

BAB II

LANDASAN TEORI

2.1. Penelitian Terkait

Penelitian pertama yang berjudul “Pembuatan Aplikasi “*Egrang Run*” berbasis Android” (Sholikhin dkk, 2016), perancangan dan pembangunan *game* menggunakan metode MDLC (*Multimedia Development Life Cycle*). Penelitian ini berfokus pada konten yaitu konten edukasi untuk memperkenalkan kembali permainan tradisional egrang itu sendiri dengan berupa *game* yang bergenre-*endless run* yang dimana *game* tidak ada akhirnya kecuali *player* mati atau bar *life point* habis, penelitian ini juga berisi edukasi bagaimana egrang dimulai, cara bermain egrang dan info-info menarik mengenai egrang. Kelebihan dari penelitian ini konten yang diberikan cukup menarik terutama dari sisi edukasi yang bermaksud untuk memperkenalkan kembali permainan tradisional dalam bentuk digital, *game* ini dapat meningkatkan motorik bagi pengguna. Kekurangan dari penelitian ini *game* tidak bersifat *multiplatform* artinya hanya bisa dimainkan di platform Android, *game* ini bergenre-*endless run* yang membuat *game* kurang menarik dan terkesan jenuh, tidak terdapat nya NPC yang bisa membuat *game* lebih menarik.

Penelitian kedua yang berjudul “Penerapan Algoritma *Collision Detection* dan *Boids* Pada *Game Dokkaebi Shooter*” (Musfiroh dkk, 2014), perancangan dan pengembangan *game* menggunakan metode *Prototype*. Penelitian ini berfokus pada penerapan algoritma *Collision Detection* yang berfungsi sebagai pendeteksi

tubrukan antara 2 buah objek antara panah yang di tembakkan *player* dengan musuh yang akan menimbulkan sebuah rekasi yaitu musuh akan mati dan algoritma lain yang di terapkan adalah algoritma *Boids* yang berfungsi sebagai pergerakan musuh yang menyerupai pergerakan burung atau ikan. Kelebihan dari penelitian ini adalah berhasil mengimplementasikan dua algoritma sekaligus yaitu algoritma *Collision Detection* dan *Boids*. Kekurangan dari penelitian ini *game* hanya bisa dimainkan di *layer* dengan ukuran 600px ke bawah dan minimal versi android KitKat.

Penelitian ketiga yang berjudul “Implementasi Algoritma *Collision Detection* Pada *Game Simulator Driving Car*” (Fasha dkk, 2018), perancangan dan pengembangan game menggunakan metode GDLC (*Game Development Life Cycle*). Penelitian ini berfokus pada implementasi algoritma *Collision Detection* yang diterapkan dalam sebuah mobil dan juga dalam *obstacle* atau rintangan, cara kerja dari algoritma ini yakni ketika mobil menabrak *obstacle* maka mobil akan di kembalikan ke titik awal atau titik *start* karena tujuan dari *game* ini menjalankan sebuah mobil dari titik awal sampai titik akhir tanpa mengenai *obstacle*. Kelebihan dari penelitian ini berhasil mengimplementasikan algoritma *Collision Detection* kedalam dua buah objek yakni mobil dan *obstacle*, *game* ini juga berhasil memenuhi standar yang mengacu pada ATSC (*Advance Television Systems Committee*), karena *game* ini mampu mencapai FPS (*frame rate per second*) sebesar 60FPS.

