PERANCANGAN AI PADA GAME FIGHTING DENGAN METODE MONTE CARLO TREE SEARCH

AI DESIGNING IN FIGHTING GAME WITH MONTE CARLO TREE SEARCH METHODE

Muhamad Rifqi Firdaus[1], Purba Daru Kusuma[2], Ratna Astuti Nugraheni, S.T[3]

[1,2,3] Universitas Telkom, Bandung
firdausrifqi@student.telkomuniversity.ac.id[1], purbodaru@telkomuniversity.ac.id[3], ratnaan@telkomuniversity.ac.id[3]

Abstrak

Artificial Intellegence (AI) atau biasa disebut kecerdasan buatan merupakan kecerdasan yang ditambahkan kepada sesuatu sistem yang bisa diatur. AI dapat diimplementasikan ke dalam sistem game untuk membuat setiap NPC (*Non-Player Character*) terlihat pintar dalam pengambilan keputusan. Dengan adanya AI, permainan terlihat menarik dan membuat pemain tidak mudah bosan. Jika NPC memberikan tingkah laku yang statis, maka hal tersebut membuat pemain dapat memprediksi apa yang NPC akan lakukan selanjutnya. Selain hal tersebut, NPC memberikan tingkah laku yang statis juga menyebabkan pemain menjadi bosan dan berhenti memainkan permainan tersebut.

Agar NPC memberikan tingkah laku yang tidak statis, maka dilakukan penelitian yang membuat NPC dapat memberikan tingkah laku yang berbeda-beda. Penelitian NPC dimaksudkan dengan membuat kecerdasan buatan melalui metode *Monte Carlo Tree Search* (MCTS). Metode MCTS diterapkan agar NPC menjadi lebih variatif dalam menanggapi stimulus pemain. NPC akan memberikan respon yang lebih menarik dan menantang pemain sesuai dengan alternatif pada *Decision Tree Flowchart*. Ketika pemain bermain *game*, dengan *Decision Tree Flowchart* yang didesign dapat membuat pemain melakukan beragam pergerakan NPC, permainan lebih menantang, dan tidak menjadi cepat bosan.

Hasil dari pengujian dengan menggunakan metode MCTS, diperoleh bahwa NPC dapat mengambil keputusan dengan baik dan sesuai dengan situasi yang sedang terjadi. Dengan melakukan 2 jenis pengujian yang dimana pengujian pertama *player* tidak diperbolehkan mengambil bola energi selama pertandingan berlangsung dan pengujian kedua *player* dapat mengambil bola energi selama pertandingan berlangsung. Pada pengujian pertama didapatkan hasil dimana NPC dapat memenangkan 6 dari 10 pertandingan, sedangkan pada pengujian kedua didapatkan hasil dimana NPC dapat memenangkan 5 dari 10 pertandingan.

Kata kunci : *Artificial Intelligence, Monte Carlo Tree Search, Game, NPC, Decision Tree Flowchart*

Abstract

Artificial Intelligence (AI) or commonly called artificial intelligence is intelligence that is added to a system that can be managed. AI can be implemented into the game system to make every NPC (Non-Player Character) look smart in decision making. With AI, the game looks interesting and makes players not easily bored. If the NPC gives static behavior, then it allows the player to predict what the NPC will do next. In addition to this, NPCs provide static behavior which also causes players to become bored and stop playing the game.

For NPCs to provide non-static behavior, research is carried out that makes NPCs able to provide different behaviors. NPC research is intended to create artificial intelligence through the Monte Carlo Tree Search (MCTS) method. The MCTS method is applied so that NPCs become more varied in responding to player stimuli. NPCs will provide more interesting and challenging responses to players according to the alternatives on the Decision Tree Flowchart. When players play games, the Decision Tree Flowchart that is designed can make players perform various NPC movements, the game is more challenging, and doesn't get bored quickly.

The results of the test using the MCTS method, it is found that the NPC can make decisions well and following the current situation. By doing 2 types of tests where the first test is that the player is not allowed to take energy balls during the match and the second test is that players can take energy balls during the match. In the first test, the results were obtained where the NPC could win 6 out of 10 matches, while in the second test the results were obtained where the NPC could win 5 out of 10 matches.

