# BAB III

# ANALISIS DAN KEBUTUHAN ALGORITMA

## 3.1 Analisis Masalah

*Bayesian Network* merupakan pengembangan dari *Bayes Theorem* yang digunakan untuk mempelajari tentang probabilitas yang digambarkan pada sebuah model terhadap sebuah *Bayesian Network* dengan memspesifikasikan pada 2 hal. Yaitu Topologi Graph dan nilai parameter dari setiap *node* melalui data. Dalam game ini NPC dibuat sedemikian rupa agar dapat menyerupai *behavior Player* dalam memainkan permainan pingpong terutama dalam melakukan serangan. Serangan yang dilancarkan oleh NPC yang bersifat tidak pasti dalam menentukan arah. Dengan ini diharapkan *Bayesian Network* dapat melakukan perubahan terhadap *behaviour* yang terjadi pada NPC dengan ditambahkannya unsur probabilitas kedalamnya, sehingga NPC tersebut menjadi cerdas dalam memainkan *game* pingpong tersebut.

## 3.2 Analisis *Game*

Analisis *game* merupakan penguraian penjelasan dari bagian – bagian pada game yang akan dikembangkan.

### 3.2.1 *Gameplay*

Pada awal *game* permainan akan ditampilkan menu utama yang terdapat pilihan *play game* dan *exit*. *Player* dapat memulai permainan dengan memilih menu *play game* dimana *Player* akan langsung dihadapkan dengan musuh, disini *Player* memulai permainan dengan melakukan servis, selama permainan berlangsung , *player* dapat menggerakan karakter dengan tombol yang tersedia dan dapat juga melakukan *pause* untuk menghentikan permainan sementara dengan menekan tombol *pause*.

Pada menu *pause*, Player dapat melanjutkan permainan, memulai permainan dari awal atau menuju ke menu utama. Ketika bola melewati *player* maka musuh akan mendapatkan poin, begitupun juga sebaliknya.

### 3.2.2 NPC

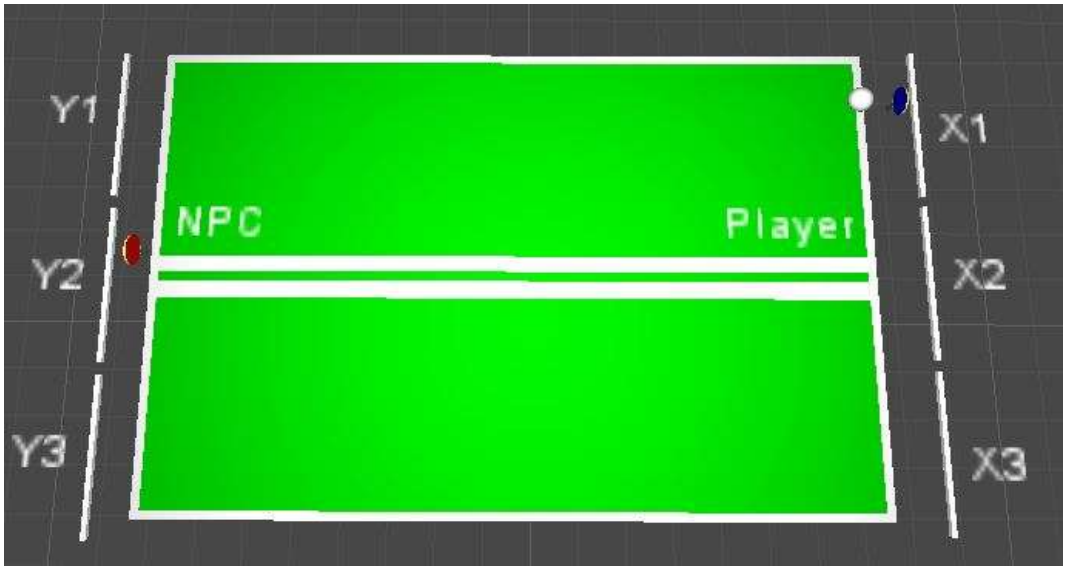
NPC yang ikut bermain dalam permainan menjadi lawan bagi *player* tersebut. NPC disini berperilaku seperti Player dimana NPC dapat memainkan permainan pingpong baik dalam melakukan penerimaan bola maupun dalam melakukan pemukulan bola.

## 3.3 Analisis Masukan

Bayesian Network merupakan sebuah model grafik probabilitas dimana setiap *node* saling terhubung. Variabel *node* X1, X2, X3, Y1, Y2, dan Y3 dimana nilai pada variabel tersebut didapat dari sebuah *Conditional Probability Table* (CPT) yang nilainya didapat dari Probabilitas Kondisional sesuai dengan kondisi yang terjadi pada alam sekitar atau pada kasus tertentu.

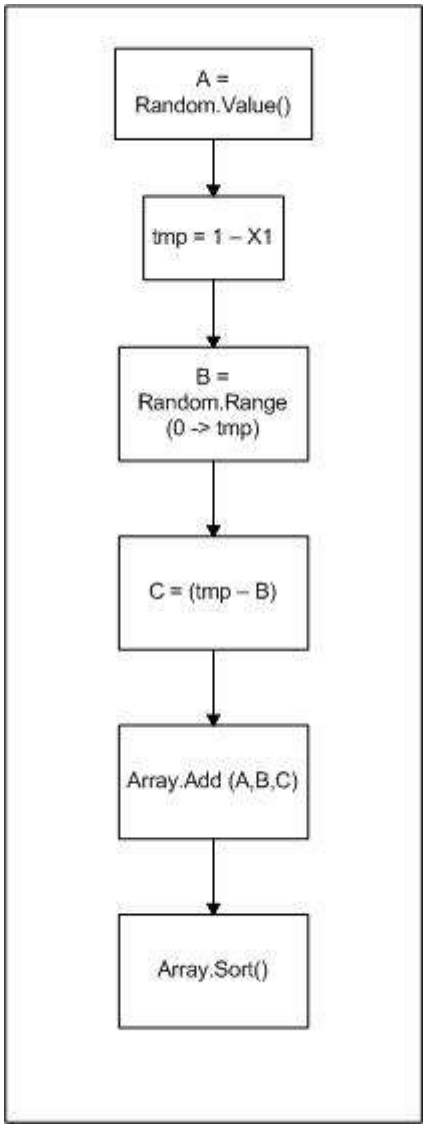
### 3.3.1 Analisis *Node*

Kasus yang terjadi dimana sebuah NPC melakukan penembakan bola, NPC akan menentukan arah penembakan oleh NPC. Area permainan dibagi 3 yaitu X1, X2, dan X3 dimana ketiga area tersebut memiliki sudut arah penembakan tersendiri sesuai dengan posisi player, ilustrasi area permainan dapat dilihat pada gambar 3.5.



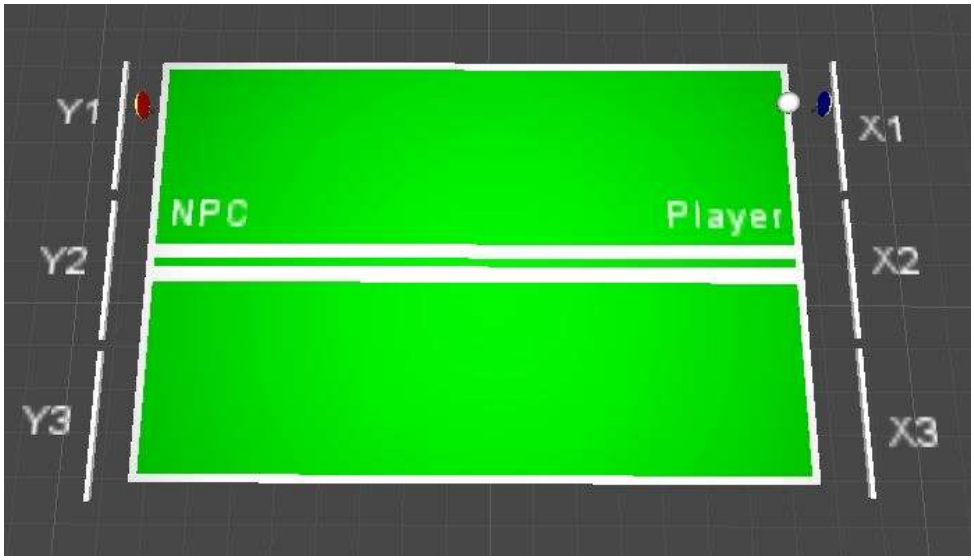

***Gambar 3.5 Ilustrasi Area Permainan***

Nilai pada ketiga *node* tersebut yang ada pada gambar 3.5 didapat secara acak (*random*) dimana terdapat batasan pada proses pengacakan berdasarkan dengan kondisi tertentu. Hal tersebut mengacu kepada posisi Player dan NPC. Posisi Player memiliki nilai kondisional paling kecil diantara area yang lain karena dalam kondisi tersebut penembakan harus dilakukan dengan selisih jarak jauh dari jangkauan Player. Kondisi dimana *player* berada pada X1 dan posisi NPC pada Y1 maka didapat batasan X3 < X2 < X1. Flowchart untuk batasan tersebut dapat dilihat pada gambar 3.6



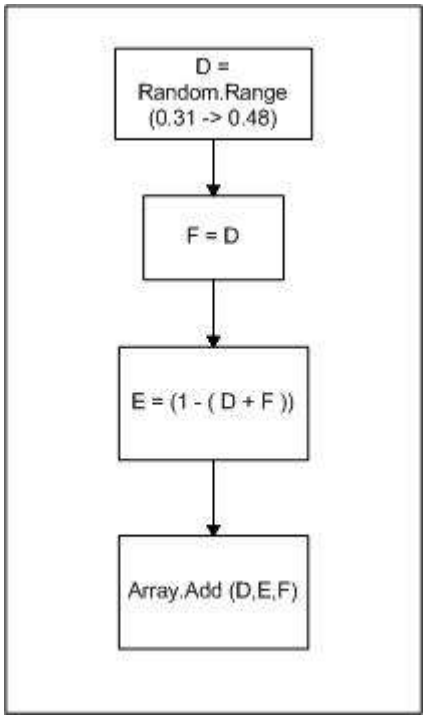

***Gambar 3.6 Flowchart Random SideNode***

Sebagai contoh akan diilustrasikan pada gambar 3.7 dimana posisi Player berada pada X1 dan NPC pada Y1.



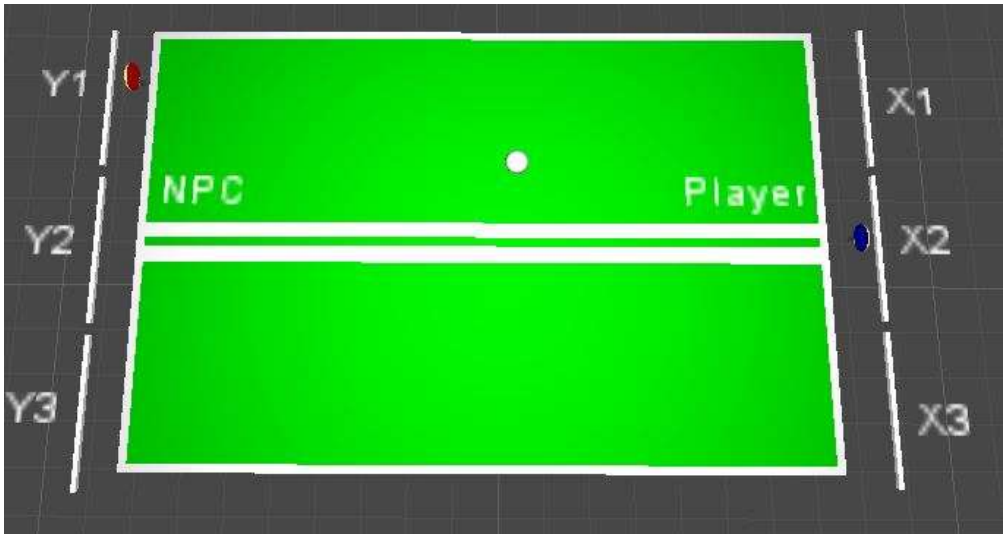

***Gambar 3.7 Ilustrasi player pada X1***

Nilai kondisional pada kondisi tersebut sesuai dengan batasan pengacakannya adalah X1 = 0.122 , X2 = 0.312, X3 = 0.566. Hal tersebut berlaku apabila posisi *player* berada pada X1 dan posisi NPC berada pada Y1. Kondisi dimana *player* berada pada X2 dan posisi NPC pada Y1 maka didapat batasan X1 > X2 < X3. Flowchart untuk batasan tersebut dapat dilihat pada gambar 3.8.




