BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Pada analisis sistem akan mencakup analisis masalah, pengenalan dan analisis *game* sejenis, analisis kebutuhan non-fungsional, dan analisis kebutuhan fungsional.

3.1.1 Analisis Masalah

Penyakit kanker pada anak dapat disebabkan oleh lingkungan yang tidak bersih, asap rokok, paparan bahan kimia yang bersifat karsinogen, makanan yang mengandung bahan pengawet, pewarna, dan pemanis buatan, dan lain-lain. Kurangnya pengetahuan mengenai kanker pada anak dapat menyebabkan anak-anak melakukan hal-hal yang dapat menyebabkan kanker, diantaranya berada pada lingkungan yang tidak sehat dan penuh polusi, serta memakan makanan yang tidak sehat, tidak bergizi dan tidak higienis.

Menurut laporan yang dikeluarkan International Data Corporation (IDC), Indonesia merupakan salah satu negara dengan tingkat pertumbuhan pengguna perangkat *mobile* paling tinggi di dunia. IDC mencatat, pada kuartal IV 2012, penjualan perangkat *mobile* di Indonesia sudah mencapai 1,6 juta unit. Dan diperkirakan pada tahun 2013 akan tumbuh hingga mencapai 2 juta unit.

Berdasarkan hasil survey yang dilakukan Nielsen terhadap pemilik perangkat *mobile* pada orang dewasa yang mempunyai anak usia di bawah 12 tahun, diketahui tujuh dari sepuluh orang dewasa yang memiliki perangkat *mobile*, digunakan juga oleh anak-anak mereka. Dari hasil survey itu juga diketahui bahwa anak-anak menggunakan perangkat *mobile* untuk bermain *game*s online sebanyak 77 persen, sebagai hiburan dalam perjalanan (55%), untuk menonton TV atau film (43%), dan untuk sarana pendukung pendidikan sebanyak 57 persen. [17]

Penggunaan perangkat *mobile* sebagai alternatif media pengajaran anak secara modern makin disukai. Karena perangkat *mobile* memiliki layar besar sebagai media visual cukup nyaman, dan juga sudah banyak tersedia perangkat lunak pendidikan. Maka, alangkah baiknya bila perangkat *mobile* yang sering digunakan oleh anak-anak ini, dimanfaatkan sebagai sarana edukasi, seperti *game* edukasi mengenai penyakit kanker

3.1.2 Pengenalan dan Analisis *Game* Sejenis

Game sejenis yang akan penulis gunakan sebagai referensi dalam pembangunan *game* melawan kanker ini antara lain adalah *game* Catch The Eggs, Memory *Game*, dan Cartoon Quiz *Game*.

3.1.2.1 *Game* Catch The Eggs

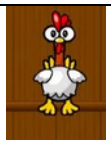
Dalam pembahasan, penulis akan membahas *game* dengan jenis *gameplay* casual dan *game* edukasi. Salah satu *game* yang akan dibahas sebagai acuan dalam pembangunan *game* melawan kanker yang akan dibangun adalah *game* Catch The Eggs. Oleh karena itu penulis menggabungkan tipe *gameplay* casual dengan menerapkan unsur edukasi mengenai penyakit kanker untuk anak-anak. Berikut ini merupakan ulasan detail mengenai *game* catch eggs.

a. Konten-Konten Pada *Game* Catch The Eggs

Berikut adalah konten-konten yang terdapat pada permainan *game* Catch The Eggs dapat dilihat pada tabel 3.1:

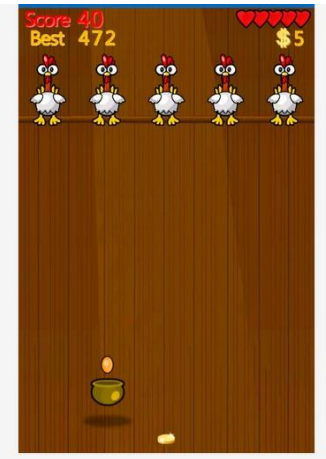
Tabel 3.1 Konten-Konten Pada *Game* Cath The Eggs

No	Nama	Gambar	Penjelasan
1	Keranjang		Keranjang merupakan karakter yang dikendalikan oleh pemain, pemain dapat menggeserkan keranjang agar posisinya tepat dengan telur yang akan ditangkap

2	Ayam		Ayam merupakan karakter yang berfungsi untuk mengeluarkan telur yang akan ditangkap oleh pemain
3	Telur Biasa		Menambahkan 1 skor bila berhasil ditangkap
4	Telur Kuning		Menambahkan 2 skor bila berhasil ditangkap
5	Telur Emas		Menambahkan 5 skor apabila berhasil ditangkap, dan juga menambahkan bonus koin.
6	Telur Granat		Tidak menambah skor, tetapi apabila tidak berhasil ditangkap, dan kemudian jatuh dan pecah, maka permainan akan langsung selesai.
7	Nyawa		Apabila pemain tidak berhasil menangkap telur, maka nyawa akan berkurang satu. Dan hanya dibatasi 5 nyawa saja.
8	Skor		Skor pada *game* ini merupakan hasil penjumlahan dari banyaknya telur yang berhasil ditangkap.

b. Tampilan *Gameplay* pada *Game* Catch The Eggs

Tampilan *game* merupakan interface yang akan membeuat pengguna merasa tertarik dalam memainkannya. Berikut adalah tampilan *gameplay* permainan Catch The Eggs



Gambar 3.1 Tampilan *Gameplay* Catch The Eggs

c. Skenario Permainan

Skenario permainan pada *Game* Catch The Eggs dapat disusun oleh beberapa hal, antara lain:

1. Alur Sistem

Sistem akan menginisialisasi seluruh entitas yang diperlukan untuk memulai permainan, seperti karakter pemain yang berupa keranjang, ayam, inisialisasi skor awal yang masih bernilai nol, dan inisialisasi nyawa awal yang berjumlah 5 buah. Karakter pemain yang berupa keranjang diletakkan di bagian tengah bawah layar sebagai area permainan. Telur yang akan dikeluarkan oleh ayam akan diacak posisi keluarnya oleh sistem. Setelah seluruh entitas yang diperlukan sudah diinisialisasi, maka permainan pun dapat segera dimulai.

2. Control Permainan

Control permainan pada game Catch The Eggs akan dijelaskan pada tabel berikut:

Tabel 3.2 Control Permainan Pada *Game* Catch The Eggs

Control Permainan
1. Permainan ini menggunakan sistem touch screen sebagai alat input

untuk mengontrol karakternya. 2. Pemain dapat menggeser karakter ke sebelah kanan dengan menyentuh atau mengetuk layar sebelah kanan, dan menggeser karakter ke sebelah kiri dengan mengetuk layar sebelah kiri.

3. Misi Permainan

Misi permainana pada game Catch The Eggs akan dijelaskan pada tabel berikut:

Tabel 3.3 Misi Permainan Pada *Game* Catch The Eggs

Misi Permainan
Pemain harus berhasil menangkap setiap telur yang dijatuhkan oleh ayam, dan mendapatkan skor tertinggi dalam setiap level permainan.

4. *Gameplay* Catch The Eggs

Gameplay pada game Catch The Eggs akan dijelaskan pada tabel berikut:

Tabel 3.4 *Gameplay* Catch The Eggs

Gameplay
1. Karater utama adalah sebuah keranjang yang bertugas menangkap telur yang dikeluarkan oleh ayam-ayam yang sedang bertelur. 2. Untuk menangkap telur tersebut, karakter harus digeser-geser agar posisinya sejajar dengan telur yang dikeluarkan oleh ayam. 3. Target jumlah telur yang harus ditangkap berbeda-beda pada setiap level permainan. 4. Jika telur tidak berhasil ditangkap, maka terlur akan jatuh dan pecah, pada setiap level permainan, dibatasi hanya ada 5 telur saja yang boleh pecah. Permainan akan berhenti apabila telur yang pecah berjumlah 5 butir. 5. Setiap telur yang pecah akan mengurangi nyawa pemain.

6. Bila pemain tidak berhasil menangkap telur granat, maka permainan akan langsung berhenti.
7. Pada awal permainan, akan ada petunjuk cara bermain *game* tersebut.
8. Dalam *game* ini ada point berupa skor apabila pemain berhasil menyelesaikan permainan.

3.1.2.2 *Game* Memory

Game ini merupakan *game* casual yang tujuan utamanya adalah melatih ingatan pemain.



Gambar 3.2 Tampilan *Game* Memory

Skenario Permainan

Skenario permainan terdiri dari beberapa hal, sebagai berikut:

1. **Alur Sistem**

   Sistem akan menginisialisasi seluruh entitas yang diperlukan untuk memulai permainan, seperti kartu-kartu gambar yang akan digunakan dalam permainan. Sistem akan mengacak letak dari kartu gambar

tersebut. Setelah seluruh entitas yang diperlukan sudah diinisialisasi, maka permainan pun dapat segera dimulai.

2. Control Permainan

Control permainan pada game Memory akan dijelaskan pada tabel berikut:

Tabel 3.5 Control Permainan *Game* Memory

Control Permainan
Control permainan dilakukan dengan cara menyentuh layar untuk membuka kartu gambar.

3. Misi Permainan

Misi permainan pada game Memory akan dijelaskan pada tabel berikut:

Tabel 3.6 Misi Permainan *Game* Memory

Misi Permainan
Misi permainan dari *game* ini dalah untuk mencocokkan kartu gambar dengan pasangannya.

4. *Gameplay*

Gameplay pada game Memory akan dijelaskan pada tabel berikut:

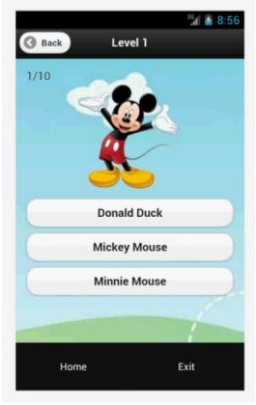
Tabel 3.7 *Gameplay Game* Memory

Gameplay
1. Akan disediakan kartu gambar berjumlah 4x3 yang terdiri dari 6 gambar dan pasangannya. 2. Gambar-gambar tersebut akan diletakkan secara acak dengan posisi dibalik, sehingga gambar tidak bisa dilihat oleh pemain. 3. Tugas pemain adalah membuka kartu gambar tersebut dengan cara

menyentuh kartu, dan juga mencari pasangan dari gambar itu. 4. Jika yang dibuka adalah pasangannya, maka kedua kartu itu akan tetap terbuka, dan pemain mendapatkan skor. 5. Jika yang dibukan bukan pasangannya, maka kedua gambar tersebut akan kembali tertutup.

3.1.2.3 *Game* Cartoon Quiz

Game ini merupakan *game* yang berisi beberapa pertanyaan mengenai kartun.



Gambar 3.3 Tampilan *Game* Cartoon Quiz

Skenario Permainan

1. **Alur Sistem**

   Sistem akan menginisialisasi seluruh entitas yang diperlukan untuk memulai permainan, seperti gambar-gambar kartun dan juga pertanyan-pertanyaan yang akan digunakan. Setelah seluruh entitas yang diperlukan sudah diinisialisasi, maka permainan pun dapat segera dimulai.

2. **Control Permainan**

   Control permainan pada game Cartoon Quiz akan dijelaskan pada tabel berikut:

Tabel 3.8 Control Permainan *Game* Cartoon Quiz

Control Permainan
Control permainan dilakukan dengan touch screen. Pemain cukup memilih jawaban yang tepat dengan menyentuh layar.

3. Misi Permaianan

Misi permainan pada game Cartoon Quiz akan dijelaskan pada tabel berikut:

Tabel 3.9 Misi Permaianan *Game* Cartoon Quiz

Misi Permaianan
Misi dari permainan ini adalah untuk menebak gambar sebanyak-banyaknya, dan menjawab pertanyaan yang ada.

4. *Gameplay*

Gameplay pada game Cartoon Quiz akan dijelaskan pada tabel berikut:

Tabel 3.10 *Gameplay Game* Cartoon Quiz

Gameplay
1. Akan disediakan beberapa gambar dan pertanyaan mengenai kartun. 2. Tugas pemain adalah menebak gambar tokoh kartun yang ada. 3. Jika jawaban benar, maka skor permainan akan bertambah.

3.1.2.4 Kekurangan Game Sejenis

Analisis kekurangan dari game sejenis yang sudah dibahas di atas akan digunakan sebagai acuan untuk membangun game edukasi melawan kanker yang akan dibangun. Analisis kekurangan tersebut akan dijelaskan pada tabel berikut:

Tabel 3.11 Kekurangan Game Sejenis

No	Nama Game	Kekurangan
1	Catch The Eggs	1. Tidak ada unsur edukasi dalam permainannya. 2. Pergerakan karakter penangkap yang kurang cepat, karena digerakkan dengan cara menyentuh layar.
2	Game Memory	1. Tidak ada unsur edukasi dalam permainannya
3	Game Cartoon Quiz	1. Tidak terdapat unsur edukasi dalam permainannya. 2. Tidak ada penjelasan mengenai jawaban setelah pemain memilih jawaban pertanyaan

3.1.3 Pengenalan *Game* Edukasi Melawan Kanker Berbasis *Mobile*

Game yang akan dibangun berjudul *game* melawan kanker yang bertipe casual. *Game* ini merupakan perpaduan dari beberapa *game* yang sudah dibahas di atas, *game* ini juga menggunakan teknologi accelerometer sebagai kontrol karakter, sehingga lebih menarik.

Berikut ini adalah fitur-fitur yang ada pada *game* ini:

1. Sistem single player.
2. Mengangkat tema mengenai penyebab penyakit kanker.
3. Grafik *game* 2D.
4. Jenis *game* bertipe casual dan disajikan secara interaktif.
5. Menerapkan teknologi accelerometer untuk kontorl karakter, yaitu dengan memiringkan perangkat *mobile* ke kanan dan kiri.
6. Aplikasi *game* ini bersifat bermain dan belajar, karena dalam *game* ini juga disajikan informasi mengenai penyebab-penyebab penyakit kanker.

3.1.3.1 *Storyline*

Game edukasi ini mempunyai *gameplay* yang sederhana dan kontrol karakter yang mudah digunakan bagi pengguna, yaitu target utama yang merupakan anak usia 7-16 tahun. Disamping itu, *game* ini juga memberikan pengetahuan mengenai penyebab-penyebab penyakit kanker, seperti jenis-jenis makanan sehat dan tidak sehat

Game ini merupakan perpaduan dari *game-game* sejenis yang sudah dibahas di atas. *Game* ini terdiri dari tiga bagian *game*. Berikut adalah storyline dari setiap bagian *game* tersebut:

1. *Game* Tangkapan Sehat

   Pada level pertama, *gameplay* permainan hampir sama dengan permainan Catch The Eggs. Pada bagian ini, akan disediakan karakter Hero dan Food. Karaker Food yaitu berupa gambar makanan yang terdiri dari makanan yang sehat dan tidak sehat. Tugas utama pemain yaitu menangkap gambar makanan sehat, dan menghindari gambar makanan yang tidak sehat. Pada level ini diharapkan dapat membantu anak untuk mempelajari beberapa makanan sehat yang dapat digunakan untuk mencegah penyakit kanker, dan menghindari makanan tidak sehat yang dapat menyebabkan kanker.

2. *Game* Ingatan Sehat

   Pada level kedua, *gameplay* permainan hampir sama dengan *game* Memory. Akan disediakan beberapa gambar mengenai hal-hal yang dapat mencegah penyakit kanker, dan juga hal-hal yang dapat menyebabkan penyakit kanker.

   Hal-hal yang dapat mencegah penyakit kanker diantaranya adalah makanan sehat yang dapat berupa sayuran dan buah, lingkungan yang sehat misalnya yang tidak tercemar polusi, dan aktivitas sehat, misalnya melakukan olah raga.

Hal-hal yang dapat menyebabkan kanker, misalnya makanan instan, makanan yang mengandung bahan pengawet, lingkungan yang tercemar sampah, dan lain-lain.

Gambar yang dapat mencegah kanker akan dipasangkan dengan gambar sejenisnya, misalnya gambar buah-buahan dengan buah-buahan, sayuran dengan sayuran.

Gambar yang dapat menyebabkan kanker akan dipasangkan dengan gambar tong sampah. Gambar-gambar tersebut akan diletakkan secara acak, dan posisi gambar dibalik, seingga pemain tidak mengetahui letak gambar yang sebenarnya. Tugas utama pemain adalah memasangkan gambar dengan pasangannya masing-masing, dengan cara membuka setiap kartu gambar yang disediakan. Apabila kartu gambar yang dibuka adalah bukan pasangannya, maka kartu tersebut akan menutup kembali.

Pada permainan ini diharapkan dapat membantu anak untuk mengingat dan membedakan hal-hal yang dapat mencegah dan menyebabkan kanker.

3. *Game* Kuis Sehat

Pada *game* ini, gamaplay permainan hampir sama dengan *game* Cartoon Quiz. Akan disediakan beberapa gambar dan pertanyaan seputar pengetahuan mengenai penyakit kanker.

Tugas utama pemain adalah memilih jawaban atas pertanyaan yang disediakan. Pada permainan ini diharapkan dapat menambah pengetahuan anak seputar penyakit kanker, baik mengenai penyebab maupun cara pencegahannya.

3.1.3.2 *Gameplay*

Gameplay pada game edukasi melawan kanker ini terdiri dari beberapa gameplay, yaitu:

1. *Gameplay* level 1 (Game Tangkapan Sehat)

Gameplay level 1 (Game Tangkapan Sehat) akan dijelaskan pada tabel berikut:

Tabel 3.12 *Game*paly Level 1 (Tangkapan Sehat)

Gameplay
1. Karakter berada di layar bagian bawah 2. Karakter memiliki kemampuan untuk bergeser secara horizontal 3. Pemain mengendalikan karakter dengan memiringkan ponsel ke kanan atau kiri. 4. Akan disediakan 3 nyawa. 5. Pemain dapat mengumpulkan poin dengan menangkap gambar jenis makanan sehat. 6. Pemain harus menghindari gambar makanan yang tidak sehat, karena dapat mengurangi nyawa. 7. Permainan akan berakhir apabila nyawa sudah habis.

2. *Gameplay* level 2 (Game Ingatan Sehat)

Gameplay level 2 (Game Ingatan Sehat) akan dijelaskan pada tabel berikut:

Tabel 3.13 *Gameplay* Level 2

Gameplay
1. Akan disediakan beberapa gambar mengenai penyebab dan pencegah penyakit kanker. 2. Gambar-gambar pencegah kanker sudah memiliki pasangan masing-masing, misalnya sayuran dengan sayuran, buah dengan buah. 3. Gambar-gambar penyebab kanker akan dipasangkan dengan gambar tong sampah. 4. Gambar-gambar tersebut akan diletakkan secara acak dengan posisi dibalik, sehingga gambar tidak bisa dilihat oleh pemain.

5. Tugas pemain adalah membuka kartu gambar tersebut dengan cara menyentuh gambar kartu, dan juga mencari pasangan dari gambar itu. 6. Jika yang dibuka adalah pasangannya, maka kedua kartu itu akan tetap terbuka, dan pemain mendapatkan skor. 7. Jika yang dibuka bukan pasangannya, maka kedua gambar tersebut akan kembali tertutup.

