# IMPLEMENTASI GERAK PARABOLA PADA GAME PROJECT:TANK

Mananda / 00000005245 / mananda.hutagalung@gmail.com

Winadi Wiratama / 00000007257 / winadiw@gmail.com

**Abstrak**

**Kata kunci : .....**

## 1. Pendahuluan

Pemilihan judul penulis dilatarbelakangi oleh keinginan penulis untuk membuat game yang terinspirasi dari game yang bernama “Shellshock Live”. Shellshock Live adalah game di mana pemain, dengan mengendalikan sebuah tank, berusaha menghancurkan semua tank lawan yang ada. Cara menghancurkan tank lawan adalah dengan menembakan peluru, dimana saat ditembakkan, amunisi tersebut memiliki gerak parabola. Fitur menarik lainnya pada game ini terdapat pada amunisi tank yang dibuat dengan sangat kreatif, beragam, dan unik satu sama lain. Fitur inilah yang menjadi fokus dari pembuatan game. Penulis berharap, dengan mengambil inspirasi dari game tersebut, game yang di buat dapat dengan mudah memenuhi kriteria utama dari projek ini, yaitu adanya penggunaan rumus Gerak Parabola di dalam game yang akan di buat.

Berdasarkan uraian diatas, maka penulis dapat merumuskan masalah yaitu “Bagaimana cara implementasi gerak parabola ke dalam game Project:Tank?”

Penulis juga merumuskan beberapa batasan yang akan di terapkan dalam pembuatan game tersebut antara lain:

1. Pembuatan game ini tidak akan menggunakan game engine apapun, hanya menggunakan software Integrated Development Environment (IDE) dengan bahasa pemrograman Java.
2. Desain dan pembuatan dari amunisi beserta rumus - rumus yang akan digunakan menjadi prioritas dari pembuatan game ini. Pelengkap - pelengkap seperti aset tekstur dan suara akan ditambahkan setelah sistem amunisi penulis telah selesai
3. Salah satu mode dalam game ini dimainkan oleh 2 orang, namun mode ini tidak dapat dimainkan secara multipemain lewat koneksi LAN/Internet.

Dengan judul yang telah penulis putuskan bersama, tujuan penulis membuat projek tersebut adalah untuk menerapkan Gerak Parabola pada game yang akan dibuat yaitu Project:Tank.

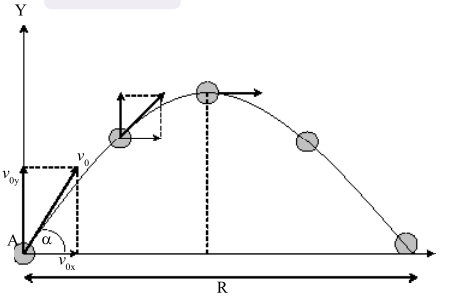
Penulis juga berharap bahwa hasil akhir yang akan di dapatkan adalah game yang dapat dijalankan dan dimainkan dengan baik, mekanika inti dan penyeimbangannya tepat, serta implementasi rumus Gerak Parabola yang menjadi kriteria utama dari projek ini berhasil di terapkan dan terlihat dalam permainan.

# 2. Landasan Teori

**2.1 Teori utama**

Dalam game ini, rumus yang akan digunakan berkaitan dengan gerak benda, terutama gerak parabola.

Menurut [3] jika suatu bola dilempar miring dengan sudut tertentu, maka gerak yang terjadi dinamakan gerak parabola atau gerak peluru. Gerak parabola dapat terjadi karena perpaduan gerak GLB (Gerak Lurus Beraturan) dan GLBB (Gerak Lurus Berubah Beraturan).



**Gambar 1** Gerak parabola dari sebuah sudut yang diberi kecepatan awal membentuk sudut tertentu.

Gravitasi, menjadi gaya ke bawah, menyebabkan proyektil untuk mempercepat ke bawah. Dalam gerak proyektil, gaya gravitasi tidak pernah bisa mengubah kecepatan horisontal dari suatu obyek karena komponen tegak lurus gerak yang independen satu sama lain, sehingga gaya vertikal tidak mempengaruhi gerakan horisontal [2, pp. 70].

