

Penerapan *Finite State Machine* dan *Atreus AI Behavior* pada *AI Musuh* dalam *Fighting Game*

Jong Jeffrey Wicaksono, Djoni Haryadi Setiabudi, Hans Juwiantho
Program Studi Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra
Jl. Siwalankerto 121-131 Surabaya 60236

Telp. (031) – 2983455, Fax. (031) - 8417658

jeffreyjong11@gmail.com, djonih@petra.ac.id, hans.juwiantho@petra.ac.id

ABSTRAK

Artificial Intelligence (AI) sudah menjadi bagian penting dalam pengembangan game. AI pada game memiliki tugas yang beragam dalam memberikan pengalaman langsung kepada pemain. Kekurangan ragam strategi pada AI musuh dalam fighting game bisa saja membuat motivasi pemain menjadi berkurang ketika bermain. Untuk menyelesaikan permasalahan tersebut, dibuatlah AI yang memiliki ragam strategi bekerjasama yang dapat menarik motivasi pemain.

Game akan dikembangkan menggunakan aplikasi Unity3D dan menggunakan bahasa pemrograman C#. Metode Finite State Machine digunakan untuk mengembangkan AI yang menjadi musuh dari pemain dan diberikan Atreus AI Behavior agar dapat memiliki ragam strategi. Probability juga ditambahkan pada pengambilan keputusan AI agar menjadi lebih tidak mudah ditebak.

Hasil pengujian menunjukkan bahwa AI dapat berjalan dengan baik menggunakan penggabungan FSM dan behavior Atreus. Pengujian juga dilakukan kepada 15 pemain yang memiliki background pemain casual atau di atasnya. Lima belas pemain ini bertugas untuk mencoba permainan. Hasilnya pengujian tersebut menunjukkan bahwa AI menjadi lebih sulit untuk dikalahkan dan memberikan motivasi kepada pemain untuk mengalahkannya.

Kata Kunci: *Finite State Machine, Atreus AI Behavior, Strategi, Probability, C#, Unity 3D, Fighting Game.*

ABSTRACT

Artificial Intelligence (AI) has already become an important part in game development. AI in game have a diverse job to give a direct experience to player. The lack of strategy in enemy AI inside fighting game can make players less motivated when playing. To solve this problem, AI is made which has variety of cooperate strategy that can make attract player's motivation

Game is developed using Unity3D Engine and using C# programming language. Finite State Machine method is used to develop AI which become player enemy and Atreus AI Behavior is given in order to have variety of strategies. Probability also added in AI decision making in order to make the AI less predictable.

The results show that AI can run well using FSM and Atreus Behavior merging. Testing is also carried out on 15 players which have a background as a casual player or above. These 15 players are in charge to try the game. The results show that AI is more difficult to beaten and player is motivated to defeat the AI.

Keywords: *Finite State Machine, Atreus AI Behavior, Strategy, Probability, C#, Unity 3D, Fighting Game.*

1. PENDAHULUAN

Artificial Intelligence (AI) sudah menjadi bagian penting dalam pengembangan game. AI dalam game biasanya bertugas untuk menyelesaikan suatu masalah di dalam game tersebut, seperti mengalahkan musuh [3]. Oleh karena itu *gameplay* menjadi unik dan menarik ketika AI diimplementasikan [2].

AI pada game memiliki tugas yang beragam dalam memberikan pengalaman langsung kepada pemain. Dalam *fighting games*, pemain akan lebih banyak berinteraksi dengan AI yang menjadi musuh dari pemain. Musuh tersebut biasanya memiliki beragam tingkat kesulitan agar memberikan pengalaman bermain yang menarik bagi pemain. Kekurangan yang ada pada ragam tingkat kesulitan tersebut yaitu kurangnya strategi dalam menghadapi pemain, dimana AI biasanya hanya menerjang pemain langsung. Strategi yang kurang beragam tersebut bisa saja membuat motivasi pemain menjadi berkurang. Oleh karena itu, pemain membutuhkan tantangan dalam bermain game agar menjadi motivasi pemain untuk melanjutkan permainannya [1].

Penelitian dengan menggunakan metode FSM pada game pernah dilakukan oleh Svensson pada tahun 2015. Penelitian ini menghasilkan AI yang dapat memilih pilihan yang tepat sesuai dengan strategi yang diberikan. Penelitian tersebut menghasilkan strategi AI yang statis, peneliti juga menyarankan agar AI dapat melakukan *information gathering* dari musuhnya atau lingkungannya agar menghasilkan AI yang lebih unik. Oleh karena itu penelitian ini memiliki perbedaan dari konsep AI yang diterapkan dimana penelitian ini akan menerapkan AI yang dapat melakukan *information gathering* dari musuhnya.

