

# PENGGUNAAN FINITE STATE MACHINE UNTUK MEMODELKAN PERILAKU MUSUH PADA ROLE-PLAYING GAME 2 DIMENSI “THE SOUL”

*Implementation of Finite State Machine To Model Enemies Behavior In The 2 Dimensional Role-Playing Game "The Soul"*

Aries Gilang Saputra<sup>1</sup>, Hanny Haryanto<sup>2</sup>

<sup>1,2</sup>Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Semarang  
E-mail: <sup>1</sup>111201710710@mhs.dinus.ac.id, <sup>2</sup>hanny.haryanto@dsn.dinus.ac.id

## Abstrak

Banyak sekali jenis permainan yang sudah beredar luas pada saat ini. Salah satu jenis game yang banyak dimainkan oleh orang-orang adalah RPG. Jenis game tersebut merupakan jenis game yang dimana terdapat peran-peran yang dimiliki tiap karakter dan mereka memainkan peran tersebut di dalam permainan. Di dalam permainan tersebut tentu saja memiliki musuh dan musuh tersebut tidak dimainkan oleh pemain akan tetapi dimainkan secara otomatis dengan menggunakan AI. Tetapi, Musuh tersebut ada yang masih monoton dan itu membuat player menjadi bosan. Oleh karena itu, disini penulis akan menggunakan *Finite State Machine* untuk memodelkan perilaku musuh di dalam permainan berjenis RPG yang dibuat penulis dan teman penulis yang bernama “The Soul” agar perilaku musuh dapat lebih dinamis. Hasil dari penelitian ini adalah akan menghasilkan perilaku musuh yang dinamis dan tidak monoton yang dibutuhkan di dalam permainan berjenis RPG.

**Kata kunci:** RPG, Finite State Machine, AI, Perilaku Musuh

## Abstract

*There are many types of games that are widely circulated nowadays. One type of game that most people play is RPG. This type of game is a game where each character has a role, and they played that role in the game. In that game surely have enemies and those enemies are not played by players but are played automatically using Artificial intelligence. But Some enemies are still monotonous and that makes players bored. Therefore, the writer will be using Finite State Machine to model enemies' behavior in the RPG type of game made by the writer and his friend which is named "The Soul" to make enemies' behavior can be more dynamic. The result of this research is that it will make dynamic and non-monotonous enemy behavior that is needed in an RPG type of game.*

**Keywords:** RPG, Finite State Machine, AI, Enemies Behavior

## 1. PENDAHULUAN

Pengertian game RPG Menurut [1] adalah game yang berisi pemain yang memerankan tokoh dan dapat berinteraksi dengan dunia tersebut. Selain itu, [2] mengatakan bahwa ciri utama dari game RPG adalah *user* di dalam game tersebut berperan sebagai tokoh utama dari game tersebut lalu untuk dapat menyelesaikan game tersebut sampai tuntas maka ada cerita yang harus diikuti oleh *player* tersebut dan pada game ini terdapat perasaan berupa penasaran tentang bagaimana akhir dari game ini atau cerita dari game ini sehingga *player* ingin sekali untuk menyelesaikan game ini agar dapat mengetahui cerita dari game ini.

Penelitian ini berfokus pada RPG game yang *offline* dimainkan hanya oleh satu *user* dan merupakan game RPG 2 dimensi. Game bergenre *Role Play Game* ini merupakan genre yang mempunyai banyak tipe seperti menurut [1] bahwa terdapat sub-genre dari RPG ini seperti

Japanese RPG dan Western RPG yang mempunyai perbedaan utama dalam model pertarungan yang digunakan, dalam Japanese RPG model pertarungan yang digunakan adalah model pertarungan berupa *turn-based* sedangkan pada Western RPG model pertarungan yang digunakan adalah berupa *real-time battle* sistem.

Disini peneliti menggunakan sebuah game yang bergenre *turn-based* RPG dalam melakukan penelitian ini. *Turn-Based Role-Playing Game* adalah sebuah genre yang mana terdapat 2 sisi yaitu sisi pemain dan sisi musuh dalam sistem pertarungannya dengan sisi pemain dikendalikan oleh player sedangkan sisi musuh dikendalikan oleh AI [1]. Tetapi pada game RPG ini biasa terjadi ketidakseimbangan seperti menurut [3] melalui grafik *flow*, menunjukkan jika terjadi ketidakseimbangan antara tingkat kesulitan game dengan kemampuan pengguna seperti kesulitan game tersebut terlalu sulit untuk *user* maka akan menyebabkan *user* akan menjadi marah dan sebal tapi sebaliknya jika kesulitan game terlalu mudah atau *user* yang memainkan game ini terlalu pandai maka akan menyebabkan *user* menjadi cepat bosan dan berhenti memainkan game ini. Karena itu, pentingnya untuk membuat perilaku NPC yang tidak monoton agar pemain yang memainkan game tersebut tidak menjadi bosan dan terus memainkan game tersebut. Terdapat penelitian yang menggunakan game RPG sebagai game edukasi yaitu penelitian [4] yang berisi tentang pembuatan game edukasi untuk mahasiswa yang kekurangpahaman dengan materi kuliah Algoritma Dan Struktur Data yang bertema RPG.

