BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya

3.1.1 Analisis Masalah

Berdasarkan analisis sistem yang dilakukan dengan mengevaluasi game sejenis yang sudah ada yaitu game perawatan karapan sapi maka masalah yang didapatkan yaitu gameplay yang kurang menarik untuk dimainkan dan juga perlu ditambahkan entitas musuh.Untuk lenih jelasnya lagi akan dijelaskan sebagai berikut :

1. Masih jarangnnya game race yang bertemakan kebudayaan dan karakter dengan ciri khas Indonesia
2. Bagaimana membuat pergerakan NPC menjadi lebih dinamis sehingga tidak hanya bisa melewati rintangan saja tetapi bisa mencari rute yang lebih pendek untuk bisa mencapai garis *finish*

3.1.2 Analisis Game Sejenis

Sebelum merancang *game* yang akan dibuat, terlebih dahulu dilakukan analisis pada *game* sejenis yang sudah ada. Ini bertujuan agar *game* yang sudah dibuat nanti memiliki solusi dari permasalahan *game* yang sudah ada.

3.1.2.1 Analisis *Game* Perawatan Sapi Kerapan

Game Perawatan Sapi Karapan merupakan *game* bergenre peternakan berbasis desktop.*Gameplay* pada *game* ini cukup sederhana yaitu pemain harus memelihara seekor sapi diantaranya memberikan makan , minum , menjemur sapi dan memberinya latihan supaya bisa diperlombakan dalam perlombaan .



Gambar 3-1 Game Perawatan Sapi Karapan

Pada *game* ini tidak mempunyai level pemain hanya memelihara sapi sampai besar dan dijual sampai mamasuki kategori mahal. Secara umum *game perawatan sapi karapan* dapat dikategorikan sebagai permainan yang sederhana dan mudah dimainkan tetapi membutuhkan konsentrasi yang tinggi. Namun *game* perawatan sapi karapan memiliki *game*play yang monoton dimana pemain hanya dapat memelihara sapi saja tanpa bisa mengikuti pertandingan karapan sapi yang sebenarnya.

3.1.3 Analisis *Game* Yang Dibangun

Aplikasi game yang akan dibangun yaitu berupa game 2D, bergenre *race*. Pada menu utama terdapat pilihan yaitu: *new game* (memulai permainan), petunjuk dan keluar. *Game* ini haya memiliki satu pemain atau *single player*, tokoh utama dari *game* bernama slamet, pemain akan melakukan balapan melawan musuh dan disetiap *stage* akan jumlah musuh yang bertanding berbeda beda.Di dalam permainan ini pemain harus mendapatkan karung rumput yang akan berguna untuk menambah kecepatan dan kekuatan sapi tersebut.Selain itu pemain harus melompati gunukan tanah yang ada ketika permainan berlangsung dan dilevel akhir pemain juga harus bisa menghindar sapi yang berkeliaran di sepanjang lintasan .

Fitur-fitur yang akan dibuat pada *gamerace* ini adalah sebagai berikut :

1. Sistem *single player*
2. Karakter utama memiliki kemampuan untuk melewati rintangan dan mendapatkan karung rumput yang akan berpengaruh terhadap kecepatan sapi pada saat balapan berlangsung
3. Memiliki 3 level dengan tingkat kesulitan yang berbeda-beda.
4. Control karakter menggunakan *touchscreen*.

5. Grafik yang digunakan merupakan grafik *game* 2D.

3.1.3.1 *Storyline*

Demi mendapatkan hadiah uang dan 5 ekor sapi. Slamet akan mengikuti pertandingan di Pulau Madura,dimana pertandingan pertama dilakukan di kabupaten Bangkalan didua tempat yang berbeda,selanjutnya Slamet bertanding di Kabupaten Sampang setelah memenangkan pertandingan di dua Kabupaten sebelumnya tiba saatnya Slamet bertanding di Kabupaten terakhir yaitu Kabupaten Pamekasan.Disetiap kabupatennya Slamet mendapat perlawanan sengit dari lawan yang dihadapinya

3.1.3.2 *Gameplay*

Pada permainan ini, tugas utama pemain adalah user akan melakukan perlombaan dalam dengan melawan musuh yang tingkat kesulitannya berbeda pada setiap levelnya dan juga harus melewati rintangan yang ada di sepanjang linitasan.*Game* ini terdiri dari tiga level dimana setiap levelnya terdapat dua stage yang harus dimainkan oleh user,jumlah musuh pada setiap levelnya berbeda-beda.Selain itu pada saat perlombaan berlangsung user juga harus mendapatkan karung rumput sebanyak banyaknya , hal ini akan memberikan kekuatan tambahan kepada sapi untuk bisa berlari lebih kencang dalam mengejar musuhnya dan juga untuk menambah poin agar bisa melanjutkan ke *stage* dan level berikutnya.Dalam permainan ini algoritma a* akan diterapkan pada sapi yang menjadi lawan *user* tujuannya untuk memberikan kecerdasan kepada musuh agar bisa melewati rintangan dan mencapai garis finish sehingga akan memberikan perlawanan yang sulit kepada user.Aspek lain yang ditanamkan adalah penerapan collision detection-nya terjadi ketika pemain menabrak rintangan,mendapatkan karung rumput dan menabrak sapi lawan .Untuk sistem kendali pada game ini akan menggunakan touchscreen atau layar sentuh,jadi pengguna game ini hanya menekan *button* yang ada dilayar dan kontroling game ini ada dua yaitu melompat dan berlari

3.1.3.3 Scoring

Pada bagian ini akan dijelaskan bagaimana pemberian skor yang ada pada *game*. Berikut cara memperoleh *score* dari game*race* karapan sapi :

1. Setiap satu karung rumput yang didapat user akan mendapat poin 100

3.1.3.4 Leveling

Dalam *game race* Karapan Sapi terdapat 3 level yang masing-masing level dibedakan berdasarkan tingkat kesulitan rintangan yang harus dihadapi. Berikut ini penjelasan tingkat level kesulitan dapat dilihat pada Tabel 3-1.

Tabel 3-1 Leveling Gaame

Level	Deskripsi
1	Musuh yang akan dihadapi 1 sapi dengan panjang lintasan 100 meter dan rintangan yang harus dilewati adalah genangan lumpur disepanjang lintasan.
2	Musuh yang akan dihadapi 1 sapi dengan panjang lintasan 200 meter dan rintangan yang harus dilewati adalah genangan air
3	Musuh yang akan dihadapi 1 sapi dengan panjang lintasan 300 meter dan rintangan yang harus dilewati adalah genangan air dan sapi yang berkeliaran.

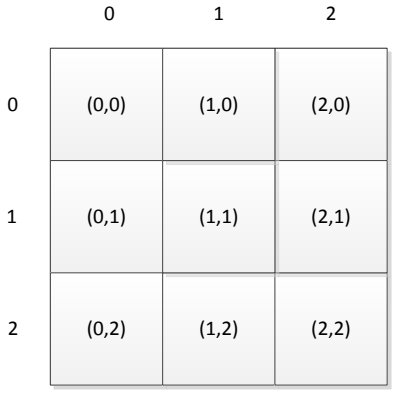
3.1.3.4 Kendali Game

Untuk memainkan game race karapan sapi ini pengguna hanya menggerakan jari tangannya di layar sentuh ke arah kanan ataupun kiri.

3.1.4 Analisis Algoritma

Agar *game* menjadi lebih menarik dapat dilakukan dengan meningkatkan kualitas *gameplay* bagi pengguna yang memainkannya. AI merupakan faktor yang penting dalam meningkatkan *gameplay* pada semua jenis *game*, termasuk pada *game* yang akan dibangun. AI yang akan digunankan pada pembangunan *gamerace*karapan sapi ini adalah A*. A* merupakan algoritma pencarian jalur terpendek menuju suatu tujuan dengan secara optimal.Pencarian jalur dalam *gamerace*karapan sapi ini diterapkan pada objek musuh besar dalam melakukan pencarian karakter pemain di area permainan. Pada *gamerace* karapan sapi arena

permainan berupa kumpulan tile atau grid yang terdiri dari orde X × Y. Tiap grid memiliki koordinat X dan Y atau berbentuk seperti array dua dimensi.



Gambar 3-2 Grip Map Orde X×Y

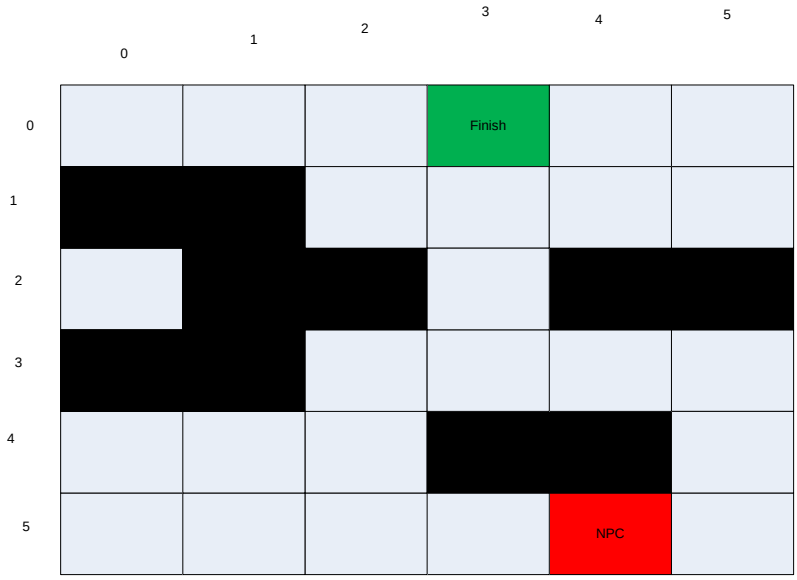
Algoritma A* dalam *game race* ini akan menganggap grid tersebut sebagai simpul-simpul yang membentuk jalan (path) dari simpul awal hingga ke simpul akhir tujuan, lalu menghitung biaya dan mengambil jalan yang memakan biaya paling sedikit.

$f(n) = g(n) + h(n)$

f(n) adalah jumlah dari g dan h yang merupakan estimasi terbaik dari biaya jalan yang akan melalui simpul saat ini. Intinya semakin rendah nilai f maka akan lebih efisien. g(n) adalah biaya yang didapatkan dari simpul awal ke simpul selanjutnya yang adalah jumlah dari semua nilai di jalan antara simpul awal dan simpul selanjutnya. h(n) merupakan singkatan dari heuristik yang merupakan perkiraan biaya dari simpul saat ini ke simpul tujuan. Untuk menghitung h digunakan fungsi heuristik manhattan distance dengan fungsi sebagai berikut:

h(n)= abs(nx-goal.x) + abs (ny-goal.y)

Arena permainan pada *game race* terdiri dari orde X×Y. Dalam analisis ini akan dijelaskan dengan ordo 6×7. Simpul awal atau posisi musuh dicontohkan berada pada koordinat (4,5), sedangkan simpul tujuan atau posisi karakter pemain berada pada koordinat (3,0) di arena permainan.



Gambar 3-3 Posisi Musuh Dan Pemain Dalam Grid

Setelah mengetahui simpul awal dan simpul tujuan, pertama-tama simpul awal yang ada di *open list* di masukkan ke *closed list*. Langkah selanjutnya dilakukan pemeriksaan terhadap simpul simpul yang dapat dilalui musuh dan memiliki nilai *cost* / biaya terkecil sebagai *bestnode*. Setelah diperiksa terdapat tiga simpul yang dapat dilalui oleh musuh yaitu simpul dengan koordinat (2,1), (1,2) dan (2,3), simpul dengan koordinat (3,2) tidak dapat dilalui karena terdapat penghalang yang tidak dapat dilalui musuh. Simpul-simpul yang diperiksa dimasukkan ke dalam open list. Langkah perhitungan untuk mendapatkan *bestnode* sebagai berikut:

1. Langkah pertama

a. Koordinat (3,5)

Nilai nx = 3

Nilai ny = 5

Simpul tujuan (3,0) sehingga goal.x = 3 dan goal.y = 0

g(3,5) = 1

h(3,5) = abs(nx-goal.x) + abs (ny-goal.y)

= abs(3-3) + abs(5-0)

= abs(0) + abs(5)

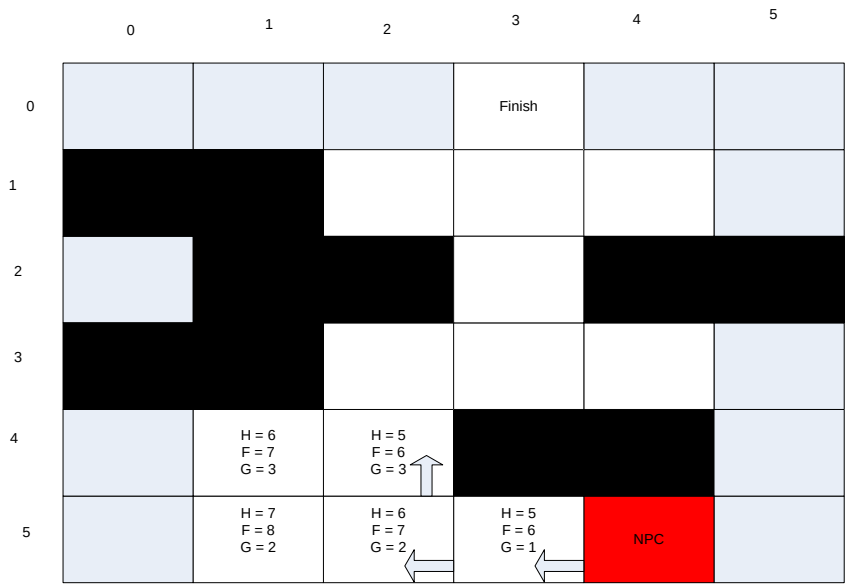
= 0 + 5

= 5

f(3,5) = g(3,5) + h(3,5)

= 1 + 5

= 6



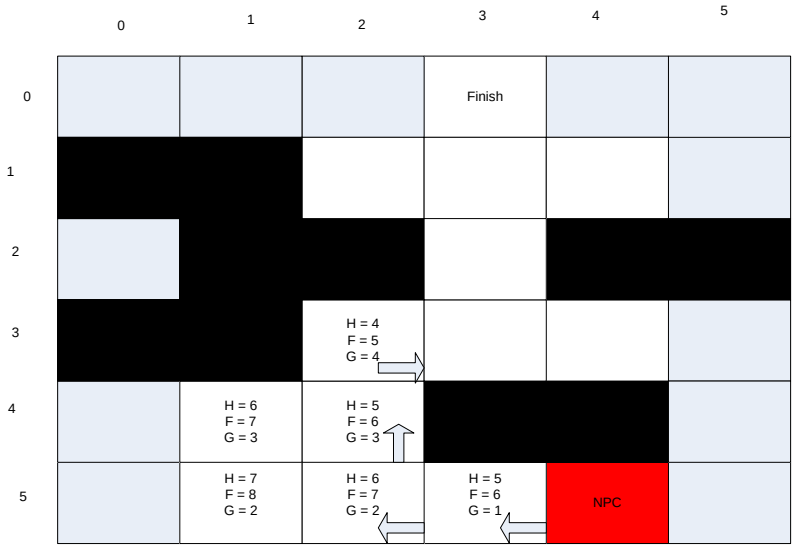
Gambar 3-4 Langkah Pencarian Pertama Bestnode

Selanjutnya simpul tetangga dari simpul (2,5) akan diperiksa dan dimasukkan ke dalam open list yaitu simpul simpul (2,4) dan simpul (1,5).Hasil pemeriksaan simpul (2,4) dan (1,5) dapat dilihat pada Tabel 3-2.

