BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Dalam analisis sistem ini akan dilakukan berbagai macam analisis tentang game sejenis dan game yang akan dibangun. Pada game sejenis akan dilakukan analisis gameplay dan kekurangan serta kelebihan game tersebut; sedangkan analisis yang akan dilakukan pada game yang akan dibangun meliputi anlisis masalah serta kebutuhan fungsional dan non-fungsional.

3.1.1 Analisis Masalah

Sebelum membangun *game* "Sugie Man Penyelamat Sugar Glider", maka harus dilakukan analisa masalahnya terlebih dahulu yaitu meliputi:

Kurangnya pengetahuan masyarakat Indonesia tentang makanan *sugar glider*.

3.1.2 Analisis *Game* Sejenis

Analisis *game* sejenis dilakukan untuk memberikan gambaran system agar lebih mudah dalam pembangunan dan sebagai acuran dalam membangun *game* yang akan dicapai. *Game* sejeni yang dijadikan acuan dalam pembangunan *game* Sugie Man Penyelemat Sugar Glider adalah *Pokemon Blue*, *Grand Theft Auto*, dan *The Legend of Zelda The Minish Cap*.

3.1.2.1 Analisis *Game Pokemon Blue*

Game Pokemon ini berjenis *adventure* dan bersudut pandang *top down* yang bergrafis dua dimensi(2D).

3.1.2.1.1 *Gameplay Pokemon Blue*

Game pokemon blue mempunyai *gameplay* sebagai berikut:

1. Diawal permainan *player* akan memilih 1 dari 3 *pokemon* yang ada.
2. Latar belakang *game* ini yaitu sebuah wilayah suatu pulau yang didalamnya terdapat beberapa kota dan ada juga goa.

3. Terdapat beberapa kota yang masing-masing memiliki tingkat kesulitan yang berbeda.
4. Untuk menambah darah *pokemon* bisa di simpan di rumah sakit atau memberikan *item* penambah darah.
5. Untuk meningkatkan level *pokemon* diharuskan melawan *pokemon-pokemon* yang berada di tanah, rumput, dan air.
6. Setiap memenangkan pertarungan *pokemon* akan mendapatkan *experience*.
7. Pada level-level tertentu *pokemon* dapat berevolusi.
8. Cara untuk menangkap *pokemon* liar adalah dengan melawan *pokemon* tersebut kemudian melempar *pokeball* ke *pokemon* lawan.
9. Untuk memperoleh mata uang didalam *game*, *player* diharuskan mengalahkan *pokemon trainer* lainnya.
10. Untuk dapat berpindah ke kota selanjutnya *player* diharuskan menyelesaikan *quest* yang ada.
11. Jika *player* kalah dalam pertempuran *player* akan langsung berpindah tempat ke rumah sakit.
12. Untuk memenangkan *game* ini *player* harus menyelesaikan *quest* dan melengkapi lencana atau *badges* dengan cara mengalahkan semua pokemon *trainer* yang berada dalam gym.

Kekurangan *Game Pokemon Blue*:

Pewarnaan pada permainannya masih hitam putih, selain itu tidak ada petunjuk permainan yang menyebabkan pemain kebingungan dalam memainkannya.



Gambar 3.1 Pokemon *Blue*

3.1.2.2 Analisis *Game Grand Theft Auto*

Game Grand Theft Auto ini berjenis *adventure* dan bersudut pandang *top down* yang bergrafik dua dimensi(2D).

3.1.2.2.1 *Gameplay Game Grand Theft Auto*

Game Grand Theft Auto mempunyai *gameplay* sebagai berikut:

1. Latar belakang *game Grand Theft Auto berupa jalan raya* yang dikelilingi oleh rumah-rumah, gedung- gedung bertingkat, dan lain sebagainya.
2. Cara memenangkan *game* ini *player* diharuskan menyelesaikan semua *quest* yang ada.
3. Setiap *quest* mempunyai tingkat kesulitan yang berbeda.
4. Jika gagal dalam menjalani *quest*, *player* dapat memilih untuk mengulang atau mencari *quest* yang lain.
5. Jika *player* tertangkap polisi maka akan di penjara dan semua *item* yang ada pada *player* akan di ambil.
6. Jika *player* terbunuh maka *player* akan kembali ke tempat penyimpanan terakhir.

Kekurangan *Game Grand Theft Auto*:

Backsound menjadi kurang jelas karena suara mesin mobil, klakson mobil, radio mobil, ban mobil ketika berbelok, dan lain-lain saling bersatu.



Gambar 3.2 *Grand Theft Auto*

3.1.2.3 Analisis *Game The Legend of Zelda The Minish Cap*

Game The Minish Cap ini berjenis *adventure* dan bersudut pandang *top down* yang bergrafis dua dimensi(2D).

3.1.2.3.1 *Gameplay The Legend of Zelda The Minish Cap*

Game the minish cap mempunyai *gameplay* sebagai berikut:

1. Karakter yang akan dimainkan *player* bernama link.
2. Terdapat beberapa kota yang masing-masing memiliki tingkat kesulitan yang berbeda.
3. Latar belakang *game* ini sebuah pulau yang didalamnya terdapat beberapa kota, hutan, goa, kuil, istana, puri, dan lainnya.
4. *Player* akan mati jika darahnya habis ketika diserang musuh.
5. Untuk berpindah ke tempat lainnya *player* diharuskan menyelesaikan *quest y*ang ada.
6. Dalam perjalanan menyelesaikan *quest*, karakter dihalangi oleh berbagai macam musuh yang dapat membunuh karakter. Karakter harus melawan musuh tersebut sebelum melanjutkan *quest*.

Kekurangan *Game The Legend of Zelda The Minish Cap*:

Terlalu banyak percakapan yang nantinya akan membuat *player* merasa bosan.



Gambar 3.3 *The Legend of Zelda The Minish Cap*

Tabel 3.1 Perbandingan Game

No	Komponen	*Game Pokemon Red*	*Game The Legend of Zelda The Minish Cap*	*Game Grand Theft Auto*	*Game Sugie Man Penyelamat Sugar Glider*
1	Bahasa	Bahasa Inggris	Bahasa Inggris	Bahasa Inggris	Bahasa Indonesia
2	Grafis	2D	2D	2D	2D
3	Suara	Backsound pada setiap map ada yang sama dan ada juga yang berbeda	Backsound pada setiap map ada yang sama dan ada juga yang berbeda	Backsound kurang sesuai dengan keadaan sekitar	Backsound pada setiap map berbeda
4	Tingkat	Tidak ada tingkatan level pada *game* ini	Tidak ada tingkatan level pada *game* ini	Tidak ada tingkatan level pada *game* ini	Terdapat 3 level pada *game* ini
5	Tombol	Pada *game* ini memiliki tombol-tombol yang berfungi sebagai : 1. Tombol navigasi 2. Tombol ok dan cancel	Pada *game* ini memiliki tombol-tombol yang berfungi sebagai : 1. Tombol navigasi 2. Tombol loncat 3. Tombol menggunakan *item* 4. Tombol cancel	Pada *game* ini memiliki tombol-tombol yang berfungi sebagai : 1. Tombol navigasi 2. Tombol menembnak	Pada *game* ini memiliki tombol-tombol yang berfungi sebagai : 1. Tombol navigasi 2. Tombol menyerang

6	Warna	Warna pada setiap map ada yang sama dan ada juga yang berbeda	Warna pada setiap map ada yang sama dan ada juga yang berbeda	Warna berbeda pada setiap objeknya	Setiap level memiliki warna yang berbeda karena backgroundnya berbeda-beda
7	Genre	Adventure	Adventure	Adventure	Adventure
8	Sudut Pandang	Top Down	Top Down	Top Down	Top Down

3.1.3 Analisis *game* yang di bangun

3.1.3.1 Pengenalan Tokoh Utama

Sugie Man merupakan tokoh utama atau karakter utama yang ada pada *game* sugieman penyelamat *sugar glider.*

3.1.3.2 Jalan Cerita

Petualangan dimulai ketika seorang anak dari tanah Papua yang bernama Sugie Man sedang bermain di hutan dan secara tidak sengaja melihat para pemburu liar menangkap *Sugar Glider* untuk dijual. Saat itu pula Sugie Man mengikuti para pemburu liar itu, dan bermaksud untuk menyelamatkan *Sugar Glider* tersebut untuk di kembalikan ke habitat aslinya.

3.1.3.3 Gameplay

3.1.3.3.1 Skenario dan misi: Level 1 (Hutan Papua)

Latar belakang pada stage ini yaitu berupa hutan. Di stage ini pemain akan bertemu dengan hewan-hewan darat yang harus di hindarinya. Untuk dapat melanjutkan ke level berikutnya pemain harus mengumpulkan 5 buah pear sebelum waktu habis.

Misi:

1. Latar belakang berupa hutan.
2. Mengambil 5 buah pear untuk dapat melanjutkan ke level berikutnya dan akan mengeluarkan tulisan dan suara yes.
3. Terdapat 1 buah makanan khas papua yaitu papeda.
4. Mengambil makanan khas papua untuk menambah darah sugie man sebesar 20, akan mengeluarkan tulisan dan suara nyam.

5. Jika mengenai kalajengking, atau anjing darah pemain akan berkurang 20 dan akan mengeluarkan tulisan dan suara aduh.
6. Menghindari kalajengking, atau anjing pemburu liar untuk dapat melanjutkan perjalanan dan level berikutnya.
7. Jika *player* menyelesaikan permainan dalam waktu maksimal 20 detik, maka skor bertambah 50.
8. Jika waktu permainan habis, maka permainan berakhir.

Adapun tingkatan musuh yang ada pada stage 1 dapat dilihat pada tabel 3.2

Tabel 3.2 tingkatan musuh pada stage 1

No	Nama	Serangan
1	Kalajengking	20
2	Anjing	20

3.1.3.3.2 Skenario dan misi: Level 2 (Rawa Papua)

Latar belakang pada stage ini yaitu berupa rawa. Di stage ini pemain akan bertemu dengan hewan-hewan rawa yang harus di hindarinya. Untuk dapat melanjutkan ke level berikutnya pemain harus mengumpulkan 5 buah melon, menyelamatkan 2 kadal yang terperangkap terlebih dahulu sebelum waktu habis.

Misi:

1. Latar belakang berupa rawa.
2. Mengambil 5 buah melon untuk dapat melanjutkan ke level berikutnya dan mengeluarkan tulisan dan suara yes.
3. Menyelamatkan 2 kadal yang terperangkap dengan cara mengambil kunci terlebih dahulu, jika berhasil menyelamatkan kadal mengeluarkan tulisan dan suara yes.
4. Mengambil kunci untuk menyelamatkan kadal yang terperangkap dan mengeluarkan tulisan dan suara yes.
5. Terdapat 2 buah makanan khas papua yaitu papeda.
6. Mengambil makanan khas papua yang ada untuk menambah darah sugie man sebesar 20 dan mengeluarkan tulisan dan suara nyam.
7. Jika mengenai buaya, atau ular darah pemain akan berkurang 20 dan mengeluarkan tulisan dan suara aduh.

8. Menghindari buaya, atau ular untuk dapat melanjutkan perjalanan dan level berikutnya.
9. Jika *player* menyelesaikan permainan dalam waktu maksimal 20 detik, maka skor bertambah 50.
10. Jika waktu permainan habis, maka permainan berakhir.

Adapun tingkatan musuh yang ada pada stage 2 dapat dilihat pada tabel 3.3

Tabel 3.3 tingkatan musuh pada stage 2

No	Nama	Serangan
1	Buaya	20
2	Ular	20

3.1.3.3.3 Skenario dan misi: Level 3 (Rumah Pemburu Liar)

Latar belakang pada stage ini yaitu berupa rawa. Di stage ini pemain akan bertemu dengan hewan darat atau bos yang harus di hindarinya. Pemain harus mengumpulkan 5 buah apel, menyelamatkan 4 sugar glider yang terperangkap dengan cara mengambil kunci yang sama warnanya dengan perangkap, setiap menyelamatkan sugar glider darah bos pemburu liar akan berkurang 20% dan 80%. Jika misi berhasil bos pemburu liar akan diserahkan ke polisi, pemain akan mendapatkan kembali *Sugar Glider* yang telah di tangkap oleh pemburu liar, dan kemudian dikembalikan ke habitatnya semula sebelum waktu habis.

Misi:

1. Latar belakang berupa rumah pemburu liar.
2. Mengambil 5 buah apel untuk dapat menyelesaikan permainan dan mengeluarkan tulisan dan suara yes.
3. Menyelamatkan 4 sugar glider yang terperangkap dengan cara mengambil kunci yang sama warnanya dengan perangkap terlebih dahulu dan mengeluarkan tulisan dan suara yes.
4. Mengambil kunci yang sama warnanya dengan kandang untuk menyelamatkan sugar glider yang terperangkap dan mengeluarkan tulisan dan suara yes.

5. Setiap menyelamatkan sugar glider, darah bos pemburu liar akan berkurang 20, jika berhasil menyelamatkan sugar glider mengeluarkan tulisan dan suara yes.
6. Terdapat 3 buah makanan khas papua yaitu papeda.
7. Mengambil makanan khas papua yang ada untuk menambah darah sugie man sebesar 20 dan mengeluarkan tulisan dan suara nyam.
8. Menghindari anjing, atau bos pemburu liar untuk dapat menyelesaikan permainan.
9. Melepaskan sugar glider ke habibat aslinya.
10. Jika *player* menyelesaikan permainan dalam waktu maksimal 20 detik, maka skor bertambah 50.
11. Jika waktu permainan habis, maka permainan berakhir.

Adapun tingkatan musuh yang ada pada stage 3 dapat dilihat pada tabel 3.4

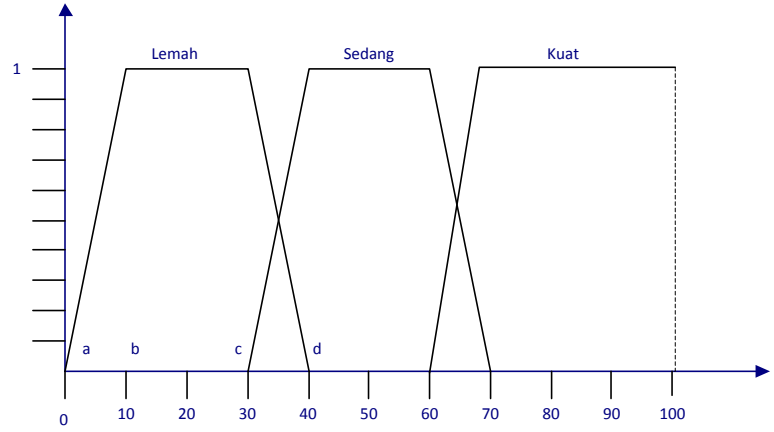
Tabel 3.4 tingkatan musuh pada stage 3

No	Nama	Darah	Serangan
1	Anjing	-	20
2	Boss pemburu liar	100	20

3.1.4 Analisis Logika *Fuzzy*

Penerapan Logika *Fuzzy* pada bos musuh

Aturan fungsi keanggotaan:



Gambar 3.4 Fungsi keanggotaan trapesium serangan *player* terhadap bos

Keterangan:

a. µlemah(x) = $0,\ x \geq 40$

$= \frac{x}{10}, 0 < x \leq 10$

$= 1, 10 < x \leq 30$

$= \frac{-(x-40)}{10}, 30 < x \leq 40$

b. µsedang(x) = $0, x \leq 30, x \geq 70$

$= \frac{x-30}{10}, 30 < x \leq 40$

$= 1, 40 < x \leq 60$

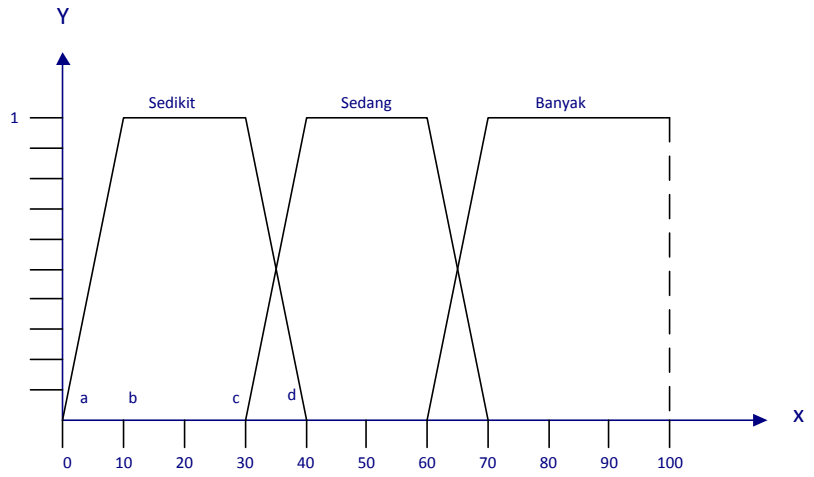
$= \frac{-(x-70)}{10}, 60 < x \leq 70$

c. µkuat(x) = $0, x \leq 60, x \geq 100$

$= \frac{x-60}{10}, 60 < x \leq 70$

$= 1, 70 < x \leq 90$

$= \frac{-(x-90)}{10}, 90 < x \leq 100$



Gambar 3.5 Fungsi keanggotaan trapesium darah bos

Keterangan:

a. µsedikit(x) = $0,\ x \geq 40$

$= \frac{x}{10}, 0 < x \leq 10$

$= 1, 10 < x \leq 30$

$= \frac{-(x-40)}{10}, 30 < x \leq 40$

b. μsedang(x) = $0, x \leq 30, x \geq 70$

$= \frac{x-30}{10}, 30 < x \leq 40$

$= 1, 40 < x \leq 60$

$= \frac{-(x-70)}{10}, 60 < x \leq 70$

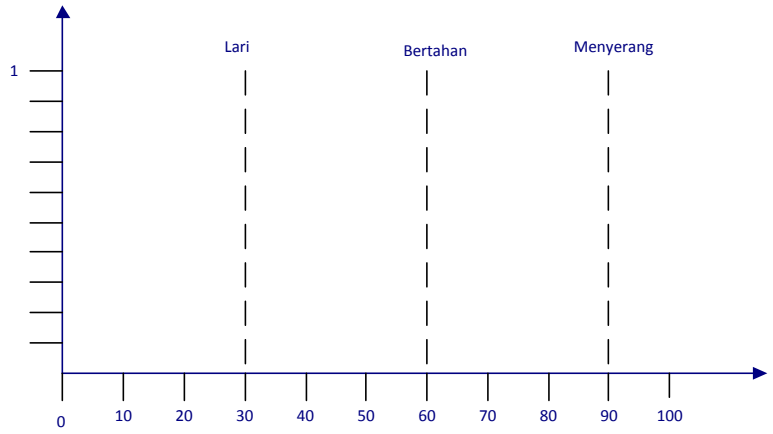
c. μbanyak(x) = $0, x \leq 60, x \geq 100$

$= \frac{x-60}{10}, 60 < x \leq 70$

$= 1, 70 < x \leq 90$

$= \frac{-(x-90)}{10}, 90 < x \leq 100$

Singleton kondisi bos:

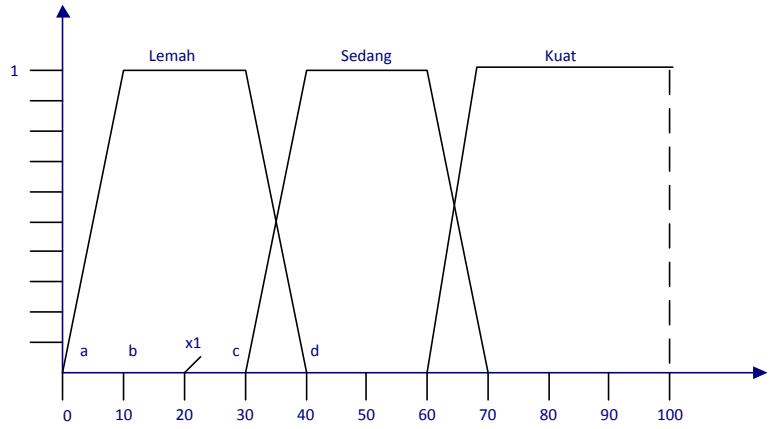


Gambar 3.6 singleton kondisi bos

Range kondisi bos: *lari(30-59)*, *bertahan(60-89)*, dan *kuat(≥90)*

a. Proses *Fuzzification* x1

Misalkan untuk serangan *player* terhadap bos menggunakan fungsi keanggotaan trapesium dengan tiga variabel linguistik yaitu: *lemah(0-40)*, *sedang(30-70)*, dan *kuat(≥60)*.