Kecerdasan buatan adalah salah satu bidang studi ilmu komputer yang dapat mengintegrasikan game dengan logika yang dapat bersaing dengan manusia [5]. Selain itu, Menurut [6] AI atau kecerdasan buatan dapat digunakan untuk banyak hal seperti menyelesaikan masalah yang rumit dan kompleks tapi, AI juga dapat digunakan untuk tempat kreativitas dan ekspresi dari *user*. Teknologi sekarang dapat menggunakan kecerdasan buatan untuk mengatur karakter musuh pada game *turn-based* RPG [1]. Selain itu, [5] juga mengatakan bahwa sistem agen cerdas dibutuhkan agar pengalaman *user* dalam bermain game dapat bertambah karena tantangan yang ada dalam game tersebut.

Penelitian tentang AI dalam game telah diteliti oleh beberapa peneliti seperti [6] melakukan penelitian berupa membandingkan apakah L-GCNN memiliki akurasi lebih tinggi dalam mengatur perilaku NPC untuk menyerang dibandingkan dengan metode *Neural Network* lainnya. Penelitian lain dilakukan oleh [7] yaitu tentang mengatur perubahan perilaku pada NPC dengan menggunakan kecerdasan buatan berupa Jaringan Syaraf Tiruan yaitu algoritma Perceptron sesuai *mufradat* dan jumlah poin pemain. Penelitian lain tentang pengaturan perilaku musuh juga dilakukan oleh peneliti [5] yang menggunakan kecerdasan buatan berupa *Finite State Machine* untuk mengatur perilaku musuh. Terdapat juga peneliti yang menggunakan logika *fuzzy* untuk mengatur pengambilan keputusan karakter musuh dalam *battle* game yaitu pada penelitian [1].

Namun, dalam penelitian yang sudah dijelaskan diatas terdapat sedikit kelemahan yang perlu diperbaiki seperti penggunaan parameter yang digunakan masih belum cukup untuk dapat mengatur musuh dengan lebih dinamis lagi sehingga perilaku musuh yang dihasilkan masih kurang dinamis. Peneliti disini meneliti untuk membuat perilaku lebih dinamis dengan memberikan perilaku sesuai dengan keadaan yang digambarkan ke dalam model kecerdasan buatan berupa FSM agar perilaku musuh dapat lebih dinamis lagi dan hal ini diterapkan terutama pada game RPG karena peneliti menggunakan obyek berupa RPG 2 dimensi pada penelitian ini.

Peneliti disini menggunakan metode yang bernama *Finite State Machine* (FSM). Menurut [5] FSM merupakan salah satu model dari kecerdasan buatan yang memiliki kelebihan pada bagian komputasi dan pemahaman serta implementasinya karena itu FSM ini sering digunakan untuk mengatur perilaku NPC di dalam game. Selain itu, Menurut [8] Finite State Machine merupakan salah satu metode yang mudah untuk digunakan dalam game yang memiliki kecerdasan buatan di dalamnya. Karena itu, metode FSM ini dapat membantu peneliti untuk menyelesaikan penelitian ini karena peneliti akan menggunakan FSM ini untuk mengatur perilaku musuh.

Terdapat beberapa penelitian terkait hal ini seperti milik [5] yang menggunakan FSM

untuk mengatur perilaku musuh. Penelitian lain yang menggunakan FSM juga terdapat pada penelitian milik [3] yang menggunakan model FSM untuk mengatur *reward* dalam game agar *reward* tersebut menjadi lebih dinamis. Pada penelitian ini peneliti akan menggunakan model FSM (*Finite State Machine*) untuk mengatur perilaku musuh. Penggunaan model FSM ini dilakukan agar musuh dalam game RPG 2 dimensi yang merupakan objek yang digunakan peneliti untuk penelitian ini menjadi lebih dinamis lagi dan musuh menjadi memiliki banyak perilaku sesuai dengan keadaan yang dihadapi.

## 2. METODE PENELITIAN

Game RPG 2 dimensi *The Soul* merupakan game yang digunakan untuk menguji perilaku musuh yang ada di dalam game ini. Pengujian tersebut dilakukan di dalam *dungeon* 1F dalam game ini yang berisi 3 lantai dengan musuh-musuh yang bertambah disetiap lantai mulai dari lantai 2.