Atreus AI Behavior merupakan sebuah *behavior* dari karakter bernama *Atreus* pada game *God of War*. Pada game tersebut, *Atreus* berperan untuk membantu pemain dalam mengalahkan musuh yang ada. Cara *Atreus* membantu pemain cukup beragam, diantaranya yaitu membantu menyerang, menjatuhkan, dan menahan musuh [6]. Oleh karena itu, penelitian ini akan menggunakan *behavior* dari karakter *Atreus* agar dapat membuat musuh yang unik dan menarik.

Pada penelitian ini, metode FSM akan diterapkan dengan *behavior Atreus* pada AI musuh sehingga akan menghasilkan variasi strategi yang beragam ketika menghadapi pemain. AI juga dapat melakukan *information gathering* sederhana dari musuhnya sehingga memberikan kesan mempelajari pemain. Dengan diterapkannya FSM dan *Behavior AI Atreus*, harapannya AI akan lebih sulit dikalahkan dengan normal tetapi tetap bisa dikalahkan

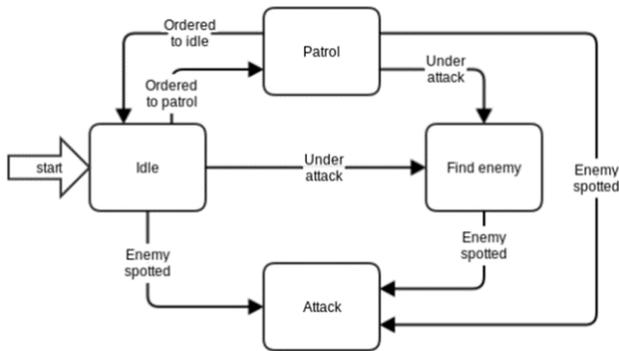
dengan strategi yang baik. Penggabungan FSM dan *Behavior AI Atreus* diharapkan juga akan memberikan nuansa yang baru dan memberikan keseruan dalam bermain *fighting games*.

2. TINJAUAN PUSTAKA

2.1 Finite State Machine

FSM digunakan untuk membuat dan menunjukkan logika dari suatu *action*. Menggunakan *State Pattern*, representasi FSM dapat dikembangkan hampir diseluruh bidang aplikasi IT, mulai dari *business simulations* hingga *game* komputer [5].

FSM terdiri dari *finite set of states* dimana hanya satu state yang dapat aktif dalam satu waktu. Biasanya satu state ditandai sebagai state awal sebagai tanda awal sistem berjalan. Untuk bisa pindah state, dapat dilakukan transisi setelah kondisi tertentu terpenuhi [4]. Contoh FSM dari *AI* musuh dapat dilihat pada Gambar 1.



Gambar 1. Contoh model *finite state machine*

Pada Gambar 1, states digambarkan dalam bentuk kotak yang memiliki suatu fungsi atau tugas. Ketika *AI* sedang melakukan idle, *AI* dapat melakukan fungsi Attack apabila kondisi tertentu terpenuhi. Kondisi tersebut digambarkan berupa garis yang mengandung suatu kalimat.

2.2 Atreus AI Behavior

Atreus adalah anak dari *Kratos* pada *game God of War*. *Atreus* berperan membantu pemain seperti seorang rekan tim, dan tindakannya pun sangat membantu pemain dalam *game* tersebut. Penelitian ini akan menerapkan *behavior* dari *Atreus* pada *AI* musuh sehingga pada penelitian ini, *AI* akan bekerja sama mengalahkan pemain menggunakan bantuan-bantuan *Atreus* pada *game God of War*.

Bantuan yang diberikan ada beragam, tetapi yang akan dipakai pada penelitian ini ada tiga yaitu *beat downed enemy*, *hold the enemy*, dan *tackle the enemy* [7].

3. DESAIN SISTEM

3.1 Alur Sistem Game

Secara umum, *game* dapat dibagi menjadi beberapa halaman, halaman tersebut memiliki alur yang dapat digambarkan dengan diagram Gambar 2.



Gambar 2. Alur sistem permainan

- *Main Menu*

Pada bagian *main menu*, pemain dapat memilih menu *Start* untuk dapat memulai permainan.

- *Tutorial Gameplay*

Pada bagian *tutorial*, pemain dapat mempelajari tombol-tombol yang diperlukan untuk dapat mengontrol karakter dari pemain.

