

IMPLEMENTASI ALGORITMA COLLISION DETECTION DAN FINITE STATE MACHINE UNTUK KARAKTER MUSUH PADA GAME BERTIPE METROIDVANIA

Firmanu Alamsyah¹, Wasum², Amak Yunus³

Teknik Informatika, Universitas Kanjuruhan Malang^{1,2,3}

manu.alamsyah@gmail.com¹, dwasum@yahoo.co.id², amakyunus@unikama.ac.id³

Abstrak. Kecerdasan buatan atau AI (Artificial Intelligence) adalah satu unsur yang diperlukan untuk pembuatan dan pengembangan game yakni untuk membuat permainan lebih dinamis, menarik dan natural. AI yang dapat diterapkan adalah AI yang dapat menentukan perilaku salah satunya adalah Finite State Machine dan didukung Algoritma Collision Detection sebagai pemicu. Penulis menerapkan kedua metode tersebut pada game bertipe metroidvania. Manfaat dari penulisan ini adalah dapat menjadi permulaan dalam project pembuatan game yang berkelanjutan sehingga dapat menghasilkan suatu game yang berkualitas. Game ini diterapkan pada smartphone berbasis android sehingga dapat dimainkan oleh siapa saja dengan harapan game dapat lebih menyenangkan dan lebih menarik bagi pengguna saat dimainkan serta minat pengguna untuk memainkan game juga akan meningkat.

Kata Kunci: *Artificial Intelligence; Collision Detection; Finite State Machine; Metroidvania*

PENDAHULUAN

Dahulu, permainan (*game*) hanya digunakan untuk mengisi waktu luang maupun sebagai media hiburan namun sekarang *game* memiliki berbagai macam kegunaan seperti media pembelajaran, penelitian bahkan dipertandingkan dan menjadi sebuah profesi yang menjanjikan. Perkembangan *game* juga mengalami perubahan yang pesat, mulai dari segi *platform* hingga desain yang semakin baik. *Platform* yang digunakan berawal dari PC (*Personal Computer*) dan berbagai macam *console* seperti *Nintendo*, *Playstation* dan *Xbox*. Tapi sekarang telah memasuki era *mobile* mulai dari *smartphone*, *tablet PCs* dan *portable media player* dengan berbagai macam platform yang ada seperti *Android*, *Apple IOS* dan *Windows Phone*. Keuntungan dari perkembangan ini adalah *game* dapat dimainkan dimana saja dan kapan saja serta menjadi semakin familiar dengan siapapun dari kalangan tua maupun muda. *Game* memiliki banyak *genre* seperti *action game*, *sport game*, *real time strategy*, *role playing game* dan masih banyak yang lain. Salah satu *genre game* yang paling populer adalah *action-adventure*, *genre* ini memiliki beberapa *sub-genre* seperti *survival horror* dan *metroidvania*.

Kecerdasan buatan adalah salah satu ilmu komputer yang mempelajari bagaimana membuat suatu sistem dapat bekerja seperti kecerdasan manusia atau bahkan lebih baik. Kecerdasan buatan dapat diterapkan di berbagai ilmu komputer seperti *microcontroller*, *game*, *network* dan aplikasi lainnya. Salah satu unsur terpenting dalam *game* adalah kecerdasan buatan yang dapat memberikan respon sebaik mungkin atau sealami mungkin. Penerapan kecerdasan buatan dalam *game* dapat dimasukkan dalam berbagai aspek seperti pada karakter utama, karakter musuh, NPC (*Non Playable Character*) dan sistem penentuan *reward* dalam *game* serta proses kenaikan level dan sebagainya. Banyak metode yang dapat diterapkan dalam sebuah *game* untuk membangun sebuah kecerdasan buatan seperti *Collision Detection* dan *Finite State Machine*.