Gambar 3.7 Fungsi keanggotaan trapesium serangan *player* terhadap bos

1. Serangan *player* terhadap bos(x1) = 20 berada pada nilai linguistik lemah.
2. Keterangan :

   x = nilai *crisp*.

   a = titik awal dari fungsi keanggotaan trapesium

   b = titik ke dua yang searah dengan sumbu y dan berpotongan pada sumbu x jika ditarik garis secara vertikal.

   c = titik ke tiga yang searah dengan sumbu y dan berpotongan pada sumbu x jika ditarik garis secara vertikal.

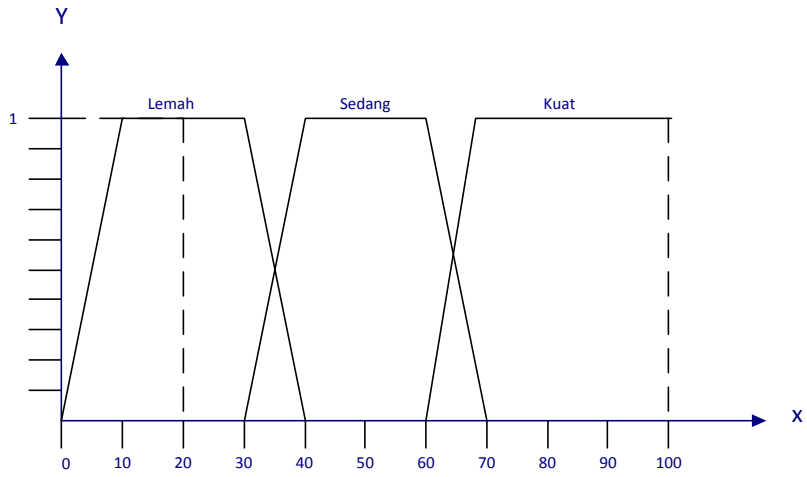
   d = titik ke empat dari fungsi keanggotaan trapesium.
3. Derajat keanggotaan untuk lemah dihitung menggunakan rumus µA(x) = $1, b \leq x \leq c$ dimana b = 10 dan c = 30

   Diketahui :

   x = 20, b = 10, c = 30

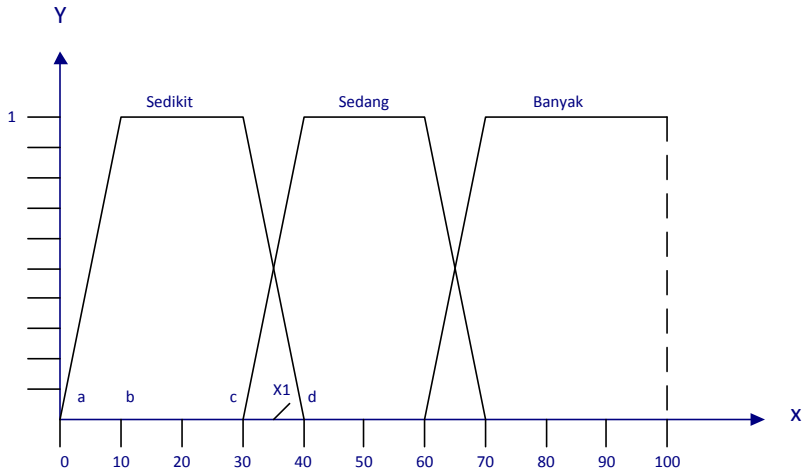
   µlemah(x) = $1, b \leq x \leq c$

   µlemah(20) = $1, 10 \leq 20 \leq 30$



Gambar 3.8 *Fuzzy* input serangan *player* terhadap bos

Misalkan untuk darah bos menggunakan fungsi keanggotaan trapesium dengan tiga variabel linguistik yaitu: *sedikit(0-40)*, *sedang(30-70)*, dan *banyak(60-100)*.



Gambar 3.9 Fungsi keanggotaan trapesium darah bos

1. Darah yang hilang dari, bos yaitu 34, berada pada nilai linguistik sedikit dan sedang.
2. Keterangan:

   x = nilai *crisp*.

   a = titik awal dari fungsi keanggotaan trapesium

   b = titik ke dua yang searah dengan sumbu y dan berpotongan pada sumbu x jika ditarik garis secara vertikal.

c = titik ke tiga yang searah dengan sumbu y dan berpotongan pada sumbu x jika ditarik garis secara vertikal.

d = titik ke empat dari fungsi keanggotaan trapesium.

3. Derajat keanggotaan untuk sedikit dihitung menggunakan rumus μA(x) = $\frac{-(x-d)}{d-c}$, $c < x \leq d$.

   Diketahui:

   x = 34

   c = 30

   d = 40

   μsedikit(x) = $\frac{-(x-d)}{d-c}$

   μsedikit(34) = $-\frac{\mathbf{34-40}}{\mathbf{40-30}} = \frac{\mathbf{6}}{\mathbf{10}}$

4. Derajat keanggotaan untuk sedang dihitung menggunakan rumus μA(x) = $\frac{x-a}{b-a}$, $a < x \leq b$.

   Diketahui:

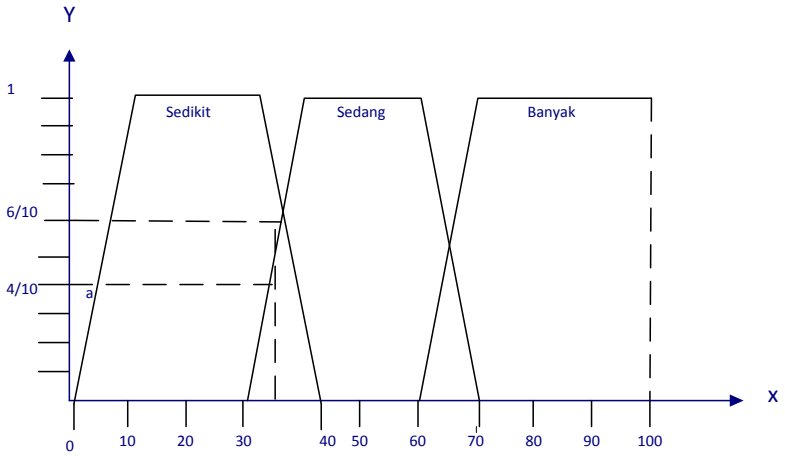
   x = 34

   a = 30

   b = 40

   μsedang(x) = $\frac{x-a}{b-a}$

   μsedang(34) = $\frac{\mathbf{34-30}}{\mathbf{40-30}} = \frac{4}{\mathbf{10}}$



Gambar 3.10 *Fuzzy* input darah bos

Jadi proses *fuzzificationnya* menghasilkan 3 *fuzzy* input:

1. Serangan *player* terhadap bos = lemah (1).
2. Darah bos = sedikit ($\frac{6}{10}$) dan sedang ($\frac{4}{10}$).

b. Proses Inference

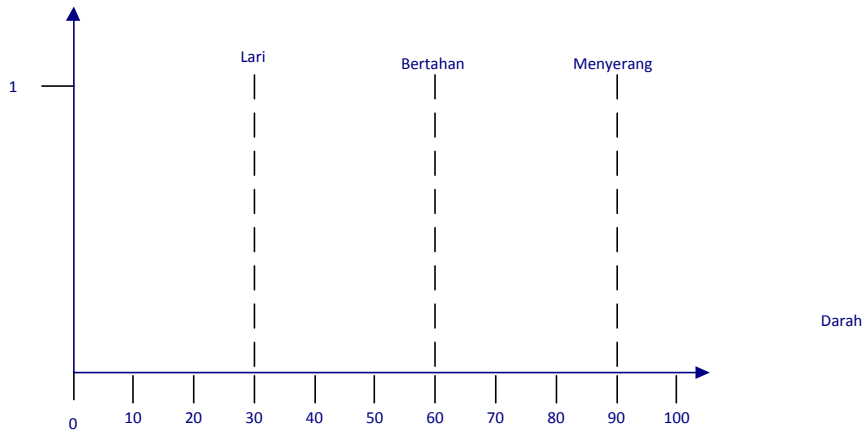
Proses Inference model Sugeno dengan menggunakan fungsi keanggotaan s*ingleton*. Fungsi keanggotaan *singleton* adalah fungsi keanggotaan yang memiliki derajat keanggotaan 1 pada suatu nilai *crisp* tunggal dan 0 pada semua nilai *crisp* yang lain.

Misal aturan *fuzzy* serangan *player* terhadap bos dengan darah bos sebagai berikut:

Tabel 3.5 Kondisi Bos

		Serangan *player* terhadap bos		
		lemah	Sedang	Besar
Darah	Sedikit	Lari	Lari	Lari
	Sedang	Bertahan	Bertahan	Lari
	Banyak	Menyerang	Menyerang	Bertahan

Dari aturan *fuzzy* keadaan serangan *player* terhadap bos dan darah misalkan didefiniskan fungsi *singleton* untuk kondisi bos yaitu:



Gambar 3.11 *singleton* kondisi bos

Dari tiga *fuzzy input* serangan *player* terhadap bos = lemah (1), darah = sedikit ($\frac{6}{10}$), dan darah = sedang ($\frac{4}{10}$), maka didapatkan 2 aturan *fuzzy* yang kemudian diaplikasikan menggunakan aturan *conjunction* untuk memilih derajat keanggotaan minimum:

a. Jika serangan *player* terhadap bos = lemah (1) AND darah = sedikit ($\frac{6}{10}$)

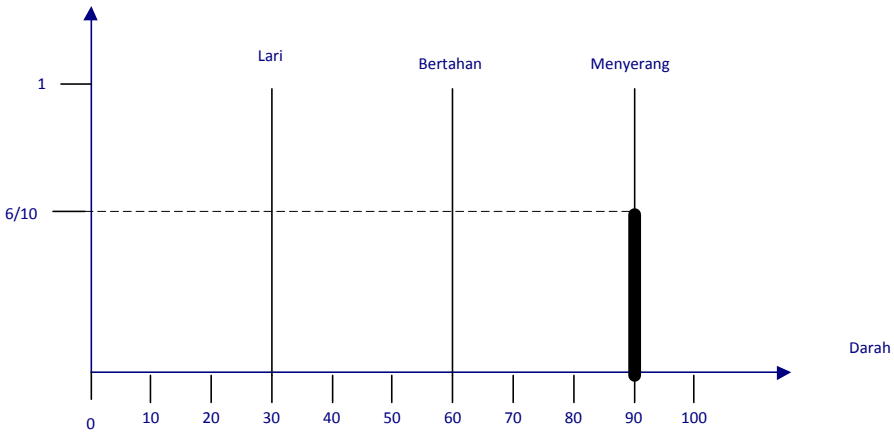
   THEN kondisi bos = menyerang ($\frac{6}{10}$).

b. Jika serangan *player* terhadap bos = lemah (1) AND darah = sedang ($\frac{4}{10}$)

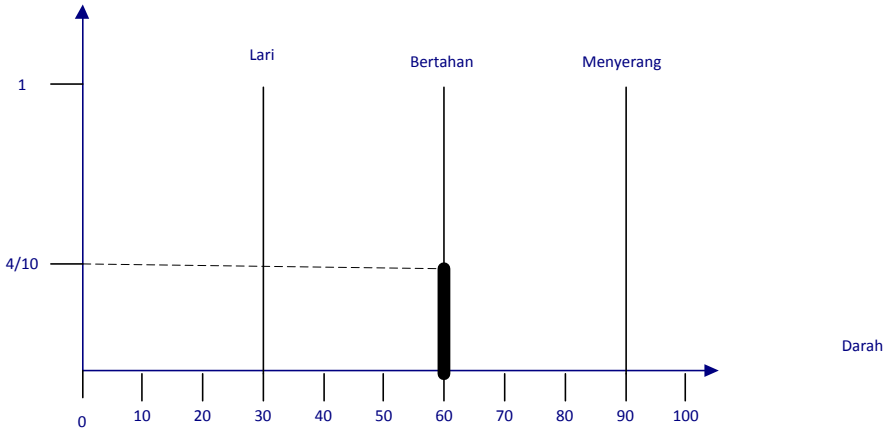
   THEN kondisi bos = bertahan ($\frac{4}{10}$).

Jika ada kondisi bos yang sama maka di lakukan penyaringan nilai maksimum dari kondisi bos dengan menggunakan aturan *disjunction*.

Proses inferensi menggunakan model sugeno menghasilkan dua derajat keanggotaan sebagai berikut:



Gambar 3.12 kondisi bos menyerang



Gambar 3.13 kondisi bos bertahan

c. Proses *Defuzzification*

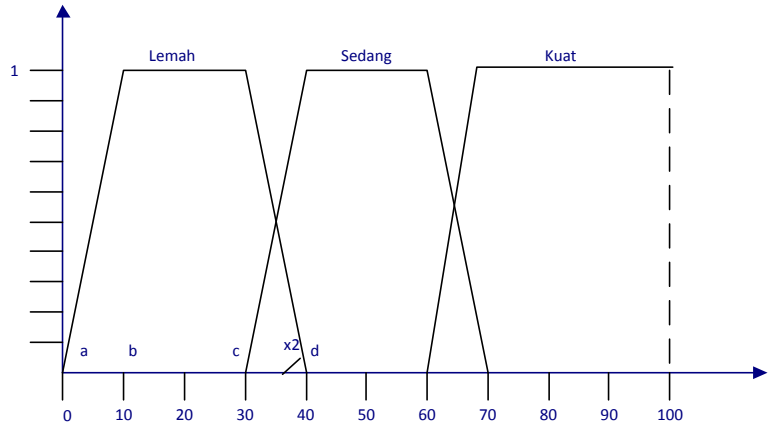
Proses *defuzzification* menggunakan proses *height method* maka dari dua *fuzzy* set menyerang ($\frac{6}{10}$), dan bertahan ($\frac{4}{10}$), maka dipilih nilai maksimumnya yaitu menyerang ($\frac{6}{10}$) karena nilai *crisp* sebesar 90 maka persentase darahnya adalah 90 dan disebut menyerang.

d. Kesimpulan

Jadi misalkan nilai *crisp* yang diterima oleh serangan *player* terhadap bos adalah 20, nilai *crisp* yang diterima oleh darah adalah 34, kondisi yang didapatkan adalah menyerang yaitu persentase darahnya ($\frac{6}{10}$), nilai ($\frac{6}{10}$) ini digunakan memberikan indikator tentang kondisi bos.

a. Proses *Fuzzification* x2

Misalkan untuk serangan *player* terhadap bos menggunakan fungsi keanggotaan trapesium dengan tiga variabel linguistik yaitu: *sedikit(0-40)*, *sedang(30-70)*, dan *banyak(≥60)*.



Gambar 3.14 Fungsi keanggotaan trapesium serangan *player* terhadap bos

1. Serangan *player* terhadap bos(x2) = 38 berada pada nilai linguistik sedikit.
2. Keterangan:

   x = nilai *crisp*.

   a = titik awal dari fungsi keanggotaan trapesium

   b = titik ke dua yang searah dengan sumbu y dan berpotongan pada sumbu x jika ditarik garis secara vertikal.

   c = titik ke tiga yang searah dengan sumbu y dan berpotongan pada sumbu x jika ditarik garis secara vertikal.

   d = titik ke empat dari fungsi keanggotaan trapesium.
3. Derajat keanggotaan untuk lemah dihitung menggunakan rumus µA(x) = $\frac{-(x-d)}{d-c}$, $c < x \leq d$.

   Diketahui:

   x = 38

   c = 30

   d = 40

   µlemah(x) $= \frac{-(x-d)}{d-c}$

   µlemah(38) = $-\frac{\mathbf{38-40}}{\mathbf{40-30}} = \frac{\mathbf{2}}{\mathbf{10}}$

4. Derajat keanggotaan untuk sedang dihitung menggunakan rumus µA(x) = $\frac{x-a}{b-a}$, $a < x \leq b$.

   Diketahui:

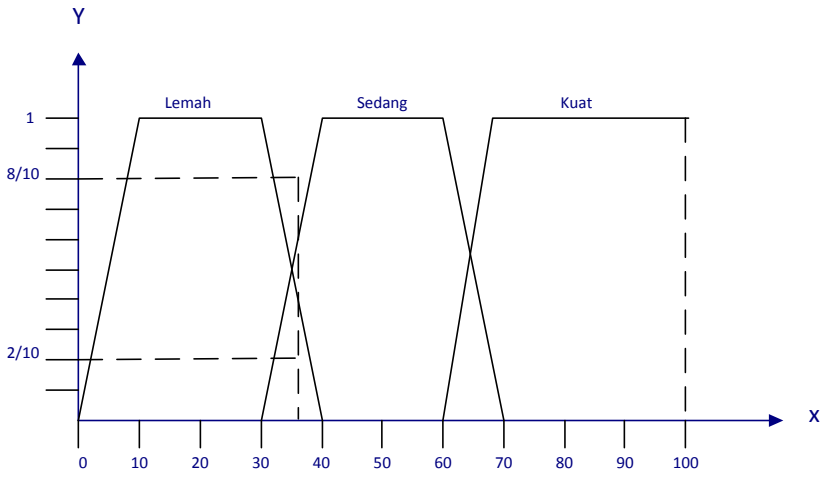
   x = 38

   a = 30

   b = 40

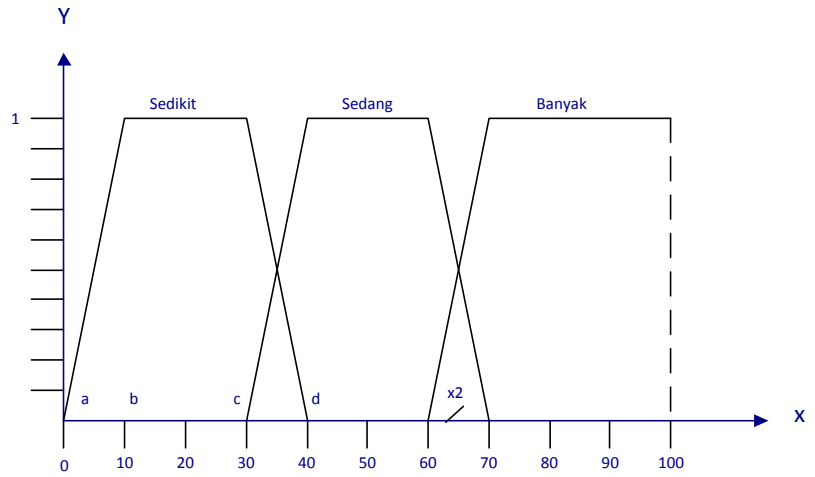
   µsedang(x) = $\frac{x-a}{b-a}$

   µsedang(38) = $\frac{\mathbf{38-30}}{\mathbf{40-30}} = \frac{8}{\mathbf{10}}$



Gambar 3.15 *Fuzzy* input serangan *player* terhadap bos

Misalkan untuk darah bos menggunakan fungsi keanggotaan trapesium dengan tiga variabel linguistik yaitu: *sedikit(0-40), sedang(30-70),* dan *banyak(60-100).*



Gambar 3.16 Fungsi keanggotaan trapesium darah bos

1. Darah yang hilang dari, bos yaitu 64, berada pada nilai linguistik sedikit dan sedang.

2. Keterangan:

   x = nilai *crisp.*

   a = titik awal dari fungsi keanggotaan trapesium

   b = titik ke dua yang searah dengan sumbu y dan berpotongan pada sumbu x jika ditarik garis secara vertikal.

   c = titik ke tiga yang searah dengan sumbu y dan berpotongan pada sumbu x jika ditarik garis secara vertikal.

   d = titik ke empat dari fungsi keanggotaan trapesium.

