BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Analisis sistem merupakan penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya yang dimaksudkan untuk mengidentifikasi dan mengevaluasi segala permasalahan dan hambatan yang terjadi serta kebutuhan yang diharapkan dapat menjadi acuan untuk diusulkannya perbaikan-perbaikan.

Tahap analisis sistem ini sangat penting karena apabila terjadi kesalahan dalam tahap ini akan mengakibatkan kesalahan pada tahap selanjutnya, untuk itu diperlukan tingkat ketelitian dan kecermatan yang tinggi untuk dapat mendapatkan kualitas kerja sistem yang baik. Untuk dapat melakukan analisis sistem maka dilakukan beberapa buah langkah yaitu sebagai berikut :

1. Analisis Masalah
2. Analisis *Game* Sejenis
3. Analisis *Game* yang Dibangun
4. Analisis Kebutuhan Non Fungsional
5. Analisis Kebutuhan Fungsional

3.1.1 Analisis Masalah

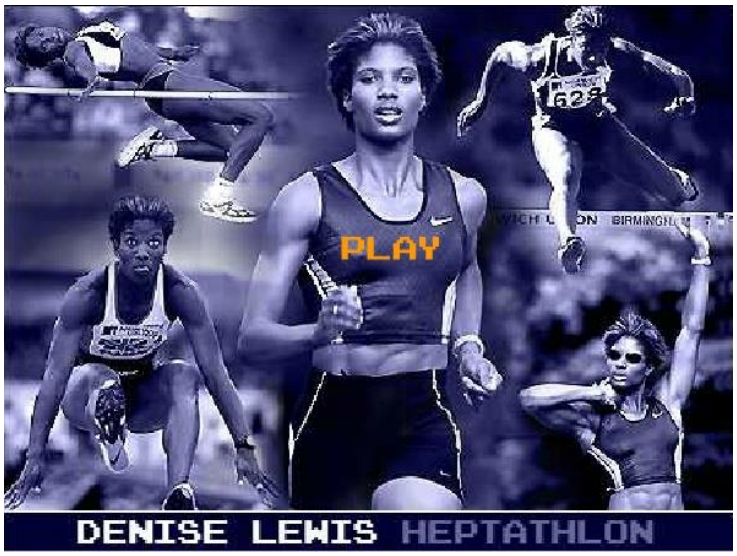
Berdasarkan latar belakang yang telah diuraikan maka masalah yang didapat adalah kurangnya pengetahuan anak-anak terhadap peraturan olahraga atletik, hal ini berdasarkan kuesioner yang telah dilakukan.

3.1.2 Analisis *Game* Sejenis

Analisis *game* sejenis merupakan analisis yang akan membahas mengenai *game* yang menjadi acuan dalam pembangunan sebuah *game*. Dalam *game* sejenis akan dibahas mengenai berbagai hal yang ada di dalam *game* sejenis diantaranya meliputi *storyline*, *gameplay*, serta komponen-komponen apa saja yang ada dalam *game* tersebut sehingga dapat menjadi acuan bagi kebutuhan yang akan dibangun. Berikut sedikit ulasan mengenai *game* yang akan dibahas yaitu *game* Heptathlon dan *game* Summer Games 2005.

3.1.2.1 Pengenalan *Game* Heptathlon

Heptathlon adalah permainan olahraga atletik, *game* ini merupakan permainan atletik yang terdiri dari *sprint*, lari gawang, lompat jauh, lempar lembing dan lompat tinggi. Disini pemain mainkan satu karakter saja tanpa bisa memilih karakter lain untuk digunakan



Gambar 3.1 Tampilan *Game* Heptathlon

1. ***Storyline Game* Heptathlon**

   Game Heptathlon adalah sebuah permainan yang menceritakan tentang olahraga atletik yang memainkan 7 cabang olahraga seperti lari gawang, *sprint* jarak 200M, lompat jauh, lompat tinggi, lempar lembing, lempar cakram, dan lari jarak 800M. Pemain akan mengikuti cabang olahraga ini dari pertama hingga akhir dari permainan.

2. ***GamePlay Game* Heptathlon**

   Berikut ini adalah penjelasan *gameplay Game* Heptathlon :

   a. Pemain tidak bisa memilih karakter lain.
   b. Permainan ini seperti berada dalam kompetisi cabang atletik dimana pemain mengikuti semua cabang permainan yang tersedia dalam *game* ini dari pertama mulai hingga selesai.
   c. Pemain akan memainkan 7 cabang olahraga seperti lari gawang, *sprint* jarak 200M, lompat jauh, lompat tinggi, lempar lembing, lempar cakram, dan lari jarak 800M
   d. Pemain menggunakan *keyboard* untuk bermain, dalam *game* ini tombol yang digunakan adalah N dan B untuk berlari dan tombol spasi untuk melompat atau melempar.

3.1.2.2 Pengenalan *Game* Summer Games 2005

Game Summer Games 2005 adalah *game* yang hampir sama dengan *game* Heptathlon yang menerapkan unsur olahraga atletik, tetapi dalam *game* ini pemain bisa memilih negara mana yang akan mereka gunakan namun jenis permainannya tidak sebanyak *game* Heptathlon.



Gambar 3.2 Tampilan Summer Games 2005

1. ***Storyline Game* Summer Games 2005**

   Summer Games 2005 ini adalah sebuah permainan yang menceritakan olahraga atletik yang memainkan 4 cabang olahraga atletik seperti lari sprint 100M, lompat tinggi, lempar lembing dan lari gawang 110M.

2. ***GamePlay Game* Summer Games 2005**

   Berikut ini adalah penjelasan *gameplay Game* Summer Games 2005:

   a. Permainan ini merupakan sebuah permainan olahraga atletik yang hanya bisa dimainkan oleh satu pemain.
   b. Pemain bisa memilih negara sebagai karakter yang akan digunakan.
   c. Pemain akan memainkan 4 cabang olahraga dari mulai lari 100M, lompat tinggi, lempar lembing dan lari gawang 110M.
   d. Pemain akan bertanding dengan waktu atau jarak yang ditentukan untuk dicapai untuk bisa melanjutkan pada permainan selanjutnya.
   e. Bila pemain kalah, maka permainan berakhir dan pemain harus mengulangginya dari awal.

f. Tombol yang digunakan adalah spasi untuk berlari dan ctrl untuk melompat atau melempar

3.1.2.3 Kesimpulan Dari *Game* Sejenis Terhadap *Game* yang Akan Dibangun

Secara garis besar perbandingan *game* sejenis terhadap *game* yang akan dibangun akan dijelaskan pada tabel 3.1.

Tabel 3.1 Perbandingan *Game* Sejenis Dengan *Game* yang Akan Dibangun

	***Game* Heptathlon**	***Game* Summer Games 2005**	***Game* yang akan dibangun**
Desain Grafis	2D	2D	2D
Genre *Game*	*Sport*	*Sport*	*Sport*
Storyline	Alur cerita yang menggambarkan tentang seorang legenda olahraga atletik yang mengikuti sebuah perlombaan	Alur cerita yang menggambarkan tentang pertandingan olahraga atletik namun peraturan tidak sesuai dengan aslinya	Alur cerita menggambarkan tentang perlombaan atletik dengan perarturan sama seperti aslinya
Gameplay	-pemain harus mengikuti alur permainan tidak peduli menang atau kalah. -tidak terdapat materi tentang cara bermain yang sesungguhnya	-pemain harus mengikuti alur permainan namun bila kalah pemain harus kembali memulai dari awal lagi. -tidak terdapat materi tentang cara bermain yang sesungguhnya	-pemain bebas memilih permain yang ingin dimainkan -terdapat materi tentang cara bermain yang sesungguhnya
Basis	Desktop	Desktop	Desktop

Setelah menganalisis *game* sejenis, maka *game* yang akan dibangun memiliki kelebihan pada pengetahuan yang didapat pemainnya berbeda dengan *game* Heptathlon dan Summer Games 2005 yang hanya menampilkan perlombaan permainan saja tanpa memberikan nilai-nilai informasi dalam olahraga atletik tersebut.

3.1.2.4 Rangkuman Hasil Analisis Terhadap *Game* yang Akan Dibangun

Dari kesimpulan tersebut maka dapat dirangkum :

1. *Game* yang akan dibangun memiliki kesamaan dengan *game* sejenis yang dianalisis yaitu menggunakan animasi 2D, namun perbedaan terdapat pada *gameplay* yang akan dibangun, pada *game* atletik ini pemain bebas memilih akan memainkan permainan yang tersedia pada menu, sedangakan pada *game* Heptathlon dan Summer Games 2005 pemain mengharuskan mengikuti alur permainan dari awal sampai akhir.
2. Pada *Game* Olahraga Atletik terdapat penyajian materi tentang olahraga tersebut yang dapat menambah pengetahuan pemainnya. Berbeda dengan *game* heptathlon dan summer games 2005.

Maka keunggulan dari *game* olahraga atletik ini adalah terdapatnya beberapa penyampaian materi tentang cara melakukan olahraga atletik tersebut dimana dapat menambah wawasan dan pengetahuan para pemainnya.

3.1.2.5 Analisis *Game* Olahraga Atletik

Game Olahraga Atletik adalah *game* yang bertema tentang olahraga atletik, *game* ini mengangkat kegiatan atletik dimana pemain akan bermain sebagai atlet yang mengikuti beberapa cabang pertandingan olahraga.

Pertama pemain akan memilih permainan apa yang ingin dimainkan. Pemain bebas memilih permainan yang ingin dimainkan seperti *sprint*, Lari gawang, Lompat Tinggi, Lompat Jauh, dan Lempar Lembing.

3.1.2.6 *Storyline*

Game atletik adalah sebuah permainan yang mempunyai sisi edukatif dimana terdapat informasi tentang seputar olahraga cabang atletik. Terdapat beberapa atlet yang ikut serta dalam kegiatan ini dan pada permainan ini tidak semua cabang menjadi sorotan, hanya cabang atletik saja yang akan dibahas dalam permainan ini. Cabang olahraga yang ada dalam permainan ini seperti cabang olahraga *sprint*, lari gawang,

lompat tinggi, lompat jauh, dan lempar lembing. Pemain bebas memilih permainan mana saja yang ingin dipilih tanpa harus mengikuti alur permain. Setiap permainan memiliki tantangan, pada *game sprint* pemain akan beradu lari dengan NPC untuk berlari mencapai garis *finish*. Pada *game* lari gawang pemain akan dihadapkan dengan rintangan berupa gawang yang menghalangi laju para atlet untuk mencapai garis *finsih*. Pada permainan lompat tinggi pemain diharuskan melewati mistar pada area lompat tinggi, lompat jauh dan lempar lembing menggunakan skor untuk pencapaian yang diraih oleh para atlet.

1. *Sprint*

   Ada 8 atlet yang bersiap beradu lari dalam sebuah lintasan atletik, 1 adalah pemain dan 7 diantaranya adalah NPC. Dimulai dengan aba-aba “bersedia”, “siap”,dan “mulai” para atlet tersebut semua berlari berlomba-lomba untuk menjadi yang pertama menyentuh garis finish.

2. Lari Gawang

   Ada 8 atlet yang bersiap beradu lari dalam sebuah lintasan atletik, 1 adalah pemain dan 7 diantaranya adalah NPC. Dimulai dengan aba-aba “bersedia”, “siap”,dan “mulai” para atlet tersebut semua berlari berlomba-lomba menjadi yang pertama untuk menyentuh garis finish namun ada rintangan berupa gawang yang harus dilewati oleh semua atlet, untuk menghindarinya para atlet harus mampu melompatinya.

3. Lompat tinggi

   Pemain akan melakukan sebuah lompatan pada sebuah matras dimana ada halang rintangan berupa mistar yang dipasangakan diantara 2 tiang, pemain harus bisa melompat tinggi untuk melewatinya mistar tersebut.

4. Lompat Jauh

   Pemain akan melakukan sebuah lompatan yang jauh pada sebuah area pasir sebagai landasannya.

5. Lempar Lembing

   Pemain akan berlari membawa sebuah lembing dan melempar lembingnya sejauh mungkin.

3.1.2.7 *GamePlay Game* Olahraga Atletik

Untuk masing-masing *game* memiliki *gameplay* yang berbeda. Berikut penjelasan tentang *gameplay* dari masing-masing *game* Olahraga Atletik.

A. *Sprint*

1. Pemain akan berlomba lari dengan 7 NPC dalam permainan untuk memenangkan pertandingan.
2. Cara bermain menggunakan tombol tanda panah kanan dan kiri pada *keyboard* untuk berlari.

B. Lari Gawang

1. Pemain akan berlomba lari gawang dengan 7 NPC lainnya untuk memenangkan pertandingan.
2. Cara bermain menggunakan tombol tanda panah kanan dan kiri pada *keyboard* untuk berlari dan *space* untuk melompat.
3. Pemain akan menghadapi rintangan berupa gawang pada saat berlari, bila pemain menabrak gawang otomatis kecepatan akan berkurang.

C. Lompat Tinggi

1. Pemain akan melakukan lompat tinggi terhadap matras tebal yang dibatasi oleh mistar.
2. Cara bermain menggunakan tombol tanda panah kanan dan kiri pada *keyboard* untuk berlari dan tombol *space* untuk melompat.

D. Lompat Jauh

1. Pemain akan melakukan lompat jauh terhadap pasir.
2. Pemain harus melakukan lompatan yang sangat jauh.

3. Cara bermain menggunakan tombol tanda panah kanan dan kiri pada *keyboard* untuk berlari dan tombol *space* untuk melompat.

E. Lempar Lembing

1. Pemain akan melakukan lemparan lembing.
2. Cara bermain menggunakan tombol tanda panah kanan dan kiri pada *keyboard* untuk berlari dan tombol *space* untuk melempar.
3. Pemain harus melakukan lemparan yang sangat jauh.

3.1.2.5 Analisis *Scoring*

Pada *Game* Olahraga Atletik ini memiliki skor pada cabang olahraga lompat jauh dan lempar lembing. Skor berupa jarak dimana pemain bisa melompat atau melempar sejauh mungkin.

1 Lompat Jauh

Dalam permainan lompat jauh skor yang di dapat pemain berupa jarak yang diperoleh melalui lompatan pemain, dimana pemain akan melompat pada pasir. Semakin jauh pemain melompat akan semakin tinggi skornya.

2 Lempar Lembing

Dalam permainan lempar lembing skor yang di dapat pemain berupa jarak yang diperoleh melalui lemparan lembing yang dilakukan oleh pemain. Semakin jauh pemain melempar akan semakin tinggi skornya.

3.1.3 Analisis Algoritma *Fuzzy Logic*

Pada *game* ini *Fuzzy Logic* digunakan untuk mengendalikan NPC. Berikut pemaparan jelas tentang Algoritma *Fuzzy Logic* dan tahapan – tahapannya sebagai berikut.

1. Variable *Fuzzy Logic*

Ada 3 variable yang digunakan dalam fungsi *fuzzy*, yaitu sebagai berikut :

1. Jarak

2. Stamina
3. Kecepatan

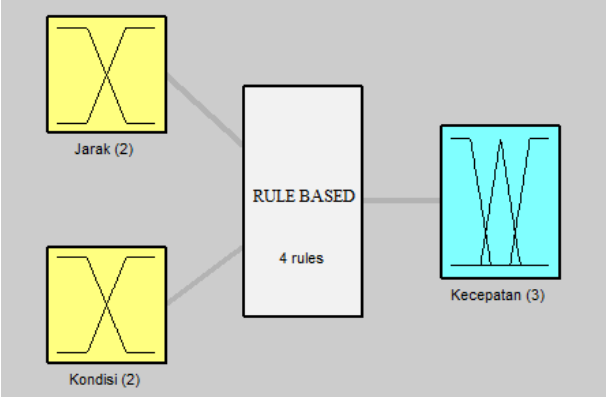
2. Nilai Linguistik

Dari 3 variable yang digunakan memiliki nilai linguistik sebagai berikut :

1. Jarak : dekat, jauh
2. Stamina : fit, cedera
3. Kecepatan : lambat, sedang, cepat

3. Fuzzyfikasi

Fuzzyfikasi merupakan proses memetakan nilai crisp (numerik) kedalam himpunan *fuzzy* dan menentukan derajat keanggotaannya. Secara garis besar pemetaan nilai crisp ke dalam himpunan *fuzzy* dijelaskan dengan gambar 3.3.