Keywords: *Artificial Intelligence, Monte Carlo Tree Search, Game, NPC, Decision Tree Flowchart*

---

1. Pendahuluan [10 pts/Bold]

Pada masa ini hampir semua *game* menggunakan *artificial intelligence* (AI). AI adalah kecerdasan buatan yang digunakan pada sebuah sistem sehingga memungkinkan pengguna untuk bermain melawan komputer dengan keadaan seperti melawan pemain lain. Kecerdasan buatan dapat diimplementasikan pada *Non-Player Character* (NPC) untuk membuat karakter yang ada di dalam game ini terlihat pintar. Dalam pengimplementasi AI pada *game*banyaknya pengembang *game* menggunakan AI bedasarkan *rulebased system* sehingga menghasilkan *output* yang statis dan menyebabkan NPC akan mudah untuk diprediksi oleh pengguna.

Pada penelitian tugas akhir ini akan dibuat kecerdasan buatan dengan menggunakan metode *monte carlo tree search* (MCTS) dengan bertujuan untuk membuat NPC dapat mengambil keputusan yang berbeda-beda, juga sesuai pada situasi yang sedang terjadi dan yang akan datang.

2. Dasar Teori /Material dan Metodologi/perancangan

2.1 Game

Game adalah bentuk hiburan yang memiliki aturan-aturan yang dapat dilakukan oleh pemain dan ada juga aturan yang tidak dapat dilakukan oleh pemain [1]. *Game* juga merupakan sebuah permainan yang mempunyai tujuan atau sub tujuan dimana pemain harus memberikan keputusan atau tindakan yang baik untuk mencapai suatu tujuan [2].

Game Fighting merupakan game yang memiliki interaksi antara pemain dengan sistem permainan, yang dibatasi oleh aturan-aturan tertentu. *Game* pertarungan dapat dikategorikan sebagai *role playing game* (RPG). *Game* harus mengenali sifat dan peran karakter yang menjadi perwakilan pemain didalam permainan yang biasanya adalah tokoh utama, dimana karakter tersebut dapat berubah sesuai yang diinginkan pemain. Game pertarungan menjadi salah satu game yang sering dimainkan bahkan untuk pertandingan [3].

2.2 Kecerdasan Buatan (Artificial Intelligence)

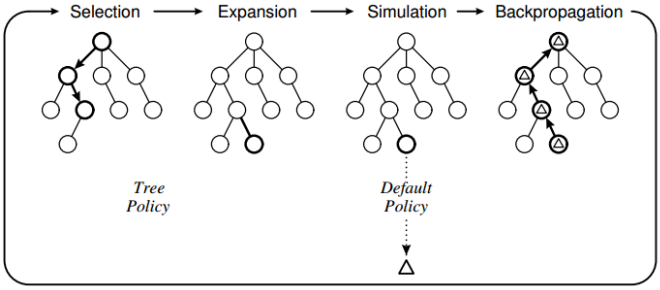
Kecerdasan buatan (*Artificial Intelligence)* adalah suatu sistem komputer yang memiliki kecerdasan layaknya manusia, dalam hal ini kecerdasan buatan banyak dikembangkan kedalam permainan untuk NPC dengan tujuan membuat NPC terlihat pintar dan membuat pemain sulit mencapai tujuan yang ingin diselesaikan. Kecerdasan buatan pada NPC pada game bertujuan untuk membuat keputusan cerdas ketika suatu kondisi memiliki beberapa pilihan dengan hasil yang berbeda, sehingga dapat menghasilkan perilaku yang relevan, efektif, dan berguna [4].

2.3 Monte Carlo Tree Search

Algoritma *Monte Carlo Tree Search* atau biasa disebut MCTS adalah suatu algoritma yang sering digunakan untuk mengambil keputusan dari beberapa sampel yang ada [5]. *Monte Carlo Tree Search* memiliki 2 proses cara pengambilan keputusan, yaitu eksplorasi dan eksploitasi. Eksplorasi merupakan sampel yang belum pernah dipilih, sedangkan eksploitasi merupakan sampel yang sudah pernah dipilih [6]. Metode *Monte Carlo Tree Search* memiliki 4 tahapan yaitu *selection*, *expansion*, *simulation*, dan *backpropagation*.

Tahap *selection* merupakan proses pemilihan node yang terdapat didalam node *tree* hingga mencapai node leaf. Tahap *exploration* atau *expansion* adalah tahapan memilih node child yang belum pernah dikunjungi, node child akan di seleksi apabila semua node sudah pernah dipilih. Jika belum, maka node tersebut akan menjadi pilihan ke tahap selanjutnya. Tahap *play-out*, node yang terpilih dan sudah diproses pada tahap sebelumnya akan disimulasikan. Pada tahap

backpropagation akan dilakukan update skor mulai dari node yang telah dipilih pada tahap *exploration* atau *selection*, hingga mencapai node root kembali [7].



