BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis Masalah

Masalah yang diangkat dalam penulisan ini skripsi ini adalah bagaimana Penerapan metode N-gram untuk *non player character* prediksi formasi agar menggunakan formasi bertahan yang tepat pada game futsal. Analisis masalah sangatlah penting dalam suatu penelitian karena pada tahapan ini merupakan salah satu proses mengidentifikasi masalah-masalah yang timbul dari penerapan sistem yang akan penulis teliti yaitu mengenai metode N-Gram dalam permainan futsal mengenai bagaimana NPC akan memprediksi formasi dari player menggunakan metode tersebut.

Seperti telah dijelaskan sebelumnya bahwa teknik metode yang digunakan dalam memprediksi formasi futsal adalah metode N-Gram yang biasa digunakan untuk memprediksi perilaku player pada game sehingga dapat memperoleh keputusan yang akan dilakukan. Dengan ini diharapkan metode N-Gram dapat memprediksi formasi futsal yang dilakukan player dengan membaca beberapa kali perilaku player sehingga NPC mempunyai kecerdasan buatan yang menunjukan perilaku yang konsisten dan realistis, bereaksi dengan tepat kepada tindakan pemain dan karakter lain seperti halnya manusia.

3.2 Analisis Game

Bagian ini menjelaskan tentang game yang akan digunakan untuk penelitian. Penjelasan terutama untuk bagian-bagian dalam game yang akan diterapkan algoritma. Misalkan jika sebuah karakter NPC yang akan diterapkan algoritma, maka jelaskan bagaimana karakter tersebut akan berperilaku.

Game yang akan dirancang ber-genre *sport*. *Game*. *Game* ini dibangun dengan bentuk prorotype. Dalam permainan yang akan dirancang dalam penelitian ini, pemain berhadapan dengan musuh-musuh berperilaku berbeda. Rincian perilaku musuh dapat dilihat sebagai berikut yaitu formasi serang dan formasi bertahan :

1. Formasi serang

   Adalah suatu formasi atau taktik untuk menyerang dipermainan futsal

   a. Formasi serang 1

      Formasi serang 1 gerakannya menggunakan *motion tween*, formasi serang 1 ini dapat diprediksi oleh bertahan 1.

   b. Formasi serang 2

      Formasi serang 2 gerakannya menggunakan *motion tween*, formasi serang *2* ini dapat diprediksi oleh bertahan 2.

   c. Formasi serang 3

      Formasi serang 3 gerakannya menggunakan *motion tween*, formasi serang 3 ini dapat diprediksi oleh bertahan 3.

2. Formasi Bertahan

   Adalah suatu formasi atau taktik untuk bertahan dan biasanya formasi bertahan digunakan untuk serangan balik dipermainan futsal

   a. Formasi bertahan 1

      Formasi bertahan 1 gerakannya menggunakan *motion tween*, formasi bertahan 1 ini dapat memprediksi serang 1.

   b. Formasi bertahan 2

      Formasi bertahan 2 gerakannya menggunakan *motion tween*, formasi bertahan 2 ini dapat memprediksi serang 2.

   c. Formasi bertahan 3

      Formasi bertahan 3 gerakannya menggunakan *motion tween*, formasi bertahan 3 ini dapat memprediksi serang 3.

3.3 Analisis Masukan

Analisis masukan yang dilakukan dalam penelitian ini adalah menganalisis parameter dalam game untuk formasi player pada *game* futsal, gerakan formasi dan bola menggunakan *motion tween* dan *motion guide* yaitu :

1. *Motion Tween*

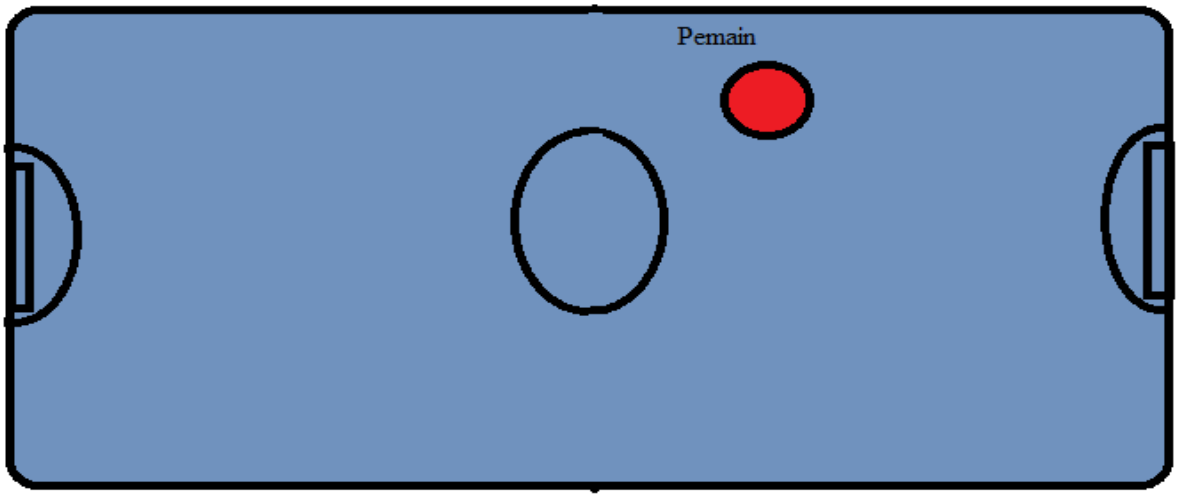
Animasi *motion tween* digunakan apabila kita ingin membuat gerakan animasi yang teratur. Animasi ini sangat mengurangi waktu karena kita tidak perlu membuat animasi secara frame per frame. Sebaliknya Anda hanya membuat frame awal dan frame akhir saja. Dua alasan utama mengapa Animasi motion tween sangat baik yaitu

karena mengurangi pekerjaan mengambar dan meminimalkan ukuran file karena isi dari setiap frame tidak perlu disimpan.

2. Animasi Motion Guide

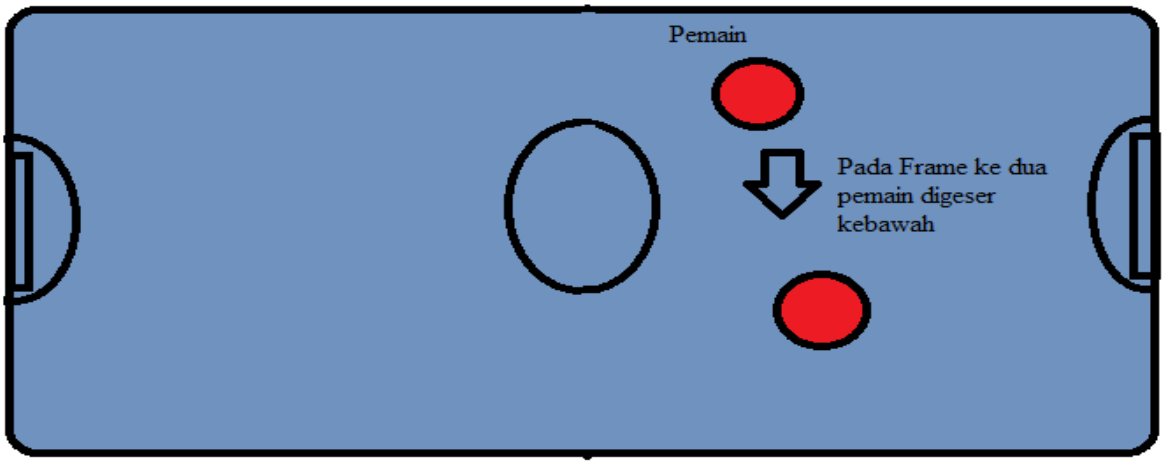
Animasi motion guide adalah animasi yang mempunyai gerakan sesuai dengan jalur yang kita buat. Animasi ini merupakan lanjutan dari animasi motion tween. Animasi ini sangat cocok digunakan untuk jenis animasi yang membutuhkan ketelitian dalam pergerakan yang dikehendaki atau sesuai keinginan pembuat animasi [14].

Pada game futsal ini pergerakan pemain dalam formasi menggunakan *motion tween* seperti di gambar berikut pada gambar 3.1.