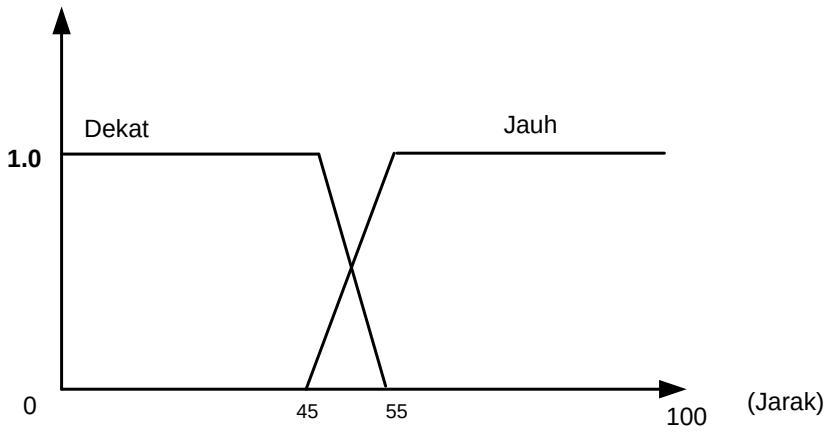


Gambar 3.3 *fuzzy Interface System*

Berdasarkan *fuzzy interface system* tersebut maka pemetaan input – output adalah sebagai berikut:

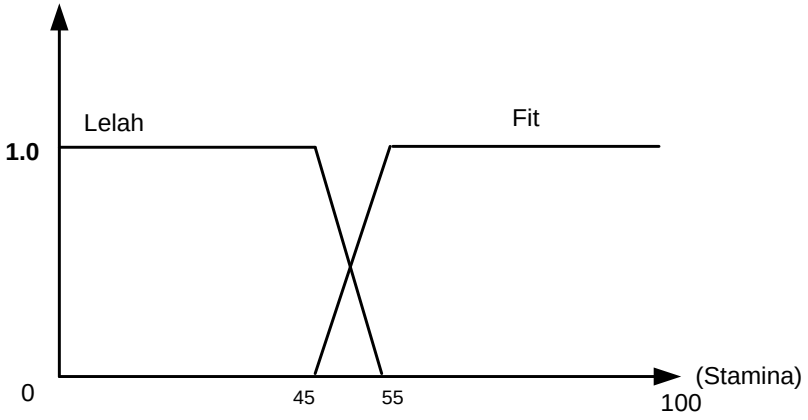
1. Variabel Jarak, terbagi menjadi 2 input yaitu dekat dan jauh. Range nilai untuk variabel jarak antara 0-100 pixel dengan penjelasan sebagai berikut :

   1 Dekat = 0-55
   2 Jauh = 45-100



Gambar 3.4 Grafik Variabel Jarak

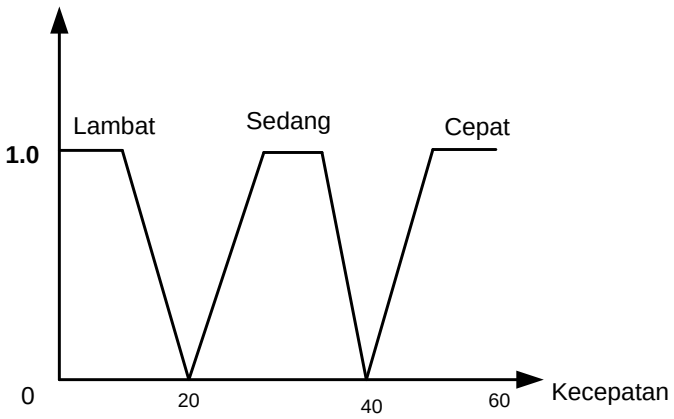
2. Variable Stamina, terbagi menjadi 2 input yaitu fit dan lelah. Range nilai untuk variabel stamina diantara 0-100 dengan penjelasan sebagai berikut :
   1 Lelah = 0-55
   2 Fit = 45-100



Gambar 3.5 Grafik Variabel Stamina

3. Variable Kecepatan, terbagi menjadi 3 output yaitu lambat, sedang dan cepat. Range nilai untuk variabel kecepatan diantara 0-60 dengan penjelasan sebagai berikut :
   1 Lambat = 0-20

2 Sedang = 20-40
3 Cepat = 40-60



Gambar 3.6 Grafik Variabel Kecepatan

4. Proses Inferensi

Proses inference memperhitungkan semua aturan yang ada dalam basis pegetahuan. Hasil proses inferensi adalah sebagai berikut :

1 IF Jarak is Dekat AND Stamina is Lelah THEN Kecepatan is Lambat
2 IF Jarak is Dekat AND Stamina is Fit THEN Kecepatan is Sedang
3 IF Jarak is Jauh AND Stamina is Lelah THEN Kecepatan is Sedang
4 IF Jarak is Jauh AND Stamina is Fit THEN Kecepatan is Cepat

5. Proses Defuzzyfikasi

Dengan menggunakan *weight Average* untuk proses *defuzzyfication* menggunakan rumus

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

Dimana y adalah nilai crips dan $\mu(y)$ adalah derajat keanggotaan dari nilai crips y.

3.1.4 Contoh Kasus

Untuk *game sprint* dan lari gawang akan menggunakan Algoritma *Fuzzy Logic*. Algoritma *Fuzzy Logic* digunakan untuk mengendalikan NPC. Berikut pemaparan jelas tentang Algoritma *Fuzzy Logic* yang digunakan untuk menyelesaikan *game sprint* dan *game* lari gawang :

1. Variable *Fuzzy Logic*

Ada 3 variable yang digunakan dalam fungsi *fuzzy*, yaitu sebagai berikut :

1. Jarak antara *player* dengan NPC
2. Stamina
3. Kecepatan yang diambil NPC

3. Nilai Linguistik

Dari 3 variable yang digunakan memiliki nilai linguistik sebagai berikut :

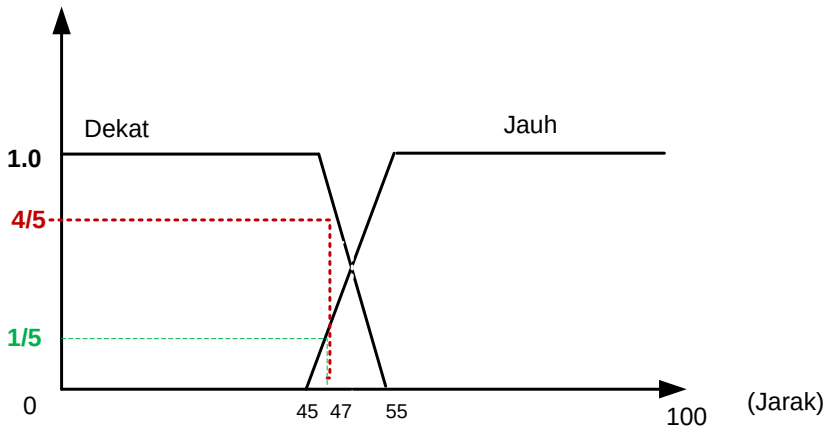
1. Jarak : dekat, jauh
2. Stamina : fit, cedera
3. Kecepatan : lambat, sedang, cepat

5. Fuzzyfikasi

1. Variabel Jarak, terbagi menjadi beberapa input yaitu dekat dan jauh. Range nilai untuk variabel jarak antara pemain dan NPC diantara 0-100 pixel dengan penjelasan sebagai berikut :

   Misalkan didapatkan nilai jarak sebesar 47 pixel, pada proses ini maka nilai crips input Jarak 47 pixel dikonversi ke nilai fuzzy dengan cara:

   1 Jarak 47 pixel berada pada nilai linguistik dekat dan jauh.
   2 Derajat keanggotaan untuk Dekat dihitung menggunakan rumus = -(x – d) / (d – c), $c < x \leq d$, dimana c = 45 dan d = 55 sehingga derajat keanggotaan Dekat = - (47 – 55) / (55 – 45) = 8/10= 4/5
   3 Derajat keanggotaan untuk Jauh menggunakan rumus (x –a) / (b-a) , $a < x < b$, dimana a=45 dan b= 55 sehingga derajat keanggotaan Jauh = (48 - 45) / (55 - 45) = 2/10 =1/5

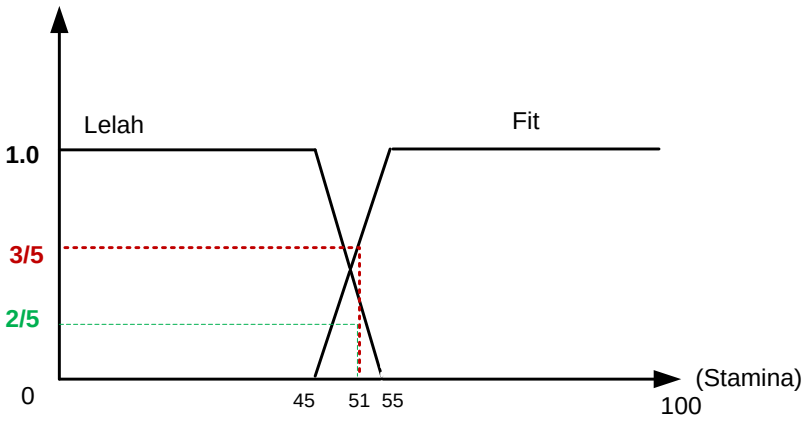


Gambar 3.7 Grafik Variabel Jarak Dengan Input 47

4. Variable Stamina, terbagi menjadi beberapa input yaitu fit dan lelah. Range nilai untuk variabel stamina diantara 0-100 dengan penjelasan sebagai berikut :

   Misalkan didapatkan nilai Stamina sebesar 51 didapatkan dari hasil random. Pada proses ini maka nilai crips input Stamina 51 dikonversi ke nilai fuzzy dengan cara:

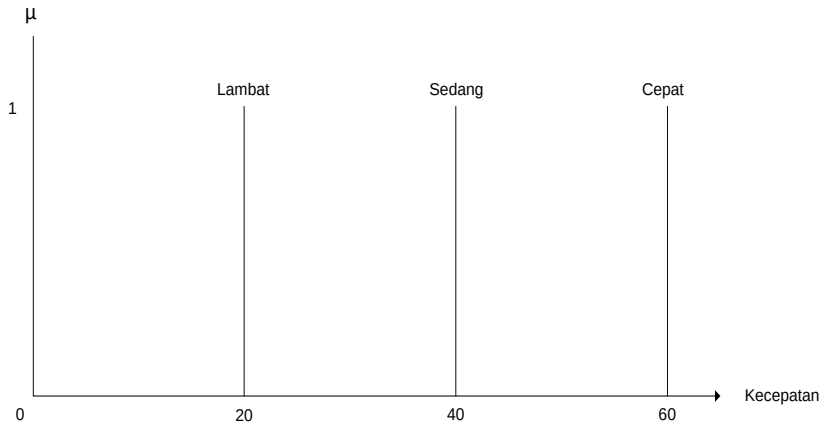
   1 Stamina berada pada nilai linguistik Lelah dan Fit.
   2 Derajat keanggotaan untuk Lelah dihitung menggunakan rumus = -(x – d) / (d – c), $c < x \leq d$, dimana c = 45 dan d = 55 sehingga derajat keanggotaan Lelah = - (51 – 55) / (55 – 45) = 4/10= 2/5
   3 Derajat keanggotaan untuk Fit menggunakan rumus (x –a) / (b-a) , $a < x < b$, dimana a=45 dan b= 55 sehingga derajat keanggotaan Fit =(51- 45) / (55 - 45) = 6/10 = 3/5



Gambar 3.8 Grafik Variabel Stamina Dengan Input 51

6. Proses Inferensi

Dengan menggunakan fungsi keanggotaan *singleton* yaitu fungsi keanggotaan yang memiliki derajat keanggotaan 1 pada suatu nilai crips tunggal dan 0 pada semua nilai crips yang lain. Grafik variabel kecepatan terdapat pada gambar 3.9 berikut.



Gambar 3.9 Grafik Variabel Kecepatan

Aturan-aturan yang diterapkan dalam penentuan perilaku NPC ada 4 *rules*, dapat dilihat pada tabel 3.2 berikut :

Tabel 3.2 *Rule-Based* NPC

	JARAK		STAMINA		KECEPATAN
IF	Dekat	And	Fit	Then	Sedang
IF	Dekat	And	Lelah	Then	Lambat
IF	Jauh	And	Fit	Then	Cepat
IF	Jauh	And	Lelah	Then	Sedang

Dari 4 aturan fuzzy dan 4 fuzzy input tersebut, maka proses *inference* yang terjadi adalah sebagai berikut :

5 IF Jarak is Dekat (4/5) AND Stamina is Lelah (2/5) THEN Kecepatan is Lambat (2/5)
6 IF Jarak is Dekat (4/5) AND Stamina is Fit (3/5) THEN Kecepatan is Sedang (3/5)
7 IF Jarak is Jauh (1/5) AND Stamina is Lelah (2/5) THEN Kecepatan is Sedang (1/5)
8 IF Jarak is Jauh (1/5) AND Stamina is Fit (2/5) THEN Kecepatan is Cepat (1/5)

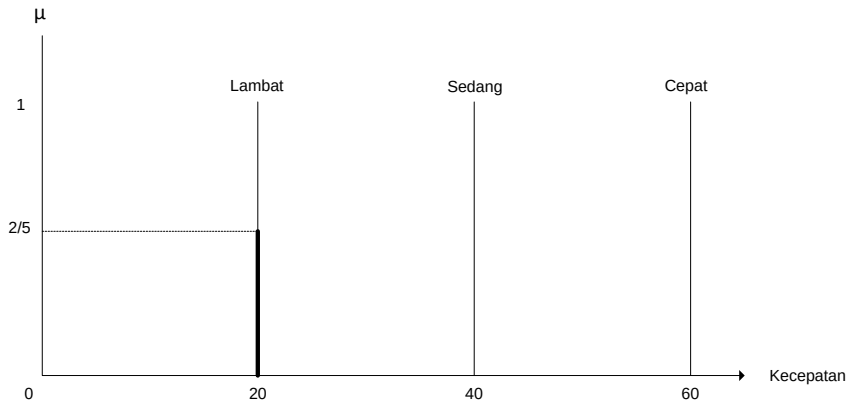
Dengan demikian diperloleh 3 pernyataan :

1. Kecepatan is Lambat (2/5)
2. Kecepatan is Sedang (3/5)
3. Kecepatan is Cepat (1/5)

Proses inferensi menggunakan model sugeno mengghasilkan 3 derajat keanggotaan yaitu :

1. Derajat keanggotaan lambat

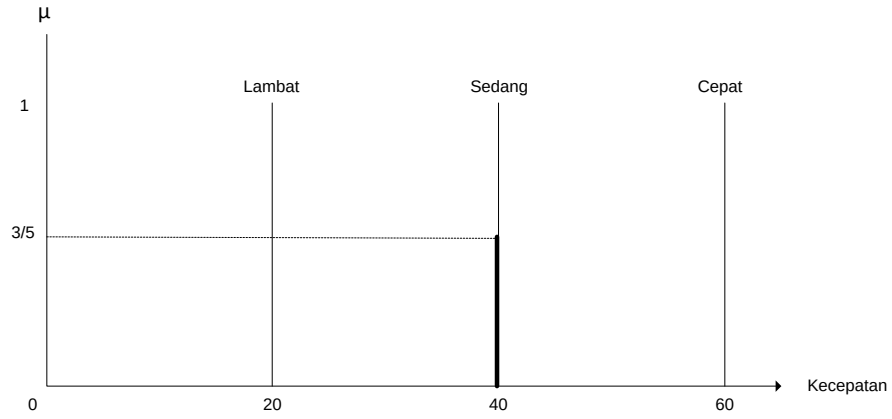
   Berikut adalah gambar derajat keanggotaan lambat pada gambar 3.10.



