BAB 3
ANALISIS DAN KEBUTUHAN ALGORITMA

3.1 Analisis Masalah

Berdasarkan latar belakang yang telah di jabarkan sebelumnya didapat suatu masalah yaitu, bagaimanakah cara menerapkan metode logika *fuzzy*, sehingga menghasilkan suatu *output* berupa keputusan yang dapat digunakan untuk mengatur perilaku NPC, agar NPC di dalam game ini dapat bersosialisasi dengan baik saat melakukan kegiatan tawar-menawar dengan karakter utama sebagai tantangan dalam memainkan game Pasar Jajanan ini.

Untuk mencapai tujuan tersebut maka akan diterapkannya suatu kecerdasan buatan dengan metode logika fuzzy yang menggunankan aturan Sistem Inferensi Fuzzy (FIS) model Mamdani, FIS model mamdani di pilih karena menggunakan perhitungan cenderung lebih rumit akan tetapi hasil yang diperoleh lebih dinamis dan manusiawi. Oleh karena pemrosesan perilaku NPC di dalam game ini membutuhkan hasil yang semanusiawi mungkin, maka dalam penelitian ini digunakan model Mamdani.

3.2 Analisis *Game*

Life-Simulation Game adalah cabang dari genre *Game* simulator, yaitu sebuah genre *Game*s yang dimana pemainnya bertanggung jawab atas sebuah tokoh atau karakter dalam memenuhi kebutuhan tokoh selayaknya kehidupan nyata namun dalam ranah virtual. Karakter memiliki kebutuhan dan kehidupan layaknya manusia, seperti kegiatan bekerja, bersosialisasi, makan, belanja, dan sebagainya.

Game Pasar Jajanan termasuk dalam genre *Game* simulasi karena *Game* ini mensimulasikan kegiatan jual beli dengan cara tawar-menawar seperti pada dunia nyata. *Game* ini berfokus pada seorang karakter yang bertugas menyelesaikan misi. Di dalam *Game* ini pemain akan menjadi seorang pembeli yang akan membeli suatu barang/item yang ada di sebuah Pasar. *Game* ini dibuat hanya bertujuan untuk menerapkan system jual beli tawar menawar seperti yang

biasa kita lakukan saat membeli barang-barang yang ada di Pasar maupun kaki lima. misi utama dalam *Game* ini adalah untuk membeli barang-barang yang tersebar di sebuah Pasar yang belatar belakang Kota Bandung. Di dalam *Game* ini terdapat suatu tantangan yaitu NPC yang bisa melakukan kegiatan tawar menawar dengan Karakter utama sebagai tantangan dalam *Game* ini.

Saat karakter utama akan membeli suatu barang di pedagang tertentu, maka pedagang akan memberikan harga tawaran pertama barang mereka dan pemain akan berusaha menawar harga barang tersebut, semakin rendah harga suatu barang maka akan semakin banyak sisa uang yang ada di kantong pemain. Setelah list seluruh barang terpenuhi sisa uang yang tersisa di kantong pemain akan di hitung dan akan di tukar menjadi point sebagai nilai dari misi. semakin banyak uang tersisa semakin banyak pula point yang di dapat.

3.2.1 Storyline

Game Pasar Jajanan bercerita tentang karakter utama yang di perankan oleh seorang pelancong yang sedang berada di Kota Bandung, dan ingin membeli sejumlah barang sebagai oleh-oleh dari perjalanan-nya, lalu sang karakter pergi mendatangi sebuah Pasar yang khusus menjual Jajanan khas Kota Bandung untuk membeli oleh-oleh tersebut. Tetapi karakter utama tidak mempunyai cukup uang untuk membeli barang-barang yang dia inginkan, jadi sang karakter utama harus mencari cara agar uang yang di milikinya bisa cukup untuk membeli oleh-oleh yang dia inginkan, maka agar seluruh barang yang di inginkannya dapat terbeli, karakter utama harus menawar harga yang di berikan oleh pedagang serendah mungkin agar uang yang di milikinya cukup untuk membeli barang tersebut.

3.2.2 Gameplay

Tugas utama pemain dalam memainkan *Game* ini yaitu menyelesaikan misi untuk membeli barang/items yang akan di berikan listnya dan sejumlah uang untuk membeli barang-barang tersebut, kesulitan atau tantangan dalam *Game* ini adalah dalam memenuhi misinya karakter akan di hadapakan dengan NPC (*Non player character*) yaitu pedagang yang mampu berinteraksi dengan karakter utama. Saat karakter utama akan membeli suatu barang di pedagang tertentu,

maka pedagang akan memberikan harga tawaran pertama barang mereka dan pemain akan berusaha menawar harga barang tersebut serendah mungkin, semakin rendah harga suatu barang maka akan semakin banyak sisa uang yang ada di kantong pemain. Setelah list seluruh barang terpenuhi sisa uang yang tersisa di kantong pemain akan di hitung dan akan di tukar menjadi point sebagai nilai dari misi. semakin banyak uang tersisa semakin banyak pula point yang di dapat.

Untuk memulai permainan, pemain akan di hadapkan dengan menu sebagai berikut:

1) Menu mulai berfungsi untuk memulai permainan.
2) Menu Petunjuk berfungsi untuk memberikan informasi tentang aturan-aturan yang diterapkan di dalam *Game*.
3) Menu keluar untuk keluar dari permainan.

Untuk memasuki permainan pemain harus memilih menu mulai, lalu pemain akan di hadapkan dengan sebuah tampilan informasi tentang misi dan jumlah uang yang ada di dompet karakter utama, di informasi misi akan di tampilkan list barang apa saja yang harus di beli oleh karakter utama untuk menyelesaikan misi, setelah itu klik tombol oke untuk melanjutkan permainan maka pemain akan masuk ke arena permainan yaitu sebuah pasar tradisional yang berada di kota bandung. lalu karakter utama akan bergerak berkeliling ke dalam pasar untuk mencari barang apa saja yang di butuhkannya untuk menyelesaikan misi.

Misalkan pemain mempunyai list misi yaitu membeli batagor,baju, dan gelang maka pemain tinggal mendatangi pedagang penjual barang tersebut. Sebagai contoh karakter utama akan membeli gelang, maka karakter akan mendatangi toko *sovenir* dan memilih item gelang, setelah item gelang dipilih maka *system* akan menampilkan harga awal tawaran gelang dari pedagang, dan akan keluar dua opsi yaitu beli dan tidak jadi beli. bila memilih beli maka pemain akan masuk ke menu negosiasi berisi opsi tawar untuk menawar dan deal untuk membeli dan bila memilih tidak jadi beli pemain akan kembali ke pasar.

bila pemain memilih opsi tawar,maka pemain akan masuk ke menu tawar. Di menu tawar akan ditampilkan *input box* untuk meng*input* harga yang kita inginkan setelah menekan tombol enter maka system akan memproses dengan metode *fuzzy* dan akan menghasilkan keputusan apakah barang tersebut akan dijual atau tidak, bila NPC setuju untuk menjual barang maka otomatis uang karakter akan terpotong dan list di misi akan berkurang, bila pedagang tidak mau menjual barang maka pemain otomatis akan di hadapkan dengan menu *input* harga tawaran baru.

Permainan akan berakhir dengan dua kondisi yang pertama bila seluruh list misi telah terpenuhi yang berarti misi selesai dan yang kedua bila uang pemain telah habis sebelum list misi terpenuhi yang berarti misi gagal dan akan langsung di hadapkan dengan tampilan *Game* over.

3.3 Analisis Masukan

Pada penelitian ini akan dikembangkan metode untuk membangun elemen *Game* yaitu NPC yang mampu merespon harga tawaran dari karakter utama, dengan menghasilkan suatu keputusan yang dapat digunakan dalam negosiasi tawar menawar.

Dalam sebuah *Game*, perilaku NPC dipengaruhi oleh attribut fisik yang dimiliki oleh setiap karakter NPC. Mascharenhas [5] dalam tulisannya menyebutkan attribut fisik pada karakter bermacam-macam sesuai dengan jenis *Game*. Attribute yang akan di gunakan dalam *Game* ini berjumlah tiga *variable* yaitu, dua *variable input* dan satu *variable output* dan masing-masing *variable* memiliki nilai *linguistic* sebagai berikut :

Variable input :

1. *Variable* Harga tawaran. Dengan nilai *linguistic* : Rendah, Sedang ,dan tinggi.

Pada *variable input* harga tawaran tersebut, nilai *input* masukan awal adalah berupa nilai mata uang yang akan di konversi kedalam persen (%)

Contoh konversi nilai *linguistic* pada *variable* harga tawar :

Bila harga barang adalah 20000 dan harga tawar yang di *input* adalah 15000, nilai tersebut di konversi kedalam persen (%) menjadi $\frac{15000}{20000} x\ 100\% = 75\%$. nilai linguistic 75% tersebut berada pada kategori rendah karena kategori sedang memiliki range (60%-90%), dapat dilihat pada gambar 3.2.

2. *Variable* Emosi. Dengan nilai *Linguistic* Marah, Tenang, dan senang.

Pada *variable input* Emosi, nilai *input* masukan awal adalah berupa nilai *random* (acak) dengan range 0-1 dan nilai *random* yang didapat akan disesuai dengan Emosi pada *variable linguistic*. Nilai *random* adalah nilai awal yang digunakan untuk proses fuzzy pada *variable* Emosi, nilai tersebut dapat berubah sesuai dengan dialog yang diterima oleh NPC.

Contoh nilai hasil random Pada *variable input* Emosi

Bila nilai hasil random dari *variable* Emosi adalah (0.7) maka Emosi yang akan berjalan pada NPC adalah Senang karena Emosi senang ada pada range (0.6-1). dapat dilihat pada gambar 3.3.

Variable Output:

1. *Variable* Keputusan Jual. Dengan nilai *Linguistic* : Tidak Jual, Naikan Tawaran, dan Jual.

Variable keputusan jual ini berfungsi sebagai *output* dari proses Defuzzifikasi dari nilai *input* yang ada. Berikut akan diterangkan kondisi yang berjalan sesuai nilai *output fuzzy* yang di hasilkan.