***Gambar 3.8 Flowchart Random MidNode***

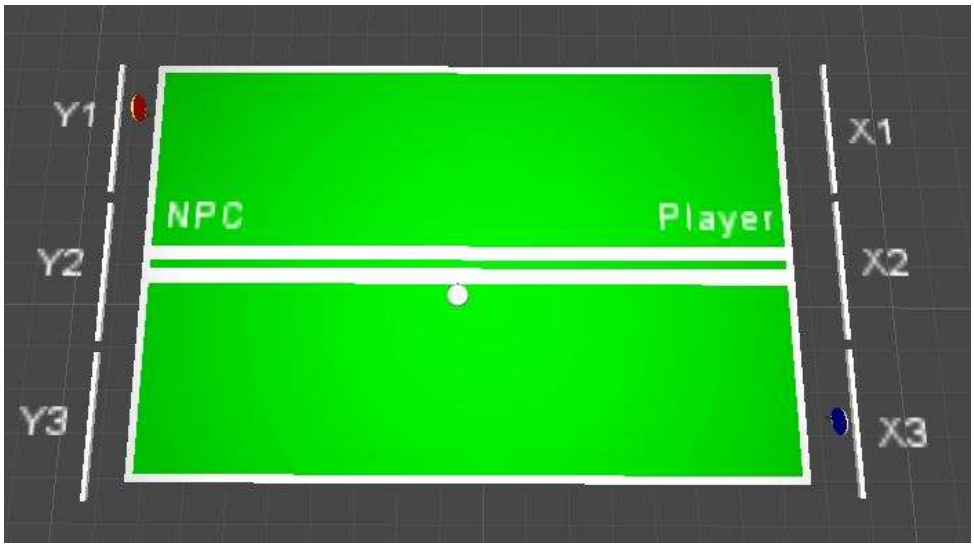
Sebagai contoh akan diilustrasikan pada gambar 3.9 dimana posisi Player berada pada X2 dan NPC pada Y1.




***Gambar 3.9 Ilustrasi player pada X2***

Nilai kondisional pada kondisi tersebut sesuai dengan batasan pengacakannya adalah X1 = X1 = 0.36, X2 = 0.28, X3 = 0.36. Kondisi dimana

*player* berada pada X3 dan NPC berada pada Y1 didapat batasan X3 < X2 < X1 serta flowchart yang sama pada gambar 3.5. Sebagai contoh akan diilustrasikan pada gambar 3.10.




***Gambar 3.10 Ilustrasi player pada X3***

Pembagian 3 sudut permainan dipengaruhi oleh lebar dari meja pingpong terhadap vektor y, jika diketahui meja pingpong berada pada vektor y = -90 sampai dengan 90 maka masing – masing sudut memiliki 60 atau lebar meja pingpong sama dengan y = 180, maka agar dapat terbagi kedalam tiga sudut, masing masing sudut mendapat nilai y = 60 dengan titik tengah dimulai dari 0. Berikut ini adalah pembagian sudut vektor y terhadap masing – masing node.

Lebar sudut meja pingpong $y = \frac{180}{3} = 60.$

Y1 = NPC berada pada sudut vektor y = 31 sampai dengan 90

Y2 = NPC berada pada sudut vektor y = -30 sampai dengan 30

Y3 = NPC berada pada sudut vektor y = -31 sampai dengan -90

X1 = Player berada pada sudut vektor y = 31 sampai dengan 90

X2 = Player berada pada sudut vektor y = -30 sampai dengan 30

X3 = Player berada pada sudut vektor y = -31 sampai dengan -90

*Conditional Probability Table* node X terhadap *parent* Y pada ilustrasi kasus di atas dapat dilihat pada tabel 3.1 dan node *parent* Y di tabel 3.2. Nilai yang didapat di generate secara random dengan ketentuan sesuai pada ilustrasi gambar 3.10 dimana nilai *node parent* Y didapat sesuai dengan ketergantungan posisi NPC saat melakukan penembakan dan *node child* atau *node* X1, X2, X3 yang digenerate dengan ketentuan nilai yang paling besar adalah nilai dengan *node* yang paling jauh dengan *player*.

***Tabel 3.1 Node X terhadap Parent Y1***

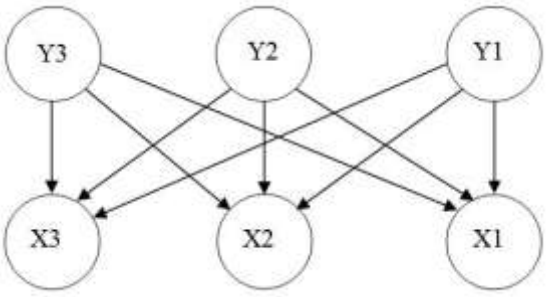
| Player / NPC | X1 | X2 | X3 |
|---|---|---|---|
| **Y1** | 0.122 | 0.312 | 0.566 |
| **Y2** | 0.36 | 0.28 | 0.36 |
| **Y3** | 0.566 | 0.312 | 0.122 |

***Tabel 3.2 Node Parent Y***

| Y1 | Y2 | Y3 |
|---|---|---|
| 0.333 | 0.333 | 0.333 |

### 3.3.2 Analisis Jaringan Bayes

Masing – masing *node* memiliki nilai berdasarkan pengacakannya, maka berikut ini adalah jaringan bayes berdasarkan pada *node* X1, X2, dan X3 dimana masing – masing node memiliki *parent* node Y1, Y2, dan Y3 yang dapat dilihat pada gambar 3.11




***Gambar 3.1 Jaringan Bayes Game Pingpong***

### 3.4 Analisis Algoritma

*Flowchart Bayesian Network* yang ada pada gambar 2.2 terdapat perhitungan probabilitas bayes menggunakan data pada hasil dari inisialisasi node dan edge kedalam *Conditional Probability Table*. Diasumsikan dalam permainan, NPC akan memukul bola kepada *player*. Hal tersebut berdasarkan nilai pada probabilitas *Teorema Bayes* terhadap *Conditional Probability Table* yang terlihat pada tabel 3.1 dan 3.2 sebelumnya. Berikut ini adalah penyelesaian contoh kasus yang ada pada gambar 3.5 menggunakan *Teorema Bayes* dimana posisi *player* berada pada X1 dan posisi NPC berada pada Y1.

$$P(Y1 \mid X1) = \frac{P(X1 \mid Y1)\,P(Y1)}{P(X1)}$$

$$= \frac{P(X1 \mid Y1)\,P(Y1)}{P(X1 \mid Y1)\,P(Y1) + P(X1 \mid Y2)\,P(Y2) + P(X1 \mid Y3)\,P(Y3)}$$

$$= \frac{0.122\,(0.3)}{0.122\,(0.333) + 0.36\,(0.333) + 0.566\,(0.333)}$$

$$= \frac{0.0366}{0.0366 + 0.1248 + 0.1698} = \frac{0.0366}{0.3312} = 0.110 = 11\%$$

Maka NPC memiliki kemungkinan 11% akan menembakkan bola ke sudut X1 terhadap Player. Sehingga dapat diambil kesimpulan bahwa NPC akan jarang menembak ke arah X1. Berikut ini adalah proses indexing array yang di dapat dari tabel 3.1 dan 3.2 masuk ke dalam *array*.

| 0.3 | 0.4 | 0.3 | 0.122 | 0.312 | 0.566 |
|---|---|---|---|---|---|
| A | A + 1 | A + 2 | A + 3 | A + 4 | A + 5 |

Pseudo code dari proses indexing di atas adalah sebagai berikut :

```
A : array[0..5] of integer
i : integer
```

```
BEGIN
      A[0] = 0.3
      A[1] = 0.4
      A[2] = 0.3
      A[3] = 0.122
      A[4] = 0.312
      A[5] = 0.566
END
```

Begitupun juga dengan probabilitas terhadap sudut X2 dimana P ( Y1 | X2 )

$$P\,(Y1 \mid X2) = \frac{P\,(X2 \mid Y1)\,P\,(Y1)}{P\,(X2)}$$

$$= \frac{P\,(X2 \mid Y1)\,P\,(Y1)}{P\,(X2 \mid Y1)\,P\,(Y1) + P\,(X2 \mid Y2)\,P\,(Y2) + P\,(X2 \mid Y3)\,P\,(Y3)}$$

$$= \frac{0.36\,(0.3)}{0.312\,(0.333) + 0.28\,(0.333) + 0.312\,(0.333)}$$

$$= \frac{0.108}{0.108 + 0.112 + 0.108} = \frac{0.108}{0.328} = 0.329 = 32{,}9\%$$

Maka NPC memiliki kemungkinan 32,9% akan menembakkan bola ke sudut X2 terhadap Player. Sehingga dapat diambil kesimpulan bahwa NPC akan menembak ke arah X2 dengan tempo normal atau sedang. Berikut ini adalah proses indexing array pada kasus P (Y1 | X2)

| 0.3 | 0.4 | 0.3 | 0.36 | 0.28 | 0.36 |
|---|---|---|---|---|---|
| A | A + 1 | A + 2 | A + 3 | A + 4 | N + 5 |

Pseudo code dari proses indexing di atas adalah sebagai berikut :

```
A : array[0..5] of integer
i : integer
```

```
BEGIN
      A[0] = 0.3
      A[1] = 0.4
      A[2] = 0.3
      A[3] = 0.36
      A[4] = 0.28
      A[5] = 0.36
END
```

Kemudian probabilitas terhadap sudut X3 dimana P (Y1 | X3)

$$P\,(Y1 \mid X3) = \frac{P\,(X3 \mid Y1)\,P\,(Y1)}{P\,(X3)}$$

$$= \frac{P\,(X3 \mid Y1)\,P\,(Y1)}{P\,(X3 \mid Y1)\,P\,(Y1) + P\,(X3 \mid Y2)\,P\,(Y2) + P\,(X3 \mid Y3)\,P\,(Y3)}$$

$$= \frac{0.566\,(0.3)}{0.566\,(0.333) + 0.36\,(0.333) + 0.122\,(0.333)}$$

$$= \frac{0.1698}{0.1698 + 0.1248 + 0.366} = \frac{0.1698}{0.3312} = 0.513$$

Maka NPC memiliki kemungkinan 51,3% akan menembakkan bola ke sudut X3 terhadap Player. Sehingga dapat diambil kesimpulan bahwa NPC akan sering menembak ke arah X3. Berikut ini adalah proses indexing array pada kasus P (Y1 | X3)

| 0.3 | 0.4 | 0.3 | 0.566 | 0.312 | 0.122 |
|---|---|---|---|---|---|
| A | A + 1 | A + 2 | A + 3 | A + 4 | N + 5 |

Pseudo code dari proses indexing di atas adalah sebagai berikut :

```
A : array[0..5] of integer
i : integer
```

```
BEGIN
      A[0] = 0.3
      A[1] = 0.4
      A[2] = 0.3
      A[3] = 0.566
      A[4] = 0.312
      A[5] = 0.122
END
```

Berdasarkan hasil analisis NPC memiliki kemungkinan memukul bola ke arah *player* dengan kemungkinan terbesar adalah X3 dengan nilai 54.5 % atau ke sudut kiri terhadap *player*.

## 3.5 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak memberikan gambaran mengenai kebutuhan dari sebuah perangkat lunak tersebut dibangun. Adapun kebutuhan perangkat lunak terbagi kedalam non fungsional dan fungsional, berikut adalah penjelasan pada setiap analisis tersebut

### 1.5.1 Kebutuhan Non Fungsional

Kebutuhan non-fungsional menggambarkan kebutuhan luar sistem yang diperlukan untuk menjalankan aplikasi yang dibangun. Adapun kebutuhan non-fungsional pada implementasi *Bayesian Network* di *game* pingpong meliputi kebutuhan perangkat keras dan kebutuhan perangkat lunak. Kebutuhan non-fungsional bertujuan agar aplikasi yang dibangun dapat digunakan sesuai dengan kebutuhan.

***Tabel 3.3 Kebutuhan Non Fungsional***

| No | Non– Fungsional | Keterangan |
|---|---|---|
| 1 | SRS – NF – 01 | Kebutuhan Perangkat Keras (*Hardware*) |
| 2 | SRS – NF – 02 | Kebutuhan Perangkat Lunak (*Software*) |

| | | |
|---|---|---|
| 3 | SRS – NF – 03 | Kebutuhan Pengguna |

#### 3.5.1.1 Kebutuhan Perangkat Keras

Menjalan suatu aplikasi diperlukan perangkat keras yang dapat mendukung proses kerja dari sistem itu sendiri. Adapun spesifikasi minimum perangkat keras komputer sebagai berikut :

1. Processor Pentium IV 2.8 GHZ
2. Kapasitas RAM 1 GB
3. VGA card 512 MB
4. Kapasitas Harddisk 80 GB
5. Monitor

#### 3.5.1.2 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 untuk menjalankan aplikasi game ini adalah sistem operasi minimal windows xp dengan DirectX 9 atau DirectX11. Game ini dikembangkan dengan menggunakan unity 3D 3.5 dan pembuatan assets menggunakan Google SketchUp.

#### 3.5.1.3 Kebutuhan Pemain

Kebutuhan pemain menentukan layak tidaknya pemain memainkan permainan ini. Berikut adalah sasaran pemain yang layak memainkan permainan ini :

1. Berdasarkan umur pengguna.
2. Pengguna dapat mempergunakan komputer, karena selain *Game* berbentuk aplikasi dekstop, *Game* pingpong juga memerlukan respon pengguna secara langsung terhadap komputer.

