BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis Masalah

Masalah yang diangkat dalam penulisan ini skripsi ini adalah bagaimana implementasi metode N-Gram pada game tenis meja untuk memprediksi arah pukulan bola pada NPC. 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 tenis mengenai bagaimana NPC akan memprediksi arah pukulan bola dari player menggunakan metode tersebut.

Seperti telah dijelaskan sebelumnya bahwa teknik metode yang digunakan dalam memprediksi arah pukulan bola 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 arah pukulan bola 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 adalah *game* tenis meja. *Game* ini dibangun dalam bentuk prototype. Dalam permainan yang akan dirancang dalam penelitian ini, *player* hanya bisa bergerak ke arah kiri, tengah dan kanan serta memukul bola. Sama halnya dengan *player*, NPC bisa untuk memukul dan bergerak menahan bola. NPC pada *game* ini akan diterapkan metode N-Gram untuk bisa

memprediksi arah pukulan bola. Berikut adalah perilaku NPC yang diterapkan metode N-Gram :

1. Sebelum memprediksi NPC akan melakukan gerakan acak untuk menahan bola.
2. NPC akan memprediksi arah pukulan bola dari player setelah membaca beberapa pukulan dari player.
3. NPC akan melakukan pukulan bola secara acak karena metode N-Gram hanya akan diterapkan untuk menahan pukulan bola dari player.

3.3 Analisis Masukan

Analisis masukan yang dilakukan dalam penelitian ini adalah menganalisis parameter pada *game* tenis meja, parameter tersebut adalah parameter arah pukulan bola yang akan dilakukan oleh *player* dan NPC. Terdapat beberapa arah pukulan bola antara lain :

1. Arah pukulan bola dari *player*

Player memiliki 3 arah pukulan yaitu pukulan ke kanan, tengah dan kiri. Berikut adalah rincian pukulan arah bola dari *player* :

a. Pukulan ke kanan

Pukulan arah bola ke kanan dari *player* memiliki titik kordinat x = 250 s.d 350 dan y = 50 s.d 300.

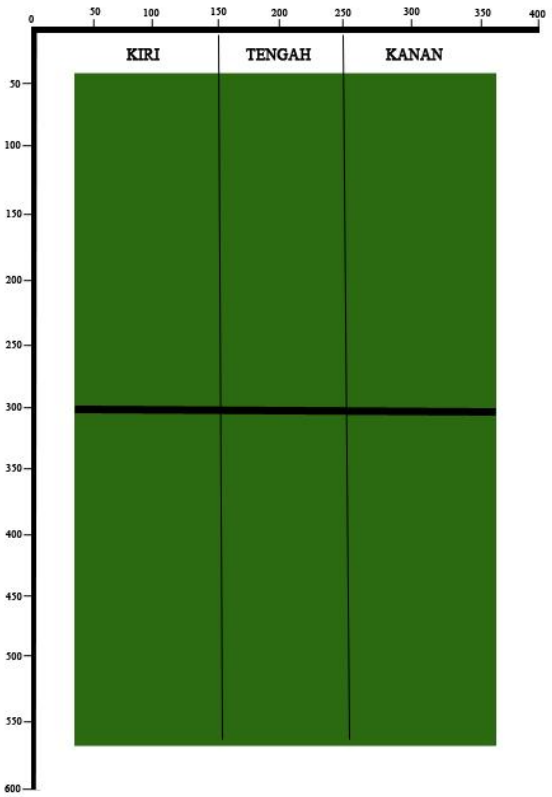
b. Pukulan ke kiri

Pukulan arah bola ke kiri dari *player* memiliki titik kordinat x = 50 s.d 150 dan y = 50 s.d 300.

c. Pukulan ke tengah

Pukulan arah bola ke tengah dari *player* memiliki titik kordinat x = 150 s.d 250 dan y = 50 s.d 300.

Berikut adalah gambar arah bola pukulan dari *player* dapat dilihat pada gambar 3.1.



Gambar 3.1 Arah pukulan dari player

2. Arah pukulan bola dari NPC *(Non Player Character)*

NPC memiliki 3 arah pukulan yaitu pukulan ke kanan, tengah dan kiri. Berikut adalah rincian pukulan arah bola dari NPC :

a. Pukulan ke kanan

Pukulan arah bola ke kanan dari NPC memiliki titik kordinat x = 250 s.d 350 dan y = 300 s.d 570.

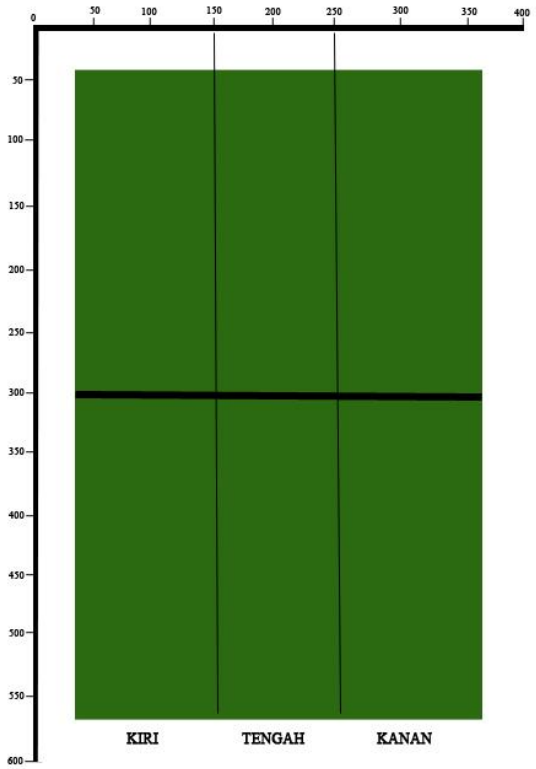
b. Pukulan ke kiri

Pukulan arah bola ke kiri dari NPC memiliki titik kordinat x = 50 s.d 150 dan y = 300 s.d 570.

c. Pukulan ke tengah

Pukulan arah bola ke tengah dari NPC memiliki titik kordinat x = 150 s.d 250 dan y = 300 s.d 570.

Berikut adalah gambar arah bola pukulan dari NPC dapat dilihat pada gambar 3.2.



