

# Pembuatan Game RPG Adventure Of The Dungeon

Muhamad Ulul Azmi<sup>1</sup>, Julian Sahertian<sup>2</sup>, Ahmad Bagus Setiawan<sup>3</sup>

<sup>1,2,3</sup>Teknik Informatika, Fakultas Teknik, Universitas Nusantara PGRI Kediri

E-mail: [e124p3n71n9@gmail.com](mailto:e124p3n71n9@gmail.com)<sup>1</sup>, [juliansahertian@unpkediri.ac.id](mailto:juliansahertian@unpkediri.ac.id)<sup>2</sup>, [bagus.este@gmail.com](mailto:bagus.este@gmail.com)<sup>3</sup>

**Abstrak**—Kebanyakan dari game RPG (Role Playing Gaming) hanya menerapkan logika sederhana kepada karakter musuh berupa pola serangan yang mudah ditebak dan pola serangannya berulang-ulang. Meskipun pola pergerakannya banyak tapi hanya berfokus pada pola serangannya saja, sedangkan untuk pertahanannya hampir tidak ada. Dan pola serangannya hanya berubah berdasarkan jumlah health point (HP) dari musuh. Berdasarkan hal ini, game RPG dibuat dengan mengimplementasikan AI pertahanan pada karakter musuh pada platform PC atau Laptop, AI yang digunakan adalah Algoritma FSM (Finite State Machine) yang berfungsi untuk mengatur pertahanan musuh. Berdasarkan tes fungsional yang telah dilakukan memperoleh hasil baik dan game dapat dikembangkan.

**Kata Kunci**— Finite State Machine, Game, RPG

## 1. PENDAHULUAN

Perkembangan teknologi dan game belakangan ini memang sangat pesat sekali. Baik game-game online maupun game-game offline. Secara umum game merupakan sebuah aplikasi yang menyimpulkan kehidupan manusia kedalam perangkat lunak. Game dapat menjadi alternatif hiburan untuk mengisi waktu luang yang sangat menyenangkan bagi anak-anak maupun orang dewasa.

Ada berbagai genre pada game salah satu genre yang paling terkenal adalah genre RPG. Genre ini merupakan salah satu genre game yang sangat menarik karena berisi unsur-unsur cerita yang kompleks, serta seni peran yang akan membuat pemain seolah-olah menjadi tokoh yang diperankan dalam game tersebut. Oleh karena itu game bergenre RPG mempunyai daya tarik tersendiri, mengapa?, karena pemain harus memerankan karakter sesuai dengan perannya untuk bisa mengembangkan permainan dan juga membuat tiap pemain bisa memilih karakter sesuai dengan yang mereka inginkan (Custom Character).

Namun kebanyakan dari game RPG hanya menerapkan logika sederhana kepada musuh berupa pola serangan yang mudah ditebak dan pola serangannya berulang-ulang. Meskipun pola pergerakannya banyak tapi hanya berfokus pada pola serangannya saja, sedangkan untuk pertahanannya hampir tidak ada. Dan pola serangannya hanya berubah berdasarkan jumlah health point (HP) musuh. Untuk mengatasi masalah tersebut diperlukan skill bertahan dan skill menghindar pada karakter musuh. Agar permainan seimbang maka skill musuh diberi durasi dan cooldown agar player tetap bisa menyerang atau memberi damage.