Jika bola dilemparkan dengan kecepatan v0 dan sudut elevasi α maka kecepatannya dapat diproyeksikan ke arah mendatar (sumbu X) dan arah vertikal (sumbu Y). Persamaannya adalah sebagai berikut:

v0x = v0 cos α

v0y = v0 sin α

Pada arah sumbu X (horisontal) v0x tidak dipengaruhi oleh percepatan, sehingga terjadi gerak lurus beraturan. Maka berlaku hubungan sebagai berikut:

Vx = v0x

x = vxt

Pada arah sumbu Y (vertikal), v0y dipengaruhi oleh percepatan gravitasi yang arahnya ke bawah dan besarnya g = 9.8 m/s2. Maka pada arah ini terjadi gerak lurus berubah beraturan (GLBB) diperlambat. Berikut perumusannya:

Vy = v0y – gt

Y = v0yt – (½gt2)

**Titik tertinggi**

Pada gambar 1, titik tertinggi terjadi di titik B. Dari hasil penurunan rumus, maka untuk mencari tinggi maksimum yang dapat dicapai pada gerak parabola memenuhi persamaan berikut:

ym = (v02 (sin α)2) / 2g

Dengan:

ym = tinggi maksimum (m)

v0 = kecepatan awal (m/s)

α = sudut elevasi

g = percepatan gravitasi

**Efek Angin**

Rumus-rumus diatas tidak memperhitungkan koefisien angin, namun dalam kenyataanya angin mempengaruhi gerak proyektil, terutama jarak yang akan ditempuh dari benda (jika angin berlawan arahnya dengan benda). Untuk menghitung koefisien gaya eksternal, maka harus diperhitungkan Fx dan Fy. Maka digunakan rumus sebagai berikut:

Fx = F \* sin (β)

Fy = F \*cos (β)

Ax = Fx / m;

Ay = (Fy / m) + g;

Dengan:

F = gaya (N)

Fx = gaya pada sumbu x (N)

Fy = gaya pada sumbu y (N)

β = sudut gaya

Ax = percepatan benda di sumbu x

Ay = percepatan benda di sumbu y

Dalam game yang akan di buat, penulis menggunakan rumus parabola pada peluru yang akan ditembakkan oleh tank. Peluru akan ditembak mulai dari tank (titik A pada Gambar 1) yang memiliki arah serta kecepatan tertentu (v0x, v0y, α, v0). Akan terdapat beberapa jenis peluru dengan karakteristik masing-masing. Kekuatan tembakan, sudut laras tank dan jenis dari peluru akan mempengaruhi jarak serta ketinggian dari proyektil peluru. Semakin kuat v0 (kecepatan awal) maka akan semakin jauh peluru yang ditembakkan. Game tersebut juga menerapkan koefisien eksternal (angin) dalam perhitungan rumus parabola.

**2.2 Teori pendukung**

Pada bagian ini, akan dijelaskan berbagai karakteristik dari suatu game. Karakteristik yang akan dijelaskan dirasa perlu dipertimbangkan agar game yang penulis buat baik dan menarik untuk dimainkan.

**4 Elemen Dasar**

Menurut [1, pp. 41 - 45], ada 4 elemen dasar dari sebuah game:

1. **Mechanics**: menjelaskan prosedur dan aturan yang ada dalam suatu game, tujuan dari suatu game, apa saja yang bisa dilakukan dan yang tidak bisa dilakukan oleh pemain dalam mencapai tujuan tersebut.
2. **Story**: alur kejadian atau cerita yang menjadi latar belakang suatu game, cerita tersebut bisa bersifat linier atau bercabang. Story membantu pemain mengerti rules serta goal dari game tersebut.
3. **Aesthetics**: bagaimana penampilan dari suatu game, penting karena berhubungan langsung dengan pengalaman pemain selama memainkan game. Aesthetics merupakan barisan pertama dari sebuah game (first impression). Aesthetics juga berkaitan dengan reaksi emotional pemain dengan game.
4. **Technology**: menjelaskan tentang medium yang digunakan dalam membuat game. Setiap pemilihan Technology memiliki kelebihan dan kekurangan masing-masing yang patut dipertimbangkan.