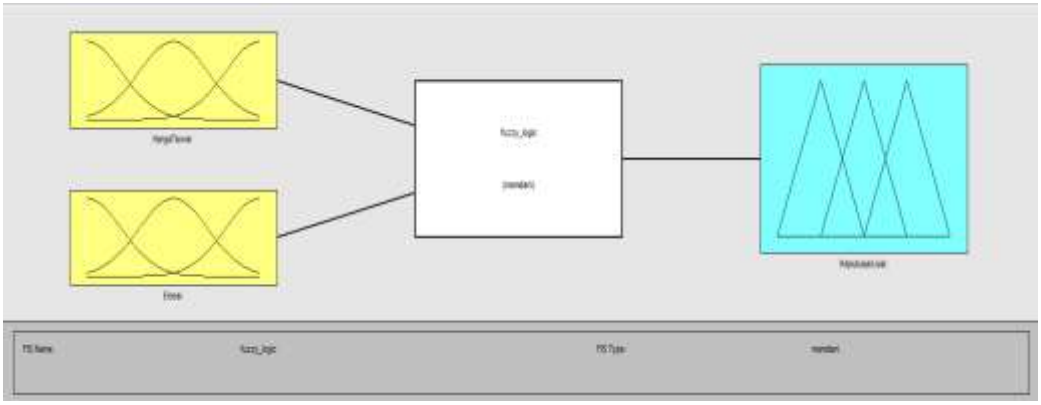
1) Bila keputusan jual adalah “jual” maka uang pemain akan berkurang sebanyak harga akhir dari negosiasi.
2) Bila keputusan jual adalah “naikan harga” maka NPC akan memberikan *alternative* harga baru diantara harga awal barang dan harga tawar pemain sesuai dengan status emosi NPC.
3) Bila keputusan jual adalah “tidak jual” maka negosiasi akan berakhir.

Proses *input* menggunakan fungsi keanggotaan trapesium dan proses *output* menggunakan fungsi keanggotaan segitiga. Seluruh proses tersebut menggunakan tiga variabel *linguistic*. Nilai dari variabel *linguistic* ini merepresentasikan besar peluang suatu *input* terhadap keputusan.

3.4 Analisis Metode / Algoritma

Dalam penelitian ini akan dikembangkan sebuah kecerdasan buatan dengan aturan logika *fuzzy* yang mempu memberikan NPC Kemampuan untuk melakukan prosese tawar menawar dengan karakter utama. logika *fuzzy* berfungsi untuk mengambil keputusan terhadap respon yang akan dihasilkan oleh NPC sesuai dengan jenis emosi yang diberikan.

Untuk lebih memahami model yang akan dikembangkan dalam penelitian ini, akan digambarkan pada bagan di bawah ini :



Gambar 3. 1 Model kecerdasan buatan

Pada gambar 3.1 diatas dijelaskan bahwa NPC memiliki 2 *variable input* yaitu *variable* Harga tawar dan *variable* Emosi yang kemudian akan di proses melalu proses *fuzzy logic* dengan system inferensi mamdani yang kemudian mengeluarkan *output* berupa *variable* keputusan jual.

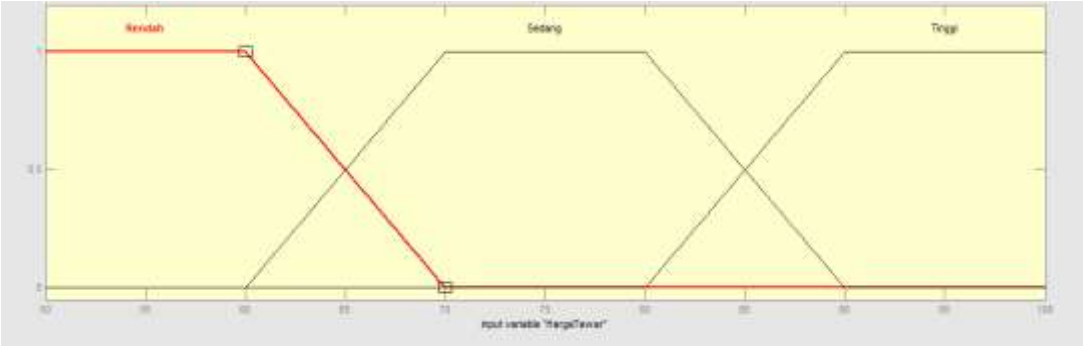
Sebuah sistem yang dibangun menggunakan logika *fuzzy* pada dasarnya akan mempunyai komponen-komponen sebagai berikut:

3.4.1 Variabel *Linguistic*

Variabel *linguistic* merupakan suatu interval numerik dan mempunyai

nilai-nilai *linguistic*, yang semantiknya didefinisikan oleh fungsi keanggotaannya [8].Dalam penelitian ini akan digunakan tiga *variable input-output* yang masing-masing memiliki tiga nilai *Linguistic* yang di gunakan sebagai parameter untuk proses *fuzzy* yaitu :

1. *Variable* Harga tawaran Dengan nilai *Linguistic* : Rendah (50%-70%), Sedang (60%-90%),dan tinggi (80%-100%).



Gambar 3. 2 Variable Harga tawaran Dengan nilai Linguistik

Pada gambar 3.2 menunjukan sebuah grafik *Variable* Harga tawaran yang mempunyai range nilai dari 50%-100%, setiap nilai *linguistic* dari *Variable* Harga tawaran seperti Rendah, Sedang, dan tinggi mempunyai nilai fuzzyfikasi yang berbeda-beda.

1) Variabel *linguistic* Rendah yang memiliki range nilai antara 50-70, nilai fuzzyfikasinya dihitung berdasarkan fungsi trapesium, berikut persamaannya :

$$\mu Rendah[x] \begin{cases} 1; x \leq 60 \\ \frac{70-x}{10}; 60 \leq x \leq 70 \\ 0; x \geq 70 \end{cases} \quad \textbf{(3.1)}$$

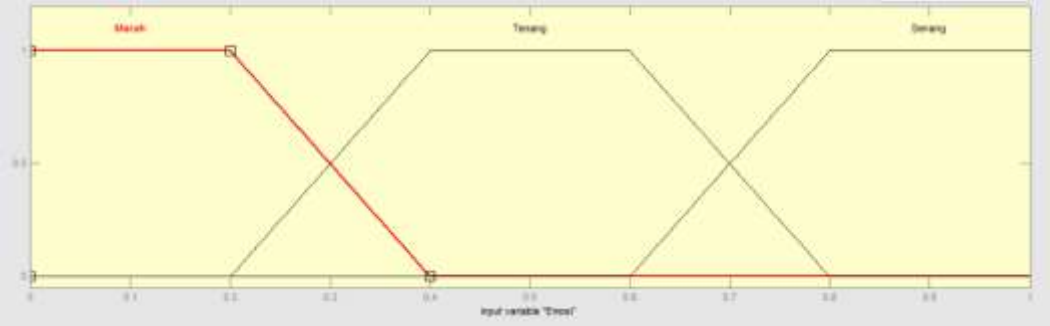
2) Variabel *linguistic* Sedang yang memiliki range nilai antara 60-90, nilai fuzzyfikasinya dihitung berdasarkan fungsi trapesium, berikut persamaannya :

$$\mu Sedang[x] \begin{cases} 0 & ;x \leq 60, atau, x \geq 90 \\ \frac{x-60}{10} & ;60 \leq x \leq 70 \\ 1 & ;70 \leq x \leq 80 \\ \frac{90-x}{10} & ;80 \leq x \leq 90 \end{cases} \quad \textbf{(3.2)}$$

3) Variabel *linguistic* Tinggi yang memiliki range nilai antara 80-100, nilai fuzzyfikasinya dihitung berdasarkan fungsi trapesium, berikut persamaannya :

$$\mu Tinggi[x]\begin{cases}0; x \leq 80\\ \frac{x-80}{10}; 80 \geq x \geq 90\\ 1; x \geq 90\end{cases} \quad \textbf{(3.3)}$$

2. *Variable* Emosi. Dengan nilai *Linguistic* : Marah (0-0.4), Tenang(0.2-0.8), dan senang(0.6-1).



Gambar 3. 3 Variable Emosi. Dengan nilai Linguistik

Pada gambar 3.3 menunjukan sebuah grafik *Variable* Emosi yang mempunyai range nilai dari 0-1, setiap nilai *linguistic* dari *Variable* Emosi seperti Marah, Tenang, dan Senang mempunyai nilai fuzzyfikasi yang berbeda-beda.

1) Variabel *linguistic* Marah yang memiliki range nilai antara 0-0.4, nilai fuzzyfikasinya dihitung berdasarkan fungsi trapesium, berikut persamaannya :

$$\mu Marah[x]\begin{cases}1; x \leq 0.2\\ \frac{0.4-x}{0.2}; 0.2 \leq x \leq 0.4\\ 0; x \geq 0.4\end{cases} \quad \textbf{(3.4)}$$

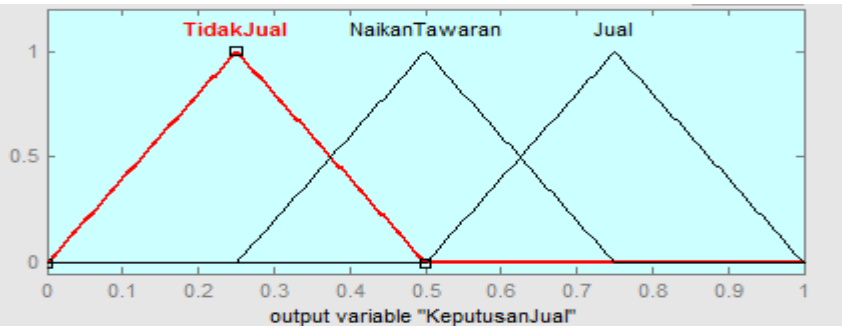
2) Variabel *linguistic* Tenang yang memiliki range nilai antara 0.2-0.8, nilai fuzzyfikasinya dihitung berdasarkan fungsi trapesium, berikut persamaannya :

$$\mu Sedang[x]\begin{cases} 0 & ;x\leq 0.2\ ,atau,x\geq 0.8 \\ \frac{x-60}{10} & ;0.2 \leq x \leq 0.4 \\ 1 & ;0.4\leq x\leq 0.6 \\ \frac{90-x}{10} & ;0.6 \leq x \leq 0.8 \end{cases} \quad \textbf{(3.5)}$$

3) Variabel *linguistic* Senang yang memiliki range nilai antara 0.6-1, nilai fuzzyfikasinya dihitung berdasarkan fungsi trapesium, berikut persamaannya :

$$\mu Senang[x]\begin{cases} 0; x \leq 0.6 \\ \frac{x-0.6}{0.2}; 0.6 \geq x \geq 0.8 \\ 1; x \geq 0.8 \end{cases} \quad \textbf{(3.6)}$$

3. *Variable* Keputusan Jual. Dengan nilai *Linguistic* : Tidak Jual (0-0.4), Naikan Tawaran (0.2-0.8), dan Jual (0.6-1).



