BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Pada bagian analisis sistem ini akan dibahas mengenai analisa-analisa masalah yang akan dilakukan terhadap pasar game khususnya game di Indonesia, analisa pengguna (user) dan juga kegunaan game yang akan dibangun.

3.1.1 Analisis Masalah

1. Pengaruh Perkembangan Teknologi Terhadap Budaya Bangsa

Kebudayaan adalah kompleks keseluruhan dari pengetahuan, keyakinan, kesenian, moral, hukum, adat-istiadat dan semua kemampuan atau kebiasaan lain yang diperoleh seseorang sebagai anggota masyarakat (Sir Edward Tylor).

Namun seiring berjalannya waktu dan teknologi yang semakin canggih, rasa tanggung jawab sudah pudar terhadap budaya. Masyarakat tidak lagi peduli dengan budayanya. Hal ini disebab kan semakin gencarnya media elektronik, khususnya TV yang selalu menayangkan kebudayaan luar. Hal ini dengan mudahnya merusak pola pikir masyarakat khususnya para generasi muda. Mereka cenderung melupakan kebudayaan sendiri dan beralih ke budaya luar.

2. Peran Masyarakat & Pemerintah dalam melestarikan budaya Indonesia

Kesenian & kebudayaan merupakan dua sisi mata uang yang tidak terpisahkan. Kesenian dapat menjadi wajah untuk mempertahankan identitas budaya Indonesia. Faktanya, sekarang ini identitas budaya Indonesia sudah mulai memudar karena arus global. Sehingga kondisi yang mengkhawatirkan ini perlu segera diselamatkan. Hal ini semakin diperparah dengan diakuinya budaya Indonesia oleh bangsa lain. Masalah yang sedang marak baru-baru ini adalah diakuinya lagu daerah yang bersal dari Maluku,"Rasa Sayang-sayange," serta "Reog Ponorogo" dari Jawa Timur oleh Malaysia. Hal ini disebabkan oleh kurang pedulinya bangsa Indonesia terhadap budayanya. Namun ketika kebudayaan itu di akui oleh bangsa lain, Indonesia mulai bingung. Berita terbaru menyebutkan bahwa kesenian "Angklung" dari Jabar juga mau dipatenkan oleh Negara tersebut. Lalu dimanakah peran masyarakat dan pemerintah dalam mengatasi hal ini?.

Kebudayaan nasional adalah kebudayaan kita bersama yakni kebudayaan yang mempunyai makna bagi kita bangsa Indonesia. Kalau bukan kita lalu siapa lagi yang akan menjaga dan melestarikannya.Hal ini sebenarnya akan menimbulkan rasa tanggung jawab untuk melestarikan kebudayaan tersebut.

Begitu juga halnya dengan pemerintah, pemerintah harus tegas dalam menjaga dan melestarikan kebudayaan Indonesia dengan cara membuat peraturan perundang-undangan yang bertujuan untuk melindungi budaya bangsa. Jika perlu pemerintah harus mematenkan budaya yang ada di Indonesia agar tidak jatuh ke tangan Negara lain

3.2 Analisis dan Kebetuhan Fungsional

3.2.1 Identifikasi Aktor

Tahap pertama yang dilakukan dalam melakukan analisis berorientasi objek menggunakan UML adalah menentukan aktor atau pengguna sistem. Kata aktor dalam konteks UML, menampilkan peran (roles) yang pengguna (atau sesuatu di luar sistem yang dikembangkan yang dapat berupa perangkat keras, end user, sistem yang lain, dan sebagainya).

3.3 Analisis dan Kebutuhan Non-Fungsional

3.3.1 Analisis dan Kebutuhan Perangkat Lunak

Perangkat lunak digunakan dalam sebuah sistem merupakan perintah-perintah yang diberikan kepada perangkat keras agar bisa saling berinteraksi diantara keduanya. Perangkat lunak yang dibutuhkan untuk membangun aplikasi game Kabayan Saving Our Cultures ini adalah sebagai berikut :

1. Sistem Operasi Windows 7.
2. Greenfoot-windows-201
3. jdk-1_5_0_07-windows-i586-p ke atas (JDK 1.5 minimal).

3.3.2 Analisis dan Kebutuhan Perangkat Keras

Komputer terdiri dari perangkat keras dan perangkat lunak yang saling berinteraksi. Perangkat lunak memberikan instruksi-instruksi kepada perangkat

keras untuk melakukan suatu tugas tertentu, sehingga dapat menjalankan suatu sistem di dalamnya.

Pada aplikasi androIslam ini, perangkat keras yang digunakan adalah sebagai berikut :

1. AMD Athlon(tm) 64 X2 Dual Core Processor 5000+ 2.60GHz
2. RAM : 3GB
3. VGA 512 MB.
4. Harddisk 750 GB.
5. Monitor.
6. Mouse dan Keyboard.

3.3.3 Pembangunan Aplikasi Game 2D Kabayan Saving Our Cultures.

3.3.3.1 Pengenalan

Game yang akan dibuat berjudul Kabayan Saving Our Cultures yang bergenre Adventure. Game ini dibangun dengan mengaplikasikan teknologi sebagai jembatan untuk menyampaikan kebudayaan Bangsa Indonesia, khususnya budaya Jawa Barat. Game ini dibangun dengan grafis 2D (2 Dimensi).

Game ini menerapkan sistem single player dengan data-data yang dibagi ke dalam objek-objek karena menggunakan pemrograman berorientasi objek.

Berikut ini adalah kelebihan-kelebihan yang ada pada game ini :

1. Sistem single player.

2. Mengangkat tema budaya, khususnya budaya Jawa Barat, umumnya budaya Indonesia.
3. Grafik game 2D yang lebih berwarna.
4. Adventure (petualangan) yang disediakan interaktif.
5. Aplikasi game bersifat bermain sambil belajar.

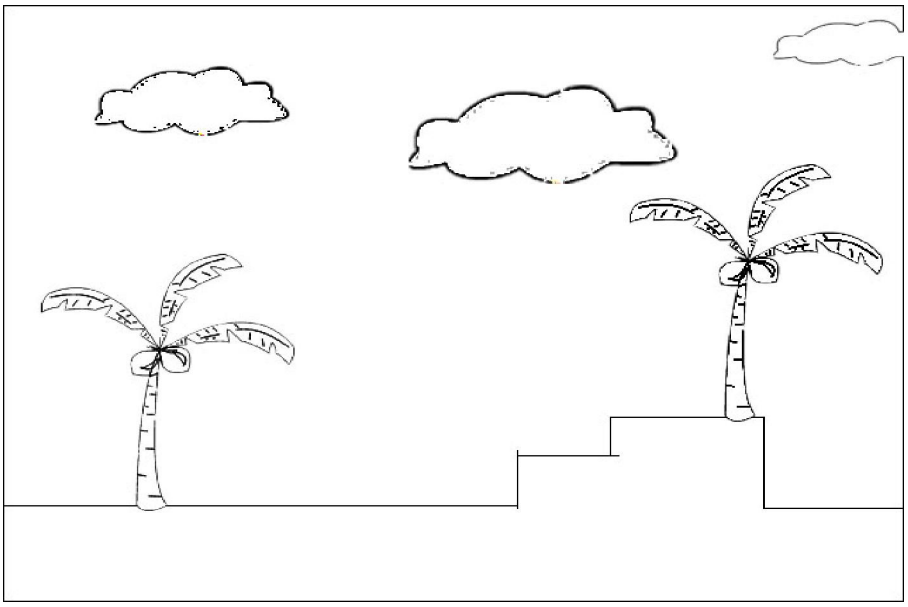
3.3.3.2 Storyline

Kabayan Saving Our Cultures menceritakan perjalanan seorang pemuda dari suku Sunda yaitu Kabayan yang harus bertualang untuk menyelamatkan alat musik tradisional yaitu angklung yang telah dicuri oleh seorang penjahat dari Negara Misterius yang bernama Juragan Asep. Disini Kabayan harus mengalami berbagai macam petualangan dalam usahanya untuk mengejar Juragan Asep yang kabur ke negeri seberang dan berniat untuk membawa Angklung ke Negaranya. Kabayan harus berjuang mati-matian untuk merebut kembali kesenian asli Jawa Barat tersebut.

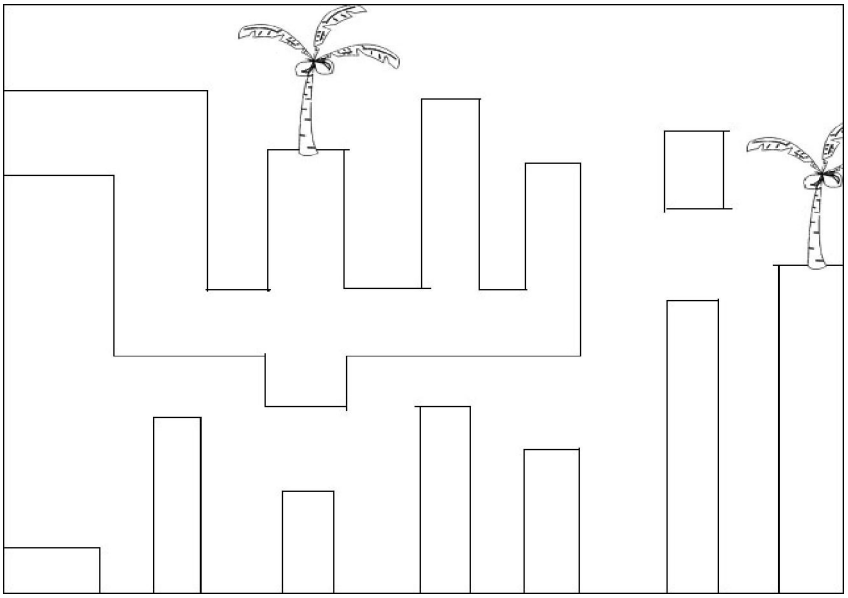
3.3.3.3 Storyboard

Storyboard adalah sketsa gambar yang disusun berurutan sesuai dengan naskah, dengan storyboard kita dapat menyampaikan ide cerita kita kepada orang lain dengan lebih mudah, karena kita dapat menggiring khayalan seseorang mengikuti gambar-gambar yang tersaji, sehingga menghasilkan persepsi yang sama pada ide cerita kita.

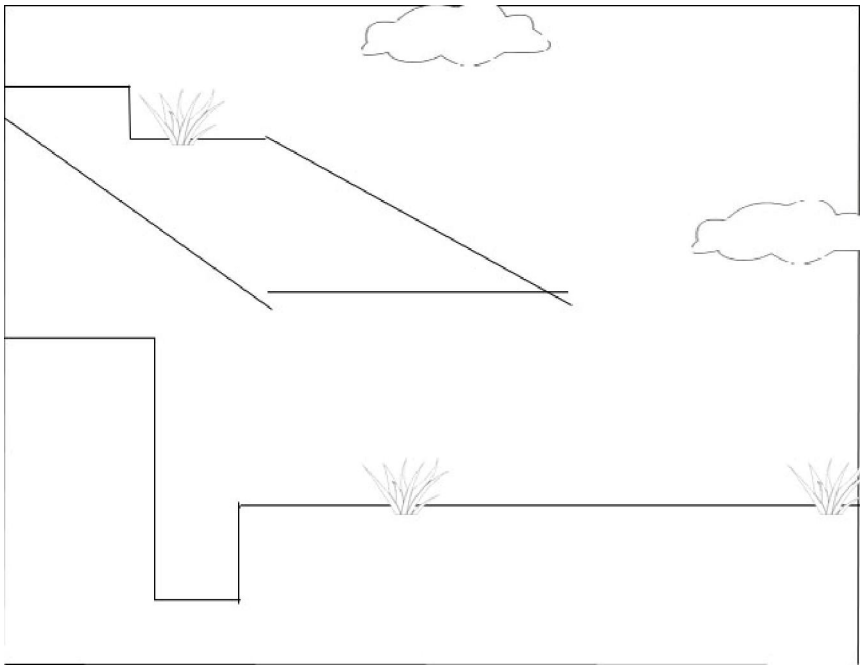
Berikut merupakan storyboard dari Aplikasi game 2D Kabayan Saving Our Cultures :



