

IMPLEMENTASI BEHAVIOR TREE DALAM MENENTUKAN PERILAKU MUSUH PADA GAME METROIDVANIA “CURSED WORLD”

*Implementation of Behavior Tree in Determining Enemy Behavior in the Metroidvania
Game "Cursed World"*

Pandji Alam¹, Hanny Haryanto²

^{1,2,3}S-1 Teknik Informatika, Universitas Dian Nuswantoro

E-mail: ¹111202012957@mhs.dinus.ac.id, ²hanny.haryanto@dsn.dinus.ac.id

Abstrak

Game Metroidvania menghadirkan tantangan eksplorasi map dan mengalahkan musuh dengan tingkat kesulitan beragam. Untuk meningkatkan kualitas permainan, penggunaan kecerdasan buatan (AI) pada karakter musuh telah menjadi pilihan. Dalam Metroidvania, musuh memiliki variasi perilaku yang kompleks. Perilaku musuh ini diatur dan ditentukan agar memiliki tantangan yang baik bagi pemain. Metode Finite State Machine (FSM) merupakan kecerdasan buatan untuk mengatur setiap perubahan pada perilaku musuh. Metode ini bisa mengatur perilaku musuh dengan baik, tetapi metode ini sangat sulit diterapkan pada perilaku musuh yang kompleks. Untuk mengatasi masalah ini, metode Behavior Tree (BT) dipilih untuk mengelola perilaku musuh dalam game Metroidvania. Empat musuh dikembangkan dengan BT, menciptakan struktur pemodelan modular dan mendalam. Metode BT, sebagai pohon dengan root, control-flow, dan execution node, merupakan metode terstruktur yang memudahkan penyusunan perilaku musuh. Hasil penelitian menunjukkan peningkatan tantangan dan pengalaman bermain yang baik, diuji dengan pendekatan Game Experience Questionnaire (GEQ) menunjukkan rata-rata sebesar 4.05. Kesimpulan menyatakan bahwa penggunaan BT memberikan kontribusi positif pada pengembangan game Metroidvania “Cursed World” melalui struktur pemodelan modular dan tingkat kesulitan yang disesuaikan serta memberikan pengalaman bermain yang baik bagi pemain.

Kata kunci: Metroidvania, Kecerdasan buatan, FSM, Behavior Tree, GEQ

Abstract

The Metroidvania game presents challenges in exploring maps and defeating enemies with various difficulty levels. To enhance the game's quality, the use of Artificial Intelligence (AI) in enemy characters has become a choice. In Metroidvania, enemies exhibit complex variations in behavior. The behavior of these enemies is organized and determined to provide a substantial challenge for players. The Finite State Machine (FSM) method is an artificial intelligence approach to regulate every change in enemy behavior. While this method can effectively manage enemy behavior, it proves challenging to implement for complex enemy behaviors. To address this issue, the Behavior Tree (BT) method was chosen to manage enemy behavior in the Metroidvania game. Four enemies were developed using BT, creating a modular and deep modeling structure. The BT method, as a tree with root, control-flow, and execution nodes, is a structured approach that facilitates the organization of enemy behavior. The research results demonstrate an improvement in challenges and a positive gaming experience, tested with the Game Experience Questionnaire (GEQ) approach, showing an average score of 4.05. The conclusion states that the use of BT contributes positively to the development of the 'Cursed World' Metroidvania game through modular modeling structure, adjusted difficulty levels, and providing a satisfying gaming experience for players.

Keywords: Metroidvania, Artificial Intelligence, FSM, Behavior Tree, GEQ

1. PENDAHULUAN

Game *Metroidvania* adalah salah satu sub-genre *game action-adventure* yang mengharuskan pemain untuk menjelajahi atau eksplorasi map dan mengalahkan musuh dengan tingkat kesulitan yang berbeda. *Metroidvania "Cursed World"* merupakan *game* dengan inti cerita seorang pemuda yang menjelajahi dunia terkutuk untuk mengalahkan musuh-musuh yang berperilaku aneh. Judul *game "Cursed World"* diambil berdasarkan latar tempat dari cerita *game* ini yaitu dunia yang terkutuk. Dalam *game* ini, pemain akan mengontrol karakter utama untuk melawan musuh dan mengalahkan bos-bosnya untuk mencapai tujuan akhir. Salah satu elemen penting dalam *game* ini adalah karakter musuh yang cerdas dan adaptif, sehingga memberikan tantangan dan kepuasan dalam bermain. Oleh karena itu, penggunaan kecerdasan buatan (*artificial intelligence/AI*) pada karakter musuh dalam *game* *Metroidvania* telah menjadi solusi yang populer dalam meningkatkan kecerdasan dan kualitas permainan.