Tabel 3-2 Langkah Kedua Pencarian f

Iterasi	Titik	g	h	f
1	1,5	3	5	**6**
	2,4	3	5	8

Dari hasil perhitungan yang dilakukan simpul yang memiliki nilai cost / biaya terkecil adalah simpul dengan koordinat (1,5) maka dipilihlah simpul (1,5) sebagai bestnode dan simpul yang telah diperiksa dimasukkan ke closed list agar tidak diperiksa ulang.



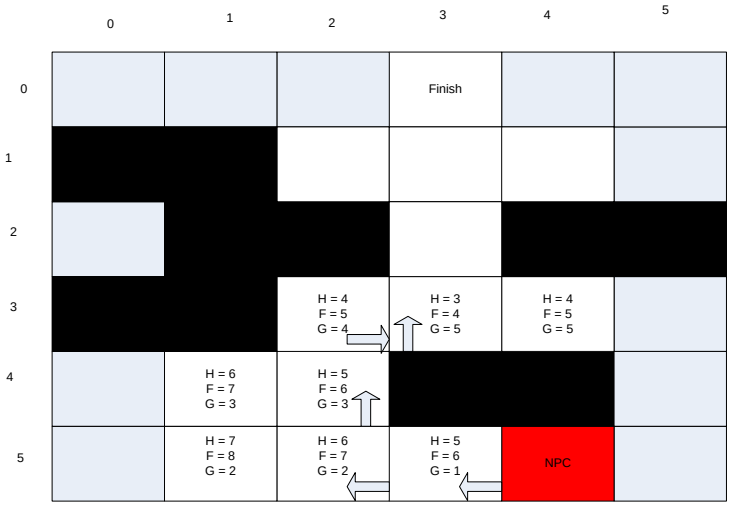
Gambar 3-5 Langkah Pencarian Kedua Bestnode

Selanjutnya simpul tetangga dari simpul (2,4) akan diperiksa dan dimasukkan ke dalam open list yaitu simpul (2,3) dan (1,4). Simpul (3,4) tidak dapat diperiksa karena terdapat penghalang yang tidak bisa dilalui. Hasil pemeriksaan simpul dapat dilihat pada Tabel 3-3.

Tabel 3-3 Langkah Ketiga Pencarian f

Iterasi	Titik	g	h	f
2	1,4	3	6	7
	2,3	3	4	**5**

Dari hasil perhitungan yang dilakukan simpul dengan koordinat (2,3) menjadi bestnode karena memiliki nilai cost / biaya terkecil.



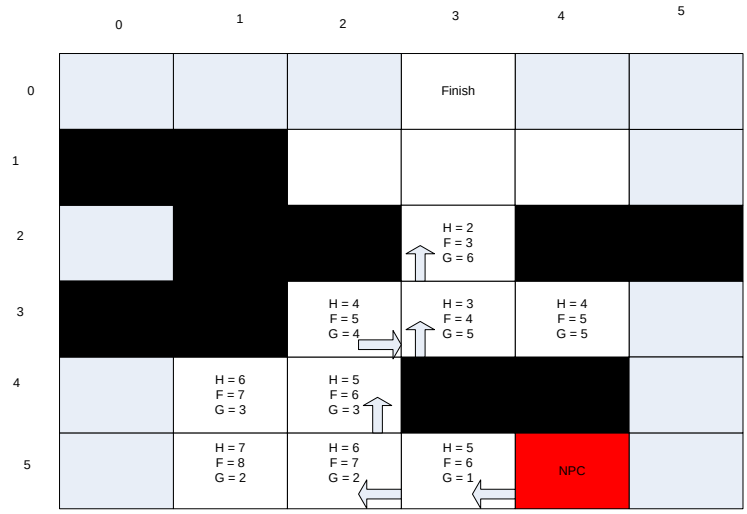
Gambar 3-6 Langkah Pencarian Ketiga Bestnode

Setelah mendapatkan bestnode dilangkah ketiga, dilakukan penelusuran selanjutnya untuk mendapatkan bestnode selanjutnya dengan melakukan penelusuran simpul-simpul yang memiliki nilai cost terkecil dari simpul tetangga (2,3) yaitu simpul dengan koordinat (3,3)Simpul yang akan diperiksa di masukan ke dalam open list. Hasil dari pemeriksaan dapat dilihat pada Tabel 3-4.

Tabel 3-4 Langkah Keempat pencarian f

Iterasi	Titik	g	h	f
3	3,3	4	3	4
	4,3	-	-	-

Dari hasil perhitungan yang dilakukan, simpul dengan koordinat (3,3) menjadi bestnode karena merupakan satu-satunya simpul yang mempunyai nilai cost terkecil dan simpul yang telah diperiksa dimasukkan ke closed list agar tidak diperiksa ulang.



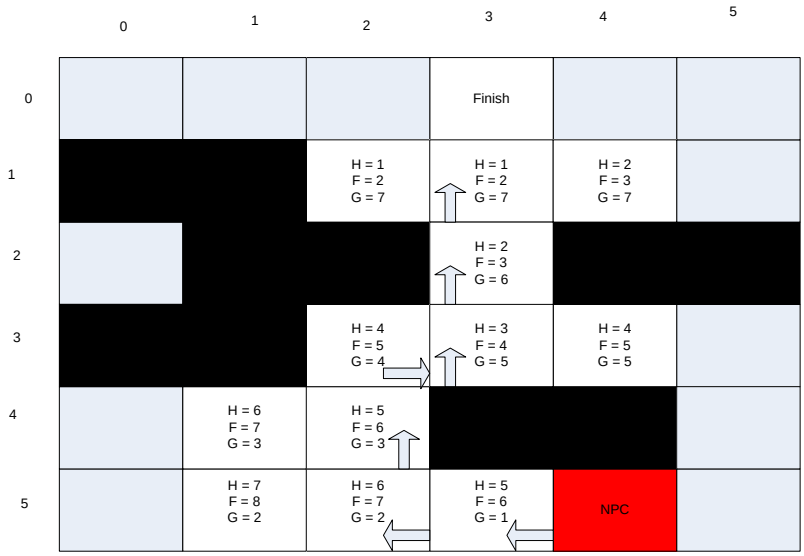
Gambar 3-7 Langkah Pencarian Keempat Bestnode

Setelah mendapatkan bestnode dilangkah keempat, dilakukan penelusuran selanjutnya untuk mendapatkan bestnode selanjutnya dengan melakukan penelusuran simpul-simpul tetangga dari simpul (3,2). Simpul yang akan diperiksa selanjutnya adalah simpul (3,1). Simpul yang akan diperiksa dimasukkan ke dalam open list. Hasil dari pemeriksaan simpul (3,1). dapat dilihat pada Tabel 3-5.

Tabel 3-5 Langkah Kelima Pencarian f

Iterasi	Titik	g	h	f
4	3,1	6	1	2

Dari hasil perhitungan yang dilakukan, simpul dengan koordinat (3,1) menjadi bestnode karena merupakan simpul yang mempunyai nilai cost terkecil dan simpul yang telah diperiksa dimasukkan ke closed list agar tidak diperiksa ulang.



Gambar 3-8 Langkah Pencarian Bestnode

Pada *game*race Karapan Sapi yang akan dibangun algoritma A* dijalankan di setiap level. Algoritma ini akan berjalan apabila NPC telah memasuki area balapan sehingga bisa mencapau garis finish.

3.1.5 Analisis Kebutuhan Non Fungsional

Analisis non-fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan.

Pada analisis kebutuhan sistem non-fungsional ini dijelaskan analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras, dan analisis pengguna.

3.1.5.1 Analisis Kebutuhan Perangkat Keras

Agar aplikasi dapat berjalan dengan baik, maka dibutuhkan perangkat keras yang sesuai dengan kebutuhan aplikasi. Di bagian ini akan dijelaskan analisis kebutuhan perangkat keras dari sisi pengembang dan dari sisi pemain.

1. Dari sisi pengembang

   Perangkat keras yang digunakan untuk membangun *game* ini adalah sebagai berikut:

   1. Prosesor 2 GHz
   2. Memori 4 GB
   3. Space kosong di harddisk 5 GB
   4. VGA card 512 MB
   5. Display LCD Monitor dengan resolusi 1366 x 768

2. Dari sisi pemain

   Spesifikasi minimum perangkat keras yang dibutuhkan agar dapat menjalankan *game* ini adalah sebagai berikut:

   1.Smartphone Android 4.0 (*icecream sandwich*) ke atas

   2.Layar touchscreen

   3.Processor 1 Ghz, 512 MB RAM

3.1.5.2 Analisis Kebutuhan Perangkat Lunak

Perangkat keras komputer tidak berarti tanpa perangkat lunak, begitu juga sebaliknya. Jadi perangkat lunak dan perangkat keras saling mendukung satu sama lain. Perangkat keras hanya berfungsi jika diberikan instruksi - intruksi kepadanya.Instruksi-instruksi inilah yang disebut dengan perangkat lunak. . Di bagian ini akan dijelaskan analisis kebutuhan perangkat keras dari sisi pengembang dan dari sisi pemain.

1. Dari sisi pengembang

Perangkat lunak yang digunakan untuk mengembangkan *game* ini adalah sebagai berikut:

   1. Sistem operasi *Windows*
   2. Adobe Flash 5.5

2. Dari sisi pemain

Gamerace “Karapan sapi ” ini dibutuhkan perangkat lunak untuk menjalankannya, yaitu Sistem Operasi Android 4.0 (*ice cream sandwich*).

3.1.5.3 Analisis Pengguna

Analisis pengguna gambaran tentang target maupun profil pengguna untuk *game* yang akan dibangun. Perangkat keras dan perangkat lunak yang ada tidak akan berguna apabila tidak ada pengguna yang mengoperasikannya. Berikut ini target maupun profil pengguna untuk *game race* “Karapan sapi ”, dapat dilihat pada Tabel 3-6.