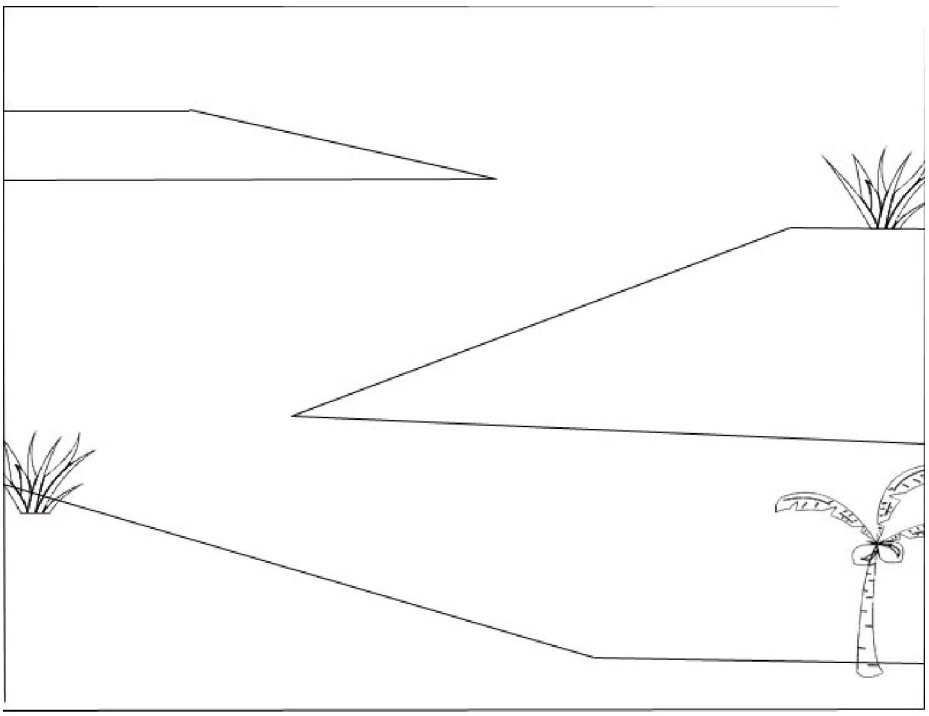
Gambar 3.1 Storyboard Level 1-1



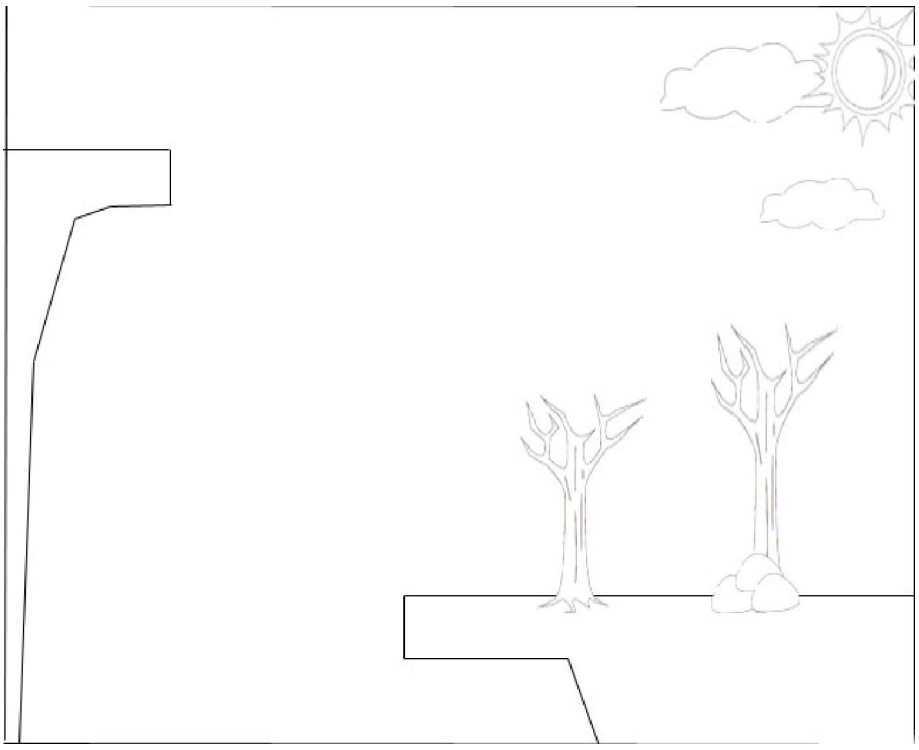
Gambar 3.2 Storyboard Level 1-2



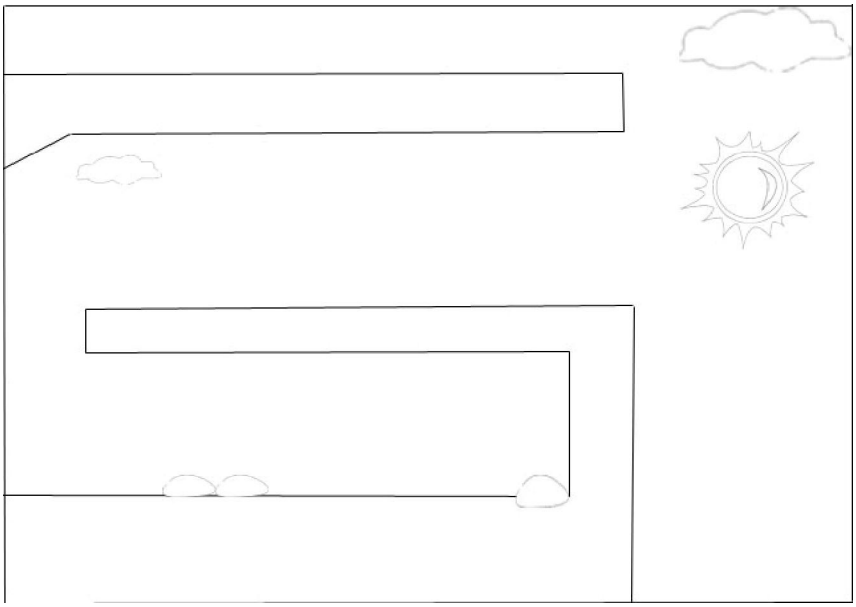
Gambar 3.3 Storyboard Level 1-3



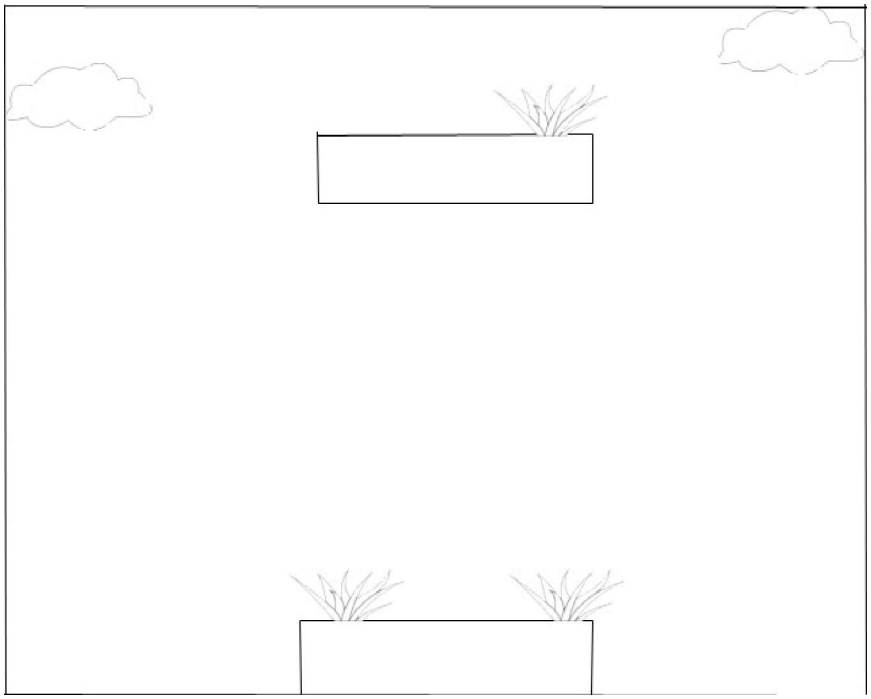
Gambar 3.4 Storyboard Level 1-4



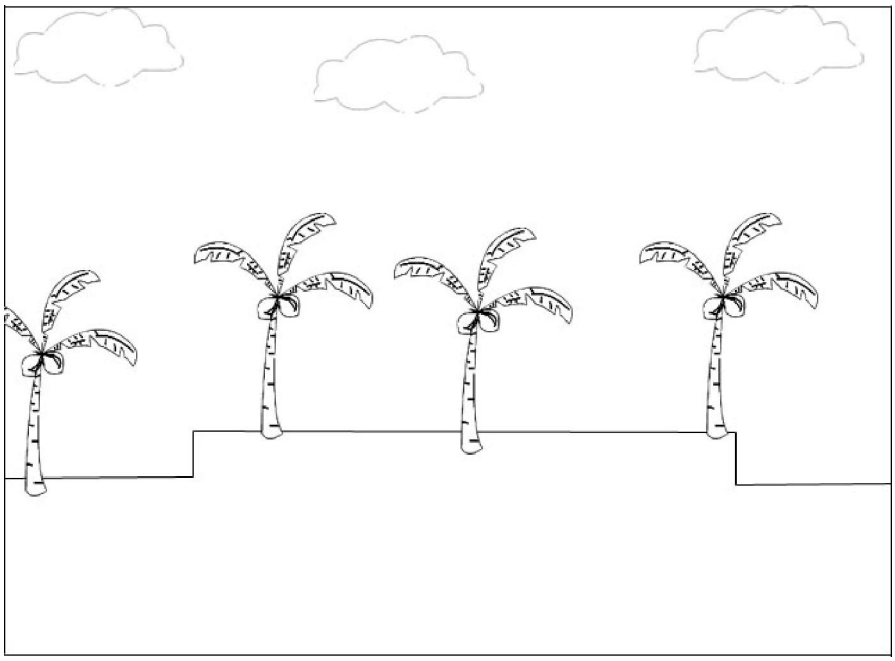
Gambar 3.5 Storyboard Level 1-5



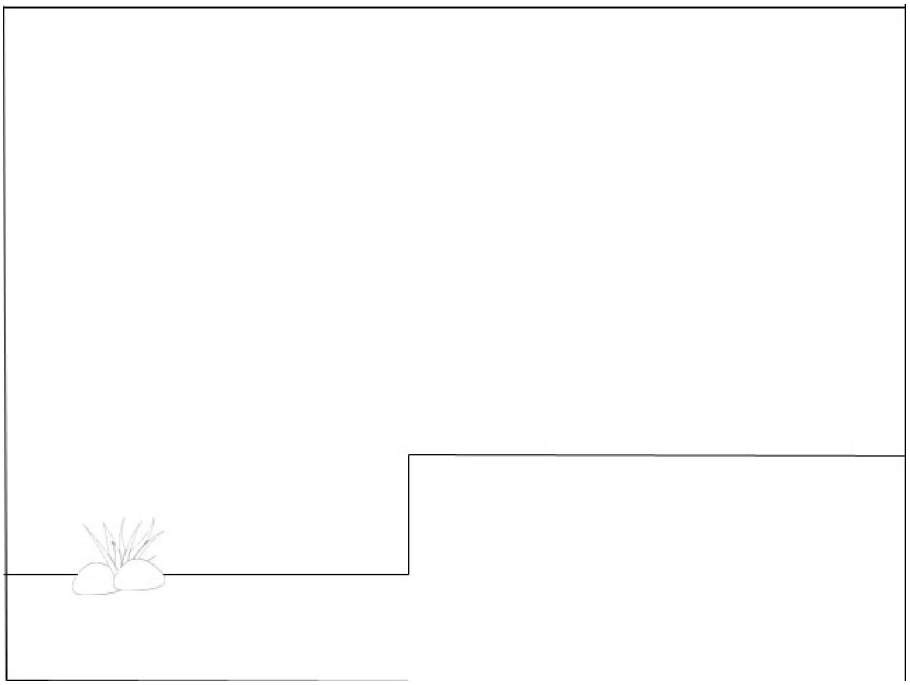
Gambar 3.6 Storyboard Level 1-6



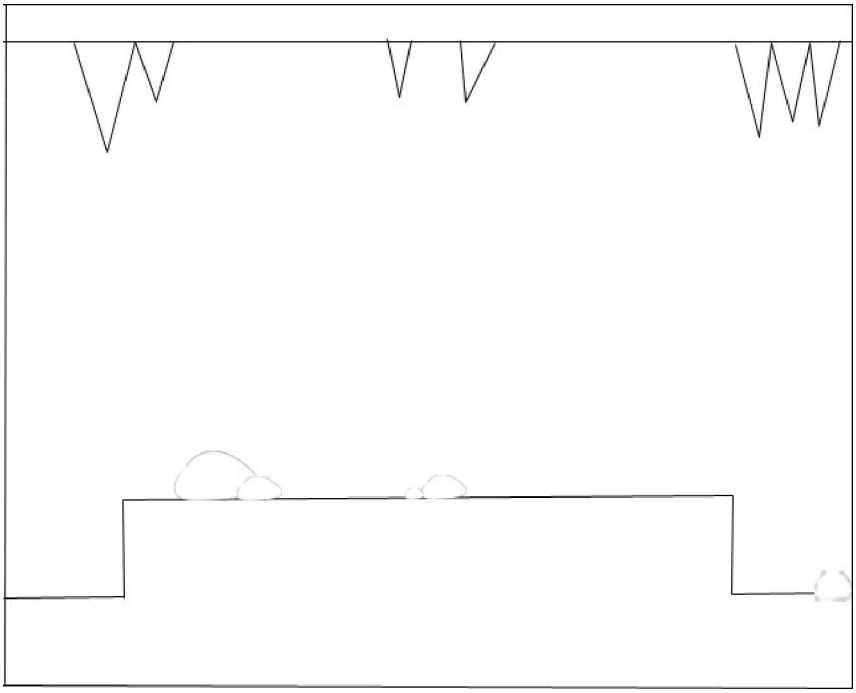
Gambar 3.7 Storyboard Level 1-7



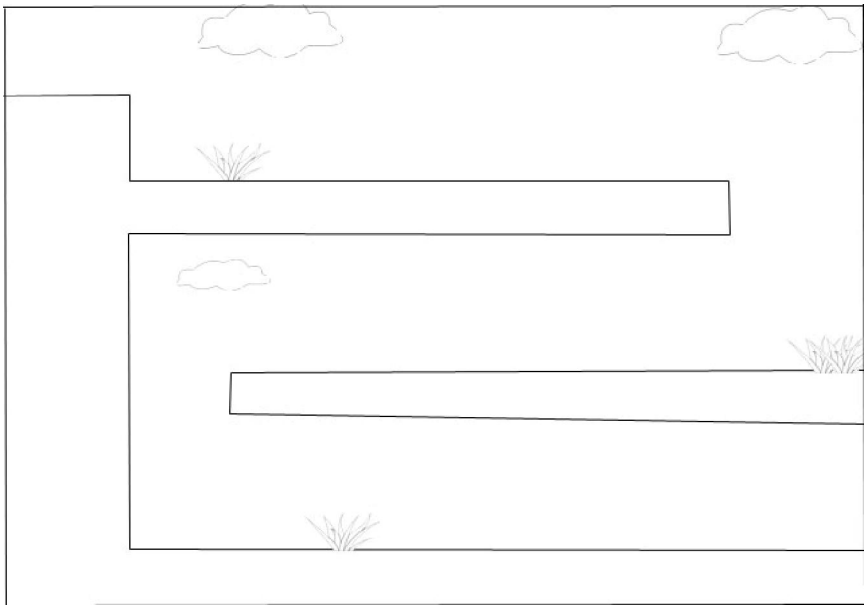
Gambar 3.8 Storyboard Level 1-8



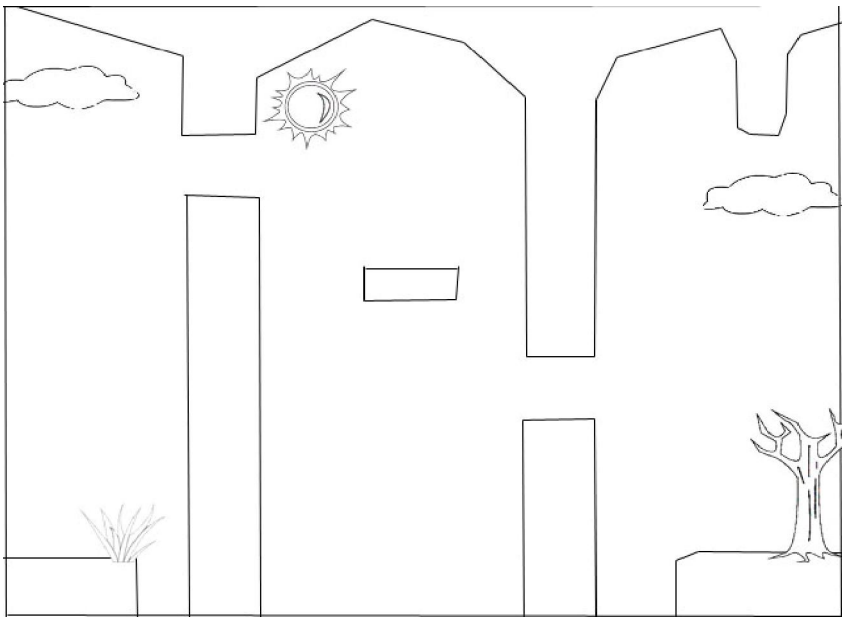
Gambar 3.9 Storyboard Level 1-9



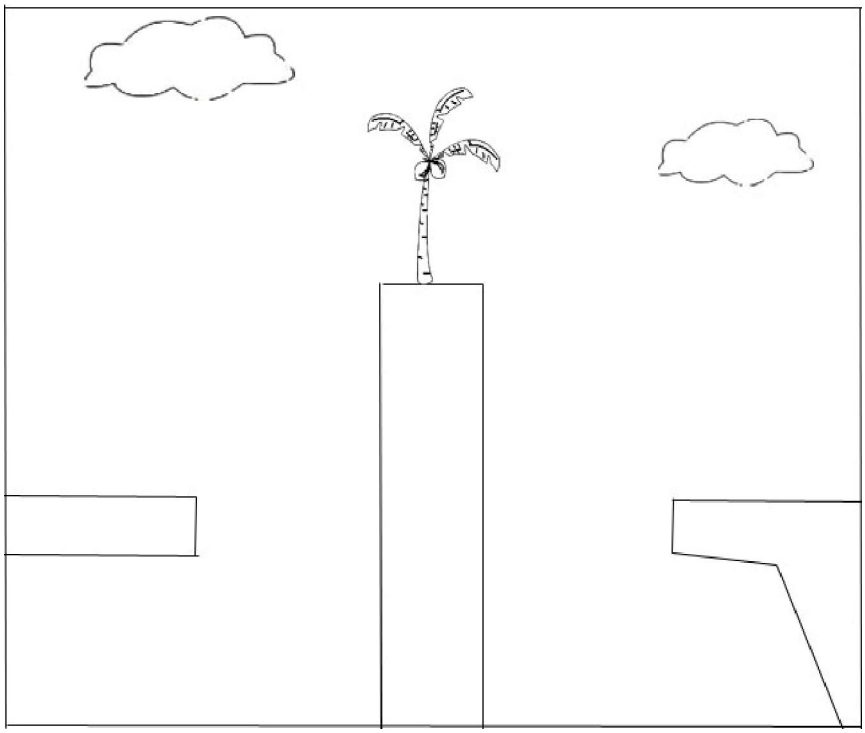
Gambar 3.10 Storyboard Level 1-Boss



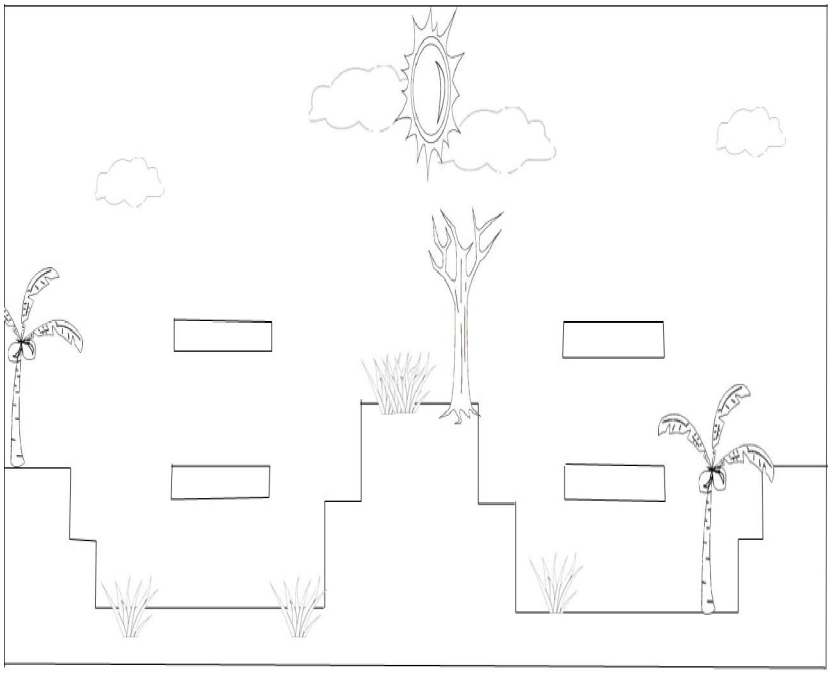
Gambar 3.11 Storyboard Level 2-1