Gambar 3. 4 *Variable* Keputusan Jual. Dengan nilai *Linguistic*

Pada gambar 3.4 menunjukan sebuah grafik *Variable* Keputusan Jual yang mempunyai range nilai dari 0-1, setiap nilai *linguistic* dari *Variable* Keputusan Jual seperti Tidak Jual , Naikan Tawaran, dan Jual. mempunyai nilai fuzzyfikasi yang berbeda-beda.

1) Variabel *linguistic* Tidak Jual yang memiliki range nilai antara 0-0.5, nilai fuzzyfikasinya dihitung berdasarkan fungsi Segitiga, berikut persamaannya :

$$\mu TidakJual[x]\begin{cases} \frac{x-0}{0.25} & ;0 \leq x \leq 0.25 \\ \frac{0.5-x}{0.25} & ;0.25 \leq x \leq 0.5 \\ 0 & ;x \geq 0.5 \end{cases} \quad \textbf{(3.7)}$$

2) Variabel *linguistic* Naikan Tawaran yang memiliki range nilai antara 0.25-0.75, nilai fuzzyfikasinya dihitung berdasarkan fungsi Segitiga, berikut

persamaannya :

$$\mu NaikanTawaran[x]\begin{cases} 0 & ;x \leq 0.25\,, atau, \geq 0.75 \\ \frac{x-0.25}{0.25} & ;0.25 \leq x \leq 0.5 \\ \frac{0.75-x}{0.25} & ;0.25 \leq x \leq 0.75 \end{cases} \quad \textbf{(3.8)}$$

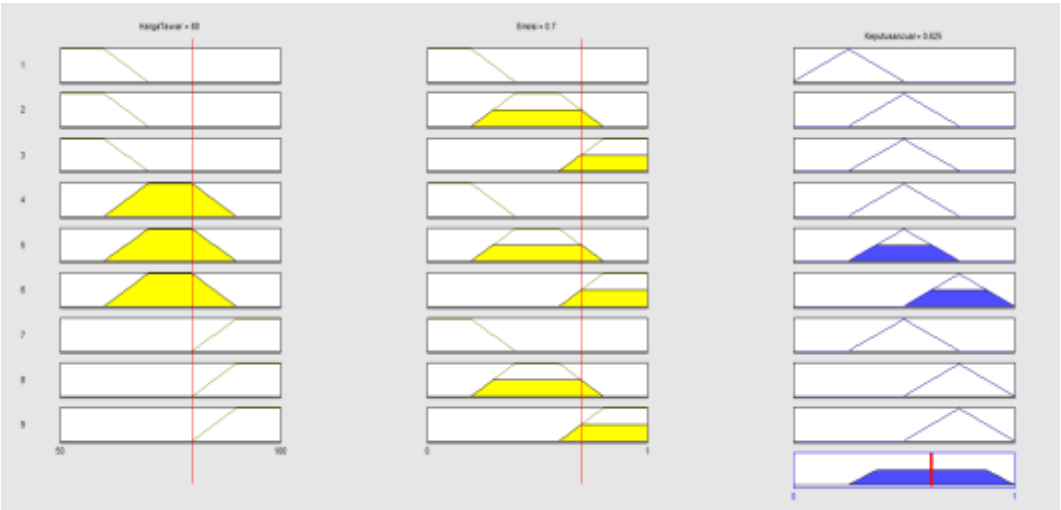
3) Variabel *linguistic* Jual yang memiliki range nilai antara 0.5-1, nilai fuzzyfikasinya dihitung berdasarkan fungsi Segitiga, berikut persamaannya :

$$\mu Senang[x]\begin{cases} 0 & ;x \leq 0.5 \\ \frac{x-0.5}{0.25} & ;0.5 \leq x \leq 0.75 \\ \frac{1-x}{0.25} & ;0.75 \leq x \leq 1 \end{cases} \quad \textbf{(3.9)}$$

3.4.2 Fuzzification

Fuzzification merupakan proses konversi nilai-nilai kebenaran yang bersifat pasti (*Crisp input*), yang digunakan sebagai *input*an, menjadi bentuk *fuzzy input*, yang berupa nilai *linguistic* yang semantiknya ditentukan berdasarkan fungsi keanggotaan [8].

Proses ini menggunakan fungsi keanggotaan trapesium dan tiga variabel *linguistic*. Nilai dari variabel *linguistic* ini merepresentasikan besar peluang *variable input* terhadap keputusan sebagai output, yang disimulasikan pada gambar 3.5 di bawah ini.



Gambar 3. 5 Simulasi besar peluang *variable input* terhadap keputusan

3.4.3 Inference

Dalam suatu sistem berbasis aturan *fuzzy*, proses inference memperhitungkan semua aturan yang ada di dalam basis pengetahuan [8]. Model yang digunakan dalam proses inference dalam penelitian ini adalah model Mamdani. Proses inference menggunakan aturan sebagai berikut :

1. If (HargaTawar is Rendah) and (Emosi is Marah) then (KeputusanJual is TidakJual) (1)
2. If (HargaTawar is Rendah) and (Emosi is Tenang) then (KeputusanJual is NaikanTawaran) (1)
3. If (HargaTawar is Rendah) and (Emosi is Senang) then (KeputusanJual is NaikanTawaran) (1)
4. If (HargaTawar is Sedang) and (Emosi is Marah) then (KeputusanJual is NaikanTawaran) (1)
5. If (HargaTawar is Sedang) and (Emosi is Tenang) then (KeputusanJual is NaikanTawaran) (1)
6. If (HargaTawar is Sedang) and (Emosi is Senang) then (KeputusanJual is Jual) (1)
7. If (HargaTawar is Tinggi) and (Emosi is Marah) then (KeputusanJual is NaikanTawaran) (1)
8. If (HargaTawar is Tinggi) and (Emosi is Tenang) then (KeputusanJual is Jual) (1)
9. If (HargaTawar is Tinggi) and (Emosi is Senang) then (KeputusanJual is Jual) (1)

Dimana ada 9 *rule* yang akan menjadi parameter dalam pengambilan keputusan.

3.4.4 Defuzzification

Proses *defuzzification* memiliki bermacam-macam metode yang bisa diaplikasikan untuk menangani berbagai macam masalah. Dalam penelitian ini akan digunakan salah satu metode yang disebut Metode *Centroid*. Metode ini sering disebut juga sebagai *center of area* atau *center of gravity*. Metode *Centroid* ini digunakan karena dirasa paling cocok untuk mengimplementasikan logika fuzzy ke dalam Npc di game pasar jajanan ini. Karena metode centroid ini dapat digunakan untuk menetapkan nilai *crisp* dengan cara mengambil titik pusat daerah fuzzy, sehingga mudah untuk di konversikan ke dalam perilaku dari NPC[8].
Metode ini menggunakan rumus sebagai berikut :

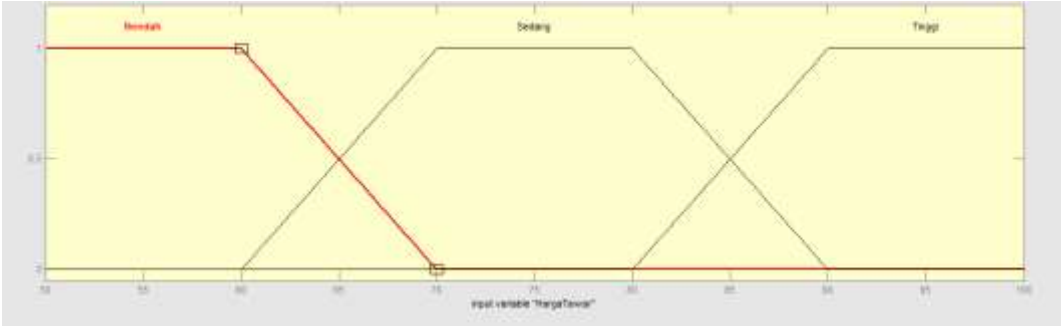
$$z^* = \frac{\sum_{j=1}^{n} z_j \mu(z_j)}{\sum_{j=1}^{n} \mu(z_j)} \qquad \textbf{(3.10)}$$

Di mana z adalah hasil proses *fuzzy* dan zμR(z) adalah derajat keanggotaan dari z. Hasil dari proses *defuzzification* adalah suatu nilai yang merupakan representasi dari perubahan nilai atribut dari NPC.

3.4.5 Perhitungan *Fuzzy*

Berdasarkan *fuzzy interface system* di atas maka pemetaan input-output adalah sebagai berikut:

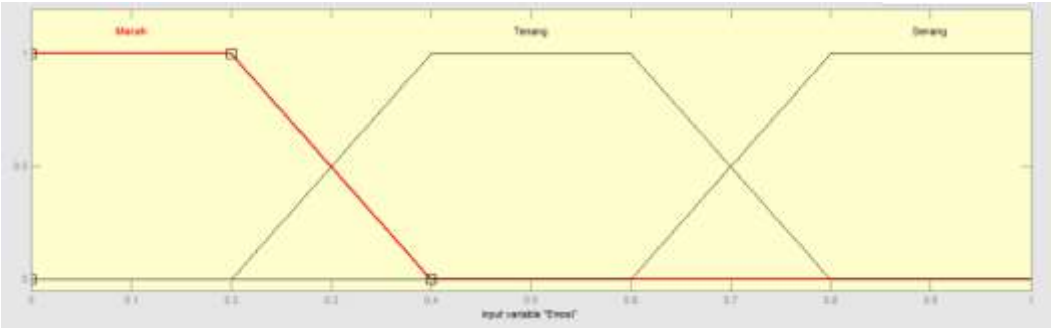
1. *Variable* Harga tawaran Dengan nilai *Linguistic* : Rendah (50%-70%), Sedang (60%-90%),dan tinggi (80%-100%).



