KATA PENGANTAR

Alhamdulillah,

Segala puji syukur penulis atas kehadirat Allah SWT dengan seluruh limpahan rahmat serta ridho-Nya, penulis dapat menyelesaikan skripsi ini. Dengan terselesaikannya skripsi ini tentu juga ada dukungan berupa semangat dan doa dari orang-orang yang berada di sekeliling penulis yang telah terlibat secara langsung maupun tidak langsung. Oleh karena itu penulis mengucapkan terima kasih yang tidak terhingga kepada:

1. Kepada orang tua, khususnya Ibu penulis yang bernama Latifah yang terus menerus menyemangati dan memberi dukungan penulis untuk menyelesaikan skripsi ini tanpa mengenal lelah.
2. Kepada Dosen Pembimbing Eriq M. Adams J., S.T., M.Kom. dan Denny Sagita Rusdianto, S.kom, M.Kom. yang tanpa menyerah telah membimbing penulis selama ini. Serta tidak lupa mengucapkan terima kasih kepada seluruh dosen dan para staff Program Teknologi Informasi dan Ilmu Komputer dalam membantu penulis dalam menyelesaikan skripsi ini.
3. Kepada asisten baru Lab. Game, Mury dan kakak-kakak tingkat Kurnia Prima, Febri Abdullah, dan M. Aminul Akbar, penulis berterimakasih atas bantuan yang tak dapat dihitung dalam menyelesaikan skripsi ini.
4. Kepada teman seperjuangan yang telah membantu dalam menyelesaikan skripsi ini. Tidak lupa kepada Ririn, Anisa, Tian, Zahrul, Fia, Fuad, Alfi, Geri, Ayis dan seluruh teman angkatan penulis yang secara langsung maupun tidak dalam menyemangati penulis untuk tetap berusaha dalam menyelesaikan skripsi pada semester ini.
5. Kepada seluruh teman SMA penulis, khususnya Dharing, Kiky, Erste dan Indra yang telah mendukung dan selalu ada pada ketika penulis membutuhkan bantuan dalam hal pembuatan aplikasi maupun pada saat menyusun laporan skripsi ini.

6. Tak lupa kepada seluruh teman kontrakan penulis, Hamdan, Jimmy, Adi, Nanda, Cahyo, Adit, Tatas dan yang lainnya yang telah memberikan dukungan kepada penulis.

Semoga Allah SWT membalas kebaikan dan ketulusan semua pihak yang telah membantu menyelesaikan skripsi ini dengan melimpahkan rahmat dan karunia-Nya dan semoga karya penelitian tugas akhir ini dapat memberikan manfaat dan kebaikan bagi banyak pihak demi kemaslahatan bersama serta bernilai ibadah di hadapan Allah SWT. Amin.

Malang, Januari 2015

Penulis

ABSTRAK

Saiful Setyawan, 2015. Rancang Bangun Game Berdasarkan Cerita Rakyat Timun Emas. Skripsi Program Studi Teknik Informatika. Program Teknologi Informasi dan Ilmu Komputer. Universitas Brawijaya. Malang. Pembimbing: Eriq Muhammad Adams J., S.T., M.Kom. dan Denny Sagita Rusdianto S.Kom., M.Kom.

Indonesia merupakan negara yang penuh dengan budaya. Budaya merupakan suatu cara hidup yang berkembang dan dimiliki oleh suatu kelompok orang dan diwariskan dari generasi ke generasi. Cerita rakyat merupakan salah satu budaya yang mulai berkurang peminatnya. Kurang komunikatif dan kurang menarik merupakan salah satu alasan dari mulai ditinggalkannya cerita rakyat. Anak-anak cenderung bosan dengan cerita rakyat yang disuguhkan pada buku dongeng. Mereka lebih memilih media yang memiliki banyak gambar dan memiliki sedikit tulisan untuk dibaca. Penggunaan gambar sebagai media utama dalam menceritakan cerita rakyat dapat meningkatkan ketertarikan bagi anak-anak. Media seperti ini akan lebih menarik bagi anak-anak, terutama jika menggunakan media multimedia seperti permainan yang dapat mengajak anak-anak untuk turun langsung atau ikut aktif dalam cerita yang ada, terutama permainan yang ada pada ponsel layar sentuh, yang dapat dimainkan dengan mudah oleh anak-anak. Dengan menggunakan ponsel layar sentuh mempermudah memainkan sebuah *game* daripada pada komputer yang menggunakan *keyboard* dan *mouse* sebagai *input device* utama. *Input* pada *smartphone* menggunakan sentuhan pada layar langsung sehingga mudah untuk dimainkan. Harapannya, dengan adanya aplikasi ini dapat membantu mengenalkan kembali cerita rakyat dengan cara menyuguhkan cerita rakyat dengan metode penyampaian yang berbeda. Metode dan perancangan *game* ini menggunakan dasar *game design document* sebagai standar dalam merancang *game*. Dari hasil pengujian yang dilakukan, mulai dari pengujian unit dan pengujian integritas dapat disimpulkan bahwa *game* Timun Emas telah memenuhi kebutuhan fungsionalitas.

Kata Kunci: Cerita rakyat, Timun Emas, *Game*

ABSTRACT

Saiful Setyawan, 2015. Design and Build Game Based on Timun Emas' Folktale. Undergraduate Thesis of Informatics Engineering Study Program, Information Technology and Computer Science Program. Brawijaya University. Malang. Pembimbing: Eriq Muhammad Adams J., S.T., M.Kom. and Denny Sagita Rusdianto S.Kom., M.Kom.

Culture is a lifestyle. Indonesia is a country with so many cultures. A folktale is a form of culture that lost it fans recently. Less communicative and considered uninteresting are the reason behind it. Children often get bored reading folktale in a physical book. They prefer read colorful media that has so many images and less text. The usage of picture image tends to be more attractive for children. This kind of method can be enhanced by using an interactive media such as a game, especially a smartphone game that easily played for children. Compared to a desktop computer which uses keyboard and mouse buttons as it main input controls, a smartphone with a touchscreen as an input control tends to be easier to be played for children. Hopefully, a smartphone game can be an alternative media that serves a folktale with some different experiences. This research proposed 'game design document' as a method to create and design the game. As a result from testing, ranging from unit test, integration it can be concluded that this Timun Emas' game has fulfilled the functional requirements product.

Keyword: Folktale, Timun Emas, Game

DAFTAR ISI

KATA PENGANTAR i
ABSTRAK iii
ABSTRACT iv
DAFTAR ISI v
DAFTAR TABEL viii
DAFTAR GAMBAR ix
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan masalah 2
1.3 Batasan Masalah 3
1.4 Tujuan 3
1.5 Manfaat 3
1.6 Sistematika Penulisan 3
BAB II LANDASAN TEORI 5
2.1 Definisi Game 5
2.2 Komponen *Game* 6
2.3 Game Design Document 7
2.4 Iterative Rapid Prototyping 7
2.5 Unity 9
2.6 Timun Emas 11
2.7 Whitebox Testing dan Blackbox Testing 12
2.8 Test Suite dan Test Case 13
2.9 Audacity 13
2.10 Paint Tool SAI 13
BAB III METODOLOGI DAN PERANCANGAN 14

3.1. Metodologi 14
3.1.1. Studi Literatur 14
3.1.2. Perancangan Game 15
3.1.3. Implementasi 15
3.1.4. Pengujian 16
3.2. Perancangan Game 17
3.2.1. Deskripsi Game 17
3.2.2. *Game*play 18
3.2.3. Rancangan Level 20
3.2.4. Konsep Art 21
3.2.5. Game Screen Flow 25
BAB IV IMPLEMENTASI 27
4.1. Spesifikasi Sistem 27
4.2. Pemilihan Teknologi dan Platform 27
4.3. Implementasi Prosedur Program 28
4.3.1. Implementasi Prosedur Program Karakter 28
4.3.2. Implementasi Prosedur Program Achievements 29
4.3.3. Implementasi Prosedur Program Platform 30
4.3.4. Implementasi Prosedur Penyimpanan Score pada Stage 32
4.3.5. Implementasi Prosedur Penyimpanan data Setiap Stage 33
4.3.6. Implementasi Algoritma Tutorial 34
4.4. Implementasi Art 35
4.4.1 Implementasi Komik 35
4.4.2 Implementasi Desain Karakter 37
4.4.3 Implementasi Desain Platform dan Latar Belakang 38
4.4.4 Implementasi Desain HUD 39

4.4.5 Implementasi Desain Main Menu 40
4.4.6 Implementasi Desain Icon Level 41
4.4.7 Implementasi Sound Effect, Suara dan Music 43
4.5. Implementasi Game Screen dan Realisasi Level 44
4.5.1. Splash Screen 44
4.5.2. Prologue 45
4.5.3. Main Menu 45
4.5.4. Tutorial 46
4.5.5. Stage Gua 46
4.5.6. Stage Hutan 47
4.5.7. Stage Padang Rumput 47
4.5.8. Stage Danau 48
4.5.9. Stage Result 48
BAB V PENGUJIAN DAN ANALISIS 49
5.1 Pengujian Whitebox 49
5.1.1. Pengujian Unit 49
5.1.2. Pengujian Integrasi 57
5.2 Pengujian Pengguna 61
5.3.1 Test Suite 61
5.3.2 Play testing 62
5.3.3 Analisis pengujian 62
BAB VI KESIMPULAN DAN SARAN 64
6.1 Kesimpulan 64
6.2 Saran 65
DAFTAR PUSTAKA 66

DAFTAR TABEL

Tabel 4. 1 Tabel Spesifikasi Hardware 27
Tabel 4. 2 Tabel Spesifikasi Minimal Aplikasi 28
Tabel 4. 3 Pseudo code Prosedur Program Timun Emas 28
Tabel 4. 4 Pseudo code Prosedur Program Raksasa 29
Tabel 4. 5 Pseudo code Prosedur Program Koin 30
Tabel 4. 6 Pseudo Code Prosedur Program Platform 31
Tabel 4. 7 Pseudo code prosedur program platform yang jatuh 31
Tabel 4. 8 Source Code ScoreResult.cs 32
Tabel 4. 9 Source Code LevelData.cs 33
Tabel 4. 10 Source Code PlayerData.cs 33
Tabel 4. 11 Pseudo code Prosedur Tutorial 34
Tabel 5. 1 Pseudo code Timun emas 49
Tabel 5. 2 Tabel Kasus Uji untuk obyek Timun Emas 51
Tabel 5. 3 Pseudocode script Horizontal.cs 52
Tabel 5. 4 Tabel Kasus Uji untuk script Horizontal.cs 53
Tabel 5. 5 Tabel Pseudocode Scrip Vertical.cs 54
Tabel 5. 6 Tabel Kasus Uji script Vertical.cs 55
Tabel 5. 7 Pseudocode script Falling.cs 56
Tabel 5. 8 Tabel Kasus Uji script Falling.cs 57
Tabel 5. 9 Tabel Pseudocode ScoreResult 58
Tabel 5. 10Tabel Kasus Uji script ScoreResult 59
Tabel 5. 11 Tabel pseudocode MusicSetting 60
Tabel 5. 12 Tabel Kasus Uji pengujian script MusicSetting 61

DAFTAR GAMBAR

Gambar 2. 1 Pengembangan Iteratif 8
Gambar 2. 2 Iterative and Rapid Prototyping 9
Gambar 2. 3 Interface pada Unity 10
Gambar 3. 1 Flowchart Metodologi Penelitian 14
Gambar 3. 2 Desain Karakter 21
Gambar 3. 3 Desain Splash Screen 22
Gambar 3. 4 Desain Prologue 22
Gambar 3. 5 Desain Menu Utama 23
Gambar 3. 6 Desain Stage Select 23
Gambar 3. 7 Desain In-game 24
Gambar 3. 8 Desain Bungkusan 24
Gambar 3. 9 Desain Score Result 25
Gambar 3. 10 Game Screen Flow 25
Gambar 4. 1 Implementasi Komik I 36
Gambar 4. 2 Implementasi Komik 2 36
Gambar 4. 3 Implementasi Desain Karakter Timun Emas 37
Gambar 4. 4 Implementasi Desain Karakter Mbok Rondo 37
Gambar 4. 5 Implementasi Desain Karakter Raksasa 38
Gambar 4. 6 Implementasi Desain Platform 38
Gambar 4. 7 Implementasi Desain Latar Belakang 39
Gambar 4. 8 Implementasi Desain Bungkusan 39
Gambar 4. 9 Implementasi Desain HUD 40
Gambar 4. 10 Implementasi Penempatan Desain HUD 40
Gambar 4. 11 Implementasi Desain Menu Utama 41
Gambar 4. 12 Implementasi Desain Stage Select 42
Gambar 4. 13 Implementasi Desain Achievements 43
Gambar 4. 14 Implementasi Scene Splash Screen 44
Gambar 4. 15 Implementasi Scene Komik 45
Gambar 4. 16 Implementasi Scene Menu Utama 45
Gambar 4. 17 Implementasi Scene Tutorial 46

Gambar 4. 18 Implementasi Scene Stage Gua 46
Gambar 4. 19 Implementasi Scene Stage Hutan 47
Gambar 4. 20 Implementasi Scene Stage Padang Rumput 47
Gambar 4. 21 Implementasi Scene Stage Danau 48
Gambar 4. 22 Implementasi Scene Score Result 48

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB I
PENDAHULUAN

1.1 Latar Belakang

Indonesia merupakan negara yang penuh dengan budaya. Budaya merupakan suatu cara hidup yang berkembang dan dimiliki oleh suatu kelompok orang dan diwariskan dari generasi ke generasi. Banyak warisan budaya yang ada di Indonesia, mulai dari tradisi yang ada pada setiap daerah, hingga cerita rakyat yang memiliki banyak variasi.

Cerita rakyat merupakan salah satu budaya yang mulai berkurang peminatnya. Kurang komunikatif dan kurang menarik merupakan salah satu alasan dari mulai ditinggalkannya cerita rakyat [VIV-14]. Anak-anak cenderung bosan dengan cerita rakyat yang disuguhkan pada buku dongeng. Mereka lebih memilih media yang memiliki banyak gambar dan memiliki sedikit tulisan untuk dibaca. Penggunaan gambar sebagai media utama dalam menceritakan cerita rakyat dapat meningkatkan ketertarikan bagi anak-anak.

Media seperti ini akan lebih menarik bagi anak-anak, terutama jika menggunakan media multimedia seperti permainan yang dapat mengajak anak-anak untuk turun langsung atau ikut aktif dalam cerita yang ada, terutama permainan yang ada pada ponsel layar sentuh, yang dapat dimainkan dengan mudah oleh anak-anak. Multimedia adalah beberapa kombinasi dari teks, gambar, suara, animasi dan video dikirim ke anda melalui komputer atau alat elektronik lainnya. Dengan multimedia, pengguna dapat lebih berinteraksi dengan hal yang disajikan sehingga informasi yang diberikan dapat mudah ditangkap oleh pengguna [MUL-13].

Dengan menggunakan ponsel layar sentuh mempermudah memainkan sebuah *game* daripada pada komputer yang menggunakan *keyboard* dan *mouse* sebagai *input device* utama. *Input* pada *smartphone* menggunakan sentuhan pada layar langsung sehingga mudah untuk dimainkan. Hampir semua *smartphone* yang digunakan oleh masyarakat Indonesia menggunakan sistem operasi berbasis *android.* Menurut [DET-13], Indonesia merupakan 5 besar negara pengguna *smartphone,* dengan total pengguna 47 juta pengguna aktif.

Dengan mengangkat salah satu cerita rakyat yang berjudul Timun Emas, diharapkan untuk melestarikan, mempertahankan, dan mengenalkan cerita rakyat Timun Emas yang mulai jarang diceritakan di dalam buku. Selain itu, pengangkatan cerita rakyat ke dalam tema *game* juga dapat memberikan pesan moral yang ada pada cerita rakyat. Saat ini, masyarakat cenderung lebih memilih untuk menikmati acara televisi seperti sinetron, sehingga melupakan tentang cerita rakyat yang dulunya merupakan cerita yang selalu diceritakan. Harapannya, dengan adanya aplikasi ini dapat membantu mengenalkan kembali cerita rakyat dengan cara menyuguhkan cerita rakyat dengan metode penyampaian yang berbeda.

Dalam merancang sebuah *game*, biasanya menggunakan paradigma perancangan perangkat lunak. menurut Scott Rogers dalam merancang sebuah *game* terdapat sebuah standar penulisan sebuah *game* dokumen yang disebut *game design document* atau yang biasa disingkat GDD. Sehingga untuk perancangan *game* Timun emas ini penulis mencoba menggunakan GDD sebagai standar dalam merancang sebuah *game.*

Sesuai dengan kebutuhan, setelah *game* dirancang dilakukan pengujian untuk mendapatkan hasil tentang manfaat dari *game* tersebut sesuai dengan permasalahan yang telah disebutkan diatas, serta fungsionalitas dari *game* tersebut.

1.2 Rumusan masalah

Berdasarkan latar belakang, permasalahan yang timbul antara lain:

a. Bagaimana merancang *gameplay* dari *game* ini yang berdasarkan cerita rakyat Timun Emas?
b. Bagaimana implementasi *gameplay* dari *game* Timun Emas?
c. Bagaimana menguji *game* Timun Emas yang telah dibuat?