3. Derajat keanggotaan untuk sedang dihitung menggunakan rumus μA(x) = $\frac{-(x-d)}{d-c}$, $c < x \leq d$.

   Diketahui:

   x = 64

   c = 60

   d = 70

   μsedang(x) = $\frac{-(x-d)}{d-c}$

   μsedang(64) = $-\frac{\mathbf{64-70}}{\mathbf{70-60}} = \frac{\mathbf{6}}{\mathbf{10}}$

4. Derajat keanggotaan untuk banyak dihitung menggunakan rumus μA(x) = $\frac{x-a}{b-a}$, $a < x \leq b$.

   Diketahui:

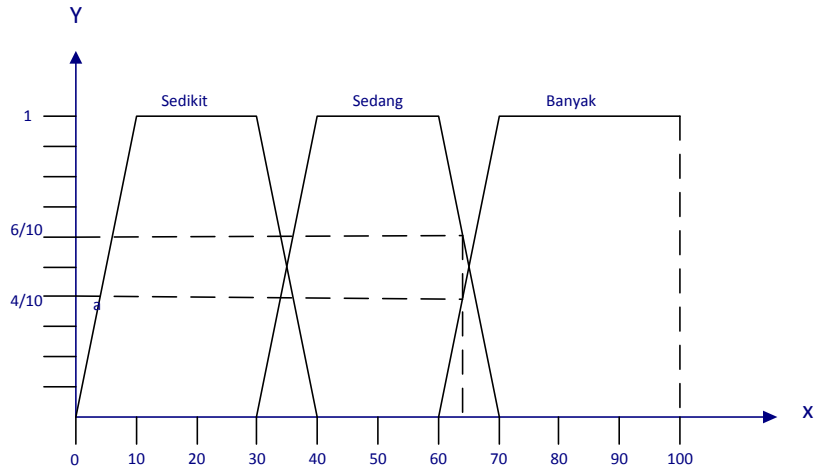
   x = 64

   a = 60

   b = 70

   μbanyak(x) $= \frac{x-a}{b-a}$

   μbanyak(64) $= \frac{64-60}{\mathbf{70-60}} = \frac{4}{\mathbf{10}}$



Gambar 3.17 *Fuzzy* input darah bos

Jadi proses *fuzzificationnya* menghasilkan 4 *fuzzy* input:

1. Serangan *player* terhadap bos = lemah ($\frac{\mathbf{2}}{\mathbf{10}}$) dan sedang ($\frac{\mathbf{8}}{\mathbf{10}}$)
2. Darah bos = sedang ($\frac{\mathbf{6}}{\mathbf{10}}$) dan banyak ($\frac{\mathbf{4}}{\mathbf{10}}$).

b. Proses *Inference*

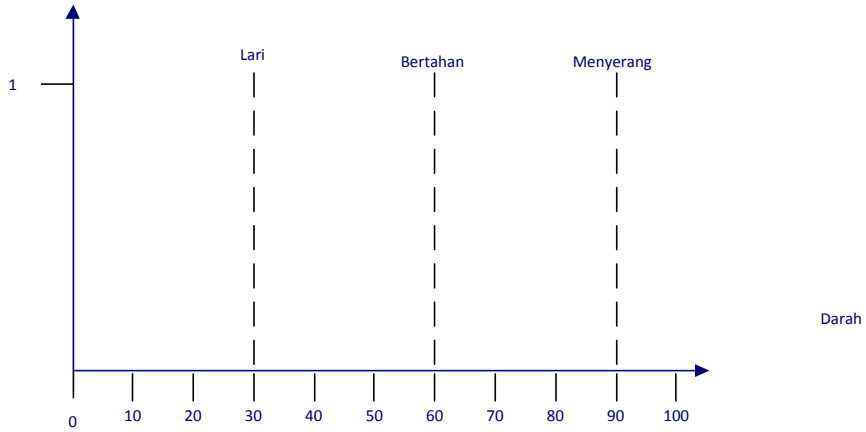
Proses *Inference* model Sugeno menggunakan fungsi keanggotaan s*ingleton*. Fungsi keanggotaan *singleton* adalah fungsi keanggotaan yang memiliki derajat keanggotaan 1 pada suatu nilai *crisp* tunggal dan 0 pada semua nilai *crisp* yang lain.

Misal aturan *fuzzy* serangan *player* terhadap bos dengan darah bos sebagai berikut:

Tabel 3.6 Kondisi Bos

		Serangan *player* terhadap bos		
		Lemah	Sedang	Besar
Darah	Sedikit	Lari	Lari	Lari
	Sedang	Bertahan	Bertahan	Lari
	Banyak	Menyerang	Menyerang	Bertahan

Dari aturan *fuzzy* keadaan serangan *player* terhadap bos dan darah misalkan didefiniskan fungsi *singleton* untuk kondisi bos yaitu:



Gambar 3.18 *singleton* kondisi bos

Dari empat *fuzzy input* serangan *player* terhadap bos = lemah ($\frac{\mathbf{2}}{\mathbf{10}}$), serangan *player* terhadap bos = sedang ($\frac{\mathbf{8}}{\mathbf{10}}$), darah bos = sedang ($\frac{6}{\mathbf{10}}$), dan darah bos = banyak ($\frac{4}{\mathbf{10}}$), maka didapatkan 4 aturan *fuzzy* yang kemudian diaplikasikan menggunakan aturan *conjunction* untuk memilih derajat keanggotaan minimum:

a. Jika serangan *player* terhadap bos = lemah ($\frac{\mathbf{2}}{\mathbf{10}}$) AND darah bos = sedang ($\frac{6}{\mathbf{10}}$)

   THEN kondisi bos = menyerang ($\frac{\mathbf{2}}{\mathbf{10}}$).

b. Jika serangan *player* terhadap bos = lemah ($\frac{\mathbf{2}}{\mathbf{10}}$) AND darah bos = banyak ($\frac{4}{\mathbf{10}}$)

   THEN kondisi bos = menyerang ($\frac{\mathbf{2}}{\mathbf{10}}$).

c. Jika serangan *player* terhadap bos = sedang ($\frac{\mathbf{8}}{\mathbf{10}}$) AND darah bos = sedang ($\frac{6}{\mathbf{10}}$)

   THEN kondisi bos = bertahan ($\frac{\mathbf{6}}{\mathbf{10}}$).

d. Jika serangan *player* terhadap bos = sedang ($\frac{\mathbf{8}}{\mathbf{10}}$) AND darah bos = sedang ($\frac{4}{\mathbf{10}}$)

   THEN kondisi bos = menyerang ($\frac{\mathbf{4}}{\mathbf{10}}$).

Menggunakan aturan *disjunction* untuk memilih derajat keanggotaan maksimum, maka hasilnya:

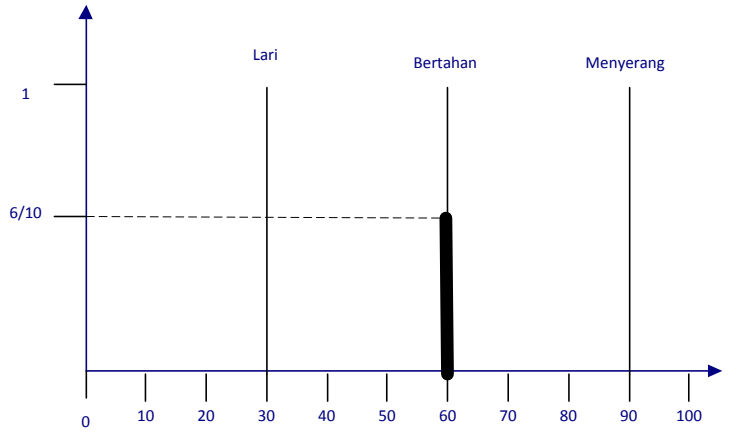
a. Jika serangan *player* terhadap bos = sedang ($\frac{\mathbf{8}}{\mathbf{10}}$) AND darah bos = sedang ($\frac{6}{\mathbf{10}}$)

   THEN kondisi bos = bertahan ($\frac{\mathbf{6}}{\mathbf{10}}$).

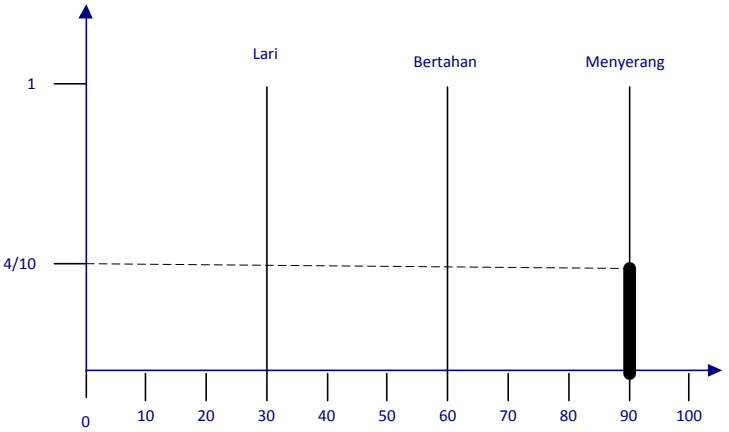
b. Jika serangan *player* terhadap bos = sedang ($\frac{\mathbf{8}}{\mathbf{10}}$) AND darah bos = sedang ($\frac{4}{\mathbf{10}}$)

THEN kondisi bos = menyerang ($\frac{4}{10}$).

Proses *inferensi* menggunakan model sugeno menghasilkan dua derajat keanggotaan sebagai berikut:



Gambar 3.19 kondisi bos bertahan



Gambar 3.20 kondisi bos menyerang

c. Proses *Defuzzification*

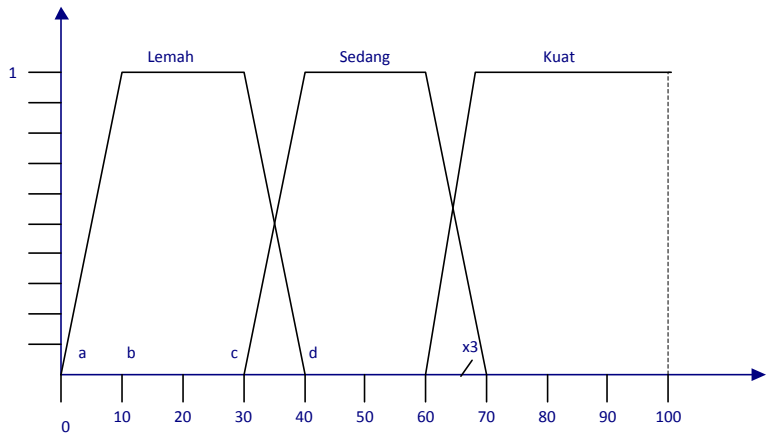
Proses *defuzzification* menggunakan proses *height method* maka dari dua *fuzzy* set bertahan ($\frac{6}{10}$), dan menyerang ($\frac{4}{10}$), maka dipilih nilai maksimumnya yaitu bertahan ($\frac{6}{10}$) karena nilai *crisp* sebesar 60 maka persentase darahnya adalah 60 dan disebut bertahan.

d. Kesimpulan

Jadi misalkan nilai *crisp* yang diterima oleh serangan *player* terhadap bos adalah 38, nilai *crisp* yang diterima oleh darah adalah 60, kondisi yang didapatkan adalah bertahan yaitu persentase darahnya ($\frac{6}{10}$), nilai ($\frac{6}{10}$) ini digunakan memberikan indikator tentang kondisi bos.

a. Proses *Fuzzification* x3

Misalkan untuk serangan *player* terhadap bos menggunakan fungsi keanggotaan trapesium dengan tiga variabel linguistik yaitu: *sedikit(0-40)*, *sedang(30-70)*, dan *banyak(≥60)*.



Gambar 3.21 Fungsi keanggotaan trapesium serangan *player* terhadap bos

1. Serangan *player* terhadap bos(x3) = 67 berada pada nilai linguistik sedikit.
2. Keterangan:

   x = nilai *crisp*.

   a = titik awal dari fungsi keanggotaan trapesium

   b = titik ke dua yang searah dengan sumbu y dan berpotongan pada sumbu x jika ditarik garis secara vertikal.

   c = titik ke tiga yang searah dengan sumbu y dan berpotongan pada sumbu x jika ditarik garis secara vertikal.

   d = titik ke empat dari fungsi keanggotaan trapesium.

3. Derajat keanggotaan untuk sedang dihitung menggunakan rumus μA(x) = $\frac{-(x-d)}{d-c}$, $c < x \leq d$.

   Diketahui:

   x = 67

   c = 60

   d = 70

   μsedang(x) = $\frac{-(x-d)}{d-c}$

   μsedang(67) = $-\frac{\mathbf{67-70}}{\mathbf{70-60}} = \frac{\mathbf{3}}{\mathbf{10}}$

4. Derajat keanggotaan untuk kuat dihitung menggunakan rumus μA(x) = $\frac{x-a}{b-a}$, $a < x \leq b$.

   Diketahui:

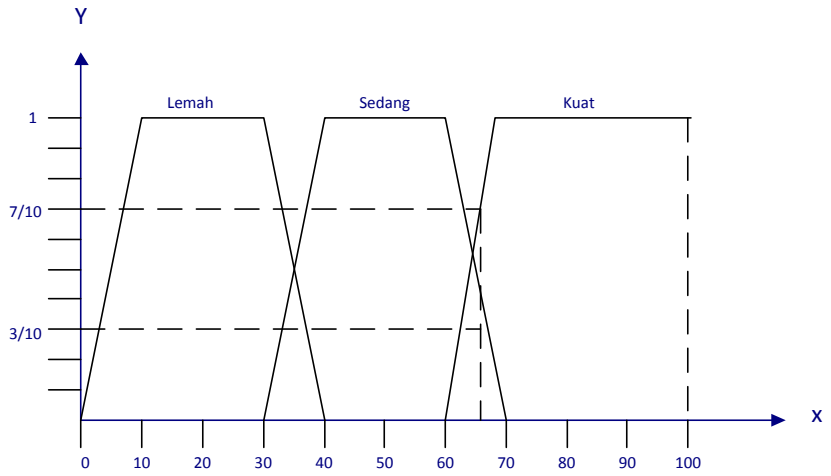
   x = 67

   a = 60

   b = 70

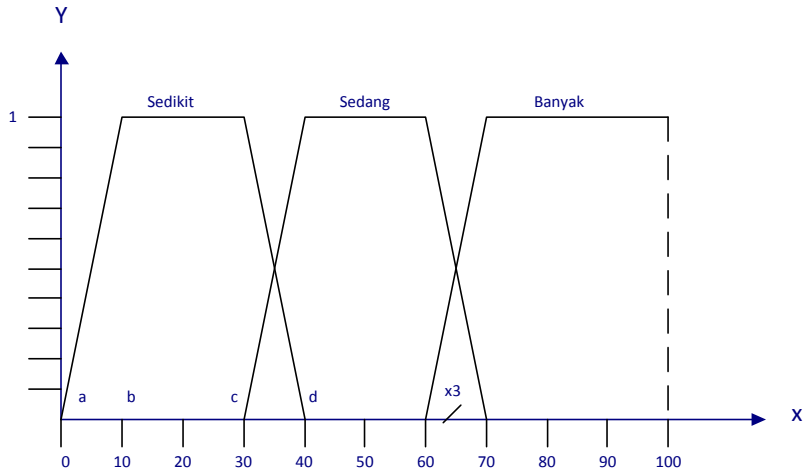
   μkuat (x) = $\frac{x-a}{b-a}$

   μkuat (67) = $\frac{\mathbf{67-60}}{\mathbf{70-60}} = \frac{7}{\mathbf{10}}$



Gambar 3.22 *Fuzzy* input serangan *player* terhadap bos

Misalkan untuk darah bos menggunakan fungsi keanggotaan trapesium dengan tiga variabel linguistik yaitu: *sedikit(0-40)*, *sedang(30-70)*, dan *banyak(60-100)*.



Gambar 3.23 Fungsi keanggotaan trapesium darah bos

1. Darah yang hilang dari, bos yaitu 61, berada pada nilai linguistik sedikit dan sedang.
2. Keterangan:

   x = nilai *crisp*.

   a = titik awal dari fungsi keanggotaan trapesium

   b = titik ke dua yang searah dengan sumbu y dan berpotongan pada sumbu x jika ditarik garis secara vertikal.

   c = titik ke tiga yang searah dengan sumbu y dan berpotongan pada sumbu x jika ditarik garis secara vertikal.

   d = titik ke empat dari fungsi keanggotaan trapesium.
3. Derajat keanggotaan untuk sedang dihitung menggunakan rumus µA(x) = $\frac{-(x-d)}{d-c}$, $c < x \leq d$.

   Diketahui:

   x = 61

   c = 60

   d = 70

   µsedang(x) = $\frac{-(x-d)}{d-c}$

μsedang(61) = $-\frac{\mathbf{61-70}}{\mathbf{70-60}} = \frac{\mathbf{9}}{\mathbf{10}}$

4. Derajat keanggotaan untuk banyak dihitung menggunakan rumus μA(x) = $\frac{x-a}{b-a}$, $a < x \leq b$.

   Diketahui:

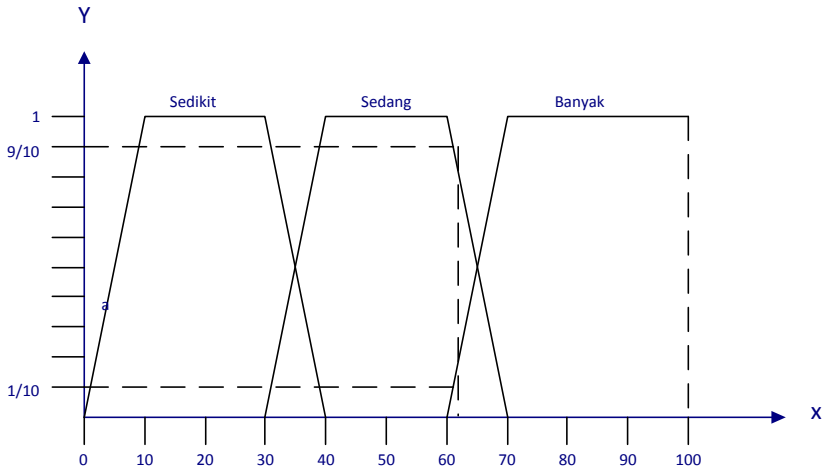
   x = 61

   a = 60

   b = 70

   μbanyak (x) = $\frac{x-a}{b-a}$

   μbanyak(61) = $\frac{61-60}{\mathbf{70-60}} = \frac{\mathbf{1}}{\mathbf{10}}$



Gambar 3.24 *Fuzzy* input darah bos

Jadi proses *fuzzificationnya* menghasilkan 4 *fuzzy* input:

1. Serangan *player* terhadap bos = sedang ($\frac{\mathbf{3}}{\mathbf{10}}$) dan kuat ($\frac{\mathbf{7}}{\mathbf{10}}$).
2. Darah bos = sedang ($\frac{\mathbf{9}}{\mathbf{10}}$) dan banyak ($\frac{\mathbf{1}}{\mathbf{10}}$).

b. Proses Inference

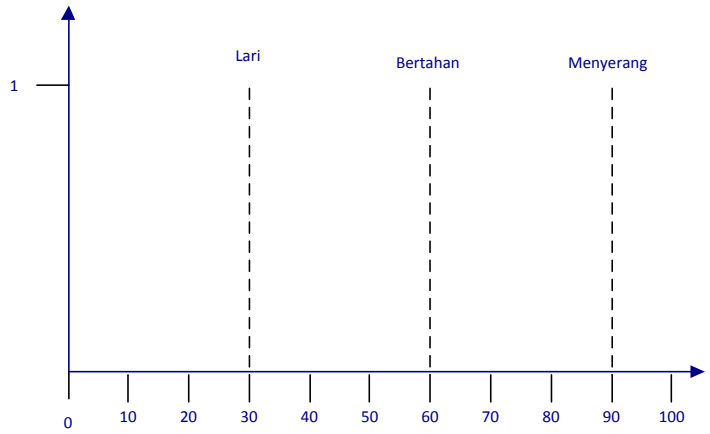
Proses *Inference* model Sugeno menggunakan fungsi keanggotaan s*ingleton*. Fungsi keanggotaan *singleton* adalah fungsi keanggotaan yang memiliki derajat keanggotaan 1 pada suatu nilai *crisp* tunggal dan 0 pada semua nilai *crisp* yang lain.

