



Penerapan *Non-Deterministic Finite Automata (NFA)* dan *Decision Making* Menggunakan Algoritma *Monte Carlo Tree Search (MCTS)* Menentukan Perilaku Non-Player Character (NPC) Pada Game *The Last Hope*

Revie Danial Pramadya^{*1}, Arbansyah², Sayekhti Harits³

Email: ¹reviedanial69@gmail.com, ²arb381@umkt.ac.id, ³shs500@umkt.ac.id

^{*1,2,3}Teknik Informatika, Sains dan Teknologi, Universitas Muhammadiyah Kalimantan Timur

Diterima: 01 Agustus 2023 | Direvisi: 13 Agustus 2023 | Disetujui: 30 Agustus 2023

©2020 Program Studi Teknik Informatika Fakultas Ilmu Komputer,
Universitas Muhammadiyah Riau, Indonesia

Abstrak

Perkembangan game juga berkembang pesat di Indonesia, banyak game baru yang tersebar di industri game. Para desainer game berlomba-lomba membuat game-game terbaru karena melihat peluang yang dihadirkan oleh banyaknya pengguna ponsel yang terbiasa bermain game. Dalam pembuatan game, Non-Player Character (NPC) merupakan salah satu keseruan dalam memainkan sebuah game, oleh karena itu, pengembangan NPC yang cerdas dan memiliki perilaku yang menarik menjadi fokus utama pada pembuatan game. Berdasarkan permasalahan itu peneliti ingin menerapkan Non-deterministic Finite Automata (NFA) pada NPC untuk mengatur pergerakannya sehingga permainan akan semakin menantang dan menggunakan algoritma Monte Carlo Tree Search (MCTS) yang sering digunakan pada game turn-base sebagai inputan perilaku pada game yang real-time dengan genre roguelike.

Pembuatan ini dilakukan menggunakan software Unity 2D, menggunakan bahasa C# (C Sharp) dan berjalan di android. Penelitian berfokus pada NPC Boss yang nantinya akan menjadi musuh terakhir pada game ini sekaligus menjadi akhir pada permainan. Dalam game ini pemain akan menyelesaikan sebuah stage dimana di dalam stage terdapat beberapa musuh, Ketika selesai menyelesaikan satu stage, stage berikutnya akan muncul Kembali dengan musuh yang lebih banyak hingga stage terakhir, sebelum memasuki stage terakhir pemain akan mendapatkan buff yang dapat meningkatkan status pada pemain. Untuk menambahkan keseruan di stage terakhir terdapat Boss untuk dikalahkan.

Uji coba dilakukan memakai metode blackbox dan mendapatkan hasil bahwa penerapan algoritma Monte Carlo Tree Search berhasil dilakukan pada game "The Last Hope" dengan melihat uji coba yang dilakukan. Tindakan yang dilakukan pada NPC Boss didasarkan oleh permodelan dari NFA tersebut yang kemudian di simulasikan dan dipilih Tindakan yang mana yang terbaik untuk dilakukan.

Kata kunci: *game, non-deterministic finite automata, NPC, unity, android*

Application of Non-Deterministic Finite Automata (NFA) and Decision Making Using the Monte Carlo Tree Search (MCTS) Algorithm to Determine Non-Player Character (NPC) Behavior in The Last Hope Game

Abstract

Game development is also growing rapidly in Indonesia, many new games are spreading in the gaming industry. Game designers are competing to make the latest games because they see the opportunities presented by the many mobile users who are used to playing games. In making games, the Non-Player Character (NPC) is one of the excitements of playing a game, therefore, the development of NPCs that are smart and have interesting behavior is the main focus in game making. Based on these problems the researcher wants to apply Non-deterministic Finite Automata (NFA) to NPCs to regulate their movements so that the game will be more challenging and use the Monte Carlo Tree Search (MCTS) algorithm which is often used in turn-base games as input behavior in real games. time with the roguelike genre.