Gambar 3.10 Kecepatan is Lambat

2. Derajat keanggotaan Sedang

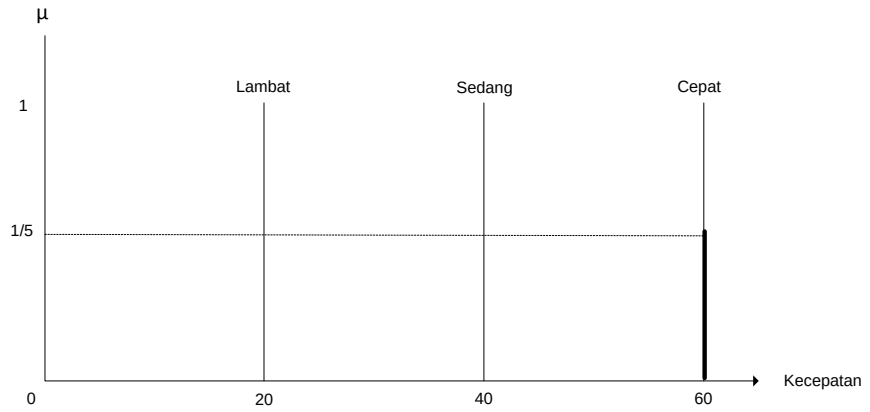
   Berikut adalah gambar derajat keanggotaan sedang pada gambar 3.11.



Gambar 3.11 Kecepatan is Sedang

3. Derajat keanggotaan Cepat

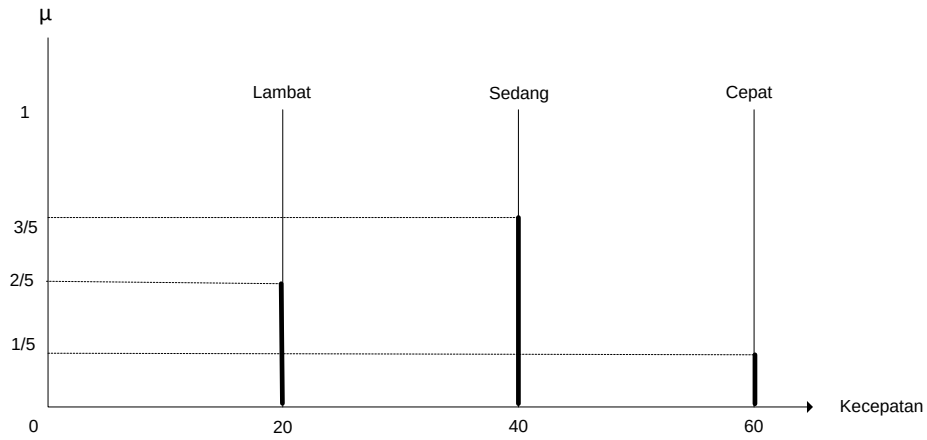
   Berikut adalah derajat keanggotaan cepat pada gambar 3.12.



Gambar 3.12 kecepatan is Cepat

2. Proses Defuzzyfikasi

Proses *composition* dari tiga fuzzy set, kecepatan is Lambat (2/5), kecepatan is Sedang (3/5), dan kecepatan is Cepat (1/5) menghasilkan satu fuzzy set tunggal yang ditunjukan pada gambar 3.13 berikut.



Gambar 3.13 Proses Composition

Dengan menggunakan *weight Average* untuk proses *defuzzyfication* menghasilkan nilai

$$y^* = \frac{2/5(20) + 3/5(40) + 1/5(60)}{2/5 + 3/5 + 1/5} = 31{,}42$$

Jadi keputusan output yang diambil pada Kecepatan bernilai 31,42 (Sedang).

3.1.4.1 *Pseudocode* Algoritma *Fuzzy Logic*

Berikut adalah *pseudocode* Algoritma *Fuzzy Logic* :

Algoritma *Fuzzy Logic*

{Algoritma untuk memberikan keputusan kepada NPC }

```
Deklarasi
        jarak : int
        stamnia : int
        kecepatan : int
        posisi NPC : int
        posisi pemain : int
Algoritma
        read (jarak)
        if jarak > 55 then
          write (‘jauh’)
        else
        if jarak < 45 then
          write (‘dekat’)
        endif
        endif
        read (jarak) ← (posisi NPC – posisi pemain)
        read (stamina)
        if stamina < 45 then
          write (‘lelah’)
        else
        if stamina > 55 then
          write (‘fit’)
        endif
        endif
        if defuzzyfikasi < 20 then
          write (‘lambat’)
        else
        if defuzzyfikasi > 20 and < 40 then
          write (‘sedang’)
        else
```

```
if defuzzyfikasi > 40 then
  write (‘cepat’)
endif
endif
endif
```

3.1.5 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non-fungsional menggambarkan kebutuhan luar sistem yang diperlukan untuk menjalankan aplikasi yang dibangun. Adapun kebutuhan non-fungsional pada *game* Olahraga Atletik ini meliputi kebutuhan perangkat keras, kebutuhan perangkat lunak dan pengguna sistem yang akan memakai aplikasi.

3.1.5.1 Analisis Kebutuhan Perangkat Keras

Analisis Perangkat keras atau *hardware* merupakan salah satu hal yang penting karena tanpa *hardware* yang memenuhi syarat, program yang akan dibuat tidak akan dapat berjalan. Berikut spesifikasi standar perangkar keras yang dapat dipergunakan untuk membangun aplikasi *game* ini dan spesifikasi perangkat keras yang dapat dipergunakan untuk menjalankan aplikasi *game* ini terdapat pada tabel 3.3.

Tabel 3.3 Kebutuhan Perangkat Keras Pembangun

No	Perangkat Keras	Spesifikasi
1	Processor	Intel 2.30 GHz
2	Monitor	14 inch
3	Graphic Card	VGA 2 GB
4	Hard disk drive	500 GB
5	Memory	RAM 4 GB
6	Keyboard dan Mouse	Standar
7	Speaker	Standar

Pembangunan aplikasi yang akan dibangun membutuhkan spesifikasi perangkat keras, spesifikasi minimum perangkat keras untuk menjalankan aplikasi ini dapat dilihat pada tabel 3.4.

Tabel 3.4 Kebutuhan Perangkat Keras Pengguna

No	Perangkat Keras	Spesifikasi
1	Processor	Processor dengan kecepatan 1.8 Ghz
2	Monitor	14 inch
3	Graphic Card	VGA 512 MB
4	Hard disk drive	Free Space 500 MB
5	Memori	RAM 512 MB
6	Keyboard dan Mouse	Standar
7	Speaker	Standar

3.1.5.2 Analisis Kebutuhan Perangkat Lunak

Analisis perangkat lunak atau *software* merupakah 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 keduannya. Perangkat lunak yang dapat dibutuhkan untuk membangun aplikasi *Game* Olahraga Atletik ini dapat dilihat pada tabel 3.5 berikut.

Tabel 3.5 Kebutuhan Perangkat Lunak yang Dibutuhkan

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Microsoft Windows XP, Windows 7
2	Tool Pembangun	Unity
3	Tool Desain	Adobe Photoshop CS5

3.1.5.3 Analisis Kebutuhan Pengguna

Analisis kebutuhan pengguna merupakan analisis terhadap *user* yang akan menggunakan sistem yang akan dibangun. Adapun kebutuhan pengguna yang terlibat dalam penggunaan *Game* Olahraga Atletik ini dapat dilihat pada tabel 3.6 berikut.

Tabel 3.6 Analisis Kebutuhan Pengguna

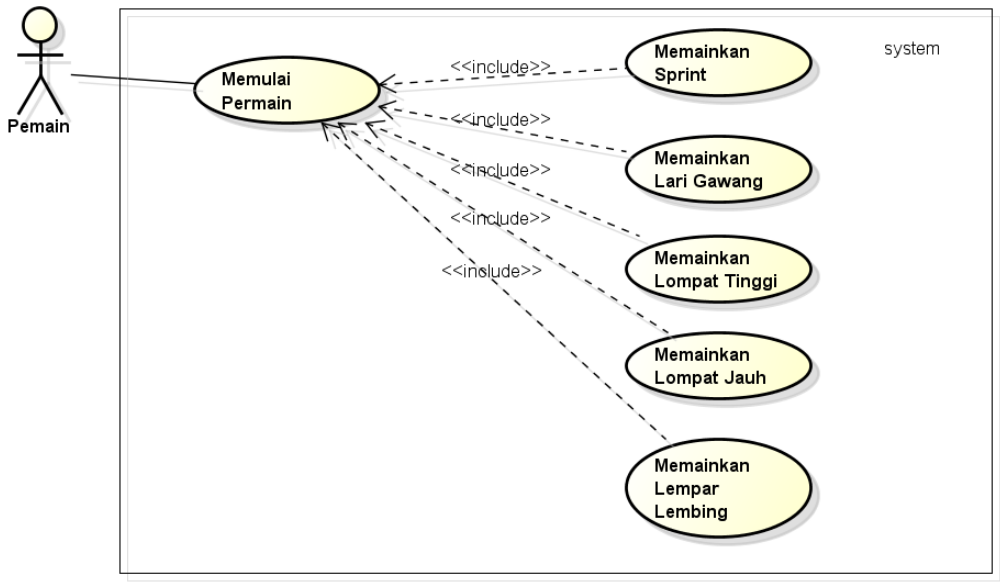
Pengguna	Anak-anak
Hak Akses	Memainan *game*
Tingkat Usia	9 tahun keatas
Tingkat Sekolah	Sekolah Dasar
User Experience	Dapat menggunakan komputer

3.1.6 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan kebutuhan sistem yang akan dibangun pada *Game* Olahraga Atletik. Adapun kebutuhan fungsional pada *game* yang akan dibangun ini dengan pemodelan berorientasi objek. Pemodelan berorientasi objek merupakan perangkat lunak yang gunakan pada pembanguan *game* ini.

3.1.6.1 *Use Case* Diagram

Diagram *Use Case* menggambarkan hubungan antara aktor dan aplikasi yang dibangun. Berikut ini adalah perancangan proses-proses yang terdapat pada *game* yang akan dibangun, yang digambarkan dengan *Use Case* Diagram pada Gambar 3.14.



Gambar 3.14 *Use Case* Diagram *Game* Olahraga Atletik

3.1.6.2 Definisi Aktor

Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem sistem yang akan dibangun. Definisi aktor diterangkan pada tabel 3.7.

Tabel 3.7 Definisi Aktor

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

3.1.6.3 Definisi *Use Case*

Definisi *Use Case* berfungsi untuk menjelaskan proses yang terdapat pada setiap *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.
2.	Memainkan S*print*	Proses penyajian permainan *sprint*
3.	Memainkan Lari Gawang	Proses penyajian permainan lari gawang
4.	Mermainkan Lompat Tinggi	Proses penyajian permainan Lompat Tinggi
5.	Memainkan Lompat Jauh	Proses penyajian permainan Lompat Jauh
6.	Memainkan Lempar Lembing	Proses penyajian permainan Lempar Lembing

3.1.6.4 Skenario *Use Case* Memulai Permainan

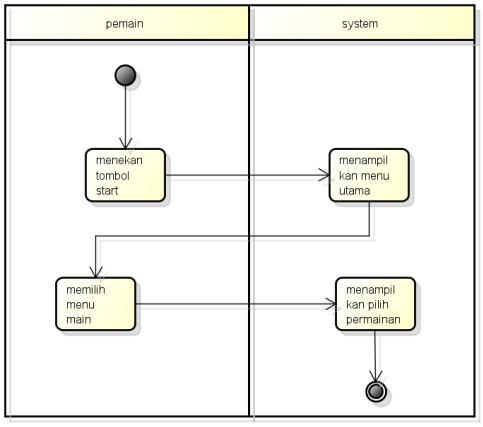
Tabel 3.9 berikut ini adalah skenario *Use Case* memulai permain pada *game* olahraga atletik.

Tabel 3.9 Skenario *Use Case* Memulai Permain

Identifikasi	
Nomor	UC-01
Nama	Memulai Permain
Tujuan	Menjalankan menu main
Deskripsi	Proses untuk mulai main
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain mengklik aplikasi *game* olahraga atletik
Aksi aktor	**Reaksi Sistem**
1. Menekan tombol start	
	2. Menampilkan menu utama
3. Memilih menu Main	
	4. Menampilkan pilih permainan
Kondisi akhir	Aplikasi menampilkan menu pilih permainan

3.1.6.5 *Activity Diagram* Memulai Permainan

Activity diagram memulai permainan menggambarkan aktifitas dan sistem yang saling berhubungan dari hasil skenario *use case* yang telah dijelaskan sebelumnya pada tabel 3.9 dapat dilihat pada gambar 3.15.



Gambar 3.15 *Activity Diagram* Mulai Main

3.1.6.6 Skenario *Use Case* Memainkan *Sprint*

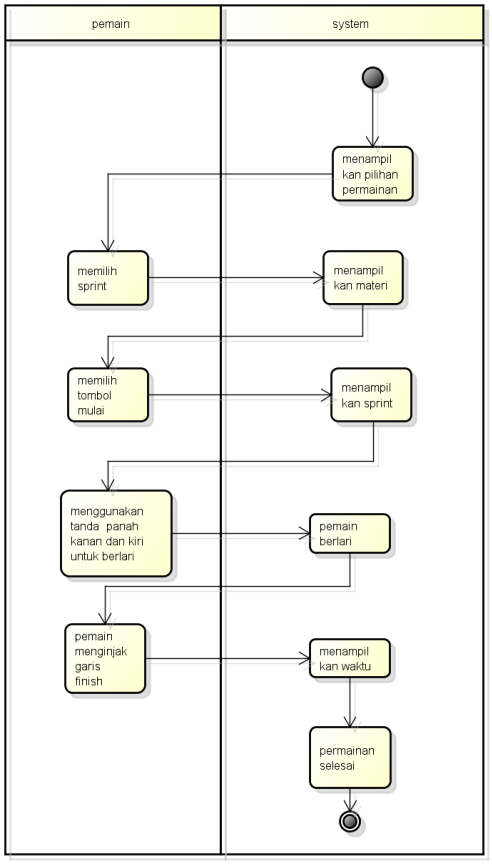
Tabel 3.10 berikut ini adalah skenario *Use Case* memainkan *sprint* pada *game* olahraga atletik.