Misal aturan *fuzzy* serangan *player* terhadap bos dengan darah bos sebagai berikut:

Tabel 3.7 Kondisi Bos

		Serangan *player* terhadap bos		
		Lemah	Sedang	Besar
Darah	Sedikit	Lari	Lari	Lari
	Sedang	Bertahan	Bertahan	Lari
	Banyak	Menyerang	Menyerang	Bertahan

Dari aturan *fuzzy* keadaan serangan *player* terhadap bos dan darah misalkan didefiniskan fungsi *singleton* untuk kondisi bos yaitu:



Gambar 3.25 *singleton* kondisi bos

Dari empat *fuzzy input* serangan *player* terhadap bos = sedang ($\frac{3}{10}$), serangan *player* terhadap bos = kuat ($\frac{7}{10}$), darah bos = sedang ($\frac{9}{10}$), dan darah bos = banyak ($\frac{1}{10}$), maka didapatkan 4 aturan *fuzzy* yang diaplikasikan menggunakan aturan *conjunction* ($\cap$) untuk memilih derajat keanggotaan minimum:

a. Jika serangan *player* terhadap bos = sedang ($\frac{3}{10}$) AND darah bos = sedang ($\frac{9}{10}$)

   THEN kondisi bos = bertahan ($\frac{3}{10}$).

b. Jika serangan *player* terhadap bos = sedang ($\frac{3}{10}$) AND darah bos = banyak ($\frac{1}{10}$)

   THEN kondisi bos = menyerang ($\frac{1}{10}$).

c. Jika serangan *player* terhadap bos = kuat ($\frac{7}{10}$) AND darah bos = sedang ($\frac{9}{10}$)

   THEN kondisi bos = lari ($\frac{7}{10}$).

d. Jika serangan *player* terhadap bos = kuat ($\frac{7}{10}$) AND darah bos = banyak ($\frac{1}{10}$)

   THEN kondisi bos = menyerang ($\frac{1}{10}$).

Menggunakan aturan *disjunction* untuk memilih derajat keanggotaan maksimum, maka hasilnya:

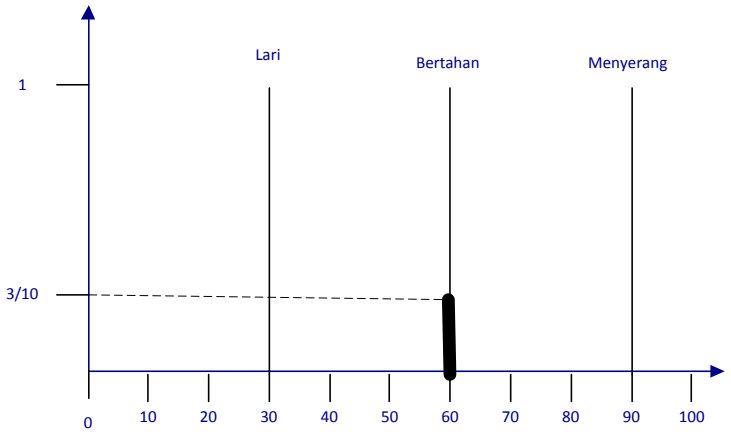
a. Jika serangan *player* terhadap bos = sedang ($\frac{3}{10}$) AND darah bos = sedang ($\frac{9}{10}$)

   THEN kondisi bos = bertahan ($\frac{3}{10}$).

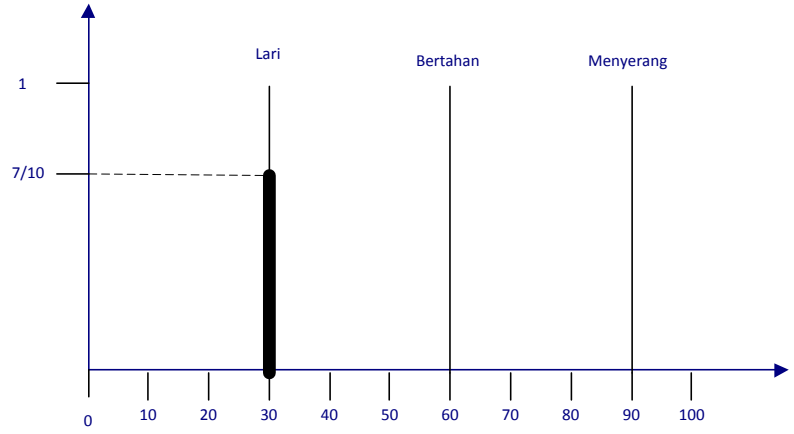
b. Jika serangan *player* terhadap bos = kuat ($\frac{7}{10}$) AND darah bos = banyak ($\frac{1}{10}$)

   THEN kondisi bos = menyerang ($\frac{1}{10}$).

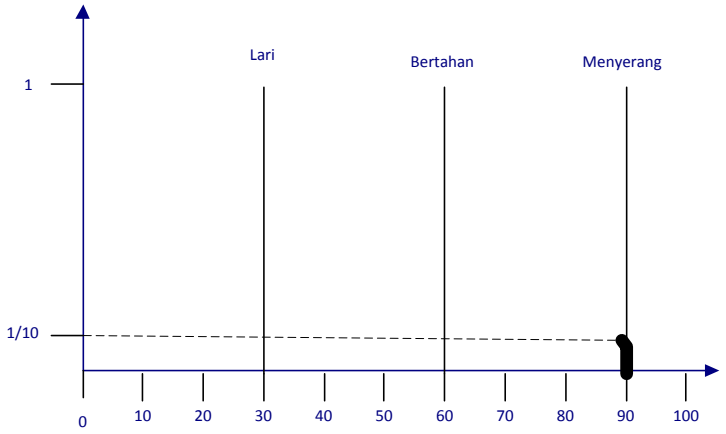
Proses *inferensi* menggunakan model sugeno menghasilkan tiga derajat keanggotaan sebagai berikut:



Gambar 3.26 kondisi bos bertahan



Gambar 3.27 kondisi bos lari



Gambar 3.28 kondisi bos menyerang

c. *Defuzzification*

Proses *defuzzification* menggunakan proses *height method* maka dari tiga *fuzzy* set bertahan ($\frac{3}{10}$), lari ($\frac{7}{10}$), dan menyerang ($\frac{1}{10}$), maka dipilih nilai maksimumnya yaitu lari ($\frac{7}{10}$) karena nilai *crisp* sebesar 30 maka persentase darahnya adalah 30 dan disebut lari.

d. Kesimpulan

Jadi misalkan nilai *crisp* yang diterima oleh serangan *player* terhadap bos adalah 67, nilai *crisp* yang diterima oleh darah adalah 61, kondisi yang didapatkan adalah lari yaitu persentase darahnya ($\frac{7}{10}$), nilai ($\frac{7}{10}$) ini digunakan memberikan indikator tentang kondisi bos.

3.1.5 Analisis Kebutuhan Non-Fungsional

Menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan analisis dan kebutuhan perangkat lunak, analisis dan kebutuhan perangkat keras, serta analisis kebutuhan user.

3.1.6 Analisis dan Kebutuhan Perangkat Lunak

Perangkat lunak digunakan dalam sebuah sistem merupakan perintah-perintah yang diberikan kepada perangkat keras agar dapat berinteraksi diantara keduanya. Adapun perangkat lunak yang dibutuhkan dalam membangun aplikasi game ini adalah:

1. Sistem Operasi Windows Xp.
2. Adobe Flash.

3.1.7 Analisis dan Kebutuhan Perangkat keras

Komputer terdiri dari bagian-bagian perangkat keras dan perangkat lunak yang saling berinteraksi diantara keduanya. Perangkat lunak memberikan perintah-perintah kepada perangkat keras untuk melakukan tugas tertentu, sehingga dapat menjalankan suatu sistem di dalamnya. Perangkat keras yang digunakan dapat dilihat pada tabel 3.5:

Tabel 3.8 Analisis dan Kebutuhan Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	*Hardisk*	40 Giga
2	*Keyboard*	Standar
3	*Mouse*	Standar
4	*Processor*	*Processor Pentium 4*
5	*RAM*	1 Giga
6	*Speaker*	Standar
7	VGA	VGA *Card 128*MB

3.1.8 Analisis Pengguna

Adapun spesifikasi pengguna untuk dapat memainkan aplikasi *game* ini, seperti pada tabel 3.6:

Tabel 3.9 Spesifikasi Pengguna

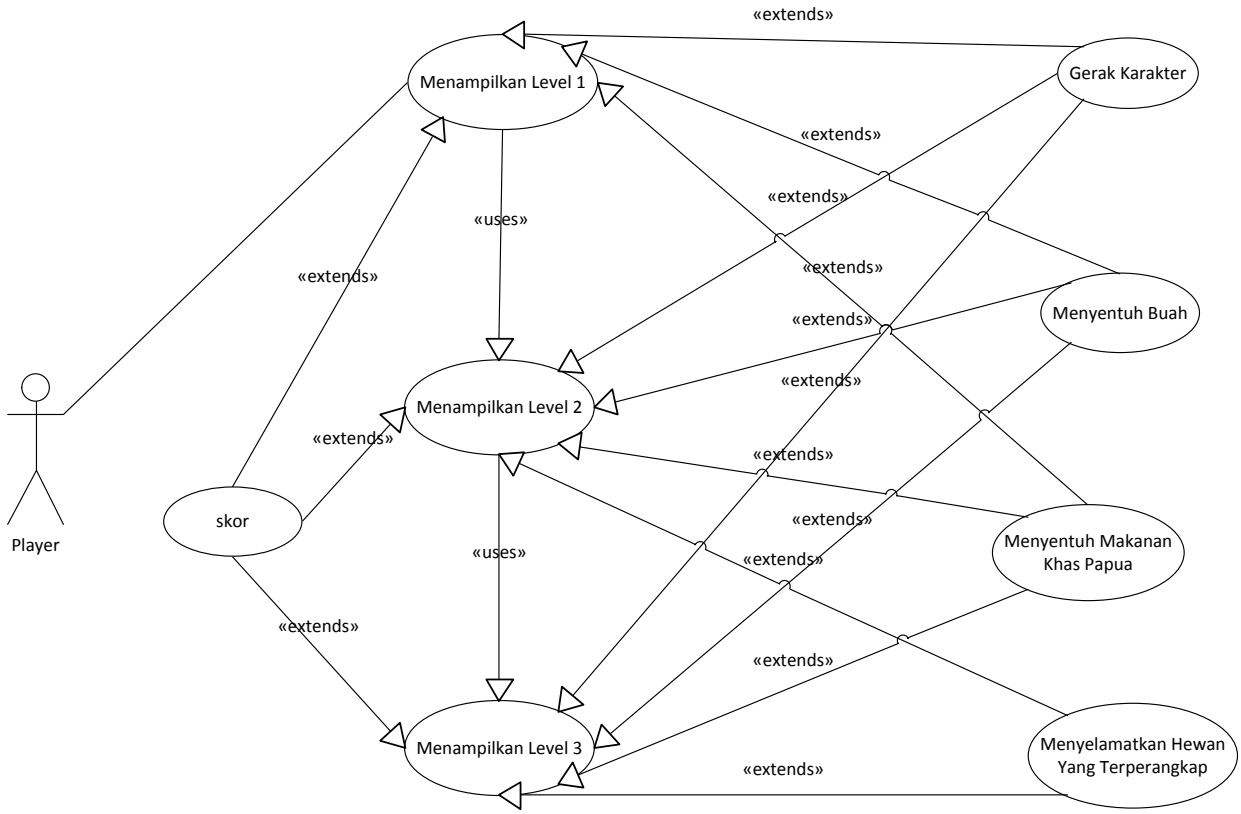
Peran	Pemain
Usia	10 tahun keatas
Tingkat Keterampilan	1. Mampu membaca 2. Mampu mengoperasikan komputer
Pengalaman	Pernah memainkan *game* seperti pokemon, dll.

3.1.9 Analisis Kebutuhan Fungsional

Menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik. Analisis yang dilakukan dimodelkan menggunakan UML (*Unified Modeling Language*). Tahap-tahap pemodelan dalam analisis tersebut antara lain use case diagram, skenario use case, activity diagram, sequence diagram, class diagram.

3.1.10 Use Case Diagram

Use case diagram merupakan konstruksi untuk mendeskripsikan hubungan-hubungan yang terjadi antar aktor dengan aktivitas yang terdapat pada sistem, yang digambarkan pada diagram use case adalah sebagai berikut:



Gambar 3.29 Diagram *Use Case* Sugie Man Penyelamat Sugar Glider

Use Case terdiri dari tiga bagian yaitu definisi aktor, definisi *use case*, dan *scenario use case*, berikut penjelasannya:

1. Definisi Aktor

Definisi Aktor yaitu seorang yang mengakses *use* case, dan hanya ada satu aktor yang berperan yaitu pemain yang memainkan *game* tersebut.

2. Definisi *Use Case*

Definisi *use case* merupakan fungsi untuk menjelaskan setiap proses yang terdapat pada setiap *use case diagram*. Definisinya dapat dilihat pada tabel di bawah ini:

Tabel 3.10 Definisi *Use Case*

No	*Use Case*	Deskripsi
1	Menampilkan Level 1	Proses tingkatan level 1
2	Menampilkan Level 2	Proses tingkatan level 2
3	Menampilkan Level 3	Proses tingkatan level 3
4	Menyentuh buah	Proses untuk menyelesaikan misi
5	Menyentuh makanan khas papua	Proses untuk menambah darah
6	Gerak karakter	Proses untuk menggerakan karakter
7	Menyelamatkan hewan yang terperangkap	Proses untuk menyelamatkan hewan yang terperangkap
8	Skor	Proses untuk menghitung *score*

3. Scenario *Use Case*

Scenario u*se case* adalah hasil instansiasi dari setiap *use case*. Berikut *scenario use case* pada *game* sugie man penyelamat sugar glider adalah sebagai berikut:

a. Nama *use case*: Menampilkan Level 1

Tabel 3.11 Skenario Menampilkan Level 1

Identifikasi	
No	1
Nama	Menampilkan Level 1
Tujuan	Menampilkan permainan level 1
Deskripsi	Proses untuk memulai permainan di level 1
Actor	Pemain
Scenario utama	
Kondisi awal	Menampilkan arena level 1
Aksi Aktor	**Reaksi Sistem**
Skenario normal	

1. Menggerakan sugie man	
2. Menyentuh kalajengking atau anjing	
3. Menyentuh makanan khas papua	
4. Menyentuh buah pear	
	5. *Player* dapat melanjutkan ke level 2
Skenario Alternatif	
1. Tidak menyentuh kalajengking atau anjing	
2. Tidak menyentuh buah pear	
	3. Melanjutkan permainan

b. Nama *use case*: Menampilkan Level 2

Tabel 3.12 Skenario Menampilkan Level 2

Identifikasi	
No	2
Nama	Menampilkan Level 2
Tujuan	Menampilkan permainan level 2
Deskripsi	Proses untuk memulai permainan di level 2
Actor	Pemain
Scenario utama	
Kondisi awal	Menampilkan arena level 2
Aksi Aktor	**Reaksi Sistem**
Skenario normal	
1. Menggerakan sugie man	
2. Menyentuh buaya atau ular	
3. Menyentuh makanan khas papua	
4. Menyentuh 2 kadal yang terperangkap	
5. Menyentuh buah melon	
	6. *Player* dapat melanjutkan ke level 3
Skenario Alternatif	
1. Tidak menyentuh buaya atau ular	
2. Tidak Menyentuh 2 kadal yang terperangkap	
3. Tidak menyentuh buah melon	
	4. Melanjutkan permainan

c. Nama *use case*: Menampilkan Level 3

Tabel 3.13 Skenario Menampilkan Level 3

Identifikasi	
No	3
Nama	Menampilkan Level 3
Tujuan	Menampilkan permainan level 3
Deskripsi	Proses untuk memulai permainan di level 3
Actor	Pemain
Scenario utama	
Kondisi awal	Menampilkan arena level 3
Aksi Aktor	**Reaksi Sistem**
Skenario normal	
1. Menggerakan sugie man	
2. Menyentuh anjing atau boss	
3. Menyentuh makanan khas papua	
4. Menyentuh 4 sugar glider yang terperangkap	
5. Menyentuh buah semangka	
	6. Permainan selesai
Skenario Alternatif	
1. Tidak menyentuh anjing atau boss	
2. Tidak Menyentuh 4 sugar glider yang terperangkap	
3. Tidak menyentuh buah semangka	
	4. Melanjutkan permainan

d. Nama *use case*: Menyentuh buah

Tabel 3.14 Skenario Menyentuh buah

Identifikasi	
No	4
Nama	Menyentuh buah
Tujuan	Menyelesaikan misi
Deskripsi	Proses untuk menyelesaikan misi
Actor	*Player*
Scenario utama	
Kondisi awal	Berada di level 1, level 2, atau level 3
Aksi Aktor	**Reaksi Sistem**
Skenario Normal	

1. Menyentuh buah	
	2. Cek buah
	3. Buah telah terpenuhi, permainan selesai
Skenario Alternatif	
1. Tidak menyentuh buah	
	2. Cek buah
	3. Buah belum terpenuhi, melanjutkan permainan

e. Nama *use case*: Menyentuh makanan khas papua

Tabel 3.15 Skenario Menyentuh makanan khas papua

Identifikasi	
No	5
Nama	Menyentuh makanan khas papua
Tujuan	Menambah darah sugie man
Deskripsi	Proses untuk menambah darah sugie man
Actor	*Player*
Scenario utama	
Kondisi awal	Berada di level 1, level 2, atau level 3
Aksi Aktor	**Reaksi Sistem**
Skenario Normal	
1. Menyentuh makanan khas papua	
	2. Darah sugie man bertambah 20
Skenario Alternatif	
1. Tidak menyentuh makanan khas papua	
	2. Darah sugie man tetap

f. Nama *use case*: Gerak Karakter

Tabel 3.16 Skenario Gerak Karakter

Identifikasi	
No	6
Nama	Gerak karakter
Tujuan	Menggerakan sugie man
Deskripsi	Proses untuk menggerakan sugie man
Actor	*Player*
Scenario utama	
Kondisi awal	Berada di level 1, level 2, atau level 3
Aksi Aktor	**Reaksi Sistem**
Skenario Normal	

1.	
2. Menggerakan karakter	
	3. Sugie man bergerak sesuai *input player*
Skenario Alternatif	
1. Tidak Menggerakan karakter	
	2. Sugie man tidak bergerak

g. Nama *use case*: Menyentuh hewan yang terperangkap

Tabel 3.17 Skenario Menyentuh hewan yang terperangkap

Identifikasi	
No	7
Nama	Menyentuh hewan yang terperangkap
Tujuan	Menyelesaikan *quest*
Deskripsi	Proses untuk menyelamatkan hewan yang terperangkap
Actor	*Player*
Scenario utama	
Kondisi awal	Berada di level 1, level 2, dan level 3
Aksi Aktor	**Reaksi Sistem**
Skenario Normal	
1. Menyentuh kadal atau sugar glider yang terperangkap	
	2. Cek kadal atau sugar glider
	3. Kadal atau sugar glider telah terpenuhi, permainan selesai
Skenario Alternatif	
1. Tidak Menyentuh kadal atau sugar glider yang terperangkap	
	2. Cek kadal atau sugar glider
	3. Kadal atau sugar glider belum terpenuhi, melanjutkan permainan

h. Nama *use case*: skor

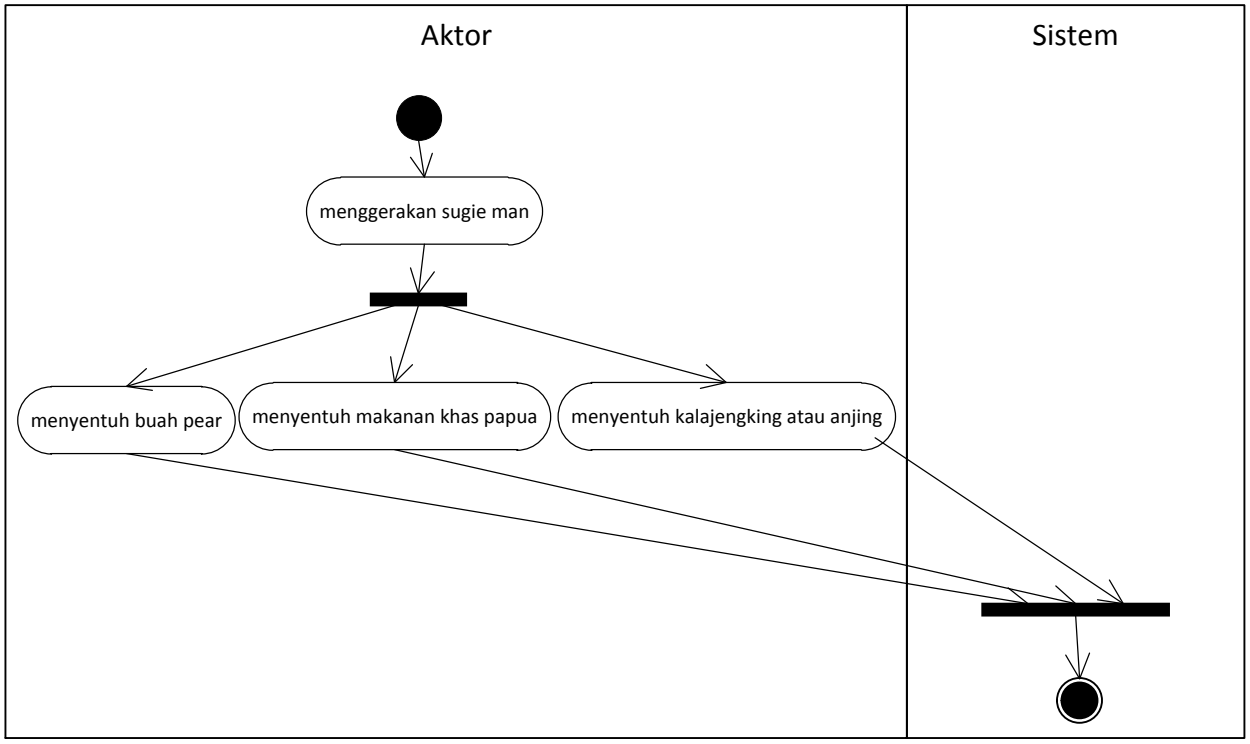
Tabel 3.18 Skenario skor

Identifikasi	
No	8
Nama	skor
Tujuan	Untuk menghitung nilai skor
Deskripsi	Proses menginformasikan nilai skor pemain
Actor	*Player*
Scenario utama	
Kondisi awal	Berada di level 1, level 2, dan level 3
Aksi Aktor	**Reaksi Sistem**
Skenario Normal	
1. Pemain menyelesaikan permainan dalam waktu maksimal 20 detik	
	2. skor pemain bertambah 50
Skenario Alternatif	
1. Pemain tidak menyelesaikan permainan dalam waktu 120 detik	
	2. Permainan berakhir