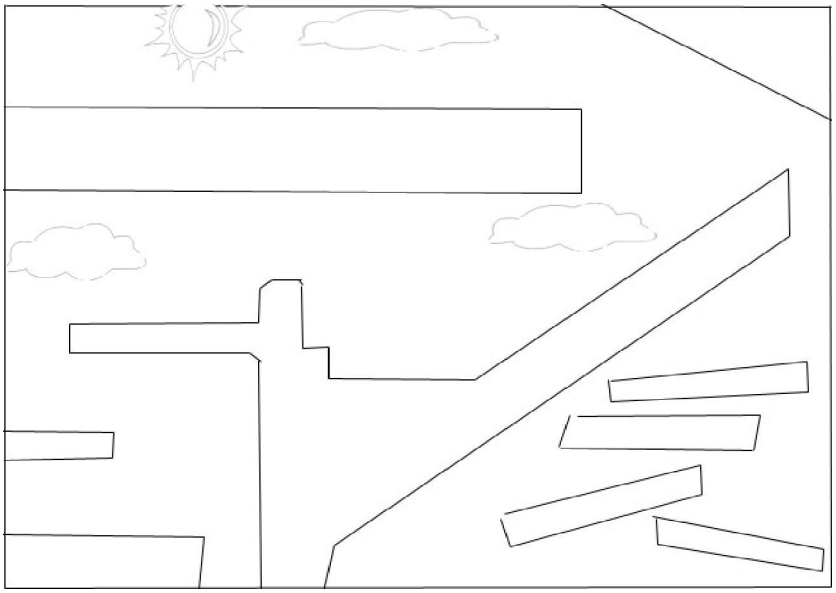
Gambar 3.12 Storyboard Level 2-2



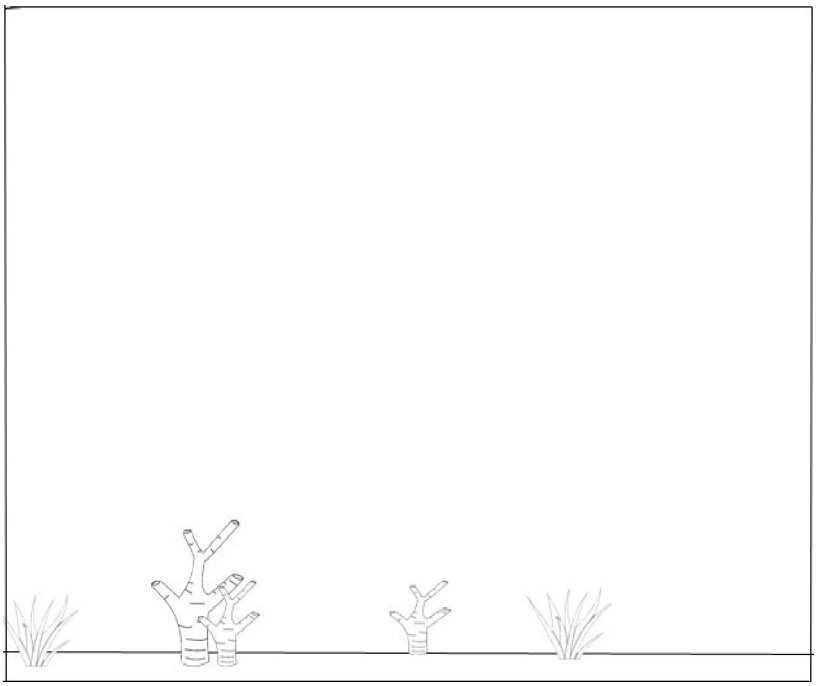
Gambar 3.13 Storyboard Level 2-3



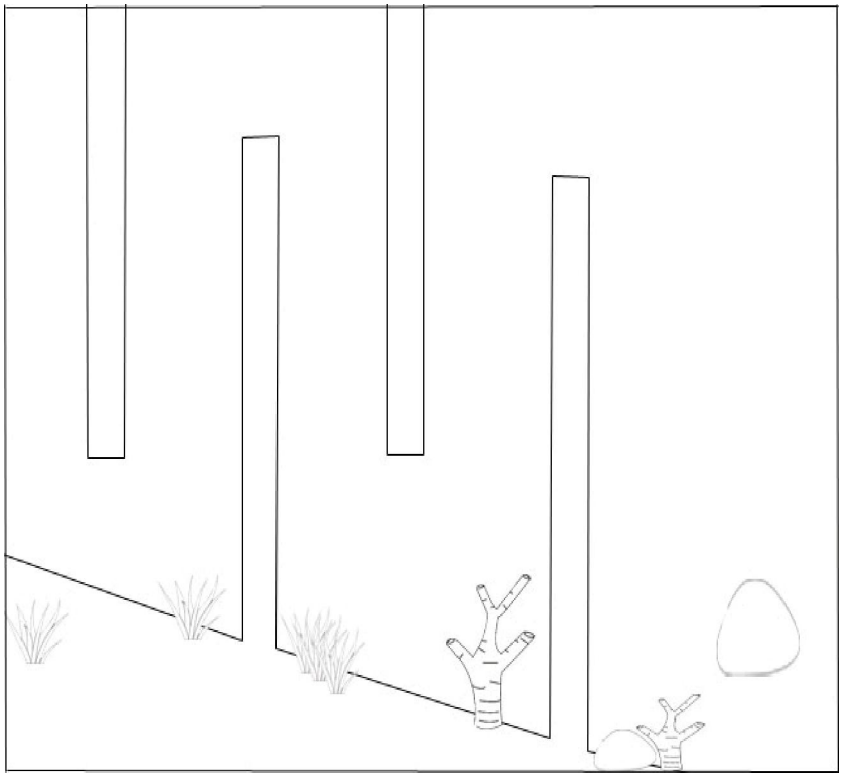
Gambar 3.14 Storyboard Level 2-4



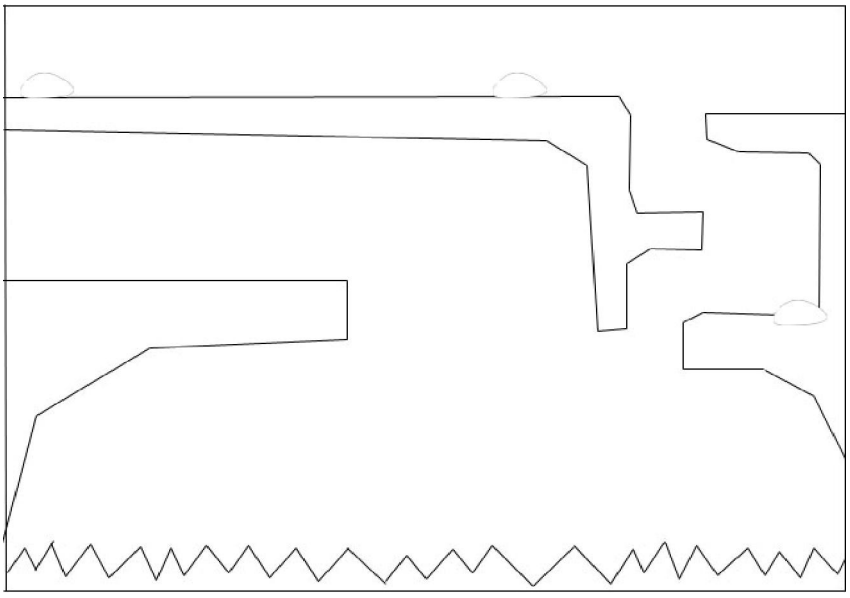
Gambar 3.15 Storyboard Level 2-5



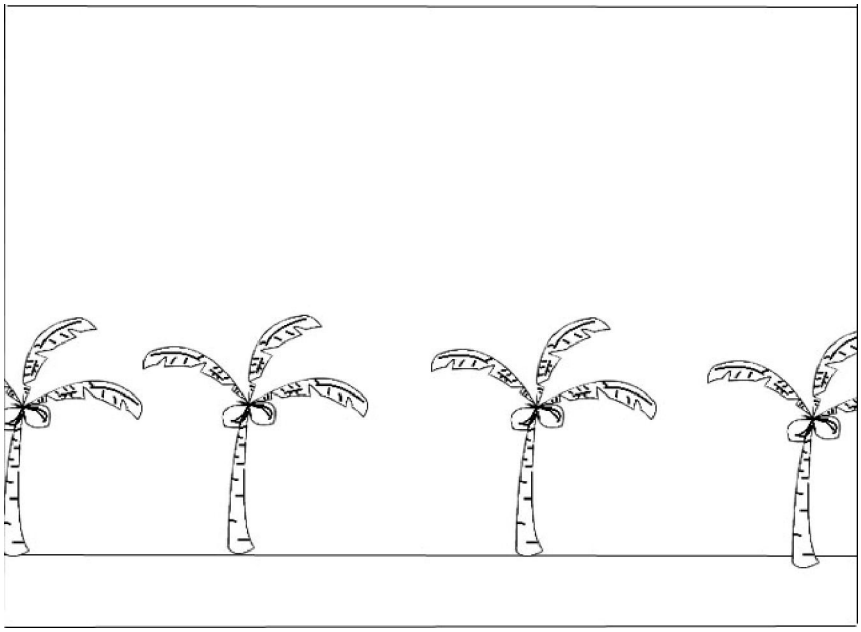
Gambar 3.16 Storyboard Level 2-6



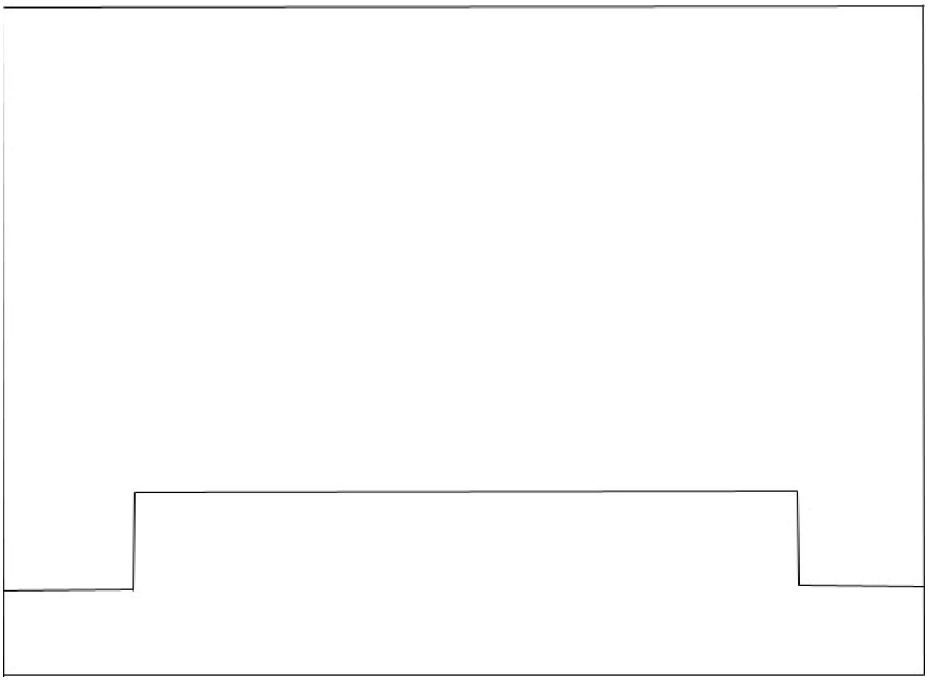
Gambar 3.17 Storyboard Level 2-7



Gambar 3.18 Storyboard Level 2-8



Gambar 3.19 Storyboard Level 2-9



Gambar 3.20 Storyboard Level 2-Boss

3.3.3.4 Gameplay

Tugas utama pemain (gamer) dalam memainkan Aplikasi ini yaitu melalui level-level dan juga sub level, dimana setiap bagian level menyediakan petualangan dengan menampilkan beberapa karakter level tersebut maupun karakter musuh yang harus dilawan, sehingga pemain memperoleh score (nilai).

