BAB III
ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Pada analisis sistem akan menguraikan suatu sistem secara utuh beserta komponen - komponennya mencakup identifikasi dan evaluasi dari permasalahan, juga kebutuhan yang diperlukan sehingga hubungan antar unsur – unsurnya menjadi jelas serta dapat diusulkan perbaikannya. Bagian ini meliputi analisis masalah, deskripsi sistem, analisis non fungsional dan analisis fungsional.

3.1.1 Analisis Masalah

Penyakit Diare masih merupakan masalah kesehatan masyarakat di Indonesia, karena angka kesakitan dan kematian yang masih tinggi. Diare adalah suatu kondisi dimana seseorang buang air besar dengan konsistensi lembek atau cair, bahkan dapat berupa air saja dan frekuansinya lebih sering (biasanya tiga kali atau lebih) dalam satu hari. Secara klinis penyebab diare dapat dikelompokan dalam 6 golongan besar yaitu infeksi (disebabkan oleh bakteri, virus atau infestasi parasit), malabsorpsi, alergi, keracunan, imunodefisiensi dan sebab-sebab lainnya. Penyebab yang sering ditemukan dilapangan ataupun secara klinis adalah diare yang disebabkan infeksi dan keracunan [1].

Sejauh ini telah dilakukan sosialiasi mengenai penyakit diare yang dilakukan oleh Lembaga Sekolah bekerja sama dengan Dinas Kesehatan Kota Cimahi seperti pemberian brosur, spanduk, dan poster. Walaupun sudah ada sosialiasi pencegahan diare namun dirasa kurang efektif. Karenanya anak-anak cenderung lebih tertarik terhadap animasi, suara, gambar dan video dibanding dengan tulisan-tulisan saja dan dari sisi psikologi anak-anak masih suka bermain dibandingkan membaca atau belajar. Oleh sebab itu maka dibuatlah penyuluhan dalam bentuk game yang diharapkan anak-anak lebih sadar akan cara pencegahan penyakit diare.

3.1.2 Deskripsi Sistem

Game yang dibangun berhubungan dengan penyakit diare yang diberi judul Lawan Diare, dimana dalam game tersebut menyisipkan unsur materi mengenai pencegahan dan pertolongan pertama terhadap masalah penyakit diare. Dikemas dalam bentuk yang menarik dengan harapan dapat menghibur pemain sekaligus memberikan informasi tentang penyakit diare. Misi dalam *game* ini awal mulanya karakter dalam permainan yang bernama Diki harus mempersiapkan berbagai jenis makanan dan minuman yang sehat untuk perbekalan, melewati rintangan dalam perjalanan ke sekolah dengan berbagai variasi halangan dan rintangan yang harus dilewati. Diki juga harus dapat menjaga kebersihan diri dan lingkungan agar sehat selalu. Berikut point–point yang ada pada game ini :

a. Dimainkan secara *single player* pada *platform desktop,*

b. Visual grafis yang ditampilkan 2D,

c. Terdiri dari 3 *level* berbeda dengan tingkat kesulitan disetiap *level*nya,

d. Kecerdasan buatan yang diterapkan pada rintangan musuh.

1. *Story Line*

Karakter adalah seorang anak lelaki siswa sekolah dasar (SD) bernama Diki, ia senang mengkonsumsi makanan yang tidak higienis hingga suatu hari ia mengalami masalah pada perutnya, perutnya terasa sakit sekali dan sering buang air besar (BAB) tidak teratur sehingga mengalami dehidrasi, orang tua Diki segera membawanya ke dokter dan dokter mendiagnosis bahwa Diki mengalami penyakit diare, dokter pun memberi Diki obat dan menyampaikan informasi mengenai penyebab diare dan menyarankan untuk mengkonsumsi makanan yang sehat dan higienis, mencuci tangan sebelum makan agar kuman penyebab diare tidak masuk ke dalam tubuh dan meminum oralit apabila sudah terlihat gejala-gejala terserang penyakit diare.

2. *Game Play*

Berikut level-level yang ada pada game ini :

a. *Level* pertama, bercerita mengenai jajanan yang sering dikonsumsi sehari-hari oleh Diki ternyata tidak higienis dan menyebabkan ia sakit diare, sehingga orang tua dari Diki menyuruh agar selalu membawa bekal kemana pun hendak berpergian agar tidak jajan / makan sembarangan. Level ini terdiri dari 2 *stage*, *stage* pertama Diki berada di dalam *stage* mode petualangan 1, dimana Diki harus melawati rintangan lalat dan mengumpulkan item buah-buahan agar dapat melanjutkan ke *stage* kedua yaitu *mini game packing* makanan. Setelah berhasil menyelesaikan *mini game packing* maka Diki kembali ke *stage* petualangan 1 dan bisa melanjutkan ke *level* 2 dengan cara masuk ke dalam pintu yang ada dalam *stage* patualangan 1.

b. *Level* kedua, bercerita mengenai masalah kebersihan lingkungan, lingkungan tempat bermain sehari-hari Diki ternyata sangat kotor dan dapat berpengaruh terhadap kesehatan, terutama kuman atau bakteri dari sampah-sampah yang dapat mengakibatkan penyakit diare. Level kedua juga terdiri dari 2 *stage*, *stage* pertama Diki berada di dalam *stage* mode petualangan 2, dimana Diki harus melawati rintangan lalat dan tong sampah kotor, Diki juga harus mengumpulkan item sampah-sampah agar dapat melanjutkan ke *stage* kedua yaitu *mini game* membuang sampah. Setelah berhasil menyelesaikan *mini game* membuang sampah maka Diki kembali ke *stage* petualangan 2 dan dapat melanjutkan ke *level* 3 dengan cara masuk ke dalam pintu yang ada dalam *stage* patualangan 2.

c. *Level* ketiga (terakhir), bercerita pada suatu hari dokter berkunjung ke rumah Diki, dokter memberi nasehat apabila Diki atau anggota keluarga terlanjur sakit diare, diharapkan untuk tidak khawatir karena penyakit diare dapat dilawan (diobati) dengan mengkonsumsi oralit sesuai dengan dosis yang tepat, level ketiga

ini terdiri dari 2 *stage*, *stage* pertama Diki berada di dalam *stage* mode petualangan 3, dimana Diki harus melawati rintangan lalat, tong sampah kotor dan bos lalat, Diki juga harus mengumpulkan item oralit agar dapat melanjutkan ke *stage* kedua yaitu *mini game* pemberian oralit. Setelah berhasil menyelesaikan *mini game* pemberian oralit maka Diki kembali ke *stage* petualangan 3 dan dapat mengakhiri permainan dengan cara masuk ke dalam pintu yang ada dalam *stage* patualangan 3.

3. *Levelling* dan *Scoring*

1. Skenario : *Level* pertama

Intro :

Pada *level* pertama, Diki menyiapkan perbekalan dengan membawa makanan-makanan yang sehat, di perjalanan Diki harus mengumpulkan buah-buahan dan adapun yang harus diperhatikan oleh Diki sewaktu melakukan pengumpulan makanan yaitu rintangan yang dapat menghambat laju permainan.

Misi :

Menyiapkan perbekalan makanan yang sehat.

Apa yang harus dilakukan :

a. Dalam stage petualangan, Diki harus mengumpulkan buah-buahan,
b. Mengalahkan / melewati rintangan (Lalat),
c. Memasukkan makanan yang sehat kedalam tas.

Konsep *Reward* :

a. Mendapatkan nilai 100 setiap mendapatkan buah,
b. Mendapatkan nilai 100 setiap mengalahkan lalat,
c. Mendapatkan nilai 100 bila berhasil menyelesaikan *stage* mini *game* menyiapkan perbekalan.

Konsep *Punishment* :

a. Mengurangi stamina apabila besentuhan dengan rintangan (lalat),
b. Game Over, apabila stamina habis.

Outro:

Setelah mencapai pintu, dapat melanjutkan ke level selanjutnya.

2. Skenario : *Level* kedua

Intro :

Pada *level* kedua, Diki harus membersihkan sampah dengan cara memasukkan sampah-sampah ke dalam tempat sampah sesuai dengan kriteria sampah tersebut, di perjalanan Diki harus mengumpulkan samaph-sampah dan adapun yang harus diperhatikan oleh Diki sewaktu mengumpulkan sampah yaitu rintangan yang dapat menghambat laju permainan.

Misi :

Membersihkan lingkungan bermain dari sampah-sampah yang berserakkan.

Apa yang harus dilakukan :

a. Dalam stage petualangan, harus mengumpulkan sampah-sampah,
b. Mengalahkan / melewati rintangan (Lalat dan tong sampah kotor),
c. Memasukkan sampah kedalam tempat yang sudah disediakan.

Konsep *Reward* :

a. Mendapatkan nilai 100 setiap mengumpulkan sampah,
b. Mendapatkan nilai 200 setiap mengalahkan lalat,
c. Mendapatkan nilai 500 bila berhasil menyelesaikan *stage* mini *game* membersihkan lingkungan.

Konsep *Punishment* :

a. Mengurangi stamina apabila besentuhan dengan rintangan (Lalat dan tong sampah kotor),

b. Game Over, apabila stamina habis.

Outro:

Setelah mencapai pintu, dapat melanjutkan ke level selanjutnya.

3. Skenario : *Level* ketiga

Intro :

Pada *level* ketiga, Diki mengobati anggota keluarga yang sakit diare dengan memberi oralit sesuai dosis yang dianjurkan, di perjalanan Diki harus mengumpulkan oralit dan adapun yang harus diperhatikan oleh Diki sewaktu melakukan mengumpulkan oralit yaitu rintangan yang dapat menghambat laju permainan.

Misi :

Mengobati anggota keluarga dengan pemberian oralit.

Apa yang harus dilakukan :

a. Dalam stage petualangan, harus mengumpulkan oralit,
b. Mengalahkan / melewati rintangan (Lalat dan bos lalat),
c. Mengobati dengan pemberian oralit.

Konsep *Reward* :

a. Mendapatkan nilai 100 setiap mendapatkan oralit diperjalanan,
b. Mendapatkan nilai 200 setiap mengalahkan lalat dan bos lalat,
c. Mendapatkan nilai 1000 bila berhasil menyelesaikan *stage* mini *game* pemberian oralit.

Konsep *Punishment* :

a. Mengurangi stamina apabila besentuhan dengan rintangan (lalat dan bos lalat),
b. Game Over, apabila stamina habis.

Outro:

Setelah mencapai pintu, akan tampil nilai yang diperoleh dan game Lawan Diare telah berakhir.

4. Deskripsi Karakter

1. Nama karakter : Diki

   Kemampuan karakter :

   - Stamina bar : berkurang setiap menyentuh rintangan (Lalat, bos lalat dan Sampah kotor),
   - Diam, Bergerak ke kanan, ke kiri, ke atas, ke bawah, meloncat dan menembak.

2. Nama karakter : Lalat (*Game Level* 1, 2 dan 3)

   Kemampuan karakter :

   - Mengurangi stamina player,
   - Bergerak ke kiri dan ke kanan.

3. Nama karakter : Bos Lalat (*Game Level* 3)

   Kemampuan karakter :

   - Mengurangi stamina player (Banyak),
   - Diam, Bergerak ke kiri, ke kanan, ke atas dan ke bawah.

4. Nama karakter : Tong sampah kotor (*Game Level* 2)

   Kemampuan karakter :

   - Mengurangi stamina player,
   - Diam.

5. *Game Control*

Media Interaksi : Mouse, Keyboard dan Speaker (sebagai pendukung).

3.1.3 Penerapan Algoritma A * (A Star)

Algoritma A* (A Star) akan diterapkan dalam pencarian jalan terpendek pada game path finding. Pencarian jalan terpendek pada game path finding akan diasumsikan dengan titik awal (keadaan pertama) dan titik tujuan (keadaan akhir) yang diimplementasikan pada pencarian di ruang kosong maupun di ruang dengan penghalang seperti halnya permainan labirin. Ruang yang akan dibangun akan berbentuk puzzle (kotak) supaya mempermudah dalam penerapan cara kerja algoritma A*. Hasil pencarian algoritma tersebut akan menghasilkan jalan terpendek dari suatu titik awal ke titik tujuan.

Dalam game Lawan Diare, Algoritma A* Star untuk mengerakan Bos lalat mengejar Diki, algoritma ini melakukan pencarian jalur terpendek dengan waktu yang diperhitungkan dari penjumlahan waktu sebenarnya ditambah dengan waktu perkiraan dengan notasi matematika

f(n) = g(n)+h(n) **(3.1)**

1. Analisis Penggunaan algoritma A* (A star)

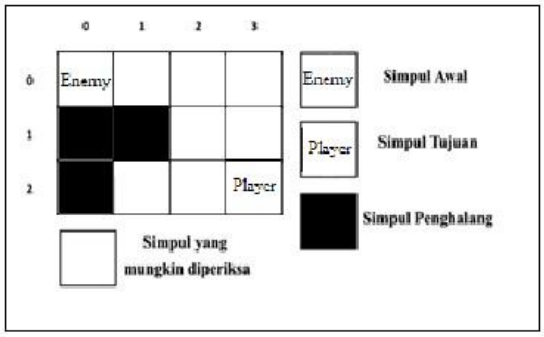
Pada game Lawan Diare, arena permainan berupa kumpulan tile-tile yang terdiri dari orde X x Y. Tiap tile memiliki koordinat X dan Y atau berbentuk seperti array dua dimensi.

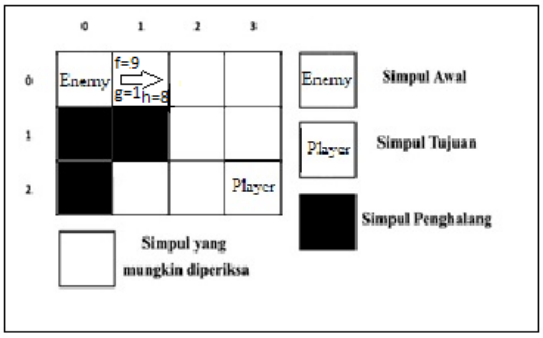
Algoritma A* akan diterapkan pada objek rintangan (Bos lalat) dalam melakukan pencarian Diki dalam arena permainan. Adapun istilah-istilah yang akan dibahas yaitu open list, Closed list, nilai f,g dan h.

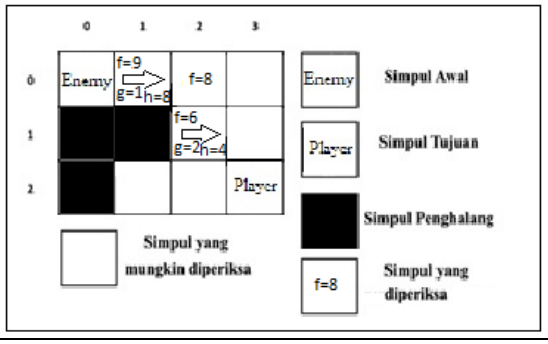
2. Ilustrasi perhitungan Algoritma A*

Arena permainan pada *game* Lawan Diare *Level* 3 terdiri dari orde X x Y. Dalam analisis ini akan dijelaskan dengan ordo 4 x 3. Simpul awal atau posisi musuh dicontohkan berada pada koordinat (0,0), sedangkan simpul tujuan atau posisi karakter pemain berada pada koordinat (3,2) di arena permainan.

Tabel 3. 1 Ilustrasi perhitungan Algoritma A *

NO	SIMULASI
1.	 Keterangan: f(n) : berada di dalam simpul di tepi kiri atas g(n) : berada didalam simpul di tepi kiri bawah h(n) : berada di dalam simpul di tepi kanan bawah. Koordinat ke-1 Koordinat yang dipilih n(1,0) Nilai nx =1 Nilai ny =0 Simpil tujuan (3,2) Goal x=3 Goal y=2 Perhitungan : g(1,0)= 1 h orthogonal : h(1,0)= (abs(nx–goal x) + abs(ny–goal y)) h(1,0)= (abs(1-3)+abs(0-2)) h(1,0)= abs(-2)+abs(-2) h(1,0)= 4 h diagonal : h(1,0)= min(abs(nx–goal x) + abs(ny–goal y)) h(1,0)= min(abs(1-3)+abs(0-2)) h(1,0)= min(abs(-2)+abs(-2) h(1,0)= min 4 h(n)= h diagonal(n) + h orhogonal(n) – (2 * h diagonal(n)) h(1,0)= (-4) + (4 - (2* -4)) h(1,0)= -4 + 12 h(1,0)= 8

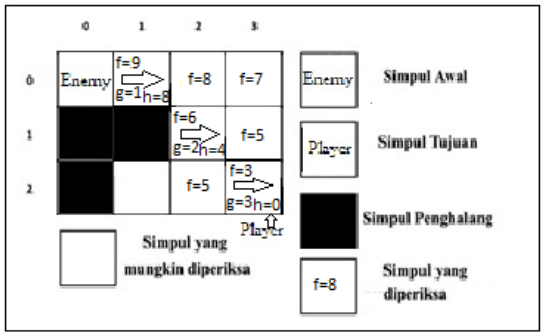
	f(1,0)= g(1,0)+h(1,0) f(1,0)= 1+8 f(1,0)= 9 Koordinat Selanjutnya tidak tersedia
2.	 Keterangan: Dari hasil perhitungan diatas dipilihlah simpul (1,0) sebagai bestNode karena memiliki waktu terkecil yaitu 9. Koordinat yang dipilih n(2,0) Nilai nx =2 Nilai ny =0 Simpil tujuan (3,2) Goal x=3 Goal y=2 Perhitungan : g(2,0)= 2 h orthogonal : h(2,0)= (abs(nx–goal x) + abs(ny–goal y)) h(2,0)= (abs(2-3)+abs(0-2)) h(2,0)= abs(-1)+abs(-2) h(2,0)= 3 h diagonal : h(2,0)= min(abs(nx–goal x) + abs(ny–goal y)) h(2,0)= min(abs(2-3)+abs(0-2)) h(2,0)= min(abs(-1)+abs(-2) h(2,0)= min 3 h(n)= h diagonal(n) + h orhogonal(n) – (2 * h diagonal(n)) h(2,0)= (-3) + (3 - (2* -3)) h(2,0)= -3 + 9 h(2,0)= 6

	f(2,0)= g(2,0)+h(2,0) f(2,0)= 2+6 f(2,0)= 8 Koordinat yang dipilih n(2,1) Nilai nx =2 Nilai ny =1 Simpil tujuan (3,2) Goal x=3 Goal y=2 Perhitungan : g(2,1)= 2 h orthogonal : h(2,1)= (abs(nx–goal x) + abs(ny–goal y)) h(2,1)= (abs(2-3)+abs(1-2)) h(2,1)= abs(-1)+abs(-1) h(2,1)= 2 h diagonal : h(2,1)= min(abs(nx–goal x) + abs(ny–goal y)) h(2,1)= min(abs(2-3)+abs(1-2)) h(2,1)= min(abs(-1)+abs(-1) h(2,1)= min 2 h(n)= h diagonal(n) + h orhogonal(n) – (2 * h diagonal(n)) h(2,1)= (-2) + (2 - (2* -2)) h(2,1)= -2 + 6 h(2,1)= 4 f(2,1)= g(2,1)+h(2,1) f(2,1)= 2+4 f(2,1)= 6
3.	