1. Target Pengguna

Target pengguna (target user) untuk *game* ini adalah pengguna dengan karakteristik sebagai berikut :

Tabel 3-6 Target Pengguna

1	Peran	Pemain
2	Usia	12 tahun keatas
3	Tingkat Keterampilan	Dapat menggunakan *smartphone* Android
4	Pengalaman	Pernah memainkan aplikasi *game mobile* Pernah memainkan aplikasi *game* dengan layar sentuh
5	Minat terhadap *game casual*	Baik

2. Profil Pengguna

Profil pengguna (user profile) untuk *game* ini dapat dikelompokkan sebagai berikut :

a. Human Information Processing System

Sistem pemroses informasi manusia yaitu berkaitan dengan antarmuka dan fungsionalitas. Antarmuka dan fungsionalitas yang ada dalam *game* ini dibuat sederhana agar pengguna dapat dengan mudah memainkan *game* ini.

b. User Pshycology Characteristic

Karakteristik psikologi pengguna untuk *game* ini haruslah pengguna yang memiliki kegemaran atau setidaknya berminat untuk memainkan *game* berjenis *adventure*. Selain itu juga tertarik dengan kebudayaan.

c. User Knowledge and Experience Characteristic

Pengguna yang memainkan *game* ini harus memiliki pengetahuan dasar tentang cara pengoperasian *smartphone* android. Selain itu, pengguna juga diharapkan pernah bermain *game* berjenis *adventure* sebelumnya.

d. User Job and Task Characteristic

Untuk lebih memudahkan dalam memainkan *game* ini, Pengguna harus memiliki tingkat frekuensi penggunaan (*frequency of use*) yang tinggi. Kategori pekerjaan pengguna yang memainkan *game* ini mencakup siswa atau siapa saja yang berusia 10 tahun ke atas yang ingin mempelajari kebudayaan dan gemar bermain *game*.

e. User Physical Characteristic

Diharapkan tidak positif mengidap buta warna karena *game* ini adalah *game* yang full color, namun tidak mempengaruhi banyak saat memainkannya. *Handedness user* tidak mempengaruhi dalam memainkannya karena tidak akan berefek *human error* bila semua *skillhandedness* di implementasikan, hanya perlu penyesuaian. Selain itu, Gender pengguna untuk *game* ini tidak dibatasi atau dengan kata lain, *game* ini dapat dimainkan oleh pengguna dengan gender apapun.

f. User Physical Environment

Dalam memainkan *game* ini, dibutuhkan cukup konsentrasi pengguna agar cerita dan pesan dalam *game* ini dapat tersampaikan dan dapat diingat dengan baik oleh pengguna. Selain itu, *game* ini tidak terlalu mempermasalahkan mengenai privasi karena pada dasarnya *game* ini dibuat untuk umum.

g. User Tools

Selain *smartphone* android, pengguna tidak perlu menggunakan alat tambahan lainnya dikarenakan hanya dengan memiliki

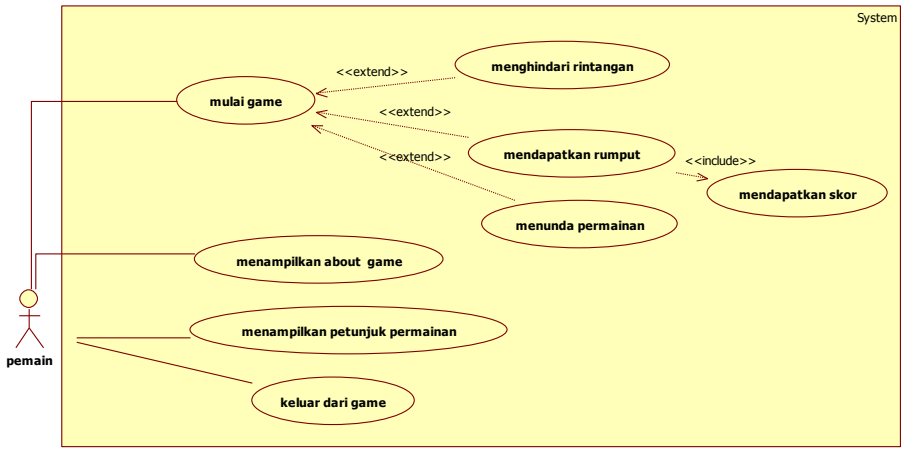
*smartphone*android pengguna dapat dengan mudah memainkan *game* ini.

3.1.6 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik sesuai kebutuhan. Pada bagian ini akan dibahas tentang *use case diagram*, *activity diagram*, *class diagram*, dan *sequence diagram*.

3.1.6.1 Use case diagram

Pada sub bab berikut ini dijelaskan bagaimana *UseCase* yang terjadi dalam pembangunan *game* ini yang dimodelkan dalam sekumpulan *UseCase* dan *Actor* dan bagaimana hubungan-hubungannya yang akan dipetakan dalam diagram *UseCase*, dilengkapi dengan skenario untuk menjelaskan dari gambaran *UseCase* yang ada dengan menggambarkan skenario per *UseCase*. *Use case diagram* dapat dilihat pada Gambar 3-9.



Gambar 3-9 Use Case Diagram Game Karapan Sapi

1. *Actor Definition*

   Actor Definition berfungsi untuk menjelaskan *Actor* yang terdapat pada *Use case diagram*. *Actor Definition* diterangkan pada Tabel 3-7.

Tabel 3-7 Actor Definition

No	*Actor*	Deskripsi
1	Pemain	Orang yang memainkan *game*

2. *Use Case Definition*

   Use Case Definition berfungsi untuk menjelaskan fungsi *Use Case* yang terdapat pada *Use case diagram. Use Case Definition* diterangkan pada Tabel 3-8.

Tabel 3-8 Use Case Definition.

No	*Use Case*	Deskripsi
1	Mulai game	Proses untuk memulai permainan
2	Menghindari rintangan	Proses untuk menghindari rintangan
3	Mendapatkan rumput	Proses untuk mendapatkan rumput
4	Menampilkan tentang game	Proses yang dilakukan oleh pemain untuk mengetahui informasi mengenai permainan yang akan dimainkan
5	Menampilkan petunjuk permainan	Proses yang dilakukan oleh pemain untuk menampilka petunjuk permainan
6	Menunda permainan	Proses menunda permainan
7	Mendapatkan Skor	Proses mendapatkan skor setelah pemain mendpatkan karung ruput
8	Keluar game	Proses keluar dari permainan

3. *Use Case Scenario*

   Use Case Scenario mendeskripsikan urutan langkah-langkah dalam proses bisnis baik yang dilakukan aktor terhadap sistem maupun yang dilikakukan oleh sistem terhadap aktor. Berdasarkan *use case diagram* pada Gambar diatas maka *use case scenario* untuk aplikasi yang dibangun akan dijelaskan sebagai berikut.

a. *Use Case Scenario* Mulai *Game*

Use Case Scenario dari *Use Case* Menampilkan Level 1 dijelaskan pada Tabel 3-9.

Tabel 3-9 Use Case Scenario Mulai Game

Identifikasi	
Nomor	1
Nama	Mulai game
Tujuan	Proses yang dilakukan pemain untuk memulai permainan dan menampilkan permainan
Deskripsi	Proses untuk memulai permainan
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di menu utama
Aksi Aktor	Reaksi Sistem
1.Memilih Tombol Menu Main	
	2.Sistem menampilkan arena permainan
	3.Sistem mengisialisasi posisi pemain
	4.Sistem mengisialisasi posisi musuh
5.Menekan Mulai	
	6.Memulai permainan
	7.Menampilkan Level 1
8.Mengendalikan karakter pemain	
	9.Karakter bergerak
	10.Cek collision saat mendapatkan karung rumput
	11.Menghitung poin saat karung rumput didapatkan
	12.Jalankan AI lalu musuh bereaksi
13.Berhasil mengalahkan musuh	
	14.Menghitung poin saat memenangkan balapan
15.Menyelesaiakan level 1	
	16.Mengupdate informasi permainan
	17.Poin bertambah
Kondisi Akhir	Pemain lanjut ke level 2 atau kembali ke awal permainan

b. Use Case Scenario Menghindari Rintangan

Use Case Scenario dari *Use Case* Menampilkan Level 2 dijelaskan pada Tabel 3-10.

Tabel 3-10 Use Case Scenario Menghindari Rintangan

Identifikasi	
Nomor	2
Nama	Menghindari Rintangan
Tujuan	Untuk menghindari rintangan
Deskripsi	Proses untuk menghindari rintangan pada permainan
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di area permainan
Aksi Aktor	Reaksi Sistem
1.Pemain menggerakkan karakter pemain	
	2.Sistem menggerakan karakter pemain
	3.Sistem mengecek collision terhadap rintangan
Kondisi akhir	Pemain melewati rintangan

c. *Use Case Scenario* Mendapatkan Rumput

Use Case Scenario dari *Use Case* Menampilkan Level 3 dijelaskan pada Tabel 3-11.

Tabel 3-11 Use Case Scenario Mendapatkan Rumput

Identifikasi	
Nomor	3
Nama	Mendapatkan rumput
Tujuan	Untuk mendapatkan rumput
Deskripsi	Proses mendapatkan rumput diarena permainan
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di area permainan
Aksi Aktor	Reaksi Sistem
1.Pemain menggerakan karakter pemain	
	2.Sistem menggerakan karakter pemain
	3.Sistem mengecek collision terhadap karung rumput
Kondisi Akhir	Pemain mendapatkan karung rumput

d. *Use Case Scenario* Menampilkan Tentang *Game*

Use Case Scenario dari *Use Case* Menampilkan *about game*dijelaskan pada Tabel 3-12.

Tabel 3-12 Use Case Scenario Menampilkan About Game

Identifikasi	
Nomor	4
Nama	Menampilkan *About Game*
Tujuan	Proses untuk menampilkan menu *about game*
Deskripsi	
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di menu utama
Aksi Aktor	Reaksi Sistem
1.Pemain memilih menu about	
	2.Sistem menampilkan menu about
Kondisi Akhr	Menu about berhasil ditampilkan oleh sistem

e. Use Case Scenario Menampilkan Petunjuk Permainan

Use Case Scenario dari *Use Case* Menampilkan Petunjuk Permainan dijelaskan pada Tabel 3-13.

Tabel 3-13 Use Case Scenario Menampilkan Petunjuk Permainan

Identifikasi	
Nomor	5
Nama	Menampilkan Petunjuk Permainan
Tujuan	Proses untuk menampilkan menu petunjuk permainan
Deskripsi	Proses untuk melihat informasi petunjuk cara bermain
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di menu utama
Aksi Aktor	Reaksi Sistem
1.Pemain memilih menu petunjuk permainan	
	2.Sistem merespon dengan cara memanggil dan milihat petunjuk
Kondisi Akhir	Menu petunjuk permainan berhasil ditampilkan oleh sistem

f. Use Case Scenario Menunda Permainan

Use Case Scenario dari *Use Case* Mengendalikan Karakter dijelaskan pada Tabel 3-14.

Tabel 3-14 Use Case Scenario Menunda Permainan

Identifikasi	
Nomor	6
Nama	Menunda Permainan
Tujuan	Untuk menunda permainan
Deskripsi	Proses untuk menunda permainan ketika *game* dimainkan
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada arena permainan
Aksi Aktor	Reaksi Sistem
1.Pemain menekan timbol pause	
	2.Sistem menghentikan permainan
Kondisi Akhir	Permainan dihentikan

g. Use Case Scenario Keluar Game

Use Case Scenario dari *Use Case* keluar *game* dijelaskan pada Tabel 3-15.

Tabel 3-15 Use Case Scenario Keluar Game

Identifikasi	
Nomor	7
Nama	Keluar Game
Tujuan	Untuk keluar dari game
Deskripsi	Proses untuk keluar dari permainan
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada arena permainan
Aksi Aktor	Reaksi Sistem
1.Pemain menekan tombol keluar	
	2.Sistem akan keluar dari game
Kondisi Akhir	Permainan keluar dari game

h. Use Case Mendapatkkan Skor

Use Case Scenario dari *Use Case* Mendapatkan Skor dijelaskan

Identifikasi	
Nomor	8
Nama	Mendapatkan Skor

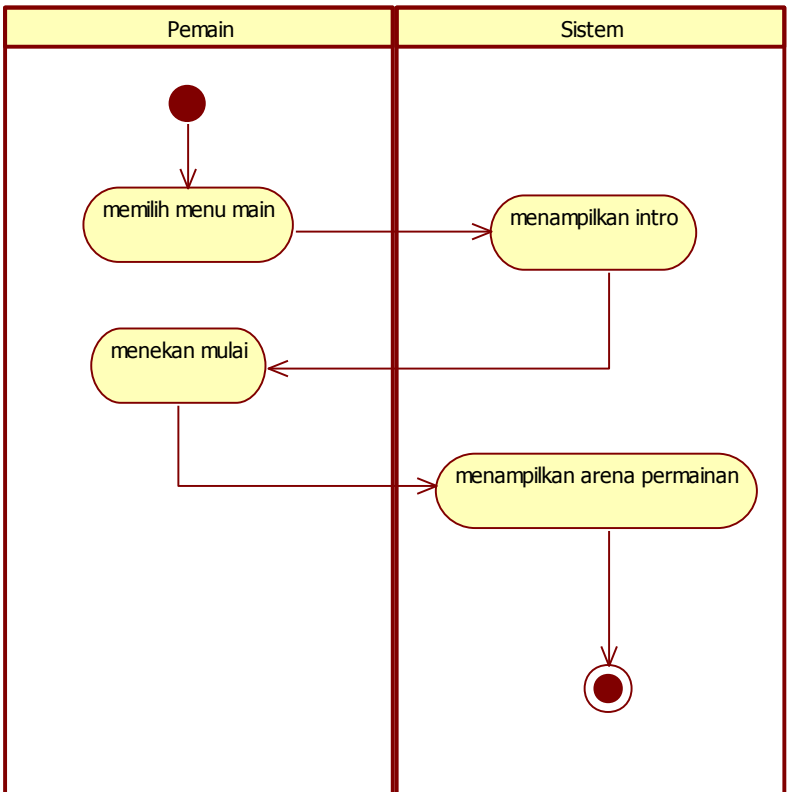
Tujuan	Untuk keluar dari game
Deskripsi	Proses untuk mendapatkan skor apabila pemain telah mendapatkan karung rumput
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada arena permainan
Aksi Aktor	Reaksi Sistem
1.Pemain mendapatkan karung rumput	
	2.Sistem akan menghitung skor berdasarkan banyaknya karung rumput yang didapatkan
Kondisi Akhir	Permainan mengetahui skor yang didapat

3.1.6.2 Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Berikut ini *activity diagram* pada aplikasi *game raceKarapan Sapi.*

1. Activity Diagram Mulai Game

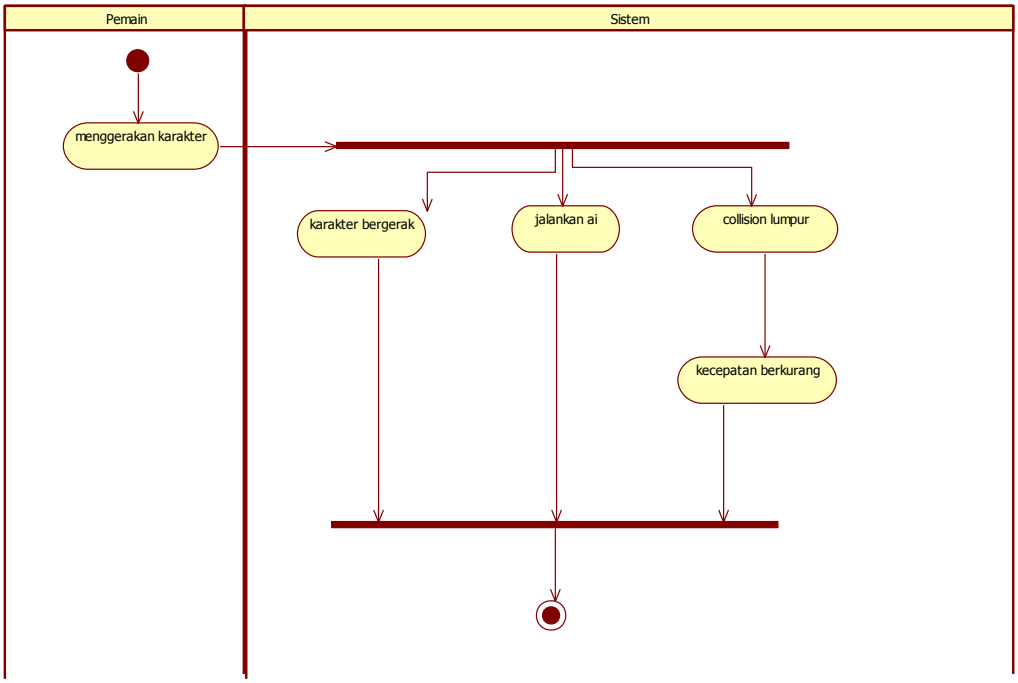
   Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas mulai game yang dapat dilihat pada Gambar 3-10 .