This production was carried out using Unity 2D software, using C# (C Sharp) and running on Android. The research focuses on the NPC Boss which will be the last enemy in this game as well as the end of the game. In this game the player will complete a stage where there are several enemies in the stage. When finished completing one stage, the next stage will appear. player. to add to the excitement on the last stage there is a Boss to beat.

The trial was carried out using the blackbox method and the result was that the application of the Monte Carlo Tree Search algorithm was successful in the game "The Last Hope" by looking at the trials carried out. The actions taken on the NPC Boss are based on the modeling of the NFA which is then simulated and the best action is selected.

Keywords: game, non-deterministic finite automata, NPC, unity, android

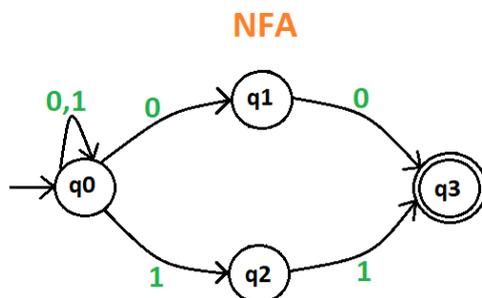
1. PENDAHULUAN

Jenis dan genre video game meningkat setiap tahun[1], Perkembangan game juga berkembang pesat di Indonesia, banyak game baru yang tersebar di industri game. Para desainer game berlomba-lomba membuat game-game terbaru karena melihat peluang yang dihadirkan oleh banyaknya pengguna ponsel yang terbiasa bermain game[2]. Saat ini semakin banyak perangkat mobile atau smartphone yang canggih yang dapat menjalankan game, mulai dari game sederhana hingga yang fungsi dan tampilannya tidak kalah dengan game komputer[3]. Game tersebut ditujukan untuk memberikan hiburan kepada pengguna smartphone[4]. Dalam pembuatan game, Non-Player Character (NPC) merupakan salah satu keseruan dalam memainkan sebuah game, oleh karena itu, pengembangan NPC yang cerdas dan memiliki perilaku yang menarik menjadi fokus utama pada pembuatan game. Pendekatan yang menarik untuk NPC adalah penggunaan automata. Automata adalah sistem yang terdiri dari sejumlah state bagian yang terbatas, di mana state bagian menentukan data input. Automata juga dianggap sebagai mesin otomatis (bukan mesin fisik), yang merupakan model matematis dari sistem yang menerima input dan menghasilkan output terdiri dari sejumlah keadaan yang terbatas[5]. Automata yang digunakan dalam pengembangan NPC adalah non-deterministic finite state automata (NFA). NFA adalah jenis Finite State Machine (FSM) di mana salah satu dari keadaan berikut tidak sepenuhnya ditentukan oleh keadaan atau input saat ini. Himpunan possible next states menunjukkan bahwa automata dapat bertransisi dari satu keadaan tertentu ke keadaan lain sebagai respons terhadap input[6]. Untuk melakukan sebuah inputan tersebut memerlukan suatu algoritma, Monte Carlo adalah algoritma komputer yang dapat menggunakan random sampling untuk memecahkan masalah. Monte Carlo dapat digunakan untuk memecahkan masalah dalam berbagai bidang seperti fisika, permainan, matematika dan bidang lainnya. Monte Carlo Tree Search (MCTS) adalah teknik pencarian kecerdasan buatan yang banyak digunakan. MCTS dapat dengan cepat menemukan langkah-langkah yang baik dalam proses keputusan sekuensial yang besar dan kompleks dengan menggabungkan pohon pencarian tradisional dengan penilaian node berdasarkan simulasi stokastik[7]. Pada penelitian ini, penulis ingin menggunakan NFA untuk memodelkan dan algoritma MCTS sebagai inputan perilaku pada NPC game "The Last Hope" untuk memberikan perilaku dengan pemain.

