# BAB 3
# ANALISIS DAN KEBUTUHAN ALGORITMA

## 3.1 Analisis Masalah

Selama ini representasi kepuasan biasanya menggunakan logika fuzzy model mamdani sebagaimana dijelaskan dalam jurnal yang berjudul "Analisis Tingkat Kepuasan Konsumen Berdasarkan Pelayanan, Harga dan Kualitas Makanan Menggunakan *Fuzzy* Mamdani" yang dilakukan oleh nuraida, iryanto, djakaria sebayang dari Department of Mathematics, Faculty of Mathematics and Natural Sciences, *University of Sumatera Utara*, Medan adalah untuk menentukan domain himpunan *fuzzy* yang dilakukan untuk menentukan skor dari setiap kriteria yang diberikan koresponden melalui kuesioner.

Berdasarkan hal tersebut *game* yang akan dikembangkan ini menggunakan logika fuzzy model sugeno bergenre Time Management, dimana pemain yang mengatur jalannya permainan sehingga keinginan atau target tercapai. Pada *game* ini pemain harus mengumpulkan ekspresi wajah pelanggan yang puas, dengan cara melayani pelanggan dari mulai menempatkan pelanggan pada kursi kosong, mengantarkan makanan dan minuman secepat mungkin sehingga tidak melewati batas waktu yang ditentukan. Ketepatan antara *input* dan *output* sangat penting sehingga dapat merepresentasikan kepuasan pelanggan dengan benar.

## 3.2 Analisis *Game*

*Game* yang akan dibangun ini akan merepresentasikan kepuasan pelanggan terhadap suatu pelayanan. Logika fuzzy akan diterapkan pada pengunjung yang akan memakan bakso. Pelanggan yang datang akan membentuk sebuah antrian, jika terjadi antrian pemain harus mengurai antrian tersebut agar kepuasan pelanggan dapat terpenuhi.

### 3.2.1. *Game*play

Berikut adalah *game*play pada *game* yang akan dibangun :

Permainan ini berlatar belakang pedagang bakso kaki lima di pinggir jalan, pelanggan yang datang akan dilayani sebaik mungkin untuk mendapatkan kepuasan pelanggan. Pemain akan menempatkan pelanggan yang datang di kursi-kursi yang tersedia. Setelah pelanggan memesan pemain harus mengantarkan pesanan pelanggan dengan tepat kepada pelanggan yang memesan. Disini pemain dituntut untuk bergerak cepat karena jika tidak cepat akan terjadi antrian yang akan mempengaruhi *input* nilai untuk logika fuzzy

## 3.3 Analisis Masukan

Parameter yang digunakan dalam *game* ini adalah :

1. makanan (ma) → Cepat, Sedang, Lambat
2. minuman (mi) → Cepat, Sedang, Lambat
3. kursi (ku) →Cepat, Sedang, Lambat

Keterangan :

1. Parameter – parameter tersebut berdasarkan durasi waktu yang dipakai.
2. Berikut ini adalah nilai interval yang digunakan :
   a. Interval waktu untuk mengantarkan makanan 0 sampai 10 menit, dibagi ke dalam tiga nilai linguistik.
      a) Cepat , 0 – 4 menit.
      b) Sedang, 3 – 7 menit.
      c) Lambat, 6 – 10 menit.

      Keterangan : Interval waktu secara linguistik untuk makan (ma) dapat dilihat pada gambar 3.1

   b. Interval waktu untuk mengantarkan minuman 0 sampai 10 menit, dibagi ke dalam tiga nilai linguistik.

   a) Cepat , 0 – 3 menit.
   b) Sedang, 2 – 6 menit.
   c) Lambat, 5 – 10 menit.

Keterangan : Interval waktu secara linguistik untuk minum (m) dapat dilihat pada gambar 3.2

c. Interval waktu untuk mendapatkan kursi 0 sampai 10 menit, dibagi ke dalam tiga nilai linguistik.
   a) Cepat , 0 – 5 menit.
   b) Sedang, 4 – 8 menit.
   c) Lambat, 7 – 10 menit.

   Keterangan : Interval waktu secara linguistik untuk kursi (ku) dapat dilihat pada gambar 3.3

3. Waktu yang digunakan bukanlah waktu yang sebenarnya, 1 menit dalam *game* ini sama dengan 10 detik dalam waktu yang sebenarnya.

## 3.4 Analisis Metode / Algoritma

Kecerdasan buatan yang akan diterapkan dalam *game* ini adalah logika fuzzy. Algoritma ini akan diterapkan kepada pelanggan-pelanggan yang datang dan dari pelanggan-pelanggan tersebutlah akan didapatkan *input* untuk logika fuzzy untuk merepresentasikan kepuasan pelanggan.

**Contoh kasus :**

Seorang mahasiswi datang dengan teman-temannya untuk makan bakso untuk makan siang. Kondisi warung bakso hari itu sedang tidak terlalu penuh dan setelah dia menunggu selama 3,9 menit. Setelah memesan, bakso tersebut diantar pada menit ke 4,7 setelah dia memesan dan pada menit ke 5,5 minuman yang dipesan pun datang. Konversikan nilai crips tersebut ke dalam nilai fuzzy.

### A. Fuzzyfication

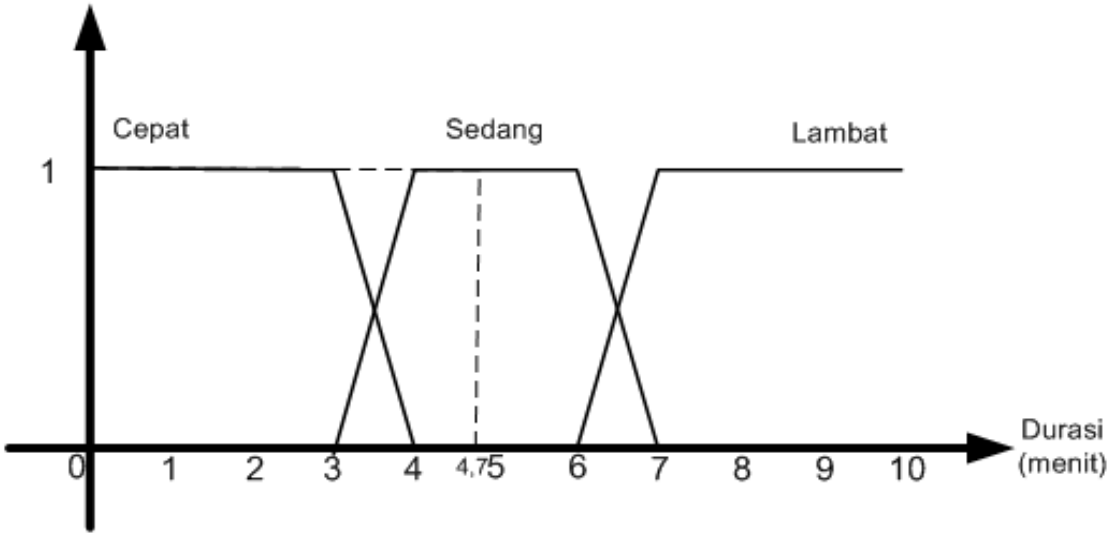
1. Didapatkan waktu untuk mengantarkan makanan selama 4,7menit.

   Pada proses ini maka nilai crips *input* ma 4,7 menit dikonversi ke nilai fuzzy dengan cara :

   Ma 4,7 menit berada pada nilai linguistik sedang

   Dengan rumus = 1, $b \leq x \leq c$

Derajat keanggotaan ma =1

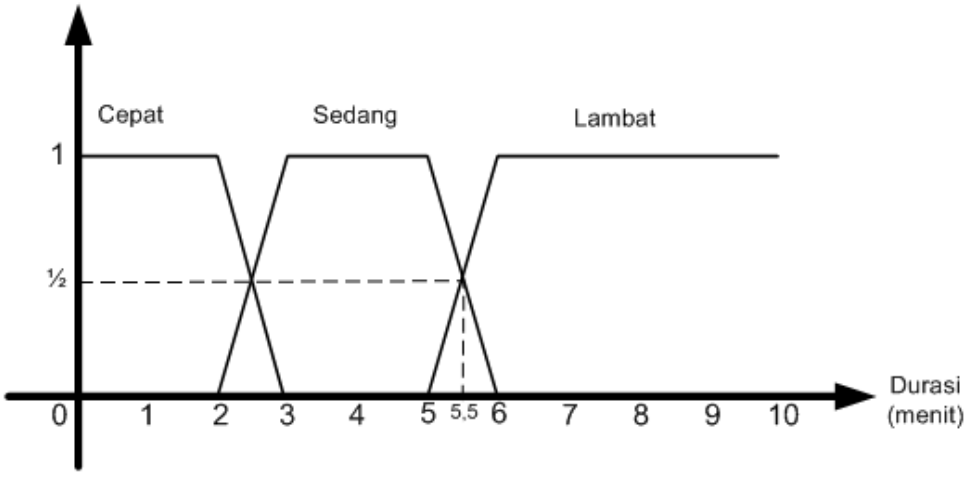



**Gambar 3. 1 Fungsi keanggotaan fuzzy (ma)**

2. Didapatkan waktu untuk mengantarkan minuman selama 5,5 menit
   Pada proses ini maka nilai crisp *input* mi 5,5 menit dikonversikan ke nilai fuzzy dengan cara :
   mi pada 5,5 menit berada pada nilai linguistik sedang dan lambat