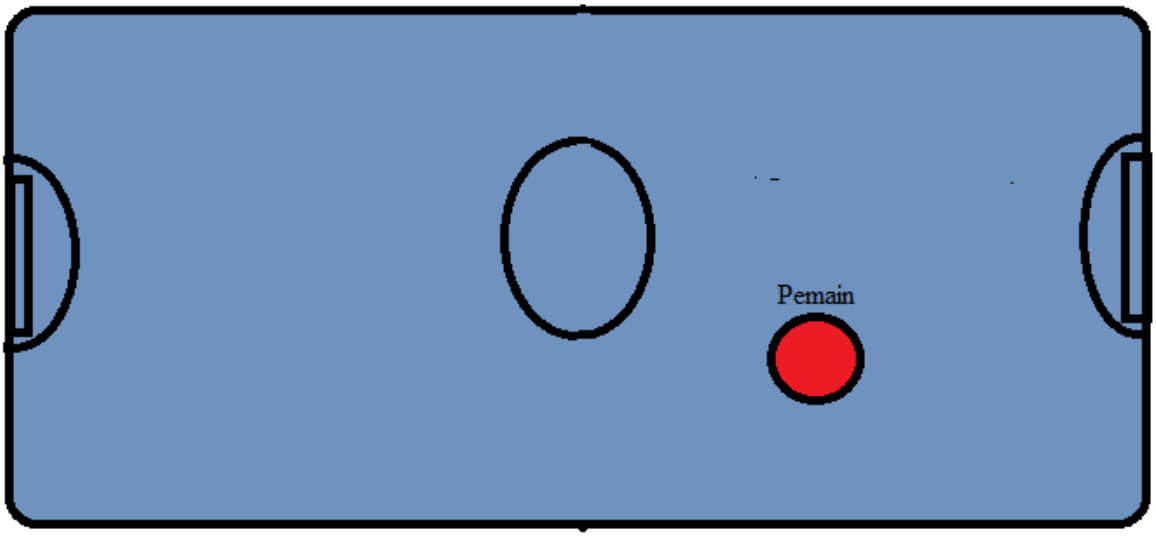
Gambar 3.1 *Motion Tween* pada formasi pemain 1

Selanjutnya pada layer ke 2 pemain pada gambar pertama digeser ke bawah seperti berikut pada gambar 3.2.



Gambar 3.2 *Motion Tween* pada formasi pemain 2

Maka hasilnya akan seperti gambar berikut posisi pemain bergerak ke bawah pada gambar 3.3



Gambar 3.3 *Motion Tween* pada formasi pemain 3

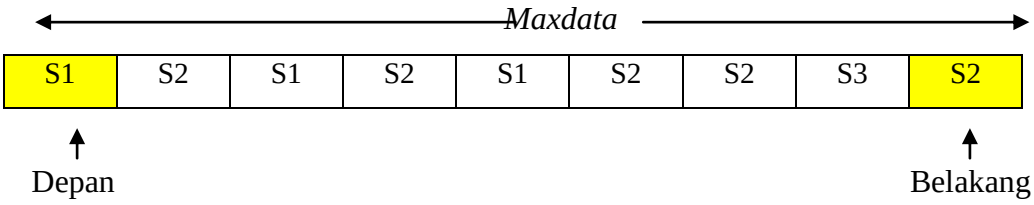
3.4 Analisis Metode

Analisis Algoritma merupakan salah satu metode/alat yang digunakan untuk membantu proses penelitian *game* dengan lebih mudah, proses penyelesaian penelitian *yang di*gunakan antara lain, metode N-gram.

3.4.1 Analisis Struktur Data

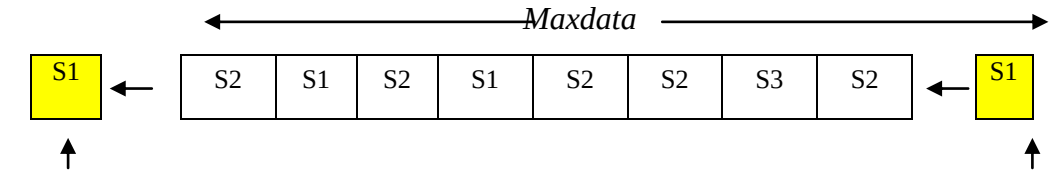
Struktur data pada metode N-gram dalam penelitian ini menggunakan struktur data majemuk yaitu struktur data *queue* (antrian) merupakan kumpulan data gerakan atau formasi dari *palyer* hanya bisa dilakukan pada suatu ujung yaitu *tail* atau belakang dan penghapusan melalui ujung yaitu depan. Antrian itu disebut fifo (*first in first out)* yaitu elemen yang lebih dulu disisipkan merupakan elemen yang lebih dlu diambil. Jika serang 1 yaitu S1, serang 2 yaitu S2 dan serang 3 yaitu S3.

Gambar berikut adalah implementasi queue atau antrian aray statis, data masuk dari belakang.



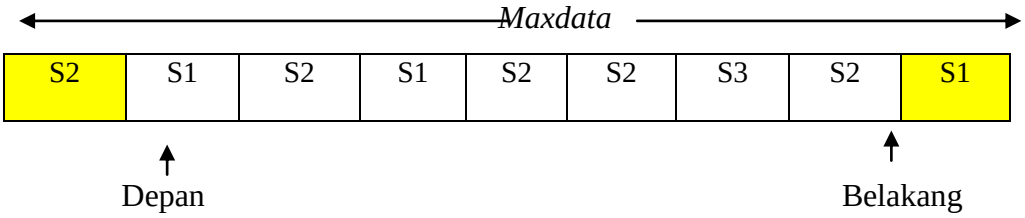
Gambar 3.4 Struktur data Queue 1

Jika *maxdata* sudah penuh maka gerakan selanjutanya akan masuk dari belakang dan gerakan yang masuk paling pertama atau yang paling depan akan dihapus dari aray seperti gambar berikut.



Gambar 3.5 Struktur data Queue 2

Maka data atau gerkan yang masuk dalam aray yang baru masuk dari belakang dan yang paling depan akan di hapus seperti pada gambar berikut



Gambar 3.6 Struktur data Queue 3

3.4.2 Metode N-gram

N-Gram dalam *game* adalah kecerdasan buatan untuk *NPC* yang digunakan untuk memprediksi perilaku *player* pada *game* sehingga dapat memperoleh keputusan yang akan dilakukan Didalam N-gram terdapat *window size*, *sequence* length, dan memory concerns yaitu sebagai berikut dengan parameternya.

3. *Windows size*

   Windows size yang digunakan untuk memprediksi gerakan *player* windows size sendiri selalu bernilai n-1. Pada game ini menggunakan 3-gram maka windows size nya adalah 2.

4. Sequence length

   Sequence length yang digunakan dalam N-gram mengikuti parameter *windows size* , jika menggunakan 3-gram maka setelah gerakan ke 3 jika sequence terbentuk sudah dapat memprediksi

b. *Memory Concerns*

   Memory concerns yang disimpan dalam game ini banyak menggunakan yaitu 100 agar lebih akurat banyak menyimpan data

3.4.3 *Study* kasus

Pada sub bab ini akan dibahas mengenai cara kerja algoritma Metode N-gram dalam game futsal ini adalah sebegai berikut yaitu :

Jika serang 1 adalah S1, serang 2 adalah S2 dan serang 3 adalah S3.

1. Memasukan gerakan dari *player* ke dalam *array* sehingga terbentuk *sequence*. Dalam 10 gerakan formasi yang dicoba, maka akan terbentuk *sequence* :

Tabel 3.1 Data dari *player*

Data dari *player*	0	1	2	3	4	5	6	7	8	9
	S1	S2	S1	S2	S1	S2	S2	S3	S1	S2

2. Memotong-motong *sequence* menjadi *key* yang terbentuk sebanyak n. Dalam penelitian ini menggunakan 3-Gram, berarti data akan dibagi sesuai nilai n nya yaitu 3 :

Tabel 3.2 *Key* yang terbentuk

***Key* yang terbentuk**	012	123	234	345	456	567	678	789
	S1S2S1	S2S1S2	S1S2S1	S2S1S2	S1S2S2	S2S2S3	S2S3S1	S3S1S2

3. Menghitung frequensi kemunculan *pattern* yang terbentuk (*key*). Setelah dibagi sesuai nilai n nya, tinggal hitung frequensinya :

Tabel 3.3 Frequensi kemunculan

Key	**Frequensi kemunculan**
S1S2S1	2
S2S1S2	2
S1S2S2	1
S2S2S3	1
S2S3S1	1
S3S1S2	1

4. Setelah didapat key nya, selanjutnya mencocokan *window* terakhir dengan *key* yang ada. Untuk memprediksi harus menentukan *window size* nya terlebih dahulu dengan rumus n-1. Karena pada penelitian ini menggunakan 3-Gram maka *window size* nya adalah 2 jadi windows terakhir nya 2.

Tabel 3.4 Key yang terakhir yang terbentuk

***Key* yang terbentuk**	012	123	234	345	456	567	678	789
	S1S2S1	S2S1S2	S1S2S1	S2S1S2	S1S2S2	S2S2S3	S2S3S1	S3S1S2

Yang diberi warna kuning adalah window terakhir yang telah dicocokan yaitu S1S2.

5. Selanjutnya mengambil dan mencari frequensi yang paling tinggi. Maka setelah mendapat *window* terakhir yaitu S1S2, selanjutnya mencari pola

sequence yang diawali oleh S1S2. Dari data yang diperoleh terdapat S1S2S1 dengan 2 kemunculan dan S1S2S2 engan 1 kemunculan.