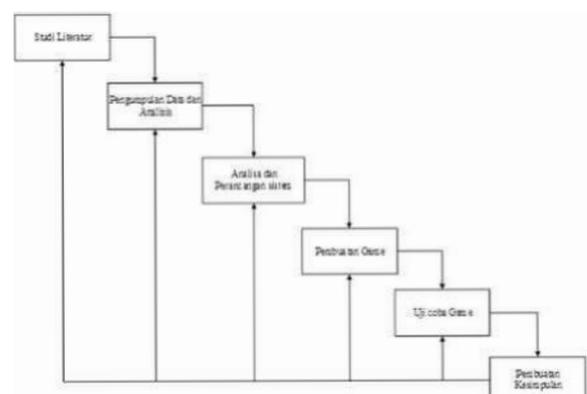
Salah satu penerapan kecerdasan buatan di game untuk pengambilan keputusan yang cerdas adalah Finite State Machine (FSM)[1]. Metode Finite State Machine sendiri dipilih karena merupakan metode

yang cukup sederhana namun sangat baik dalam memodelkan perilaku agen. Metode finite state machine menggunakan state (kondisi), event (kejadian) dan action (aksi). Yang saling berkaitan. Penerapan metode finite state machine pada agen digunakan agar agen dapat terlihat seperti mengambil keputusan saat state yang terdapat pada agen tersebut terpenuhi[2]. Didalam RPG, agen permainan yang berperan dapat sebagai musuh ini biasa disebut dengan NPC.

Pada penelitian-penelitian sebelumnya metode ini dapat membuat permainan lebih tertata dan teratur serta respon perilaku NPC terhadap perubahan kondisi yang terjadi pada pemain dan pada lingkungan game[3].

## 2. METODE PENELITIAN

Alur metode penelitian yang digunakan dalam penelitian ini menggunakan metode waterfall sebagai berikut :



Gambar 1. Metode Waterfall

## 2.1 Studi Literatur

Pada tahap ini dipelajari literature dari perencanaan serta konsep awal untuk merancang *game* yang akan dibuat yaitu didapat dari referensi buku, internet, maupun sumber-sumber yang lain.

## 2.2 Pengumpulan Data dan Analisis

Pada tahap ini adalah proses pengumpulan data yang dibutuhkan untuk pembuatan *game*, serta melakukan analisa atau pengamatan pada data yang sudah terkumpul untuk selanjutnya diolah lebih lanjut.



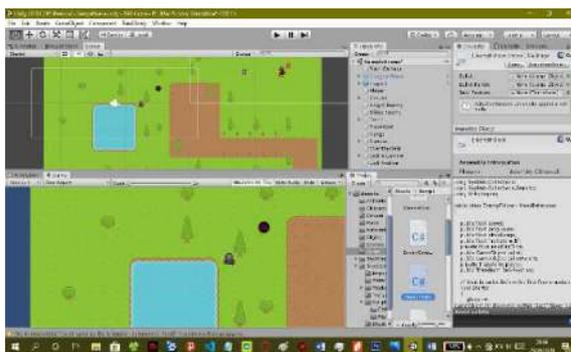
Gambar 2. Proses pengeditan karakter musuh

Pada Gambar 2 proses pengeditan karakter musuh untuk animasi idle nya.



Gambar 3. Tampilan menu utama

Pada menu utama terdapat tombol *Continue*, *New Game*, *Options* dan *Quit*.



Gambar 4. proses pembuatan game

Pada gambar 4 dilakukan proses pembuatan game dari data yang sebelumnya sudah terkumpul. Proses pembuatan game ini menggunakan *Software Unity 3D*.

## 2.3 Analisa dan Perancangan Sistem

Setelah selesai pada tahap pengumpulan data dan analisis maka tahap selanjutnya adalah melakukan analisa dan perancangan sistem. Pada tahap ini adalah proses perancangan dari *game* yang akan dibuat untuk selanjutnya akan diproses lebih lanjut.

## 2.4 Pembuatan Game

Setelah tahap perancangan sistem maka tahap selanjutnya adalah pembuatan *game*. Pada tahap ini rancangan yang sebelumnya telah dibuat akan diterapkan pada program yang akan dibuat.

## 2.5 Uji Coba Game

Setelah gameselesai dibuat maka dilakukan pengujian program untuk mengetahui apakah *game* tersebut telah beralan dengan benar sesuai dengan sistem yang dibuat.