Gambar 3. 6 Variable Harga tawaran Dengan nilai *linguistic* range dari (0-1)

Pada gambar 3.6 menunjukan sebuah grafik Harga tawar yang mempunyai range nilai dari 50-100, setiap nilai *linguistic* dari variabel Harga tawar memiliki nilai *fuzzyfikasi* yang berbeda-beda.

2. *Variable* Emosi, Dengan nilai *Linguistic* range dari (1-0) :
   Marah (0-0.4), Tenang(0.2-0.8), dan senang(0.6-1).

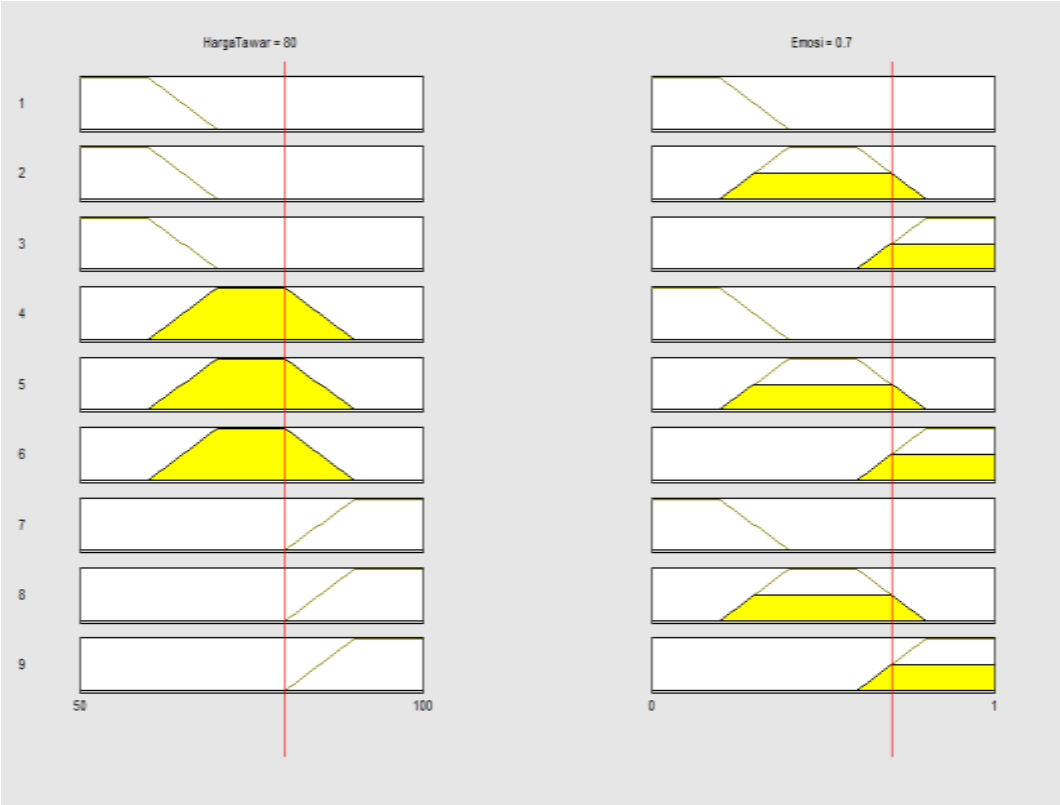


Gambar 3. 7 Variable Emosi. Dengan nilai *linguistic* range dari (1-0)

Pada gambar 3.7 menunjukan sebuah grafik *Variable* Emosi yang mempunyai range nilai dari 0-1, setiap nilai *linguistic* dari variabel Emosi memiliki nilai fuzzyfikasi yang berbeda-beda.

1. *Fuzzyfication*

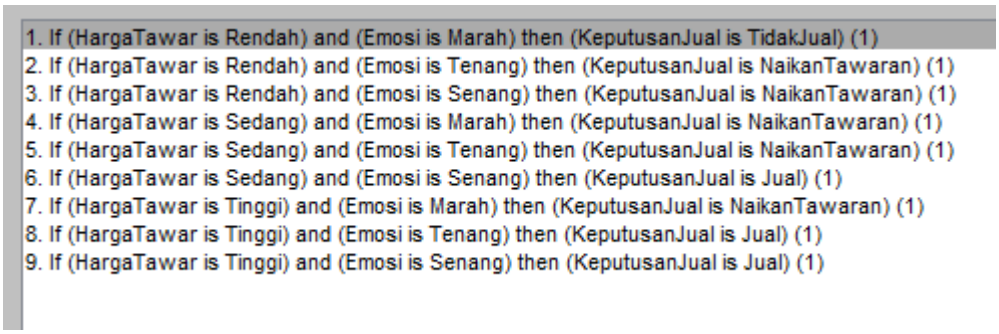
Fuzzyfication yaitu suatu proses untuk mengubah suatu masukan dari bentuk tegas (*Crisp*) menjadi *fuzzy* (variabel linguistik). Proses *Fuzzy*fikasi untuk Harga tawar 80% dan nilai emosi 0.7 adalah sebagai berikut :



Gambar 3. 8 Perhitungan *Fuzzyfication* untuk Harga tawar

2. *Inference*

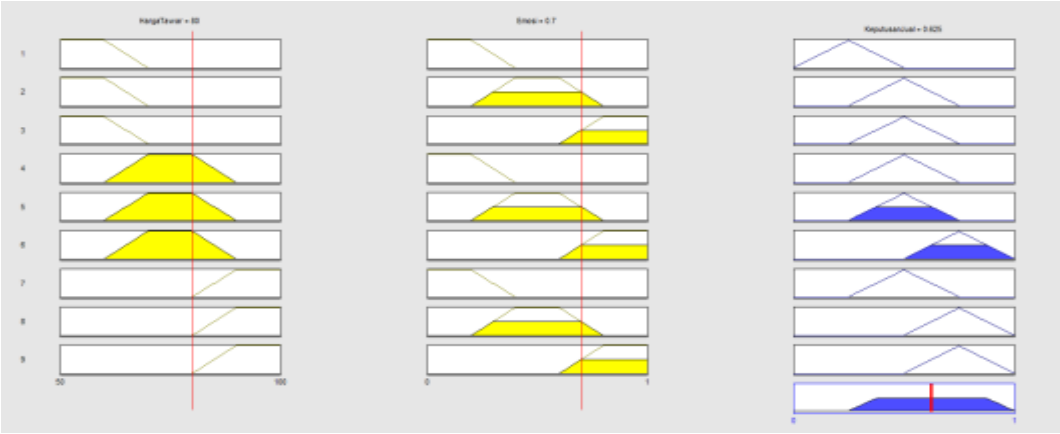
Inference yaitu proses melakukan penalaran menggunakan *fuzzy input* dan *fuzzy rules* yang telah ditentukan sehingga menghasilkan *fuzzy output* . Setelah nilai derajat keanggotaan didapatkan, maka nilai-nilai derajat tersebut harus dibandingkan sesuai dengan *Rule* yang ada, tujuannya untuk mencari nilai minimum untuk setiap *rule* yang dihitung. *rule* yang di pakai dalam proses inference adalah sebagai berikut.



Gambar 3. 9 *Rule* pada proses *fuzzy*

3. Defuzzification

Defuzzification yaitu mengubah *fuzzy output* menjadi *Crisp* value (nilai pasti) berdasarkan fungsi keanggotaan yang telah ditentukan. Gambar 3.10 Berikut ini menggambarkan proses defuzzifikasi yang berjalan.



Gambar 3. 10 Proses *defuzzification*

Jadi Keputusan yang dihasilkan berdasarkan dari nilai harga tawar [80] dan emosi [0.7] adalah keputusan naikan tawaran karena sesuai dengan nilai Z* yang didapatkan yaitu [0.625]. dapat dilihat di gambar 3.4.

3.5 Kebutuhan Non Fungsional

Analisis kebutuhan non-fungsional dilakukan untuk mengetahui spesifikasi kebutuhan untuk sistem. Spesifikasi kebutuhan melibatkan analisis perangkat keras/hardware dan analisis perangkat keras Software.

3.5.1 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.

Analisis kebutuhan perangkat keras digunakan untuk mengetahui spesifikasi minimum perangkat keras dalam mengimplementasikan metode logika *fuzzy* dan menjalankan *Game* ini.

Table 3. 1 Spesifikasi Perangkat Keras minimum

Nama Perangkat	Spesifikasi
Processor	1.8 Ghz
RAM	512 Mb
Hard disk	1 Gb
Display	Led 20 inc (800x600)
VGA	256 Mb
Keyboard & Mouse	Standar

Table 3.1 mendeskripsikan spesifikasi perangkat keras minimum untuk mengimplementasikan algoritma fuzzy dan menjalankan game ini.

3.5.2 Analisis Kebutuhan Perangkat Lunak

Analisi kebutuhan Perangkat Lunak adalah suatu proses dalam mempelajari kebutuhan pembuat sistem, dengan tujuan untuk mendapatkan seluruh kebutuhan di dalam sistem atau perangkat yang digunakan dalam sebuah *system*.