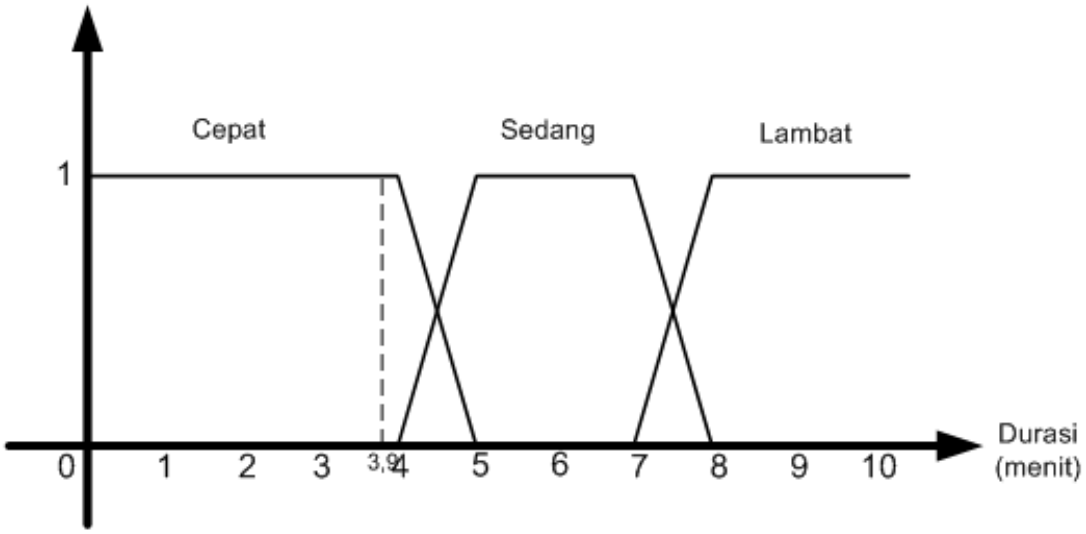
   Derajat keanggotaan untuk sedang dihitung menggunakan rumus =
   -(x-d)/(d-c), $c \leq x \leq d$, dimana c=5 dan d=6
   sehingga derajat keanggotaan sedang : -(5,5 – 6)/(6 – 5) = 0,5/1 = ½

   Derajat keanggotaan untuk lambat dihitung menggunakan rumus =
   (x – a)/(b – a),$a \leq x \leq b$ dimana a = 5 dan b = 6 sehingga derajat keanggotaan lambat : (5,5 – 5)/(6 – 5) = 0,5/1 = ½




**Gambar 3. 2 Fungsi keanggotaan fuzzy (mi)**

3. Didapatkan waktu untuk pelanggan mendapatkan kursi 3,9 menit
   Pada proses ini maka nilai crips *input* ku 3,9 menit dikonversikan ke nilai fuzzy dengan cara :
   ku 3,9 menit berada pada nilai linguistik cepat
   Dengan rumus = 1, $b \leq x \leq c$




**Gambar 3. 3 Fungsi keanggotaan fuzzy (ku)**

Kesimpulan :

Pada Fuzzyfication menghasilkan 4 *input* :

Ma (makanan) = sedang (1)

Mi (minuman) = sedang (½) dan lambat (½)

Ku (kursi) = cepat (1)

## B. Inference Model Sugeno

Model sugeno menggunakan fungsi keanggotaan singleton, yaitu fungsi keanggotaan yang memiliki derajat keanggotaan 1 pada suatu nilai crips tunggal dan 0 pada semua crips yang lain.

Rule untuk karakter dapat dilihat pada tabel 3.1

**Tabel 3. 1 Rule untuk karakter**

| Rule | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IF | ma | IS | lambat | AND | Mi | IS | lambat | AND | ku | IS | Lambat | THEN | hasil | IS | Tidak Puas |
| IF | ma | IS | lambat | AND | Mi | IS | lambat | AND | ku | IS | Sedang | THEN | hasil | IS | Tidak Puas |
| IF | ma | IS | lambat | AND | Mi | IS | lambat | AND | ku | IS | Cepat | THEN | hasil | IS | Tidak Puas |
| IF | ma | IS | lambat | AND | Mi | IS | sedang | AND | ku | IS | Lambat | THEN | hasil | IS | Tidak Puas |
| IF | ma | IS | lambat | AND | Mi | IS | sedang | AND | ku | IS | Sedang | THEN | hasil | IS | Cukup Puas |
| IF | ma | IS | lambat | AND | Mi | IS | sedang | AND | ku | IS | Cepat | THEN | hasil | IS | Puas |
| IF | ma | IS | lambat | AND | Mi | IS | cepat | AND | ku | IS | Lambat | THEN | hasil | IS | Tidak Puas |
| IF | ma | IS | lambat | AND | Mi | IS | cepat | AND | ku | IS | Sedang | THEN | hasil | IS | Cukup Puas |
| IF | ma | IS | lambat | AND | Mi | IS | cepat | AND | ku | IS | Cepat | THEN | hasil | IS | Puas |
| IF | ma | IS | sedang | AND | Mi | IS | lambat | AND | ku | IS | Lambat | THEN | hasil | IS | Tidak Puas |
| IF | ma | IS | sedang | AND | Mi | IS | lambat | AND | ku | IS | Sedang | THEN | hasil | IS | Cukup Puas |
| IF | ma | IS | sedang | AND | Mi | IS | lambat | AND | ku | IS | Cepat | THEN | hasil | IS | Puas |
| IF | ma | IS | sedang | AND | Mi | IS | sedang | AND | ku | IS | Lambat | THEN | hasil | IS | Cukup Puas |
| IF | ma | IS | sedang | AND | Mi | IS | sedang | AND | ku | IS | Sedang | THEN | hasil | IS | Puas |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IF | ma | IS | sedang | AND | Mi | IS | sedang | AND | ku | IS | Cepat | THEN | hasil | IS | Puas |
| IF | ma | IS | sedang | AND | Mi | IS | cepat | AND | ku | IS | Lambat | THEN | hasil | IS | Puas |
| IF | ma | IS | sedang | AND | Mi | IS | cepat | AND | ku | IS | Sedang | THEN | hasil | IS | Puas |
| IF | ma | IS | sedang | AND | Mi | IS | cepat | AND | ku | IS | Cepat | THEN | hasil | IS | Puas |
| IF | ma | IS | cepat | AND | Mi | IS | lambat | AND | ku | IS | Lambat | THEN | hasil | IS | Cukup Puas |
| IF | ma | IS | cepat | AND | Mi | IS | lambat | AND | ku | IS | Sedang | THEN | hasil | IS | Puas |
| IF | ma | IS | cepat | AND | Mi | IS | lambat | AND | ku | IS | Cepat | THEN | hasil | IS | Puas |
| IF | ma | IS | cepat | AND | Mi | IS | sedang | AND | ku | IS | Lambat | THEN | hasil | IS | Puas |
| IF | ma | IS | cepat | AND | Mi | IS | sedang | AND | ku | IS | Sedang | THEN | hasil | IS | Puas |
| IF | ma | IS | cepat | AND | Mi | IS | sedang | AND | ku | IS | Cepat | THEN | hasil | IS | Puas |
| IF | ma | IS | cepat | AND | Mi | IS | cepat | AND | ku | IS | Lambat | THEN | hasil | IS | Puas |
| IF | ma | IS | cepat | AND | Mi | IS | cepat | AND | ku | IS | Sedang | THEN | hasil | IS | Puas |
| IF | ma | IS | cepat | AND | Mi | is | cepat | AND | ku | IS | Cepat | THEN | hasil | IS | Puas |

Dari 4 fuzzy *input* maka di dapat dua aturan dari 27 aturan yang dapat diaplikasikan

IF ma IS sedang AND mi IS sedang AND ku IS cepat THEN hasil IS puas

IF ma IS sedang AND mi IS lambat AND ku IS cepat THEN hasil IS puas

Menggunakan aturan Conjuction ( ^ ) dengan memilih derajat keanggotaan minimum sehingga diperoleh