Metode-metode seperti *Finite State Machine* dalam menentukan pergerakan musuh di *game* edukasi COVID-19 [1], *Finite State Machine* juga digunakan untuk menentukan pergerakan di *game* *Rabbit vs Zombie* [2], *Behavior Tree* digunakan pada perilaku NPC pada *game* *Sidescroller* [3], dan *Hierarchical Finite State Machine* digunakan pada perilaku musuh *game* *hack and slash* [4].

Metode *Finite State Machine* menggunakan state yang bisa meng-handle tingkahlaku musuh yang kompleks, tetapi *game* hari ini membutuhkan banyak state dimana NPC di metode *FSM* hanya bisa berada di satu state saja [5]. Banyaknya state tersebut bisa membuat *FSM* sulit untuk menambah, menghapus atau mengubah state karena membutuhkan evaluasi ulang. Disamping itu, *Hierarchical Finite State Machine* bisa mengatasi masalah *FSM* dengan hierarki, tetapi masih belum cukup karena penggunaan logika yang berulang [3]. *HFSM* adalah versi hierarki dari *FSM* [6], tetapi pembuatan hierarki tersebut bisa sulit karena dibuat secara manual dan terkoneksi oleh subgraph lainnya.

Behavior Tree (BT) merupakan metode yang menggunakan modularitas yang setiap state membentuk cabang seperti pohon, sehingga bisa mengatasi kekurangan dari *Finite State Machine* yaitu state-nya tidak mudah digunakan kembali dan permasalahan ketika banyak state dan transisi [7]. Kelemahan *HFSM* bisa diatasi oleh *BT*, karena *BT* mudah untuk diubah, dihapus dan bisa digunakan kembali setiap kali node dibutuhkan.

Namun, dalam *game* *Metroidvania*, terdapat tantangan khusus dalam penerapan *BT* pada karakter musuh. Salah satu tantangan tersebut adalah kompleksitas perilaku musuh dalam desain *BT* yang harus disesuaikan dengan kondisi *game* *Metroidvania*. Karakter musuh dalam *game* *Metroidvania* harus dapat menyesuaikan perilakunya dengan lingkungan *game* yang berubah-ubah, serta harus mampu menyesuaikan diri dengan kemampuan dan taktik yang digunakan oleh pemain. Oleh karena itu, penelitian ini akan mengimplementasikan kecerdasan buatan menggunakan *BT* dalam menentukan perilaku musuh pada *game* *Metroidvania*. Tujuan dari penelitian ini adalah mendapatkan hasil dari penerapan *BT* dalam menangani perilaku pada karakter musuh *game* *Metroidvania*.

2. METODE PENELITIAN

2.1 Pengumpulan Data

Data diambil dari berbagai sumber yang bisa mendukung kebutuhan sistem yang akan dikembangkan. Sumber ini bisa berasal dari buku, jurnal, artikel, dan data dari *game* *metroidvania*. Metode survei juga digunakan untuk mendapatkan perspektif langsung dari pemain. Penulis akan melakukan wawancara dan kuesioner terstruktur dengan pemain yang telah berpartisipasi dalam sesi permainan. Pertanyaan akan difokuskan pada persepsi mereka terhadap

tingkat kesulitan, pengalaman keseluruhan, dan saran perbaikan. Survei ini akan memberikan wawasan yang lebih mendalam tentang preferensi pemain, tantangan yang mereka hadapi, serta aspek-aspek tertentu yang dapat ditingkatkan dalam permainan. Data yang terkumpul akan dianalisis secara keseluruhan untuk memperoleh wawasan tentang tingkat kepuasan dan kesulitan pemain.