Pemain (gamer) dapat memulai permainan dengan memilih pilihan mulai baru, dimana nantinya pemain memasuki sub pilihan yaitu memilih level dengan karakter permainan “Mudah” atau karakter permainan “Susah”. Setelah memilih salah satu pilihan, pemain kembali diantarkan pada sub pilihan, yaitu pilihan untuk memilih level, dengan tujuan apakah pemain ingin bermain dari level awal atau pada hal ini level 1, sedangkan pilihan lainnya adalah pemain langsung memulai untuk bermain pada level berikutnya atau level 2.

Setelah pemain telah memilih salah satu level, maka pemain langsung dapat memulai petualangannya. Selain pilihan “Mulai baru” pada tampilan awal juga disediakan pilihan “High score” yaitu pilihan untuk melihat score-score yang sudah tercetak sebagai hasil dari permainan yang telah dilakukan oleh pemain dan pilihan lainnya adalah pilihan “Pembuat”, sebagai keterangan yang menyediakan informasi mengenai pembuat game ini.

3.3.4 Sasaran User dan Kegunaan Aplikasi

Game yang akan dibuat ditujukan untuk user dengan usia dari kelompok anak-anak sampai remaja dengan range usia 5-15 tahun, alasannya adalah berusaha menyisipkan pesan budaya dengan menerapkan konsep belajar sambil bermain, sehingga tanpa disadari gamer telah belajar mengenai budaya Indonesia, tetapi tidak dengan cara yang membosankan, usia ini juga merupakan usia yang efektif untuk mempelajari berbagai hal dan merupakan kurun waktu yang sangat penting dan kritis dalam hal tumbuh kembang fisik, mental, dan psikososial, yang berjalan sedemikian cepatnya. Pada masa usia ini juga sering disebut dengan masa usia emas (the golden age).

Game yang dibuat merupakan game PC (Personal Computer) atau bisa disebut juga game desktop.

3.3.5 Analisa Penggunaan Algoritma BFS pada game Kabayan SOC

Breadth-First Search (BFS). Algoritma BFS akan melakukan penelusuran pada semua cabang yang dibuka dari simpul induknya. Pencarian dilakukan dalam arah horisontal sehingga semua cabang yang dibuka dijamin akan mendapatkan giliran yang adil. Bila pada kedalaman (depth) yang sama GS belum ditemukan, maka dilakukan pembukaan cabang baru lagi sesuai dengan urutan pembukaan cabang sebelumnya. Secara sederhana pseudo code algoritma BFS adalah sebagai berikut:

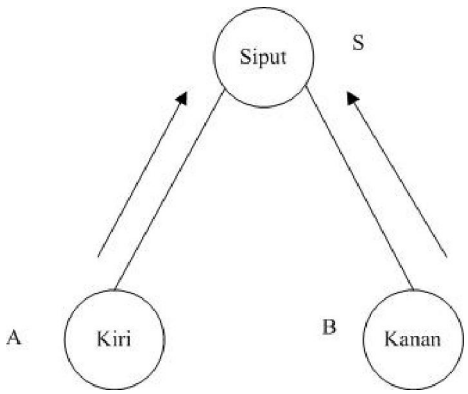
1. Berikan simpul awal pada daftar “open”.
2. Loop : if open = kosong then exit(fail).
3. n := first(open)

4. if goal(n) then exit(success)
5. Remove(n, open) dan Add(n, closed)
6. Ekspansikan n, berikan semua simpul anak pada ekor “open” dan bubuhkan pointer dari simpul anak ke-n.
7. Kembali ke Loop.

Penggunaan Algoritma BFS pada game Kabayan SOC digunakan pada beberapa Class yaitu :

1. Struktur Data tree pada Class Siput

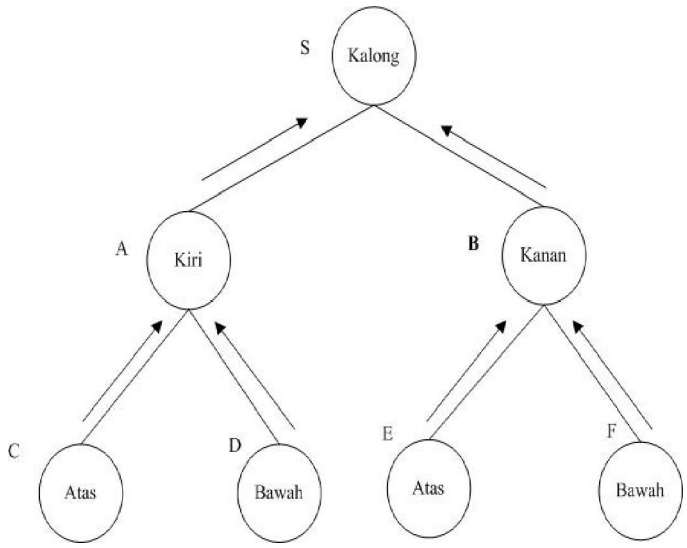
Tabel 3.1 Struktur Data Tree pada Class Siput

Nama Class	Siput
Tree Diagram	 Keterangan : S= Siput A = Kiri B = Kanan
Urutan Searching	[S] [AB]

	Urutan Pelacakan : [S, A, B]
Algoritma	IF siput gerak ke kiri AND Kabayan NOT ketemu THEN gerak ke kanan END IF

2. Struktur Data tree pada Class Kalong

Tabel 3.2 Tabel 3.2 Struktur Data Tree pada Class Kalong

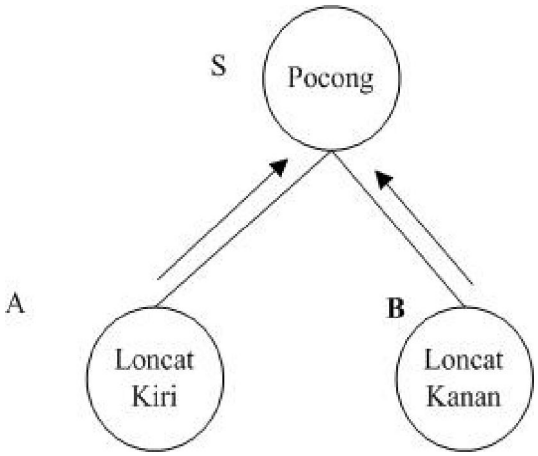
Nama Class	Kalong
Tree Diagram	 Keterangan : S= Kalong A = Kiri B = Kanan

	C dan E = Atas D dan F = Bawah
Urutan Searching	[S] [AB] [BCD] [CDEF] [DEF] Urutan Pelacakan : [S, A, B, C, D, E, F]
Algoritma	IF Kalong gerak ke kiri AND Kabayan NOT ketemu THEN gerak ke kanan ELSE IF Kabayan NOT ketemu THEN gerak ke kiri atas OR gerak ke kiri bawah ELSE IF Kabayan NOT ketemu THEN gerak ke kanan atas OR Gerak ke kanan bawah ELSE IF Kabayan ketemu THEN keluarkan listrik END IF END IF END IF

	END IF

3. Struktur Data tree pada Class Pocong

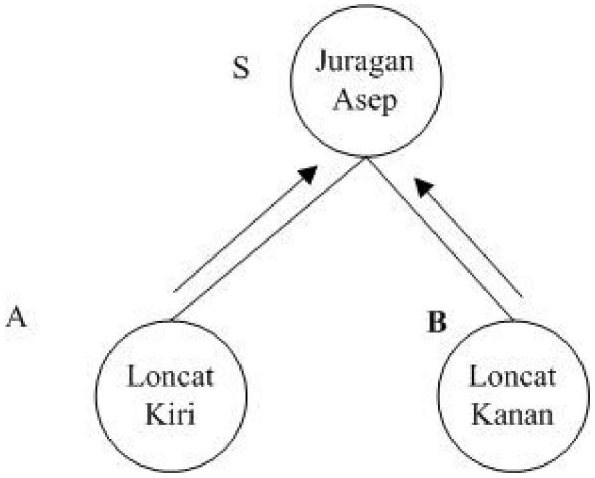
Tabel 3.3 Tabel 3.3 Struktur Data Tree pada Class Pocong

Nama Class	Pocong
Tree Diagram	 Keterangan : S= Pocong A = Loncat Kiri B = Loncat Kanan
Urutan Searching	[S] [AB] Urutan Pelacakan : [S, A, B]
Algoritma	IF pocong loncat ke kiri AND Kabayan NOT ketemu

	THEN loncat ke kanan END IF

4. Struktur Data tree pada Class Juragan Asep

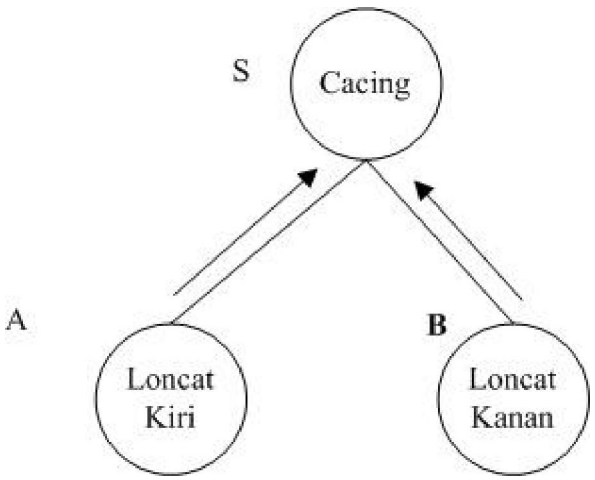
Tabel 3.4 Struktur Data Tree pada Class Juragan Asep

Nama Class	Juragan Asep
Tree Diagram	 Keterangan : S = Juragan Asep A = Loncat Kiri B = Loncat Kanan
Urutan Searching	[S] [AB] Urutan Pelacakan : [S, A, B]
Algoritma	IF JuraganAsep loncat ke kiri

	AND Kabayan NOT ketemu THEN loncat ke kanan END IF

5. Struktur Data tree pada Class Cacing

Tabel 3.5 Struktur Data Tree pada Class Cacing

Nama Class	Cacing
Tree Diagram	 Keterangan : S = Juragan Asep A = Loncat Kiri B = Loncat Kanan
Urutan Searching	[S] [AB] Urutan Pelacakan :

	[S, A, B]
Algoritma	IF cacing gerak ke kiri AND Kabayan NOT ketemu THEN gerak ke kanan END IF

3.3.6 Analisa Penggunaan Algoritma DFS pada game Kabayan SOC

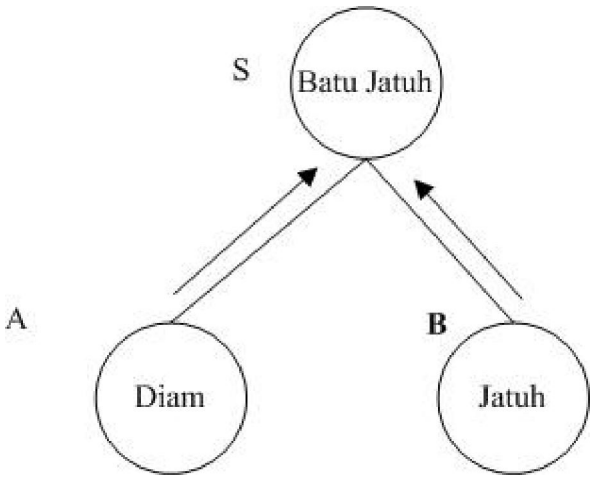
Depth-First Search (DFS). Algoritma DFS akan melakukan penelusuran dengan cara mengekplorasi simpul anak yang pertama kali ia buka. Bila GS tidak ditemukan pada simpul yang dibuka, ia akan bergerak mundur untuk membuka simpul anak berikutnya. Hal ini akan dilakukan secara terus menerus hingga GS ditemukan. Secara sederhana pseudo code algoritma DFS adalah sebagai berikut :

1. Berikan simpul awal pada daftar “open”.
2. Loop : if open = kosong then exit(fail).
3. n := first(open)
4. if goal(n) then exit(success)
5. Remove(n, open) dan Add(n, closed)
6. Ekspansikan n. Berikan pada kepala “open” semua simpul anak yang belum muncul pada “open” atau “closed” dan bubuhkan pointer pada n.
7. Ekspansikan n, berikan semua simpul anak pada kepala “open” dan bubuhkan pointer dari simpul anak ke-n.