Gambar 3.2 Arah pukulan dari NPC

3.4 Analisis Metode

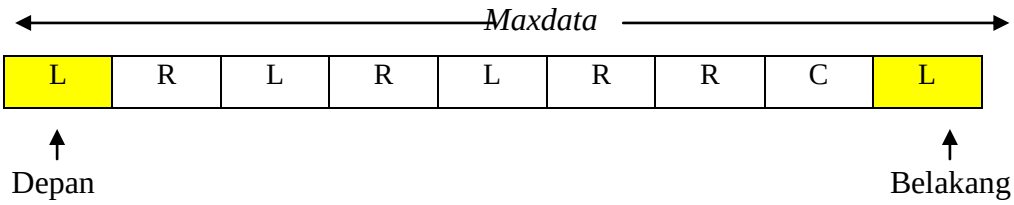
Analisis metode merupakan salah satu alat yang digunakan untuk membantu proses memprediksi dalam game tenis meja untuk NPC untuk membaca gerakan dari player. Metode yang akan penulis gunakan ialah metode N-Gram.

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 pukulam dari *player* 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

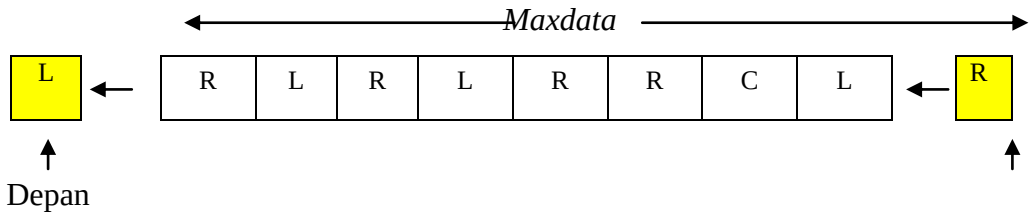
elemen yang lebih dulu diambil. Jika pukulan kiri yaitu L, kanan yaitu R dan tengah yaitu C.

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



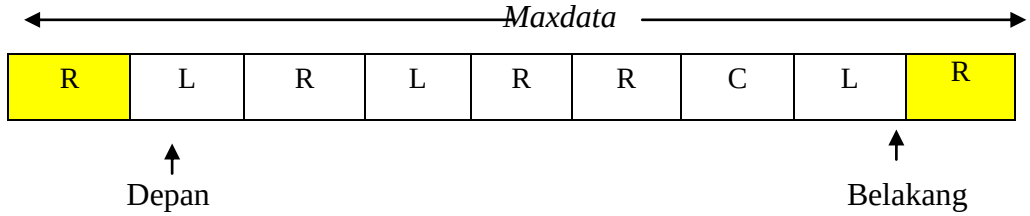
Gambar 3.3 Struktur data Queue 1

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



Gambar 3.4 Struktur data Queue 2

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



Gambar 3.5 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. Di dalam N-gram terdapat *window size*, *sequence* length, dan *memory concerns.*

1. ***Windows size***

   Windows size adalah banyak data yang digunakan untuk memprediksi gerakan *player* windows size sendiri selalu bernilai n-1. Pada game ini menggunakan 3-gram karena pada penelitian ini memiliki 3 gerakan yang dapat diprediksi yaitu gerakan pukulan bola ke kanan, ke kiri dan tengah. Karena menggunakan 3-gram maka windows size nya adalah 2.

2. ***Sequence length***

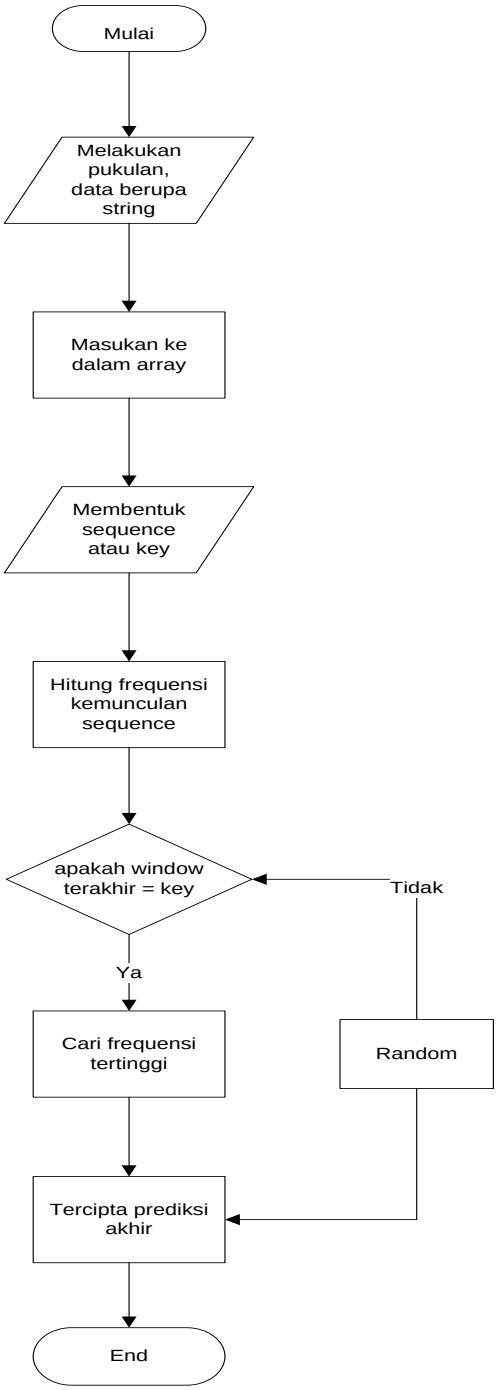
   Sequence length adalah panjang urutan yang digunakan dalam N-gram. Pada awalnya prediksi tidak akan pernah akurat, ini karena belum ada data yang masuk sehingga semua kemungkinan masih dapat terbentuk, atau masih mempunyai frekuensi yang masih terlalu kecil dan juga nilainya masih sama satu sama lain.

3. ***Memory Concerns***

   Memory concerns adalah banyak data yang disimpan dalam N-gram. Prediksi akan menajdi lebih akurat jika AI menyimpan banyak data dari hasil aksi seorang *player*. Pada game ini banyak menggunakan memory concerns nya yaitu 10 agar lebih akurat banyak menyimpan data.

3.4.3 Blok Diagram Metode N-Gram

Berikut adalah blok diagram metode N-Gram yang menunjukan alur dari metode N-Gram dapat dilihat pada gambar 3.6 dibawah ini.



Gambar 3.6 Blok Diagram Metode N-Gram

3.4.4 Studi Kasus

Berikut studi kasus metode N-Gram pada penelitian *game* tenis meja ini adalah sebagai berikut :

Jika pukulan ke kiri adalah L *(left*), ke tengah C (*center*), dan ke kanan R (*right)*

1. Memasukan gerakan dari *player* berupa string ke dalam *array* sehingga terbentuk *sequence*. Dalam 10 kali *throw* atau pukulan 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
	L	R	L	R	L	R	R	C	L	R

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
	LRL	RLR	LRL	RLR	LRR	RRC	RCL	CLR

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**
LRL	2
RLR	2
LRR	1
RRC	1
RCL	1
CLR	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.