### 2.1 Prosedur Pengumpulan Data

Prosedur pengumpulan data yang dilakukan di dalam penelitian ini adalah dengan menggunakan metode studi pustaka yaitu mengumpulkan referensi-referensi yang dapat ditemukan di dalam jurnal, buku-buku, ataupun artikel online yang sesuai dengan permasalahan pada penelitian ini seperti berikut :

1. Referensi tentang pengertian game khususnya dengan genre *Role-Playing Game* dari jurnal penelitian.
2. Referensi tentang pengertian *Non-Playable Character* terutama musuh pada game dari jurnal penelitian.
3. Referensi tentang *Artificial Intelligence* pada game dari jurnal penelitian.
4. Referensi tentang *Finite State Machine* pada game dari jurnal penelitian.

### 2.2 Teknik Analisis Data

Data yang diperoleh dari pengumpulan data yang digunakan untuk kebutuhan penelitian ini berupa data kualitatif. Data kualitatif adalah data yang bukan merupakan angka atau non numerik dan merupakan bahasan konseptual dari suatu permasalahan. Dalam pengumpulan data yang digunakan oleh peneliti yaitu dari jurnal-jurnal penelitian sebelumnya ditemukan bahwa pada penelitian sebelumnya menggunakan *Finite State Machine* untuk memodelkan *Non-Playable Character* yang bermacam-macam seperti contohnya NPC yang berinteraksi dengan *player*, NPC musuh sampai NPC yang diam juga menggunakan *Finite State Machine* untuk memodelkannya.

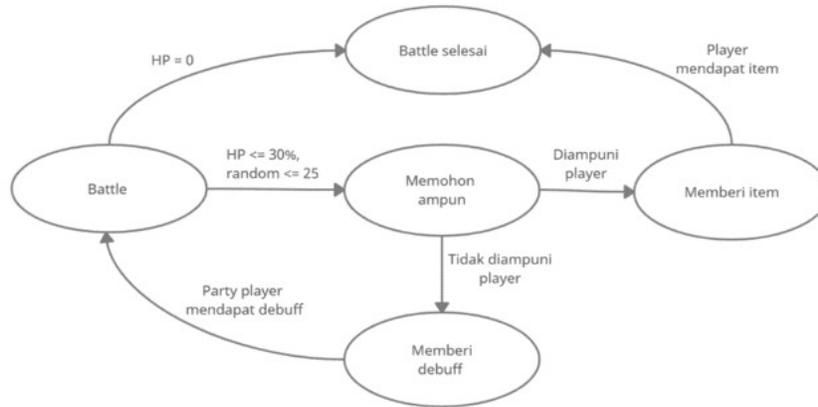
Penelitian ini merupakan penelitian yang berfokus pada perilaku yang dimiliki oleh NPC musuh. Parameter utama yang ada di dalam penelitian ini adalah HP (*Health Point*) karena jika *player* ketika melawan musuh HP (*Health Point*) ada di angka 0 atau habis maka permainan akan berakhir dan akan mengulangi di *save* terakhir yang disimpan. Jika HP (*Health Point*) musuh berkurang hingga 0 atau habis maka itu membuktikan bahwa *player* telah berhasil untuk mengalahkan musuh yang ada. Hasil yang didapat dari mengalahkan musuh tersebut adalah dapat berupa *exp* (*experience*), *item*, dan *gold* (uang).

Tetapi untuk mengalahkan musuh yang ada di dalam game ini tidak mudah karena NPC musuh yang ada di dalam game ini telah diberi perilaku oleh peneliti sehingga NPC musuh tersebut memiliki perilaku masing-masing sesuai musuh yang telah ditetapkan oleh peneliti sehingga *player* tidak mudah bosan dan merasa tertantang dengan musuh-musuh yang ada.

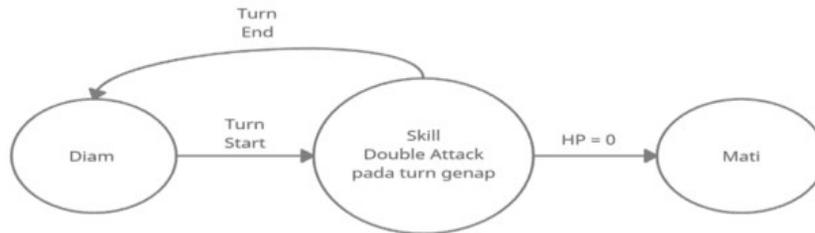
### 2.3 Metode yang diusulkan

Berdasarkan analisis data diatas, maka untuk menggambarkan perilaku musuh yang sudah dibuat oleh peneliti membutuhkan metode berupa *Finite State Machine*. Metode ini sering digunakan karena kemudahan dalam implementasinya. *Finite State Machine* ini sudah