Gambar 1 Monte Carlo Tree Search

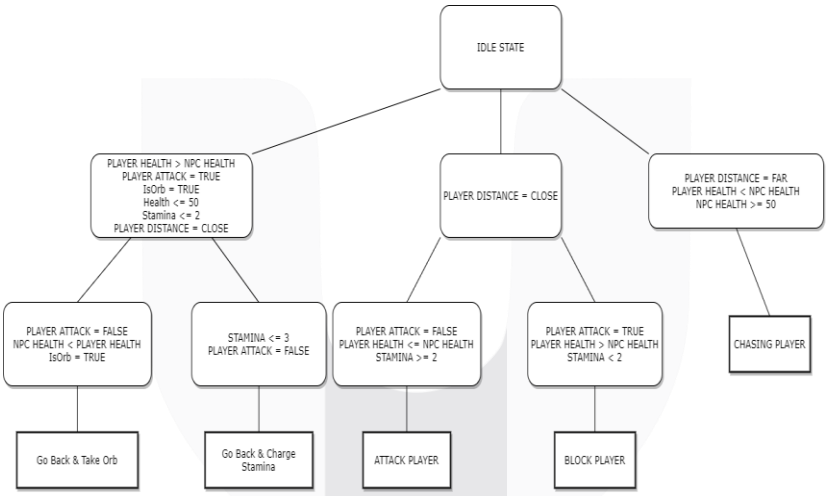
3. Implementasi

3.1 Desain Sistem

Dalam bab ini diuraikan secara rinci cara dan pelaksanaan kerja, yang bertujuan membuat perancangan hasil pengamatan percobaan atau pengumpulan data dan informasi lapangan, serta pengolahan data dan informasinya.

3.1.1 Diagram

Pada gambar di bawah merupakan beberapa aksi yang dimiliki oleh NPC saat permainan dimulai.



***Gambar 2** Diagram Tree*

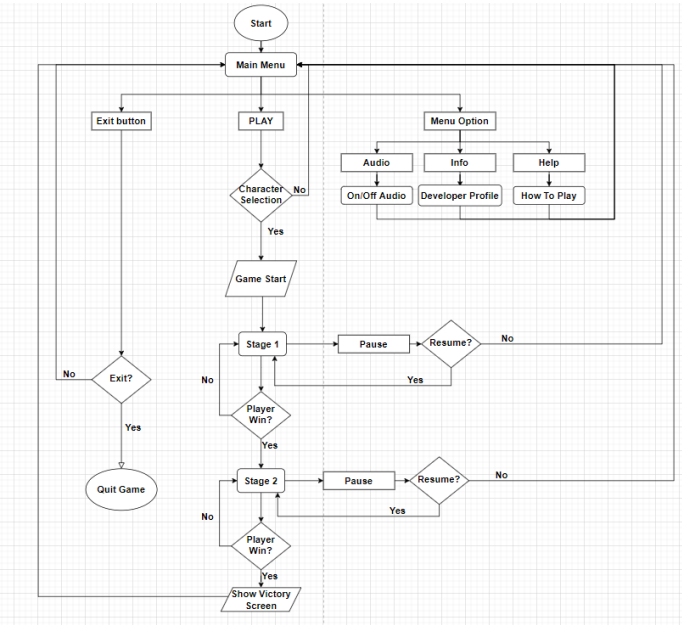
Berikut adalah pembahasan dari *Diagram* Pohon:

a) Ketika NPC berada dalam jarak yang cukup jauh dengan pengguna, darah yang dimiliki oleh NPC itu lebih besar dari 50 dan lebih besar dari dara yang dimiliki oleh pengguna, maka NPC akan mendekati pengguna.
b) Ketika NPC berada dalam jarak dekat dengan pengguna, NPC akan memilih dua pilihan dengan kondisi tertentu, NPC akan menyerang pengguna jika pengguna tidak melakukan serangan kepada NPC, darah yang dimiliki oleh NPC lebih banyak daripada darah si pengguna, dan stamina yang dimiliki harus lebih dari 2.
c) Ketika NPC berada dalam jarak yang dekat dengan pengguna, darah yang dimiliki oleh NPC lebih sedikit daripada darah pengguna, dan pengguna sedang melakukan serangan kepada NPC, maka NPC akan menghindari serangan player dan memilih dua pilihan. Yaitu jika NPC memiliki stamina kurang dari 3 dan pengguna tidak melakukan serangan, maka NPC akan mengisi ulang stamina agar dapat melakukan serangan kepada pengguna, tetapi jika NPC memiliki darah yang lebih sedikit daripada pengguna, dan terdapat bola energi di area tersebut, maka NPC akan mengambil bola energi tersebut untuk mengisi darah NPC.