**Tingkat Kesulitan**

Pada [1, pp. 177 - 179], diperlihatkan pentingnya memperhatikan tingkat kesulitan dari sebuah game. Hal ini disebabkan karena tingkat kesulitan berpengaruh kepada keinginan pemain untuk terus bermain atau tidak. Ketika suatu game terlalu sulit untuk dimainkan, pemain menjadi frustrasi dan mungkin meninggalkan game itu, tidak memainkannya lagi. Sebaliknya, ketika suatu game terlalu mudah untuk dimainkan, pemain menjadi tidak tertarik untuk terus memainkan game tersebut. [1, pp. 177 - 179] menjelaskan juga bahwa untuk melakukan penyesuaian tingkat kesulitan adalah sesuatu yang sulit. Ini dikarenakan setiap pemain mempunyai kemampuan yang berbeda - beda dan tidak ada tolak ukur yang sama. Apa yang dianggap mudah oleh seorang pemain, belum tentu mudah untuk pemain lainnya. Sebaliknya, apa yang dianggap sulit oleh seorang pemain, belum tentu sulit untuk pemain lainnya. Oleh karena itu, [1, pp. 177 - 179] menjelaskan beberapa teknik yang umumnya digunakan untuk melakukan penyeimbangan tingkat kesulitan dalam game:

1. Menerapkan peningkatan kesulitan untuk setiap level. Peningkatan kesulitan suatu level bisa berbentuk progress (semakin lama semakin sulit).
2. Menerapkan tipe kesulitan yang berbeda (mudah, sedang, sulit), yang kemudian pemain bisa pilih sebelum bermain. Pilihan ini memungkinkan pemain menyesuaikan style/gaya bermainnya serta keahlian pemain dalam game tersebut.
3. Menerapkan sistem penilaian (A, B, C, D, dst.) untuk menilai sebaik apa seorang pemain menyelesaikan suatu level dalam game. Misalnya, untuk mendapatkan nilai A dan B seorang pemain harus melakukan suatu misi yang sulit selama ia sedang menyelesaikan suatu level, untuk mendapatkan nilai C ke bawah seorang pemain cukup melakukan misi yang mudah atau hanya sekedar menyelesaikan level tersebut. Sistem penilaian pada game membuat pemain ingin berusaha mendapatkan yang terbaik, sehingga meningkatkan tingkat playability dari sebuah game.

**Pilihan yang Berpengaruh**

Menurut [1, pp. 179] “A good game gives the pemain meaningful choices.”. Pada [1, pp. 179 - 181] dijelaskan bagaimana memberikan pilihan kepada pemain yang mampu mempengaruhi bagaimana pemain memainkan suatu game. Setiap pilihan yang diberikan kepada pemain harus seimbang satu sama lain, punya kelebihan dan kekurangannya masing - masing. Tidak boleh ada yang terlalu mengungguli pilihan yang lainnya sehingga menjadi pilihan yang pasti. Jika itu terjadi, game itu akan menjadi tidak menarik lagi. Pemain tidak terdorong untuk melakukan eksplorasi dari pilihan - pilihan yang ada. Selain itu perlu diperhatikan juga jumlah pilihan yang diberikan. Terlalu banyak akan membuat pemain kebingungan, terlalu sedikit akan membuat pemain frustrasi dan bosan. Perlu dicari jumlah pilihan yang tepat sehingga pemain dapat memenuhi keinginannya tentang bagaimana ia akan memainkan game itu.