Collision detection atau pendeteksian tumpukan adalah proses pengecekan dua atau lebih buah objek spasial tersebut saling bertumpukan atau tidak. Pada ruang spasial dua dimensi objek dikatakan saling bertumpuk apabila dua buah objek atau lebih saling beririsan (Nugraha, 2010). Penerapan algoritma ini akan membuat karakter dalam *game* dapat mendeteksi objek-objek di sekitar dan membuat pergerakan akan semakin mendekati realistis. Selain pergerakan menjadi lebih realistis, *collision detection* juga dapat di gunakan sebagai pemicu akan tindakan yang akan dilakukan oleh metode *Finite State Machine* (FSM). FSM akan menentukan tindakan untuk menyerang, berjalan maupun bertahan ketika mengetahui objek apa yang menjadi pemicu sehingga permainan akan menjadi lebih menarik. Oleh sebab itu, munculah pemikiran untuk menerapkan *Collision Detection* dan *Finite State Machine* untuk karakter musuh agar menghasilkan sebuah *game* yang memiliki pergerakan yang lebih dinamis dan realistis serta menghasilkan kualitas karakter musuh yang lebih menarik. Penelitian ini nantinya akan menjelaskan tentang hubungan *Collision Detection* dan *Finite State Machine* yang diharapkan dapat berguna pada pembuatan penelitian yang sejenis dan bisa dijadikan patokan untuk membuat pergerakan karakter dalam *game* lebih realistis.

METODE PENELITIAN

1. Artificial Intelligence

Salah satu bagian terpenting dalam sebuah *game* adalah kecerdasan buatan. Kecerdasan buatan yang diterapkan dapat menghasilkan perilaku-perilaku dalam *game* sealam mungkin selayaknya manusia. *Game* dengan AI adalah aplikasi untuk memodelkan karakter yang ada dalam permainan baik sebagai lawan, ataupun karakter pendukung yang merupakan bagian dari permainan. Beberapa jenis kecerdasan buatan dalam *game* adalah :

- a. *Pathfinding*
- b. *Neural Network*
- c. Algoritma genetika
- d. Algoritma, Struktur Data dan Representasi

2. Collision Detection

Collision Detection atau pendeteksi tumpukan adalah proses pengecekan dua atau lebih buah objek spasial tersebut saling bertumpukan atau tidak. Pada ruang spasial dua dimensi objek dikatakan saling bertumpuk apabila dua buah objek atau lebih saling beririsan.

Teknik pengecekan tumpukan yang digunakan dapat dikelompokkan menjadi dua yaitu *priori detection* dan *post detection*. Pengecekan tumpukan sebelum tumpukan terjadi disebut *priori detection*, sedangkan pengecekan yang dilakukan setelah tumpukan terjadi adalah *post detection* (Nugraha, 2010).

Algoritma collision adalah sebagai berikut (Nisa dkk, 2016)

Algoritma Collision Detection

Deklarasi:

$x1, x2, y1, y2, lebar1, lebar2, panjang1, panjang2 = float$

Deskripsi:

- a. Inisialisasi $x1, x2, y1, y2, lebar1, lebar2, panjang1, panjang2$.
- b. Terjadi Pergerakan
- c. $hitTesObj = ((x1 < x2 + lebar2) \&\& (x1 + lebar1 > x2) \&\& (y2 < y2 + panjang2) \&\& (y1 + panjang1 > y2))$
- d. Jika kondisi method *hitTesObj* terpenuhi maka akan lanjut ke proses metode FSM nya.
- e. Jika kondisi method *hitTesObj* tidak terpenuhi maka akan terus melakukan perbandingan selesai.

3. *Finite State Machine*

Menurut Marzain dan Qamal (2017), *Finite State Machine* (FSM) adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan perilaku atau prinsip kerja sistem dengan berpaku pada tiga hal, yaitu *state* (keadaan), *event* (kejadian), dan *action* (aksi). Pada saat dalam periode waktu yang cukup signifikan, sistem akan berada pada salah satu *state* yang aktif. Sistem dapat beralih atau berpindah menuju *state* lain jika mendapat rangsangan atau *event* tertentu, baik yang berasal dari perangkat luar atau komponen dalam sistemnya itu sendiri. Transisi keadaan ini biasanya juga disertai oleh aksi yang dilakukan oleh sistem ketika menanggapi rangsangan yang terjadi. Aksi yang dilakukan tersebut dapat berupa aksi yang sederhana atau melibatkan rangkaian proses yang relatif kompleks.

HASIL DAN PEMBAHASAN

1. Perancangan alur cerita

Game Arsa Ijen Ing Alas ini menceritakan seorang anak manusia yang bertualang di dalam alas khatulistiwa. Adapun alur cerita dari *game* ini adalah sebagai berikut :

- a. Pada sebuah negeri terdapat legenda di hutan yang bernama Alas Khatulistiwa. Legenda itu adalah Arsa Ijen Ing Alas. Dalam legenda ini diceritakan bahwa seorang anak manusia ditakdirkan untuk menjadi seorang pahlawan yang melawan kegelapan. Nama anak manusia tersebut adalah Arsa.
- b. Pada suatu hari di seribu tahun yang lalu. Di dalam Alas Khatulistiwa terdapat iblis yang tersegel yang bernama Loka. Namun, kini Loka telah bangkit kembali untuk menjalankan rencana jahatnya yang sempat tertunda. Arsa ditakdirkan untuk bertarung dan mengagalkan rencana jahat dari Loka yang jahat.