8. Kembali ke Loop.

1. Struktur Data tree pada BatuJatuh

Tabel 3.6 Struktur Data Tree pada Class Batu Jatuh

Nama Class	BatuJatuh
Tree Diagram	 Keterangan : S= BatuJatuh A = Diam B = Jatuh
Urutan Searching	[S] [AB] Urutan Pelacakan : [S, A, B]
Algoritma	IF Kabayan ketemu THEN BatuJatuh Jatuh ELSE Kabayan NOT ketemu THEN BatuJatuh Diam

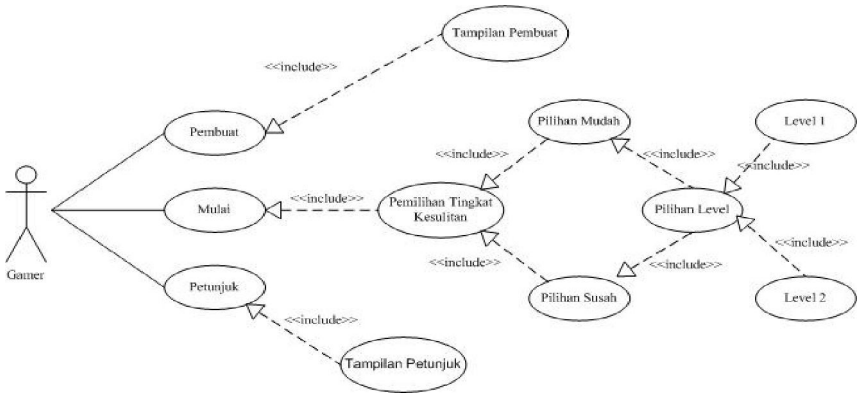
	END IF END IF

3.4 Perancangan

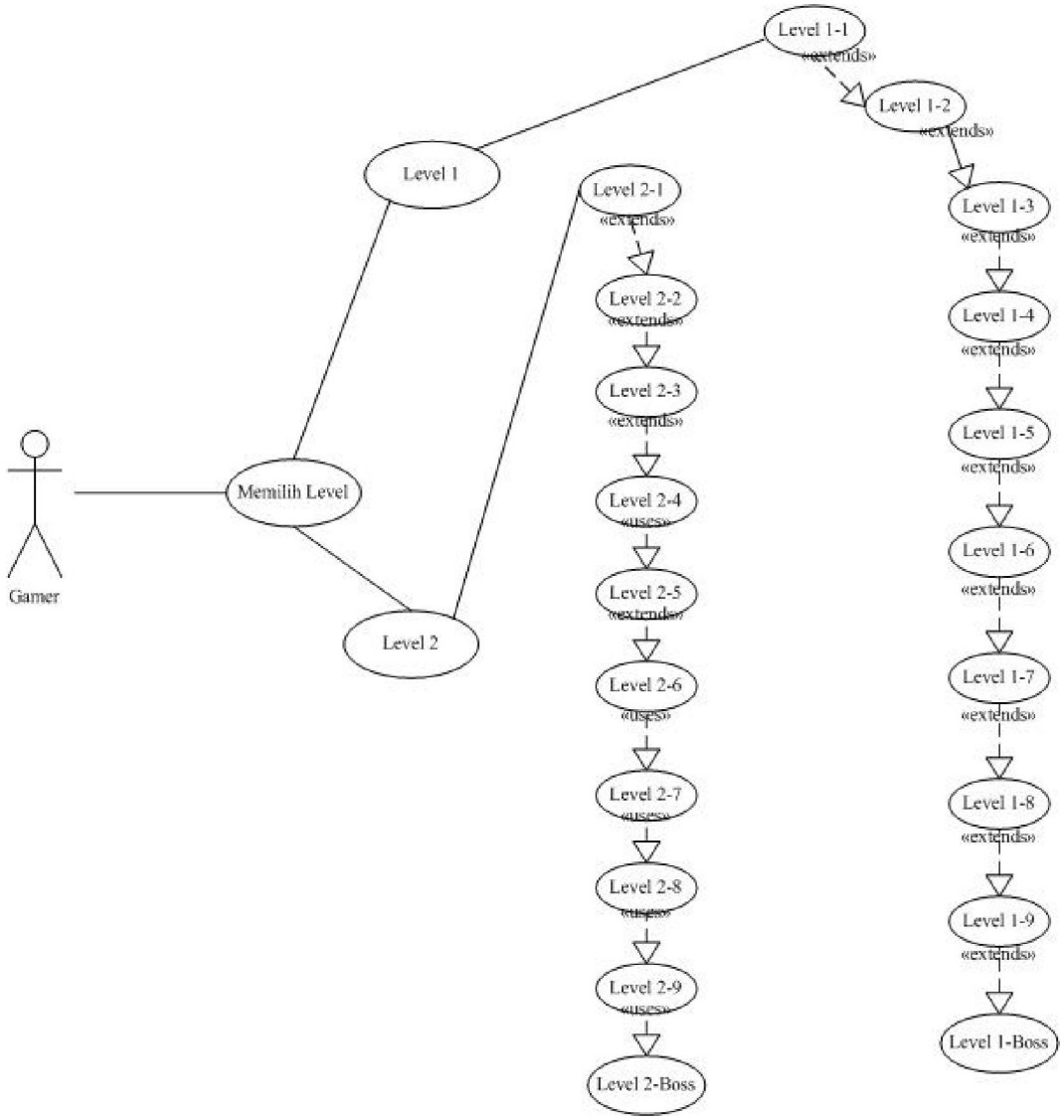
Perancangan ini bertujuan untuk menerapkan solusi pemecahan masalah yang telah diajukan pada analisis sistem

3.4.1 Pendefinisian Fungsi Game

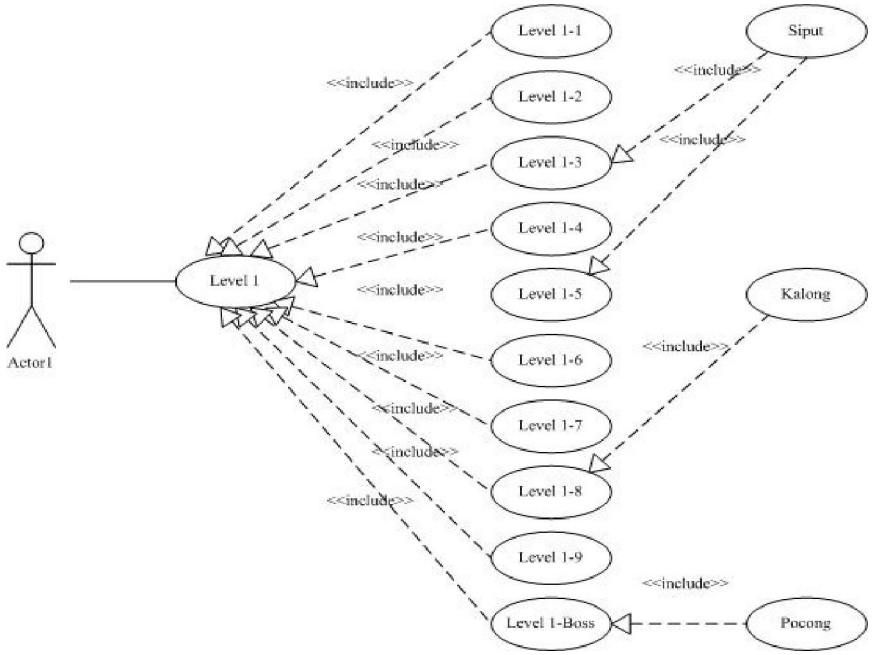
Berikut ini adalah perancangan proses-proses yang terdapat pada game Kabayan Saving Our Cultures :



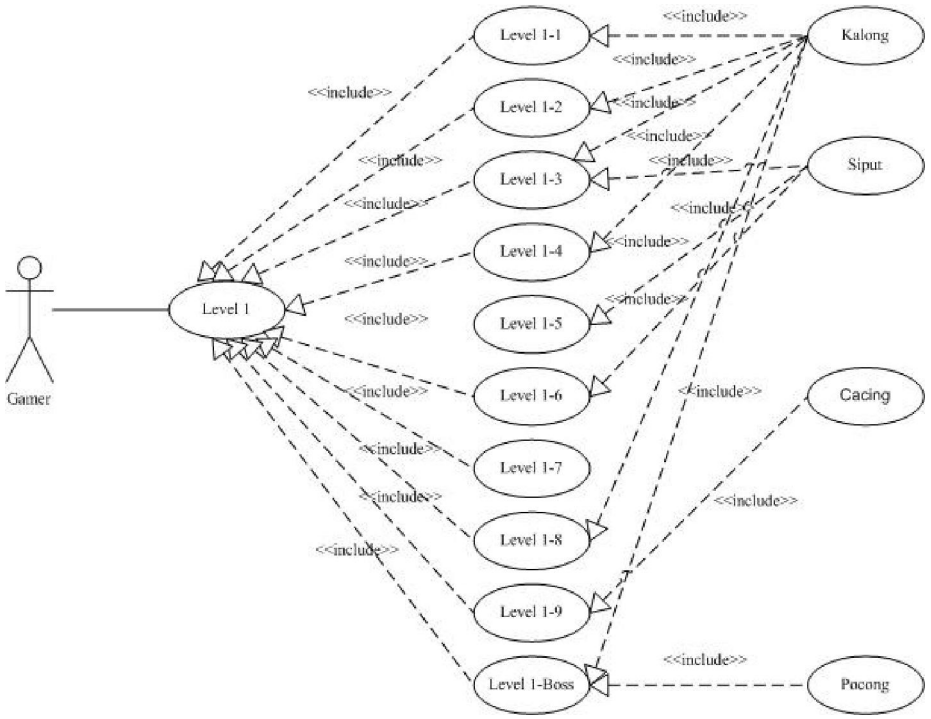
Gambar 3.21 Use Case Diagram Tampilan Utama (User)



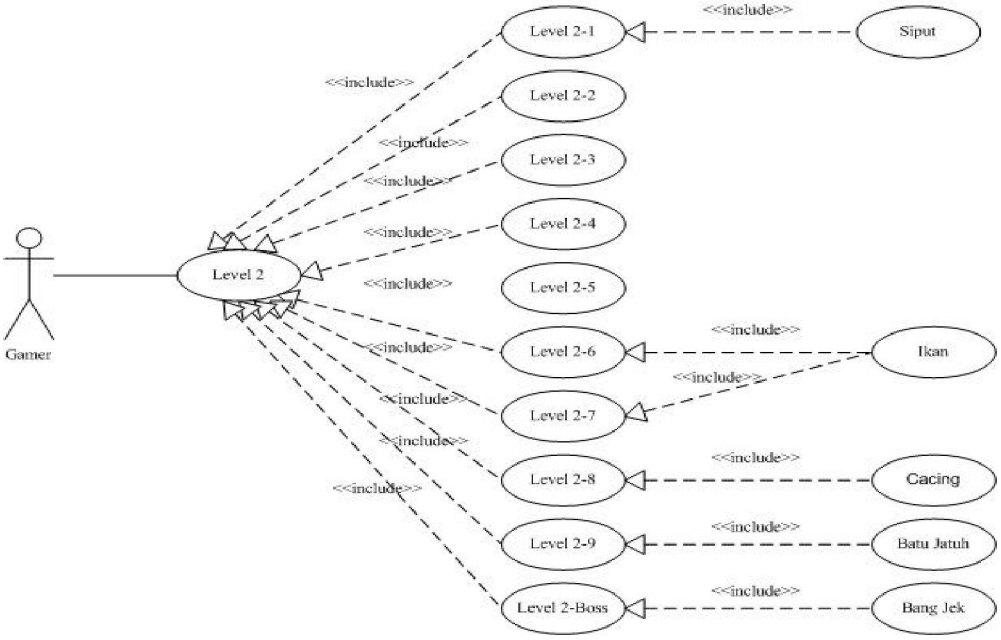
Gambar 3.22 Use Case Diagram Pemilihan Level



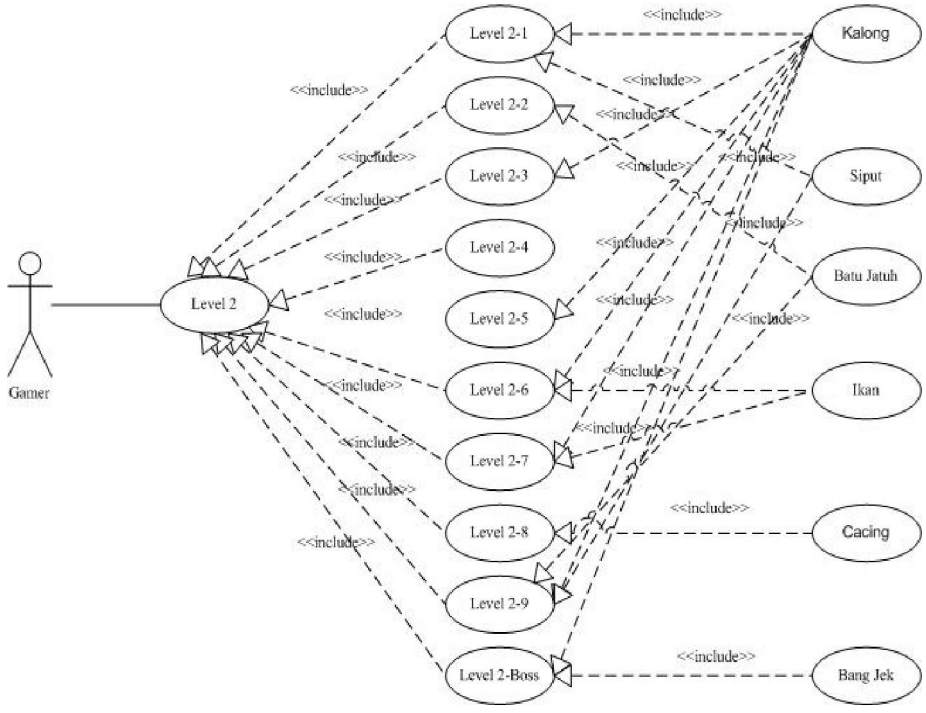
Gambar 3.23 Use Case Musuh Level 1 (Mudah)



Gambar 3.24 Use Case Musuh Level 1 (Susah)



Gambar 3.25 Use Case Musuh Level 2 (Mudah)



Gambar 3.26 Use Case Musuh Level 2 (Susah)

3.4.2 Defenisi Use Case

Definisi Use Case berfungsi untuk menjelaskan proses yang terdapat pada setiap Use Case.

Tabel 3.7 Definisi Use Case

NO	Use Case	Deskripsi
1	Mulai Baru	Proses untuk memulai permainan aplikasi
3	Pembuat	Proses untuk melihat informasi pembuat aplikasi
4	Memilih Level	Proses untuk memilih Level permainan
5	Memilih Tingkat Kesulitan	Proses untuk memilih tingkat kesulitan
6	Level 1	Tingkatan Level 1
7	Level 2	Tingkatan Level 2
8	Cerita	Proses dimana GPS mengambil nilai lintang, bujur
9	Petunjuk	Proses untuk melihat pentunjuk cara bermain aplikasi game tersebut
10	Pilihan Mudah	Pilihan level dengan tingkat kesulitan Mudah
11	Pilihan Susah	Pilihan level dengan tingkat kesulitan

		Susah

3.5 Skenario Use Case

Skenario setiap bagian pada use case menunjukkan proses apa yang terjadi pada setiap bagian didalam use case tersebut, dimana user memberikan perintah pada setiap bagian dan respon apa yang diberikan oleh sistem kepada user setelah user memberikan perintah pada setiap bagian – bagian use case.