**Reward**

[1, pp. 188] menyatakan bahwa kenapa pemain bisa menghabiskan banyak waktu untuk bermain game? Salah satu alasannya adalah untuk mendapat reward yang diberikan oleh game. Reward yang diberikan pun bisa dari hal yang sangat simpel seperti lewat kata-kata “Well Done!” hingga memberikan bonus berapa item in-game. Terdapat beberapa jenis reward yang game bisa berikan kepada pemain untuk menambah motivasi dalam bermain [, pp. 189]:

1. **Praise (Pujian):** reward yang paling simpel, seperti “Good Work” atau lewat sound-effect ketika berhasil memenangkan stage dari game.
2. **Points:** banyak game menerapkan point untuk menjadi tolak ukur dari kesuksesan pemain dalam bermain game tersebut. Adanya point membuat pemain ingin mencoba yang terbaik, agar mendapatkan score tertinggi (High Score).
3. **Power (kekuatan)**: menjadi lebih kuat dari sebelumnya di dalam game tentunya membuat pemain lebih termotivasi dan bisa menjadi reward bagi pemain juga. Dengan power yang semakin kuat, maka pemain bisa mengatasi problem yang ada di dalam game dengan lebih mudah.
4. **Completion:** ketika pemain sudah menyelesaikan seluruh goal yang ada di dalam game, maka ada rasa kepuasan tersendiri bagi pemain. Ini adalah reward terbesar dari sebuah game, ketika sudah sampai titik ini maka pemain tidak dapat melanjutkan game.

Game yang baik tentu saja memberikan reward bagi pemain ketika menyelesaikan suatu task atau goal tertentu. Motivasi untuk mendapatkan reward adalah alasan terutama player ingin memainkan game tersebut. Oleh karena inilah, game yang baik harus memperhatikan reward bagi pemain setelah menyelesaikan sebuah task.

**Kompleksitas**

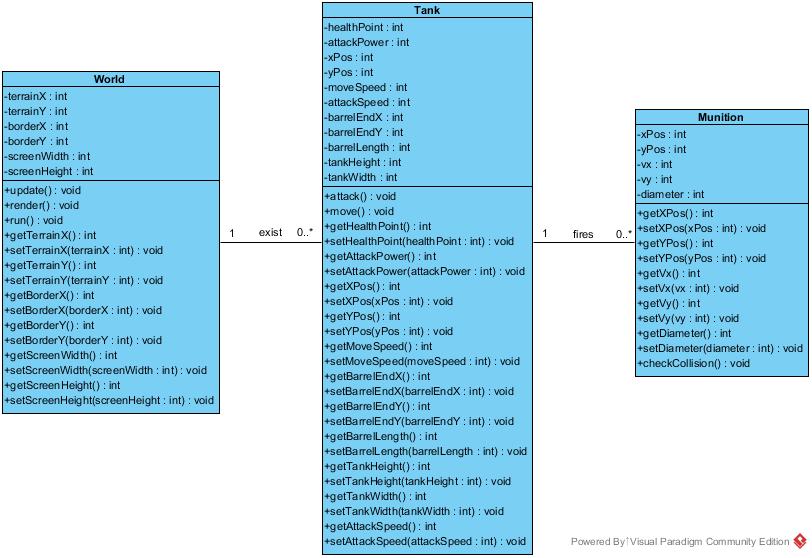
Pada [1, pp. 195 - 197] dijelaskan bagaimana mengevaluasi kompleksitas dari sebuah game. Ada 2 jenis kompleksitas yang umumnya muncul dalam sebuah game:

1. **Innate complexity**: aturan - aturan yang ada dalam game rumit, ciri khasnya adalah setiap aturan tersebut terdapat beberapa pengecualian, berusaha untuk mengatasi berbagai situasi yang mungkin muncul, sehingga aturan tersebut akan diubah sesuai dengan situasi yang ada.
2. **Emergent complexity**: aturan - aturan yang ada dalam game sederhana, namun mampu memunculkan situasi yang rumit

Perlu disesuaikan antara jenis kompleksitas yang akan diterapkan dengan konteks dari game itu sendiri. Umumnya, game yang berhasil dan dimainkan oleh orang banyak adalah game yang di dalamnya diterapkan konsep emergent complexity. Hal ini disebabkan konsep tersebut mampu menyediakan kejutan - kejutan kepada pemain saat memainkannya, sesuai dengan naturnya yang bisa memunculkan situasi baru yang rumit, sehingga player tertantang untuk memecahkan situasi yang baru itu. Pada akhirnya, ini menjadikan game tersebut menarik untuk dimainkan dan tidak akan ditinggalkan dalam waktu yang cukup lama.