3. Pengguna yang tidak dapat membedakan warna yang satu dengan yang lainnya (buta warna) masih mampu menggunakan aplikasi ini, karena tidak ada indikator warna – warna khusus yang membedakan fungsional yang satu dengan fungsional yang lainnya. Pengguna cukup fokus terhadap objek bola, objek musuh, dan sudut tembakan.

### 3.5.2 Kebutuhan Fungsional

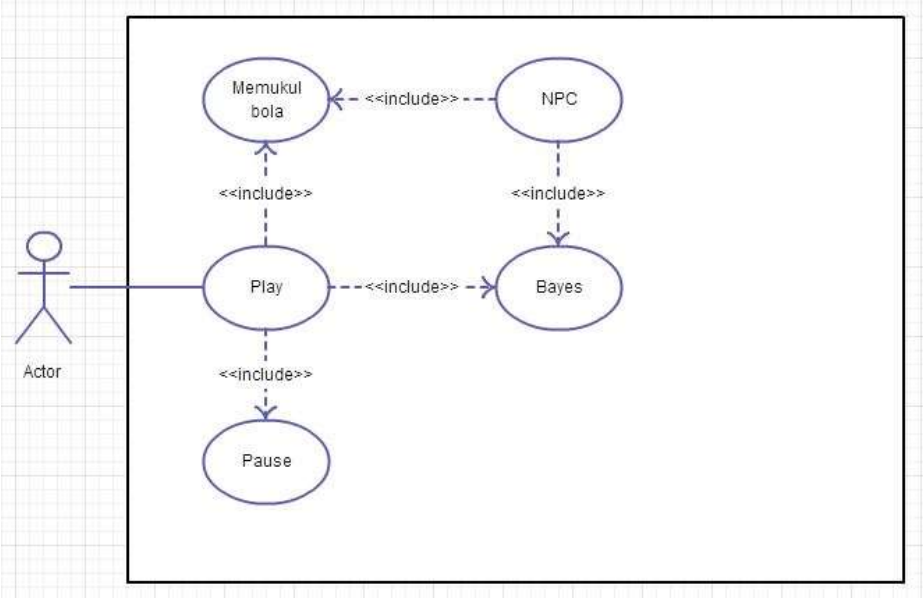
Identifikasi aktor dapat dilakukan dalam analisis berorientasi objek dengan menggunakan UML yaitu menentukan aktor atau Player sistem. Aktor dalam konteks UML menampilkan peran Player atau sesuatu diluar sistem yang dikembangkan dapat berupa perangkat keras, end user, sistem yang lain dan sebagainya.

***Tabel 3.4 Tabel Kebutuhan Fungsional***

| No | Fungsional | Keterangan |
|---|---|---|
| 1 | SRS – F – 01 | Player dapat bergerak, memukul bola dan mengarahkannya |
| 2 | SRS – F – 02 | NPC dapat bergerak, memukul bola dan mengarahkannya |
| 3 | SRS – F – 03 | Skema permainan hanya menyediakan bermain dan pause |
| 4 | SRS – F – 04 | NPC melakukan pemukulan bola berdasarkan proses Bayesian Network |
| 5 | SRS – F – 05 | Bayesian Network dapat dijadikan alternative terhadap player dalam melakukan penembakan bola |
| 6 | SRS – F – 06 | Permainan selesai ketika salah satu pemain mencapai poin 12 |

#### 3.5.2.1 Use Case Diagram

Use case diagram merupakan pemodelan untuk mendeskripsikan interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Use case diagram pada permainan game pingpong dapat dilihat pada gambar 3.12.




***Gambar 3.12 Use Case Diagram Game Pingpong***

### 3.5.2.1.1 Definisi Aktor

Definisi Aktor berfungsi untuk menjelaskan aktor yang terdapat pada use case diagram. Definisi aktor diterangkan pada tabel 3.5 berikut.

***Tabel 3.5 Definisi Aktor***

| No | Aktor | Deskripsi |
|---|---|---|
| 1 | *Player* | Orang yang memainkan game |

### 3.5.2.1.2 Definisi Use Case

Definisi use case berfungsi untuk menjelaskan proses yang terdapat pada setiap use case. Definisi use case diterangkan pada tabel 3.6 :

***Tabel 3.6 Definisi Use Case***

| No | Use Case | Deskripsi |
|---|---|---|
| 1 | *Play* | Player memilih Play dari menu utama. Player memainkan permainan pingpong. Player mendapatkan poin bila bola melewati NPC. Permainan berakhir bila |

| No | Use Case | Deskripsi |
|---|---|---|
| | | telah menyelesaikan 2 ronde. |
| 2 | Memukul bola | Player memukul bola ke arah sudut lawan saat permainan berlangsung baik dengan ataupun tidak dengan bayes. |
| 3 | NPC | Sebuah sistem yang berperan sebagai lawan dalam permainan pingpong |
| 4 | Bayes | Sebuah sistem yang memberikan keputusan dalam melakukan penembakan. |
| 5 | Pause | Player dapat menghentikan permainan untuk sementara |

### 3.5.2.1.3 Skenario Use Case

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

***Tabel 3.7 Play***

| Identifikasi | | |
|---|---|---|
| Nomor | 1 | |
| Nama | Play | |
| Tujuan | Player bermain game dengan menggerakan karakter | |
| Deskripsi | Player menggerakan karakter untuk bermain pingpong | |
| Aktor | Player | |
| Skenario Utama | | |
| Kondisi Awal | Player telah memilih menu play dan game siap dimainkan. | |
| Aksi Aktor | | Reaksi Sistem |
| 1. Player mengarahkan dan menekan tombol yang disediakan untuk menggerakkan karakter di dalam game | | |

| Identifikasi | |
|---|---|
| | 2. Karakter bergerak sesuai dengan arahan dan tombol yang ditekan oleh player |
| Kondisi Akhir | Karakter bergerak sesuai dengan tombol yang ditekan oleh<br>player sampai permainan berakhir |

***Tabel 3.8 Memukul Bola***

| Identifikasi | | |
|---|---|---|
| Nomor | 2 | |
| Nama | Memukul bola | |
| Tujuan | Player memukul bola untuk bermain pingpong | |
| Deskripsi | Player memukul bola melalui sebuah karakter | |
| Aktor | Player | |
| Skenario Utama | | |
| Kondisi Awal | Player telah memilih menu play dan game tengah berlangsung. | |
| Aksi Aktor | | Reaksi Sistem |
| 1. Player mengarahkan dan menekan tombol yang disediakan untuk menggerakkan karakter dan menembakkan bola di dalam game. Player juga bisa mengikuti saran dari perhitungan Bayesian Network sebagai alternatif penembakan bola | | |
| | | 2. Bola bergerak sesuai dengan arahan dan tombol yang ditekan oleh player |
| Kondisi Akhir | | Bola bergerak sesuai dengan tombol yang ditekan oleh player sampai permainan berakhir |

***Tabel 3.9 NPC***

| Identifikasi |
|---|

| Identifikasi | | |
|---|---|---|
| Nomor | 3 | |
| Nama | NPC | |
| Tujuan | Menjadi lawan yang semakin cerdas dalam permainan pingpong | |
| Deskripsi | Sebuah sistem yang berperan sebagai lawan dalam permainan pingpong | |
| Aktor | | |
| Skenario Utama | | |
| Kondisi Awal | Player telah memilih menu play dan game tengah berlangsung. | |
| Aksi Aktor | | Reaksi Sistem |
| 1. Player memulai permainan | | |
| | | 2. NPC mengikuti arah tembakan player |
| | | 3. NPC menembak ke arah player sesuai dengan sudut yang ditentukan oleh Bayes |
| 4. Player kembali melanjutkan permainan | | |
| | | 5. NPC mengikuti arah tembakan player |
| | | 6. NPC mempelajari nilai Bayes yang sebelumnya, dan dilakukan proses iterasi |
| | | 7. NPC menembak ke arah player sesuai dengan hasil iterasi pada Bayes |
| Kondisi Akhir | | NPC memukul bola |

***Tabel 3.10 Bayes***

| Identifikasi | |
|---|---|
| Nomor | 4 |
| Nama | Bayes |
| Tujuan | Menjadikan alternatif bantuan dalam memukul bola |
| Deskripsi | Player mendapatkan alternatif bantuan dalam memukul bola dan NPC menjadi semakin cerdas dalam menembak bola |

| Identifikasi | |
|---|---|
| Aktor | Player |
| Skenario Utama | |
| Kondisi Awal | Player telah memilih menu play dan game tengah berlangsung. |
| Aksi Aktor | Reaksi Sistem |
| 1. Player memulai permainan | |
| | 2. Sistem melakukan pengambilan posisi player |
| | 3. Sistem memulai pengacakan nilai |
| | 2. Sistem memulai perhitungan Bayes |
| | 3. Sistem melakukan iterasi setelah perhitungan yang pertama selesai. |
| | 4. Iterasi dilakukan hingga iterasi tersebut berhenti |
| Kondisi Akhir | Iterasi berhenti ketika nilai iterasi sebelumnya sesuai dengan iterasi yang sekarang |

***Tabel 3.11 Pause***

| Identifikasi | |
|---|---|
| Nomor | 5 |
| Nama | Pause |
| Tujuan | Menghentikan permainan untuk sementara |
| Deskripsi | Player dapat menghentikan permainan dan kemudian melanjutkan permainan ketika player menekan tombol resume |
| Aktor | Player |
| Skenario Utama | |
| Kondisi Awal | Permainan sudah berjalan dan sedang dimainkan |
| Aksi Aktor | Reaksi Sistem |
| 1. Player menekan tombol pause | |
| | 2. Menampilkan menu yang menanyakan player untuk melanjutkan atau mengakhiri |

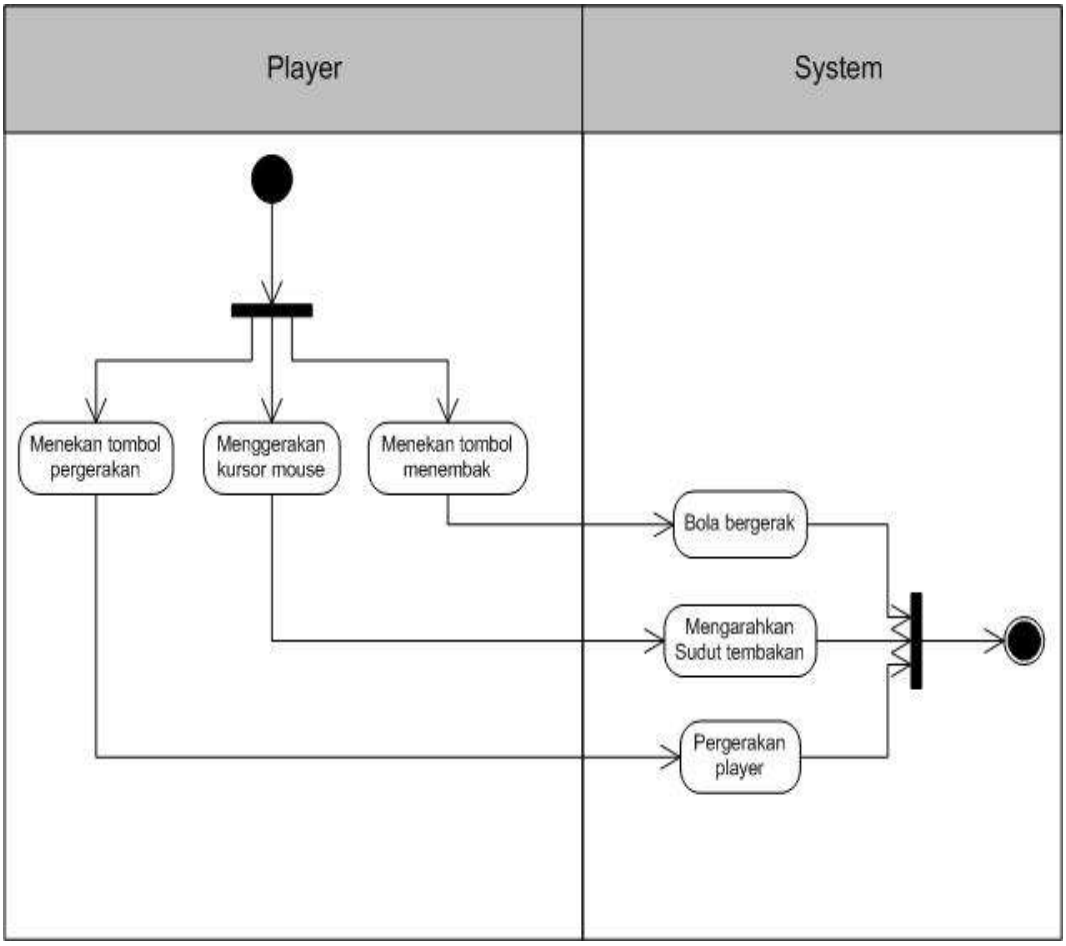
| Identifikasi | |
|---|---|
| | permainan |
| 3. Player memilih menu yang diinginkan | |
| | 4. Jilka player memilih resume maka permainan akan dikembalikan seperti sebelum tombol pause ditekan. |
| Kondisi Akhir | Permainan kembali seperti semula atau keluar dari permainan |