Gambar 3-10 Activity Diagram Mulai Game

2. *Activity Diagram* Menghindari Rintangan

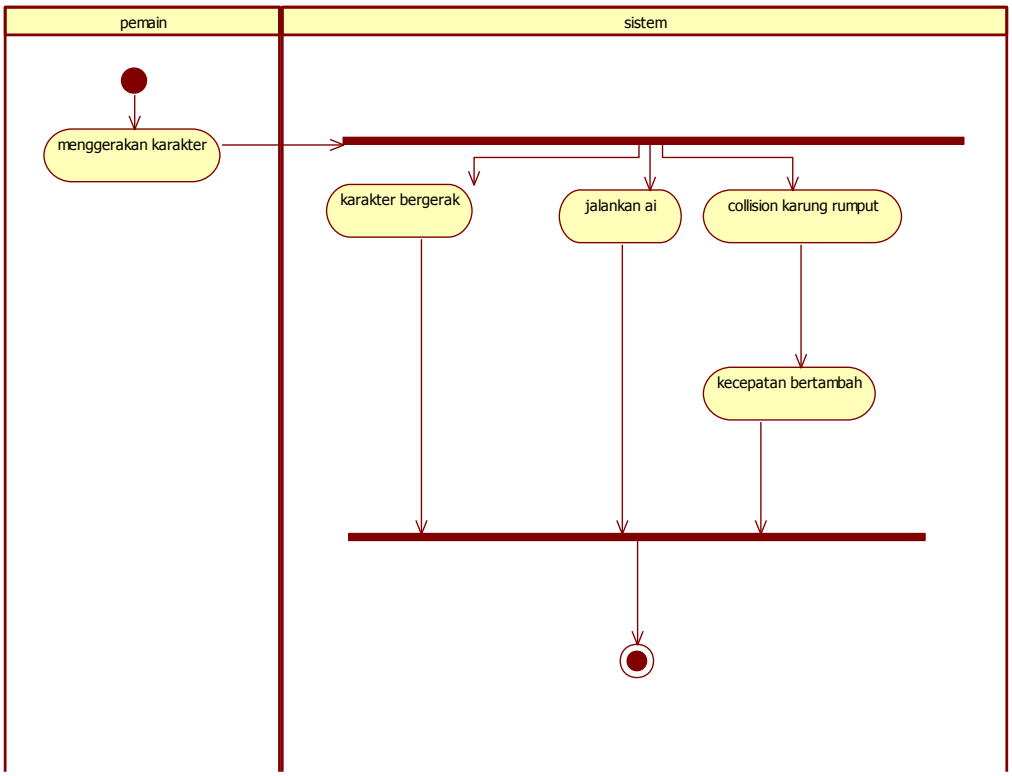
   Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas menghindari rintangan yangdapat dilihat pada Gambar 3-11



Gambar 3-11 Activity Diagram Menghindari Rintangan

3. Activity Diagram Mendapatkan Rumput

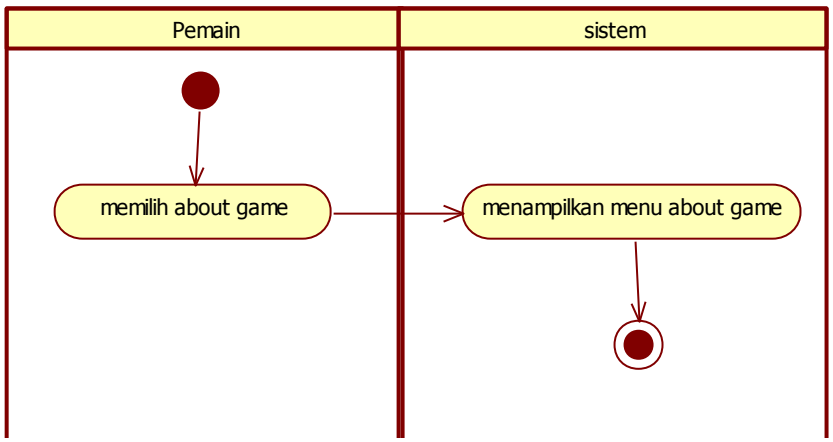
   Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas mendapatkan rumput yang dapat dilihat pada Gambar 3-12.



Gambar 3-12 Activity Diagram Mendapatkan Rumput

4. *Activity Diagram* Menampilkan *About Game*

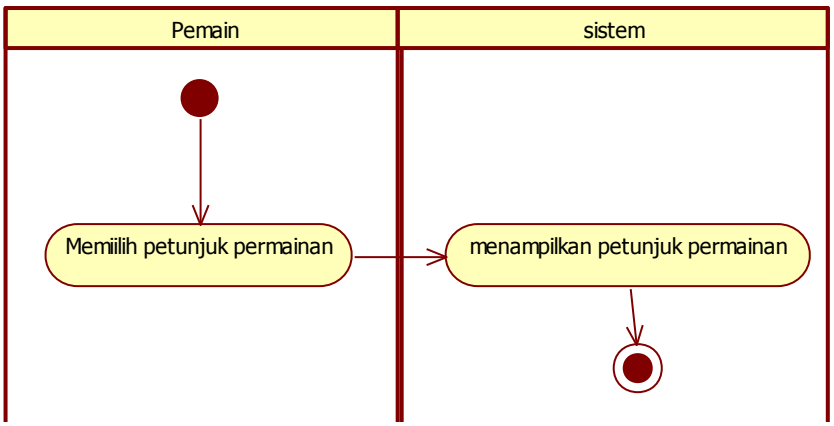
   Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas menampilkan *about game* yang dapat dilihat pada Gambar 3-13



Gambar 3-13 Activity Diagram Menampilkan about game

5. Activity Diagram Menampilkan Petunjuk Permainan

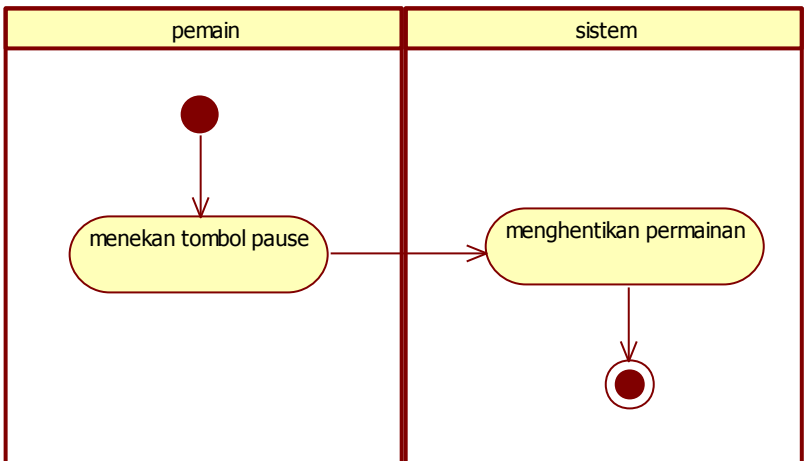
   Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas menampilkan petunjuk permainan yang dapat dilihat pada Gambar 3-14



Gambar 3-14 Activity Diagram Menampilkan Petunjuk Permainan

6. Activity Diagram Menunda Permainan

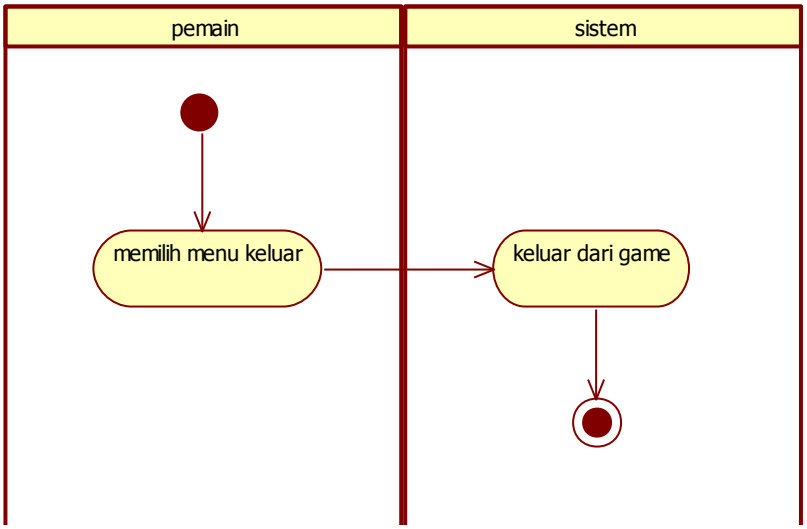
   Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas menunda permainan yangdapat dilihat pada Gambar 3-15



Gambar 3-15 Activity Diagram Menunda Permainan

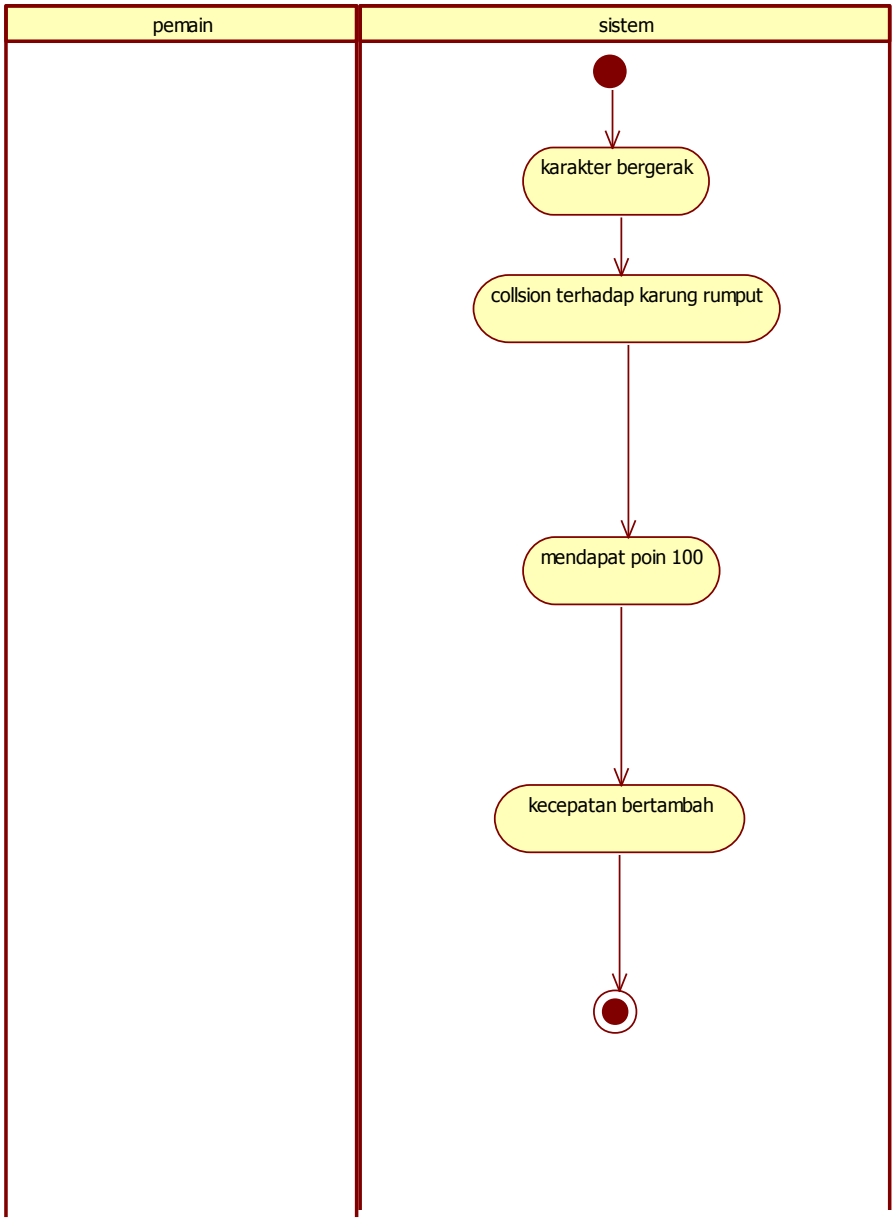
7. Activity Diagram Keluar Game

   Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas keluar dari game yang dapat dilihat pada Gambar 3-16.