LRL RLR LRL RLR LRR RRC RCL CL**LR**

Yang dilingkari adalah window terakhir yang telah dicocokan yaitu LR.

5. Selanjutnya mengambil dan mencari frequensi yang paling tinggi. Maka setelah mendapat *window* terakhir yaitu LR, selanjutnya mencari pola *sequence* yang diawali oleh LR. Dari data yang diperoleh terdapat LRL dengan 2 kemunculan dan LRR dengan 1 kemunculan.
6. Terciptanya prediksi akhirnya setelah menghitung kemunculan frequensi yang tinggi yaitu LRL dengan 2 kemunculan maka selanjutnya NPC akan memprediksi arah pergerakan *player* akan memukul ke kiri (*left*).

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* tenis meja meliputi spesifikasi kebutuhan perangkat lunak, kebutuhan perangkat lunak dan kebutuhan perangkat keras. Analisis kebutuhan *non*-fungsional bertujuan agar aplikasi yang dibangun dapat digunakan sesuai dengan kebutuhan.

3.5.2 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 pengembangan game tenis meja ini antara lain:

Tabel 3.4 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 3.5 Spesifikasi Perangkat Keras yang Dibutuhkan

No	PerangkatKeras	Spesifikasi
1	Prosesor	Kecepatan 2.4 GHz
2	Monitor	Monitor 17'', resolusi 1024x768
3	VGA On-board	Kecepatan 512 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 3.6 SKPL *User Requirement*

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

Tabel 3.7 SKPL *System Requirement*

Kode	Kebutuhan
SKPL–F1	Gerakan pukulan berupa pukulan ke kiri, kanan dan tengah.
SKPL-F2	Proses masukan yang diterima system yang kemudian diolah menjadi keluaran sesuai keinginan *user*.

Tabel 3.8 SKPL Non-Fungsional

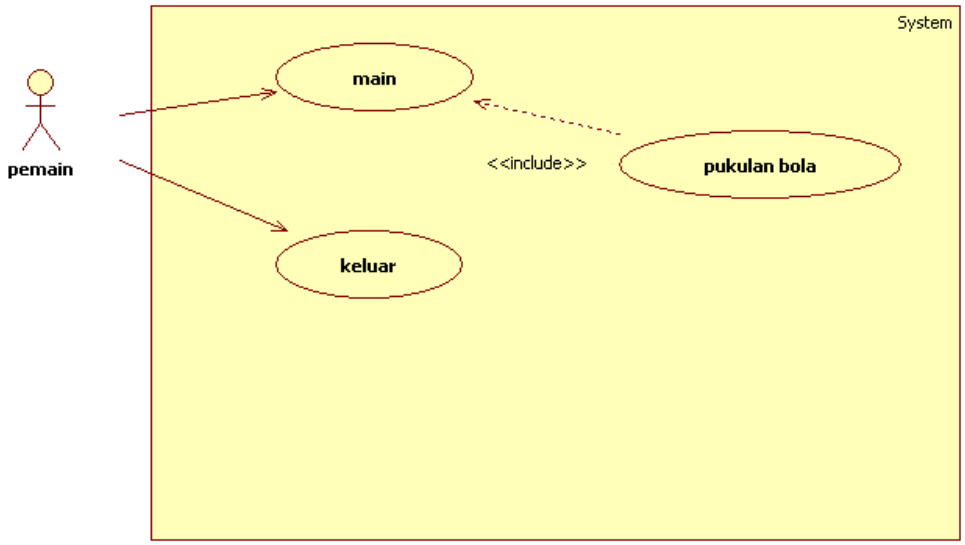
Kode	Kategori	Kebutuhan
SKPL-NF1	*Efficiency (Product Requirement)*	Sistem dapat memproses masukan parameter arah pukulan bola.
SKPL-NF2	*Development (Organizatonal Requirement)*	Untuk menjalankan sistem yang di bangun maka dibutuhkan : a. *Microsoft Windows* sebagai sistem operasi b. *Adobe Flash* CS5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

3.6 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 Ca*se 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. Berikut *Use Case* pada implementasi metode N-Gram untuk menentukan arah pukulan bola pada *game* tenis meja ini yaitu :



Gambar 3.7 *Use Case* Diagram *Game* Tenis Meja

1. *Actor Definition*

Actor Definition berfungsi untuk menjelaskan *Actor* yang terdapat pada *Use case diagram*. *Actor Definition* diterangkan pada Tabel 3.6.

Tabel 3.9 *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. Use Case Definition* diterangkan pada Tabel 3.7.

Tabel 3.10 *Use Case Definition.*

No	***Use Case***	**Deskripsi**
1	Memulai Permainan	Proses untuk memulai permainan
2	Pukulan Bola	Proses untuk karakter/player memukul bola
3	Keluar *Game*	Proses untuk keluar atau menghentikan *game* ini.

3.6.2 *Use Case* Skenario

Skenario *use case* merupakan bagian pada *use case* yang menunjukkan proses apa saja yang terjadi pada setiap bagian di dalam *use case*, dimana pemain memberikan perintah pada setiap bagian dan respon apa yang diberikan oleh sistem kepada pemain setelah pemain memberikan perintah pada setiap bagian-bagian *use case*. Berikut ini beberapa skenario *use case* berdasarkan *use case* yang ada yaitu :

Tabel 3.11 Main

Identifikasi		
Nomor	1	
Nama	Main	
Tujuan	Pemain dapat memulai dan memainkan game ini	
Deskripsi	Pemain dapat memulai game ini dan bertanding melawan NPC	
Aktor	Pemain	
Skenario Utama		
Kondisi Awal		Menu utama ditampilkan
Aksi Aktor		Reaksi Sistem
1. Pemain masuk ke dalam game/aplikasi ini.		
		2. Menampilkan layar menu awal game
3. Pemain memilih menu main		
		4. Menampilkan layar main.
Kondisi Akhir		Permainan dimulai

Tabel 3.12 Pukulan Bola

Identifikasi	
Nomor	2
Nama	Pukulan Bola
Tujuan	Untuk menentukan arah pukulan bola
Deskripsi	Pemain menggerakan karakter pada game untuk menentukan arah pukulan bola
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Permaianan dimulai
Aksi Aktor	Reaksi Sistem
1. Pemain menekan tombol *keyboard* untuk menggerakan karakter dan memukul bola. Tombol keyboard yang digunakan adalah A,S dan D.	
	2. Setelah menekan tombol bola akan mengarah sesuai apa yang pemain tekan. Tombol A untuk pukulan ke kiri, S untuk pukulan ke tengah dan D untuk pukulan ke kanan.
	3. Setelah memukul sistem mulai cek gerakan dan memasukan data ke dalam array. Jika terprediksi bola akan tertahan NPC dan jika tidak NPC akan bergerak random.
Kondisi Akhir	Pemain akan dapat menentukan arah pukulan bola.