2.2 Metode

Dalam penelitian ini akan menggunakan Behavior Tree pada semua musuh. Behavior Tree (BT) merupakan suatu metode terstruktur untuk mengalihkan tugas-tugas kepada agen otonom atau kecerdasan buatan seperti robot atau entitas virtual dalam *game* komputer (NPC) [6]. Dalam Behavior Tree, terdapat beberapa jenis node, yaitu *root node*, *control-flow node*, dan *execution node*. Hubungan antara node-node ini dieksekusi dari kiri ke kanan dan mengembalikan status *success*, *running*, *failure*.

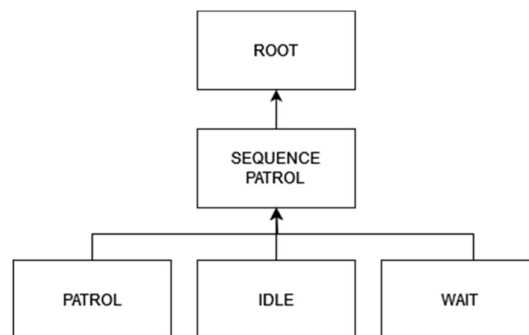
Adapun node-node Behavior Tree yang digunakan dijelaskan sebagai berikut.

1. *Root node*, merupakan induk dari setiap node dibawahnya.
2. *Control-flow node (sequence, selector, & decorator)*, merupakan node-node anak/child untuk mengontrol kembalian status dari setiap node.
 - a. *Sequence node* dijalankan dari kiri ke kanan dan setiap node anaknya harus mengembalikan status *success*, tetapi jika ada yang *failure*, maka node *sequence* akan gagal dijalankan dan dilanjutkan ke node selanjutnya.
 - b. *Selector node* dijalankan dengan memilih/menemukan node anak yang mengembalikan status *running/success* saja.
 - c. *Decorator* merupakan node untuk memanipulasi kembalian dari node anaknya, contohnya *inverter*, *repeat until fail*.
3. *Execution node*, merupakan node untuk mengeksekusi tugas yang ada dalam modul Behavior Tree, contohnya perilaku musuh.

Behavior Tree diterapkan pada musuh di *game* “Cursed World” memiliki tipe musuh berbeda dan perilaku yang bervariasi. Adapun struktur dan penjelasan beberapa musuh tersebut antara lain:

1. Musuh Lalat

Musuh ini akan terbang ke kanan dan ke kiri atau patrol dan *idle* selama beberapa detik. Musuh tipe ini akan meledak dan mati jika bersentuhan dengan pemain. Pemain akan menerima *damage* ketika dalam jangkauan ledakan. Struktur BT musuh lalat tertera pada gambar 1.