Gambar 3-16 Activity Diagram Keluar Game

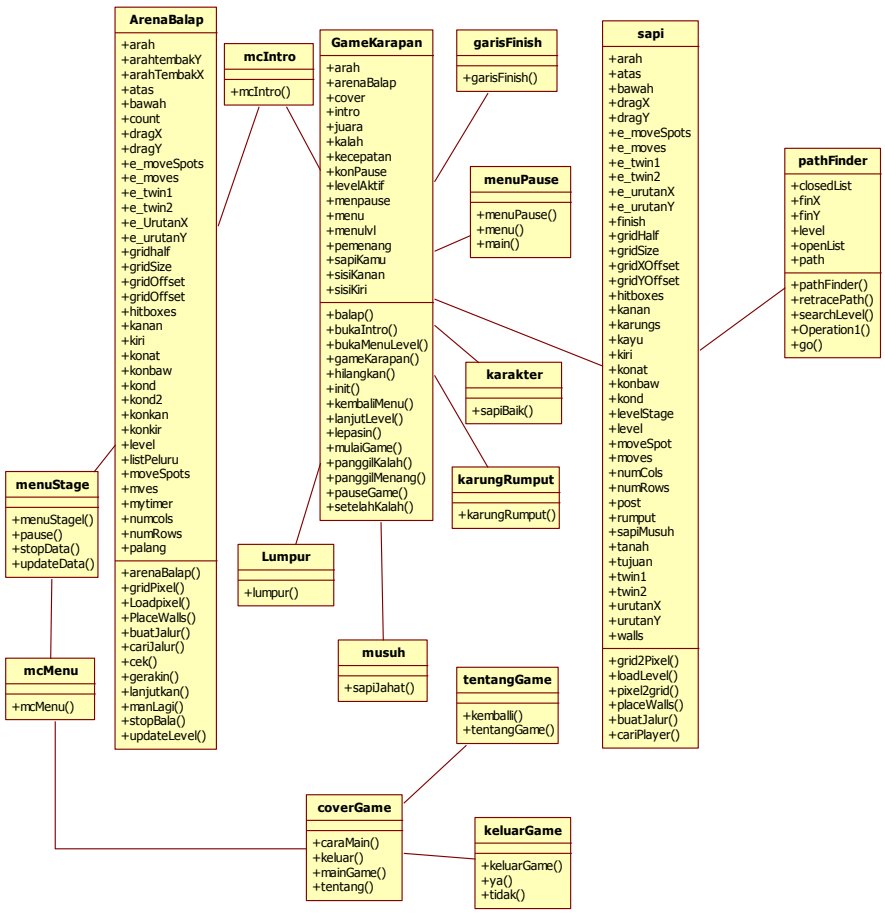
8. Activity Diagram Mendapatkan Skor
   Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas mendapatkan skor yang dapat dilihat pada



Gambar 3-17 Activity Diagram Mendapatkan Skor

3.1.6.3 Class Diagram

Class Diagram menggambarkan struktur dan hubungan antar objek-objek yang ada pada sistem.Struktur itu meliputi atribut dan *method* yang ada pada masing-masing *class*.Hubungannya dari masing-masing *class* yang ada tersebutdigambarkan dengan menggunakan pewarisan dan generalisasi seperti terlihat pada Gambar 3-18.

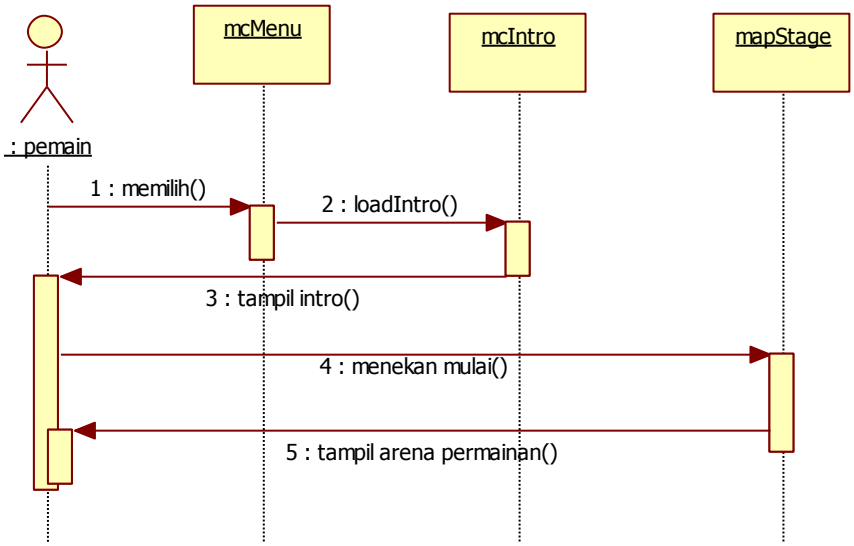


Gambar 3-18 Class Diagram Game Karapan Sapi

3.1.6.4 Sequence Diagram

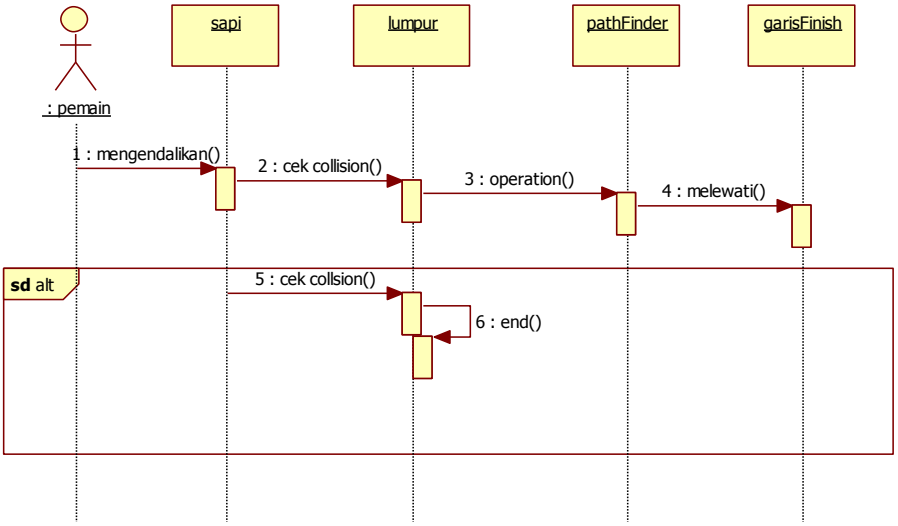
Sequence Diagram menggambarkan interaksi antar masing-masing objek pada setiap *use case* dalam urutan waktu. Interaksi ini berupa pengiriman serangkaian data antar objek-objek yang saling berinteraksi.*Sequence Diagram* pada *game race* Karapan Sapiterdiri dari *sequence diagram* menampilkan *score,sequence diagram* petunjuk permainan,dan*sequence diagram* menampilkan *About.*Berikut ini penjelasan dari masing-masing sequence diagram:

1.Sequence Diagram Mulai Game



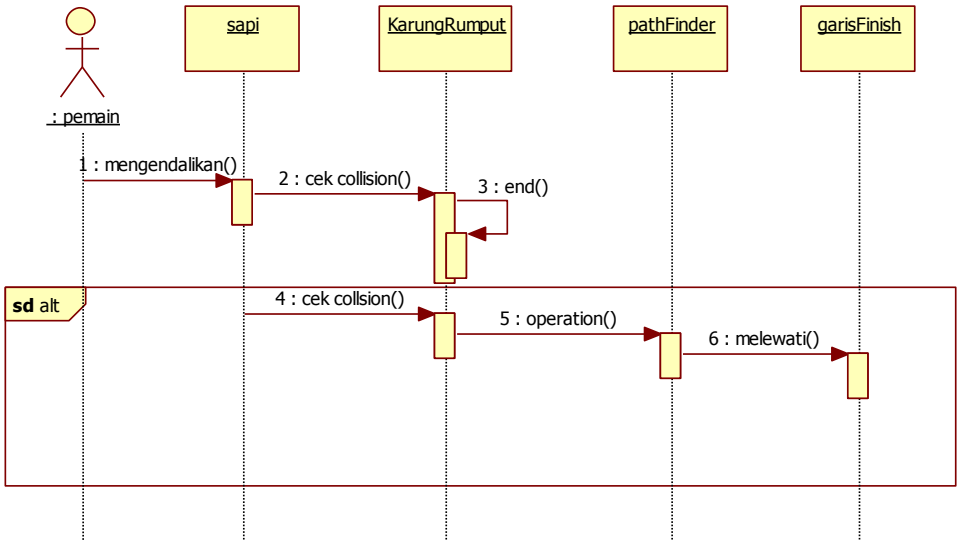
Gambar 3-19 Sequence Diagram Mulai Game

2.Sequence Diagram Menghindari Rintangan



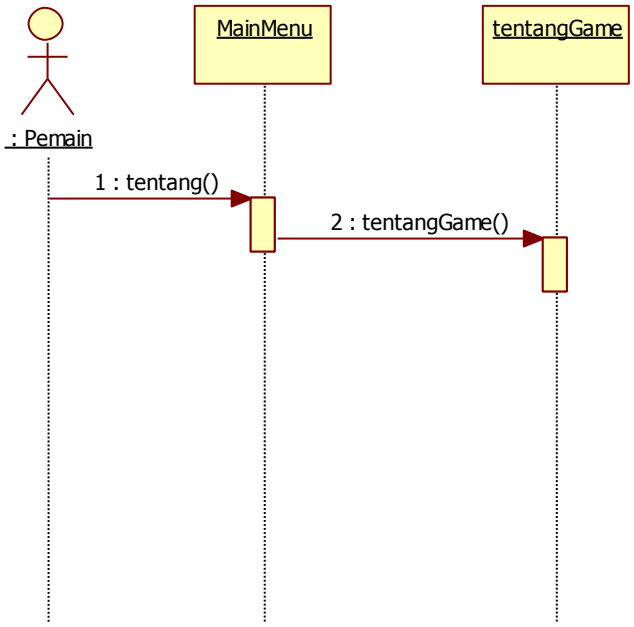
Gambar 3-20 Sequence Diagram Menghindari Rintangan

3.Sequence Diagram Mendapatkan Rumput



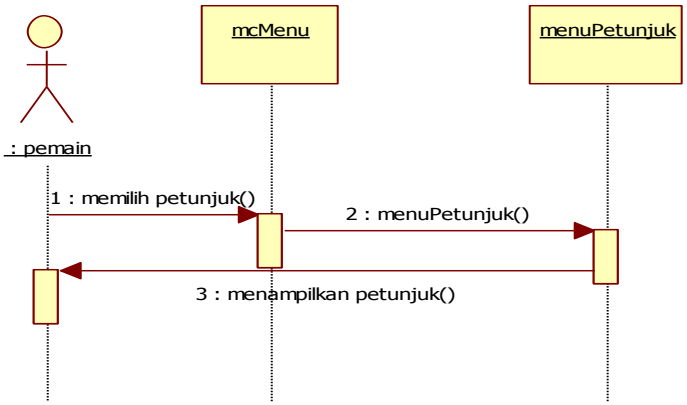
Gambar 3-21 Sequence Diagram Mendapatkan Rumput

4.Sequence Diagram Menampilkan *About*



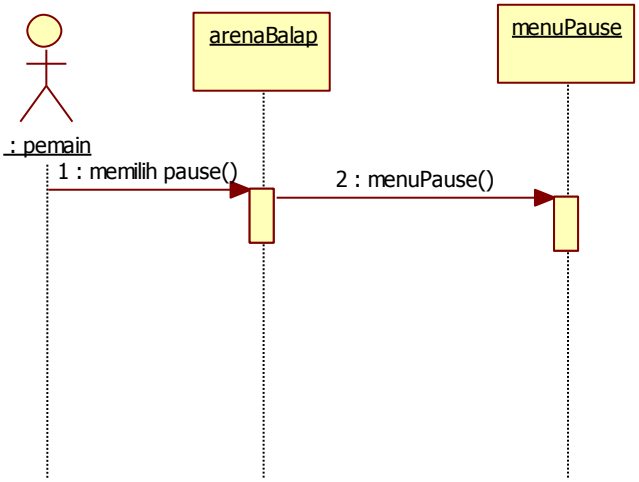
Gambar 3-22 Sequence Diagram Menampilkan About

5.Sequence Diagram Menampilkan Petunjuk Permainan



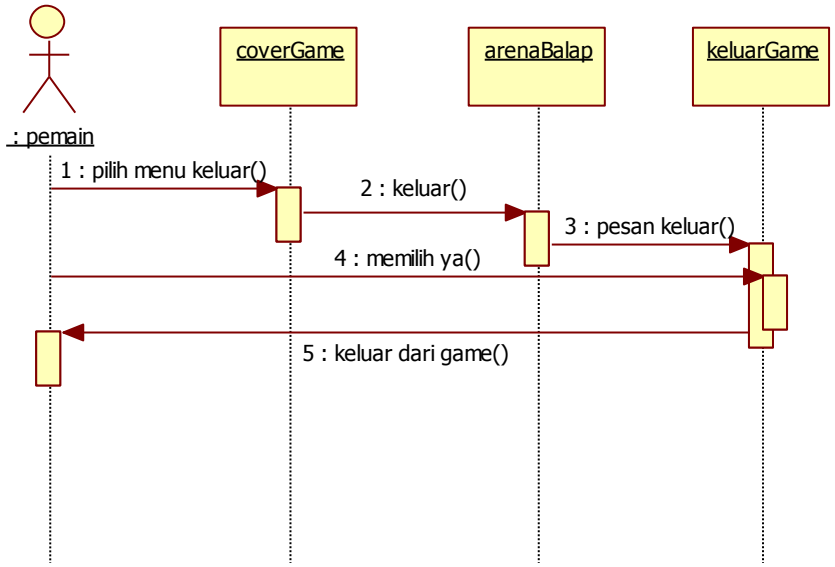
Gambar 3-23 Sequence Diagram Menampilkan Petunjuk Permainan