## 2.6 Pembuatan Kesimpulan

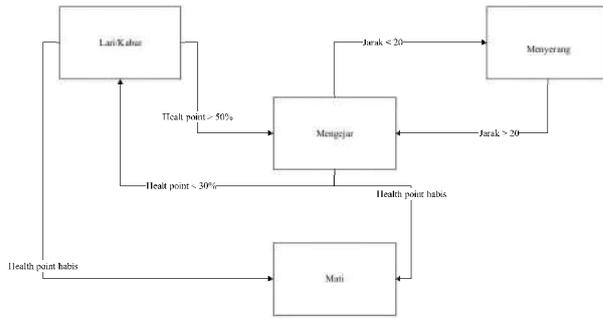
Pada tahap akhir ini adalah pembuatan kesimpulan atau ringkasan dari *game* yang dibuat.

## 2.7 Finite State Machine

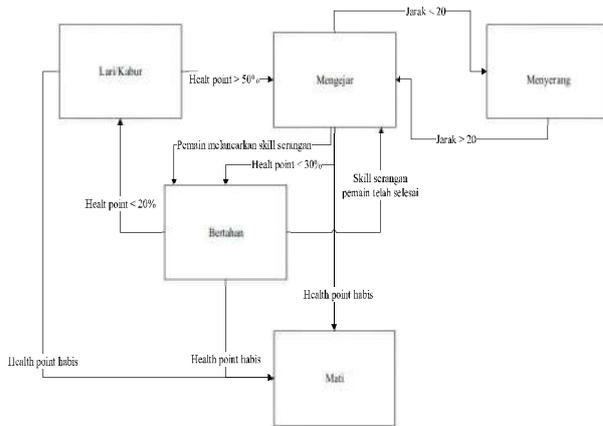
*Finite State Machine* (FSM) adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal, yaitu *state* (keadaan), *event* (kejadian), dan *action* (aksi). Pada saat dalam periode waktu yang cukup signifikan, sistem akan berada pada salah satu *state* yang aktif. Sistem dapat beralih atau bertransisi menuju *state* lain jika mendapat masukan atau event tertentu, baik yang berasal dari perangkat luar atau komponen dalam sistemnya itu sendiri. Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan oleh sistem ketika menanggapi masukan yang terjadi [4].

## 2.8 Algoritma

Penggunaan AI pada karakter musuh akan menggunakan metode *finite state machine* dimana *state* (keadaan), *event* (kejadian), dan *action* (aksi). Jika pemain diam maka musuh akan maju menyerang dan jika pemain menyerang musuh, dan jika skill pertahanan aktif maka gunakan skill untuk bertahan, jika serangan selesai dan skill serangan musuh aktif maka serang pemain dengan skill tersebut.



Gambar 5. Diagram State FSM Kategori Musuh normal



Gambar 6. Diagram State FSM Kategori Musuh Bertahan

Tindakan pemain	Boss skill attack	Boss skill defence	Tindakan bos
Bertahan	Ready	Cooldown	Menyerang pemain dengan skill
Serangan biasa sambil bergerak	Ready	Cooldown	Mengejar sambil menyerang pemain
Serangan biasa	Cooldown	Ready	Menyerang pemain
Serangan skill	Cooldown	Ready	Bertahan dengan skill defence
Bertahan	Cooldown	Ready	Menyerang pemain
Serangan biasa sambil bergerak	Cooldown	Ready	Gunakan skill bertahan sambil kemudian Mengejar sambil menyerang pemain

### 3. HASIL DAN PEMBAHASAN

Dalam penelitian ini menggunakan *Artificial Intelligence* (kecerdasan buatan) dengan metode *Finite State Machine* yang diterapkan kepada karakter musuh. Parameter *input* yang akan digunakan adalah sebagai berikut :

