4	<i>Formula untuk menghitung penambahan waktu</i>	28
Gambar 3.5	<i>Posisi camera utama dalam permainan</i>	31
Gambar 3.6	<i>Konsep logo game</i>	32
Gambar 3.7	<i>Konsep HUD bottom-left</i>	32
Gambar 3.8	<i>Konsep HUD bottom-right</i>	33
Gambar 3.9	<i>Konsep koin pada game</i>	33
Gambar 3.10	<i>Konsep challenge title</i>	34
Gambar 3.11	<i>Konsep tempat parkir</i>	34
Gambar 3.12	<i>Susunan kota dalam game</i>	35
Gambar 3.13	<i>Game screen flow keseluruhan</i>	36
Gambar 3.14	<i>Sketsa Screen Mainmenu</i>	37
Gambar 3.15	<i>Sketsa screen option</i>	37
Gambar 3.16	<i>Sketsa screen credit</i>	38
Gambar 3.17	<i>Sketsa dari screen exit</i>	38
Gambar 3.18	<i>sketsa dari screen garage</i>	39
Gambar 3.19	<i>Sketsa dari screen level</i>	40
Gambar 3.20	<i>Sketsa dari screen result-completed</i>	40
Gambar 3.21	<i>Sketsa dari screen result – failed</i>	41
Gambar 4.1	<i>Implementasi class data</i>	49
Gambar 4.2	<i>Implementasi gameobject challenge 1</i>	51
Gambar 4.3	<i>Implementasi gameobject challenge 2</i>	52
Gambar 4.4	<i>Implementasi gameobject challenge 3</i>	52
Gambar 4.5	<i>Implementasi gameobject challenge 4</i>	53
Gambar 4.6	<i>Implementasi gameobject challenge 5</i>	53
Gambar 4.7	<i>Implementasi gameobject challenge 6</i>	54

Gambar 4.8	Implementasi gameobject challenge 7	55
Gambar 4.9	Implementasi gameobject challenge 8	55
Gambar 4.10	Implementasi gameobject challenge 9	56
Gambar 4.11	Implementasi gameobject challenge 10	57
Gambar 4.12	Implementasi logo SDS	58
Gambar 4.13	Base dari HUD bottom left	58
Gambar 4.14	Implementasi throttle bar	58
Gambar 4.15	Implementasi gear indicator	59
Gambar 4.16	Implementasi angka pada speed indicator	59
Gambar 4.17	Implementasi brake bar	59
Gambar 4.18	Implementasi clutch bar	59
Gambar 4.19	Implementasi health bar	60
Gambar 4.20	Base dari HUD bottom-right	60
Gambar 4.21	Left sein lamp sprite	60
Gambar 4.22	Right sein lamp sprite	60
Gambar 4.23	Handbrake sprite	61
Gambar 4.24	uv dan texture koin	61
Gambar 4.25	Implementasi dari challenge title	62
Gambar 4.26	Implementasi dunia game	62
Gambar 4.27	Implementasi mainmenu	63
Gambar 4.28	Implementasi menu quit	63
Gambar 4.29	Implementasi layar game	64
Gambar 4.30	Implementasi layar Garage	64
Gambar 4.31	Implementasi screen level	65
Gambar 4.32	Implementasi screen result saat kondisi menang	66
Gambar 4.33	Implementasi screen result saat kondisi gagal	66
Gambar 4.34	Implementasi screen allcompleted	67
Gambar 5.1	Penambahan baris baru	75



BAB I

PENDAHULUAN

1.1. Latar Belakang

Game adalah sebuah latihan dari pelaksanaan sistem kendali, di mana terdapat pertandingan antar kekuatan yang dibatasi oleh aturan dalam rangka untuk mendapatkan hasil *disequibrial* [SCH-08]. *Game* mempunyai sebuah tujuan yang harus dicapai dan harus memiliki sebuah konflik. *Game* dibatasi oleh aturan, dimana hal ini merupakan salah satu aspek yang mendefinisikan sebuah *game*. Dalam sebuah *game* terdapat hasil tidak seimbang yang berarti ada pemain yang menang dan kalah.

Selama bertahun-tahun, *game* telah dibagi menjadi banyak *genre* dan *subgenre*. Sebuah *genre* atau jenis *game* biasanya di gunakan untuk menjelaskan gaya *gameplay game* tersebut. *Game* terdiri atas berbagai *genre*, *genre* tersebut antara lain *action*, *shooter*, *adventure*, *construction*, *life simulation*, *music*, *party*, *puzzle*, *sports*, *strategy*, dan *vehicle simulation* [ROG-10].

Vehicle Simulation atau Simulasi kendaraan merupakan salah satu jenis dari *game* simulasi yang membuat pemain dapat melakukan kegiatan pilot atau menyetir sebuah kendaraan, serta fokus kepada penciptaan pengalaman nyata mungkin [ROG-10]. Di PTIIK sendiri, *game* semacam ini pernah di buat dengan judul “Simulasi Berkendara 3D” [HER-13]. *Game* tersebut secara garis besar membahas tentang pengenalan dan pembelajaran rambu-rambu lalu lintas dan *game* yang akan di kembangkan mengambil ide dasar dari *game* tersebut.

Di Indonesia sendiri juga terdapat alat simulator yang di gunakan pada ujian SIM, namun hal tersebut tidak terealisasikan dengan jelas karena tersangkut kasus. Dalam ujian SIM, syarat kelulusan pembuatan SIM adalah ujian teori, ujian praktik, dan ujian ketrampilan melalui simulator [UUR-09]. Dalam ujian praktik SIM, Terdapat banyak materi yang di ujikan dalam ujian SIM, seperti uji parkir seri maju mundur , uji zigzag, uji parkir paralel, uji melewati jalan sempit, dan juga tanjakan [KUM-14]. Sedangkan untuk uji ketrampilan melalui simulator , materi yang di ujikan lebih ke arah penguasaan mobil di jalan.

Berdasarkan kebutuhan *game* yang akan di kembangkan nantinya akan lebih fokus ke arah simulasi ujian praktik sebelum mendapatkan sim. Dan berdasarkan kebutuhan akan pembelajaran dan aplikasi dalam dunia nyata, penulis mengambil “Rancang Bangun Permainan Simulasi Ujian Berkendara 3D” dengan tujuan mengenalkan kepada masyarakat yang belum dapat mengemudi bagaimana cara mengemudi.

1.2. Perumusan Masalah

Berdasarkan uraian latar belakang diatas maka dapat diambil rumusan masalah sebagai berikut:

1. Bagaimana merancang dan mengimplementasikan permainan simulasi ujian berkendara 3D dengan menggunakan steering wheel sebagai kontrol ?
2. Bagaimana menerapkan kegiatan ujian berkendara menjadi sebuah *gameplay* simulasi berkendara ?
3. Bagaimana menguji permainan simulasi ujian berkendara 3D dengan menggunakan *playtesting* dan *design test* ?

1.3. Ruang Lingkup Masalah

Ruang lingkup dari penelitian ini adalah:

1. *Game Smart Driving Simulation* di tujukan untuk platform PC.
2. *Game Engine* yang digunakan untuk merancang dan membangun *game Smart Driving Simulation* adalah Unity version 4.3.
3. Sudut pandang pemain dalam game in adalah kamera orang pertama.
4. *Game Smart Driving Simulation* hanya menggunakan uji parkir seri maju dan mundur serta uji tanjakan.
5. Coding dilakukan dengan IDE (*Integrated Development Environment*) Visual Studio menggunakan bahasa pemrograman C#.

1.4. Tujuan

Tujuan dari penelitian ini adalah untuk merancang, mengimplementasi dan menguji permainan simulasi ujian berkendara yang di bangun dengan menggunakan *game engine Unity* pada *platform* desktop PC.

1.5. Manfaat

Penulisan skripsi ini diharapkan dapat bermanfaat untuk berbagai pihak.

1. Bagi Penulis

- a. Mengaplikasikan ilmu yang di dapat selama mengikuti perkuliahan di Teknik Informatika Universitas Brawijaya
- b. Mendapatkan pemahaman tentang perancangan dan pengembangan *game* pada *platform* PC dengan bantuan *game engine*

2. Bagi Pengguna

- a. Memberikan sarana hiburan dan pendidikan secara bersamaan pada *game* ber-*platform* PC
- b. Menyediakan *game* yang memberi pengalaman sebagai gambaran saat berkendara.

1.6. Sistematika Penulisan

Sistematika penulisan dalam skripsi ini sebagai berikut:

1. Pendahuluan

Dalam bab ini diuraikan mengenai latar belakang, perumusan masalah, ruang lingkup masalah, tujuan, manfaat hasil penelitian, dan juga menjelaskan sistematika penulisan.

2. Kajian Pustaka dan dasar Teori

Bab ini menguraikan tentang teori yang berhubungan dengan judul kerja praktik terkait dalam penyusunan aplikasi *game* dan proses pengembangannya.

3. Metode Penelitian dan Perancangan

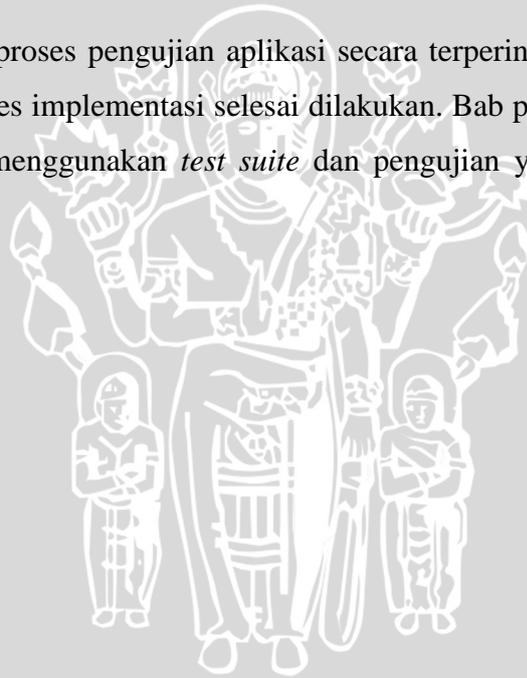
Bab ini berisi tentang metodologi yang akan digunakan dalam penelitian ini secara terperinci, dan juga hal yang terkait dengan perancangan aplikasi dari penentuan judul hingga sebelum proses implementasi. Tersusun secara structural mengikuti aturan *game design document*.

4. Implementasi

Bab ini berisikan tentang proses implementasi / penerapan yang akan dilakukan terhadap *game* yang dikembangkan secara terperinci dan berurutan.

5. Pengujian

Bab ini berisi proses pengujian aplikasi secara terperinci yang akan dilakukan setelah proses implementasi selesai dilakukan. Bab pengujian ini terdiri atas pengujian yang menggunakan *test suite* dan pengujian yang menggunakan *play testing*.



BAB II

KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini menjelaskan tentang apa saja dasar-dasar yang mendukung pengembangan *game SDS* ini. Dapat berisikan tentang penjelasan istilah yang terkait dalam penelitian yang sedang dikerjakan, mulai dari *library*, *car tutorial*, *blender* dan *game engine*.

2.1. Game

Elliot Avedon and Brian Sutton-Smith menyatakan bahwa *game* adalah sebuah latihan dari pelaksanaan sistem kendali, di mana terdapat pertandingan antar kekuatan yang dibatasi oleh aturan dalam rangka untuk mendapatkan hasil *disequibrial* [SCH-08:31]. Penjelasan lebih rinci mengenai pengertian tersebut dapat dilihat melalui paparan berikut ini [SCH-08:31].

- a. Terdapat pertandingan antar kekuatan, bagian ini menjelaskan bahwa *game* mempunyai sebuah tujuan yang harus dicapai dan *game* harus memiliki sebuah konflik
- b. Dibatasi oleh aturan, bagian ini menjelaskan bahwa aturan adalah salah satu aspek yang mendefinisikan sebuah *game*
- c. Hasil *disequibrial*, bagian ini menjelaskan bahwa dalam sebuah *game* didapatkan hasil yang tidak seimbang yang berarti terdapat pemain yang menang dan kalah.

Menurut pendapat lainnya, yaitu Greg Costikyan menyatakan bahwa *game* adalah sebuah struktur interaktif dari nilai endogen yang membutuhkan pemain yang berusaha untuk mencapai sebuah tujuan [SCH-08:31]. Penjelasan lebih rinci mengenai pengertian tersebut dapat dilihat melalui paparan berikut ini [SCH-08:31-32].

- a. Struktur interaktif, dalam bagian ini dijelaskan bahwa pemain bersifat aktif, dan tidak pasif, dan antara pemain dengan *game* terdapat interaksi antar satu sama lainnya. Sehingga hal tersebut menjelaskan bahwa *game* memiliki sebuah struktur (yang didefinisikan dengan aturan-aturan) yang

dapat membuat pemain dapat berinteraksi dan bagaimana dapat menginteraksi pemainnya.

- b. Berusaha mencapai sebuah tujuan, bagian ini menjelaskan *game* yang baik terdapat tantangan yang tidak terlalu banyak dan tidak terlalu sedikit untuk diselesaikan sehingga tujuan dapat tercapai.
- c. Nilai endogen berarti bahwa nilai yang terdapat dalam *game* hanya akan memiliki nilai didalam *game* tersebut. Misalnya, pada *game* monopoli terdapat uang monopoli, yang mana uang monopoli tersebut akan memiliki nilai jika dimainkan dalam *game* monopoli.

Menurut pendapat lainnya, yaitu Tracy Fullerton, Chris Swain, dan Steven Hofman, *game* adalah sebuah sistem formal dan tertutup yang melibatkan pemain dalam sebuah konflik terstruktur dan terselesaikan dengan hasil yang tidak seimbang [SCH-08:33]. Penjelasan lebih rinci mengenai pengertian tersebut dapat dilihat melalui paparan berikut ini [SCH-08:33-34].

- a. Menyertakan pemain, bagian ini menjelaskan bahwa *game* harus memiliki kualitas permainan yang baik sehingga dapat membuat pemain merasa tertarik untuk memainkannya
- b. Sistem formal dan tertutup, sebagai sebuah sistem *game* terdiri dari berbagai elemen yang bekerja bersamaan. Formal merupakan kata lain yang menjelaskan bahwa *game* merupakan sebuah sistem yang secara jelas terdefinisi dengan menggunakan aturan-aturan. Tertutup menjelaskan bahwa terdapat batasan dalam sebuah sistem.
- c. Hasil yang tidak seimbang, bagian ini menjelaskan bahwa dalam sebuah *game* didapatkan hasil yang tidak seimbang yaitu menang atau kalah.

Dari keseluruhan pengertian yang telah dipaparkan diatas, maka dapat diambil sebuah kesimpulan bahwa *game* adalah sebuah sistem interaktif terstruktur (didefinisikan dengan menggunakan aturan-aturan) yang melibatkan pemain untuk mencapai tujuan melalui konflik dan tantangan yang ada dan menghasilkan hasil yang tidak seimbang (menang atau kalah).

2.2. Force Feedback Toolkit

Force Feedback toolkit adalah sebuah *toolkit* yang di kembangkan oleh *user* pada *forum unity* dengan *id zaddo67*, untuk para pengembang *game* yang ingin mengembangkan *game* dengan menggunakan perangkat keras *steering wheel* sebagai *controller* [ZAD-12]. *Force feedback* memudahkan para pengembang *game* dalam mengintegrasikan *force feedback device* dengan *game*.

Toolkit ini memiliki banyak fitur, diantaranya [ZAO-12]:

- a. Terdapat *Force effects function*.
- b. Mendukung 2 dimensional *force feedback joystick*.
- c. Mendukung fungsi *rumble* pada *gamepad*.
- d. Mendukung sampai 4 *device* yang berjalan secara bersamaan.
- e. Terdapat contoh kode yang sudah terintegrasi dengan *game*.
- f. LED.
- g. Terdapat input mapping untuk kendaraan.
- h. Terdapat *sample scene* yang dapat terhubung dengan 4 *controller* sekaligus.
- i. Terdapat intruksi yang menjelaskan tentang fungsi-fungsi yang di gunakan.

Toolkit ini menyediakan satu set fungsi yang dapat digunakan untuk *force feedback* dan langsung terkoneksi dengan *unity*. *Toolkit* ini telah di rancang khusus agar bisa digunakan pada *unity free*. *Force feedback* atau *ffb toolkit* di kemas dalam satu *package* yang berada pada *folder standard aset*, hal ini di lakukan untuk menjamin agar *script* di *compile* terlebih sebelum *script* yang ada *project* di *compile*.

a. Force feedback Toolkit Files

1. **FFWheelInput.DLL** merupakan library pada *ffb*, yang mengatur koneksi pada *device* dan mengirim *force feedback* ke *game controller*. *File* ini harus berada pada *directory* utama dari folder *aset*.
2. **DInputProxy.cs** yang berfungsi sebagai proxy penghubung antara *unity* dan **FFWheelInput.DLL**.

RawJoystickState.cs di gunakan untuk menyusun state pada *game controller*.

3. **ControllerProperites.cs** merupakan sebuah *class* yang menyusun *properties* pada *game controller*.

b. Sample Scene Programs

1. **WheelInput.cs**, class ini di gunakan untuk mengambil dan memproses data input dari *DinputProxy.cs*, sehingga mudah digunakan.
2. **InputScan.cs**, class ini menscans kegiatan dari input.
3. **WheelMenu.cs**, class ini biasanya di pasangkan pada main camera pada scene. Class ini merupakan satu-satunya class yang mengakses *WheelInput*. Dengan menekan tombol “M” nantinya akan menampilkan menu setup untuk input.
4. **FFB Scene**, merupakan contoh scene yang menampilkan wheelmenu GUI untuk mengetest efek dari *forcefeedback* dan menampilkan map controller.
5. **FFBGenerator.cs**, class ini merupakan sebuah class yang akan generate instruksi dari *forcefeedback* ke game controller yang digunakan.

c. MultipleControllers Folder

1. *Ball.prefab* & *car.prefab*, merupakan sebuah prefab yang di gunakan pada game yang akan di coba.
2. *FFBCarController.cs*, merupakan class yang berfungsi sebagai control dari car prefab.
3. *FFBGame*, berisi game controller yang memanggil prefab.
4. *MultipleControllers Scene* contoh game yang mendemonstrasikan *forcefeedback*, *multiple controller* dan *direct input*.

d. Popular Vehicle Package Examples

1. *UnityCar.ZIP*, package ini berisi control file pada unity car.
2. *Edy.Zip*, package ini berisi control file pada edy’s vehicle physic .
3. *CarTutorial.Zip*, package ini berisi control file pada car tutorial.