- *Gameplay*

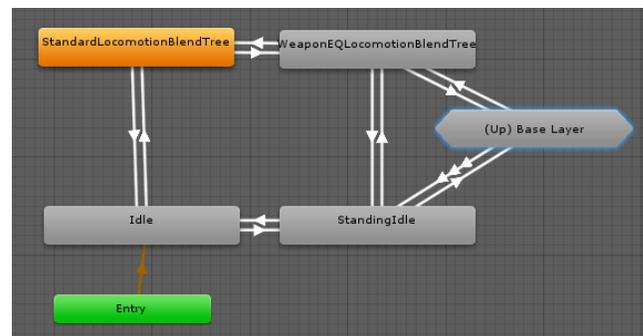
Pada bagian *gameplay*, pemain akan memainkan bagian utama dari *game* dan berusaha untuk mengalahkan musuh agar dapat menyelesaikan permainan.

- *Result*

Halaman result akan menampilkan hasil permainan dimana pemain dinyatakan menang atau kalah dan pemain akan diarahkan ke bagian main menu.

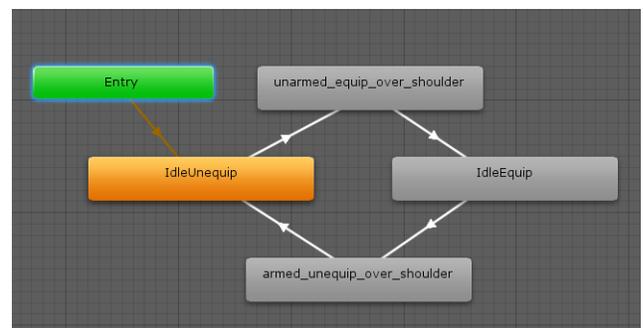
3.2 Desain Animasi

Desain animasi pada karakter milik pemain dan musuh *AI* memiliki desain yang sama. Desain animasi tersebut meliputi animasi *movement*, *movement* dengan senjata, memegang dan menyimpan senjata, *block* dan *attack*. Desain tersebut dapat dilihat pada Gambar 3, Gambar 4, Gambar 5, dan Gambar 6.



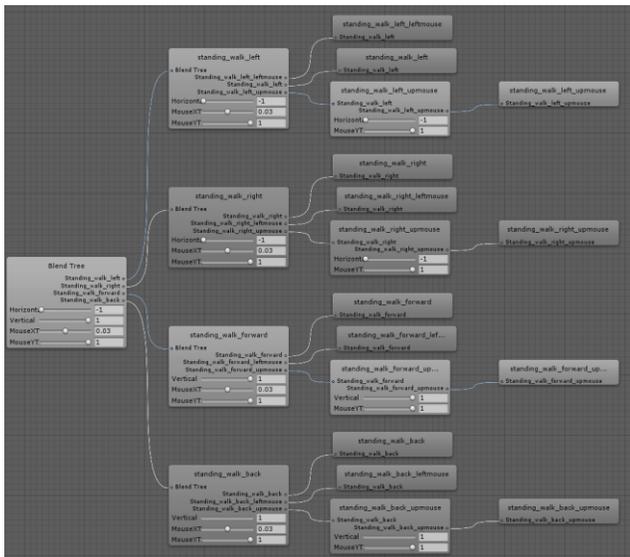
Gambar 3. Desain animasi *movement* tanpa senjata

Desain Gambar 3 untuk *movement* tanpa senjata cukup sederhana dimana karakter berpindah dari *state idle* yang berguna untuk memainkan animasi ketika pemain diam ke *state* berwarna jingga untuk memainkan animasi *movement*.



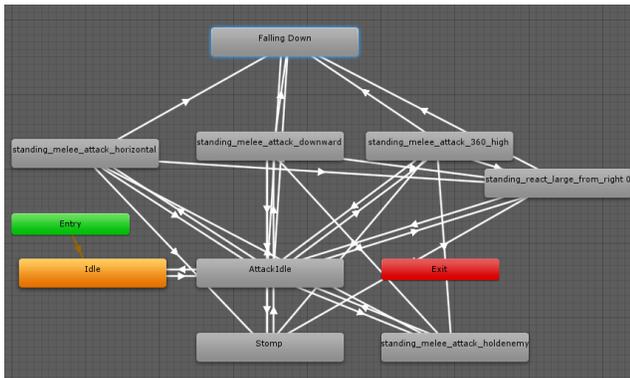
Gambar 4. Desain animasi memegang dan menyimpan senjata

Desain animasi Gambar 4 berguna untuk memanggil animasi yang berfungsi untuk memegang senjata pada tangan kanan karakter pemain dan karakter *AI*. Desain ini juga berguna untuk memanggil animasi ketika karakter pemain menyimpan kembali senjatanya.