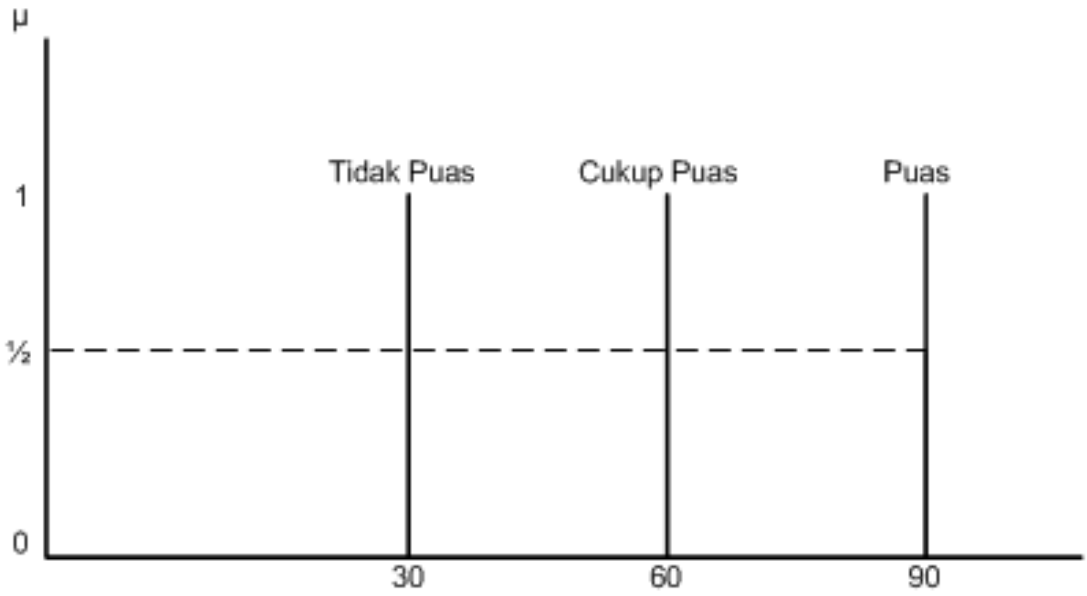
IF ma IS sedang(1) AND mi IS sedang (½) AND ku IS cepat(1) THEN hasil IS puas (½)

IF ma IS sedang(1) AND mi IS lambat (½) AND ku IS cepat(1) THEN hasil IS puas (½)

Menggunakan aturan Disjunction (v) dengan memilih derajat keanggotaan maksimum.

Sehingga dihasilkan hasil is puas (½) v hasil is puas (½) .

Dihasilkan hasil is puas (½)




**Gambar 3. 4 Inference model sugeno**

## C. Defuzzyfication Model Sugeno

Hasil is puas sudah menghasilkan satu fuzzy set tunggal. Dengan menggunakan height method untuk proses defuzzyfication hasil is puas (½)

Didapatkan nilai crips puas adalah 90 maka proses defuzzyfication menghasilkan nilai crips 90.

$$y* = \sum \frac{\mu(y)y}{\mu(y)},$$

$$Y = \frac{½\,(90) + ½\,(90)}{½ + ½}$$

Kesimpulan

Nilai yang didapat dari proses defuzzyfication tersebut selanjutnya digunakan untuk penentuan keputusan dengan aturan sebagai berikut

Tidak Puas 0 – 30

Cukup Puas 31 – 60

Puas 61 – 90

Hasil nilai dari hasil = 90 adalah Puas

## 3.5 Analisis kebutuhan non fungsional

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

### 3.5.1 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan dalam sebuah sistem merupakan perintah-perintah yang diberikan kepada perangkat keras agar bisa saling berinteraksi diantara keduanya.

1. **Pengembang**

   Spesifikasi perangkat lunak yang digunakan pengembang untuk membangun *game* ini dapat dilihat pada tabel 3.2

**Tabel 3. 2 Spesifikasi perangkat lunak pengembang**

| Nama Perangkat Lunak | Spesifikasi |
|---|---|
| Sistem Operasi | Microsoft Windows 7 SP1 |
| Tools | Unity<br>Inkscape |

**2. Pengguna**

Spesifikasi minimum perangkat lunak yang dibutuhkan pengguna untuk memainkan *game* ini dapat dilihat dari tabel 3.3

**Tabel 3. 3 Spesifikasi minimum perangkat lunak pengguna**

| Nama Perangkat | Spesifikasi |
|---|---|
| *Processor* | 1.80 GHz |
| *RAM* | 2 GB |
| *VGA Card* | 250 MB |
| *Harddisk* | Tersedia 100 MB |
| Monitor | Resolusi 800 x 600 |
| *Mouse* dan *keyboard* | Standar |

### 3.5.2 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.

a) Pengembang

Spesifikasi perangkat keras yang digunakan pengembang untuk membangun *game* ini dapat dilihat pada Tabel 3.4

**Tabel 3. 4 Spesifikasi perangkat keras pengembang**

| Nama Perangkat | Spesifikasi |
|---|---|
| Processor | Intel Core 2 duo |
| RAM | 2 GB |
| VGA Card | 250 MB |
| Harddisk | 200 GB |
| Monitor | Resolusi |
| Mouse dan Keyboard | Standar |

### 3.5.3 Analisis Pengguna

Pada analisis pengguna ini akan mencakup analisis beberapa parameter terhadap calon pengguna dari aplikasi.

1. *User Knowledge and Experience* dari target pengguna yang akan menggunakan *game*

*Game* yang akan dibangun ini bisa digunakan oleh kalangan apapun, tetapi pengetahuan dan pengalaman akan memudahkan pengguna dalam penggunaannya. Klasifikasi *knowledge and experience* dari aplikasi dapat di lihat pada tabel 3.5

**Tabel 3. 5 Analisis klasifikasi knowledge and experience**

| | |
|---|---|
| Educational Level | *Game* ini bisa digunakan oleh berbagai kalangan seperti pelajar, mahasiswa hingga masyarakat awam |
| Reading Level | *Game* ini bisa digunakan oleh berbagai level pendidikan dengan *reading* level rendah |
| Typing Skills | *Game* ini tidak memerlukan *typing skills* tinggi |
| Computer Literacy | *Game* ini bisa digunakan oleh pengguna yang memiliki kemampuan komputer yang rendah |
| Task Experience | *Game* ini bisa digunakan oleh dengan pengalaman *game* yang rendah |
| System Experience | *Game* ini bisa digunakan oleh dengan pengalaman computer dan *game* yang rendah |
| Application Experience | *Game* ini bisa digunakan dalam semua system operasi Windows minimal XP |
| Native Language | *Game* ini menggunakan bahasa Indonesia |
| Use of Other System | *Game* ini bisa dijalankan tanpa perlu install aplikasi lain |

2. *User Physical Characteristic*

Keadaan fisik seseorang mungkin akan berpengaruh pada penggunaan aplikasi *game* ini. Ada hal-hal yang harus diperhatikan juga terhadap pengguna dari

karakteristik fisiknya untuk dapat menggunakan aplikasi ini yaitu, *color blind, handedness*, dan *gender* dapat dilihat pada tabel 3.6

**Tabel 3. 6 Analisis users physical characteristik**

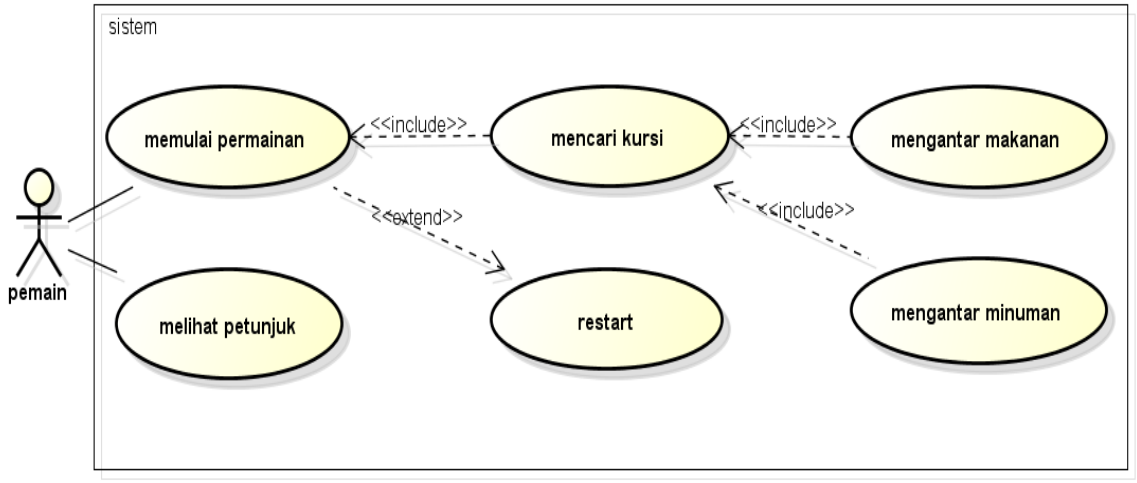
| | |
|---|---|
| Age | 10 tahunkeatas |
| Gender | Pria dan wanita |
| Disabilities | Pengguna yang buta tidak akan bias menggunakan *game* ini, untuk yang penglihatannya kurang dan tuli masih dapat memainkan tetapi tidak akan optimal. |

## 3.6 Analisis kebutuhan fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik sesuai kebutuhan. Analisis kebutuhan fungsional dimodelkan dengan menggunakan UML *(Unified Modeling Language).*

### 3.6.1 Use Case Diagram

Diagram *use case* menggambarkan hubungan antara aktor dan sistem yang dibangun. Perancangan proses-proses yang terdapat pada *game* yang akan dibangun akan digambarkan dengan diagram *use case.*




**Gambar 3. 5 use case diagram *game* tukang bakso**

1. Definisi Aktor

Definisi aktor berfungsi untuk menjelaskan aktor yang terdapat pada diagram *use case.* Definisi aktor diterangkan pada Tabel 3.7

**Tabel 3. 7 Definisi Aktor**

| No. | Aktor | Deskripsi |
|---|---|---|
| 1. | Pemain | Pengguna atau orang yang memainkan *game*. |

2. Definisi *Use Case*

Definisi *use case* berfungsi untuk menjelaskan fungsi *use case* yang terdapat pada diagram *use case*. Definisi *use case* diterangkan pada Tabel 3.8

**Tabel 3. 8 Definisi *Use Case***

| No. | *Use Case* | Deskripsi |
|---|---|---|
| 1. | Memulai permainan | Proses untuk memulai permainan atau *game*. |
| 2. | Melihat petunjuk | Proses untuk menampilkan petunjuk atau panduan cara bermain. |
| 3. | Mencari kursi | Proses penempatan pelanggan pada kursi yang kosong |
| 4. | Mengantar makanan | Proses mengantarkan makanan pada pelanggan |
| 5. | Mengantar minuman | Proses mengantarkan minuman pada pelanggan |
| 6. | *Restart* | Proses untuk mengulang kembali permainan |

### 3.6.2 Skenario *Use Case*

Skenario setiap bagian pada *use case* menunjukkan proses apa yang terjadi pada setiap bagian didalam *use case* tersebut, dimana pengguna memberikan perintah pada setiap bagian dan respon apa yang diberikan oleh sistem kepada pengguna setelah pengguna memberikan perintah pada setiap bagian - bagian *use case*.