Tabel 3.13 Keluar Permainan.

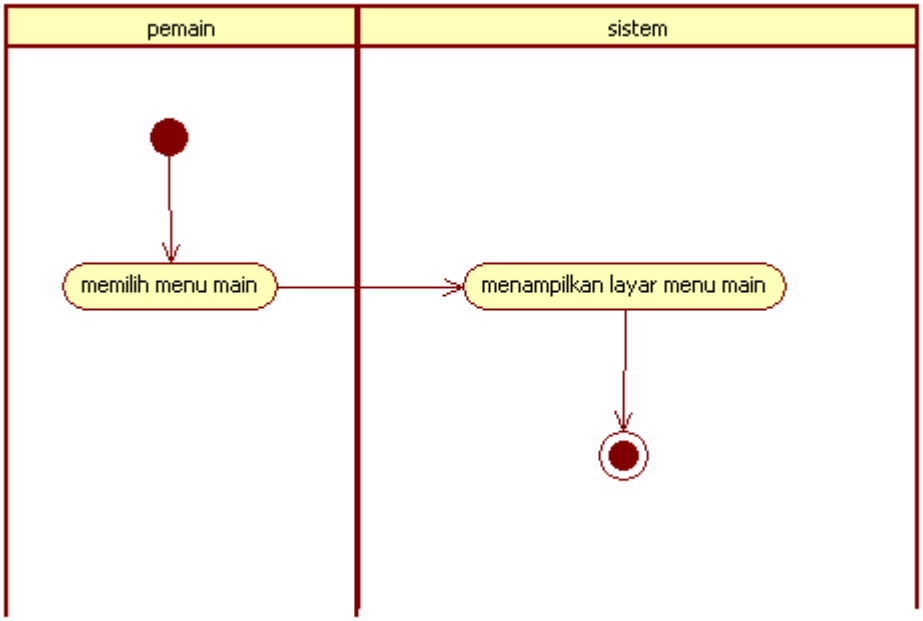
Identifikasi	
No	3
Nama	Keluar Permainan
Tujuan	Keluar dari permainan
Deskripsi	Ketika permainan berakhir atau pemain ingin mengakhiri game, system harus tahu langkah untuk keluar game berdasarkan jenis keluar dari pemain.
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Menu utama ditampilkan
Aksi Aktor	Reaksi Sistem
1. Pemain menekan tombol 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*). Berikut *Activity Diagram* pada implementasi metode N-Gram untuk menentukan arah pukulan bola pada *game* tenis meja ini yaitu :

1. ***Activity Diagram* Main**

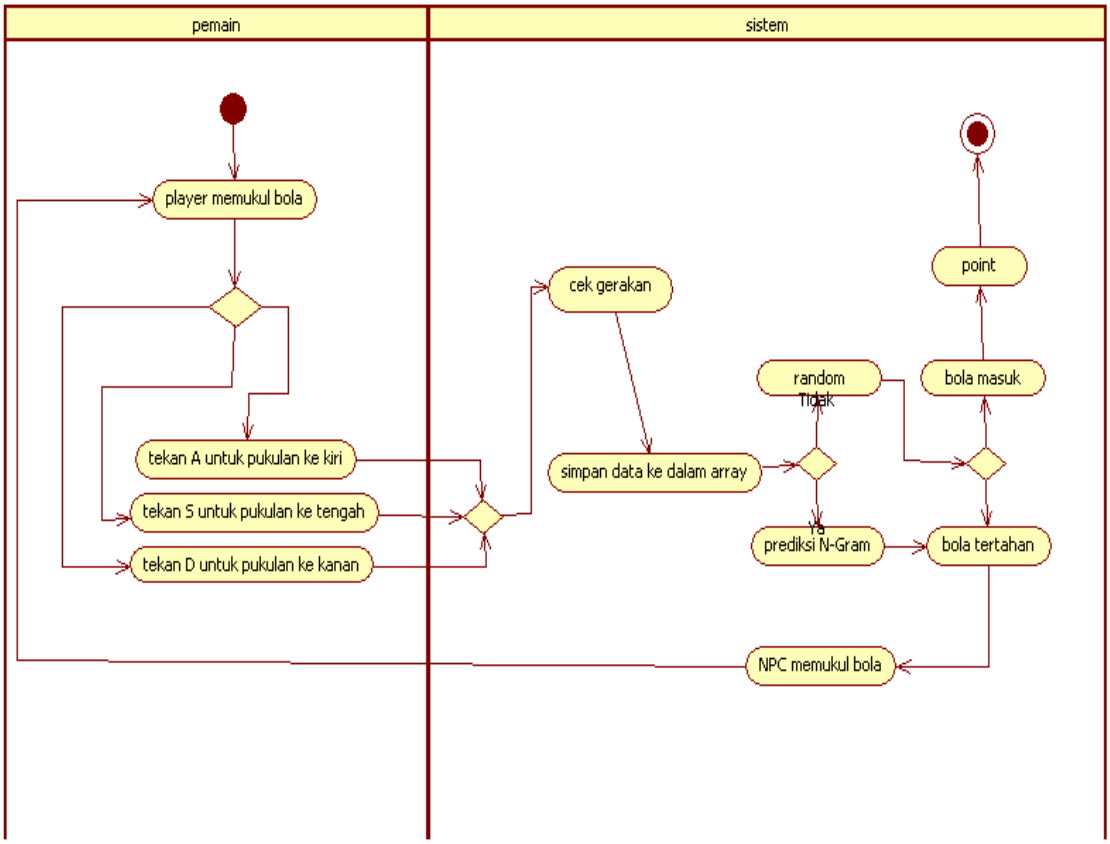
   Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas main atau memulai permainan yang dapat dilihat pada Gambar 3.8.