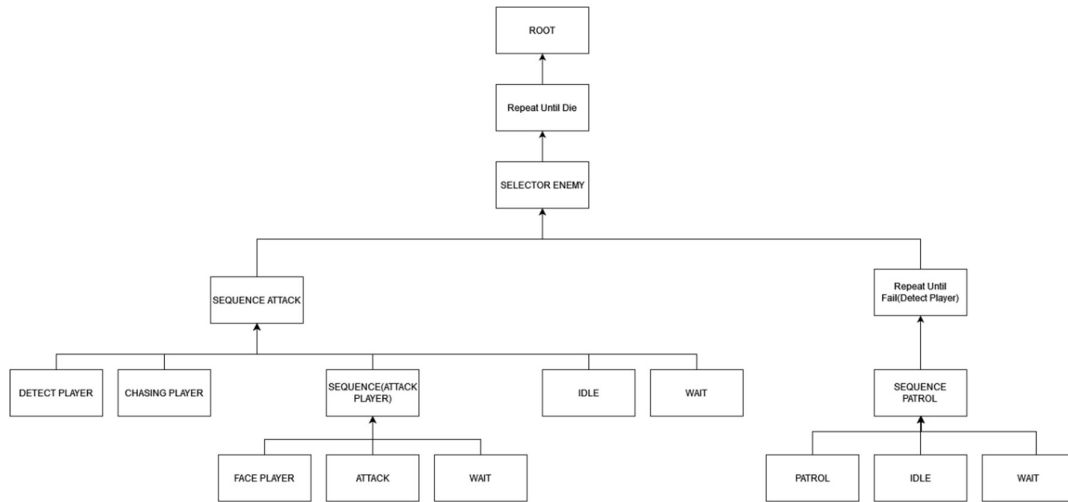


Gambar 1. Behavior Tree Musuh Lalat

2. Musuh Kesatria

Gambar 2 menggambarkan musuh tipe ini akan mengejar jika mendeteksi keberadaan pemain pada jarak 6 m. Jika pemain cukup dekat 1.5 m dengan musuh, maka musuh akan melakukan serangan *melee* dengan menghadap pemain dan menyerang, kemudian menunggu animasi serangan selesai dan jeda *idle* baru menyerang lagi. Ketika pemain kabur dari musuh atau hilang pandangan dari musuh, maka musuh akan masuk

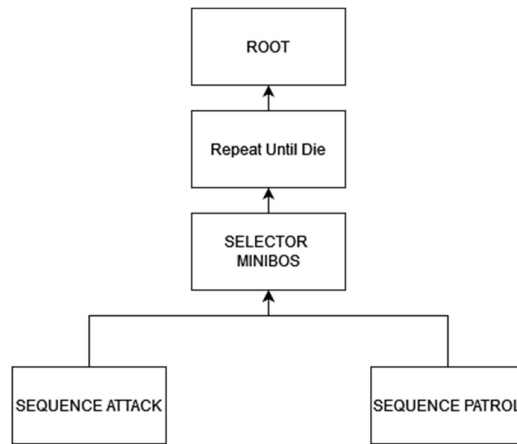
dalam perilaku patrol yang akan diulang sampai pemain terdeteksi kembali. Musuh akan melakukan perilaku tersebut jika darah masih ada dan jika darah musuh kurang dari sama dengan nol, maka musuh akan mati.



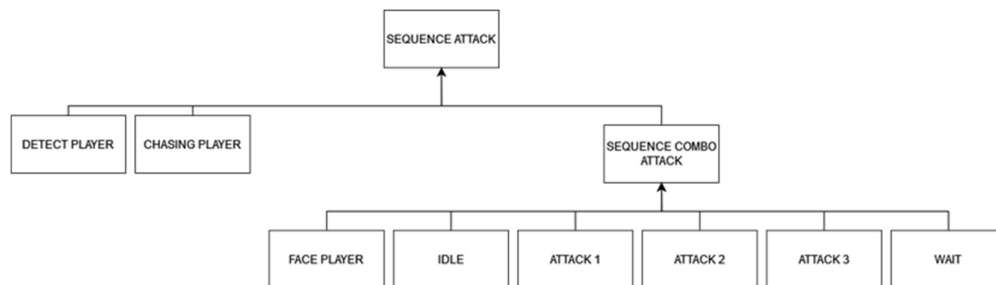
Gambar 2. Behavior Tree Musuh Kesatria

3. Musuh Miniboss

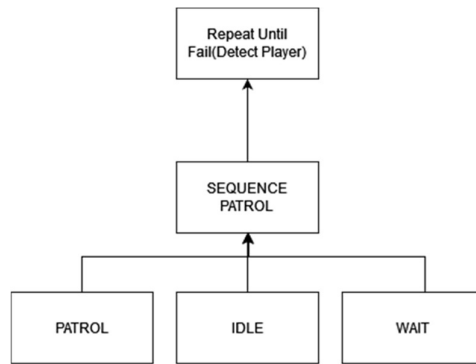
Musuh ini akan mengejar pemain jika terdeteksi di areanya. Memiliki perilaku yang mirip seperti musuh kesatria pada gambar 2. Namun, musuh minibos memiliki tiga serangan beruntun yang diarahkan ke pemain ketika berada 3 m didekatnya. Struktur BT musuh ini digambarkan pada gambar 3, 4, dan 5.