Tabel 3.8 Scenario Use Case Pilihan Mulai Baru

Identifikasi		
Nomor	1	
Nama	Mulai Baru	
Tujuan	Memulai permainan aplikasi	
Deskripsi	Proses untuk memulai permainan aplikasi	
Aktor	Gamer	
Skenario Utama		
Kondisi awal	Pengguna berada di menu utama aplikasi game Kabayan Saving Our Cultures	
Aksi Aktor		Reaksi Sistem

Memilih menu Mulai Baru	Aplikasi merespon dengan cara masuk ke sistem dari aplikasi permainan
Kondisi akhir	Aplikasi masuk ke dalam menu pilihan tingkat kesulitan permainan

Tabel 3.9 Scenario Use Case Pilihan Petunjuk

Identifikasi	
Nomor	2
Nama	Petunjuk
Tujuan	Melihat informasi mengenai gameplay dan tips bermain
Deskripsi	Proses untuk melihat informasi mengenai gameplay dan tips bermain
Aktor	Gamer
Skenario Utama	
Kondisi awal	Pengguna berada di menu utama aplikasi game Kabayan Saving Our Cultures
Aksi Aktor	Reaksi Sistem

Memilih menu Pembuat	Aplikasi merespon dengan cara memanggil dan melihat kelas Petunjuk
Kondisi akhir	Aplikasi masuk ke dalam informasi petunjuk aplikasi game

Tabel 3.10 Scenario Use Case Pilihan Pembuat

Identifikasi	
Nomor	3
Nama	Pembuat
Tujuan	Melihat informasi mengenai pembuat aplikasi
Deskripsi	Proses untuk melihat informasi mengenai pembuat aplikasi
Aktor	Gamer

Skenario Utama		
Kondisi awal	Pengguna berada di menu utama aplikasi game Kabayan Saving Our Cultures	
Aksi Aktor		Reaksi Sistem
Memilih menu Pembuat		Aplikasi merespon dengan cara memanggil dan melihat kelas pembuat
Kondisi akhir	Aplikasi masuk ke dalam informasi pembuat aplikasi game	

Tabel 3.11 Scenario Use Case Pilihan Level

Identifikasi	
Nomor	4
Nama	Memilih Level
Tujuan	Memilih level permainan yang ingin dimainkan
Deskripsi	Proses untuk memilih level permainan yang ingin dimainkan
Aktor	Gamer

Skenario Utama		
Kondisi awal	Pengguna berada di menu utama aplikasi game Kabayan Saving Our Cultures	
Aksi Aktor		Reaksi Sistem
Memilih menu Pembuat		Aplikasi merespon dengan cara menyediakan dua pilihan level, yaitu level 1 dan level 2
Kondisi akhir	Aplikasi masuk ke dalam level 1 atau level 2	

Tabel 3.12 Scenario Use Case Pilihan Tingkat Kesulitan

Identifikasi	
Nomor	5
Nama	Memilih Tingkat Kesulitan
Tujuan	Memilih Tingkat Kesulitan yang akan dihadapi oleh gamer
Deskripsi	Proses untuk memilih tingkat kesulitan yang akan dihadapi oleh gamer
Aktor	Gamer

Skenario Utama		
Kondisi awal	Pengguna berada di menu utama aplikasi game Kabayan Saving Our Cultures	
Aksi Aktor		Reaksi Sistem
Memilih menu Pembuat		Aplikasi merespon dengan cara menyediakan dua pilihan tingkat kesulitan, yaitu pilihan mudah dan pilihan susah.
Kondisi akhir	Aplikasi masuk ke dalam pilihan level Mudah atau pilihan level Susah	

Tabel 3.13 Scenario Use Case Pilihan Level 1

Identifikasi	
Nomor	6
Nama	Level 1
Tujuan	Memilih pilihan level 1
Deskripsi	Proses untuk memilih pilihan level 1
Aktor	Gamer

Skenario Utama		
Kondisi awal	Pengguna berada di menu utama aplikasi game Kabayan Saving Our Cultures	
Aksi Aktor		Reaksi Sistem
Memilih menu Pembuat		Aplikasi merespon dengan memanggil dan masuk ke kelas level 1
Kondisi akhir	Aplikasi masuk ke dalam pilihan level 1	

Tabel 3.14 Scenario Use Case Pilihan Level 2

Identifikasi	
Nomor	7
Nama	Level 2
Tujuan	Memilih pilihan level 2
Deskripsi	Proses untuk memilih pilihan level 2
Aktor	Gamer
Skenario Utama	
Kondisi awal	Pengguna berada di menu utama aplikasi game Kabayan

	Saving Our Cultures
Aksi Aktor	Reaksi Sistem
Memilih menu Pembuat	Aplikasi merespon dengan memanggil dan masuk ke kelas level 2
Kondisi akhir	Aplikasi masuk ke dalam pilihan level 2

Tabel 3.15 Scenario Use Case Pilihan Petunjuk

Identifikasi	
Nomor	9
Nama	Petunjuk
Tujuan	Mengetahui cara bermain game yang akan dimainkan
Deskripsi	Proses untuk mengetahui cara bermain game yang akan dimainkan
Aktor	Gamer

Skenario Utama		
Kondisi awal	Pengguna berada di menu utama aplikasi game Kabayan Saving Our Cultures	
Aksi Aktor		Reaksi Sistem
Memilih menu Pembuat		Aplikasi merespon dengan memanggil dan masuk ke kelas pemilihan tingkat kesulitan dengan petunjuk permainan yang tertera pada background
Kondisi akhir	Aplikasi masuk ke dalam pilihan tingkat kesulitan dengan background petunjuk permainan	

Tabel 3.16 Scenario Use Case Pilihan Mudah

Identifikasi	
Nomor	10
Nama	Pilihan Mudah
Tujuan	Memilih tingkat permainan pada level yang mudah dimainkan
Deskripsi	Proses untuk memilih tingkat permainan pada level yang mudah dimainkan

Aktor	Gamer
Skenario Utama	
Kondisi awal	Pengguna berada di menu utama aplikasi game Kabayan Saving Our Cultures
Aksi Aktor	Reaksi Sistem
Memilih menu Pembuat	Aplikasi merespon dengan memanggil dan masuk ke kelas pemilihan tingkat kesulitan dengan pilihan kelas Mudah
Kondisi akhir	Aplikasi masuk ke dalam pilihan tingkat kesulitan Mudah

Tabel 3.17 Scenario Use Case Pilihan Susah

Identifikasi	
Nomor	11
Nama	Pilihan Susah
Tujuan	Memilih tingkat permainan pada level yang susah dimainkan
Deskripsi	Proses untuk memilih tingkat permainan pada level yang susah dimainkan
Aktor	Gamer

Skenario Utama		
Kondisi awal	Pengguna berada di menu utama aplikasi game Kabayan Saving Our Cultures	
Aksi Aktor		Reaksi Sistem
Memilih menu Pembuat		Aplikasi merespon dengan memanggil dan masuk ke kelas pemilihan tingkat kesulitan dengan pilihan kelas Susah
Kondisi akhir	Aplikasi masuk ke dalam pilihan tingkat kesulitan Susah	

3.6 Class Diagram



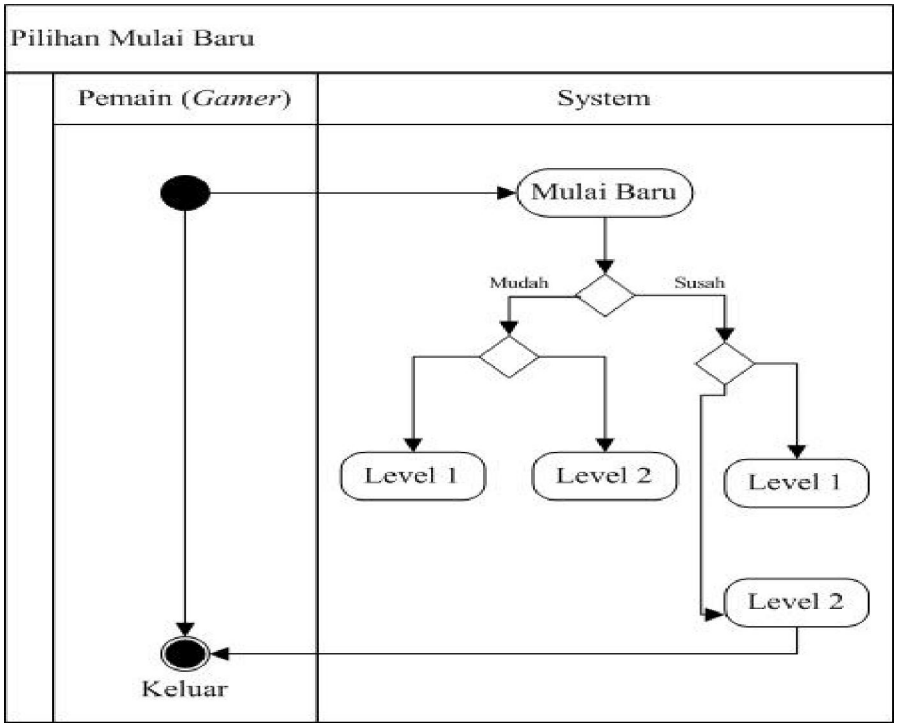
Gambar 3.27 Class Diagram Game Kabayan Saving Our Cultures

3.7 Activity Diagram

3.7.1 Proses View Mulai Baru

Tabel 3.18 Penjelasan View Mulai Baru

Deskripsi	Tampilan ini menampilkan pilihan Mulai Baru, dimana pemain diberikan sub pilihan untuk memilih Tingkat Kesulitan dan juga Level yang ingin dimainkan.
Normal Flow	1. Pemain memilih pilihan Mulai Baru. 2. Pemain memilih tingkat kesulitan yaitu antara pilihan mudah dan pilihan susah. 3. Pemain memilih level yang ingin dimainkan yaitu antara level 1 atau level 2.

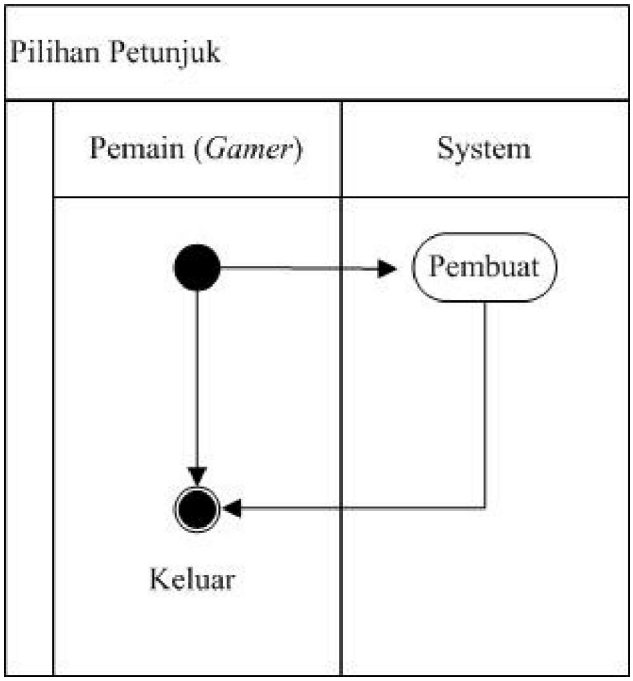


Gambar 3.28 Activity Diagram Pada Proses View Mulai Baru

3.7.2 Proses View Petunjuk

Tabel 3.19 Penjelasan View Petunjuk

Deskripsi	Tampilan ini merupakan tampilan untuk menampilkan informasi petunjuk mengenai gameplay dan juga tips mengalahkan musuh
Normal Flow	1. Pemain memilih pilihan Petunjuk melihat tips dan petunjuk permainan. 2. Pemain keluar dari tampilan Petunjuk.