Tabel 3.10 Skenario *Use Case* Memainkan *Sprint*

Identifikasi	
Nomor	UC-02
Nama	Memainkan *Sprint*
Tujuan	Menampilkan permainan *sprint*
Deskripsi	Proses untuk menampilkan permainan *sprint*
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada pada menu pilih permainan
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan menu pilih permainan
2. Memilih permainan *sprint*	
	3. Menampilkan materi permainan
4. Memilih mulai	
	5. Menampilkan permainan *sprint*
6. Menggunakan tombol panah kanan dan kiri	
	7. Pemain berlari
8. Pemain menginjak garis finish	
	9. Menampilkan waktu
	10. Permainan selesai
Kondisi akhir	Aplikasi menampilkan permainan *sprint*

3.1.6.7 *Activity Diagram* Memainkan *Sprint*

Activity diagram memainan *sprint* menggambarkan aktifitas dan sistem yang saling berhubungan dari hasil skenario *use case* yang telah dijelaskan sebelumnya pada tabel 3.10 dapat dilihat pada gambar 3.16.



Gambar 3.16 *Activity Diagram* Memainkan *Sprint*

3.1.6.8 Skenario *Use Case* Memainkan Lari Gawang

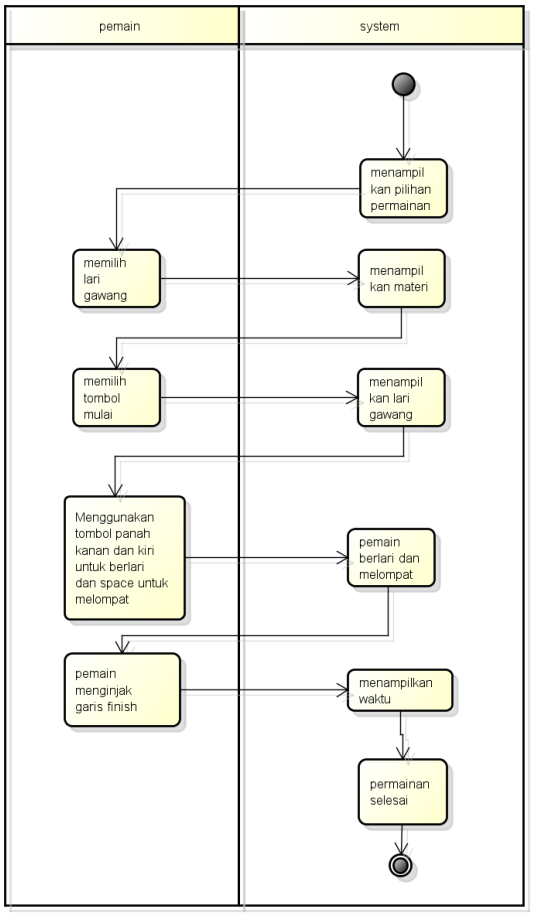
Tabel 3.11 berikut ini adalah skenario *Use Case* memainkan lari gawang pada *game* olahraga atletik.

Tabel 3.11 Skenario *Use Case* Memainkan Lari Gawang

Identifikasi	
Nomor	UC-03
Nama	Memainkan Lari gawang
Tujuan	Menampilkan permainan lari gawang
Deskripsi	Proses untuk menampilkan permainan lari gawng
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada pada menu pilih permainan
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan menu pilih permainan
2. Memilih permainan lari gawang	
	3. Menampilkan materi permainan
4. Memilih mulai	
	5. Menampilkan permainan lari gawang
6. Menggunakan tombol panah kanan dan kiri	
	7. Pemain berlari
8. Menggunakan tombol space	
	9. Pemain melompat
10. Pemain menginjak garis finish	
	11. Menampilkan waktu
	12. Permainan selesai
Kondisi akhir	Aplikasi menampilkan permainan lari gawang

3.1.6.9 *Activity Diagram* Memainkan Lari Gawang

Activity diagram memainkan lari gawang menggambarkan aktifitas dan sistem yang saling berhubungan dari hasil skenario *use case* yang telah dijelaskan sebelumnya pada tabel 3.11 dapat dilihat pada gambar 3.17.



Gambar 3.17 *Activity Diagram* Memainkan Lari Gawang

3.1.6.10 Skenario *Use Case* Memainkan Lompat Tinggi

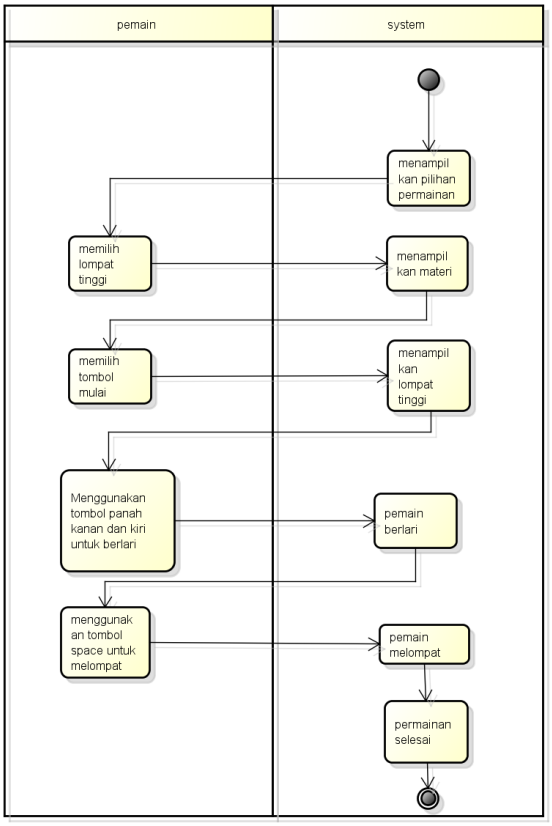
Tabel 3.12 berikut ini adalah skenario *Use Case* memainkan lompat tinggi pada *game* olahraga atletik.

Tabel 3.12 Skenario *Use Case* Memainkan Lompat Tinggi

Identifikasi	
Nomor	UC-04
Nama	Memainkan lompat tinggi
Tujuan	Menampilkan permainan lompat tinggi
Deskripsi	Proses untuk menampilkan permainan lompat tinggi
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada pada menu pilih permainan
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan menu pilih permainan
2. Memilih permainan lompat tinggi	
	3. Menampilkan materi permainan
4. Memilih mulai	
	5. Menampilkan permainan lompat tinggi
6. Menggunakan tombol panah kanan dan kiri	
	7. Pemain berlari
8. Menggunakan tombol space	
	9. Pemain melompat
	10. Permainan selesai
Kondisi akhir	Aplikasi menampilkan permainan lompat tinggi

3.1.6.11 *Activity Diagram* Memainkan Lompat Tinggi

Activity diagram memainkan lompat tinggi menggambarkan aktifitas dan sistem yang saling berhubungan dari hasil skenario *use case* yang telah dijelaskan sebelumnya pada tabel 3.12 dapat dilihat pada gambar 3.18.



Gambar 3.18 *Activity Diagram* Memainkan Lompat Tinggi

3.1.6.12 Skenario *Use Case* Memainkan Lompat Jauh

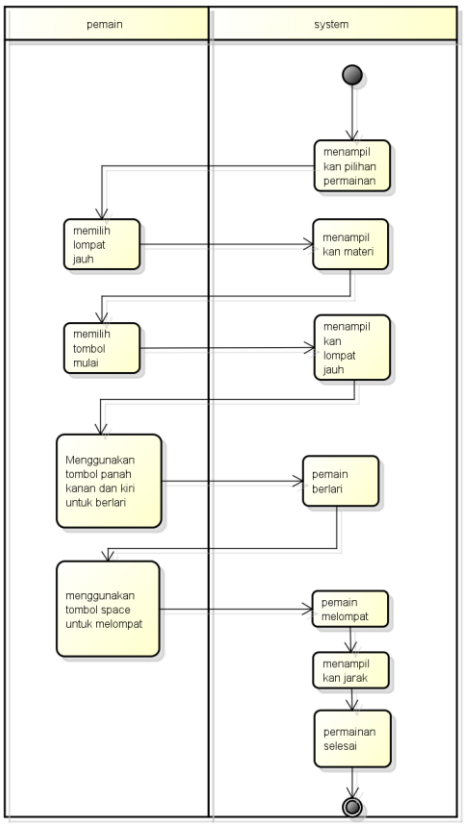
Tabel 3.13 berikut ini adalah skenario *Use Case* memainkan lompat jauh pada *game* olahraga atletik.

Tabel 3.13 Skenario *Use Case* Memainkan Lompat Jauh

Identifikasi	
Nomor	UC-05
Nama	Memainkan lompat jauh
Tujuan	Menampilkan permainan lompat jauh
Deskripsi	Proses untuk menampilkan permainan lompat jauh
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada pada menu pilih permainan
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan menu pilih permainan
2. Memilih permainan lompat jauh	
	3. Menampilkan materi permainan
4. Memilih mulai	
	5. Menampilkan permainan lompat jauh
6. Menggunakan tombol panah kanan dan kiri	
	7. Pemain berlari
8. Menggunakan tombol panah space	
	9. Pemain melompat
	10. Menampilkan jarak
	11. Permainan selesai
Kondisi akhir	Aplikasi menampilkan permainan lompat jauh

3.1.6.13 *Activity Diagram* Memainkan Lompat Jauh

Activity diagram memainkan lompat jauh menggambarkan aktifitas dan sistem yang saling berhubungan dari hasil skenario *use case* yang telah dijelaskan sebelumnya pada tabel 3.13 dapat dilihat pada gambar 3.19.



Gambar 3.19 *Activity Diagram* Memainkan Lompat Jauh

3.1.6.14 Skenario *Use Case* Memainkan Lempar Lembing

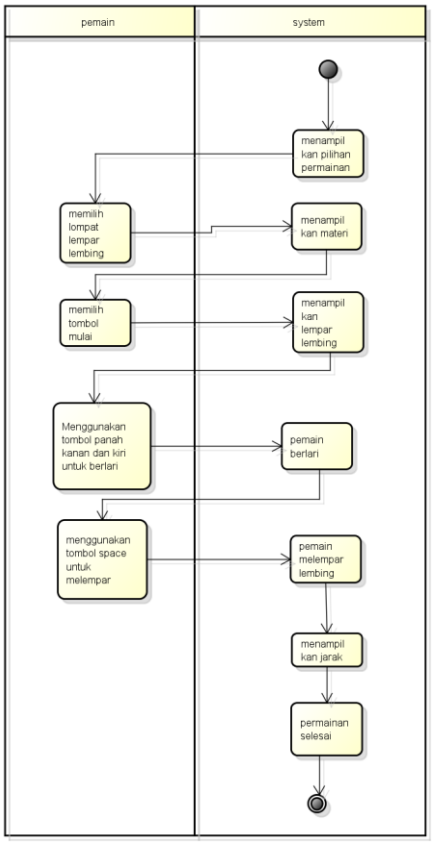
Tabel 3.14 berikut ini adalah skenario *Use Case* memainkan lempar lembing pada *game* olahraga atletik.

Tabel 3.14 Skenario *Use Case* Memainkan Lempar Lembing

Identifikasi	
Nomor	UC-06
Nama	Memainkan lempar lembing
Tujuan	Menampilkan permainan lempar lembing
Deskripsi	Proses untuk menampilkan permainan lempar lembing
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada pada menu pilih permainan
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan menu pilih permainan
2. Memilih permainan lempar lembing	
	3. Menampilkan materi permainan
4. Memilih mulai	
	5. Menampilkan permainan lempar lembing
6. Menggunakan tombol panah kanan dan kiri untuk berlari	
	7. Pemain berlari
8. Menggunakan tombol space untuk melempar	
	9. Pemain melempar
	10. Menampilkan jarak
	11. Permainan selesai
Kondisi akhir	Aplikasi menampilkan permainan lempar lembing

3.1.6.15 *Activity Diagram* Memainkan Lempar Lembing

Activity diagram memainkan lempar lembing menggambarkan aktifitas dan sistem yang saling berhubungan dari hasil skenario *use case* yang telah dijelaskan sebelumnya pada tabel 3.14 dapat dilihat pada gambar 3.20.



Gambar 3.20 *Activity Diagram* Memainkan Lempar Lembing

3.1.7 *Sequence Diagram*

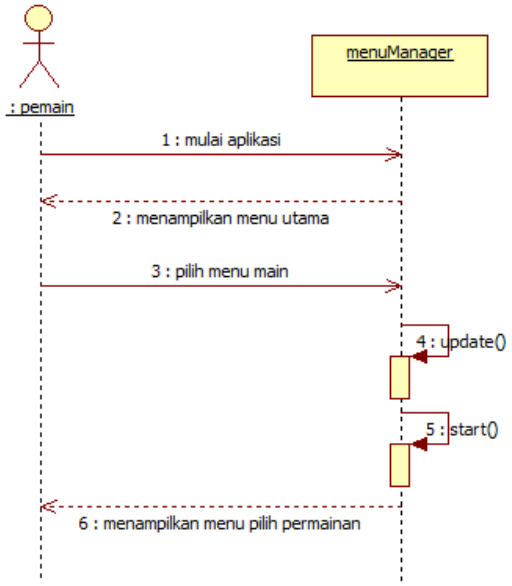
Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup dan *message* yang dikirimkan dan diterima antar objek.

1. *Use Case* Memulai Permainan

No	Nama Kelas	Jenis Kelas *(Interface, Control, Entity)*
1	menuManager	Control

2. *Sequence Diagram* Memulai Permainan

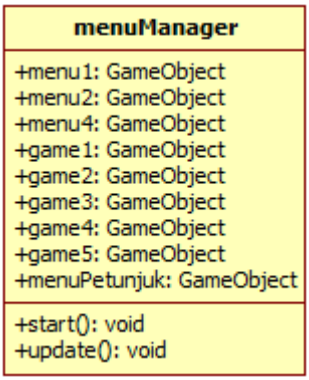
Berikut ini adalah *sequence diagram* memulai permainan dapat dilihat pada gambar 3.21.



Gambar 3.21 *Sequence Diagram* Memulai Permainan

3. *Class Diagram* Perancangan Memulai Permainan

Berikut ini adalah *class diagram* memulai permainan dapat dilihat pada gambar 3.22.