1.3 Batasan Masalah

Agar pembahasan tidak meluas dan menyimpang dari permasalahan yang ada, maka penulis membatasi permasalahan pada:

a. Cerita rakyat yang digunakan dalam *game* ini adalah Timun Emas.
b. *Game* yang dibangun merupakan *game* dua dimensi (2D).
c. Perancangan yang dibuat sesuai dengan pembuatan *game design document* pada buku *level up* karya Scott Rogers.
d. Pembuatan *game* dilakukan pada *smartphone android.*

1.4 Tujuan

Tujuan penulisan skripsi ini adalah untuk merancang, membangun dan menguji sebuah game sebagai pembelajaran serta mengenalkan cerita rakyat Timun Emas.

1.5 Manfaat

Penulisan skripsi ini diharapkan dapat bermanfaat untuk berbagai pihak.

1. Bagi Penulis
    a. Mengaplikasikan ilmu yang didapat selama mengikuti perkuliahan di Teknik Informatika Universitas Brawijaya.
    b. Mendapatkan pengetahuan dan pengalaman dalam merancang sebuah *game*.
    c. Membantu mengenalkan kembali salah satu cerita rakyat yang merupakan salah satu warisan budaya Indonesia
2. Bagi Pengguna
    a. Dapat memahami lebih tentang cerita rakyat Timun Emas.
    b. Secara tidak langsung melatih daya pikir, kreativitas dan imajinasi.

1.6 Sistematika Penulisan

Sistematika penulisan dalam skripsi ini sebagai berikut:

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, tujuan, batasan masalah, manfaat tujuan, dan sistematika penulisan.

BAB II Dasar Teori

Membahas teori dasar dan teori penunjang yang berhubungan dengan pembuatan *game* pada *game engine Unity*.

BAB III Metodologi dan Perancangan

Berisi tentang langkah-langkah dalam merancang pembuatan *game*. Serta membahas analisis kebutuhan dan perancangan yang sesuai dengan teori yang ada dalam membangun *game* agar sesuai dengan keinginan penulis serta bermanfaat bagi pengguna.

BAB IV Implementasi

Membahas tentang implementasi *game* yang dibangun. Meliputi pembuatan *environtment*, implementasi desain karakter, alur *game*, menu *game* menggunakan *software-software* yang dibutuhkan. Tidak terkecuali pemrograman *game* menggunakan bahasa pemrograman C#.

BAB V Pengujian

Pada bab ini memuat hasil pengujian serta analisis terhadap *game* yang telah dibuat.

BAB VI Penutup

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

BAB II
LANDASAN TEORI

2.1 Definisi Game

Game, atau dalam bahasa Indonesia disebut dengan *game* adalah aktivitas yang dimainkan untuk olahraga atau hiburan menurut aturan. Seperti yang kita ketahui, pada *game* terdapat juga kondisi menang atau kalah [MAH-10].

Menurut Scott Rogers, terdapat banyak *genre game* yang ada, diantaranya adalah:

1. *Action game,* merupakan *game* yang membutuhkan koordinasi dan kecekatan mata dan tangan.
2. *Shooter game,* merupakan *game* yang fokus utama pada menembak musuh dengan senjata api.
3. *Adventure game,* merupakan game yang memfokuskan pada memecahkan teka-teki dan pengumpulan item. Merupakan sebuah *game* yang membutuhkan waktu lama untuk diselesaikan.
4. *Construction/management,* pada *game* ini, pemain diharuskan untuk membangun dan meluaskan wilayah dengan sumber daya yang terbatas.
5. *Music/rhythm,* pemain bermain dengan lagu dan *rhythm* yang ada untuk mendapatkan skor tertinggi.
6. *Party game, game* ini dikhususkan untuk dimainkan oleh lebih dari satu pemain.
7. *Puzzle game, game* yang membutuhkan logika dan pola untuk diselesaikan.
8. *Sports game, game* yang mempunyai tema dasar olahraga.
9. *Strategy game,* merupakan *game* yang membutuhkan pemikiran seperti membuat rencana untuk menyelesaikannya.
10. *Simulation game, game* yang dibuat berdasarkan pengalaman nyata. *Game* ini dibuat untuk mendekati sesuatu yang nyata.

Action game dikategorikan menjadi beberapa tipe, diantaranya:

a. *Action-adventure,* merupakan game *action,* dengan penekanan pada pengumpulan *item,* dan pemecahan teka-teki. Umumnya merupakan *game* jangka panjang yang didasari cerita dalam game.
b. *Action-arcade*, merupakan game *action,* dengan penekanan pada pencapaian suatu skor tinggi. Umumnya merupakan *game* jangka pendek.
c. *Platformer*, adalah tipe *game* dimana karakter utama melakukan aksi melompat (berayun atau memantul) untuk melalui tantangan pada *platform*. *Platform* itu sendiri adalah benda atau obyek yang tersusun secara horizontal yang menyerupai koridor dan dapat digunakan oleh karakter pada sebuah *game* sebagai jalan yang dapat dilalui.
d. *Stealth*, *game action* dengan penekanan pada menghindari musuh bukan langsung bertarung dengan mereka.
e. Fighting, permainan di mana dua atau lebih lawan arena. *Game fighting* dibedakan dari kontrol pada setiap karakter yang disediakan.
f. Hack ‘n Slash, pada *game* ini pemain bertarung dengan banyak musuh yang datang gelombang demi gelombang dalam jumlah yang banyak. [LVL-10]

2.2 Komponen *Game*

Dalam *game,* terdapat beberapa istilah yang sering digunakan. Untuk mempermudah pemahaman, diantaranya:

a. *Splash Screen,* merupakan tampilan awal ketika *game* dibuka. Pada umumnya, *splash screen* berisi logo dari *developer,* distributor atau sponsor yang berperan dalam pembuatan dari *game* tersebut.
b. *HUD* atau *Heads-up Display,* merupakan sebuah desain antarmuka atau *user interface (UI)* pada *game* yang berisi tentang informasi ini pada *game,* seperti kondisi pemain, alat atau senjata yang digunakan, tombol *pause, minimap, high score* dan sebagainya.
c. *Level*, merupakan tingkat kesulitan pada *game.* Umumnya, setiap *game* memiliki tiga tingkat kesulitan, yaitu mudah *(easy),* sedang *(normal),* dan sulit *(hard).*

d. *Achievements,* merupakan penghargaan pada *game* jika pemain mencapai pada kondisi tertentu.
e. *Objectives,* merupakan perintah-perintah yang harus pemain selesaikan untuk dapat menyelesaikan *stage* yang ada. [LVL-10]

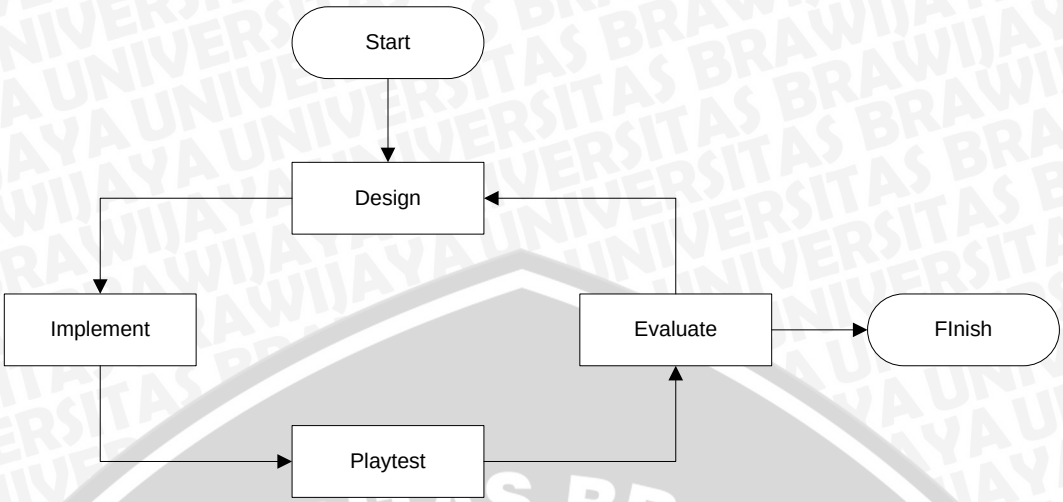
2.3 Game Design Document

Game design document (GDD) merupakan sebuah metode penulisan dalam perancangan sebuah *game.* GDD dibuat bertujuan untuk menjelaskan komponen dari *game* yang akan dibuat agar dapat terdokumentasi dengan baik dan mudah dimengerti bagi siapapun yang membacanya.

Dalam pembuatannya, GDD melalui beberapa tahap terlebih dahulu. Sebelum pembuatan GDD secara keseluruhan, terlebih dahulu dibuat *one-sheet document*. *One-sheet document* yang merupakan kerangka dasar dalam pembuatan GDD berisi tentang deskripsi umum *game* seperti judul *game, ERSB rating,* deskripsi singkat tentang *gameplay* serta rangkuman singkat tentang cerita pada *game,* serta *unique selling point. One-sheet document* biasanya ditujukan pada tim pengembang serta *publisher*. Seperti namanya, o*ne-sheet document dibuat* secara ringkas pada dokumen satu halaman. Selanjutnya setelah pembuatan *one-sheet document* membuat *ten pager document*. Tujuan dari *ten pager document* agar pembaca cepat mengerti tentang konsep dasar dari hasil akhir *game* yang akan dibuat tanpa detail yang menyeluruh. Pembuatan *ten pager document* digunakan untuk *publisher* sehingga sebaiknya pembuatan *ten pager document* dalam bentuk *power point.* Setelah itu *ten pager document* dapat dikembangkan untuk menjadi GDD yang mempunyai isi lebih detail tentang isi dari *game* yang akan dibuat. [LVL-10]

2.4 Iterative Rapid Prototyping

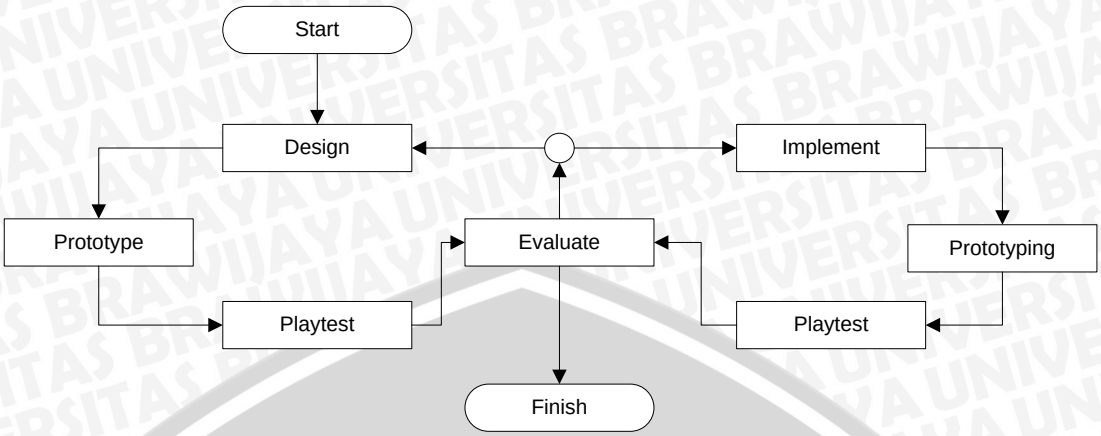
Iterative rapid prototyping merupakan sebuah metode dalam perancangan *game design.* Sesuai dengan namanya, metode ini dasarnya merupakan pengembangan atau perulangan yang dilakukan dalam merancang sebuah game untuk mendapatkan ide atau konsep dari sebuah yang lebih matang. Semakin banyak iterasi yang dilakukan, ide atau konsep dari game yang dirancang akan semakin kuat. [GMD-09]



Gambar 2. 1 Pengembangan Iteratif

Pada gambar 2.1, merupakan pengembangan dengan metode iterasi, dimana pada awal pengembangan dibuat rancangan desain dulu, lalu diimplementasikan dan setelah selesai dibuat dilakukan *playtest.* Pada fase evaluasi menentukan keputusan apakah *game* yang dibuat sudah berhasil dan siap dipasarkan, atau terdapat beberapa elemen yang perlu diperbaiki sehingga memasuki iterasi desain-implementasi-*playtest* sekali lagi, sehingga pada pengembangan ini dapat disimpulkan semakin banyak iterasi yang dilakukan, akan lebih baik hasil final yang dikembangkan.

Tetapi, jika dilakukan pengembangan sesuai dengan gambar 2.1, pada setiap memasuki fase *implement* (implementasi pada program dan *debugging*) akan membutuhkan waktu lama sehingga setiap dilakukan iterasi, maka waktu yang dibutuhkan akan jauh lebih banyak, sehingga waktu untuk melakukan *playtest* (*game testing* dan pencarian *bug*) akan terpangkas. Sehingga, perancangan desain dilakukan diatas kertas terlebih dahulu dan dilakukan iterasi sebanyak yang dibutuhkan untuk mendapatkan desain *game* yang lebih baik sebelum desain tersebut memasuki fase implementasi.



Gambar 2. 2 Iterative and Rapid Prototyping

Pada *iterative and rapid prototyping,* desain dan implementasi dilakukan dalam iterasi yang berbeda, hal ini dilakukan agar iterasi yang ada fase desain tidak terpengaruhi oleh lamanya implementasi khususnya pada implementasi programming dan debugging yang relative memakan waktu lama. [GDC-13]

2.5 Unity

Unity adalah sebuah *game engine* multi platform. Unity digunakan untuk membuat *game* untuk *web plug in, desktop platform, console*, dan perangkat *mobile*. *Unity* dirilis pada tahun 2005, oleh David Helgason, Nicholas Francis, dan Joachim Ante. [UNI-NN]

Pada pembuatan *game,* terdapat beberapa istilah umum yang ada pada *unity,* diantaranya:

a. *Scene,* merupakan istilah pada *unity* yang digunakan untuk menyebutkan sebuah *level* atau menu.
b. *Gameobject,* merupakan sebuah *object* yang ada pada *unity*. *Gameobject* memiliki beberapa komponen yang dapat ditambahkan atau dikurangi dengan cara *drag and drop*.
c. *Asset Store,* merupakan sebuah *website* yang disediakan oleh *unity,* yang menyediakan desain, *plug in,* hingga *sounds* baik yang open source maupun yang berbayar.

Gambar 2. 3 Interface pada Unity

Pada gambar 2.3, adalah contoh dari *interface* pada *unity*. Dalam obyek pada *unity*, terdapat beberapa *component* yang bisa dilihat pada *inspector*. Pada kolom *hierarchy* menampilkan daftar obyek yang ada pada suatu *scene* beserta dengan hierarchy dari *parent-child* pada setiap obyek. Pada kolom *project*, menampilkan *asset* dari *project* sesuai dengan penempatan folder pada *explorer*. Pada kolom *scene* menampilkan rancangan *game* yang dibuat secara keseluruhan.

Dalam *unity*, setiap *script* diturunkan dari *parent class* utama yang ada pada *library unity* yang bernama *MonoBehaviour*. Pada *MonoBehaviour* terdapat *method* dasar yang biasa digunakan pada pemrograman *game*, seperti *method Start* yang dieksekusi pada saat awal *game* dijalankan, dan *method Update* yang dieksekusi pada setiap *frame* pada game. *Script* pada unity dikategorikan sebagai *component* sehingga setiap *script* dapat di masukan ke dalam obyek seperti *component* lainnya.

Fitur utama yang disediakan oleh *unity* antara lain:

1. *Drag and drop*, fitur yang memudahkan pengguna untuk *import component* seperti memasukkan suatu *script* pada obyek yang diinginkan,

atau memasukkan *file* gambar pada obyek untuk dijadikan sebagai *sprite* atau sebagai material pada obyek 3D.

2. *Prefab,* merupakan sebuah *gameObject* yang disimpan dalam bentuk file. Dengan menggunakan *prefab,* memudahkan untuk memasukkan beberapa obyek yang sama dalam *game.*
3. *Animator* dan *animation, animator* digunakan sebagai heriarki *animation* pada suatu obyek dalam *unity,* sedangkan *animation* digunakan untuk membuat animasi pada obyek 2D ataupun obyek 3D.

Pada *unity* mulai versi 4.3, pengguna dapat membuat *game* berbasis 2D untuk mempermudah pembuatan *game* versi dua dimensi. Dengan fitur ini, terdapat beberapa perbedaan pada saat pembuatan *game* tiga dimensi dengan dua dimensi, antara lain:

1. Sudut pandang dari kamera yang bersifat orthodox, yaitu sumbu z pada koordinat *cartesius* dihiraukan.
2. *Texture mode* pada gambar yang telah diimport pada unity terdapat pilihan *sprite.*
3. Terdapat *window sprite editor,* yang dapat digunakan untuk memotong tiap *sprite* yang tersimpan pada satu gambar.
4. Fitur *sorting layer,* yang berguna untuk mengurutkan tumpukan gambar yang terjadi pada *screen.*
5. Komponen *physic* dibedakan antara tiga dimensi dan dua dimensi untuk mempermudah dalam pemakaiannya, seperti *collider.*

2.6 Timun Emas

Timun emas adalah sebuah cerita rakyat yang berasal dari Jawa Tengah, yang menceritakan seorang anak perempuan yang dikejar oleh raksasa. Dalam cerita ini, terdapat seorang wanita tua yang hidup seorang diri di sebuah hutan, yang bernama Mbok Rondo. Pada suatu ketika Mbok Rondo bertemu sesosok raksasa, dan kemudian raksasa tersebut memberikan sebiji bibit mentimun yang akan memberinya seorang anak, dengan syarat anak tersebut akan dimakan oleh raksasa tersebut ketika sudah cukup besar.