2. METODE PENELITIAN

2.1. Non-deterministic Finite Automata (NFA)

Non-deterministic Finite Automata (NFA) adalah salah satu jenis finite state Automata (FSA). Perbedaan antara NFA dan jenis FSA lainnya, deterministic finite automata (DFA) hanya memiliki transisi keadaan satu arah, sedangkan NFA dapat memiliki beberapa arah transisi status[8]. Non-Deterministic Finite Automata atau disebut NFA adalah salah satu Bahasa Automata yang mudah untuk diaplikasikan dan sesuai dengan logika manusia[9].



Gambar 1. Diagram NFA sederhana

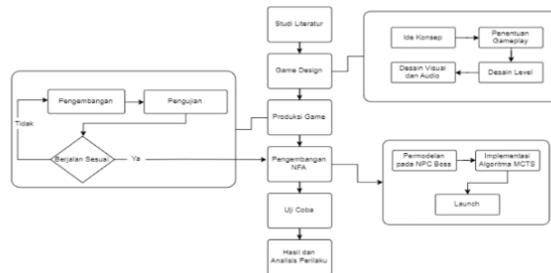
Pada gambar diatas menampilkan status FSM dengan empat status dan dua input dan output yang berbeda. Jika masukan diberikan pada keadaan awal (Q0), maka FSM berubah menjadi salah satu keadaan yaitu Q1 atau Q2. Dengan kekuatan yang berbeda-beda tergantung aktivitas yang dilakukan.

2.2. Monte Carlo Tree Search

Algoritma Monte Carlo Tree Search biasanya sering digunakan untuk mengambil sebuah keputusan berdasarkan beberapa sampel yang ada. Monte Carlo Tree Search memiliki dua proses keputusan yaitu eksplorasi dan eksploitasi. Survei adalah sampel yang tidak pernah dipilih, sedangkan eksploitasi adalah sampel yang telah dipilih. Metode Monte Carlo Tree Search memiliki 4 tahapan yaitu selection, expansion, simulation, dan backpropagation[10].

2.3. Tahapan Penelitian

Dalam tahapan penelitian ini, akan dilakukan perancangan kecerdasan buatan dengan menggunakan Algoritma MCTS (Monte Carlo Tree Search) pada game The Last Hope sebagai bukti konsep. Adapun konsep pada tahap penelitian dapat dilihat pada gambar berikut:



Gambar 3. Tahapan Penelitian

Tahapan penelitian dimulai dari Studi Literatur, Game Design, Produksi Game, Pengembangan NFA, Uji Coba, serta Hasil dan Analisis Perilaku

3. HASIL DAN PEMBAHASAN

3.1. Produksi Game

Setelah semua tahap game design dengan pengumpulan ide dan asset yaitu visual maupun audio. Tahap selanjutnya adalah produksi game, dimana tahap ini adalah implementasi dari ide-ide yang sudah dibuat.

Berikut ini adalah pengembangan dari ide-ide yang sudah dikumpulkan kemudian di implementasikan pada unity 2D



Gambar 7. Tampilan Awal

Tampilan awal diperkenalkannya logo “The Last Hope”, pemain bisa menekan layarnya yang nantinya akan menuju ke main menu.



Gambar 8. Main Menu

Di main menu terdapat gambar character utama, dan beberapa button yaitu new game, options, dan quit. Di tombol options terdapat tombol Audio untuk mengatur suara dan credit yang dimana isinya adalah lisensi dari assets yang telah digunakan.



Gambar 9. Cutscene

Di cutscene ini menceritakan tentang bagaimana jalan cerita game the last hope itu. Dan gimana konflik pada game ini terjadi.



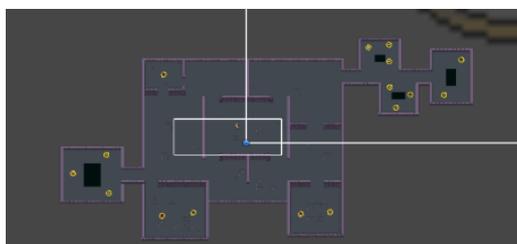
Gambar 10. Lobby

Ketika cutscene telah selesai pemain akan menuju ke lobi, yang dimana tempat ini ada safe zone yang dimana tampilan ini sesuai dengan storyboard yang dibuat pada tahap game design.