### 3.5.2.2 Activity Diagram

Activity diagram menggambarkan berbagai aliran aktivitas dalam sistem yang sedang dirancang. Berikut ini beberapa activity diagram yang terdapat pada game ini yaitu :

A. Activity Diagram *Play*

Activity Diagram *Play* dapat dilihat pada gambar 3.13



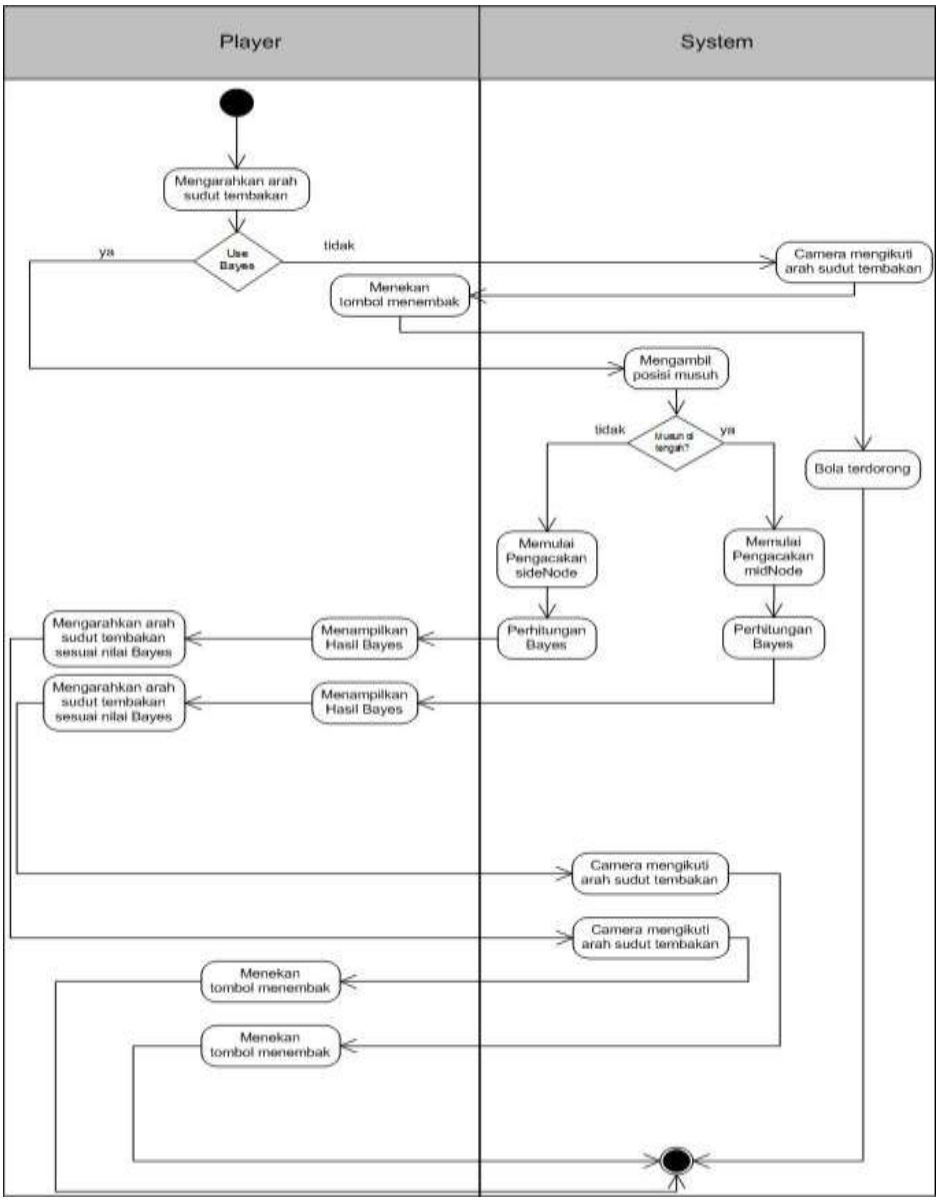

***Gambar 3.13 Activity Diagram Play***

***Tabel 3.12 Penjelasan Play***

| Deskripsi | Menjelaskan proses Play |
|---|---|
| Normal Flow | 1. Player menggerakkan kursor atau menekan tombol yang disediakan<br>2. Sistem merespon berdasarkan input yang dipilih oleh Player |

B. Activity Diagram Memukul bola

Activity Diagram Memukul bola dapat dilihat pada gambar 3.14




***Gambar 3.14 Activity Diagram Memukul Bola***

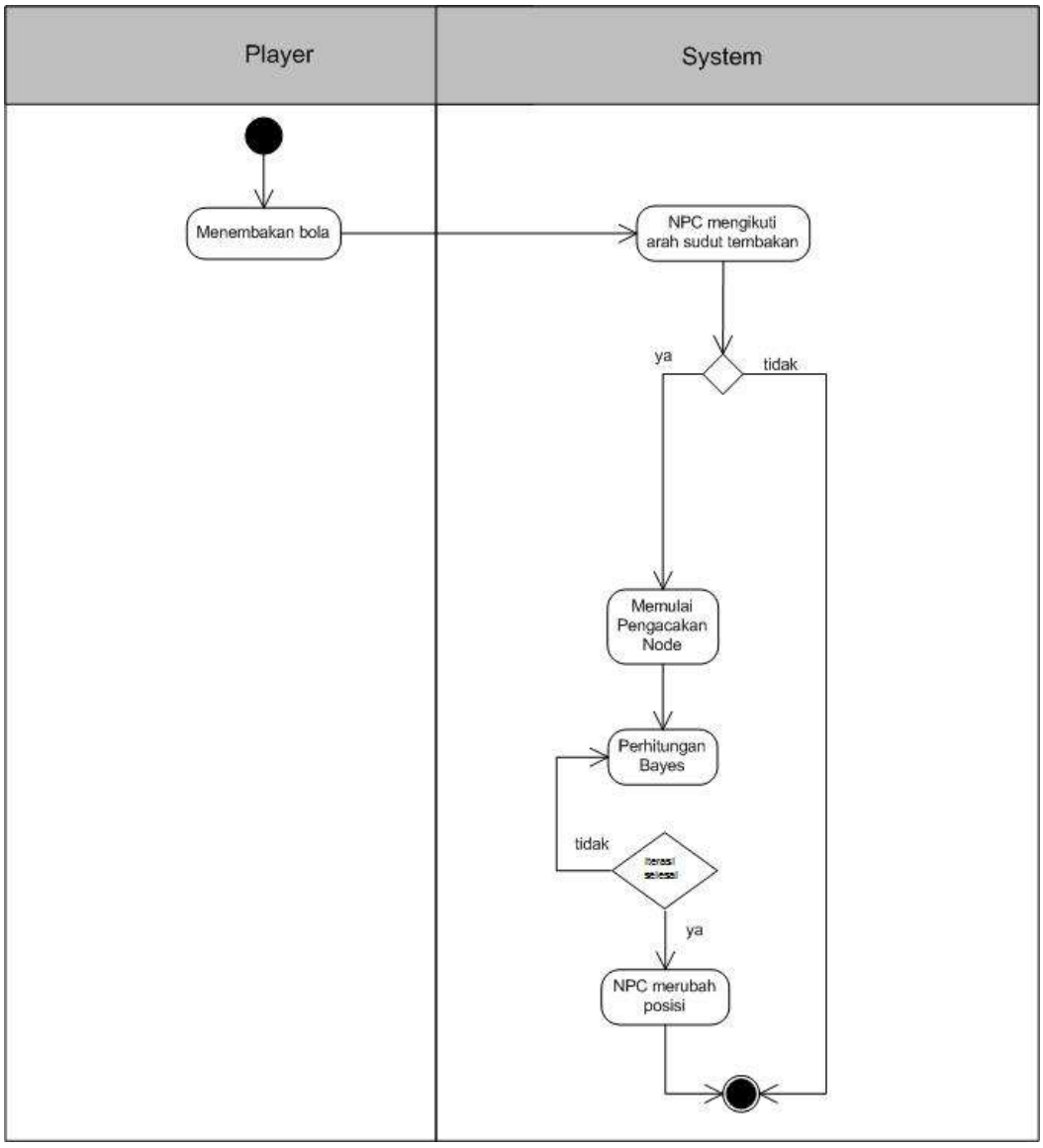
***Tabel 3.13 Penjelasan Memukul bola***

| Deskripsi | Menjelaskan proses memukul bola |
|---|---|
| Normal Flow | 1. Player mengarahkan arah sudut tembakan<br>2. Camera mengikuti arah sudut tembakan berdasarkan arahan |

| Deskripsi | Menjelaskan proses memukul bola |
|---|---|
| | player<br>3. Player menekan tombol tembak<br>4. Maka bola akan terdorong sesuai dengan arahan sudut tembakan player |

C. Activity Diagram NPC

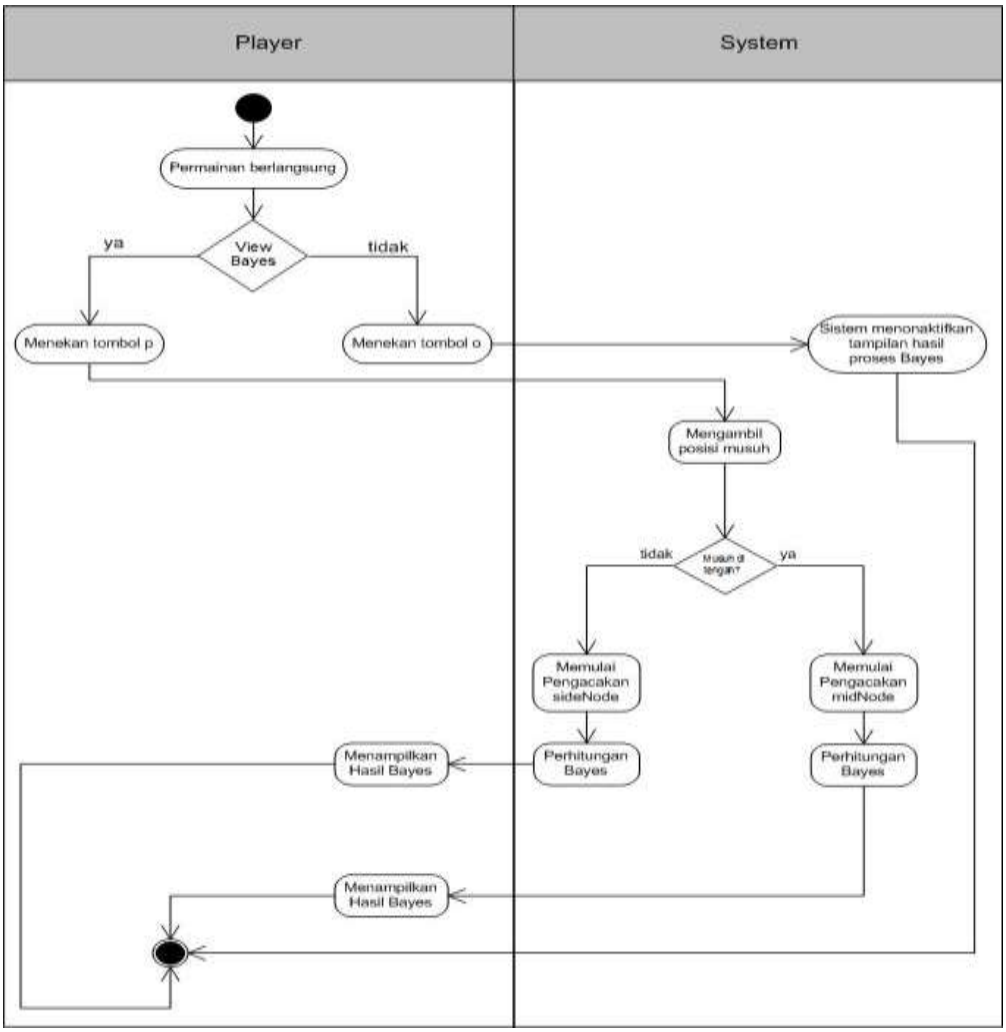
Activity Diagram *Bayes* dapat dilihat pada gambar 3.15