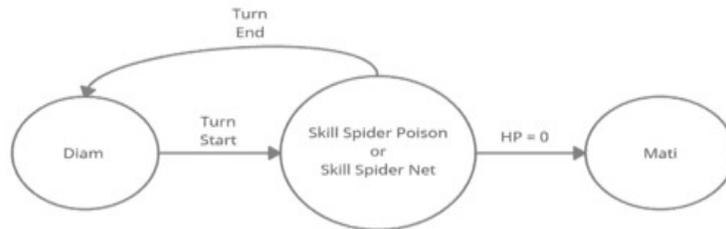
disesuaikan dengan perilaku musuh yang ada di dalam game.



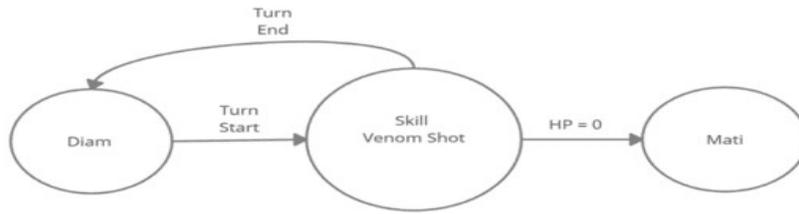
Gambar 1. *Finite State Machine* Musuh Secara Umum



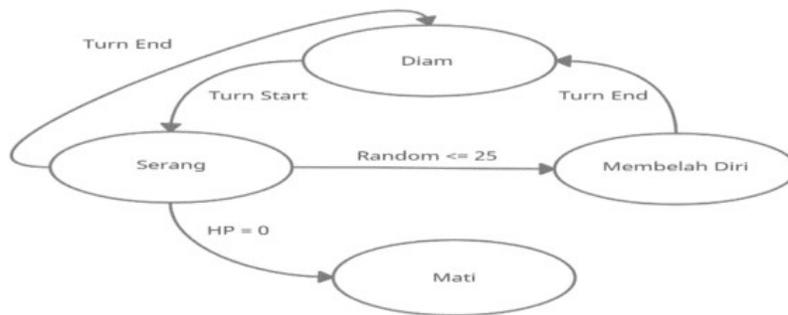
Gambar 2. *Finite State Machine* Perilaku Musuh Bat



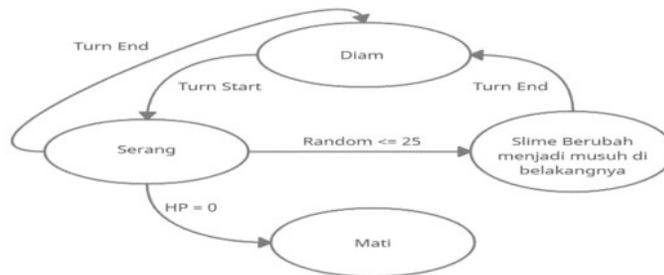
Gambar 3. *Finite State Machine* Perilaku Musuh Spider



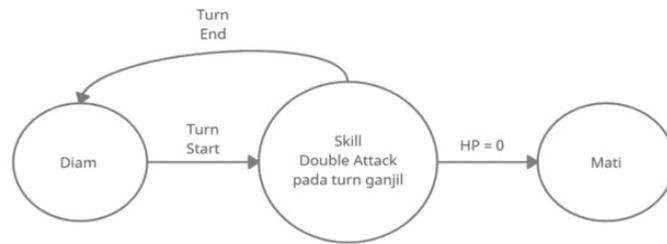
Gambar 4. *Finite State Machine* Perilaku Musuh Snake



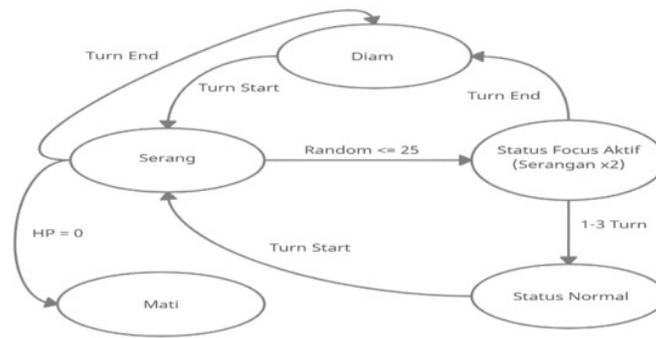
Gambar 5. *Finite State Machine* Perilaku Musuh Slime 1