3.1.11 *Activity Diagram*

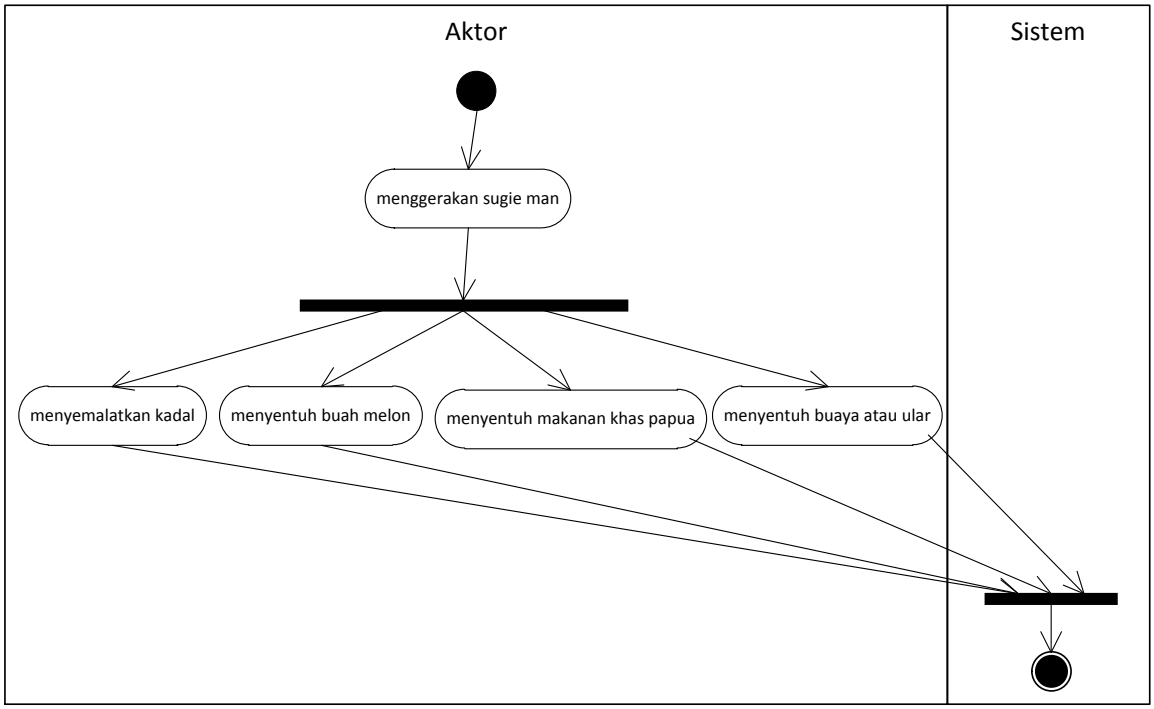
Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, desicion yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

a. *Activity Diagram* menampilkan level 1



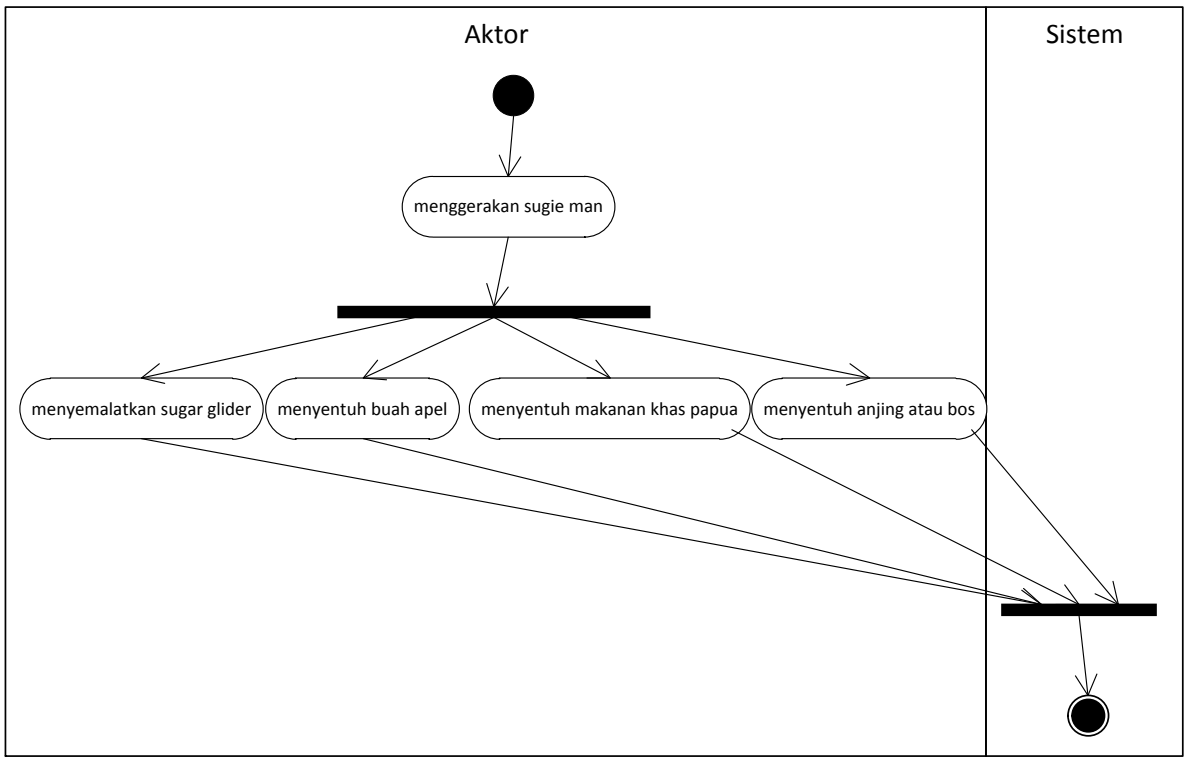
Gambar 3.30 *Activity Diagram* menampilkan level 1

b. *Activity Diagram* menampilkan level 2



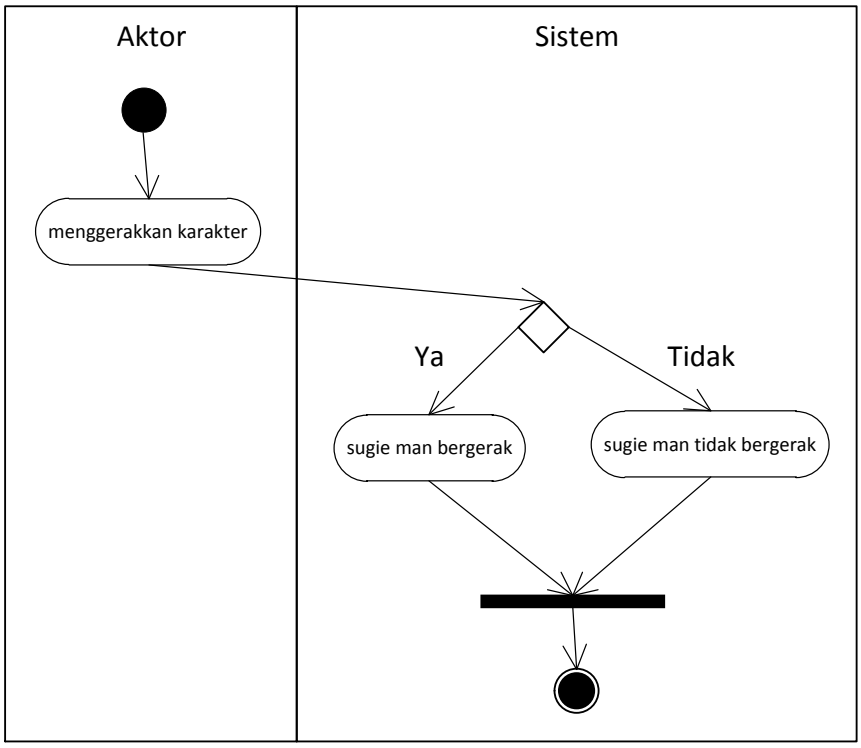
Gambar 3.31 *Activity Diagram* menampilkan level 2

c. *Activity Diagram* menampilkan level 3



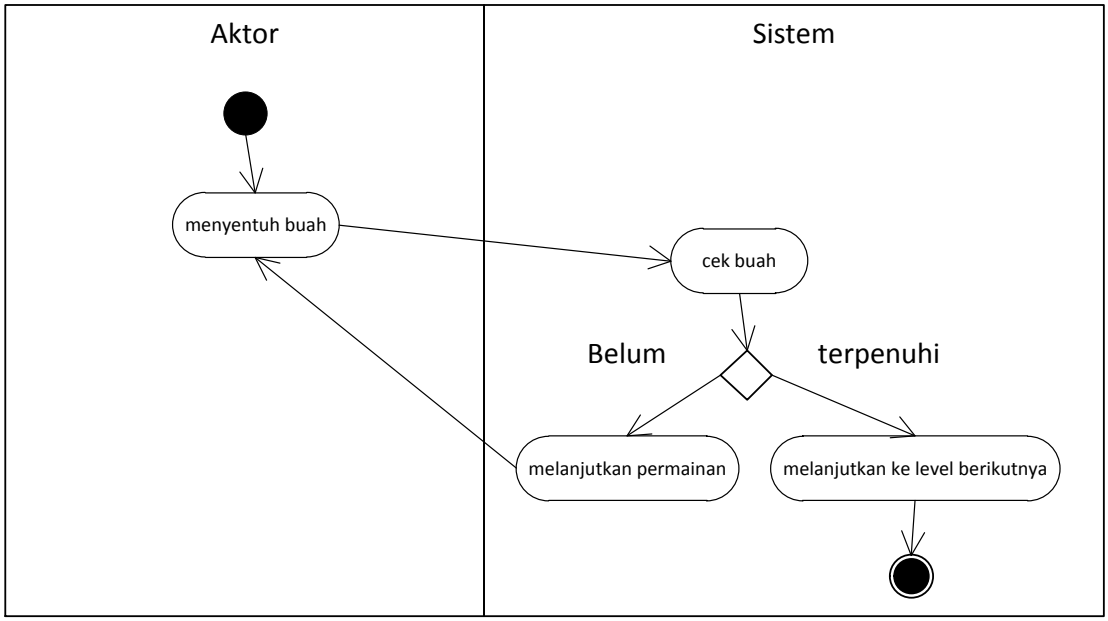
Gambar 3.32 *Activity Diagram* menampilkan level 3

d. *Activity Diagram* gerak karakter



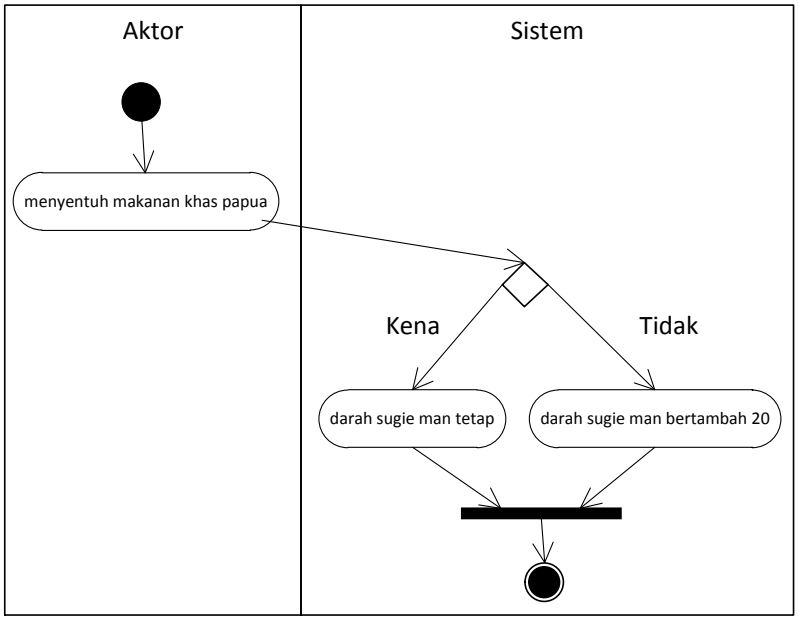
Gambar 3.33 *Activity Diagram* gerak karakter

e. *Activity Diagram* menyentuh buah



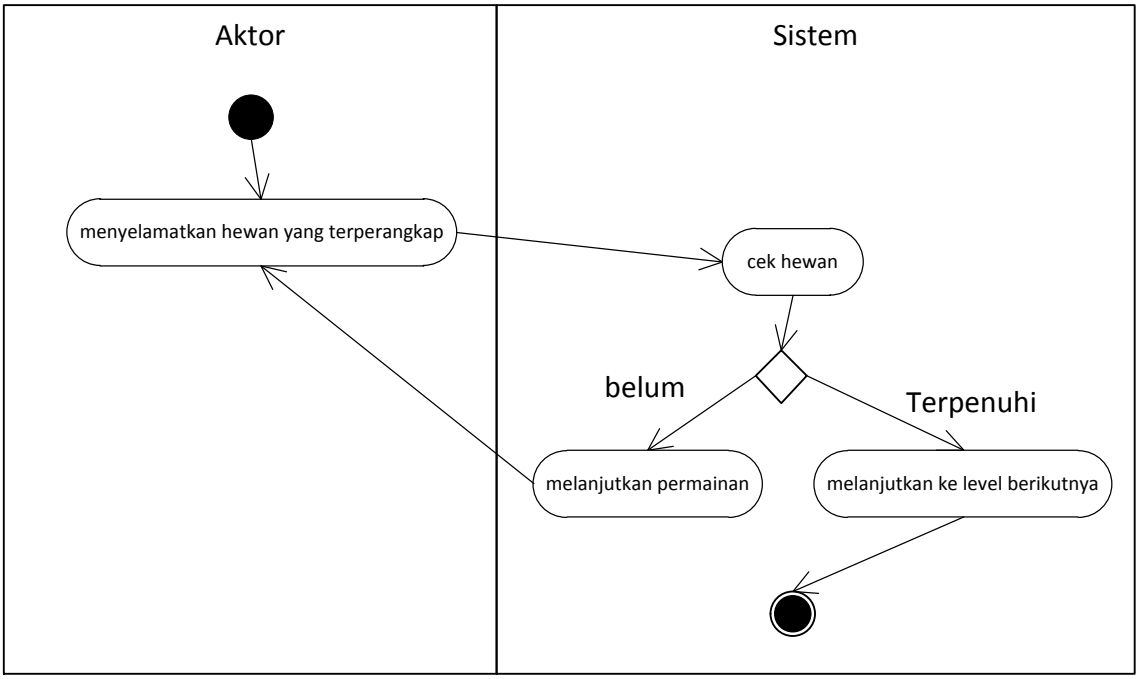
Gambar 3.34 *Activity Diagram* menyentuh buah

f. *Activity Diagram* menyentuh makanan khas papua



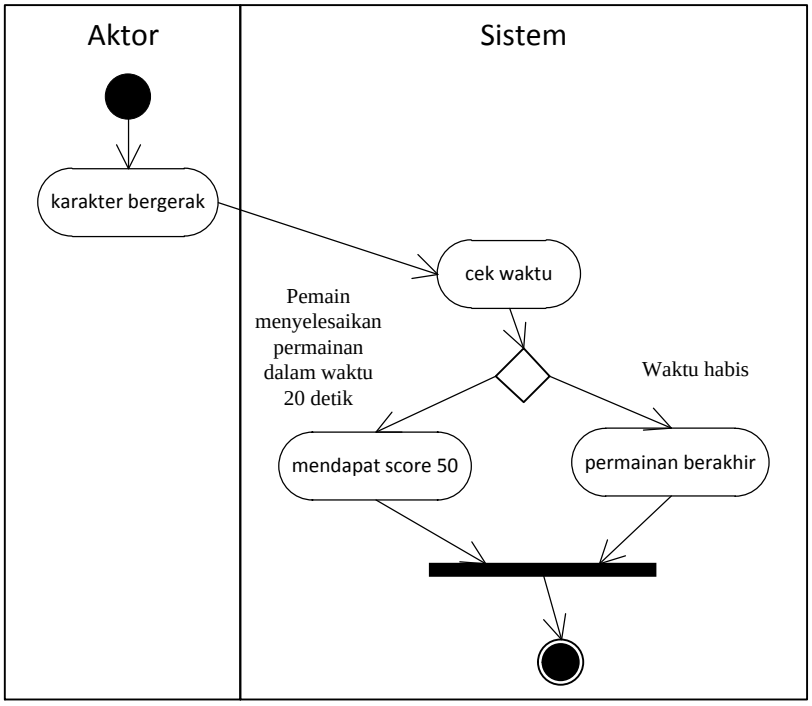
Gambar 3.35 *Activity Diagram* menyentuh makanan khas papua