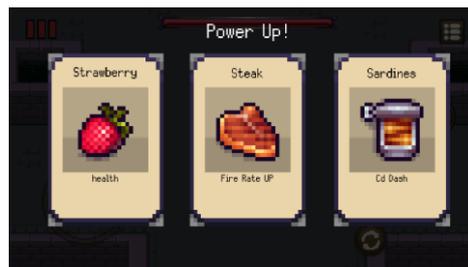
Gambar 11. Tampilan Kontrol Player

Tampilan kontrol pada pemain disesuaikan dengan referensi beberapa game di tahap sebelumnya, dimana ada sebelah kanan ada tombol menembak, dash, reload, dan pause. Sedangkan dikiri ada tombol pergerakan player dan juga ada visual darah di pojok kiri atas.



Gambar 12. Map Dungeon

Ketika player menuju kebawah lobi player akan memasuki map dungeon, yang dimana nantinya berisikan berbagai macam musuh, dan player harus mengalahkan semua itu.



Gambar 13. Buff Menu

Buff dibuat untuk menyeimbangkan gameplay pada game, buff ini menambahkan status pada player, buff tersebut muncul Ketika telah menyelesaikan sebuah stage dan menuju ke stage berikutnya.



Gambar 14. Map Boss

Map boss akan muncul pada stage terakhir game, sama halnya tujuan dari game ini, player harus mengalahkan boss untuk memenangkan game ini.



Gambar 15. Cutscene Map Boss

Sebelum melawan boss di map ini akan ada cutscene seperti pada storyboard yang sudah dibuat.



Gambar 16. Tampilan Game Over

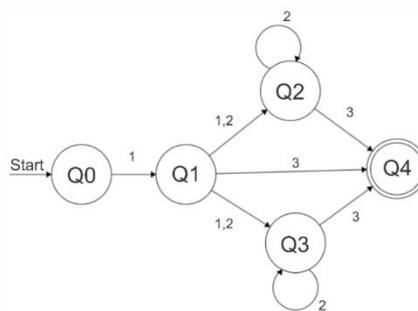
Tampilan game over akan muncul ketika darah player 0, di bagian ini terdapat button yaitu try again dan back to menu. Jika pemain menekan try again maka pemain akan Kembali ke lobi, namun ketika pemain menekan back to menu maka akan Kembali ke menu



Gambar 17. Cutscene ketika mengalahkan NPC Boss

Setelah mengalahkan NPC Boss akan ada cutscene, di cutscene ini menceritakan bagaimana kelanjutan dalam game The Last Hope.

3.2. Pengembangan NFA



Gambar 18. Non-deterministic Finite Automata

Tabel 1. State Pada NFA

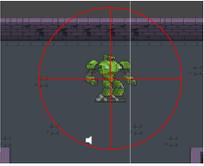
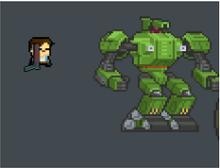
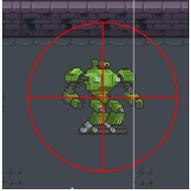
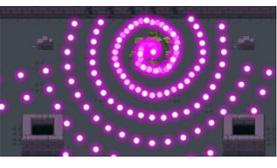
Perilaku	State(Q)
Idle	Q0
Walking	Q1
Shot	Q2
SpesialShot	Q3
Dead	Q4

Tabel 2. Inputan Action Pada NFA

Action	Input Symbol Σ
Mencari Keberadaan Player	1
Melakukan Tembakan di Area Tertentu	2
Jika HP Boss dibawah 0	3

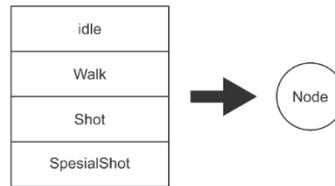
Tabel 3. Perilaku NPC Boss

No	Gambar	Keterangan
1		Keadaan diam (idle) pada NPC dimana keadaan ini sedang memproses tindakan apa yang akan digunakan
2		ketika NPC Boss berjalan(walk) maka NPC Boss akan menuju kearah player kemanapun player itu bergerak.