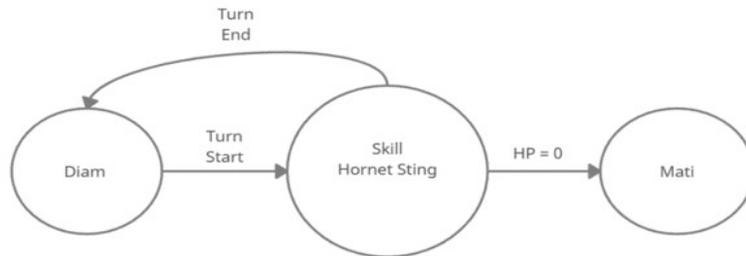
Gambar 6. *Finite State Machine* Perilaku Musuh Slime 2



Gambar 7. Finite State Machine Perilaku Musuh 3



Gambar 8. Finite State Machine Perilaku Musuh Hornet 1



Gambar 9. Finite State Machine Perilaku Musuh Hornet 2

#### 2.4 Kerangka Pemikiran

Musuh merupakan salah satu hal yang penting untuk diperhatikan karena di dalam permainan terutama dengan genre RPG, musuh merupakan bagian dari permainan yang akan membuat permainan tersebut menjadi lebih seru. Tetapi, untuk membuat musuh yang seru bukanlah hal yang mudah karena jika AI yang dimiliki musuh kurang, maka akan membuat pemain menjadi bosan dan akan meninggalkan permainan tersebut. Maka dari itu, dibutuhkan AI yang membuat musuh berperilaku sesuai dengan keadaan saat di dalam *battle* agar membuat *battle* tersebut menjadi lebih seru dan menantang.

Penelitian yang dilakukan peneliti saat ini bertujuan untuk hal tersebut yaitu merancang perilaku musuh terutama dalam permainan dengan genre RPG 2 Dimensi agar memiliki perilaku yang lebih dinamis agar dalam *battle* di dalam permainan tersebut menjadi lebih menarik dan

lebih menantang sehingga *player* tidak akan bosan dengan permainan ini dengan bantuan metode *Finite State Machine* untuk menggambarkan perilaku musuh yang dibuat oleh peneliti dengan harapan dapat membuat musuh yang ada menjadi lebih dinamis sesuai dengan keadaan di dalam battle tersebut.



Gambar 10. Kerangka Pemikiran

### 3. HASIL DAN PEMBAHASAN

Perilaku agen cerdas musuh yang dibuat oleh peneliti pada game RPG “The Soul” ini ada 6 untuk masing-masing agen cerdas musuh yang berbeda. Perilaku ini memiliki kondisi yang berbeda-beda untuk mengaktifkannya. Setelah perilaku ini diterapkan pada semua agen cerdas musuh di dalam game ini, maka perilaku ini perlu untuk dilakukan uji coba apakah perilaku ini dapat berjalan dengan lancar atau tidak.

Percobaan tersebut dilakukan di dalam game langsung, lebih tepatnya di dalam *dungeon* 1 yang sudah dibuat oleh peneliti untuk keperluan uji coba perilaku ini. *Dungeon* ini terdapat 3 lantai yang tiap lantainya memiliki agen cerdas musuh yang berbeda-beda dimana semakin tinggi lantainya maka semakin banyak karena agen cerdas musuh pada lantai sebelumnya juga terdapat atau ikut muncul pada lantai selanjutnya. Hasil percobaan dari *Finite State Machine* tersebut dijelaskan sebagai berikut.

Tabel 1 Tabel Hasil Percobaan Untuk Agen Cerdas Musuh Secara Umum

| Kondisi                                  | Perilaku                        |
|--|---------------------------------|
| HP $\leq$ 30% dan Angka Random $\leq$ 25 | Agen cerdas musuh memohon ampun |
| Player mengampuni agen cerdas musuh      | Agen cerdas musuh memberi item  |

|   |   |
|---|---|
| Player menolak untuk mengampuni agen cerdas musuh | Agen cerdas musuh memberi <i>debuff</i> |
| HP = 0  | Agen cerdas musuh mati                  |



Gambar 11. Dialog Perilaku Agen Cerdas Musuh Menyerah

Perilaku pertama ini adalah perilaku yang diterapkan ke semua agen cerdas musuh yang ada di dalam game ini yaitu berupa perilaku menyerah apabila kondisi terpenuhi. Gambar di atas menunjukkan dialog yang muncul setelah agen cerdas musuh menyerah ketika kondisi untuk munculnya perilaku tersebut terpenuhi. Dialog di atas sendiri adalah ketika player diberikan pilihan apakah ingin memberikan ampun untuk agen cerdas musuh yang dihadapi saat itu atau tidak. Pilihan ini akan memberikan hasil yang berbeda berdasarkan pilihan yang dipilih oleh player.