Setelah menanam biji mentimun itu, ada sebuah mentimun yang bentuknya sangat besar, setelah di belah terdapat bayi di dalamnya, dan diberi nama Timun Emas. Setelah beberapa tahun, Sang Raksasa menemui Mbok Rondo untuk meminta Timun Emas untuk dimakan. Merasa Timun Emas dalam bahaya, Mbok Rondo meminta kepada Sang Raksasa untuk memberikan waktu dua tahun lebih lama, dan raksasa pun menyetujui untuk memberikan waktu dua tahun lagi.

Pada suatu malam, Mbok Rondo bermimpi bertemu seseorang yang memberitahu Mbok Rondo untuk menemui seorang pertapa yang bertapa di sebuah bukit. Keesokan harinya pun Mbok Rondo langsung mencari pertapa tersebut untuk meminta bantuan. Sang Pertapa pun memberikan tiga buah bungkusan kepada Mbok Rondo yang dapat digunakan sebagai senjata untuk mengusir Sang Raksasa.

Dua tahun kemudian, raksasa kembali ke Mbok Rondo untuk mengambil Timun Emas. Tetapi, Timun Emas berhasil kabur lewat pintu belakang rumahnya. Sang Raksasa pun mengejar Timun Emas, namun Raksasa tersebut berhasil dikalahkan berkat tiga buah bungkusan yang diberikan oleh Sang Pertapa. [TIM-NN]

2.7 Whitebox Testing dan Blackbox Testing

Whitebox dan *blackbox* merupakan metode dalam pengujian sebuah perangkat lunak. *Whitebox* adalah pengujian yang digunakan pada sisi pengembang dan *blackbox* adalah pengujian yang digunakan pada pengguna. *Whitebox testing* menguji apa yang terjadi pada sistem, seperti struktur sistem, jalur algoritma, proses dalam algoritma dan tingkat komplektisitas pada sebuah perangkat lunak. Salah satu metode pengujian *whitebox* adalah *basic path testing*. Metode ini digunakan untuk menguji tingkat komplektisitas pada sebuah algoritma dan menguji setiap jalur pada percabangan dan perulangan dalam batasan tertentu.

Pengujian *blackbox* adalah sebuah metode yang digunakan untuk mengamati, apakah keluaran dari sebuah perangkat lunak sesuai dengan input yang dimasukkan sesuai yang dibutuhkan tanpa harus mengetahui proses yang dilakukan oleh perangkat lunak tersebut. [RED-08]

2.8 Test Suite dan Test Case

Test suite adalah sekumpulan kasus uji *(test case)* yang disusun secara berurutan, dan dideskripsikan lebih detail. *Test suite* memberikan langkah-langkah instruksi tentang pengoperasian *game*, seperti apa yang harus dilakukan pada *game* dan detail apa yang harus dicek sebagai hasil dari setiap langkah. [ALL-12]

Dalam pembuatan *test suite*, awalnya ditentukan isi dari *test case*. *Test case* merupakan sebuah *test* yang memiliki suatu tujuan untuk mengamati setiap fungsionalitas dari sebuah *game. Test case* juga mendeskripsikan operasi atau tindakan apa yang harus dilakukan untuk mencapai tujuan tersebut. Dalam merancang *test suite* umumnya dibuat berdasarkan urutan dari *event* yang ada pada *game* yang akan diuji, sehingga *test case* yang dirancangpun tersusun secara berurutan sesuai dengan jalannya *game*, mulai dari *game* pertama dibuka sampai *game* ditutup.

2.9 Audacity

Audacity merupakan sebuah perangkat lunak yang digunakan untuk mengolah suara. *Audacity* dapat digunakan untuk merekam, memotong, dan menambahkan suara pada *track* yang diinginkan. Audacity dapat mendukung *file* dengan tipe WAV, AIFF, FLAC, MP2, MP3 dan Ogg. [AUD-NN]

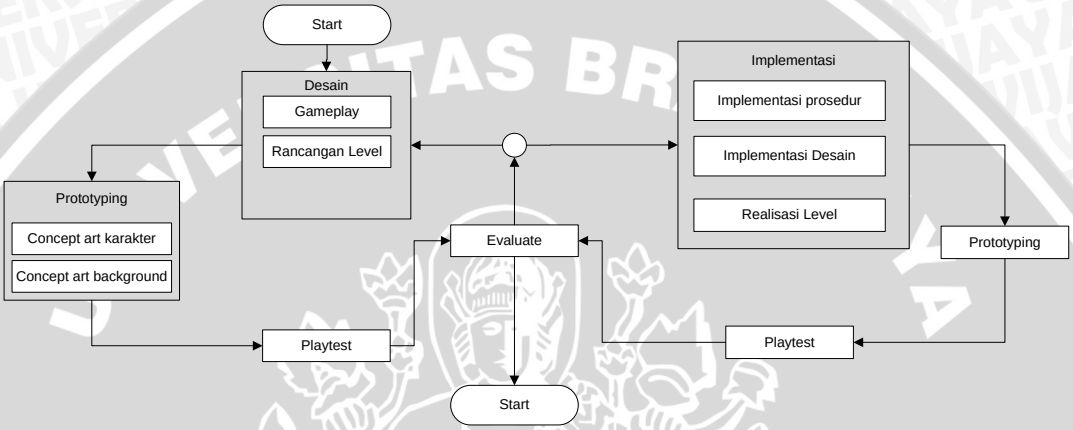
2.10 Paint Tool SAI

Paint Tool SAI merupakan sebuah perangkat lunak dikembangkan oleh *systemax* yang digunakan untuk pengolahan gambar. Berbeda dengan *Adobe Photoshop*, *Paint Tool SAI* mempunyai fitur yang lebih sedikit sehingga membuat kapasitas yang dibutuhkan dari perangkat lunak ini menjadi lebih sedikit. Fitur dari *Paint Tool SAI* sebagian besar mempermudah dalam menggambar, seperti *brush* yang bisa di *customize* sesuai dengan keinginan penggunanya. [SYS-NN]

BAB III
METODOLOGI DAN PERANCANGAN

3.1.Metodologi

Pada bab ini akan dibahas mengenai langkah-langkah yang akan dilakukan dalam pengerjaan skripsi, yaitu studi literatur, perancangan *game*, implementasi, pengujian, pengambilan kesimpulan dan juga saran dari aplikasi yang akan dibuat.



Gambar 3. 1 Flowchart Metodologi Penelitian

Metodologi yang digunakan adalah *iterative rapid prototyping*, yang merupakan sebuah metodologi dalam *prototyping* atau merancang sebuah ide dari *game*. Pada dasarnya, metode ini melakukan iterasi atau perulangan secara terus menerus hingga mendapatkan ide atau konsep dari sebuah *game* yang lebih matang dari konsep yang telah dibuat sebelumnya. Untuk lebih jelasnya dapat dilihat pada gambar 3.1.

3.1.1. Studi Literatur

Studi literatur menjelaskan dasar teori yang digunakan untuk menunjang penulisan skripsi. Referensi dapat diambil dari sebuah buku, jurnal, laporan penelitian, maupun artikel yang berhubungan. Teori-teori pendukung tersebut meliputi:

- Perancangan *game*
- *Unity3D*
- *Photoshop CS5*
- *SAI Paint Tool*

- *Audacity*
- Buku Cerita Timun Emas karya terbitan CV. Pustaka Agung Harapan

3.1.2. Perancangan Game

Dalam pembuatan sebuah *game*, tahap pertama yang dilakukan adalah untuk merancang desain awal *game* yang berisi tentang *gameplay* sesuai dengan cerita dasar yang diadaptasi dari cerita Timun Emas. Gameplay meliputi cara bermain, peraturan permainan atau *rules*, kontrol yang digunakan, goal, genre, karakter, sudut kamera, serta *game items*. Dari *gameplay* yang telah ditentukan sebelumnya, dapat ditentukan *game* tersebut termasuk pada genre yang sesuai dengan *rule*, *goal* dan *objectives* yang telah ditentukan.

Setelah semua konsep dasar telah dikumpulkan tahap berikutnya yang dilakukan pembuatan rancangan *level*, mekanis *game*, desain karakter, desain HUD serta desain latar belakang yang akan digunakan. Rancangan *level* menjelaskan deskripsi dari setiap *level*, apa saja bagian yang membedakan dari setiap *level* pada *game* yang akan dibuat. Mekanis *game* adalah suatu *game object* yang berinteraksi dengan pemain untuk tujuan tertentu.

Setelah itu, dari rancangan *game* yang telah dibuat, akan ditentukan apakah rancangan tersebut sudah cukup, atau perlu diulang dengan menambah, mengurangi atau merubah satu atau beberapa ide pada rancangan tersebut.

3.1.3. Implementasi

Implementasi pembuatan *game* dilakukan dengan mengacu pada pengguna yang memiliki *smartphone android* yang menjalankan sistem operasi *android* versi 4.0 keatas, dan menggunakan arsitektur CPU ARMv7. Hal ini dikarenakan *Unity 4.6* hanya bisa membuat *game* untuk *android* dengan CPU versi ARMv7 saja. Hasil dari pembuatan *game* pada *Unity* akan disesuaikan dengan aplikasi *android* yang berekstensi .apk yang dapat dipasangkan pada ponsel *android.*

Pada implementasi juga Membahas tentang tampilan antarmuka *(User Interface)*, *game art*, dan *gameplay*. Hal-hal yang mempengaruhi dalam implementasi pembuatan *game*, antara lain:

1. Mengangkat cerita Timun Emas terbitan dari CV. Pustaka Agung Harapan sebagai acuan utama dalam pembuatan alur cerita.
2. Penggunaan *Unity* sebagai *Game Engine* dalam pembuatan *game.*
3. Penggunaan Bahasa C# sebagai bahasa pemrograman utama.
4. Penggunaan *Photoshop CS5* dan *SAI Paint Tool*, sebagai aplikasi dalam pembuatan desain karakter, desain *environment*, HUD, dan material 2D lainnya.
5. Penggunaan *Audacity*, sebagai aplikasi dalam pembuatan dan pengolahan suara untuk musik dan *sound effect*.

Seperti pada fase perancangan, pada proses implementasi juga dilakukan iterasi jika terdapat satu atau beberapa bagian dari game yang perlu dirubah. Fase implementasi meliputi realisasi *gameplay* yang telah ditentukan dengan programming, pembuatan model karakter sesuai dengan desain awal yang ada pada *concept art*, serta realisasi *level* sesuai dengan rancangan *level* yang telah ditentukan pada perancangan.

3.1.4. Pengujian

Pengujian dilakukan setelah proses implementasi selesai dilakukan. Pengujian dilakukan dengan tiga metode, yaitu *whitebox testing, test suite* dan *play testing. Whitebox testing* digunakan untuk menguji dari performa dari sebuah perangkat lunak.

Salah satu metode pengujian *whitebox* yang digunakan adalah *basic path testing.* Metode ini digunakan untuk menentukan tingkat komplektisitas dari sebuah algoritma pada perangkat lunak, serta menguji setiap percabangan dan perulangan *(loop)* yang ada pada sebuah algoritma dalam batasan tertentu. Penggunaan *basic path testing* ini digunakan untuk mengamati jalur algoritma pada setiap jalur percabangan dan perulangan sesuai dengan yang dibutuhkan.

Pengujian *test suite* adalah pengujian yang dilakukan dengan serangkaian *test case* untuk memantau fungsionalitas dari sebuah *game. Test suite* memberikan instruksi langkah demi langkah tentang apa yang harus dilakukan untuk

menjalankan sebuah game dan apakah keluaran akhir pada setiap langkah sesuai dengan ekspektasi.

Pengujian *play testing* merupakan salah satu metode dalam menguji sebuah *game*. Pengujian *play testing* akan dilakukan kepada anak-anak berumur dibawah tujuh tahun, yang merupakan target pemain dari *game* ini. Untuk mendapatkan hasil uji yang dilakukan kepada anak-anak, digunakan kuesioner sebagai media untuk pengujian ini. Kuesioner sendiri adalah sebuah metode yang digunakan untuk pengumpulan data dan informasi pada suatu masalah. Partisipan tidak dibatasi umur, jenis kelamin, pangkat dan sebagainya. Dengan penggunaan kuesioner ini akan didapatkan informasi mengenai informasi seperti tingkat kesulitan, tingkat pemahaman pemain pada cerita yang disampaikan, dan seberapa menarik *gameplay* yang dimainkan. Hasil dari kuesioner tersebut akan menjelaskan seberapa banyak anak-anak yang mengetahui cerita Timun Emas. Pada hasil kuesioner ini dapat diketahui pula pengetahuan anak kecil tentang cerita Timun Emas dan seberapa menarik *game* Timun Emas.

3.2.Perancangan Game

Perancangan berisikan tentang konsep *game* yang akan dikembangkan. Perancangan ini berisikan mulai dari Deskripsi *game*, judul, *genre*, *overview*, *ESRB rating*, *gameplay*, rancangan *level*, konsep *art*, dan seterusnya hingga pada *game screen flow* yang akan digunakan pada *game* yang akan dibuat.

3.2.1. Deskripsi Game

Pada *game* ini, pemain membantu Timun Emas untuk lolos dari kejaran Raksasa dengan membuatkan jalan dari *platform* yang ada pada setiap *level* untuk dilalui oleh Timun Emas, serta memperlambat gerakan Raksasa dengan cara melemparkan bungkusan kepada Raksasa. Pemain harus mencegah Timun Emas ditangkap oleh Raksasa, atau permainan akan berakhir dan pemain harus mengulang permainan mulai dari awal. Selain pemain harus lolos dari kejaran Raksasa pada setiap level, pemain juga dapat mengumpulkan *achievement* pada setiap *level*. Ketika pemain berhasil melalui setiap *level* yang ada, Timun Emas dapat pulang ke rumahnya dan *game* akan berakhir.

Bagian-bagian yang ada pada deskripsi *game* antara lain:

1. Judul Game, adalah Timun Emas, yang diambil dari judul cerita rakyat Timun Emas itu sendiri.
2. *Genre, game* ini merupakan *platformer*.
3. *ESRB Rating, Everyone*
4. Tujuan *(Goal),* adalah agar Timun Emas kembali ke rumahnya dan lolos dari kejaran Raksasa.

3.2.2. *Game*play

*Game*play menjelaskan tentang dasar dari sebuah game, seperti tentang

a. *Rule* atau peraturan, yang berisi tentang aturan permainan seperti kondisi kalah dan memang.
b. Kontrol, yang berisi tentang cara memberikan perintah pada karakter atau benda yang berinteraksi pada permainan.
c. Kamera, menjelaskan tentang sudut kamera yang digunakan.
d. Karakter, menjelaskan tentang karakter yang ada pada *game.*
e. *Game items,* menjelaskan tentang benda yang berkaitan pada sebuah *game.*
f. Game mechanics, menjelaskan tentang benda yang berinteraksi langsung dengan karakter dan *game.*

a. Rule

Peraturan dasar yang ada pada *game* ini terbagi menjadi kondisi menang dan kondisi kalah.

1. Kondisi Menang, ketika pemain dapat dinyatakan menang apabila lebih dulu mencapai *goal point* dibandingkan Raksasa yang mengejar Timun Emas.
2. Kondisi Kalah, ketika pemain Timun Emas tertangkap oleh Raksasa, atau jatuh dari *platform*.

b. Kontrol Game

Pemain dapat mengontrol posisi *platform* dengan mengklik/menyentuh platform yang dapat digerakkan. Area dari *collider* pada setiap *platform* yang dapat digerakkan diperbesar untuk mempermudah pemain untuk menggerakkan *platform* yang ada.

Pemain juga bisa meletakkan bungkusan pada saat permainan. Bungkusan tersebut dapat digerakkan dengan cara *drag and drop*, sehingga pemain dapat mengklik/menyentuh dan menahan bungkusan tersebut, menggerakkan ke posisi yang diinginkan pemain dan melepaskan bungkusan tersebut jika posisinya telah sesuai.

c. Kamera

Pada *game* Timun Emas, sudut kamera yang digunakan adalah *parallax scrolling. Parallax scrolling* adalah sudut kamera dimana pada saat kamera bergerak, lingkungan pada *game* tersebut ikut bergerak. Terdapat dua macam dari *parallax scrolling*, yaitu pergerakan kamera mengikuti karakter, dan karakter dipaksa untuk mengikuti kecepatan konstan kamera.

d. Karakter

Pada *game* Timun Emas terdapat beberapa karakter yang ada, diantaranya:

1. Timun Emas, sebagai karakter utama yang dapat dimainkan oleh pemain.
2. Raksasa, sebagai karakter antagonis utama. Raksasa merupakan seseorang yang sombong dan serakah.
3. Mbok Rondo, sebagai karakter pendukung. Mbok Rondo merupakan karakter yang tidak bisa dimainkan.
4. Pertapa, merupakan karakter pendukung pada cerita Timun Emas. Pada cerita ini, Pertapa memberikan bungkusan kepada Mbok Rondo yang digunakan untuk melawan Raksasa.

e. Game Items

Item yang tersedia pada *game* ini adalah tiga bungkusan yang ada pada cerita Timun Emas, tiga bungkusan tersebut berisi jarum, garam dan terasi. Tiap

bungkusan dapat membantu pemain untuk memperlambat Raksasa dalam jangka waktu selama dua detik.

f. Game Mechanic

Berbeda dengan *game items, game mechanic* adalah sebuah atau beberapa obyek pada sebuah game yang berinteraksi langsung yang bertujuan untuk membantu pemain. Pada *game* Timun Emas, *game object* yang dikategorikan pada *game mechanic* adalah *platform* yang digerakkan oleh pemain. Untuk menggerakkan *platform*, pemain cukup meng-klik atau menyentuh *game object* tersebut untuk bergerak. Berdasarkan fungsinya, terdapat empat macam *platform*, diantaranya *platform* yang tidak bergerak, *platform* yang bergerak secara horizontal, *platform* yang bergerak secara vertikal, *platform* yang akan jatuh jika Timun Emas berada ditengahnya.