# 3. Perancangan Program

**Class Diagram**



**Gambar 2** Class Diagram

Class diagram di atas terdiri atas 3 class dengan keterangan sebagai berikut:

1. World: Class ini akan digunakan sebagai class yang menggambar frame utama dari game

2. Tank: Class ini akan digunakan sebagai class untuk tank. Class Tank akan menyimpan attribute yang bersesuaian dengan tank, serta fungsi-fungsinya. Karena terdapat 2 pemain dalam game, maka akan dibuat 2 objek tank dari class Tank.

3. Munition: Class ini akan digunakan sebagai class yang menyimpan attribute dan fungsi yang diperlukan untuk 1 objek peluru. Setiap pembuatan objek class ini akan membuat 1 peluru.

**Pseudocode**

Dalam game, penulis akan menerapkan sistem attack speed untuk mengendalikan tingkat kecepatan tembakan dari tiap tank. Berikut adalah pseudocode yang penulis akan gunakan sebagai panduan dalam menerapkan sistem tersebut:

//Attack speed

delaycounter = 0;

delay = 50;

player\_fire()

if player fired then

if (delaycounter <= 0) then

fireweapon()

delaycounter = delay

player\_update(decrement)

if (delaycounter > 0) then

delaycounter -= decrement;

Pseudocode di atas bekerja sebagai berikut:

1. Inisialisasi 2 variabel, delaycounter untuk menghitung sisa waktu delay, dan delay untuk menentukan seberapa lama waktu delay yang diterapkan
2. Di dalam fungsi serangan, ketika akan menembak, diperiksa apakah delaycounter sudah habis
3. Jika ya, dilanjutkan dengan proses menembak, kemudian masukkan nilai delay kepada delaycounter
4. Jika tidak, proses menembak tidak akan dijalankan
5. Di dalam fungsi update, ditambahkan parameter untuk menyimpan nilai pengurangan delaycounter. Kemudian nilai delaycounter akan dikurangi dengan nilai pengurangan selama nilai delaycounter masih lebih besar dari 0

**//**Cek apakah peluru mengenai tank

void detectCollision(Player target)

{

if(bulletY+r >= (target.getY() - target.getHeight()) &&

bulletY +r <= (target.getY()))

{

if(bulletX >= (target.getX() - target.getWidth()) &&

bulletX <= (target.getX() + target.getWidth()))

{

target.hit(); target.hit();