6. Terciptanya prediksi akhirnya setelah menghitung kemunculan frequensi yang tinggi yaitu S1S2S1 dengan 2 kemunculan maka selanjutnya NPC akan memprediksi arah pergerakan *player* yaitu S2 atau serang2.

3.5 Analisis Kebutuhan *Non-Fungsional*

Analisis kebutuhan non-fungsional menggambarkan kebutuhan luar sistem yang diperlukan untuk menjalankan aplikasi yang dirancang. Adapun kebutuhan non-fungsional pada implementasi metode N-Gram pada game futsal meliputi kebutuhan perangkat keras dan kebutuhan perangkat lunak. Analisis kebutuhan non-fungsional bertujuan agar aplikasi yang dibangun dapat digunakan sesuai dengan kebutuhan.

3.5.1 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak atau software merupakan hal yang terpenting dalam mendukung kinerja sebuah sistem. Perangkat lunak digunakan dalam sebuah sistem merupakan perintah-perintah yang diberikan kepada perangkat keras agar dapat saling berinteraksi diantara keduanya. Perangkat lunak yang dibutuhkan sebagai system operasi yang ada pada komputer dan sebagai media penelitian game futsal ini antara lain:

Tabel 3.5 Analisis Kebutuhan Perangkat Lunak

No	PerangkatLunak	Keterangan
1	SistemOperasi	*Windows* 7
2	*Tool*s Program	*Adobe Flash* CS5

3.5.2 Analisis Kebutuhan Perangkat Keras

Untuk menjalankan suatu aplikasi maka diperlukan perangkat keras yang dapat mendukung proses kerja dari sistem itu sendiri. Pada dasarnya game ini dapat dijalankan di semua perangkat komputer/desktop tapi untuk kenyamanan sebaiknya dijalankan di perangkat yang mempunyai spesifikasi sebagai berikut :

Tabel 3.6 Analisis Kebutuhan Perangkat Keras

No	PerangkatKeras	Spesifikasi
1	Prosesor	Kecepatan 2.4 GHz
2	Monitor	Monitor 17'', resolusi 1024x768
3	VGA On-board	Kecepatan 256 MB
4	Optical Drive DVD-ROM	Standard
5	Keyboard	Standard
6	Mouse	Standard
7	Memori	1GB

3.5.3 Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak dilakukan berdasarkan kebutuhan *game*. Spesifikasi kebutuhan perangkat lunak akan dibagi kedalam dua bagian yaitu SKPL-F (Spesifikasi Kebutuhan Perangkat Lunak Fungsional) dan SKPL-NF (Spesifikasi Kebutuhan Perangkat Lunak Non-Fungsional).

Spesifikasi kebutuhan perangkat lunak dalam penelitian ini dapat dilihat pada.

Tabel 3.7 SKPL *User Requirement*

Kode	Kebutuhan
SKPL-F1	Perangkat Lunak dapat memproses hasil masukan untuk kemudian dimulai proses gerakan formasi.

Tabel 3.8 SKPL *System Requirement*

Kode	Kebutuhan
SKPL–F1	Gerakan formasi berupa formasi serang 1, serang 2 dan serang 3
SKPL-F2	Proses masukan yang diterima system yang kemudian diolah menjadi keluaran sesuai keinginan *user*.

Tabel 3.9 SKPL Non-Fungsional

Kode	Kategori	Kebutuhan
SKPL-NF1	*Efficiency (Product Requirement)*	Sistem dapat memproses masukan parameter gerakan formasi.
SKPL-NF2	*Development (Organizatonal Requirement)*	Untuk menjalankan sistem yang di bangun maka dibutuhkan : a. *Microsoft Windows* sebagai sistem operasi b. *Adobe Flash* CS5 sebagai *tools* pemrograman.

3.5.4 Analisis Pengguna

Analisis pengguna gambaran tentang target maupun profil pengguna untuk *game* yang akan dibangun. Perangkat keras dan perangkat lunak yang ada tidak akan berguna apabila tidak ada pengguna yang mengoperasikannya. Berikut ini analisis pengguna untuk *game* futsal dapat dilihat pada Tabel .

Tabel 3.10 Analisis Pengguna

Karakeristik fisik pengguna	
Gender	Laki-lagi dan perempuan
Handedness	Tangan kanan dan kiri
Disabilities	Tidak buta warna total
Pengetahuan dan pengalaman pengguna	
Computer Literacy	Memiliki pengalaman dalam mengoperasikan sebuah komputer
Task Experience	Mengetahui cara menggunakan komputer
Native language or culture	Bahasa Indonesia

Dari analisis pengguna yang telah dilakukan, maka dapat disimpulkan bahwa pengguna yang akan memainkan *game* merupakan pengguna tidak memiliki cacat pada penglihatan, dapat mengoperasikan komputer, dapat atau pernah memainkan aplikasi *game* dikomputer.

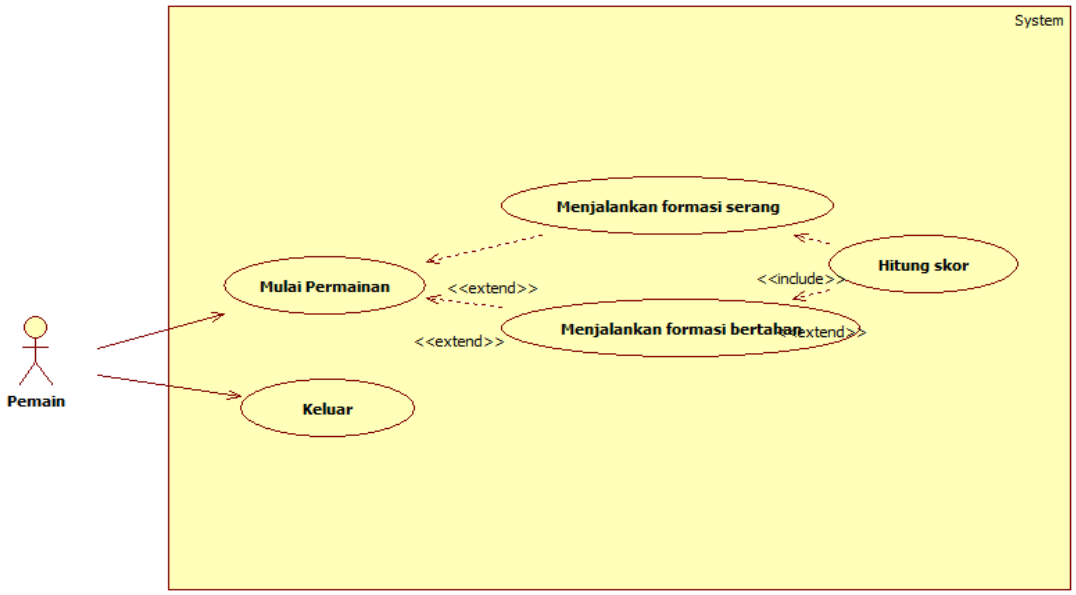
3.6 Analisis kebutuhan Fungsional

Spesifikasi kebutuhan fungsional adalah spesifikasi tentang hal-hal yang akan dilakukan system ketika diimplementasikan. Analisis kebutuhan ini diperlukan untuk menentukan keluaran yang akan dihasilkan sistem, masukan yang diperlukan sistem, lingkup proses yang digunakan untuk mengolah masukan menjadi keluaran.

3.6.1 Use Case Diagram

Use Case atau diagram use case merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. Use Case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

Use Case pada implementasi metode N-gram untuk prediksi formasi pada *game* futsal



Gambar 3.7 Use Case Diagram

1. *Actor Definition*

   berfungsi untuk menjelaskan *Actor* yang terdapat pada *Use case diagram*.

Tabel 3.11 Actor Definition

No	*Actor*	Deskripsi
1	Pemain	Orang yang memainkan *game*

2. Use Case Definition

   Use Case Definition berfungsi untuk menjelaskan fungsi *Use Case* yang terdapat pada *Use case diagram.*

Tabel 3.12 Use Case Definition

No	*Use Case*	Deskripsi
1	Mulai permainan	Proses untuk memulai permainan
2	Menjalankan formasi serang	Proses untuk pergerakan formasi serang
3	Menjalankan formasi bertahan	Proses untuk pergerakan formasi bertahan
4	Hitung skor	Menghitung skor
5	Keluar	Keluar game

3.6.2 Use Case Sekenario

Skenario pada diagram *use case* berisi mengenai alur kejadian yang di dalamnya menjelaskan urutan interaksi antara *actor* dengan *use case* tersebut.

1. Skenario *Use Case* Mulai Permainan

Skenario *use case* ini menjelaskan interaksi antara *actor* yaitu pemain, dengan *use case* mulai permainan berikut.