3		<p>Kemudian ketika NPC Boss Menembak(shot), maka tembakan itu akan menuju kearah posisi terakhir dari player. Jadi player harus kabur dari peluru itu atau akan terkena tembakan.</p>
4		<p>Ini adalah jarak pada keadaan menembak, jika player di dalam lingkaran tersebut. Maka NPC akan menembak.</p>
5		<p>Ketika player mencoba mendekati NPC Boss, maka NPC Boss akan menjauh dari player. Hal ini digunakan agar tidak adanya tabrakan dari NPC Boss kepada Player maupun sebaliknya.</p>
6		<p>Ini adalah jarak pada keadaan dimana ketika player mendekati area itu, maka NPC Boss akan bergerak mundur.</p>
7		<p>Kemudian ada beberapa special shot dari NPC Boss itu sendiri, salah satunya adalah rocket ini, sama halnya dengan shot tadi, rocket ini akan mengarah ke tempat terakhir player itu berada. Jadi player harus berpindah agar tidak terkena serangan itu.</p>
8		<p>Pada special shot ini melakukan serangan dengan seluruh arah berbentuk bulat(circle)</p>
9		<p>Pada special shot ini melakukan serangan yang serupa namun serangan ini berputar 360 derajat.</p>
10		<p>Lalu pada shot ini melakukan hal yang sama, namun melakukannya dengan secara dua arah.</p>
11		<p>NPC Boss akan melakukan Tindakan ini ketika darah NPC Boss itu berada dibawah 0, ketika animasinya selesai game object NPC Boss tersebut akan hilang</p>

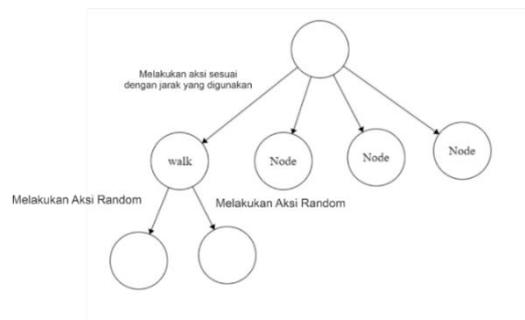
3.3. Implementasi Algoritma MCTS

Setelah permodelan dibuat, kemudian memasukan Algoritma Monte Carlo Tree Search.



Gambar 19. Algoritma MCTS

Membuat node dari perilaku yang sudah dibuat pada NPC Boss, yang nantinya akan dilakukan simulasi untuk pemilihan interaksi.



Gambar 20. Algoritma MCTS

1) Selection

Seleksi dilakukan dengan memilih nilai heuristik yang dilakukan oleh perilaku NPC Boss, nilai tersebut diambil ketika perilaku itu memberikan damage pada player dan sebaliknya. Jika perilaku itu tidak ada nilai heuristiknya, maka akan dilakukannya pemilihan perilaku dengan UCB1.

$$UCB1_i = V_i + C \sqrt{\frac{\ln N}{n_i}} \quad (2)$$

Dimana V_i adalah nilai rata-rata dari node, C adalah jumlah perbandingan eksplorasi dan eksploitasi, $\ln(N)$ merupakan logaritma natural dari total jumlah iterasi yang telah dilakukan, dan n_i adalah jumlah iterasi yang telah dilakukan.

2) Expansion

Ekspansi dilakukan ketika ada perilaku baru yang tidak tersimpan di dalam node ketika melakukan simulasi.