}

}

}

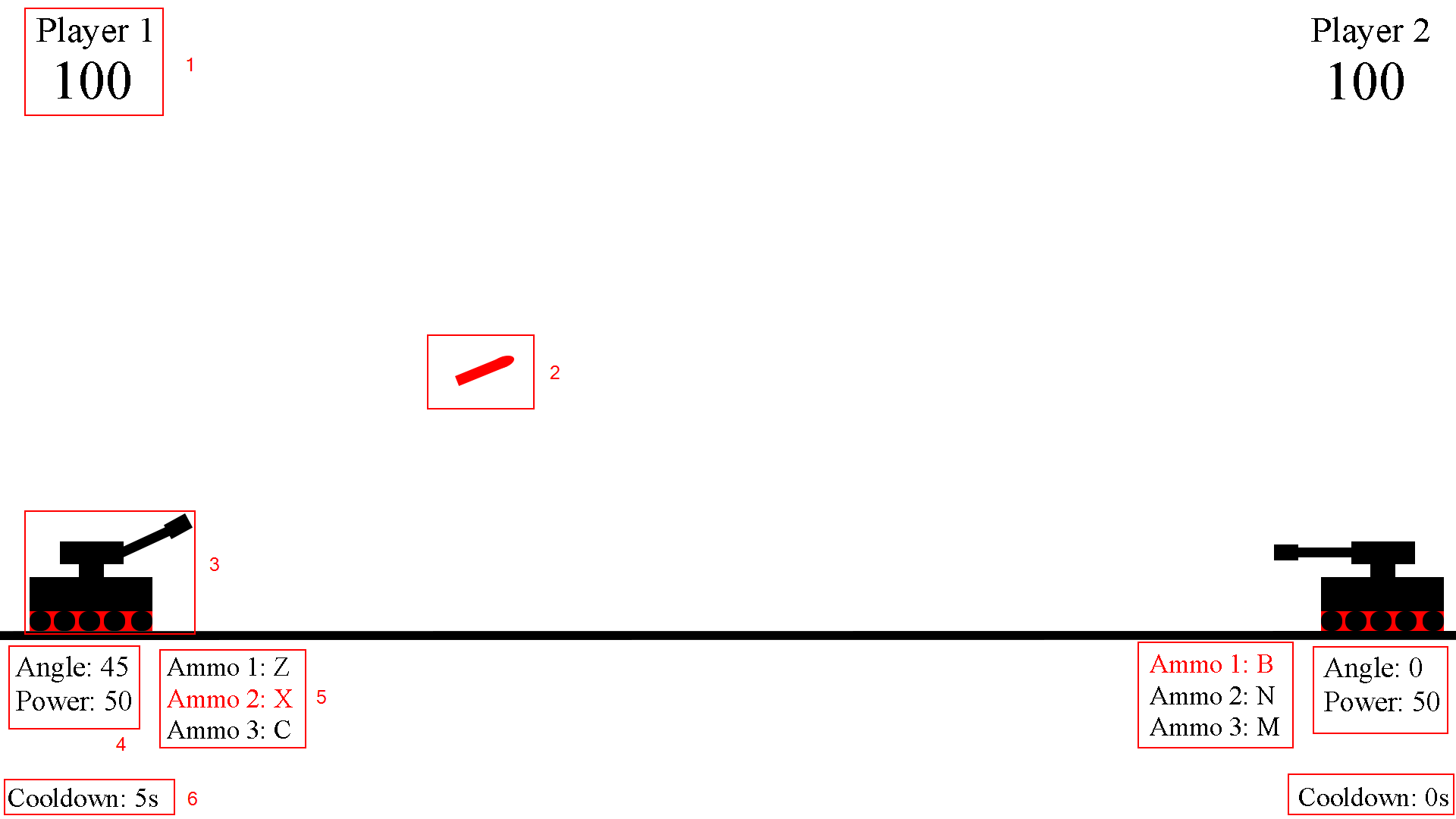
Pseudocode diatas digunakan untuk cek apakah peluru mengenai tank. Berikut adalah penjelasan pseudocode diatas:

1. IF pertama digunakan untuk cek apakah posisi Y dari bullet berada di area Y dari tank. Tinggi dari tank juga harus diperhitungkan, sehingga dibutuhkan variable yang di return oleh fungsi target.getHeight().

2. IF kedua digunakan untuk cek apakah posisi X dari bullet berada di area X dari tank tersebut. Lebar dari tank juga harus diperhitungkan, sehingga dibutuhkan variable lebar tank yang di return oleh target.getWidth().

3. Jika kedua IF memenuhi kondisi, maka fungsi target.hit() akan dipanggil.

**Rancangan UI**



**Gambar 3** Rancangan UI

Gambar 3 menunjukkan rancangan tampilan utama dari game Project:Tank. Berikut adalah penjelasan tiap bagian dari gambar:

1. Pada Bagian 1 akan tertulis siapa nama pemain, serta HP (health point) dari tank milik pemain tersebut. Jika HP berkurang karena terkena peluru, maka HP akan otomatis di update dan pemain dapat melihat HP nya lewat tampilan tersebut.