Gambar 5. Desain animasi *movement* dengan senjata dan *block*

Desain animasi pada Gambar 5 digunakan oleh karakter pemain dan AI ketika berjalan dengan menggunakan senjata. Desain tersebut juga berguna untuk memanggil animasi *block* ke berbagai arah ketika pemain dan AI melakukan *block* sambil berjalan.

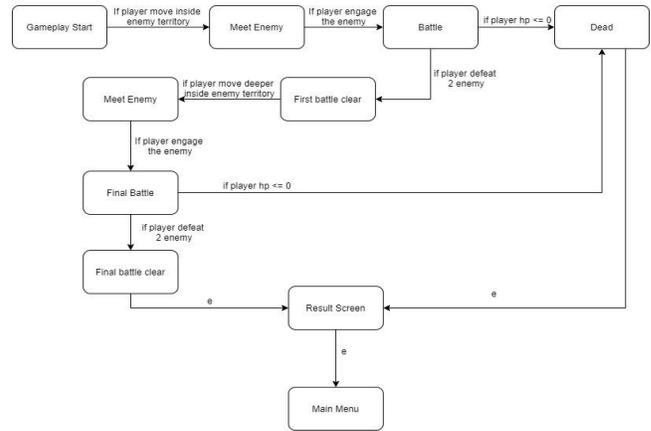


Gambar 6. Desain animasi *attack*

Desain animasi pada Gambar 6 berguna untuk memanggil animasi *attack* dan animasi menerima serangan dari karakter pemain dan karakter AI. Desain ini juga berguna untuk memanggil animasi khusus milik karakter pemain yaitu animasi terjatuh, dan juga animasi khusus AI yaitu *stomp* dan *hold enemy*.

3.3 Gameplay

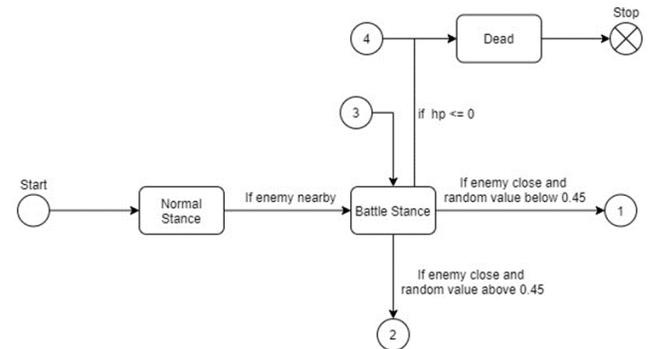
Saat memulai permainan *player* akan muncul atau *spawn* di depan benteng kerajaan musuh. Ketika gerbang benteng dibuka *player* akan menghadapi dua prajurit biasa musuh, disini *player* harus dapat membuktikan kemampuannya dengan mengalahkan dua musuh tersebut agar dapat menghadapi petinggi prajurit musuh. Setelah *player* mengalahkan dua prajurit musuh, dua petinggi prajurit musuh akan masuk ke dalam pertempuran dan bekerjasama mengalahkan *player*. *Game* akan berakhir dengan dikalahkannya petinggi prajurit musuh. Gambar 7 merupakan alur pada *gameplay*.



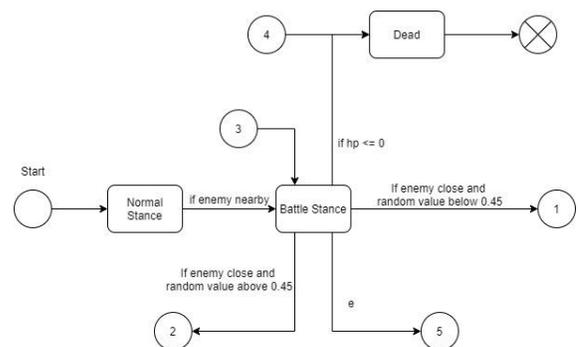
Gambar 7. Diagram alur *gameplay*

3.4 Desain FSM AI

Empat musuh yang akan dihadapi oleh pemain dalam *game* ini dibagi menjadi dua jenis musuh, yaitu *basic enemy* dan *advanced enemy*. Pada awal *gameplay* pemain akan menghadapi dua musuh yang berupa *basic enemy* dan dua musuh yang berupa *advanced enemy* pada akhir *game*. Pengaturan pergerakan dari *basic enemy* dan *advanced enemy* akan dibuat menggunakan *Finite State Machine*. Desain dasar FSM kedua jenis AI dapat dilihat pada Gambar 8 dan Gambar 9. Untuk desain *fsm* kedua jenis AI dapat dilihat pada Gambar 10 dan Gambar 11. *Advanced enemy* memiliki desain *fsm* tambahan yang berisikan *behavior* dari *Atreus*, desain *fsm* tersebut diberi nama *special move* dan dapat dilihat pada Gambar 12.