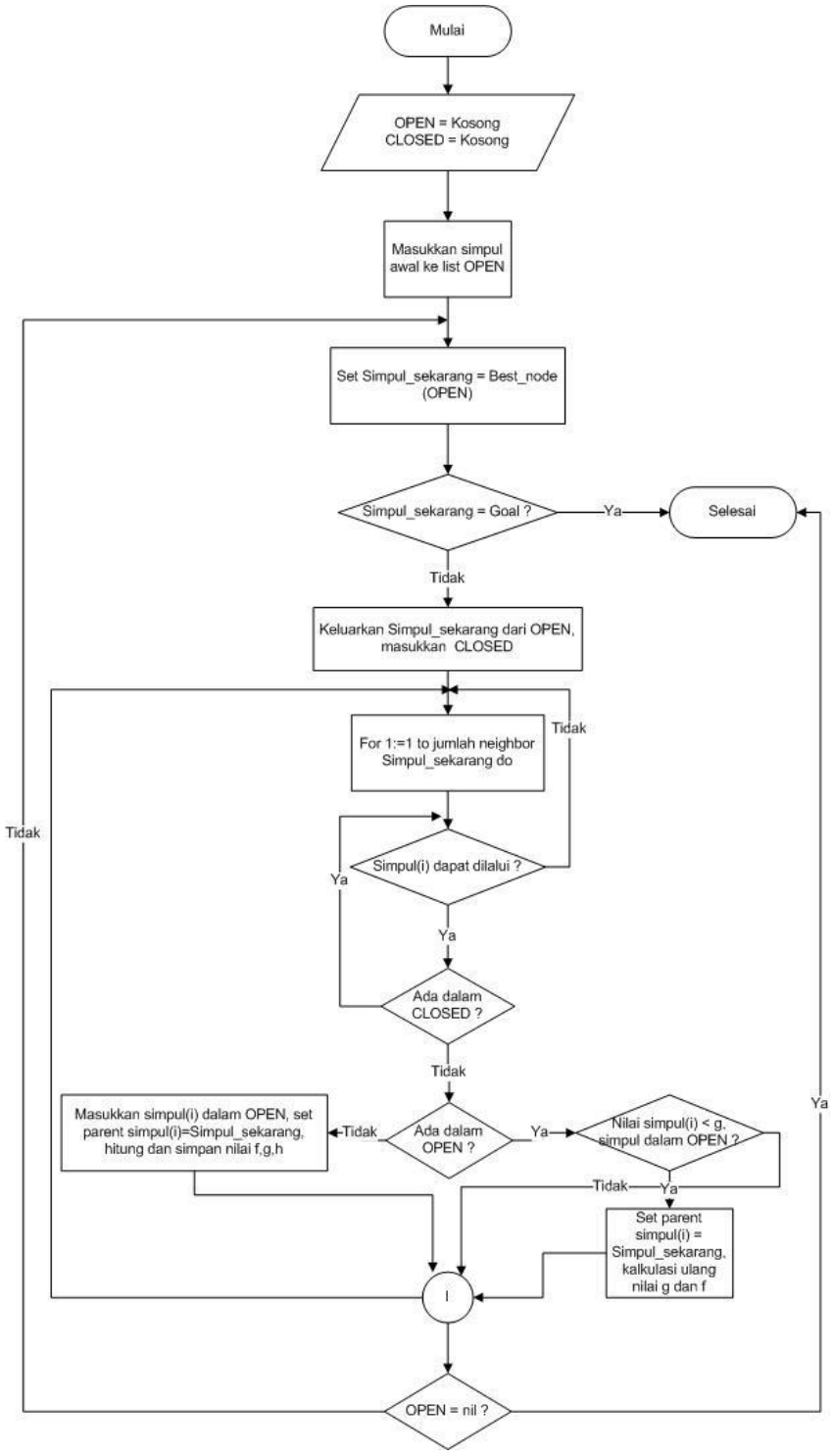
	Keterangan: Dari hasil perhitungan diatas dipilihlah simpul (2,1) sebagai bestNode karena memiliki waktu terkecil yaitu 6. Koordinat yang dipilih n(3,1) Nilai nx =3 Nilai ny =1 Simpil tujuan (3,2) Goal x=3 Goal y=2 Perhitungan : g(3,1)= 3 h orthogonal : h(3,1)= (abs(nx–goal x) + abs(ny–goal y)) h(3,1)= (abs(3-3)+abs(1-2)) h(3,1)= abs(0)+abs(-1) h(3,1)= 1 h diagonal : h(3,1)= min(abs(nx–goal x) + abs(ny–goal y)) h(3,1)= min(abs(3-3)+abs(1-2)) h(3,1)= min(abs(0)+abs(-1) h(3,1)= min 1 h(n)= h diagonal(n) + h orhogonal(n) – (2 * h diagonal(n)) h(3,1)= (-1) + (1 - (2* -1)) h(3,1)= -1 + 3 h(3,1)= 2 f(3,1)= g(3,1)+h(3,1) f(3,1)= 3+2 f(3,1)= 5 Koordinat yang dipilih n(2,2) Nilai nx =2 Nilai ny =2 Simpil tujuan (3,2) Goal x=3 Goal y=2 Perhitungan : g(2,2)= 3

	h orthogonal : h(2,2)= (abs(nx–goal x) + abs(ny–goal y)) h(2,2)= (abs(2-3)+abs(2-2)) h(2,2)= abs(-1)+abs(0) h(2,2)= 1 h diagonal : h(2,2)= min(abs(nx–goal x) + abs(ny–goal y)) h(2,2)= min(abs(2-3)+abs(2-2)) h(2,2)= min(abs(-1)+abs(0) h(2,2)= min 1 h(n)= h diagonal(n) + h orhogonal(n) – (2 * h diagonal(n)) h(2,2)= (-1) + (1 - (2* -1)) h(2,2)= -1 + 3 h(2,2)= 2 f(2,2)= g(2,2)+h(2,2) f(2,0)= 3+2 f(2,0)= 5 Koordinat yang dipilih n(3,0) Nilai nx =3 Nilai ny =0 Simpil tujuan (3,2) Goal x=3 Goal y=2 Perhitungan : g(3,0)= 3 h orthogonal : h(3,0)= (abs(nx–goal x) + abs(ny–goal y)) h(3,0)= (abs(3-3)+abs(0-2)) h(3,0)= abs(0)+abs(-2) h(3,0)= 2 h diagonal : h(3,0)= min(abs(nx–goal x) + abs(ny–goal y)) h(3,0)= min(abs(3-3)+abs(0-2)) h(3,0)= min(abs(0)+abs(-2) h(3,0)= min 2 h(n)= h diagonal(n) + h orhogonal(n) – (2 * h diagonal(n))

	h(3,0)= (-2) + (2 - (2* -2)) h(3,0)= -2 + 6 h(3,0)= 4 f(3,0)= g(3,0)+h(3,0) f(2,0)= 3+4 f(2,0)= 7 Koordinat yang dipilih n(3,2) Nilai nx =2 Nilai ny =0 Simpil tujuan (3,2) Goal x=3 Goal y=2 Perhitungan : g(3,2)= 3 h orthogonal : h(3,2)= (abs(nx–goal x) + abs(ny–goal y)) h(3,2)= (abs(3-3)+abs(2-2)) h(3,2)= abs(0)+abs(0) h(3,2)= 0 h diagonal : h(3,2)= min(abs(nx–goal x) + abs(ny–goal y)) h(3,2)= min(abs(3-3)+abs(2-2)) h(3,2)= min(abs(0)+abs(0) h(3,2)= 0 h(n)= h diagonal(n) + h orhogonal(n) – (2 * h diagonal(n)) h(3,2)= (0) + (0 - (2* -0)) h(3,2)= 0 + 0 h(3,2)= 0 f(3,2)= g(3,2)+h(3,2) f(3,2)= 3+0 f(3,2)= 3

4.	 Setelah mendapatkan bestNode dilangkah ketiga, dilakukan penelusuran selanjutnya untuk mendapatkan bestnode selanjutnya dengan melakukan penelusuran simpul-simpul yang memiliki nilai cost terkecil. Simpul yang dihitung selanjutnya adalah koordinat (3,1), (3.0), (2,2) dan (3,2) dan koordinat (3,2) mempunyai cost terkecil yaitu 3 sekaligus enemy berhasil ke tempat player (simpul tujuan).

Untuk lebih jelas mengenai pencarian dengan menggunakan algoritma A* dapat dilihat dalam diagram *flowchart* pada Gambar dibawah ini.



Gambar 3. 1 Diagram Flowchart Algoritma A*

Secara detail pseudo code algoritma A* untuk musuh mencari player adalah sebagai berikut:

```
inisialisasi list
        openList = kosong
        closedList = kosong
        masukkan simpul awal ke openList
                openList = node
         menentukan perhitungan A* (A star)
                curNode.g = 0
                curNode.h = Pathfinder.manhattanHeuristic (curNode,Finished)
                curF = node.g + node.h
        while curNode tidak sama dengan Finished
                SearchLevel()
                for i = 1 to level.length
                        openList = level[row][col]
                if openList = curNode & level.obstacle = false
                                        g = node.g + travelCost
                                        h = Pathfinder.manhattanHeuristic(level, Finished)
                                        f = g + h
                        end if
                         if Pathfinder.isOpen(level, openList) dan
                        Pathfinder.isClosed( level, closedList)
                         if curF > f
                                        curF = f
                                        node.g = g
                                        node.h = h
                                        level.parentNode = curNode
                                        openList. parentNode (level)
                        end if
                        else
                                        level.f = f
                                        level.g = g
                                        level.h = h
                                        level.parentNode = curNode
                                        openList.push(level)
                        end if
                        end for
                        if openList.length = 0
                                        return null
                        end if
                        openList.sortOn('f', Array.NUMERIC)
                                        curNode = openList.push()
        end while
        return Pathfinder.RetracePath(node)
end
```

Gambar 3. 2 Pseudo Code Algoritma A*

3.2 Analisis Kebutuhan *Non* Fungsional

Analisis kebutuhan *non* fungsional biasanya dilakukan untuk mengetahui spesifikasi kebutuhan yang melibatkan analisis perangkat keras / *hardware*, analisis perangkat lunak / *software*, dan juga analisis pengguna / *user*.

3.2.1 Analisis Kebutuhan Perangkat Keras

Komputer terdiri dari perangkat keras dan perangkat lunak yang saling berinteraksi. Perangkat lunak memberikan instruksi - instruksi kepada perangkat keras untuk melakukan suatu tugas tertentu, sehingga dapat menjalankan suatu sistem di dalamnya.

Kebutuhan *hardware* (perangkat keras) yang dipakai adalah komputer / *notebook* dengan spesifikasi yang digunakan oleh pengembang sebagai berikut:

1. *Processor* 2.4 Ghz
2. *Harddisk* 720GB
3. *RAM* 4 GB
4. *VGA* 2 GB
5. *LCD Monitor* dengan resolusi 1366 x 768

Sedangkan perangkat keras minimum yang dibutuhkan untuk menjalankan aplikasi *game* ini adalah sebagai berikut :

1. *Processor* 1.3 *Ghz*
2. *RAM* 512 MB
3. *Harddisk* 75 MB
4. Kartu grafik dengan DirectX 9 *compatible sound card*
5. *CRT monitor* dengan resolusi 800 x 600

3.2.2 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang dipergunakan dalam pembangunan aplikasi game yang akan dibangun sebagai berikut :

1. Sistem operasi *Window* 7 *Ultimate 32-bit*
2. *Adobe Flash CS6*
3. *Flash Illustrator CS5*
4. *Flash Player*

Sedangka perangkat lunak yang dibutuhkan pengguna untuk menjalankan aplikasi *game* ini adalah sebagai berikut :

1. Sistem Operasi *Window XP, Window Vista,* dan *Window 7*
2. *Flash Player*

3.2.3 Analisis Pengguna

Game yang akan dibuat ditujukan khusus untuk siswa siswi sekolah dasar (SD) rentang usia 8 hingga 13 tahun. Analisis pengguna menggunakan observasi berdasarkan analisis *game* sejenis bergenre *action* yang diminati oleh kebanyakan pemain. Maka analisis penggunanya berdasarkan elemen yang ada adalah sebagai berikut :

1. Pemain menyukai *game action* yang mempunyai beberapa level dalam permainannya.
2. Pemain menyukai karakter yang mempunyai darah pada permainan. Artinya, ketika karakter pemain terkena sekali serangan musuh, permainan tidak akan langsung berakhir.
3. Pemain menyukai permainan yang memiliki *scores*. Hal ini berhubungan dengan *achieve* bagi pemain untuk memainkan permainannya
4. *Dungeon exit and level warps*, pemain menyukai permainan dengan adanya sebuah konten untuk fase perpindahan level, karena hal ini menyebabkan misi dari setiap level menjadi jelas.
5. Pemain menginginkan *victory conditions* pada permainan. Elemen ini dibutuhkan untuk menyelesaikan cerita dari permainan yang sedang dimainkan.
6. Pemain menginginkan *user interface* yang sederhana dan mudah dimengerti, tetapi memenuhi kebutuhan pemain.

3.3 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 serta sesuai dengan kebutuhan.

3.3.1 Analisis Kebutuhan Sistem

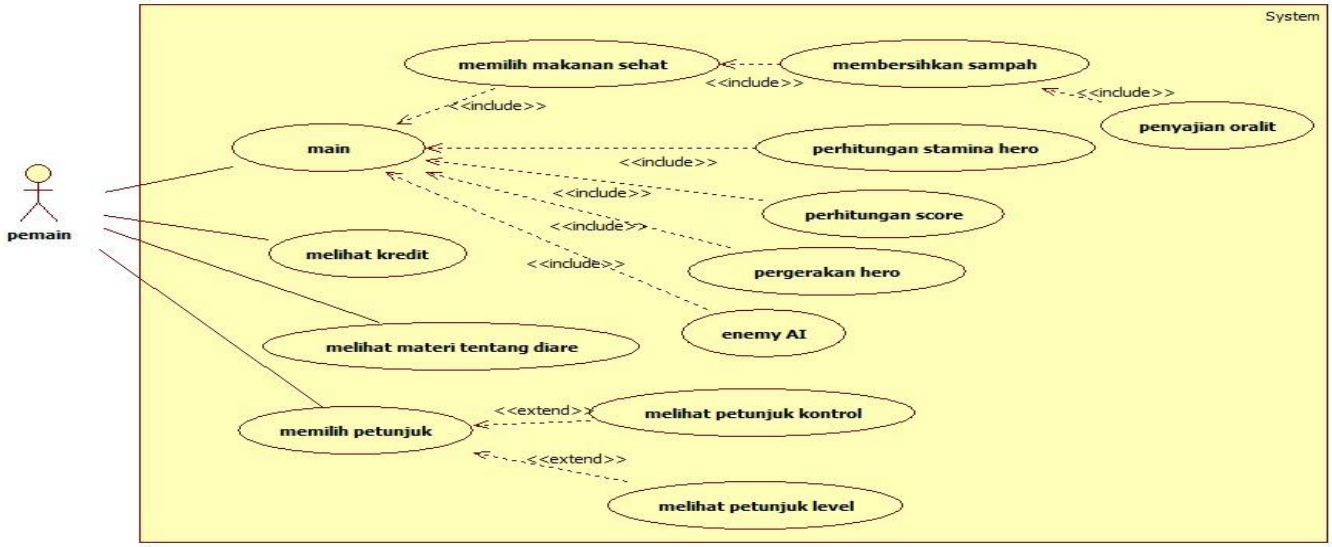
Dalam pembangunan aplikasi perancangan sistem, sebelumnya dilakukan analisis kebutuhan sistem agar sistem yang dibangun sesuai dengan yang diharapkan atau dibutuhkan pengguna.