Table 3. 2 Spesifikasi Perangkat Lunak Pembangun

Nama Perangkat Lunak	Spesifikasi
Sistem Operasi	Microsoft Windows 8.1
Tools	Unity versi 4.0
	3ds Max 2011
	Google skechup 8
	Matlap

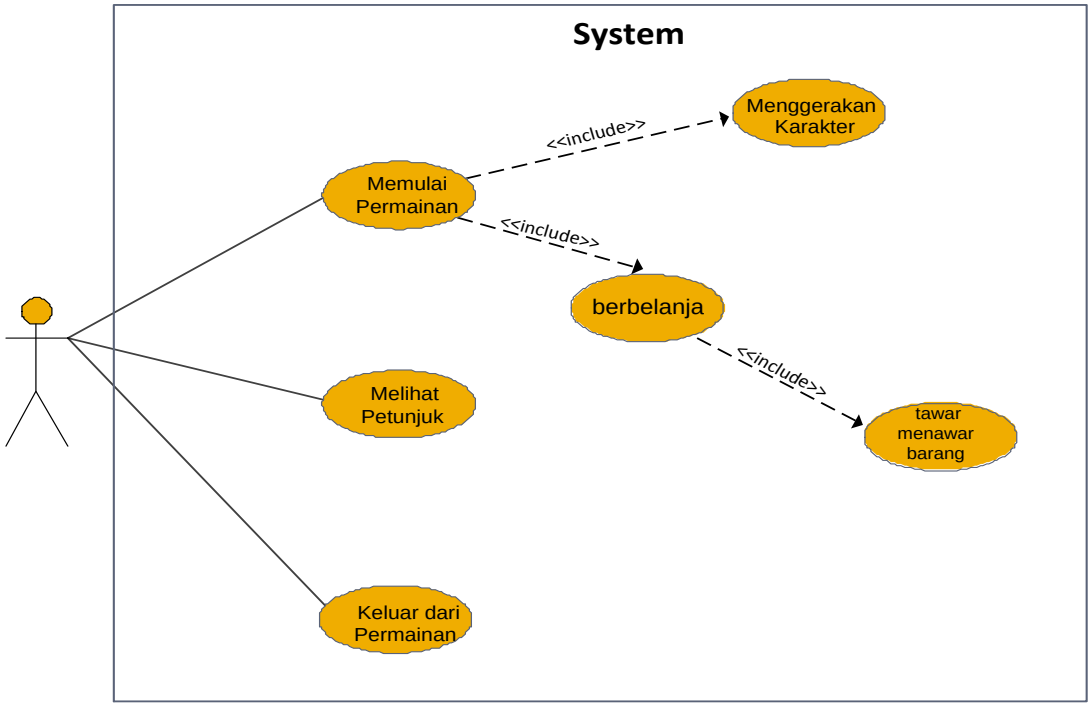
Table 3.2 mendeskripsikan spesifikasi perangkat lunak dalam mengimplementasikan algoritma pada Game ini.

3.6 Kebutuhan Fungsional

Analisis kebutuhan fungsional dilakukan untuk mengetahui gambaran sistem yang akan dibangun. Analisis kebutuhan fungsional terdiri dari pendefinisan fungsi *Game*, penggambaran alir aktivitas pada *Game* dan penggambaran interaksi antar objek pada *Game*.

3.6.1 Use Case Diagram

Penggambaran fungsi *Game* berdasarkan interaksi antar aktor dan objek pada sistem digambarkan dengan menggunakan *Diagram* use case yang terdapat pada gambar 3.11 berikut ini.



3.6.1.1 Definisi Actor

Definisi aktor berfungsi untuk menjelaskan aktor yang terdapat pada *Diagram use case.*

Table 3. 3 Definisi Aktor

No.	Aktor	Deskripsi
1.	Pemain	orang yang memainkan *Game*.

3.6.1.2 Definisi *Use Case*

Definisi *use case* berfungsi untuk menjelaskan fungsi *use case* yang terdapat pada *Diagram* use case. Definisi use case diterangkan pada Table 3.4 dibawah ini.

Table 3. 4 Definisi Use Case

No	Use Case	Deskripsi
1.	Memulai permainan	Proses bermain *Game*.
2.	Melihat petunjuk	Proses untuk menampilkan petunjuk atau panduan cara bermain.
3.	Keluar dari permainan	Proses keluar dari aplikasi permainan.
4.	Menggerakan karakter	Proses untuk menggerakkan karakter dengan animasi berjalan
5.	Berbelanja	Proses jual beli antar karakter utama dan NPC
6.	Tawar-menawar barang	Proses negosiasi karakter utama dengan pedagang untuk mendapatkan harga termurah.

3.6.1.3 Skenario Use Case

Skenario Use Case menunjukan proses apa yang terjadi pada setiap bagian di dalam use case tersebut, dimana *user* memberikan perintah pada bagian-bagian use case dan repon apa yang di berikan oleh system kepada *user* setelah *user* memberikan perintahnya. Skenario use case di tunjukan pada tabel 3.5 sampai tabel 3.10 dibawah ini

Table 3. 5 Scenario Use Case Memulai Permainan

Identifikasi	
Nomor	1

Nama	Memulai Permainan
Tujuan	Untuk memulai bermain *game*
Deskripsi	Proses untuk menampilkan permainan
Aktor	Pengguna
Skenario Utama	
Kondisi awal	Pemain berada di menu utama
Aksi Aktor	**Reaksi Sistem**
1.Membuka aplikasi *Game*	2.Menampilkan menu utama.
3.Memilih menu mainkan.	4.Menampilkan permainan.

Table 3. 6 Scenario Use Case Melihat petunjuk

Identifikasi	
Nomor	2
Nama	Melihat Petunjuk
Tujuan	Untuk Menampilkan Petunjuk permainan
Deskripsi	Proses untuk menampilkan Petunjuk
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di menu utama
Aksi Aktor	**Reaksi Sistem**
1.Membuka aplikasi *Game*	2.Menampilkan menu utama.
3. Memilih menu Petunjuk	4. Menampilkan petunjuk

Table 3. 7 Scenario Use Case Keluar dari permainan

Identifikasi	
Nomor	3
Nama	Keluar dari permainan
Tujuan	Untuk keluar dari permainan
Deskripsi	Proses untuk meninggalkan permainan
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di menu utama
Aksi Aktor	**Reaksi Sistem**
1.Membuka aplikasi *Game*	2.Menampilkan menu utama.
3. Memilih menu keluar	4.Keluar dari permainan

Table 3. 8 Scenario Use Case Menggerakan karakter

Identifikasi	
Nomor	4
Nama	Menggerakan karakter
Tujuan	Menggerakkan karakter
Deskripsi	Proses untuk menggerakan karakter atau tokoh utama yang dimainkan dalam permainan menggunakan tombol *keybord*
Aktor	Pengguna
Skenario Utama	
Kondisi awal	Pemain menekan tombol pada *keyboard*
Aksi Aktor	**Reaksi Sistem**
1.Menekan tombol w Pada *Keyboard*.	2.Menggerakan karakter berjalan kedepan
Skenario Alternatif 1	
1.Menekan tombol S Pada *Keyboard*.	2.Menggerakan karakter berjalan mundur
Skenario Alternatif 2	
1.menekan tombol D Pada *Keyboard*.	2.Menggerakan karakter berjalan kekanan
Skenario Alternatif 3	
1.Pemain menekan tombol A Pada *Keyboard*.	2.Menggerakan karakter berjalan kekiri

Table 3. 9 Scenario Use Case Belanja

Identifikasi	
Nomor	5
Nama	Berbelanja
Tujuan	Membeli barang yang di jual oleh NPC
Deskripsi	Proses kareakter utama membeli barang yang di jual oleh NPC
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di depan pasar untuk mencari barang yang ingin di beli
Aksi Aktor	**Reaksi Sistem**
1. Mendatangi pedagang	
2. Klik Pedagang	3.Menampilkan barang dagangan

4.Memilih barang	5. Menampilkan harga barang

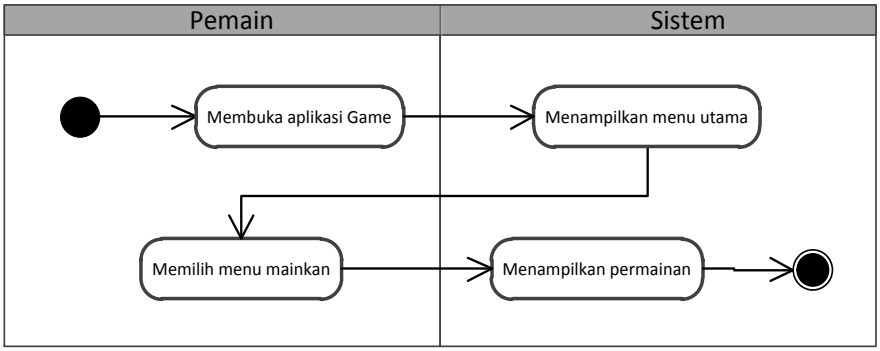
Table 3. 10 Scenario Use Case Tawar-menawar

Identifikasi	
Nomor	6
Nama	tawar menawar
Tujuan	Membeli barang dengan harga lebih murah dari harga yang diberikan oleh pedagang/NPC
Deskripsi	Proses Menawar harga
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain telah memilih barang yang ingin di beli.
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan harga barang
2. Memilih opsi beli	3. Memotong uang pemain
	4. Menampilkan informasi transaksi
Skenario alternative 1	
	1. Menampilkan harga barang
2.Memilih opsi tawar	2. Menampilkan form input harga tawar
3.Menginput harga tawar	4. Mengeluarkan keputusan jual
	5.bila keputusan jual adalah “jual” maka akan lanjut ke Skenario utama proses ke 3
	6.bila keputusan jual adalah “naikan harga” maka sistem akan menampilkan harga barang alternatif dan kembali ke proses 3
	7.bila keputusan jual adalah “tidak jual” maka negosiasi berakhir
Skenario alternative 2	
	1.Menampilkan harga barang
2.Memilih opsi keluar	3.Keluar dari lapak pedagang

3.6.2 Activity Diagram

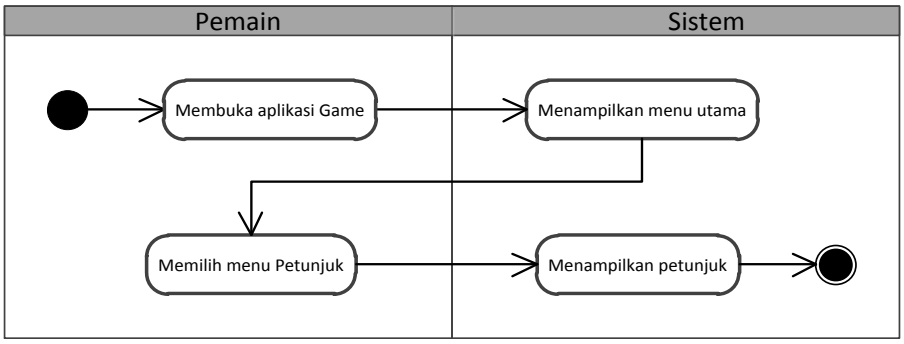
Diagram Activity menggambarkan proses-proses yang terjadi dari dimulai aktifitas sampai aktifitas berhenti pada sistem yang sedang dirancang. Bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir.