Gambar 12. Dialog Agen Cerdas Musuh Memberikan Item

Gambar di atas menunjukkan ketika player telah memberikan ampun kepada agen cerdas musuh dan sebagai imbalannya, mereka memberikan *item* kepada player. Ada 4 *item* yang memiliki kemungkinan berbeda untuk diberikan kepada player. Keempat *item* tersebut adalah Potion, Antidote, Magic Water, dan Excalibur. *Item-item* tersebut memiliki kemungkinan yang berbeda untuk masing-masing *item*. Setelah mereka memberikan *item* kepada player maka *battle* pun berakhir dan player melanjutkan perjalanan.



Gambar 13. Agen Cerdas Musuh Memberikan *Debuff*

Gambar di atas menunjukkan ketika player menolak untuk memberikan ampun kepada agen cerdas musuh dan mereka marah lalu memberikan *debuff* kepada party player. Terdapat 2 *debuff* yang memiliki kemungkinan diberikan kepada *party player* yaitu Fascination dan Poison yang memiliki efek berbeda.

Selain Fascination dan Poison, mereka juga memiliki 1 lagi perilaku yang terjadi saat player menolak untuk memberikan ampun yaitu dapat melakukan Heal kepada *party* mereka sehingga HP (*Health Point*) nya menjadi penuh kembali seperti pada saat awal *battle*. Semua perilaku tersebut memiliki kemungkinan yang berbeda untuk masing-masing perilakunya. Setelah mereka memberikan *debuff* atau melakukan Heal dan belum mati maka *battle* akan berlanjut hingga mati. Tetapi, jika setelah memberikan *debuff* mati maka *battle* pun berakhir.

Tabel 2 Tabel Hasil Percobaan Untuk Agen Cerdas Musuh Bat

| Kondisi    | Perilaku                               |
|------------|--|
| Turn Start | Serang atau <i>Skill Double Attack</i> |
| Turn End   | Diam                                   |
| HP = 0     | Mati                                   |



Gambar 14. *Skill Double Attack* Pada Bat

Gambar di atas menunjukkan hasil dari percobaan *Finite State Machine* untuk perilaku yang dimiliki oleh Bat dimana Bat ini dapat mengeluarkan *skill* yang bernama Double Attack. Pada gambar tersebut terlihat bahwa Bat mengeluarkan *skill*nya yaitu Double Attack yang digunakan pada *turn* genap. *Skill* tersebut menyerang sebanyak 2x ke character yang sama atau

berbeda. Setelah mengeluarkan *skill* tersebut Bat akan diam untuk menunggu turn berikutnya.

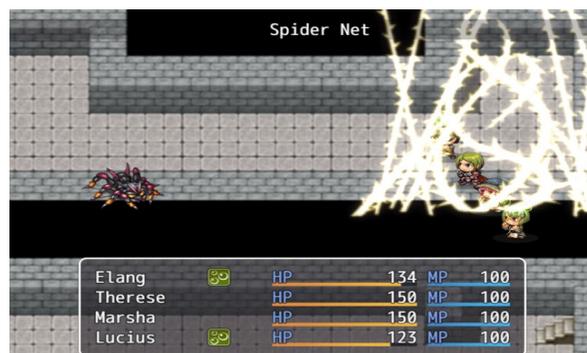
Tabel 3 Tabel Hasil Percobaan Untuk Agen Cerdas Musuh Spider

| Kondisi    | Perilaku   |
|------------|--|
| Turn Start | Serang atau <i>Skill</i> Spider Poison atau Spider Net |
| Turn End   | Diam   |
| HP = 0     | Mati   |



Gambar 15. *Skill* Spider Poison Pada Spider

Gambar di atas menunjukkan hasil percobaan untuk *skill* pertama yang dimiliki oleh Spider yaitu Spider Poison. Pada gambar tersebut terlihat bahwa Spider sedang menggunakan *skill* yang dimilikinya yaitu Spider Poison pada saat *turn* nya dimulai. *Skill* ini menyebabkan status poison dan status tersebut dapat dihilangkan dengan menggunakan *potion* seperti Antidote atau dapat menghilang otomatis pada saat battle berakhir. Setelah mengeluarkan *skill* tersebut Spider akan diam kembali untuk menunggu *turn* berikutnya.



Gambar 16. *Skill* Spider Net Pada Spider

Gambar di atas menunjukkan hasil percobaan untuk *skill* kedua yang dimiliki oleh Spider yaitu Spider Net. Pada gambar di atas terlihat bahwa Spider sedang mengeluarkan *skill* kedua yang dimilikinya yaitu Spider Net yang akan menyebabkan *character* yang terserang oleh *skill* ini akan memiliki status *Netted* dan tidak dapat bergerak selama 1-3 *turn*. Setelah menggunakan *skill*

tersebut, Spider akan kembali diam untuk menunggu *turn* berikutnya.