3) Play-out

Simulasi dilakukan sampai batas iterasi, AI akan membuat jalur node yang nantinya akan melakukan simulasi perilaku secara random, hingga sampai batas iterasi itu tercapai.

4) Backpropagation

Setelah mencapai akhir simulasi, pembaruan dilakukan, dengan memberi nilai pada setiap aksi yang dilakukan.

Untuk mengukur seberapa efisien atau beratnya algoritma, biasanya menggunakan kompleksitas algoritma. Oleh sebab itu algoritma yang sudah dibuat diukur untuk melihat seberapa kompleksitas waktu yang dibutuhkan.

Tabel 4. Kompleksitas Algoritma

Code	Keterangan
<pre> simumodel = new Model(model); GameSimul.copymodel = simumodel; while (!GameSimul.isFinished && iterationCount < maxIteration) { float heuristicScore = calculateHeuristicScore(action); System.Array actions = GameSimul.GetNextPossibleAction(action); State choice = (State) GameSimul.GetRandomAction(actions); Node exitanteNode = action.Exist(choice); if (exitanteNode == null) { Node selectedAction = action.AddChild(new Register(0, 0)); selectedAction.parent = action; selectedAction.setState(choice); action = selectedAction; }else{ action = exitanteNode; } GameSimul.PlayAction(action); iterationCount++; } </pre>	<p>Kita asumsikan bahwa while adalah big $O(n)$ kemudian akan dikalikan pada pada big $O(1)$ di dalam while sebagai berikut.</p> $O(1)+O(1)+O(n)*\{O(1)+O(1)+O(1)+O(1)+O(1)+O(1)+O(1)+O(1)+O(1)+O(1)+O(1)\}+O(1)+O(1)$ <p>Kemudian disederhanakan sebagai berikut</p> $O(1)+O(n)*12*O(1)$ <p>Lalu kalikan $O(n)*12*O(1)$ dan hasilnya berikut</p> $O(1)+O(n)$ <p>kemudian tambahkan hasilnya itu maka hasil dari $O(1)+O(n)$ adalah $O(n)$</p>

3.4. Uji Coba

Tahap uji coba ini dilakukan pada orang-orang sekitar tujuannya untuk melihat hasil. Hasil itu akan dinilai dalam bentuk kuisioner yang telah dibuat, yang nantinya akan di analisis. Uji coba ini ada dua tahap yaitu pengujian UI (User Interface) dan pada NPC Boss yang sudah dibuat.

3.5. Hasil dan Analisis Perilaku

Setelah uji coba dilakukan data dari uji coba tersebut diuji menggunakan pengujian fungsional dan pengujian black box, hal tersebut dibedakan karna tujuan dari pengujian fungsional ini dilakukan berfokus pada UI (User Interface) pada game The Last Hope, sedangkan pengujian pada black box bertujuan untuk menganalisis perilaku yang di dapatkan dari NPC Boss.

Tabel 5. Pengujian Fungsional

No	Aktivitas Pengujian Button	Aktivitas yang dilakukan	Keberhasilan	
			Berhasil	Tidak Berhasil
1	Tap To Start Game	Masuk ke menu utama	Berhasil	
2	Button New Game	Masuk ke map Lobby	Berhasil	
3	Button Options	Masuk ke pengaturan	Berhasil	
4	Button Quit	Keluar dari game	Berhasil	
5	Button pause	Menampilkan panel pause	Berhasil	
6	Button Dash	Menghindari NPC dan perpindahan player	Berhasil	
7	Button Reload	Mengisi ulang peluru	Berhasil	
8	Analog gerakan karater	Menggerakan karakter	Berhasil	
9	Analog gerakan senjata	Menggerakan senjata dan mengeluarkan peluru	Berhasil	
10	Button Skip	Melewati Cutscene yang berjalan	Berhasil	