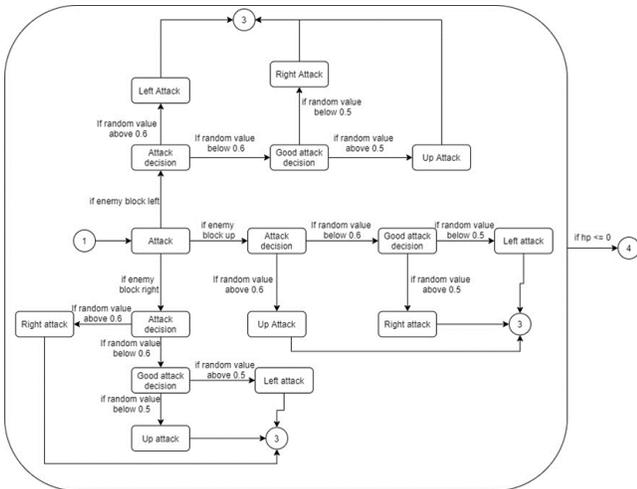


Gambar 8. Desain *fsm battle stance* dari *basic enemy*



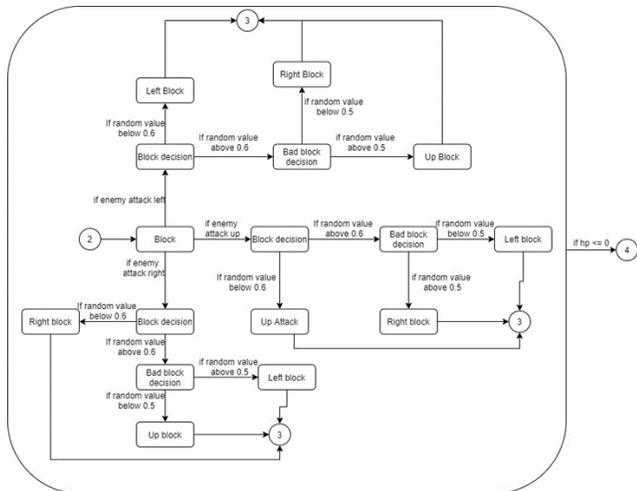
Gambar 9. Desain *fsm battle stance* dari *advanced enemy*

Perbedaan dari kedua desain *fsm battle stance* diatas yaitu ada tambahan pada desain *fsm* milik *advanced enemy*. Desain *fsm advanced enemy* memiliki sebuah kondisi tambahan menuju nomor 5. Nomor 5 tersebut mengarah ke desain *fsm* tambahan milik *advanced enemy* yaitu desain *fsm special move* yang merupakan *behavior* dari *Atreus*. Desain tersebut dapat dilihat pada Gambar 9.



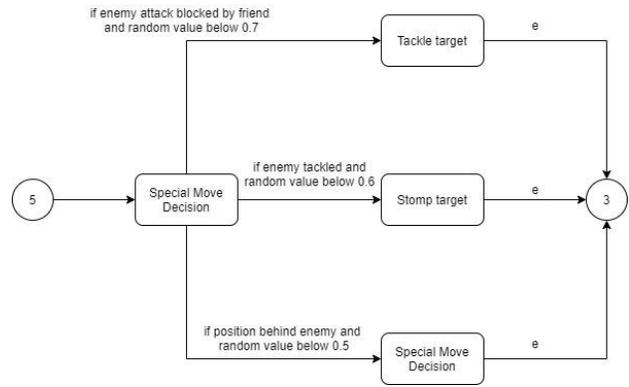
Gambar 10. Desain *fsm attack* dari *basic enemy* dan *advanced enemy*

Desain *FSM* pada Gambar 10 menunjukkan bagaimana proses pengambilan keputusan oleh *AI* untuk memilih arah kemana *AI* melakukan *attack*. Pemilihan tersebut memiliki kemungkinan 60% untuk melakukan *attack* ke dua arah yang tidak di *block* oleh pemain dan 40% untuk *attack* ke arah yang di *block* oleh pemain.