3.2 Sistem *Game*

3.2.1 Flowchart Sistem Game

Pada gambar *flowchart* di bawah merupakan alur dari dari desain sistem *game fighting* yang dibuat oleh peneliti.



Gambar 3 *Flowchart System Game*

Berikut merupakan penjelasan secara rinci tentang fungsi dan fitur yang tersedia di dalam *game* berdasarkan *flowchart* di atas.

a) Ketika pemain membuka aplikasi *game fighting rumble* maka tampilan awal adalah *Main Menu* dari *game* yang berisi beberapa pilihan seperti *Play*, *Exit*, dan *Menu Option*, pilihan tersebut dapat dipilih oleh pemain.
b) Ketika pemain memilih *Menu Option* pemain akan diberi beberapa pilihan yang dapat dipilih oleh pemain, yaitu pilihan *Audio* yang berfungsi untuk menyalakan atau mematikan suara di dalam permainan, lalu pilihan *Info* yang berfungsi untuk memberikan informasi berupa nama-nama yang membuat *game fighting rumble*, dan yang terakhir adalah pilihan *Help* yang berfungsi untuk memberitahu pengguna tombol-tombol yang digunakan untuk mengendalikan karakter pada saat permainan dimainkan.
c) Ketika pemain memilih tombol *Exit* pemain akan diberikan pilihan untuk keluar dari aplikasi permainan tersebut atau kembali ke *Main Menu*.
d) Ketika pemain memilih tombol *Play* maka pemain akan langsung memulai permainan dan pemain akan masuk ke dalam pemilihan karakter, setelah pemain memilih karakter yang diinginkan maka pemain akan langsung masuk kedalam permainan.
e) Saat didalam permainan, pemain dapat memilih tombol *Pause* yang berfungsi memberhentikan permainan untuk sementara, pemain dapat melanjutkan permainan atau mengakhiri permainan dan pemain akan kembali ke *Main Menu*.
f) Setelah permainan berhasil menyelesaikan *stage* 1 dan *stage* 2, pemain akan mendapat informasi bahwa pemain sudah memenangkan permainan, pemain akan langsung kembali ke *Main Menu*.

3.2.2 Sistem Kendali *Game*

Berikut merupakan tabel rincian dari tombol kendali dengan

Tabel 1 *Kontrol Permainan*

Tombol	Fungsi
D / → / *Right Arrow*	Untuk jalan ke arah kanan
A /← / *Left Arrow*	Untuk jalan ke arah kiri

W / ↑ / *Up Arrow*	Untuk melompat
S / ↓ / *Down Arrow*	Untuk turun ke bawah jika berada di atas
Mouse 0/Z	Untuk menyerang
Mouse 1/X	Untuk mengisi energi

3.3.3 Desain Antarmuka

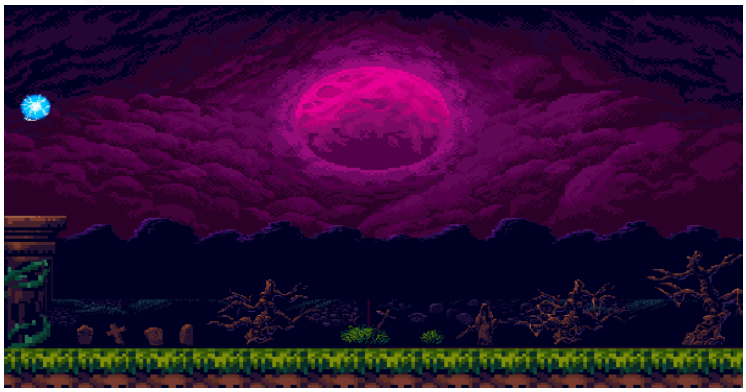
Berikut adalah rancangan awal *user interface* (UI) dari *game* yang telah dibuat, perancangan UI berguna sebagai gambaran tampilan awal dari *game* bagi pengembang *game* khususnya *game designer*.

Gambar 4 Tampilan Awal

Pada gambar 4 merupakan antarmuka yang telah dikembangkan pada menu utama.