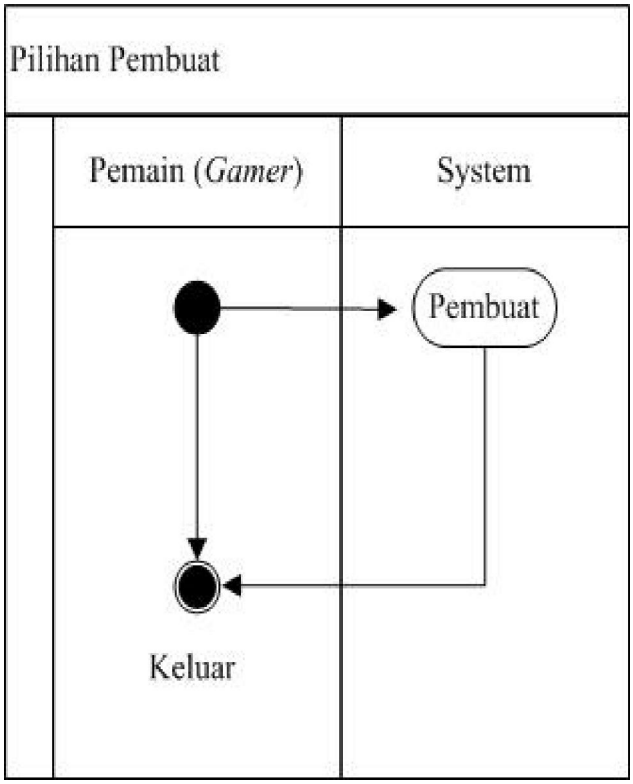


Gambar 3.29 Activity Diagram Pada Proses View Petunjuk

3.7.3 Proses View Pembuat

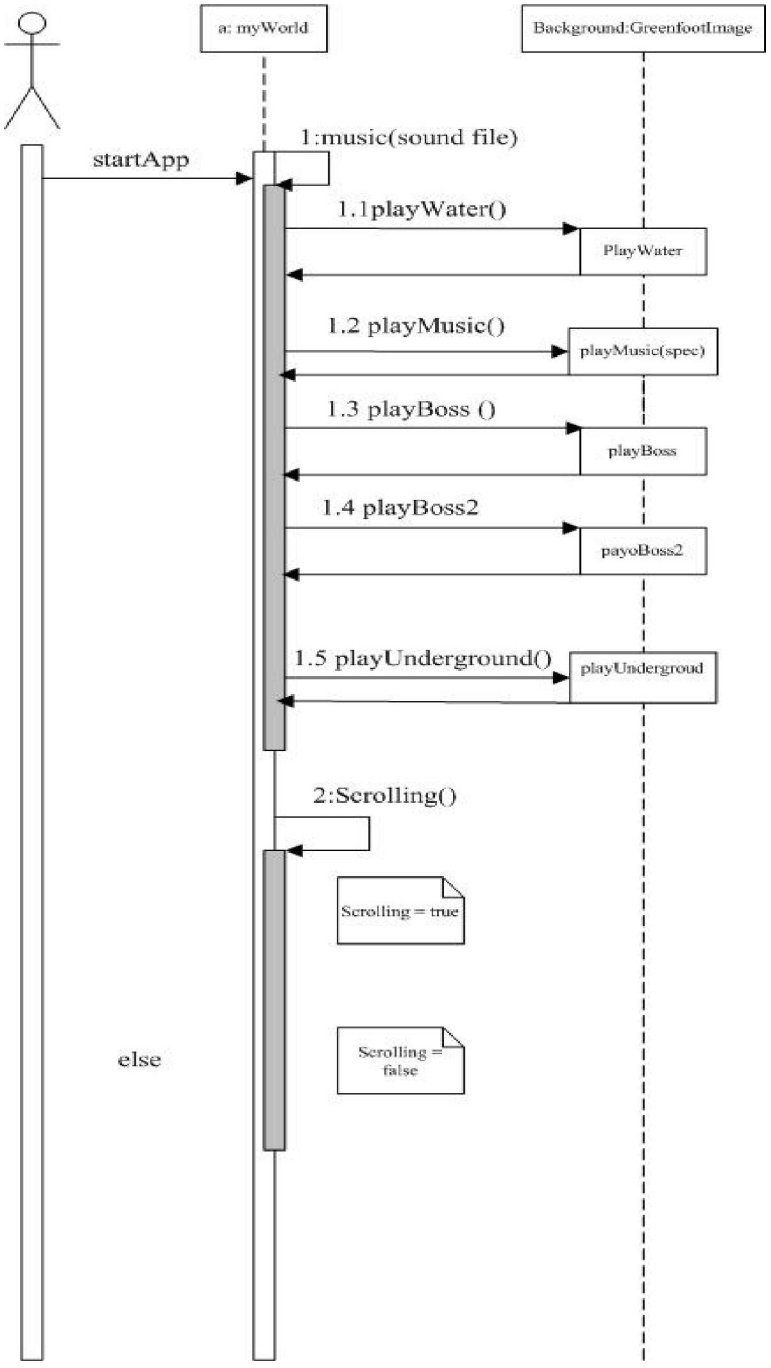
Tabel 3.20 Penjelasan View Pembuat

Deskripsi	Tampilan ini merupakan tampilan yang menampilkan keterangan atau informasi mengenai pembuat Aplikasi
Normal Flow	1. Pemain memilih pilihan pembuat untuk melihat informasi mengenai pembuat Aplikasi. 2. Pemain keluar dari tampilan pembuat.

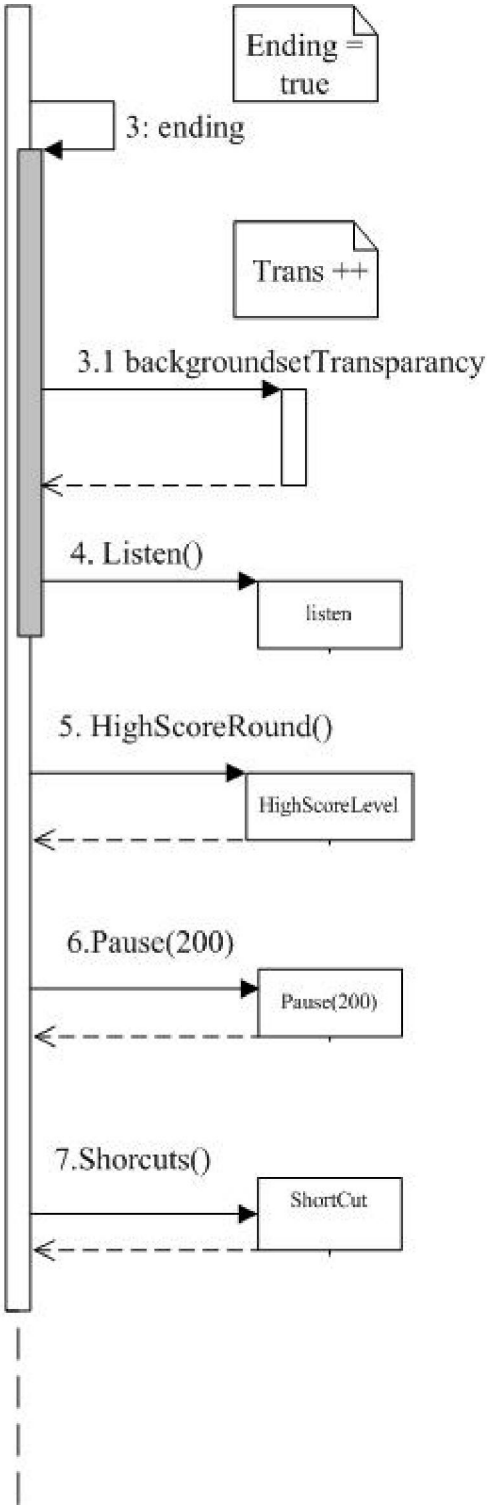


Gambar 3.30 Activity Diagram Pada Proses View Pembuat

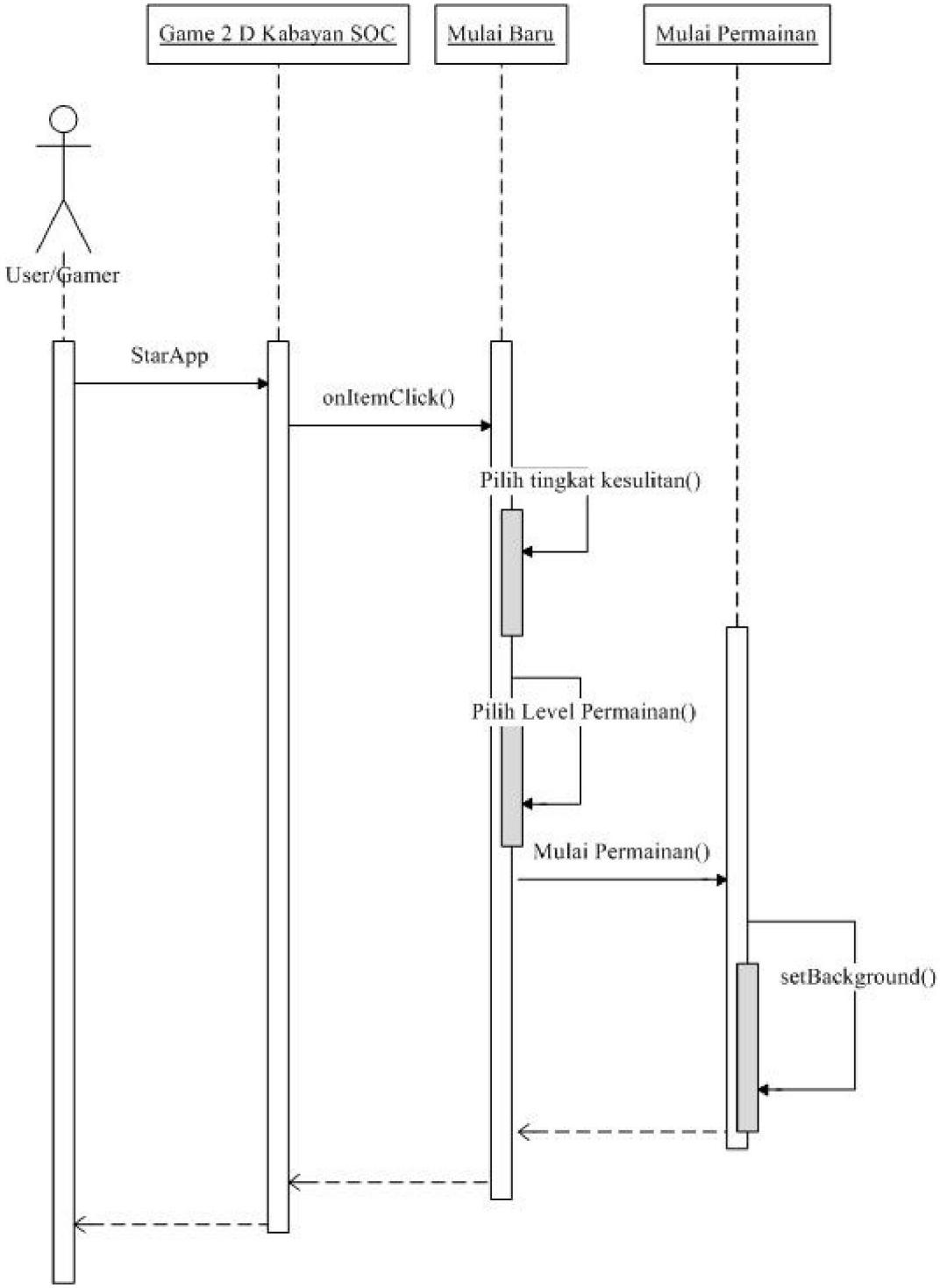
3.7.4 Perancangan Sequence Diagram



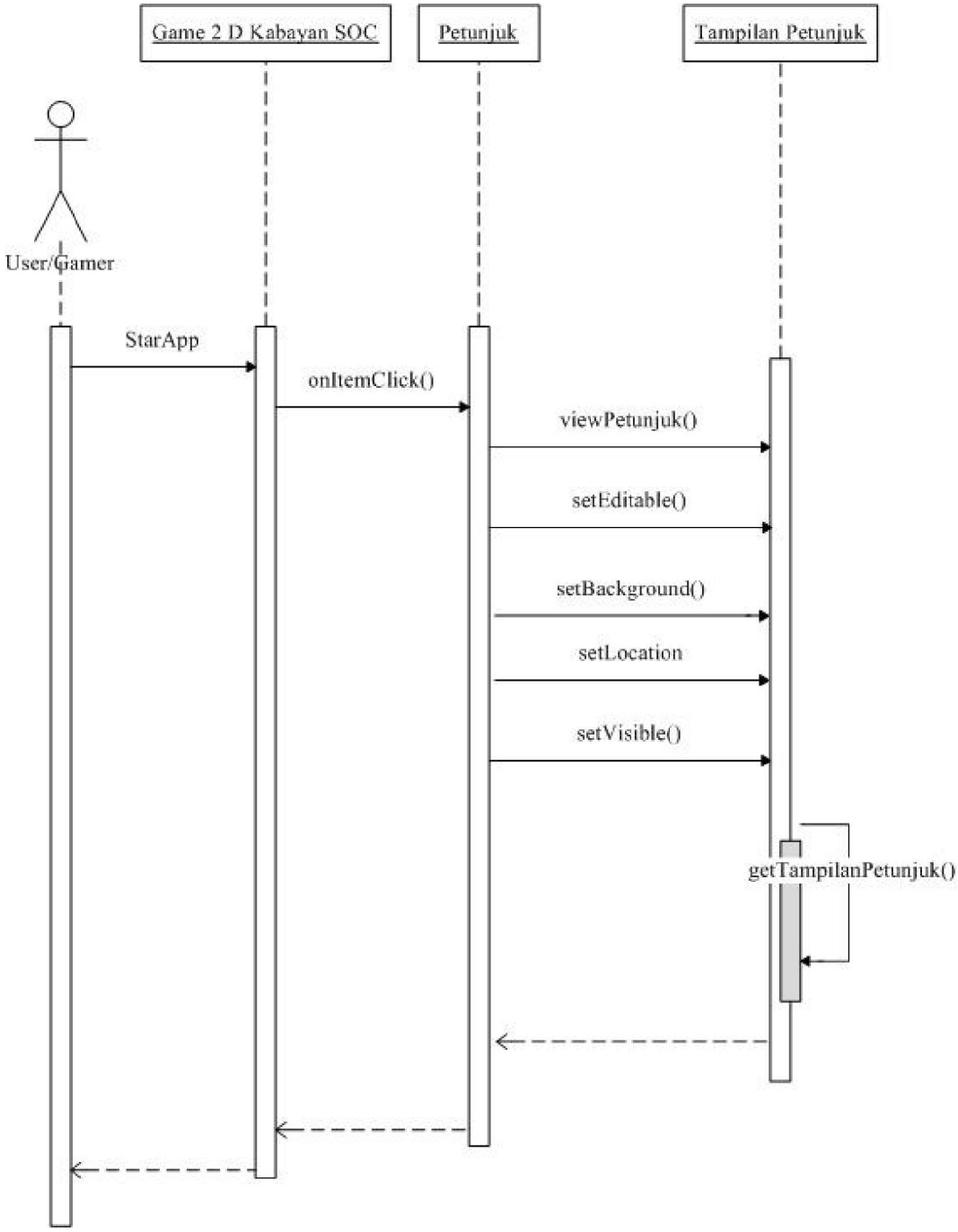
Gambar 3.31 Sequential Diagram myWorld



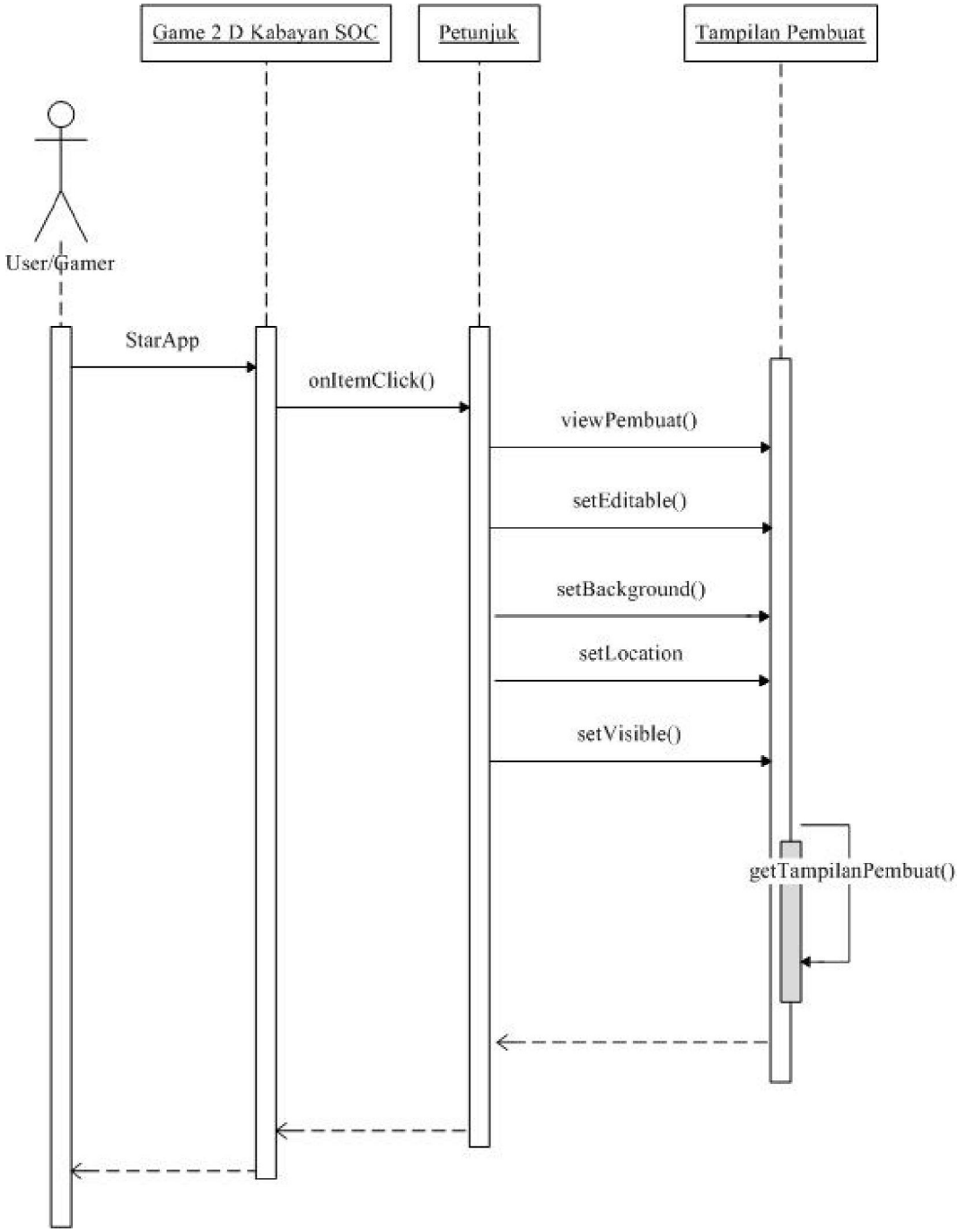
Gambar 3.32 Sequential Diagram myWorld Lanjutan