Gambar 11. Desain *fsm block* dari *basic enemy* dan *advanced enemy*

Desain *FSM* pada Gambar 11 berguna untuk pengambilan keputusan arah *block* dari *AI*. Pemilihan tersebut memiliki kemungkinan 60% untuk melakukan *block* ke arah yang di serang oleh pemain dan kemungkinan 40% untuk melakukan *block* ke dua arah yang tidak diserang oleh pemain.



Gambar 12. Desain *fsm special move* dari *advanced enemy*

4. PENGUJIAN SISTEM

4.1 Pengujian fitur *AI*

Bagian ini akan menguji apakah fitur-fitur yang dimiliki oleh *AI* dapat berjalan dengan baik dan benar. Fitur-fitur yang dimiliki oleh *basic enemy* merupakan fitur dasar yang juga akan dimiliki oleh *advanced enemy*. Fitur dasar tersebut meliputi *movement*, *battle stance*, *block*, dan *attack*. Fitur tambahan milik *advanced enemy* yang tidak dimiliki oleh *basic enemy* yaitu *tackle*, *stomp*, dan *hold enemy*. Fitur-fitur tersebut dapat dilihat pada Gambar 13, Gambar 14, Gambar 15, Gambar 16, Gambar 17, dan Gambar 18.



Gambar 13. *AI* berada dalam kondisi *battle stance* dan bergerak mendekati pemain

Gambar 13 memperlihatkan *AI* yang bergerak maju dalam kondisi *battle stance*. *AI* akan terus mengikuti pemain apabila pemain sudah pernah memasuki radius penjagaannya.



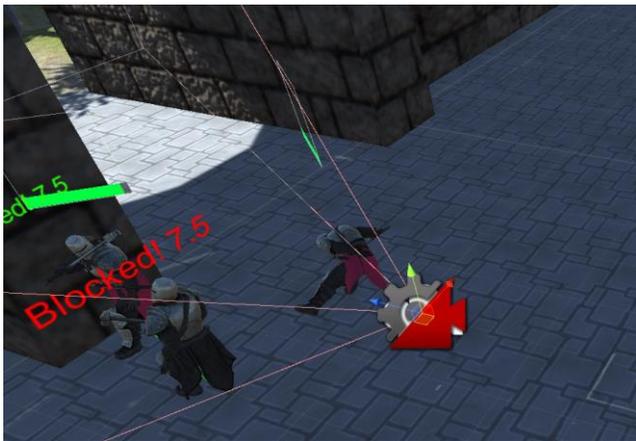
Gambar 14. Serangan pemain yang berhasil di *block AI*

Gambar 14 memperlihatkan bagaimana AI dapat dengan berhasil melakukan *block* dari serangan pemain. Apabila serangan berhasil di *block*, *damage* yang diterima dibagi dua.



Gambar 15. Attack dari AI yang tidak berhasil di *block* pemain

Gambar 15 memperlihatkan bagaimana AI berhasil melakukan *attack* dengan sempurna terhadap pemain. *Attack* yang tidak berhasil di *block* akan memberikan *damage* penuh yaitu 15.



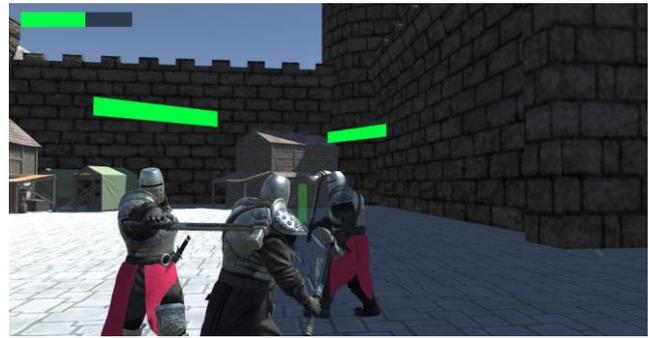
Gambar 16. Salah satu AI melakukan *tackle*

Gambar 16 merupakan fitur dari *advanced enemy* dimana AI dapat melakukan *tackle*. *Tackle* dapat dilakukan oleh AI apabila teman dari AI tersebut berhasil melakukan *block* dari serangan pemain.



Gambar 17. AI melakukan *stomp*

Gambar 17 memperlihatkan fitur dari *advanced enemy* dimana AI dapat melakukan *stomp*. *Stomp* dapat dilakukan oleh AI apabila teman dari AI tersebut berhasil melakukan *tackle*. *Stomp* memberikan *damage* sebesar 7.5.