Tabel 3.13 Skenaro Use Case Mulai Permainan

Identifikasi	
No	1
Nama	Mulai Permainan
Tujuan	Untuk dapat memulai permainan
Actor	Pemain
Skenario Utama	
Kondisi Awal	Menu Utama ditampilkan
Aksi Aktor	**Reaksi Sistem**
1. Memilih mulai permainan	
	2. Menampilkan ke *scene* permainan
3. Memilih formasi serang	
	4. Menjalankan formasi serang 5. Menyimpan data dalam *array* 6. Jika tidak terprediksi formasi random NPC memasukan bola atau *player* memasukan bola 7. Jika terprediksi NPC memasukan bola
8. Memilih formasi bertahan	
Kondisi akhir	Menampilkan formasi bertahan

2. Skenario *Use Case* menjalankan formasi serang

Skenario *use case* ini menjelaskan interaksi antara *actor* yaitu pemain, dengan *use case* menjalan formasi serang berikut.

Tabel 3.14 Skenario Use Case Menjalankan Formasi serang

Identifikasi	
No	2
Nama	Menjalankan serang
Tujuan	Menampilkan pergerakan formasi serang

Actor	Pemain
Skenario Utama	
Kondisi Awal	Permainan dimulai
Aksi Aktor	**Reaksi Sistem**
1. Memilih formasi serang	
	1. Menjalankan formasi serang 2. Menyimpan data dalam *array* 3. Jika tidak terprediksi formasi random NPC memasukan bola atau *player* memasukan bola 4. Jika terprediksi NPC memasukan bola
Kondisi Akhir	Menampilkan formasi bertahan

3. Skenario *Use Case* menjalankan formasi bertahan

Skenario *use case* ini menjelaskan interaksi antara *actor* yaitu pemain, dengan *use case* menjalankan formasi bertahan berikut.

3.15 Skenario Use Case menjalankan formasi bertahan

Identifikasi	
No	3
Nama	Menjalankan formasi bertahan
Tujuan	Menampilkan pergerakan formasi bertahan
Actor	Pemain
Skenario Utama	
Kondisi Awal	Permainan dimulai
Aksi Aktor	**Reaksi Sistem**
1. Memilih formasi bertahan	
	2. Menjalankan formasi bertahan
Kondisi Akhir	Menampilkan formasi bertahan

4. Skenario *Use case* hitung skor

Skenario *use case* ini menjelaskan interaksi antara *actor* yaitu pemain, dengan *use case* hitung skor berikut.

Tabel 3.16 Skenario Use Case hitung skor

Identifikasi	
No	4
Nama	Hitung Skor
Tujuan	Untuk dapat menghitung Skor yang diraih pemain
Actor	Pemain
Skenario Utama	
Kondisi Awal	Permainan dimulai

Aksi Aktor	**Reaksi Sistem**
1. Memasukan bola	
	2. Skor bertambah untuk player
	3. Menghitung skor
Skenario tambahan	
	4. Jika bola tidak masuk maka skor bertambah untuk NPC
	5. Menghitung skor
Kondisi Akhir	Hasil perhitungan skor akhir

5. Skenario *Use case* keluar

Skenario *use case* ini menjelaskan interaksi antara *actor* yaitu pemain, dengan *use case* keluar berikut.

Tabel 3.17 Skenario Use Case keluar

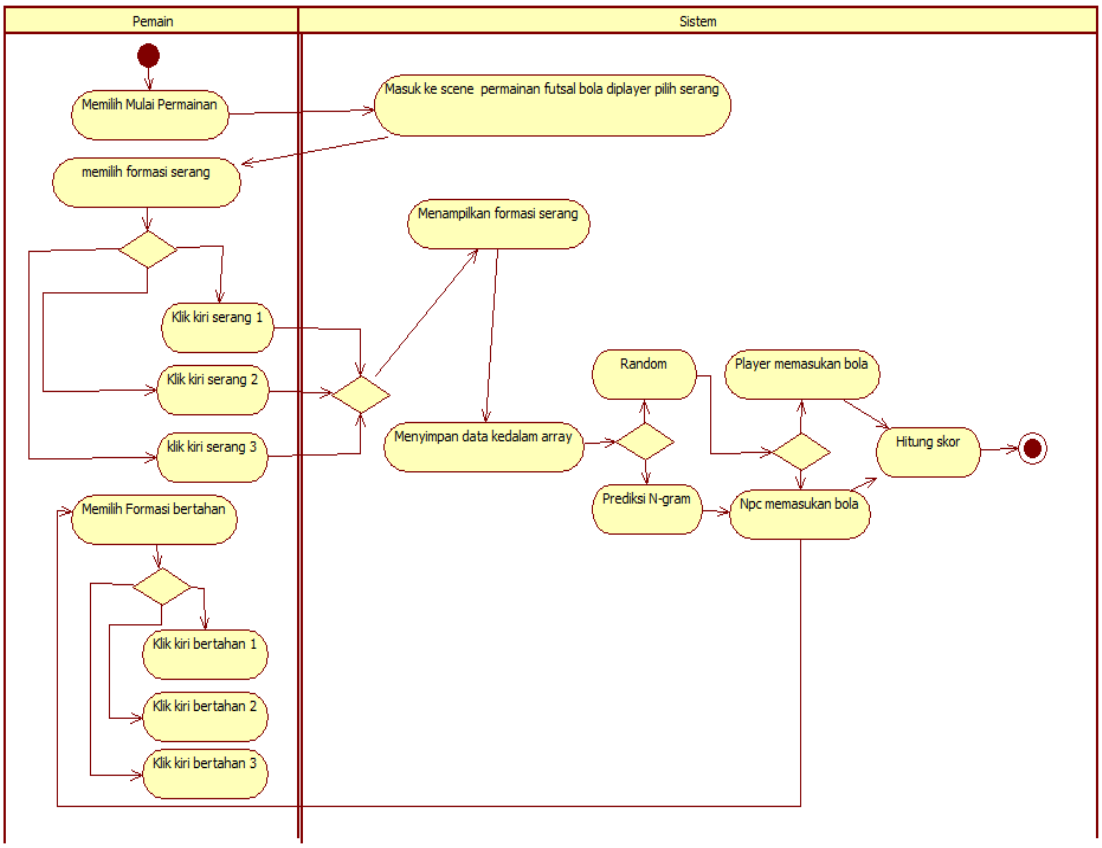
Identifikasi	
No	5
Nama	Keluar
Tujuan	Untuk keluar dari game
Actor	Pemain
Skenario Utama	
Kondisi Awal	Menu utama tampilkan
Aksi Aktor	**Reaksi Sistem**
1. Memilih keluar	
	2. Keluar dari game
Kondisi Akhir	Keluar dari game

3.6.3 Activity Diagram

Activity diagram menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, dimana sebagian besar *state* adalah action dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelum (*internal processing*).

1. *Activity Diagram* Mulai Permainan

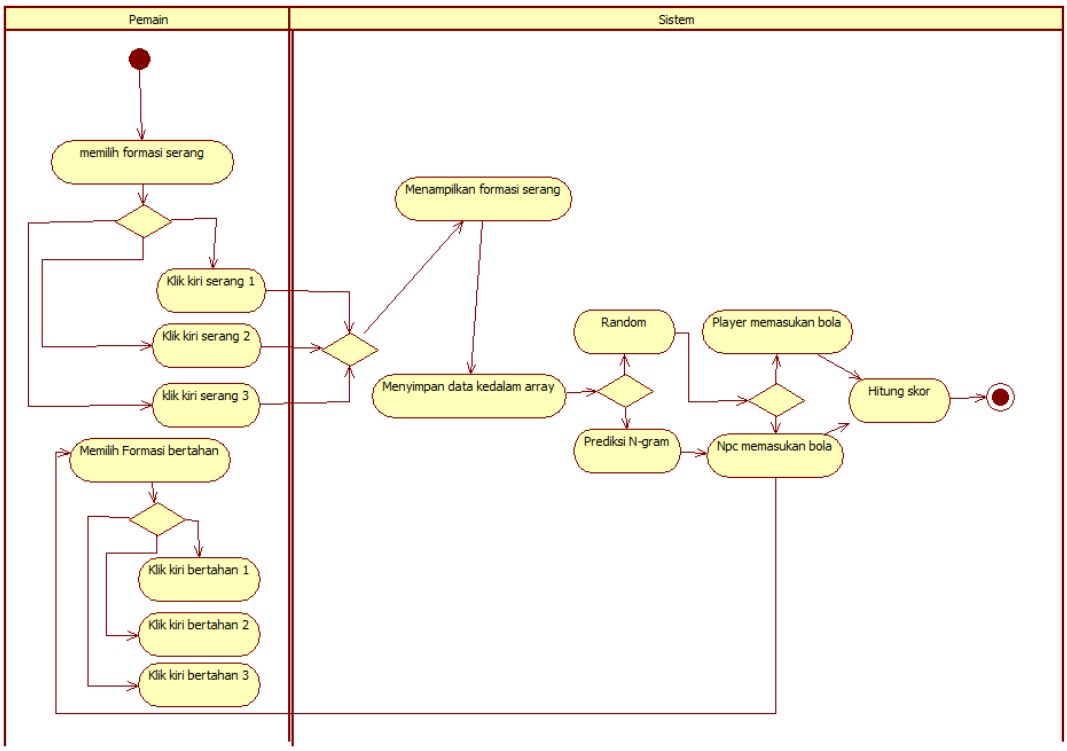
   Berikut ini adalah diagram yang menunjukan alur aksi pada mulai permainan yang dapat dilihat pada Gambar berikut.