3. *Gameplay* level 3 (Game Kuis Sehat)

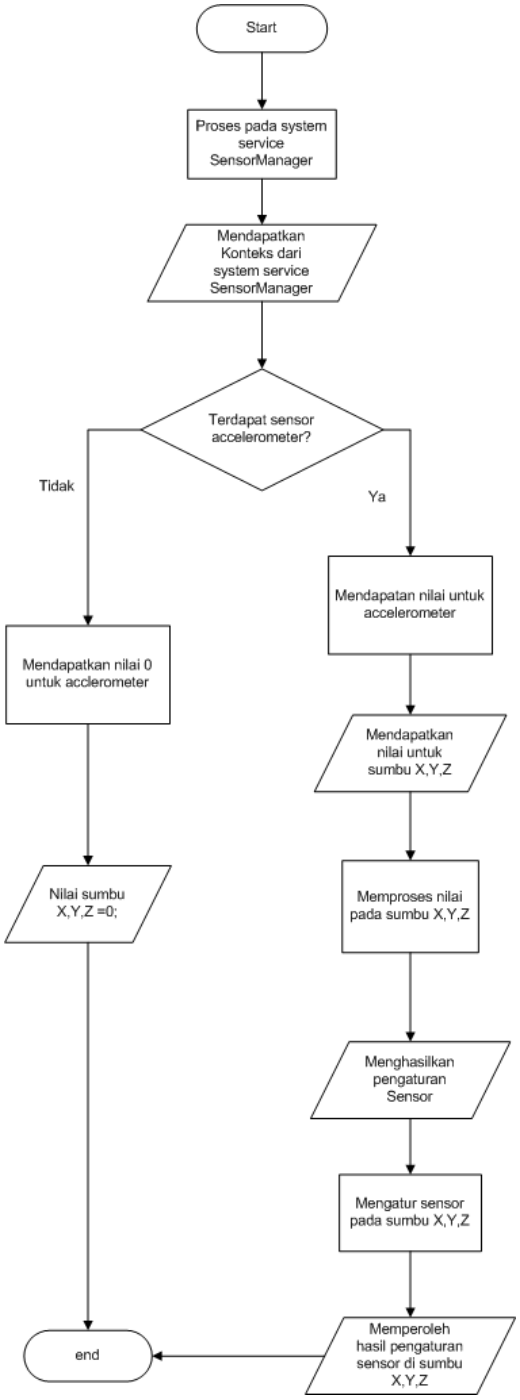
Gameplay level 3 (Game Kuis Sehat) akan dijelaskan pada tabel berikut:

Tabel 3.14 *Gameplay* Level 3

Gameplay
1. Akan disediakan beberapa pertanyaan seputar penyakit kanker. 2. Tugas pemain adalah menjawab pertanyaan tersebut dengan memilih pilihan jawaban yang sudah disediakan. 3. Jika jawaban benar, maka skor permainan akan bertambah. 4. Jika jawaban salah, maka skor permainan akan berkurang.

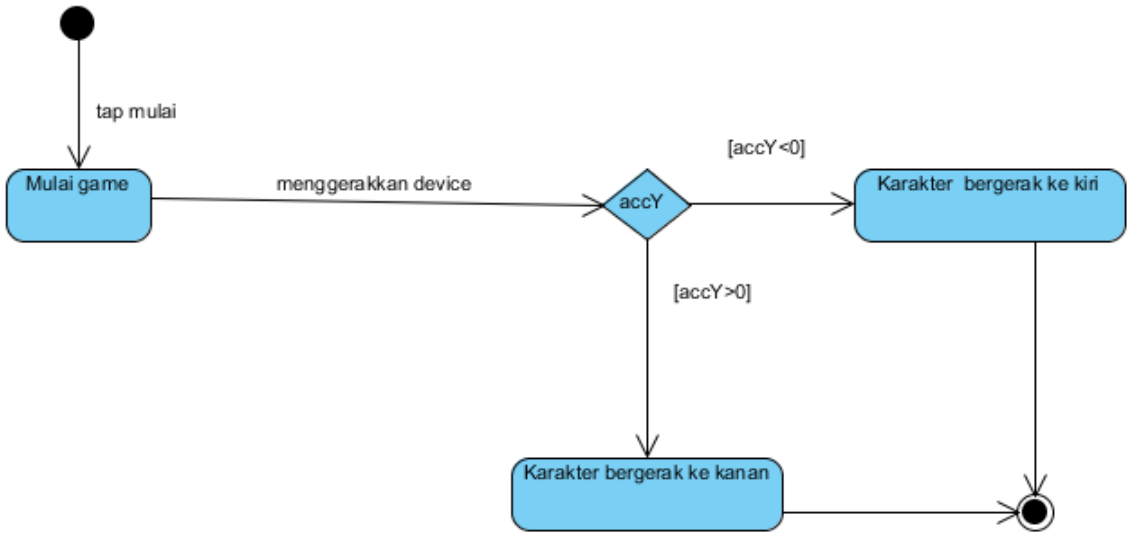
3.1.3.3 Analisis Penerapan Accelerometer Pada Game Melawan Kanker Berbasis *Mobile*

Pada game ini, diperlukan adanya sensor accelerometer yang digunakan dalam Game Tangkapan Sehat. Karena karakter Hero pada game ini digerakkan dengan cara menggerakkan device, maka accelerometer digunakan untuk mendeteksi pergerakan device tersebut. Jika device bergerak ke kanan, maka Hero akan bergereak ke kanan, jika device bergerak ke kiri, maka Hero juga akan bergerak ke kiri. Proses accelerometer dapat dilihat pada flowchart berikut ini:



Gambar 3.4 Flowchart Proses Accelerometer

Sedangkan untuk state diagram proses accelerometer akan dijelaskan pada gambar berikut:



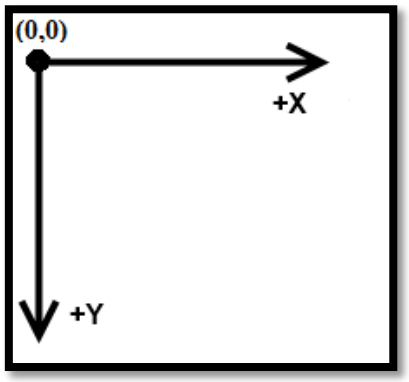
Gambar 3.5 State Diagram Proses Accelerometer

3.1.3.4 Analisis Penerapan Collision Detection Pada Game Melawan Kanker Berbasis *Mobile*

Dalam pembuatan game ini diperlukan adanya penggunaan metode collision detection yang merupakan metode untuk mengecek obyek yang saling bersinggungan, baik itu sprite dengan sprite, maupun sprite dengan objek lain.

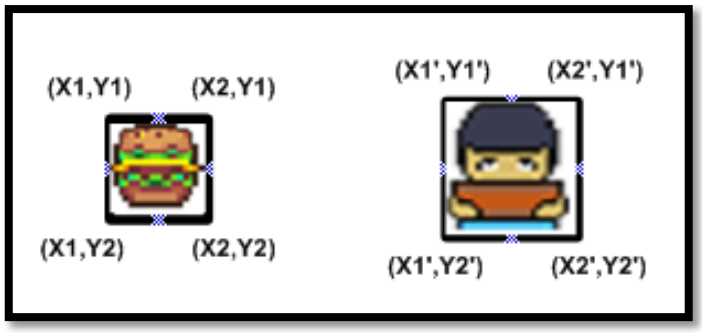
Pada game ini, metode yang akan diterapkan adalah Rectangles Collision Detection. Teknik ini merupakan pendeteksian tabrakan menggunakan bound berbentuk persegi, dimana setiap sprite atau objek memiliki bound masing-masing. Cara ini cukup mudah karena hanya perlu membandingkkan apakah ada intersection (irisan) pada kedua objek pada koordinat tertentu. Teknik ini dipilih karena dalam game ini menggunakan sprite yang bentuknya mendekati persegi dan grafis 2D. Teknik ini digunakan pada game Tangkapan sehat, dimana Karakter Hero akan menangkap makanan yang baik atau tidak baik. Untuk mengetahui apakah makanan itu tertangkap oleh Hero atau tidak, maka digunakan teknik Rectangles Collision Detection.

Cara mengetahui bagaimana tubrukan terjadi, pertama adalah menentukan titik koordinat dari masing-masing objek. Koordinat yang digunakan dalam pemrograman java adalah koordinat (0,0) terletak pada pojok kiri atas layar, X akan bernilai positif bila bergeser ke kanan, dan Y akan bernilai positif apabila bergeser ke bawah. Hal ini dapat dijelaskan pada gambar berikut:



Gambar 3.6 Letak Koordinat Awal

Pada game ini, ada dua karakter, yaitu karakter food dan karakter hero. Misalnya karakter Food diumpamakan sebagai P1 dengan koordinat (X1,Y1), (X1,Y2), (X2,Y1), (X2,Y2), sedangkan karakter Hero diumpamakan sebagai P2 dengan koordinat (X1',Y1'), (X1',Y2'), (X2',Y1'), (X2',Y2').

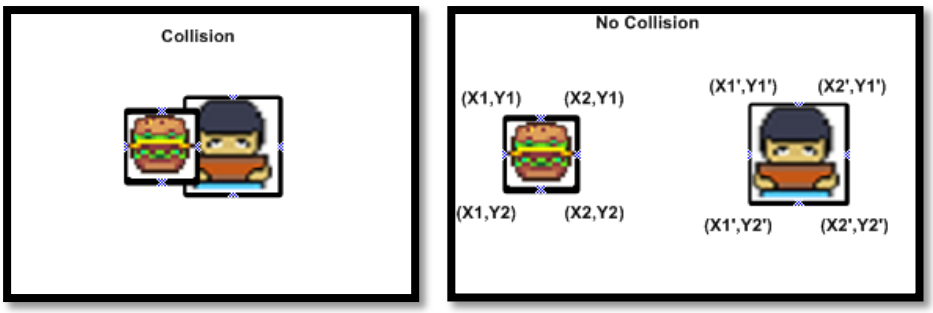


Gambar 3.7 Koordinat Persegi

a. Horizontal Collision

   Untuk memeriksa collision detection secara horizontal dapat dilakukan dengan cara sebagai berikut:

1. Memeriksa tepi kanan P1, jika lebih besar dari tepi kiri P2, maka kemungkinan terjadi collision, X1>X2’
2. Memeriksa tepi kiri P1, jika lebih kecil dari tepi kanan P2, maka kemungkinan terjadi collision, X2<X1’
3. Jika kedua kondisi di atas terpenuhi, maka kedua persegi tersebut akan bertabrakan secara horizontal. Sehingga diperoleh persamaan = X1>X2’ && X2<X1’

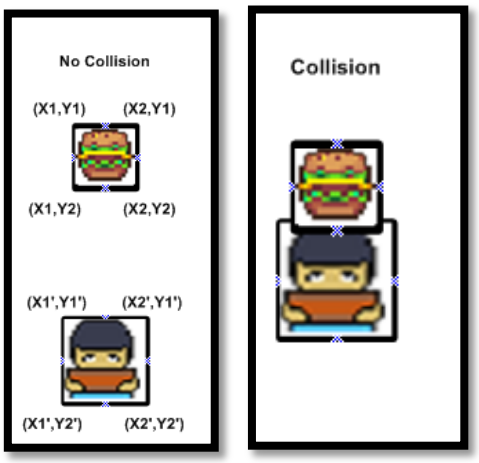


Gambar 3.8 Horizontal Collision

b. Vertical Collision

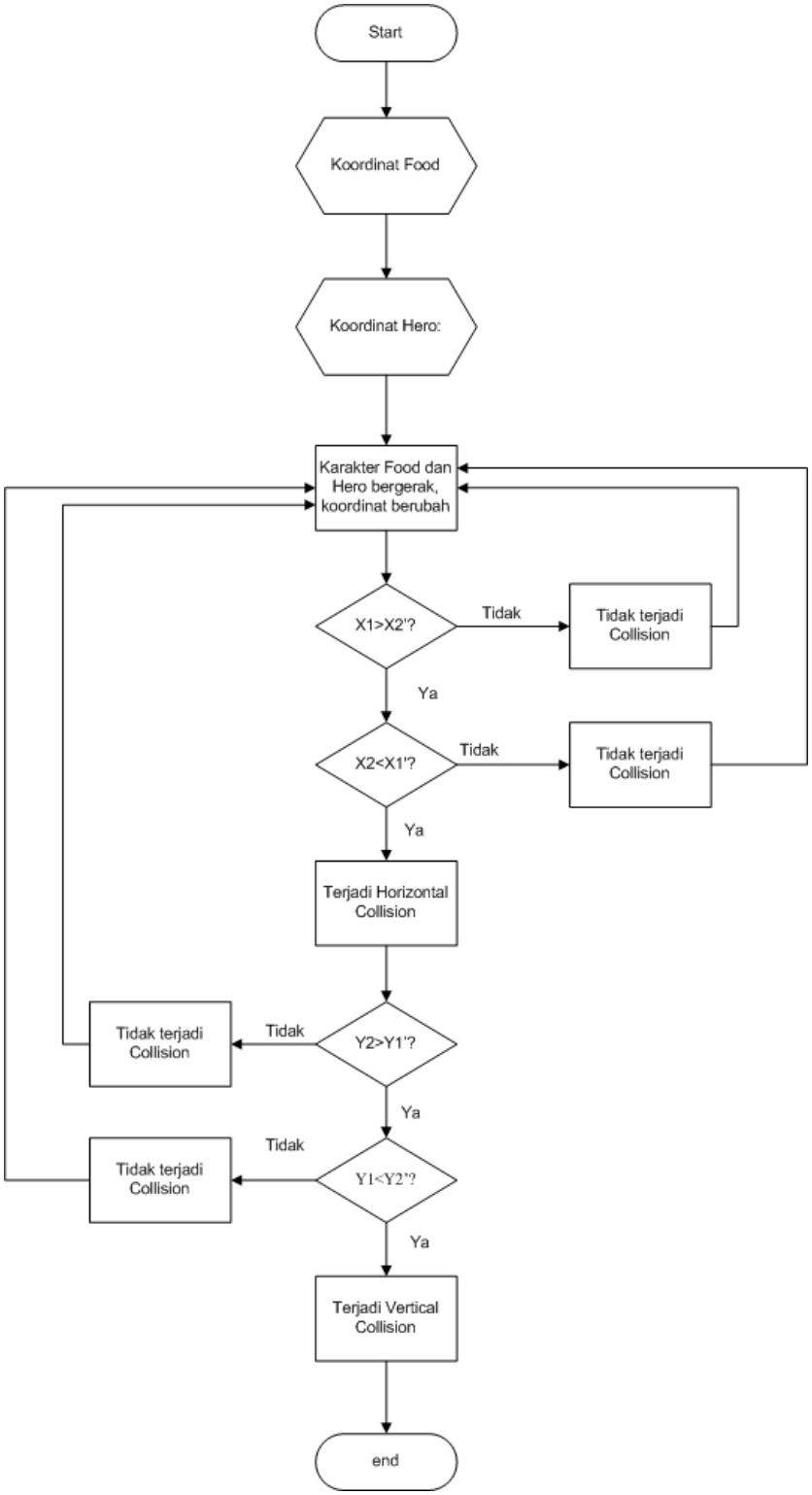
Untuk memeriksa collision detection secara vertical dapat dilakukan dengan cara sebagai berikut:

1. Memeriksa sisi bawah P1, apabila lebih besar dari sisi atas P2, maka kemungkinan terjadi collision, Y2>Y1’.
2. Memeriksa sisi atas P1, apabila lebih kecil dari sisi bawah P2, maka kemungkinan terjadi collision, Y1<Y2’.
3. Jika kedua kondisi tersebut terpenuhi, maka kedua persegi tersebut akan bertabrakan secara vertikal, sehingga diperoleh persamaan Y2>Y1’ && Y1<Y2’.



Gambar 3.9 Vertical Collision

Sedangkan flow chart untuk proses collision detection akan dijelaskan pada gambar berikut ini:



Gambar 3.10 Flow chart Collision Detection

3.1.4 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 dibutuhhkan untuk sistem yang akan dibangun sampai 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 akan dijelaskan mengenai analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras, dan analisis pengguna.

3.1.4.1 Analisis Kebutuhan Perangkat Lunak

Game edukasi melawan kankaer berbasisi *mobile* ini memiliki kebutuhan perangkat lunak sebagai berikut:

1. Sistem aplikasi Android.
2. API android terendah dan build target yang digunakan adalah 2.3.3
3. Konektifitas offline.
4. Minimal prosessor 1Gz.
5. Grafis pada *game* ini adalah 2 dimensi.
6. Menggunakan fitur accelerometer atau screen shake

Dari hasil pengamatan perangkat lunak, terdapat beberapa hal penting yang harus diperhatikan agar sistem yang akan dirancang dapat berjalan dengan baik sesuai yang diharapkan. Pengguna diharapkan dapat memenuhi kebutuhan perangkat lunak yang telah direkomendasikan agar aplikasi dapat berjalan dengan baik.

3.1.4.2 *Analisis Kebutuhan Perangkat Keras*

Agar aplikasi berjalan dengan baik, maka dibutuhkan perangkat keras yang sesuai dengan kebutuhan aplikasi. Spesifikasi minimum perangkat keras yang dibutuhkan agar dapat menjalankan *game* Edukasi Melawan Kanker ini adalah sebagai berikut:

1. Perangkat *mobile* dengan software android
2. Memori procesor minimal 1 Ghz
3. Kapasitas memory external minimal 256 MB

3.1.4.3 *Analisis Pengguna*

Analisis pengguna untuk *game* edukasi melawan kanker berbasis *mobile* akan dijelaskan pada tabel berikut:

Tabel 3.15 Analisis Pengguna

Karakteristik Fisik Pegguna	
Umur	7-14 tahun
Gender	Laki-laki dan perempuan
Handedness	Cenderung menggunakan tangan kanan
Disabilities	Tidak buta dan tidak mempunyai cacat fisik
Pengetahuan dan Pengalaman Pengguna	
Computer Literacy	Memiliki pengalaman dalam mengoperasikan perangkat *mobile*
System Experience	Memiliki pengetahuan berinteraksi dengan input *mobile* phone
Application Experience	Pernah memainan *game* sejenis sebelumnya
Education	Sekolah Dasar
Skills	Sudah bisa membaca
Task Experience	Mengetahui cara menggunakan *mobile* phone
Native Language or Culture	Bahasa Indonesia
Karakteristik Psikologi Pengguna	
Motivation	Mempunyai motivasi yang cukup dalam menyelesaikan permainan

Patience	Mempunyai kesabaran dalam menyelesaikan permainan
Stress Level	Tidak gampang stress
Attitude	Possitive

3.1.5 Analisis Kebutuhan Fungsional

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

3.1.5.1 SKPL (Spesifikasi Kebutuhan Perangkat Lunak)

Spesifikasi kebutuhan perangkat lunak pada pembangunan Game Edukasi Melawan Kanker ini dibagi menjadi dua, yaitu Kebutuhan Fungsional dan Kebutuhan Non Fungsional.

a. Kebutuhan Fungsional

Kebutuhan fungsional pada pembangunan Game Edukasi Melawan Kanker ini akan dijelaskan pada tabel berikut:

Tabel 3.16 Kebutuhan Fungsional

SKPL-Id	Kebutuhan
SKPL-F-001	Aplikasi harus dapat menampilkan halaman *About*, dan Petunjuk cara bermain masing-masing *game*.
SKPL-F-002	Aplikasi harus dapat menampilkan *game* Tangkapan Sehat, Ingatan Sehat, dan Kuis Sehat.
SKPL-F-003	Aplikasi harus dapat mengelola skor permainan pada masing-masing *game*.
SKPL-F-004	Aplikasi harus dapat menggerakkan karakter pemain sesuai dengan inputan pemain.
SKPL-F-005	Aplikasi harus dapat mengecek collision yang terjadi

	pada *game* Tangkapan Sehat.
SKPL-F-006	Aplikasi harus dapat mengecek kartu yang dibuka oleh pemain pada *game* Ingatan Sehat.
SKPL-F-007	Aplikasi harus dapat mengecek jawaban pemain pada *game* Kuis Sehat.
SKPL-F-008	Aplikasi harus dapat menampilkan hasil akhir permainan pada game Tangkapan Sehat, Ingatan Sehat, dan Kuis Sehat.

b. Kebutuhan Non Fungsional

Kebutuhan non fungsional pada pembangunan Game Edukasi Melawan Kanker ini akan dijelaskan pada tabel berikut:

Tabel 3.17 Kebutuhan Non Fungsional

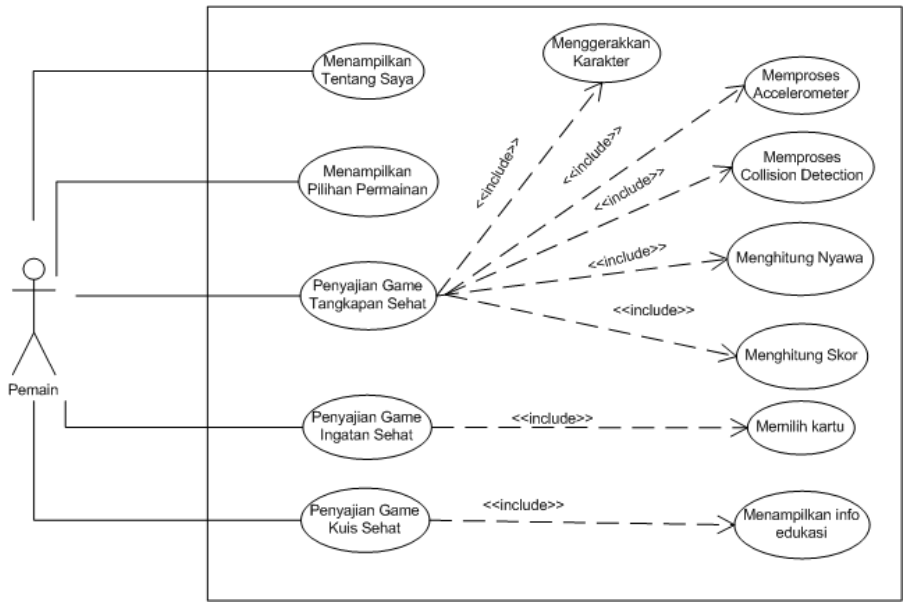
SKPL-Id	Kebutuhan
SKPL-NF-001	Aplikasi harus memiliki nilai edukasi mengenai penyakit kanker.
SKPL-NF-002	Aplikasi ini harus dapat diakses oleh user selama kebutuhan perangkat memungkinkan.
SKPL-NF-003	Aplikasi ini harus memiliki nilai ergonomi/ kenyamanan dipakai yang tinggi bagi user. Aplikasi akan dibangun dengan antarmuka user yang mudah dimengerti, indah dilihat, konsisten, mudah dioperasikan dan tidak membingungkan.
SKPL-NF-004	Aplikasi harus memiliki waktu response yang cepat, agar membuat proses interaksi permainan menjadi lebih baik.
SKPL-NF-005	Bahasa yang digunakan harus komunikatif dan mudah dimengerti oleh user.