Tabel 3.9 dibawah ini adalah skenario *Use Case* mulai main pada *game* tukang bakso

**Tabel 3. 9 Skenario Use Case Memulai permainan**

| **Identifikasi** | |
|---|---|
| **Nomor** | UC-01 |
| **Nama** | Memulai Permainan |
| **Tujuan** | Menjalankan menu main |
| **Deskripsi** | Proses untuk mulai main |
| **Aktor** | Pemain |
| **Skenario Utama** | |
| **Kondisi awal** | Pemain masuk ke aplikasi |
| **Aksi aktor** | **Reaksi Sistem** |
| 4. Masuk ke aplikasi | |
| | 5. Menampilkan menu utama |
| 6. Memilih menu Main | |
| | 7. Menampilkan suasana tempat makan |
| **Kondisi akhir** | Aplikasi menampilkan suasana tempat makan |

Tabel 3.10 dibawah ini adalah skenario *Use Case* melihat petunjuk pada *game* tukang bakso

**Tabel 3. 10 Skenario *use case* melihat petunjuk**

| **Identifikasi** | |
|---|---|
| **Nomor** | UC-02 |
| **Nama** | Melihat petunjuk |
| **Tujuan** | Menampilkan petunjuk permainan |
| **Deskripsi** | Proses untuk menampilkan cara bermain |
| **Aktor** | Pemain |
| **Skenario Utama** | |
| **Kondisi awal** | Pemain berada pada menu utama |

| Aksi aktor | Reaksi Sistem |
|---|---|
| 1. Masuk ke aplikasi | |
| | 2. Menampilkan menu utama |
| 3. Memilih menu petunjuk | |
| | 4. Menampilkan petunjuk permainan |
| 5. Menutup tampilan petunjuk | |
| | 6. Menampilkan menu utama |
| **Kondisi akhir** | Aplikasi menampilkan menu utama |

Tabel 3.11 dibawah ini adalah skenario *Use Case* mencari kursi pada *game* tukang bakso

**Tabel 3. 11 Skenario use case mencari kursi**

| **Identifikasi** | |
|---|---|
| **Nomor** | UC-03 |
| **Nama** | Mencari kursi |
| **Tujuan** | Menempatkan pelanggan pada kursi kosong |
| **Deskripsi** | Proses menempatkan pelanggan pada kursi kosong untuk mendapatkan crisp *input* |
| **Aktor** | Pemain |
| **Skenario Utama** | |
| **Kondisi awal** | Pemain berada pada permainan |
| **Aksi aktor** | **Reaksi Sistem** |
| | 1. Menampilkan menu utama |
| 2. Memilih menu main | |
| | 3. Menampilkan area permainan |
| 4. Drag karakter pembeli ke kursi kosong | |
| | 5. Menempatkan karakter pada |

| | |
|---|---|
| | kursi |
| **Kondisi akhir** | Karakter duduk di kursi |

Tabel 3.12 dibawah ini adalah skenario *Use Case* mengantarkan makanan pada *game* tukang bakso

**Tabel 3. 12 Skenario use case mengantarkan makanan**

| **Identifikasi** | |
|---|---|
| **Nomor** | UC-04 |
| **Nama** | Mengantarkan makanan |
| **Tujuan** | Mengantarkan makanan kepada pelanggan |
| **Deskripsi** | Proses mengantarkan makanan kepada pelanggan untuk mendapatkan crisp *input* |
| **Aktor** | Pemain |
| **Skenario Utama** | |
| **Kondisi awal** | Pemain berada pada permainan |
| **Aksi aktor** | **Reaksi Sistem** |
| 1. Memilih menu main | |
| | 2. Menampilkan layar permainan |
| 3. pilih karakter yang sedang menunggu | |
| | 4. mengambil pesanan makanan |
| 5. menunggu pesanan | |
| | 6. membuatkan pesanan makanan |
| 7. Mengantarkan makanan | |
| | 8. Karakter makan |
| **Kondisi akhir** | Karakter sedang memakan baso |

Tabel 3.13 dibawah ini adalah skenario *Use Case* mengantarkan minuman pada *game* tukang bakso

**Tabel 3. 13 Skenario use case mengantarkan minuman**

| **Identifikasi** | |
|---|---|
| **Nomor** | UC-05 |
| **Nama** | Mengantarkan minuman |
| **Tujuan** | Mengantarkan minuman kepada pelanggan |
| **Deskripsi** | Proses mengantarkan minuman kepada pelanggan untuk mendapatkan crisp *input* |
| **Aktor** | Pemain |
| **Skenario Utama** | |
| **Kondisi awal** | Pemain berada pada permainan |
| **Aksi aktor** | **Reaksi Sistem** |
| 1. Memilih menu main | |
| | 2. Menampilkan layar permainan |
| 3. pilih karakter yang sedang menunggu | |
| | 4. mengambil pesanan minuman |
| 5. menunggu pesanan | |
| | 6. Membuatkan pesanan minuman |
| 7. Mengantarkan minuman | |
| | 8. Karakter minum |
| **Kondisi akhir** | Aplikasi menampilkan pelanggan yang sedang minum |

Tabel 3.14 dibawah ini adalah skenario *Use Case* restart pada *game* tukang bakso

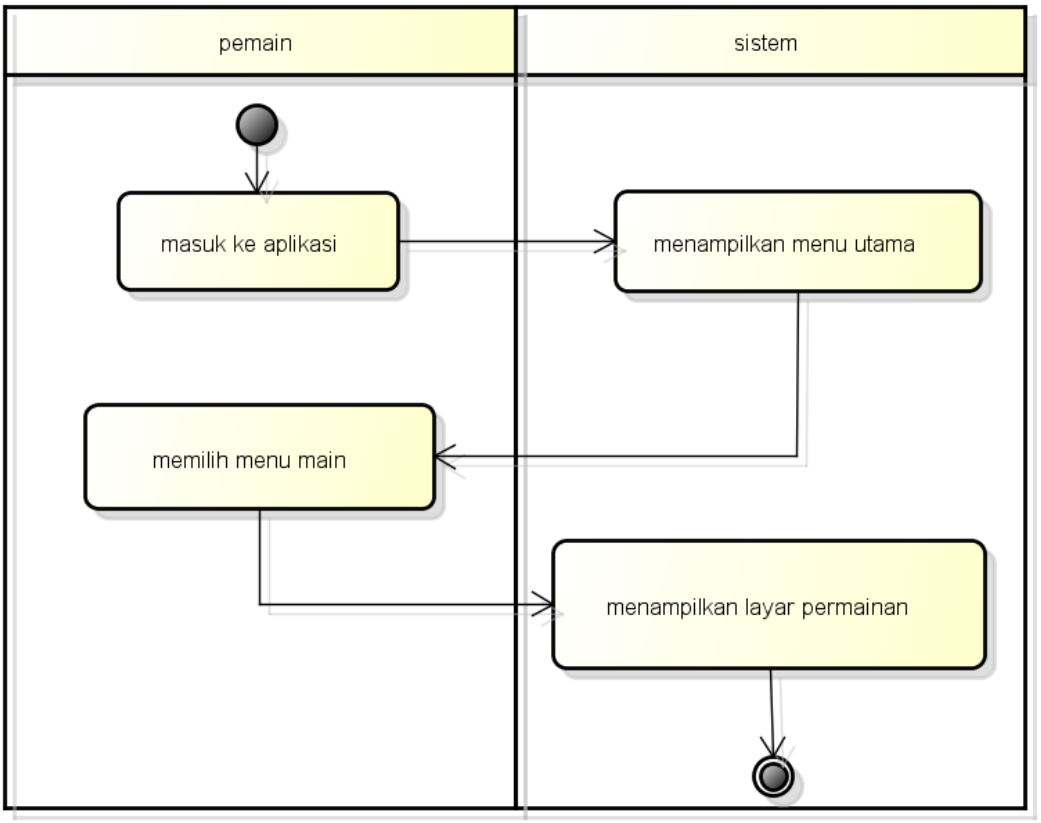
**Tabel 3. 14 Skenario Use Case restart**

| **Identifikasi** | |
|---|---|
| **Nomor** | UC-06 |
| **Nama** | Restart |
| **Tujuan** | Mengulang permainan dari awal |
| **Deskripsi** | Proses mengulang permainan |
| **Aktor** | Pemain |
| **Skenario Utama** | |
| **Kondisi awal** | Pemain berada pada permainan |
| **Aksi aktor** | **Reaksi Sistem** |
| | 1. Menampilkan layar main |
| 2. Tekan tombol pause | |
| | 3. Menampilkan validasi |
| 4. Pilih menu restart | |
| | 5. Ulangi *game* |
| **Kondisi akhir** | Aplikasi menampilkan permainan seperti pada awal permainan |