Gambar 3.8 *Activity Diagram* Main

2. *Activity Diagram* Pukulan Bola

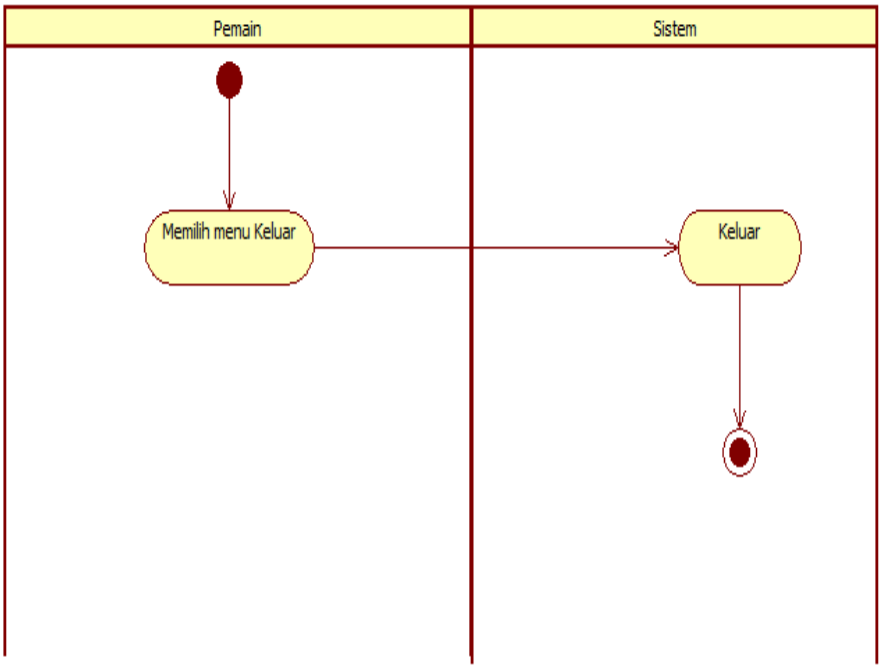
Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas pukulan bola yang dapat dilihat pada Gambar 3.9.



Gambar 3.9 *Activity Diagram* Pukulan Bola

3. *Activity Diagram* Keluar

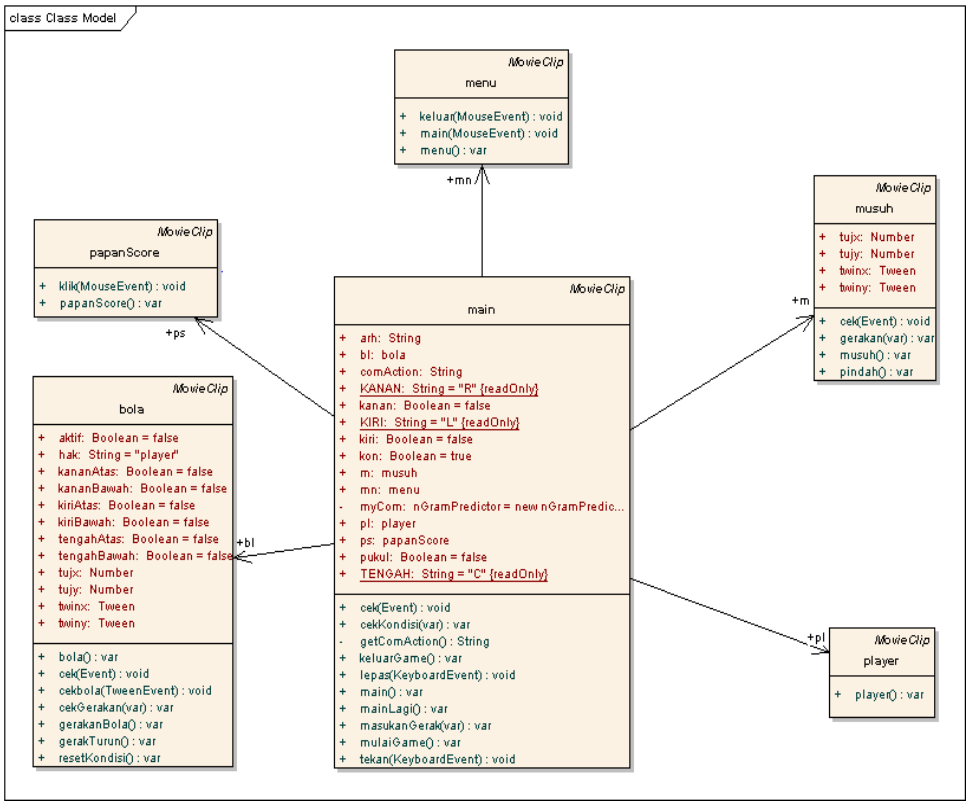
Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas keluar game yang dapat dilihat pada Gambar 3.10.



Gambar 3.10 *Activity diagram* Keluar

3.6.4 Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang dibuat untuk membangun sistem. Berikut adalah *class* diagram pada *game* tenis meja :



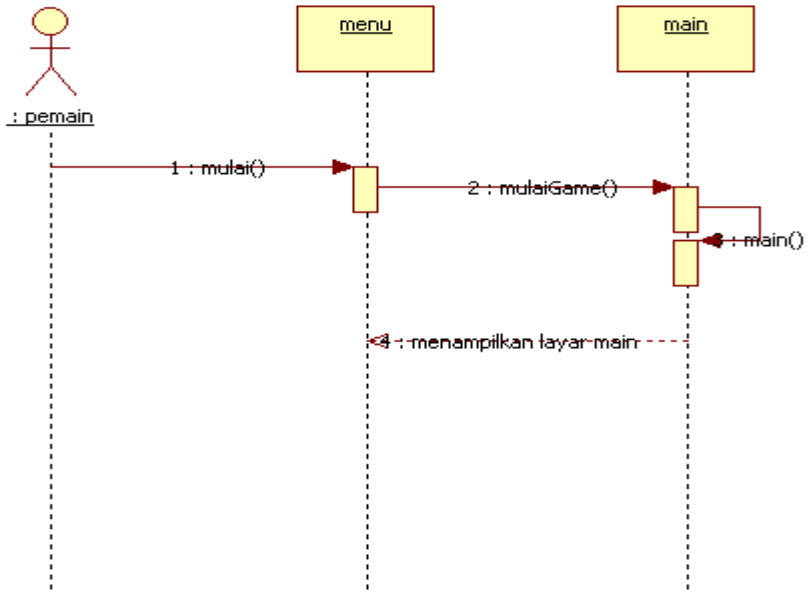
Gambar 3.11 *Class Diagram*

3.6.5 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan disekitar sistem berupa *message* yang digambarkan terhadap waktu. Berikut ini beberapa *sequence* diagram yang terdapat pada *game* tersebut yaitu :