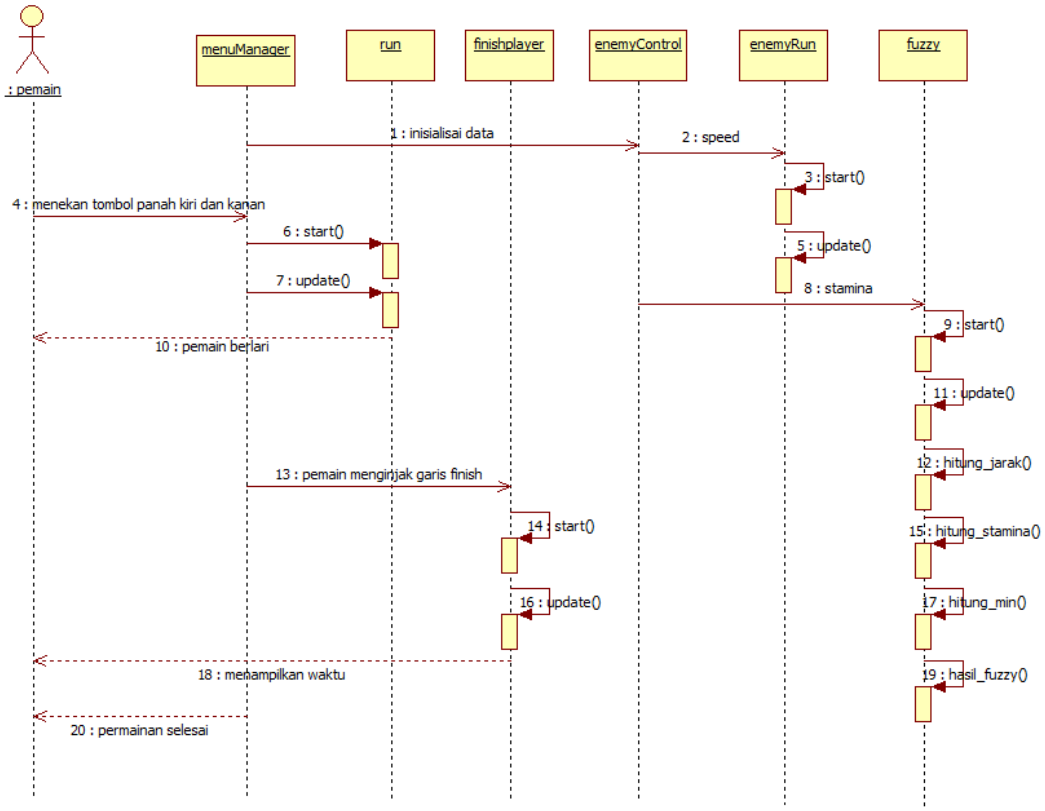
Gambar 3.22 *Class Diagram* Memulai Permainan

4. *Use Case* Permainan *Sprint*

No	Nama Kelas	Jenis Kelas *(Interface, Control, Entity)*
1	menuManager	Control
2	run	Control
3	finishplayer	Control
4	enemyControl	Control
5	enemyRun	Control
6	Fuzzy	Control

5. *Sequence diagram* Memainkan S*print*

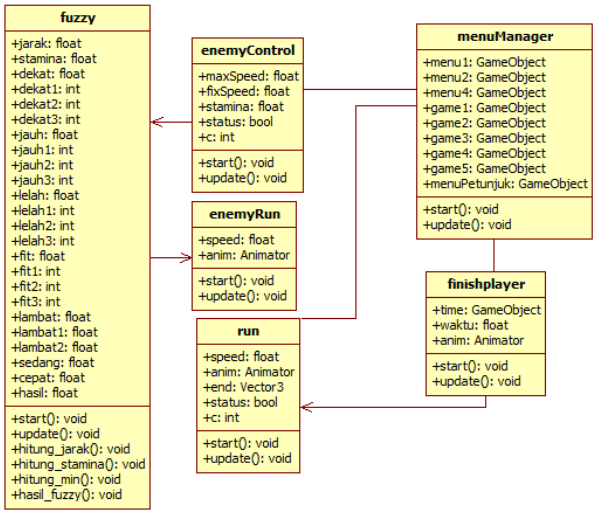
Berikut ini adalah *sequence diagram* memainkan *sprint* dapat dilihat pada gambar 3.23.



Gambar 3.23 *Sequence Diagram* Memainkan S*print*

6. *Class Diagram* Memainkan *Sprint*

Berikut ini adalah *class diagram* memainkan *sprint* dapat dilihat pada gambar 3.24.



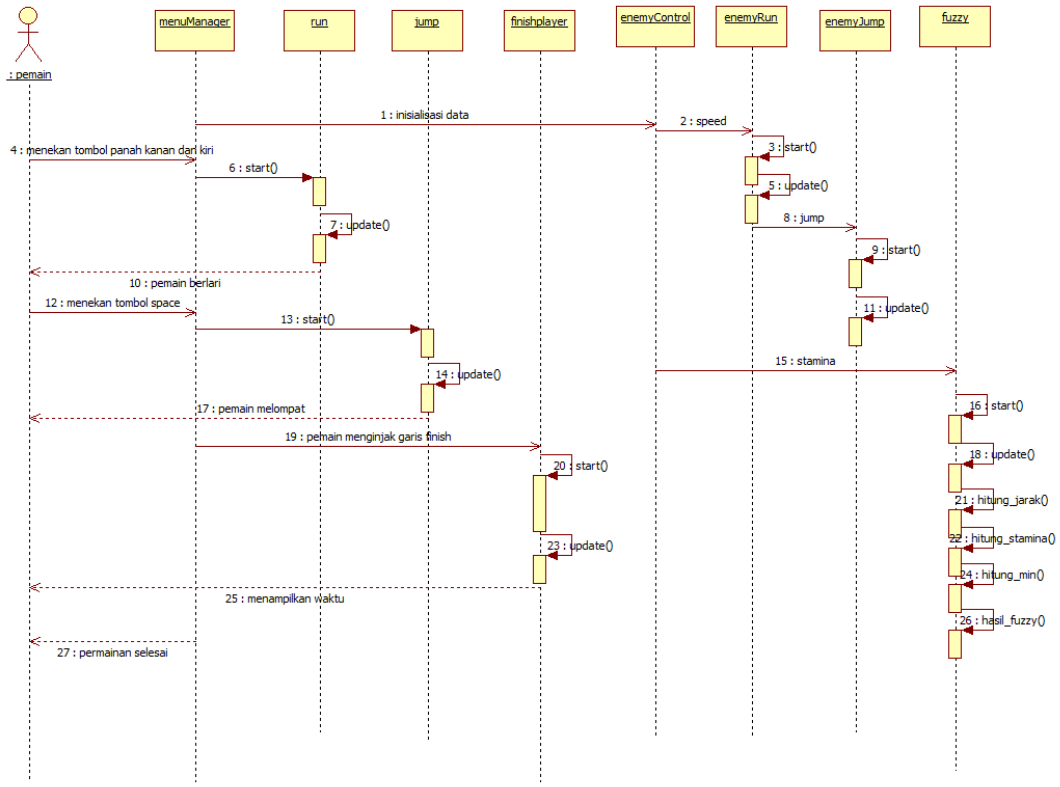
Gambar 3.24 Class Diagram Memainkan *Sprint*

7. *Use Case* Memainkan Lari Gawang

No	Nama Kelas	Jenis Kelas *(Interface, Control, Entity)*
1	menuManager	Control
2	Run	Control
3	Jump	Control
4	finishplayer	Control
5	enemyControl	Control
6	enemyRun	Control
7	enemyJump	Control
8	Fuzzy	Control

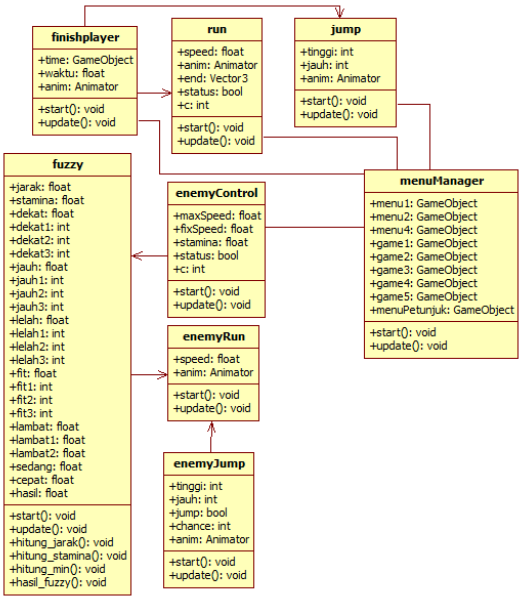
8. *Sequence Diagram* Memainkan Lari Gawang

Berikut ini adalah *sequence diagram* memainkan lari gawang dapat dilihat pada gambar 3.25.



Gambar 3.25 *Sequence Diagram* Memainkan Lari Gawang

9. *Class Diagram* Memainkan Lari Gawang

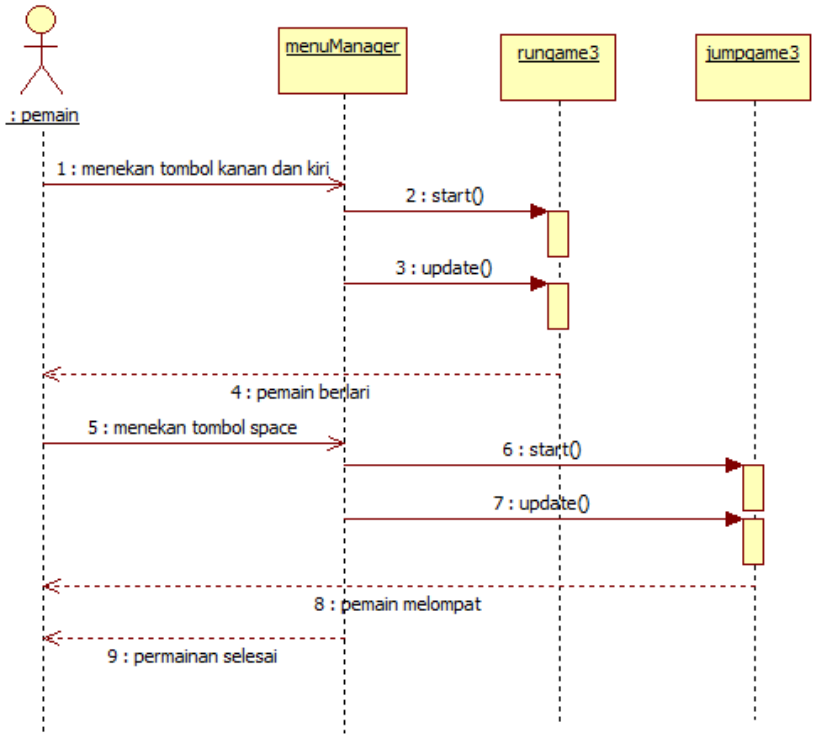


Gambar 3.26 *Class Diagram* Memainkan Lari Gawang

10. *Use case diagram* Memainkan Lompat Tinggi

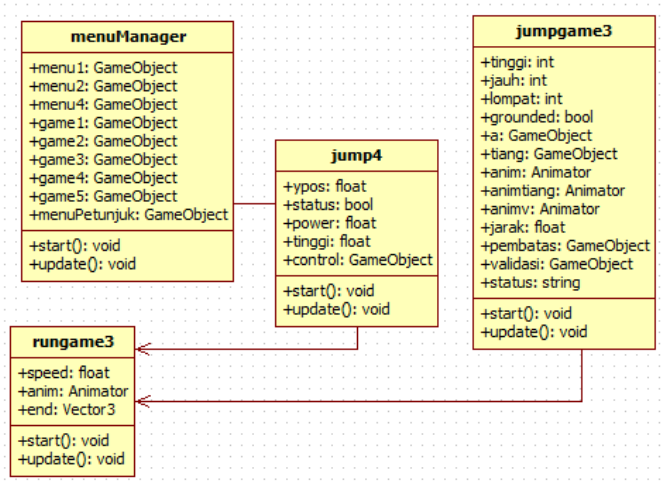
No	Nama Kelas	Jenis Kelas *(Interface, Control, Entity)*
1	menuManager	Control
2	rungame3	Control
3	jumpgame3	Control

11. *Sequence diagram* Memainkan Lompat Tinggi



Gambar 3.27 *Sequence Diagram* permainan lompat tinggi

12. *Class diagram* Memainkan Lompat Tinggi

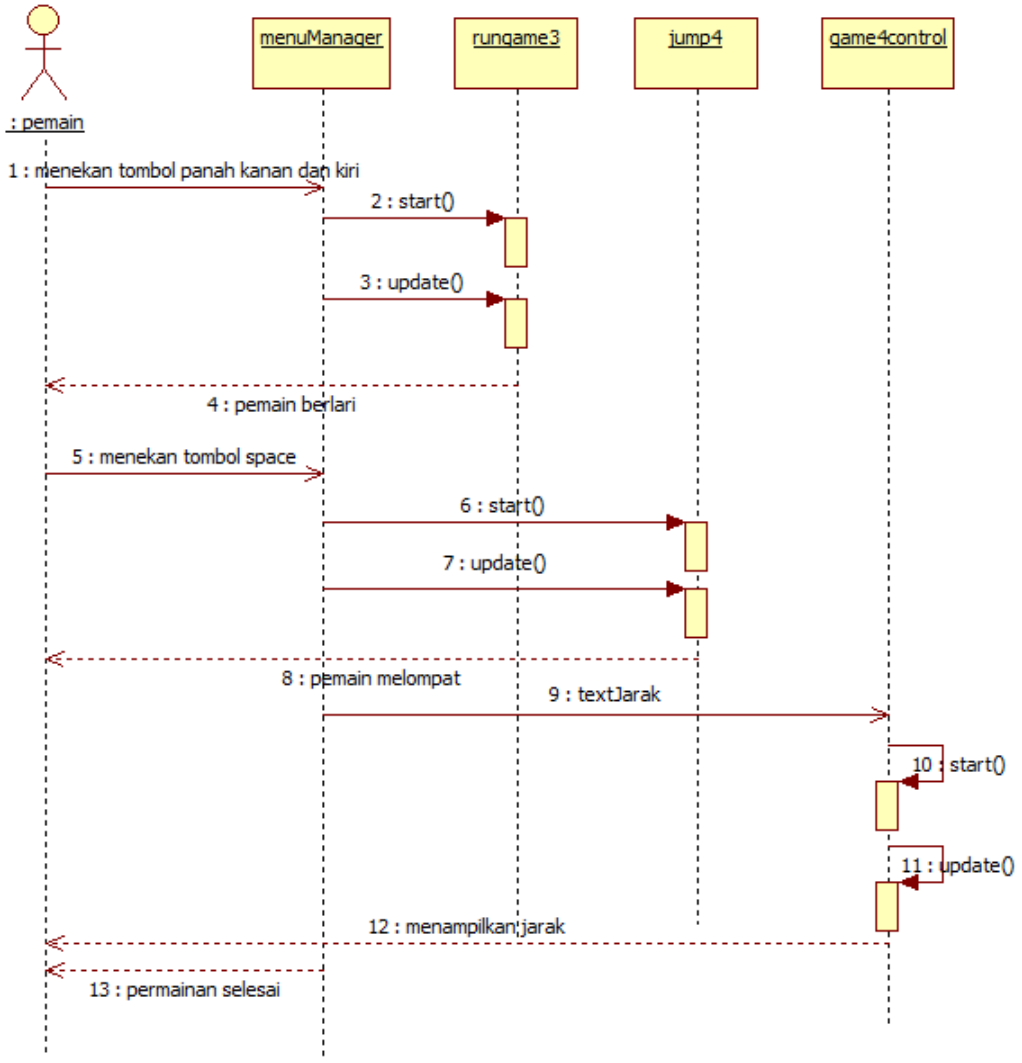


Gambar 3.28 *Class diagram* Memainkan Lompat Tinggi

13. *Use Case diagram* Memainkan Lompat Jauh

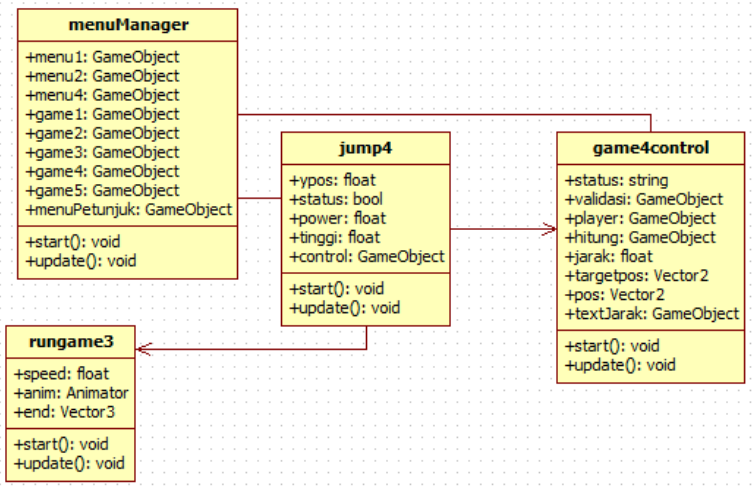
No	Nama Kelas	Jenis Kelas *(Interface, Control, Entity)*
1	menuManager	Control
2	rungame3	Control
3	jump4	Control
4	game4control	Control