### 3.6.3 Activity Diagram

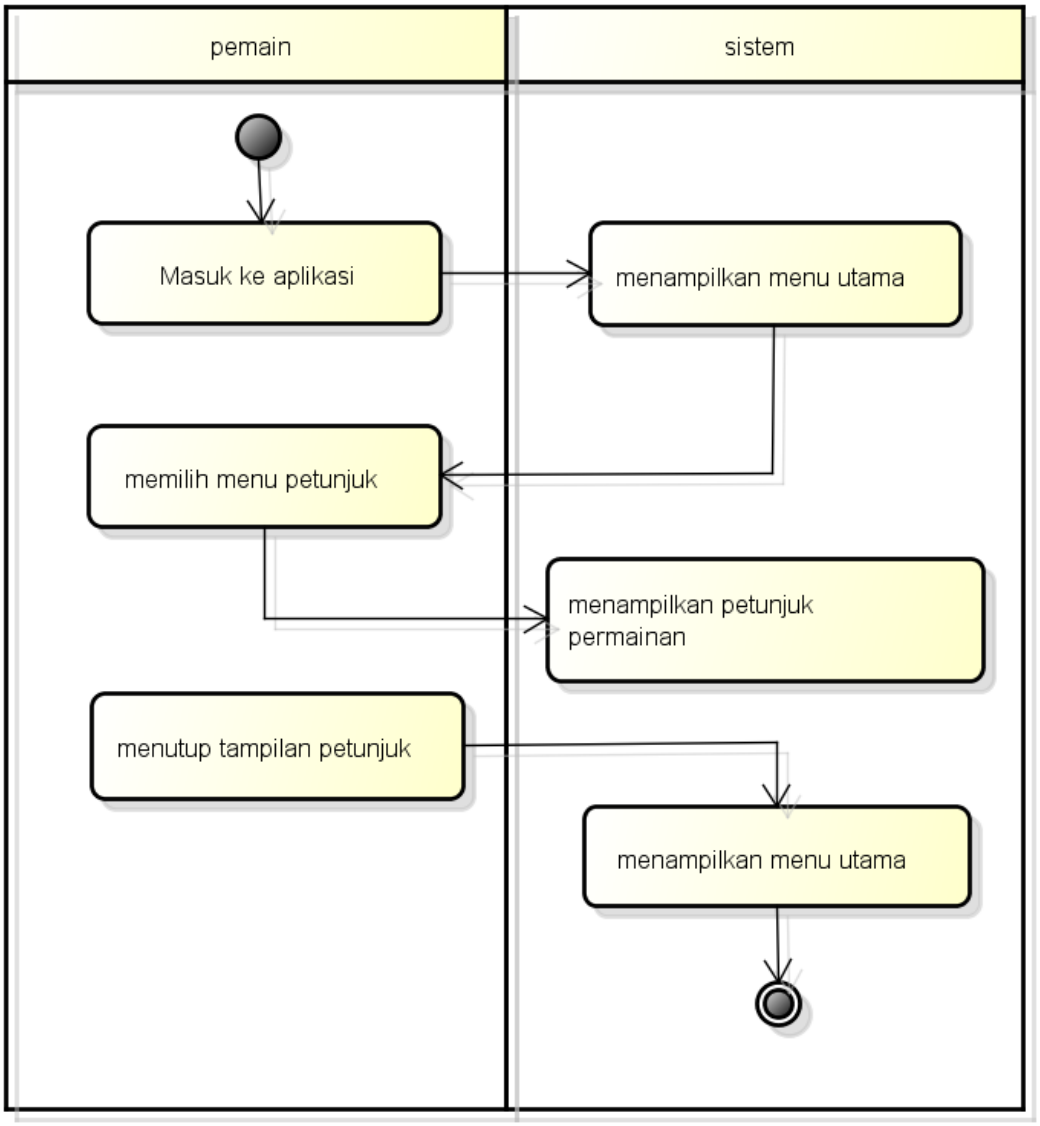
Diagram *activity* menggambarkan proses-proses yang terjadi mulai aktivitas dimulai sampai aktivitas berhenti pada sistem yang sedang dirancang. Bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir.