***Gambar 3.15 Activity Diagram NPC***

D. Activity Diagram *Bayes*

Activity Diagram *Bayes* dapat dilihat pada gambar 3.16




***Gambar 3.16 Activity Diagram Bayes***

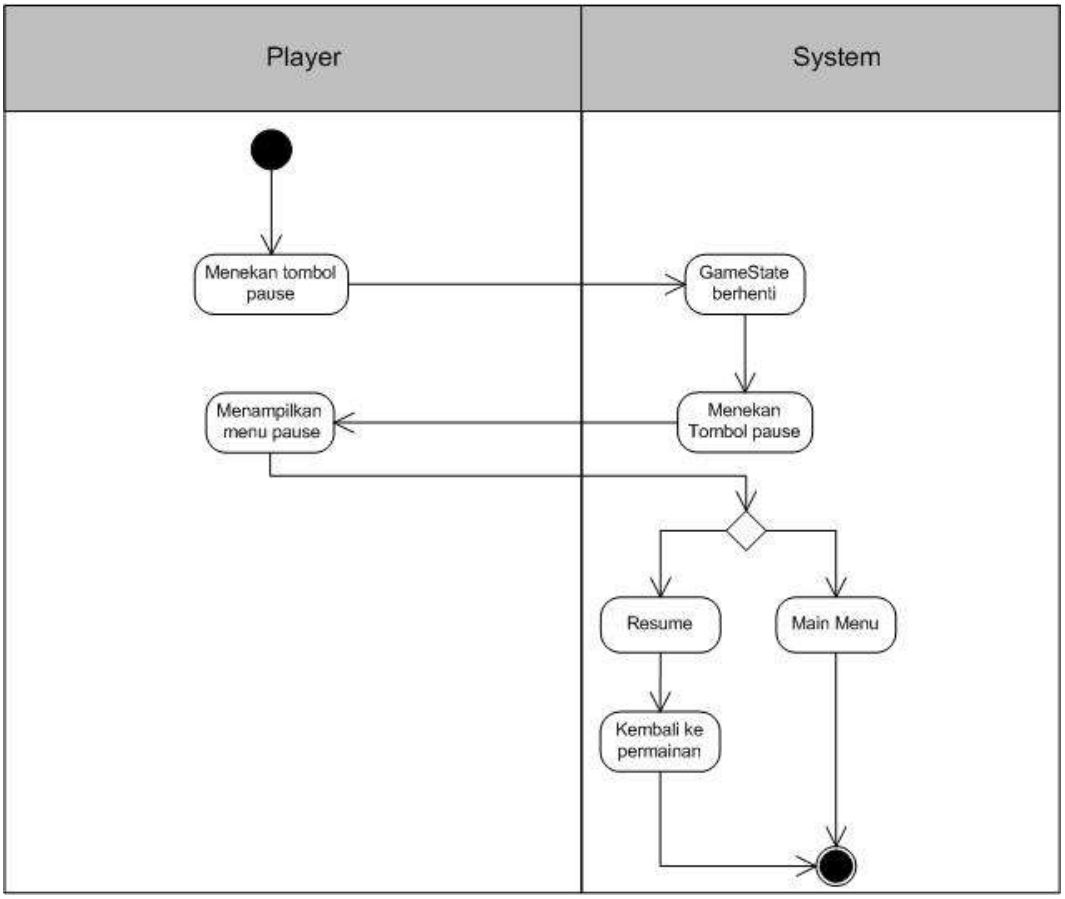
***Tabel 3.14 Penjelasan Bayes***

| Deskripsi | Menjelaskan proses bayes |
|---|---|
| Normal Flow | 1. Player mengaktifkan bayes<br>2. Sistem menampilkan hasil proses bayes<br>3. Player mendapat pilihan apakah akan mengikuti hasil dari alternatif bayes atau tetap |

| Deskripsi | Menjelaskan proses bayes |
|---|---|
| | menentukan tembakan berdasarkan pilihan diri sendiri |

E. Activity Diagram *Pause*

Activity Diagram *Pause* dapat dilihat pada gambar 3.17




***Gambar 3.17 Activity Diagram Pause***

***Tabel 3.15 Penjelasan* Pause**

| Deskripsi | Menjelaskan proses *Pause* |
|---|---|
| Normal Flow | 1. Player menekan tombol pause<br>2. Player masuk ke dalam menu pause<br>3. Jika memilih main menu player mengakhiri permainan dan masuk main menu |

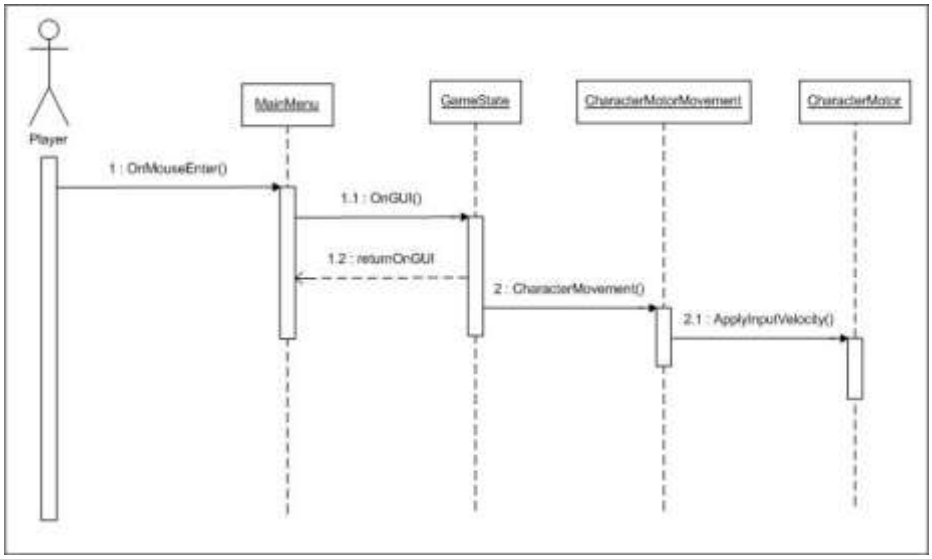
| | |
|---|---|
| Deskripsi | Menjelaskan proses *Pause* |
| | 4. Jika memilih resume player kembali ke permainan |

### 3.5.2.3 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 :

A. *Sequence* Diagram *Play*

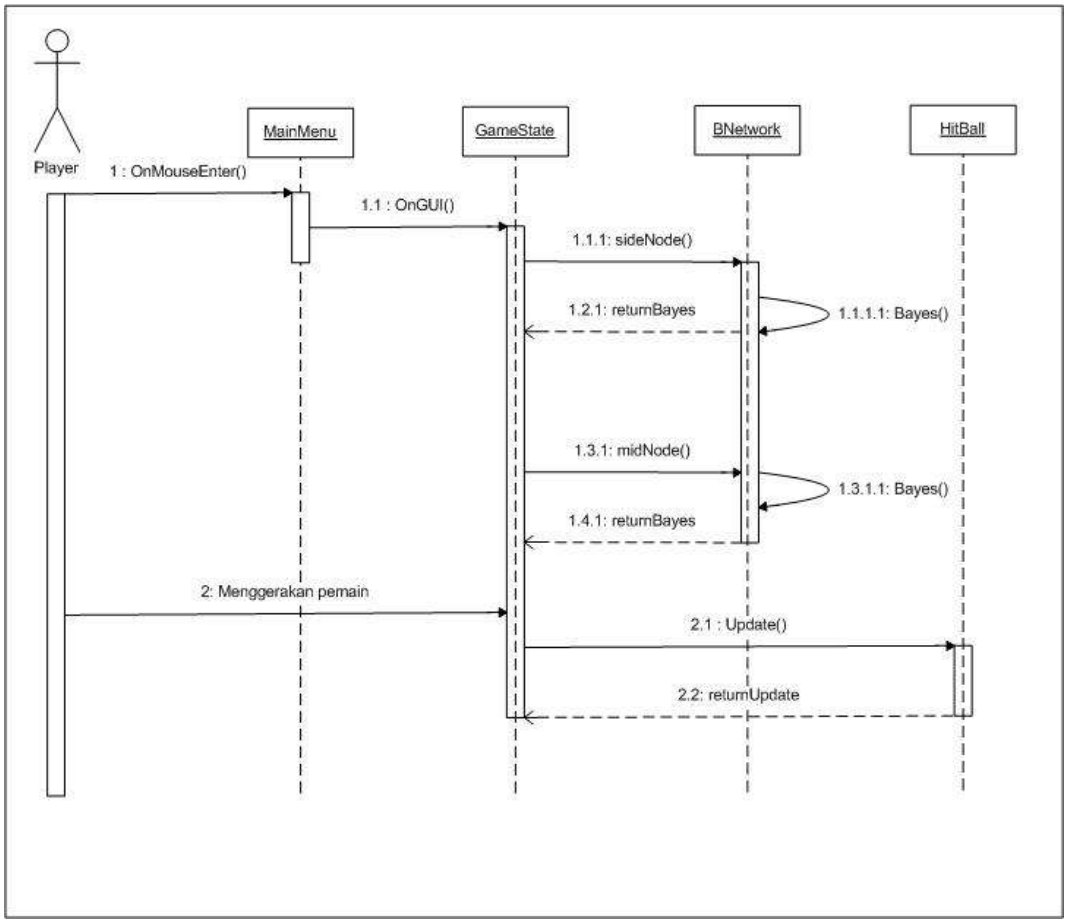
*Sequence* Diagram Bermainan permainan dapat dilihat pada gambar 3.18 dimana aktor berinteraksi dengan *GameState*, *CharacterMotorMovement*, *CharacterMotor* dan *MainMenu*




***Gambar 3.18 Sequence Diagram Play***

B. *Sequence* Diagram Memukul bola

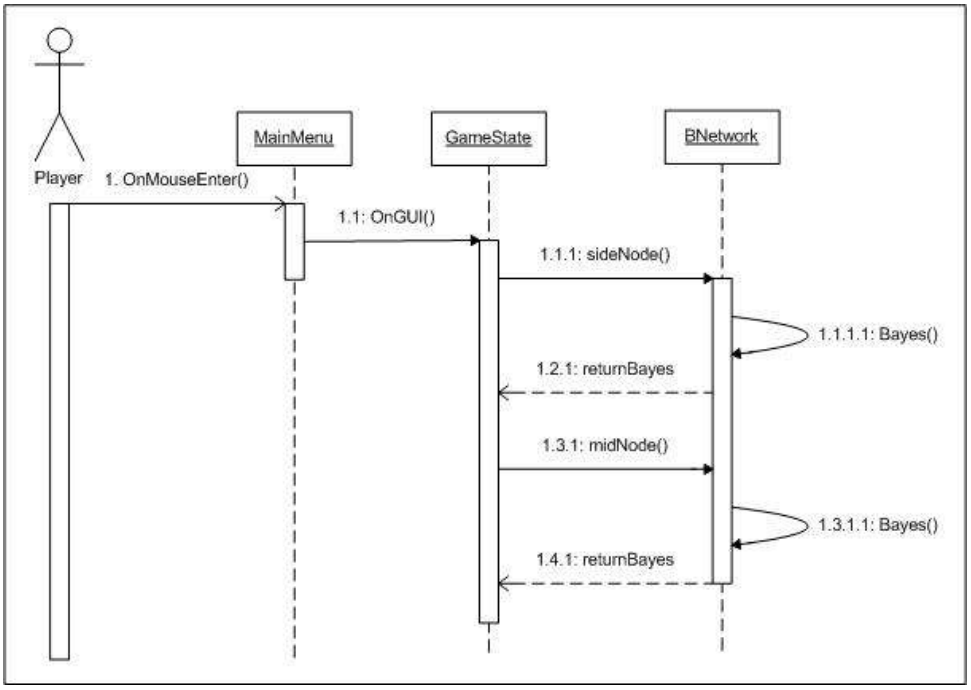
*Sequence* Diagram Memukul bola dapat dilihat pada gambar 3.19 dimana aktor berinteraksi dengan *GameState, BNetwork, CharacterMotorMovingPlatform* dan *CharacterMotor*