g. *Activity Diagram* menyelamatkan hewan yang terperangkap



Gambar 3.36 *Activity Diagram* menyelamatkan hewan yang terperangkap

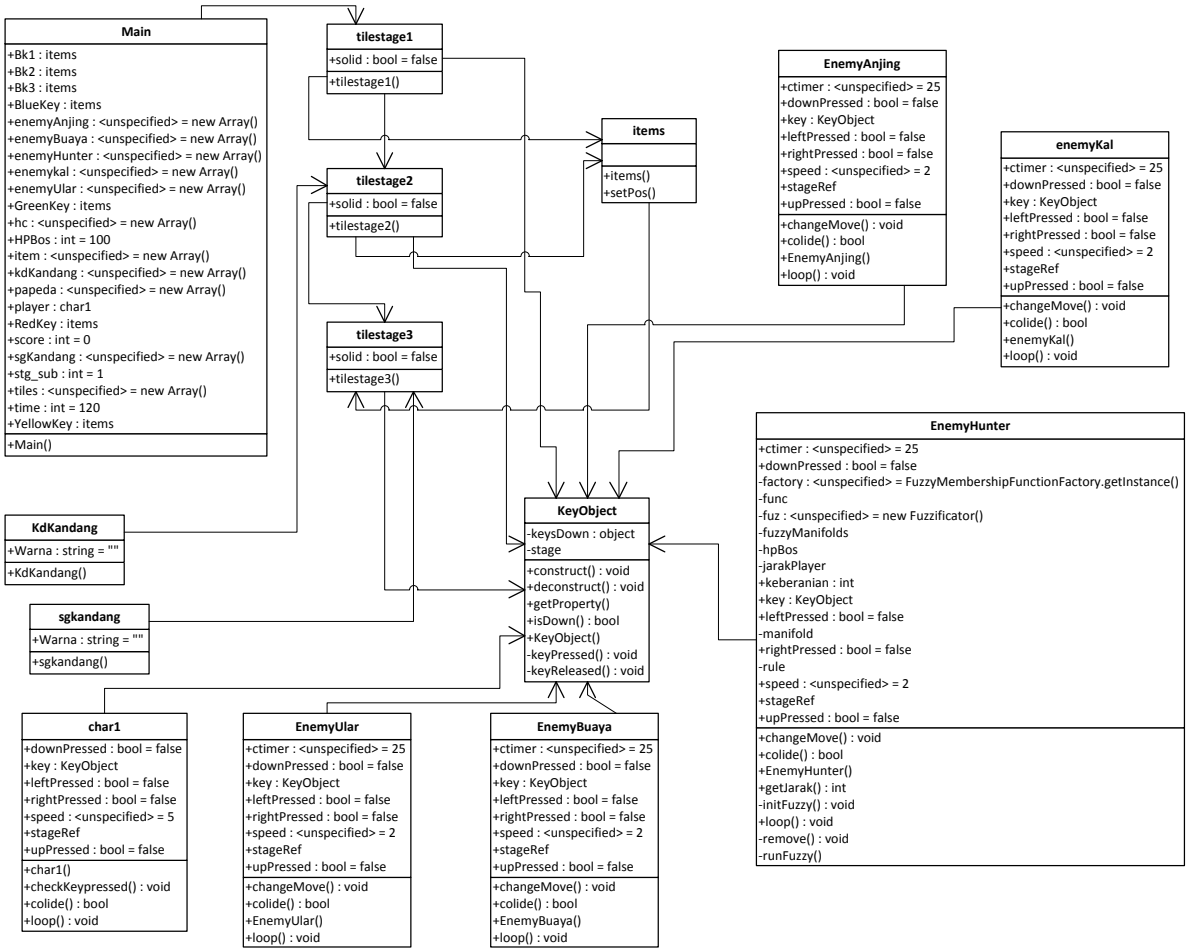
h. *Activity Diagram* skor



Gambar 3.37 *Activity Diagram* skor

3.1.12 *Class Diagram*

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

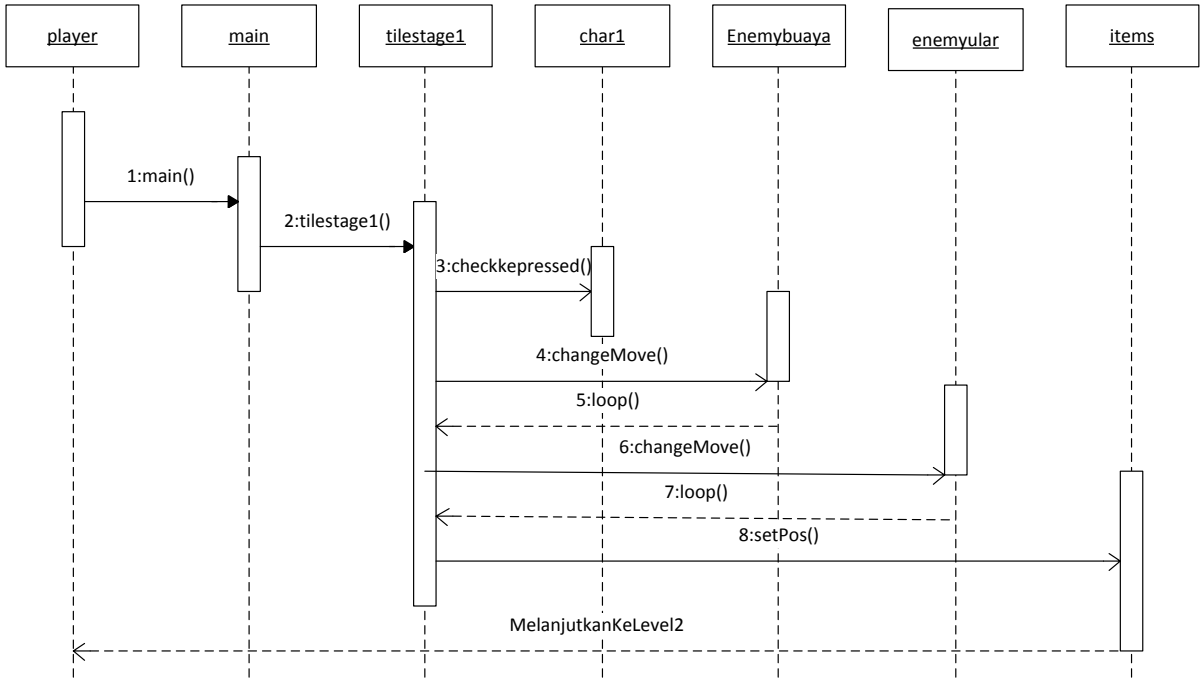


Gambar 3.38 *Class Diagram*

3.1.13 *Sequence Diagram*

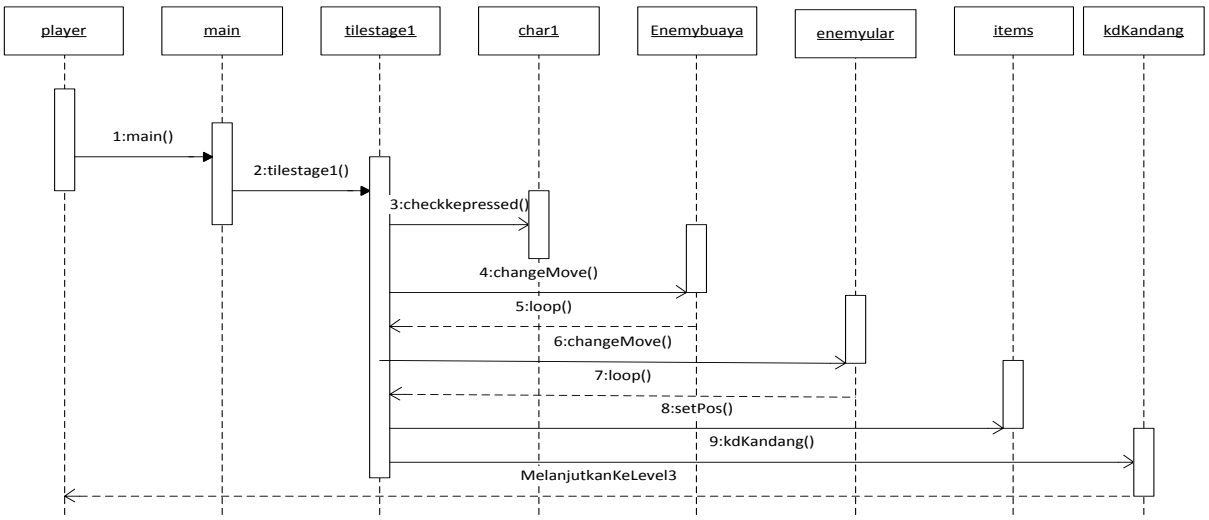
Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (pengguna, display, dan sebagainya) berupa pesan yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertical (waktu) dan horizontal (objek-objek yang terkair). Berikut *sequence diagram* pada *game* Sugie Man Penyelamat Sugar Glider:

1. *Sequence Diagram* level 1



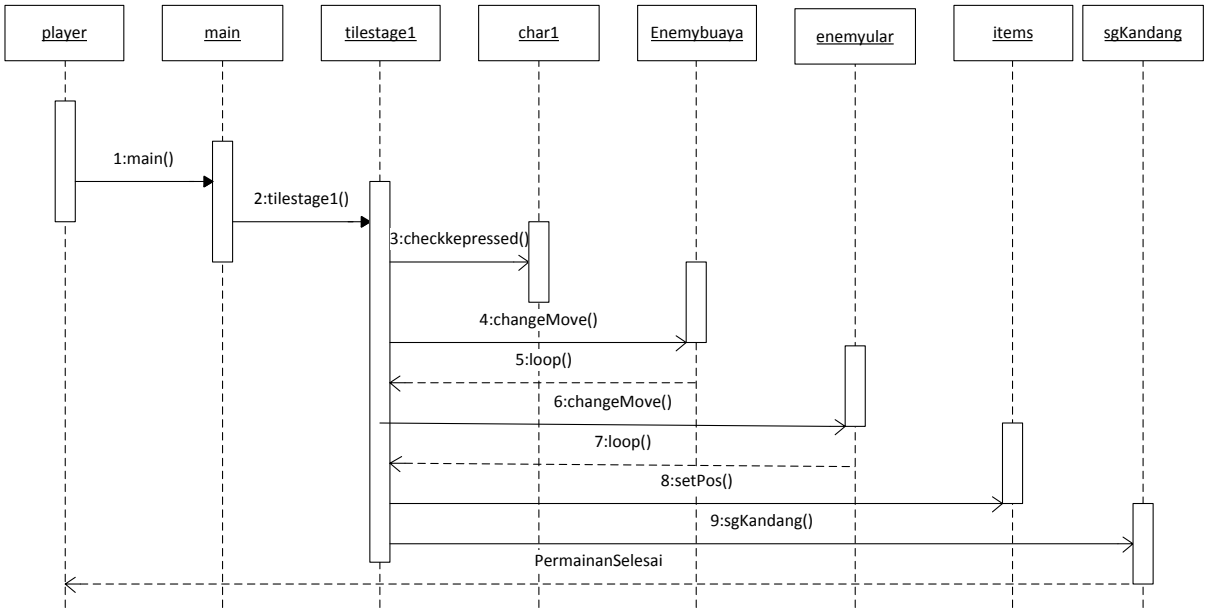
Gambar 3.39 *Sequence Diagram* level 1

2. *Sequence Diagram* level 2



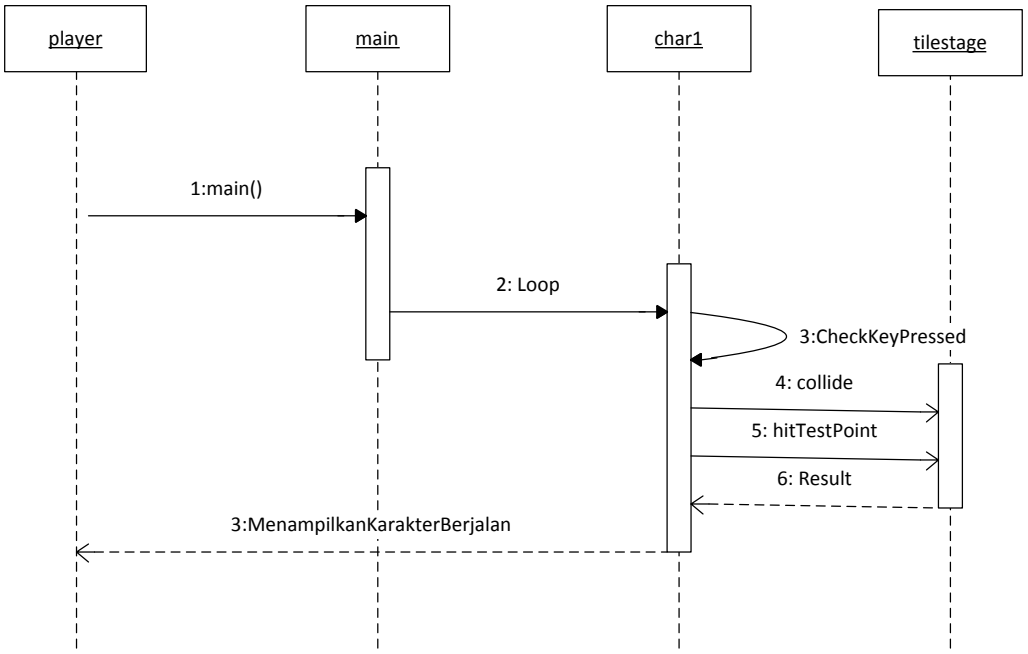
Gambar 3.40 *Sequence Diagram* level 2

3. *Sequence Diagram* level 3



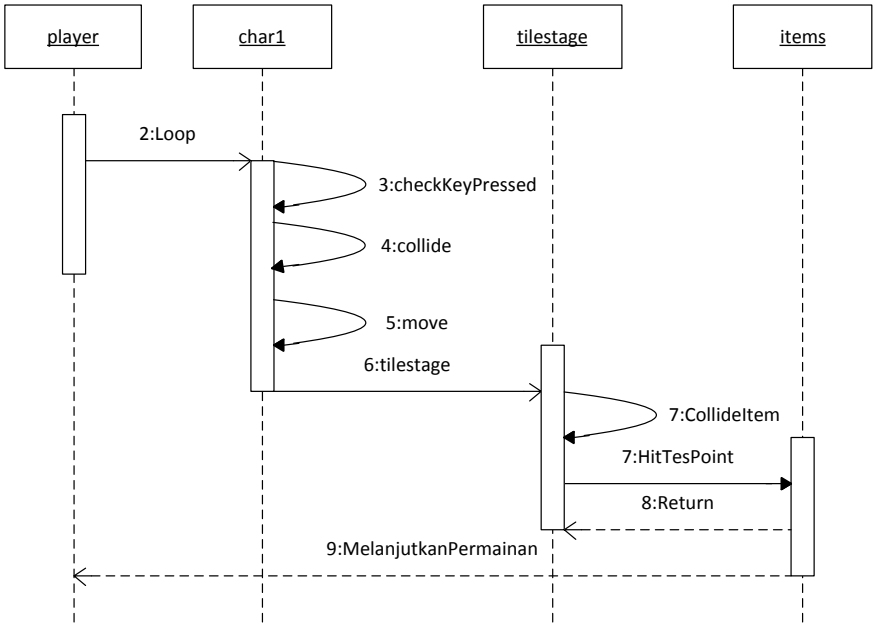
Gambar 3.41 *Sequence Diagram* level 3

4. *Sequence Diagram* gerak karakter



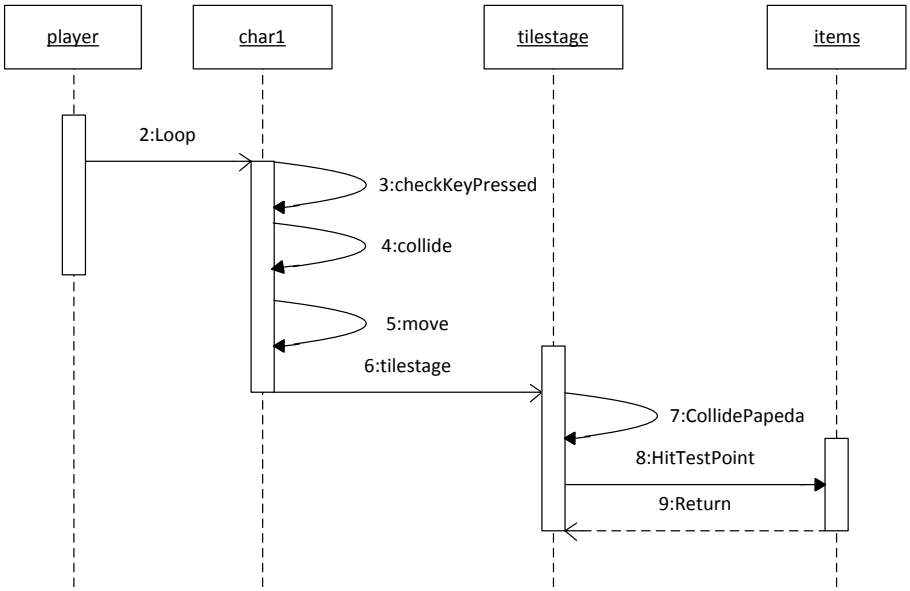
Gambar 3.42 *Sequence Diagram* gerak karakter

5. *Sequence Diagram* menyentuh buah



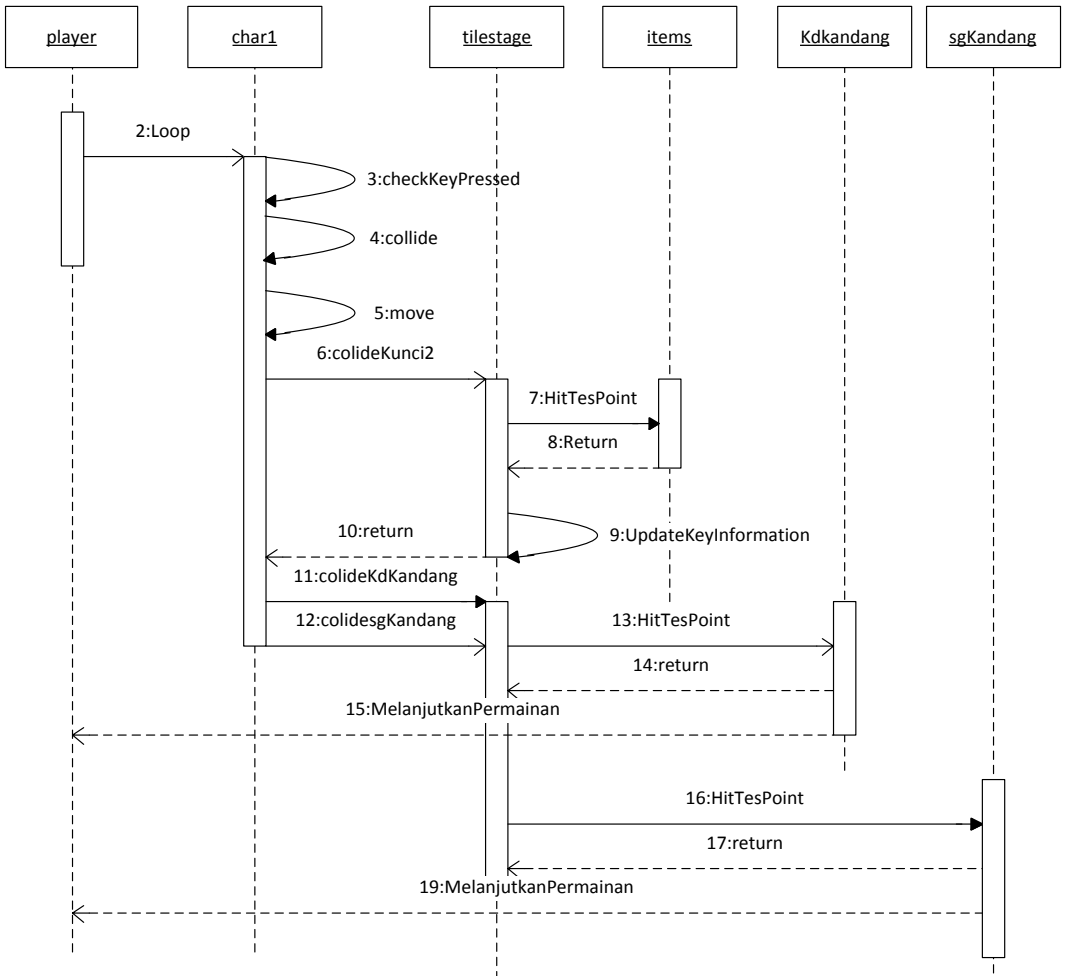
Gambar 3.43 *Sequence Diagram* menyentuh buah

6. *Sequence Diagram* menyentuh makanan khas papua



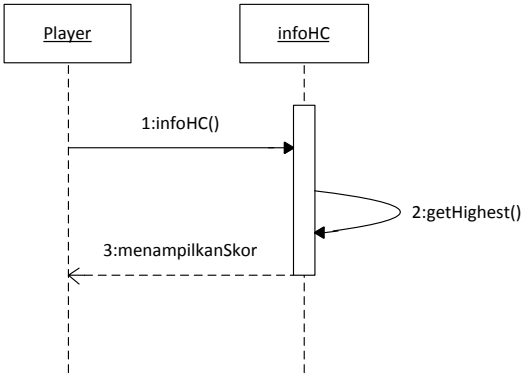
Gambar 3.44 *Sequence Diagram* menyentuh makanan khas papua

7. *Sequence Diagram* menyelamatkan hewan yang terperangkap



Gambar 3.45 *Sequence Diagram* menyelamatkan hewan yang terperangkap

8. *Sequence Diagram* skor



Gambar 3.46 *Sequence Diagram* skor

3.2 Perancangan sistem

Pada tahap ini akan menggambarkan rancangan sistem yang akan dibangun sebelum dilakukan pengkodean ke dalam suatu bahasa pemrograman.

3.2.1 Perancangan Komponen *Game*

Perancangan komponen permainan terdiri dari karakter dan *storyboard*. Perancangan komponen permainan menjelaskan tentang komponen-komponen yang terdapat pada aplikasi permainan *Sugie Man Penyelamat Sugar Glider*.