3.6.2.1 Activity Diagram Memulai Permainan



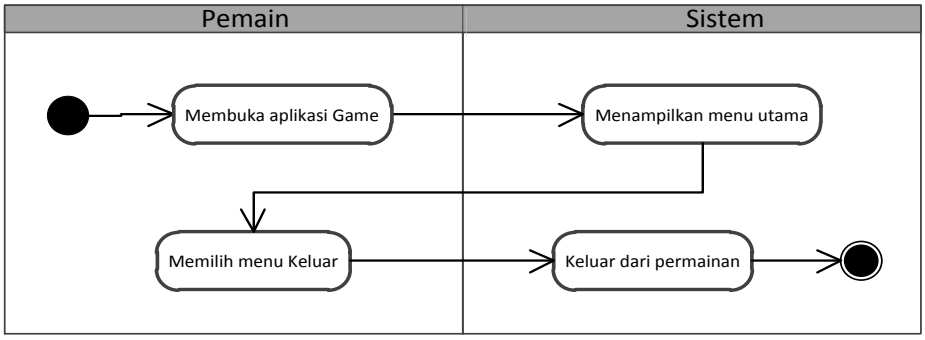
Gambar 3. 12 Activity Diagram memulai permainan

3.6.2.2 Activity Diagram Melihat Petunjuk



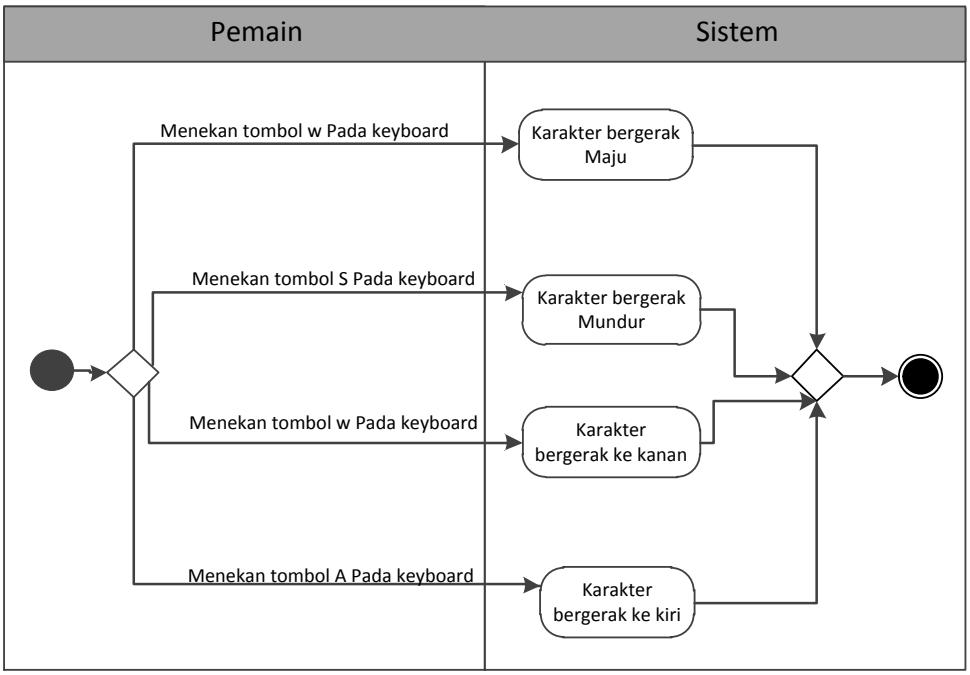
Gambar 3. 13 Activity Diagram melihat Petunjuk

3.6.2.3 *Activity Diagram* keluar dari permainan



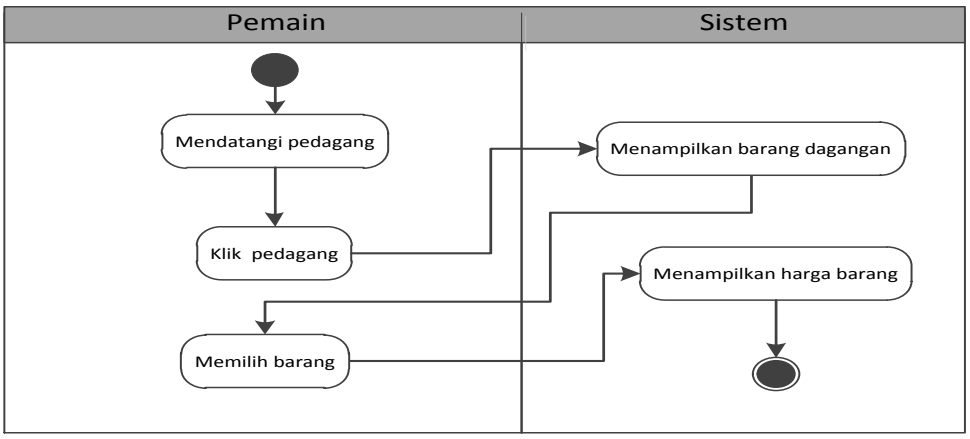
Gambar 3. 14 Activity Diagram keluar dari permainan

3.6.2.4 *Activity Diagram* Menggerakan karakter



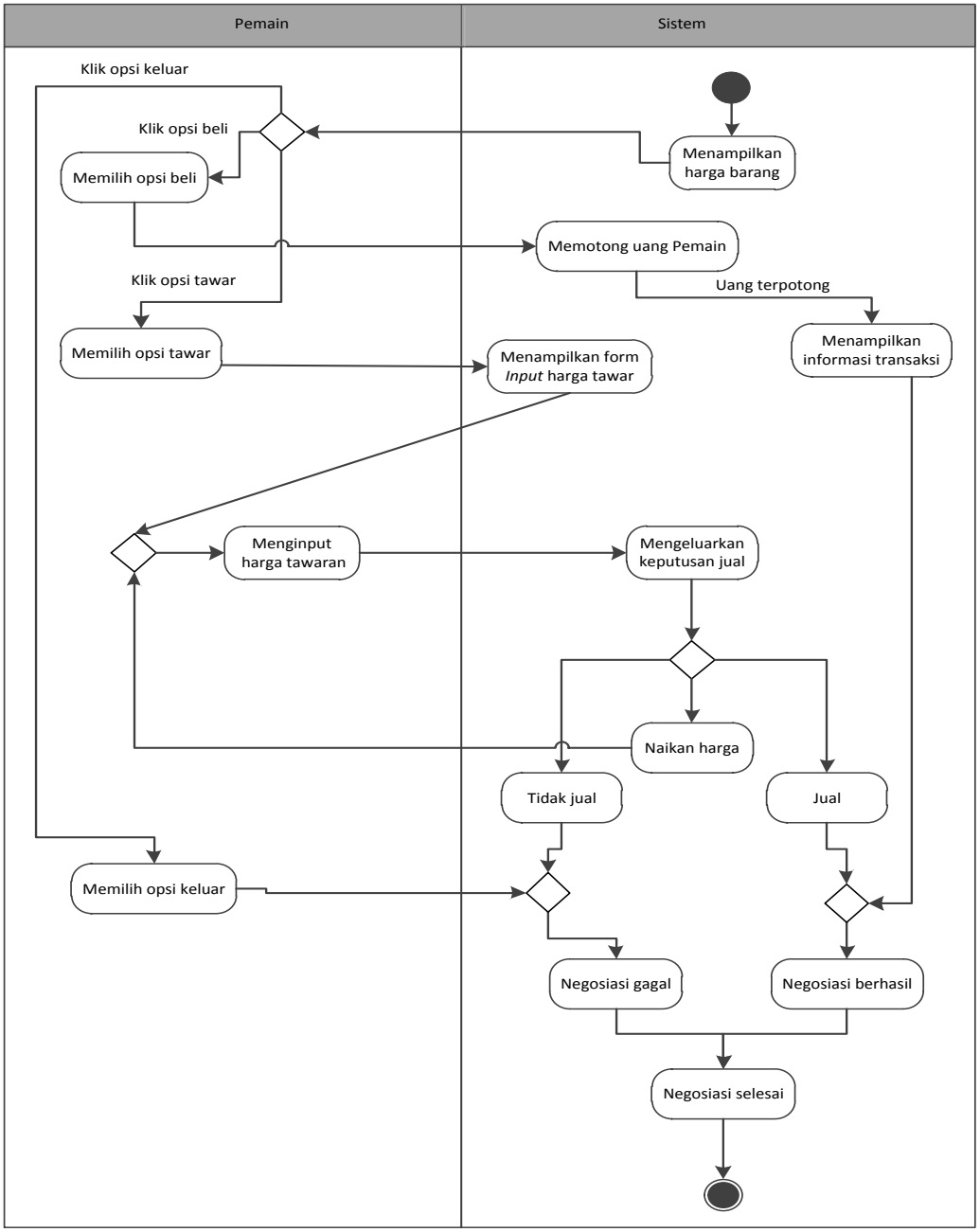
Gambar 3. 15 Activity Diagram menggerakan karakter

3.6.2.5 Activity Diagram Belanja



Gambar 3. 16 Activity Diagram belanja

3.6.2.6 *Activity Diagram* Tawar-menawar

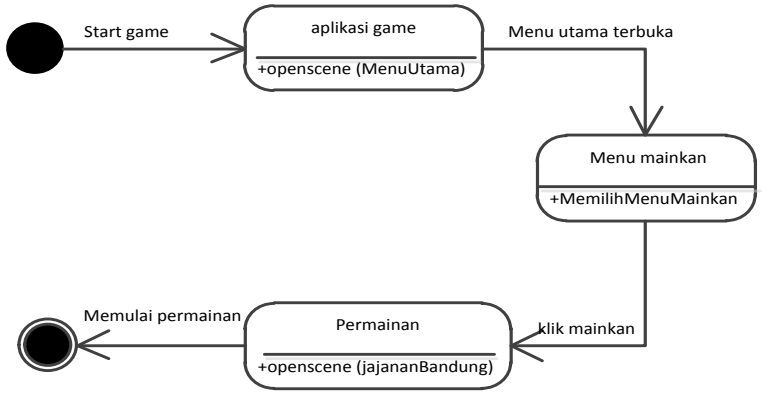


Gambar 3. 17 Activity Diagram tawar menawar

3.6.3 Statechart Diagram

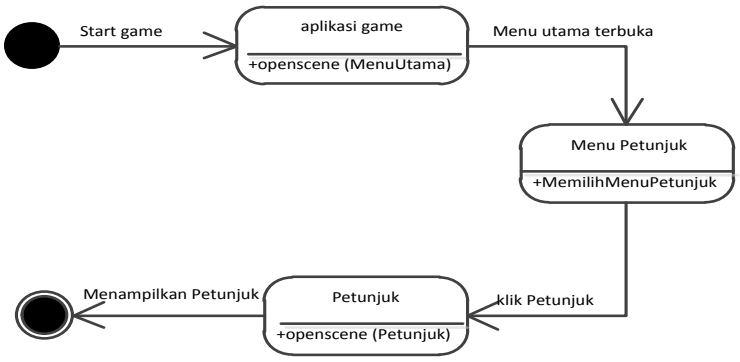
Statechart diagram digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek. Perubahan tersebut digambarkan dalam suatu grafik berarah.