### 1. *Activity Diagram* Memulai Permainan



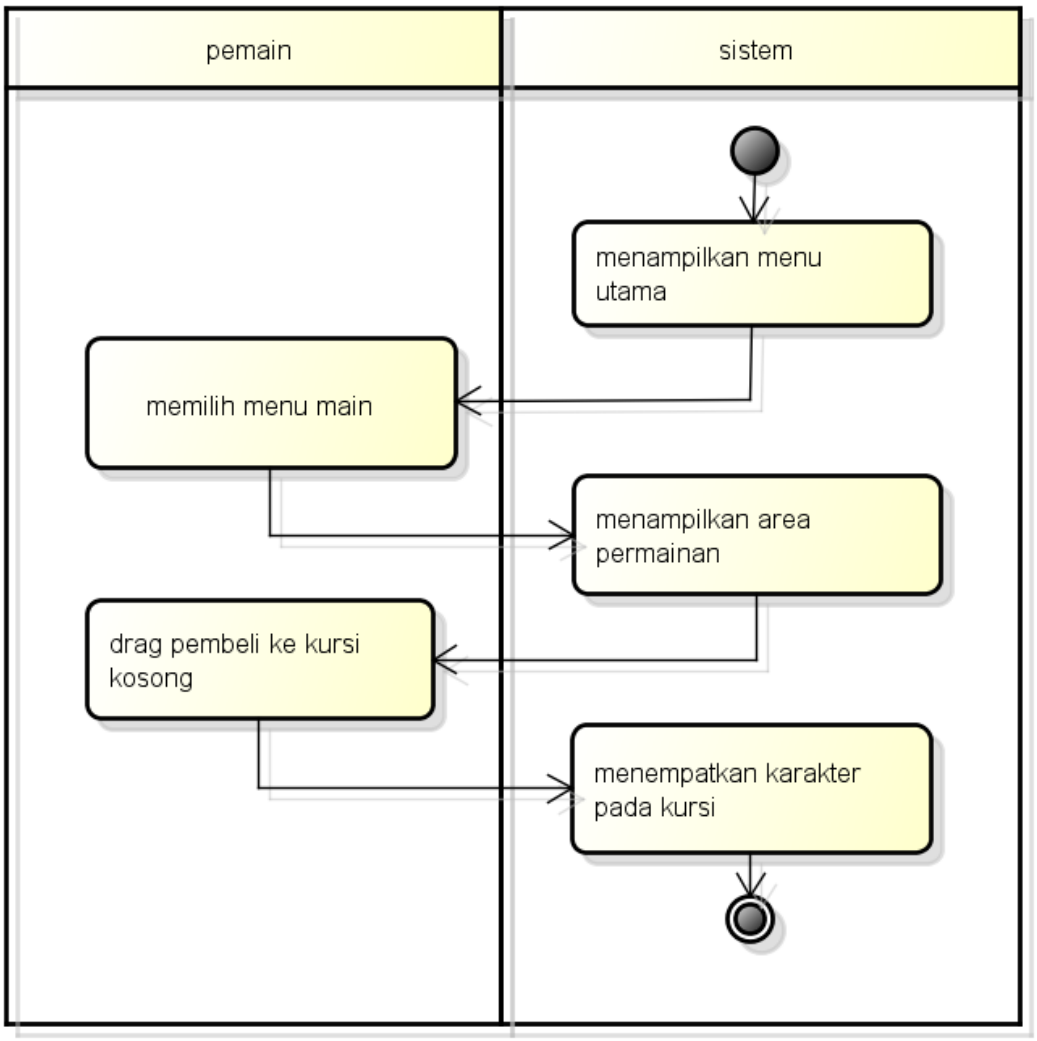

**Gambar 3. 6 Activity diagram memulai permainan**

### 2. *Activity Diagram* Melihat Petunjuk



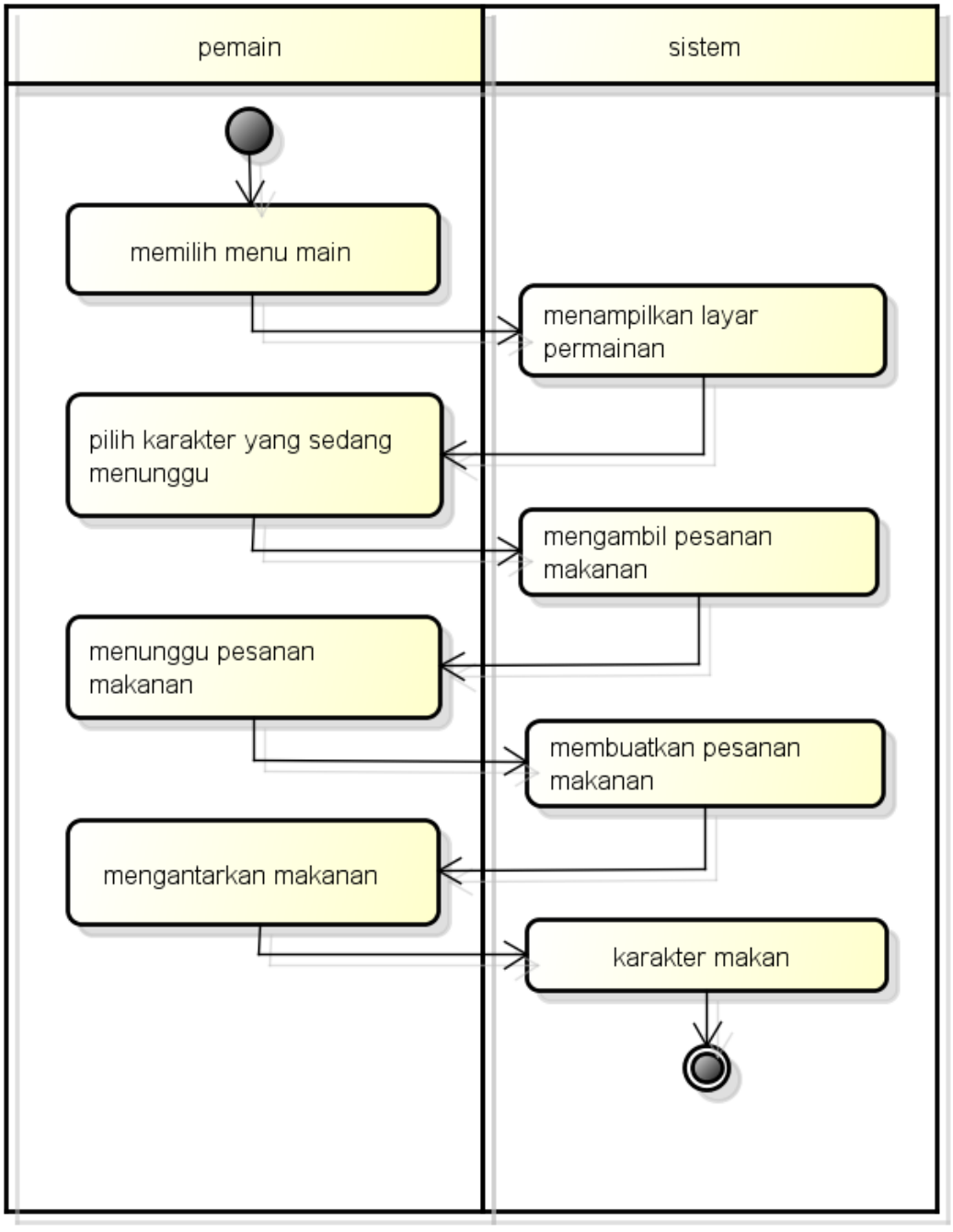

**Gambar 3. 7 Activity diagram memilih menu lihat petunjuk**

### *3. Activity Diagram mencari kursi*



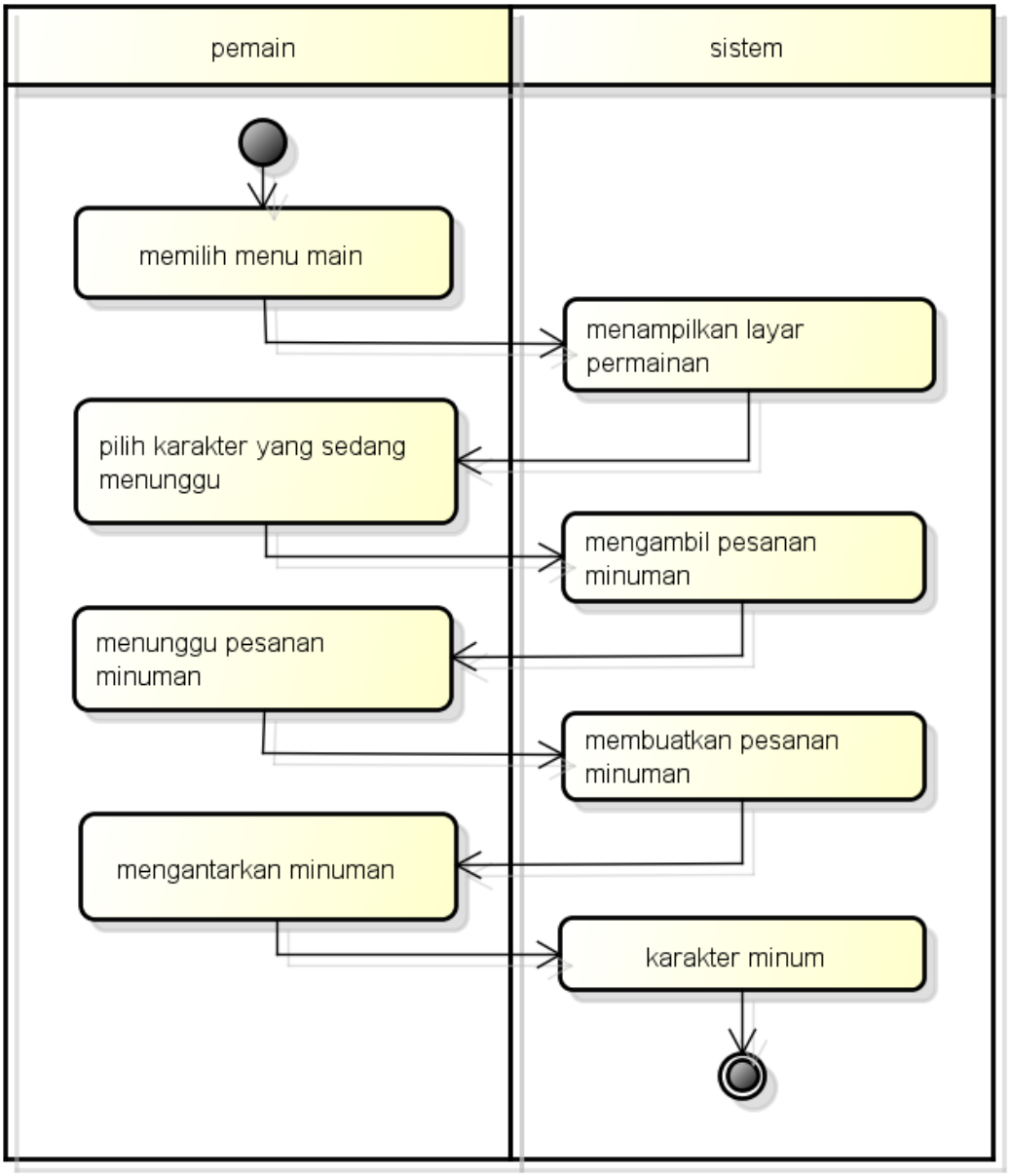

**Gambar 3. 8 Activity diagram mencari kursi**

### *4. Activity Diagram mengantar makanan*



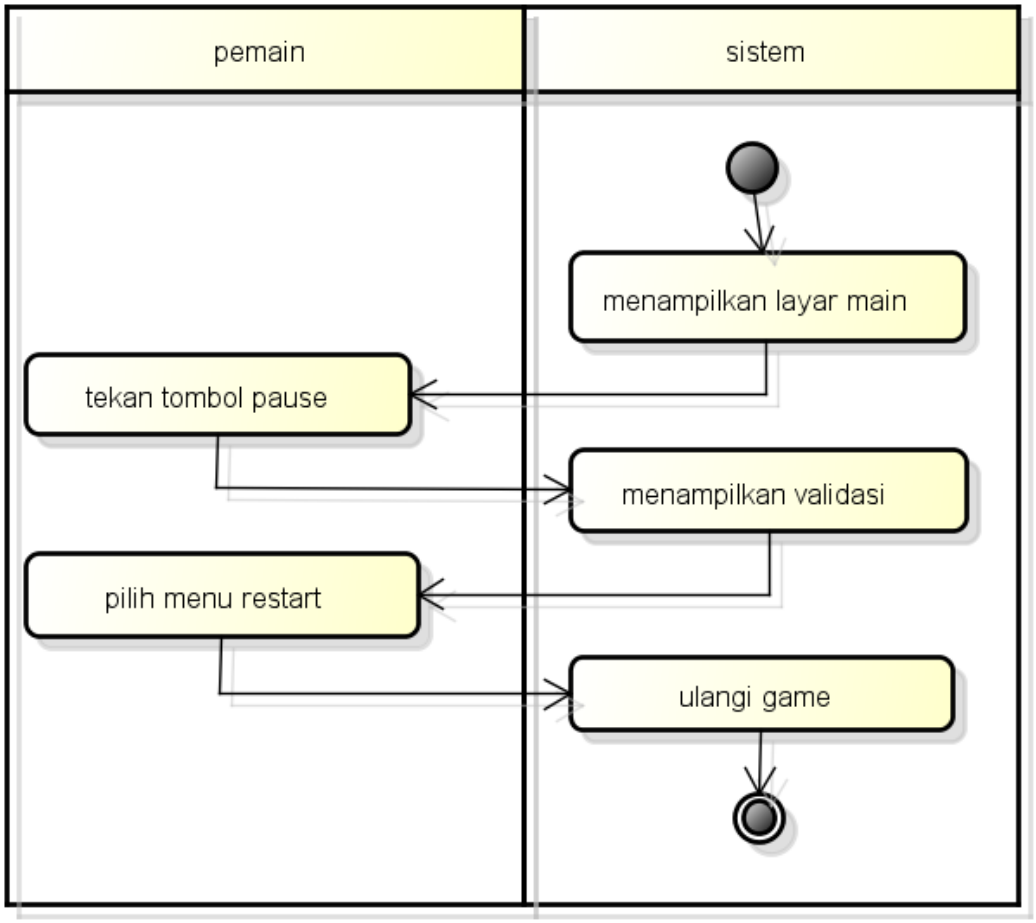