2. Bagian 2 merupakan peluru yang ditembakkan oleh Player 1. Kecepatan dan arah dari peluru ditentukan oleh player pada, ditampilkan pada bagian 4.

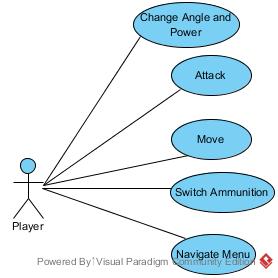
3. Bagian 3 merupakan tank yang digerakkan oleh Player 1

4. Bagian 4 menunjukkan sudut (Angle) serta kekuatan dari peluru (Power) yang akan ditembakkan.

5. Bagian 5 menampilkan jenis-jenis ammo atau peluru yang dapat digunakan untuk ditembakkan oleh pemain, serta hotkey yang harus ditekan untuk mengaktifkan ammo tersebut. Pada tulisan yang di warnai merah (pada Gambar 3 adalah Ammo 2: X) berarti Ammo 2 sedang aktif.

6. Bagian 6 menunjukkan cooldown dari Ammo. Setiap ammo memiliki cooldown yang berbeda-beda. Waktu cooldown akan disesuaikan dengan kelebihan ataupun kelemahan dari tipe ammo.

**Use Case Diagram**



**Gambar 5** Use Case Diagram

Gambar 5 menampilkan use case diagram dari game yang akan dibuat. Primary actor dari kelima use case adalah pemain (Player). Berikut adalah penjelasan setiap use case:

1. Change Angle and Power: sesuai namanya, use case tersebut bertujuan untuk menambah atau mengurangi power senjata maupun sudut dari laras tank.

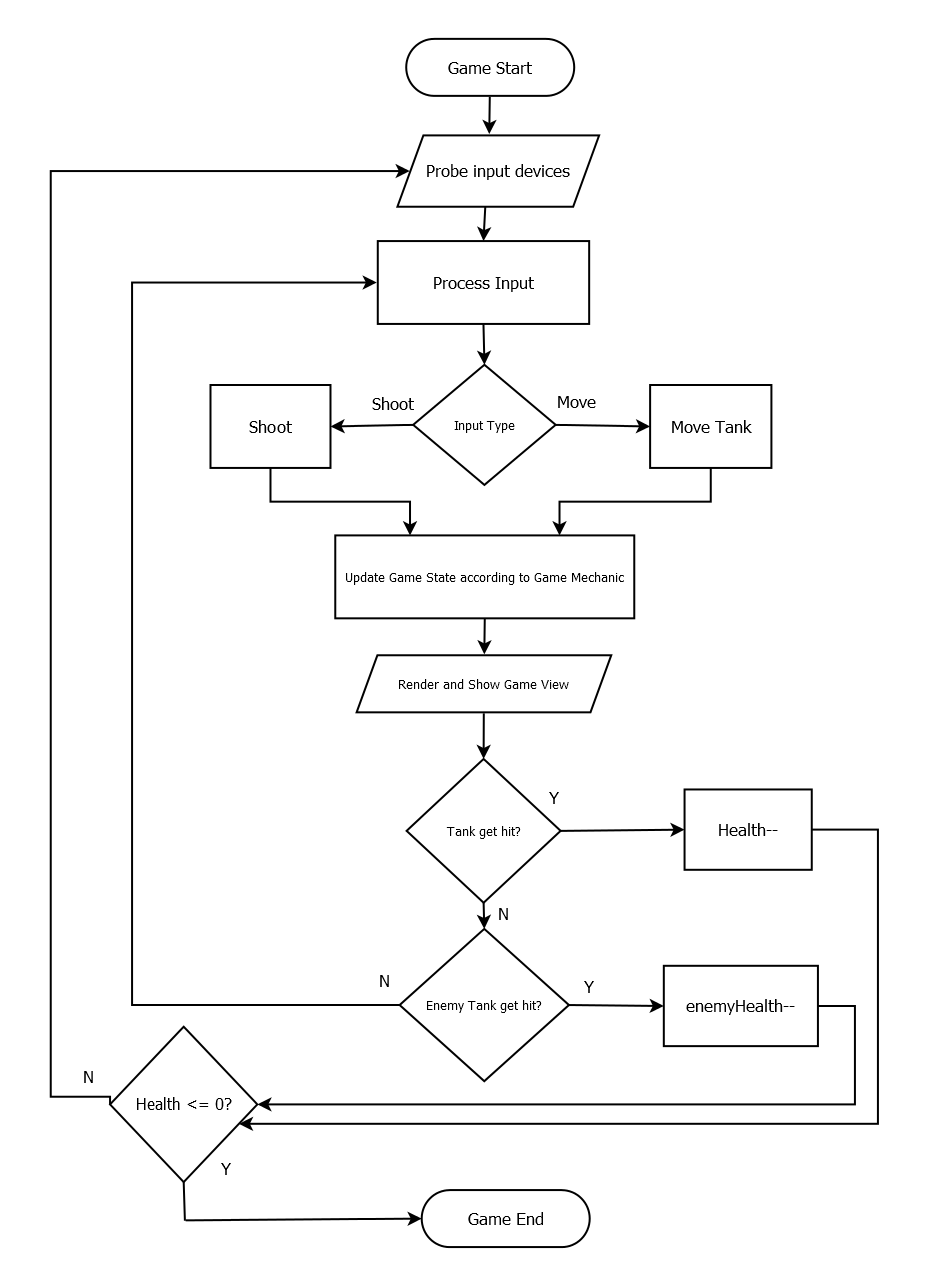
2. Attack: use case tersebut memiliki goal untuk menembakkan peluru atau ammo, sesuai dengan sudut dan power nya.