3.1.5.2 *Use Case Diagram*

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

3.1.5.2.1 *Use Case* Diagram

Use case diagram yaitu menggambaran fungsionalitas atau perilaku dari sistem.



Gambar 3.11 *Use case* Diagram

3.1.5.2.2 *Use Case Scenario*

Use case Scenario merupakan suatu penggambaran langkah-langkah aksi aktor terhadap sistem.

a. *Use case Scenario* Menampilkan Menu Tentang Saya

   Use case Scenario Menampilkan Menu Tentang Saya akan dijelaskan pada tabel berikut:

Tabel 3.18 *Use case Scenario* Menampilkan Menu Tentang Saya

Identifikasi	
Nomor	1
Nama	Menampilkan Menu Tentang Saya
Tujuan	Untuk menampilkan menu Tentang Saya
Deskripsi	Proses untuk menampilkan menu Tentang Saya
Aktor	Pemain
Skenario Utama	
Aksi Aktor Pemain	Reaksi Sistem
1. Memilih menu Tentang Saya	
	2. Membuka load halaman *About*
	3. Menampilkan halaman *About*
4. Menerima tampilan halaman *About*	
Kondisi Akhir	Aplikasi *game* menampilkan halaman *About*

b. *Use case Scenario* Menampilkan Pilihan Permainan

Use case Scenario Menampilkan Pilihan Permainan akan dijelaskan pada tabel berikut:

Tabel 3.19 *Use case Scenario* Menampilkan Pilihan Permainan

Identifikasi	
Nomor	2
Nama	Menampilkan Pilihan Permainan
Tujuan	Untuk menampilkan menu Pilihan Permainan
Deskripsi	Proses untuk menampilkan menu Pilihan Permainan
Aktor	Pemain
Skenario Utama	
Aksi Aktor Pemain	Reaksi Sistem
1. Memilih Game Melawan Kanker	
	2. Membuka load halaman utama
	3. Menampilkan halaman utama yang

	berisi tampilan pilihan permainan.
4. Menerima tampilan halaman utama yang berisi tampilan pilihan permainan.	
Kondisi Akhir	Aplikasi *game* menampilkan menu utama yang berisi tampilan pilihan permainan

c. *Use case Scenario* Penyajian *Game Tangkapan sehat*

Use case Scenario Penyajian *Game* Tangkapan sehat akan dijelaskan pada tabel berikut:

Tabel 3.20 *Use case Scenario* Penyajian *Game Tangkapan Sehat*

Identifikasi	
Nomor	3
Nama	Penyajian *Game* Tangkapan Sehat
Tujuan	Untuk menampilkan penyajian *Game* Tangkapan Sehat
Deskripsi	Proses penyajian *Game* Tangkapan Sehat
Aktor	Pemain
Skenario Utama	
Aksi Aktor Pemain	Reaksi Sistem
1. Memilih menu *Game* Tangkapan Sehat	
	2. Memproses loading sebelum masuk ke dalam permainan
	3. Memulai *Game* Tangkapan Sehat
	4. Menampilkan petunjuk permainan dan info edukasi
5. Memilih tombol main untuk memulai permainan	
6. Menggerakkan karakter	
	7. Menggerakkan karakter sesuai

	inputan
	8. Melakukan pengecekan *Collision* dengan karakter food.
	9. Jika berhasil menangkap karakter food sehat, maka skor bertambah
	10. Jika menangkap karakter food yang tidak sehat, maka nyawa berkurang
	11. Melakukan pengecekan nyawa
	12. Jika nyawa >0, maka melanjutkan permainan
	13. Jika nyawa=0, maka permainan berakhir
	14. Menampilkan hasil permainan
15. Menerima tampilan hasil permainan	
Kondisi Akhir	Aplikasi *game* menampilkan hasil akhir permainan.

d. *Use case Scenario* Menggerakkan Karakter

Use case Scenario Menggerakkan Karakter akan dijelaskan pada tabel berikut:

Tabel 3.21 *Use case Scenario* Menggerakkan Karakter

Identifikasi	
Nomor	4
Nama	Menggerakkan Karakter
Tujuan	Untuk memproses pergerakkan karakter pada game Tangkapan Sehat
Deskripsi	Proses pergerakkan karakter
Aktor	Pemain
Skenario Utama	
Aksi Aktor Pemain	Reaksi Sistem
1. Memiringkan device ke kanan	
	2. Karakter akan bergerak ke kanan

3. Memiringkan device ke kiri	
	4. Karakter akan bergerak ke kiri
5. Menerima tampilan pergereakan karakter	
Kondisi Akhir	Aplikasi *game* menampilkan pergerakkan karakter pada game Tangkapan sehat

e. *Use case Scenario* Memproses Accelerometer

Use case Scenario Memproses Accelerometer akan dijelaskan pada tabel berikut:

Tabel 3.22 *Use case Scenario* Memproses Accelerometer

Identifikasi	
Nomor	5
Nama	Memproses Accelerometer
Tujuan	Untuk memproses sensor accelerometer pada perangkat
Deskripsi	Proses Accelerometer
Aktor	Pemain
Skenario Utama	
Aksi Aktor Pemain	Reaksi Sistem
	1. Memeriksa ketersediaan sensor accelerometer
	2. Jika sensor accelerometer tidak ada, maka accY akan selalu bernilai nol (0)
	3. Jika sensor accelerometer ada, maka akan memulai proses berikutnya
4. Memiringkan device ke kanan	
	5. Mendapatkan nilai positive (+) pada accY untuk karakter yang bergerak ke kanan sebesar kemiringan

6. Memiringkan device ke kiri	
	7. Mendapatkan nilai negative (-) pada accY untuk karakter yang bergerak ke kiri sebesar kemiringan
Kondisi Akhir	Aplikasi *game* menampilkan permainan Game Tangkapan sehat.

f. *Use case Scenario* Memproses Collision Detection

Use case Scenario Memproses Collision Detection akan dijelaskan pada tabel berikut:

Tabel 3.23 *Use case Scenario* Memproses Collision Detection

Identifikasi	
Nomor	6
Nama	Memproses Collision Detection
Tujuan	Untuk mengecek proses collision detection pada Game Tangkapan sehat
Deskripsi	Proses Collision Detection
Aktor	Pemain
Skenario Utama	
Aksi Aktor Pemain	Reaksi Sistem
	1. Membuat karakter food yang akan dijatuhkan
	2. Menjatuhkan karakter food yang sudah dibuat
	3. Memasukkan semua karakter food yang ada di screen ke array listFoodAktif.
4. Memiringkan device untuk menggerakkan karakter hero.	
	5. Mengecek apakah karakter food bertabrakan dengan hero atau tidak.
	6. Jika bertabrakan, maka sistem akan

	mengecek apakah karakter food termasuk makanan yang sehat atau bukan?
	7. Jika termasuk makanan sehat, maka akan menambah skor.
	8. Jika termasuk makanan yang tidak sehat, maka akan mengurangi nyawa.
	9. Jika tidak bertabrakan, maka food akan langsung dimasukkan ke dalam array listFoodNonAktif
	10. Menghapus semua food yang ada di listFoodNonAktif.
Kondisi Akhir	Aplikasi *game* menampilkan permainan Game Tangkapan sehat

g. *Use case Scenario* Menghitung Nyawa

Use case Scenario Menghitung Nyawa akan dijelaskan pada tabel berikut:

Tabel 3.24 *Use case Scenario* Menghitung Nyawa

Identifikasi	
Nomor	7
Nama	Menghitung Nyawa
Tujuan	Untuk memproses perhitungan nyawa
Deskripsi	Proses perhitungan nyawa pada *Game* Ingatan Sehat
Aktor	Pemain
Skenario Utama	
Aksi Aktor Pemain	Reaksi Sistem
	1. Inisialisasi nyawa=3;
2. Menggerakkan karakter hero	
	3. Menggerakkan karakter hero sesuai inputan

	4. Mengecek apakah karakter hero bertabrakan dengan karakter food atau tidak?
	5. Jika bertabrakan dengan kerakter food, maka sistem akan mengecek apakah karakter food termasuk makanan sehat atau makanan tidak sehat?
	6. Jika itu makanan sehat maka kondisi nyawa tidak berubah.
	7. Jika itu makanan yang tidak sehat, maka akan mengurangi nyawa.
	8. Jika tidak terjadi tabrakan, maka nyawa tetap.
	9. Menampilkan hasil perhitungan nyawa
10. Menerima hasil perhitungan nyawa	
Kondisi Akhir	Aplikasi *game* menampilkan hasil perhitungan nyawa.

h. *Use case Scenario* Menghitung Skor

Use case Scenario Menghitung Skor akan dijelaskan pada tabel berikut:

Tabel 3.25 *Use case Scenario* Menghitung Skor

Identifikasi	
Nomor	8
Nama	Menghitung Skor
Tujuan	Untuk memproses perhitungan skor
Deskripsi	Proses perhitungan skor pada *Game* Ingatan Sehat
Aktor	Pemain
Skenario Utama	

Aksi Aktor Pemain	Reaksi Sistem
	1. Inisialisasi skor=0;
2. Menggerakkan karakter hero	
	3. Menggerakkan karakter hero sesuai inputan
	4. Mengecek apakah karakter bertabrakan dengan makanan atau tidak?
	5. Jika bertabrakan dengan karakter food, maka sistem akan mengecek apakah karakter food tersebut termasuk makanan sehat atau makanan tidak sehat?
	6. Jika itu makanan sehat maka skor akan bertambah 1.
	7. Jika itu makanan yang tidak sehat, maka skor tetap.
	8. Jika tidak terjadi tabrakan, maka skor tetap.
	9. Menampilkan hasil perhitungan skor
10. Menerima tampilan perhitungan skor	
Kondisi Akhir	Aplikasi *game* menampilkan hasil perhitungan nyawa.

i. *Use case Scenario* Penyajian *Game* Ingatan Sehat

Use case Scenario Penyajian *Game* Ingatan Sehat akan dijelaskan pada tabel berikut:

Tabel 3.26 *Use case Scenario* Penyajian *Game* Ingatan Sehat

Identifikasi	
Nomor	9
Nama	Menampilkan penyajian *Game* Ingatan Sehat

Tujuan	Untuk menampilkan penyajian *Game* Ingatan Sehat
Deskripsi	Proses penyajian *Game* Ingatan Sehat
Aktor	Pemain

Skenario Utama	
Aksi Aktor Pemain	Reaksi Sistem
1. Memilih menu Mulai *Game* Ingatan Sehat	
	2. Mempersiapkan entitas game
	3. Menampilkan petunjuk permainan dan info edukasi permainan.
4. Memilih tombol main untuk memulai permainan	
	5. Mengacak posisi kartu
	6. Menampilkan kartu dalam posisi tertutup
7. Membuka kartu pertama	
	8. Melakukan pengecekan pada kartu pertama, apakah termasuk kartu sehat atau kartu tidak?
	9. Jika kartu pertama adalah kartu sehat, maka akan dipasangkan dengan kartu sehat.
	10. Jika kartu pertama adalah kartu tidak sehat, maka akan dipasangkan dengan kartu tong sampah.
11. Membuka kartu kedua	
	12. Melakukan pengecekan pada kartu kedua, apakah kartu kedua sesuai pasangan dengan kartu pertama atau tidak?

	13. Jika kartu sesuai pasangan, maka kedua kartu akan menghilang dari permainan
	14. Jika kartu tidak sesuai pasangan, maka kedua kartu akan kembali tertutup
	15. Melakukan pengecekan apakah kartu masih tersedia atau tidak
	16. Jika kartu masih tersedia, maka melanjutkan permainan
	17. Jika kartu sudah habis, maka akan menampilkan hasil permainan
18. Menerima tampilan hasil permainan	
Kondisi Akhir	Aplikasi *game* menampilkan hasil akhir permainan.

j. *Use case Scenario* Memilih Kartu

Use case Scenario Memilih Kartu akan dijelaskan pada tabel berikut:

Tabel 3.27 *Use case Scenario* Memilih Kartu

Identifikasi	
Nomor	10
Nama	Memilih Kartu
Tujuan	Untuk memproses pemilihan kartu pada *Game* Ingatan Sehat
Deskripsi	Proses pemilihan kartu pada *Game* Ingatan Sehat
Aktor	Pemain
Skenario Utama	
Aksi Aktor Pemain	Reaksi Sistem
1. Membuka kartu pertama	
	2. Melakukan pengecekan pada kartu pertama, apakah termasuk kartu sehat atau tidak?

	3. Jika kartu pertama adalah kartu sehat, maka akan dipasangkan dengan kartu sehat.
	4. Jika kartu pertama adalah kartu tidak sehat, maka akan dipasangkan dengan kartu tong sampah.
5. Membuka kartu kedua	
	6. Melakukan pengecekan pada kartu kedua, apakah kartu kedua susah sesuai dengan pasangan kartu pertama?
	7. Jika kartu sesuai pasangan, maka kedua kartu akan menghilang
	8. Jika kartu tidak sesuai pasangan, maka kedua kartu akan kembali tertutup.
	9. Melakukan pengecekan apakah kartu masih tersedia atau tidak
	10. Jika kartu masih tersedia, maka melanjutkan permainan
	11. Jika kartu sudah habis, maka akan menampilkan hasil permainan
12. Menerima tampilan hasil permainan	
Kondisi Akhir	Aplikasi *game* menampilkan game Ingatan Sehat.

k. *Use case Scenario* Penyajian *Game* Kuis Sehat

Use case Scenario Penyajian *Game* Kuis Sehat akan dijelaskan pada gambar berikut:

Tabel 3.28 *Use case Scenario* Penyajian *Game Kuis Sehat*

Identifikasi	
Nomor	11
Nama	Penyajian *Game* Kuis Sehat
Tujuan	Untuk menampilkan penyajian *Game* Kuis Sehat
Deskripsi	Proses penyajian *Game* Kuis Sehat
Aktor	Pemain
Skenario Utama	
Aksi Aktor Pemain	Reaksi Sistem
1. Memilih menu Mulai *Game* Kuis Sehat	
	2. Memproses loading sebelum masuk ke dalam permainan
	3. Mempersiapkan soal
	4. Menampilkan petunjuk permainan
5. Memilih tombol main untuk memulai permainan	
	6. Menampilkan soal
7. Memilih jawaban	
	8. Melakukan pengecekan pada jawaban
	9. Jika jawaban benar, maka akan menampilkan penjelasan jawaban
	10. Jika jawaban salah, nilai salah bertambah
	11. Mengecek apakah nilai salah =5?
	12. Jika salah=5, maka permainan berakhir
	13. Jika salah <5, maka akan menampilkan penjelasan jawaban

	14. Melakukan pengecekan apakah soal masih ada?
	15. Jika soal masih ada, maka sistem akan menampilkan soal selanjutnya
	16. Jika soal sudah habis, maka sistem akan menampilkan hasil akhir permainan.
17. Menerima tampilan hasil akhir permainan	
Kondisi Akhir	Aplikasi *game* menampilkan hasil akhir permainan.

l. *Use case Scenario* Menampilkan Info Edukasi

Use case Scenario Menampilkan Info Edukasi akan dijelaskan pada tabel berikut:

Tabel 3.29 *Use case Scenario* Menampilkan Info Edukasi

Identifikasi	
Nomor	12
Nama	Menampilkan Info Edukasi
Tujuan	Untuk menampilkan info edukasi pada *Game* Kuis Sehat
Deskripsi	Proses untuk menampilkan info edukasi pada *Game* Kuis Sehat
Aktor	Pemain
Skenario Utama	
Aksi Aktor Pemain	Reaksi Sistem
1. Menjawab soal	
	2. Melakukan pengecekan pada jawaban
	3. Jika jawaban benar, maka akan menampilkan info edukasi
	4. Jika jawaban salah, nilai salah bertambah

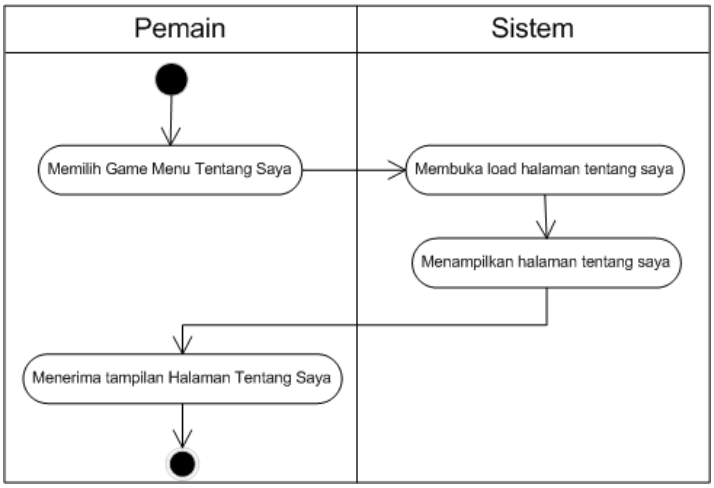
	5. Melakukan pengecekan apakah salah=5?
	6. Jika salah=5, maka permainan berakhir
	7. Jika salah<5, maka menampilkan info edukasi.
8. Menerima info edukasi berupa penjelasan jawaban.	
	9. Mengecek apakah soal masih ada atau tidak?
	10. Jika soal masih ada, maka akan menampilkan soal berikutnya
	11. Jika soal sudah tidak ada, maka akan menampilkan hasil akhir permainan.
12. Menerima hasil akhir permainan	
Kondisi Akhir	Aplikasi *game* menampilkan info edukasi berupa penjelasan jawaban.

3.1.5.2.3 Activity Diagram

Activity diagram menunjukkan aktivitas oleh manusia atau objek dalam cara yang mirip dengan Flow Chart. Berikut ini adalah diagram yang menunjukan aksi pada aktivitas.