3.3.2 Pemodelan Sistem

Pemodelan sistem berfungsi untuk menentukan fungsi-fungsi yang dapat dilakukan oleh sistem pada aplikasi serta menentukan kelas yang dibutuhkan untuk realisasi fungsi - fungsi sistem yang telah dianalisis sebelumnya dan mendeskripsikannya kedalam bentuk diagram.

1. *Use Case* Diagram

Berikut ini adalah rancangan dari use case diagram game Lawan Diare. *Use case* diagram memberikan gambaran tentang bagaimana pengguna berinteraksi dengan sistem.



Gambar 3. 3 Diagram Use Case Game Lawan Diare

1.1 Definisi Aktor

Definisi Aktor berfungsi untuk menjelaskan peran dari tiap aktor yang terlibat dalam sistem. Berikut definisi aktor dibawah ini :

1. Aktor : Pemain

   Deskripsi : Pemain merupakan orang yang memainkan game dan berinteraksi langsung dengan antar muka game

1.2 Definisi Use Case

Definisi *use case* berfungsi untuk menjelaskan proses yang terdapat pada setiap *use case*. Berikut definisi use case dibawah ini :

Tabel 3. 2 Definisi *Use Case*

No.	Use Case	Deskripsi
1	Melihat Kredit	Fungsi untuk melihat credits pembuat game
2	Melihat Materi Tentang Diare	Fungsi untuk menampilkan pengetahuan umum tentang penyakit diare
3	Memilih Petunjuk	Fungsi untuk memilih petunjuk permainan yang terdapat dalam menu petunjuk
4	Melihat Petunjuk Kontrol	Fungsi untuk mengetahui petunjuk kontrol permainan menggunakan keyboard dan mouse
5	Melihat Petunjuk Level	Fungsi untuk mengetahui petunjuk atau aturan permainan sesuai dengan level pada game
6	Main	Fungsi untuk memulai memainkan game
7	Perhitungan Stamina *Hero*	Fungsi untuk menghitung stamina *hero* dalam game
8	Pergerakan *Hero*	Fungsi untuk mengotrol pergerakan *hero* dalam game
9	Perhitungan *Score*	Fungsi untuk menghitung banyaknya nilai yang diraih dalam game
10	Memilih Makanan Sehat	Fungsi untuk memainkan game materi memilih makanan sehat
11	Membersihkan Sampah	Fungsi untuk memainkan game materi membersihkan sampah
12	Penyajian Oralit	Fungsi untuk memainkan game materi penyajian oralit
13	*Enemy* AI	Fungsi algoritma A Star yang terdapat dalam game

1.3 Skenario *Use Case*

Skenario *use case* merupakan sekenario dari setiap bagian pada *use case* menunjukkan proses apa yang terjadi pada setiap bagian didalam *use case*.

a. Skenario *Use Case* Melihat Kredit

Skenario *use case* ini menjelaskan interaksi antara *actor* dengan *use case* melihat kredit.

Tabel 3. 3 Skenario *Use Case* Melihat Kredit

Identifikasi	
Nama	Melihat kredit
Tujuan	Untuk mengatahui credits pembuat game
Actor	Pemain
Skenario Utama	
Kondisi Awal	Menu Utama ditampilkan
Aksi Aktor	Reaksi Sistem
1. Memilih menu kredit	
	2. Menampilkan tentang pembuat *game*
Kondisi Akhir	Menampilkan menu kredit

b. Skenario *Use Case* Melihat Materi Tentang Diare

Skenario *use case* ini menjelaskan interaksi antara *actor* dengan *use case* melihat materi tentang diare.

Tabel 3. 4 Skenario *Use Case* Melihat Materi Tentang Diare

Identifikasi	
Nama	Melihat materi tentang diare
Tujuan	Untuk menampilkan pengetahuan umum tentang penyakit diare
Actor	Pemain
Skenario Utama	
Kondisi Awal	Menu Utama ditampilkan
Aksi Aktor	Reaksi Sistem
1. Memilih menu materi tentang diare	
	2. Menampilkan materi diare
Kondisi Akhir	Menampilkan menu materi diare

c. Skenario Use Case Memilih Petunjuk

Skenario *use case* ini menjelaskan interaksi antara *actor* yaitu pemain dengan use case memilih petunjuk.

Tabel 3. 5 Skenario *Use Case* Memilih Petunjuk

Identifikasi		
Nama	Memilih Petunjuk	
Tujuan	Menampilkan petunjuk permainan yang terdapat dalam menu petunjuk	
Actor	Pemain	
Skenario Utama		
Kondisi Awal	Menu Utama ditampilkan	
Aksi Aktor		Reaksi Sistem
1. Memilih menu petunjuk		
		2. Menampilkan pilihan petunjuk
Kondisi Akhir	Menampilkan menu pilihan petunjuk	

d. Skenario *Use Case* Melihat Petunjuk Kontrol

Skenario *use case* ini akan menjelaskan interaksi antara *actor* yaitu pemain, dengan *use case* melihat petunjuk kontrol.

Tabel 3. 6 Skenario Use Case Melihat Petunjuk Kontrol

Identifikasi		
Nama	Melihat Petunjuk Kontrol	
Tujuan	Untuk mengatahui petunjuk kontrol permainan	
Actor	Pemain	
Skenario Utama		
Kondisi Awal	Menu memlih petunjuk	
Aksi Aktor		Reaksi Sistem
1. Memilih menu petunjuk		
		2. Menampilkan pilihan petunjuk
3. Memilih menu petunjuk kontrol		
		4. Menampilkan petunjuk kontrol
Kondisi Akhir	Menampilkan menu petunjuk kontrol	

e. Skenario *Use Case* Melihat Petunjuk *Level*

Skenario *use case* ini menjelaskan interaksi antara *actor* yaitu pemain, dengan *use case* melihat petunjuk *level*

Tabel 3. 7 Skenario *Use Case* Melihat Petunjuk *Level*

Identifikasi		
Nama	Melihat Petunjuk *Level*	
Tujuan	Untuk mengatahui pentunjuk atau aturan permainan sesuai dengan *level* pada game	
Actor	Pemain	
Skenario Utama		
Kondisi Awal	Menu memilih petunjuk	
Aksi Aktor		Reaksi Sistem
1. Memilih menu petunjuk		
		2. Menampilkan pilihan petunjuk
3. Memilih menu petunjuk l*evel*		
		4. Menampilkan petunjuk *level*
Kondisi Akhir	Menampilkan menu petunjuk *level*	

f. Skenario *Use Case* Main

Skenario *use case* ini menjelaskan interaksi antara *actor* dengan *use case* Main.

Tabel 3. 8 Skenario *Use Case* Main

Identifikasi		
Nama	Main	
Tujuan	Untuk dapat memulai permainan	
Aktor	Pemain	
Skenario utama		
Kondisi awal	Menu Utama ditampilkan	
Aksi aktor		Reaksi Sistem
1. Memilih mulai permainan		
		2. Memasukkan *hero* ke *screen* permainan 3. Memunculkan musuh 4. Memunculkan musuh khusus 5. Memunculkan target item 6. Memunculkan *Warp Zone* 7. Memunculkan item 8. Apabila *hero* mati, maka menampilkan *screen game over*
9. Menekan tombol navigasi kiri, kanan dan atas		
		10. Menggerakkan *hero*
11. Jika menekan tombol *keyboard space*		
		12. Menyemprotkan pembasmi lalat 13. Apabila musuh terkena asap pembasmi lalat dan mati, maka *score* ditambahkan
14. Menyentuh target item		
		15. Masuk kedalam *stage game* materi 16. Apabila berhasil menyelesaikan *game* materi, maka *hero* masuk kembali ke *scene* permainan 17. Menghapus target item 18. Warp zone terbuka
19. Menyentuh warp zone		
		20. Menghapus musuh 21. Menampilkan *win screen* 22. Memasukkan *hero* ke *screen* permainan selanjutnya
Kondisi akhir	Menampilkan *win screen*	

g. Skenario *Use Case* Perhitungan Stamina *Hero*

Skenario *use case* ini menjelaskan interaksi antara *actor* dengan *use case* perhitungan stamina *hero*.

Tabel 3. 9 Skenario *Use Case* Perhitungan Stamina *Hero*

Identifikasi	
Nama	Perhitungan stamina *hero*
Tujuan	Untuk menghitung stamina *hero* dalam game
Actor	Pemain
Skenario Utama	
Kondisi Awal	Permainan dimulai
Aksi Aktor	Reaksi Sistem
	1. Menentukan stamina *hero* sebesar 1000
2. Menyentuh musuh	
	3. Stamina *hero* berkurang 50
4. Menyentuh musuh khusus	
	5. Stamina *hero* berkurang 100
6. Menyentuh sampah kotor	
	7. Stamina *hero* berkurang 30
8. Menyentuh jurang	
	9. Stamina *hero* habis
Kondisi Akhir	Hasil perhitungan stamina *hero*

h. Skenario *Use Case* Pergerakan *Hero*

Skenario *use case* ini menjelaskan interaksi antara *actor* dengan *use case* pergerakan *hero*.

Tabel 3. 10 Skenario *Use Case* Pergerakan *Hero*

Identifikasi	
Nama	Pergerakan *Hero*
Tujuan	Untuk melakukan pergerakan *hero* dalam game
Actor	Pemain
Skenario Utama	
Kondisi Awal	*Hero* diam

Aksi Aktor	Reaksi Sistem
1. Jika pemain menekan tombol kanan pada *keyboard*	
	2. *Hero* bergerak ke kanan
3. Jika pemain menekan tombol kiri pada *keyboard*	
	4. *Hero* bergerak ke kiri
5. Jika pemain menekan tombol atas pada *keyboard*	
	6. *Hero* loncat
7. Jika pemain menekan klik kiri pada *mouse*	
	8. *Hero* menyemprotkan pembasmi lalat
Kondisi Akhir	Pemain melakukan pergerakan *hero*

i. Skenario *Use Case* Perhitungan *Score*

Skenario *use case* ini menjelaskan interaksi antara *actor* dengan *use case* perhitungan *score*.

Tabel 3. 11 Skenario *Use Case* Perhitungan *Score*

Identifikasi	
Nama	Perhitungan *Score*
Tujuan	Untuk menghitung banyaknya nilai yang diraih dalam game
Actor	Pemain
Skenario Utama	
Kondisi Awal	Permainan dimulai
Aksi Aktor	Reaksi Sistem
	1. Menentukan *score* awal sebesar 0
2. Mengumpulkan item diperjalanan	
	3. Menambah *score* sebesar 100 pada permainan
4. Membunuh musuh	
	5. Menambah *score* sebesar 100 atau 200 sesuai dengan *level* permainan
6. Menyelesaikan game materi	
	7. Menambah *score* sebesar 100 atau 500 atau 1000 sesuai dengan level permainan
Kondisi Akhir	Hasil perhitungan *score*

j. Skenario *Use Case* Memilih Makanan Sehat

Skenario *use case* ini menjelaskan interaksi antara *actor* dengan *use case* memilih makanan sehat.

Tabel 3. 12 Skenario *Use Case* Memilih Makanan Sehat

Identifikasi		
Nama	Memilih makanan sehat	
Tujuan	Untuk memainkan *game* materi memilih makanan sehat	
Actor	Pemain	
Skenario Utama		
Kondisi Awal	Menyentuh Target item Tas	
Aksi Aktor		Reaksi Sistem
1. Menyentuh target item		
		2. Memasukkan karakter *hero* ke *screen game* materi memilih makana sehat 3. Memunculkan target berbentuk tas 4. Memunculkan item berbentuk macam-macam hidangan
5. Memindahkan item hidangan		
		6. Mengecek target item yang dimaksud 7. Item berpindah tempat kedalam target 8. Apabila item tidak cocok dengan target, maka item kembali ke posisi awal 9. Menampilkan tombol lanjut
10. Menekan tombol lanjut		
		11. Menutup game materi 12. Membuka warp zone
Kondisi Akhir	Menampilkan *Game* materi memilih makanan sehat	

k. Skenario *Use Case* Membersihkan sampah

Skenario *use case* ini menjelaskan interaksi antara *actor* dengan *use case* membersihkan sampah.

Tabel 3. 13 Skenario *Use Case* Membersihkan Sampah

Identifikasi	
Nama	Membersihkan sampah

Tujuan	Untuk memainkan *game* materi membersihkan sampah	
Actor	Pemain	
Skenario Utama		
Kondisi Awal	Menyentuh target item tempat sampah	
Aksi Aktor		Reaksi Sistem
1. Menyentuh target item		
		2. Memasukkan karakter *hero* ke *screen game* materi membersihkan sampah 3. Memunculkan target berbentuk tempat sampah 4. Memunculkan item berbentuk macam-macam sampah
5. Memindahkan item sampah		
		6. Mengecek target item yang dimaksud 7. Item berpindah tempat kedalam target 8. Apabila item tidak cocok dengan target, maka item kembali ke posisi awal 9. Menampilkan tombol Cuci tangan
10. Menekan tombol cuci tangan		
		11. Menutup game materi 12. Membuka warp zone
Kondisi Akhir	Menampilkan *Game* materi membersihkan sampah	

l. Skenario *Use Case* Penyajian Oralit

Skenario *use case* ini menjelaskan interaksi antara *actor* dengan *use case* penyajian oralit.

Tabel 3. 14 Skenario *Use Case* Penyajian Oralit

Identifikasi		
Nama	Penyajian oralit	
Tujuan	Untuk memainkan *game* materi penyajian oralit	
Actor	Pemain	
Skenario Utama		
Kondisi Awal	Menyentuh target item tas P3K	
Aksi Aktor		Reaksi Sistem
1. Menyentuh target item		
		2. Memasukkan karakter *hero* ke *screen game* materi penyajian oralit 3. Memunculkan target berupa anggota

	keluarga yang terjangkit penyakit diare 4. Memunculkan item berbentuk oralit
5. Memindahkan item oralit	
	6. Mengecek target item yang dimaksud 7. Item berpindah tempat kedalam target 8. Apabila item tidak cocok dengan target, maka item kembali ke posisi awal 9. Menampilkan tombol sehat
10. Menekan tombol sehat	
	11. Menutup game materi 12. Membuka warp zone
Kondisi Akhir	Menampilkan *Game* materi penyajian oralit

m. Skenario *Use Case Enemy* AI

Skenario *use case* ini menjelaskan interaksi antara *actor* dengan *use case enemy* AI *dalam game level* 3.

Tabel 3. 15 Skenario *Use Case Enemy* AI

Identifikasi	
Nama	*Enemy* AI
Tujuan	Untuk meunculkan musuh khusus
Actor	Pemain
Skenario Utama	
Kondisi Awal	Didalam *screen* permainan
Aksi Aktor	Reaksi Sistem
	1. Memunculkan musuh khusus 2. Menggerakkan musuh khusus mengejar *hero* 3. Apabila *hero* mati, maka menampilkan *screen game over*
4. Menekan tombol navigasi kiri, kanan dan atas	
	5. Aksi pergerakan *hero*
6. Jika menekan tombol *keyboard space*	
	7. Menyemprotkan pembasmi lalat 8. Darah musuh khusus berkurang 9. Jika darah musuh khusus habis, maka musuh khusus akan mati

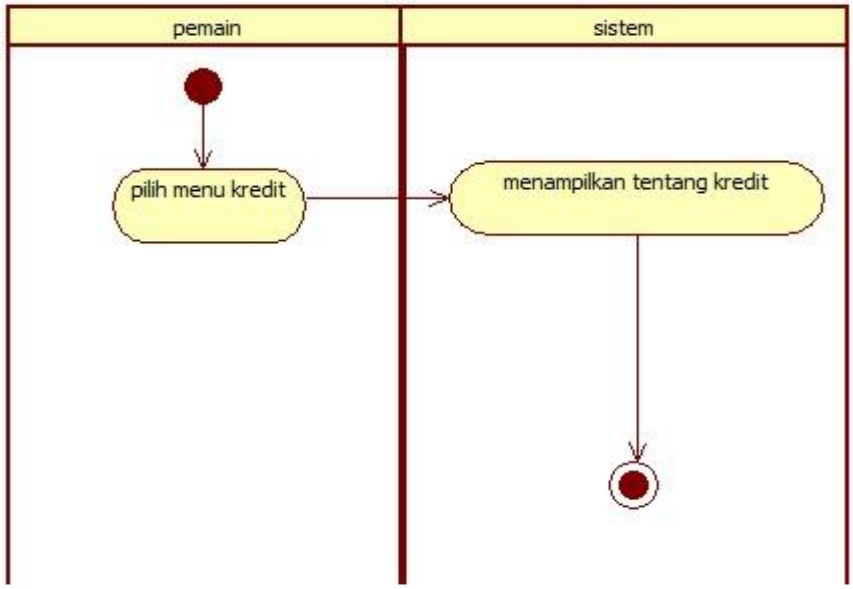
	10. Apabila musuh terkena asap pembasmi lalat dan mati, maka *score* ditambahkan
Kondisi Akhir	Musuh Khusus mati

23. *Activity* Diagram

Activity diagram adalah teknik untuk mendeskripsikan logika *procedural*, proses bisnis dan aliran kerja dari urutan aktivitas dalam suatu proses yang mengacu pada *use case* diagram.

a. *Activity* Diagram Melihat *Credit*

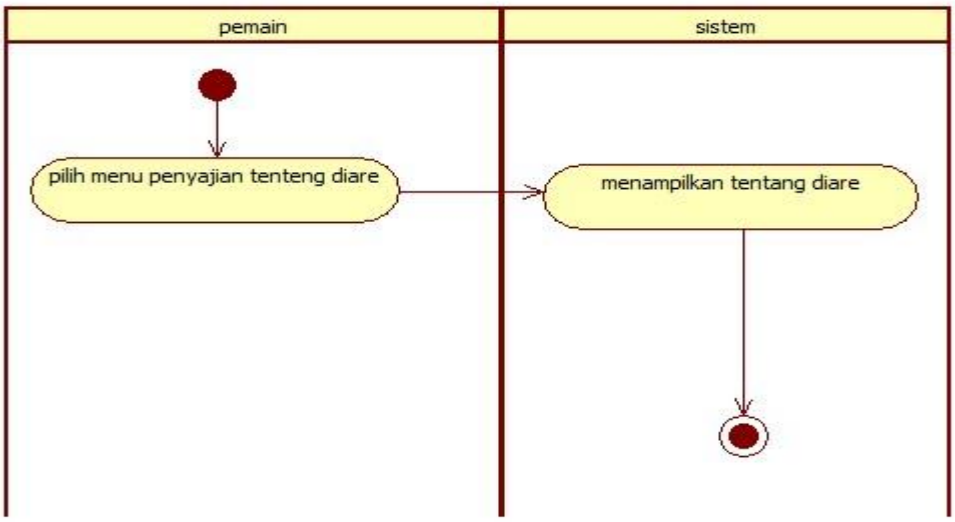
Activity diagram ini menjelaskan interaksi antara pemain dengan sistem.