3.6.3.1 *Statechart Diagram* Memulai Permainan



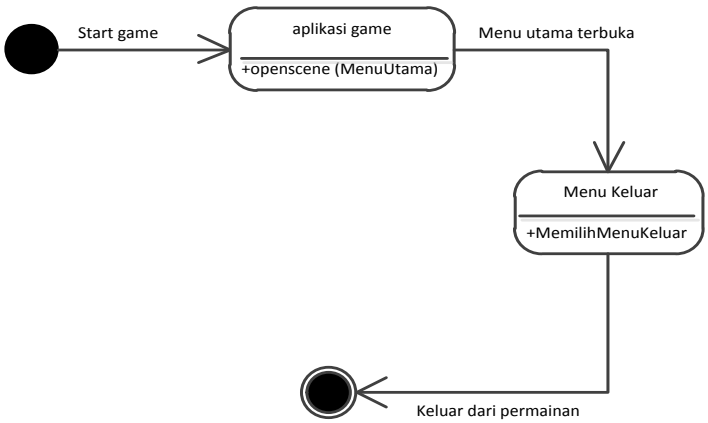
Gambar 3. 18 Statechart Diagram Memulai Permainan

3.6.3.2 Statechart Diagram Melihat Petunjuk



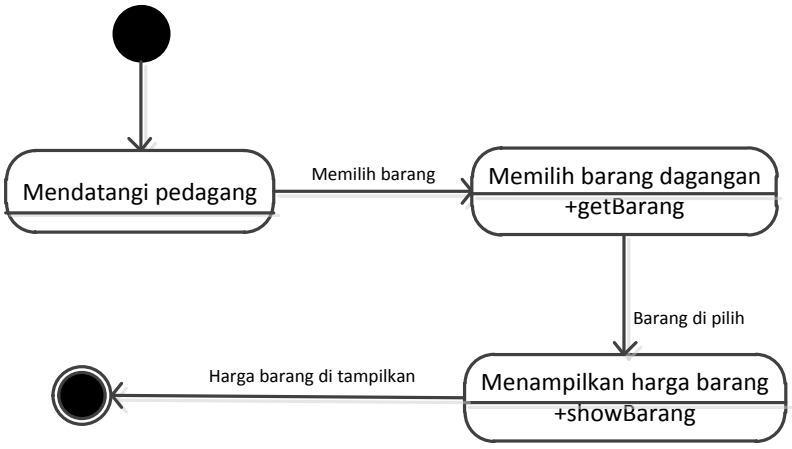
Gambar 3. 19 Statechart Diagram Melihat Petunjuk

3.6.3.3 Statechart Diagram Keluar dari Permainan



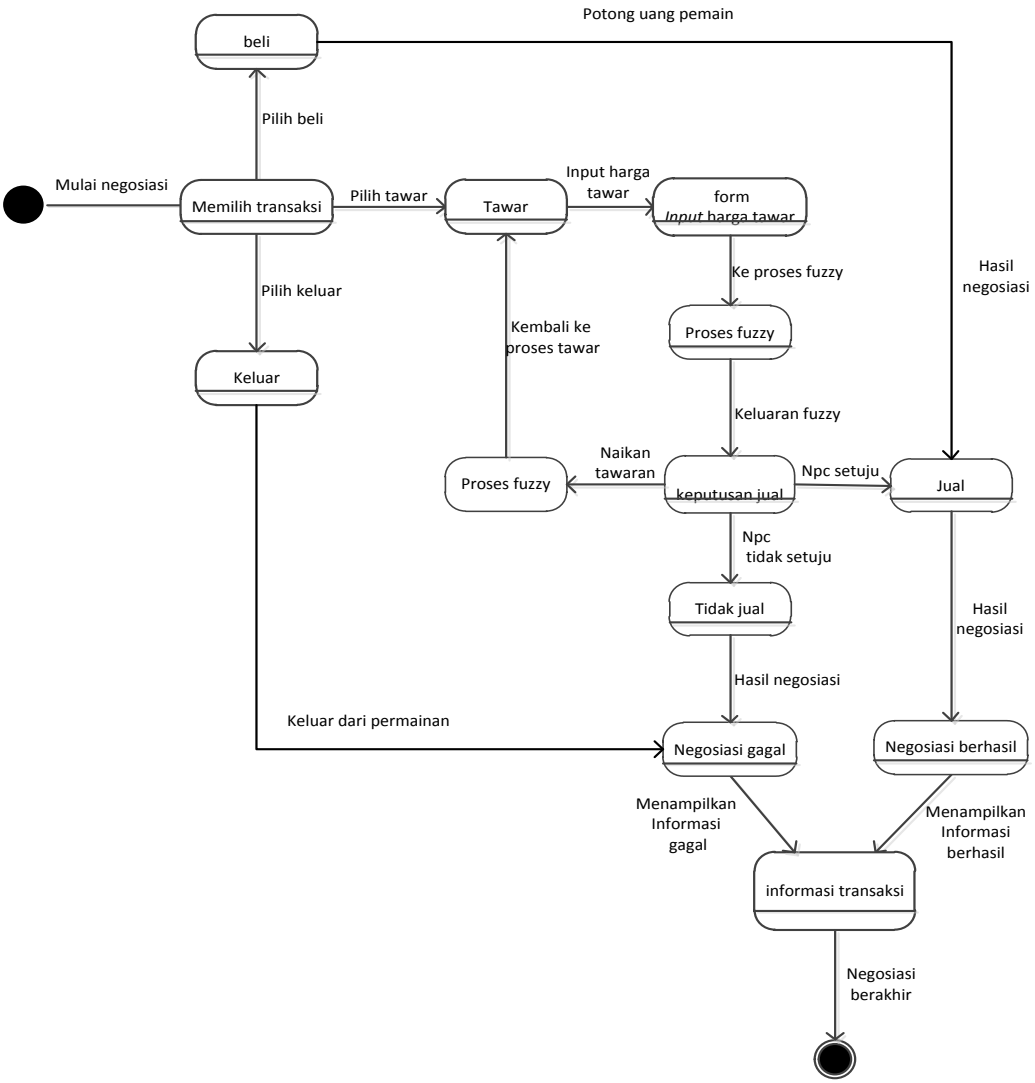
Gambar 3. 20 Statechart Diagram Keluar dari Permainan

3.6.3.4 *Statechart Diagram* Belanja



Gambar 3. 21 *Statechart Diagram* Belanja

3.6.3.5 *Statechart Diagram* Tawar-menawar

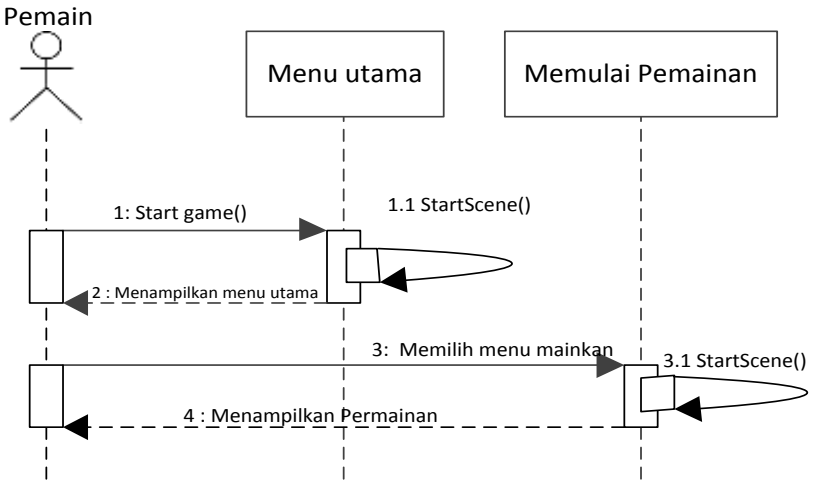


Gambar 3. 22 Statechart Diagram Tawar-Menawar

3.6.4 Sequence Diagram

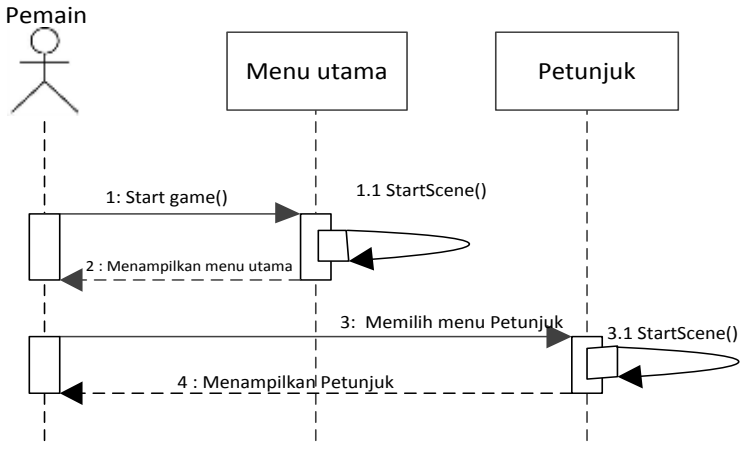
Sequence Diagram merupakan penggambaran dari interaksi antar objek di dalam dan sekitar sistem (termasuk pengguna, display, dan sebagainya). *Sequence Diagram* terdiri atas dimensi vertical (waktu) dan dimensi horizontal (objek-objek yang terkait)