2.3.Car Tutorial

Car tutorial merupakan *free tutorial* yang di sediakan oleh unity. *Car tutorial* berupa sebuah projek lengkap, dimana pada aset *car tutorial* berisi 3d model dari mobil, tutorial, *script*, dan *sample scene* yang sudah langsung di gunakan. Tutoial *in-depth* di buat Morten Summer, dengan grafik dan *coding* berasal dari *unity studio*. Di dalam Projek tutorial ini juga sudah terdapat realistik model dan fisik mobil yang akurat yang di buat oleh Jonas Echterhoff [TEC-12]. Gambar 2.1 adalah gambar dari car tutorial pada *unity asset store*.



Gambar 2.1 *Car tutorial* pada *unity asset store*

Car tutorial di bagi menjadi 3 sesi yang dapat dilihat melalui paparan berikut [TEC-12].

a. *Section 1 – Assembling the Car*

Pada sesi ini, car tutorial menjelaskan Bagaimana cara merakit 3d model mobil, script, dan komponen-komponen lainnya pada aset prefab mobil.

b. *Section 2 – Tweaking the Car*

Pada sesi 2, car tutorial menjelaskan Bagaimanacara agar mobil bisa berkelok.

c. *Section 3 – Under the Hood*

Pada sesi terakhir akan di jelaskan lebih detail tentang cara agar scripts bisa membuat mobil bergerak.

Car tutorial dapat di download di *unity asset store* pada link berikut <https://www.assetstore.unity3d.com/en/#!/content/10>.

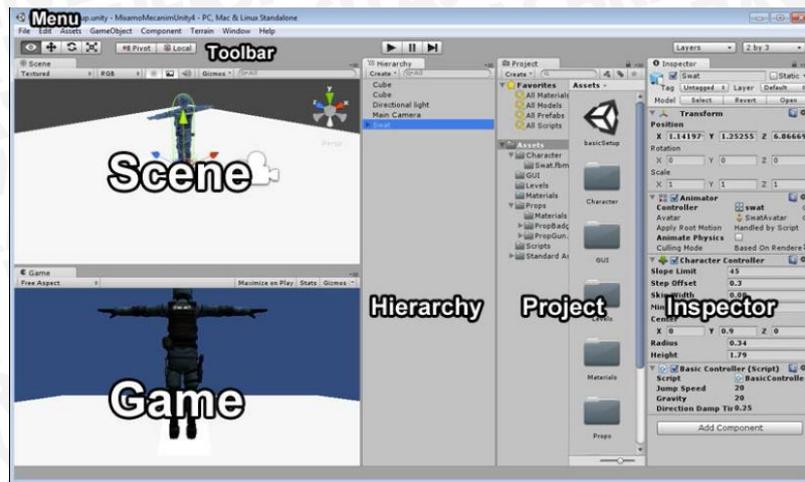
2.4. Unity3D Game Engine

Unity adalah sebuah 3D *game engine* multi platform yang dapat di gunakan untuk membuat dan mengembangkan *game* dengan berbagai macam pilihan platform, seperti Windows, Mac, Linux, iOS, Android, XBox 360, Wii U, dan PS3 serta web-based menggunakan *Unity plugins*. *Unity* berbasis object-oriented, yang artinya setiap object dalam *unity* mempunyai class dan setiap class dapat berasal dari “base” atau “parent” class. Sebagai *game engine* dengan basis object-oriented, *unity* memiliki hierarchy class [CHU-10].

Setiap *scene* terdiri dari banyak *game object*, di dalam *game engine* lain sering di kenal *entities*. Masing-masing memiliki nama, posisi, dan orientasi. Selain itu, *game object* juga bisa di hubungkan dengan parent-child yang posisi orientasi nya bergantung pada *game object* induknya. Sistem *scripting* pada *unity* di dasarkan pada *mono*, dimana *mono* merupakan versi *open-source* dari *.net*. Layaknya *.net*, *mono* mendukung banyak bahasa pemrograman dalam pengembangannya, namun pada *unity* hanya mendukung bahasa pemrograman *c#*, *boo*, dan *javascript* [CHU-10]. Dalam *unity*, *scripts* merupakan sebuah komponen dan dapat di bubuhkan pada *game object* seperti seperti komponen lainnya, hal ini bisa di lihat di *Unity Manual* pada bagian *using scripts* yang menjelaskan bagaimana cara membuat *scripts* dan membubuhkannya ke *game object*.

2.4.1. Memahami dan Mengoptimisasi Antarmuka Pengguna

Dalam antarmuka *unity* terdapat beberapa mode tata letak yang dapat di pilih. Tata letak *Unity* selalu dimulai dengan mode *Wide*, hal ini berbeda apabila pengguna telah merubahnya [SMI-13]. Cara merubah nya pada akses ke *Window-Layouts* dan memilih opsi lain, seperti 4 *Split* atau 2 *by 3*, dan perhatikan bagaimana antarmuka diatur dalam *Gambar 2.2*.



Gambar 2.2 Interface pada unity

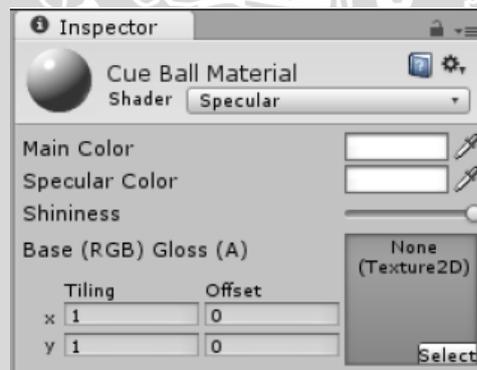
- Scene:** Tampilan ini digunakan untuk *position*, *rotate*, *scale*, dan *select* objek pada *game* dan juga mengarahkan level pengguna.
- Game:** Tempat untuk memainkan dan menguji *game*. Ini akan menghasilkan pengalaman bermain dari *player* seakurat mungkin.
- Hierarchy:** Berisi list objek-objek yang ada di *game* (seperti karakter, kamera, level geometri, pencahayaan, dan tekstur GUI)
- Project:** Tempat untuk membuat, mengatur, dan mengakses asset-asset *game* dari model 3D dan tekstur 2D sampai *script* C#, seluruh elemen akan dilist disini.
- Inspector:** Melakukan konfigurasi berbagai objek *game* (dari tampilan *Hierarchy*) atau asset (dari tampilan *Project*). Termasuk merubah peraturan ***Transform***, mengkonfigurasi komponen yang telah ada dan membuat komponen baru. Pada *inspector* juga dapat dilakukan penyesuaian preferensi lain untuk *game*.
- Toolbar:** Termasuk merubah alat yang digunakan untuk memanipulasi objek permainan dan navigasi adegan, alat control yang digunakan untuk bermain/berhenti dan menghentikan level, dan alat *drop-down* yang digunakan untuk mengelola *layers* dan *layouts*.

- g. **Menu:** Memberikan akses ke beragam daftar perintah yang meliputi impor/ekspor asset, pengaturan preferensi, pembuatan objek permainan, medan, tata letak, dan dokumentasi.

2.4.2. Saving Asset

Membuat *geometri primitive* dapat dilakukan dengan mudah pada *Unity*. Berikut adalah cara membuat objek dari *Unity* dan menyimpannya ke dalam proyek sebagai Prefab [SMI-13] :

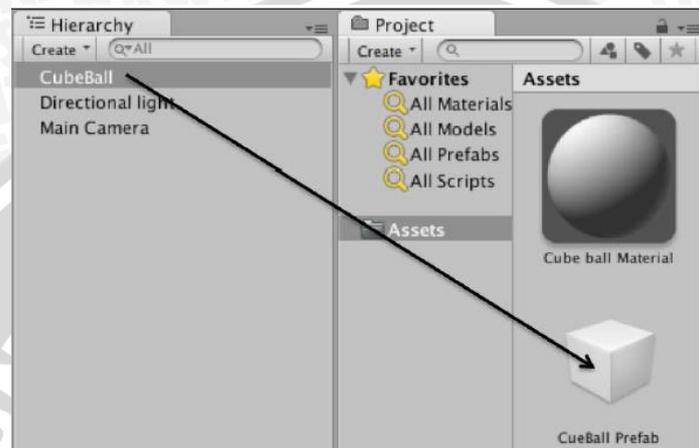
1. Di dalam editor *Unity*, arahkan ke *GameObject-Create Other-Sphere*
2. Dalam tampilan *Hierarchy*, klik kanan *Sphere* dan pilih opsi yang sesuai dari menu konteks untuk mengubah nama ke *Cue Ball*
3. Pada tampilan *Project*, klik tombol *Create* dan pilih opsi *Material*. Lalu, ubah nama material ke *Cue Ball Material*
4. Pada tampilan *Project*, pilih *Cue Ball Material*, kemudia dalam tampilan *Inspector*, ubah nilai *Shader* ke *Specular*
5. Mengatur warna *Specular* menjadi putih dan mengatur *Shininess* yang maksimal seperti pada Gambar 2.3.



Gambar 2.3 *Specular Color*

6. Dari tampilan *Project*, Tarik *Cue Ball* ke objek game *Cue Ball* pada tampilan *Hierarchy*
7. Pilih *Cue Ball*. Kemudia, akses *Component|Physics|Rigidbody*

- Objek *game* telah selesai dibuat, klik tombol *Create* pada tampilan *Project View* dan pilih opsi *Prefab* dan ubah nama menjadi *Cue Ball Prefab*,
- Tarik objek *Cue Ball* dari *Hierarchy* ke *Prefab* di *Project*. Objek game telah siap untuk digunakan pada proyek ini.

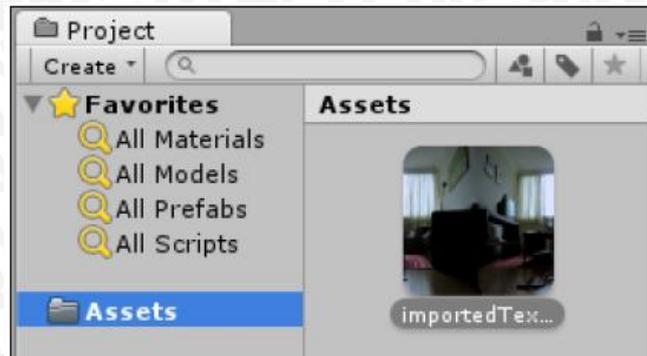


Gambar 2.4 *Cueball Prefab*

2.4.3. Import Konten

Dalam pengembangan sebuah game, para pengembang memerlukan aset tambahan di luar aset unity. Maka dari itu, para pengembang mengimport aset dan konten. Setelah membuat sebuah model 3D, audio clip, movie clip, atau texture, bisa diimpor ke dalam project yang diinginkan. Berikut adalah langkah-langkah untuk mengimpor sebuah asset ke dalam menu [SMI-13]:

- Dalam editor *Unity*, akses menu *Assets*
- Pilih opsi *Import New Asset*.
- Cari file yang diinginkan, lalu pilih impor
- File tersebut sudah masuk ke dalam list *Project* seperti gambar dibawah ini:



Gambar 2.5 Aset pada project

2.5. Blender

Aplikasi blender merupakan aplikasi Pengolah grafis 3D *open source* yang dikembangkan oleh Blender Foundation, sebuah organisasi independen dari Belanda. Aplikasi blender ini sangat berguna bagi penggemar grafis 3D, animasi bergerak, dan lain-lain. Aplikasi ini cocok digunakan bagi siapapun yang ingin berkarya dengan biaya yang murah. Meskipun Blender merupakan Aplikasi yang gratis, aplikasi ini juga menawarkan fitur yang sangat canggih dan tidak bisa dipandang remeh [FLA-10].

Ton Roosendaal, cofounder dari studio animasi NeoGeo, bertanggung jawab terhadap perusahaan rumah *software*-nya. Namun perusahaan tersebut tertimpa masalah financial, dan di tahun 2002 dengan investor mulai menarik aset, Ton mengambil langkah radikal dengan menawarkan *software* Blender sebagai *software open source*, dengan ketentuan bahwa penggemar Blender dapat meningkatkan €100.000 untuk melepaskan utang. Dalam beberapa minggu Blender telah memperoleh pengikut setia, dan tawaran berhasil dalam beberapa minggu. Sesuai dengan janji Ton, Blender dikeluarkan sebagai *open source software* dibawah lisensi GNU *General Public Lisence* (GPL), dan Ton telah mendedikasikan dirinya untuk akan mengembangkan Blender. Hingga kini Blender berkembang karena kuatnya penggemar-penggemar dan pengembang-pengembang yang tanpa lelah bekerja untuk menyempurnakan aplikasi dan

penggunaannya [FLA-10]. Seperti yang ditampilkan dalam Gambar 2.6 yang merupakan tampilan Blender versi 2.68a.



Gambar 2.6 Blender 2.68a

Blender memiliki peran yang sangat penting dalam pengembangan animasi, modeling, dan *game* khususnya pada konten 3D. Dengan beberapa fitur yang tidak kalah bagus dengan *software* 3D berbayar lainnya, Blender mampu menunjukkan kemampuannya dan menghasilkan beberapa karya hebat yang memotivasi para blender *artist* (sebutan untuk pengguna Blender) dalam menghasilkan karya-karya lainnya. Blender memiliki fitur yang hampir menyerupai dengan *software* 3D yang berbayar dan dapat digunakan secara maksimal walaupun memiliki ukuran yang kecil dibandingkan *software* 3D lainnya.

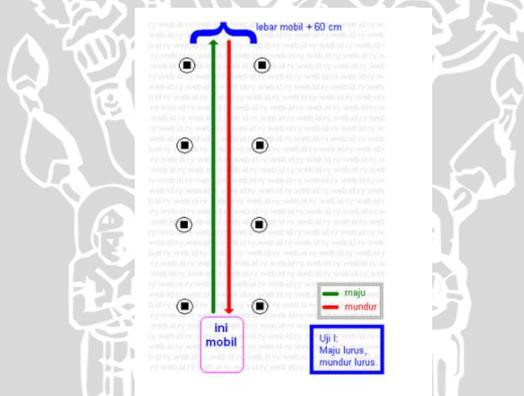
Untuk membuat transisi 3D menggunakan blender, ada beberapa hal yang perlu diingat yang akan membuat lebih terasa mudah :

- Blender adalah aplikasi gratis. Seorang yang ingin membuat kreasi dengan aplikasi ini tidak perlu memikirkan lagi masalah pembayaran untuk meng-upgrade atau isu mengenai lisensi.
- Blender adalah aplikasi yang mempunyai komunitas. Jika dalam penggunaannya terdapat kesulitan kita dapat langsung mengunjungi halaman forum di <http://blenderartist.org>

Banyaknya sumber-sumber yang ada pada tutorial online, termasuk <http://blendercookie.com>, <http://blenderguru.com>, dan <http://blendernewbies.com> [JON-12].

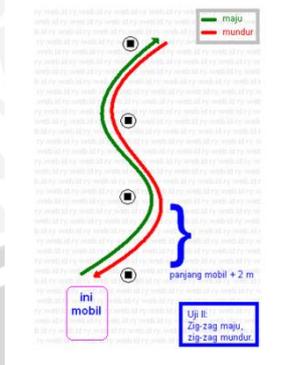
2.6. Ujian berkendara

Salah satu syarat lulus ujian berkendara yang di jelaskan pada undang-undang no 22 tahun 2009 tentang lalu lintas di jalan meliputi ujian teori, ujian praktik, dan ujian keterampilan melalui simulator [UUR-09]. Untuk ujian teori, materi yang di gunakan adalah tentang pengetahuan umum berkendara. Untuk ujian praktik, materi yang di gunakan adalah tentang bagaimana cara berkendara yang baik. Pada materi ujian praktik kendaraan roda empat, meliputi : Uji menjalankan Ranmor maju dan mundur pada jalur sempit, Uji zig zag maju dan mundur, Uji parkir parallel dan parkir seri, dan Uji mengemudikan Ranmor berhenti di tanjakan dan turunan [KUM-14].



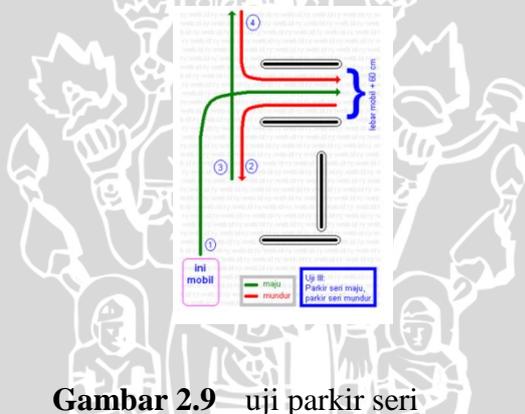
Gambar 2.7 uji maju dan mundur.

Pada Gambar 2.7 menerangkan tentang uji maju dan mundur pada jalan sempit, seolah-olah peserta ujian melewati gang sempit sepanjang 60 m.



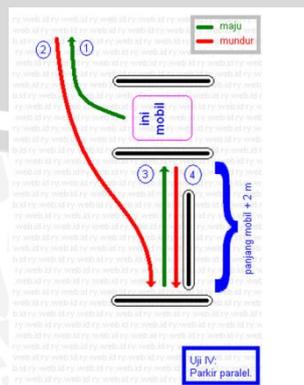
Gambar 2.8 uji zigzag

Uji zigzag pada Gambar 2.8 menerangkan tentang peserta ujian melewati rintangan zigzag. Tidak boleh ada satupun balok yang roboh. Bodi dari mobil tidak boleh melewati garis batas kiri dan kanan.



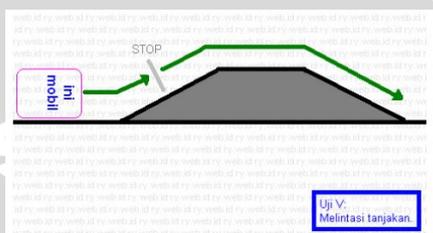
Gambar 2.9 uji parkir seri

Pada Gambar 2.9 menjelaskan tentang uji parkir seri, yang mana peserta ujian akan di minta untuk memarkir mobil dengan posisi seri maju. Kemudian pada uji parkir seri mundur, peserta di minta untuk mengeluarkan mobil dengan cara berjalan mundur.



Gambar 2.10 uji parkir paralel

Pada Gambar 2.10 menjelaskan tentang uji parkir paralel, yang mana peserta ujian akan di minta untuk memarkir mobil dengan posisi paralel maju. Kemudian pada uji parkir paralel mundur, peserta di minta untuk mengeluarkan mobil dengan cara berjalan mundur.



Gambar 2.11 uji tanjakan

Pada uji tanjakan yang di tunjukkan pada Gambar 2.11 menjelaskan bahwa praktik di lakukan dengan cara, mobil berjalan maju naik tanjakan. Kemudian ketika berada di tengah tanjakan mobil harus berhenti. Setelah penguji memberi aba-aba untuk melanjutkan perjalanan menaiki tanjakan, maka mobil di haruskan untuk bergerak maju tanpa mundur sedikitpun.