1. Activity Diagram Menampilkan Menu Tentang Saya

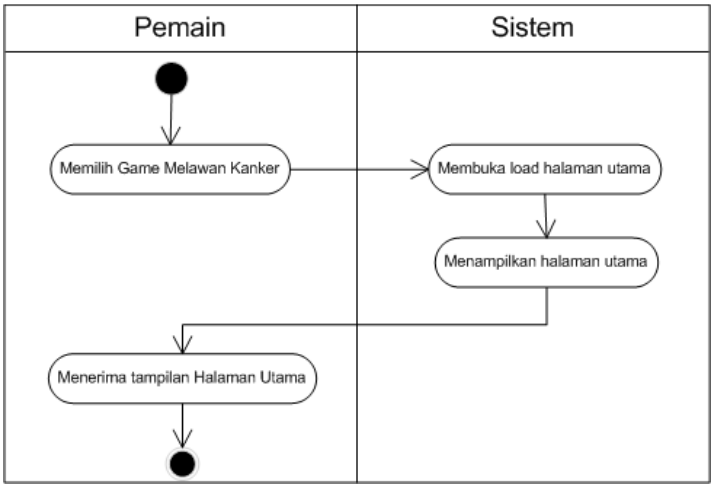
   Activity diagram Menampilkan Menu Tentang Saya akan dijelaskan pada gambar berikut:



Gambar 3.12 Activity Diagram Menampilkan Menu Tentang Saya

2. Activity Diagram Menampilkan Pilihan Permainan

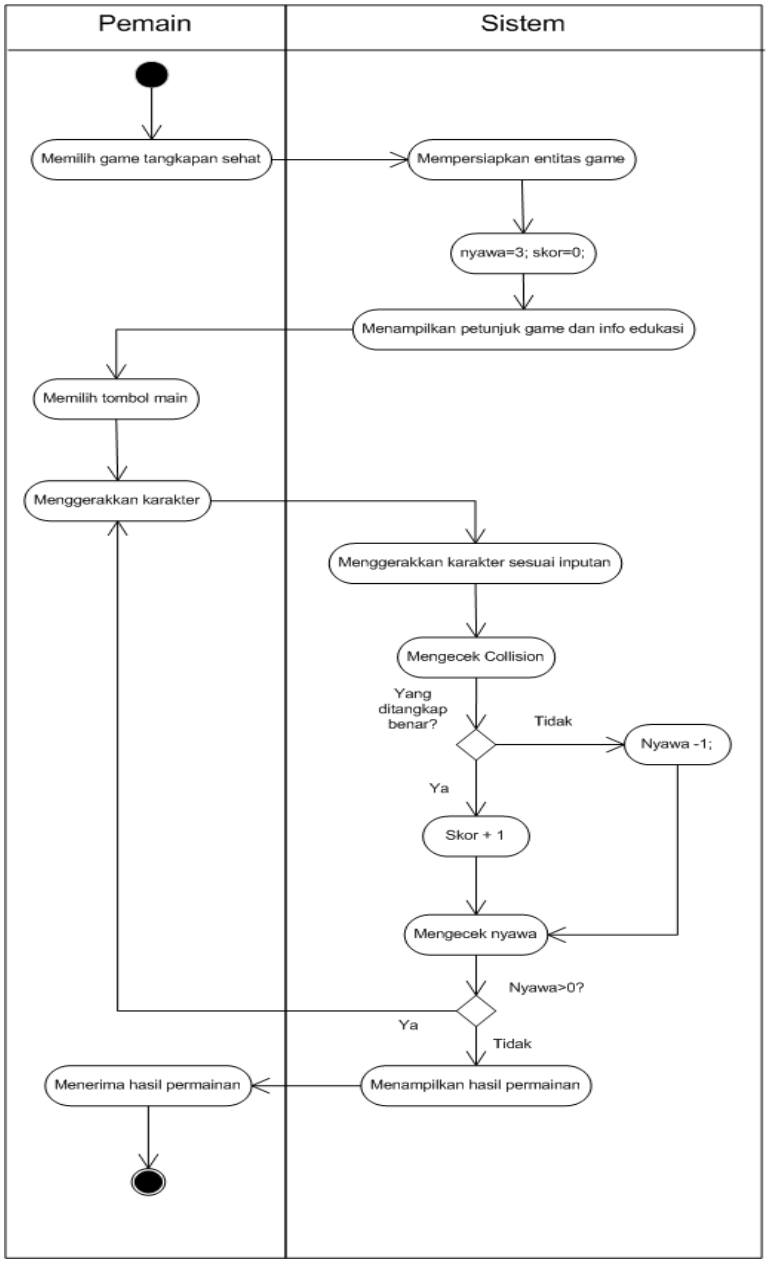
   Activity Diagram Menampilkan Pilihan Permainan akan dijelaskan pada gambar berikut:



Gambar 3.13 Activity Diagram Menampilkan Pilihan Permainan

3. Activity Diagram Penyajian *Game* Tangkapan sehat

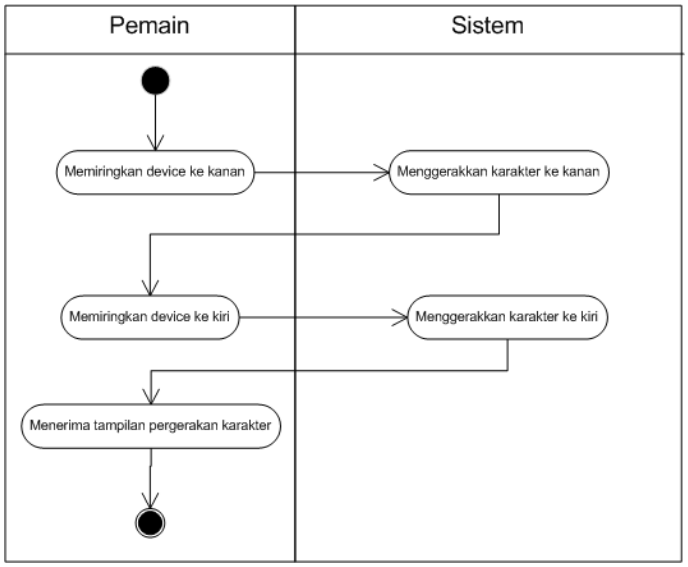
   Activity Diagram Penyajian *Game* Tangkapan sehat akan dijelaskan pada gambar berikut:



Gambar 3.14 Activity Diagram Penyajian *Game Tangkapan sehat*

4. Activity Diagram Menggerakkan Karakter

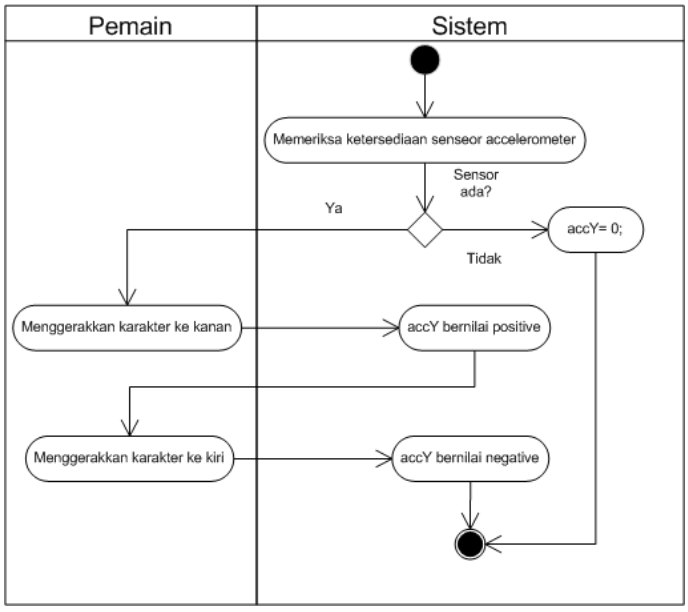
   Activity diagram proses menggerakkan karakter akan dijelaskan pada gambar berikut:



Gambar 3.15 Activity Diagram Menggerakkan Karakter

5. Activity Diagram Memproses Accelerometer

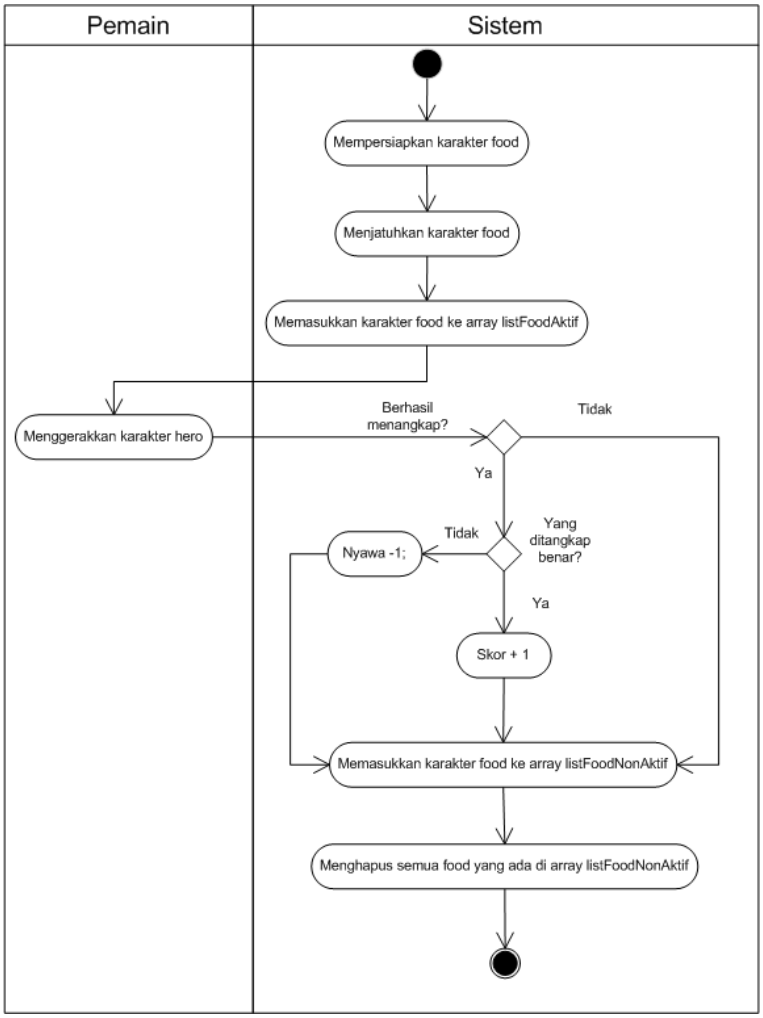
   Activity Diagram Memproses Accelerometer akan dijelaskan pada gambar berikut:



Gambar 3.16 Activity Diagram Memproses Accelerometer

6. Activity Diagram Memproses Collision Detection

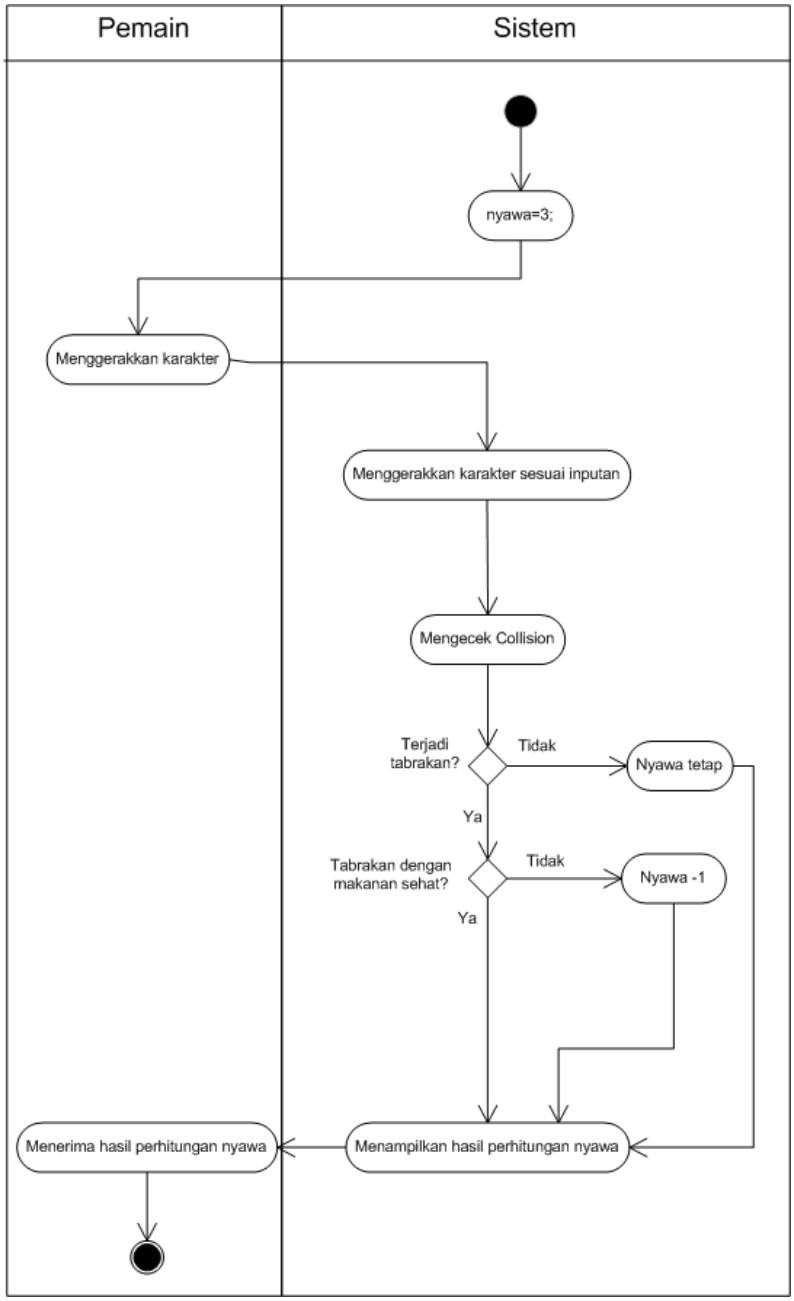
   Activity Diagram Memproses Collision Detection akan dijelaskan pada gamber berikut:



Gambar 3.17 Activity Diagram Memproses Collision Detection

7. Activity Diagram Menghitung Nyawa

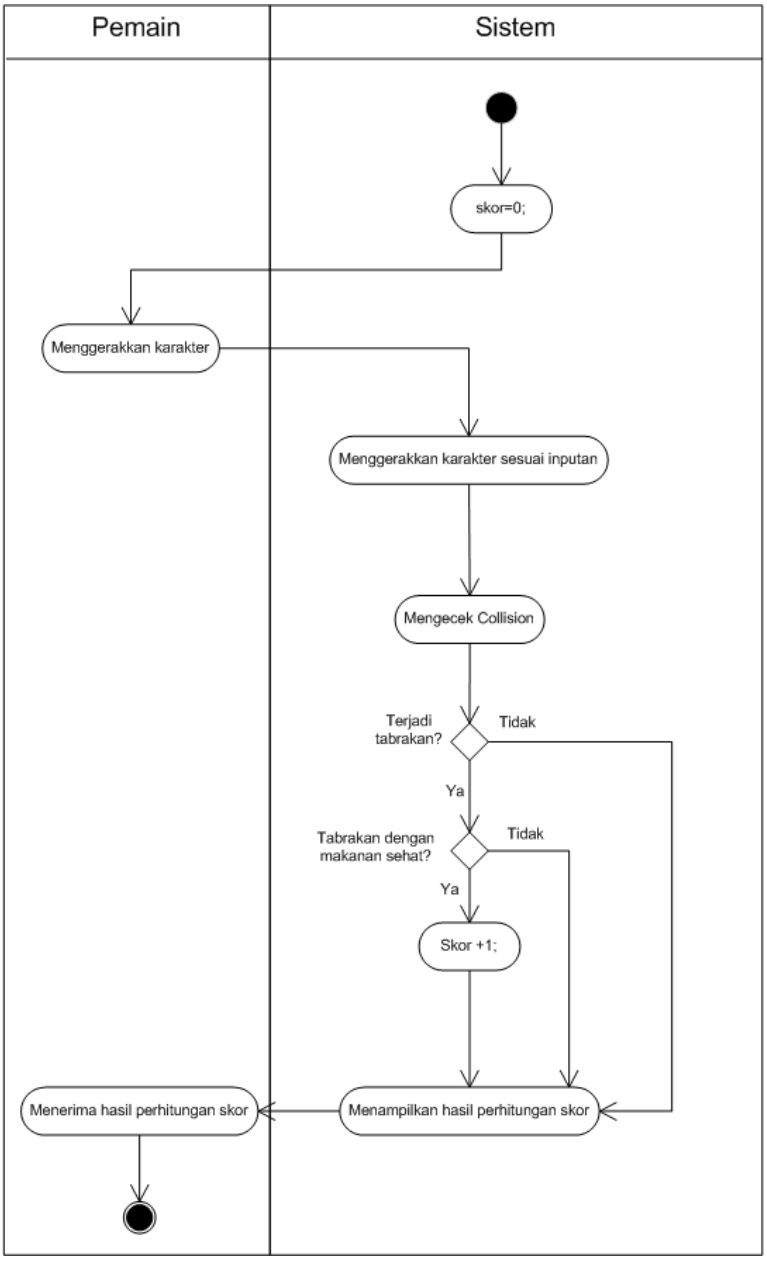
   Activity Diagram Menghitung Nyawa dapat dijelaskan pada gambar berikut:



Gambar 3.18 Activity Diagram Menghitung Nyawa

8. Activity Diagram Menghitung Skor

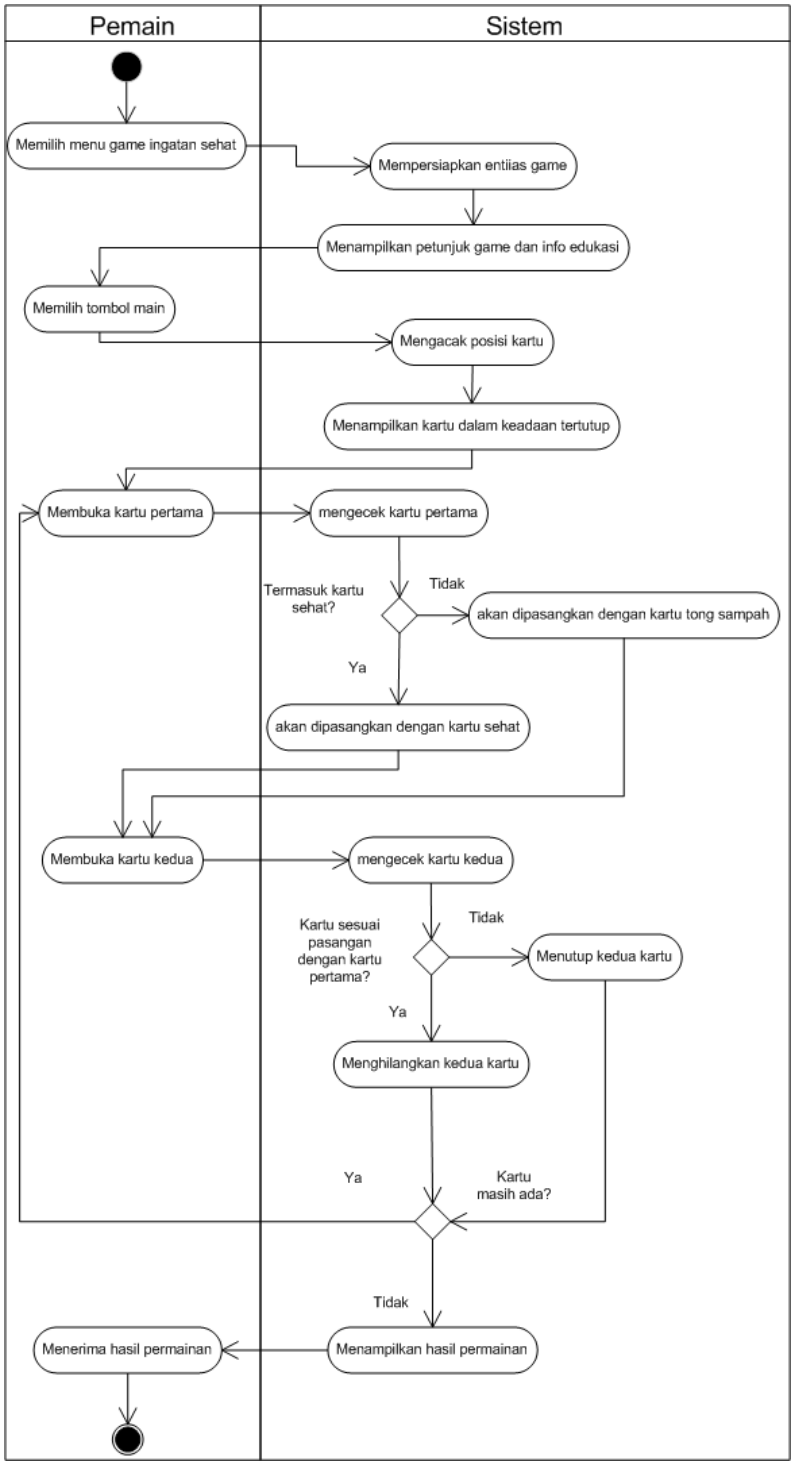
   Activity Diagram Menghitung Skor akan dijelaskan pada gambar berikut:



Gambar 3.19 Activity Diagram Menghitung Skor

9. Activity Diagram Penyajian *Game* Ingatan Sehat

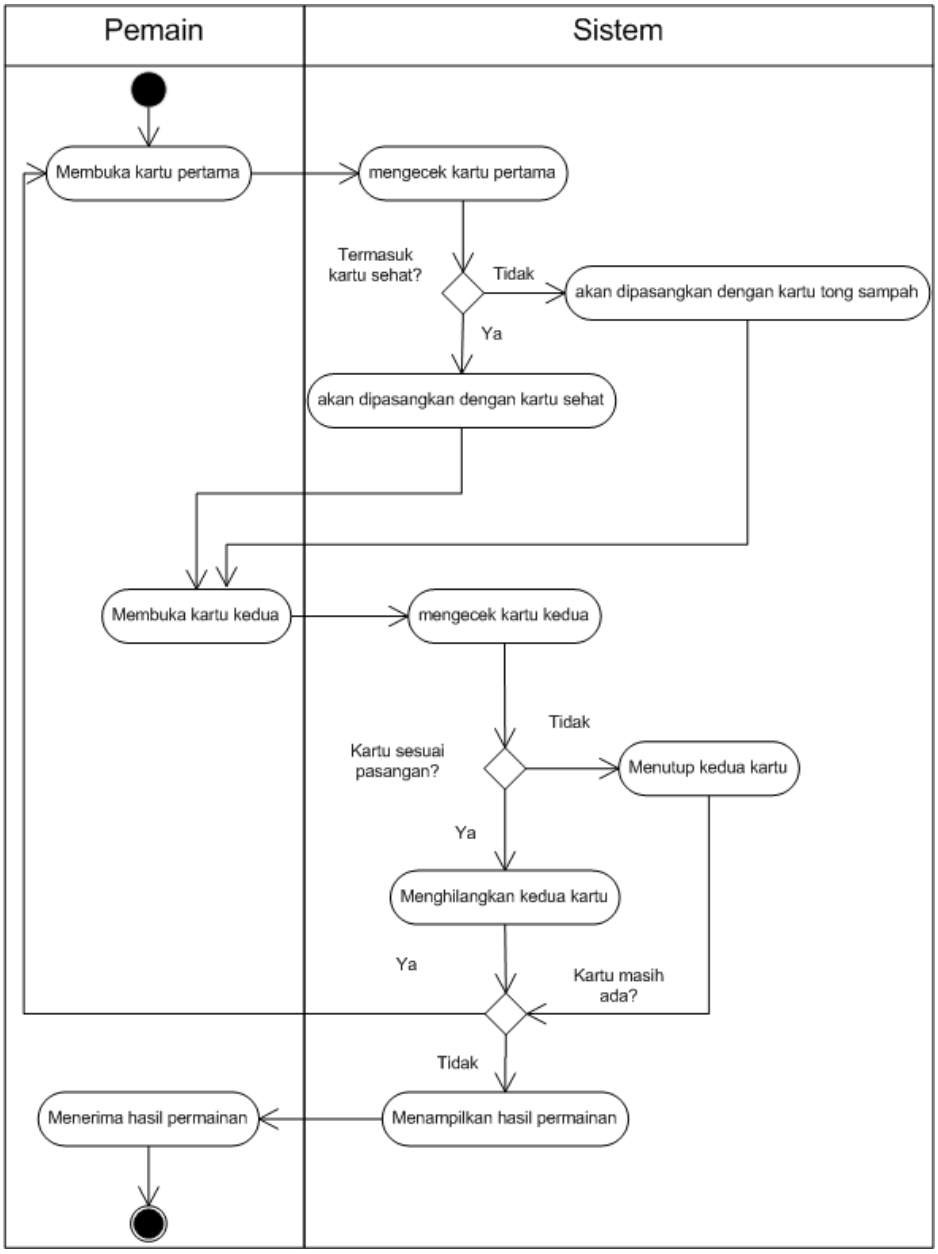
   Activity Diagram Penyajian *Game* Ingatan Sehat akan dijelaskan pada gambar berikut:



Gambar 3.20 Activity Diagram Penyajian *Game* Ingatan Sehat

10. Activity Diagram Memilih Kartu

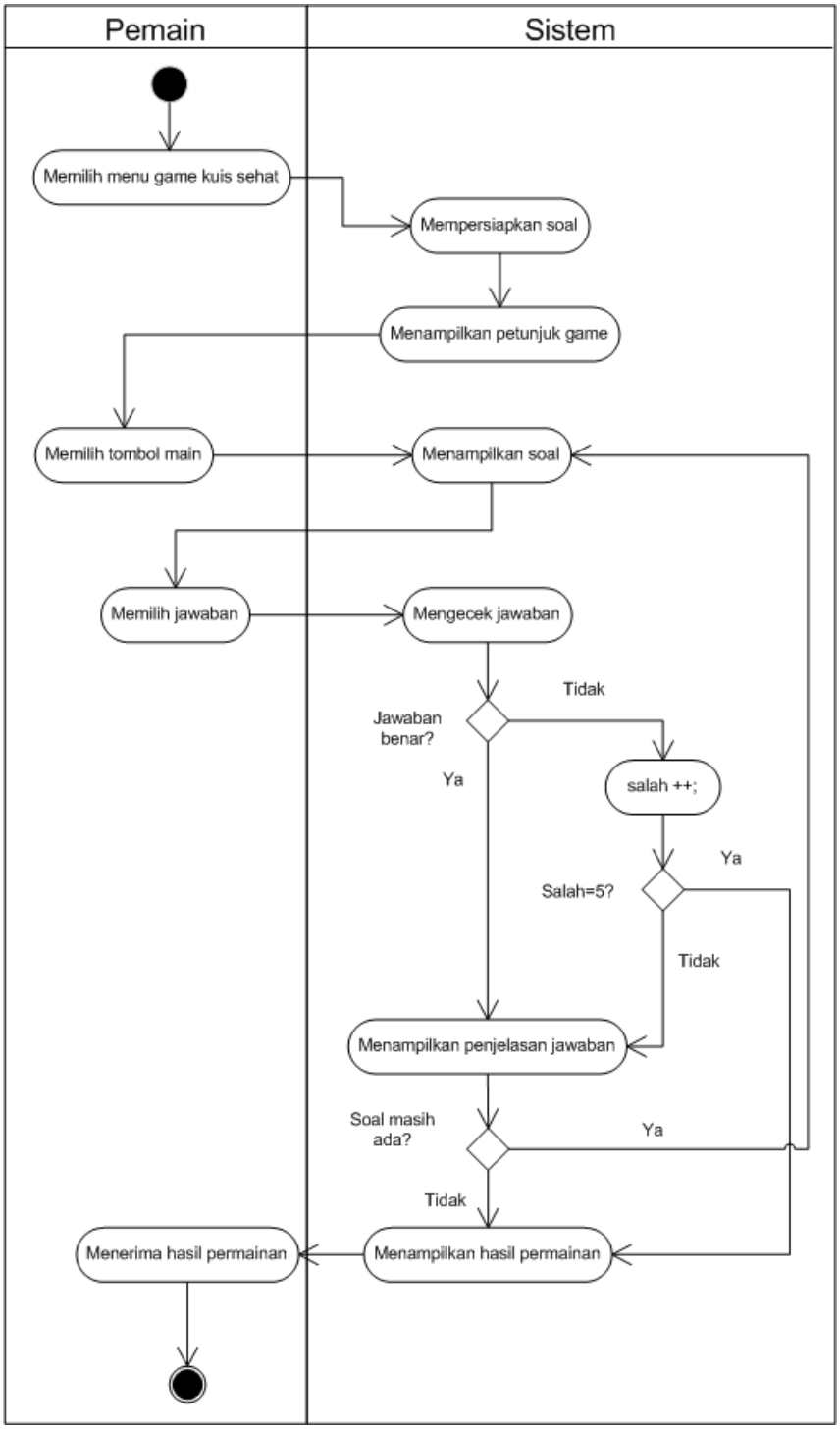
Activity Diagram Memilih Kartu akan dijelaskan pada gambar berikut:



Gambar 3.21 Activity Diagram Memilih Kartu

11. Activity Diagram Penyajian *Game* Kuis Sehat

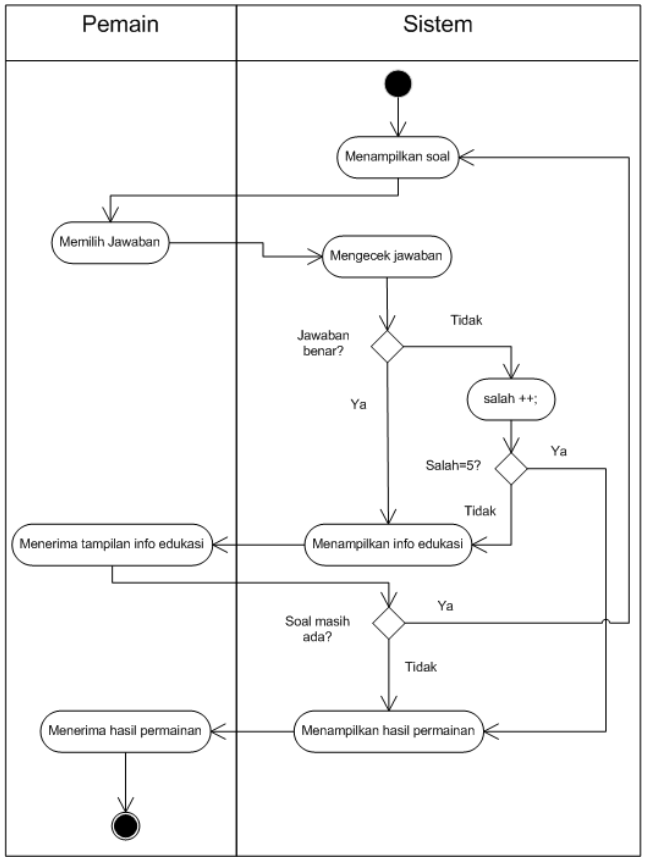
Activity Diagram Penyajian *Game* Kuis Sehat akan dijelaskan pada gambar berikut:



Gambar 3.22 Activity Diagram Penyajian *Game* Kuis Sehat

12. Activity Diagram Menampilkan Info Edukasi

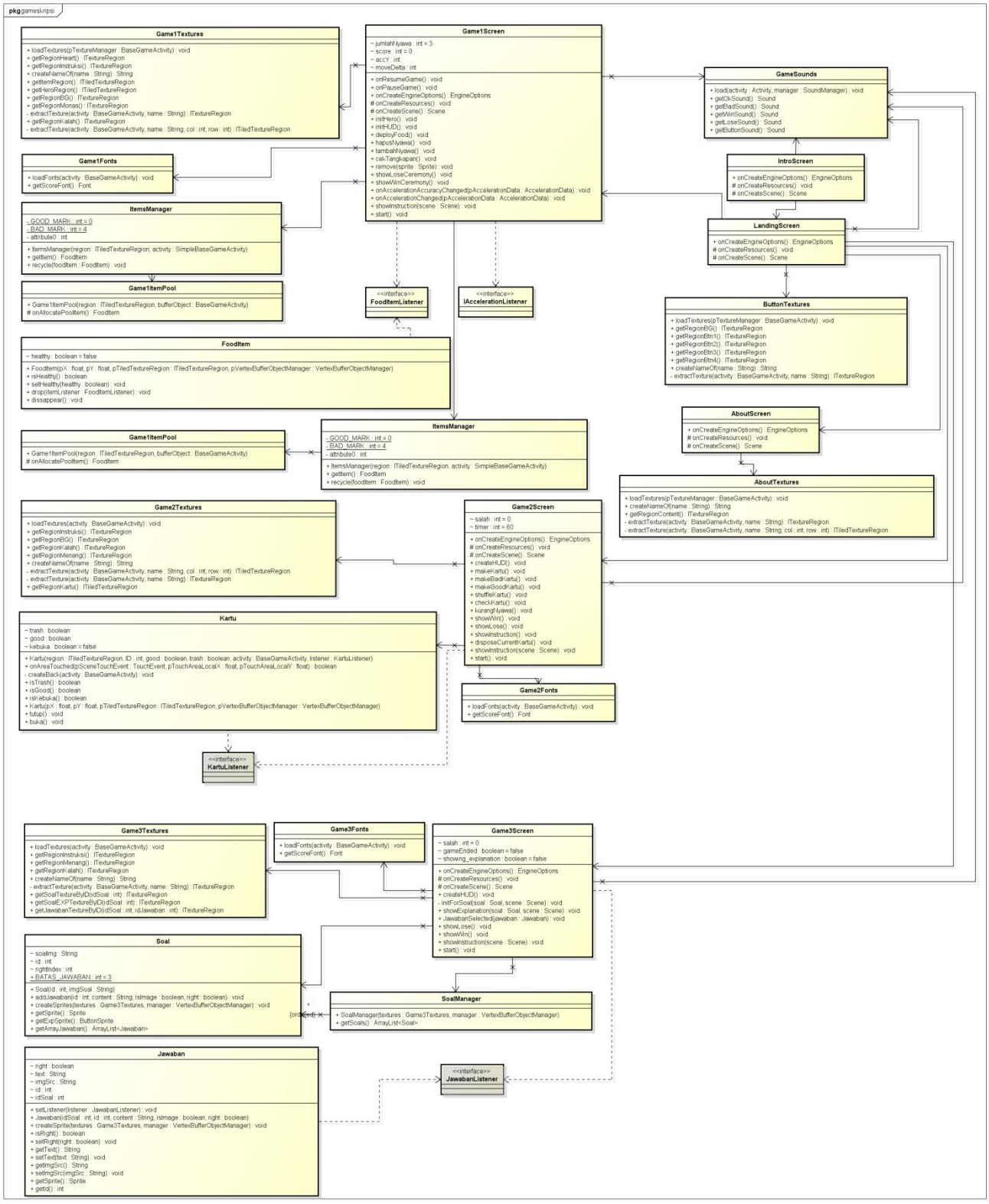
Activity Diagram Menampilkan Info Edukasi akan dijelaskan pada gambar berikut:



Gambar 3.23 Activity Diagram Menampilkan Info Edukasi

3.1.5.2.4 Class Diagram

Class diagram menggambarkan struktur dan hubungan antar *Class* - *Class* yang ada pada sistem. Struktur itu meliputi atribut-atribut dan metode-metode yang ada pada masing-masing *Class*.



Gambar 3.24 Class Diagram

Deskripsi untuk masing-masing class akan dijelaskan lebih lanjut pada tabel-tabel berikut ini:

Tabel 3.30 Deskripsi Class IntroScreen

Class	**IntroScreen**		
Ket.	*Class* untuk menampilkan halaman utama		
Metode	**Nama**	**Sifat**	**Keterangan**
	IntroScreen()	**Public**	Constructor
	onCreateEngineOptions():EngineOptions	Public	Membuat preference untuk engine game
	onCreateResources():void	Protected	Meload resource game
	onCreateScene():Scene	Protected	Membuat scene untuk game

Tabel 3.31 Deskripsi Class LandingScreen

Class	**LandingScreen**		
Ket.	*Class* untuk menampilkan Pilihan Menu		
Metode	**Nama**	**Sifat**	**Keterangan**
	LandingScreen()	**Public**	Constructor
	onCreateEngineOptions():EngineOptions	Public	Membuat preference untuk engine game
	onCreateResources():void	Protected	Meload resource game
	onCreateScene():Scene	Protected	Membuat scene untuk game

Tabel 3.32 Deskripsi Class ButtonTextures

Class	**ButtonTextures**		
Ket.	Class untuk menampilkan tombol pada halaman utama		
Metode	**Nama**	**Sifat**	**Keterangan**
	ButtonTextures()	**Public**	Constructor
	loadTextures(BaseGameActivity):void	Public	Meload texture
	getRegionBtn1():ITextureRegion	Public	mengambil texture untuk button 1
	getRegionBtn2():ITextureRegion	Public	mengambil texture untuk

			button 2
	getRegionBtn3():ITextureRegion	Public	mengambil texture untuk button 3
	getRegionBtn4():ITextureRegion	Public	mengambil texture untuk button 4
	createNameOf(String):String	Public	membuat nama yang sesuai dengan struktur game
	extractTexture(BaseGameActivity,String): ITextureRegion	Privat	mengextract gambar yang di load menjadi textureobjek

Tabel 3.33 Deskripsi Class Game1Screen

Class	**Game1Screen**		
Ket.	Class untuk menampilkan game Tangkapan Sehat		
Metode	**Nama**	**Sifat**	**Keterangan**
	Game1Screen()	**Public**	constructor
	onResumeGame():void	Public	Game resume
	onPauseGame():void	Public	Game pause
	onCreateEngineOptions():EngineOptions	Public	membuat preference untuk engine game
	onCreateResources():void	Protected	meload resource game
	onCreateScene():Scene	Protected	membuat scene
	initHero():void	Public	Inisialisasi hero, membuat, set posisi, gambar, dll
	InitHUD():void	Public	Inisalisasi HUD
	deployFood():void	Public	masukan food ke game
	hapusNyawa():void	Public	Menghapus nyawa
	tambahNyawa():void	Public	Menambah nyawa
	cekTangkapan():void	Public	Mengecek tangkapan
	Remove(Sprite):void	Public	Menghapus sprite dari scene

	showLoseCeremony():void	Public	Memunculkan tampilan kalah
	showWinCeremony():void	Public	Memunculkan tampilan menang
	onAccelerationAccuracyChanged(AccelerationData):void	Public	dipanggil ketika akselerasi handphone berubah
	onAccelerationChanged(AccelerationData):void	Public	dipanggil ketika akselerasi handphone berubah
	showInstruction(Scene):void	Public	Menampilkan instruksi
	Start():void	Public	Memulai game

Tabel 3.34 Deskripsi Class Game1Fonts

Class	**Game1Fonts**		
Ket.	Class untuk menampilkan huruf pada Game Tangkapan Sehat		
Metode	**Nama**	**Sifat**	**Keterangan**
	Game1Fonts()	**Public**	Constructor
	loadFonts(BaseGameActivity):void	Public	load font untuk game
	getScoreFont():Font	Public	mengambil font untuk teks score

Tabel 3.35 Deskripsi Class Game1Textures

Class	**Game1Textures**		
Ket.	Class untuk menampilkan texture pada Game Tangkapan Sehat		
Metode	**Nama**	**Sifat**	**Keterangan**
	Game1Textures()	**Public**	Constructor
	loadTextures(BaseGameActivity):void	Public	meload texture untuk game
	getRegionInstruksi():ITextureRegion	Public	mendapatkan texture untuk sprite instruksi
	createNameOf(String):String	Public	membuat nama file yang sesuai dengan struktur game
	getItemRegion():ITiledTextureRegion	Public	mendapatkan

			texture untuk item makanan
	getHeroRegion():ITiledTextureRegion	Public	mendapatkan texture untuk hero
	extractTexture(BaseGameActivity,String): ITextureRegion	Privat	mengextract texture
	getRegionKalah()ITextureRegion	Public	mendapatkan texture untuk sprite kalah
	extractTexture(BaseGameActivity,String,int,int): ITiledTextureRegion	Privat	mengextract texture

Tabel 3.36 Deskripsi Class Game1ItemPool

Class	**Game1ItemPool**		
Ket.	Class untuk pool item yang akan dipakai dalam game		
Metode	**Nama**	**Sifat**	**Keterangan**
	Game1ItemPool(ItiledTextureRegion,BaseGameActivity)	**Public**	Constructor
	loadFonts(BaseGameActivity):void	Public	load font untuk game
	onAllocatePoolItem():FoodItem	Protected	dipanggil saat membuat item baru

Tabel 3.37 Deskripsi Class ItemsManager

Class	**ItemsManager**		
Ket.	Class untuk memanage item yang dipakai		
Metode	**Nama**	**Sifat**	**Keterangan**
	ItemsManager(ItiledTextureRegion,BaseGameActivity)	**Public**	Constructor
	getItem():FoodItem	Public	mendapat kan item makanan
	Recycle(FoodItem):void	Public	recycle makanan agar bisa digunakan lagi

Tabel 3.38 Deskripsi Class FoodItem

Class	**FoodItem**		
Ket.	Class untuk mengelola karakter Food		
Metode	**Nama**	**Sifat**	**Keterangan**