Penelitian keempat yang berjudul “Penerapan *Game Artificial Inteligence Patrol Collison, Dynamic Movement* dan *Turret* Pada *Game* “USAGI

WATADOS” Berbasis 2D *Platformer*” (Putri dkk, 2017), perancangan dan pengembangan game menggunakan metode MDLC (*Multimedia Development Life Cycle*). Penelitian ini berfokus pada penerapan AI (*Artificial Inteligence*) dengan berbagai tipe seperti *Patrol Collison* yang berfungsi sebagai pergerakan karakter dan ketika karakter bertubrukan dengan objek maka akan terjadi sesuatu yang sudah ditentukan pada karakter seperti pengurangan *life point*, *Dynamic Movement* yang berfungsi sebagai pergerakan untuk NPC secara dinamis dan *Turret* yang berfungsi sebagai perlawanan yang dilakukan oleh karakter NPC berupa tembakan kepada karakter utama. Kelebihan dari penelitian ini adalah berhasil diterapkannya beberapa algoritma untuk karakter NPC sehingga membuat *game* ini menarik untuk dimainkan karena berbeda dari yang lain, *game* ini bisa dimainkan tanpa harus melalui proses instalasi dan juga bersifat *multiplatform* yaitu bisa berjalan di sistem operasi Windows, Mac dan Linux. Kekurangan dari penelitian ini *game* tidak dapat dimainkan secara *multiplayer*, pada pengaturan suara masih terjadi kesalahan seperti ketika sudah dimatikan suaranya tetapi ketika masuk ke menu lain musik kembali menyala padahal di pengaturan sudah dimatikan.

Penelitian kelima yang berjudul “Aplikasi *Game Win Start LA* Berbasis Android” (Adib dkk, 2016), perancangan dan pengembangan game menggunakan metode MDLC (*Multimedia Development Life Cycle*). Penelitian ini yakni berupa merancang dan mengembangkan sebuah *game Win Start LA* yakni sebuah permainan labirin, dimana karakter utama yang bernama *Win Start LA* yang harus mengitari titik-titik yang tersedia yang merupakan makanannya yang berada

disebuah labirin tanpa harus diketahui oleh musuhnya, *game* ini menggunakan Algoritma *Greedy* yang diterapkan pada NPC yang berfungsi untuk mencari rute terpendek ke posisi karakter utama. Kelebihan dari penelitian ini berhasilnya mengimplementasikan algoritma *Greedy* yang diterapkan pada NPC, *game* ini berfungsi untuk meningkatkan minat, kecerdasan, kreatifitas maupun motorik bagi yang memainkannya sekaligus mengetahui letak geografis Kota Lamongan. Kekurangan dari penelitian ini adalah *interface* atau tampilan dari *game* ini terlalu kaku atau tidak menarik apalagi *game* ini dikhususkan untuk kalangan anak-anak dan tidak terteranya bagaimana cara kerja dari Algoritma *Greedy* tersebut seperti perhitungan dari algoritma yang diterapkan.

Penelitian keenam yang berjudul “Implementasi Algoritma *Mixed Congruential Random Number Generator* Untuk *Game* Siaga Bencana Alam Berbasis Android” (Napitupulu dkk, 2017), perancangan dan pengembangan *game* menggunakan metode MDLC (*Multimedia Development Life Cycle*). Penelitian ini berfokus pada penerapan algoritma *Random Number Generator* yang berfungsi untuk menentukan bencana secara acak berdasarkan tempat yang dilalui oleh karakter yang dimainkan dan juga benda-benda yang jatuh secara acak sesuai bencana yang sedang terjadi. Kelebihan dari penelitian ini adalah berhasil di implementasikannya Algoritma *RNG* yang diterapkan pada *game* yang berfungsi untuk menampilkan secara acak objek yang sudah ditentukan sesuai aturan yang sudah di buat, tampilan dari *game* ini menarik dan cenderung enak untuk dimainkan. Kekurangan dari penelitian hanya mengimplementasikan 1 algoritma dan *gameplay* yang monoton.