Pada ujian ketrampilan motor dan mobil akan memberikan nilai setelah si pemohon SIM melakukan simulator Dalam tes simulator motor, ada 6 poin yang diuji yakni sebagai berikut [AME-12] :

1. Slalom. Tes slalom ini adalah tes mengemudikan motor secara zigzag. Pemohon SIM harus mendapatkan nilai minimal 70 agar dapat lolos dalam tes.
2. Tikana. Tes ini adalah mengemudikan motor dengan membentuk formasi angka 8. Peserta uji SIM harus mendapatkan nilai minimal 70
3. Reaksi. Tes yang dilakukan bertujuan untuk melihat reaksi peserta dalam pengereman. Dalam tes ini, peserta akan berada di jalan lurus kemudian nanti akan menemui cabang jalan berbentuk 'V', di mana dia di situ harus mengerem ketika akan belok kiri atau kanan. Nilai ujian ini minimal 60.

4. Tes balik arah dilakukan pada model jalan letter U, di mana motor tidak boleh menabrak kiri-kanan jalan. Nilai ujian minimal 60.
5. Tes pada jembatan. Dimisalkan pada jembatan Cawang-Priok. Pada jembatan ini, peserta tidak boleh terguling ke kiri atau ke bawah karena motor akan terjatuh. Nilai yang harus didapatkan minimal 60.
6. Pengereman pada minimal kecepatan 30 km/jam pada jalan lurus, kemudian nanti ada tanda berhenti dan harus mengerem. Poin yang harus dikumpulkan adalah 60.

Kemudian untuk tes dalam simulator mobil ada 4 poin yang diuji yakni reaksi, antisipasi, konsentrasi dan perpaduan ketiga poin tadi,

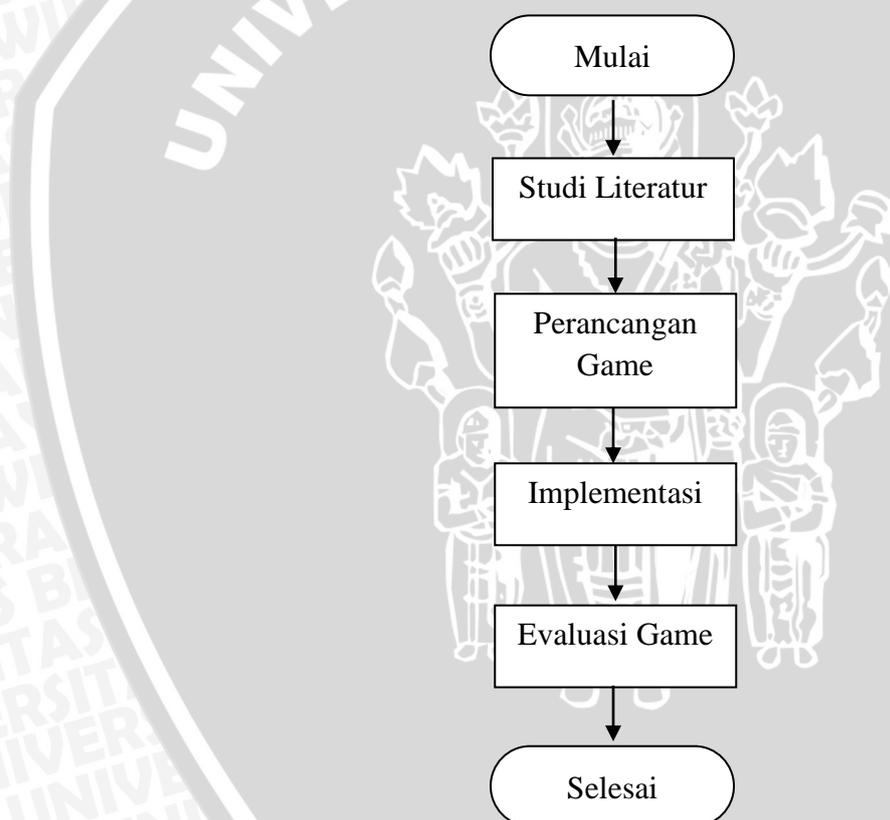
1. Reaksi : melihat reaksi pemohon SIM saat melihat tanda lampu lalu lintas menyala 'hijau', 'kuning' dan 'merah'.
2. Antisipasi: Mobil dalam keadaan jalan, setelah ada garis titik dia harus menggunakan lampu dim.
3. Konsentrasi: Lebih menitikberatkan pada kendali setir mobil di mana harus ada keseimbangan di kanan dan kiri. Dalam tes ini, pemohon SIM seolah-olah berada pada jalan kemudian ada garis hitam yang tidak boleh bertabrakan dengan garis merah. Tabrakan kedua garis ini tidak boleh lebih dari 4 kali.
4. Perpaduan ketiga poin tadi, di mana pemohon SIM harus menyalakan lampu saat mendahului, kendali setir, rem dan reaksi zigzag.

BAB III

METODE PENELITIAN DAN PERANCANGAN

3.1. Metode Penelitian

Pada bab ini dijelaskan langkah - langkah yang akan dilakukan dalam perancangan, implementasi dan pengujian dari *game* yang akan dibuat. Langkah-langkah metodologi dalam tugas akhir ini terdiri dari: studi literatur, perancangan *game*, implementasi dan pengujian.. Gambar 3.1 merupakan penjelasan singkat tentang alur metodologi penelitian yang dilakukan pada pengembangan *game* ini.



Gambar 3.1 Metodologi penelitian

3.1.1. Studi Literatur

Studi literatur menjelaskan dasar teori dan referensi yang akan digunakan dalam perancangan dan pembuatan *game*. Dalam studi literatur data-data

dikumpulkan dari buku, website, materi pembelajaran, maupun game sejenis yang sudah ada yang menjadi sumber dari teori dan referensi yang akan diimplementasikan lebih lanjut pada permainan yang dibuat. Dalam penelitian ini studi literatur yang digunakan antara lain :

- a. Permainan (*Game*)
- b. Unity Game Engine
- c. *ForceFeedback Toolkit*
- d. *The Car Tutorial*
- e. Perancangan *game*
- f. Pengujian *game*.

3.1.2. Perancangan *Game*

Setelah semua analisa kebutuhan dilaksanakan, hal berikutnya merupakan perancangan dari *game* yang ingin dikembangkan. Dalam perancangan ini dilakukan semua tahap-tahap awal pengembangan *game*. Sehingga dapat memudahkan implementasi *game* ini kedepannya. Perancangan berisi tentang keseluruhan *game concept* dan perancangan teknis dari *game* itu sendiri. Bagian perancangan dapat berisi hal-hal berikut, antara lain:

1. Deskripsi *Game*
2. Gameplay
3. Rancangan Level.
4. Konsep Art
5. Game Screen Flow

3.1.3. Implementasi

Implementasi aplikasi akan dilakukan dengan mengacu kepada pengguna yang memiliki perangkat *steering wheel*, dan memiliki *personal computer* masing-masing. Implementasi ini juga dilakukan dengan menggunakan bahasa pemrograman C# dengan *library* dan *Game Engine*. Hasil dari aplikasi ini adalah sebuah *pack file* berisikan program ber ekstensi windows dan *file-file* bawaan

game, dan langsung dapat digunakan oleh pengguna yang memiliki *Logitech Steering Wheel G27* dan *personal computer*. Implementasi juga membahas tentang tampilan antarmuka *game*, *art*, dan cara bermain. Dengan kata lain, pengimplementasian *game* dapat dibagi menjadi beberapa bagian agar mudah dipahami. Bagian-bagian tersebut antara lain :

1. Pemilihan Teknologi dan Platform
2. Implementasi Gameplay.
3. Realisasi Level.
4. Implementasi Arts.
5. Implementasi Game Screen.

Tabel 3.1 berikut menunjukkan program atau aplikasi yang digunakan dalam melakukan implementasi pembuatan permainan *SDS*:

Tabel 3.1 Implementasi Pembuatan Permainan

Jenis Implementasi	Nama Program
Desain 3D	Blender 2.71
Desain 2D	Paint Tool SAI
Game Engine	Unity 4.3.4f1
Bahasa Pemrograman	C#

3.1.4. Pengujian

Pengujian atau evaluasi *game* dilakukan agar dapat menunjukkan bahwa permainan yang dibuat telah mampu bekerja sesuai dengan kebutuhan yang melandasinya. Pengujian juga bertujuan untuk menemukan kesalahan-kesalahan atau kekurangan-kekurangan pada permainan yang diuji. Dalam penelitian ini evaluasi game terdiri dari

1. *Design test (test suite)*
2. *Playtesting*

Dari hasil pengujian yang dilakukan didapatkan umpan balik yang dapat digunakan untuk memperbaiki kesalahan-kesalahan atau kekurangan-kekurangan yang ditemukan dalam permainan yang diuji.

3.2. Perancangan Game

Perancangan berisikan tentang konsep *game* yang akan dikembangkan, seperti yang dijelaskan pada metodologi, dimulai dari konsep *game*, judul, dan lain sebagainya.

3.2.1. Deskripsi *game*

Game ini hanya dapat dimainkan oleh satu orang pemain. Pemain menggunakan steering wheel sebagai control utama untuk menjalankan karakter. Karakter pada game ini berupa kendaraan roda empat yaitu mobil.

1. Judul *Game*

Game yang akan di kembangkan ini nantinya akan berjudul Smart Driving Simulation atau bisa di singkat dengan sebutan SDS. Judul *game* akan seterusnya muncul dalam *game* dengan bentuk logo yang unik. Kata *Driving Simulation* yang ada dalam judul sendiri berasal dari genre *game* yang akan di kembangkan. Pemain di tuntut untuk menyelesaikan setiap level dengan waktu dengan waktu tercepat, yaitu dengan mengalahkan rekor nya sendiri tanpa mengurangi *healthbar* yang di miliki, karena waktu akhir pemain akan ditambah jika *healthbar* pemain berkurang. Pemain harus pintar – pintar mendapatkan waktu tercepat, maka dari itu *game* terdapat kata *Smart*.

2. Genre dan Topik

Genre dari *game* ini sendiri merupakan *Simulation* atau dapat dikatakan simulasi, dimana pemain akan lebih ditantang bersimulasi untuk mengendarai sebuah mobil. Topik dari *game* ini sendiri adalah *Driving*.

3. Goal (tujuan game)

Tujuan *game SDS* ini sederhana, pemain menyelesaikan setiap level dari *game SDS* ini dengan waktu terbaik yang pemain bisa dapatkan. Waktu dari pemain dapatkan akan di ubah dalam bentuk *Rank*. Terdapat 6 tingkatan *rank* di setiap *level*, yaitu : Rank S, Rank A, Rank B, Rank C, Rank D, Rank E.

Pemain juga di berikan *healthbar* di saat setiap level di mainkan jika pemain menabrak sesuatu di level berjalan, maka *healthbar* akan berkurang. Di akhir level waktu akan di hitung, jika *healthbar* pemain berkurang, maka pemain akanh mendapatkan *penalty* berupa penambahan waktu. Dari penambahan waktu tersebut akan di total menjadi waktu akhir, kemudian pemain mendapatkan *rank* dari waktu akhir tersebut.

4. Platform

Penggunaan *game* ini akan ditujukan untuk perangkat yang didukung oleh penggunaan *Steering Wheel*. Untuk saat ini *game* ini di hanya di dukung oleh perangkat keras *Logitech Steering Wheel G27*. *Game* ini menggunakan *Unity Game Engine* yang mendukung beberapa platform. Namun untuk penelitian dan pengembangan ini, hanya dipusatkan pada perangkat sistem operasi windows.

5. ESRB Rating

Game SDS ini termasuk pada peringkat *everyone*, karena konten dalam *game* ini dapat dikatakan layak untuk semua umur. Konten layak tersebut diantaranya adalah tidak adanya unsur kekerasan, dan seksual di dalamnya. Oleh karena itu, dapat dikatakan bahwa target pengguna dari *game* ini merupakan semua kalangan, mulai dari anak kecil, remaja, dan dewasa yang memiliki perangkat komputer personal dengan system operasi windows dan steering wheel.

3.2.2. Gameplay

Bagian *gameplay* akan menjelaskan tatanan dan alur *game* dalam *game SDS* ini. Mulai dari rule, karakter, mekanik dalam *game*, hingga perhitungan dari skor *game*.

Permainan dimulai ketika pemain memilih start pada mainmenu. Mobil akan berada Posisi awal gear mobi l adalah Terdapat koin melayang yang berada di depan pintu mesin parkir. Koin merupakan *trigger* untuk memasuki permainan *challenge*. Challenge merupakan nama dari tantangan yang di berikan game untuk pemain. Terdapat 10 challenge yang bisa di selesaikan pemain. Pemain hanya perlu menabrak koin melayang yang berada pada pintu mesin parkir. Jika pemain menyelesaikan ke-10 *challenge* yang terdapat pada *game*, maka pemain di nyatakan telah lulus atau menyelesaikan seluruh *challenge*.

1. Karakter

Dalam *game* SDS ini, karakter yang di mainkan oleh pemain adalah sebuah kendaraan beroda empat yaitu mobil. Mobil berupa *3D model* yang di buat dari perangkat lunak dengan feature membuat objek tiga dimensi. Objek model mobil sendiri di dapatkan dari sebuah website yang menyediakan berbagai macam model tiga dimensi dengan lisensi penggunaan tertentu. Objek yang di dapat dari website tadi, kemudian di *remodel* kembali agar efisiensi dari *game* SDS dapat tercapai.

Pemain di hadapkan langsung pada posisi dimana pemain berada di dalam mobil, sehingga objek mobil tidak terlihat sepenuhnya dan hanya terlihat bagian dalam dalam mobil.

2. Kontrol Game

Pada *game* SDS ini, pemain tidak menggunakan tombol pada keyboard sebagai kontrol utama, namun menggunakan perangkat keras Steering Wheel.

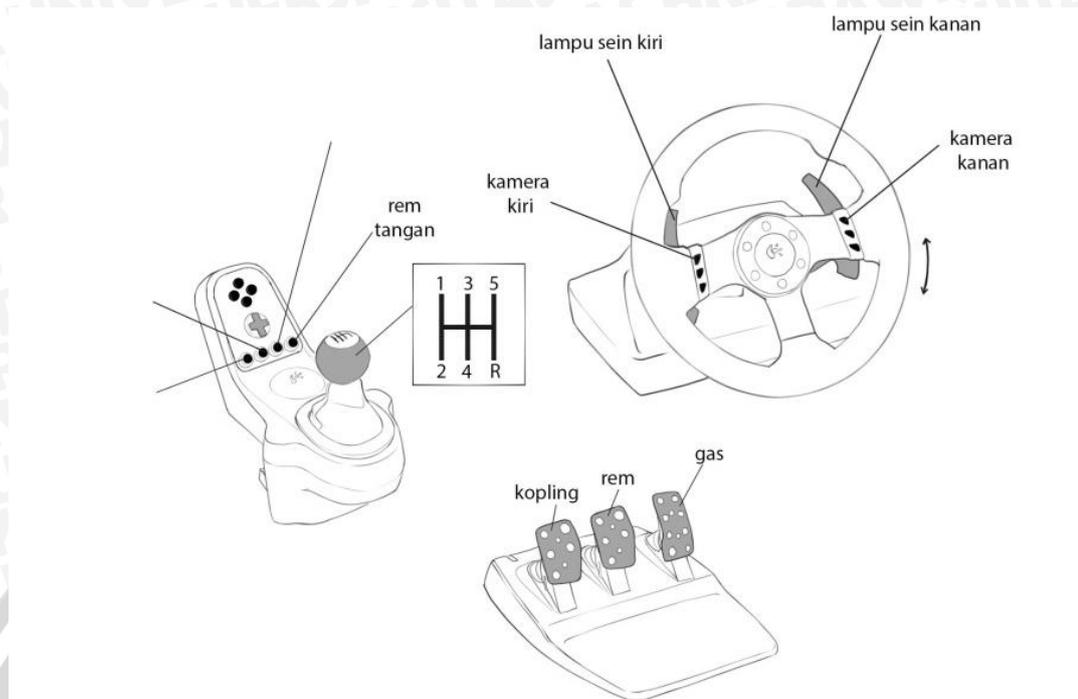


Gambar 3.2 Logitech G27 Racing Wheel

Pada gambar 3.2 merupakan perangkat keras dari kontrol yang akan di gunakan dalam pengembangan game ini. Perangkat keras tersebut memiliki 3 alat utama. Yang pertama *Racing Wheel*, alat ini di gunakan untuk membelok kan kendali dari karakter yaitu mobil.

Kontrol yang kedua adalah *foot Pedal*, alat ini memiliki 3 pedal pijakan yang masing-masing *axis* sendiri – sendiri. Dimulai dari yang paling kanan, pedal pertama nantinya di gunakan untuk pedal gas yaitu pedal yang apabila di pijak akan menambah kecepatan mobil. Pedal kedua , yaitu pedal pada bagian tengah nantinya akan di gunakan untuk fungsi rem pada mobil. Pedal yang terakhir, pedal yang paling kiri dari *Foot Pedal* nantinya akan digunakan untuk pedal kopling.

Alat yang terakhir pada kontrol perangkat keras yaitu Gear Stick, fungsi utama dari alat ini nantinya akan di gunakan untuk mengganti ger pada mobil. Pada alat ini juga terdapat tombol – tombol lain yang bisa di gunakan pada mobil. Gambar 3.3 merupakan peta kontrol steering wheel yang akan di gunakan pada *game SDS*.