14. *Sequence diagram* Memainkan Lompat Jauh



Gambar 3.29 *Sequence Diagram* Memainkan Lompat Jauh

15. *Class Diagram* Memainkan Lompat Jauh

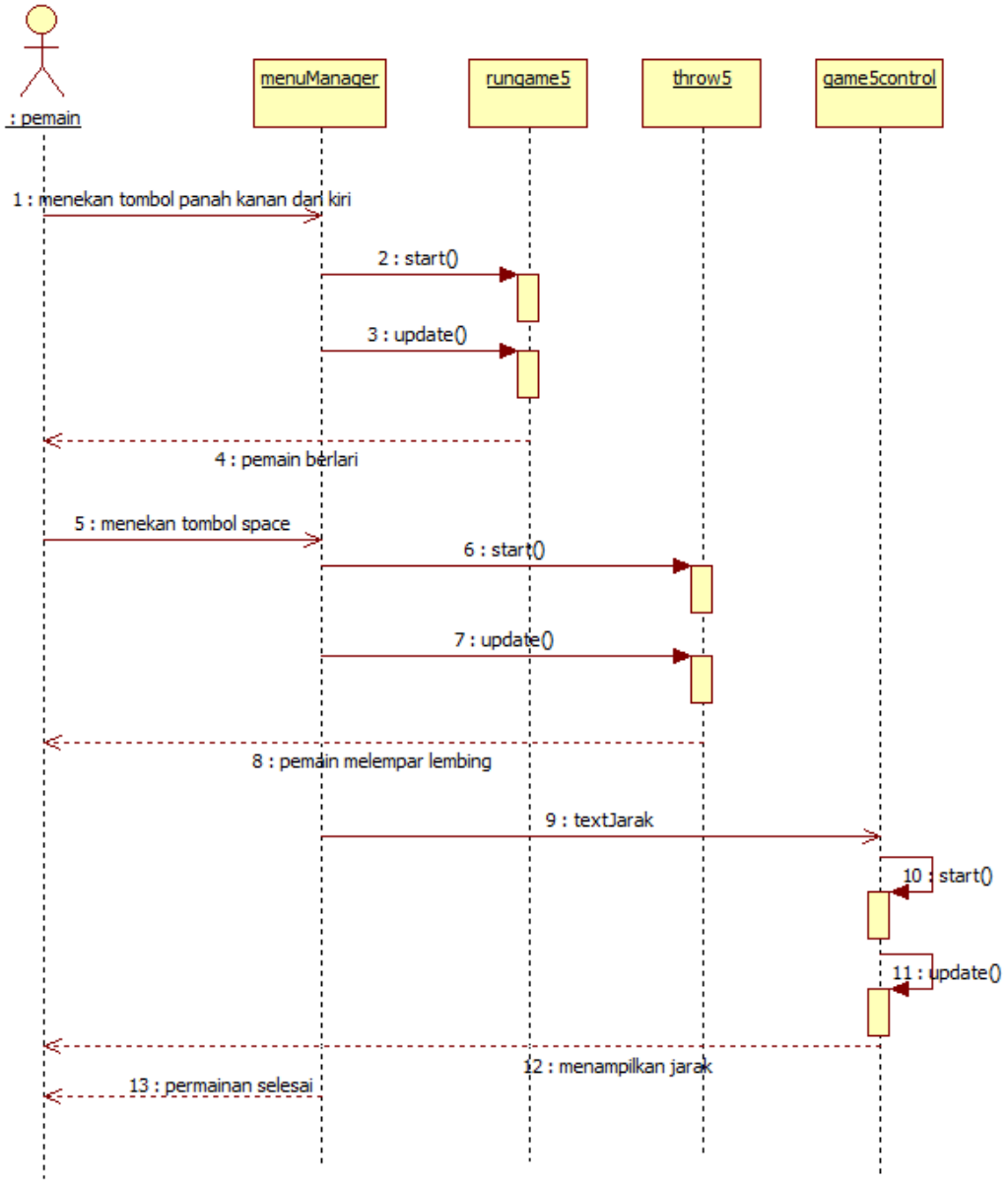


Gambar 3.30 *Class Diagram* Memainkan Lompat Jauh

16. *Use Case diagram* Memainkan Lempar Lembing

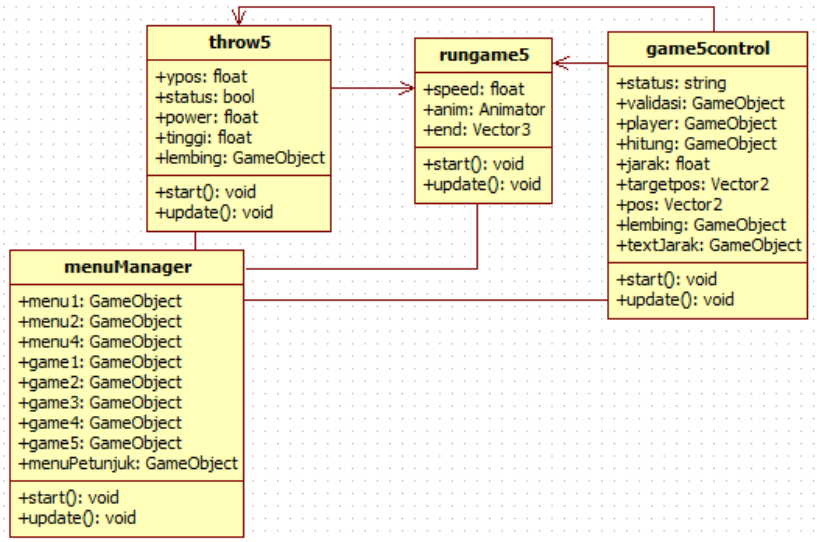
No	Nama Kelas	Jenis Kelas *(Interface, Control, Entity)*
1	menuManager	Control
2	rungame5	Control
3	throw5	Control
4	game5control	Control

17. *Sequence diagram* Memainkan Lempar Lembing



Gambar 3.31 *Sequence Diagram* Memainkan Lempar Lembing

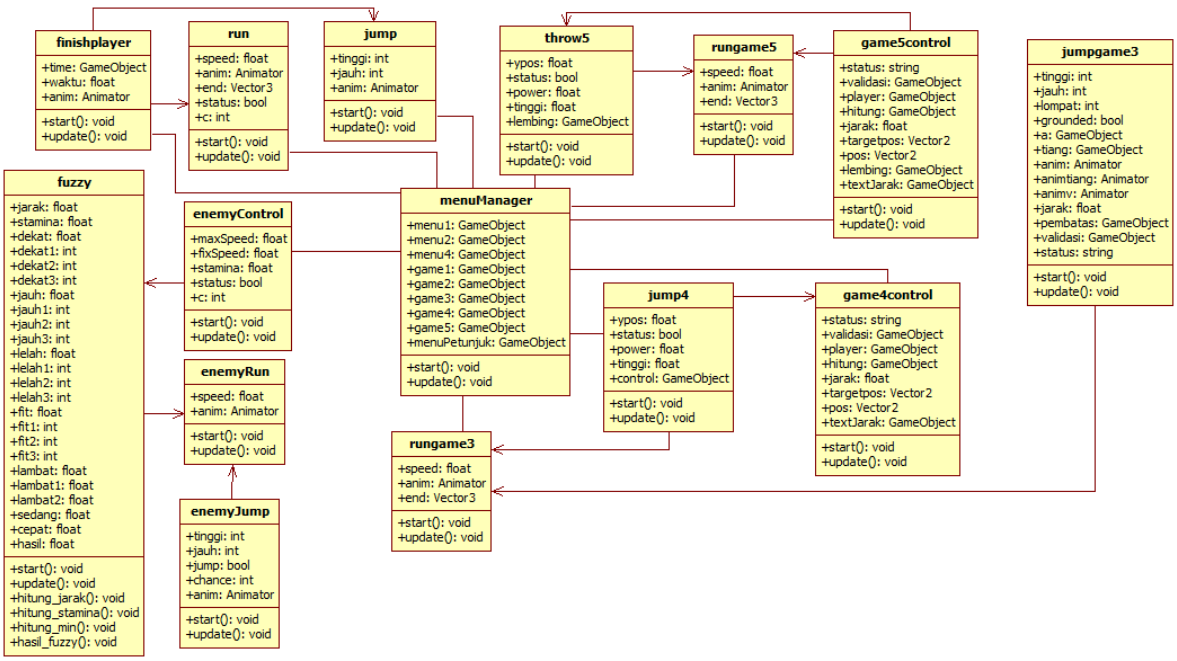
18. *Class Diagram* permainan lempar lembing



Gambar 3.32 *Class Diagram* Memainkan Lempar Lembing

3.1.8 *Class Diagram*

Class diagram digunakan untuk menampilkan kelas-kelas dan paket-paket di dalam sistem. Class diagram memberikan gambaran sistem secara statis dan relasi antar mereka. Beberapa diagram akan menampilkan subset dari kelas-kelas dan relasinya. Berikut adalah class diagram dari *game* yang akan dibangun pada gambar 3.33.



Gambar 3.33 *Class Diagram* Keseluruhan

3.1.8.1 Rincian *Class Diagram*

Rincian *Class diagram* adalah pendeskripsian mengenai kelas-kelas keseluruhan yang terlibat pada gambar 3.33 sebelumnya.

1. *Class* jump

   Deskripsi *class* jump : *class* yang mengatur lompatan pemain yang terdapat pada *game* olahraga atletik adalah sebagai berikut :

Nama Operasi	***Visibility* *(private, public)***	**Keterangan**
start()	*Public*	Ketika proses dimulai
update()	*Public*	Proses untuk melompat
Nama Atribut	***Visibility* *(private, public)***	**Tipe**
tinggi	*Public*	Int
jauh	*Public*	Int
anim	*Public*	Animator

2. *Class* run

Deskripsi *class* run: *class* yang mengatur lari pemain yang terdapat pada *game* olahraga atletik adalah sebagai berikut :

Nama Operasi	***Visibility* *(private, public)***	**Keterangan**
start()	*Public*	Ketika proses dimulai
update()	*Public*	Proses untuk berlari
Nama Atribut	***Visibility* *(private, public)***	**Tipe**
speed	*Public*	Int
anim	*Public*	Animator
end	*Public*	Vector3
Status	*Public*	Bool
C	*Public*	Int

3. *Class* finishplayer

Deskripsi *class* finishplayer: *class* yang mengatur garis finish yang terdapat pada *game* olahraga atletik adalah sebagai berikut :

Nama Operasi	***Visibility* *(private, public)***	**Keterangan**
start()	*Public*	Ketika proses dimulai
update()	*Public*	Proses ketika pemain menyentuh garis finish
Nama Atribut	***Visibility* *(private, public)***	**Tipe**
time	*Public*	gameObject
waktu	*Public*	Float
anim	*Public*	Animator

4. *Class* menuManager

Deskripsi *class* menuManager: *class* yang mengatur menu-menu yang terdapat pada *game* olahraga atletik adalah sebagai berikut :

Nama Operasi	***Visibility* *(private, public)***	**Keterangan**
start()	*Public*	Ketika proses dimulai
update()	*Public*	Proses untuk mengatur menu-menu dalam permain
Nama Atribut	***Visibility* *(private, public)***	**Tipe**
menu1	*Public*	GameObject
menu2	*Public*	GameObject
menu4	*Public*	GameObject
game1	*Public*	GameObject
game2	*Public*	GameObject
game3	*Public*	GameObject
game4	*Public*	GameObject
game5	*Public*	GameObject
menuPetunjuk	*Public*	GameObject

5. Class enemyControl

   Deskripsi *class* enemyControl: *class* entitas yang mengatur NPC yang terdapat pada *game* olahraga atletik adalah sebagai berikut :

Nama Operasi	***Visibility* *(private, public)***	**Keterangan**
start()	*Public*	Ketika proses dimulai
update()	*Public*	Proses untuk mengatur NPC
Nama Atribut	***Visibility* *(private, public)***	**Tipe**
maxSpeed	*Public*	Float
fixSpedd	*Public*	Float
stamina	*Public*	Float
status	*Public*	Bool
c	*Public*	Int

6. *Class* fuzzy

   Deskripsi *class* fuzzy: *class* yang mengatur *fuzzy logic* yang terdapat pada *game* olahraga atletik adalah sebagai berikut :

Nama Operasi	***Visibility* *(private, public)***	**Keterangan**
start()	*Public*	Ketika proses dimulai
update()	*Public*	Ketika proses mengalami perubahan
hitung_jarak()	*Public*	Proses menghitung jarak
hitung_stamina()	*Public*	Proses menghitung stamina
hitung_min()	*Public*	Proses pengecekan infrence minimal
hasil_fuzzy()	*Public*	Proses pemberitahuan hasil fuzzy
Nama Atribut	***Visibility* *(private, public)***	**Tipe**
jarak	*Public*	Float
stamina	*Public*	Float
dekat	*Public*	Float
dekat1	*Public*	Int
dekat2	*Public*	Int
dekat3	*Public*	Int
jauh	*Public*	Float
jauh1	*Public*	Int
jauh2	*Public*	Int
jauh3	*Public*	Int
lelah	*Public*	Float
lelah1	*Public*	Int
lelah2	*Public*	Int
lelah3	*Public*	Int
fit	*Public*	Float
fit1	*Public*	Int
fit2	*Public*	Int
fit3	*Public*	Int
Lambat	*Public*	Float
Lambat1	*Public*	Float
Lambat2	*Public*	Float
Sedang	*Public*	Float
Cepat	*Public*	Float
hasil	*public*	Float

7. *Class* throw5

   Deskripsi *class* throw5: *class* yang mengatur lemparan pemain yang terdapat pada *game* olahraga atletik adalah sebagai berikut :

Nama Operasi	***Visibility* *(private, public)***	**Keterangan**
start()	*Public*	Ketika proses dimulai
update()	*Public*	Proses ketika pemain melakukan lemparan
Nama Atribut	***Visibility* *(private, public)***	**Tipe**
ypos	*Public*	Float
status	*Public*	Bool
power	*Public*	Float
tinggi	*Public*	Float
lembing	*Public*	GameObject

8. *Class* enemyRun

   Deskripsi *class* enemyRun: *class* yang mengatur lari NPC yang terdapat pada *game* olahraga atletik adalah sebagai berikut :

Nama Operasi	***Visibility* *(private, public)***	**Keterangan**
start()	*Public*	Ketika proses dimulai
update()	*Public*	Proses ketika NPC berlari
Nama Atribut	***Visibility* *(private, public)***	**Tipe**
speed	*Public*	Float
anim	*Public*	Animator

9. *Class* enemyJump

   Deskripsi *class* enemyJump: *class* yang mengatur lompatan NPC yang terdapat pada *game* olahraga atletik adalah sebagai berikut :

Nama Operasi	***Visibility* *(private, public)***	**Keterangan**
start()	*Public*	Ketika proses dimulai
update()	*Public*	Proses ketika NPC melompat
Nama Atribut	***Visibility* *(private, public)***	**Tipe**
tinggi	*Public*	Int
jauh	*Public*	Int
jump	*Public*	Bool
chance	*Public*	Int
Anim	*Public*	Animator

10. *Class* rungame5

Deskripsi *class* rungame5: *class* yang mengatur lari pemain pada permainan lempar lembing yang terdapat pada *game* olahraga atletik adalah sebagai berikut :

Nama Operasi	***Visibility* *(private, public)***	**Keterangan**
start()	*Public*	Ketika proses dimulai
update()	*Public*	Proses ketika pemain berlari pada permainan lempar lembing
Nama Atribut	***Visibility* *(private, public)***	**Tipe**
speed	*Public*	Float
anim	*Public*	Animator
end	*Public*	Vector3

11. Class jump4

Deskripsi *class* jump4: *class* yang mengatur lompatan pemain pada permainan lompat jauh yang terdapat pada *game* olahraga atletik adalah sebagai berikut :