Gambar 3.8 Activity Diagram Main

6. *Activity Diagram* menjalankan formasi serang

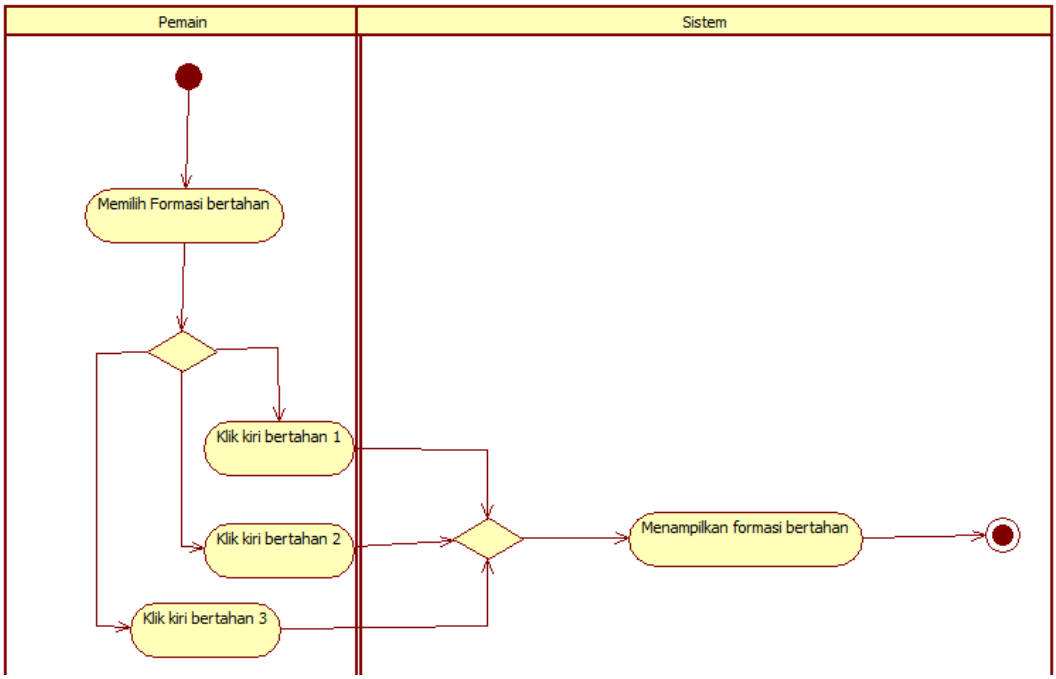
   Berikut ini adalah diagram yang menunjukan alur aksi pada formasi serang yang dapat dilihat pada Gambar berikut.



Gambar 3.9 Activity Diagram formasi serang

7. *Activity Diagram* menjalankan formasi bertahan

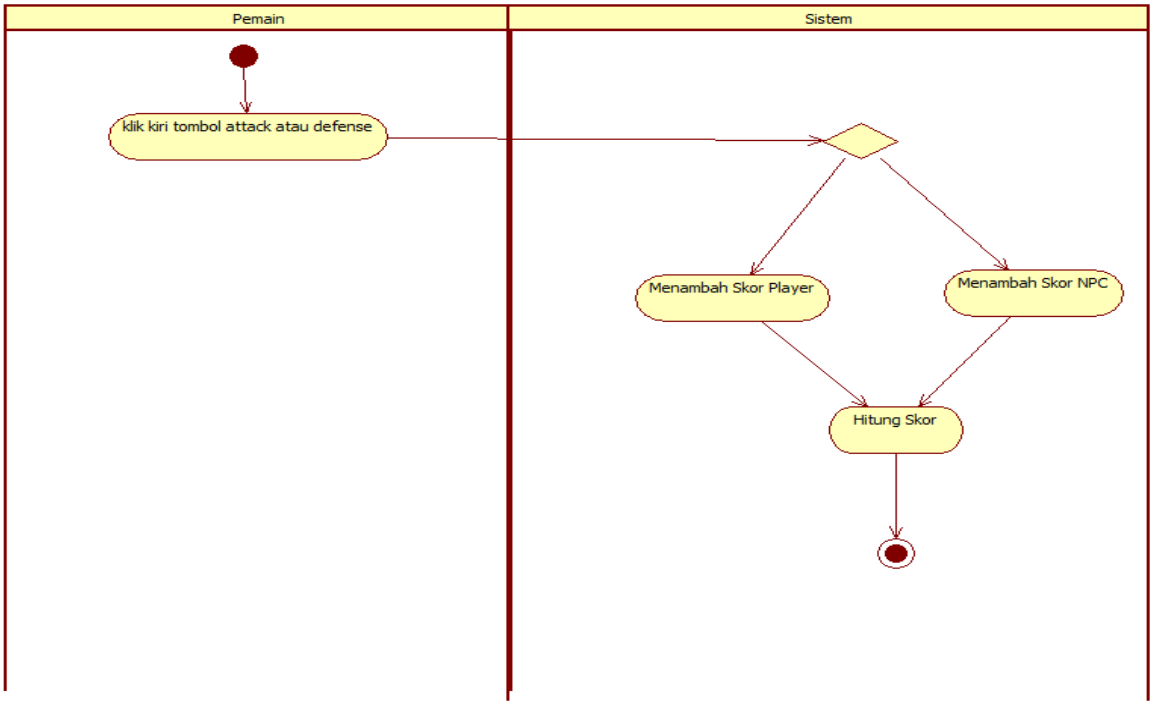
   Berikut ini adalah diagram yang menunjukan alur aksi pada formasi bertahan yang dapat dilihat pada Gambar berikut.



Gambar 3.10 Activity Diagram formasi bertahan

8. Activity Diagram Hitung Skor

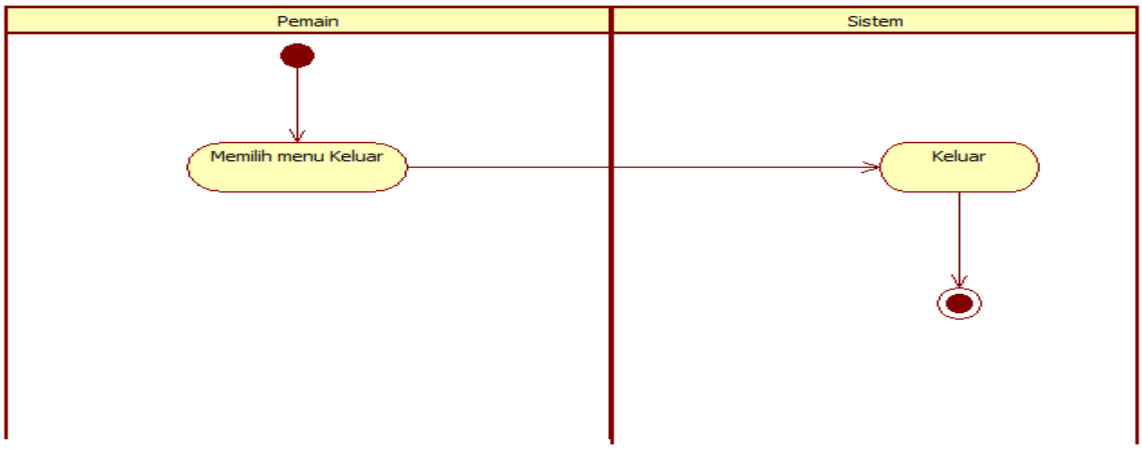
   Berikut ini adalah diagram yang menunjukan alur aksi pada hitung skor yang dapat dilihat pada Gambar berikut.