1. *Sequence Diagram* Main

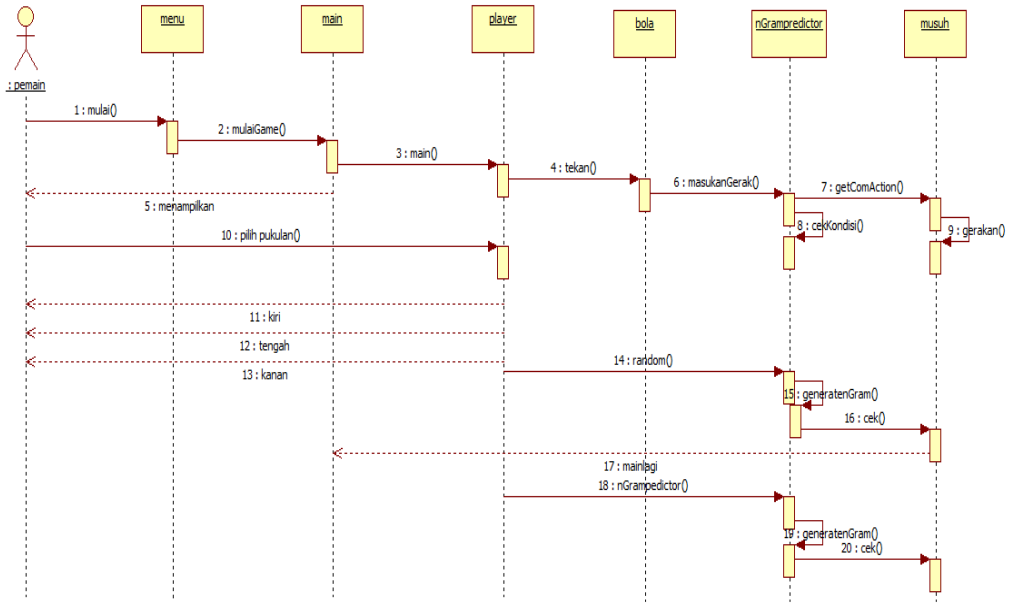
Sequence diagram main pada aplikasi ini dapat dilihat pada Gambar 3.12.



Gambar 3.12 *Sequence Diagram* Main

2. *Sequence Diagram* Pukulan Bola

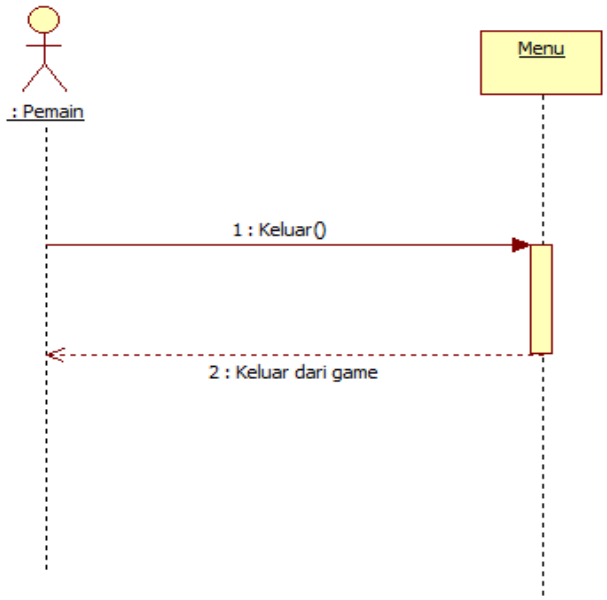
Sequence diagram pukulan bola pada aplikasi ini dapat dilihat pada Gambar 3.13.



Gambar 3.13 *Sequence Diagram* Pukulan Bola

3. *Sequence Diagram* keluar

Sequence diagram keluar pada aplikasi ini dapat dilihat pada Gambar 3.14.



Gambar 3.14 *Sequence Diagram* Keluar

3.7 Perancangan Sistem

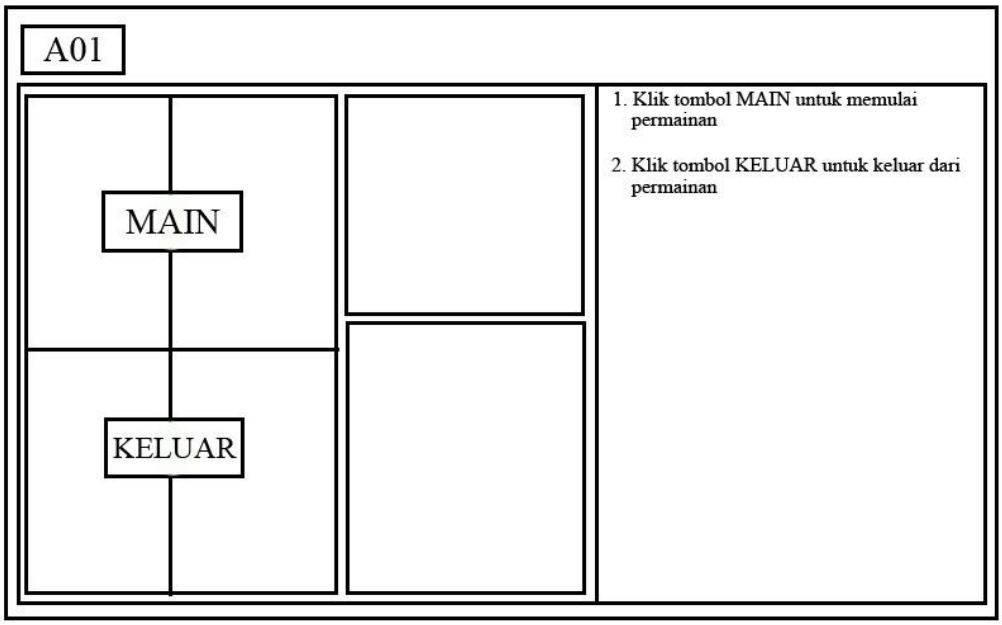
Perancangan sistem merupakan kegiatan untuk merancang aplikasi yang akan dibangun, tahapannya dimulai dari perancangan komponen permainan, perancangan antarmuka, perancangan pesan dan jaringan semantik dari *game* tenis meja.