3.6.4.1 Sequence Diagram Memulai Permainan



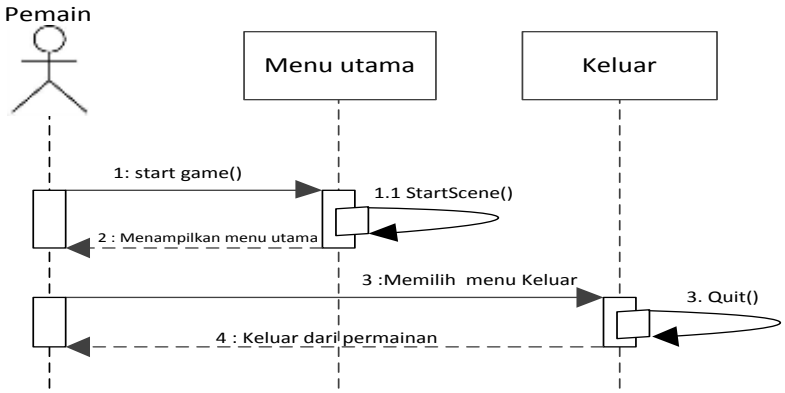
Gambar 3. 23 Sequence Diagram Permainan

3.6.4.2 *Sequence Diagram* Melihat Petunjuk



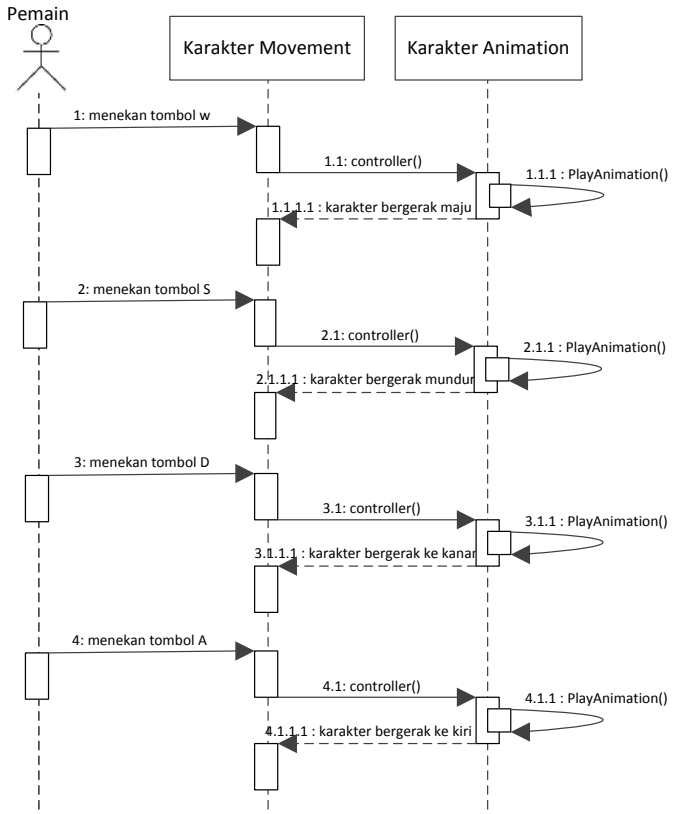
Gambar 3. 24 Sequence Diagram Melihat petunjuk

3.6.4.3 *Sequence Diagram* Keluar dari permainan



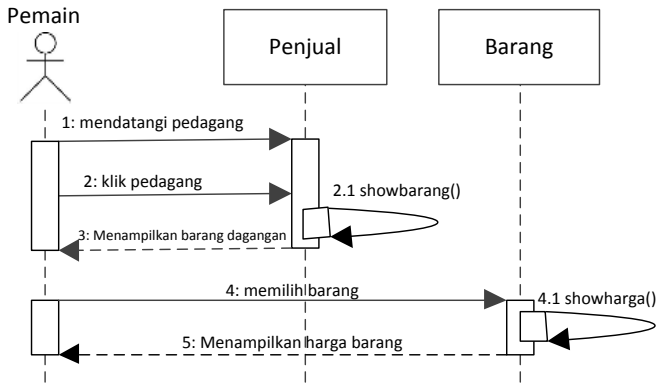
Gambar 3. 25 Sequence Diagram keluar dari permainan

3.6.4.4 *Sequence Diagram* Menggerakan karakter



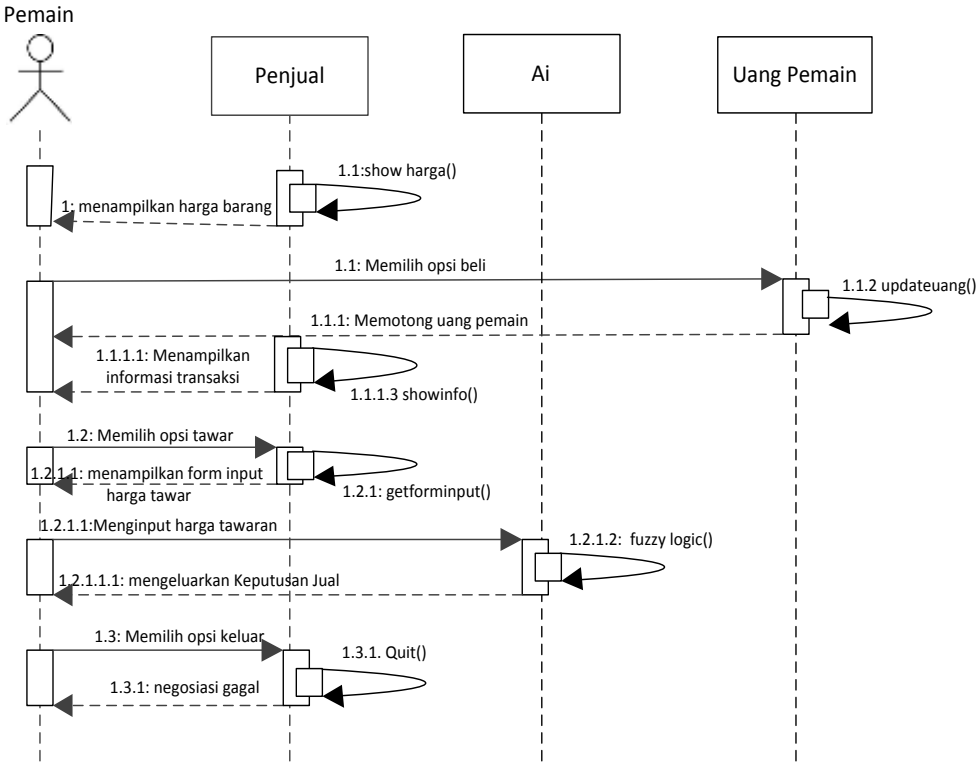
Gambar 3. 26 Sequence Diagram menggerakan karakter

3.6.4.5 Sequence Diagram belanja



Gambar 3. 27 Sequence Diagram Berbelanja

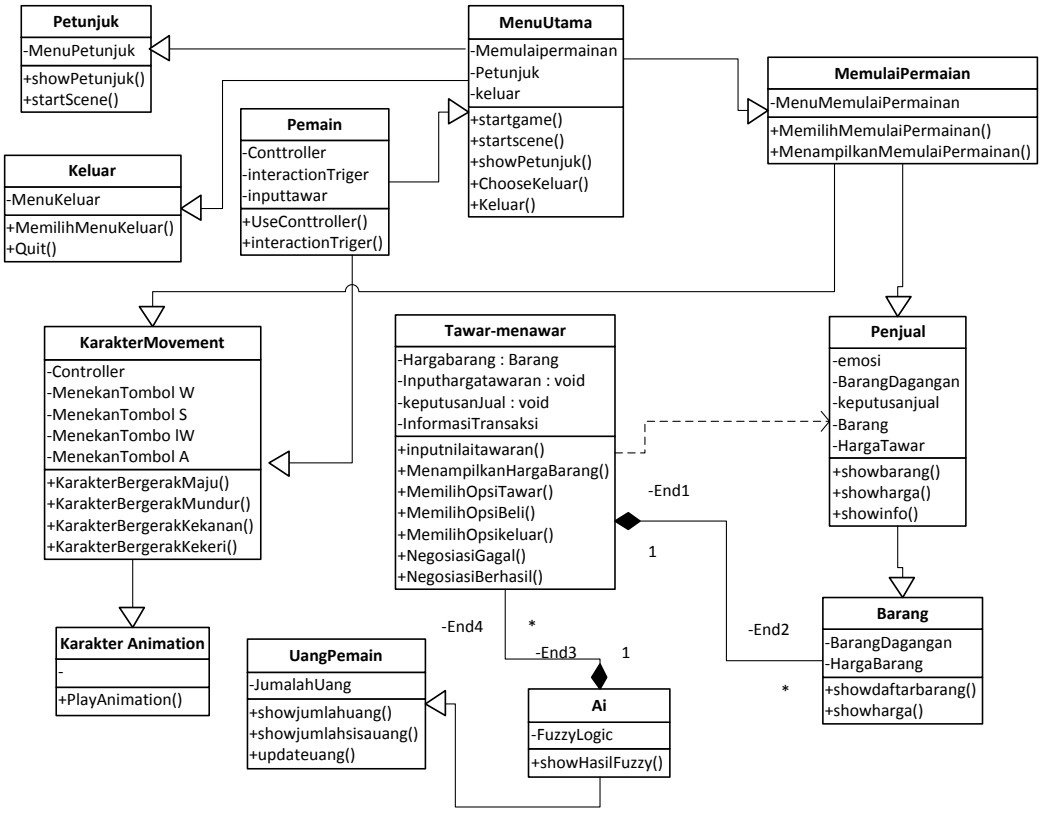
3.6.4.6 *Sequence Diagram* Tawar-Menawar



Gambar 3. 28 Sequence Diagram Tawar-Menawar

3.6.5 Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak dapat membuat kelas-kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas. *Class diagram* dari aplikasi *Game* yang akan dibangun dapat dilihat pada Gambar 3.29



Gambar 3. 29 Class Diagram