Tabel 1. Algoritma Musuh

Tindakan pemain	Boss skill attack	Boss skill defence	Tindakan bos
Serangan biasa	Ready	Ready	Menyerang pemain
Serangan skill	Ready	Ready	Gunakan skill defence
Bertahan	Ready	Ready	Gunakan skill attack
Serangan biasa sambil bergerak	Ready	Ready	Gunakan skill defence kemudian Mengejar sambil menyerang
Serangan biasa	Cooldown	Cooldown	Menyerang pemain
Serangan skill	Cooldown	Cooldown	Menghindar sambil menyerang
Bertahan	Cooldown	Cooldown	Menyerang pemain
Serangan biasa sambil bergerak	Cooldown	Cooldown	Mengejar sambil menyerang
Serangan biasa	Ready	Cooldown	Menyerang pemain
Serangan skill	Ready	Cooldown	Menyerang dengan skill

Tabel 2. Rancangan Alur FSM

Current State (Q)	Event (e)	Action (a)	State Transition
Idle	Jika player memasuki area musuh	Perilaku objek musuh <i>follow player</i>	<i>Follow player</i>
Idle	Jika player memasuki area serangan musuh	Perilaku objek musuh <i>attack player</i>	<i>Attack player</i>
Idle	Jika player memasuki area musuh dan jika player masuk dalam area serangan musuh	Perilaku musuh <i>follow player</i> dan juga <i>attack player</i>	<i>Follow dan attack player</i>
Idle	Jika nyawa musuh $\leq 0$	Objek musuh <i>Death</i>	<i>Death</i>
Idle	Jika player didalam area serangan	Perilaku musuh <i>idle</i> dan <i>attack player</i>	<i>Idle dan attack player</i>
Idle	Jika player menyerang musuh	Perilaku musuh menghindari atau bertahan	Menggunakan skill bertahan

<i>Curen State (Q)</i>	<i>Event (e)</i>	<i>Action (a)</i>	<i>State Transision</i>
<i>Run</i>	Jika player memasuki area musuh	Perilaku objek musuh <i>follow player</i>	<i>Follow player</i>
<i>Run</i>	Jika player memasuki area serangan musuh	Perilaku objek musuh <i>attack player</i>	<i>Attack player</i>
<i>Run</i>	Jika player memasuki area musuh dan jika player masuk dalam area serangan musuh	Perilaku musuh <i>follow player</i> dan juga <i>attack player</i>	<i>Follow dan attack player</i>
<i>Run</i>	Jika player diluar area musuh	Perilaku musuh <i>idle</i>	<i>Idle</i>
<i>Run</i>	Jika nyawa musuh $\leq 0$	Objek musuh <i>Death</i>	<i>Death</i>
<i>Run</i>	Jika <i>player</i> didalam area serangan	Perilaku musuh <i>idle</i> dan <i>attack player</i>	<i>Idle dan attack player</i>
<i>Run</i>	Jika player menyerang	Perilaku musuh menghindar atau bertahan	Menggunakan skill bertahan
<i>Attack</i>	Jika player memasuki area serangan musuh	Perilaku objek musuh <i>attack player</i>	<i>Attack player</i>
<i>Attack</i>	Jika nyawa musuh $\leq 0$	Objek musuh <i>Death</i>	<i>Death</i>
<i>Attack</i>	Jika <i>player</i> didalam area serangan	Perilaku musuh <i>idle</i> dan <i>attack player</i>	<i>Idle dan attack player</i>
<i>Attack</i>	Jika player menyerang	Perilaku musuh menghindar atau bertahan	Menggunakan skil bertahan

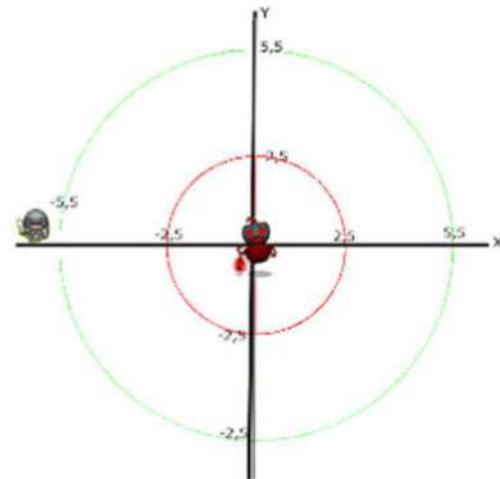
Keterangan:

1. *State(Q)* adalah keadaan
2. *Event(e)* adalah kondisi
3. *Action(a)* adalah aksi

Keterangan:

1. Lingkaran hijau merupakan area jangkauan pada musuh.
2. Lingkaran merah merupakan area serangan pada musuh.

Berikut implementasi Algoritma FSM pada Karakter Musuh.



Gambar 7. Posisi Musuh dan Player

Tabel 3. Algoritma FSM pada Karakter musuh

```

start
Mencari jarak antara player dan musuh
If jarak player < 5,5 && jarak player > 2,5 then
Karakter musuh berjalan mendekati player
Else if jarak player <= 2,5 && serangan selanjutnya < waktu then
Serang musuh
Else if jika jarak player > 5,5 && jarak player > 2,5 then
Karakter musuh kembali ke posisi semula
end
    
```

Kode program ini menjelaskan situasi saat posisi pemain didalam jangkauan musuh dan diluar area serangan musuh, maka musuh mendekati pemain. Dan jika posisi pemain berada dijangkauan serangan musuh maka musuh menyerang pemain dan jika posisi pemain diluar jangkauan dan seerangan musuh maka musuh kembali ke posisi semula.

Tabel 4. Algoritma area jangkauan dan serangan

```

Draw lingkaran warna hijau berdiameter 5,5 dan karakter musuh sebagai pusat
Draw lingkaran warna merah berdiameter 2,5 dan karakter musuh sebagai pusat
    
```

Kode program ini menjelaskan dimana batas target musuh untuk tindakan yang akan diambil saat player mendekati musuh.

Tabel 5. Pengujian menu utama

No	Skenario	Hasil yang diharapkan	Respon aplikasi	hasil
1	Tombol Continue	Menampilkan permainan yang tersimpan	Menu utama ke <i>gameplay</i> yang tersimpan	Diterima
2	Tombol New Game	Menampilkan permainan utama	Menu utama ke <i>gameplay</i>	Diterima
3	Tombol Options	Menampilkan menu pengaturan	Menu utama ke menu pengaturan	Diterima
4	Tombol Quit	Mematikan permainan atau keluar dari permainan	Menu utama ke tampilan pc atau laptop	Diterima

Tabel 6. Pengujian pada karakter musuh

No	Skenario	Hasil yang diharapkan	Respon aplikasi	hasil
1	Karakter musuh	Musuh bergerak mendekati pemain saat pemain dalam jangkauan musuh	Musuh bergerak mendekati pemain	Diterima
2	Karakter musuh	Musuh menyerang pemain saat pemain dalam jangkauan serangan musuh	Musuh menyerang pemain	Diterima
3	Karakter musuh	Pemain berada diluar jangkauan musuh	Musuh kembali ke posisi awal	Diterima
4	Karakter musuh	Saat pemain menyerang musuh	Musuh bertahan	Diterima
5	Karakter musuh	Saat pemain mengalahkan musuh	Musuh mati	Diterima
6	Karakter musuh	Musuh lari menjauh dari pemain jika HP dibawah 30	Musuh bergerak menjauhi pemain	Diterima