Gambar 3.33 Sequence Diagram Mulai Baru



Gambar 3.34 Sequence Diagram Petunjuk



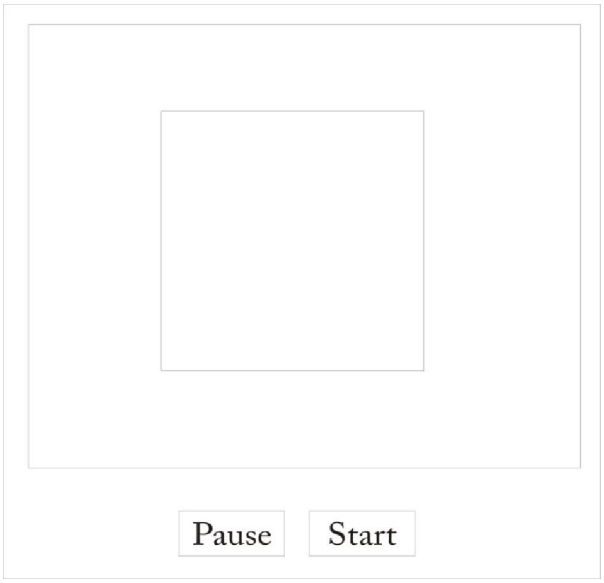
Gambar 3.35 Sequence Diagram Pembuat

3.8 Perancangan Antar Muka

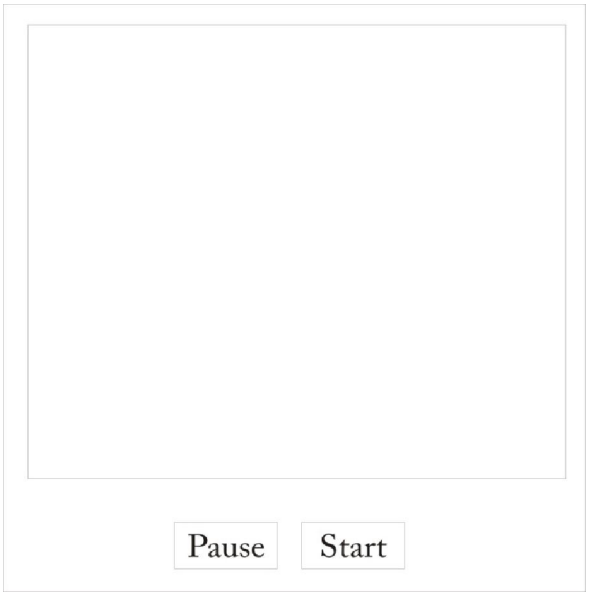
Perancangan antarmuka dibutuhkan untuk mewakili keadaan sebenarnya dari aplikasi yang akan dibangun, berikut akan disajikan perancangan antarmuka dari aplikasi yang akan dibangun:



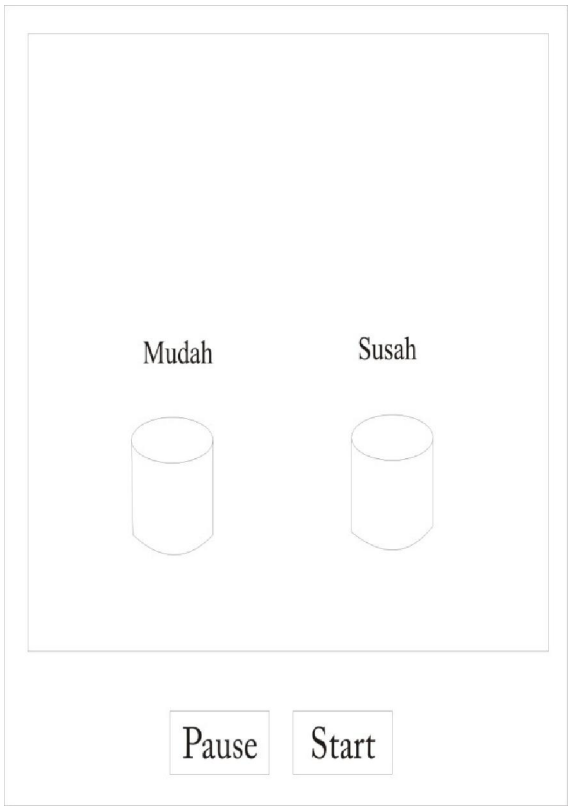
Gambar 3.36 Antar Muka Tampilan Utama



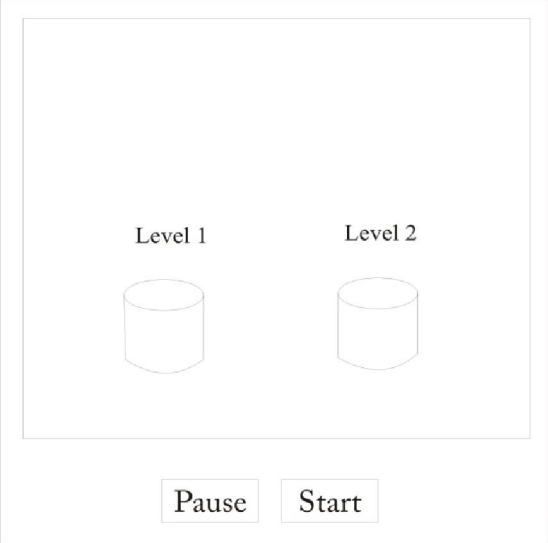
Gambar 3.37 Antar Muka Pembuat



Gambar 3.38 Antar Muka Petunjuk



Gambar 3.39 Antar Muka Mulai Baru



Gambar 3.40 Antar Muka Pilih Level

3.9 Perancangan Algoritma Secara Keseluruhan

Implementasi algoritma secara keseluruhan dibagi ke dalam tiga class utama yaitu :

1. Class World
2. Class Actor
3. Class Vector

Tiga class utama diatas merupakan API yang sudah tersedia pada IDE Greenfoot untuk kemudian pada nantinya kita membuat sub-sub class berdasarkan bahasa pemrograman java. Hal tersebut bisa kita lakukan dengan cara membungkus (polimorfism) atau dengan cara mewariskan (inheritance) dari objek-objek yang akan kita bangun.

Algoritma game Kabayan Saving Our Cultures sendiri secara umum algoritma yang dirancang dirangkum ke dalam class myWorld di dalam kelas World yaitu :

Class myWorld

java.lang.Object

greenfoot.World

myWorld

public class myWorldextends greenfoot.World

Tabel 3.21 Inisialisasi Class myWorld

Field Summary (Pembagian field-field)	
static boolean	bpicked
int	c
boolean	deadKabayan
int	deaths
static int	fireX

static int	fireY
int	gravity
static int	KabayanX
static int	KabayanY
int	kills
boolean	moosic
int	objectX
int	objectY
boolean	Scrolling
static int	startX
static int	startY
int	sumurPos
int	sumurX
int	t

Konstruktor Class myWorld

Inisialisasi myWorld()

Tabel 3.22 Method-method pada class myWorld

Method Summary (Method-method yang digunakan)	
void	act()
void	addedToWorld(greenfoot.World world)
void	bubbles()
void	cleanUp()
void	ending()
void	HighScoreRound()
boolean	isAcid(int x, int y)
boolean	isGround(java.awt.Color mg)
boolean	isGround(int x, int y)
boolean	isHurting(int x, int y)
boolean	isIce(int x, int y)

boolean	isJumpPad(int x, int y)
boolean	isPatrol(int x, int y)
boolean	isPipe(int x, int y)
boolean	isWater(int x, int y)
void	music()
void	music(int spec)
void	nextLevel()
boolean	onLadder(int x, int y)
static void	paintBg(int x, int y)
void	pause(int time)
void	playBoss()
void	playBoss2()
void	playMusic()
void	playMusic(int spec)

void	playUnderground()
void	playWater()
boolean	radarsDestroyed()
void	reload()
void	restartGame()
void	restartLevel()
void	reverseGravity()
void	Scrolling()
void	setKabayanCoords(int px, int py)
void	setLevel_boss1()
void	setLevel_boss2()
void	setLevel_devLevel()
void	setLevel_episode1to2()
void	setLevel_introLevel()

void	setLevel_lvl1()
void	setLevel_lvl12()
void	setLevel_lvl13()
void	setLevel_lvl14()
void	setLevel_lvl15()
void	setLevel_lvl16()
void	setLevel_lvl17()
void	setLevel_lvl18()
void	setLevel_lvl2()
void	setLevel_lvl3()
void	setLevel_lvl4()
void	setLevel_lvl5()
void	setLevel_lvl6()
void	setLevel_lvl7()

void	setLevel_lvl8()
void	setLevel_lvlending()
void	setLevel_preboss1()
void	setLevel_preboss2()
void	setLevel_select()
void	setLevel_select2()
void	setLevel(int a)
void	setRound(int b)
void	started()
void	stopAllAudio()
void	stopped()
boolean	toggleDifficulty()
boolean	toggleFireDifficulty()

Tabel 3.23 Method-method yang diimpor dari class greenfoot

Methods inherited from class java.lang.Object (Method-method yang mewarisi/diimpor dari class java.lang.Object)
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Tabel 3.24Tabel 3.19 Method-method yang diimpor dari class java.lang.object

Methods inherited from class greenfoot.World (Method-method yang diwarisi/diimpor dari class greenfoot)
addObject, getBackground, getCellSize, getColorAt, getHeight, getObjects, getObjectsAt, getWidth, numberOfObjects, removeObject, removeObjects, repaint, setActOrder, setBackground, setBackground, setPaintOrder

Field Detail

bpicked

public static boolean bpicked

c

public int c

deadKabayan

public boolean deadKabayan

deaths

public int deaths

fireX

public static int fireX

fireY

public static int fireY

gravity

public int gravity

KabayanX

public static int KabayanX

KabayanY

public static int KabayanY

kills

public int kills

moosic

public boolean moosic

objectX

public int objectX

objectY

public int objectY

Scrolling

public boolean Scrolling

startX

public static int startX

startY

public static int startY

sumurPos

public int sumurPos

sumurX

public int sumurX

t

public int t

Constructor Detail

myWorld

public myWorld()

Method Detail

act

public void act()

Overrides:

act in class greenfoot.World

addedToWorld

public void addedToWorld(greenfoot.World world)

bubbles

public void bubbles()

cleanUp

public void cleanUp()

ending

public void ending()

HighScoreRound

public void HighScoreRound()

isAcid

public boolean isAcid(int x, int y)

isGround

public boolean isGround(java.awt.Color mg)

isGround

public boolean isGround(int x, int y)

isHurting

public boolean isHurting(int x, int y)

isIce

public boolean isIce(int x, int y)

isJumpPad

public boolean isJumpPad(int x, int y)

isPatrol

public boolean isPatrol(int x, int y)

isPipe

public boolean isPipe(int x, int y)

isWater

public boolean isWater(int x, int y)

music

public void music()

music

public void music(int spec)

nextLevel

public void nextLevel()

paintBg

public static void paintBg(int x, int y)

pause

public void pause(int time)

playBoss

public void playBoss()

playBoss2

public void playBoss2()

playMusic

public void playMusic()

playMusic

public void playMusic(int spec)

playUnderground

public void playUnderground()

playWater

public void playWater()

radarsDestroyed

public boolean radarsDestroyed()

reload

public void reload()

restartGame

public void restartGame()

restartLevel

public void restartLevel()

reverseGravity

public void reverseGravity()

Scrolling

public void Scrolling()

setKabayanCoords

public void setKabayanCoords(int px, int py)

setLevel_boss1

public void setLevel_boss1()

setLevel_boss2

public void setLevel_boss2()

setLevel_devLevel

public void setLevel_devLevel()

setLevel_episode1to2

public void setLevel_episode1to2()

setLevel_introLevel

public void setLevel_introLevel()

setLevel_lvl1

public void setLevel_lvl1()

setLevel_lvl12

public void setLevel_lvl12()

setLevel_lvl13

public void setLevel_lvl13()

setLevel_lvl14

public void setLevel_lvl14()

setLevel_lvl15

public void setLevel_lvl15()

setLevel_lvl16

public void setLevel_lvl16()

setLevel_lvl17

public void setLevel_lvl17()

setLevel_lvl18

public void setLevel_lvl18()

setLevel_lvl2

public void setLevel_lvl2()

setLevel_lvl3

public void setLevel_lvl3()

setLevel_lvl4

public void setLevel_lvl4()

setLevel_lvl5

public void setLevel_lvl5()

setLevel_lvl6

public void setLevel_lvl6()

setLevel_lvl7

public void setLevel_lvl7()

setLevel_lvl8

public void setLevel_lvl8()

setLevel_lvlending

public void setLevel_lvlending()

setLevel_preboss1

public void setLevel_preboss1()

setLevel_preboss2

public void setLevel_preboss2()

setLevel_select

public void setLevel_select()

setLevel_select2

public void setLevel_select2()

setLevel

public void setLevel(int a)

setRound

public void setRound(int b)

started

public void started()

Overrides:

started in class greenfoot.World

stopAllAudio

public void stopAllAudio()

stopped

public void stopped()

Overrides:

stopped in class greenfoot.World

toggleDifficulty

public boolean toggleDifficulty()

toggleFireDifficulty

public boolean toggleFireDifficulty()