3.2.3. Rancangan Level

Pada setiap level, yang membedakan antara level yang satu dengan level yang lain adalah tingkat kesulitannya. Point yang menentukan tingkat kesulitan itu antara lain:

a. Tingkat kerumitan platform, mulai dari letak dari platform itu sendiri, sampai banyaknya jebakan yang ada pada setiap level.
b. Kecepatan lari dari Timun Emas dan Raksasa.
c. Penambahan kecepatan maksimal Raksasa setiap Raksasa jatuh.
d. Waktu yang tersedia pada setiap level semakin diperpendek.

Level yang ada pada *game* Timun Emas sebanyak sembilan level yang dibagi berdasarkan latar belakang menjadi tiga stage utama dan satu stage terakhir, yaitu:

a. *Stage* Gua, berisi *level* pada awal *game*. Berisi dua *level*, pada *level* pertama pemain akan bermain tanpa raksasa yang mengejar. Pada *level* kedua Raksasa mulai mengejar Timun Emas. Pada *level* ini, kecepatan Raksasa ditambah sebanyak 5% setiap kali Raksasa jatuh.
b. *Stage* Hutan, berisi *level* pada pertengahan *game*. Pada *level-level* ini, jumlah dari *platform* bertambah. Jumlah *platform* yang menjebak pemain lebih banyak jika dibandingkan pada *stage* gua. Penambahan kecepatan

Raksasa setiap kali terjatuh ditambah, berurutan pada *level* satu sebesar 10%, *level* dua sebesar 13%, dan *level* tiga sebesar 15%.

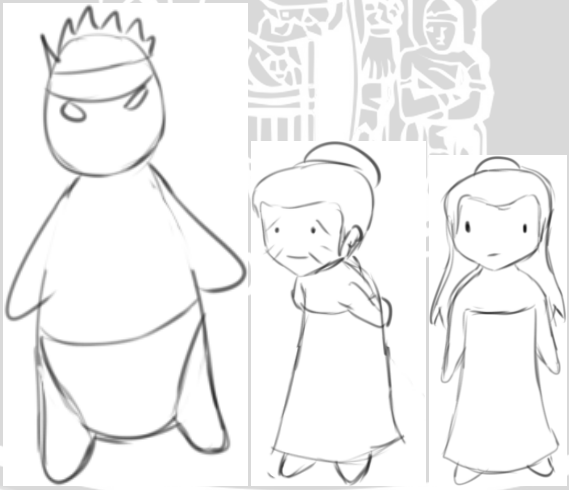
c. *Stage* Padang Rumput berisi *level* pada bagian *mid-end game.* Tingkat kesulitan pada *level-level* ini menjadi sulit. Banyaknya *platform* yang dapat jatuh ditambahkan dan penambahan kecepatan Raksasa pada *level* satu sebesar 20%, pada *level* dua sebesar 23% dan *level* tiga sebesar 25%.

d. *Stage* danau berisi satu level dan merupakan *level* terakhir. Pada *level* ini tidak ada Raksasa yang mengejar Timun Emas, namun jebakan ditambah dan jumlah *platform* sangat banyak jika dibandingkan dengan *level* lainnya.

3.2.4. Konsep Art

a. Konsep Art Karakter

Karakter utama pada *game* ini adalah Timun Emas, Raksasa, dan Mbok Rondo. Untuk konsep art pada setiap karakter, desain dibuat berdasarkan tema budaya Indonesia. Seperti pemakaian batik sebagai bagian dari pakaian Timun Emas dan Mbok Rondo, juga konde pada rambut Mbok Rondo. Konsep art dari setiap karakter dapat dilihat pada gambar 3.2.

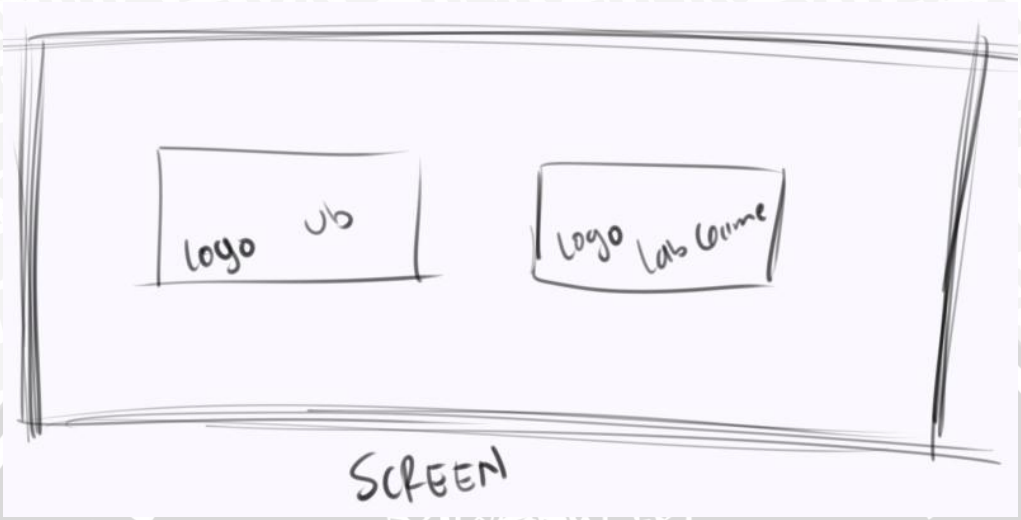


Gambar 3. 2 Desain Karakter

b. Konsep Art Scene

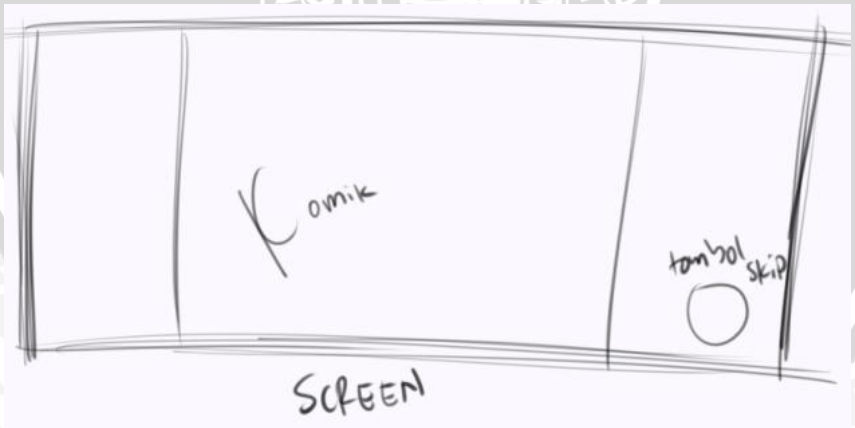
Pada saat pemain memulai *game*, pemain akan masuk pada *splash screen*, yang memuat gambar logo Universitas Brawijaya dan log Lab *Game* seperti pada

gambar 3.3. Pada umumnya, *splash screen* memuat gambar dari logo perusahaan *publisher* atau distributor serta informasi dari sponsor yang berkontribusi pada pembuatan *game.*



Gambar 3. 3 Desain Splash Screen

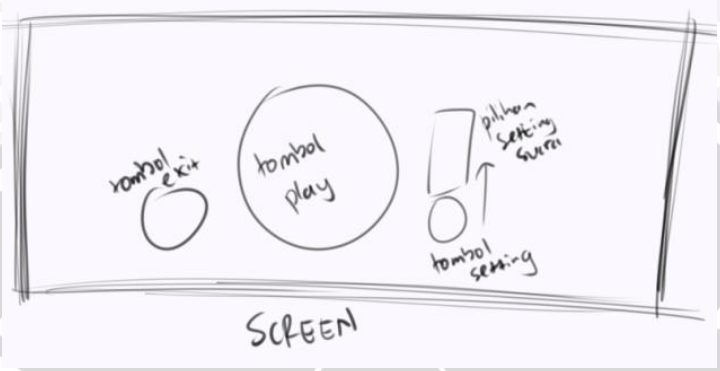
Setelah melewati *splash screen*, pemain akan diberi cerita bernarasi singkat tentang awal cerita dari Timun Emas. Seperti yang ditunjukkan gambar 3.4, pada *scene* ini, selain berisi narasi, terdapat komik singkat yang mudah dipahami agar pemain tidak bosan dengan cerita yang hanya berisi kata – kata dan kalimat.



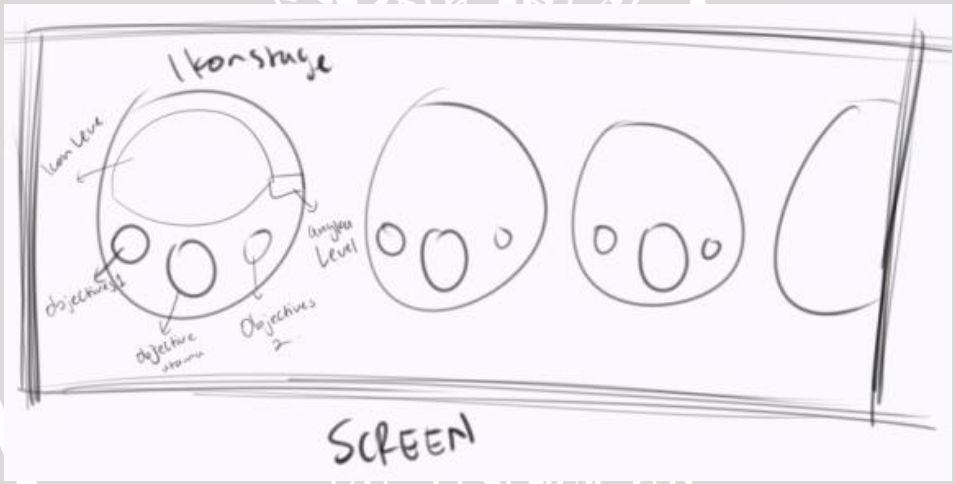
Gambar 3. 4 Desain Prologue

Seperti yang ditunjukkan gambar 3.5, *main menu*, berisi beberapa tombol yang dapat dipilih, untuk langsung bermain atau melakukan pengaturan dahulu

pada *setting*. Setelah pemain memilih *play,* terdapat beberapa tombol yang mewakili setiap *level* yang ada. Pada gambar 3.6, setiap tombol terdapat tiga ikon yang menunjukkan semua *achievements* yang tersedia. Warna awal dari ikon ini adalah abu-abu, warna tersebut akan berganti menjadi kecokelat-cokelatan saat pemain menyelesaikan *achievements* tersebut.

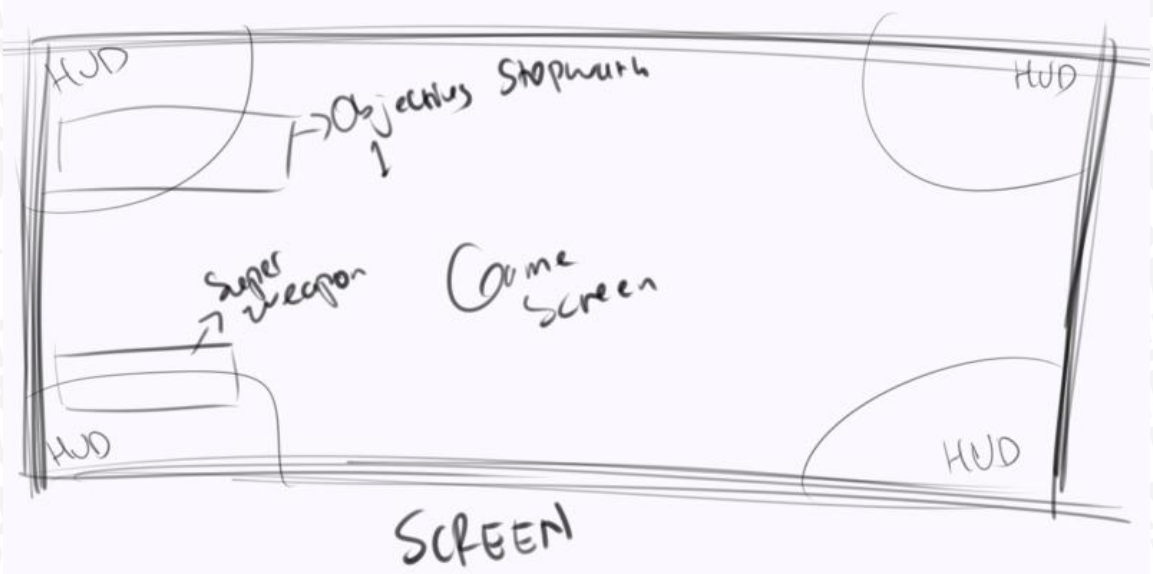


Gambar 3. 5 Desain Menu Utama



Gambar 3. 6 Desain Stage Select

Pada saat pemain berada dalam *game,* terdapat beberapa *Head-up Design (HUD)* yang berisi tentang informasi waktu, koin yang sudah dikumpulkan, dan senjata yang tersedia, seperti yang ditunjukkan pada gambar 3.7.

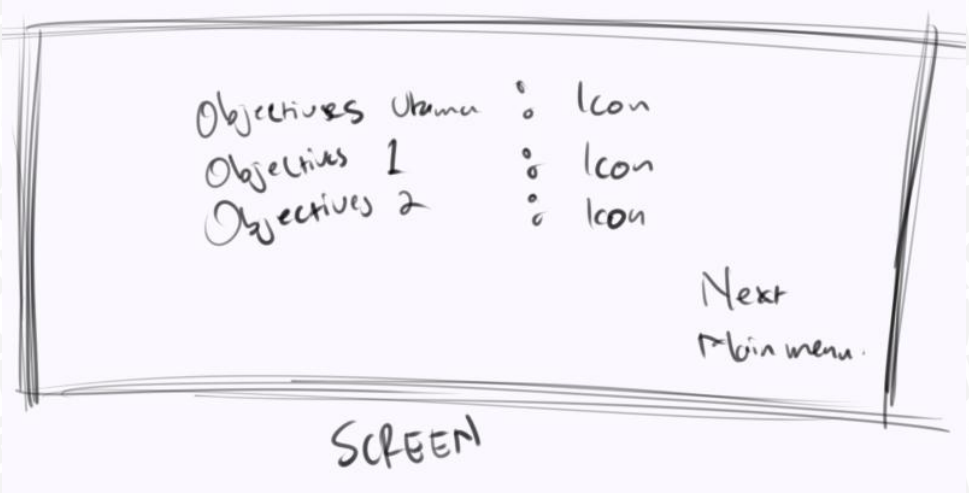


Gambar 3. 7 Desain In-game

Desain dari senjatanya berbentuk bungkusan yang dapat dilihat pada gambar 3.8. Bungkusan-bungkusan tersebut dibedakan berdasarkan warna merah, kuning, dan nila. Ketiga warna tersebut digunakan untuk pembeda pada setiap bungkusan, merah untuk terasi, kuning untuk jarum, dan nila untuk garam.

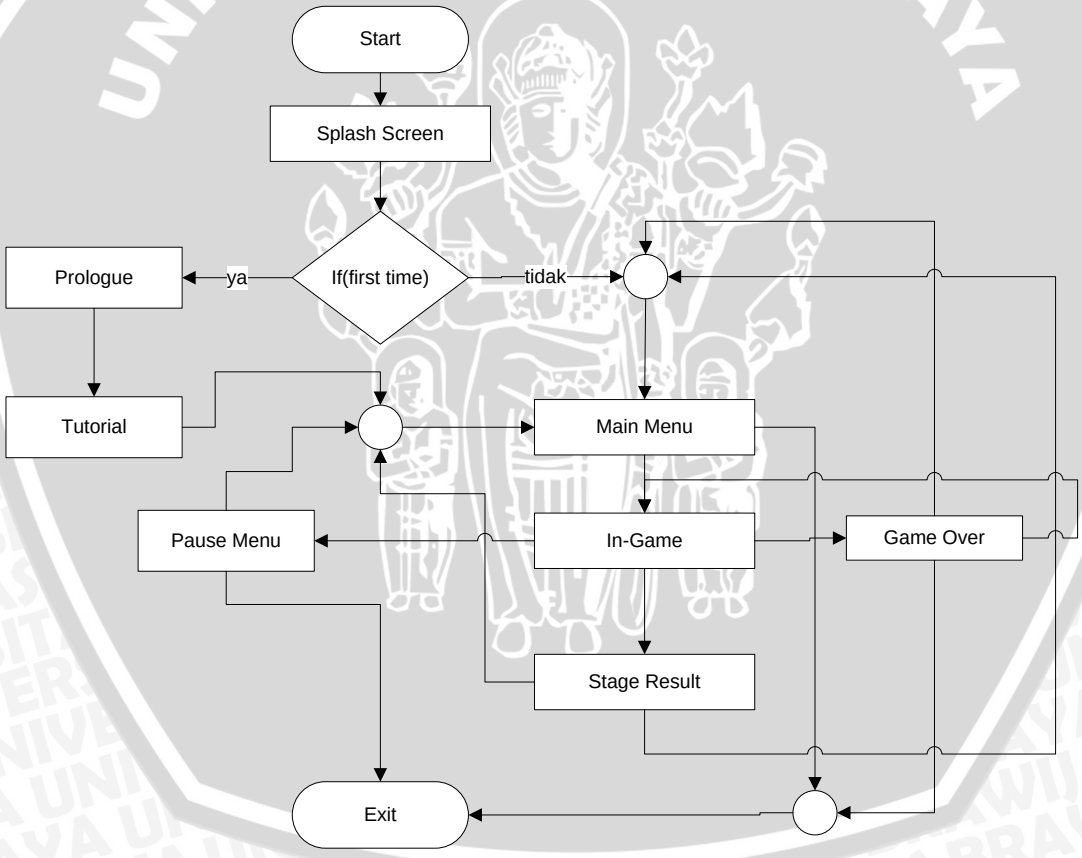
Gambar 3. 8 Desain Bungkusan

Setelah pemain menyelesaikan *game*, pemain masuk pada *stage result scene* yang berisi tentang review pada *level* yang pemain mainkan sebelumnya. Pada *scene* ini pemain dapat melanjutkan ke *level* selanjutnya, atau kembali ke menu utama. Untuk lebih jelasnya dapat dilihat pada gambar 3.9.