***Gambar 3.19 Sequence Diagram Memukul Bola***

C. *Sequence* Diagram *Bayes*

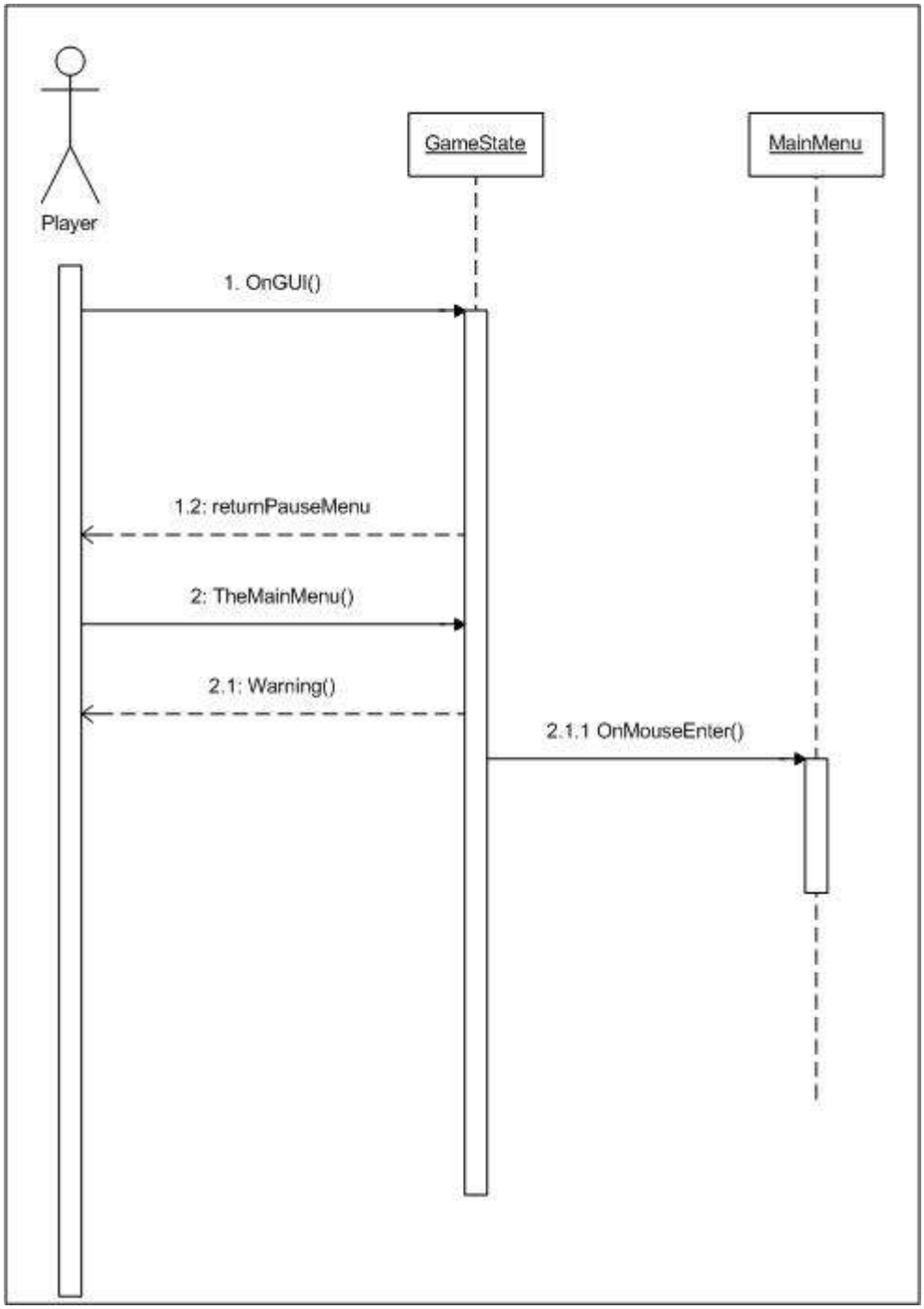
*Sequence* Diagram Pergerakan karakter dapat dilihat pada gambar 3.20 dimana aktor berinteraksi dengan MainMenu ,*BNetwork* dan *GameState*.




***Gambar 3.20 Sequence Diagram Bayes***

D. *Sequence* Diagram *Pause*

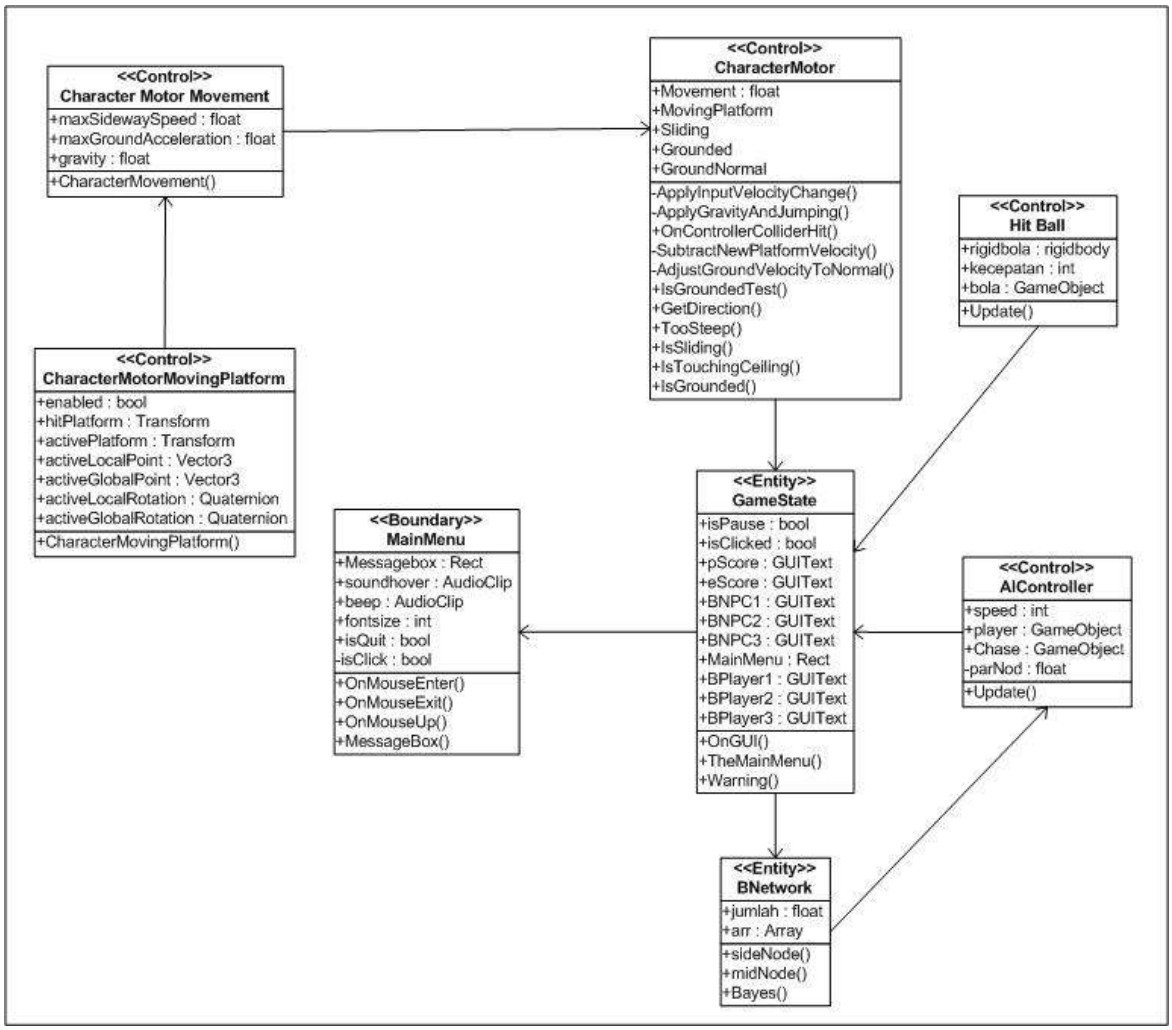
*Sequence* Diagram Menunda Permainan dapat dilihat pada gambar 3.21 dimana aktor berinteraksi dengan *MainMenu* dan *GameState*.




***Gambar 3.21 Sequence Diagram Pause***

### 3.5.2.4 Class Diagram

Diagram ini menunjukkan sekumpulan kelas, *interface* dan kolaborasi keterhubungannya. Class Diagram yang ada pada permainan pingpong dapat dilihat pada gambar 3.22




***Gambar 3.22 Class Diagram Game Pingpong***

## 3.6 Perancangan Sistem

Perancangan sistem merupakan suatu pengembangan perangkat lunak yang dilakukan setelah melalui tahapan analisis. Dalam tahap ini digambarkan rancangan sistem yang akan dibangun sebelum dilakukan pengkodean ke dalam suatu bahasa pemrograman.

### 3.6.1 Perancangan Komponen Permainan

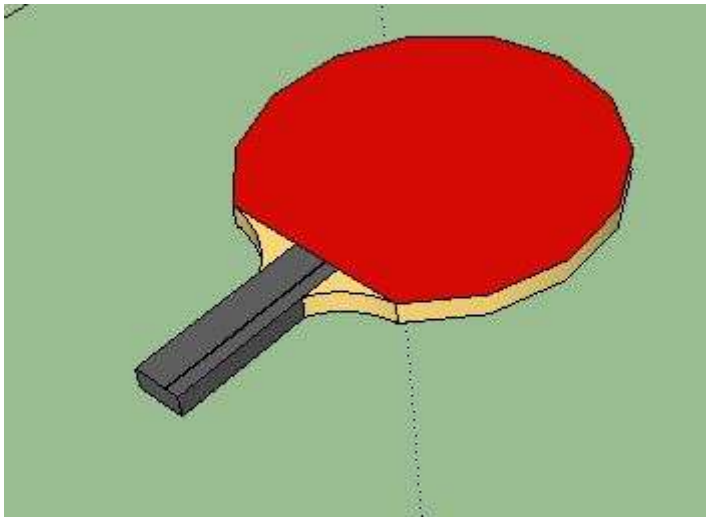
Perancangan komponen yang ada pada game pingpong meliputi karakter dan beberapa material yang menjadi unsur penting dalam permainan pingpong.

#### 3.6.1.1 Karakter

Berikut ini adalah beberapa karakter yang ada didalam permainan game pingpong

A. Player

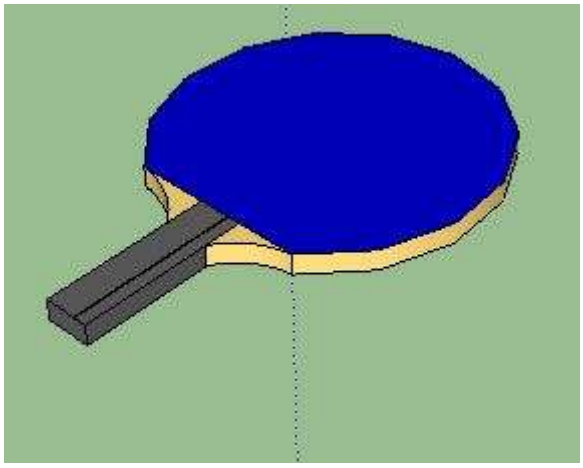
Berikut ini adalah perancangan karakter player dalam permainan pingpong pada gambar 3.23



***Gambar 3.23 Karakter player***

B. NPC

Berikut ini adalah perancangan karakter NPC dalam permainan pingpong pada gambar 3.24



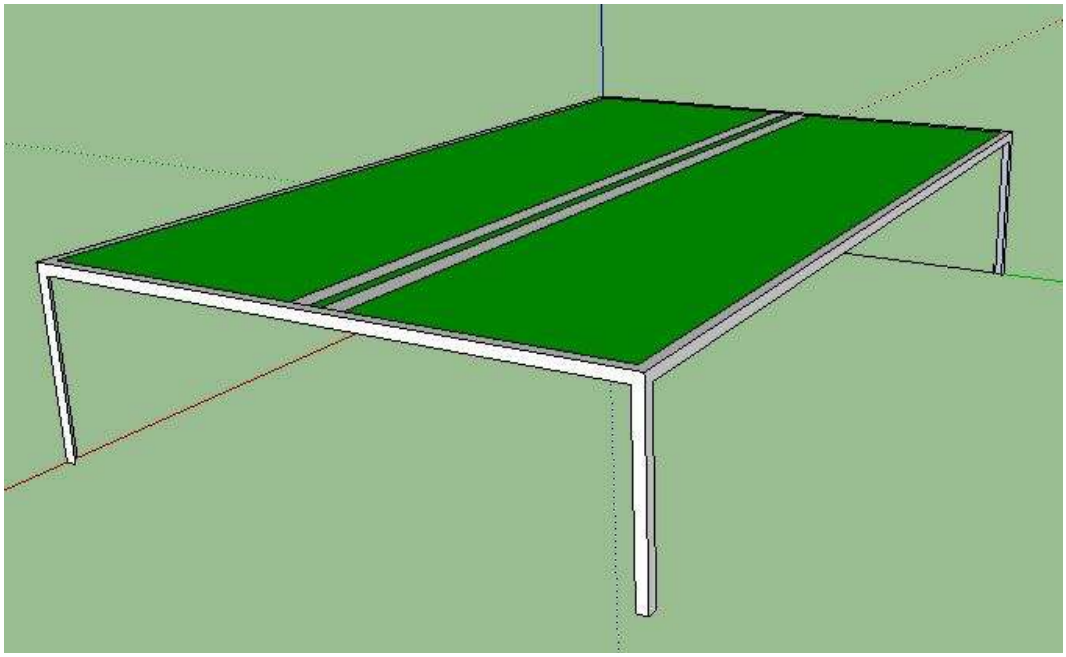
*Gambar 3.24 Karakter NPC*

### 3.6.1.2 Material

Berikut ini adalah beberapa material yang digunakan dalam permainan pingpong :

A. Meja permainan

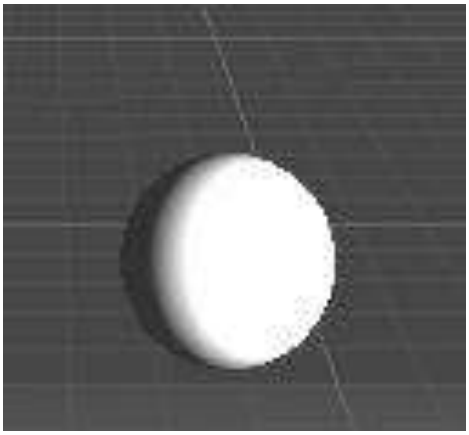
Meja yang digunakan dalam permainan dapat dilihat pada gambar 3.25



*Gambar 3.25 Meja Permainan*

B. Bola Pingpong

Bola yang digunakan dalam permainan dapat dilihat pada gambar 3.26



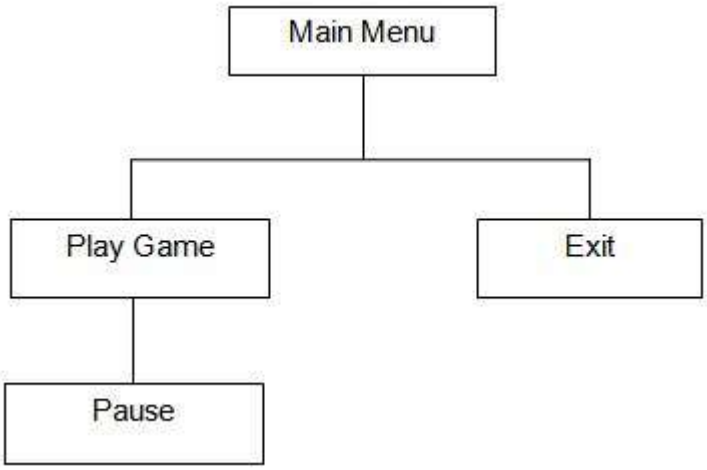
***Gambar 3.26 Bola Pingpong***

### 3.6.2 Perancangan Arsitektur

Perancangan arsitektur yang telah dibuat meliputi beberapa perancangan di antaranya struktur menu dan perancangan antarmuka.