Gambar 3.3 Peta kontrol steering wheel

3. Game Mechanics

a. Pintu Garasi

Lokasi garasi merupakan tempat awal mula pemain memainkan game ini. Pintu garasi akan terbuka dengan sendirinya apabila pemain berada di depan pintu garasi. Begitu juga sebaliknya, pintu akan tertutup dengan sendirinya jika pemain menjauh dari area garasi.

b. Mesin parkir

Mesin parkir berada di gedung parkir. Pada game ini terdapat 2 mesin parkir. Lokasi yang pertama berada di bawah dekat dengan pintu masuk gedung parkir. Lokasi mesin parkir yang kedua berada di lantai 2 gedung parkir. Mesin parkir akan terbuka sendiri apabila pemain berada di depan pintu parkir, namun jika pemain sudah pernah menyelesaikan tantangan 1 kali. Apabila pemain belum pernah menyelesaikan tantangan maka pintu mesin tidak akan terbuka.

c. Sistem Healthbar

Pada game ini SDS terdapat sistem *healthbar*. Healthbar ini akan berfungsi pada saat pemain memasuki level. Apabila pemain menabrak rintangan, tembok, ataupun pembatas jalan maka *healthbar* akan berkurang sesuai dengan benturan yang di dapat pada mobil. Apabila *healthbar* telah habis pada saat *level* berlangsung maka pemain di anggap gagal menyelesaikan level tersebut sehingga pemain harus mengurangi dari awal level tersebut. Jika pemain berhasil menyelesaikan level, sistem *healthbar* akan menghitung nilai *healthbar* yang tersisa. Jika *healthbar* berkurang maka waktu pemain menyelesaikan level tersebut akan di tambah. Penambahan waktu tersebut di dapatkan dari rumus pada Gambar 3.4.

$$extendTime = \frac{100 - health}{10}$$

Gambar 3.4 Formula untuk menghitung penambahan waktu

3.2.3. Rancangan Level

Dalam game SDS ini terdapat 10 level dengan 3 kategori. Kategori yang di gunakan yaitu yang pertama kategori ujian kendaraan parkir depan. Kategori level akan ada pada level 1 sampai dengan 4. Pada kategori ini pemain harus memarkirkan kendaraan di tempat tertentu dengan ketentuan posisi mobil menghadap ke depan. Pada level 1 sampai dengan 4 memiliki tingkat kesulitan yang berbeda – beda, misalnya dengan memberikan rintangan yang lebih banyak atau memindahkan lokasi finish ketempat yang berbeda.

Kategori yang kedua, yaitu kategori ujian kendaraan parkir belakang, kategori ini dimulai dari level 5 sampai dengan 8. Tidak seperti pada kategori sebelumnya yang di haruskan pemain memarkirkan kendaraan dengan posisi mobil menghadap ke depan. Pada kategori ini pemain memarkirkan kendaraan di tempat tertentu dengan posisi mobil menghadap belakang.

Kategori yang terakhir, kategori tentang melewati tanjakan. Posisi kendaraan yang di akan parkir sama dengan level 1 sampai dengan 4 yaitu parkir

dengan posisi mobil kedepan. Namun pada level 9 dan 10 ada sedikit perbedaan pada penempatan lokasi parkir. Pada level sebelumnya lokasi parkir pada lantai 1 gedung parkir, tetapi untuk level 9 dan 10 lokasi finsih parkir di pindahkan ke lantai 2 gedung parkir. Pada tanjakan di level 9 dan 10 pemain di tuntutan untuk melewati mesin parkir, pemain berhenti di tengah – tengah tanjakan dan pemain harus menginjak pedal rem dan kopling secara bersama – sama namun pedal tidak di injak secara penuh. Kemudian pintu mesin parkir akan terbuka dan player dapat menyelesaikan *level*. Jika pemain menyelesaikan seluruh challenge atau tidak terdapat nilai *rank e* di 10 *challenge* yang ada pada *game*, maka pemain dinyatakan lulus atau telah menyelesaikan seluruh *challenge*.

Pada setiap challenge nilai dari time dan health akan di hitung dan di nilai waktu final result. Nilai ini akan di masukkan ke variable dan kemudian bandingkan dengan nilai waktu rank yang sudah tentukan, seperti pada Tabel 3.2.

Tabel 3.2 Rancangan challenge

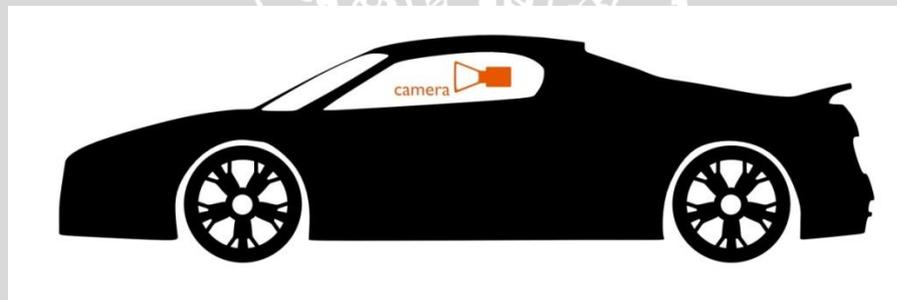
Nama Level	Atribut		Keterangan
Challenge 1	Rintang	Car	2
		barricade	4
	Rank	S	14
		A	20
		B	25
		C	30
		D	40
E	50		
Challenge 2	Rintang	Car	4
		barricade	4
	Rank	S	16
		A	20
		B	25
		C	30
		D	40
E	50		
Challenge 3	Rintang	Car	3
		barricade	6
	Rank	S	19
		A	25
		B	30
		C	35
		D	40
E	50		

Challenge 4	Rintangangan	Car	3
		barricade	8
	Rank	S	19
		A	25
		B	30
		C	35
		D	40
E	50		
Challenge 5	Rintangangan	Car	2
		barricade	3
	Rank	S	19
		A	25
		B	30
		C	35
		D	40
E	50		
Challenge 6	Rintangangan	Car	2
		barricade	3
	Rank	S	19
		A	25
		B	30
		C	35
		D	40
E	50		
Challenge 7	Rintangangan	Car	2
		barricade	5
	Rank	S	25
		A	30
		B	35
		C	40
		D	50
E	60		
Challenge 8	Rintangangan	Car	2
		barricade	5
	Rank	S	28
		A	35
		B	40
		C	50
		D	60
E	70		
Challenge 9	Rintangangan	Car	2
		barricade	3
	Rank	S	30
		A	35
		B	40
		C	45
D	55		
E	65		
Challenge 10	Rintangangan	Car	2

Rank	barricade	7
	S	30
	A	35
	B	40
	C	50
	D	60
E	70	

3.2.4. Konsep Art

Game SDS di rancang dengan konsep 3D dengan lokasi permainan berada do kota - kota yang memiliki gedung – gedung tinggi. Game SDS menggunakan camera yang ditaruh di dalam mobil, camera akan bergerak sesuai dengan gerakab mobil, jika mobil berbelok ke kanan, maka camera akan ikut berbelok ke kanan. Keuntungan dari penggunaan camera ini adalah agar dapat lebih memberikan konsep 3D kepada pemain. Gambar 3.5 menunjukkan posisi camera dalam *game* SDS.



Gambar 3.5 Posisi camera utama dalam permainan.

1. Konsep Logo *game*

SDS adalah singkatan nama *game* ini, Smart Driving Simulation. Konsep logo *game* terinspirasi dari nama *game* itu sendiri. Penggunaan nama logo ini bertujuan untuk memudahkan *user* untuk mengenali *game* ini. Konsep logo ditunjukkan pada Gambar 3.6.

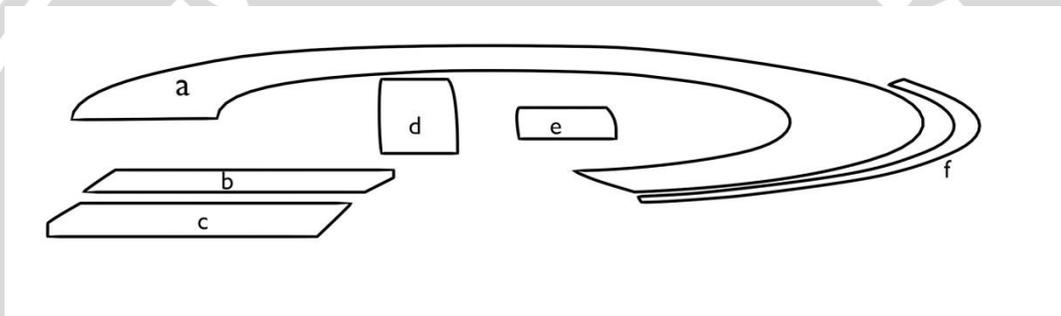
SDS
Smart driving simulation

Gambar 3.6 Konsep logo *game*

2. Konsep Art HUD (Head Up Display)

Saat *game* di mulai, pemain akan di berikan tampilan *Head-Up Display* (HUD). HUD system merupakan informasi penting untuk pemain yang ditampilkan dalam layar permainan. Pada *game* SDS terdapat beberapa HUD yang akan di tampilkan pada sudut – sudut layar.

Pada sudut kiri bawah layar terdapat HUD *bottom-left*, dimana HUD ini berisi informasi tentang *speed indicator*, *gear indicator*, *throttle bar*, *brake bar*, *clutch bar*, dan juga *health bar*. Gambar 3.7 menunjukkan konsep HUD *bottom left*.

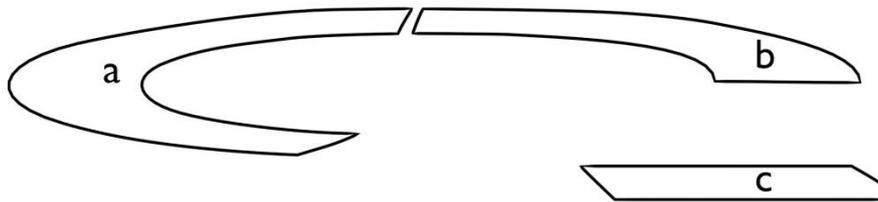


Gambar 3.7 Konsep HUD *bottom-left*

Pada HUD *bottom – left* terdapat beberapa informasi yang di tampilkan.

- a. **Throttle bar** : menunjukkan informasi tentang tekanan gas yang di tekan pemain.
- b. **Brake bar** : menunjukkan informasi tentang tekanan rem yang di tekan pemain.
- c. **Clutch bar** : menunjukkan informasi tentang tekanan kopling yang di tekan pemain.
- d. **Gear Indicator** : menunjukkan informasi tentang gear berapa yang sedang di gunakan pemain
- e. **Speed Indicator** : menunjukkan informasi tentang kecepatan mobil.
- f. **Health bar** : menunjukkan informasi tentang nyawa pemain dalam *game*. Art ini akan aktif ketika pemain memasuki *challenge game*.

Pada sudut kanan bawah terdapat HUD bottom-right, dimana HUD ini berisi mengenai indikator *sein lamp*, dan indikator *handbrake*. Konsep HUD tersebut di tunjukkan pada gambar 3.8.



Gambar 3.8 Konsep HUD *bottom-right*

Pada HUD *bottom-right*, di tampilkan beberapa informasi, diantaranya :

- a. **Left-lamp** : HUD ini menunjukkan informasi lampu sein kiri
- b. **Right-lamp** : HUD ini menunjukkan informasi lampu sein kiri
- c. **Handbrake** : menunjukkan informasi tentang *handbrake*.

3. Konsep art koin

Dalam game SDS terdapat sebuah koin yang melayang di depan pintu mesin parkir, koin ini adalah *trigger* untuk memasuki *challenge*. Koin ini di desain seperti koin pada umumnya, namun terdapat logo *game* di antarmuka nya. Konsep koin di tunjukkan pada gambar 3.9.



Gambar 3.9 Konsep koin pada game

4. Konsep Art Pendukung

Game SDS memiliki beberapa art pendukung, diantaranya

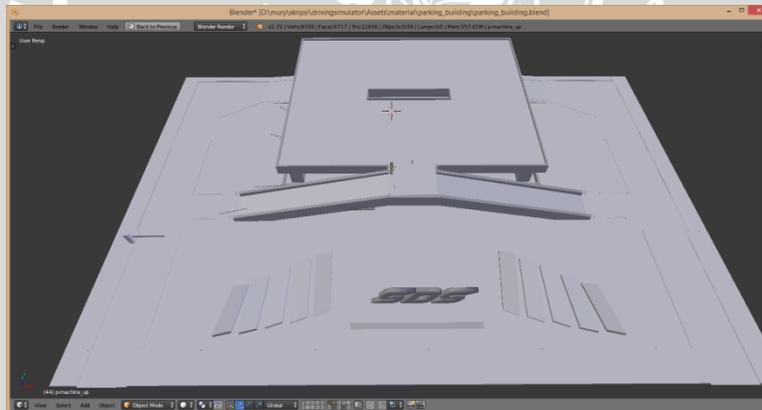
a. Challenge title

Art ini menunjukkan nama challenge yang sedang berlangsung. Konsep di tunjukkan pada gambar 3.10.



Gambar 3.10 Konsep challenge title

5. Konsep dunia dalam game



Gambar 3.11 Konsep tempat parkir

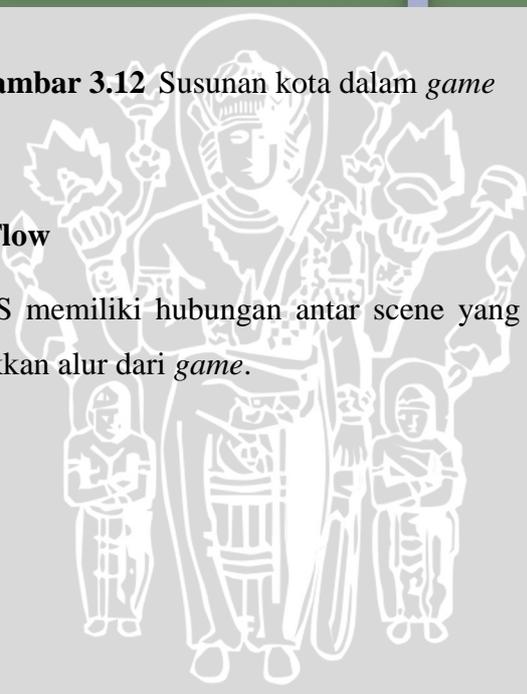
Dunia dalam *game* Smart Driving Simulation adalah perkotaan kecil dengan jalan yang berputar – putar. Dalam konsep untuk dunia game terdapat beberapa tempat yang akan dibuat, diantaranya tempat parkir yang dapat dilihat pada gambar 3.11. Dan juga terdapat pom bensin, taman, beberapa gedung yang saling berjajar, dan juga garasi. Sketsa dunia game di tunjukkan dalam gambar 3.12.

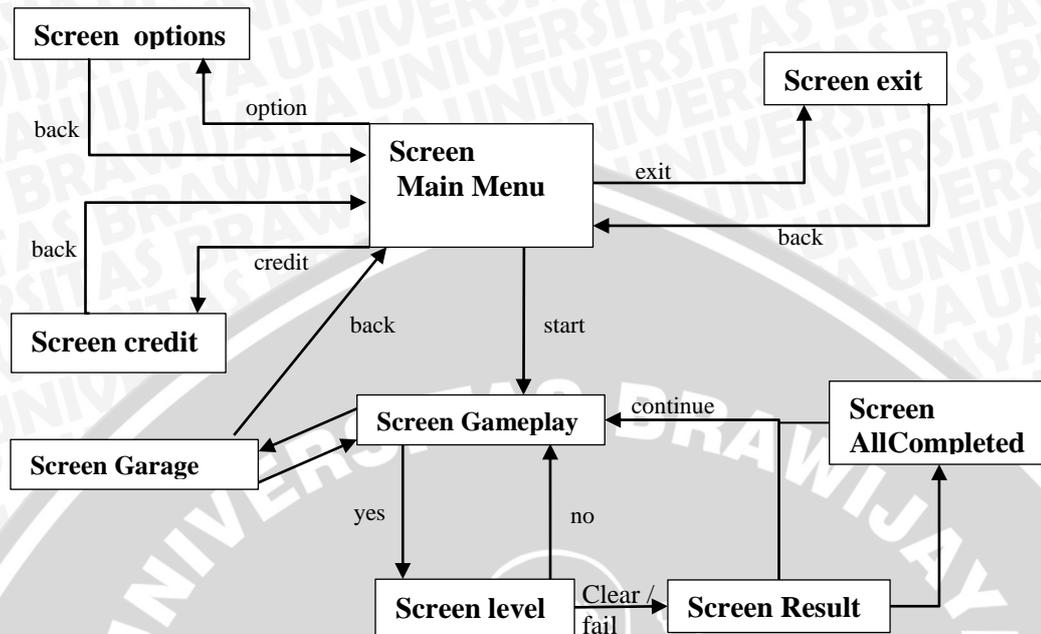


Gambar 3.12. Susunan kota dalam *game*

3.2.5. *Game* Screen Flow

Pada *game* SDS memiliki hubungan antar scene yang sangat kompleks. Gambar 3.13 menunjukkan alur dari *game*.

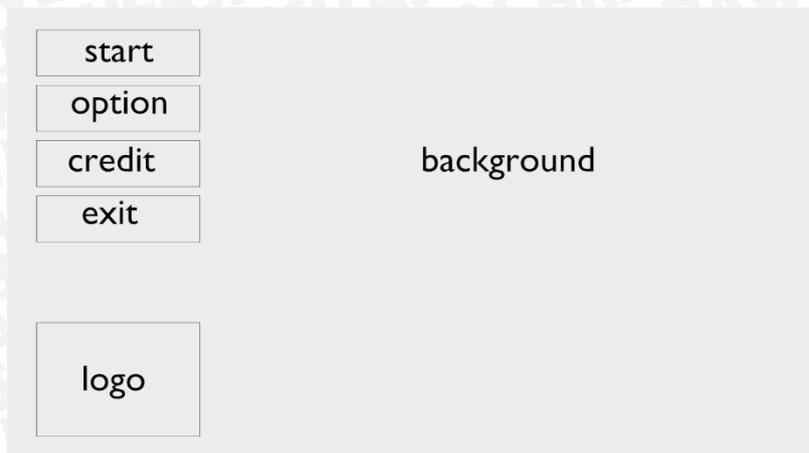




Gambar 3.13 Game screen flow keseluruhan

1. Screen MainMenu

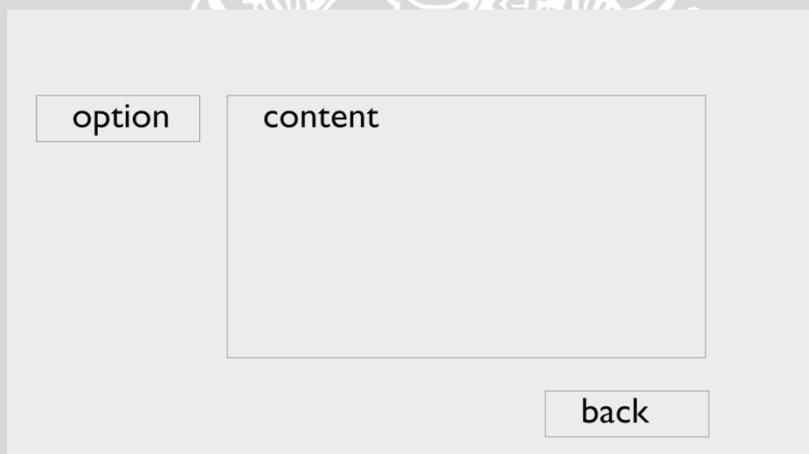
Screen main menu adalah screen pertama yang di lihat oleh pemain. Pada screen main menu terdapat 4 tombol yang di tampilkan di layar. 4 tombol itu adalah tombol start, tombol option, tombol credit, tombol exit. Apabila salah satu tombol itu di tekan, maka men- trigger ke screen selanjutnya. Tema background untuk screen main menu adalah tema gelap, di sini warna untuk background menggunakan warna hitam dan ada gambar mobil di sisi kanan layar. Terdapat logo SDS sisi kiri bawah layar. Konsep Screen mainmenu di ilustrasikan pada Gambar 3.14.