Gambar 3. Behavior Tree Musuh Miniboss



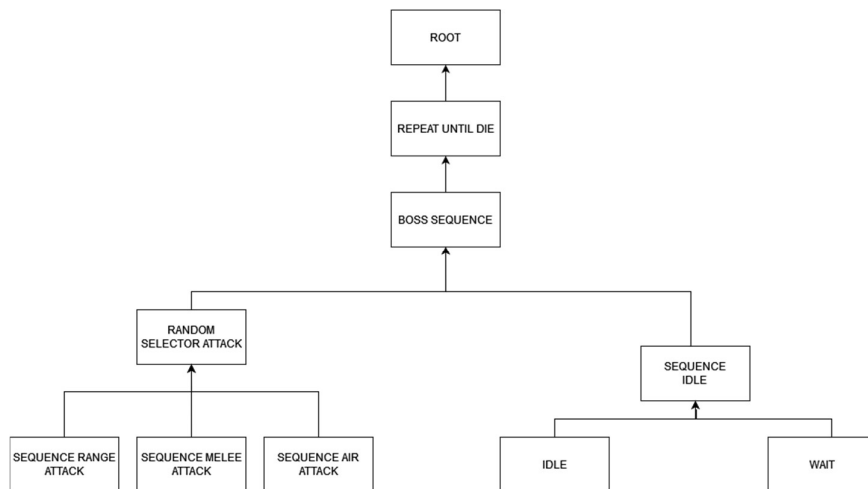
Gambar 4. Sequence Attack Musuh Miniboss



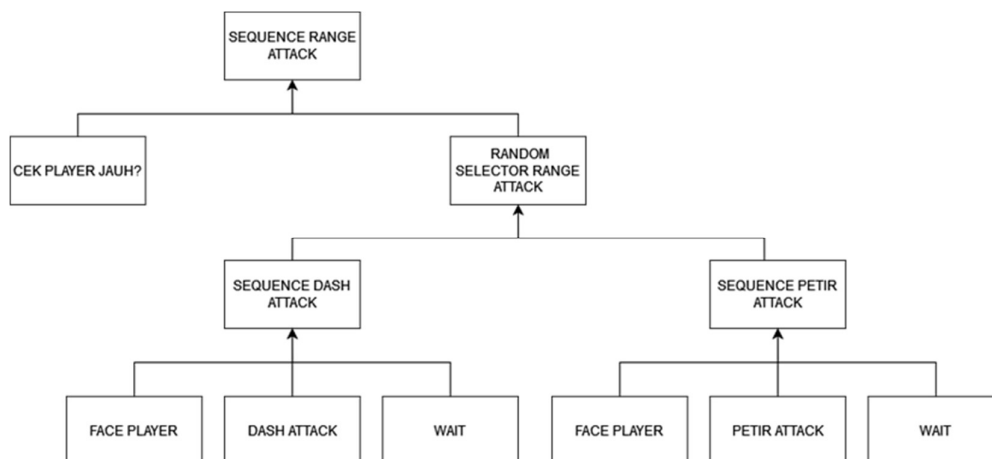
Gambar 5. Sequence Patrol Musuh Miniboss