#### 3.6.2.1 Perancangan Struktur Menu

Perancangan struktur menu dirancang untuk menyediakan fungsi-fungsi yang akan digunakan. Berikut ini adalah rancangan struktur menu pada gambar 3.27



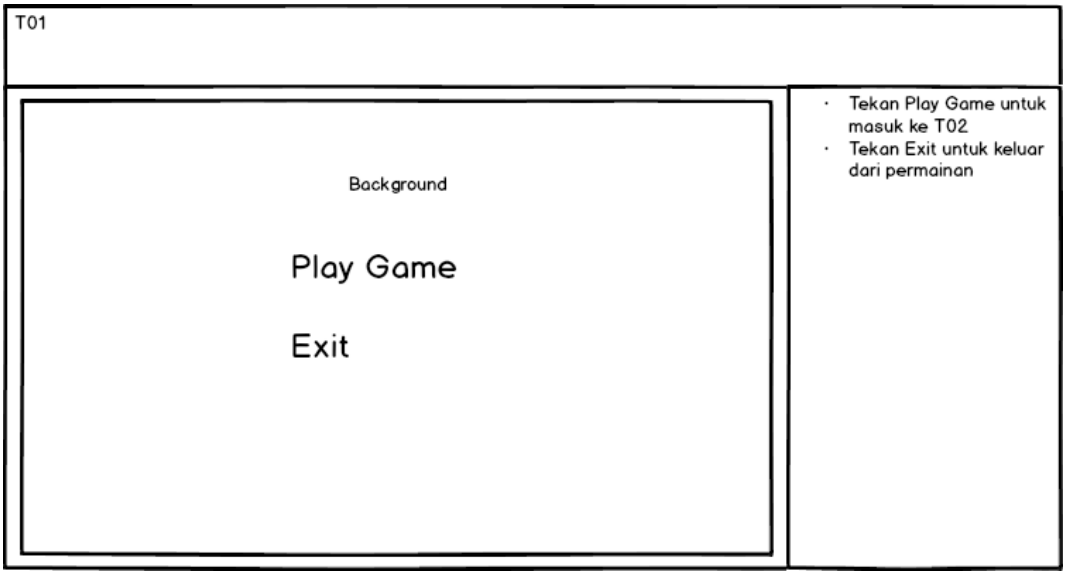

***Gambar 3.27 Struktur menu game pingpong***

### 3.6.2.2 Perancangan Antarmuka

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

1. Main Menu

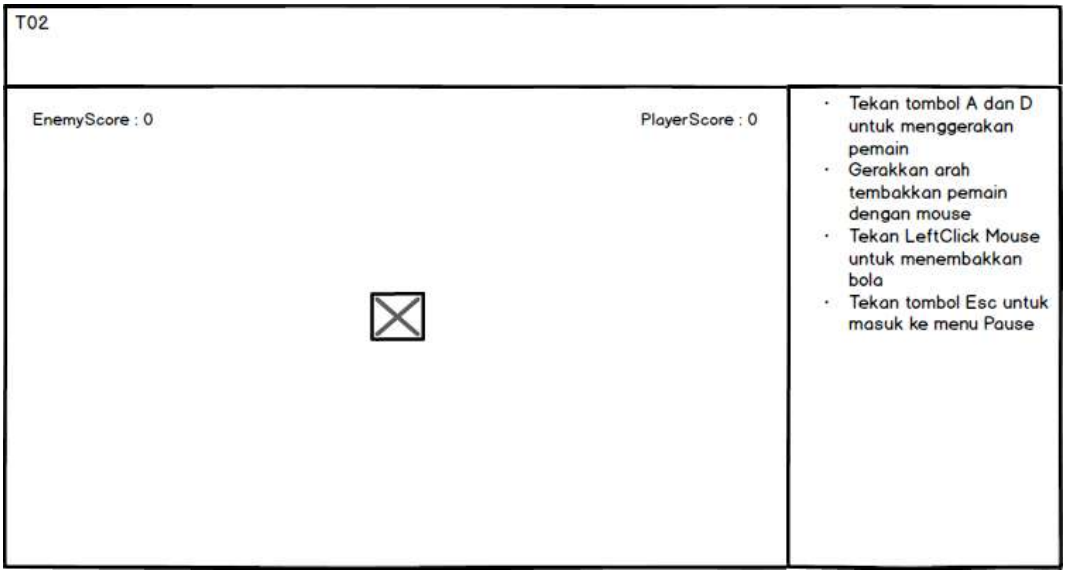
Berikut ini adalah perancangan antarmuka main menu dari pembangunan aplikasi game pingpong pada gambar 3.28




***Gambar 3.28 Perancangan antarmuka main menu***

2. Play

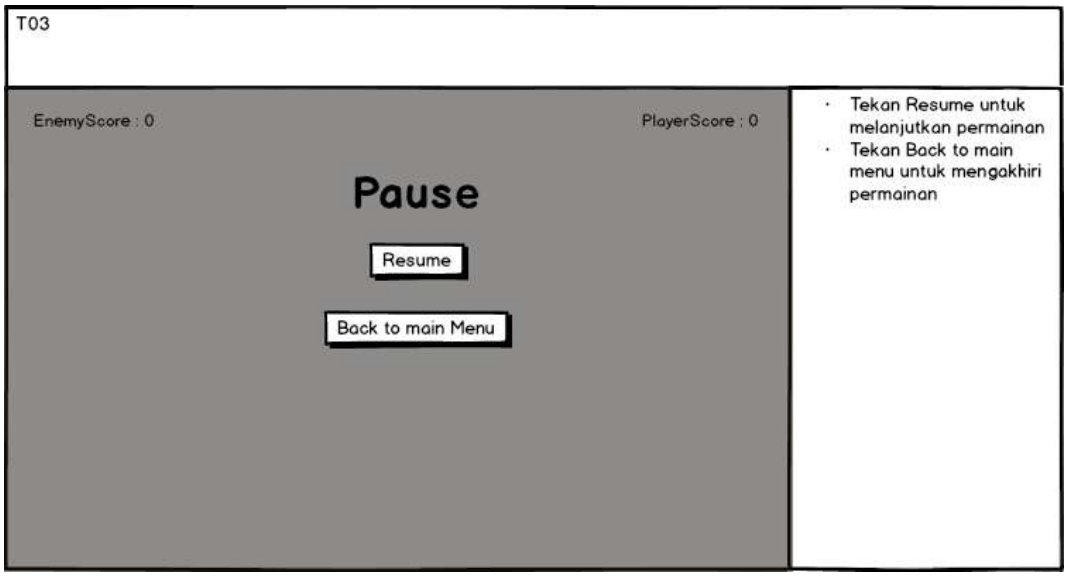
Berikut ini adalah perancangan antarmuka permainan dari pembangunan aplikasi game pingpong 3.29




***Gambar 3.29 Perancangan Antarmuka Permainan***

3. Pause Menu

Berikut ini adalah perancangan antarmuka pause menu dari pembangunan aplikasi game pingpong pada gambar 3.30



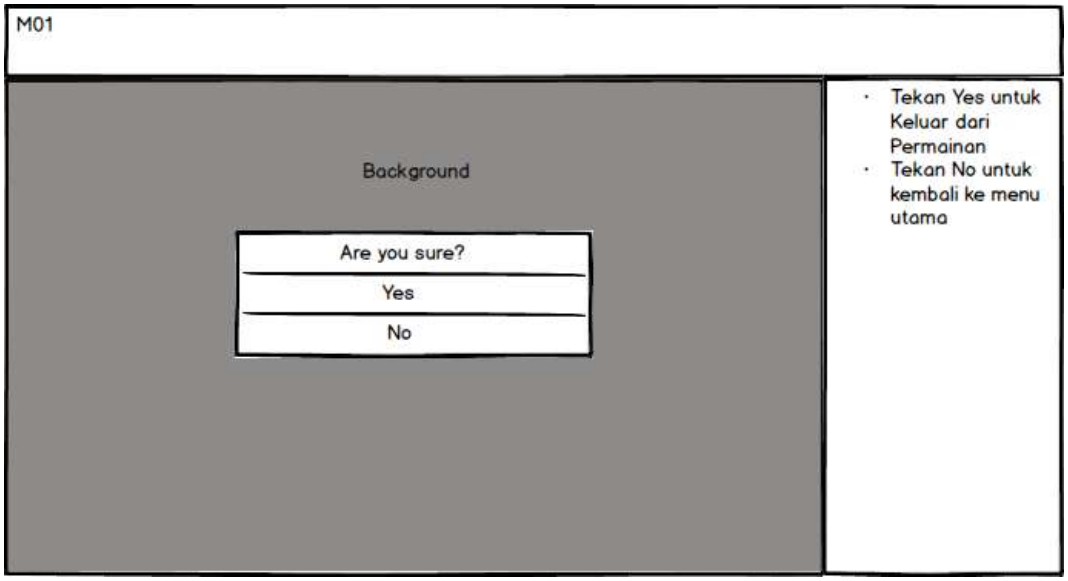

***Gambar 3.30 Perancangan Antarmuka Pause Menu***

#### 3.6.2.2.1 Perancangan Pesan

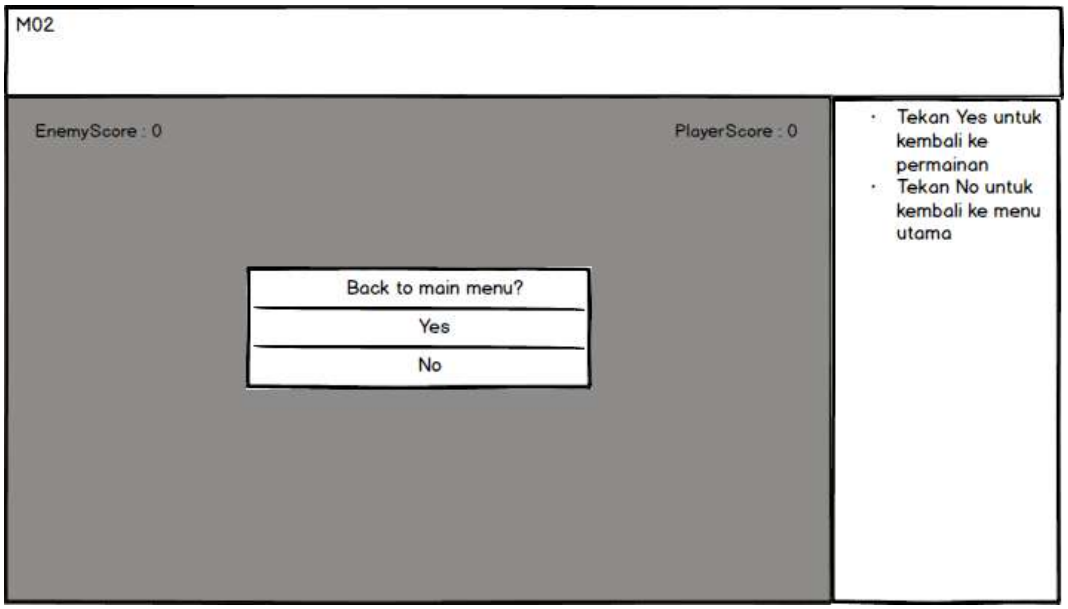
Berikut ini adalah form perancangan pesan yang ada pada aplikasi permainan pingpong pada tabel 3.16

***Tabel 3.16 Form Perancangan Pesan***

| No | No Pesan | Isi Pesan | No Form |
|---|---|---|---|
| 1 | M01 | *Are you sure to quit?* | T01 |
| 2 | M02 | Are you sure back to main menu? | T03 |