Gambar 3. 9 Desain Score Result

3.2.5. Game Screen Flow



Gambar 3. 10 Game Screen Flow

Pada gambar 3.10, saat pertama kali pemain memulai *game*, pemain akan memasuki *splash screen.* Jika pemain pernah memainkan *game* Timun Emas

sebelumnya, setelah memasuki *splash screen* pemain akan langsung menuju menu utama. Jika belum, pemain akan memasuki *prologue* kemudian memasuki *tutorial* untuk memahami cerita awal dan cara bermain pada *game* ini. Pada menu utama, pemain dapat langsung menuju *level select*, atau melakukan pengaturan suara terlebih dahulu. Pada *level select* pemain dapat langsung memilih *level* untuk dimainkan.

Pada saat pemain berada pada permainan, pemain dapat memilih *pause menu* untuk menghentikan *game flow*. Pada *pause menu*, pemain dapat memilih antara melanjutkan permainan, mengulang *level*, kembali ke menu utama, atau menutup aplikasi. Jika pemain gagal dalam menyelesaikan suatu level, pemain akan masuk pada *scene game over*. Pada *scene* ini terdapat beberapa pilihan, diantaranya pemain dapat mengulang stage tersebut, kembali ke menu utama atau menutup *game.* Setiap pemain selesai menyelesaikan *level*, pemain akan memasuki *stage result. Scene* ini akan merekap setiap *achievements* yang telah dikumpulkan oleh pemain pada *level* yang dimainkan sebelumnya. Pada *scene* ini pemain dapat memilih melanjutkan ke *level* berikutnya atau kembali ke menu utama.

BAB IV
IMPLEMENTASI

Bab ini membahas implementasi dari pembuatan *game* Timun Emas dengan spesifikasi sesuai dengan bab sebelumnya.

4.1.Spesifikasi Sistem

Game ini dikembangkan dalam lingkungan implementasi yang terdiri dari perangkat keras dan perangkat lunak.

Tabel 4. 1 Tabel Spesifikasi Hardware

Spesifikasi *Hardware*	
Processor	Intel Core i3 CPU M370 @2.40GHz (4CPUss), ~2.5GHz
Memory (RAM)	2048MB
Hard disk	512GB HDD
Graphic Card	ATI Mobility Radeon HD 5470
Spesifikasi *Software*	
Operating System	*Windows 7 Home Premium 32bit*
DirectX Version	*DirextX 11*
Programming Language	*C#*
Graphics Editor	*Adobe Photoshop CS5* *SAI Paint Tool*
Integrated Development Environment	*Microsoft Visual Studio*

4.2.Pemilihan Teknologi dan Platform

Dalam pembuatan *game* Timun Emas, dibutuhkan teknologi pendukung seperti *game engine,* serta *open source asset* yang dapat memudahkan proses pengembangan *game.*

Unity3D digunakan sebagai *Game Engine* utama dalam pengimplementasian *game* Timun Emas ini. Unity pada versi 4.0 ke atas mendukung pembuatan *game* dua dimensi sehingga sangat membantu dalam pembuatan *game* ini. *Unity3D* memiliki banyak *library* yang telah terpasang bersama *Unity* pada saat instalasi.

Untuk *platform* yang digunakan adalah *platform* ponsel *android.* Android merupakan salah satu *operating system smartphone* yang banyak dipakai di Indonesia. Untuk spesifikasi minimal dapat dilihat di tabel dibawah ini:

Tabel 4. 2 Tabel Spesifikasi Minimal Aplikasi

Operating System	Android versi 2.2
RAM	256MB
Space Disk	~52.0Mb
CPU	CPU ARMv7

4.3.Implementasi Prosedur Program

Dalam implementasi prosedur program, terdapat faktor utama dalam *game* ini, diantaranya adalah implementasi obyek pada setiap karakter, implementasi desain karakter, implementasi desain *platform*, implementasi suara narrator, hingga implementasi suara musik dan suara efek.

4.3.1. Implementasi Prosedur Program Karakter

Tabel 4. 3 Pseudo code Prosedur Program Timun Emas

```
Pseudo Code Timun Emas
DECLARATION
    anim is Animator
    MaxSpeed is float
    Speed is float
    isGround, isEnded, die is Boolean
    turun is Animator
    DESCRIPTION
1.  START
2.  anim.SetFloat("Speed", Mathf.Abs(speed))
3.
4.  anim.SetBool("Jump", isGround)
5.      IF(Rigidbody2D.velocity.x < MaxSpeed &&
6.      Time.TimeScale == 1) THEN
7.          Addforce(Vector2D.right * speed)
8.      END IF
9.
10.     IF(die)THEN
11.         isEnded <- TRUE
12.         CALL fungsi GameOver()
13.     END IF
14. END
```

Pada tabel 4.3 merupakan implementasi dari karakter Timun Emas. Timun Emas akan berjalan pada saat menyentuh *platform.* Jika Timun Emas berinteraksi dengan *game objects* lain selain *platform* seperti *ground* dan Raksasa, maka permainan berakhir. Jika Timun Emas berinteraksi dengan *goal point,* maka pemain telah berhasil menyelesaikan permainan dan akan masuk ke *stage result scene.*

Tabel 4. 4 Pseudo code Prosedur Program Raksasa

```
Pseudo Code ButoIjo
DECLARATION
    Speed is FLOAT
1.  DESCRIPTION
2.  START
3.      Jalan dengan kecepatan speed;
        IF(collider raksasa berinteraksi dengan collider
4.
5.      obj)
6.          IF(obj == player collider)gameover();
            ELSE IF(obj == weapons collider)
7.             jatuh(); max speed= max speed +((n/100)*max
8.          speed);}
9.          END IF;
10.     END IF;
    END
```

Pada tabel 4.4, Raksasa akan berjalan mengejar Timun Emas dari belakang. Raksasa akan menghiraukan *platform* yang ada. Dalam *game,* Raksasa hanya berinteraksi dengan senjata (bungkusan) dan Timun Emas. Jika Raksasa berinteraksi dengan bungkusan yang telah dipasang, Raksasa akan terjatuh dan melambat, namun pada saat berjalan lagi kecepatan maksimal Raksasa akan bertambah.

4.3.2. Implementasi Prosedur Program Achievements

Terdapat tiga *achievement* yang dapat diperoleh pemain pada setiap *stage*. Setiap *achievement* memiliki kondisi yang berbeda, seperti *achievement* waktu yang didapat ketika pemain dapat menyelesaikan *level* sebelum *timer* yang tertera pada setiap *stage* habis. Sama halnya dengan koin, untuk mendapatkan *achievement* ini pemain harus mengumpulkan tiga koin yang tersebar pada masing-masing level.

Sementara untuk *achievement goal*, *achievement* ini merupakan *achievement* utama, dimana pemain harus menyelesaikan *achievement* ini untuk dapat melanjutkan *game* ke *level* selanjutnya.

Tabel 4. 5 Pseudo code Prosedur Program Koin

CoinBehaviour.cs	
1.	counter = 0;
2.	Update()
3.	{
4.	if (coin collider berinteraksi dengan collider
5.	obj)
6.	{
7.	if(obj == collider player)
8.	{
9.	counter++;
10.	destroy coin;
11.	}end if;
12.	}
13.	Switch(counter)
14.	{
15.	Counter = 1: tampilkan koin1 di HUD;
16.	Counter = 2: tampilkan koin2 di HUD;
17.	Counter = 3:
18.	{
19.	tampilkan koin3 di HUD;
20.	setScore();
21.	}
22.	}
	}

Pada saat semua koin yang telah dikumpulkan pada setiap level, data tersebut akan dikirim pada suatu *class* yang bernama *Score* yang akan menyimpan data *achievement* pada *stage* yang dijalankan oleh pemain, sama seperti *achievement timer* dan *goal* seperti yang dijelaskan pada tabel 4.5.

4.3.3. Implementasi Prosedur Program Platform

Berdasarkan dari *game* mekanis, terdapat beberapa macam tipe *platform*, *platform* yang tidak dapat digerakkan, *platform* yang bergerak secara horizontal, *platform* yang bergerak secara vertikal dan *platform* yang akan jatuh jika Timun Emas berada ditengahnya. Untuk membedakan pada saat dimainkan, warna dan pola bentuk pada setiap *platform* dibedakan agar pemain tidak bingung.

Tabel 4. 6 Pseudo Code Prosedur Program Platform

	Pseudo code Platform
	DECLARATION
	Awal is Vector3 Akhir is Vector3 maju, aktif is Boolean speed is float endY is float
	DESCRIPTION
1.	START
2.	Awal <- transform.position
3.	Akhir.y <- endY
4.	IF(INPUT mouse/touch)THEN
5.	aktif = true
6.	maju = !maju
7.	END IF
8.	IF(aktif && maju)THEN
9.	transform.position = Vector3.lerp(awal,
10.	akhir, speed)
11.	END IF
12.	
13.	ELSE IF aktif && !maju)THEN
14.	transform.position = Vector3.lerp(akhir,
15.	awal, speed)
16.	END IF
17.	END

Pada tabel 4.6, penggunaan dua *boolean* untuk membatasi *platform* agar bergerak sesuai yang jarak, dan posisi *platform* tersebut. Jika *platform* telah bergerak secara horizontal ke kiri, maka pada saat pengaktifan selanjutnya *platform* akan bergerak secara horizontal ke kanan, begitu juga sebaliknya.

Tabel 4. 7 Pseudo code prosedur program platform yang jatuh

	Pseudo code pada script Falling
	DECLARATION
	plt is GameObject falling is Boolean sprite is SpriteRenderer
	DESCRIPTION
1.	START
2.	falling = false;
3.	sprite = plt.GetComponent<SpriteRenderer>()
4.	IF(falling)THEN
5.	plt.collider2D.isTrigger <- true

```
6.              plt.transform.translate(0f, -8.5f, 0f)
7.          END IF
8.  END
```

Pada tabel 4.7, terdapat sebuah obyek yang diberi nama *checker* untuk mengaktifkan *method* jatuh pada saat Timun Emas menabrak *collider checker* tersebut. *Layer* platform diganti menjadi *default* agar Timun Emas tidak melakukan animasi berlari ketika *platform* mulai terjatuh. Timun Emas akan melakukan animasi berlari ketika Timun Emas mendarat pada *platform* yang memiliki nama *layer ground* dan *touch input.*

4.3.4. Implementasi Prosedur Penyimpanan Score pada Stage

Pada *scene ScoreResult,* terdapat *class* utama yang digunakan untuk menyimpan data *achievement* yang telah didapatkan oleh pemain pada *level* sebelumnya. Namun penyimpanan ini hanya bersifat sementara karena *scene ScoreResult* akan dipanggil pada saat pemain menyelesaikan pada setiap *level.*

Sebelum pemain memasuki *scene* ini, *achievement* yang telah didapatkan pemain pada setiap *level* akan langsung dikirim ke *class ScoreResult.*

Tabel 4. 8 Source Code ScoreResult.cs

ScoreResult.cs

```
1.  public static void setCoin(bool a)
2.      {
3.         coin = a;
4.         if (!PlayerData.LevelDatas[id].CoinFlag)
5.         PlayerData.LevelDatas[id].CoinFlag = coin;
6.      }
7.      public static void setFlag(bool a)
8.      {
9.         flag = a;
10.        if(!PlayerData.LevelDatas[id].CompleteFlag)
11.        PlayerData.LevelDatas[id].CompleteFlag = flag;
12.     }
13.     public static void setTimer(bool a)
14.     {
15.        timer = a;
16.        if(!PlayerData.LevelDatas[id].TimeFlag)
17.        PlayerData.LevelDatas[id].TimeFlag = timer;
18.     }
19.     public static bool getTimer()
20.     {
21.         return timer;
22.     }
```

```
23.     public static bool getFlag()
24.     {
25.         return flag;
26.     }
27.     public static bool getCoin()
28.     {
29.         return coin;
30.     }
```

Pada tabel 4.8, sebelum data dari *level* tersebut dikirim ke *class playerdata*, *class* ini akan mengecek terlebih dahulu, apakah pemain sebelumnya telah mendapatkan *achievement* tersebut atau belum. Hal ini bertujuan untuk menghindari tumpang tindih dari data yang dikirim.

4.3.5. Implementasi Prosedur Penyimpanan data Setiap Stage

Tabel 4. 9 Source Code LevelData.cs

LevelData.cs

```
1.  public class LevelData
2.  {
3.      public int ID { get; private set; }
4.      public string Name { get; private set; }
5.      public bool CompleteFlag { get; set; }
6.      public bool TimeFlag { get; set; }
7.      public bool CoinFlag { get; set; }
8.
9.      public LevelData(int ID, string Name)
10.     {
11.         this.ID = ID;
12.         this.Name = Name;
13.         CompleteFlag = false;
14.         TimeFlag = false;
15.         CoinFlag = false;
16.     }
}
```

Pada tabel 4.9, untuk menyimpan data pada *achievement*, dibuat sebuah *class* yang menyimpan setiap *achievement* pada sebuah obyek. Setelah dibuat *class* seperti gambar diatas, dibuat *class* lain yang dapat digunakan untuk membuat *static object* dari *class* tersebut untuk dapat membuat *object* pada setiap *stage* yang ada.

Tabel 4. 10 Source Code PlayerData.cs

PlayerData.cs

```
1.  public class PlayerData
2.  {
3.      public static List<LevelData> LevelDatas =
```

```
4.  defaultLevelData();
5.     private static List<LevelData> defaultLevelData()
6.      {
7.          List<LevelData> val = new List<LevelData>();
8.
9.          val.Add(new LevelData(0, "0"));
10.         val.Add(new LevelData(1, "1"));
11.         val.Add(new LevelData(2, "2"));
12.         val.Add(new LevelData(3, "3"));
13.         val.Add(new LevelData(4, "4"));
14.         val.Add(new LevelData(5, "5"));
15.         val.Add(new LevelData(6, "6"));
16.         val.Add(new LevelData(7, "7"));
17.         val.Add(new LevelData(8, "8"));
18.         val.Add(new LevelData(9, "9"));
19.         return val;
20.     }
21. }
```

Pada tabel 4.10, untuk menyimpan data pada setiap *level* dibuat sebuah *array list* yang berjumlah sama dengan jumlah *level* yang ada pada *game.* Berdasarkan *LevelData.class,* pada setiap array akan menyimpan id dan nama *level*, yang berisikan nomor *level*, *achievements flag*, *achievement coin*, dan *achievement time.* Selain untuk mempermudah dalam penyimpanan data *game*, dengan menggunakan *static object* dapat mempermudah pengaksesan data pada setiap *scene* berdasarkan dengan id yang telah diinisialisasi pada setiap *stage*.

4.3.6. Implementasi Algoritma Tutorial

Pada tutorial, terdapat beberapa *script* tambahan yang bertujuan untuk memberikan jeda pada beberapa tempat. Pada saat jeda, pemain diberi beberapa instruksi berupa teks. Hal ini bertujuan agar pemain dapat mengerti apa yang harus dilakukan pada saat menemukan beberapa masalah, diantaranya jalan yang berada di depan Timun Emas terputus. Untuk lebih jelasnya dapat dilihat pada tabel 4.11.

Tabel 4. 11 Pseudo code Prosedur Tutorial

Pseudo code Tutorial	
DESCRIPTION Checker is BoxCOllider2D Dialog is STRING[]	
1.	MULAI

```
2.      Print dialog[0]
3.      Switch(checker.index)
4.          Case 1: print dialog[1]
5.          Case 2: print dialog[2]
6.          Case 3: print dialog[3]
7.          Case 4: print dialog[4]
8.          Case 5: print dialog[5]
9.      AKHIR SWITCH
10.     IF(player collider berinteraksi dengan goal
11.     collider)
12.         Set first tutorial = true
13.         Go to "Main Menu"
14.     END IF
15. AKHIR
16.
    Method OnMouseUp()
17. START
18.     jeda = false
19.     dialog clear
    END
```

4.4.Implementasi Art

Dalam implementasi desain pada *game* Timun Emas, bagian utama yang paling berpengaruh adalah pengimplementasian desain karakter, desain *platform*, dan desain latar belakang pada setiap level. Berikut bagian-bagian dari implementasi *art* pada *game* ini:

4.4.1 Implementasi Komik

Desain komik pada *game* ini ditampilkan pada saat awal pemain membuka *game* ini. Ini bertujuan agar pemain mengetahui cerita awal Timun Emas, sehingga pada saat memainkan *game* ini pemain sudah mengerti konsep dasar dari tema *game* ini. Desain komik dapat dilihat pada gambar 4.1 dan gambar 4.2.