3.2.1.1 Pengenalan Karakter

Karakter – Karakter yang digunakan pada *game* sugie man penyelamat sugar glider yaitu bisa dilihat pada tabel 3.19:

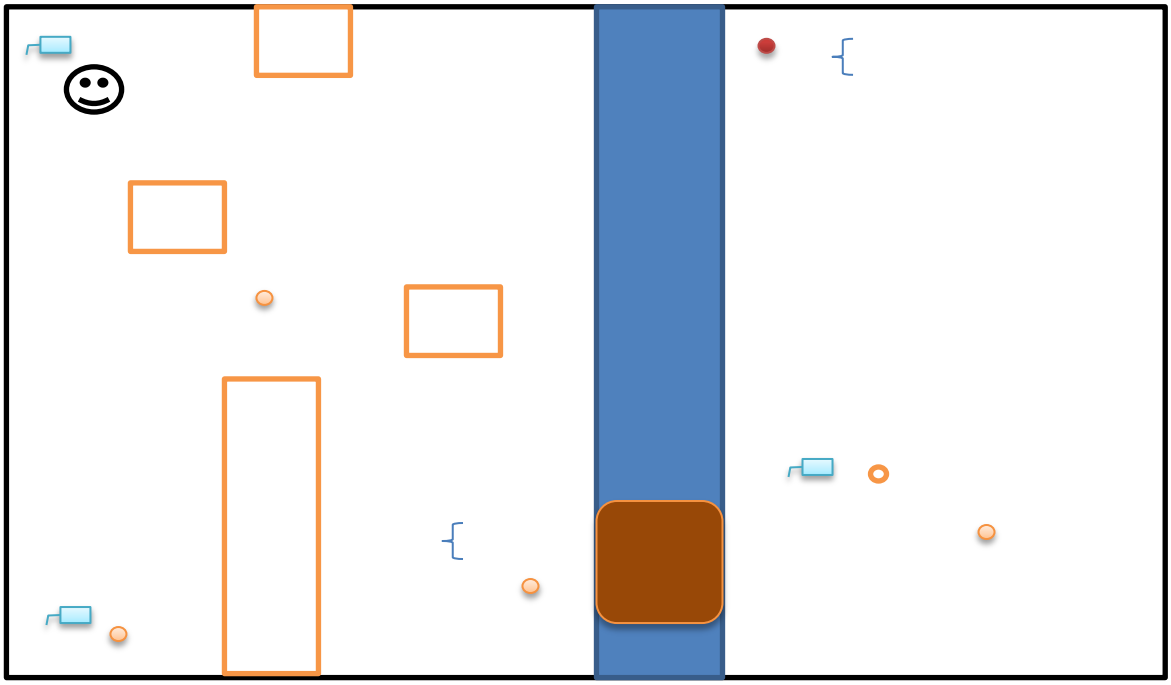
Tabel 3.19 Pengenalan Karakter

No	Karakter	Keterangan
1	Sugie Man	1. Merupakan karakter utama pada *game* ini 2. Sugie man dapat berjalan ke kanan, kiri, atas, dan bawah 3. Sugie man dapat memukul
2	Kalajengking	1. Musuh yang muncul di level 1 2. Dapat menyerang sugie man
3	Anjing	1. Musuh yang muncul di level 1 2. Dapat menyerang sugie man
4	Buaya	1. Musuh yang muncul di level 2 2. Dapat menyerang sugie man
5	Ular	1. Musuh yang muncul di level 2 2. Dapat menyerang sugie man
7	Anjing	1. Musuh yang muncul di level 3 2. Dapat menyerang sugie man
8	Bos Pemburu Liar	1. Musuh yang muncul di level 3 2. Dapat menyerang sugie man

3.2.1.2 *Storyboard*

Storyboard adalah sketsa gambar yang disusun berurutan sesuai dengan naskah, dengan *storyboard* kita dapat menyampaikan ide cerita kita kepada orang lain dengan cara lebih mudah.

Berikut adalah *storyboard game Sugie Man Penyelamat Sugar Glider*:



Gambar 3.47 *Storyboard* level 1

Keterangan:

= Sugie Man

= Anjing

= Kalajengking

= Buah Pear

= Papeda

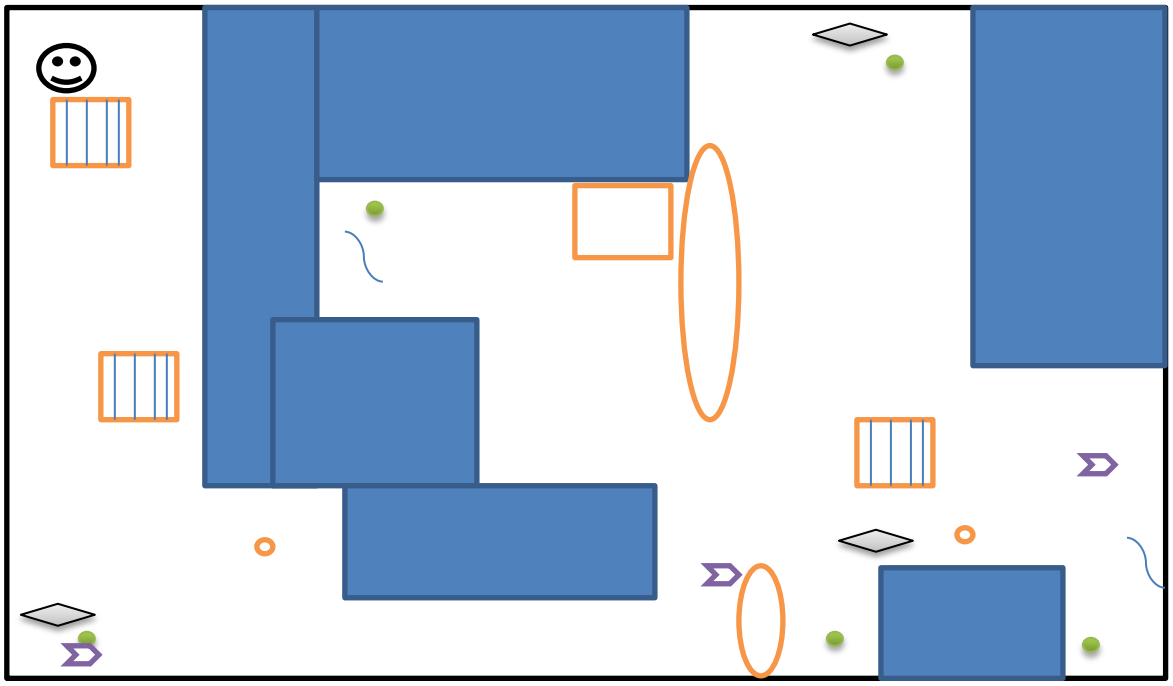
= Pohon

= Jembatan

= Sungai

1. Latar belakang berupa hutan.

2. Mengambil 5 buah pear untuk dapat melanjutkan ke level berikutnya dan akan mengeluarkan tulisan dan suara yes.
3. Terdapat 1 buah makanan khas papua yaitu papeda.
4. Mengambil makanan khas papua untuk menambah darah sugie man sebesar 20, akan mengeluarkan tulisan dan suara nyam.
5. Jika mengenai kalajengking, atau anjing darah pemain akan berkurang 20 dan akan mengeluarkan tulisan dan suara aduh.
6. Menghindari kalajengking, atau anjing pemburu liar untuk dapat melanjutkan perjalanan dan level berikutnya.
7. Jika *player* menyelesaikan permainan dalam waktu maksimal 20 detik, maka skor bertambah 50.
8. Jika waktu permainan habis, maka permainan berakhir.



Gambar 3.48 *Storyboard* level 2

Keterangan:

= Sugie Man

= Ular

= Buaya


= Melon


= Papeda


= Jembatan


= Sungai

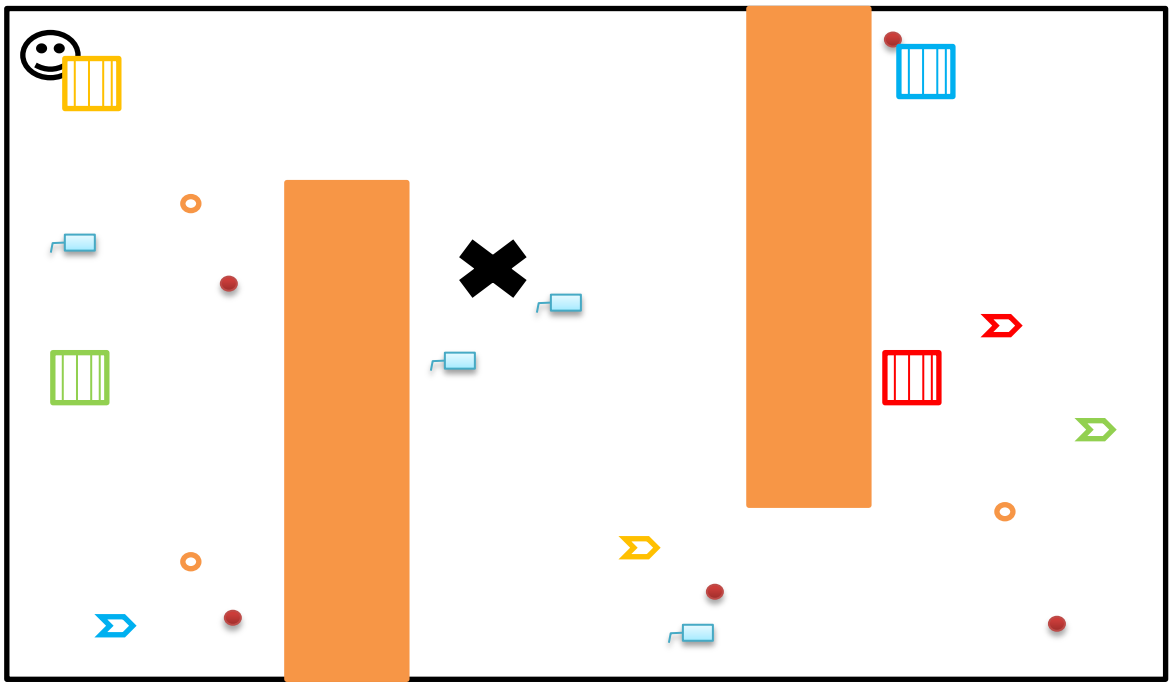

= Kandang Kadal


= Kunci Kandang


= Batu


= Pohon

1. Latar belakang berupa rawa.
2. Mengambil 5 buah melon untuk dapat melanjutkan ke level berikutnya dan mengeluarkan tulisan dan suara yes.
3. Menyelamatkan 2 kadal yang terperangkap dengan cara mengambil kunci terlebih dahulu, jika berhasil menyelamatkan kadal mengeluarkan tulisan dan suara yes.
4. Mengambil kunci untuk menyelamatkan kadal yang terperangkap dan mengeluarkan tulisan dan suara yes.
5. Terdapat 2 buah makanan khas papua yaitu papeda.
6. Mengambil makanan khas papua yang ada untuk menambah darah sugie man sebesar 20 dan mengeluarkan tulisan dan suara nyam.
7. Jika mengenai buaya, atau ular darah pemain akan berkurang 20 dan mengeluarkan tulisan dan suara aduh.
8. Menghindari buaya, atau ular untuk dapat melanjutkan perjalanan dan level berikutnya.
9. Jika *player* menyelesaikan permainan dalam waktu maksimal 20 detik, maka skor bertambah 50.
10. Jika waktu permainan habis, maka permainan berakhir.



Gambar 3.49 *Storyboard* level 3

Keterangan:

= Sugie Man

= Anjing

= Boss Pemburu Liar

= Kayu

= Buah Apel

= Papeda

= Kandang Sugar Glider Kuning

= Kunci Kandang Kuning

= Kandang Sugar Glider Biru

= Kunci Kandang Biru

= Kandang Sugar Glider Hijau

= Kunci Kandang Hijau

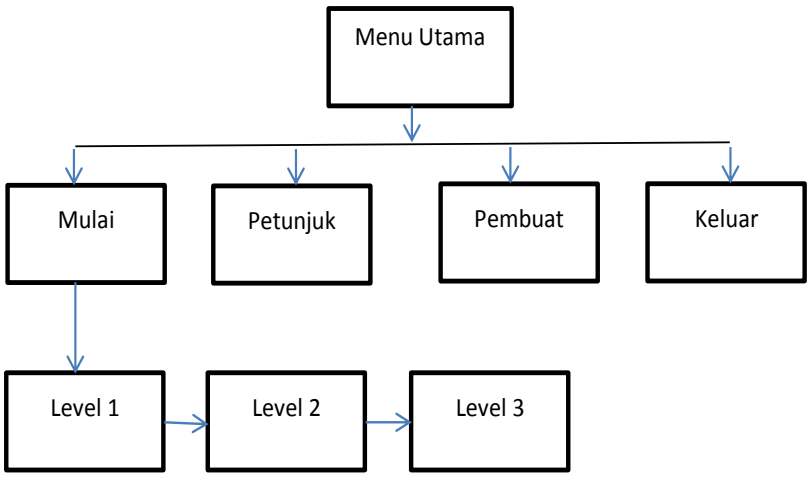
= Kandang Sugar Glider Merah

= Kunci Kandang Merah

1. Latar belakang berupa rumah pemburu liar.
2. Mengambil 5 buah apel untuk dapat menyelesaikan permainan dan mengeluarkan tulisan dan suara yes.
3. Menyelamatkan 4 sugar glider yang terperangkap dengan cara mengambil kunci yang sama warnanya dengan perangkap terlebih dahulu dan mengeluarkan tulisan dan suara yes.
4. Mengambil kunci yang sama warnanya dengan kandang untuk menyelamatkan sugar glider yang terperangkap dan mengeluarkan tulisan dan suara yes.
5. Setiap menyelamatkan sugar glider, darah bos pemburu liar akan berkurang 20, jika berhasil menyelamatkan sugar glider mengeluarkan tulisan dan suara yes.
6. Terdapat 3 buah makanan khas papua yaitu papeda.
7. Mengambil makanan khas papua yang ada untuk menambah darah sugie man sebesar 20 dan mengeluarkan tulisan dan suara nyam.
8. Menghindari anjing, atau bos pemburu liar untuk dapat menyelesaikan permainan.
9. Melepaskan sugar glider ke habibat aslinya.
10. Jika *player* menyelesaikan permainan dalam waktu maksimal 20 detik, maka skor bertambah 50.
11. Jika waktu permainan habis, maka permainan berakhir.

3.2.2 Perancangan Struktur Menu

Perancangan struktur menu pada *game* Sugie Man Penyelemat Sugar Glider adalah sebagai berikut:

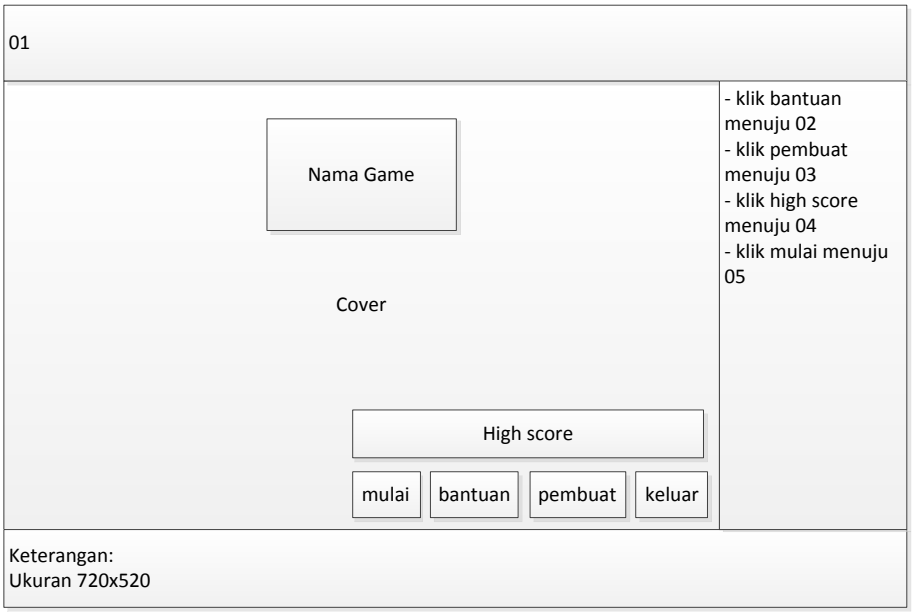


Gambar 3.50 Perancangan Struktur Menu Utama

3.2.3 Perancangan Antarmuka

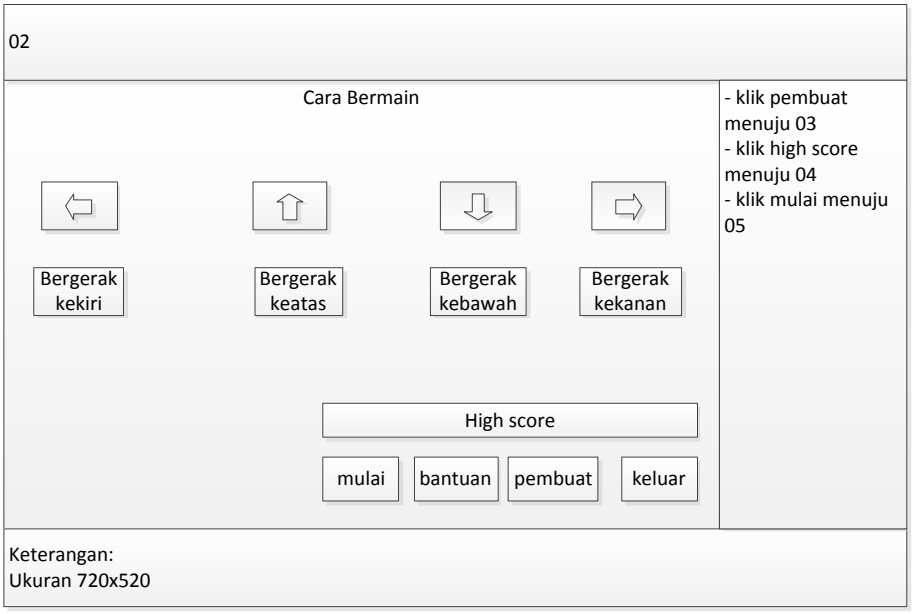
Perancangan antarmuka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun, sehingga akan mempermudah dalam mengimplementasikan dan pembuatan aplikasi.