Gambar 3.11 Activity Diagram hitung Skor

9. Activity Diagram keluar

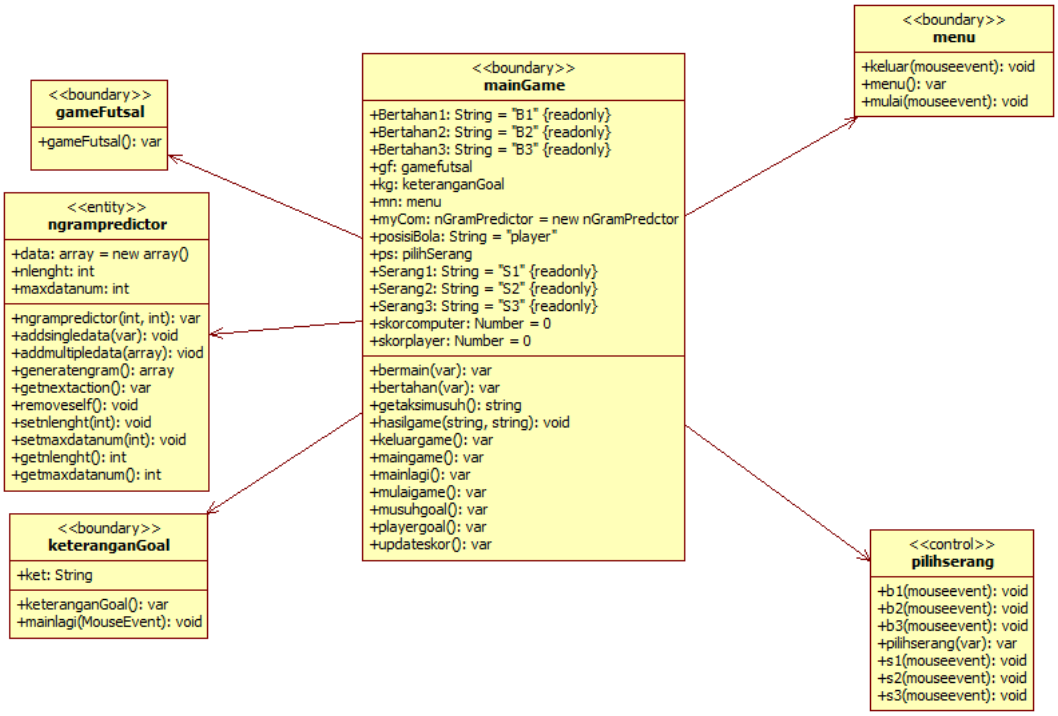
   Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas keluar game yang dapat dilihat pada Gambar berikut.



Gambar 3.12 Activity Diagram keluar

3.6.4 Class Diagram

Class Diagram menggambarkan struktur dan hubungan antar objek-objek yang ada pada sistem.Struktur itu meliputi atribut dan *method* yang ada pada masing-masing *class*. Hubungannya dari masing-masing *class* yang ada tersebut digambarkan dengan menggunakan pewarisan dan generalisasi seperti terlihat pada Gambar berikut.



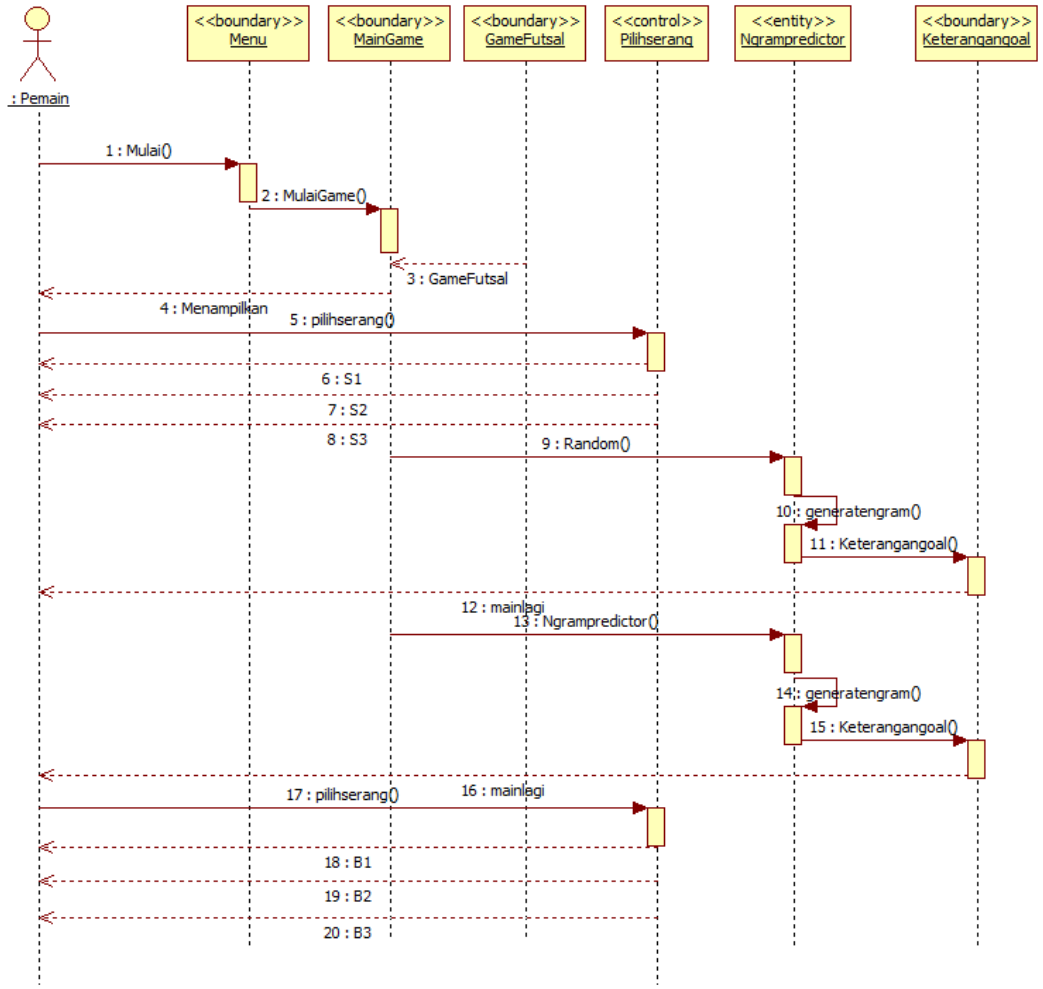
Gambar 3.13 Class Diagram

3.6.5 Sequence Diagram

Sequence Diagram menggambarkan interaksi antar masing-masing objek pada setiap *use case* dalam urutan waktu. Interaksi ini berupa pengiriman serangkaian data antar objek-objek yang saling berinteraksi. *Sequence Diagram* pada Implementasi metode N-gram untuk formasi pada *game* futsal menampilkan *diagram*. Berikut ini penjelasan dari masing-masing sequence diagram:

1. *Sequence Diagram* Mulai Permainan

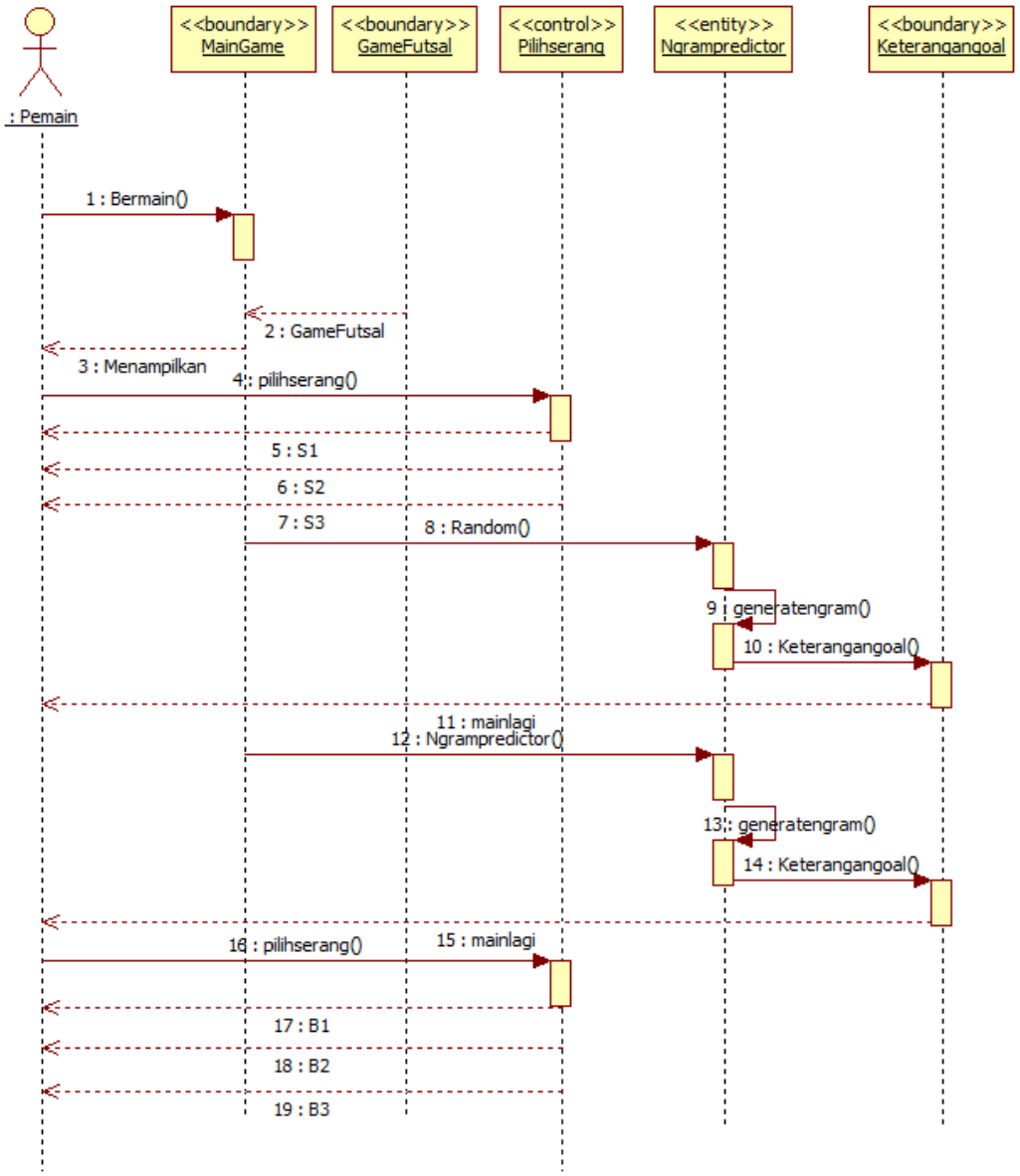
   Sequence diagram mulai permainan pada aplikasi ini dapat dilihat pada Gambar 3.14