*Game* RPG ini dinilai layak karena hasil pengujian melalui metode black-box. Berikut

merupakan hasil pengujian *game* Adventure Of the Dungeon : pada pengujian black-box, (a). presentasi uji tombol *Continue* menampilkan permainan yang tersimpan lalu masuk *game play* yan tersimpan dalam kategori diterima, (b). presentasi uji tombol *New Game* menampilkan permainan utama lalu masuk *game play* dalam kategori diterima, (c). presentasi uji tombol *Options* menampilkan pengaturan lalu masuk ke *menu pengaturan dalam* dalam kategori diterima, (d). presentasi uji tombol *Quit* mematikan permainan atau keluar dari permainan dalam kategori diterima, (e) presentasi uji karakter musuh, musuh bergerak mendekati pemain saat pemain dalam jangkauan musuh lalu musuh bergerak mendekati pemain dalam kategori diterima, (f) presentasi uji karakter musuh, musuh menyerang pemain saat pemain dalam jangkauan serangan musuh lalu musuh menyerang pemain dalam kategori diterima, (g) presentasi uji karakter musuh, pemain berada diluar jangkauan musuh lalu musuh kembali ke posisi awal dalam kategori diterima, (h) presentasi uji karakter musuh, saat pemain menyerang musuh lalu Musuh bertahan dalam kategori diterima, (i) presentasi uji karakter musuh, saat pemain mengalahkan presentasi uji karakter musuh, musuh lalu Musuh mati dalam kategori diterima, (j) presentasi uji karakter musuh, musuh lari menjauh dari pemain jika HP dibawah 30 lalu Musuh bergerak menjauhi pemain dalam kategori diterima.

#### 4. SIMPULAN

Berdasarkan hasil dari penelitian yang telah dilakukan dapat ditarik kesimpulan sebagai berikut : telah berhasil mengembangkan *game* “Adventure Of the Dungeon” menggunakan *Software Unity 3D Game Engine*. *Game* Adventure Of the Dungeon ber-genre RPG. *Game* ini memiliki 3 level yang berbeda level 1, level 2, level 3, pada level 1 ini karakter musuh dan tingkat kesulitan permainan berada pada tingkat rendah dan sebagai awal mula cerita permainan. Pada level 2 ini karakter musuh dan tingkat kesulitan dari permainan berada pada tingkat sedang dan sebagai pertengahan cerita permainan. Pada level 3 ini karakter musuh dan tingkat kesulitan dari permainan berada pada tingkat sulit dan sebagai akhir dari cerita permainan. *Game* ini akan menampilkan halaman kredit *game* kemudian kembali ke menu utama *game* jika pemain sudah menyelesaikan ketiga level yang ada pada *game*.

#### 5. SARAN

Pada penelitian ini masih terdapat banyak kekurangan. Penelitian ini masih dapat dikembangkan lagi untuk mendapatkan hasil permainan yang lebih baik, lebih seru dan lebih menantang lagi. Saran yang direkomendasikan oleh peneliti adalah menambah *state* (keadaan), *event*(kejadian), dan *action*(aksi) yang lebih banyak lagi dan lebih menarik lagi baik pada karakter musuh

atau karakter pemain. Saran kedua oleh peneliti adalah mengembangkan ke permainan smartphone atau android. Ketiga saran oleh peneliti adalah mengembangkan tampilan dari *game* 2D ke *game* 3D.

#### DAFTAR PUSTAKA

- [1] Jones, M. T. 2015. Artificial Intelligence: A Systems Approach: A System Approach. Jones & Bartlett Learning.
- [2] Andrea, R., & Palupi, S. 2018. Membangun Edugame “Boni Kids–Borneo Animal Kids” Permainan Match-up dengan Teknik Pengacakan Shuffle dan Pengembangan Agen Cerdas dengan Model Finite State Machine (Fsm). *Sebatik*, 19(1), 6-10.
- [3] Yulsilviana, E., & Ekawati, H. (2019). PENERAPAN METODE FINITE STATE MACHINE(FSM) PADA GAME AGENT LEGENDA ANAK BORNEO. *SEBATIK*, 116.
- [4] Setiawan, I., 2016. Perancangan Software Embedded System Berbasis FSM Online: <http://elektro.undip.ac.id/iwan/perancangan>, p.20.