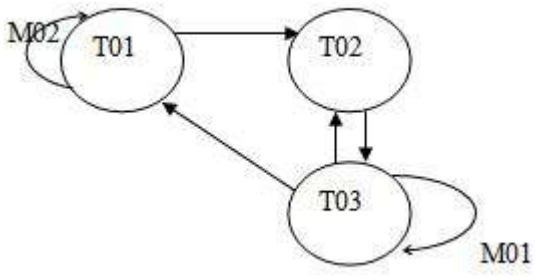

*Gambar 3.31 Perancangan Pesan M01*




*Gambar 3.32 Perancangan Pesan M02*

#### 3.6.2.2.2 Jaringan Semantik

Setelah melakukan perancangan antarmuka dan perancangan pesan maka dilakukanlah perancangan terhadap aliran dari menu-menu yang ada di program yang mana akan digambarkan dalam sebuah jaringan semantik. Berikut ini adalah jaringan semantik yang ada pada gambar 3.23



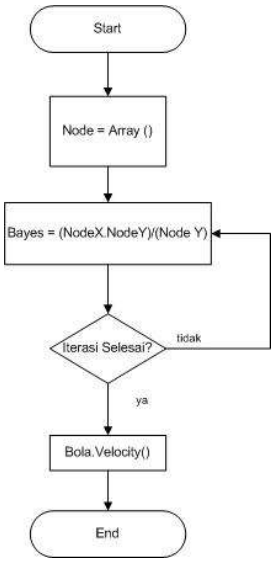

***Gambar 3.33 Jaringan Semantik Game Pingpong***

### 3.1.1 Perancangan Method

Perancangan *method* merupakan perncangan yang berfungsi untuk mendeskripsikan *method* - *method* yang berada di dalam aplikasi. *Method* dapat dipanggil dengan menyertakan variabel, baik hanya satu variabel, banyak variabel atau bahkan tidak ada sama sekali. Adapun *method* - *method* yang terdapat dalam game pingpong yang akan dibangun adalah sebagai berikut :

A. BNetwork.sideNode()

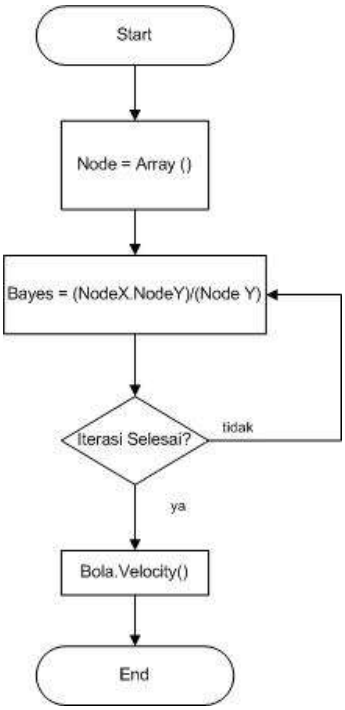
*Method* ini digunakan untuk inisialisasi sideNode pada struktur data game pingpong dan nilai yang nantinya akan diproses pada tahap Bayes Theorem. *Method* ini ditunjukan pada gambar 3.34 berikut ini




***Gambar 3.34 BNetwork.sideNode()***

B.BNetwork.Bayes()

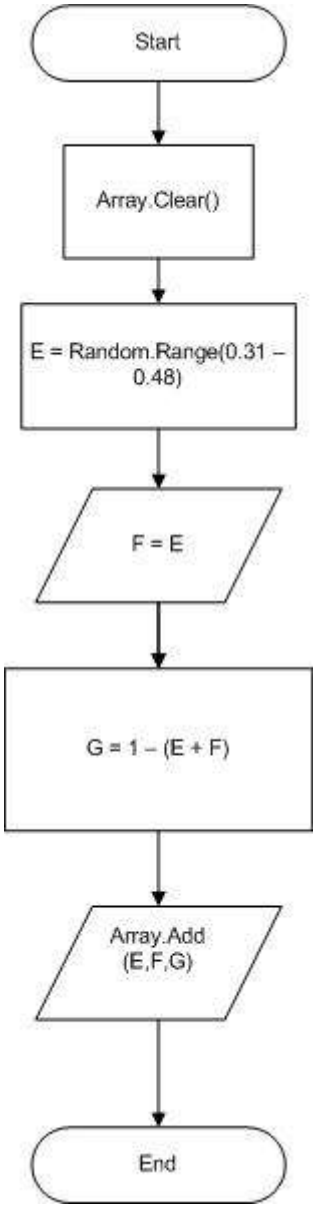
*Method* ini digunakan untuk proses perhitungan Bayes Theorem berdasarkan pada nilai dari node – node yang telah diinisialisasi. *Method* ini ditunjukan pada gambar 3.35 berikut ini




***Gambar 3.35 BNetwork.Bayes()***

C.BNetwork.midNode()

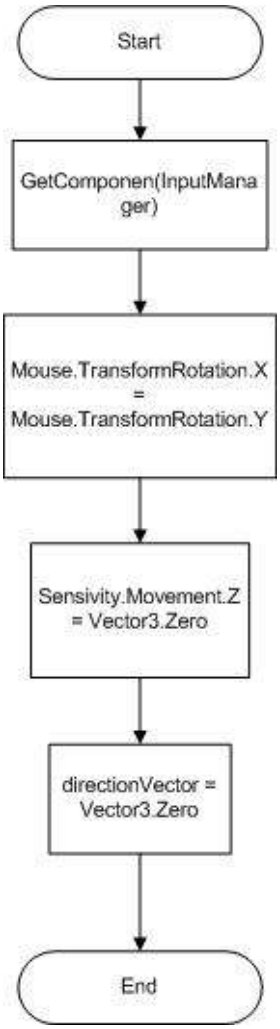
*Method* ini digunakan untuk inisialisasi midNode pada struktur data game pingpong dan nilai yang nantinya akan diproses pada tahap Bayes Theorem. *Method* ini ditunjukan pada gambar 3.36 berikut ini




***Gambar 3.36 BNetwork.midNode()***

D. CharacterMotor.CharacterMotorMovement()

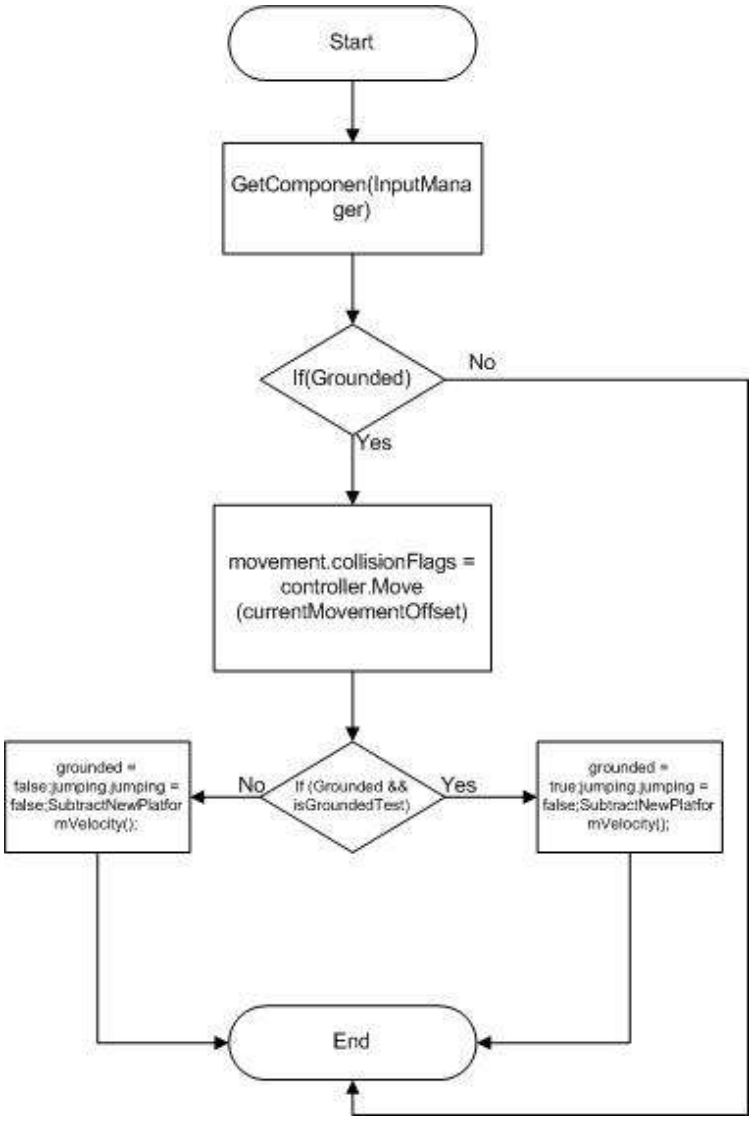
*Method* ini digunakan untuk pergerakan yang dilakukan oleh *player* terhadap karakter yang ada dalam permainan. Pergerakan tersebut meliputi rotasi terhadap karakter yang akan berpengaruh terhadap kamera. *Method* ini ditunjukan pada gambar 3.37 berikut ini




***Gambar 3.37 CharacterMotor.CharacterMotorMovement()***

E. CharacterMotor.ApplyInputVelocityChange()

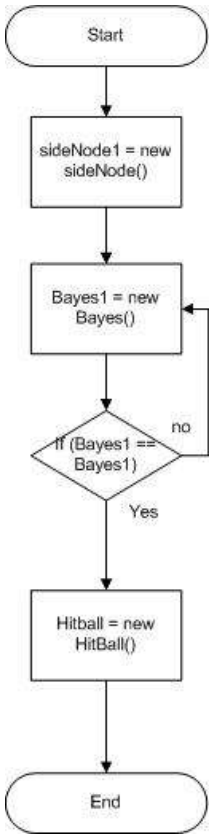
*Method* ini digunakan untuk menstabilkan lompatan terhadap gravitasi dimana pijakan karakter tersebut sewaktu – waktu bisa berubah. *Method* ini ditunjukan pada gambar 3.38 berikut ini




***Gambar 3.38 CharacterMotor.ApplyInputVelocityChange()***

F. AIController.Update()

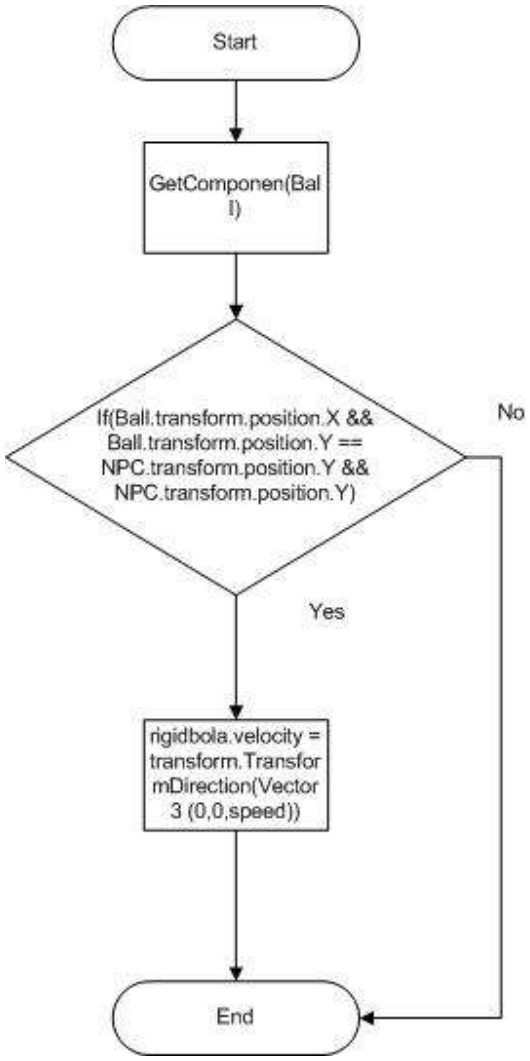
*Method* ini digunakan untuk inisialisasi sideNode pada struktur data game pingpong dan nilai yang nantinya akan diproses pada tahap Bayes Theorem. *Method* ini ditunjukan pada gambar 3.39 berikut ini




***Gambar 3.39 AIController.Update()***

G. HitBall.Update()

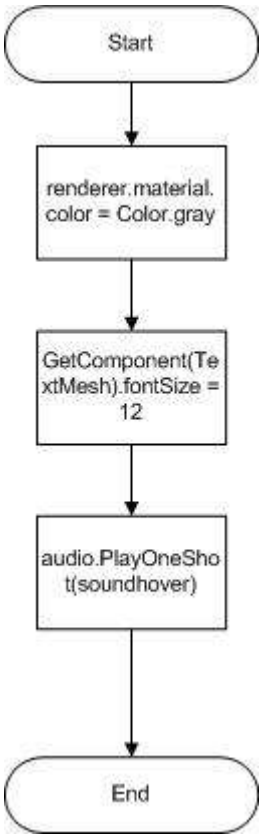
*Method* ini digunakan agar NPC mampu menembak sebuah bola berdasarkan *velocity* atau arah terhadap tembakannya. *Method* ini ditunjukan pada gambar 3.40 berikut ini




***Gambar 3.40 HitBall.Update()***

H. MainMenu.OnMouseEnter()

*Method* ini digunakan untuk kursor pada menu permainan menekan sebuah objek. *Method* ini ditunjukan pada gambar 3.41 berikut ini




***Gambar 3.41 MainMenu.OnMouseEnter()***