6. Sequence Diagram Menunda Permainan



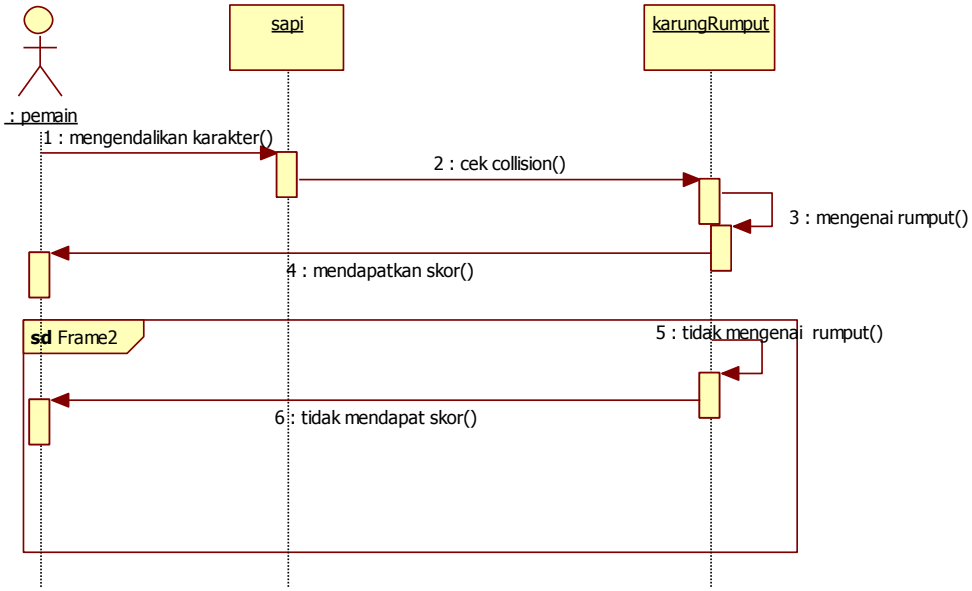
Gambar 3-24 Sequence Diagram Menunda Permainan

7.Sequence Diagram Keluar Game



Gambar 3-25 Sequence Diagram Keluar Game

8.Sequence Diagram Mendapatkan Skor



Gambar 3-26 Sequence Diagram Mendapatkan Skor

3.2 Perancangan Sistem

Perancangan sistem merupakan kegiatan untuk merancang aplikasi yang akan di bangun, tahapannya dimulai dari perancangan komponen permainan,

perancangan antarmuka, perancangan pesan dan jaringan semantik dari *game race* Karapan Sapi

3.2.1 Perancangan Kompnen Permainan

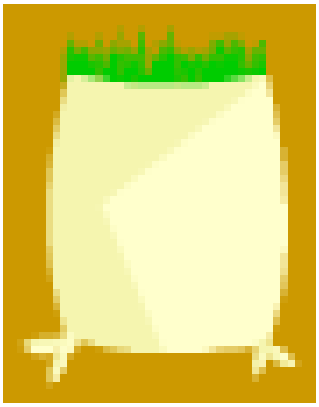
Di dalam bagian ini akan dijelaskan item yang terdapat pada *gamerace* Karapan Sapi

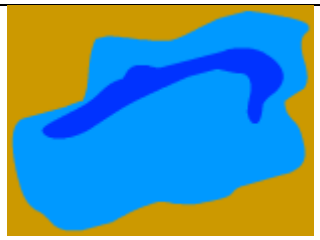
3.2.1.1 Item Game

Didalam bagian ini akan dijelaskan karakter karakter yang terdapat pada game race karapan sapi yang bisa dilihat pada Tabel 3-16.

Tabel 3-16 Tabel Item

No	Item	Keterangan
1		NamaKarakter : Slamet , merupakan karakter utama didalam game ini.

2		Nama Item : Karung rumput , jika karakter mendapatakan item tersebut maka kecepatan sapi akan bertambah.
3		Nama item : Lumpur , jika karakter mengenai lumpur maka kecepatan peman akan berkurang
4		Nama item : Sapi penghalang , jika karakter bertabrakan dengan dengan sapi penghalang maka karakter akan terhalang lajunya

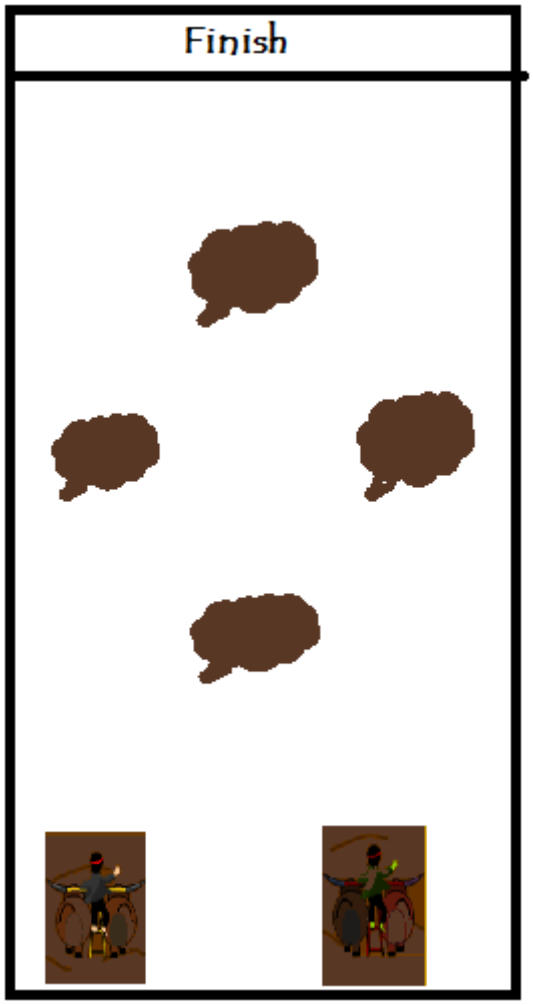
5		Nama item : Genagan air jika karater mengenai genangan air maka kecepstsn akan berkurang

3.2.1.2 *Storyboard*

Storyboard adalah sketsa gambar yang disusun berurutan sesuai dengan naskah, dengan storyboard ide cerita dapat disampaikan kepada orang lain dengan lebih mudah, karena dapat menggiring khayalan seseorang mengikuti gambar-gambar yang tersaji sehingga menghasilkan persepsi yang sama. Berikut merupakan story board dari *game race* Karapan Sapi

1. *Storyboard Level* 1

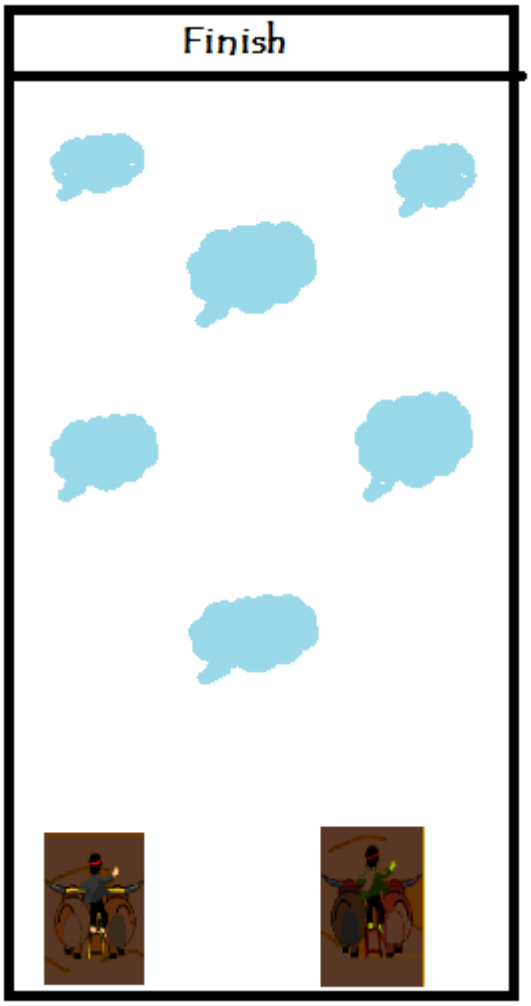
Pada level 1 rintangan yang harus dilewati adalah genangan lumpur yang ditunjukan pada Gambar 3-27 Storyboard Level 1.



Gambar 3-27 Storyboard Level 1

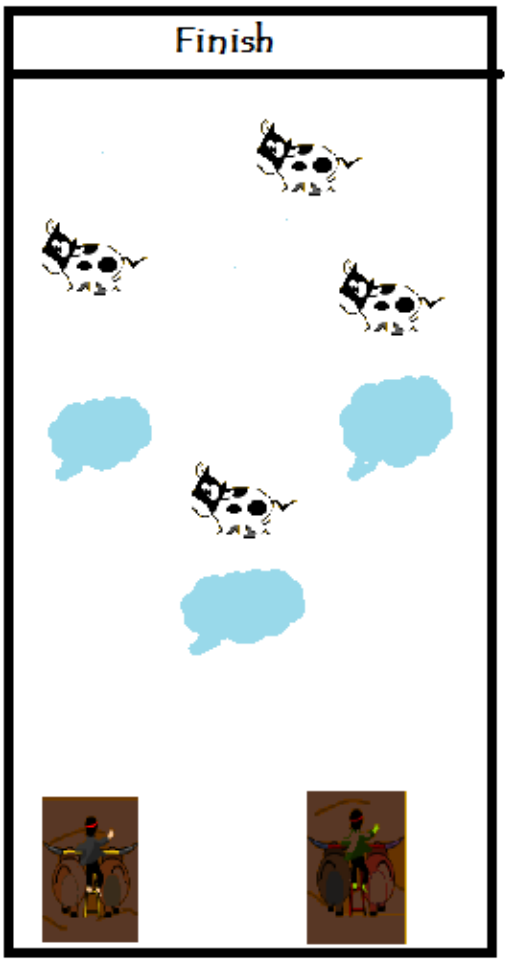
2. *Storyboard Level* 2

   Pada level 2 rintangan yang harus dilewati adalah genangan air yang ditunjukan pada Gambar 3-28.



Gambar 3-28 Storyboard Level 2

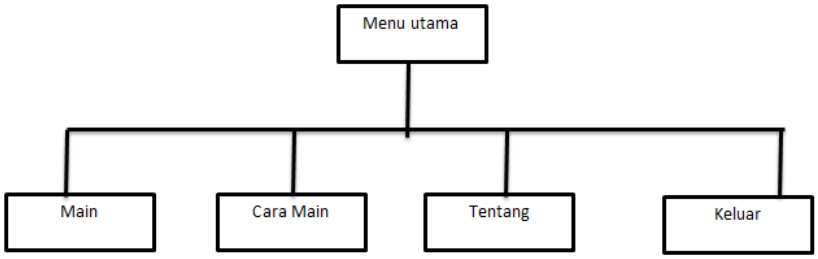
3. *Storyboard Level* 3
   Pada level 3 rintangan yang harus dilewati adalah genangan air dan sapi liar yang ditunjukan pada Gambar 3-29



Gambar 3-29 Storyboard Level 3

3.2.2 Perancangan Struktur Menu

Perancangan struktur menu pada *game race* Karapan Sapi pada gambar berikut :



Gambar 3-30 Struktur Menu Game Karapan Sapi

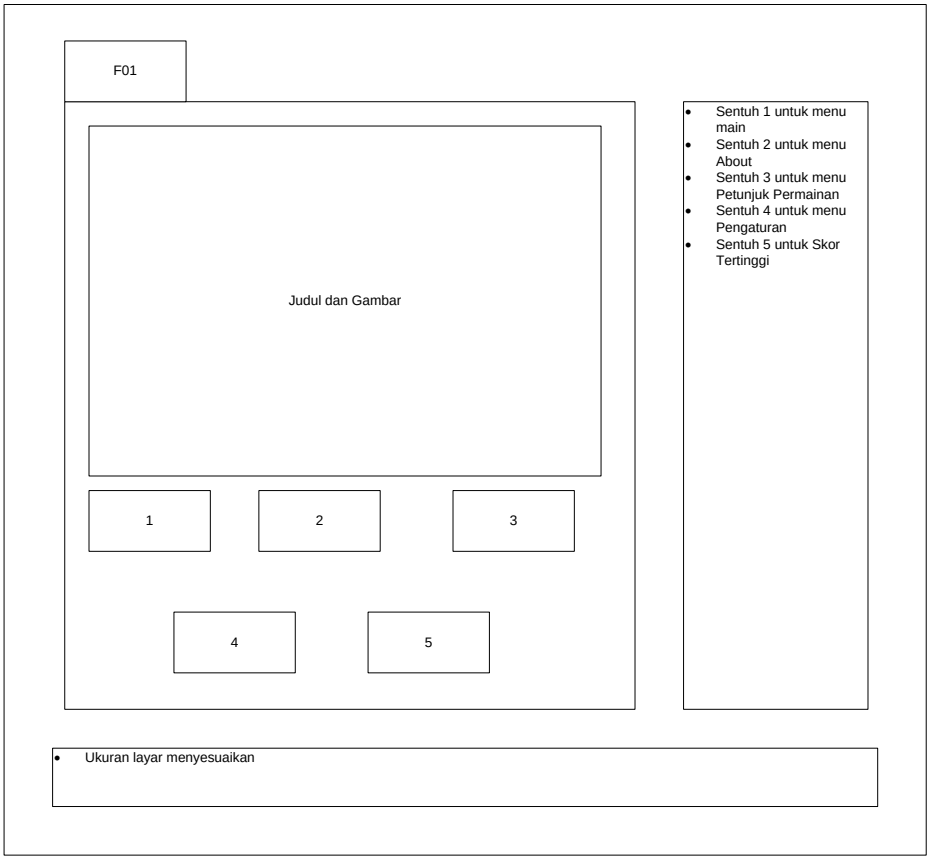
3.2.3 Perancangan Antar Muka

Perancangan antarmuka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun, sehingga akan mempermudah dalam mengimplementasikan aplikasi serta akan memudahkan pembuatan aplikasi.