Penelitian ketujuh yang berjudul “*Game Artificial Intelegent: Ram City Tower Dengan Algoritma A**” (Meilani dan Suryadi, 2017), perancangan dan pengembangan game menggunakan metode *Waterfall*. Penelitian ini adalah perancangan dan pembangunan *game* edukasi seperti *game Tower Defense* yaitu *player* menempatkan sebuah tower untuk menghancurkan musuh yang akan melewati jalur yang ditentukan, penelitian ini mengimplementasikan algoritma A* yang diterapkan pada musuh untuk menemukan rute terpendek, penelitian ini memiliki edukasi dimana setiap tower dan musuh memiliki nama yang berlawanan atau bisa disebut antonim yang berfungsi sebuah tower hanya bisa menyerang musuh yang memiliki nama yang berlawanan dengan towernya. Kelebihan dari penelitian ini adalah membuat pengguna dapat belajar kata antonim, berhasil mengimplementasikan algoritma A*. Kekurangan dari penelitian ini kata yang terdapat pada *game* ini hanya sedikit yang membuat *game* terasa bosan karena mempelajari kata antonim yang sama.

Penelitian kedelapan yang berjudul “*Algoritma Pathfinding A* pada Game RPG Tanaman Higienis*” (Pramono, 2015), perancangan dan pengembangan game menggunakan metode *Prosedural*. Penelitian ini berfokus pada pembuatan sebuah *game* edukasi yang memiliki tema pertanian dimana dalam *game* ini memiliki tujuan utama untuk bercocok tanam mulai dari menanam, memanen dan melakukan perawatan pada tanaman yang memiliki tingkat kesulitan yang berbeda-beda, penelitian ini menerapkan algoritma A* yang diterapkan pada penghalang yang ada di area atau *map* dalam *game* tersebut yang berfungsi sebagai memeberikan jalan tercepat yang harus dilalui oleh *player*. Kelebihan dari

penelitian ini adalah memberikan edukasi bagaimana cara bercocok tanam yang baik, tampilan *game* pada karakter terlihat menarik dengan menggunakan tampilan kamera perspektif sehingga membuat objek lain seperti rumah dan sebagainya terlihat jelas. Kekurangan dari penelitian ini tidak menerapkan algoritma A* terhadap NPC yang membuat *game* kurang menarik karena NPC tidak melakukan aktivitas.

Penelitian kesembilan yang berjudul “Penerapan Algoritma A *Star* Dan *Fuzzy Logic Sugeno* Pada *Game Pac-Man*” (Halim dkk, 2016), perancangan dan pengembangan *game* menggunakan metode *Prototype*. Penelitian ini merupakan perancangan dan pengembangan *game Pac-Man* yang mengimplementasikan algoritma A* yang diterapkan pada NPC yang berfungsi untuk mencari jalur terpendek terhadap karakter utama dan mengejar karakter utama, sedangkan logika *Fuzzy Sugeno* diterapkan juga terhadap NPC yang berfungsi untuk perilaku dari NPC terhadap karakter utama jika bertabrakan, terdapat beberapa *input/output* yang digunakan dalam algoritma ini pada NPC yakni *distance*, *health*, *bonus* dan *score* yang memiliki fungsi masing-masing dan mempengaruhi karakter NPC maupun karakter utama. Kelebihan dari penelitian ini adalah berhasilnya mengimplementasikan 2 algoritma sekaligus dalam satu kasus yang membuat *game* menjadi menarik.

Penelitian kesepuluh yang berjudul “Penerapan Algoritma *Fuzzy Logic Sugeno* dan Algoritma A* pada *Game Battle City*” (Attoriq dkk, 2016), perancangan dan pengembangan *game* menggunakan metode *Prototype*. Penelitian ini merupakan implementasi dari algoritma A* dan Logika *Fuzzy*

Sugeno yang diterapkan pada *Game Battle City* yang memiliki *gameplay* dimana *tank* yang harus dimainkan harus bertahan melawan *tank* musuh sekaligus menjaga benteng pertahanan (*base*), Algoritma A* diterapkan pada karakter NPC yang berfungsi mencari jalur terpendek untuk menemukan dan mengejar karakter utama, sedangkan logika Fuzzy Sugeno berfungsi sebagai perilaku dari NPC seperti menyerang, bertahan bahkan melarikan diri dari karakter utama. Kelebihan dari penelitian ini adalah berhasilnya implementasi kedua algoritma dilihat dari beberapa kali percobaan dalam game tersebut.