Gambar 3. 4 Activity Diagram Melihat *Credit*

b. *Activity* Diagram Melihat Materi Tentang Diare

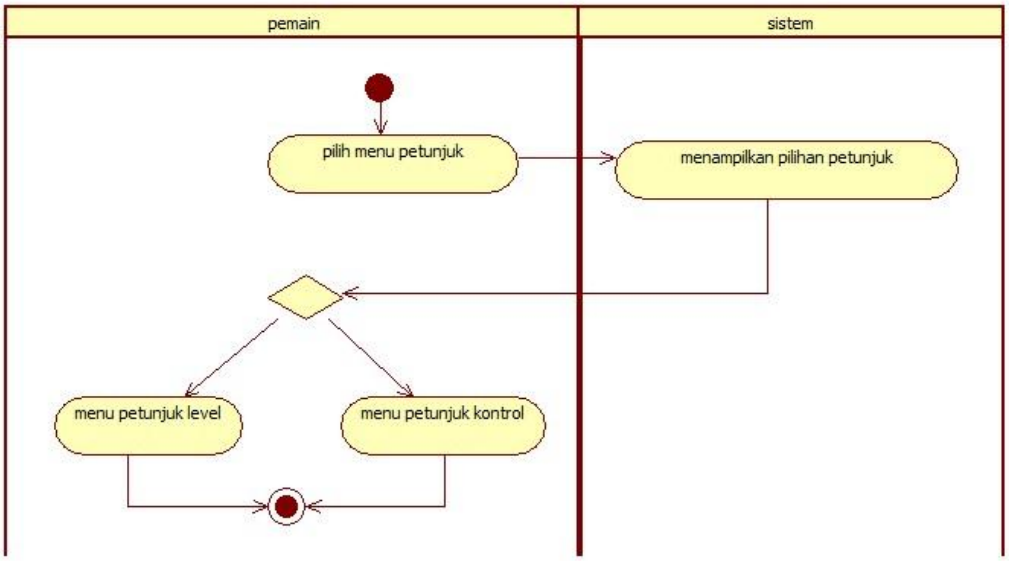
Activity diagram ini menjelaskan interaksi antara pemain dengan sistem.



Gambar 3. 5 Activity Diagram Melihat Materi Tentang Diare

c. *Activity* Diagram Memilih Petunjuk

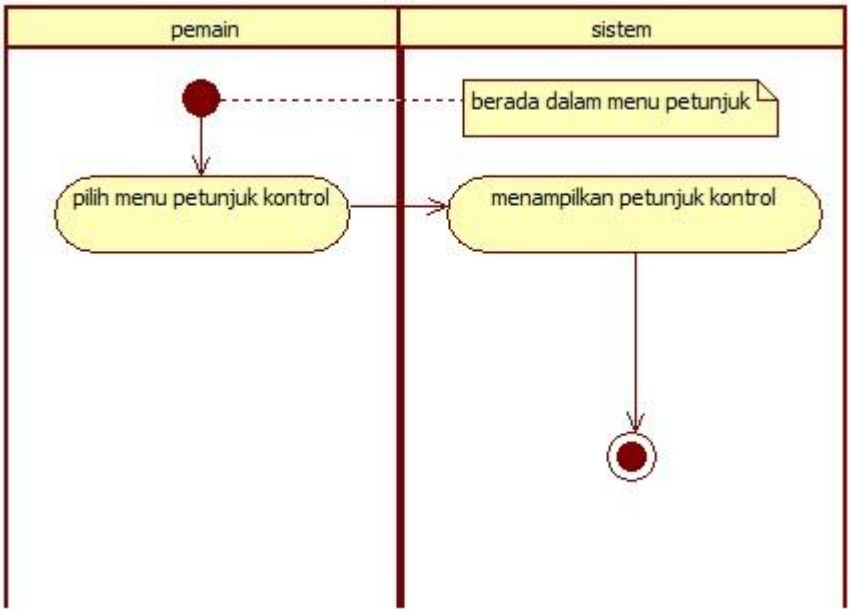
Activity diagram ini menjelaskan interaksi antara pemain dengan sistem.



Gambar 3. 6 Activity Diagram Memilih Petunjuk

d. *Activity* Diagram Melihat Petunjuk Kontrol

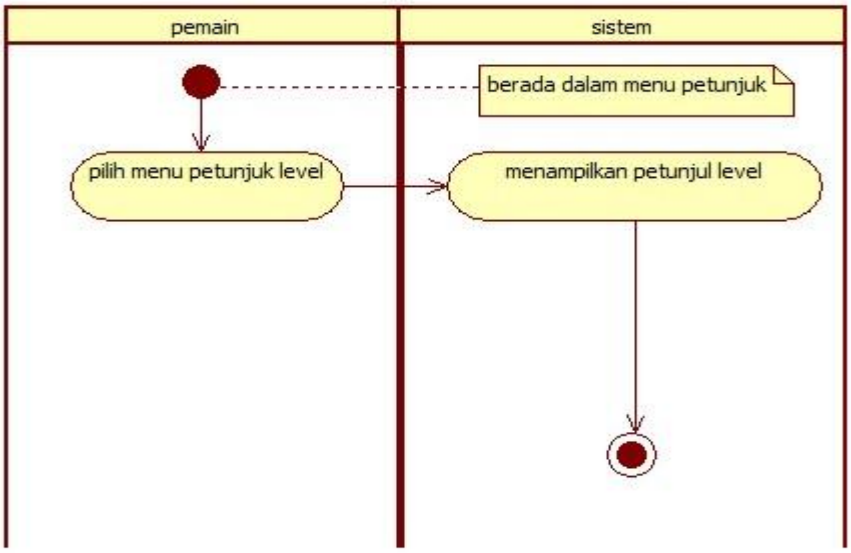
Activity diagram ini menjelaskan interaksi antara pemain dengan sistem.



Gambar 3. 7 Activity Diagram Melihat Petunjuk Kontrol

e. *Activity* Diagram Melihat Petunjuk *Level*

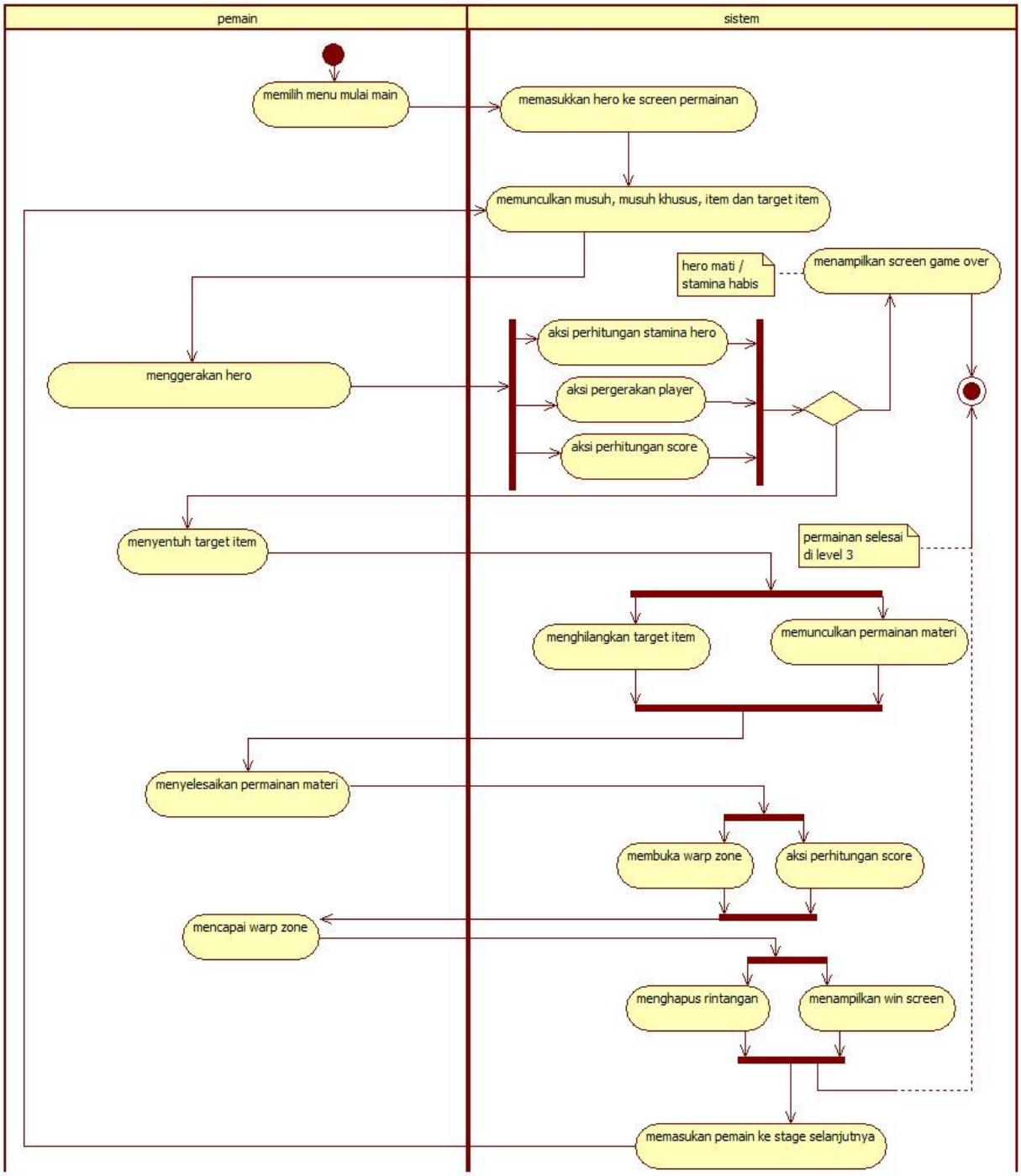
Activity diagram ini menjelaskan interaksi antara pemain dengan sistem.



Gambar 3. 8 *Activity* Diagram Melihat Petunjuk *Level*

f. *Activity* Diagram Main

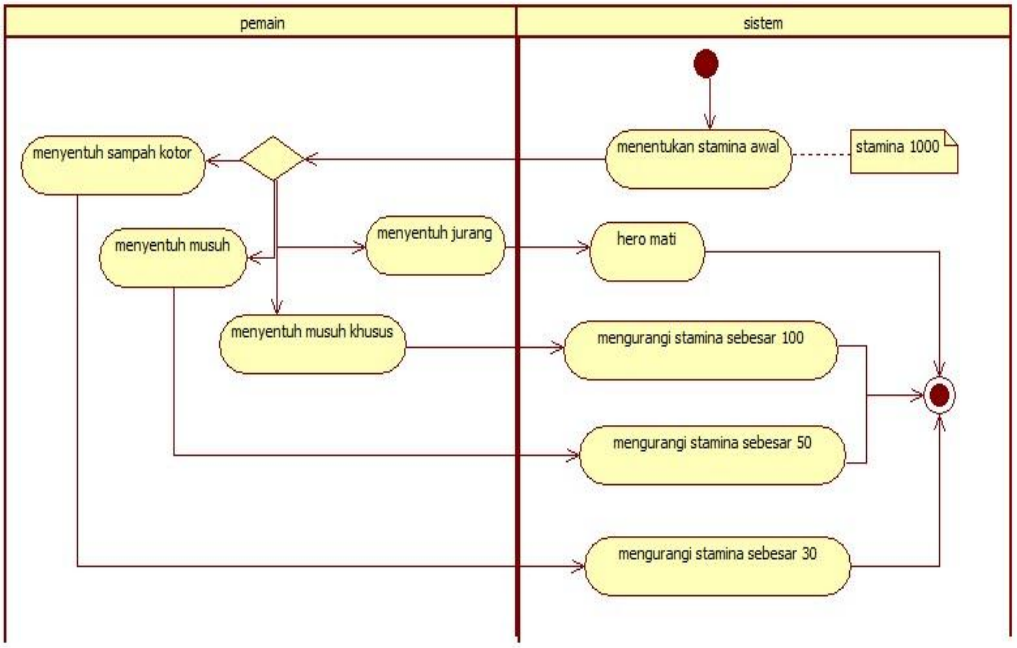
Activity diagram ini menjelaskan interaksi antara pemain dengan sistem.



Gambar 3. 9 *Activity* Diagram Main

g. *Activity* Diagram Perhitungan Stamina Hero

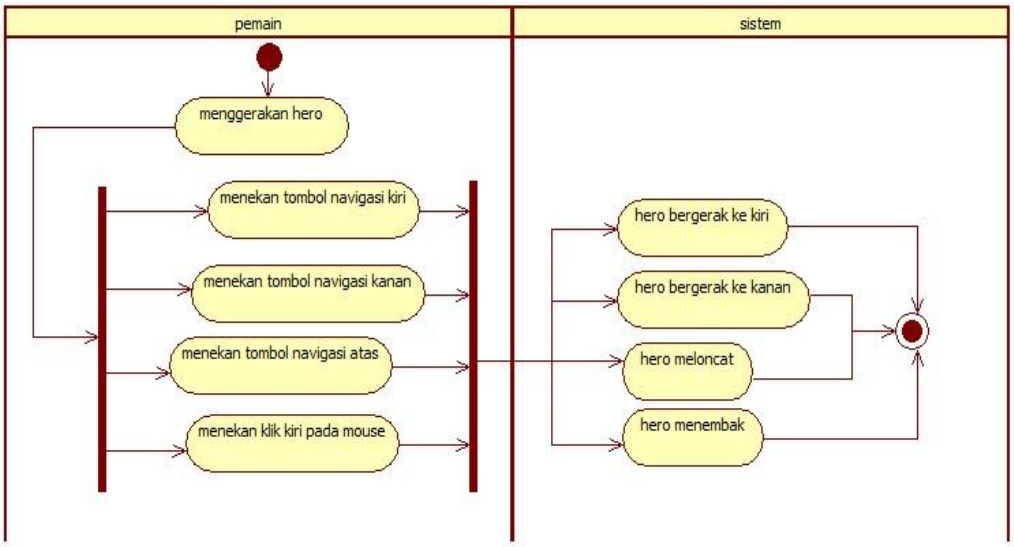
Activity diagram ini menjelaskan interaksi antara pemain dengan sistem.



Gambar 3. 10 *Activity* Diagram Perhitungan Stamina Hero

h. *Activity* Diagram Pergerakan *Hero*

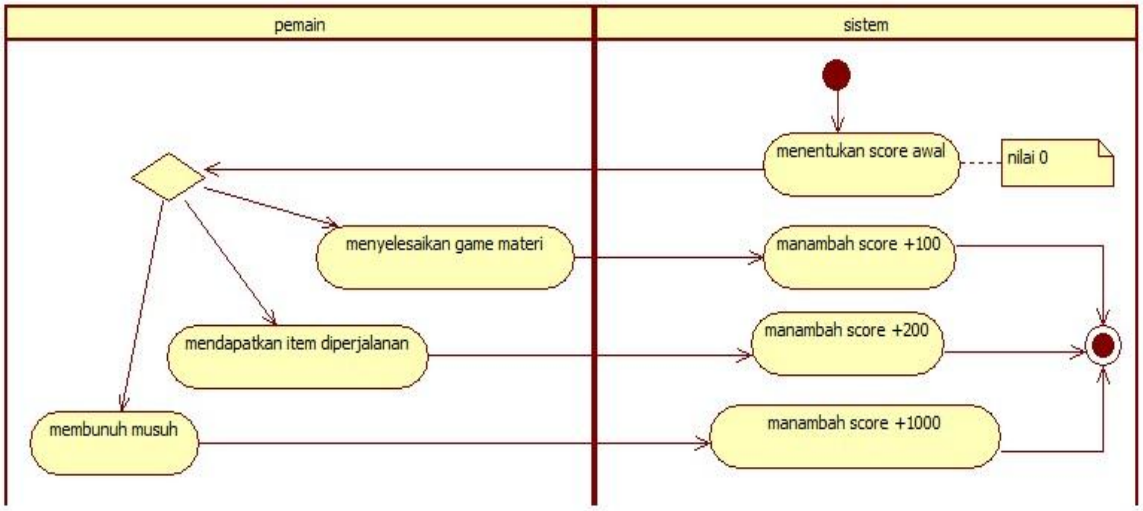
Activity diagram ini menjelaskan interaksi antara pemain dengan sistem.