Gambar 3.14 Sequence Diagram Main

2. *Sequence Diagram* Menjalankan serang

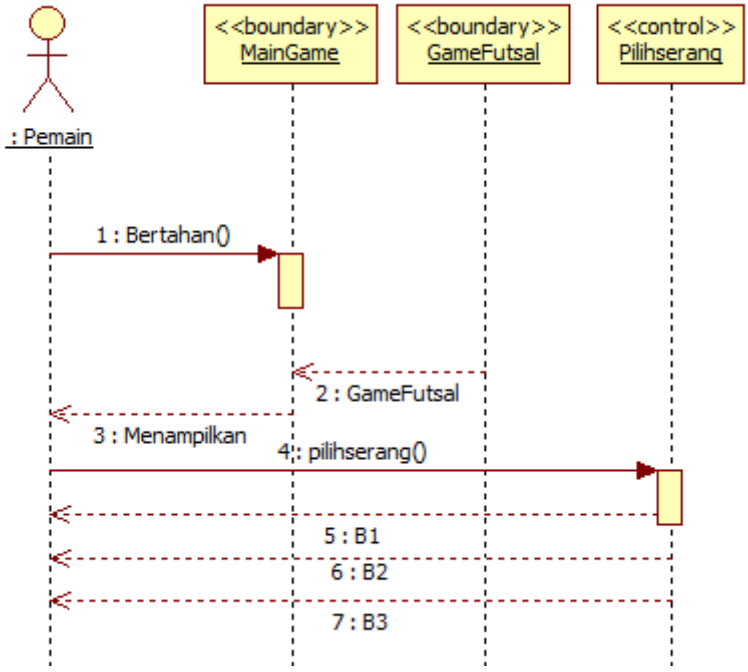
Sequence diagram Menjalankan Formasi serang pada aplikasi ini dapat dilihat pada Gambar 3.15



Gambar 3.15 Sequence Diagram Formasi serang

3. *Sequence Diagram* Menjalankan bertahan

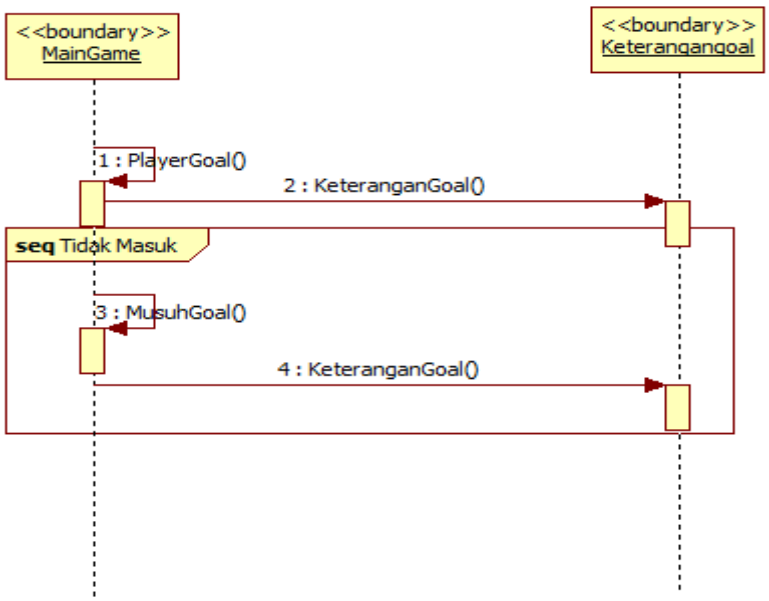
Sequence diagram Menjalankan Formasi bertahan pada aplikasi ini dapat dilihat pada Gambar 3.16.



Gambar 3.16 Sequence Diagram formasi bertahan

4. *Sequence Diagram* Hitung Skor

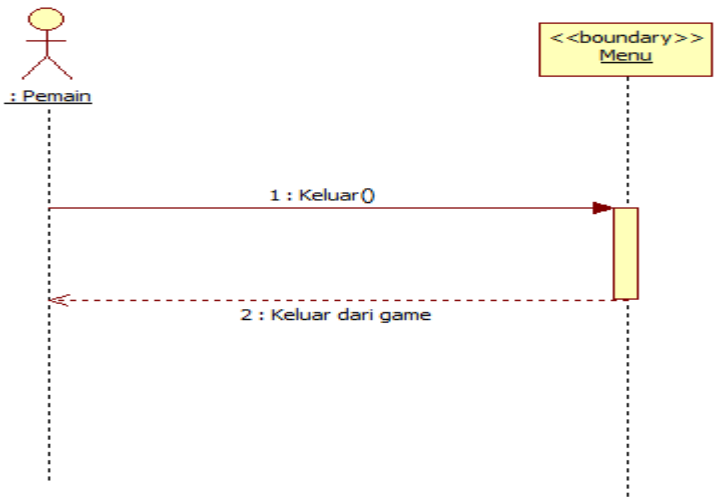
Sequence diagram Hitung Skor pada aplikasi ini dapat dilihat pada Gambar 3.17.



Gambar 3.17 Sequence hitung skor

5. *Sequence Diagram* Keluar

Sequence diagram keluar pada aplikasi ini dapat dilihat pada Gambar 3.18.



Gambar 3.18 Sequence Diagram keluar

3.7 Perancangan Sistem

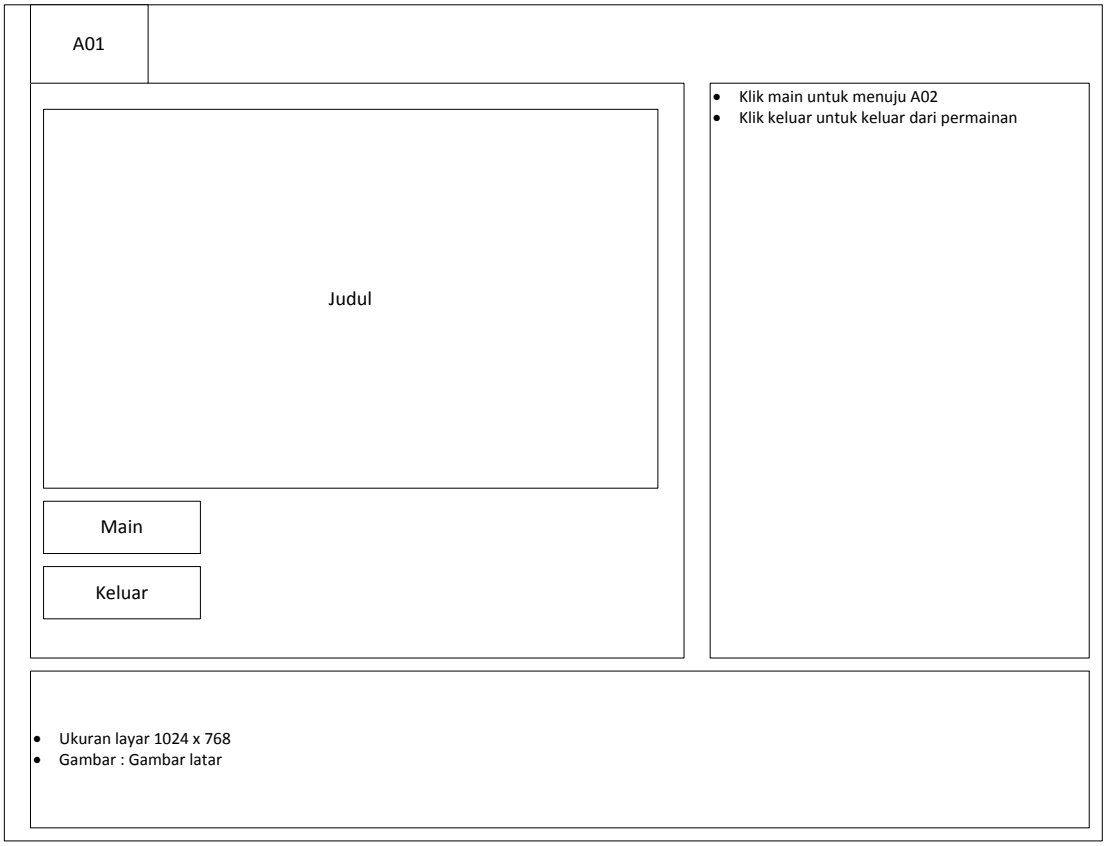
Perancangan sistem merupakan kegiatan untuk merancang aplikasi yang akan dibangun untuk penelitian.