3.3.4 Environment

Berikut merupakan environment yang terdapat pada game yang digunakan sebagai arena permainan:



Gambar 5 *Arena* Permainan

Pada Gambar 5 merupakan implementasi dari lingkungan yang dijadikan arena permainan dan merupakan *asset* yang didapatkan secara gratis dari tautan berikut:

- https://assetstore.unity.com/packages/2d/characters/gothicvania-cemetery-120509

3.3.5 Karakter

Berikut merupakan model karakter yang ada dalam permainan yang akan digunakan oleh player maupun NPC.

Gambar 6 *Tampilan Akhir Karakter*

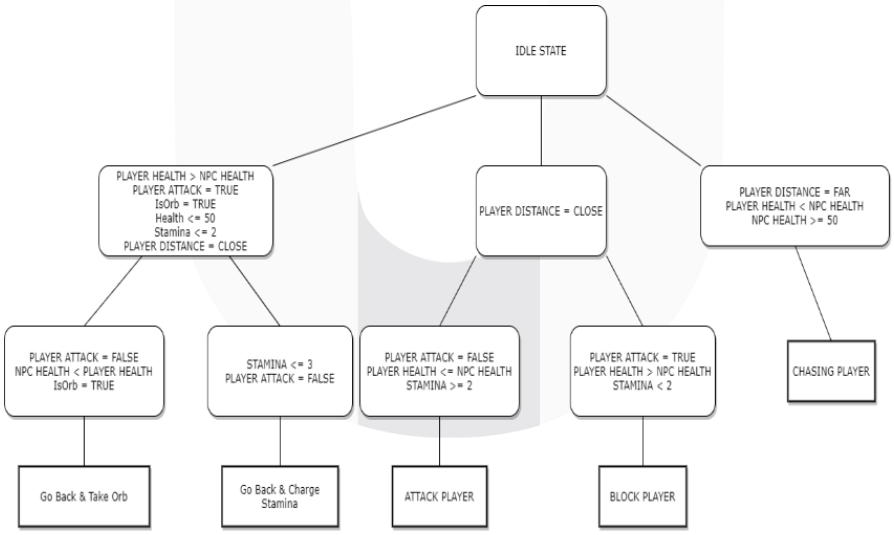
Pada gambar 6 merupakan *asset* karakter yang didapatkan dari *Unity Store* secara gratis dari tautan berikut:

1. https://assetstore.unity.com/packages/2d/characters/bandits-pixel-art-104130
2. https://assetstore.unity.com/packages/2d/characters/hero-knight-pixel-art-165188

4. Pengujian NPC

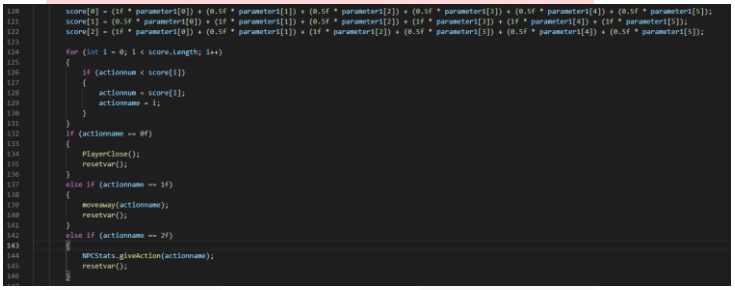
4.3.1 Pengujian NPC

Berikut merupakan skenario pengujian dengan menguji penggunaan metode MCTS terhadap performansi dari NPC yang ada di dalam game.



Gambar 7 *Diagram Tree Search NPC*

Gambar 8 *Perbandingan kondisi NPC dan Player*



Gambar 9 *Penghitungan dari hasil kondisi yang didapatkan*

Pada gambar 8 dan 9 merupakan potongan implementasi dari diagram kedalam Bahasa pemrograman C# yang akan digunakan untuk pergerakan *NPC*. Pada gambar 8 berfungsi untuk melakukan perbandingan untuk setiap kondisi yang dimiliki oleh *NPC* dan *player*. Setelah setiap kondisi sudah selesai dibandingkan dan sudah dimasukkan kedalam setiap parameter yang ada dan untuk setiap parameter akan memiliki nilai 1 atau 0 yang berarti *true* atau *false*, Pada baris 120 sampai 122 di Gambar 9 parameter tersebut akan dilakukan penghitungan untuk setiap aksi yang dimiliki oleh *NPC*, lalu pada baris 124 sampai 131 akan dilakukan pengurutan dari nilai tertinggi ke terendah, nilai tertinggi akan disimpan kedalam *variable* actionname. Pada baris 132 sampai 146 akan dilakukan pengecekan nilai dari *variable* actionname ke dalam kondisi yang sudah ditentukan, jika kondisi tersebut sesuai dengan *variable* actionname maka kondisi tersebut merupakan aksi yang akan dilakukan oleh NPC pada saat permainan berlangsung.