Gambar 3. 11 *Activity* Diagram Pergerakan *Hero*

i. *Activity* Diagram Perhitungan *Score*

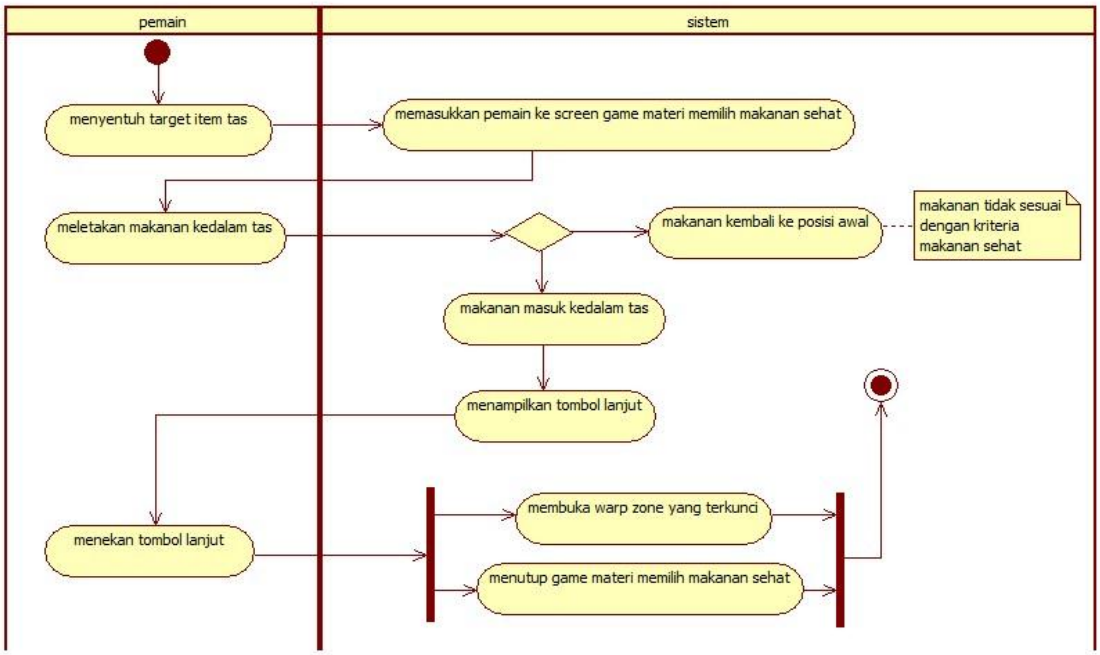
Activity diagram ini menjelaskan interaksi antara pemain dengan sistem.



Gambar 3. 12 *Activity* Diagram Perhitungan *Score*

j. *Activity* Diagram Memilih Makanan Sehat

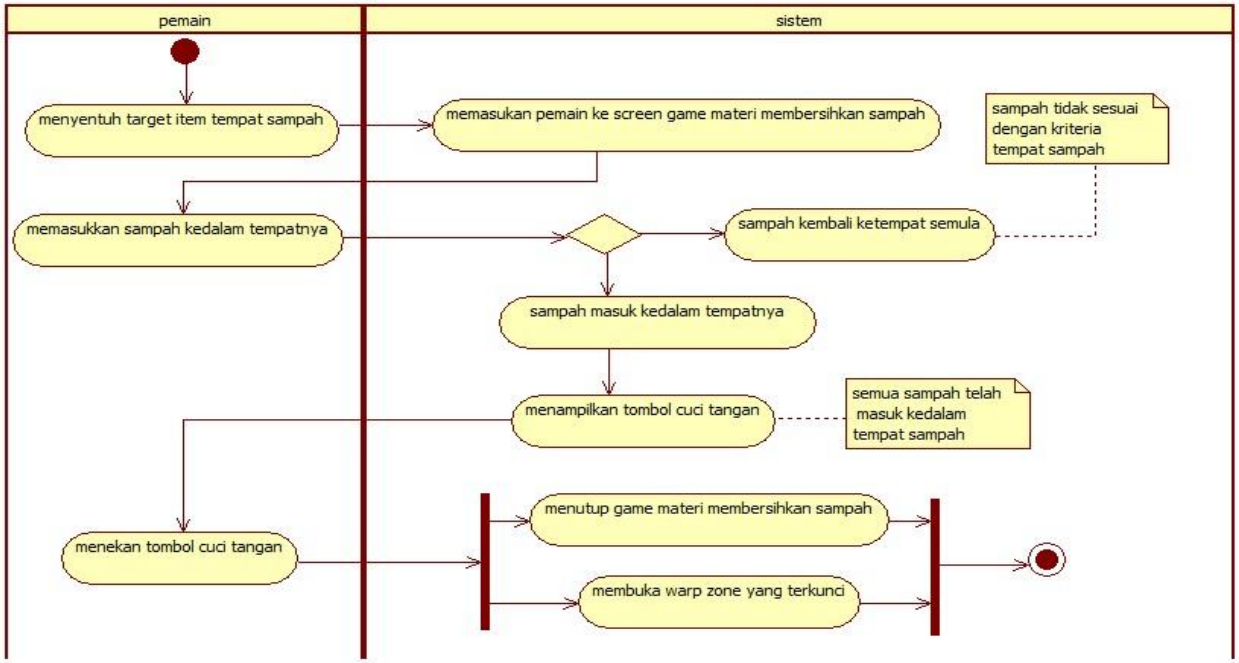
Activity diagram ini menjelaskan interaksi antara pemain dengan sistem.



Gambar 3. 13 *Activity* Diagram Memilih Makanan Sehat

k. *Activity* Diagram Membersihkan Sampah

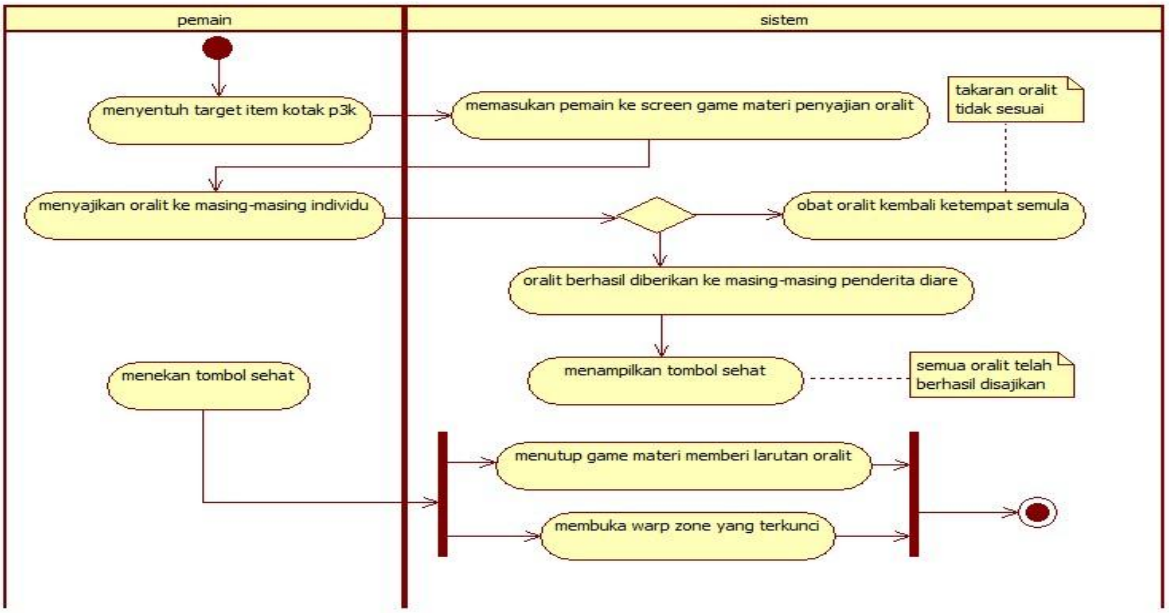
Activity diagram ini menjelaskan interaksi antara pemain dengan sistem.



Gambar 3. 14 *Activity* Diagram Membersihkan Sampah

l. *Activity* Diagram Penyajian Oralit

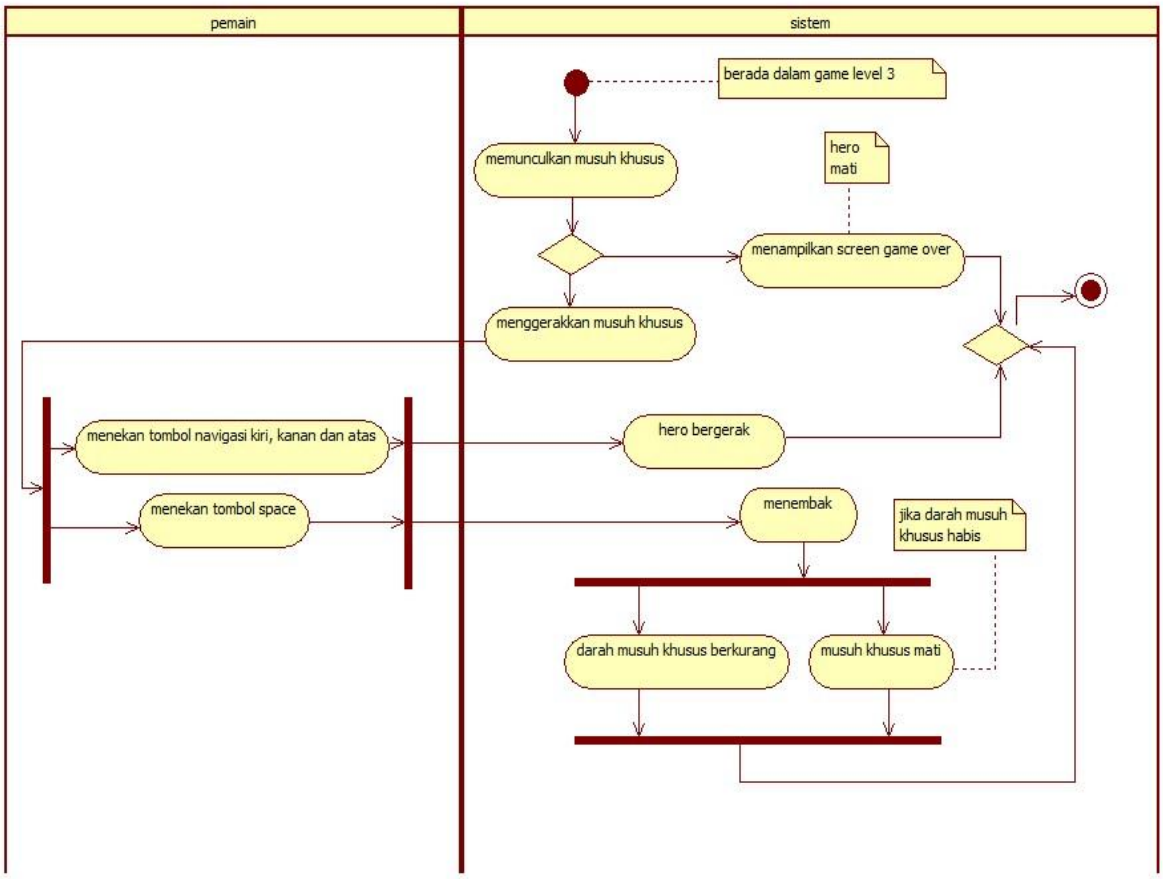
Activity diagram ini menjelaskan interaksi antara pemain dengan sistem.



Gambar 3. 14 *Activity* Diagram Penyajian Oralit

m. Activity Diagram *Enemy* AI

Activity diagram ini menjelaskan interaksi antara pemain sistem.

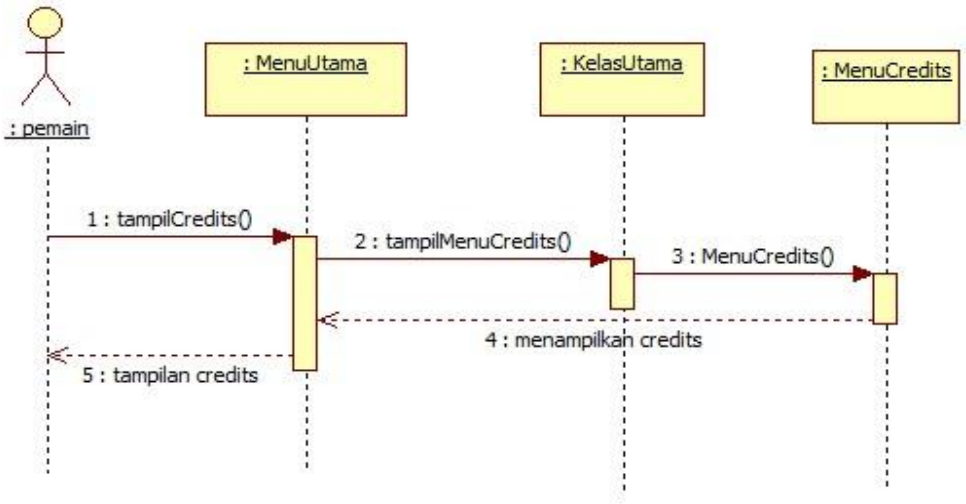


Gambar 3. 16 *Activity* Diagram *Enemy* AI

24. *Sequence Diagram*

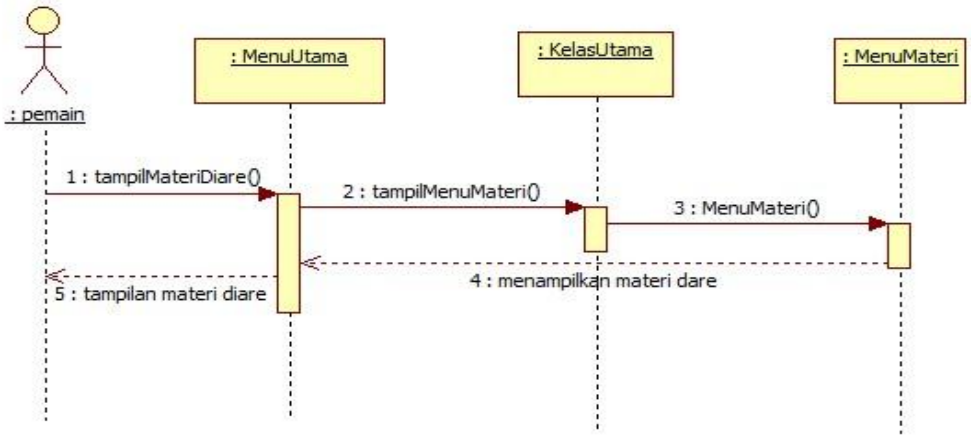
Sequence diagram menggambarkan interaksi antar masing-masing objek pada setiap *use case* dalam urutan waktu. Interaksi ini berupa pengiriman serangkaian data antar objek - objek yang saling berinteraksi.

a. *Sequence* Diagram Melihat *Credit*



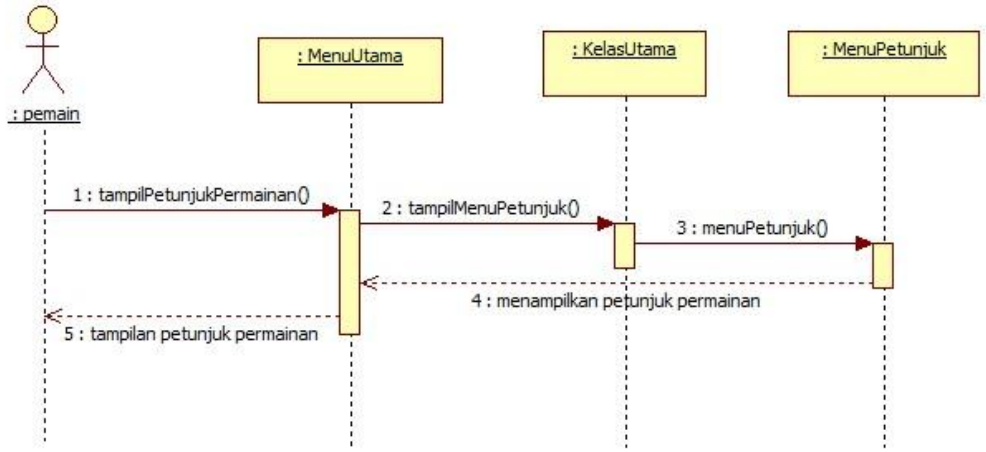
Gambar 3. 17 *Sequence* Diagram Melihat *Credit*

b. *Sequence* Diagram Melihat Materi Tentang Diare



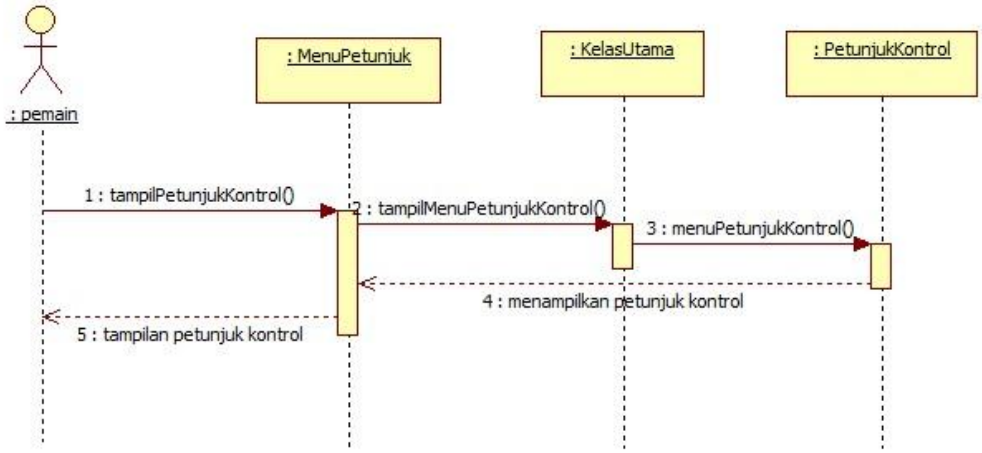
Gambar 3. 18 *Sequence* Diagram Melihat Materi Tentang Diare