	FoodItem(float,float,ItiledTextureRegion,VertexBufferObjectManager)	**Public**	Constructor
	isHeathy():boolean	Public	cek apakah makanan termasuk makanan sehat
	setHealthy(boolean):void	Public	set makanan menjadi makanan sehat
	Drop(FoodItemListener):void	Public	masukan makanan ke game
	Dissappear():void	Public	hilangkan makanan dari game

Class	**FoodItemListener**		
Ket.			
Metode	**Nama**	**Sifat**	**Keterangan**
	onDropComplete(FoodItem):void	**Public**	dipanggil saat makanan sudah offscreen
	onCatched(FoodItem):void	Public	dipanggil saat makanan di tangkap

Tabel 3.39 Deskripsi Class Game2Screen

Class	**Game2Screen**		
Ket.	Class untuk menampilkan game Ingatanan Sehat		
Metode	**Nama**	**Sifat**	**Keterangan**
	Game2Screen()	**Public**	Constructor
	onCreateEngineOptions():EngineOptions	Public	
	onCreateResources():void	Protected	load resource game
	onCreateScene():Scene	Protected	membuat scene
	createHUD():void		membuat hud
	makeKartu():void		membuat kartu
	makeBadKartu():void	Public	membuat kartu – kartu tidak sehat
	makeGoodKartu():void	Public	membuat kartu –

			kartu sehat
	shuffleKartu():void	Public	mengacak kartu
	checkKartu():void	Public	mengcek kartu yang di pilih user
	showWin():void	Public	memunculkan tampilan menang
	showLose():void	Public	memunculkan tampilan kalah
	showInstruction():void	Public	memunculkan tampilan instruksi
	disposeCurretKartu():void	Public	menghapus kartu yang di pilih
	showInstruction(Scene):void	Public	memunculkan tampilan instruksi
	Start():void	Public	start game

Tabel 3.40 Deskripsi Class Game2Fonts

Class	**Game2Fonts**		
Ket.	Class untuk menampilkan huruf pada Game Ingatan Sehat		
Metode	**Nama**	**Sifat**	**Keterangan**
	Game2Fonts()	**Public**	constructor
	loadFonts(BaseGameActivity):void	Public	load font
	getScoreFont():Font	Public	get font untuk score

Tabel 3.41 Deskripsi Class Game2Textures

Class	**Game2Textures**		
Ket.	Class untuk menampilkan texture pada Game Ingatan Sehat		
Metode	**Nama**	**Sifat**	**Keterangan**
	Game2Textures()	**Public**	Constructor
	loadTextures(BaseGameActivity):void	Public	Meload texture untuk game
	getRegionInstruksi():ITextureRegion	Public	Mendapatkan texture untuk sprite instruksi
	getRegionKalah():ITextureRegion	Public	Mendapatkan texture untuk sprite kalah

	getRegionMenang():ITextureRegion	Public	Mendapatkan texture untuk sprite menang
	createNameOf(String):String	Public	Membuat nama fike yang sesuai dengan struktur game
	extractTexture(BaseGameActivity,String): ITextureRegion	Privat	Mengextrak texture
	extractTexture(BaseGameActivity,String): ITextureRegion	Privat	Mengextrak texture
	getRegionKartu():TiledTextureRegion		Mendapatkan texture untuk kartu

Tabel 3.42 Deskripsi Class Kartu

Class	**Kartu**		
Ket.	Class untuk mengelola kartu pada game Ingatan Sehat		
Metode	**Nama**	**Sifat**	**Keterangan**
	Kartu(ItiledTextureRegion,int,boolean,boolean,BaseGameActivity,KartuListener)	**Public**	Constructor
	onAreaTouched(TouchEvent,float,float):boolean	Public	dipanggil saat kartu di tap
	CreateBack(baseGameActivity):void	Privat	
	isTrash():boolean	Public	mengecek tipe kartu
	isGood():boolean	Public	cek status kartu apakah sehat
	isKebuka():boolean	Public	cek apakah kartu sudah terbuka
	Kartu(flat,flat,iTiledTextureRegion,VertexBufferObjectManager)	Privat	constructor
	tutup()void	Privat	mentutup kartu
	buka()void		membuka kartu

Tabel 3.43 Deskripsi Class Game3Scren

Class	**Game3Screen**		
Ket.	Class untuk menampilkan game Kuis Sehat		
Metode	**Nama**	**Sifat**	**Keterangan**
	Game3Screen()	**Public**	Constructor

	onCreateEngineOptions():EngineOptions	Public	Membuat preference untuk engine game
	onCreateResources():void	Protected	Meload resource
	onCreateScene():Scene	Protected	Membuat scene
	createHUD():void	Public	Membuat HUD
	initForSoal(Soal,Scene):void	Privat	Inisialisasi soal
	JawabanSelected(Jawaban):void	Public	Mengecek jawaban yang dipilih
	showLose():void	Public	Menampilkan tampilan kalah
	showWin():void	Public	Menampilkan tampilan menang
	showInstruction(Scene):void	Public	Menampilkan instruksi
	start():void	Public	Start game

Tabel 3.44 Deskripsi Class Game3Fonts

Class	**Game3Fonts**		
Ket.	Class untuk menampilkan huruf pada Game Kuis Sehat		
Metode	**Nama**	**Sifat**	**Keterangan**
	Game3Fonts()	**Public**	Constructor
	loadFonts(BaseGameActivity):void	Public	Meload font untuk game
	getScoreFont():Font	Public	Mengambil font untuk teks score

Tabel 3.45 Deskripsi Class Game3Textures

Class	**Game3Textures**		
Ket.	Class untuk menampilkan texture pada game Kuis Sehat		
Metode	**Nama**	**Sifat**	**Keterangan**
	Game3Textures()	**Public**	Constructor
	loadTextures(BaseGameActivity):void	Public	Meload texture untuk game
	getRegionInstruksi():ITextureRegion	Public	Mendapatkan texture untuk sprite instruksi
	getRegionKalah():ITextureRegion	Public	Mendapatkan texture untuk

			sprite kalah
	getRegionMenang():ITextureRegion	Public	Mendapatkan texture untuk sprite menang
	createNameOf(String):String	Public	Membuat nama file yang sesuai dengan struktur game
	extractTexture(BaseGameActivity,String): ITextureRegion	Privat	Mengextrak texture
	getSoalTexture(baseGameActivity,String):ItextureRegion	Public	Mendapatkan texture untuk soal
	GetJawabanTextureByID(int,int):ITextureRegion	Public	Mendapatkan texture untuk jawaban

Tabel 3.46 Deskripsi Class Soal

Class	**Soal**		
Ket.	Class untuk menampilkan soal pada game Kuis Sehat		
Metode	**Nama**	**Sifat**	**Keterangan**
	Soal(int,String)	**Public**	Constructor
	addJawaban(int,String,boolean,boolean):void	Public	menambahkan jawaban ke soal
	createSprites(Game3Textures,VertexBufferObjectManager):void	Privat	membuat sprite untuk soal
	getSprite():Sprite	Public	mendapatkan sprite untuk soal
	getArrayJawaban():ArrayList<Jawaban>	Public	mendapatkan list jawaban dari soal

Tabel 3.47 Deskripsi Class SaolManager

Class	**SoalManager**		
Ket.	Class untuk mengelola soal pada game Kuis Sehat		
Metode	**Nama**	**Sifat**	**Keterangan**
	SoalManager(Game3Textures,VertexBufferObjectManager)	**Public**	Constructor
	getSoals():ArrayList<Soal>	Public	mendapatkan list soal

Tabel 3.48 Deskripsi Class Jawaban

Class	**Jawaban**		
Ket.	Class untuk menampilkan jawaban pada game Kuis Sehat		
Metode	**Nama**	**Sifat**	**Keterangan**
	setListener(JawabanListener):void	**Public**	mengeset interface ke class
	Jawaban(int,int,String,boolean,boolean)	Public	constructor
	createSprites(Game3Textures,VertexBufferObjectManager):void	Public	membuat sprite untuk jawaban
	isRight():String	Public	mengecek jawaban menjadi jawaban benar
	setRight(boolean):void	Public	mengeset jawaban menjadi jawaban benar
	getText():String	Public	mendapatkan text jawaban
	setText(String):void	Public	mengeset text jawaban
	getImgSrc():String	Public	mendapatkan nama gambar untuk jawaban
	setImgSrc():void	Public	mengeset nama gambar untuk jawaban
	getSprite():Sprite	Public	mendapatkan sprite untuk jawaban
	getId():int	Public	Mendapatkan id untuk jawaban

Tabel 3.49 Deskripsi Class AboutScreen

Class	**AboutScreen**		
Ket.	Class untuk menampilkan halaman Tentang Saya		
Metode	**Nama**	**Sifat**	**Keterangan**
	AboutScreen()	**Public**	Constructor
	onCreateEngineOptions():EngineOptions	Public	Membuat preference untuk

			engine game
	onCreateResources():void	Protected	Meload resource game
	onCreateScene():Scene	Protected	Membuat scene

Tabel 3.50 Deskripsi Class AboutTextures

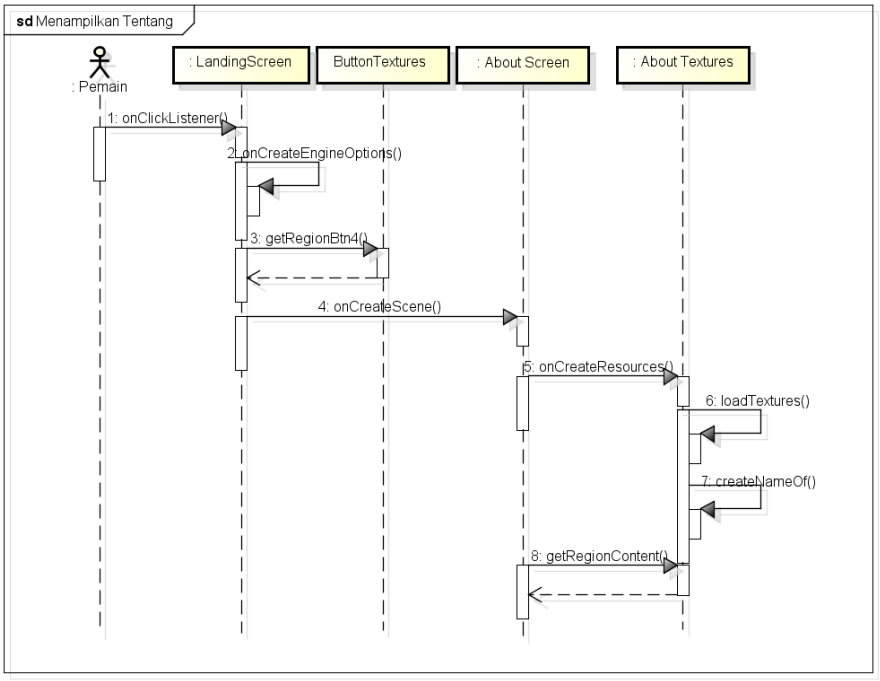
Class	**AboutTextures**		
Ket.	Class untuk menampilkan texture pada halaman Tentang Saya		
Metode	**Nama**	**Sifat**	**Keterangan**
	AboutTextures()	**Public**	Constructor
	loadTextures(BaseGameActivity):void	Public	Meload texture untuk game
	createNameOf(String):String	Public	Membuat nama file yang sesuai dengan struktur game
	getRegionContent():ITextureRegion	Public	Mendapatkan texture untuk konten halaman
	extractTexture(BaseGameActivity,String): ITextureRegion	Privat	Mengextrak texture
	extractTexture(BaseGameActivity,String,int,int):ITiledTextureRegion	Privat	Mengextrak texture

3.1.5.2.5 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.

1. **Sequence Diagram Menampilkan Tentang**

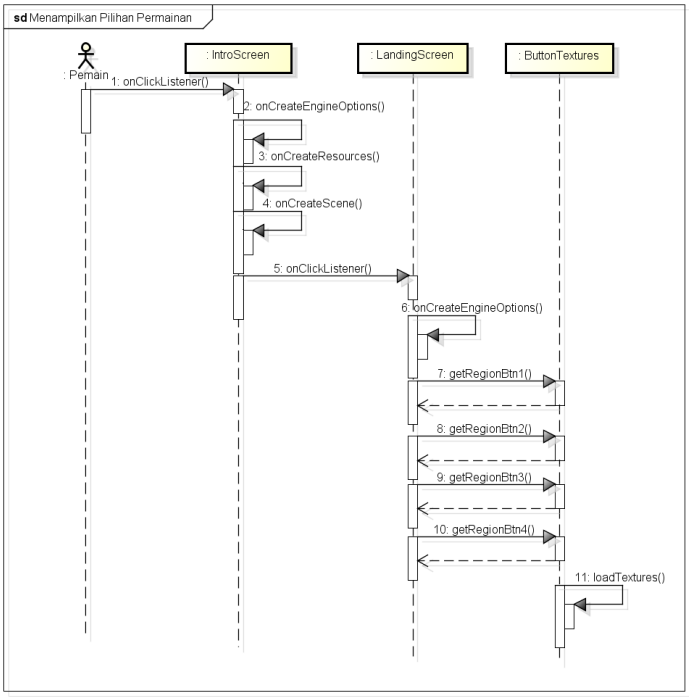
   Sequence Diagram Menampilkan Tentang Saya akan dijelaskan pada gambar berikut:



Gambar 3.25 Sequence Diagram Menampilkan Tentang Saya

2. Sequence Diagram Menampilkan Pilihan Permainan

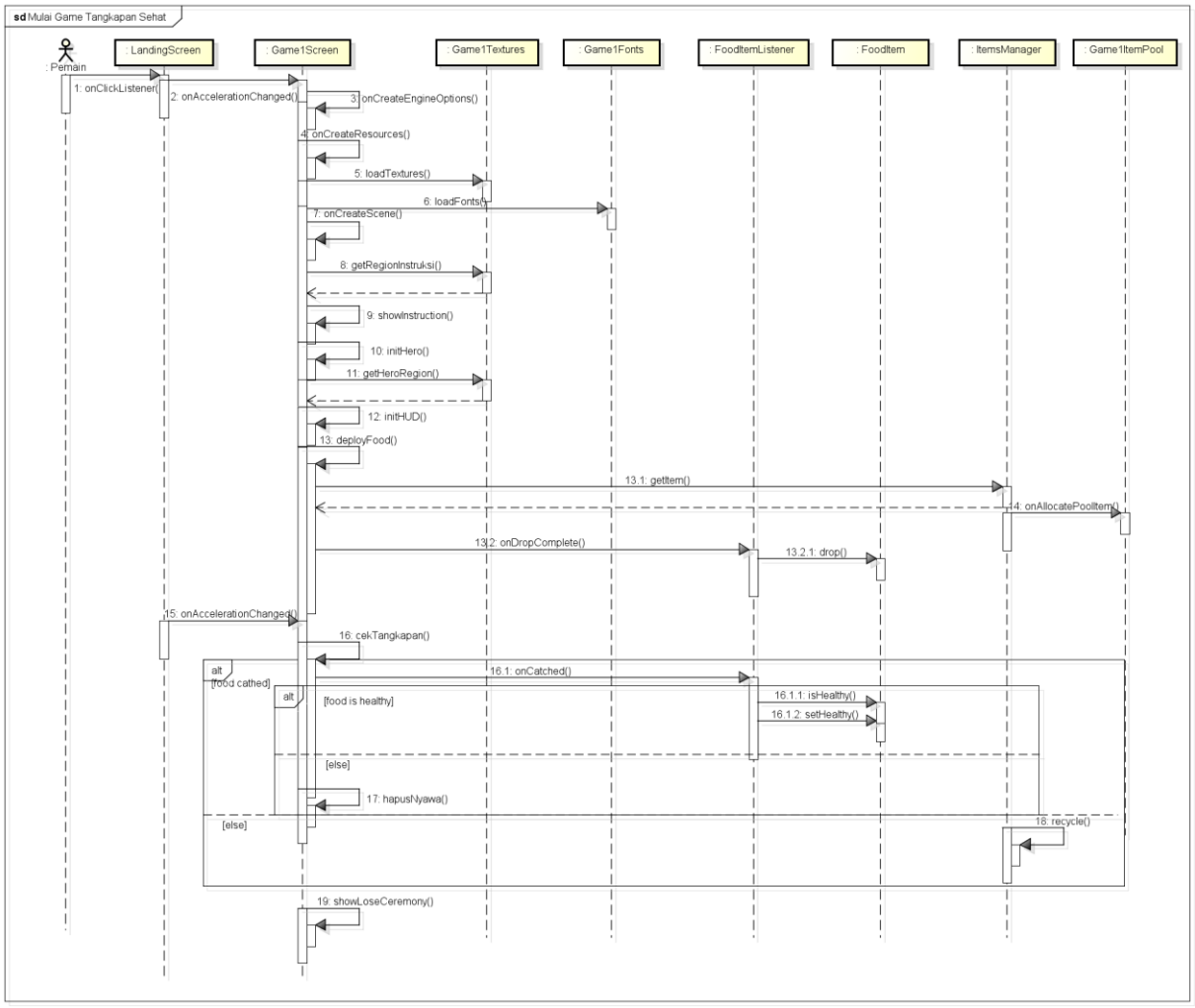
Sequence Diagram Menampilkan Pilihan Permainan akan dijelaskan pada gambar berikut:



Gambar 3.26 Sequence Diagram Menampilkan Pilihan Permainan

3. Sequence Diagram Penyajian Game Tangkapan Sehat

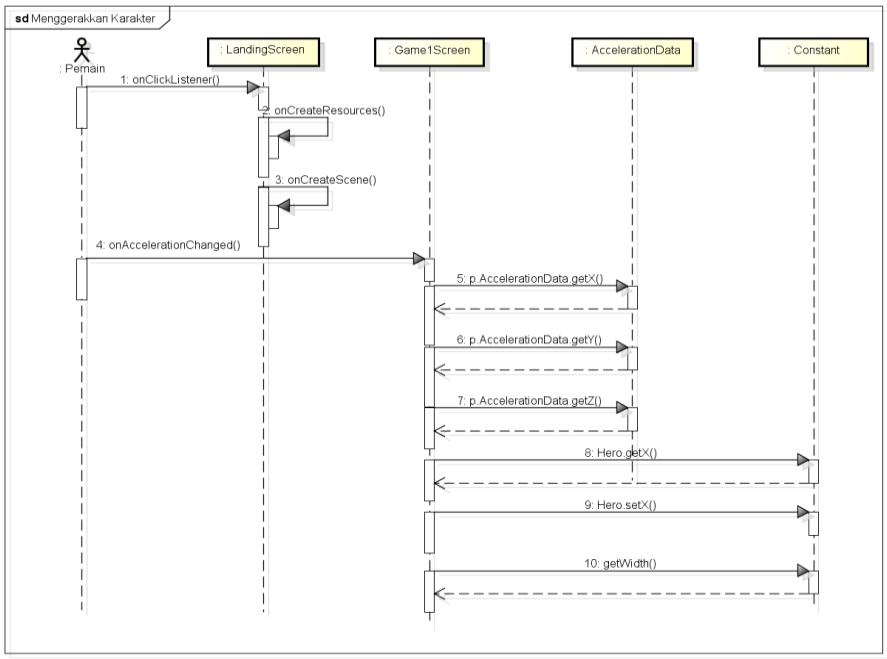
Sequence Diagram Penyajian Game Tangkapan Sehat akan dijelaskan pada gambar berikut:



Gambar 3.27 Sequence Diagram Penyajian Game Tangkapan Sehat

4. Sequence Diagram Menggerakkan Karakter

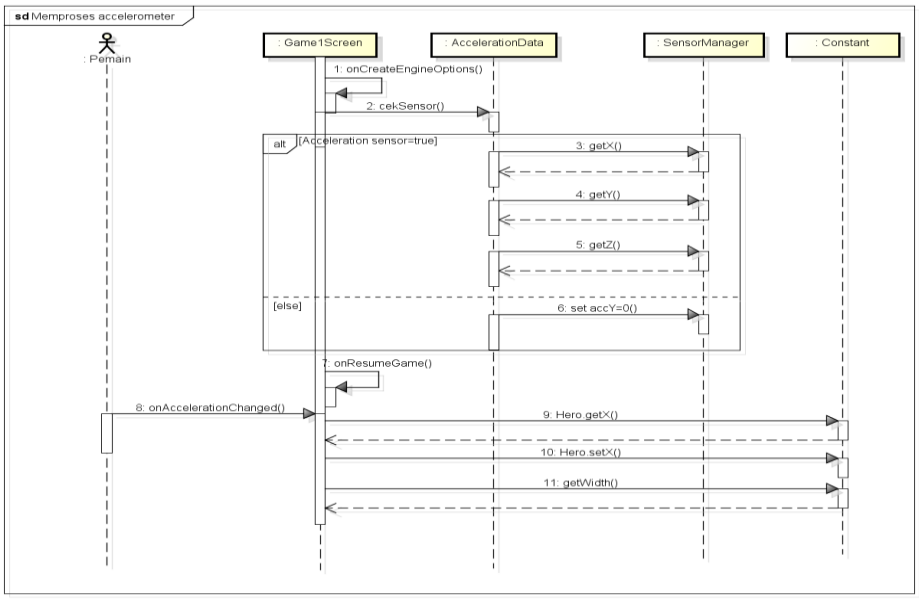
Sequence Diagram Menggerakkan Karakter akan dijelaskan pada gambar berikut:



Gambar 3.28 Sequence Diagram Menggerakkan Karakter

5. Sequence Diagram Memproses Accelerometer

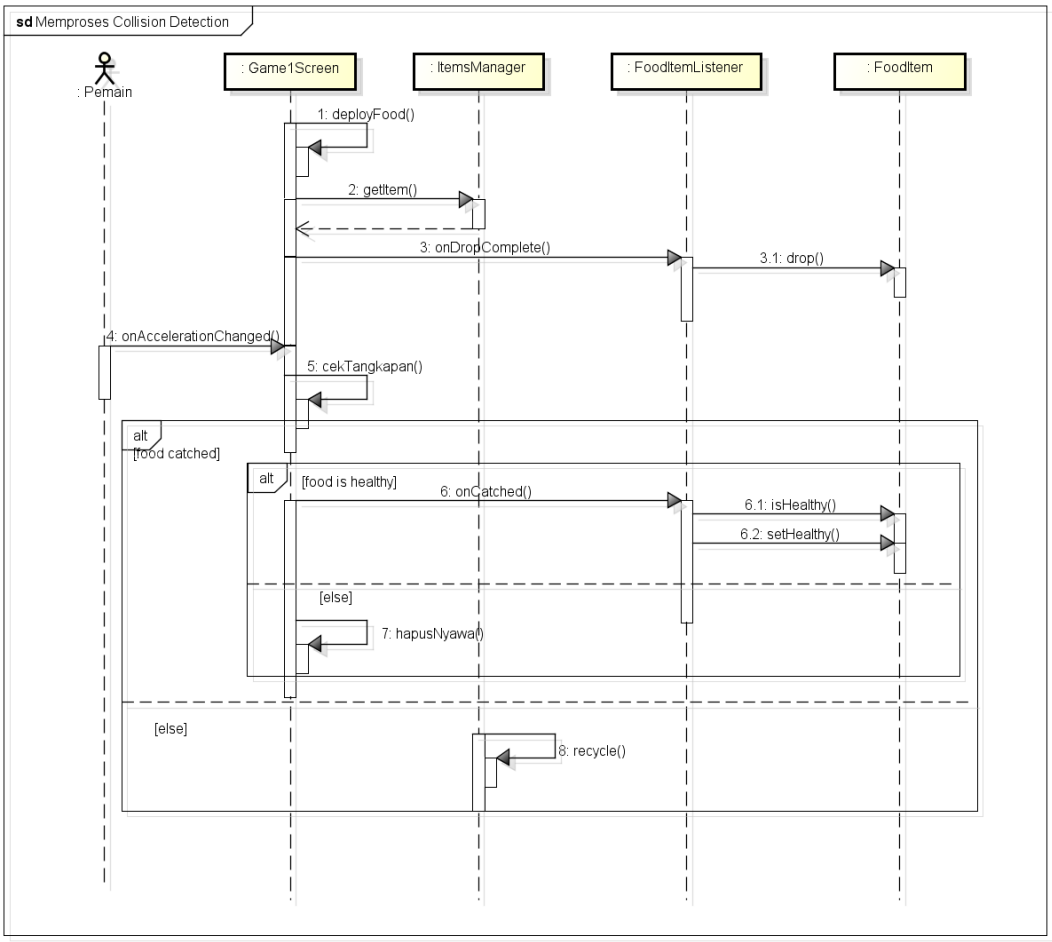
Sequence Diagram Memproses Accelerometer akan dijelaskan pada gambar berikut:



Gambar 3.29 Sequence Diagram Memproses Accelerometer

6. Sequence Diagram Memproses Collision Detection

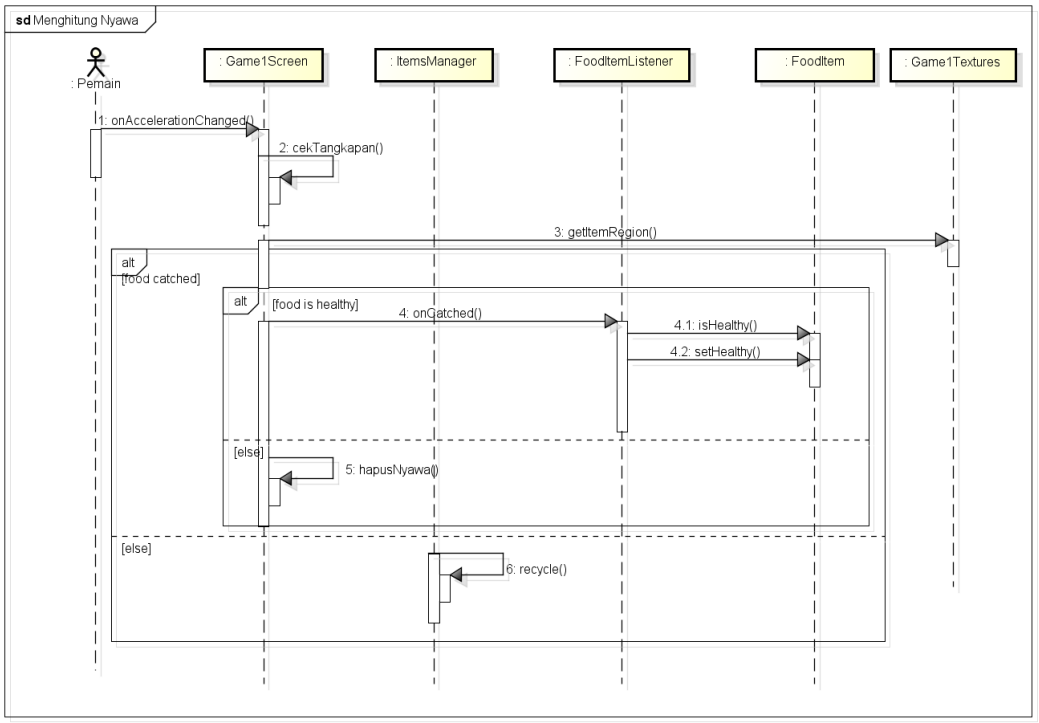
Sequence Diagram Memproses Collision Detection akan dijelaskan pada gambar berikut:



Gambar 3.30 Sequence Diagram Memproses Collision Detection

7. Sequence Diagram Menghitung Nyawa

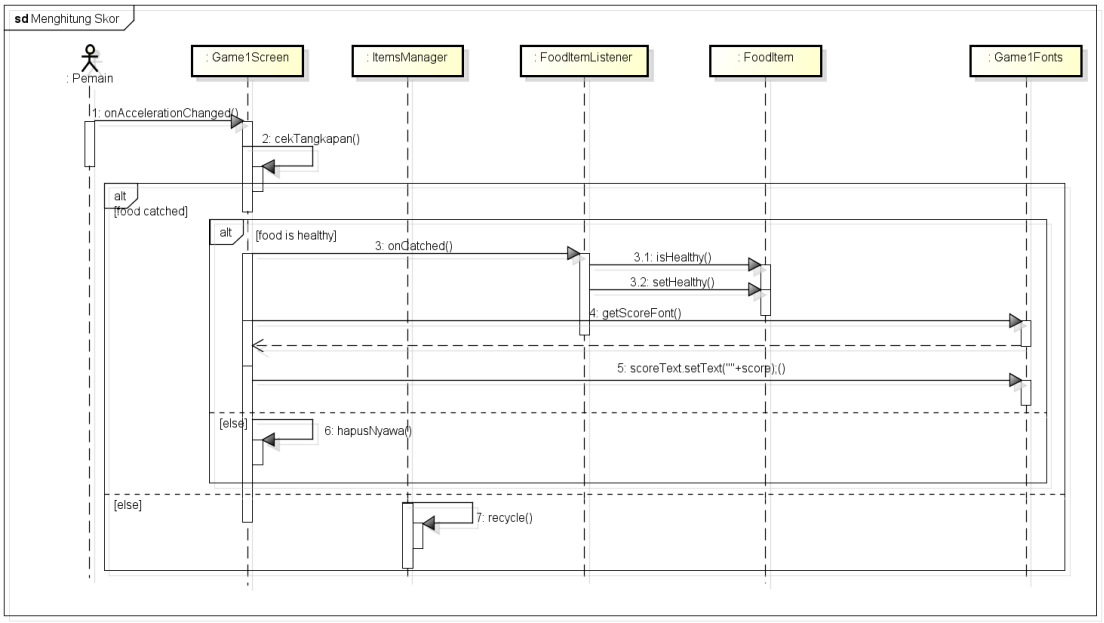
Sequence Diagram Menghitung Nyawa akan dijelaskan pada gambar berikut:



Gambar 3.31 Sequence Diagram Menghitung Nyawa

8. Sequence Diagram Menghitung Skor

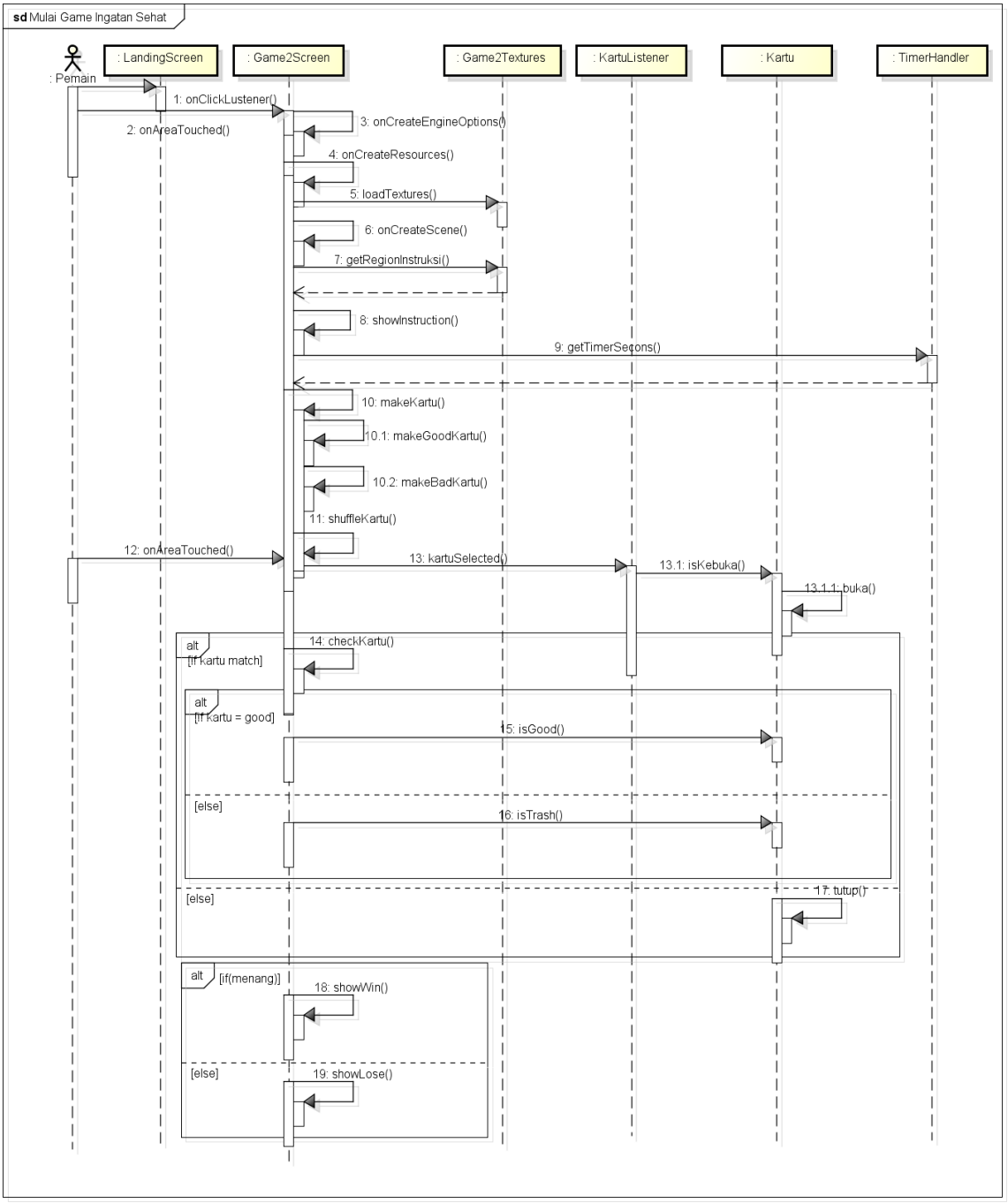
Sequence Diagram Menghitung Skor akan dijelaskan pada gambar berikut:



Gambar 3.32 Sequence Diagram Menghitung Skor

9. Sequence Diagram Penyajian Game Ingatan Sehat

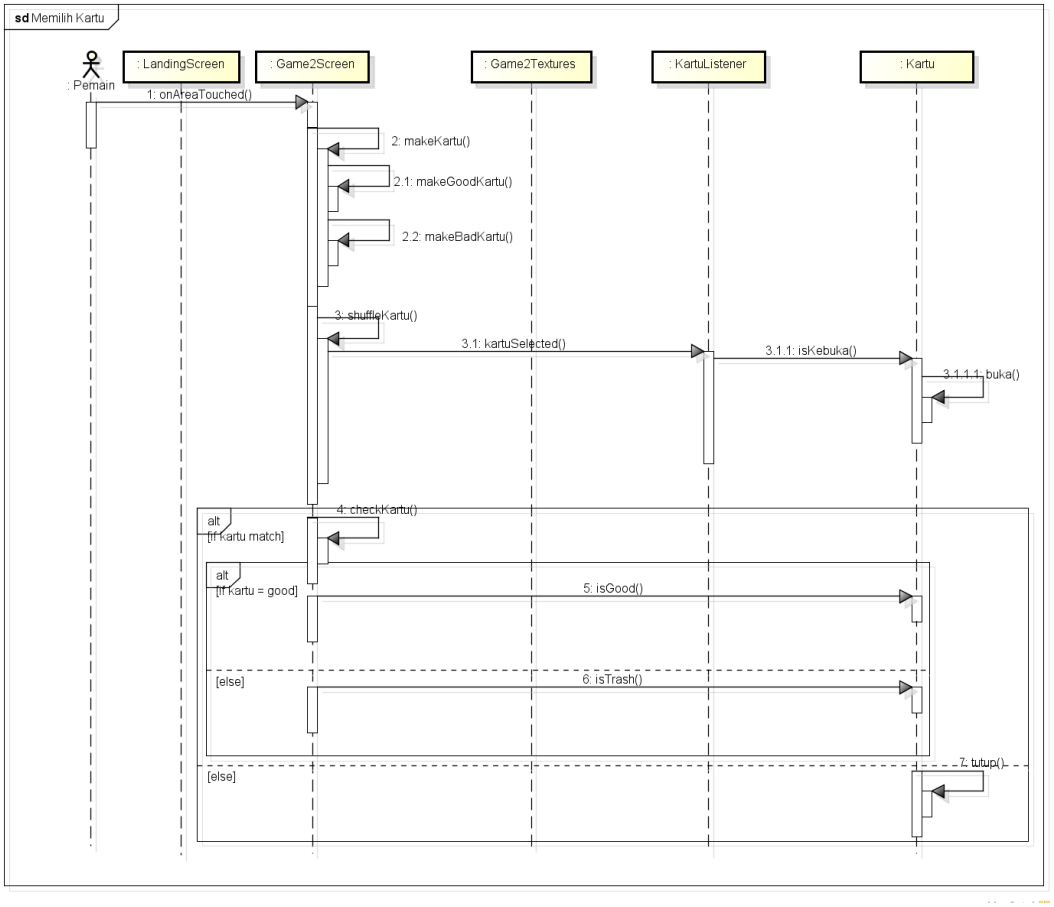
Sequence Diagram Penyajian Game Ingatan Sehat akan dijelaskan pada gambar berikut:



Gambar 3.33 Sequence Diagram Penyajian Game Ingatan Sehat

10. Sequence Diagram Memilih Kartu

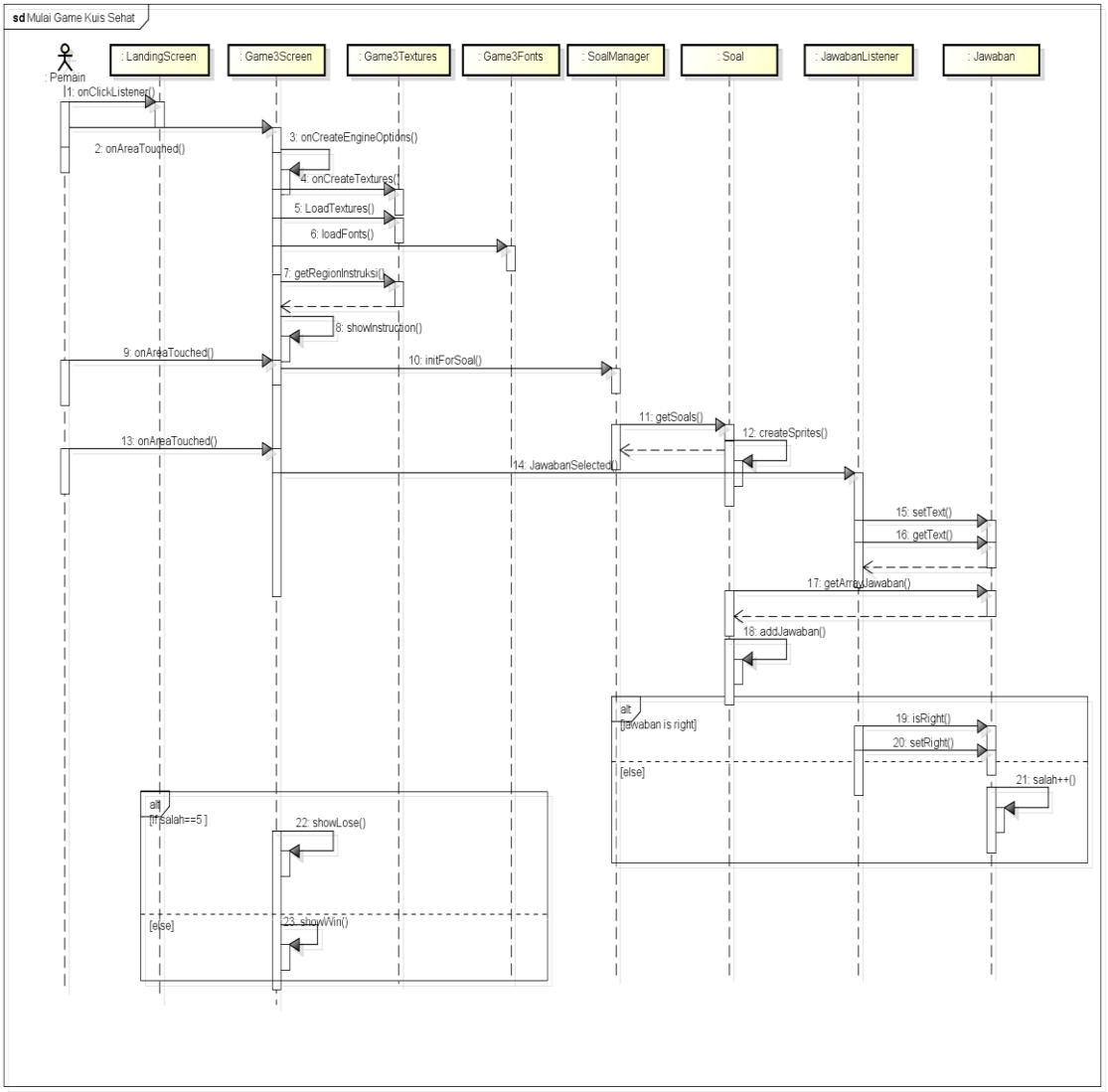
Sequence Diagram Memilih Kartu akan dijelaskan pada gambar berikut:



Gambar 3.34 Sequence Diagram Memilih Kartu

11. Sequence Diagram Penyajian Game Kuis Sehat

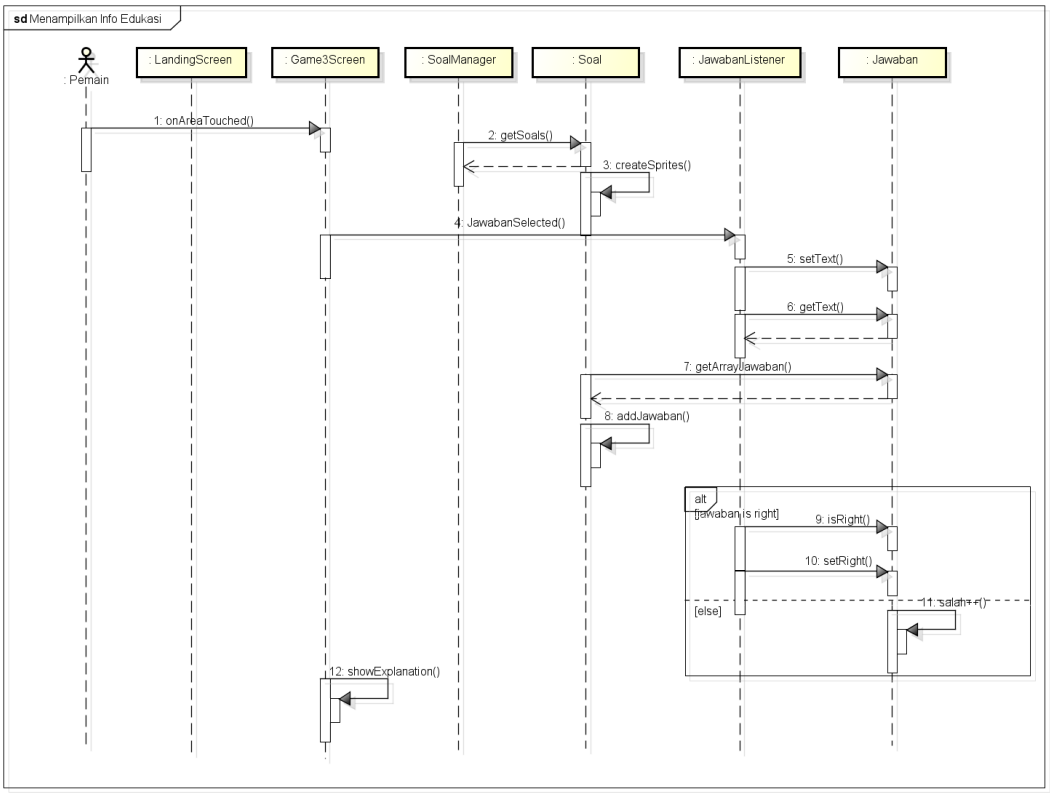
Sequence Diagram Penyajian Game Kuis Sehat akan dijelaskan pada gambar berikut:



Gambar 3.35 Sequence Diagram Penyajian Game Kuis Sehat

12. Sequence Diagram Menampilkan Info Edukasi

Sequence Diagram Menampilkan Info Edukasi akan dijelaskan pada gambar berikut:

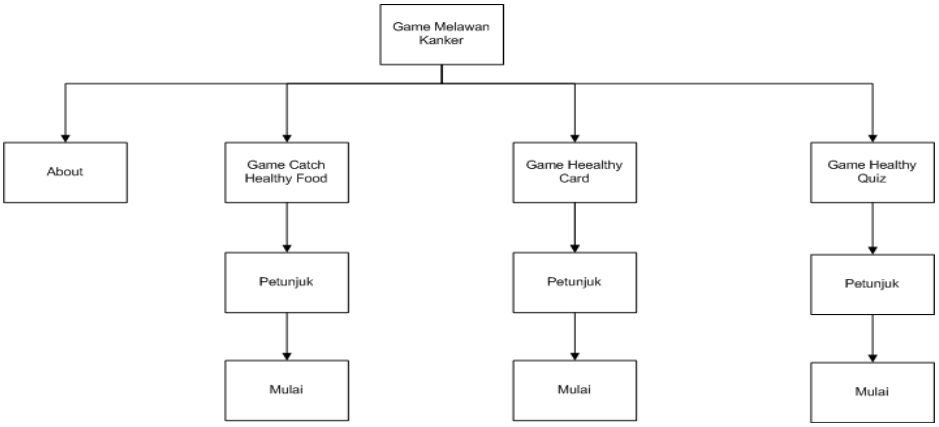


Gambar 3.36 Sequence Diagram Menampilkan Info Edukasi

3.2 Perancangan Sistem

3.2.1 Struktur Menu

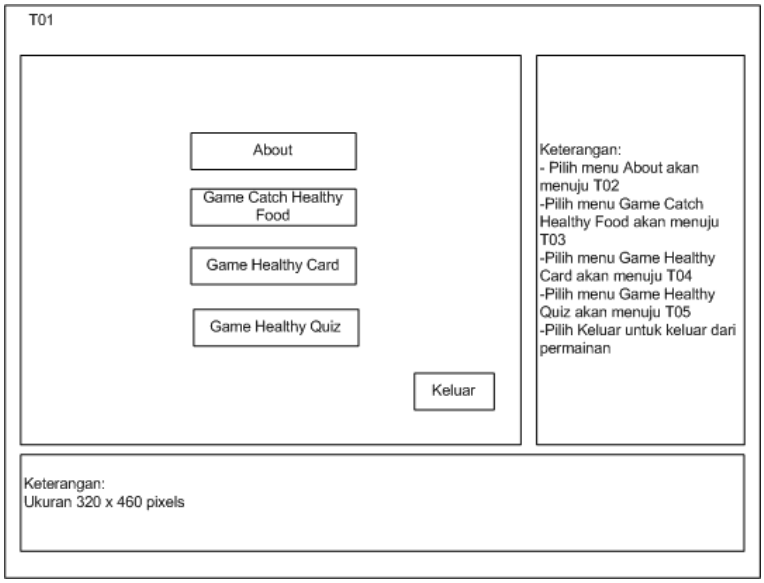
Perancangan struktur menu pada aplikasi *game* melawan kanker dapat dilihat pada gambar berikut:



Gambar 3.37 Struktur Menu

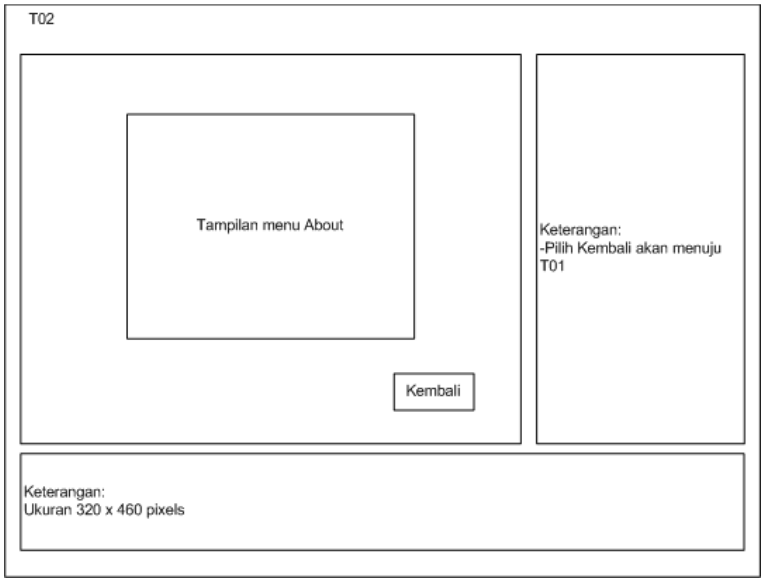
3.2.2 Perancangan Antar Muka

3.2.2.1 Perancangan Antar Muka Menu Utama



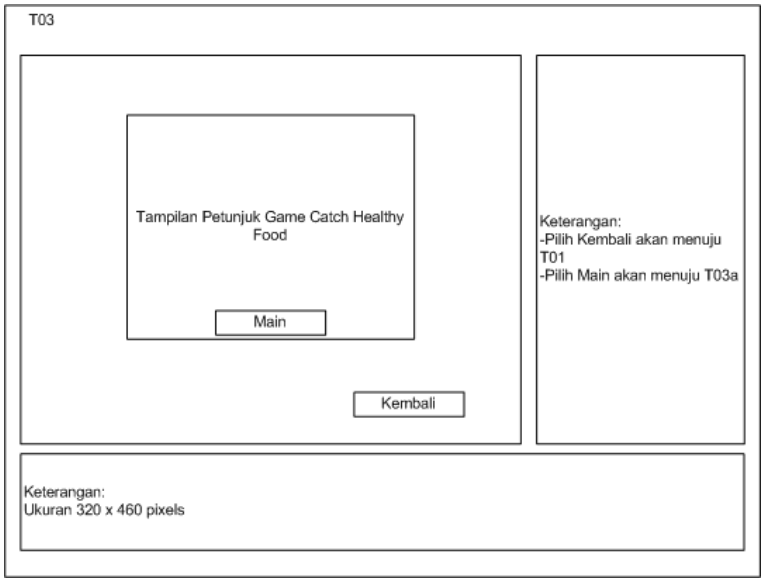
Gambar 3.38 Perancangan Antar Muka Menu Utama

3.2.2.2 Perancangan Antar Muka *About*



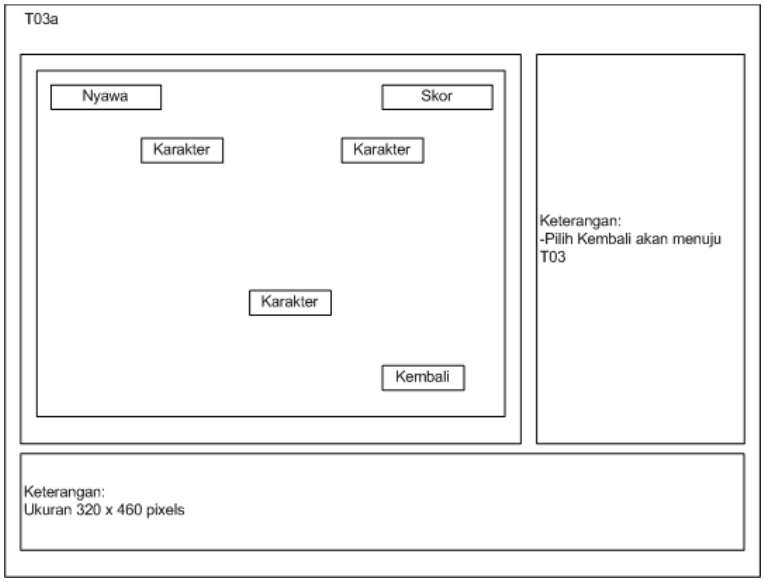
Gambar 3.39 Perancangan Antar Muka *About*

3.2.2.3 Perancangan Antar Muka Petunjuk *Game Tangkapan sehat*



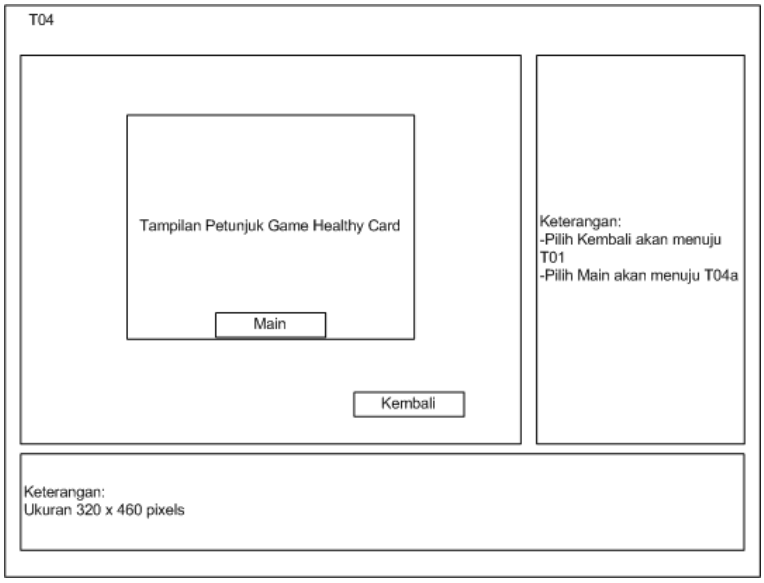
Gambar 3.40 Perancangan Antar Muka Petunjuk *Game Tangkapan sehat*

3.2.2.4 Perancangan Antar Muka Mulai *Game Tangkapan sehat*



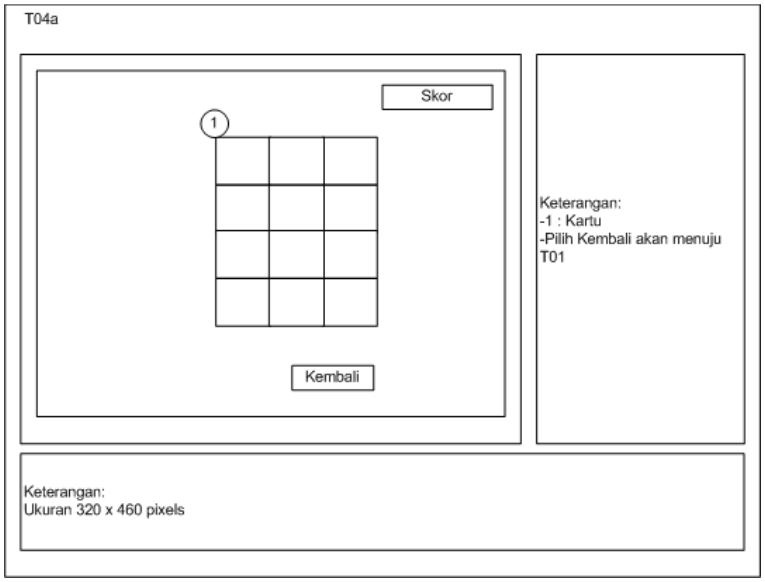
Gambar 3.41 Perancangan Antar Muka Mulai *Game Tangkapan sehat*

3.2.2.5 Perancangan Antar Muka Petunjuk *Game Ingatan Sehat*



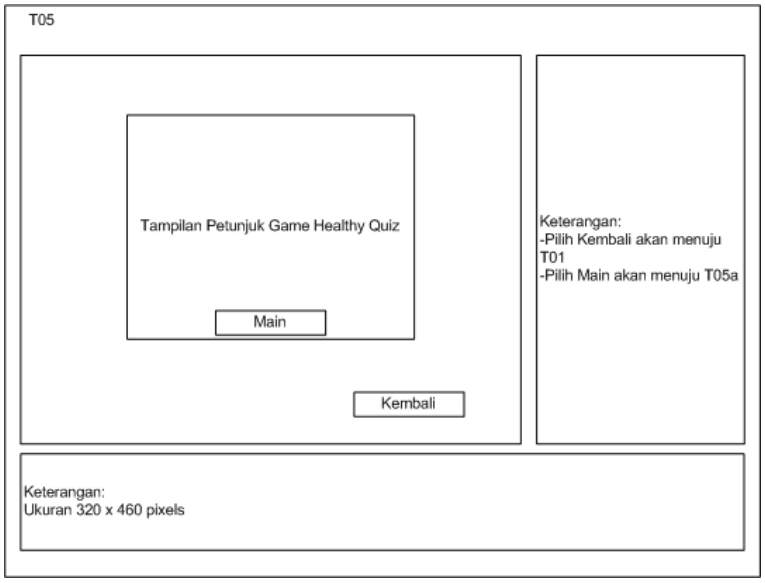
Gambar 3.42 Perancangan Antar Muka Petunjuk *Game Ingatan Sehat*

3.2.2.6 Perancangan Antar Muka Mulai *Game Ingatan Sehat*



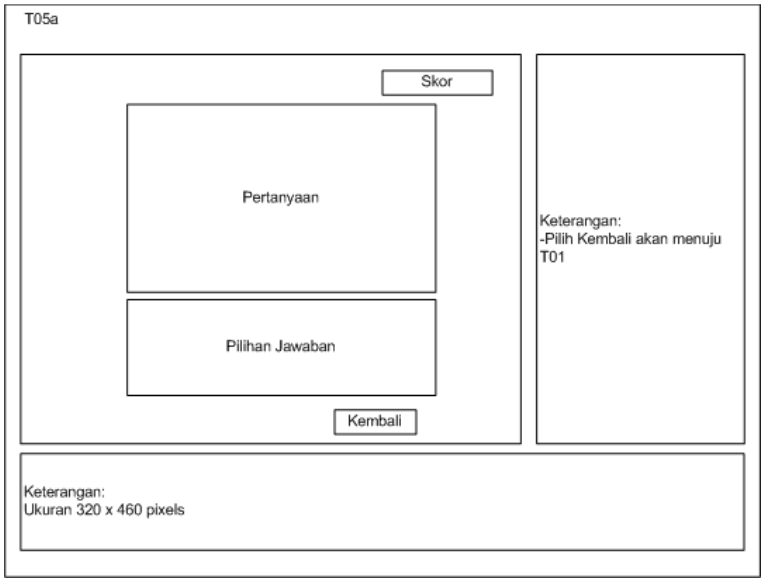
Gambar 3.43 Perancangan Antar Muka Mulai *Game Ingatan Sehat*

3.2.2.7 Perancangan Antar Muka Petunjuk *Game Kuis Sehat*



Gambar 3.44 Perancangan Antar Muka Petunjuk *Game Kuis Sehat*

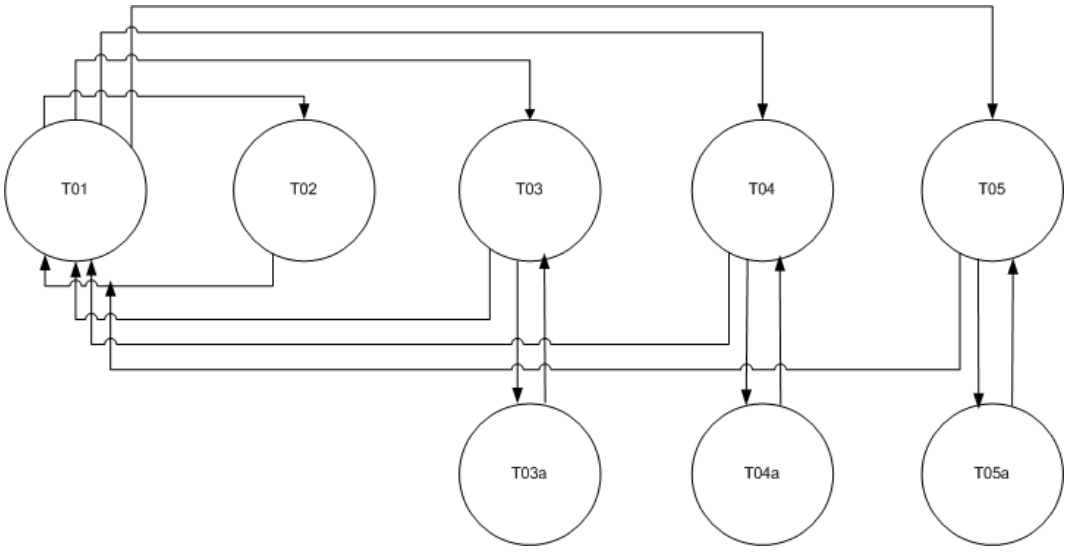
3.2.2.8 Perancangan Antar Muka Mulai *Game Kuis Sehat*



Gambar 3.45 Perancangan Antar Muka Mulai *Game Kuis Sehat*

3.2.3 Jaringan Semantik

Setelah melakukan perancangan antar muka maka dilakukanlah perancangan terhadap hubungan antar *form*. Hubungan antar form tersebut dapat digambarkan dalam sebuah jaringan semantik.



Gambar 3.46 Jaringan Semantik