Gambar 3.14 Sketsa Screen Mainmenu

2. Screen Option

pada screen option, terdapat pengaturan yang akan dilakukan untuk game. Terdapat tombol back yang di gunakan untuk kembali ke screen sebelumnya. Gambar 3.15 menunjukkan sketsa dari screen option pada game ini.

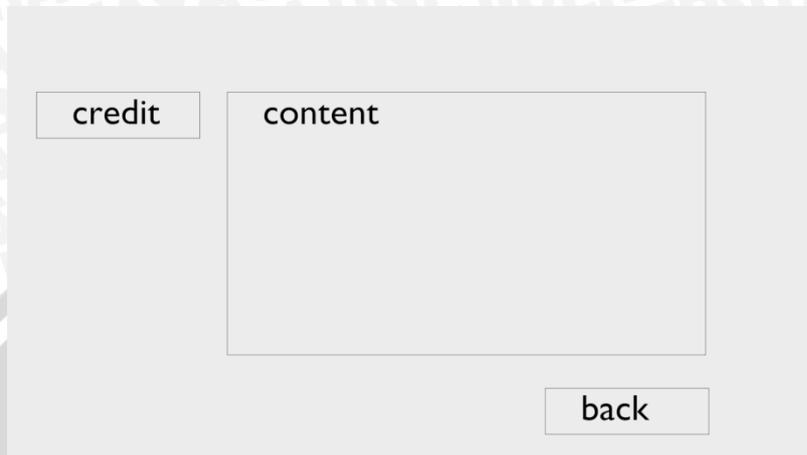


Gambar 3.15 Sketsa screen option

3. Screen Credit

Screen credit akan di temui pemain apabila pemain menu credit pada main menu. Pada screen credit, berisikan tentang profil developer. Di dalam screen ini hanya ada satu tombol yang di gunakan yaitu tombol *back*, tombol ini di gunakan untuk kembali mainmenu. Screen credit pada game SDS ini di gunakan untuk menunjukkan identitas yang berperan dalam pengembangan game ini, di

screen ini di tampilkan profil dari team developer, mulai dari *programmer*, *graphic illustrator*, dan juga *promotional art*. Screen credit di ilustrasikan pada gambar 3.16.



Gambar 3.16 Sketsa screen credit

4. Screen Exit

Pada screen sebelumnya yaitu screen mainmenu terdapat tombol exit. Apabila tombol tersebut di tekan, maka akan screen akan berpindah ke screen exit. Kemudian akan muncul 3 *dialogbox*, 1 *dialog box* berisi kata – kata apakah si pemain mau keluar dari window game dan 2 *dialogbox* adalah tombol *yes* dan *no*. Gambar 3.17 menunjukkan sketsa dari screen exit.



Gambar 3.17 Sketsa dari screen exit

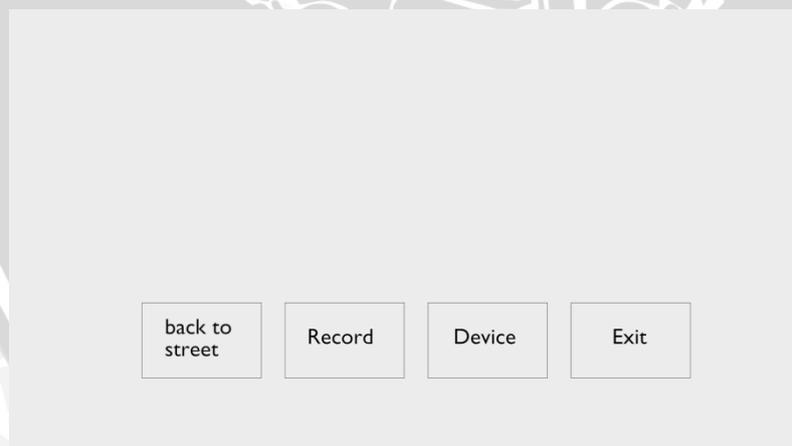
5. Screen Game

Pada *scene game*, HUD akan ditampilkan pada layar *game*. HUD utama berada pada sudut bawah layar *game*. HUD utama ini menunjukkan aktifitas – aktifitas yang di lakukan pemain. Pada sudut kiri bawah terdapat HUD yang menunjukkan kecepatan kendaraan, tekanan pijakan gas, tekanan pijakan kopling, tekanan pijakan rem, dan gear.

6. Screen Garage

Screen Garage bisa di temui ketika pemain memasuki garasi. Pada scene Garage terdapat 4 menu dan memilih menu tersebut menggunakan perangkat keras *mouse*. Menu yang pertama adalah menu *back to street*, jika tombol ini di tekan akan menutup *screen garage* dan kembali ke *screen game*.

Menu yang kedua adalah menu *record*, menu ini berisi catatan waktu dari challenge yang di mainkan oleh player. Menu yang ketiga adalah menu *device*, di sini terdapat contoh koneksi perangkat keras yang di gunakan. Menu yang terakhir adalah *menu exit*, menu ini di gunakan untuk kembali ke *screen main menu*. Gambar 3.18 menunjukkan sketsa dari *screen garage*.



Gambar 3.18 sketsa dari screen garage

7. Screen Level

Screen level ini terjadi ketika pemain mengambil koin yang ada di depan pintu parkir. *Scene* me- load data *prefab* yang ada pada *class data* dan

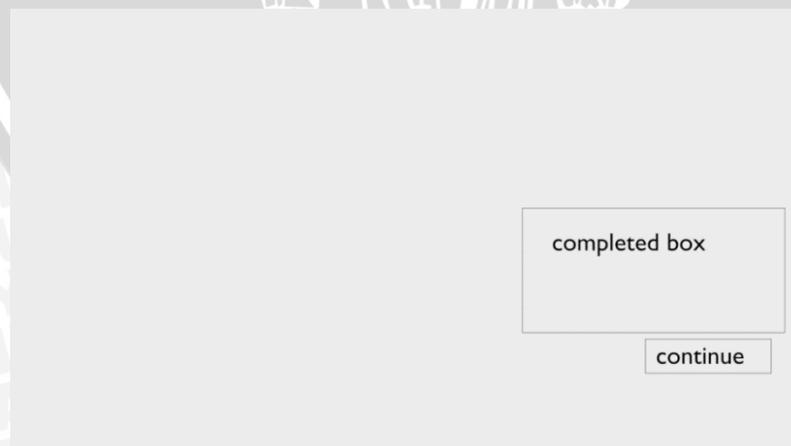
menampilkannya pada *scene level*. Terdapat dua tombol yang dapat ditekan, tombol tersebut adalah tombol *yes* dan *no*. Jika pemain menekan tombol *yes*, maka *challenge* akan dimulai. Gambar 3.19 menunjukkan sketsa dari *screen level*.



Gambar 3.19 Sketsa dari screen level

8. Screen Result - completed

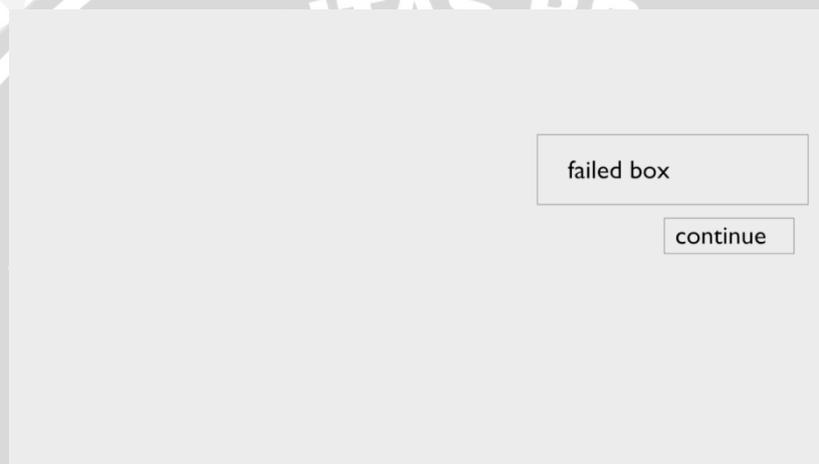
Screen result-completed merupakan screen yang di temui pemain ketika pemain menyelesaikan *challenge*. *Screen* ini akan menampilkan nilai nilai yang telah di dapat pemain setelah pemain menyelesaikan *challenge*, nilai tersebut di tampilkan dalam 2 jenis yaitu angka dan huruf. Nilai – nilai tersebut adalah nilai *health*, *time*, *timeExtend*, *final result*, dan juga *rank*. Gambar 3.20 menunjukkan sketsa dari *screen result-completed*.



Gambar 3.20 Sketsa dari screen result-completed

9. Screen Result-failed

Screen result-failed merupakan screen yang di dapatkan ketika health yang di berikan challenge kepada pemain telah habis. Setiap kali pemain menabrak tembok, rintangan, atau mobil lain, nilai tabrakan tersebut akan di kurangkan ke nilai health yang di berikan oleh challenge. Apabila nilai health tersebut berkurang sampai habis, maka permainan akan berhenti dan akan muncul screen result- failed. Pada gambar di 3.21 menunjukkan sketsa dari screen result- failed.



Gambar 3.21. Sketsa dari screen result – failed

10. Screen AllCompleted

Screen AllCompleted merupakan *screen* yang akan di temui pemain ketika pemain menyelesaikan ke 10 *challenge* yang terdapat pada *game* 10 atau dengan ketentuan tidak ada nilai *rank E* pada setiap *challenge*. Setelah pemain menyelesaikan ke-10 *challenge*, object bintang yang melayang akan muncul di depan garasi. Jika pemain menabrak bintang tersebut, pemain akan mendapatkan *screen allcompleted* ini.

BAB IV

IMPLEMENTASI

Bab ini akan membahas implementasi dari pembuatan *game SDS* sesuai dengan perancangan pada bab sebelumnya.

4.1. Pemilihan Teknologi

Pengimplementasian *game Smart Driving Simulation* atau yang di singkat dengan *SDS* di buat dalam lingkungan yang terdiri dari perangkat keras dan perangkat lunak. Spesifikasi perangkat keras yang dipakai dalam proses pengembangan dijelaskan pada tabel 4.1 berikut:

Tabel 4.1. Spesifikasi perangkat keras

PC	
<i>Processor</i>	Intel(R) Core(TM) i5-2320 CPU @ 3.00 GHz (4CPUs), ~3.0GHz
<i>Memory (RAM)</i>	4096 MB
<i>Harddisk</i>	1024 GB HDD
<i>Motherboard</i>	Acer –veriton M4610
<i>Graphic Card</i>	AMD RADEON HD 6450

Spesifikasi perangkat lunak yang dipakai dalam proses pengembangan aplikasi dijelaskan pada tabel 4.2 berikut:

Tabel 4.2. Spesifikasi perangkat lunak

SOFTWARE	
<i>Operating System</i>	Windows 8.1 Pro 64-bit (6.3, Build 9600)
<i>DirectX Version</i>	DirectX 11
<i>Game Engine</i>	Unity 4.3.4 fi
<i>Integrated Development Environment</i>	Microsoft Visual Studio 2010
<i>Programming Language</i>	C#
<i>2D Graphics Editor</i>	Paint Tool SAI
<i>3D Graphic Editor</i>	Blender

Perangkat keras dan perangkat lunak diatas digunakan dalam implementasi *game SDS* sesuai dengan kebutuhan dan batasan *game*. Beberapa batasan dalam mengimplementasikan *game* ini adalah sebagai berikut:

1. *Game SDS* di rancang untuk platform PC dengan sistem operasi *windows*,
2. *Game SDS* merupakan *game* single player dengan *graphic* 3D,
3. *Game SDS* memiliki 3 kategori dalam implementasi level yaitu : parkir seri maju, parkir seri mundur, dan tanjakan.

4.2. Implementasi Gameplay

Banyak faktor yang berpengaruh dalam pengimplementasian *gameplay* *game SDS*, di antaranya adalah implementasi karakter, implementasi *class level*, dan implementasi *class player data*.

4.2.1. Implementasi karakter

Game SDS menggunakan perangkat keras *Steering Wheel* sebagai kontrol utama. Agar mobil dapat di gerakkan dengan menggunakan *steering wheel* di butuhkan *class* yang berisi *method* yang mengakses *library forceFeedback*, *class* tersebut adalah *class check_control*. *Class car* merupakan *class* yang berada pada objek mobil, *class* ini di gunakan untuk menggerakkan mobil dan juga mengakses *class check_control*. kontrol mobil menggunakan gear manual, yang artinya pemain harus memindahkan posisi gear mobil tidak sama dengan *netral* agar mobil bisa bergerak.

Tabel 4.3. *Source code* pergantian gear

<i>Source code gear</i>	
1	<code>ingear = 0;</code>
	<code>if (checkControl.control_Gear.firstGear == 1 </code>
	<code>checkControl.control_Gear.secondGear == 1 </code>
	<code>checkControl.control_Gear.thirdGear == 1 </code>
	<code>checkControl.control_Gear.fourthGear == 1 </code>
	<code>checkControl.control_Gear.fifthGear == 1 </code>
	<code>checkControl.control_Gear.reverseGear == 1)</code>
2	<code>{</code>
3	<code> ingear = 1;</code>
4	<code> inGear = true;</code>
5	<code>}</code>

```
6 else
7 {
8     ingear = 0;
9     inGear = false;
10 }
11
12 if (checkControl.control_Basic.clutch > 0.4f)
13 {
14     if (checkControl.control_Basic.clutch > 0.8f)
15     {
16         if (clutchTime <= 0 && ingear != 1)
17         {
18             clutchTime = Time.deltaTime;
19         }
20     }
21     else
22     {
23         if (clutchTime <= 0 && ingear == 1)
24             clutchTimeHalf = Time.deltaTime;
25     }
26 }
27 else if (checkControl.control_Basic.clutch < 0.4f)
28 {
29     clutchTime = 0;
30     clutchTimeHalf = 0;
31 }
32
33 if (clutchTime != 0)
34 {
35     if (checkControl.control_Gear.firstGear == 1)
36     {
37         gearTime1 = Time.deltaTime;
38     }
39     else if (checkControl.control_Gear.secondGear ==
40 1)
41     {
42         gearTime2 = Time.deltaTime;
43     }
44     else if (checkControl.control_Gear.thirdGear == 1)
45     {
46         gearTime3 = Time.deltaTime;
47     }
48     else if (checkControl.control_Gear.fourthGear == 1)
49     {
50         gearTime4 = Time.deltaTime;
51     }
52     else if (checkControl.control_Gear.fifthGear == 1)
53     {
54         gearTime5 = Time.deltaTime;
55     }
56     else if (checkControl.control_Gear.reverseGear == 1)
57     {
58         gearTime6 = Time.deltaTime;
59     }
60     else
61     {
62         gearTime1 = 0;
```

```
63         gearTime2 = 0;
64         gearTime3 = 0;
65         gearTime4 = 0;
66         gearTime5 = 0;
67         gearTime6 = 0;
68     }
69 }
70 else
71 {
72     if (ingear == 1 || inGear)
73     {
74         gearTime1 = 0;
75         gearTime2 = 0;
76         gearTime3 = 0;
77         gearTime4 = 0;
78         gearTime5 = 0;
79         gearTime6 = 0;
80     }
81     else
82     {
83         canBackward = false;
84         canForward = false;
85     }
86 }
87 if (clutchTime < gearTime1)
88 {
89     canForward = true;
90     canBackward = false;
91     currentGear = 0;
92     stateGearUI = 1;
93 }
94 else if (clutchTime < gearTime2)
95 {
96     canForward = true;
97     canBackward = false;
98     currentGear = 1;
99     stateGearUI = 2;
100 }
101 else if (clutchTime < gearTime3)
102 {
103     canForward = true;
104     canBackward = false;
105     currentGear = 2;
106     stateGearUI = 3;
107 }
108 else if (clutchTime < gearTime4)
109 {
110     canForward = true;
111     canBackward = false;
112     currentGear = 3;
113     stateGearUI = 4;
114 }
115 else if (clutchTime < gearTime5)
116 {
117     canForward = true;
118     canBackward = false;
119     currentGear = 4;
```

```
120     stateGearUI = 5;
121 }
122 else if (clutchTime < gearTime6)
123 {
124     canForward = false;
125     canBackward = true;
126     currentGear = 0;
127     stateGearUI = 6;
128 }
129
130 else if(ingear== 0&& checkControl.control_Basic.clutch==
131 0)
132 {
133     canForward = false;
134     canBackward = false;
135     stateGearUI = 0;
136 }
137 if (canForward)
138 {
139     throttle = checkControl.control_Basic.throttle;
140     brake = checkControl.control_Basic.brake;
141     if (brake > 0)
142     {
143         if (dragTemp.z > 0)
144         {
145             brake = 0;
146         }
147         throttle = brake;
148         throttle = -throttle;
149     }
150 }
151 else if (canBackward)
152 {
153     throttle = -checkControl.control_Basic.throttle;
154     brake = checkControl.control_Basic.brake;
155
156     if (brake > 0)
157     {
158         if (dragTemp.z < 0)
159         {
160             brake = 0;
161         }
162         throttle = brake;
163     }
164 }
165 else
166 {
167     if (CurrentEnginePower > 0)
168     {
169         CurrentEnginePower -= Time.deltaTime * 200;
170     }
171     else if (CurrentEnginePower < 0)
172     {
173         CurrentEnginePower += Time.deltaTime * 200;
174     }
175 }
176 getInGear = ingear;
```

Pada tabel 4.3 merupakan *source code* yang di gunakan untuk memindahkan posisi gear mobil. Pada Baris 1 sampai baris sampai 5 merupakan sebuah kondisi apakah terjadi inputan pada *gearstick*, jika terdapat inputan maka status dari *ingear* akan berubah dari 0 menjadi 1. Jika inputan dilepas dan kembali ke semula maka status *ingear* akan kembli ke 0.

Baris 12 – 26 di gunakan untuk kondisi apakah kopling di injak atau tdak, jika kopling di injak penuh, maka menyimpan *delta time* ke variabel *clutchtime*. Baris 27 – 31 merupakan kondisi untuk kopling apabila terjadi tekanan atau inputan. Baris 33 adalah kondisi jika varibel *clutchtime* tidak sama denganm 0.