3.7.1 Perancangan antarmuka

Perancangan antarmuka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun untuk penelitian, sehingga akan mempermudah dalam mengimplementasikan aplikasi serta akan memudahkan pembuatan aplikasi. Pada bagian ini akan dijelaskan perancangan antarmuka menu utama, perancangan antarmuka menu main.

3.7.1.1 Perancangan Antarmuka Menu Utama

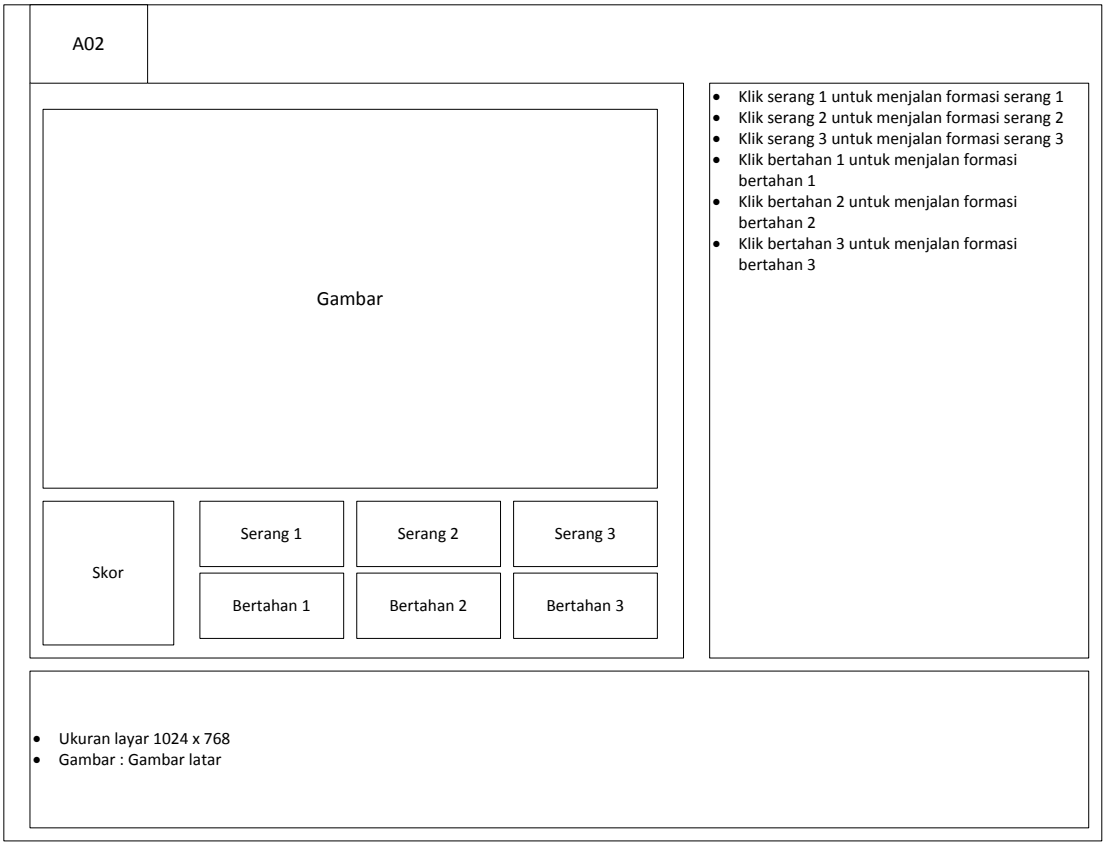
Menu utama merupakan menu yang akan muncul pertama kali pada saat pengguna membuka *game* futsal. Gambar 3.19 menunjukan perancangan antarmuka menu utama.



Gambar 3.19 Perancangan Antarmuka Menu Utama

3.7.1.2 Perancangan Antarmuka Menu Main

Antarmuka menu main merupakan perancangan antarmuka yang menampilkan permainan *game* futsal. Gambar 3.20 merupakan perancangan antarmuka menu main.



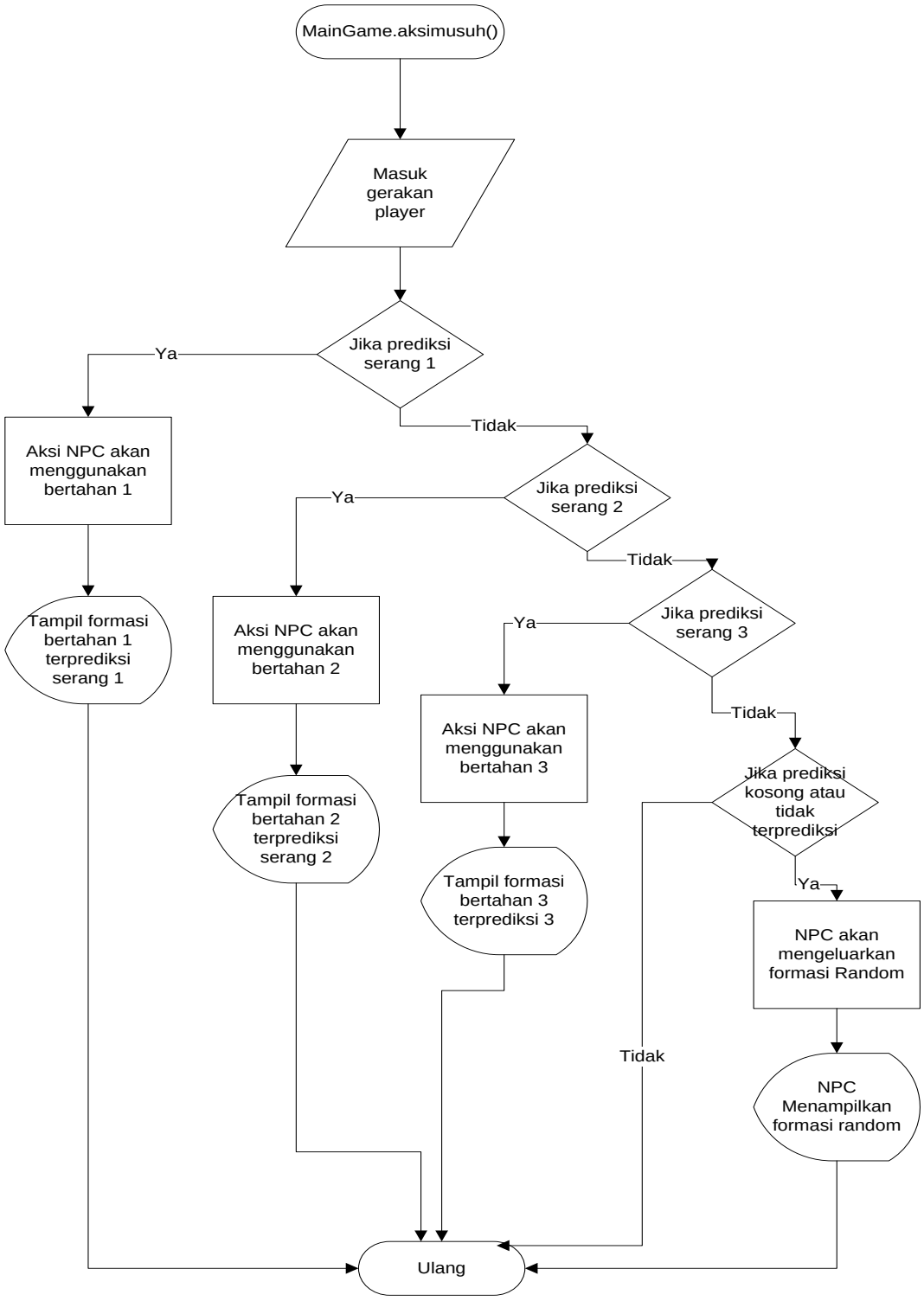
Gambar 3.20 Perancangan Antarmuka Menu Main

3.7.2 Perancangan *Method*

Perancangan *method* perlu dirancang untuk mengetahui alur proses dari program.

1. Perancangan *Method* class Maingame.aksimusuh

Gambar berikut dibawah ini adalah perancangan *method* class Maingame.aksimusuh yaitu aksi NPC pada gambar 3.21.



Gambar 3.21 *Method* class Maingame.aksimusuh