4. Musuh Boss

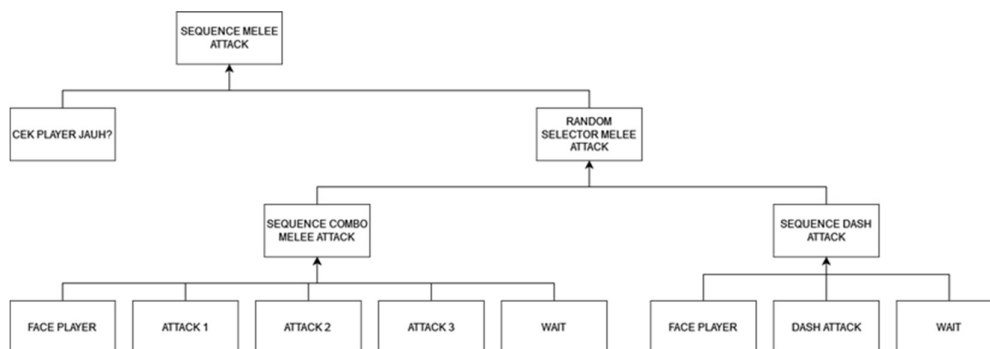
Gambar 6 adalah struktur musuh boss yang memiliki 3 tipe serangan. Ketika pemain masuk dalam ruangan, musuh akan mengecek apakah pemain lebih jauh 4 m dari musuh atau tidak. Jika ya atau sukses, maka musuh akan melakukan serangan jarak jauh (gambar 7). Terdapat dua serangan jarak jauh seperti, serangan dash yaitu musuh bergerak dengan kecepatan tinggi menuju arah pemain sejauh 8 m dan serangan petir yaitu musuh menyerang dengan menggunakan petir yang diarahkan ke pemain sejauh 8m. Musuh akan memilih secara acak serangan jarak jauh mana yang akan dilakukan. Jika pemain berada dekat kurang dari samadengan 4 m dari musuh, maka musuh akan melakukan kombo tiga serangan atau serangan dash (gambar 8). Musuh akan melakukan tipe serangan ketiga pada gambar 9 yaitu tipe serangan udara yang memiliki *cooldown* 10 detik, ketika *cooldown* 0 detik, musuh berpindah tempat menuju ke tengah atas area dan menghempaskan tubuhnya kebawah lalu menciptakan hempasan yang memberi serangan fatal pada pemain. Setiap perilaku serangan akan ada jeda waktu 1 detik untuk perilaku *idle*.



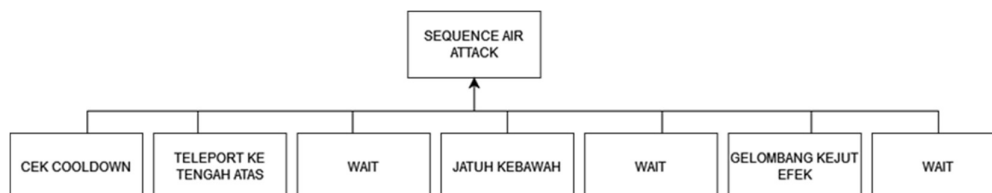
Gambar 6. Behavior Tree Musuh Boss



Gambar 7. Sequence Range Attack Musuh Boss



Gambar 8. Sequence Melee Attack Musuh Boss



Gambar 9. Sequence Air Attack Musuh Boss

2.3 Rencana Pengujian

Pengujian akan dilakukan dengan menggunakan pendekatan *Game Experience Questionnaire* (GEQ). GEQ merupakan kuesioner dalam gim yang dibuat untuk menilai pengalaman bermain pada beberapa interval selama sesi gim dimainkan [8]. Pengujian ini dilakukan dengan memberikan beberapa pertanyaan kepada pemain setelah mencoba atau memainkan gim ini. Pertanyaan yang diberikan berkaitan dengan topik pembahasan penelitian, seperti *gameplay* melawan musuh dan kesulitan pada kecerdasan buatan musuh.

3. HASIL DAN PEMBAHASAN

3.1 Musuh Lalat

Tabel 1 menunjukkan hasil dari penerapan Behavior Tree pada musuh lalat sesuai dengan struktur dan penjelasan Behavior Tree perilaku musuh lalat pada gambar 1.

Tabel 1. Hasil Implementasi BT Musuh Lalat

Perilaku	Pemain menyentuh/ darah ≤ 0	Jarak > 0
Patrol	Meledak/mati	patrol
Meledak/mati	Meledak/mati	patrol

3.2 Musuh Kesatria

Tabel 2 menunjukkan hasil dari penerapan Behavior Tree pada musuh kesatria sesuai dengan struktur dan penjelasan Behavior Tree perilaku musuh kesatria pada gambar 2.

Tabel 2. Hasil Implementasi BT Musuh Kesatria

Perilaku	Jarak ≤ 1.5	Jarak ≤ 6	Jarak > 6	Darah ≤ 0
Patrol	<i>attack</i>	<i>chasing</i>	patrol	mati
<i>Attack</i>	<i>attack</i>	<i>chasing</i>	patrol	mati
<i>Chasing</i>	<i>attack</i>	<i>chasing</i>	patrol	mati

3.3 Musuh Miniboss

Tabel 3 menunjukkan hasil dari penerapan Behavior Tree pada musuh miniboss sesuai dengan struktur dan penjelasan Behavior Tree perilaku musuh miniboss pada gambar 3, gambar 4, dan gambar 5.