Baris 35 – 68 merupakan baris- baris kondisi untuk mendapatkan nilai variabel *geartime*. Variabel *gear time* akan di gunakan apabila kopling ditekan dan posisi *gearstick* di rubah dari tempat semula. Selanjutnya baris 87 – 176, baris ini merupakan baris yang di gunakan untuk menentukan nilai gear. Nilai *cluthtime* di bandingkan dengan nilai *geartime*. Jika nilai *clutchtime* kurang dari nilai *geartime* , maka akan mengganti nilai *gear* sesuai dengan gear yang di masukkan. Nilai yang di set ketika kondisi berhasil adalah , mengganti nilai *canforward* menjadi true, mengganti nilai *canbackward* menjadi false jika sedang masuk gear maju.

4.2.2. Implementasi level

Koin yang berada di depan mesin parkir merupakan trigger yang memanggil challenge. *Class systemManager* merupakan *class* utama yang berisi rule yang ada pada *game*, *class* ini juga yang nantinya akan memanggil objek – objek yang akan di gunakan pada saat *challenge*. Tabel 4.4 merupakan *pseudocode* pada *class systemManager* yang mengatur memanggil *prefab challenge*. Baris 4 merupakan baris kondisi yang mengecek apakah karakter menyentuh koin atau tidak. Pada baris 17 counter di tambah 1 agar kondisi tidak terulang secara terus menerus. Sedangkan pada baris 18 dan baris 19 merupakan baris yang memanggil fungsi.

Tabel 4.4. *Pseudocode challenge*

Pseudocode challenge

```

1  Int counter
2  GameObject prefab
3  START
4      If collision with coin
5          Screen pemilihan level TRUE
6      END IF
7      If Screen pemilihan level TRUE
8          If YES
9              missionUp TRUE
10         Else
11             Screen pemilihan level FALSE
12             missionUp FALSE
13         END IF
14     END IF
15
16     If(missionUp && counter == 0)
17         Counter++
18         Gameplay(true)
19         Loadlevel(prefab)
20     END IF
21 END
22

```

Pemain memasuki *challenge dialog* dan kemudian memilih *yes*, setelah itu class *systemManager* akan memanggil *game object prefab*. Dalam *prefab* terdapat objek dan class yang menentukan nilai pemain. Class yang terdapat pada *prefab*, diantaranya adalah *class levelsection* dan *class level*.

4.2.3. Implementasi player dan data**Tabel 4.5.** *Source code class data*

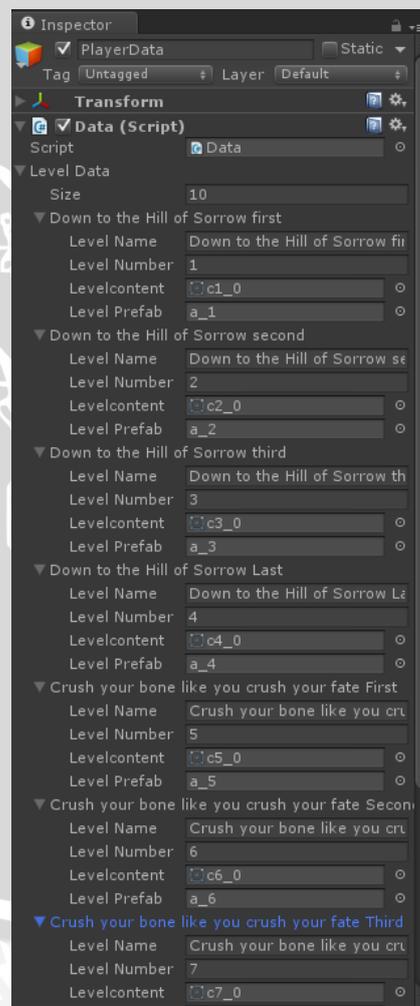
```

Data.cs
1  public class LevelSetting
2      {
3          public string levelName = null;
4          public int levelNumber;
5          public Sprite Levelcontent = null;
6          public GameObject LevelPrefab;
7      }
8  public LevelSetting[] levelData

```

Tabel 4.5 merupakan isi dari *data.cs* yang menyimpan *data-data prefab*. Baris 1 sampai baris 7 merupakan isi dari class *levelSetting*. Dalam class *levelSetting* terdapat 4 variabel yaitu :variabel *levelname*, *levelnumber*, *levelcontent*, dan *levelprefab*.

Variabel *levelname* dengan tipe *string* yang di tunjukkan pada baris 3. Pada baris 4 terdapat deklarasi variabel *levelNumber* dengan tipe *integer*, variabel ini berfungsi sebagai nomor level. Baris 5 terdapat variabel *levelContent* dengan tipe *sprite*, variabel akan di gunakan pada saat *screen pemilihan level*. Baris 8 merupakan deklarasi dari variabel *levelprefab* dengan tipe *gameobject*, variabel ini berisi *data prefab* yang akan di panggil pada setiap *challenge.Class levelSetting* nantinya akan di deklarasikan dengan menggunakan array seperti pada baris 8, yang mendeklarasikan variable *levelData* dengan tipe *LevelSetting*. Gambar 4.1 menunjukkan impementasi dari class data yang di lihat dari *inspector unity*.



Gambar 4.1 Implementasi class data

Pada saat pemain menyelesaikan level, waktu yang di dapat oleh pemain akan di *convert* ke ranking, jika nilai ranking yang di dapat kurang dari nilai E maka pemain tidak bisa membuka challenge selanjutnya. Data waktu dan ranking pemain di simpan pada *class player*. Selanjutnya di buat static function untuk mengakses class player dan menyimpan data waktu dan ranking. Tabel 4.6 menunjukkan pseudocode player menyimpan data waktu dan rank pada class player.

Tabel 4.6. Pseudocode player

<i>Pseudocode player</i>	
1	Int counter, level
2	Char rank
3	Float time
4	START
5	Static void point(level, rank, time, counter)
6	If(counter == 1)
7	Counter ++
8	CurrentLevel++
9	saveData()
10	END IF
11	END

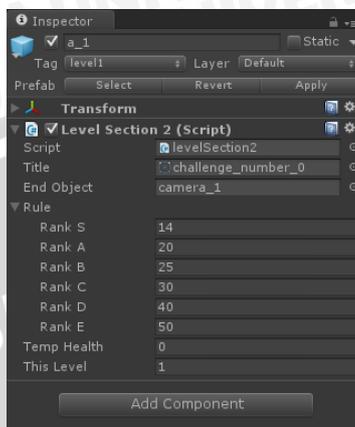
4.3. Realisasi Level

Dalam merealisasikan level pada game SDS ini, terdapat class yang menyimpan data pada setiap level. Data yang di simpan berupa data rule level yang di sebutkan pada rancangan level sebelumnya. Class tersebut di tampilkan pada Tabel 4.7.

Tabel 4.7. Rule level

<i>rule</i>	
1	public class Rule
2	{
3	public float rankS = 0;
4	public float rankA = 0;
5	public float rankB = 0;
6	public float rankC = 0;
7	public float rankD = 0;
8	public float rankE = 0;
9	}
10	public Rule _rule;

4.3.1. Challenge 1

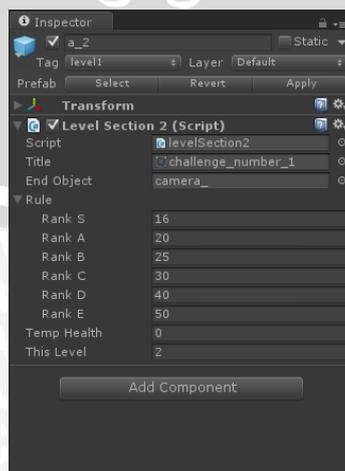


Gambar 4.2 Implementasi gameobject challenge 1

Gambar 4.2 menunjukkan implementasi dari prefab gameobject yang di panggil saat pemain memainkan challenge 1. Variabel *this Level* menunjukkan level yang di jalankan. Pada *rule* terdapat angka-angka yang menunjukkan batas rule untuk tiap rank. Saat di jalankan nilai *temp health* akan di ganti dengan nilai 100, nilai tersebut menunjukkan *health point* pemain.

4.3.2. Challenge 2

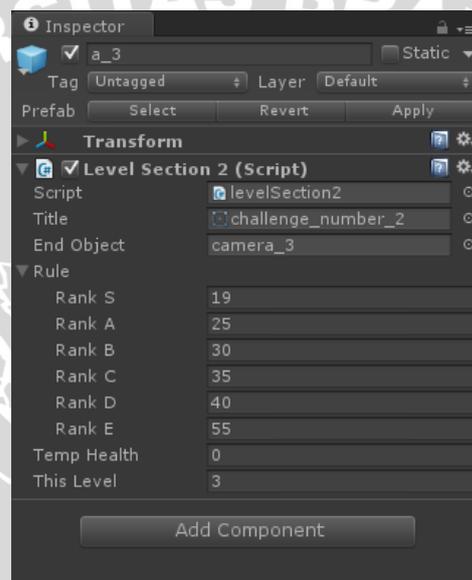
Gambar 4.3 merupakan gambar yang menunjukkan implementasi dari prefab gameobject a_2. Nilai rank pada challenge 2 untuk minimal nilai rank s di naikkan sedikit daripada challenge 1.



Gambar 4.3 Implementasi gameobject challenge 2

4.3.3. Challenge 3

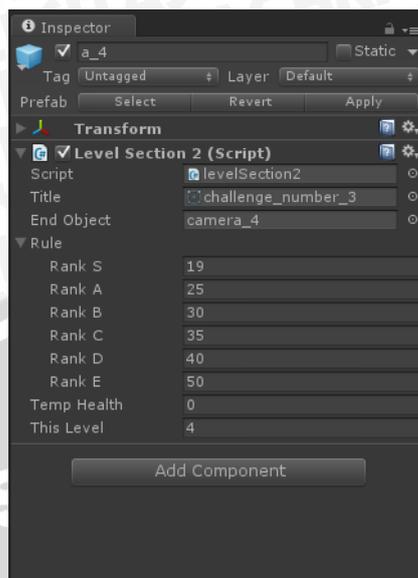
Pada challenge 3, nilai pada rule untuk menentukan nilai rank sedikit di naikan daripada challenge sebelumnya. Jumlah mobil yang ada pada gedung parkir ditambah untuk menambah kemungkinan pemain menabrak mobil, sehingga kesulitan pada challenge ini bertambah. This Level diisi dengan nilai 3 agar di ketahui bahwa yang di jalankan saat ini adalah challenge 3. Implementasi tersebut di tunjukkan pada gambar 4.4



Gambar 4.4 Implementasi gameobject challenge 3

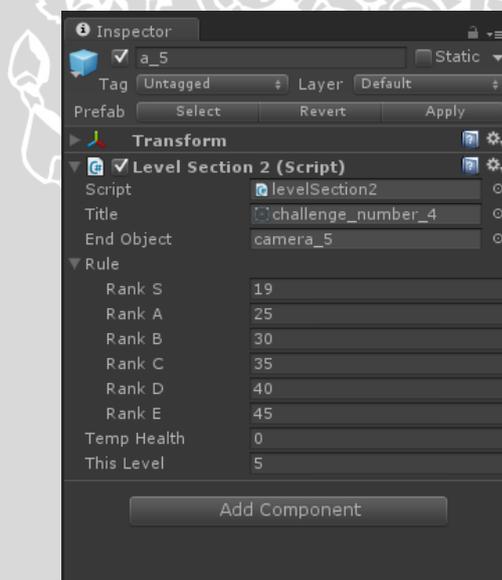
4.3.4. Challenge 4

Gambar 4.5 menunjukkan implementasi dari prefab gameobject yang di panggil saat pemain memainkan challenge 3. Pada *rule* terdapat angka-angka yang menunjukkan batas rule untuk tiap rank. *Variabel this Level* menunjukkan level yang di jalankan. Saat di jalankan nilai *temp health* akan di ganti dengan nilai 100, nilai tersebut menunjukkan *health point* pemain.



Gambar 4.5 Implementasi gameobject challenge 4

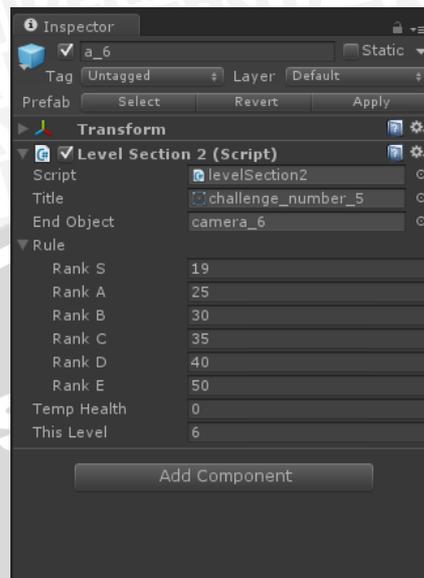
4.3.5. Challenge 5



Gambar 4.6 Implementasi gameobject challenge 5

Gambar 4.6 adalah implementasi dari gameobject *challenge 5*. *Challenge 5* merupakan *challenge* yang menggunakan parkir mundur agar dapat menyelesaikan *challenge* ini, tidak seperti *challenge 1* sampai *challenge 4* yang menggunakan parkir maju.

4.3.6. Challenge 6

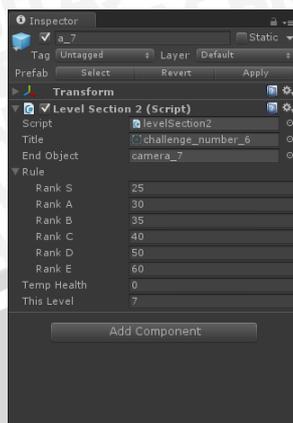


Gambar 4.7 Implementasi gameobject challenge 6

Pada *challenge* ini pemain menggunakan parkir mundur untuk menyelesaikan permainan. *This level* menunjukkan *challenge* yang sedang di jalankan. *End Object* di isi gameobject camera, saat pemain menyelesaikan permainan, object camera akan aktif. *Variabel title* merupakan variable yang di deklarasikan dengan tipe sprite. Sprite yang di isi berupa sprite title yang akan muncul selama challenge di jalankan. Challenge tersebut di implementasikan pada Gambar 4.7.

4.3.7. Challenge 7

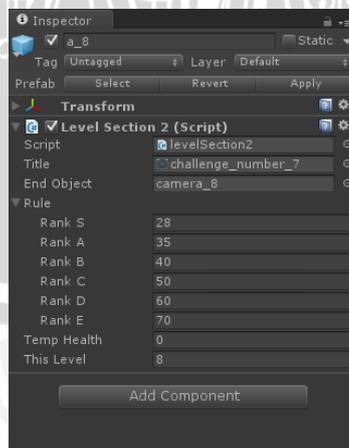
Gambar 4.8 menunjukkan implementasi dari prefab gameobject yang di panggil saat pemain memainkan challenge 7. Saat di jalankan nilai *temp health* akan di ganti dengan nilai 100, nilai tersebut menunjukkan *health point* pemain. Pada *rule* terdapat angka-angka yang menunjukkan batas rule untuk tiap rank. *Variabel this Level* menunjukkan level yang di jalankan. *End Object* di isi gameobject camera, saat pemain menyelesaikan permainan, object camera akan aktif.



Gambar 4.8 Implementasi gameobject challenge 7

4.3.8. Challenge 8

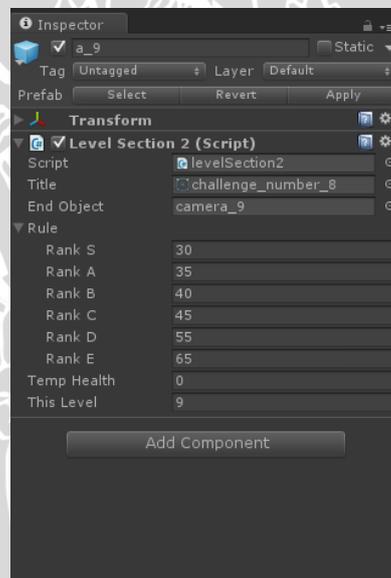
Pada *challenge* ini pemain menggunakan parkir mundur untuk menyelesaikan permainan. *This level* merupakan variabel bertipe integer menunjukkan *challenge* yang sedang di jalankan. *End Object* di isi gameobject camera, saat pemain menyelesaikan permainan, object camera akan aktif. *Variabel title* merupakan variable yang di deklarasikan dengan tipe sprite. Sprite yang di isi berupa sprite title yang akan muncul selama challenge di jalankan. Saat di jalankan nilai *temp health* akan di ganti dengan nilai 100, nilai tersebut menunjukkan *health point* pemain. Pada *rule* terdapat angka-angka yang menunjukkan batas rule untuk tiap rank. Challenge tersebut di implementasikan pada Gambar 4.9.



Gambar 4.9 Implementasi gameobject challenge 8

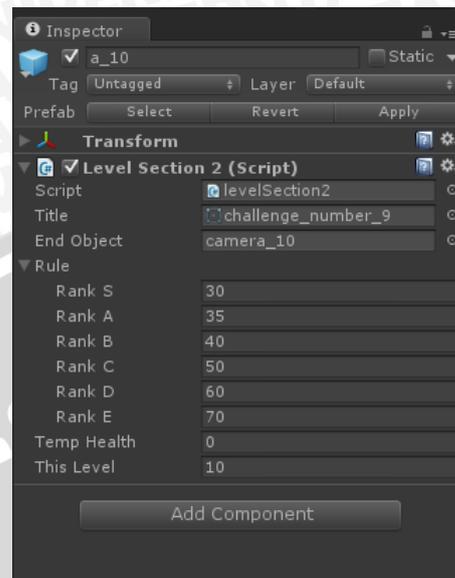
4.3.9. Challenge 9

Gambar 4.10 merupakan implementasi dari gameobject pada challenge 9. Pada challenge 9 lokasi permainan di pindah ke lantai 2. Untuk memarkir mobil pemain harus melewati rintangan berupa tanjakan. Rintangan tanjakan ini berada pada lokasi di depan pintu mesin parkir yang ada di lantai 2. Pemain harus berhenti di tengah – tengah tanjakan dan menahan mobil agar tidak turun ke bawah. Dalam waktu yang telah di tentukan pintu mesin parkir akan terbuka dan pemain dapat memarkir mobil, selanjutnya menyelesaikan permainan. Gambar 4.10 menampilkan implementasi dari game object challenge 9.



Gambar 4.10 Implementasi gameobject challenge 9

4.3.10. Challenge 10



Gambar 4.11 Implementasi gameobject challenge 10

Pada challenge ini lokasi tempat parkir sama seperti challenge 9 yaitu berada di lantai 2. Pada kolom this level menunjukkan angka 10, dimana angka ini menunjukkan nomor challenge tiap objek. Implementasi tersebut di tunjukkan pada gambar 4.11

4.4. Implementasi Arts

Berdasarkan pada konsep yang telah di buat pada bab sebelumnya, implementasi art meliputi : implementasi logo, implementasi HUD, implementasi art koin, implementasi art pendukung, dan implementasi dunia dalam game.