2. Perancangan *Assets Game*

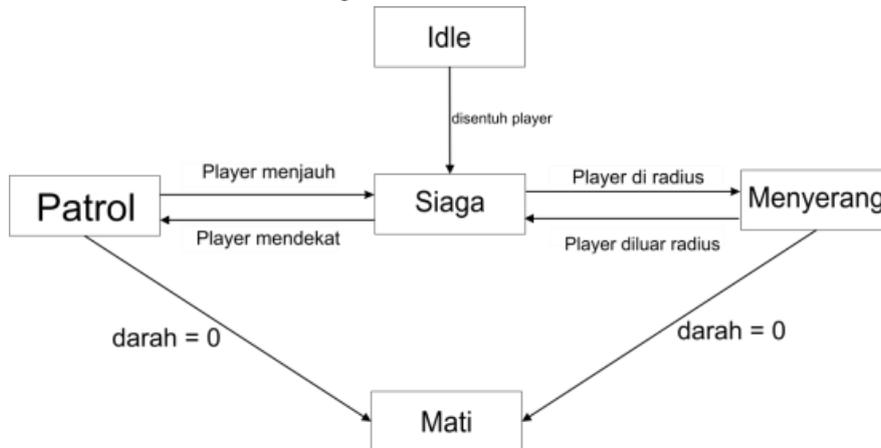
Berdasarkan perancangan alat yang sudah dilakukan perlu dilakukan pengujian dan analisa untuk Dalam proses pembuatan sebuah *game* tidak terlepas dari *assets*. *Assets* dalam sebuah *game* bisa berupa 2D, 3D model dan animasi, teks, ikon, gambar statis, efek suara dan musik latar. Beberapa *assets* 2D yang digunakan pada *game* Arsa Ijen Ing Alas dapat dilihat pada Tabel 1.

Tabel 1. Rincian *Assets Game* Nama, Tampilan dan Keterangan

NAMA	GAMBAR	KETERANGAN
ARSA		Arsa adalah karakter utama pada game Arsa Ijen Ing Alas.
MORTAR KOMANG		Makhluk sejenis katak, Jika karakter utama mendekati Komang, ia akan menembakan proyektil dari mulutnya untuk menyerang.
GRAGAL		Licik dan kejam, Gragal adalah iblis batu

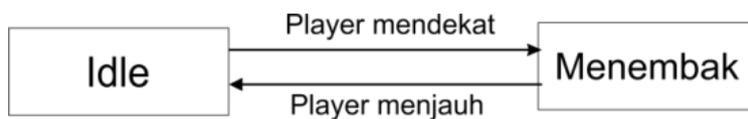
4. Penerapan Metode

Perancangan *Finite State Machine* Gagal.



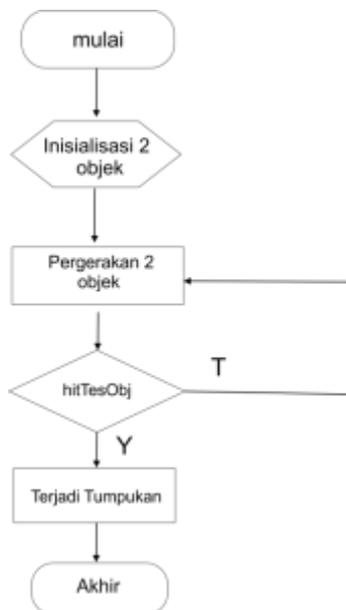
Gambar 1. FSM dari Gagal

Perancangan *Finite State Machine* Mortar Komang



Gambar 2. FSM dari Mortar Komang

Penerapan algoritma *Collision Detection* pada game Arsa Ijen Ing Alas berikut gambar *flowchart Collision Detection* antara karakter utama dan musuh reguler.