c. *Sequence* Diagram Memilih Petunjuk



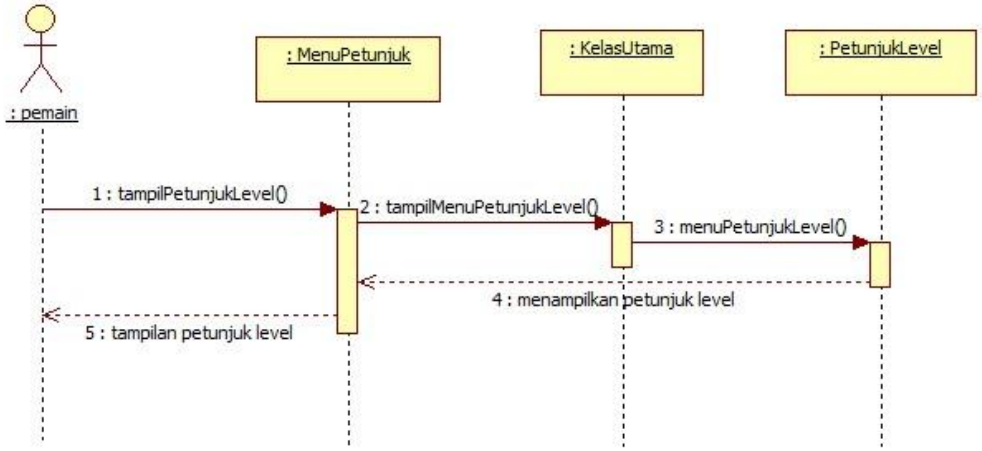
Gambar 3. 19 *Sequence* Diagram Memilih Petunjuk

d. *Sequence* Diagram Melihat Petunjuk Kontrol



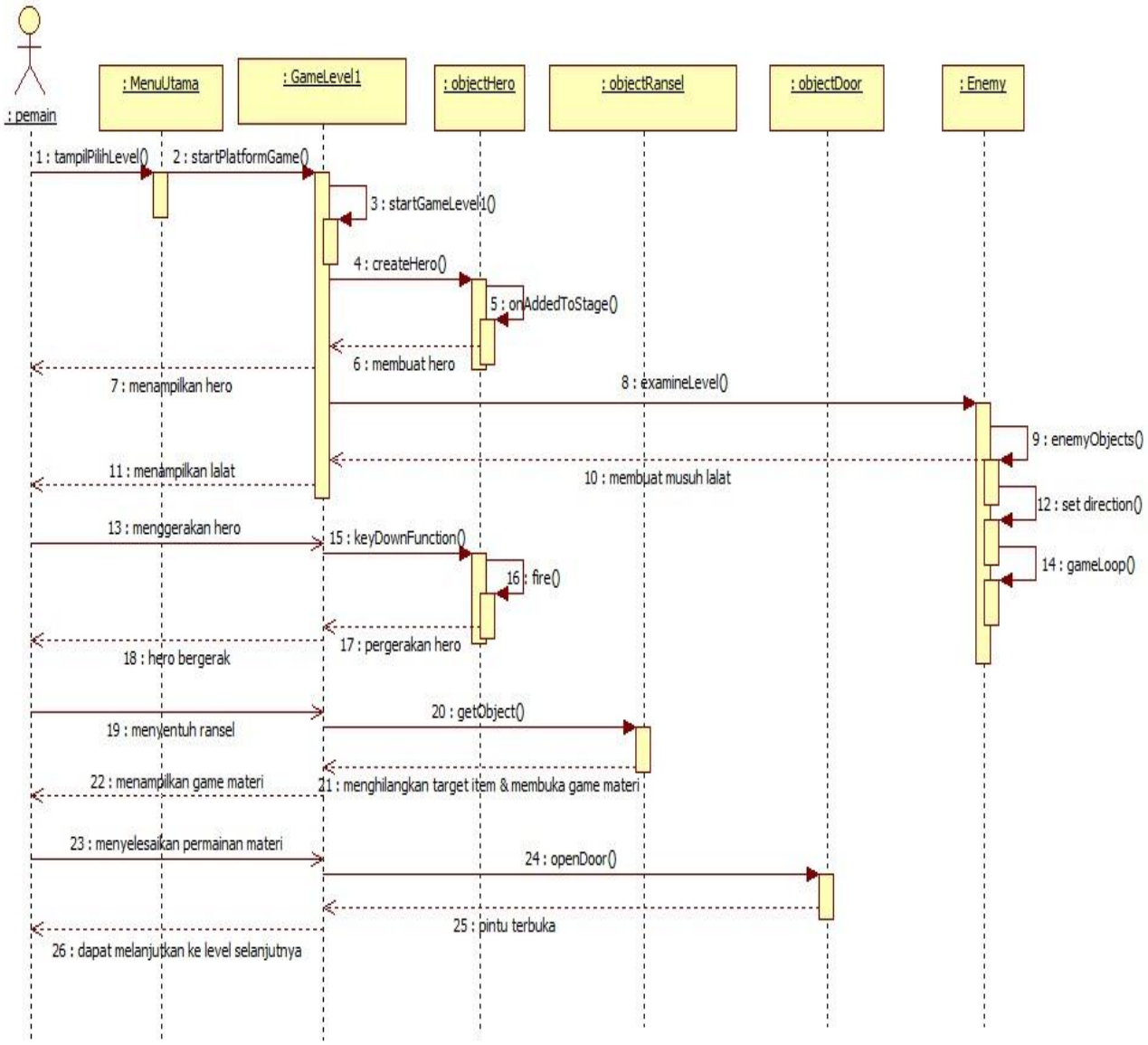
Gambar 3. 20 *Sequence* Diagram Melihat Petunjuk Kontrol

e. *Sequence* Diagram Petunjuk Level



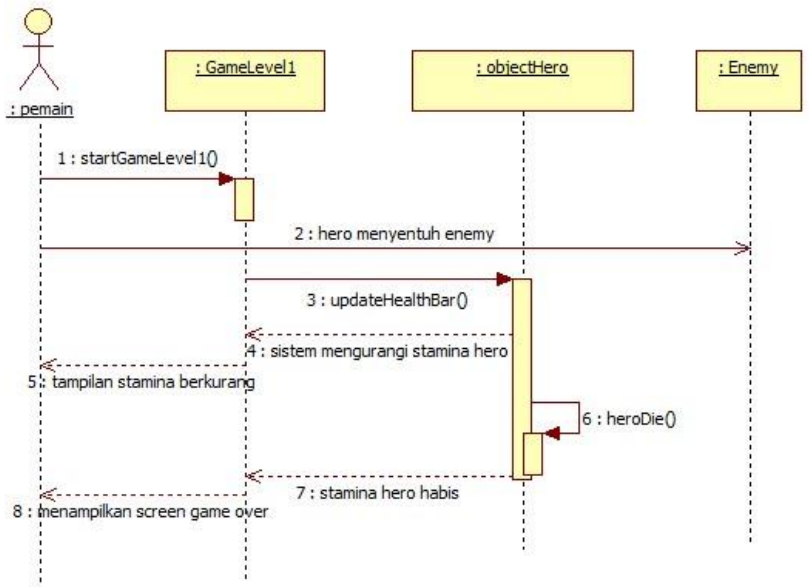
Gambar 3. 21 *Sequence* Diagram Petunjuk *Level*

f. *Sequence* Diagram Main



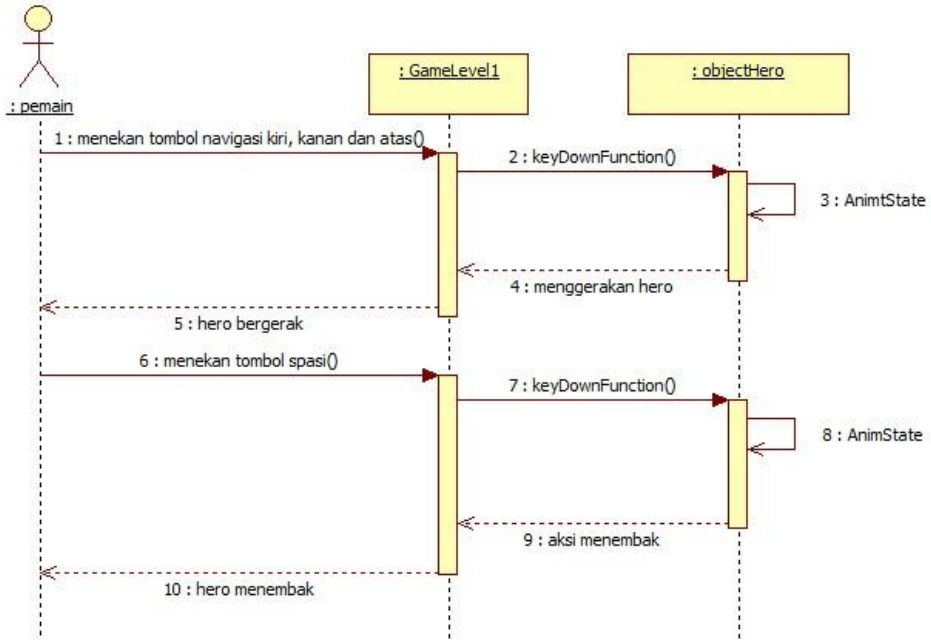
Gambar 3. 22 *Sequence* Diagram Main

g. *Sequence* Diagram Perhitungan Stamina *Hero*



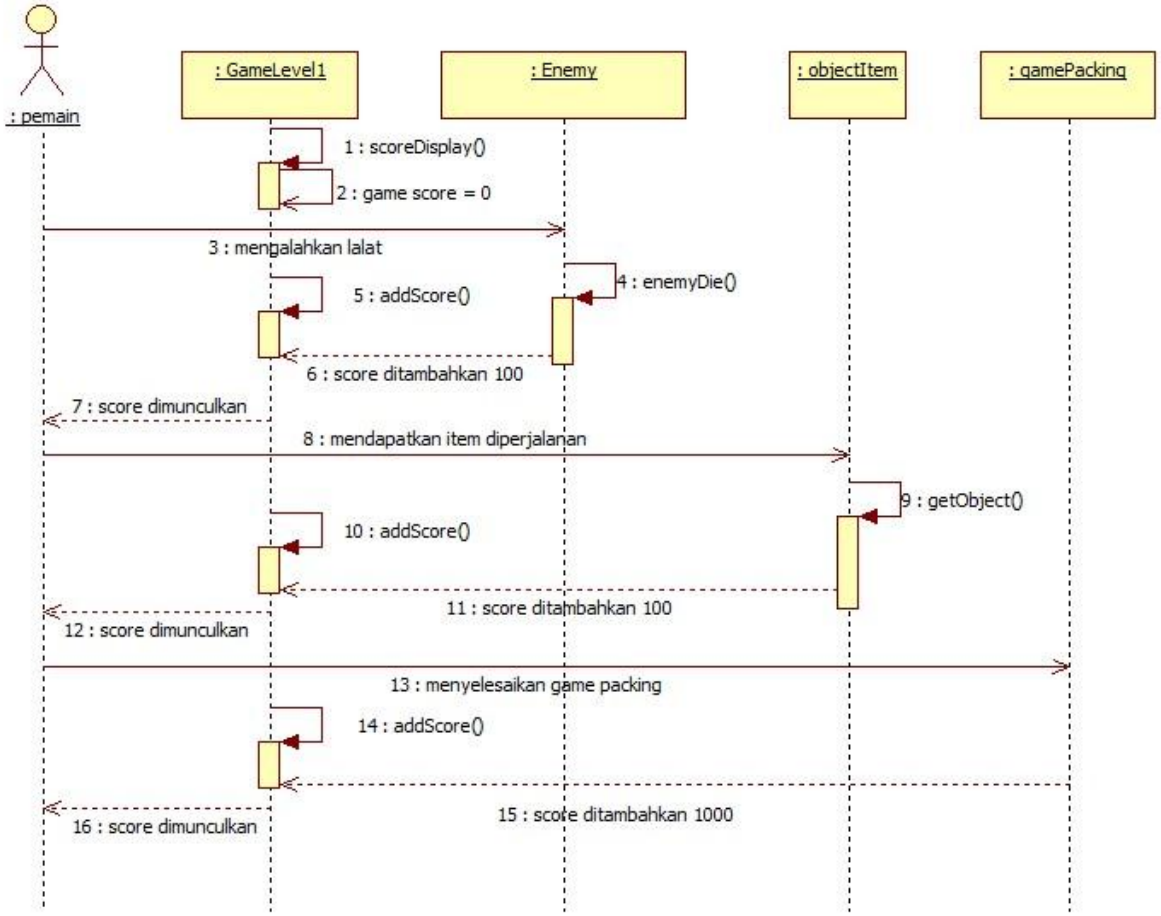
Gambar 3. 23 *Sequence* Diagram Perhitungan Stamina *Hero*

h. *Sequence* Diagram Pergerakan *Hero*



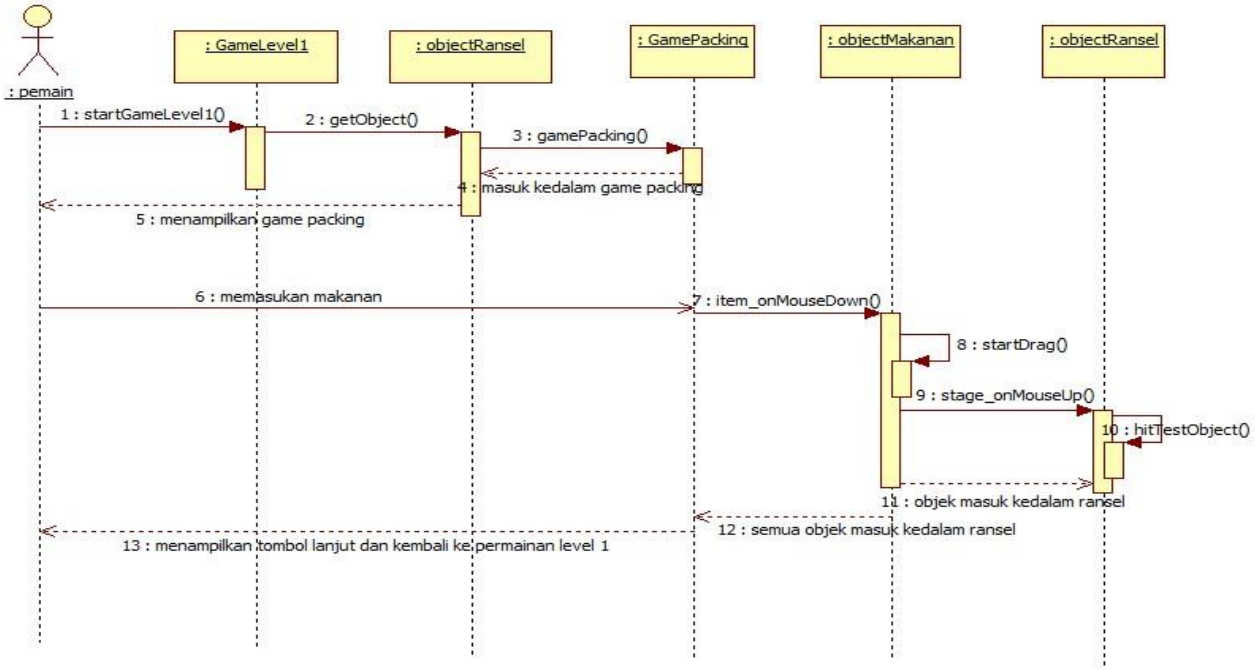
Gambar 3. 24 *Sequence* Diagram Pergerakan *Hero*

i. *Sequence* Diagram Perhitungan *Score*



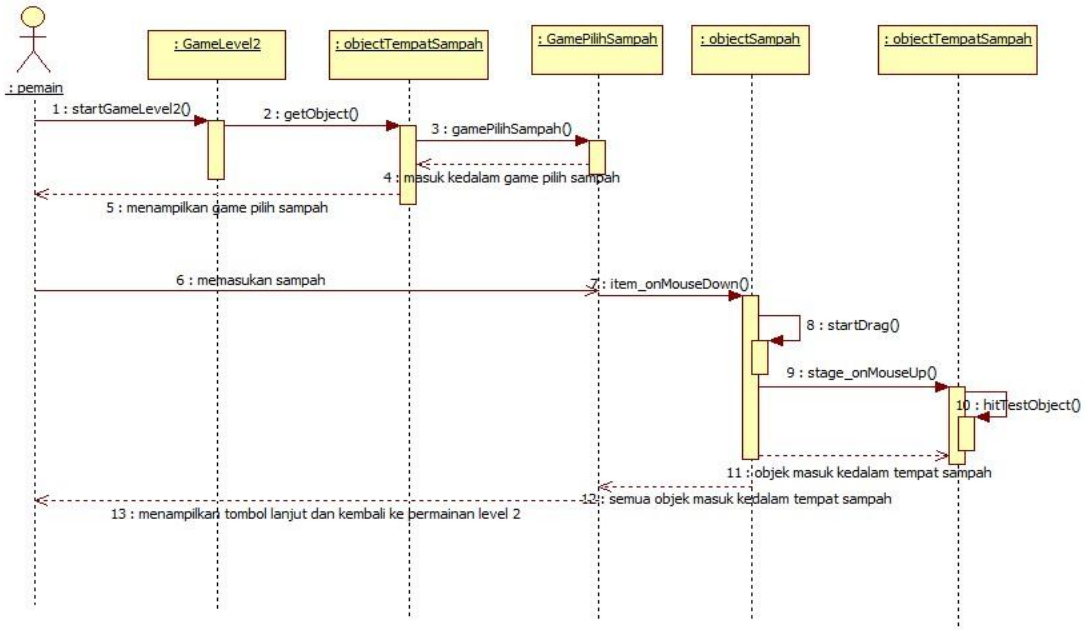
Gambar 3. 25 Sequence Diagram Perhitungan Score