4.4.1. Implemeantasi Logo

Logo dari *game SDS* diimplementasikan sesuai konsep yang ada pada bab sebelumnya. Logo *game SDS* di buat dengan menggunakan aplikasi 3d modeller, yaitu Blender 2.7 dan font yang digunakan untuk logo adalah *NFS _font_unofficial*, font tersebut bisa di download di website penyedia font yang di sediakan secara *free*. Hasil Implementasi logo game di tunjukkan pada gambar 4.12.



Gambar 4.12 Implementasi logo SDS

4.4.2. Implementasi Head Up Display (HUD)

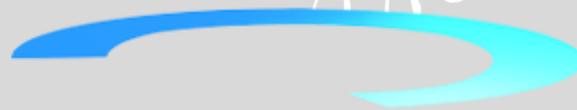
Implementasi HUD merupakan implementasi dari konsep HUD pada bab sebelumnya.



Gambar 4.13 Base dari HUD *bottom left*

Gambar 4.13 merupakan base dari HUD *bottom-left* yang berada pada. Base ini di gunakan sebagai background pada HUD *bottomleft*. Pada konsep HUD *bottom-left*, HUD ini berisi *speed indicator*, *gear indicator*, *throttle bar*, *brake bar*, *clutch bar*, dan juga *health bar*.

- a. **Throttle bar**, merupakan bar yang menunjukkan tekanan gas pada mobil, seperti Gambar 4.14.



Gambar 4.14 Implementasi *throttle bar*

- b. **Gear indicator**, merupakan indikator yang menunjukkan gear mobil. Indikator ini akan berganti gambar ketika gear pada mobil berganti. Art ini di tunjukkan dengan pergantian gear dari netral,

gear 1, gear 2, gear 3, gear 4, gear 5, dan gear R, seperti pada Gambar 4.15.



Gambar 4.15 Implementasi *gear indicator*

- c. **Speed indicator**, merupakan indikator yang di gunakan untuk menunjukkan kecepatan mobil dalam *game*, seperti pada gambar 4.16.



Gambar 4.16 Implementasi angka pada *speed indicator*

- d. **Brake bar**, merupakan bar yang menunjukkan tekana rem pada mobil, seperti pada Gambar 4.17.

Gambar 4.17 Implementasi brake bar

- e. **Clutch bar**, merupakan bar yang menunjukkan tekanan kopling pada mobil, seperti pada Gambar 4.18.

Gambar 4.18 Implementasi clutch bar

- f. **Health bar**, merupakan bar yang menunjukkan nyawa pemain. Bar akan aktif ketika pemain memasuki *challenge*, seperti pada Gambar 4.19.



Gambar 4.19 Implementasi health bar

Di sebelah bawah kanan layar terdapat HUD *bottom-right* yang menunjukkan kegiatan *sein lamp* dan *handbrake*. Pada gambar 4.20 ditunjukkan background yang di gunakan pada HUD *bottom-right*.

Gambar 4.20 Base dari HUD *bottom-right*

Pada HUD *bottom-right* yang berada pada kanan bawah layar, menunjukkan indikasi tombol lampu sein dan tombol *handbrake*.

- g. **Left Sein lamp**, menunjukkan indikator ketika lampu sein sebelah kiri di tekan. Gambar 4.21 menunjukkan implementasi dari konsep art *left sein lamp*.

Gambar 4.21 Left sein lamp sprite

- h. **Right Sein Lamp**, menunjukkan indikator ketika lampu sein sebelah kanan di tekan. Gambar 4.22 menunjukkan implementasi dari konsep art *right sein lamp*.

Gambar 4.22 Right sein lamp sprite

- i. **Handbrake**, menunjukkan indikator ketika tombol handbrake di tekan. Gambar 4.23 menunjukkan implementasi dari konsep art handbrake.



Gambar 4.23 Handbrake sprite

4.4.3. Implementasi art koin

Sesuai dengan konsep art koin, koin berbentuk lingkaran dengan logo *game* di permukaannya. Konsep *game* SDS adalah konsep *game* 3D, maka dari itu koin di beri warna emas mengkilat agar dapat mudah terlihat pada saat *game*. Gambar 4.24 menunjukkan uv dan texture yang di gunakan pada koin.



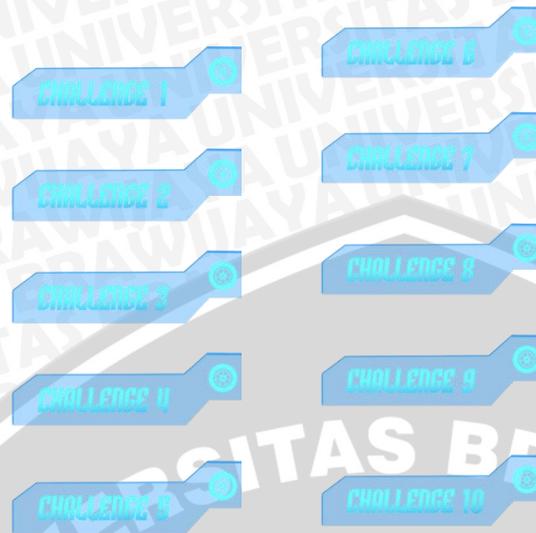
Gambar 4.24 uv dan texture koin

4.4.4. Implementasi Art Pendukung

Terdapat juga art tambahan yang mendukung dalam pengembangan *game* SDS ini

a. Challenge title

Art ini merupakan art yang menunjukkan nama *challenge* yang sedang berlangsung. Art ini hanya akan muncul pada saat pemain bermain di *challenge*. Implementasi dari challenge title di tunjukkan pada Gambar 4.25.



Gambar 4.25 Implementasi dari challenge title

4.4.5. Implementasi dunia dalam game

Implementasi dunia dalam game dibuat dengan konsep 3D menggunakan aplikasi 3d modeller *Blender* dengan versi 2.7, namun untuk pembuatan pohon dan air menggunakan aset yang sudah di sediakan *Unity3D*. Pada kota game ini terdapat gedung-gedung yang saling berjajar, pom bensin, taman, gedung parkir, dan juga garasi. Hasil aset yang telah di buat di implementasikan pada gambar 4.26.



Gambar 4.26 Implementasi dunia game

4.5. Implementasi Game Screen

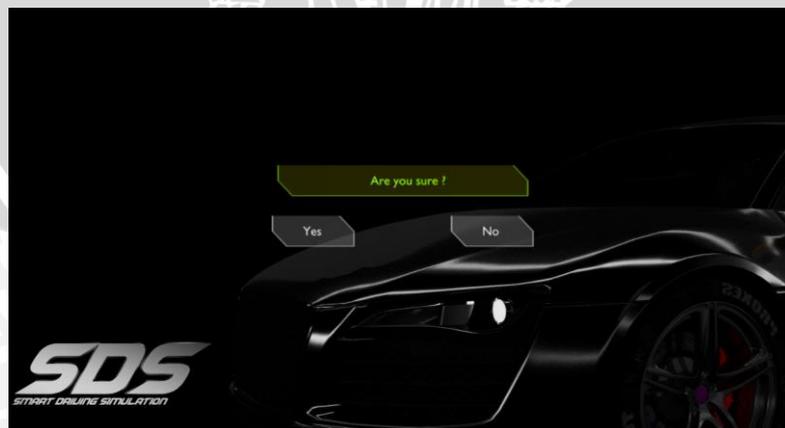
Berdasarkan pada perancangan, terdapat 9 *game screen* yang ada pada *game SDS* ini.

4.5.1. Screen Menu Utama

Gambar 4.27 merupakan implementasi dari mainmenu utama. *Game SDS* memiliki 4 menu pada mainmenu utama, yaitu : *start*, *option*, *credit*, dan *quit*. Menu *start* di gunakan untuk berpindah ke *screen game* dimana pemain memulai permainan. Menu *option* adalah menu yang di gunakan pemain untuk melihat dan mengganti settingan *game*. Menu *credit* berisi tentang profil dari *developer game SDS*. Menu *quit* di gunakan untuk mengakhiri *game SDS*, seperti pada Gambar 4.28.



Gambar 4.27 Implementasi mainmenu



Gambar 4.28 Implementasi menu quit

4.5.2. Screen Game

Gambar 4.29 menunjukkan implementasi dari screen game. Pada *scene game*, HUD akan ditampilkan pada layar *game*. HUD utama berada pada sudut bawah layar *game*. HUD utama ini menunjukkan aktifitas – aktifitas yang dilakukan pemain. Pada sudut kiri bawah terdapat HUD yang menunjukkan kecepatan kendaraan, tekanan pijakan gas, tekanan pijakan kopling, tekanan pijakan rem, dan *gear*. Dan pada sudut kanan bawah adalah HUD yang menunjukkan lampu sein dan *handbrake*.



Gambar 4.29 Implementasi layar *game*

4.5.3. Screen Garage



Gambar 4.30 Implementasi layar *Garage*

Pada gambar 4.30 menunjukkan implementasi dari screen garage, dimana screen ini dapat di temui pemain apabila pemain memasuki garasi. Terdapat 4 menu dalam screen ini, dimulai dari kiri, yaitu : *menu back to street*, *menu record*, *menu device* dan *menu exit*.

4.5.4. Screen level

Pada bab sebelumnya yang membahas tentang adanya koin yang berada di depan pintu mesin parkir *game*. Koin tersebut bila di ambil, maka akan mengaktifkan *trigger* menuju screen level. Pada screen level terdapat informasi dan petunjuk tentang level yang akan di mainkan oleh pemain, juga terdapat 2 tombol pilihan yang dapat di tekan dengan menggunakan perangkat keras *mouse*, tombol tersebut adalah tombol pilihan *yes* dan *no*. Tombol pilihan tersebut menentukan apakah pemain menerima *challenge* yang akan di berikan atau tidak. Gambar 4.31 menunjukkan implementasi screen level.

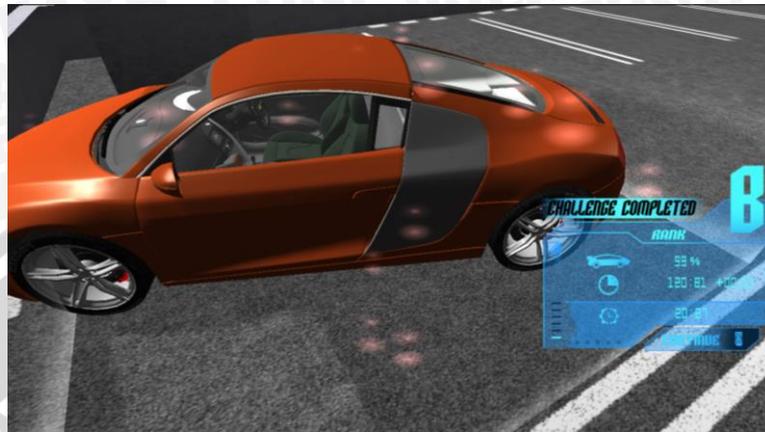


Gambar 4.31 Implementasi screen level

4.5.5. Screen result

Saat *challenge* berlangsung waktu akan di hitung dan *health bar* akan di aktif kan. Waktu akan mulai di hitung ketika pemain melewati pintu mesin parkir dan *health* akan berkurang jika pemain menabrak. Setelah pemain berhasil

memarkir mobil pada tempat yang sudah di tentukan, maka otomatis akan berhenti dan akan menemui *screen result* dengan kondisi pemain menyelesaikan *challenge*. *Screen result* tersebut dapat di lihat pada Gambar 4.32.



Gambar 4.32 Implementasi screen result saat kondisi menang

Health yang di miliki pemain hanya 100 point, dan akan berkurang jika pemain menabrak sesuatu. Apabila *health* pemain berkurang sampai mencapai 0 point, maka mobil akan berhenti dan pemain di anggap gagal dalam menyelesaikan *challenge* yang di berikan *game*. *Screen* gagal di implementasikan pada Gambar 4.33.



Gambar 4.33 Implementasi screen result saat kondisi gagal

4.5.6. Screen Allcompleted

Screen ini merupakan screen yang menunjukkan bahwa pemain telah menyelesaikan seluruh challenge yang terdapat pada game. Screen ini bisa di temui pemain ketika pemain menyelesaikan ke-10 challenge yang terdapat pada game atau dengan ketentuan tidak ada nilai rank E yang di dapat pemain saat menyelesaikan setiap challenge. Setelah pemain menyelesaikan ke-10 *challenge*, object bintang yang melayang akan muncul di depan garasi. Jika pemain menabrak bintang tersebut, pemain akan mendapatkan *screen allcompleted* ini.



Gambar 4.34 Implementasi screen allcompleted

BAB V

PENGUJIAN

Bab ini membahas mengenai tahapan pengujian dan analisis *game Smart Driving Simulation* yang telah dibangun. Pengujian berfungsi untuk melihat sejauh mana permainan berjalan dan untuk menemukan kesalahan yang terdapat pada permainan. Tahap pengujian *game Smart Driving Simulation* ini terdiri dari perancangan pengujian (*test suite*) dan *playtesting*.

5.1.Design Test (*Test Suite*)

Test suite adalah sekumpulan kasus uji (*test case*) yang dideskripsikan lebih detail. *Test suite* memberikan langkah-langkah instruksi tentang operasi apa yang harus dilakukan pada game dan detail apa yang harus dicek sebagai hasil dari setiap langkah. Kasus uji dalam *test suite* berupa sebuah pertanyaan dimana penguji harus melakukan pengujian pada permainan untuk menjawab pertanyaan tersebut. Pertanyaan dalam *test suite* sebaiknya dibuat dengan jawaban berupa “Ya” atau “Tidak” dimana jawaban “Ya” mengindikasikan bahwa aplikasi telah bekerja sesuai keinginan dan tidak ada kesalahan yang ditemui sedangkan jawaban “Tidak” mengindikasikan bahwa ada permasalahan pada aplikasi dan ada kesalahan yang harus dicatat.

Berikut adalah pertanyaan pada kuisioner menggunakan *test suite* yang akan dipakai untuk menguji *game Smart Driving Simulation*:

Jalankan SDS

- 1 Apakah menu dapat di tampilkan ?
- 2 Apakah menu Start dapat bekerja ?
- 3 Apakah menu Options dapat bekerja ?
- 4 Apakah menu Credit dapat bekerja ?

In-Game

- 1 Apakah HUD di tampilkan
- 2 Art HUD ok ?
- 3 Apakah animasi gear bekerja ?
- 4 Apakah animasi speedometer bekerja
- 5 Apakah animasi throttle (gas) bar bekerja ?
- 6 Apakah animasi brake (rem) bar bekerja ?
- 7 Apakah animasi clutch (kopling) bar bekerja ?
- 8 Apakah animasi lampu sein bekerja ?
- 9 Apakah animasi handbrake bekerja ?

- 10 Apakah steering bekerja ?
 - 11 Animasi Koin bekerja ?
 - 12 Art world ok ?
 - 13 Apakah hardware steer bisa di putar ?
 - 14 Apakah sein button sebelah kiri bekerja ?
 - 15 Apakah sein button sebelah kanan bekerja ?
 - 16 Apakah pedal gas bekerja ?
 - 17 Apakah pedal rem bekerja ?
 - 18 Apakah pedal kopling bekerja ?
 - 19 Apakah gear stick untuk gear 1 bekerja ?
 - 20 Apakah gear stick untuk gear 2 bekerja ?
 - 21 Apakah gear stick untuk gear 3 bekerja ?
 - 22 Apakah gear stick untuk gear 4 bekerja ?
 - 23 Apakah gear stick untuk gear 5 bekerja ?
 - 24 Apakah gear stick untuk gear 6 bekerja ?
- In-game saat challenge completed
- 1 Muncul result ?
 - Animasi camera berputar
 - 2 Art result ok ?
 - 3 Animasi ketika result keluar ?
 - 4 Muncul health result ?
 - 5 Muncul nilai waktu pemain ?
 - 6 Muncul tambahan nilai waktu pemain ketika health < 100 ?
 - 7 Muncul total waktu ?
 - 8 Muncul rank pemain ?
 - 9 Muncul instruksi continue...
- In-game saat challenge failed
- 1 Muncul art failed ?
 - 2 Art ok ?
 - 3 Animasi failed keluar ?
 - 4 Animasi instuksi continue ?

Pada pengujian menggunakan test suite, terdapat 20 kuisisioner yang di bagikan dan tidak ditemukan jawaban tidak pada kuisisioner, sehingga fitur-fitur yang di ujikan dalam kuisisioner tidak terjadi masalah atau berjalan sesuai yang di harapkan. Pada *in-game* pertanyaan 13 sampai dengan pertanyaan 24 merupakan pertanyaan yang terkait dengan hardware steering wheel. 20 responden menjawab ya untuk bagian yang berkaitan dengan steering wheel, sehingga kontrol *game smart driving simulation* menggunakan hardware steering wheel berjalan dengan baik. Tabel 5.2 menunjukkan hasil pengujian menggunakan test suite.