Gambar 18. AI dikiri melakukan *hold enemy*

Gambar 18 juga memperlihatkan fitur dari *advanced enemy* dimana AI (yang di posisi kiri) yang tidak berada di *field of view* dari karakter milik pemain dapat melakukan *hold enemy*. *Hold enemy* berfungsi untuk menahan *movement* dari pemain selama beberapa detik.

4.2 Pengujian *Gameplay*

Bagian ini akan menguji *gameplay* dari *game* apakah *game* sudah cukup *balance* untuk dimainkan. Tujuannya agar pemain yang memainkan *game* ini akan cukup kesulitan menyelesaikan *game*, tetapi tetap bisa menyelesaikannya agar menjadi tantangan yang menarik dan tidak kehilangan motivasi bermain. Pengujian didapatkan dari jumlah mengulang bermain untuk mengalahkan musuh. Tujuan dari mengulang bermain agar mengetahui perbedaan kesulitan ketika menghadapi *basic enemy* dan *advanced enemy*. Semakin banyak jumlah mengulang, menandakan bahwa musuh sudah cukup sulit untuk dikalahkan. Hasil mencoba *game* dapat dilihat pada Tabel 2.

Tabel 2. Jumlah mengulang melawan AI

Jenis AI	Jumlah Mengulang
<i>Basic Enemy</i>	1
<i>Advanced Enemy</i>	4

Dari data yang didapat pada Tabel 2, *basic enemy* sudah memberikan tantangan sebagaimana yang cukup pada pemain. Tetapi pada *advanced enemy*, jumlah mengulang sebanyak empat kali terlalu banyak. Untuk memberikan *balance* pada permainan, fitur *power up* diterapkan agar pemain mendapatkan keringanan sedikit untuk menyelesaikan *game*. *Game* dicoba kembali dengan fitur *power up* dan menghasilkan data sesuai pada Tabel 3.

Tabel 3. Jumlah mengulang melawan AI menggunakan fitur *power up*

Jenis AI	Jumlah Mengulang
<i>Basic Enemy</i>	0
<i>Advanced Enemy</i>	2

Sesuai dengan Tabel 3, jumlah mengulang sebanyak dua kali ketika menghadapi *advanced enemy* dirasa sudah cukup memberikan tantangan kepada pemain dan memberikan motivasi untuk mengalahkan musuh ketika pemain kalah.

4.3 Pengujian *Gameplay* oleh Audiens

Game yang sudah jadi dibuat disebar ke 15 pemain dengan memiliki *background* kriteria berjenis kelamin laki-laki, berusia dibawah 25 tahun, dan setidaknya seorang pemain *game casual* (kadang-kadang bermain).

Setelah mencoba *game*, pemain akan diminta untuk mengisi kuesioner terkait berapa kali pemain mengulang menghadapi AI, seberapa sulit melawan AI, apakah AI memberikan variasi strategi, dan seberapa termotivasi pemain untuk mengalahkan AI. Tabel 4 menunjukkan hasil bermain dari pengisi kuesioner dan Tabel 5 menunjukkan seberapa sulit pemain mengalahkan AI.

Tabel 4. Jumlah mengulang pengisi kuesioner ketika melawan AI

Jenis AI	Jumlah Mengulang			
	Tidak Mengulang	1 - 2 kali	3 - 4 kali	>5 kali
<i>Basic Enemy</i>	2	6	1	1
<i>Advanced Enemy</i>	0	3	3	4

Tabel 5. Seberapa sulit pemain mengalahkan AI

Jenis AI	Seberapa Sulit				
	Sangat Mudah	Cukup Mudah	Biasa Saja	Cukup Kesulitan	Sangat Kesulitan
<i>Basic Enemy</i>	0	4	2	9	0
<i>Advanced Enemy</i>	0	0	0	4	11

Hasil dari Tabel 4 menunjukkan bahwa jumlah mengulang bermain *game* diatas lima kali ada sebanyak 46.6% audiens. Tabel 5 menunjukkan bahwa 73.3% audiens merasa sangat kesulitan dan 26.6% audiens merasa cukup kesulitan. Jumlah audiens yang merasa sangat kesulitan masih terbilang banyak sehingga fitur *power up* diterapkan pada pemain. Ketika pemain diberikan fitur *power up*, jumlah mengulang dan tantangan kesulitannya dapat diturunkan ke angka yang lebih baik. Hasilnya dapat dilihat pada Tabel 6 dan Tabel 7 dibawah ini.