Rangkuman dari penelitian diatas digambarkan dalam tabel 2.1 untuk melihat perbandingan dengan penelitian yang sedang dilakukan yaitu sebagai berikut:

No	Penulis	Metode Penelitian			Metode Pengembangan				Genre Game						Objek		Sistem Operasi		Algoritma								
		Deskriptif	Prosedural	D&C	MDLC	GDLC	RUP	Waterfall	Prototype	Adventure	Arcade	Puzzle	Racing	RPG	Simulator	2D	3D	Android	Windows	Boids	A*	Collision	Patrol Collision	Fuzzy Logic	Greedy	RNG	PNRG & Avoidance System
1	Sholikhin dkk, 2016	√			√					√					√		√										
2	Musfiroh dkk, 2014	√						√		√					√		√		√								
3	Fasha dkk, 2018	√				√								√		√		√									
4	Putri dkk, 2017	√			√				√						√			√					√				
5	Adib dkk, 2016	√			√						√				√		√							√			

No	Penulis	Metode Penelitian	Metode Pengembangan	Genre Game	Objek	Sistem Operasi	Algoritma
----	---------	-------------------	---------------------	------------	-------	----------------	-----------

Tabel 2.1 Perbandingan Penelitian Sebelumnya

Tabel 2.1 Perbandingan Penelitian Sebelumnya (Lanjutan 1)

		PNRG & Avoidance System	RNG	Greedy	Fuzzy Logic	Patrol Collision	Collision	A*	Boids	Windows	Android	3D	2D	Simulator	RPG	Racing	Puzzle	Arcade	Adventure	Prototype	Waterfall	RUP	GDLC	MDLC	D&C	Prosedural	Deskriptif
6	Napitupulu dkk, 2017		√								√	√	√	√										√			√
7	Meilani dan Suryadi, 2017							√			√		√					√			√						√
8	Pramono, 2015							√		√		√			√								√		√		
9	Halim dkk, 2016				√			√		√		√						√		√						√	
10	Attoriq dkk, 2016				√			√			√	√			√					√						√	

Tabel 2.1 Perbandingan Penelitian Sebelumnya (Lanjutan 2)

2.2. *State of The Art*

State of the art dilakukan dengan maksud untuk menganalisis penelitian terdahulu sehingga mendapatkan informasi hasil yang telah dilakukan sebelumnya dan hubungan dengan masalah yang sedang diteliti.

Hasil dari beberapa penelitian sebelumnya, dapat disimpulkan bahwa peneliti akan mengimplementasikan Algoritma *Collision Avoidance System* dan *Random Number Generator* pada game “Balap Egrang” yang bergenre *Racing* dengan objek 2 Dimensi dan berbasis Android, game ini akan dikembangkan menggunakan perangkat lunak *game engine* yaitu *Unity* dengan metode GDLC (*Game Development Life Cycle*), game ini memiliki karakter yang harus dimainkan dengan cara berlomba dengan musuhnya mulai dari garis *start* sampai garis *finish* dan harus melewati rintangan yang ada dalam setiap lintasan, jika *player* berhasil mendahului musuh sampai garis *finish* maka dinyatakan menang dan dapat memainkan *level* selanjutnya, begitu juga sebaliknya.

Penerapan algoritma *Collision Avoidance System* dan *Random Number Generator* pada game ini adalah sebagai kecerdasan buatan yang akan diterapkan pada AI (*Artificial Intelligence*) NPC (*Non-Player Character*) atau karakter yang akan menjadi lawan pada game ini yang berfungsi sebagai pergerakan dari NPC untuk menghindari rintangan yang akan muncul secara acak.