Tabel 5.1. Hasil pengujian test suite

No.	Step	Tidak	Ya
Jalankan SDS			
1	Apakah menu dapat di tampilkan ?	0	20
2	Apakah menu Start dapat bekerja ?	0	20
3	Apakah menu Options dapat bekerja ?	0	20
4	Apakah menu Credit dapat bekerja ?	0	20
5	Apakah menu Exit dapat bekerja ?	0	20
6	Animasi menu hover bekerja	0	20
7	Muncul logo SDS ?	0	20
8	Art menu ok?	0	20
9	Animasi SDS setelah exit ?	0	20
In-Game			
1	Apakah HUD di tampilkan	0	20
2	Art HUD ok ?	0	20
3	Apakah animasi gear bekerja ?	0	20
4	Apakah animasi speedometer bekerja	0	20
5	Apakah animasi throttle (gas) bar bekerja ?	0	20
6	Apakah animasi brake (rem) bar bekerja ?	0	20
7	Apakah animasi clutch (kopling) bar bekerja ?	0	20
8	Apakah animasi lampu sein bekerja ?	0	20
9	Apakah animasi handbrake bekerja ?	0	20
10	Apakah steering bekerja ?	0	20
11	Animasi Koin bekerja ?	0	20
12	Art world ok ?	0	20
13	Apakah hardware steer bisa di putar ?	0	20
14	Apakah sein button sebelah kiri bekerja ?	0	20
15	Apakah sein button sebelah kanan bekerja ?	0	20
16	Apakah pedal gas bekerja ?	0	20
17	Apakah pedal rem bekerja ?	0	20
18	Apakah pedal kopling bekerja ?	0	20
19	Apakah gear stick untuk gear 1 bekerja ?	0	20
20	Apakah gear stick untuk gear 2 bekerja ?	0	20
21	Apakah gear stick untuk gear 3 bekerja ?	0	20
22	Apakah gear stick untuk gear 4 bekerja ?	0	20
23	Apakah gear stick untuk gear 5 bekerja ?	0	20
24	Apakah gear stick untuk gear 6 bekerja ?	0	20
Scene pemilihan challenge			
1	Effect fade saat masuk ke scene	0	20

2	Art pemilihan level ok ?	0	20
3	Hover button yes dan no bekerja ?	0	20
4	Muncul content tentang level ?	0	20
5	Effect fade saat keluar dari scene ?	0	20
In-Game saat challenge			
1	Effect fade saat masuk scene ?	0	20
2	Muncul challenge title ?	0	20
3	Art challenge title	0	20
4	Hardware steer bergetar saat menabrak	0	20
5	Animasi pintu mesin parker	0	20
6	Rintang dan mobil muncul di world ?	0	20
7	Effect particle pada lokasi tempat parkir ?	0	20
8	Apakah HUD di tampilkan ?	0	20
9	Apakah animasi health bekerja ?	0	20
10	Apakah animasi gear bekerja ?	0	20
11	Apakah animasi speedometer bekerja	0	20
12	Apakah animasi throttle (gas) bar bekerja ?	0	20
13	Apakah animasi brake (rem) bar bekerja ?	0	20
14	Apakah animasi clutch (kopling) bar bekerja ?	0	20
15	Apakah animasi lampu sein bekerja ?	0	20
16	Apakah animasi handbrake bekerja ?	0	20
17	Apakah steering bekerja ?	0	20
18	Animasi Koin bekerja ?	0	20
19	Art world ok ?	0	20
20	Apakah hardware steer bisa di putar ?	0	20
21	Apakah sein button sebelah kiri bekerja ?	0	20
22	Apakah sein button sebelah kanan bekerja ?	0	20
23	Apakah pedal gas bekerja ?	0	20
24	Apakah pedal rem bekerja ?	0	20
25	Apakah pedal kopling bekerja ?	0	20
26	Apakah gear stick untuk gear 1 bekerja ?	0	20
27	Apakah gear stick untuk gear 2 bekerja ?	0	20
28	Apakah gear stick untuk gear 3 bekerja ?	0	20
29	Apakah gear stick untuk gear 4 bekerja ?	0	20
30	Apakah gear stick untuk gear 5 bekerja ?	0	20
31	Apakah gear stick untuk gear 6 bekerja ?	0	20
In-game saat challenge completed			
1	Muncul result ?	0	20
	Animasi camera berputar	0	20
2	Art result ok ?	0	20

3	Animasi ketika result keluar ?	0	20
4	Muncul health result ?	0	20
5	Muncul nilai waktu pemain ?	0	20
6	Muncul tambahan nilai waktu pemain ketika health < 100 ?	0	20
7	Muncul total waktu ?	0	20
8	Muncul rank pemain ?	0	20
9	Muncul instruksi continue...	0	20
In-game saat challenge failed			
1	Muncul art failed ?	0	20
2	Art ok ?	0	20
3	Animasi failed keluar ?	0	20
4	Animasi instuksi continue ?	0	20

5.2. Play Testing

Play testing dilakukan saat *game* sudah dirasa dalam keadaan *stable*. Fokus yang ditujukan dalam permainan ini adalah *feedback* pengguna tentang *gameplay*, tampilan *art*, penyajian, dan lain sebagainya yang dapat dilihat pada *game* secara global. Kuisisioner *feedback* telah disediakan sebelumnya, sehingga pemain dapat memberi tanggapan sesuai topik yang diberikan pada kuisisioner. Adapun para penguji *game SDS* yang telah release setelah dilakukan design test ini ialah penguji yang sama yang telah melakukan pengujian berdasarkan *test suite* yang diberikan. Berikut adalah pertanyaan yang tersedia pada kuisisioner *playtesting*.

1. Bagaimana dengan *gameplay* yang disajikan ?
2. Bagaimana tentang efisiensi saat memainkan ?
3. Bagaimana dengan tingkat Kesulitan *game* ?
4. Seberapa "Fun" kah anda memainkan *game SDS*?
5. Bagaimana dengan *art* yang disajikan dalam *game* ?

Hasil yang didapat dari pengujian *play testing*, dari 20 kuisisioner yang di bagikan memiliki hasil yang cukup memuaskan. 20 kuisisioner terisi dengan jawaban bagus dan cukup bagus. Tabel 5.4 merupakan tabel yang menunjukkan hasil dari *playtesting*.

Tabel 5.2. Hasil kuisisioner *play testing*

No.	Pertanyaan	Sangat Kurang	Kurang	Cukup	Bagus	Sangat Bagus
1.	Bagaimana dengan gameplay yang disajikan ?				10	10
Saran tentang Gameplay* :						
2.	Bagaimana tentang efisiensi saat memainkan ?			5	12	7
Saran tentang efisiensi game* :						
3.	Bagaimana dengan tingkat Kesulitan game ?			2	10	8
Saran tentang level game* :						
4.	Seberapa “Fun” kah anda memainkan game SDS?			2	5	13
Saran tentang tingkat fun dalam game* :						
5.	Bagaimana dengan art yang disajikan dalam game ?				5	15
Saran tentang art dalam game* :						
6.	Manfaat Game Menurut Anda :					

Dari hasil pengujian playtesting diatas maka di dapatkan hasil sebagai berikut :

1. Pada pertanyaan “bagaimana dengan gameplay yang di sajikan”, dari 20 orang sebanyak 10 orang atau sekitar 50% responden memilih bagus dan 10 orang lain menjawab gameplay dari game sangat bagus

2. Pada pertanyaan kedua “bagaimana tentang efisiensi saat memainkan ?”, dari 20 orang 1 responden atau 5% memilih cukup, sebanyak 12 orang atau 60 % responden memilih bahwa efisiensi *game SDS* bagus, dan sebanyak 7 orang memilih.
3. Pada pertanyaan “bagaimana dengan tingkat kesulitan game ?”, dari 20 orang responden 2 orang atau 10% responden memilih tingkat kesulitan *game* sudah cukup. Sebanyak 10 orang atau 50% responden memilih bahwa kesulitan game sudah bagus, dan sebanyak 8 orang atau 40% responden memilih sudah sangat bagus.
4. Pada pertanyaan keempat “seberapa fun kah anda memainkan game SDS ?”, dari 20 orang responden 2 orang atau 10% dari keseluruhan responden memilih bahwa game sudah cukup fun. Sebanyak 5 orang atau 25% responden memilih bagus, dan 13 orang atau 65% dari keseluruhan responden memilih bahwa game ini sudah sangat menyenangkan.
5. Pada pertanyaan “ bagaimana art dan dunia yang di sajikan dalam game ? “ , 5 orang dari 20 responden memilih art dan dunia pada game sudah bagus. Sebanyak 15 orang atau 75% dari keseluruhan responden memilih bahwa game ini memiliki art dan dunia yang sangat bagus.

5.3. Hasil Analisa

Dari hasil *test suite* yang dilakukan telah di temukan kesalahan. Kesalahan ini terjadi pada fitur yang tidak di ujikan pada lembar kuisisioner yang di edarkan. Kesalahan tersebut terjadi ketika pemain berada pada *challenge*. Pemain menabrak sesuatu saat challenge akan mengurangi health point pemain. Health yang menjadi 0 akan membuat pemain gagal dalam challenge. Kemudian akan terlihat screen gagal di layar. Untuk keluar dari screen gagal dan kembali ke gameplay, pemain harus memasukkan gear ke posisi satu kemudian menginjak pedal gas. Saat keluar dari screen gagal title challenge keluar dari layar, namun saat playtesting title challenge tidak keluar layar dan tetap tampil di bagian atas layar. Kesalahan ini terjadi akibat sprite title tidak melakukan animasi yang

seharusnya di jalankan. Solusi yang di lakukan adalah menambah baris baru untuk memanggil animasi selanjutnya, seperti pada Gambar 5.1.

```
253     if (levelSection2.Die)
254     {
255         UiFail.showing(false);
256         UiTitle.spriteBackward();|
257         levelSection2.Die = false;
258     }
259     else
260     {
261         UiRank.hide();
262         UiHealthResult.hide();
263         UiTimeExtend.hide();
264         UiTimeResult.hide();
265         UiFinalResult.hide();
266         UiResult.spriteBackward();
267         UiButtonContinue.spriteBackward();
268     }
```

Gambar 5.1 Penambahan baris baru

Penambahan baris baru di lakukan pada *class systemmanager* pada baris ke 256. Baris tersebut di gunakan untuk menjalankan animasi sprite agar keluar dari layar saat keluar dari screen gagal.

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

1. Perancangan permainan simulasi ujian berkendara 3D dibuat berdasarkan hasil analisa kebutuhan. Selanjutnya, informasi tersebut di gunakan untuk membuat Deskripsi *Game*, *Gameplay*, Rancangan Level, Konsep Art, dan *Game Screen Flow*. Pada implementasi dibagi lagi menjadi dua sistem kerja objek integrasi yaitu di luar dan di dalam *game engine (Unity 3d)*. Pengujian yang di lakukan pada penelitian ini menggunakan *design test* dan *play testing*.
2. *Game Smart Driving Simulation* merupakan permainan 3D single player bergenre simulator dengan konten edukasi membahas tentang ujian berkendara dengan kategori parker seri maju, parker seri mundur, dan tanjakan.
3. Pada pengujian menggunakan *playtesting*, terdapat 20 orang responden yang mengisi 5 pertanyaan pada kuisisioner dengan hasil 5 orang yang memilih pilihan cukup, 42 orang memilih bagus, dan 53 orang memilih sangat bagus.
4. Pada pengujian menggunakan *design test*, 20 orang responden menjawab ya pada setiap pertanyaan pada kuisisioner. Hal ini menunjukkan bahwa tidak ditemukan kesalahan, sehingga dapat di katakan bahwa fitur-fitur yang terdapat pada game SDS 100% berhasil atau berjalan dengan baik. Terdapat 24 pertanyaan yang terkait dengan hardware steering wheel dan seluruh responden menjawab ya pada kuisisioner, sehingga hardware steering wheel berjalan baik pada *game smart driving simulator*.

6.2.Saran

1. Untuk pengembangan dari *game* dapat dilakukan dengan Penambahan materi ujian berkendara seperti zigzag dan juga parkir paralel
2. Pengembangan *game* ini juga dapat dilakukan secara bebas, namun ada baiknya untuk tetap mencantumkan nama pembuat sebelumnya agar dapat dijadikan referensi untuk pengembangan *game* kedepannya
3. Dalam membuat sebuah pemodelan lingkungan / *terrain* yang besar ada perlunya menggunakan teknik *culling* untuk meningkatkan performa permainan.
4. Perhitungan kecepatan yang mirip dengan realita dapat di jadikan pada pengembangan selanjutnya.



DAFTAR PUSTAKA

- AME-12 Amelia , Emi. 2012. Inilah yang di ujikan di Tes Simulator . diakses dari : <http://news.detik.com/read/2012/07/31/160254/1979548/10/inilah-yang-diujiikan-di-tes-simulator-sim> pada tanggal 2-12-2014
- CHU-10 Chu, Philip. 2010. *Game Development with Unity*
- FLA-10 Flavell, Lance. 2010. *Beginning Blender: Open Source 3D Modeling, Animation, and Game Design*. USA : Paul Manning.
- HER-13 Heru Irawan, Kukuh. 2013. “RANCANG BANGUN PERMAINAN SIMULASI BERKENDARA 3D PADA PLATFORM PC”. PTIIK.
- KUM-14 Kumpulan Ujian Teori Sim. 2014. Ujian Praktik SIM mobil dengan Gambar. Diakses dari : <http://ujiansim-online.blogspot.com/2014/11/ujian-praktek-sim-disertai-gambar-video.html>. Tanggak akses 28-12-2014.
- JON-12 Jonathan, William. 2012. *Character Development in Blender 2.5*. Course Technology Cengage Learning.
- Opini Opini. 2013. kasus-simulator-sim. Diakses dari : <http://www.merdeka.com/tag/k/kasus-simulator-sim/> diakses tanggal 18-12-2014.
- ROG-10 Roger, Scott. 2010. *Level Up! The Guide to Great Video Games*. West Sussex : John Wiley & Sons, Ltd
- SMI-13 Smith, Matt. Queiroz, Chico. 2013. *Unity 4.x Cookbook*. Birmingham, Mumbai. Packt Publishing, Ltd
- TEC-12 Technologies, unity. 2012. “*The Car TuTorial*” – *Creating Racing Game for Unity* . Unity Technologies
- UUR-09 Undang-Undang Republik Indonesia. 2009. Undang-Undang Republik Indonesia Nomor 22 Tahun 2009 Tentang Lalu Lintas Dan Angkutan Jalan
- ZAD-12 Zaddo67. 2012. *FORCE FEEDBACK TOOLKIT*.

LAMPIRAN



Test Suite

Nama :

Jawaban hanya meliputi “Ya” yang berarti sudah sesuai, atau “Tidak” yang berarti fungsi tersebut gagal atau perlu di perbaiki. Jawab dengan tanda centang (√) pada salah satu kolom.

No.	Step	Tidak	Ya	Tanggapan
Jalankan SDS				
1	Apakah menu dapat di tampilkan ?			
2	Apakah menu Start dapat bekerja ?			
3	Apakah menu Options dapat bekerja ?			
4	Apakah menu Credit dapat bekerja ?			
5	Apakah menu Exit dapat bekerja ?			
6	Animasi menu hover bekerja			
7	Muncul logo SDS ?			
8	Art menu ok?			
9	Animasi SDS setelah exit ?			

In-Game				
1	Apakah HUD di tampilkan			
2	Art HUD ok ?			
3	Apakah animasi gear bekerja ?			



4	Apakah animasi speedometer bekerja			
5	Apakah animasi throttle (gas) bar bekerja ?			
6	Apakah animasi brake (rem) bar bekerja ?			
7	Apakah animasi clutch (kopling) bar bekerja ?			
8	Apakah animasi lampu sein bekerja ?			
9	Apakah animasi handbrake bekerja ?			
10	Apakah steering bekerja ?			
11	Animasi Koin bekerja ?			
12	Art world ok ?			
13	Apakah hardware steer bisa di putar ?			
14	Apakah sein button sebelah kiri bekerja ?			
15	Apakah sein button sebelah kanan bekerja ?			
16	Apakah pedal gas bekerja ?			
17	Apakah pedal rem bekerja ?			
18	Apakah pedal kopling bekerja ?			
19	Apakah gear stick untuk gear 1 bekerja ?			
20	Apakah gear stick untuk gear 2 bekerja ?			
21	Apakah gear stick untuk gear 3 bekerja ?			
22	Apakah gear stick untuk gear 4 bekerja ?			
23	Apakah gear stick untuk gear 5 bekerja ?			
24	Apakah gear stick untuk gear 6 bekerja ?			

Scene pemilihan challenge

1	Effect fade saat masuk ke scene			
2	Art pemilihan level ok ?			
3	Hover button yes dan no bekerja ?			
4	Muncul content tentang			

	level ?			
5	Effect fade saat keluar dari scene ?			

In-Game saat challenge				
1	Effect fade saat masuk scene ?			
2	Muncul challenge title ?			
3	Art challenge title ok ?			
4	Hardware steer bergetar saat menabrak			
5	Animasi pintu mesin parkir			
6	Rintangan dan mobil muncul di world ?			
7	Effect particle pada lokasi tempat parkir ?			
8	Apakah HUD di tampilkan ?			
9	Apakah animasi health bekerja ?			
10	Apakah animasi gear bekerja ?			
11	Apakah animasi speedometer bekerja			
12	Apakah animasi throttle (gas) bar bekerja ?			
13	Apakah animasi brake (rem) bar bekerja ?			
14	Apakah animasi clutch (kopling) bar bekerja ?			
15	Apakah animasi lampu sein bekerja ?			
16	Apakah animasi handbrake bekerja ?			
17	Apakah steering bekerja ?			
18	Animasi Koin bekerja ?			
19	Art world ok ?			
20	Apakah hardware steer bisa di putar ?			
21	Apakah sein button sebelah kiri bekerja ?			
22	Apakah sein button sebelah kanan bekerja ?			
23	Apakah pedal gas bekerja ?			
24	Apakah pedal rem bekerja ?			
25	Apakah pedal kopling			

	bekerja ?			
26	Apakah gear stick untuk gear 1 bekerja ?			
27	Apakah gear stick untuk gear 2 bekerja ?			
28	Apakah gear stick untuk gear 3 bekerja ?			
29	Apakah gear stick untuk gear 4 bekerja ?			
30	Apakah gear stick untuk gear 5 bekerja ?			
31	Apakah gear stick untuk gear 6 bekerja ?			

In-game saat challenge completed				
1	Muncul result ?			
	Animasi camera berputar			
2	Art result ok ?			
3	Animasi ketika result keluar ?			
4	Muncul health result ?			
5	Muncul nilai waktu pemain ?			
6	Muncul tambahan nilai waktu pemain ketika health < 100 ?			
7	Muncul total waktu ?			
8	Muncul rank pemain ?			
9	Muncul instruksi continue...			

In-game saat challenge failed				
1	Muncul art failed ?			
2	Art ok ?			
3	Animasi failed keluar ?			
4	Animasi instuksi continue ?			



Playtesting

Nama :

NIM :

*** dimohon untuk tidak mengisi dengan gurauan dan harap diisi dengan sebenarnya – benarnya.

No.	Pertanyaan	Sangat Kurang	Kurang	Cukup	Bagus	Sangat Bagus
1.	Bagaimana dengan gameplay yang disajikan ?					
Saran tentang Gameplay* :						
2.	Bagaimana tentang efisiensi saat memainkan ?					
Saran tentang efisiensi game* :						
3.	Bagaimana dengan tingkat Kesulitan game ?					
Saran tentang level game* :						
4.	Seberapa “Fun” kah anda memainkan game SDS?					



Saran tentang tingkat fun dalam game* :					
5.	Bagaimana dengan art dan dunia yang disajikan dalam game ?				
Saran tentang art dalam game* :					
6.	Manfaat Game Menurut Anda :				

Ctt :

- **Art : design object 2D dan tampilan game**
- **Dunia : design object 3D world pada game**
- **Isi dengan tanda “X” pada tiap kolom**
- *** : silahkan di kosongi jika tidak ada**

Saran lain yang meliputi Game Smart Driving Simulation :