Tabel 6. Jumlah mengulang pengisi kuesioner menggunakan fitur *power up*

Jenis AI	Jumlah Mengulang			
	Tidak Mengulang	1 - 2 kali	3 - 4 kali	>5 kali
<i>Advanced Enemy</i>	3	6	5	1

Tabel 7. Seberapa sulit mengalahkan AI dengan fitur *power up*

Jenis AI	Seberapa Sulit				
	Sangat Mudah	Cukup Mudah	Biasa Saja	Cukup Kesulitan	Sangat Kesulitan
<i>Advanced Enemy</i>	0	0	4	10	1

Hasil dari Tabel 6 menunjukkan 93.3 % audiens tidak mengulang permainan hingga lima kali. Hasil dari Tabel 7 juga menunjukkan bahwa 66.6% audiens merasa cukup kesulitan dan 6.6% merasa sangat kesulitan. Pemain menjadi lebih termotivasi melawan *Advanced Enemy* dimana pemain tidak perlu mengulang sangat banyak, tetapi tetap bisa mengalahkan dan mendapatkan pengalaman yang menarik ketika melawan *advanced enemy*. Hal tersebut ditunjukkan pada Tabel 8 dimana 86.6% dari keseluruhan audiens merasa cukup dan sangat termotivasi mengalahkan AI.

Tabel 8. Jumlah pemain yang termotivasi untuk mengalahkan *advanced enemy*

Jenis AI	Seberapa Termotivasi				
	Tidak	Kurang	Biasa	Cukup	Sangat
<i>Advanced Enemy</i>	0	1	1	8	5

Selain itu, audiens juga menjawab pertanyaan apakah *advanced enemy* dapat memberikan variasi strategi yang lebih beragam daripada *basic enemy*. Hasil pada Tabel 9 menunjukkan 100% audiens mengatakan setuju dan sangat setuju bahwa *advanced enemy* dapat memberikan variasi strategi ketika menghadapi pemain.

Tabel 9. Jumlah pemain yang setuju bahwa *advanced enemy* memberikan variasi strategi

Jenis AI	Apakah Memberikan Variasi Strategi				
	Sangat Tidak	Tidak Setuju	Netral	Setuju	Sangat Setuju
<i>Advanced Enemy</i>	0	0	0	7	8

5. KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil pengamatan dan pengujian pada *game*, dapat disimpulkan beberapa hal sebagai berikut:

- Berdasarkan hasil pengujian pada *game*, dapat disimpulkan bahwa *advanced enemy* bisa menjalankan keseluruhan strategi dari *Atreus AI Behavior*.
- Berdasarkan hasil pengujian dapat disimpulkan bahwa *advanced enemy* dapat membuat 66.6% audiens merasa cukup kesulitan melawan AI tetapi masih dapat mengalahkan AI.

3. Berdasarkan hasil pengujian dapat disimpulkan penggabungan *FSM* dan *Atreus AI Behavior* membuat 86.6% audiens lebih termotivasi untuk mengalahkan *advanced enemy*.
4. Berdasarkan hasil pengujian dapat disimpulkan 100% audiens setuju bahwa *advanced enemy* dapat memberikan variasi strategi ketika menghadapi pemain.

6. DAFTAR PUSTAKA

- [1] Brandse, M., & Tomimatsu, K. 2014. Challenge Design and Categorization in Video Game Design. *Design, User Experience, and Usability. User Experience Design for Diverse Interaction Platforms and Environments*, 669–677. https://doi.org/10.1007/978-3-319-07626-3_63
- [2] Kopel, M., & Hajas, T. 2018. Implementing AI for Non-player Characters in 3D Video Games. *Intelligent Information and Database Systems*, 610–619. https://doi.org/10.1007/978-3-319-75417-8_57
- [3] Safadi, F., Fonteneau, R., & Ernst, D. 2015. Artificial Intelligence in Video Games: Towards a Unified Framework. *International Journal of Computer Games Technology*, 2015. <https://doi.org/10.1155/2015/271296>
- [4] Svensson, M. 2015. *Dynamic Strategy in Real-Time Strategy Games: with the use of finite-state machines*. Retrieved from <http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A784853&dswid=4421>
- [5] Utesch, M. C., Hauer, A., Heining, R., & Krcmar, H. 2017. Automated Stock Trading – Developing the Serious Game FSTG to Teach the Topic of *Finite State Machines*. *International Journal of Engineering Pedagogy (iJEP)*, 7(1), 17–33. Retrieved from <https://online-journals.org/index.php/i-jep/article/view/6524>
- [6] Yoshidome, H. 2019. *Raising Atreus for Battle in “God of War.”* Retrieved from <https://www.gdcvault.com/play/1025768/Raising-Atreus-for-Battle->