Tabel 4 Tabel Hasil Percobaan Untuk Agen Cerdas Musuh Snake

| Kondisi           | Perilaku                            |
|-------------------|-------------------------------------|
| <i>Turn</i> Start | Serang atau <i>Skill</i> Venom Shot |
| <i>Turn</i> End   | Diam                                |
| HP = 0            | Mati                                |



Gambar 17. *Skill* Venom Shot Pada Snake

Gambar di atas menunjukkan hasil percobaan untuk perilaku yang dimiliki oleh Snake yaitu *skill* yang bernama Venom Shot dan *skill* ini akan memberikan status poison. Setelah menggunakan *skill* ini, Snake akan diam kembali untuk menunggu *turn* berikutnya.

Tabel 5 Tabel Hasil Percobaan Untuk Agen Cerdas Musuh Slime Membelah Diri

| Kondisi                       | Perilaku            |
|-------------------------------|---------------------|
| <i>Turn</i> Start             | Serang              |
| Angka <i>Random</i> $\leq 25$ | Slime membelah diri |
| <i>Turn</i> End               | Diam                |
| HP = 0                        | Slime mati          |



Gambar 18. Perilaku Membelah Diri Pada Slime

Gambar di atas menunjukkan hasil percobaan untuk perilaku pertama yang dimiliki oleh Slime yaitu membelah diri. Sesuai dengan gambar di atas, Slime membelah dirinya sendiri menjadi 2 karena di dalam *battle* tersebut terlihat bahwa sebelumnya hanya ada 1 Slime maka ketika membelah akan menjadi 2 Slime ketika kondisi yang dibutuhkan untuk membelah terpenuhi. Jika sebelumnya terdapat 2 Slime maka setelah membelah diri akan menjadi 3 Slime. Setelah membelah diri maka *turn* Slime akan berakhir sehingga Slime akan diam untuk menunggu *turn* berikutnya.

Tabel 6 Tabel Hasil Percobaan Untuk Agen Cerdas Musuh Slime Berubah Bentuk

| <b>Kondisi</b>                | <b>Perilaku</b>      |
|-------------------------------|----------------------|
| <i>Turn</i> Start             | Serang               |
| Angka <i>Random</i> $\leq 25$ | Slime Berubah Bentuk |
| <i>Turn</i> End               | Diam                 |
| HP = 0                        | Slime mati           |



Gambar 19. Perilaku Berubah Bentuk Pada Slime

Gambar di atas adalah hasil percobaan untuk perilaku dari Slime selanjutnya yaitu Slime dapat berubah bentuk menjadi sama seperti agen cerdas musuh yang ada di belakangnya saat itu ketika kondisi terpenuhi. Karena pada gambar tersebut agen cerdas musuh yang ada adalah Bat\*2 dan Slime, maka Slime akan berubah bentuk menjadi Bat.

Selain pada Bat\*2 Slime perilaku ini juga terjadi pada Spider\*2 Slime dan Snake\*2 Slime. Perubahan dari Slime sesuai dengan agen cerdas musuh yang ada di belakangnya pada saat itu dan ketika kondisi yang dibutuhkan terpenuhi. Setelah membelah diri maka *turn* dari Slime akan berakhir dan Slime akan diam untuk menunggu *turn* berikutnya.

Tabel 7 Tabel Hasil Percobaan Untuk Agen Cerdas Musuh Slime *Skill* Double Attack

| Kondisi           | Perilaku                               |
|-------------------|--|
| <i>Turn</i> Start | Serang atau <i>Skill</i> Double Attack |
| <i>Turn</i> End   | Diam                                   |
| HP = 0            | Mati                                   |



Gambar 20. *Skill* Double Attack Pada Slime

Gambar di atas adalah hasil percobaan dari *skill* yang dimiliki oleh Slime yaitu Double Attack. Setelah mengeluarkan *skill* ini maka *turn* dari Slime akan berakhir lalu kembali diam untuk menunggu *turn* berikutnya.

Tabel 8 Tabel Hasil Percobaan Untuk Agen Cerdas Musuh Slime *Skill Focus*

| Kondisi                       | Perilaku                   |
|-------------------------------|----------------------------|
| <i>Turn Start</i>             | Serang                     |
| Angka <i>Random</i> $\leq 25$ | <i>Skill Focus</i> Aktif   |
| Setelah 1-3 <i>turn</i>       | <i>Skill Focus</i> Inaktif |
| <i>Turn End</i>               | Diam                       |
| HP = 0                        | Hornet mati                |



Gambar 21. *Skill Focus* Pada Hornet

Gambar di atas menunjukkan hasil percobaan untuk perilaku Hornet yaitu kemampuan untuk menggunakan *skill Focus* yang akan menerapkan efek *buff* kepada Hornet selama 1-3 *turn* dimana serangan dari Hornet ini akan menjadi 2x lipat dari sebelumnya. Setelah menggunakan *skill Focus* ini *turn* Hornet akan berakhir dan Hornet pun akan diam untuk menunggu *turn* berikutnya.