3. Move: use case tersebut berfungsi untuk mengerakkan tank baik ke kanan atau ke kiri

4. Switch Ammunition: use case tersebut bertujuan untuk mengubah ammunisi yang sedang digunakan.

5. Navigate menu: use case tersebut bertujuan untuk navigasi menu, baik itu menu utama maupun menu pause.

**Flowchart**



**Gambar 5** Flowchart

Flowchart di atas menunjukkan alur utama dari game yang akan dibuat. Berikut adalah penjelasan setiap proses pada flowchart:

1. Probe input devices: flowchart Input/Output. memeriksa input devices, tombol keyboard mana yang ditekan

2. Process input: memproses input sesuai dengan tombol yang ditekan

3. Decision: Input type. Jika setelah di proses menghasilkan key Shoot, maka akan masuk ke Shoot. Jika Move, maka Tank akan bergerak.

4. Update State: State atau kondisi dari game akan diupdate sesuai dengan input yang sudah di proses.

5. Render: gambar akan dirender dari hasil input

6. Decision: pada tahap ini posisi dari peluru dan tank musuh akan di cek. Jika tank pemain terkena peluru, maka nyawa pemain akan berkurang. Jika peluru pemain mengenai tank musuh, maka HP tank musuh akan berkurang.

7. Decision: Health < 0. Jika health (baik pemain maupun musuh) lebih kecil dari 0, maka game akan berakhir. Jika tidak, system akan kembali ke probe input devices untuk memeriksa kembali input yang ada.

# 4. Analisis dan Hasil Percobaan

# 5. Kesimpulan dan Saran

**Daftar Pustaka**

[1] J. Schell, *The Art of Game Design*. Amsterdam: Elsevier/Morgan Kaufmann, 2008.

[2] J. Walker, R. Resnick and D. Halliday, *Fundamentals of Physics*. Hoboken, NJ: Wiley, 2007.

[3] H. Sri and A. Damari, *Fisika untuk SMA dan MA kelas XI*. Jakarta, Indonesia: Pusat Perbukuan Departemen Pendidikan Nasional, 2009.