2.3. *Game*

2.3.1. Definisi *Game*

Game dalam bahasa Inggris berarti permainan. Permainan itu sendiri adalah sesuatu yang dapat dimainkan dengan aturan tertentu, memiliki tujuan yang ingin dicapai, dan pada akhirnya menghasilkan menang dan kalah, dalam penggunaannya, saat ini kata *game* lebih mengacu kepada video *game* yang sebenarnya mengarah kepada sebuah alat, komputer yang membuat tampilan video untuk televisi, tapi kata video *game* sekarang sudah mencakup banyak istilah, seperti permainan yang dimainkan di komputer, *mobile phone*, *console*, dan lain sebagainya (Chandler Heather, Chandler Rafael, 2011: 1-2). *Game* atau permainan adalah sesuatu yang dapat dimainkan dengan aturan tertentu sehingga ada yang menang dan ada yang kalah, bermain *game* sudah dapat dikatakan sebagai *life style* masyarakat dimasa kini, dimulai dari usia anak – anak hingga orang dewasa pun menyukai *video game*, itu semua dikarenakan bermain *video game* adalah hal yang menyenangkan. (Anggara, 2008). *Game* adalah sebuah sistem yang terencana dan secara subjektif yang menggambarkan bagian dari kenyataan. Serangkaian bagian yang berinteraksi satu sama lain. Menciptakan gambaran kehidupan emosional yang subjektif dan mudah untuk dimainkan oleh para pemain (Crawford,1997).

2.3.2. Elemen Dasar *Game*

Menurut Schell (dalam *The Art of Game Design*, 2008) terdapat 4 elemen dasar yang saling berkaitan satu sama lain dalam pembuatan game yaitu:

1. *Mechanics*

Mechanics merupakan prosedur atau aturan-aturan dari sebuah *game*. Elemen *mechanics* menjelaskan *goal* atau tujuan dari *game*, bagaimana pemain mampu ataupun tidak mampu mendapatkan *goal* tersebut. Elemen ini tidak akan ditemui dalam media *entertainment* lain seperti televisi, buku dan media lainnya. Elemen *mechanics* merupakan elemen yang paling membedakan *game* dengan media lainnya.

2. *Story*

Cerita (*story*) merupakan poin yang penting dalam sebuah perancangan *game*. Cerita dalam *game* bisa beraneka ragam, dan ketika *designer* sudah menentukan cerita yang akan mereka rangkai, maka elemen *mechanics* juga harus dikembangkan.

3. *Aesthetics*

Elemen ini terdiri dari tampilan visual *game*, suara, hingga *feel* yang ingin diciptakan dalam *game*. *Aesthetics* memiliki hubungan yang paling mendalam dengan pemain karena menyangkut pengalaman bermain *game* mereka, ketika seorang *designer* sudah menentukan elemen *aesthetic* dalam *game*, maka

teknologi pembuatan *game* juga harus dipertimbangkan untuk menghadirkan *game* yang mampu membangun pengalaman tidak terlupakan bagi pemain.

4. *Technology*

Teknologi yang dimaksud dalam elemen ini tidak hanya menyangkut teknologi tinggi (*high technology*) saja, tetapi setiap material dan interaksi yang mampu membangun *game* bisa disebut sebagai elemen teknologi. Elemen teknologi pada dasarnya merupakan media berlangsungnya estetika dan mekanika melalui cerita yang disampaikan dalam *game* (Schell, 2008: 41-42).

2.3.3. **Jenis - Jenis *Game***

Menurut Tanjung (2013) secara umum *game* dapat dibagi menjadi beberapa jenis berdasarkan *genre* yang diterapkannya, yaitu:

1. *Game* Strategi (*Strategy Games*)

Asal-usul dari *game* strategi berasal dari *game* papan seperti catur dan *Othello*, pada *game* strategi biasanya pemain dapat mengendalikan tidak hanya satu karakter, melainkan beberapa karakter dalam *game* tersebut dengan berbagai jenis tipe kemampuan, kendaraan, hingga pembuatan berbagai bangunan, pabrik dan pusat pelatihan tempur, tergantung dari tema ceritanya.