3.7.1 Perancangan antarmuka

Perancangan antarmuka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun, 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

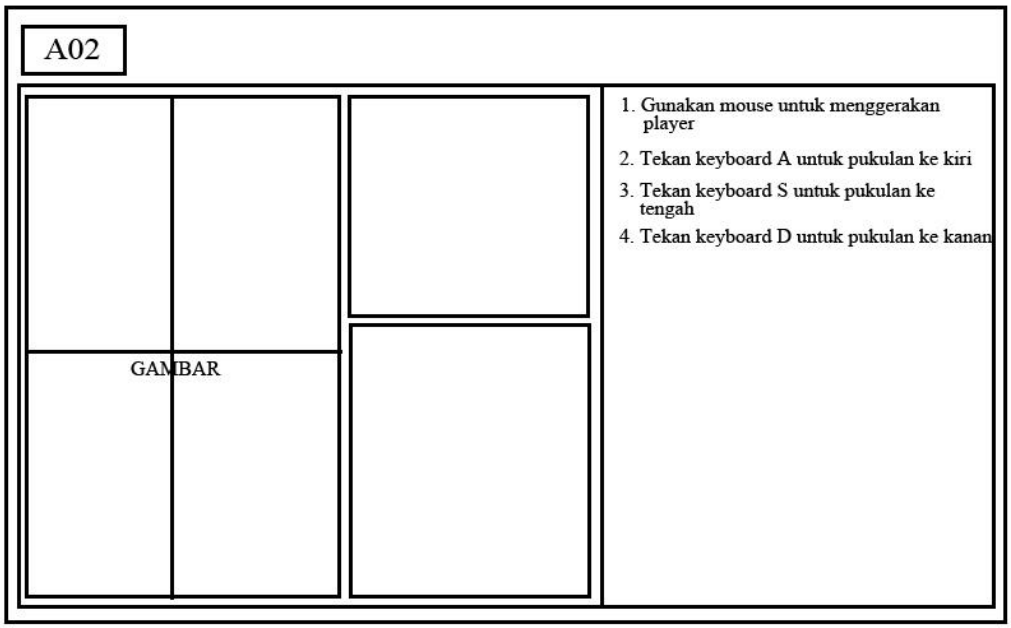
Menu utama merupakan menu yang akan muncul pertama kali pada saat pengguna membuka *game*. Gambar 3.15 menunjukan perancangan antarmuka menu utama.



Gambar 3.15 Antarmuka menu utama

3.7.1.2 Perancangan Antarmuka Menu Main

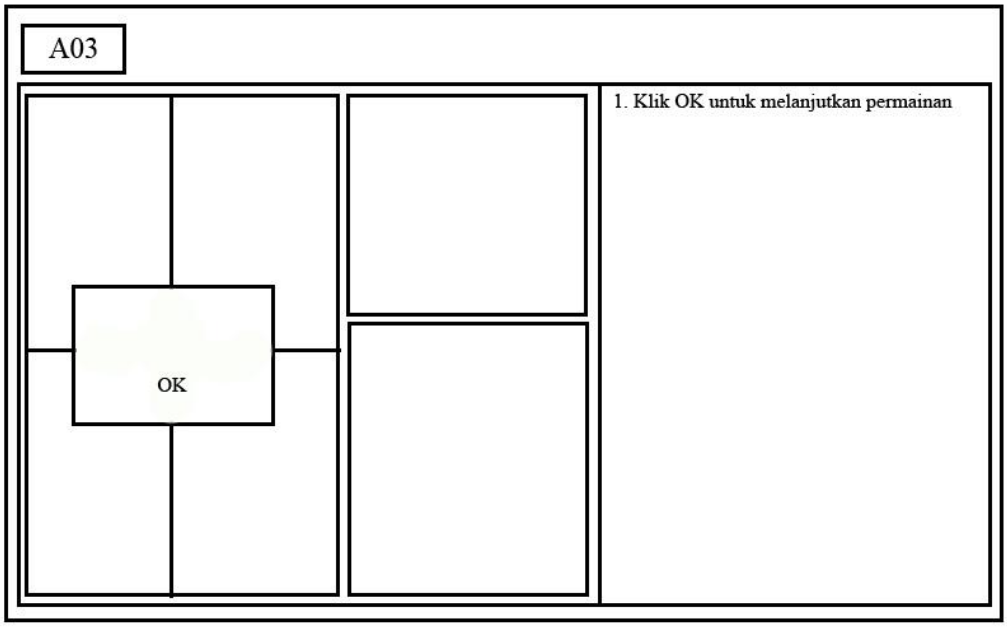
Antarmuka menu main merupakan perancangan antarmuka yang menampilkan permainan *game*. Gambar 3.16 merupakan perancangan antarmuka menu main.



Gambar 3.16 Antarmuka menu main

3.7.1.3 Perancangan Antarmuka *Point*

Antarmuka menu *point* merupakan perancangan antarmuka yang menampilkan *point*. Gambar 3.17 merupakan perancangan antarmuka *point*.