Tabel 3. Hasil Implementasi BT Musuh Miniboss

Perilaku	Jarak ≤ 3	Pemain dalam area	Pemain luar Area	Darah ≤ 0
Patrol	<i>attack</i>	<i>chasing</i>	patrol	mati
<i>Combo attack</i>	<i>attack</i>	<i>chasing</i>	patrol	mati
<i>Chasing</i>	<i>attack</i>	<i>chasing</i>	patrol	mati

3.4 Musuh Boss

Tabel 4 menunjukkan hasil dari penerapan Behavior Tree pada musuh boss sesuai dengan struktur dan penjelasan Behavior Tree perilaku musuh boss pada gambar 6, gambar 7, gambar 8, dan gambar 9.

Tabel 4. Hasil Implementasi BT Musuh Boss

Perilaku	Jarak ≤ 4	Jarak > 4	<i>Cooldown = 0</i>	Darah ≤ 0
<i>Melee combo attack</i>	<i>Melee combo attack / Dash Attack</i>	<i>Petir Attack / Dash Attack</i>	<i>Air Attack</i>	mati
<i>Dash Attack</i>	<i>Melee combo attack / Dash Attack</i>	<i>Petir Attack / Dash Attack</i>	<i>Air Attack</i>	mati
<i>Petir Attack</i>	<i>Melee combo attack / Dash Attack</i>	<i>Petir Attack / Dash Attack</i>	<i>Air Attack</i>	mati
<i>Air Attack</i>	<i>Melee combo attack / Dash Attack</i>	<i>Petir Attack / Dash Attack</i>	<i>Air Attack</i>	mati

3.5 Hasil Pengujian GEQ

Tabel 5 menjelaskan nilai rata-rata dari 22 responden dengan total keseluruhan rata-rata GEQ bernilai 4.05. Dari hasil tersebut, maka bisa disimpulkan bahwa tantangan pada gim “*Cursed World*” berhasil membuat pemain merasa tertantang.

Adapun pertanyaan yang diberikan di dalam kuesioner pengalaman bermain *game* sebagai berikut.

1. Saya tertarik dengan *gameplay* yang diberikan.
2. Musuh selain boss sangat sulit dikalahkan.
3. Musuh Boss sulit untuk dikalahkan.
4. Saya puas dengan *gameplay* yang diberikan.
5. Saya merasa *game* ini sangat sulit.
6. Saya merasa *game* ini tidak seimbang.
7. Saya merasa tertantang untuk mengalahkan semua musuh di *game* ini.

Tabel 5. Hasil GEQ

Responden\Pertanyaan	1	2	3	4	5	6	7	rata-rata
1	5	1	5	5	4	4	5	4,14
2	3	3	5	3	4	5	3	3,71
3	4	5	5	4	4	3	4	4,14
4	3	5	5	2	4	3	3	3,57
5	5	2	5	5	3	2	5	3,86
6	5	5	5	5	5	5	5	5,00
7	5	3	5	5	4	2	5	4,14
8	5	3	5	5	4	2	4	4,00
9	5	5	5	5	5	4	5	4,86
10	4	3	4	3	3	4	5	3,71
11	5	3	5	5	4	3	5	4,29
12	4	4	5	4	5	3	3	4,00
13	5	3	3	4	3	1	5	3,43
14	5	3	5	4	2	3	5	3,86
15	5	5	5	5	5	3	5	4,71
16	1	1	5	1	5	5	3	3,00
17	4	3	4	3	3	3	5	3,57
18	5	5	5	5	5	5	5	5,00
19	5	4	3	5	3	2	5	3,86
20	4	3	4	5	3	2	5	3,71
21	4	3	3	4	4	3	4	3,57
22	5	5	5	5	5	5	5	5,00
rata-rata GEQ								4,05

4. KESIMPULAN DAN PENELITIAN SELANJUTNYA

4.1 Kesimpulan

Kesimpulan yang diperoleh dari penelitian ini antara lain:

1. Penggunaan Behavior Tree untuk mengatur perilaku musuh pada *game* *Metroidvania*

ini menghasilkan struktur pemodelan yang mudah dipahami. Sifat modularitas sangat membantu dalam penyusunan perilaku pada musuh.