j. *Sequence* Diagram Memilih Makanan Sehat



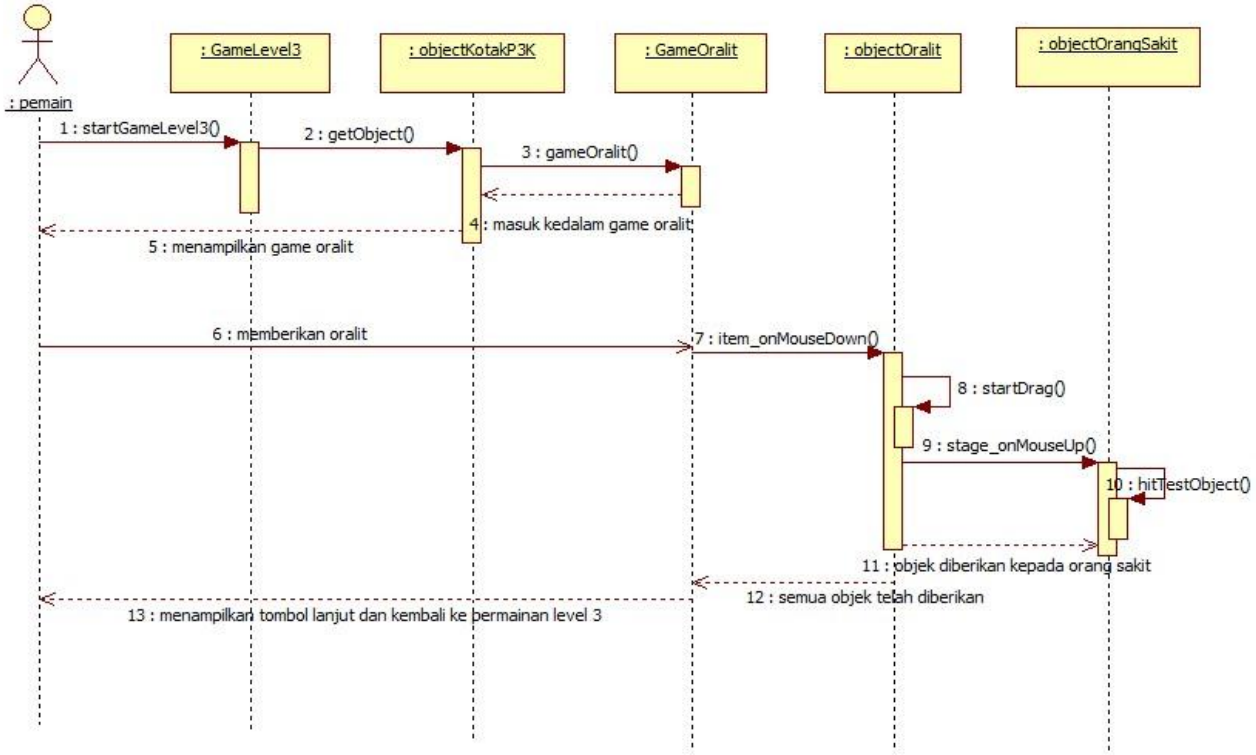
Gambar 3. 26 *Sequence* Diagram Memilih Makanan Sehat

k. *Sequence* Diagram Membersihkan Sampah



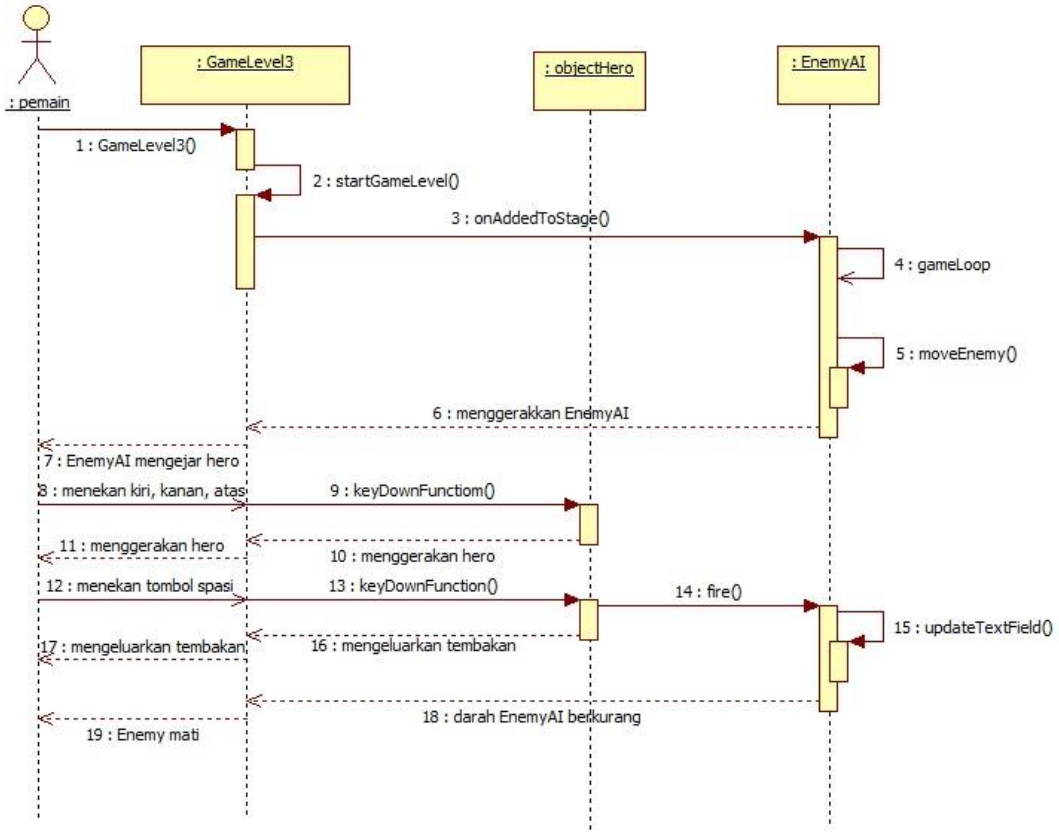
Gambar 3. 27 *Sequence* Diagram Membersihkan Sampah

l. *Sequence* Diagram Penyajian Oralit



Gambar 3. 28 *Sequence* Diagram Penyajian Oralit

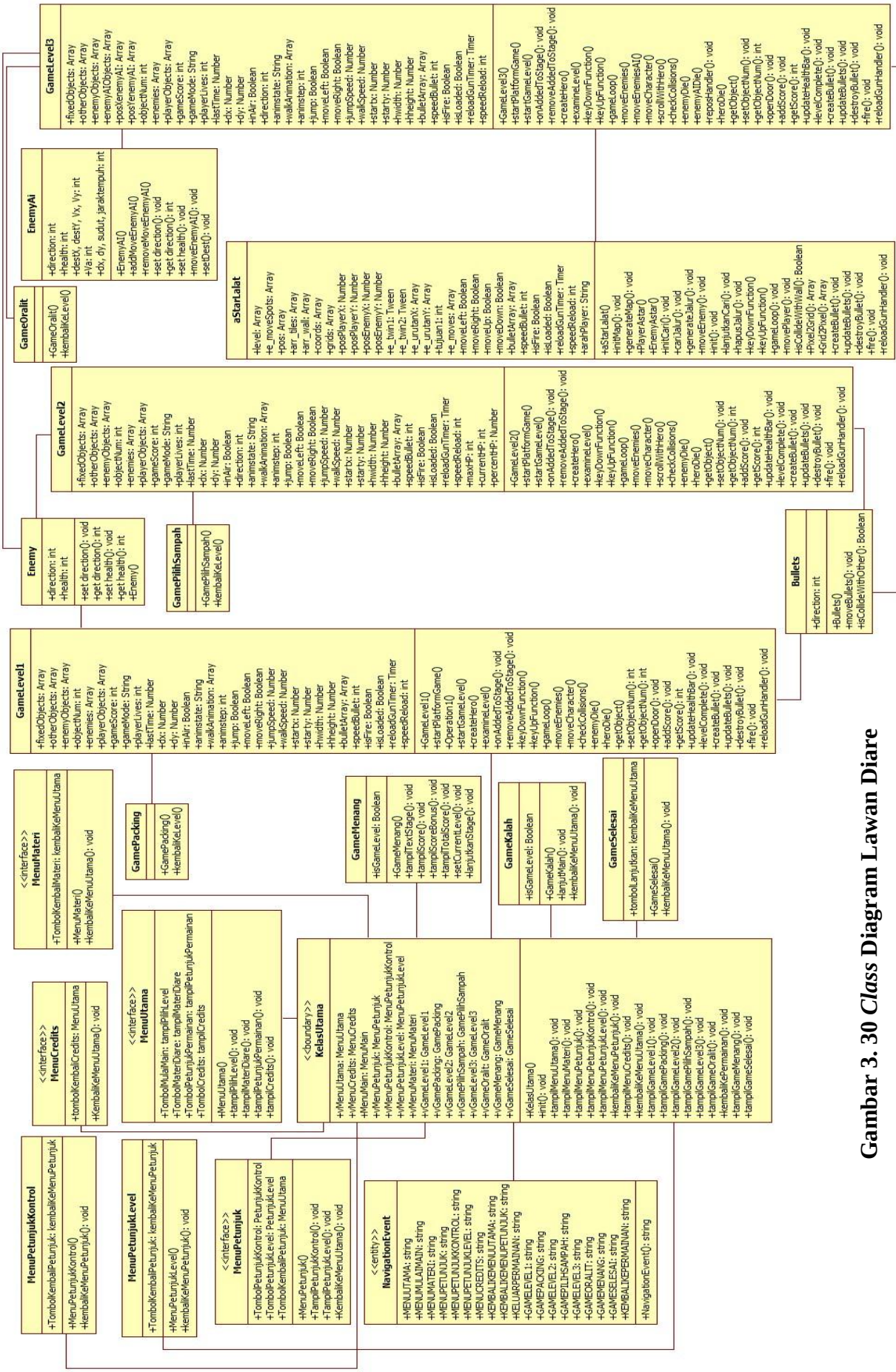
m. *Sequence* Diagram *Enemy* AI



Gambar 3. 29 *Sequence* Diagram *Enemy* AI

25. *Class Diagram*

Diagram Kelas menggambarkan struktur dan hubungan antar objek-objek yang ada pada sistem. Struktur itu meliputi atribut-atribut dan method-method yang ada pada masing-masing kelas, sedangkan hubungnnya meliputi pewarisan asosiasi, generalilasi.



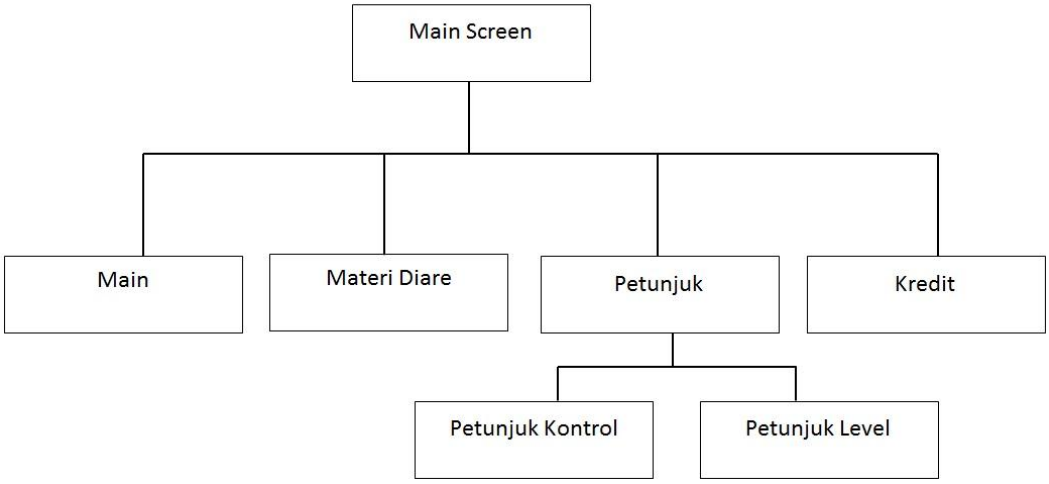
Gambar 3. 30 *Class* Diagram Lawan Diare

3.4 Perancangan Sistem

Perancangan sistem adalah merancang atau mendesain suatu sistem yang isinya adalah langkah langkah operasi dalam proses pengolahan data dan proses prosedur-prosedur untuk mendukung operasi sistem.

3.4.1 Perancangan Struktur Menu

Merupakan fitur - fitur yang disediakan pada sistem yang dirancang dan untuk mempermudah dalam penggunaan sistem nantinya. Perancangan struktur menu dapat dilihat pada gambar berikut :

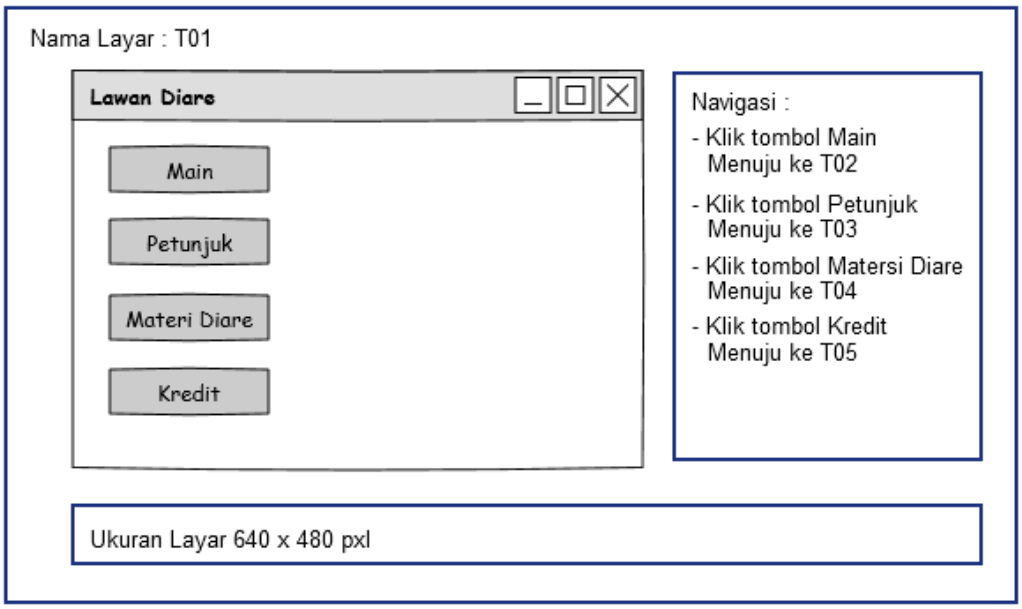


Gambar 3. 31 Perancangan Struktur Menu

3.4.2 Perancangan Antarmuka

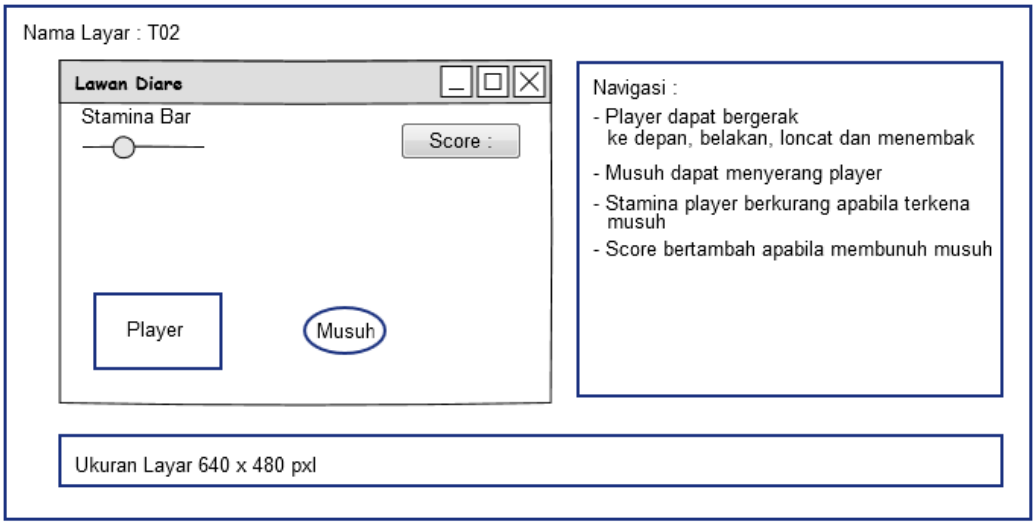
Merancang antarmuka merupakan bagian yang paling penting dari merancang sebuah system. Biasanya hal tersebut juga merupakan bagian yang paling sulit, karena dalam merancang antarmuka harus memenuhi tiga persyaratan yaitu antarmuka harus sederhana, antarmuka harus lengkap dan antarmuka harus memiliki kinerja yang cepat. Perancangan antarmuka dari game dapat dilihat pada gambar berikut :