2. *Role-Playing Games* (RPG)

Game RPG sama seperti *game* strategi, yaitu salah satu *genre* yang dibuat dari *game* yang berasal dari kertas dan pena. Dua hal yang hampir sama pada semua *game* RPG, yaitu konfigurasi dari karakter pemain yang meningkat

berdasarkan *experience* dan jalan cerita (*storyline*) yang kuat. Oleh karena itu, terdapat dua elemen utama yang menjadi kunci suksesnya sebuah *game* RPG.

3. *Game* Olahraga (*Sports Games*)

Game olahraga mempunyai tantangan yang tidak biasa bagi seorang desainer *game*, tidak seperti *game-game* yang lain, dimana pemain memiliki sedikit pengetahuan mengenai dunianya, tetapi *game* olahraga meniru aturan pada olahraga di kehidupan nyata.

4. Simulasi Kendaraan (*Vehicles Simulation*)

Game simulasi kendaraan, pemain dihadapkan dalam suatu kondisi seolah-olah pemain mengemudikan atau menerbangkan sebuah kendaraan, secara nyata atau imajinasi. *Game* simulasi kendaraan yang nyata, salah satu tujuan yang harus dicapai adalah kemiripan kendaraan, seperti karakteristik kinerja mesin (kecepatan dan manuver) dengan kendaraan yang sebenarnya.

5. *Game* Petualangan (*Adventure Games*)

Game petualangan bukan merupakan sebuah kompetisi atau simulasi seperti *game* yang lainnya. *Game* jenis ini tidak menawarkan proses untuk dikelola atau mengalahkan musuh melalui strategi dan taktik. *Game* petualangan merupakan cerita interaktif mengenai karakter yang dikontrol oleh pemain.

6. *Game* Puzzle (*Puzzle Games*)

Game puzzle merupakan *game* yang bertujuan untuk memecahkan sebuah puzzle, terkadang tanpa menyatukan dengan jalan cerita atau tujuan yang lebih besar.

Game jenis ini biasanya bervariasi pada satu tema saja, untuk kesuksesan secara komersil, sebuah *game puzzle* haruslah memiliki tantangan, visual yang atraktif, dan disamping itu, nyaman untuk dimainkan.

7. *Game Aksi (Action Games)*

Games aksi merupakan sumber yang baik bagi elemen desain dari sebuah *game*. *Game* ini relatif sederhana karena membuat analisis dari *game* tersebut lebih mudah, jika dibandingkan dengan *genre* lain. Aturan dari sebuah *game action* menjelaskan dasar dari mekanika *game*.

2.4. *Artificial Intelligence (AI)*

Kecerdasan Buatan atau *Artificial Intelligence (AI)* adalah teknik yang digunakan untuk meniru kecerdasan yang dimiliki oleh makhluk hidup maupun benda mati untuk menyelesaikan sebuah persoalan, untuk melakukan hal ini, setidaknya ada tiga metode yang dikembangkan.

1. *Fuzzy Logic (FL)*

Teknik ini digunakan oleh mesin untuk mengadaptasi bagaimana makhluk hidup menyesuaikan kondisi dengan memberikan keputusan yang tidak kaku 0 atau 1, sehingga dimunculkan sistem logika fuzzy yang tidak kaku. Penerapan logika fuzzy ini salah satunya adalah untuk sistem pengereman kereta api di Jepang.

2. *Evolutionary Computing (EC)*

Teknik Pendekatan ini menggunakan skema evolusi yang menggunakan jumlah individu yang banyak dan memberikan sebuah ujian untuk menyeleksi individu terbaik untuk membangkitkan generasi selanjutnya. Seleksi tersebut digunakan

untuk mencari solusi dari suatu permasalahan. Contoh dari pendekatan ini adalah Algoritma Genetika yang menggunakan ide mutasi dan kawin silang, *Particle Swarm Optimization* (PSO) yang meniru kumpulan binatang seperti burung dan ikan dalam mencari mangsa, *Simulated Annealing* yang menirukan bagaimana logam ditempa, dan masih banyak lagi.