2. Penggunaan Behavior Tree membuat perilaku musuh menjadi mudah didetailkan untuk setiap perilakunya.
3. Penggunaan Behavior Tree membuat pemain menjadi tertantang untuk mengalahkan musuh dan membuat tingkat kesulitan pada musuh menjadi tinggi.
4. Pengujian GEQ dihasilkan rata-rata 4.05 dari keseluruhan responden, sehingga dari hasil itu menjadikan gim “*Cursed World*” memiliki tantangan yang baik bagi pemain.

4.2 Penelitian Selanjutnya

Dalam penelitian ini masih terdapat kekurangan yang dapat dikembangkan lagi pada penelitian selanjutnya. Adapun saran sebagai pertimbangan untuk diterapkan pada penelitian selanjutnya antara lain:

1. Metode ini bisa dikombinasikan dengan metode lainnya seperti metode Fuzzy untuk menambah tingkat kesulitan dan kondisi yang semakin kompleks lagi.
2. Menambahkan komponen dekorator lain di Behavior Tree seperti Repeat Until Success, Return Success, Return Failure dan lainnya.
3. Menambahkan banyak perilaku dan animasi pada musuh.
4. Menambahkan jumlah musuh yang bervariasi.

DAFTAR PUSTAKA

- [1] D. E. Wijaya, K. Auliasari, and H. Z. Zahro’, “KOMBINASI METODE METODE FINITE STATE MACHINE DAN GAME-BASED LEARNING PADA GAME ‘ESCAPE FROM COV-MADNESS,’” *J. Mhs. Tek. Inform.*, vol. 5, no. 1, 2021.
- [2] A. Solihin, E. W. Hidayat, and A. P. Aldya, “Application of the Finite State Machine Algorithm on 2D Platformer Rabbit Games vs Zombies,” *J. Online Inform.*, vol. 4, no. 1, p. 33, Sep. 2019, doi: 10.15575/join.v4i1.293.
- [3] A. Junaidi, A. Yunus, and A. S. Wiguna, “IMPLEMENTASI BEHAVIOR TREE PADA PERILAKU NPC DI GAME SIDESCROLLER,” *Kurawal - J. Teknol. Inf. Dan Ind.*, vol. 4, no. 2, pp. 92–103, Oct. 2021, doi: 10.33479/kurawal.v4i2.459.
- [4] A. F. Naharu, E. M. A. Jonemaro, and M. A. Akbar, “Penerapan Hierarchical Finite State Machine untuk Pengambilan Keputusan Non-Player Character (Studi Kasus: Gim Hack and Slash),” *J. Pengemb. Teknol. Inf. Dan Ilmu Komput.*, vol. 5, Mar. 2021.
- [5] Y. A. Sekhavat, “Behavior Trees for Computer Games,” *Int. J. Artif. Intell. Tools*, vol. 26, no. 02, p. 1730001, Apr. 2017, doi: 10.1142/S0218213017300010.
- [6] M. Colledanchise and P. Ögren, *Behavior Trees in Robotics and AI*, 0 ed. CRC Press, 2018. doi: 10.1201/9780429489105.
- [7] E. M. Trianto, H. Junaedi, and H. Sutiksno, “AGEN CERDAS BERBASIS CONTROLLER FUZZY PADA PERMAINAN STRATEGI PERTEMPURAN DENGAN BEHAVIOR TREE,” *Semin. Nas. Ilmu Terap.*, vol. 1, no. 1, Art. no. 1, Aug. 2017.
- [8] W. A. IJsselsteijn, Y. A. De Kort, and K. Poels, “The game experience questionnaire,” *Tech. Univ. Eindh.*, 2013, Accessed: Dec. 27, 2023. [Online]. Available: <https://research.tue.nl/en/publications/the-game-experience-questionnaire>