Gambar 4. 1 Implementasi Komik I

Gambar 4. 2 Implementasi Komik 2

4.4.2 Implementasi Desain Karakter

Pada setiap karakter memiliki desain masing-masing. Timun Emas, Raksasa, serta Mbok Rondo memiliki desain masing-masing.

Timun Emas didesain sebagai anak perempuan yang berumur sekitar enam belas tahun yang memakai kain batik berwarna dasar cokelat yang mencerminkan orang suku Jawa yang hidup pada jaman dahulu, seperti yang ditunjukkan pada gambar 4.3.

Gambar 4. 3 Implementasi Desain Karakter Timun Emas

Berdasarkan gambar 4.4, Mbok Rondo yang merupakan ibu dari Timun Emas, merupakan seorang wanita tua yang sangat sayang kepada Timun Emas. Pada *game* Mbok Rondo hanya muncul pada cerita awal dan akhir saja.

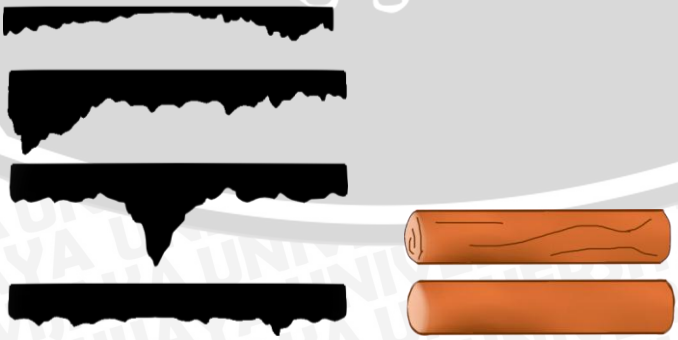
Gambar 4. 4 Implementasi Desain Karakter Mbok Rondo

Raksasa adalah karakter antagonis pada *game* ini. Raksasa digambarkan mempunyai ukuran yang besar dan kuat. Desain dari karakter raksasa dapat dilihat pada gambar 4.5

Gambar 4. 5 Implementasi Desain Karakter Raksasa

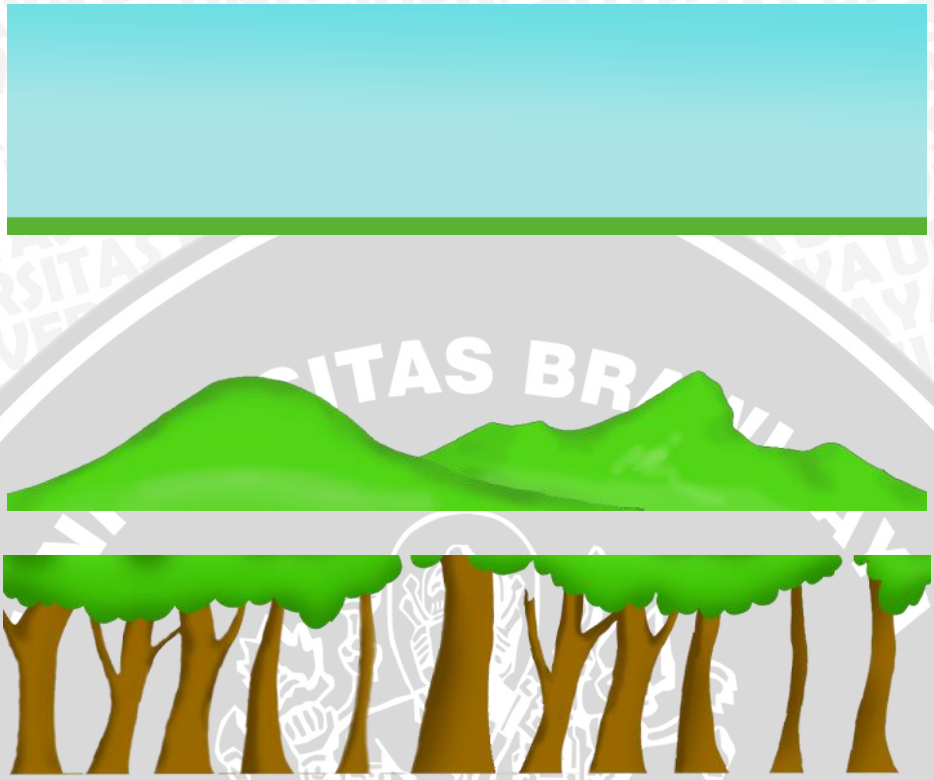
4.4.3 Implementasi Desain Platform dan Latar Belakang

Berdasarkan gambar 4.6, desain *platform*, terdapat beberapa macam yang tersedia pada *game* ini. Yang membedakan dari setiap *platform* itu adalah fungsi dan bentuk. Menurut fungsinya, *platform* yang ada dibedakan menjadi *platform* yang tidak bisa digerakkan, *platform* yang bergerak secara horizontal, platform yang bergerak secara vertikal, dan *platform* yang akan jatuh jika Timun Emas berada diatasnya. Sedangkan menurut bentuknya, *platform* dibagi menjadi kayu dan stalaktit.



Gambar 4. 6 Implementasi Desain Platform

Berdasarkan gambar 4.7, desain latar belakang dibagi menjadi beberapa macam sesuai dengan tema yang ada pada setiap level.

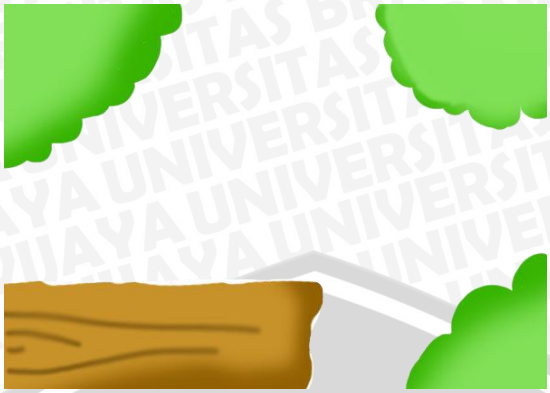


Gambar 4. 7 Implementasi Desain Latar Belakang

4.4.4 Implementasi Desain HUD

Head-up Display atau yang biasa disingkat HUD, merupakan sebuah *user interface* yang menampilkan informasi dari sebuah *game*, seperti waktu, skor, tombol *pause*, dan sebagainya.

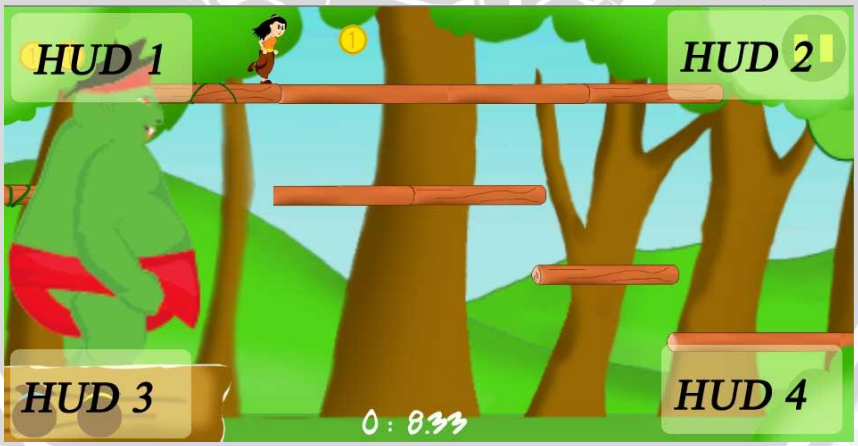
Gambar 4. 8 Implementasi Desain Bungkusan



Gambar 4. 9 Implementasi Desain HUD

Pada gambar 4.10 desain HUD dibuat pada masing-masing sudut layar, yang menampilkan:

1. HUD 1 yang berada di pojok kiri atas, menampilkan koin yang sudah dikumpulkan.
2. HUD 2 yang berada di pojok kanan atas, menampilkan tombol *pause*.
3. HUD 3 yang berada di pojok kiri bawah, menampilkan senjata yang dapat digunakan oleh pemain
4. Waktu tersisa yang dibutuhkan untuk menyelesaikan setiap *level*, berada di tengah.



Gambar 4. 10 Implementasi Penempatan Desain HUD

4.4.5 Implementasi Desain Main Menu

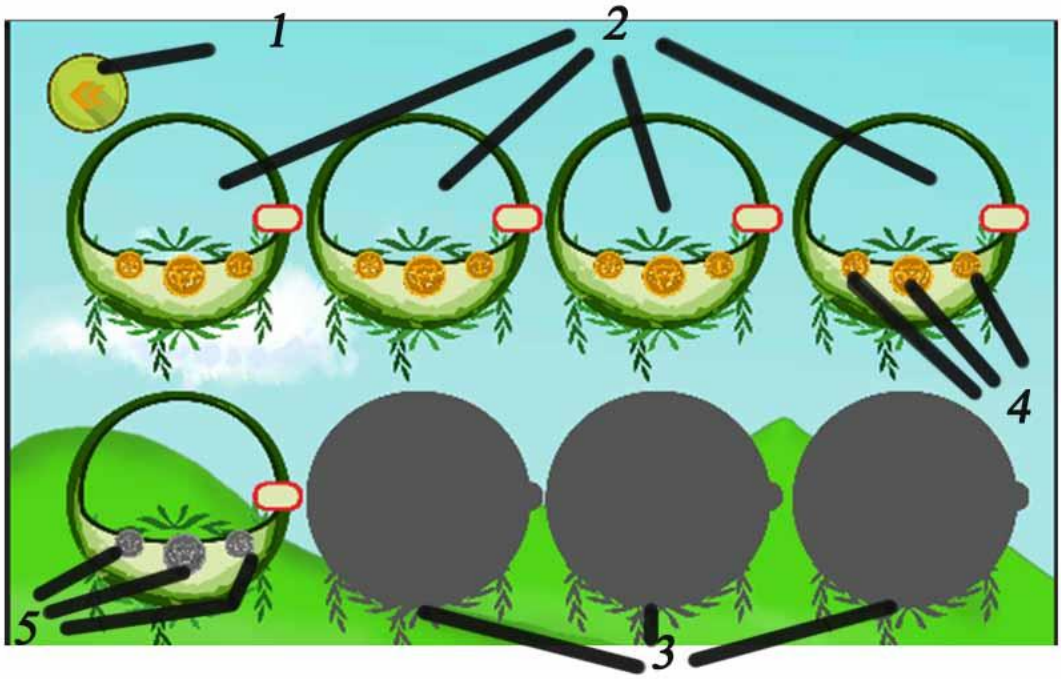
Main menu merupakan tampilan awal sebelum pemain dapat bermain langsung pada *game*. Pada *main menu*, pemain diberikan beberapa pilihan utama, diantaranya:

1. *Play*, digunakan untuk memasuki menu *level*. Pada menu *level*, pemain dapat memilih *level* yang ingin dimainkan.
2. *Setting*, digunakan untuk melakukan pengaturan pada musik dan efek suara.
3. *Exit*, digunakan untuk menutup aplikasi.

Gambar 4. 11 Implementasi Desain Menu Utama

4.4.6 Implementasi Desain Icon Level

Setiap *level* pada *game* Timun Emas ditunjukkan dengan gambar *icon*. Berdasarkan icon yang ada, setiap *level* dibedakan menjadi dua, yaitu *level* yang bisa dimainkan dan *level* yang belum bisa dimainkan. Syarat untuk dapat memainkan setiap pada *level* adalah pemain harus menyelesaikan *level* sebelumnya dengan mendapatkan minimal *achievement flag* yang akan didapatkan ketika pemain menyelesaikan suatu *level*.



Gambar 4. 12 Implementasi Desain Stage Select

Keterangan:

1. Tombol *Back.*
2. *Level* yang dapat dimainkan.
3. *Level* yang belum dapat dimainkan.
4. *Achievement* yang telah didapatkan pada *level*.
5. *Achievement* yang belum didapatkan pada *level*.

Pada gambar 4.12, setiap *icon* tersebut terdapat informasi tentang *achievement* apa saja yang telah didapatkan oleh pemain pada setiap *level.* Jika pemain telah mendapatkan *achievement* tersebut warnanya akan berbuah menjadi kecokelat-cokelatan, jika belum didapatkan warnanya adalah abu-abu.

Gambar 4. 13 Implementasi Desain Achievements

Pada gambar 4.13, terdapat tiga *achievement* diantaranya adalah *achievement flag* yang berada di tengah. *Achievement* ini didapatkan ketika pemain menyelesaikan suatu *level*, *achievement coin* berada pada sebelah kiri, dimana pemain harus mengumpulkan semua koin pada suatu *level* untuk mendapatkan *achievement* ini. Dan *achievement time* yang berada pada sebelah kanan. Pemain harus menyelesaikan *level* pada waktu tertentu untuk mendapatkan *achievement ini.*

4.4.7 Implementasi Sound Effect, Suara dan Music

Suara-suara yang digunakan pada *game* ini menggunakan *open source* yang disediakan pada beberapa situs penyedia *sound effect* untuk *game,* seperti opengameart.org. Selain menggunakan *sound effect* yang tersedia secara *open source,* suara narator yang digunakan pada *prologue* dan suara latar yang digunakan pada menu utama direkam dan diolah menggunakan *audacity.*

Berikut daftar *sound* yang digunakan pada *game* Timun Emas berikut dengan sumber:

1. misc_menu_4.wav, digunakan untuk efek suara pada saat menekan tombol. Sumber: *http://opengameart.org/content/gui-sound-effects*
2. applause_1.wav, digunakan untuk suara latar pada *scene stage result.* Sumber: *www.pacdv.com/sounds/people_sound_effects/*
3. With love from Vertex Studio (2), With love from Vertex Studio (3), With love from Vertex Studio (16), With love from Vertex Studio (24), With love from Vertex Studio (26), With love from Vertex Studio (32), With love from Vertex Studio (41), digunakan untuk suara latar pada *level* dan

suara latar pada prologue. Sumber *https://www.assetstore.unity3d.com/en/#!/content/4883*

4. Steel Guitar.wav, digunakan untuk suara latar pada *scene* menu utama. Sumber: penulis.
5. Folder narrator, berisi suara narrator yang pada setiap kolom. Sumber: penulis.

4.5.Implementasi Game Screen dan Realisasi Level

Dalam merealisasikan level pada *game* ini, terdapat sebuah *class* yang menyimpan data pada setiap level. Class ini juga menyimpan id pada setiap level, yang nantinya id tersebut digunakan untuk memanggil level sesuai dengan pemain pilih pada menu utama. Setiap tombol level pada menu utama menyimpan id level pada *PlayerData* dan id level yang disimpan pada *Unity*. Hal ini dikarenakan pada saat membangun *game* pada *unity*, setiap *scene* diberikan id berbeda untuk memudahkan pada saat pemanggilan dilakukan.

Game Screen merupakan tampilan dari sebuah *game* pada setiap *scene* yang ada. Pada *game* Timun Emas, *Scene* utama terdiri dari *splash screen, prologue scene, tutorial, main menu, stage* gua, *stage* hutan, *stage* padang rumput, *stage result*, dan *credit.*

4.5.1. Splash Screen

Splash Screen merupakan tampilan awal yang ada pada setiap *game.* Pada umumnya, *splash screen* menampilkan lambang, logo atau nama dari pihak-pihak dalam pembuatan *game* seperti *publisher*, atau sponsor.

Gambar 4. 14 Implementasi Scene Splash Screen

4.5.2. Prologue

Pada *scene* ini berisi cerita awal Timun Emas. Mulai dari pada saat Mbok Rondo bertemu dengan Raksasa, sampai pada cerita dimana Timun Emas berusaha kabur dari Raksasa.

Gambar 4. 15 Implementasi Scene Komik

4.5.3. Main Menu

Seperti yang dijelaskan pada implementasi desain *main menu*, pada *scene* ini menampilkan beberapa pilihan menu yang dapat pemain pilih. Pada *scene* ini pemain dapat langsung bermain, atau melakukan pengaturan dulu pada pilihan *setting*¸ yang berada pada sebelah kanan tombol *play*.

Gambar 4. 16 Implementasi Scene Menu Utama

Setelah pemain menekan tombol *play* yang berada ditengah, tampilan akan bergeser dan menunjukkan pilihan level yang bisa dimainkan oleh pemain.

Pemain dapat kembali ke menu awal dengan menekan tombol kembali pada layar, atau tombol pada ponsel.

4.5.4. Tutorial

Gambar 4. 17 Implementasi Scene Tutorial

Pemain akan memasuki tutorial pada saat pertama kali membuka *game* Timun Emas. Pada tutorial pemain akan diberi arahan singkat tentang mekanis *game* pada saat bermain. Tingkat kesulitan pada level ini sangat mudah karena level ini bertujuan agar pemain dapat mengerti konsep dasar dari *game* ini. Pemain diberikan instruksi dasar dari cara memainkan *game* Timun Emas, seperti mengetahui *platform* mana yang dapat digerakkan dan *platform* yang jatuh jika Timun Emas berada ditengahnya.