Nama Operasi	***Visibility* *(private, public)***	**Keterangan**
start()	*Public*	Ketika proses dimulai
update()	*Public*	proses ketika pemain melompat pada permainan lompat jauh
Nama Atribut	***Visibility* *(private, public)***	**Tipe**
ypos	*Public*	Float
status	*Public*	Boll
power	*Public*	Float
tinggi	*Public*	Float

12. *Class* game5control

Deskripsi *class* game5control: *class* yang mengatur permainan lempar lembing yang terdapat pada *game* olahraga atletik adalah sebagai berikut :

Nama Operasi	***Visibility* *(private, public)***	**Keterangan**
start()	*Public*	Ketika proses dimulai
update()	*Public*	proses untuk mengatur permainan lempar lembing
Nama Atribut	***Visibility* *(private, public)***	**Tipe**
status	*Public*	String
validasi	*Public*	GameObject
player	*Public*	GameObject
hitung	*Public*	GameObject
jarak	*Public*	Float
targetpos	*Public*	Vector2
pos	*Public*	Vector2
lembing	*Public*	GameObject
textJarak	*Public*	GameObject

13. *Class* game4control

Deskripsi *class* game4control: *class* yang mengatur permainan lompat jauh yang terdapat pada *game* olahraga atletik adalah sebagai berikut :

Nama Operasi	***Visibility* *(private, public)***	**Keterangan**
start()	*Public*	Ketika proses dimulai
update()	*Public*	proses untuk mengatur permainan lompat jauh
Nama Atribut	***Visibility* *(private, public)***	**Tipe**
Status	*Public*	String
Validasi	*Public*	GameObject
Player	*Public*	GameObject
Hitung	*Public*	GameObject
Jarak	*Public*	Float
targetpos	*Public*	Vector2
Pos	*Public*	Vector2
textJarak	*Public*	GameObject

14. *Class* jumpgame3

Deskripsi *class* jumpgame3: class yang mengatur lompatan pemain pada permainan lompat tinggi yang terdapat pada *game* olahraga atletik adalah sebagai berikut :

Nama Operasi	***Visibility* *(private, public)***	**Keterangan**
start()	*Public*	Ketika proses dimulai
update()	*Public*	Proses ketika pemain melompat pada permainan lompat tinggi
Nama Atribut	***Visibility* *(private, public)***	**Tipe**
tinggi	*Public*	Int
jauh	*Public*	Int
lompat	*Public*	Int
grounded	*Public*	Bool
a	*Public*	GameObject
tiang	*Public*	GameObject
anim	*Public*	Animator
animtiang	*Public*	Animator
animv	*Public*	Animator
jarak	*Public*	Float
pembatas	*Public*	GameObject
validasi	*Public*	GameObject
status	*Public*	String

15. *Class* rungame3

Deskripsi *class* rungame3: *class* yang mengatur lari pada permainan lompat tinggi yang terdapat pada *game* olahraga atletik adalah sebagai berikut :

Nama Operasi	***Visibility* *(private, public)***	**Keterangan**
start()	*Public*	Ketika proses dimulai
update()	*Public*	Proses ketika pemain berlari pada permain lompat tinggi
Nama Atribut	***Visibility* *(private, public)***	**Tipe**
speed	*Public*	Float
anim	*Public*	Animator
end	*Public*	Vector3

3.2 Perancangan Sistem

Perancangan merupakan bagian dari metodologi pembangunan suatu perangkat lunak yang harus dilakukan setelah melalui tahapan analisis. Untuk dapat melakukan perancangan sistem maka dilakukan beberapa langkah yaitu sebagai berikut :

1. Perancangan Komponen Permainan
2. Perancangan Struktur Menu
3. Perancangan Antarmuka
4. Jaringan Semantik

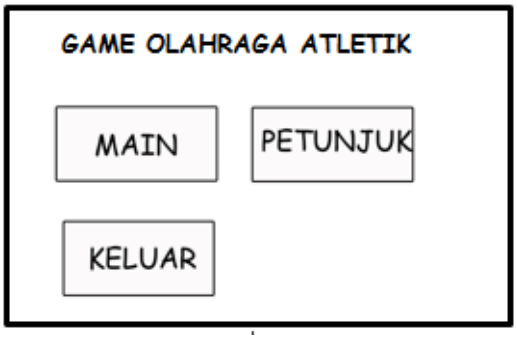
3.2.1 Perancangan Komponen Permainan

3.2.1.1 *Storyboard*

Storyboard adalah serangkaian sketsa yang dibuat berbentuk persegi yang menggambarkan suatu urutan (alur cerita) elemen-elemen yang diusulkan untuk aplikasi multimedia. *Storyboard* menggabungkan alat bantu narasi dan visual pada selembar kertas sehingga naskah dan visual menjadi terkoordinasi. S*toryboard* dapat diartikan sebagai alat perencanaan yang menggambarkan urutan kejadian berupa kumpulan gambar dalam sketsa sederhana.

a. *Storyboard* Menu Utama

Tampilan menu ini adalah menu utama. Terdapat menu main, menu petunjuk, dan menu keluar. Berikut *storyboard* menu utama pada gambar 3.34.



Gambar 3.34 *Storyboard Menu* Utama

b. S*toryboard Game Sprint*

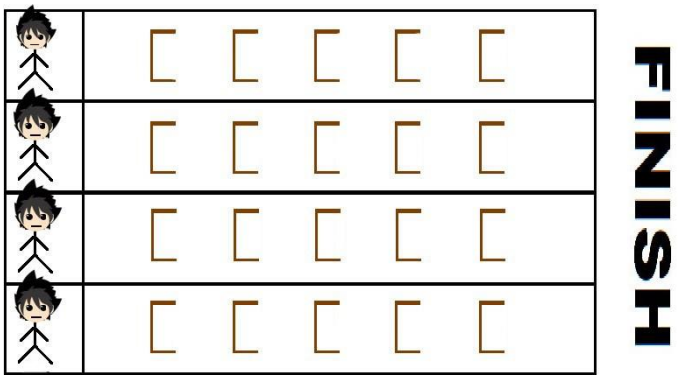
Tampilan menu ini adalah tampilan *game sprint*, pemain dan NPC berlomba-lomba menuju garis *finish*. Berikut *storyboard game sprint* terdapat pada gambar 3.35.



Gambar 3.35 *Storyboard Game Sprint*

c. *Storyboard Game* Lari Gawang

Tampilan ini merupakan tampilan *game* lari gawang, pemain dan NPC berlari menuju garis *finsih* dengan melewati beberapa halangan gawang untuk mencapai garis *finish.* Berikut *storyboard game* lari gawang terdapat pada gambar 3.36.



Gambar 3.36 *Storyboard Game* Lari Gawang

d. *Storyboard* Lompat Tinggi

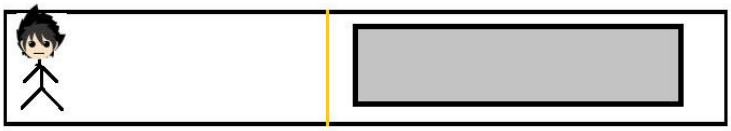
Tampilan ini merupakan tampilan *game* lompat tinggi, pemain harus melompat melewati batas mistar. Berikut *storyboard game* lompat tinggi terdapat pada gambar 3.37.



Gambar 3.37 *Storyboard* Lompat Tinggi

e. *Storyboard* Lompat Jauh

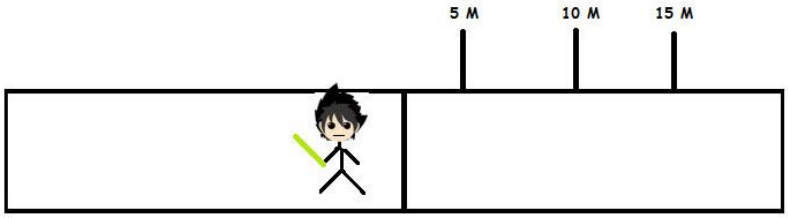
Tampilan ini merupakan tampilan *game* lompat jauh, pemain harus melompat pada pasir sejauh mungkin. Berikut *storyboard game* lompat jauh terdapat pada gambar 3.38.



Gambar 3.38 *Storyboard* Lompat Jauh

f. *Storyboard* Lempar Lembing

Tampilan ini merupakan tampilan *game* lempar lembing, pemain harus melempar lembing sejauh mungkin. Berikut *storyboard game* lempar lembing terdapat pada gambar 3.39.

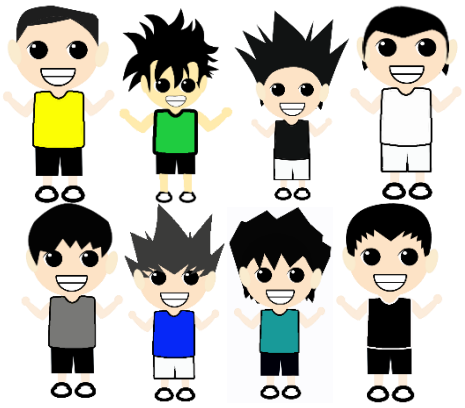


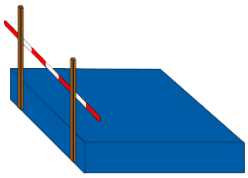
Gambar 3.39 *Storyboard* Lempar Lembing

3.2.1.2 Perancangan Karakter

Pengenalan karakter merupakan pembahasan mengenai karakter-karakter yang terlibat dalam *game* yang akan dibangun. Pengenalan karakter dapat dilihat pada tabel 3.15.

Tabel 3.15 Pengenalan Karakter Dan Item

No.	Gambar	Nama	Keterangan
1		Atlet	Karakter yang terdapat pada permainan.
2		Gawang	Objek yang digunakan sebagai rintangan

3		Matras	Matras yang digunakan untuk lompat tinggi
4		Lembing	Objek yang dilemparkan pada permainan lempar lembing

3.2.2 Perancangan Struktur Menu

Struktur menu adalah bentuk umum dari suatu rancangan program untuk memudahkan pemakai dalam menjalankan program komputer. Sehingga saat menjalankan program, pengguna tidak mengalami kesulitan dalam memilih menu-menu yang diinginkan. Berikut ini perancangan struktur menu pada *Game* Olahraga Atletik pada gambar 3.40 berikut ini.



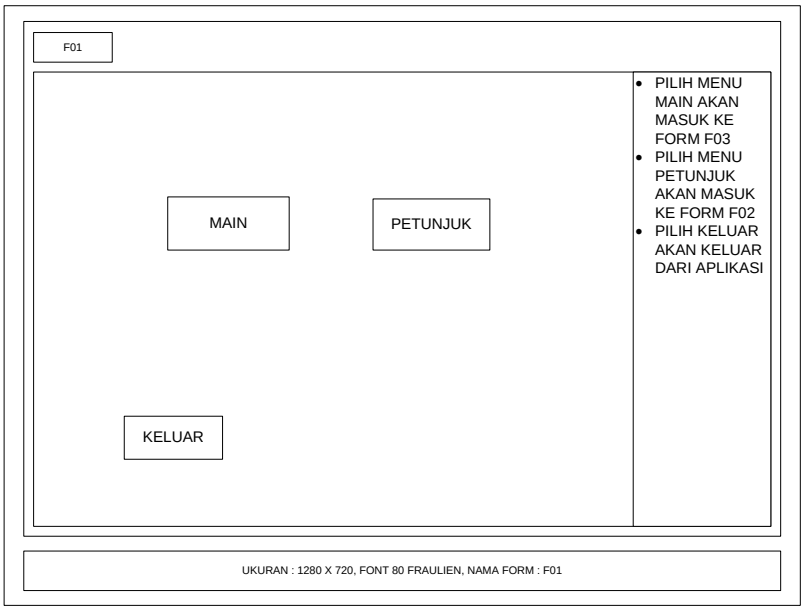
Gambar 3.40 Struktur Menu *Game* Olahraga Atletik

3.2.3 Perancangan Antarmuka

Perancangan antarmuka dibutuhkan untuk mewakili keadaan sebenarnya dari aplikasi yang akan dibangun. Berikut contoh perancangan antarmuka yang ada dari aplikasi yang akan dibangun yaitu :

3.2.3.1 Perancangan Antarmuka Menu Utama

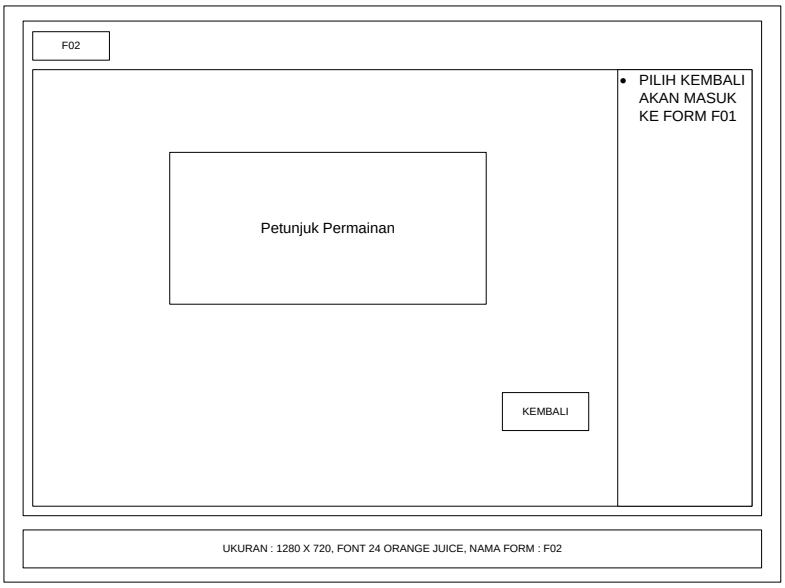
Berikut ini perancangan antarmuka menu utama G*ame* Olahraga Atletik pada gambar 3.41 berikut :



Gambar 3.41 Perancangan Antarmuka Menu Utama

3.2.3.2 Perancangan Antarmuka Petunjuk

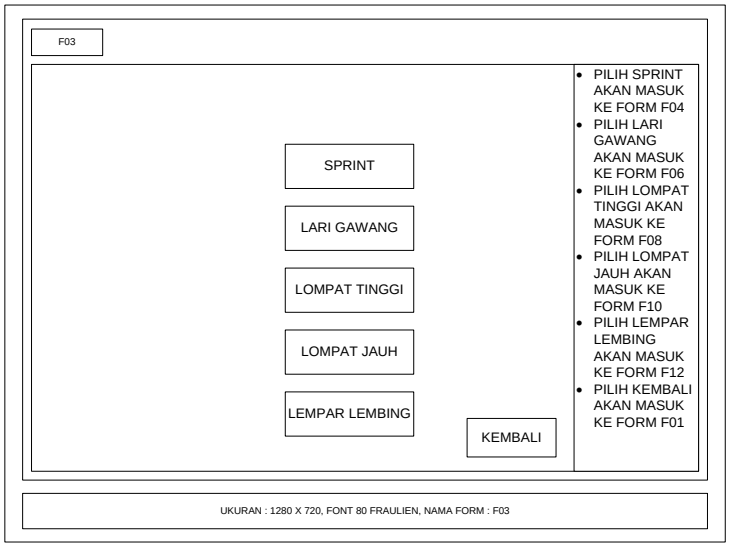
Berikut ini perancangan antarmuka petunjuk *Game* Olahraga Atletik pada gambar 3.42 berikut :