1. **Perancangan Antarmuka *MainScreen***



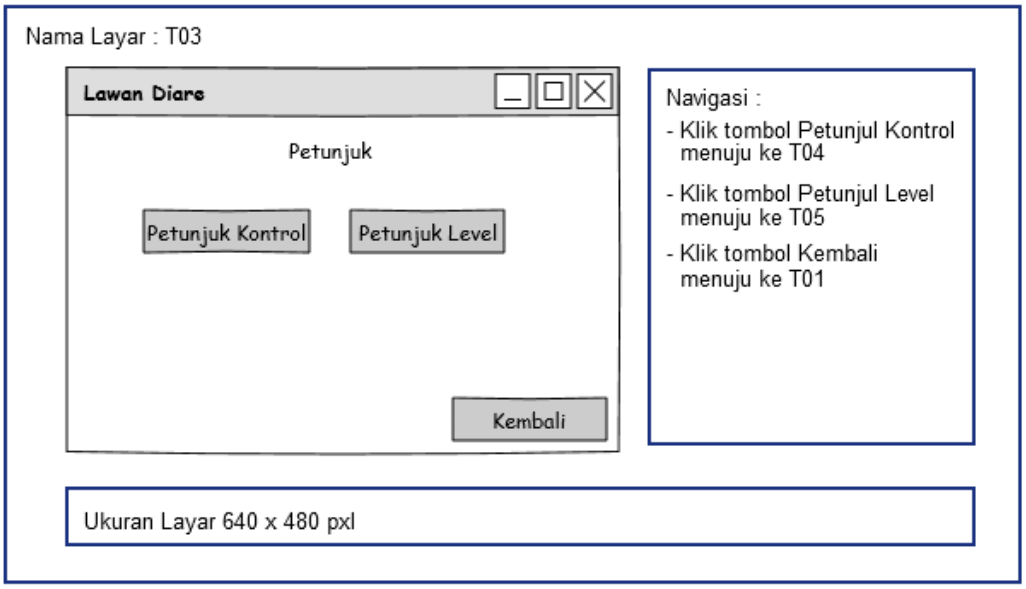
Gambar 3. 32 Perancangan Antarmuka *MainScreen*

2. Perancangan Antarmuka Petualangan



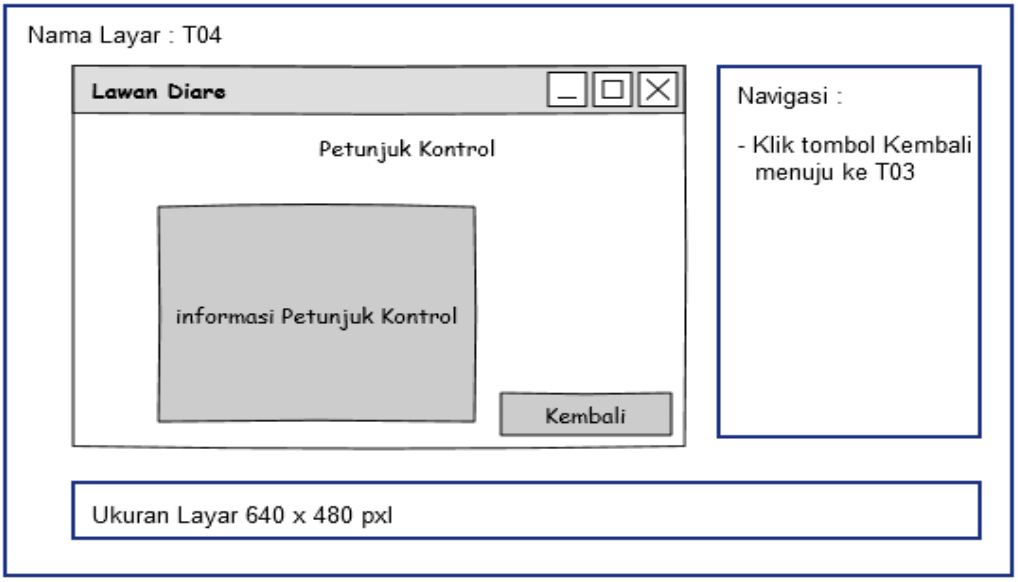
Gambar 3. 33 Perancangan Antarmuka Petualangan

3. Perancangan Antarmuka Petunjuk



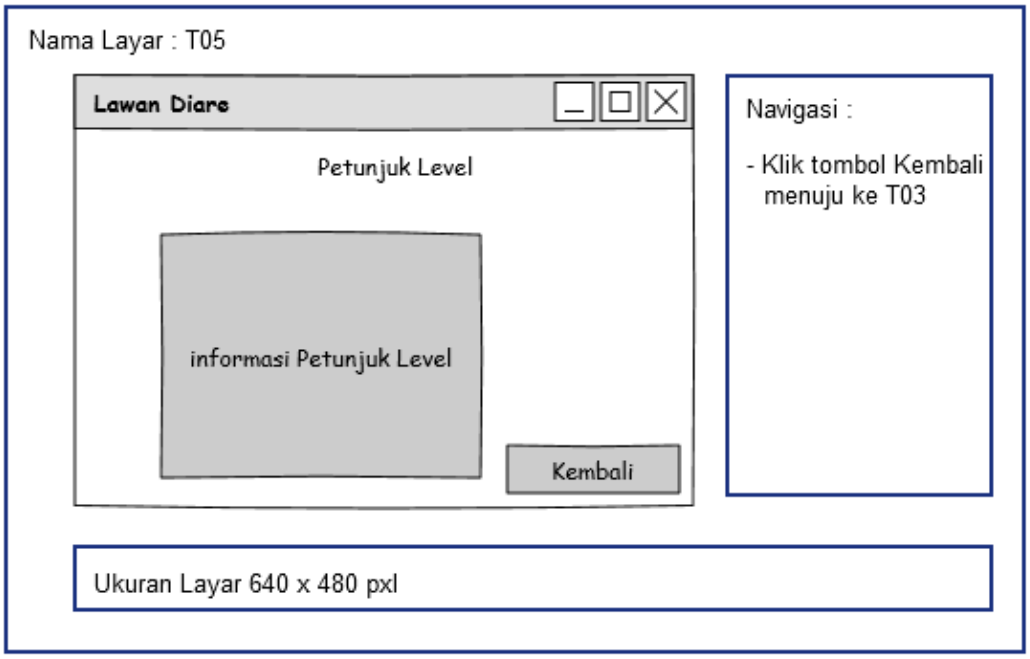
Gambar 3. 34 Perancangan Antrarmuka Petunjuk

4. Perancangan Antarmuka Petunjuk Kontrol



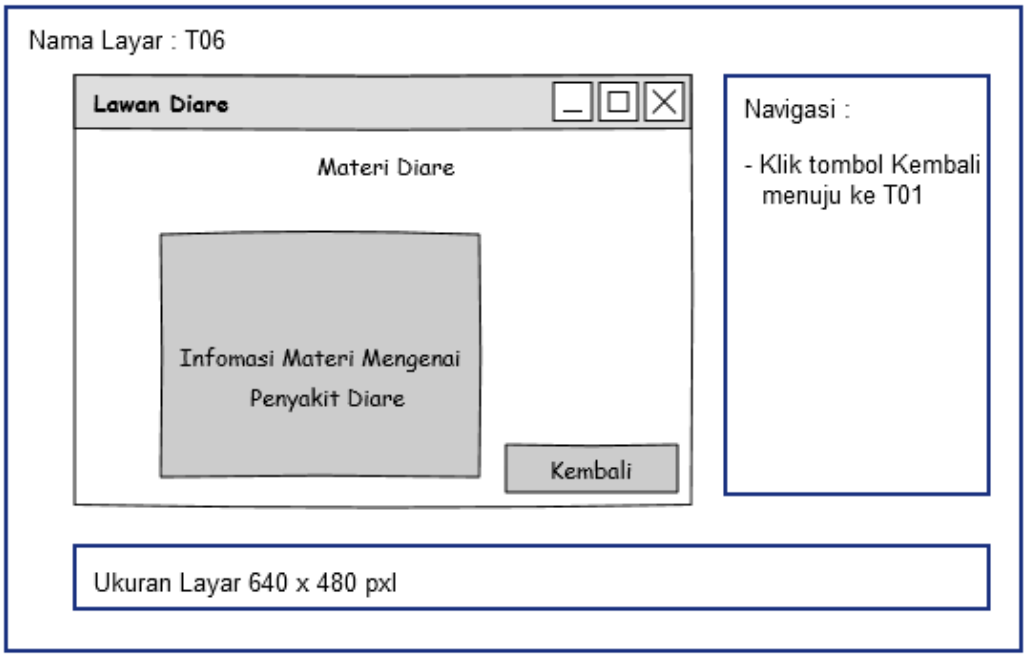
Gambar 3. 35 Perancangan Antarmuka Petunjuk Kontrol

5. Perancangan Antarmuka Petunjuk *Level*



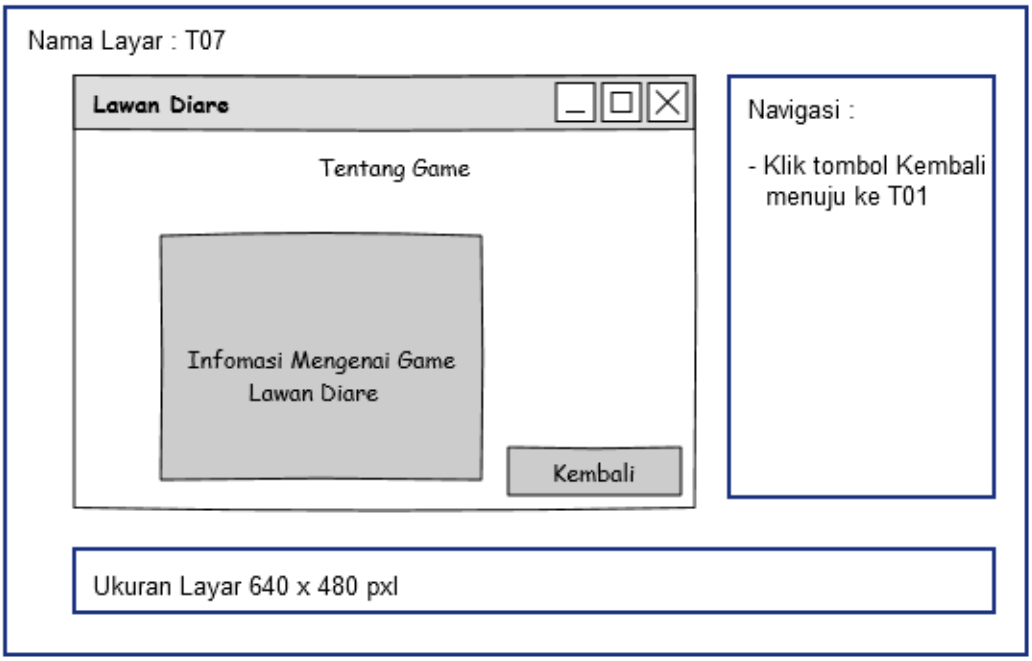
Gambar 3. 36 Perancangan Antarmuka Petunjuk *Level*

6. Perancangan Antarmuka Materi Diare



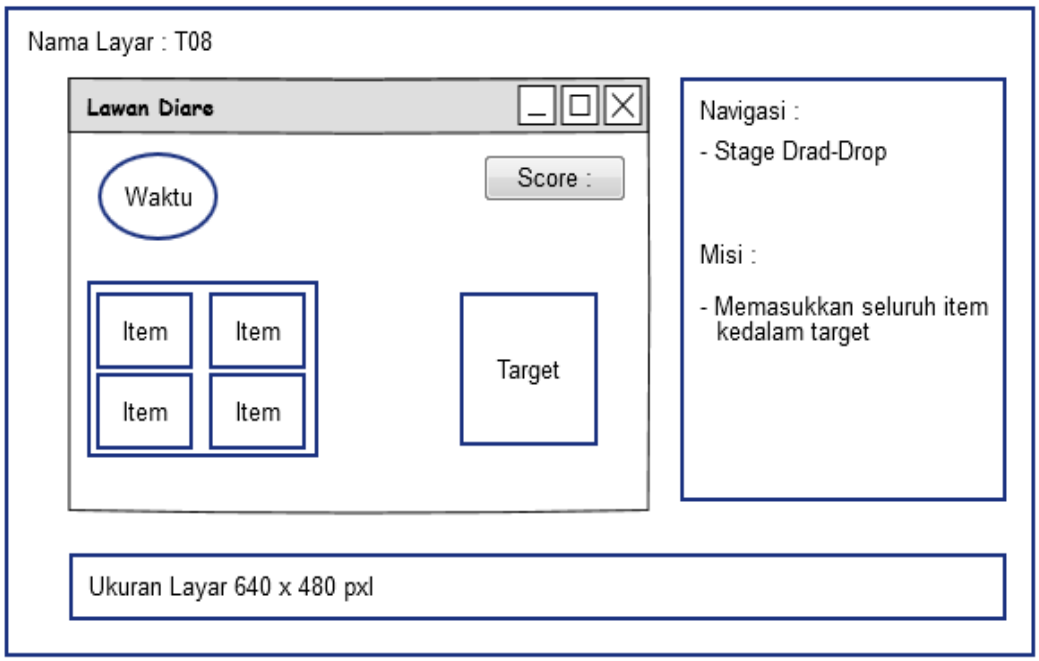
Gambar 3. 37 Perancangan Antarmuka Materi Diare

7. Perancangan Antarmuka Kredit



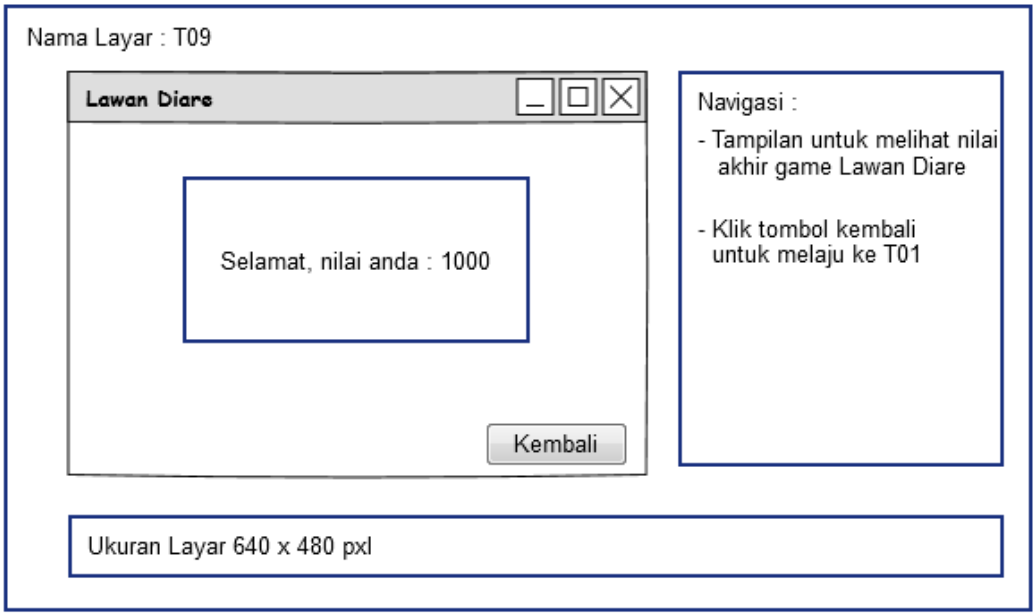
Gambar 3. 38 Perancangan Antarmuka *Credit*

8. Perancangan Antarmuka *Game* Materi



Gambar 3. 39 Perancangan Antarmuka *Game* Materi

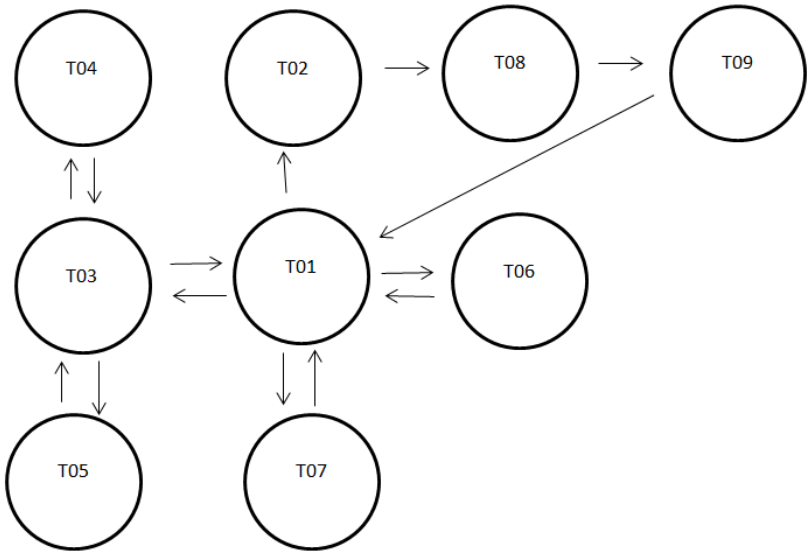
9. Perancangan Antarmuka Nilai Akhir



Gambar 3. 40 Perancangan Antarmuka Nilai Akhir

3.4.3 Perancangan Jaringan Semantik

Jaringan semantik digunakan untuk menghubungkan setiap tampilan (interface) dan pesan pada sebuah aplikasi yang dirancang. Untuk lebih jelasnya kita lihat gambar berikut :



Gambar 3. 41 Perancangan Jaringan Semantik