Gambar 10 *Hasil dari perhitungan untuk aksi NPC yang akan dilakukan*

Pada Gambar 10 dapat diketahui perhitungan yang dilakukan telah sesuai, nilai terbesar pada perhitungan tersebut yaitu *NPC* bergerak mendekati *player*.

4.3.5 Pengujian Tingkat Kemenangan NPC

Berikut perupakan skenario pengujian dengan menguji tingkat kemenangan NPC ketika melawan *player*.

Tabel 2 *Pengujian Tingkat Kemenangan NPC pada Permainan*

No	Poin Uji	Jenis Pengujian	Hasil
1	Berapa presentase kemungkinan *player* untuk bisa menang melawan *NPC* dalam 10 kali permainan.	*Black box*	40%
2	Berapa presentase kemungkinan *player* untuk bisa menang melawan *NPC* dalam 10 kali permainan, tanpa mengambil bola energi.	*Black box*	50%

Hasil dari pengujian di atas yaitu *player* dapat berhasil memenangkan pertandingan melawan NPC sebesar 40% jika terdapat bola energi dalam arena pertandingan, sedangkan *player* dapat berhasil memenangkan pertandingan sebesar 50% jika tidak terdapat bola energi dalam arena pertandingan.

5. Kesimpulan

5.1 Kesimpulan

Berikut merupakan kesimpulan dari hasil penelitian dan pengujian yang dilakukan pada tugas akhir ini.

1. Hasil dari pengujian *Artificial Intelligence* (AI) dengan menggunakan metode *Monte Carlo Tree Search* untuk pergerakan NPC pada *game Fighting Rumble* dapat berfungsi dengan baik dan sesuai seperti perancangan yang telah dibuat.
2. Berdasarkan tingkat kemenangan dengan menggunakan metode *Monte Carlo Tree Search* pada *game Fighting Rumble* dapat disimpulkan bahwa pemain masih mendapat kesulitan untuk mengalahkan NPC dikarenakan NPC dapat mengetahui kondisi yang akan dilakukan oleh pemain, dan NPC dapat mengambil bola energi sama seperti pemain.
3. Pada pengujian yang dilakukan terhadap 54 *user*, diketahui bahwa rata-rata 76.5% responden merasa suka dan tertarik untuk memainkan *game Fighting Rumble.*

5.2. Saran

Berdasarkan hasil penelitian dan pengujian yang dilakukan pada tugas akhir ini, maka saran yang dapat diusulkan untuk penelitian selanjutnya yaitu:

1. *Game* dikembangkan dengan menambah rintangan, *multiplayer*, penambahan karakter, kemampuan yang dimiliki oleh pemain atau NPC, dan *Artificial Intelligence* (AI) yang lebih kompleks.
2. *Game* dapat dikembangkan kedalam bentuk *game mobile.*

REFERENSI

[1] Eko Teguh Prasetyo, N. D. (2015). PENERAPAN KECERDASAN BUATAN PADA GAME "AIR STRIKE. *Compiler*.

[2] Matthew S.Emigh, E. G. (2014). Reinforcement Learning in Video Games Using Nearest Neighbor Interpolation and Metric Learning. *IEEE*.

[3] Santiago Bedoya-Rodriguez, C. G.-U.-Q. (2014). Augmented reality RPG card-based game. *IEEE*.

[4] Intishar Fadi Abdillah, E. M. (2018). Implementasi Adaptive AI Pada Game Turn-Based RPG Dengan Menggunakan Metode Hierarchial Dynamic Scripting. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 703-714.

[5] Nazzun Hanif Ahsani, E. M. (2017). Penerapan Algoritma Monte Carlo Tree Search Pada Permainan Komputer Maze Treasure. *JPTIIK*.

[6] Yuka Bimatara Putra, E. M. (2018). Penerapan Adaptive AI pada Game Turn Based RPG Dengan Menggunakan Metode Monte Carlo Tree Search. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2644-2648.

[7] Musta'inul Abdi, D. H. (2017). Analisis Perbandingan Kecerdasan Buatan pada Computer Player dalam Mengambil Keputusan pada Game Battle RPG. *Jurnal Ilmiah Teknologi Informasi*.

Lampiran

Jika diperlukan, tulisan dapat dilengkapi dengan lampir