Pada bagian ini akan dijelaskan perancangan antarmuka menu utama, perancangan antarmuka *menu play*, perancangan antarmuka menu *highscores*, perancangan antarmuka menu *option*, perancangan antarmuka menu *help*, perancangan antarmuka menu *About* dan perancangan antarmuka menu *exit.*

3.2.3.1 Perancangan Antarmuka Menu Utama

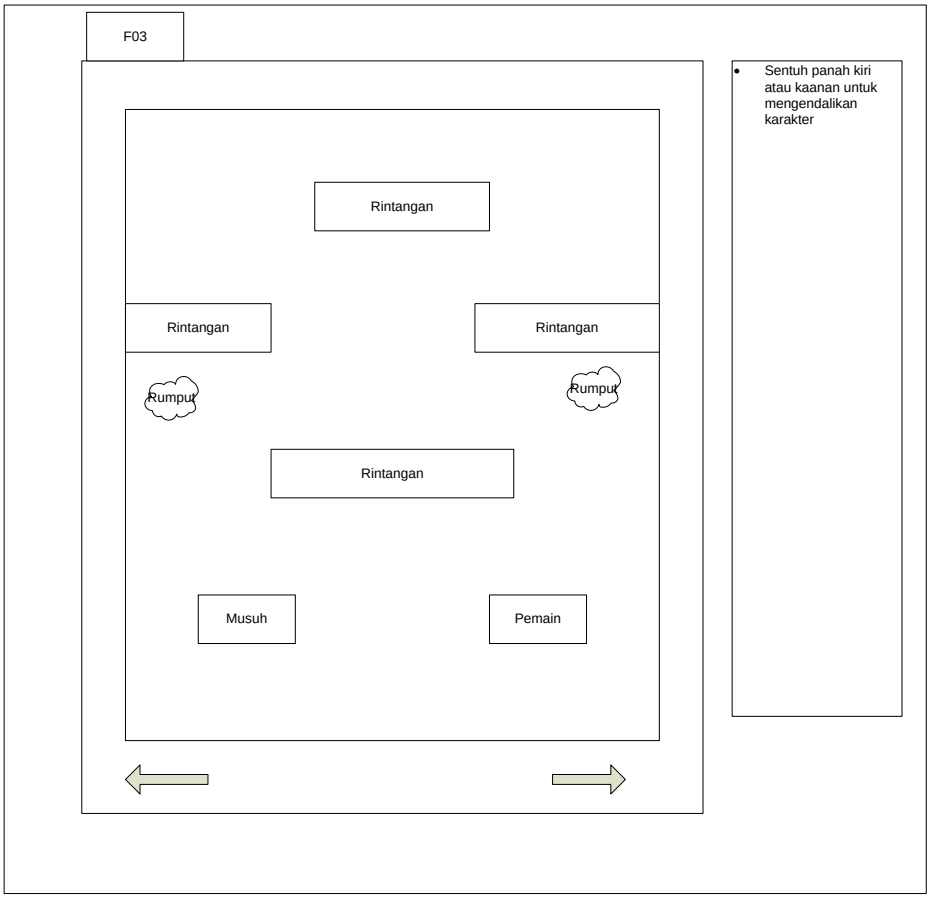
Menu utama merupakan menu yang akan muncul pertama kali pada saat pengguna membuka *game*. Gambar 3-31 menunjukan perancangan antarmuka menu utama.



Gambar 3-31 Perancangan Antarmuka Menu Utama

3.2.3.2 Perancangan Antarmuka Menu Main

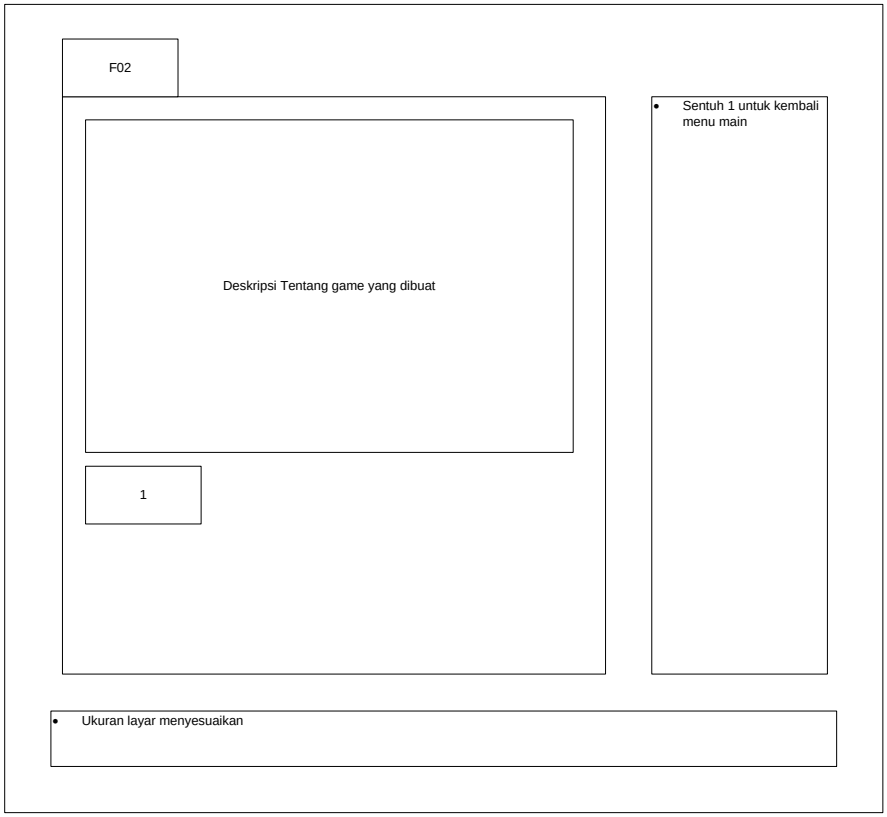
Antarmuka menu main merupakan perancangan antarmuka yang menampilkan permainan game *race* karapan sapi.Gambar 3-32 merupakan perancangan antarmuka menu main.



Gambar 3-32 Perancangan Antarmuka Menu Main

3.2.3.3 Perancagan Antarmuka Menu Cara Main

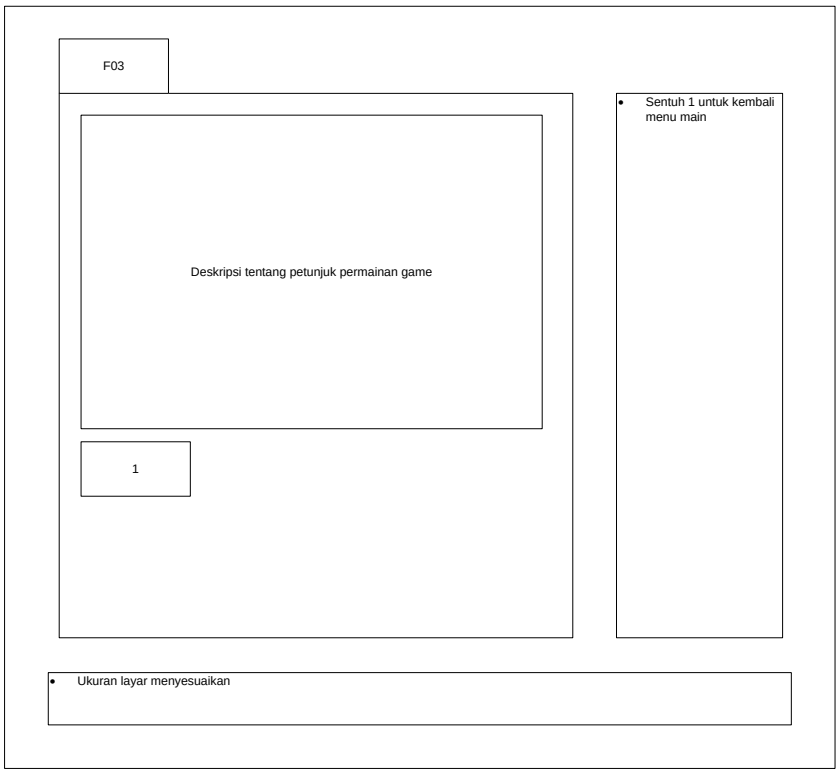
Menu *About* merupakan menu yang menampilkan profil penulis.Gambar 3-33 menunjukan perancangan antarmuka Menu *About.*



Gambar 3-33 Perancangan Antarmuka Menu About

3.2.3.4 Perancangan Antarmuka Menu Tentang

Menu petunjuk permainanmerupakan menu yang menampilkan petunjuk cara bermain game Karapan Sapi .Gambar 3-34 menunjukan perancangan antarmuka Menu Petunjuk Permainan.



Gambar 3-34 Perancangan Antarmuka Menu Petunjuk Permainan

3.2.3.5 Perancangan Antarmuka Menu Keluar

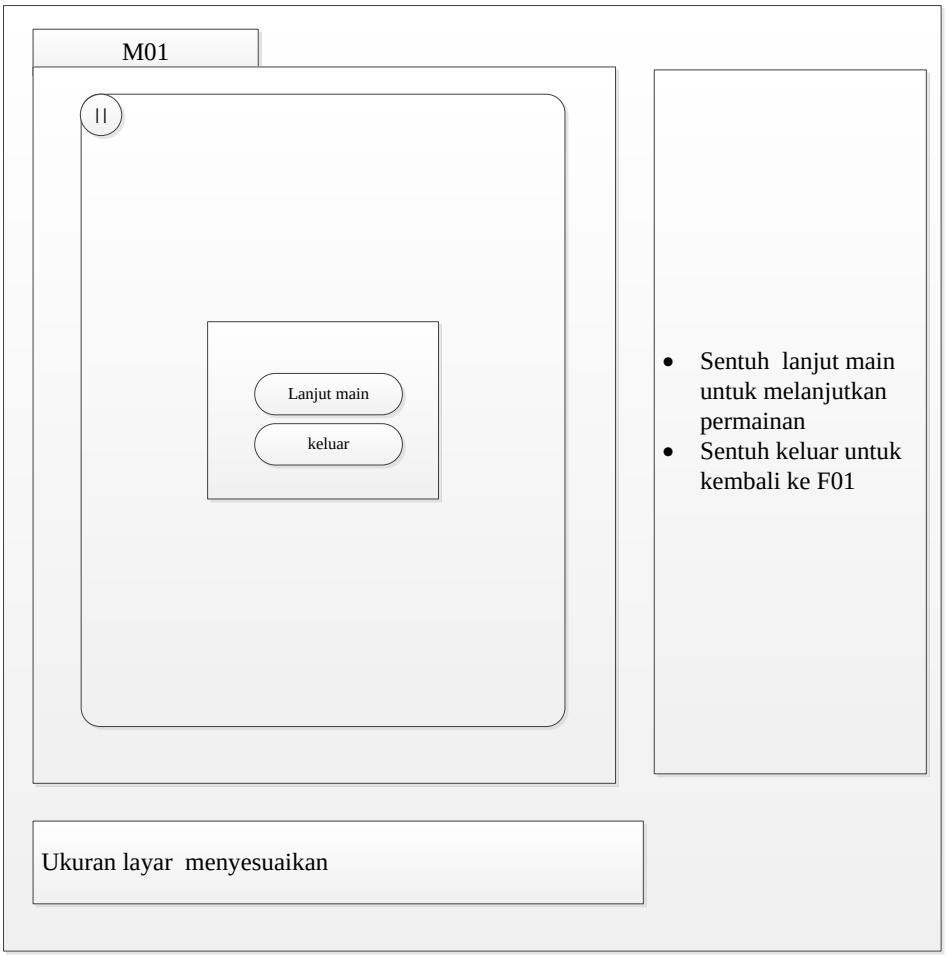
Menu pengaturan merupakan menu yang menampilkan pilihan sapi dan lintasan yang bisa dipilih untuk bermain.Gambar dibawah ini menunjukan perancangan antarmuka menu pengaturan

3.2.4 Perancangan Pesan

Pada bagian ini akan dijelaskan perancangan pesan yang terdapat pada game *race*Karapan Sapi yaitu perancangan pesan *pause menu*dan perancangan pesan perlengkapan.