3. *Machine Learning (ML)*

Teknik *Machine Learning* (ML) atau pembelajaran mesin merupakan teknik yang paling populer karena banyak digunakan untuk menggantikan atau menirukan perilaku manusia untuk menyelesaikan masalah. Sesuai namanya ML mencoba menirukan bagaimana proses manusia atau makhluk cerdas belajar dan menggeneralisasi (Hania, 2017)

2.5. *Artificial Intelligence (AI) pada Game*

Kecerdasan buatan adalah penciptaan program komputer yang meniru cara bertindak dan berpikir manusia, suatu proses meniru cara bertindak dan berpikir secara rasional. Definisi cerdas ini mencakup kecerdasan kognitif dan kecerdasan berperilaku (suatu bentuk emulsi dari tindakan dan berpikir), ini juga termasuk pada penerapan sikap rasionalitas dan “kemanusiaan”, karena menjadi manusia kadang-kadang jauh dari rasional, tetapi masih dianggap cerdas. Namun, kecerdasan buatan pada *game* tidak membutuhkan penerapan seluas itu, *game* berbasis kecerdasan buatan tidak mengharapkan segala gagasan mengenai kecerdasan buatan diterapkan padanya. *Game* berbasis kecerdasan buatan dikhususkan pada kemampuan komputer mengendalikan unsur-unsur dalam *game* membuat keputusan cerdas ketika suatu kondisi memiliki beberapa pilihan dengan

hasil akhir yang berbeda, sehingga menghasilkan perilaku yang relevan, efektif, dan berguna. Kecerdasan buatan dalam permainan sangat berorientasi pada hasil, dan dengan demikian, kita dapat mengatakan bahwa dunia *game* sangat berkaitan dengan perkembangan ilmu kecerdasan buatan (Troy, 2015).

2.6. Algoritma *Random Number Generator*

Random Number Generator (RNG) adalah sebuah program atau alat untuk menghasilkan urutan angka atau simbol secara tidak teratur. Sistem ini diaplikasikan ke dalam banyak bidang, seperti sampel statistika, simulasi komputer, kriptografi, bahkan untuk desain. *Random Number Generator* dalam *Game* memiliki kegunaan dari angka acak pertama kali diteliti di dalam konteks perjudian, banyak alat yang menghasilkan angka acak seperti dadu dan rolet dikembangkan untuk digunakan di dalam *games of chance*. *Game* elektronik modern biasanya memuat lebih dari satu *random number generation* untuk menghasilkan angka acak di dalam *game*. Sistem *random number generator* diterapkan juga untuk mekanisme *loot system* dalam *massively multiplayer online role-playing games* (MMORPG). Aspek di dalam *online game* adalah *loot* (barang yang dijatuhkan monster). Mekanisme *random number generator* mengabaikan semua input dari pemain dan memberikan statistika yang sama untuk semua pemain, sehingga membuat terciptanya sebuah sistem yang adil, karena adanya kesamaan antara menggunakan *random number generator* dan melempar (*rolling*) dadu, proses ini juga dikenal sebagai “*rolling*”, dan “*rolling for loot*” adalah proses dimana setiap pemain diberikan nomor acak dengan *random number*

generator yang menentukan apakah mereka akan mendapatkan barang yang diinginkan atau tidak (Hidayat 2011).

1. *Pseudocode Random Number Generator*

```
Start
Int rand()
Void
Rand = random.range()
    Transform local position rand into position x or y or z
End.
```