Gambar 3.42 Perancangan Antarmuka Petunjuk

3.2.3.3 Perancangan Antarmuka Pilih Permainan

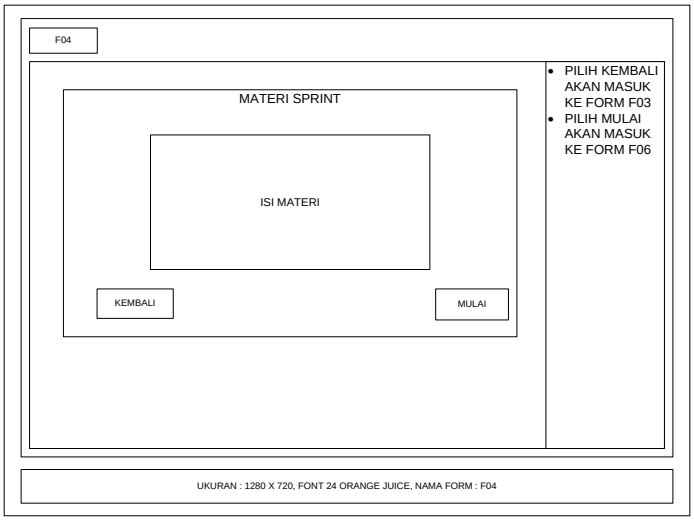
Berikut ini perancangan antarmuka pilih permain *Game* Olahraga Atletik pada gambar 3.43 berikut :



Gambar 3.43 Perancangan Antarmuka Pilih Permainan

3.2.3.4 Perancangan Antarmuka Materi *Sprint*

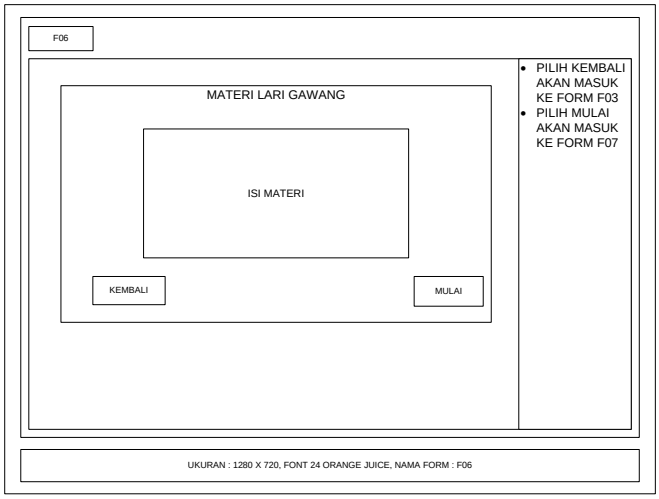
Berikut ini perancangan antarmuka materi s*print Game* Olahraga Atletik pada gambar 3.44 berikut :



Gambar 3.44 Perancangan Antarmuka Materi *Sprint*

3.2.3.5 Perancangan Antarmuka Materi Lari Gawang

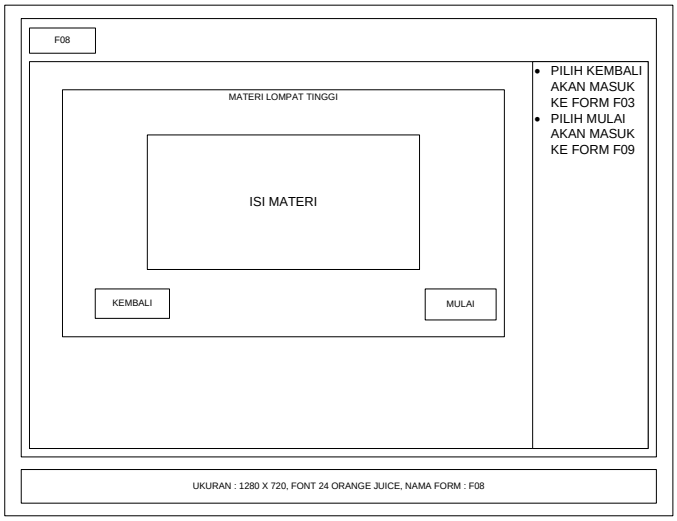
Berikut ini perancangan antarmuka materi lari gawang *Game* Olahraga Atletik pada gambar 3.45 berikut :



Gambar 3.45 Perancangan Antarmuka Materi Lari Gawang

3.2.3.6 Perancangan Antarmuka Materi Lompat Tinggi

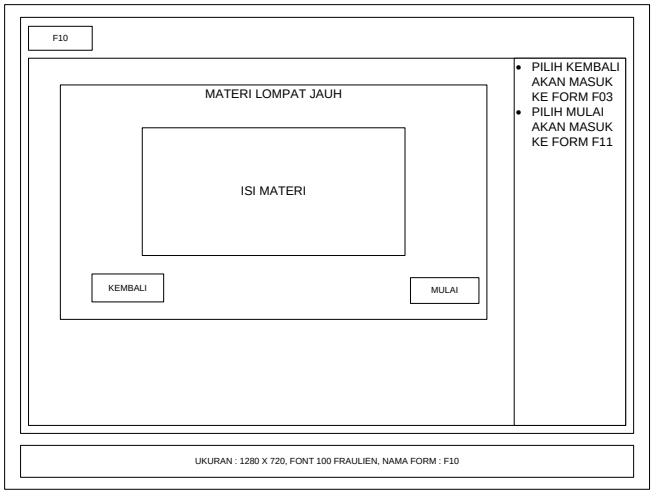
Berikut adalah perancangan antarmuka materi lompat tinggi *Game* Olahraga Atletik pada gambar 3.46 berikut :



Gambar 3.46 Perancangan Antarmuka Materi Lompat Tinggi

3.2.3.7 Perancangan Antarmuka Materi Lompat Jauh

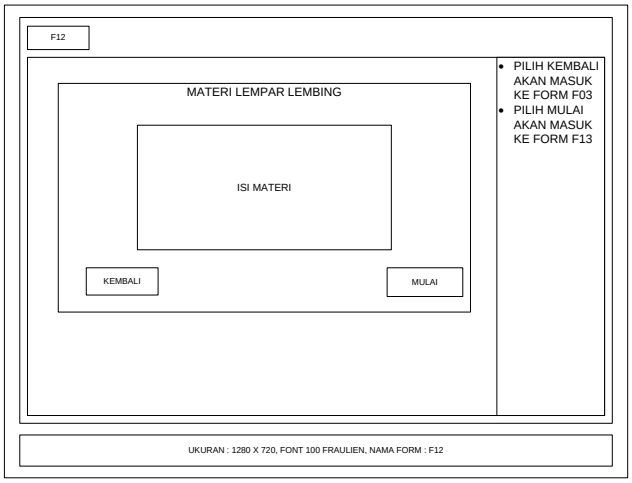
Berikut ini perancangan antarmuka materi lompat jauh *Game* Olahraga Atletik pada gambar 3.47 berikut :



Gambar 3.47 Perancangan Antarmuka Materi Lompat Jauh

3.2.3.8 Perancangan Antarmuka Materi Lempar Lembing

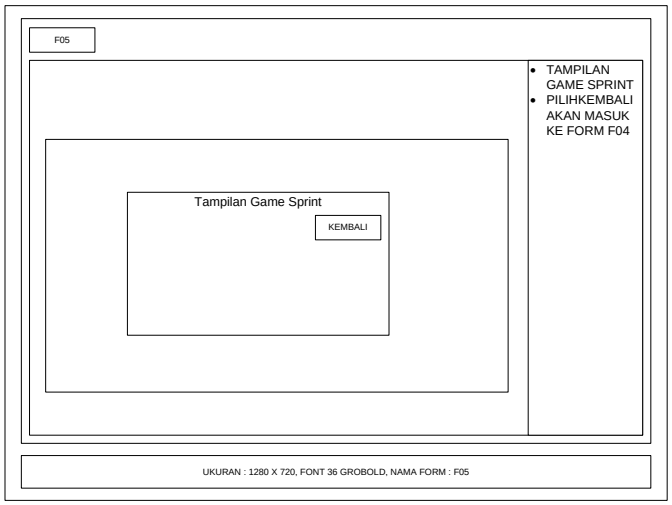
Berikut ini perancangan antarmuka materi lempar lembing *Game* Olahraga Atletik pada gambar 3.48 berikut :



Gambar 3.48 Perancangan Antarmuka Materi Lempar Lembing

3.2.3.9 Perancangan Antarmuka *Game Sprint*

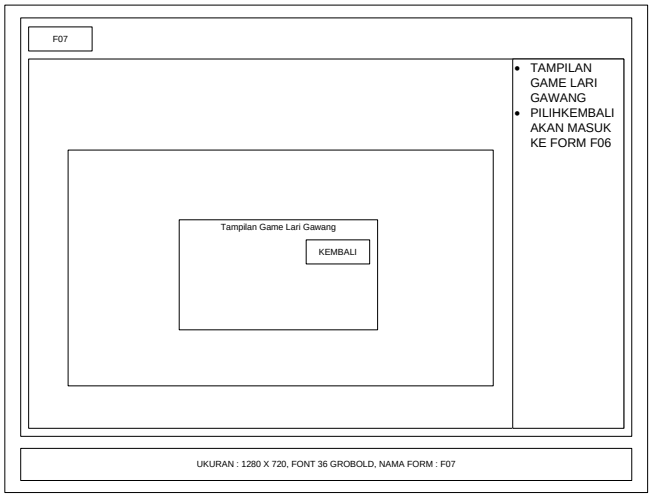
Berikut ini perancangan antarmuka *game sprint Game* Olahraga Atletik pada gambar 3.49 berikut :



Gambar 3.49 Perancangan Antarmuka *Game Sprint*

3.2.3.10 Perancangan Antarmuka *Game* Lari Gawang

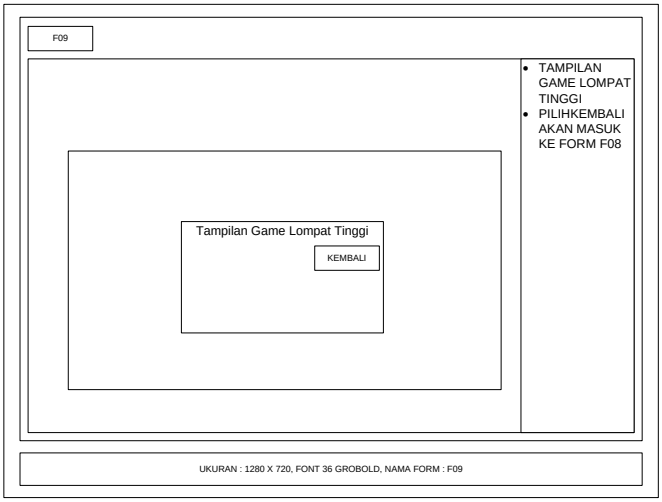
Berikut ini perancangan antarmuka *game* lari gawang *Game* Olahraga Atletik pada gambar 3.50 berikut :



Gambar 3.50 Perancangan Antarmuka *Game* Lari Gawang

3.2.3.11 Perancangan Antarmuka *Game* Lompat Tinggi

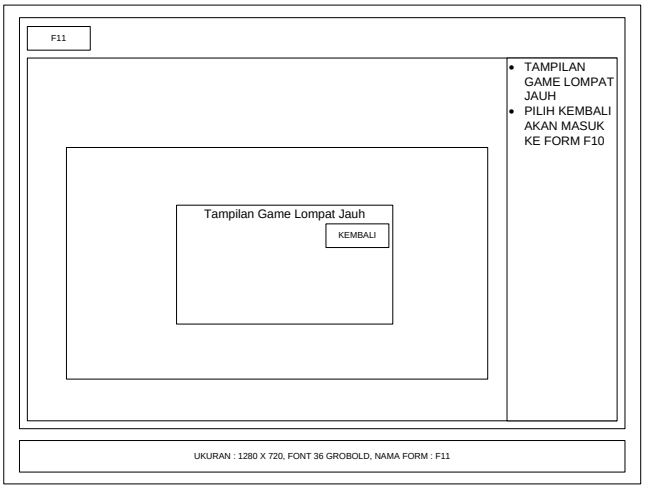
Berikut ini perancangan antarmuka *game* lompat tinggi *Game* Olahraga Atletik pada gambar 3.51 berikut :



Gambar 3.51 Perancangan Antarmuka *Game* Lompat Tinggi

3.2.3.12 Perancangan Antarmuka *Game* Lompat Jauh

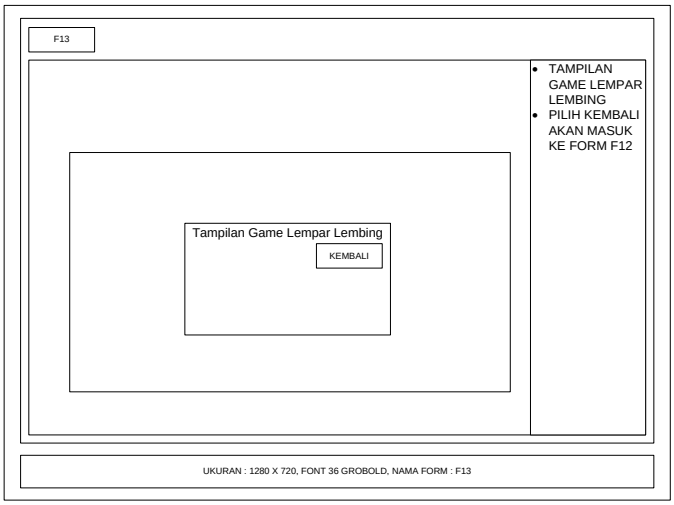
Berikut ini perancangan antarmuka *game l*ompat jauh *Game* Olahraga Atletik pada gambar 3.52 berikut :



Gambar 3.52 Perancangan Antarmuka *Game* Lompat Jauh

3.2.3.13 Perancangan Antarmuka *Game* Lempar Lembing

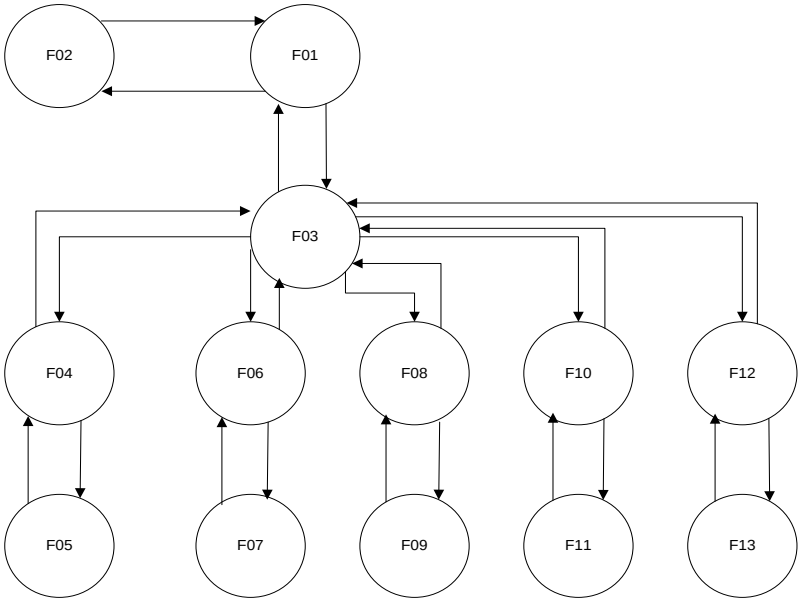
Berikut ini perancangan antarmuka *game* lempar lembing *Game* Olahraga Atletik pada gambar 3.53 berikut :



Gambar 3.53 Perancangan Antarmuka *Game* Lempar Lembing

3.2.4 Jaringan Semantik

Jaringan semantik merupakan jaringan data yang menunjukan hubungan antar berbagai objek. Berikut tampilan jaringan semantik *Game* Olahraga Atletik pada gambar 3.54.



Gambar 3.54 Jaringan Semantik