3.2.4.1 Perancangan Pesan Pause

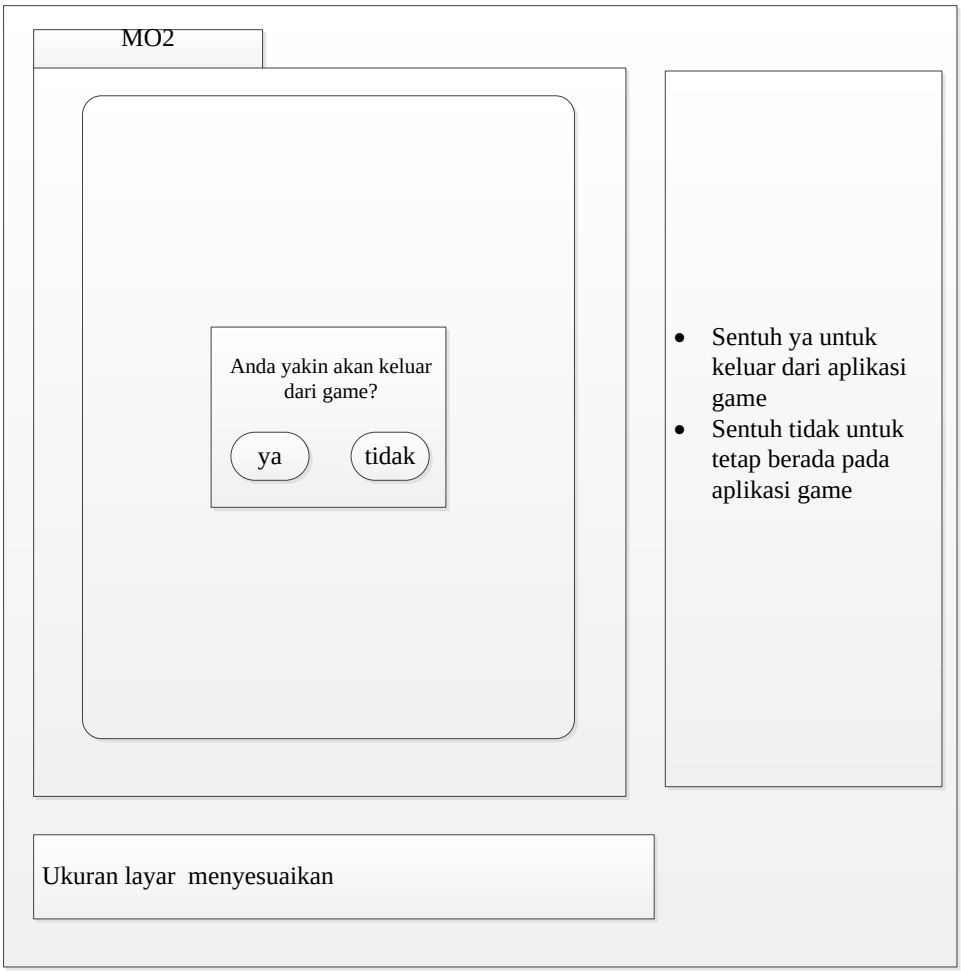
Pesan pause menu merupakan pesan yang muncul ketika pemain menampilkan pause menu.Untuk lebih jelasnya dapat dilihat pada Gambar 3-35.



Gambar 3-35 Perancangan Pesan Pause

3.2.4.2 Perancangan Pesan Exit

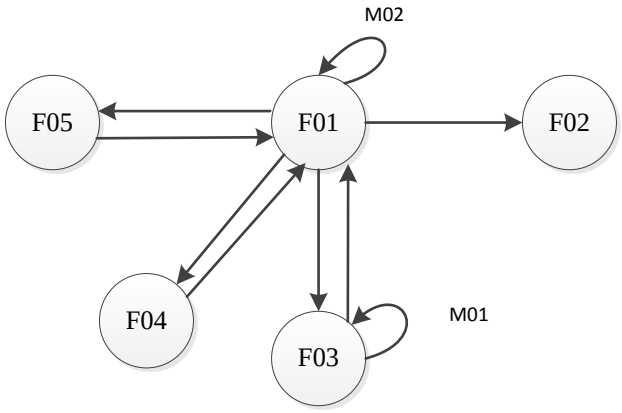
Pesan perlengkapan merupakan pesan yang muncul ketika pengguna akan keluar dari aplikasi game. Untuk lebih jelasnya dapat dilihat pada Gambar 3-36



Gambar 3-36 Perancangan Pesan Exit

3.2.5 Jaringan Semantik

Untuk mengetahui keterhubungan dari setiap tampilan yang dibuat dapat menggunakan jaringan semantik.Gambar 3-37 adalah jaringan semantik pada *game race* Karapan Sapi.



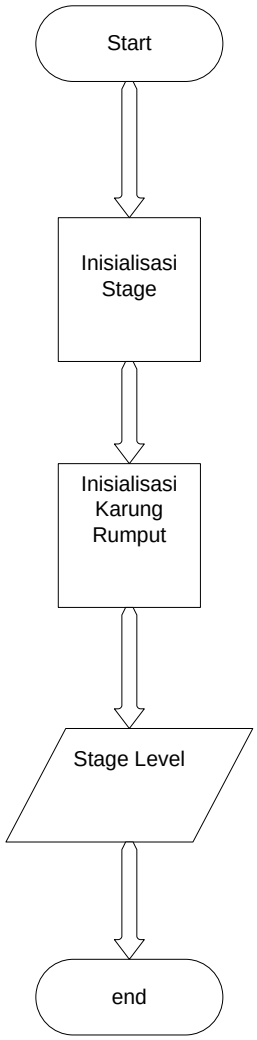
Gambar 3-37 Jaringan Semantik

3.2.6 Perancangan Metode

Perancangan metode merupakan perancangan yang dilakukan untuk menetapkan detail algoritma yang akan dinyatakan ke dalam suatu program. Perancangan metode menjelaskan tentang bagaimana program aplikasi berjalan.Perancangan metode yang ada pada game karapan sapi ini yaitu Method FindPath ,Method load level,Method Ser Moving dan Method Check Collison.

1. ***Method load level***

   Method ini digunakan untuk meload tage dari setiap level game.

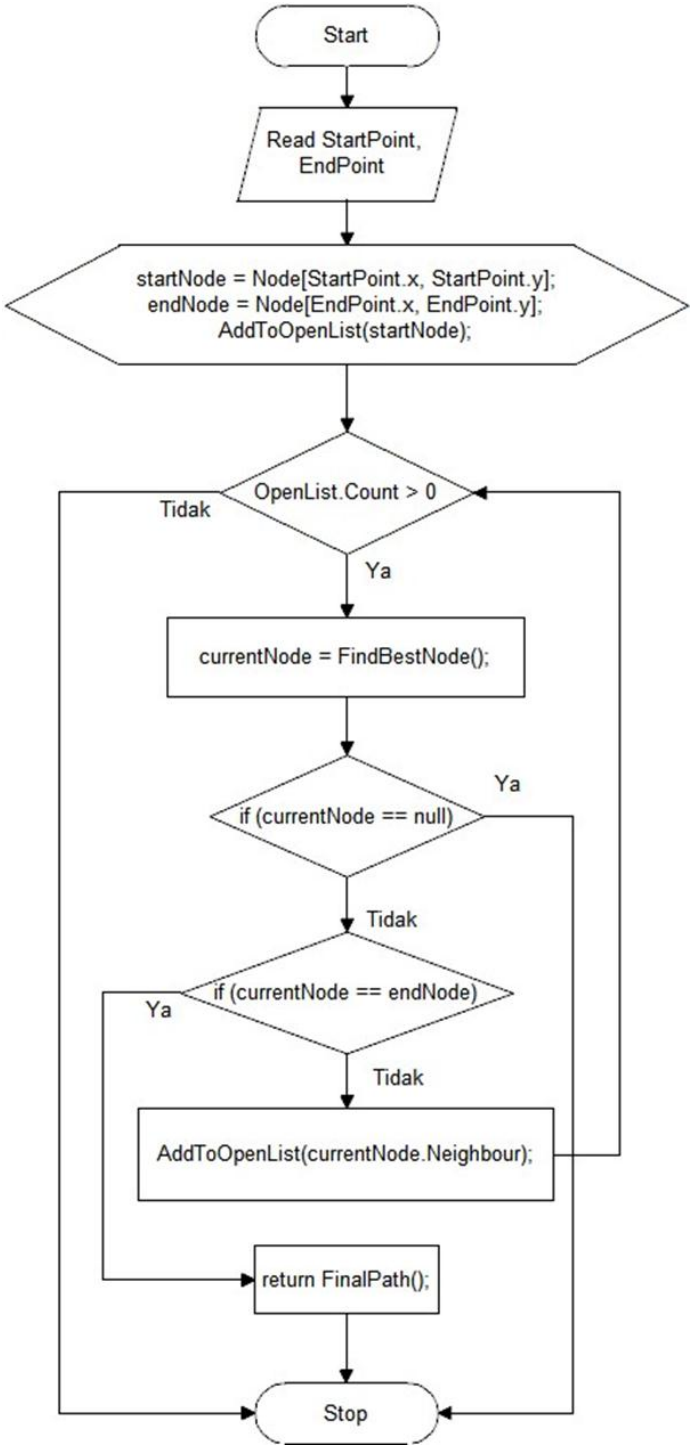


.

Gambar 3-38 Method load level

2. Method FindPath

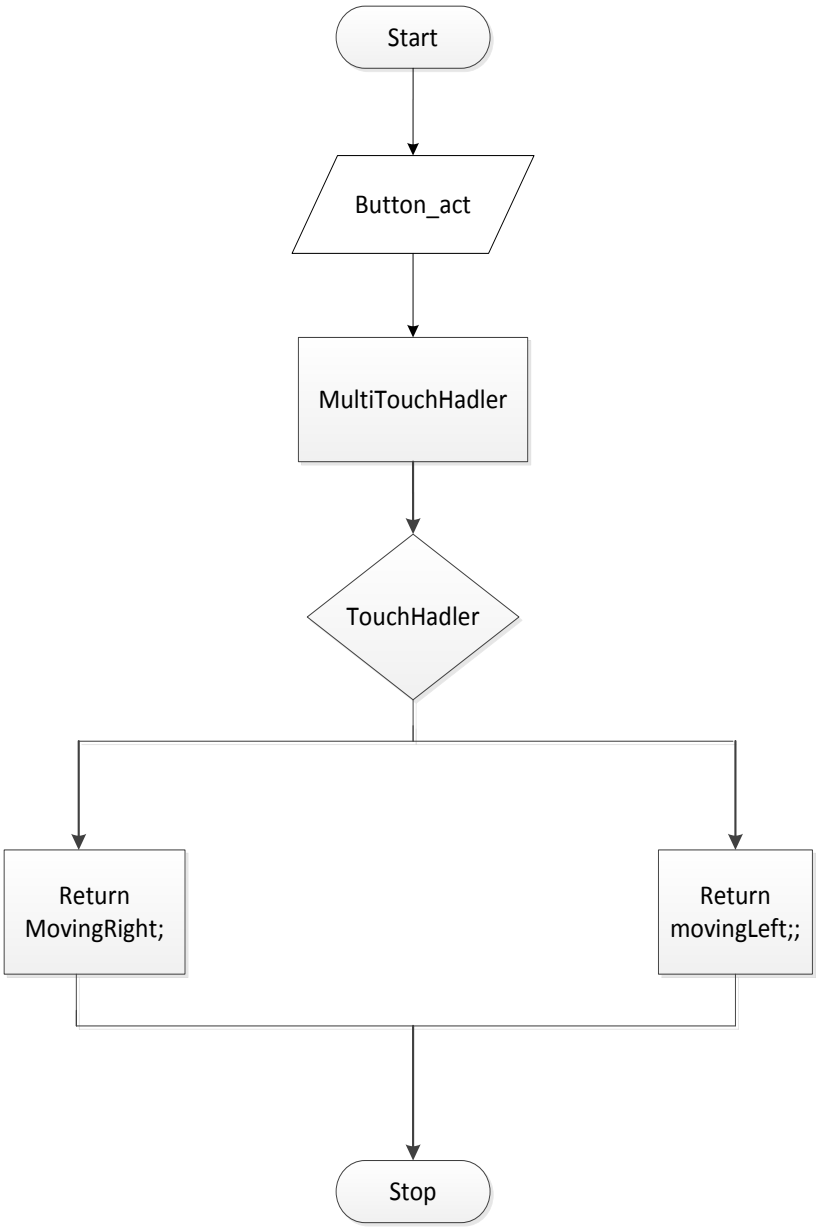
Method ini digunakan NPC untuk mencari garis finish.



Gambar 3-39 ***Method FindPath***

3. Method Set Moving

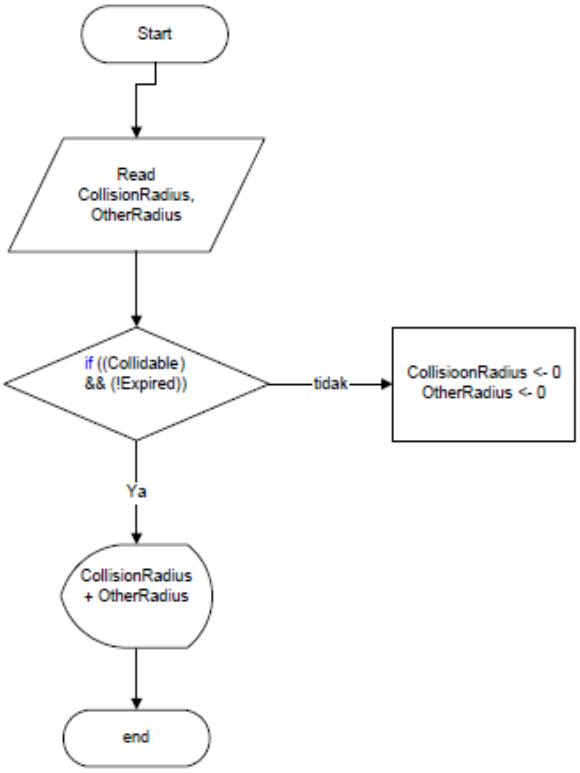
Method ini digunakan untuk membaca input dari pengguna, input tesebut berupa tombol yang di tekan pada touchscreen. Peracangan *method* ini bisa dilihat pada Gambar 3-40 Method Set Moving



Gambar 3-40 Method Set Moving

4. Method Check Collision

Method ini akan melakukan pengecekan terhadap suatu objek apabila ada kedua objek yang saling bertabrakan. Perancangan method check collision bisa dilihat pada Gambar 3-41 Method Check Collision



Gambar 3-41 Method Check Collision