**Gambar 3. 9 Activity diagram mengantar makanan**

***5. Activity Diagram mengantar minuman***




**Gambar 3. 10 Activity diagram mengantarkan minuman**

***6. Activity Diagram Restart***

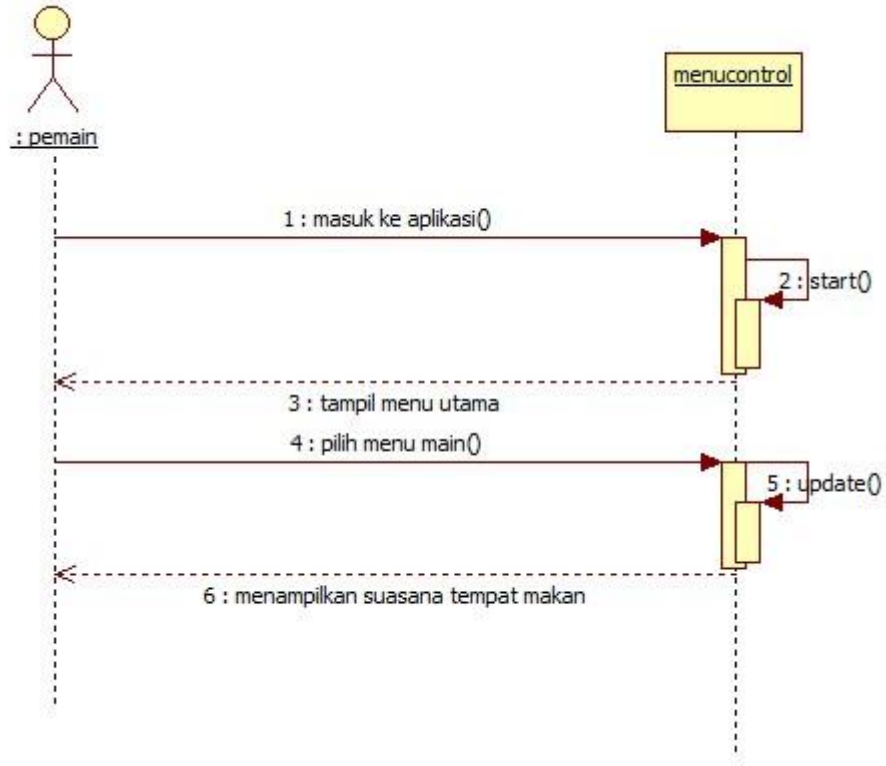



**Gambar 3. 11 Activity diagram Restart**

### 3.6.4 Sequence Diagram

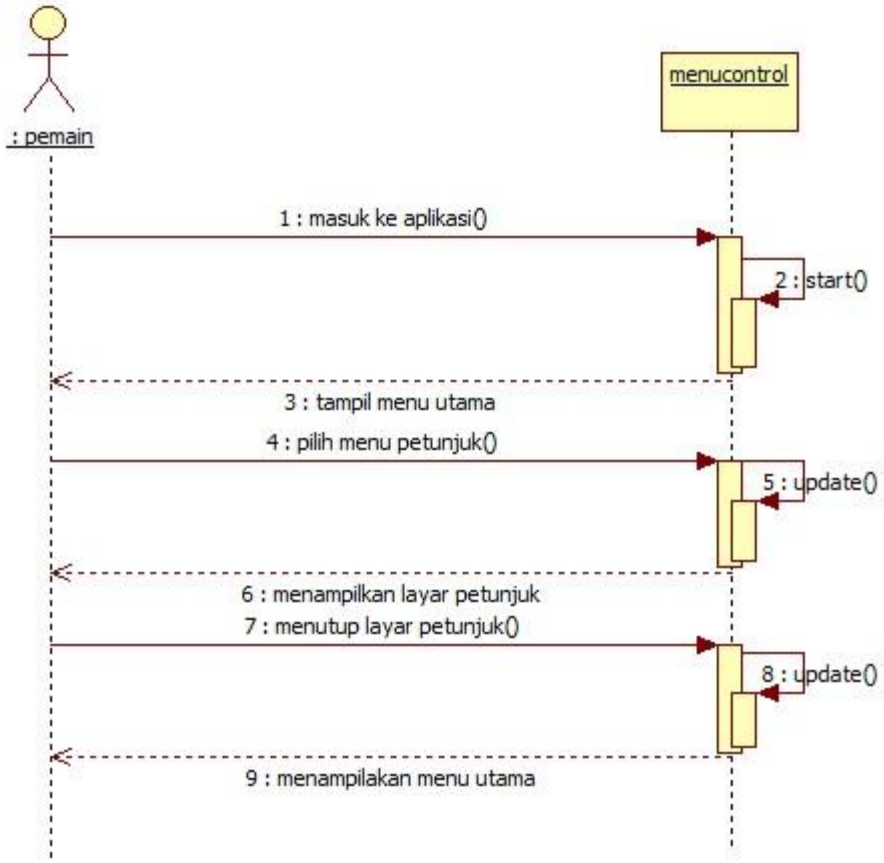
*Sequence diagram* merupakan penggambaran dari interaksi antar objek di dalam dan sekitar sistem (termasuk pengguna, *display*, dan sebagainya). *Sequence diagram* terdiri atas dimensi vertical (waktu) dan dimensi horizontal (objek-objek yang terkait).