4.5.5. Stage Gua

Gambar 4. 18 Implementasi Scene Stage Gua

Pada *stage* ini tingkat kesulitan yang digunakan berada pada tingkatan mudah. Raksasa hanya mengejar dengan kecepatan yang konstan. Tingkat kerumitan

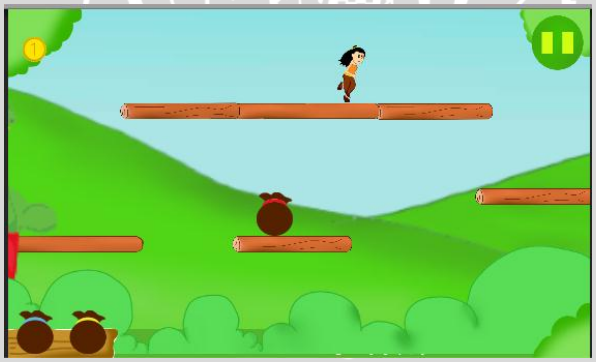
platform juga dibatasi tanpa ada percabangan untuk memudahkan pemain dalam menyelesaikan *stage* ini.

4.5.6. Stage Hutan

Gambar 4. 19 Implementasi Scene Stage Hutan

Setelah keluar dari gua, pemain akan memasuki *stage* hutan. Pada *stage* ini, kecepatan Raksasa akan bertambah setiap kali Raksasa terjatuh. Pada tingkat kerumitan *platform*, terdapat beberapa percabangan yang salah satu diantaranya akan memandu pemain menuju koin untuk menyelesaikan salah satu *achievement*.

4.5.7. Stage Padang Rumput



Gambar 4. 20 Implementasi Scene Stage Padang Rumput

Pada *stage* ini, tingkat kerumitan *platform* ditambah, seperti terdapat beberapa percabangan, penempatan *platform* jebakan diperbanyak, serta waktu yang dibutuhkan untuk menyelesaikan *stage* akan dikurangi.

4.5.8. Stage Danau

Gambar 4. 21 Implementasi Scene Stage Danau

Stage ini berisi *level* terakhir dan pada akhir. Raksasa tidak ikut mengejar Timun Emas lagi pada *level* ini, tetapi jumlah *platform* ditambah terutama *platform* jebakan yang jatuh jika Timun Emas berdiri diatasnya. Pada akhir *level* terdapat rumah dari Timun Emas dan Mbok Rondo.

4.5.9. Stage Result

Setiap kali pemain berhasil menyelesaikan *stage*, pemain akan memasuki *scene* ini. *Scene* ini berisi tentang review dari *stage* yang sudah diselesaikan.

Gambar 4. 22 Implementasi Scene Score Result

Pada *scene* ini pemain dapat memilih apakah akan melanjutkan *game* ke *stage* berikutnya atau kembali ke menu utama sesuai dengan gambar 4.22.

BAB V
PENGUJIAN DAN ANALISIS

Bab ini membahas mengenai tahapan pengujian dan analisis *game* Timun Emas yang telah dirancang dan diimplementasikan. Pengujian berfungsi untuk memantau sejauh mana *game* Timun Emas berjalan sesuai ekspektasi penulis, serta untuk menemukan kesalahan yang terdapat pada permainan. Tahap pengujian dari *game* Timun Emas ini terdiri dari pengujian pengujian *whitebox*, *test suite* dan *play testing.*

5.1 Pengujian Whitebox

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

5.1.1. Pengujian Unit

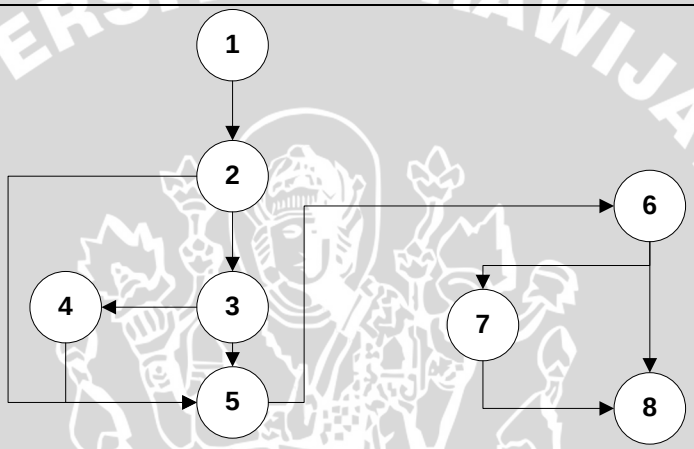
1. Pengujian Unit Timun Emas

Timun Emas merupakan obyek utama pada *game* ini. Pada tabel 5.1 menunjukkan sebagian besar algoritma yang digunakan pada obyek Timun Emas, diantaranya seperti untuk berlari, menampilkan animasi lari, dan kondisi pada saat Timun Emas mati. Pemodelan obyek Timun Emas dalam bentuk *flow graph* ditunjukkan pada tabel 5.1

Tabel 5. 1 Pseudo code Timun emas

Pseudo Code Timun Emas
DECLARATION anim is Animator MaxSpeed is float Speed is float isGround, isEnded, die is Boolean turun is Animator

```
    DESCRIPTION
    START
1.  anim.SetFloat("Speed", Mathf.Abs(speed))
2.  anim.SetBool("Jump", isGround)
3.      IF(Rigidbody2D.velocity.x < MaxSpeed &&
4.       Time.TimeScale == 1) THEN
5.           Addforce(Vector2D.right * speed)
6.      END IF
7.      IF(die)THEN
8.           isEnded <- TRUE
9.           CALL fungsi GameOver()
10.     END IF
11. END
```



Gambar 5. 1 *Flow graph* pengujian obyek Timun Emas

Pemodelan ke dalam *flow graph* yang telah dilakukan untuk obyek Timun Emas menghasilkan jumlah kompleksitas siklomatis *(cyclomatic complexity)* melalui tiga persamaan, V(G) = E – N + 2, dimana V(G) merupakan jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul *(node).* Pada persamaan V(G) = Region, dimana region merupakan daerah pada percabangan. Dan persamaan V(G) = Predikat node +1, dimana predikat node adalah sebuah node yang memiliki dua atau lebih jalur (percabangan).

V(G) = E – N + 2 V(G) = Jumlah Region V(G) = Predikat+1

= 10 – 8+ 2 = 4 = 4 = 4

Berdasarkan dari nilai *cyclomatic complexity* yang telah didapatkan dari perhitungan maka ditentukan tiga buah basis set dari jalur *independent*, yaitu:

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

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

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

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

Tabel 5. 2 Tabel Kasus Uji untuk obyek Timun Emas

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Kecepatan karakter belum maksimal dan karakter belum mati	Kecepatan dari karakter akan bertambah sampai hingga kecepatan mencapai angka maksimal	Kecepatan dari karakter akan bertambah sampai hingga kecepatan mencapai angka maksimal
2	Kecepatan karakter mencapai angka maksimal dan karakter belum mati	Karakter akan berlari dengan kecepatan konstan dengan nilai sama dengan kecepatan maksimal	Karakter akan berlari dengan kecepatan konstan dengan nilai sama dengan kecepatan maksimal
3	Kecepatan karakter belum maksimal dan karakter mati	Kecepatan dari karakter akan diubah menjadi nol, timescale diubah menjadi nol dan animasi turun dimainkan	Kecepatan dari karakter akan diubah menjadi nol, timescale diubah menjadi nol dan animasi turun dimainkan
4	Kecepatan karakter mencapai angka maksimal dan karakter mati	Kecepatan dari karakter akan diubah menjadi nol, timescale diubah menjadi nol dan animasi turun dimainkan	Kecepatan dari karakter akan diubah menjadi nol, timescale diubah menjadi nol dan animasi turun dimainkan

2. Pengujian Unit Obyek Platform

Obyek *platform* dibedakan menjadi beberapa macam. Pada pengujian ini, dilakukan pada *platform* yang dapat digerakkan oleh pemain, diantaranya

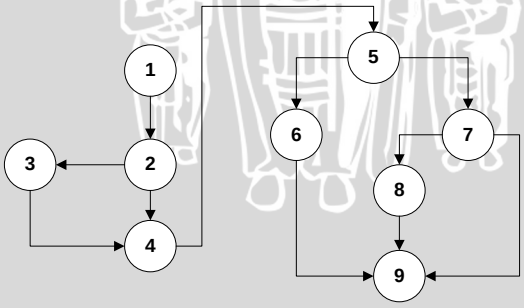
platform yang menggunakan *script horizontal*, *vertical*, dan *falling*. Berikut merupakan pseudo code dari *script* tersebut.

Tabel 5. 3 Pseudo code script Horizontal.cs

Pseudo code pada script Horizontal.cs

```
DECLARATION
    Awal is Vector3
    Akhir is Vector3
    maju, aktif is Boolean
    speed is float
    endX is float
```

```
    DESCRIPTION
1.  START
2.      Awal <- transform.position                    1
3.      Akhir.x <- endX
4.      IF(INPUT mouse/touch)THEN                     2
5.          aktif = true                              3
6.          maju = !maju
7.      END IF                                        4
8.      IF(aktif && maju)THEN                         5
9.          transform.position  =  Vector3.lerp(awal,     6
10.         akhir, speed)                                 7
11.     ELSE IF aktif && !maju)THEN                       8
12.         transform.position  =  Vector3.lerp(akhir,
13.         awal, speed)
            9
        END IF
    END
```



Gambar 5. 2 *Flow graph* pengujian *script* Horizontal

Pemodelan ke dalam *flow graph* yang telah dilakukan untuk *script* horizontal menghasilkan jumlah kompleksitas siklomatis *(cyclomatic complexity)* melalui persamaan V(G) = E – N + 2, dimana V(G) merupakan jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul *(node)*. Pada persamaan V(G) = Region,

dimana region merupakan daerah pada percabangan. Dan persamaan V(G) = Predikat node +1, dimana predikat node adalah sebuah node yang memiliki dua atau lebih jalur (percabangan).

V(G) = E – N + 2 V(G) = Jumlah Region V(G) = Predikat+1

= 11 – 9+ 2 = 4 = 4 = 4

Berdasarkan dari nilai *cyclomatic complexity* yang telah didapatkan dari perhitungan maka ditentukan tiga buah basis set dari jalur *independent*, yaitu:

Jalur 1 : 1-2-3-4-5-6-9

Jalur 2 : 1-2-3-4-5-7-8-9

Jalur 3 : 1-2-4-5-6-9

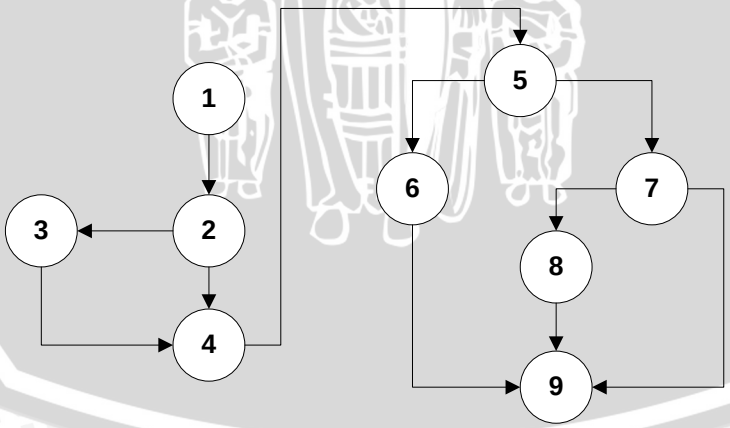
Jalur 4 : 1-2-4-5-6-7-8-9

Tabel 5. 4 Tabel Kasus Uji untuk script Horizontal.cs

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Terdapat input mouse dan nilai maju adalah benar	*Platform* akan bergerak secara horizontal menuju pada koordinat yang tersimpan pada variabel akhir	*Platform* akan bergerak secara horizontal menuju pada koordinat yang tersimpan pada variabel akhir
2	Terdapat input mouse dan nilai maju adalah salah	*Platform* akan bergerak secara horizontal menuju pada koordinat yang tersimpan pada variabel awal	*Platform* akan bergerak secara horizontal menuju pada koordinat yang tersimpan pada variabel awal
3	Tidak ada input dari pemain dan nilai maju adalah benar	*Platform* tidak bergerak sama sekali	*Platform* tidak bergerak sama sekali
4.	Tidak ada input dari pemain dan nilai maju adalah salah	*Platform* tidak bergerak sama sekali	*Platform* tidak bergerak sama sekali

Tabel 5. 5 Tabel Pseudocode Scrip Vertical.cs

Pseudo code pada script Vertical.cs	
DECLARATION	
Awal is Vector3 Akhir is Vector3 maju, aktif is Boolean speed is float endY is float	
	DESCRIPTION
1.	START
2.	Awal <- transform.position
3.	Akhir.y <- endY
4.	IF(INPUT mouse/touch)THEN
5.	aktif = true
6.	maju = !maju
7.	END IF
8.	IF(aktif && maju)THEN
9.	transform.position = Vector3.lerp(awal,
10.	akhir, speed)
11.	END IF
12.	
13.	ELSE IF aktif && !maju)THEN
14.	transform.position = Vector3.lerp(akhir,
15.	awal, speed)
	END IF
	END



Gambar 5. 3 *Flow graph* pengujian *script vertical*

Pemodelan ke dalam *flow graph* yang telah dilakukan untuk *script* vertikal menghasilkan jumlah kompleksitas siklomatis *(cyclomatic complexity)* melalui persamaan V(G) = E – N + 2, dimana V(G) merupakan jumlah kompleksitas

siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul *(node).* Pada persamaan V(G) = Region, dimana region merupakan daerah pada percabangan. Dan persamaan V(G) = Predikat node +1, dimana predikat node adalah sebuah node yang memiliki dua atau lebih jalur (percabangan).

V(G) = E – N + 2 V(G) = Jumlah Region V(G) = Predikat+1

= 11 – 9+ 2 = 4 = 4 = 4

Berdasarkan dari nilai *cyclomatic complexity* yang telah didapatkan dari perhitungan maka ditentukan tiga buah basis set dari jalur *independent*, yaitu:

Jalur 1 : 1-2-3-4-5-6-9

Jalur 2 : 1-2-3-4-5-7-8-9

Jalur 3 : 1-2-4-5-6-9

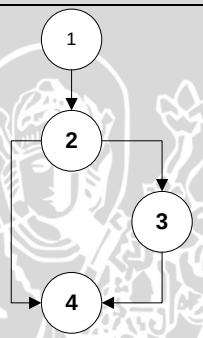
Jalur 4 : 1-2-4-5-6-7-8-9

Tabel 5. 6 Tabel Kasus Uji script Vertical.cs

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Terdapat input mouse dan nilai maju adalah benar	*Platform* akan bergerak secara horizontal menuju pada koordinat yang tersimpan pada variabel akhir	*Platform* akan bergerak secara horizontal menuju pada koordinat yang tersimpan pada variabel akhir
2	Terdapat input mouse dan nilai maju adalah salah	*Platform* akan bergerak secara horizontal menuju pada koordinat yang tersimpan pada variabel awal	*Platform* akan bergerak secara horizontal menuju pada koordinat yang tersimpan pada variabel awal
3	Tidak ada input dari pemain dan nilai maju adalah benar	*Platform* tidak bergerak sama sekali	*Platform* tidak bergerak sama sekali
4.	Tidak ada input dari pemain dan nilai maju adalah salah	*Platform* tidak bergerak sama sekali	*Platform* tidak bergerak sama sekali

Tabel 5. 7 Pseudocode script Falling.cs

Pseudo code pada script Falling	
DECLARATION plt is GameObject falling is Boolean sprite is SpriteRenderer	
	DESCRIPTION
1.	START
2.	falling = false;
3.	sprite = plt.GetComponent<SpriteRenderer>()
4.	IF(falling)THEN
5.	plt.collider2D.isTrigger <- true
6.	plt.transform.translate(0f, -8.5f, 0f)
7.	END IF
8.	END



Gambar 5. 4 *Flow graph* pengujian *script Falling*

Pemodelan ke dalam *flow graph* yang telah dilakukan untuk *script* falling menghasilkan jumlah kompleksitas siklomatis *(cyclomatic complexity)* melalui persamaan V(G) = E – N + 2, dimana V(G) merupakan jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul *(node).* Pada persamaan V(G) = Region, dimana region merupakan daerah pada percabangan. Dan persamaan V(G) = Predikat node +1, dimana predikat node adalah sebuah node yang memiliki dua atau lebih jalur (percabangan).

V(G) = E – N + 2 V(G) = Jumlah Region V(G) = Predikat+1

= 4 – 4 + 2 = 2 = 2 = 2

Berdasarkan perhitungan di atas, maka ditentukan tiga buah basis set dari jalur *independent*, yaitu:

Jalur 1 : 1-2-3-4

Jalur 2 : 1-3-4

Tabel 5. 8 Tabel Kasus Uji script Falling.cs

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Jika falling bernilai benar	*Platform* tetap berada pada posisi semula	*Platform* tetap berada pada posisi semula
2	Jika falling bernilai salah	*Platform* akan jatuh dan collider yang ada pada *platform* diubah menjadi collider isTrigger	*Platform* akan jatuh dan collider yang ada pada *platform* diubah menjadi collider isTrigger

3. Analisis Hasil Pengujian Unit

Hasil analisis yang didapatkan dari pengujian unit yaitu:

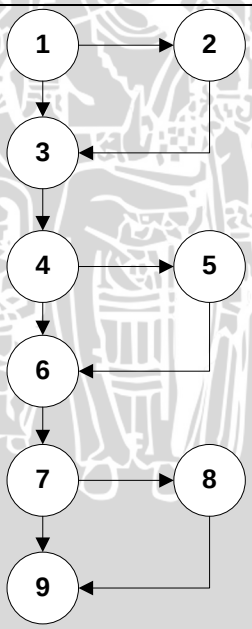
1. Berdasarkan kesesuaian antara hasil pengujian tiap unit dengan output unit yang diharapkan pada permainan edukasi pengukuran maka dapat diambil kesimpulan bahwa unit modul dari program sudah sesuai dengan output yang diharapkan.
2. Berdasarkan hasil perhitungan *cyclomatic complexity* dari tiap *flow graph* kode unit, kode unit yang menghasilkan jalur kasus uji terbanyak adalah kode operasi unit Timun Emas. Hal ini dipengaruhi oleh tiap kondisi yang dibutuhkan agar obyek Timun Emas dapat berjalan dengan lancar.