Tabel 9 Tabel Hasil Percobaan Untuk Agen Cerdas Musuh Hornet *Skill Hornet Sting*

| Kondisi           | Perilaku                              |
|-------------------|---------------------------------------|
| <i>Turn Start</i> | Serang atau <i>Skill Hornet Sting</i> |
| <i>Turn End</i>   | Diam                                  |
| HP = 0            | Mati                                  |



Gambar 22. Skill Hornet Sting Pada Hornet

Gambar di atas menunjukkan hasil percobaan untuk *skill* lain yang dimiliki oleh Hornet yaitu Hornet Sting. Skill ini akan menyebabkan status poison. Setelah mengeluarkan *skill* ini maka *turn* Hornet akan berakhir dan akan kembali diam untuk menunggu *turn* berikutnya.

#### 4. KESIMPULAN DAN SARAN

Pada penelitian ini, telah dimodelkan perilaku musuh dengan menggunakan *Finite State Machine* di dalam game “The Soul” yang menghasilkan perilaku musuh di dalam game ini menjadi lebih dinamis sesuai dengan keadaan yang dihadapi. Banyak keterbatasan di dalam penelitian ini seperti perilaku yang dimodelkan dapat dibuat menjadi lebih dinamis lagi. Kekurangan lainnya adalah perilaku musuh yang masih sedikit. Pada penelitian selanjutnya dapat menambahkan perilaku yang lebih banyak lagi kepada setiap musuhnya agar perilaku untuk setiap musuhnya lebih bervariasi lagi.

#### DAFTAR PUSTAKA

- [1] D. Ratanajaya and H. A. Wibawa, “Implementasi Kecerdasan Buatan dalam Menentukan Aksi Karakter pada Game RPG dengan Logika Fuzzy Tsukamoto,” *Khazanah Inform. J. Ilmu Komput. dan Inform.*, vol. 4, no. 2, p. 82, 2018, doi: 10.23917/khif.v4i2.6744.
- [2] S. A. Pramuditya, M. S. Noto, and H. Purwono, “Desain Game Edukasi Berbasis Android pada Materi Logika Matematika,” *JNPM (Jurnal Nas. Pendidik. Mat.*, vol. 2, no. 2, p. 165, 2018, doi: 10.33603/jnpm.v2i2.919.
- [3] H. Haryanto, “Reward Dinamis dalam Skenario Adaptif Menggunakan Metode Finite State Machine pada Game Edukasi,” *J. Appl. Intell. Syst.*, vol. 1, no. 2, pp. 144–153, 2016.
- [4] C. Taurusta, “Rancang Bangun Game Algoritma Dan Struktur Data Berbasis Role Playing Game (Rpg) Sebagai Media Pembelajaran Mahasiswa Teknik Informatika Universitas Muhammadiyah Sidoarjo,” *Kinetik*, vol. 2, no. 3, pp. 175–188, 2017, doi: 10.22219/kinetik.v2i3.167.
- [5] H. Tito, Bimantoro1, Hanny, “Pemodelan Perilaku Musuh Menggunakan Finite State Machine (FSM) Pada Game Pengenalan Unsur Kimia,” *Appl. Intell. Syst.*, vol. Vol.1, No., no. 3, pp. 210–219, 2016.
- [6] I. Mabruroh and D. Herumurti, “Adaptive Non Playable Character in RPG Game Using Logarithmic Learning For Generalized Classifier Neural Network (L-GCNN),” *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, no. 2, pp. 127–136, 2019, doi: 10.22219/kinetik.v4i2.755.
- [7] S. Karim, “Perubahan perilaku Non-Player Character (NPC) pada Game Arabic Hunter menggunakan Jaringan Syaraf Tiruan Perceptron,” *Regist. J. Ilm. Teknol. Sist. Inf.*, vol. 3, no. 1, p. 34, 2017, doi: 10.26594/register.v3i1.622.
- [8] K. Fathoni, R. Y. Hakkun, and H. A. T. Nurhadi, “Finite State Machines for Building Believable Non-Playable Character in the Game of Khalid ibn Al-Walid,” *J. Phys. Conf. Ser.*, vol. 1577, no. 1, 2020, doi: 10.1088/1742-6596/1577/1/012018.