1. Perancangan antarmuka Menu Utama



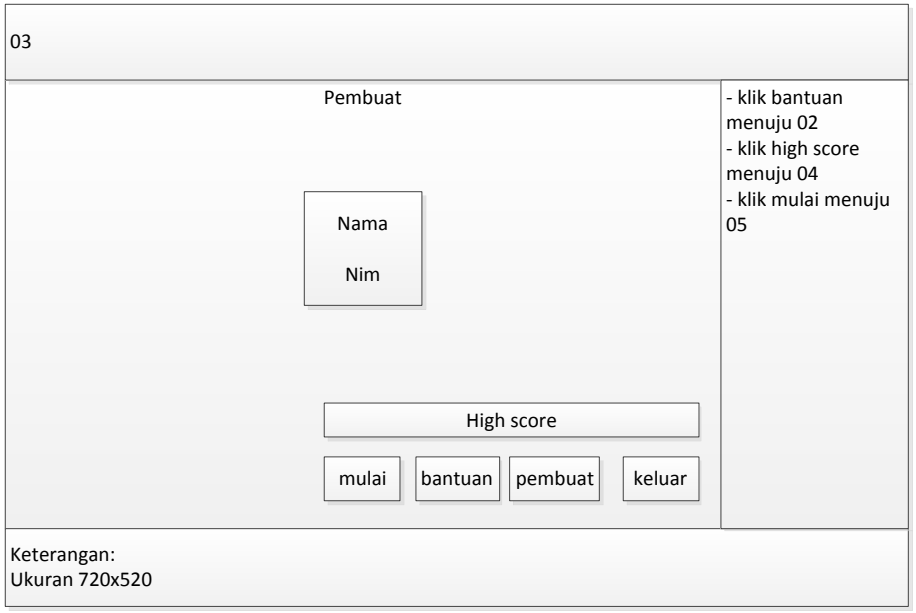
Gambar 3.51 Antarmuka Menu Utama

2. Perancangan antarmuka bantuan



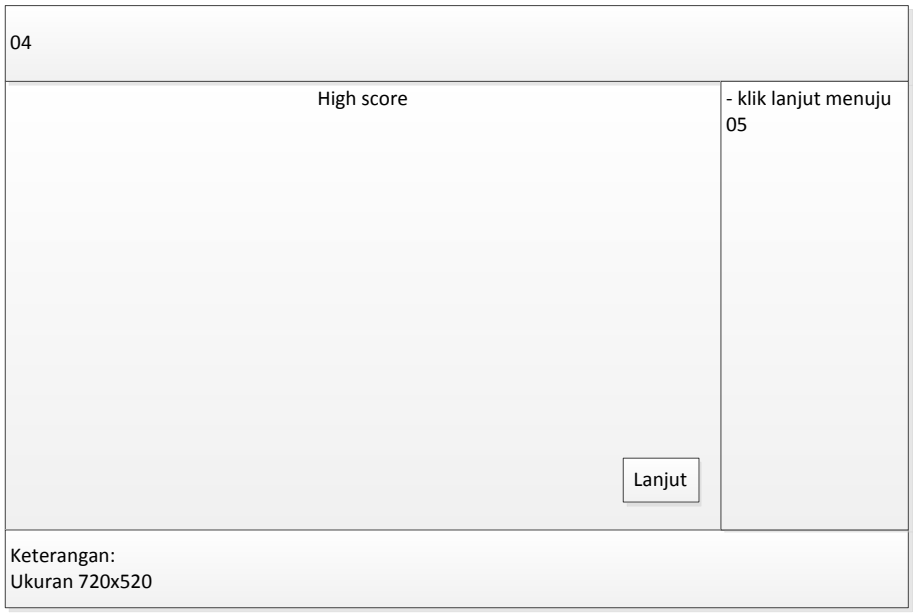
Gambar 3.52 Antarmuka bantuan

3. Perancangan antarmuka Pembuat



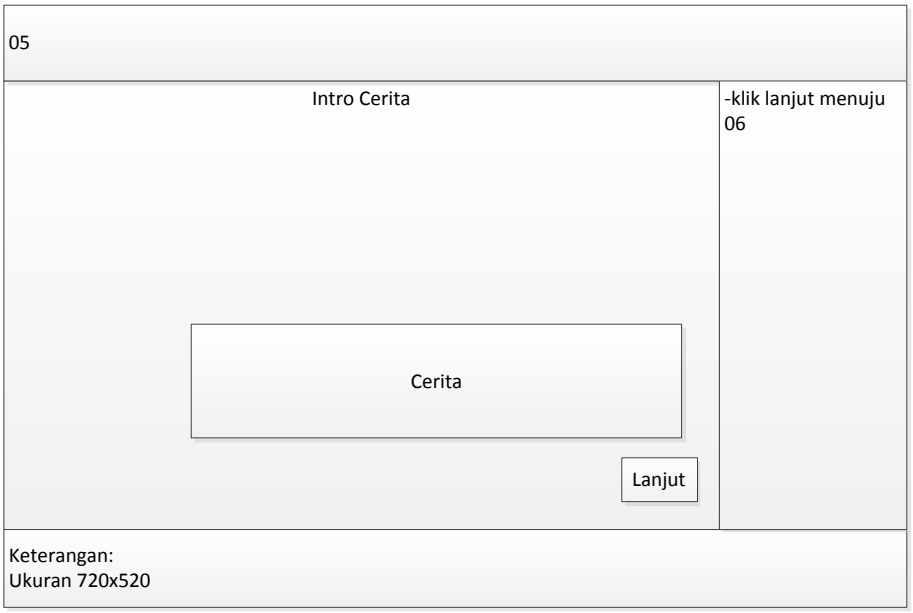
Gambar 3.53 Antarmuka Pembuat

4. Perancangan high score



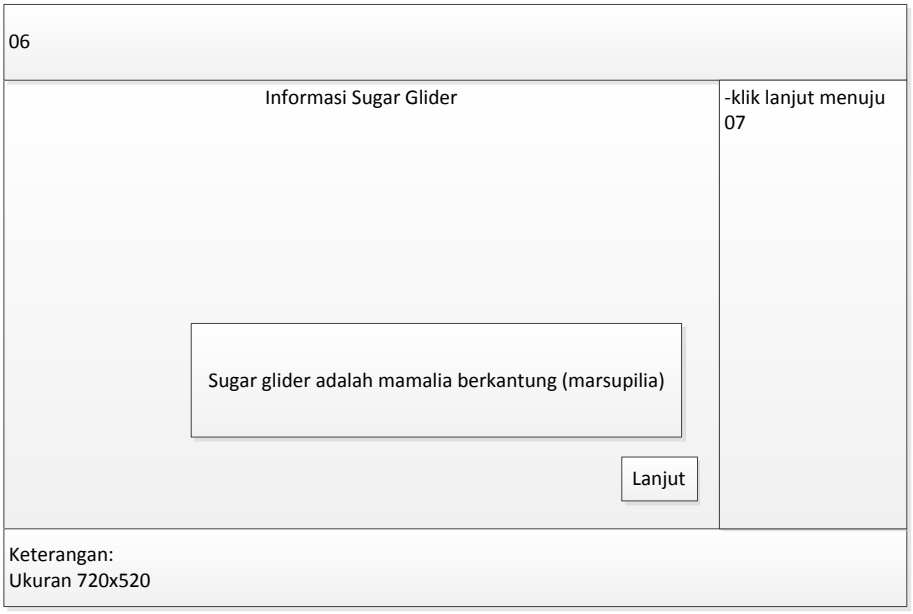
Gambar 3.54 Antarmuka high score

5. Perancangan antarmuka intro cerita



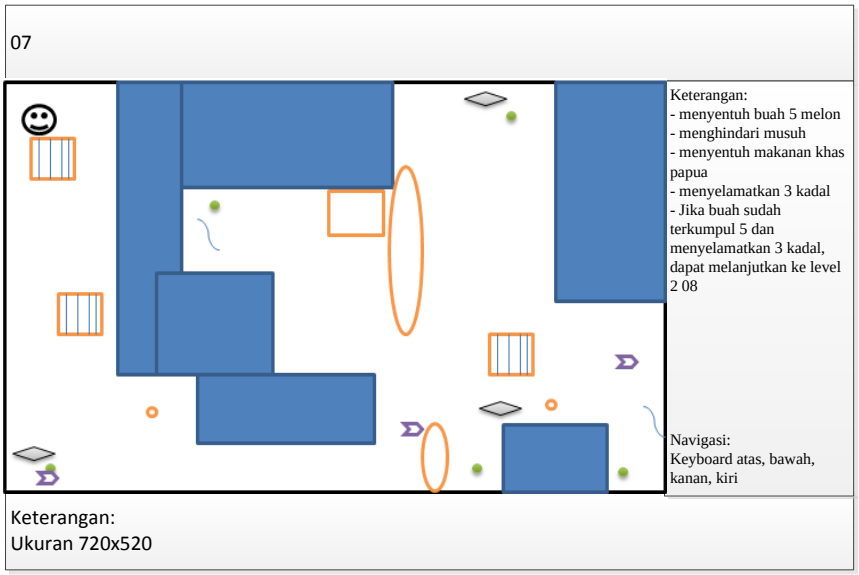
Gambar 3.55 Antarmuka intro cerita

6. Perancangan antarmuka informasi sugar glider



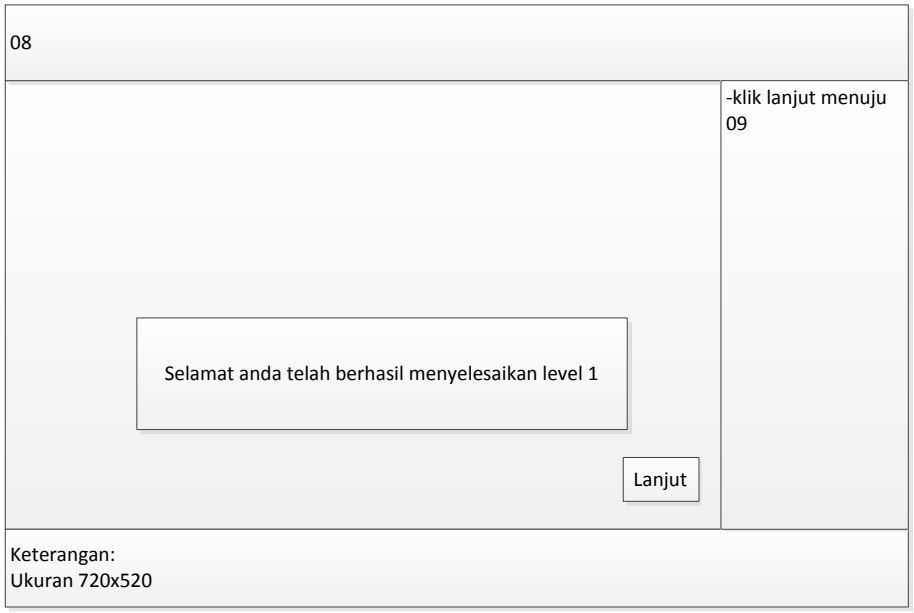
Gambar 3.56 Antarmuka informasi sugar glider

7. Perancangan antarmuka level 1



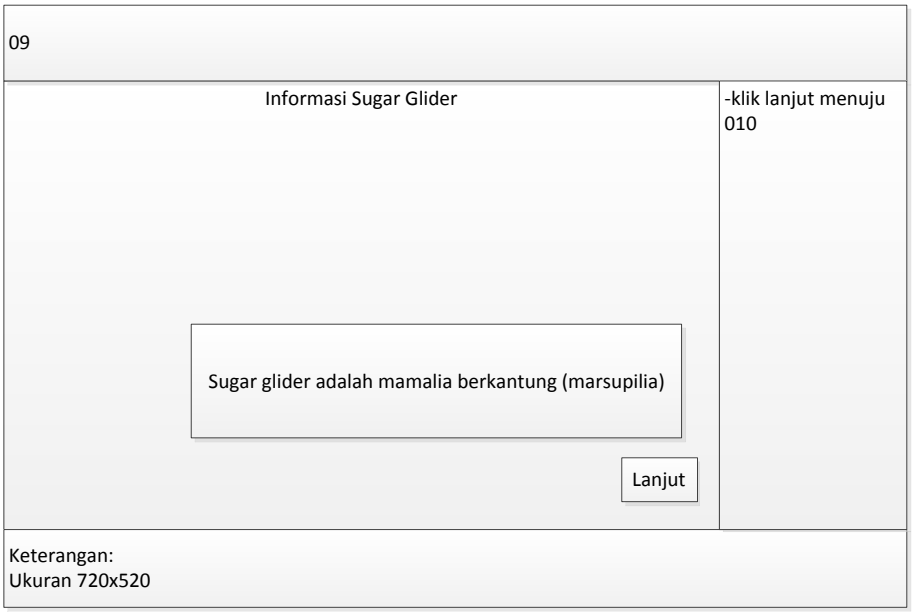
Gambar 3.57 Antarmuka level 1

8. Perancangan antarmuka berhasil menyelesaikan level 1



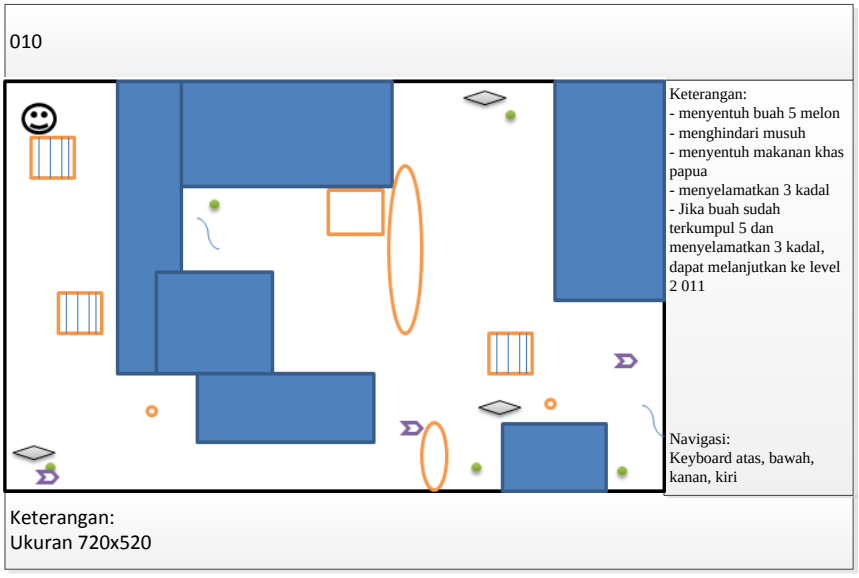
Gambar 3.58 Antarmuka berhasil menyelesaikan level 1

9. Perancangan antarmuka informasi sugar glider



Gambar 3.59 Antarmuka informasi sugar glider

10. Perancangan antarmuka level 2



Gambar 3.60 Antarmuka level 2

11. Perancangan antarmuka berhasil menyelesaikan level 2



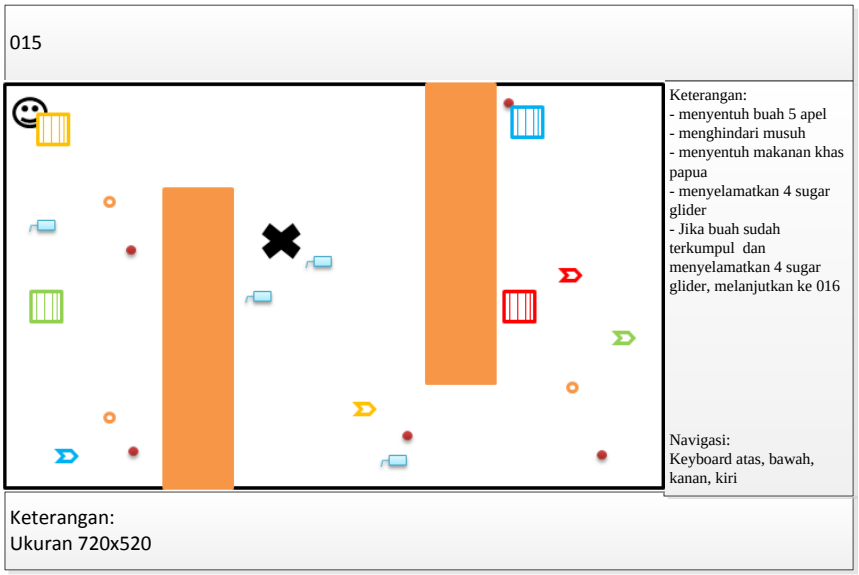
Gambar 3.61 Antarmuka berhasil menyelesaikan level 2

12. Perancangan antarmuka informasi sugar glider



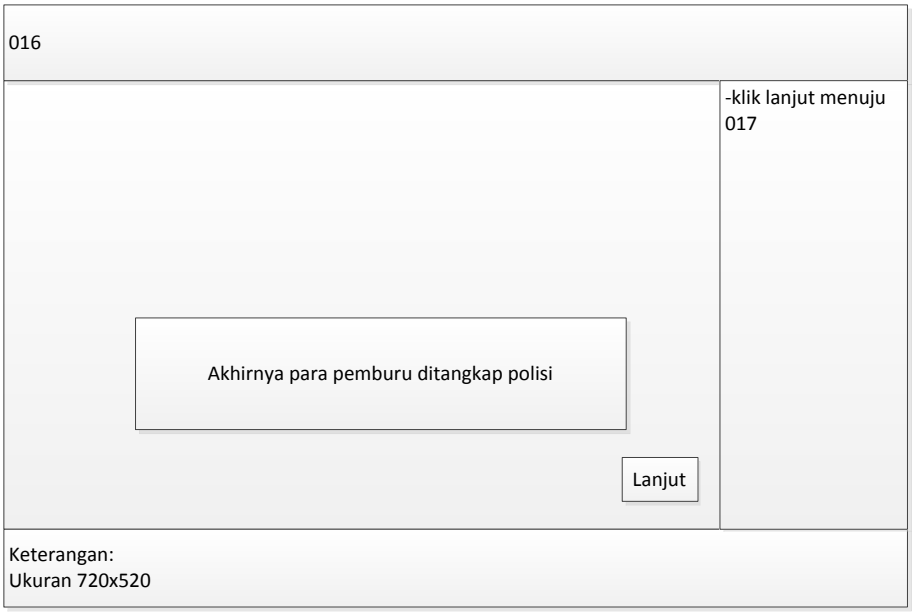
Gambar 3.62 Antarmuka informasi sugar glider

13. Perancangan antarmuka level 3



Gambar 3.63 Antarmuka level 3

14. Perancangan antarmuka bos pemburu ditangkap polisi



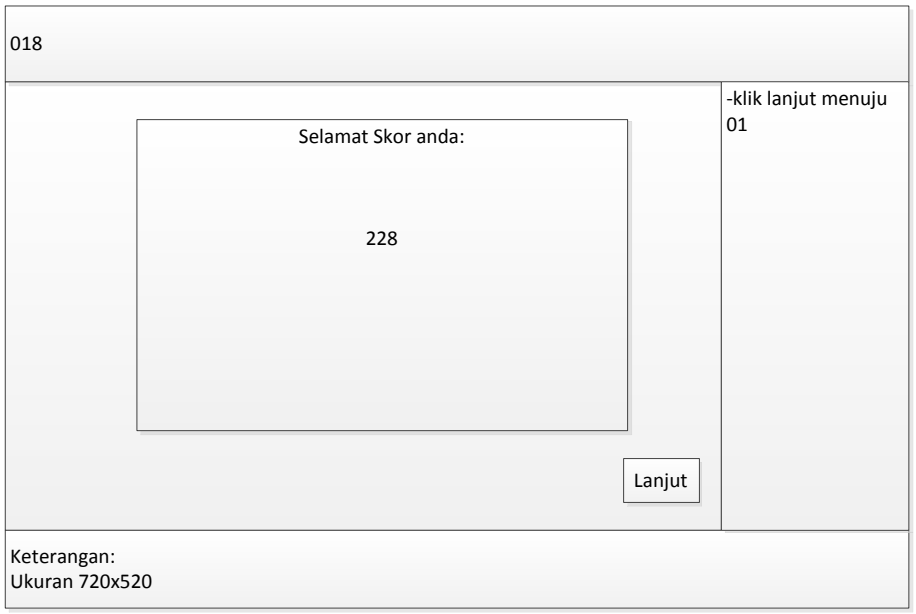
Gambar 3.64 Antarmuka bos pemburu ditangkap polisi

15. Perancangan antarmuka melepaskan sugar glider ke habitatnya



Gambar 3.65 Antarmuka melepaskan sugar glider ke habitatnya

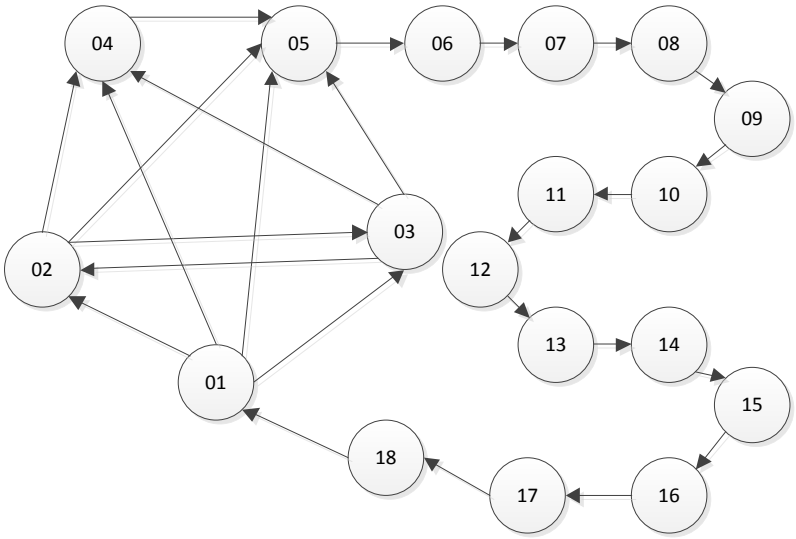
16. Perancangan antarmuka skor



Gambar 3.66 Antarmuka skor

3.2.4 Jaringan Semantik

Untuk mengetahui keterhubungan dari setiap tampilan yang dibuat dapat menggunakan jaringan semantik. Berikut gambar jaringan semantik pada *game* sugie man penyelamat sugar glider:



Gambar 3.67 Jaringan Semantik