5.1.2. Pengujian Integrasi

Pengujian integrasi diterapkan pada method yang mengintegrasikan fungsionalitas dari *class-class* lain untuk melakukan sebuah operasi tertentu. Pada pengujian integrasi perangkat lunak ini digunakan teknik *White-Box Testing* dengan teknik *Basis Path Testing*. Pada teknik *Basis Path Testing*, proses pengujian dilakukan dengan memodelkan algoritma pada suatu *flow graph*, menentukan jumlah kompleksitas siklomatis *(cyclomatic complexity)*, menentukan sebuah basis set dari jalur independen dan memberikan kasus uji *(test case)* pada setiap basis set yang telah ditentukan.

Tabel 5. 9 Tabel Pseudo code ScoreResult

Pseudocode ScoreResult.cs	
DECLARATION Inisilisasi variabel	
	DESCRIPTION
1.	START
2.	IF(!PlayerData.coin)THEN
3.	PlayerData.coin = coin
4.	END IF
5.	IF(!PlayerData.Flag)THEN
6.	PlayerData.Flag = flag
7.	END IF
8.	IF(!PlayerData.Timer)THEN
9.	PlayerData.Timer = time
10.	END IF
11.	END



Gambar 5. 5 *Flow graph* pengujian integritas *class ScoreResult*

Pemodelan ke dalam *flow graph* yang telah dilakukan untuk mengubah nilai b*oolean* pada *class PlayerData* menghasilkan jumlah kompleksitas siklomatis *(cyclomatic complexity)* melalui persamaan V(G) = E – N + 2, dimana V(G) merupakan jumlah kompleksitas siklomatis, E merupakan sisi atau *edge*

(garis penghubung antar *node*) dan N merupakan jumlah simpul *(node).* Pada persamaan V(G) = Region, dimana region merupakan daerah pada percabangan. Dan persamaan V(G) = Predikat node +1, dimana predikat node adalah sebuah node yang memiliki dua atau lebih jalur (percabangan).

V(G) = E – N + 2 V(G) = Jumlah Region V(G) = Predikat+1

= 11 – 9 + 2 = 4 = 4 = 4

Berdasarkan dari nilai *cyclomatic complexity* yang telah didapatkan dari perhitungan maka ditentukan dua buah basis set dari jalur *independent*, yaitu:

Jalur 1 : 1-2-3-4-6-7-9

Jalur 2 : 1-3-4-5-6-7-9

Jalur 3 : 1-3-4-6-7-8-9

Jalur 4 : 1-2-3-4-5-6-7-8-9

Tabel 5. 10Tabel Kasus Uji script ScoreResult

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1.	Jika PlayerData.coin bernilai salah dan yang lain bernilai benar	PlayerData.coin diisi dengan nilai boolean yang disimpan pada coin	PlayerData.coin diisi dengan nilai boolean yang disimpan pada coin
2.	Jika PlayerData.Flag bernilai salah dan yang lain bernilai benar	PlayerData.Flag diisi dengan nilai boolean yang disimpan pada flag	PlayerData.Flag diisi dengan nilai boolean yang disimpan pada flag
3.	Jika PlayerData.Timer bernilai salah dan yang lain bernilai benar	PlayerData.Timer diisi dengan nilai boolean yang disimpan pada time	PlayerData.Timer diisi dengan nilai boolean yang disimpan pada time
4.	Jika ketiganya bernilai salah	PlayerData.coin, PlayerData.Flag, dan PlayerData.Timer masing-masing akan	PlayerData.coin, PlayerData.Flag, dan PlayerData.Timer masing-masing akan

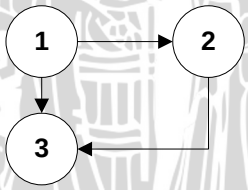
		diisi dengan nilai boolean yang disimpan pada coin, flag, dan time	diisi dengan nilai boolean yang disimpan pada coin, flag, dan time

1. Pengujian class MusicSetting

Class MusicSetting digunakan untuk merubah nilai *boolean* yang ada pada *class SoundBank* untuk menghidupkan atau mematikan suara musik pada *game.*

Tabel 5. 11 Tabel pseudocode MusicSetting

Pseudocode MusicSetting	
DECLARATION isSound is BOOL	
	DESCRIPTION
1.	START
2.	IF(INPUT mouse/touch)THEN
3.	isSound = !isSound
4.	END IF
5.	END



Gambar 5. 6 *Flow graph* pengujian intergrasi *script MusicSetting*

Pemodelan ke dalam *flow graph* yang telah dilakukan *class MusicSetting* menghasilkan jumlah kompleksitas siklomatis *(cyclomatic complexity)* melalui persamaan V(G) = E – N + 2, dimana V(G) merupakan jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul *(node).* Pada persamaan V(G) = Region, dimana region merupakan daerah pada percabangan. Dan persamaan V(G) = Predikat node +1, dimana predikat node adalah sebuah node yang memiliki dua atau lebih jalur (percabangan).

V(G) = E – N + 2 V(G) = Jumlah Region V(G) = Predikat+1

= 3 – 3 + 2 = 2 = 2 = 2

Sementara untuk V(G) = jumlah region, gambar 5.6 terdapat 2 region, diantaranya:

Berdasarkan dari nilai *cyclomatic complexity* yang telah didapatkan dari perhitungan maka ditentukan dua buah basis set dari jalur *independent*, yaitu:

Jalur 1 : 1-2-3

Jalur 2 : 1-3

Tabel 5. 12 Tabel Kasus Uji pengujian script MusicSetting

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1.	Input perintah music hidup	Volume dari music diubah menjadi satu	Volume dari music diubah menjadi satu
2.	Input perintah music mati	Volume dari music diubah menjadi nol	Volume dari music diubah menjadi nol

2. Analisa Hasil Pengujian Integrasi

Berdasarkan kesesuaian antara hasil pengujian integrasi dengan output integrasi yang diharapkan pada *game* Timun Emas, maka dapat diambil kesimpulan bahwa modul integrasi dari program sudah sesuai dengan output yang diharapkan. Pengujian Validasi.

5.2 Pengujian Pengguna

5.3.1 Test Suite

Kasus uji yang terdapat pada *test suite* merupakan beberapa pertanyaan yang memberikan instruksi yang digunakan untuk pengujian pada *game.* Dimana pengguna harus menjalankan sesuai dengan Pertanyaan tersebut untuk dapat menguji. Pernyataan yang ada pada *test suite* dapat dijawab dengan jawaban “Ya” dan “Tidak”, dimana jawaban “Ya” mengindikasikan *game* berjalan sesuai dengan instruksi dan “Tidak” sebagai jawaban bahwa *game* tidak berjalan sesuai dengan instruksi atau jika terjadi kesalahan.

Hasil dari *test suite* diatas akan menunjukkan apakah *game* tersebut berjalan dengan lancar atau tidak. Jika ditemukan jawaban "Tidak" pada salah satu kasus uji coba, maka terdapat kesalahan pada *game* tersebut dan harus diperbaiki.

5.3.2 Play testing

Play testing adalah sebuah pengujian yang dilakukan pada sebuah *game*, dengan tujuan untuk mendapatkan respon dari penguji, seberapa menarik dan menyenangkan *game* tersebut. *Playtesting* lebih menguji pada *game design.*

Pengujian ini dilakukan terhadap anak-anak SD yang mencoba *game* Timun Emas. Pengujian ini berguna untuk mengetahui apakah *game* Timun Emas menarik untuk dimainkan dan dapat digunakan untuk menceritakan cerita Timun Emas. Hasil dari pengujian ini dapat digunakan untuk memperbaiki beberapa bagian *game* seperti *gameplay* dan *level design.*

Pengujian dilakukan kepada siswa kelas 1 pada SD Negeri Gondanglegi Wetan 01 pada hari sabtu, 20 Desember 2014. Didapatkan sebanyak 17 hasil kuesioner dari siswa tersebut. Setiap siswa yang diuji, sebelumnya harus menjawab pertanyaan nomor 1 pada kuesioner dahulu, yang berisi apakah mereka mengetahui tentang cerita Rakyat Timun Emas atau tidak. Jika terdapat siswa yang menjawab tidak tahu maka akan dimainkan terlebih dahulu *scene prologue* yang ada pada *game.* Jika terdapat yang menjawab mengetahui cerita tersebut bisa langsung memainkan permainan. Setelah selesai memainkan *game,* selanjutnya para siswa yang diuji bisa meneruskan menjawab pertanyaan kuesioner selanjutnya.

5.3.3 Analisis pengujian

Berdasarkan dari pengujian pada *test suite* dan *play testing,* didapat hasil sebagai berikut:

1. Terdapat beberapa *bug* yang dialami oleh pengguna pada saat pengujian *test suite,* namun tidak semua pengguna yang mengalami *bug* tersebut. Dari dua puluh lima penguji pada *test suite* ini, terdapat satu atau dua penguji saja yang mengalami *bug* pada beberapa *test case.* Sebagian besar *bug* yang ditemukan saat pengujian ini ada bagian tutorial. *Bug* terjadi

ketika *tutorial scene* dijalankan secara terus menerus. *Bug* telah diperbaiki setelah pengujian ini selesai dilaksanakan.

2. Pada saat pengujian *play testing*, dari tujuh belas anak yang berpartisipasi tujuh diantaranya tidak mengetahui tentang cerita Timun Emas. Beberapa yang mengetahui cerita tersebut hanya mengetahui cerita Timun Emas secara singkat saja.
3. Sebanyak 88% menjawab *game* Timun Emas sangat menyenangkan. Sisanya terdapat 6% yang menjawab cukup menyenangkan dan 6% lainnya tidak menjawab. Dengan jawaban ini dapat diambil kesimpulan bahwa *game* Timun Emas cukup menarik bagi anak-anak untuk dimainkan.
4. Pada tingkat kesulitan, didapat 35.3% menjawab mudah, 17.6% yang menjawab cukup, dan sisanya sebesar 47.1% menjawab susah. Dari anak-anak yang menjawab mudah dan cukup, diketahui anak-anak tersebut sudah terbiasa dengan *smartphone* sehingga mudah untuk beradaptasi tingkat kesulitan dari *game* Timun Emas. Dengan hasil ini, dapat dikatakan bahwa *game* Timun Emas cukup sulit untuk dimainkan oleh anak-anak yang berumur tujuh tahun.

BAB VI
KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil pengamatan dari perancangan, implementasi dan proses pengujian permainan yang dilakukan, maka diambil kesimpulan sebagai berikut:

1. Menurut Scott Rogers, dalam merancang sebuah *game* terdapat beberapa bagian penting pada setiap game, diantaranya gameplay, rule, mekanis dari sebuah *game,* konsep art hingga HUD yang merupakan bagian dasar dari sebuah *game.* Dengan menggunakan format perancangan yang telah disusun oleh Scott Rogers dapat mempermudah pendeskripsian setiap bagian dari *game* yang akan dirancang.
2. Penggunaan metode *rapid and iterative prototyping* dalam perancangan *game design* adalah mempermudah dalam proses perbaikan pada bagian desain maupun implementasi. Konsep dasar dari *rapid and iterative prototyping* adalah semakin banyak perulangan dilakukan, ide atau *gameplay* dari *game* yang dibuat menjadi lebih matang, dikarenakan pada setiap akhir perulangan terdapat fase evaluasi dimana, *game* yang telah di desain akan ditentukan apakah *game* tersebut sesuai dengan konsep dasar yang telah ditentukan atau perlu dilakukan penambahan atau pengurangan fitur.
3. Dalam mengimplementasikan *game,* terdapat beberapa point penting yang harus diperhatikan sesuai dengan perancangan *gameplay* yang telah dilaksanakan sebelumnya. Point tersebut antara lain implementasi *gameplay* yang diantaranya mengimplementasikan sistem, peraturan, *goal*, *rule* hingga *scoring.* Point lain yang berpengaruh dalam implementasi game yaitu realisasi level, dan implementasi art mulai dari karakter, latar belakang, HUD hingga menu.
4. Penggunaan *game engine Unity* dapat mempermudah dalam proses implementasi, khususnya pada proses implementasi *scipt. Unity* telah menyediakan sebuah *library* yang telah mencakup banyak bagian umum ada *game,* seperti *collider, rigid body,* dan *camera.*
5. Dari hasil pengujian yang dilakukan pada siswa kelas 1 SD didapat kesimpulan bahwa tingkat kesulitan dari *game* Timun Emas disimpulkan

terlalu susah untuk tingkatan anak berusia tujuh tahun. Dari hasil pengujian didapat bahwa terdapat 47.05% dari tujuh belas responden yang merasa *game* Timun Emas susah. Sedangkan 17.65% dari tujuh belas responden menjawab cukup susah dan sisanya sebanyak 35.3% menjawab mudah.

6.2 Saran

1. Dari hasil pengujian *play testing* pada siswa kelas 1 SD, gameplay dari pengembangan game Timun emas dapat diajukan suatu cara untuk membuat gameplay permainan menjadi lebih mudah dengan cara menurunkan kecepatan dari setiap karakter atau menambah waktu yang disediakan untuk menyelesaikan setiap level, sehingga pada saat diujikan pada target pemain, penguji tidak merasa terlalu kesulitan dalam memainkan *game* tersebut.
2. Penambahan fitur, sehingga dapat membuat *game* yang dibuat lebih menarik dan menyenangkan.
3. *Game* Timun Emas dapat diadaptasikan menjadi sebuah *game multiplatform* yang dapat dimainkan bukan hanya pada ponsel *android,* tetapi juga pada ponsel lain seperti *windows phone, iOS,* hingga pada *desktop PC.*

DAFTAR PUSTAKA

[ALL-12] Schultz Charles P., Bryant Robert, Langdell Tim, 2005, *Game Testing All in One*, Course Technology: Boston.

[AUD-NN] Audacity: About Audacity. *http://audacity.sourceforge.net/about/* [29 Desember 2014].

[BER-12] Survey: Game di Ponsel Jadi Alat Membunuh Waktu18 Mei 2012; h*ttp://www.beritasatu.com/iptek/48836-survei-game-di-ponsel-jadi-alat-membunuh-waktu.html* [9 Juni 2014].

[DET-13] Heriyanto, Trisno Indonesia Masuk 5 Besar Negara Pengguna Smartphone. 3 Februari 2014; *http://inet.detik.com/read/2014/02/03/171002/2485920/317/indonesia-masuk-5-besar-negara-pengguna-smartphone* [6 Juni 2014].

[GMD-09] Second Lesson, Game Design: Iterative Rapid Prototyping. 7 Februari 2009; https://gamedesignconcepts.wordpress.com/2009/07/02/level-2-game-design-iteration-and-rapid-prototyping/ [21 Maret 2015].

[GDC-13] Game Design Concepts! Quarter 1 2013; https://learn.canvas.net/courses/3 [15 Maret 2015]

[GUN-08] Gunnerson, Tate. Playing Video Games May Boost Older Brains. 11 Desember 2008; *http://http://www.washingtonpost.com/wp-dyn/content/article/2008/12/11/AR2008121102640.html* [9 Juni 2014].

[LVL-10] Roger, Scott. 2010, *Level UP! The Guide to Great Video Game design*, Wiley: West Sussex

[MUL-13] Apa itu multimedia, Sastra multimedia *http://www.satriamultimedia.com/artikel_apa_itu_multimedia.html/* [22 Desember 2014].

[PAT- NN] Gregory, Luke, Schligloff, H. and Roggenbach, M. Path Testing

[RED-08] Red Stone Software. 2008, Black-box vs. White-box Testing: Choosing the Right Approach to Deliver Quality Applications.

[SD-14] Jadwal Pelajaran Temastik Kelas 1 Kurikulum 2014. *http://www.sekolahdasar.net/2014/07/rpp-kurikulum-2013-kelas-1-2-4-dan-5-sd.html* [22 Desember 2014].

[SAI-NN] PaintTool SAI. *; http://www.systemax.jp/en/sai/ [*30 Desember 2014].

[TIM-NN] Rahimsyah, MB, Timun Emas Melawan Raksasa, CV. Pustaka Agung Harapan.

[TOM-12] Suara Merdeka. Game Membuat Anak Lebih Kreatif. 23 September 2012; *http://www.gametechnology.info/2012/10/game-membuat-anak-lebih-kreatif.html* [19 November 2014].

[UNI-NN] Learn About Unity; *http://unity3d.com/learn* [30 Desember 2014].

[VIV-14] Sofia, Maya, Tasya Paramitha Senin. Cara Seru Tarik Minat Anak Terhadap Cerita Rakyat. 8 September 2014; *http://life.viva.co.id/news/read/536242-cara-seru-tarik-minat-anak-terhadap-cerita-rakyat*. [19 November 2014].