1. *Sequence Diagram* untuk memulai permainan



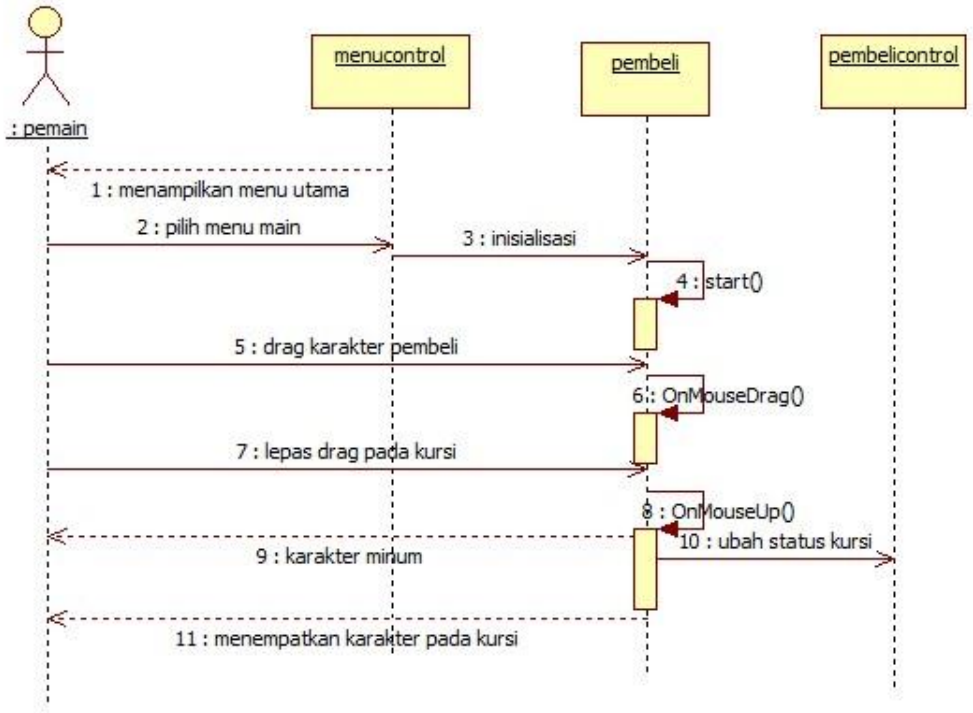

**Gambar 3. 12 Sequence Diagram untuk memulai permainan**

2. *Sequence Diagram* untuk petunjuk



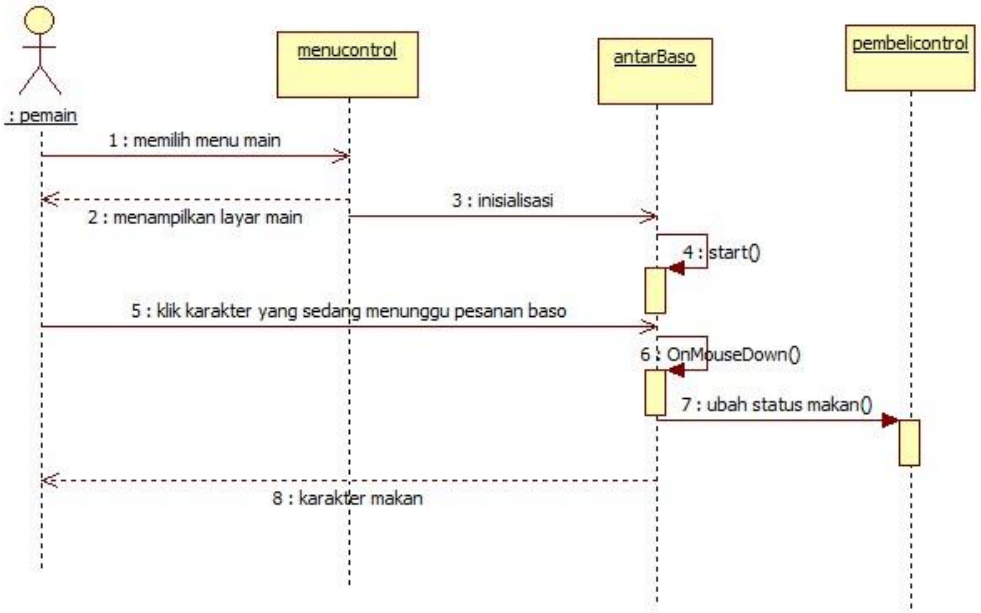

**Gambar 3.13 Sequence Diagram untuk petunjuk**

3. *Sequence Diagram* untuk mencari kursi



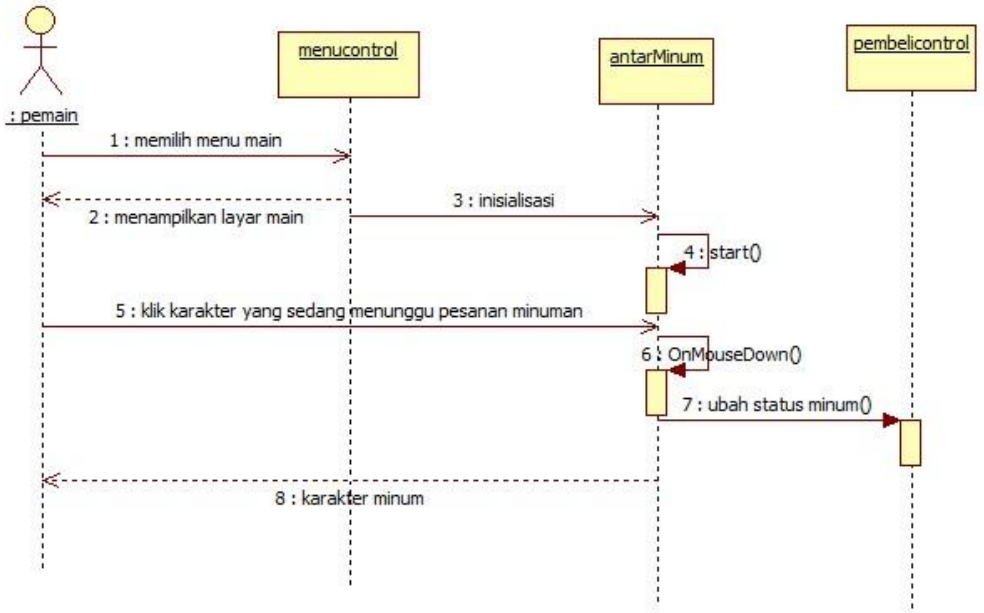

**Gambar 3. 14 Sequence Diagram untuk mencari kursi**

4. *Sequence Diagram* untuk mengantar makanan



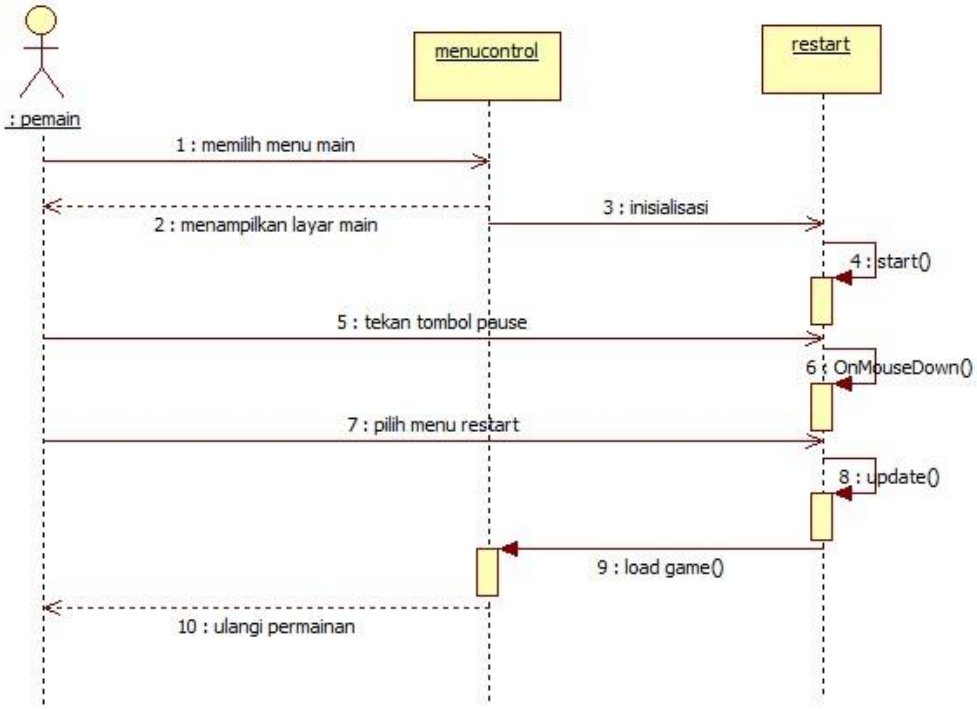

**Gambar 3. 15 Sequence Diagram untuk mengantarkan makanan**

5. *Sequence Diagram* untuk mengantar minuman




**Gambar 3. 16 Sequence Diagram untuk mengantarkan minuman**

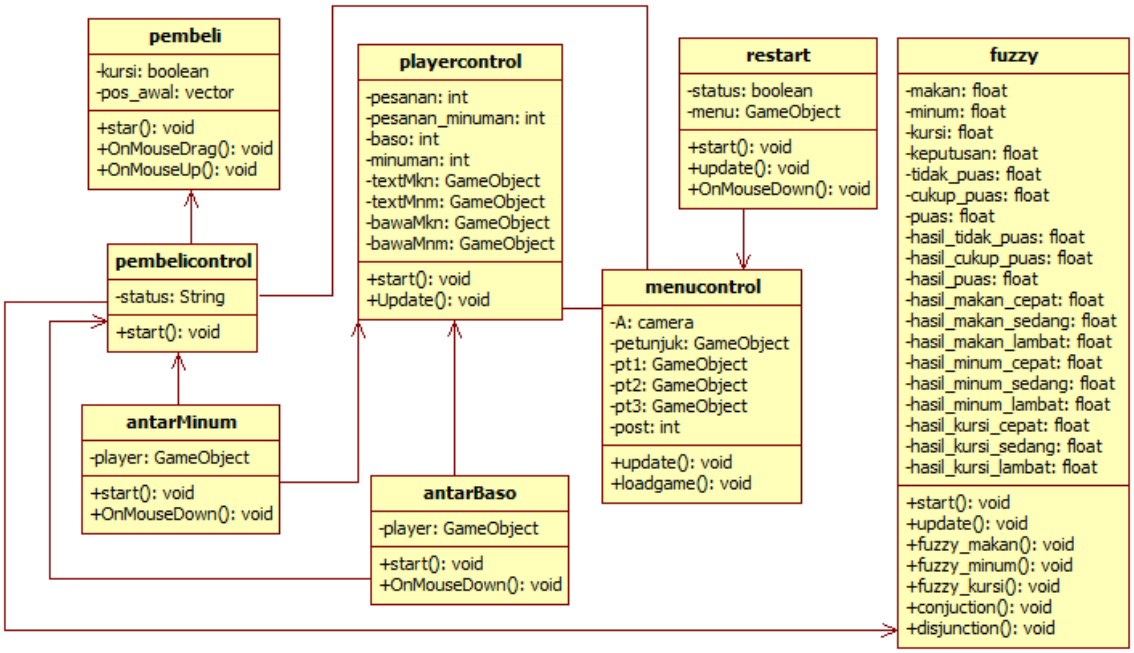
6. *Sequence Diagram* untuk *restart*




**Gambar 3. 17 Sequence Diagram untuk restart**

### 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.




**Gambar 3. 18 *Class Diagram game* tukang bakso**