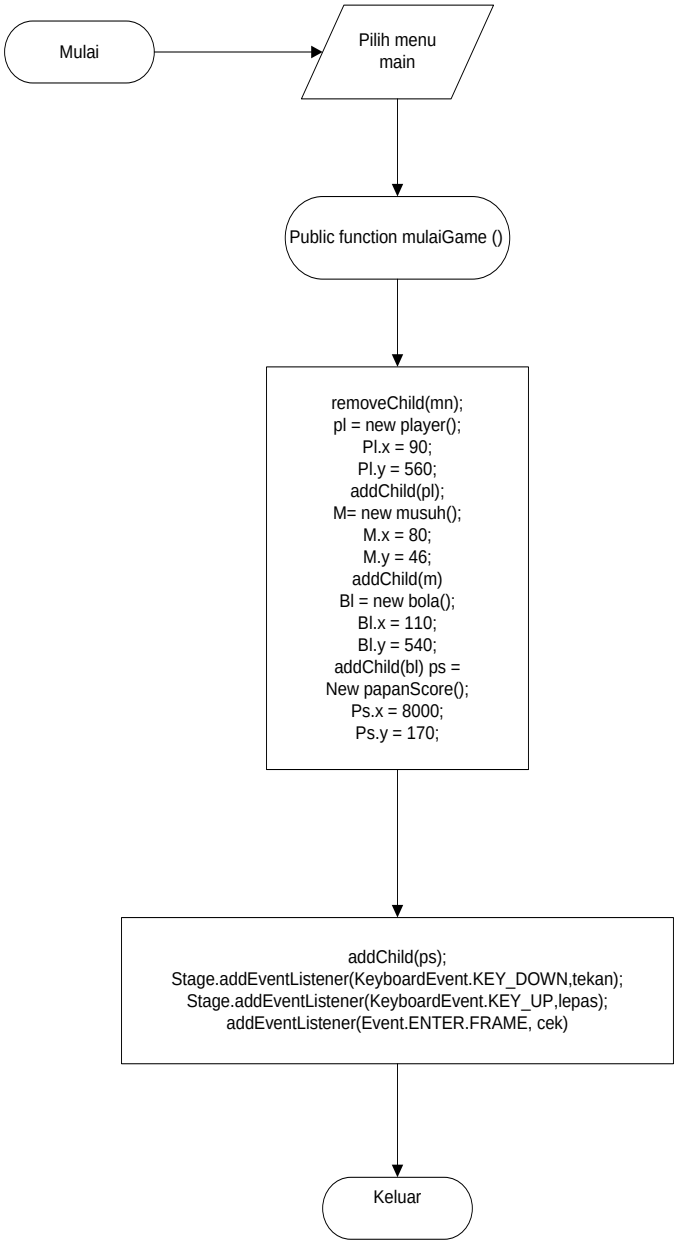
Gambar 3.17 Antarmuka *Point*

3.8 Perancangan *Method*

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

1. Perancangan *Method* Main

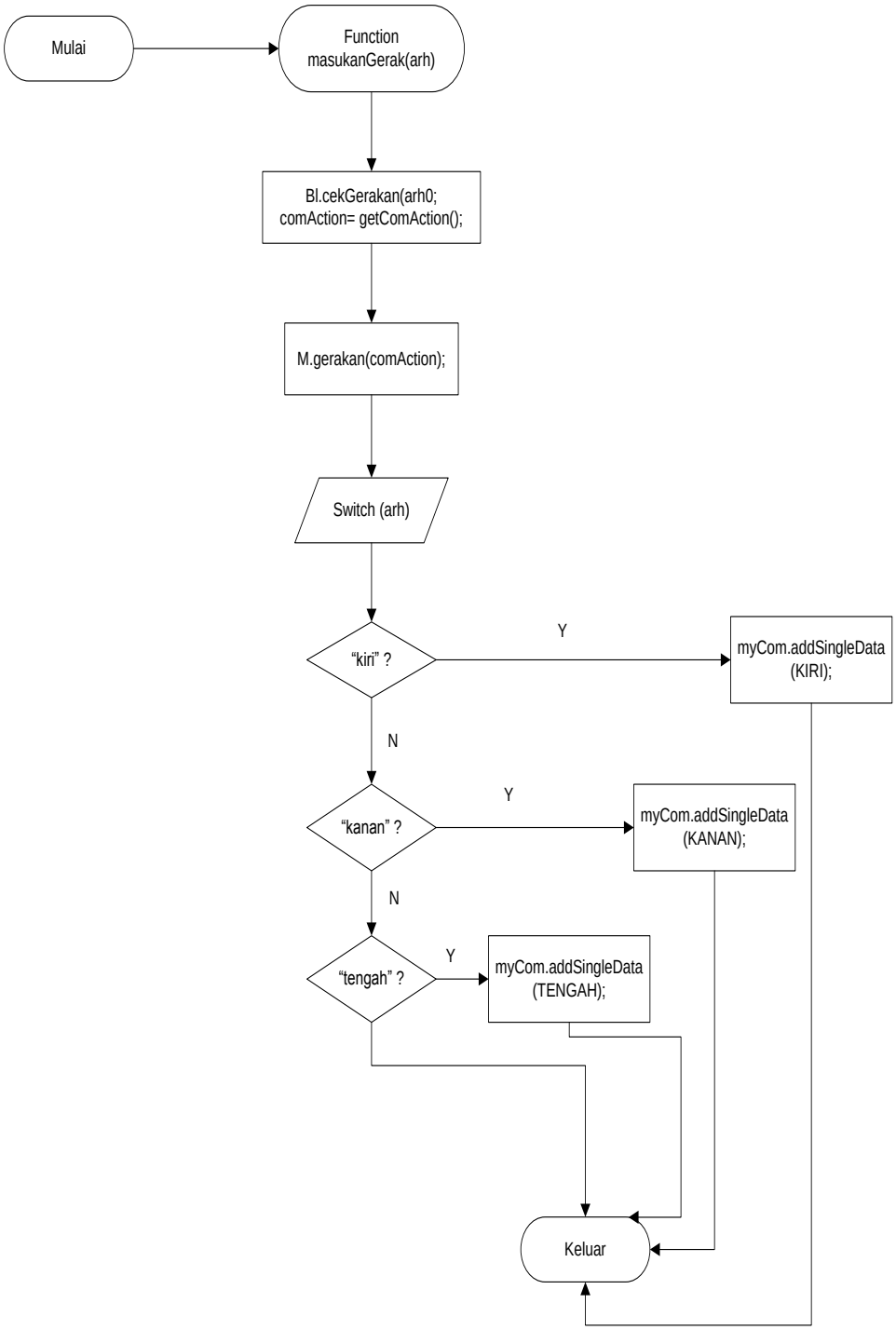
Gambar 3.18 di bawah ini adalah perancangan *method* main.



Gambar 3.18 Perancang *Method* Main

2. Perancangan *Method* Pukulan Bola

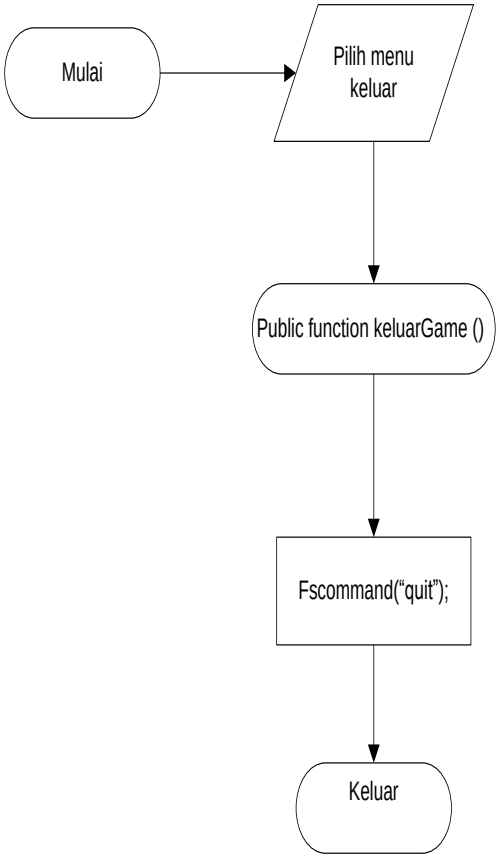
Gambar 3.19 di bawah ini adalah perancangan *method* pukulan bola.



Gambar 3.19 Perancang *Method* Pukulan Bola

3. Perancangan *Method* Keluar

Gambar 3.20 di bawah ini adalah perancangan *method* keluar.



Gambar 3.20 Perancangan *Method* Keluar