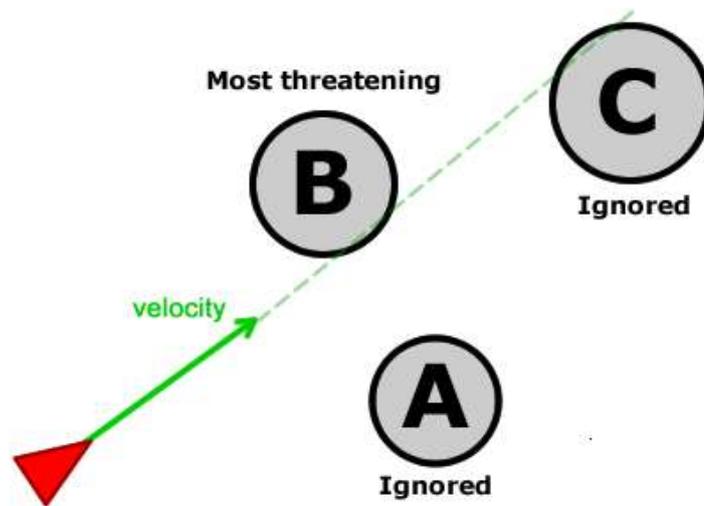
2. Tahapan *Random Number Generator*

Tahapan algoritma *Random Number Generator* sebagai berikut:

- a. Algoritma akan memunculkan objek yang sudah ditentukan secara acak.
- b. Objek akan muncul di posisi x, y maupun z
- c. Objek sudah muncul maka algoritma akan terus bekerja sampai jarak atau waktu yang ditentukan.
- d. Algoritma tidak akan memunculkan objek secara bertumpuk dalam posisi yang sama.

2.7. Algoritma *Collision Avoidance System*

Algoritma *Collision Avoidance System* adalah suatu algoritma yang berfungsi untuk melakukan suatu interaksi terhadap rintangan yang paling dekat kemudian menghindarinya yang di sisipkan kedalam *Non-Player Character* (NPC),



Gambar 2.1 Cara Kerja *Collision Avoidance System* (Bevilacqua, 2013).

Rintangan di depan karakter dianalisis dan yang paling dekat (paling mengancam) dipilih. *Collision Avoidance System* bukan termasuk algoritma *Path Finding* yakni algoritma yang akan membuat NPC bergerak menghindari rintangan (*Obstacle*), langkah pertama untuk menghindari rintangan adalah dengan melihatnya, satu-satunya rintangan yang menjadi ancaman oleh karakter adalah karakter yang ada di depannya dan secara langsung memblokir rute saat ini. (Bevilacqua, 2013).

1. *Pseudocode Collision Avoidance System*

```

Start
  If Obstacle true
    then avoid active
  Else avoid nonactive
Offset
  If obstacle is in right
    Set avoid active then going left
  Else
    If obstacle is in right
      Set avoid active then going
left
      Else
        move Forward
    Endif
  Endif
Endif.

```

2. Tahapan *Collision Avoidance System*

Tahapan algoritma *Collision Avoidance System* sebagai berikut:

- e. Algoritma akan mendeteksi *Obstacle* atau rintangan yang ada didepan.
- f. Dua kemungkinan munculnya rintangan yang harus dilalui
- g. *Obstacle* berada depan kanan maka algoritma akan membaca dan melakukan hindaran ke arah kiri.
- h. *Obstacle* berada depan kiri maka algoritma akan membaca dan melakukan hindaran ke arah kanan.

2.8. Android

Android adalah sistem operasi untuk telepon seluler yang berbasis linux. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri sehingga dapat digunakan oleh bermacam piranti penggerak. Pemrograman android membutuhkan Android SDK (*Software Development Kit*) dan juga *Java Development Kit* (Hanggoro dkk, 2015).

2.9. Unity

Unity merupakan salah satu *game engine* yang banyak digunakan. Unity menyediakan fitur pengembangan *game* dalam berbagai *platform*, yaitu *Unity Web, Windows, Mac, Android, iOS, Xbox, Playstation 3* dan *Wii*, dalam unity disediakan berbagai pilihan Bahasa pemrograman untuk mengembangkan *game*, antara lain *JavaScript, C#, dan BooScript*, namun meskipun disediakan tiga bahasa pemrograman, kebanyakan pengembang menggunakan *JavaScript* dan *C#* sebagai bahasa yang digunakan untuk mengembangkan *game* nya. Unity mendukung pembuatan *game* 2D dan 3D, namun lebih ditekankan pada 3D. Pengembangan *game* lebih ditekankan pada desain dan tampilan visual daripada pemrograman (Hanggoro dkk, 2015).