Gambar 3. Flowchart Collision Detection

hitTesObj adalah sebuah *method* untuk menghitung terjadinya tumpukan sesuai dengan rumus pada algoritma. Bisa dikatakan terjadi tumpukan (*Collision Detection*) jika semua kondisi dari rumus tersebut terpenuhi. Setelah diketahui terjadi tumpukan maka akan memicu metode

Finite State Machine yang telah ditentukan sehingga menghasilkan respon atau pergerakan karakter yang sesuai dengan metode.

PENUTUP

Dari hasil pengujian LVCB (*Low Voltage Circuit Breaker*) di Laboratorium Teknik Berdasarkan pembahasan pada implementasi algoritma *Collision Detection* dan *Finite State Machine* untuk menentukan aksi pada karakter musuh dalam *game* bertipe *Metroidvania* dengan nama Arsa Ijen Ing Alas dapat disimpulkan telah berhasil menghasilkan perilaku musuh yang realistis dan dapat menyesuaikan dengan keadaan yang terjadi. Perilaku musuh yang realistis adalah musuh dapat mendeteksi adanya tumpukan atau *Collision Detection* yang terjadi dan memberikan respon sesuai dengan metode *Finite State Machine* yang telah ditentukan. Seperti ketika *state* awal Gagal dalam keadaan diam dan terjadi tumpukan *collision* dengan karakter Arsa, *method* hitTesObj dapat melakukan pengecekan tumpukan. Ketika terdeteksi tumpukan maka *state* awal Gagal yang semula diam menjadi bergerak.

Saran

Berdasarkan perancangan dan implementasi program, *game* yang telah dibuat masih memiliki banyak kekurangan, jadi perlu dikembangkan lebih lanjut untuk kedepannya. Maka penulis menyarankan beberapa hal yang perlu diperhatikan dalam mengembangkan *game* ini, sebagai berikut :

1. Penelitian lanjutan perlu dilakukan untuk penerapan *Collision Detection* dan *Finite State Machine* agar dapat diterapkan secara maksimal.
2. Memperbaiki efek suara, musik dan animasi agar *game* lebih baik. Sehingga, ketika *game* dimainkan akan terasa lebih natural dan lebih hidup.

Demikian saran yang dapat penulis berikan, semoga dapat dijadikan sebagai bahan masukan yang bermanfaat bagi penulis khususnya dan bagi pengembang pada umumnya.

DAFTAR PUSTAKA

- Abadi, Rizky Gita. 2016. Rancang Bangun Aplikasi Game Fun With Physic Berbasis Android. Makasar : Universitas Islam Negeri Alaudin
- Asmiatun, Siti. 2016. Penerapan Algoritma *Collision Detection* dan Bayesian untuk Strategi Menyerang Jarak Dekat pada NPC (Non Player Character) Menggunakan Unity 3D. Semarang : Fakultas Teknologi Informas dan Komuniasi Universitas Semarang.
- Asmiatun, Siti dan Putri, Astrid Novita. 2017. Belajar Membuat Game 2D dan 3D Menggunakan Unity. Yogyakarta:Deepublish.
- Bimantoro, Tito dan Haryanto, Hanny. 2016. Pemodelan Perilaku Musuh Menggunakan Finite State Machine (FSM) Pada Game Pengenalan Unsur Kimia. Semarang:Universitas Dian Nuswantoro.
- Fitria, Veni. 2016. Game 2 Dimensi Berbasis Android untuk Membantu Terapi Anak Autis Menggunakan Metode PECS. Malang : Universitas Kanjuruhan Malang.
- Mustaqbal, M.Sidi. dkk. 2015. Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi Kasus : Aplikasi Prediksi Kelulusan Snmptn). Bandung : Universitas Widyatama.
- Nisa, Yulianti Haerum. dkk. Penerapan Metode *Collision Detection* Dalam Permainan Berbasis Android. Bogor: Universitas Pakuan

Nugraha, Rizky Maulana. 2010. Penggunaan Struktur Data Quad-Tree dalam Algoritma Collision Detection pada Vertical Shooter Game . Bandung : Institut Teknologi Bandung.

Nurmansyah, Niman. 2012. Pembangunan Game V-Moe Attack. Bandung : Universitas Komputer.

Putra, Ridho Pratama. 2018. Implementasi Behaviour Tree untuk mengatur perilaku musuh dalam game bertipe metroidvania. Malang : Universitas Kanjuruhan Malang.

Suyanto, ST, Msc, Artificial Intelligence : Searching, Reasoning, Planning, Learning, Informatika, 2007

W. J. Teahan, Artificial Intelligence - Agents and Environments, Bookboon, 2010