Tabel 6. Pengujian Black Box

No.	Hasil	Kesimpulan	
		Sesuai	Tidak Sesuai
1.	Bagaimana tingkat kesulitan pada NPC Boss?	Sesuai	

2.	Apakah ada pattern atau perilaku yang tidak tertebak ketika melawan NPC Boss?	Sesuai	
3.	Bagaimana serangan pada NPC Boss apakah ada serangan yang menarik?	Sesuai	
4.	apakah ketika melawan NPC Boss memberikan rasa tantangan?	Sesuai	
5.	Apakah penambahan buff dapat menyeimbangi dari perlawanan NPC Boss?	Sesuai	

4. KESIMPULAN

Kesimpulan yang di dapat setelah melakukan pengujian pada NPC Boss untuk menentukan perilaku yaitu,

1. Permodelan NFA dapat digunakan untuk memodelkan perilaku yang akan dilakukan NPC Boss pada game “The Last Hope” NFA menggambarkan perilaku dan inputan yang akan dilakukan pada NPC Boss.

Penerapan algoritma Monte Carlo Tree Search berhasil dilakukan pada game “The Last Hope” dengan melihat uji coba yang dilakukan. Tindakan yang dilakukan pada NPC Boss didasarkan oleh permodelan dari NFA tersebut yang kemudian di simulasikan dan dipilih Tindakan yang mana yang terbaik untuk dilakukan. Dari hasil uji coba pada orang-orang sekitar, bahwa algoritma MCTS dapat memberikan Tindakan yang baik atau sesuai dengan yang diharapkan.

DAFTAR PUSTAKA

- [1] M. A. Mahendrata, E. Muhammad, A. Jonemaro, and M. Aminul, “Penerapan Decision Making NPC dengan Metode Hierarchical Finite State Machine Pada 2D Endless Runner Game,” vol. 3, no. 9, pp. 8423–8428, 2019.
- [2] R. Antonio, J. Pragantha, and D. Andana, “PEMBUATAN GAME ARCADE 2D ‘WEAPOWIZE’ BERBASIS ANDROID,” pp. 156–160.
- [3] E. Yulsilviana and H. Ekawati, “Penerapan Metode Finite State Machine (Fsm) Pada Game Agent Legenda Anak Borneo,” *Sebatik*, vol. 23, no. 1, pp. 116–123, 2019, doi: 10.46984/sebatik.v23i1.453.
- [4] M. Budiwansyah and M. Malabay, “Pembuatan Game Zombie Smasher dengan Unity berbasis Android,” *Ikraith-Informatika*, vol. 7, no. 1, pp. 116–125, 2022, doi: 10.37817/ikraith-informatika.v7i1.2243.
- [5] B. Asrun, “Penerapan Konsep Non-Deterministic Finite Automata dalam Diagnosa Penyakit Jantung,” vol. 03, no. 01, pp. 2774–2777, 2022.
- [6] Sahrul, F. Karimah, A. Muhazabah, A. D. Prasetyo, A. Yunita, and N. L. Zahra, “Pengembangan Aplikasi Permainan ‘Pilah Sampah’ Menggunakan Pemodelan Finite State Machine,” *J. Teknol.*, vol. 1, no. 1, pp. 38–46, 2018, [Online]. Available: <https://aperti.e-journal.id/teknologia/article/view/6>
- [7] A. Kennedy, N. F. Savira, S. A. Gunawan, and L. Hakim, “Implementasi Algoritma Monte Carlo Tree Search Pada Permainan 2048 Berbasis Web,” *J. Ilm. Matrik*, vol. 22, no. 3, pp. 241–248, 2020, doi: 10.33557/jurnalmatrik.v22i3.1060.
- [8] R. Agastya Nugraha, A. Mulyani, and W. Gata, “IJCIT (Indonesian Journal on Computer and Information Technology) Desain Vending Machine Rujak Buah Dengan Finite State Automata,” *IJCIT (Indonesian J. Comput. Inf. Technol.)*, vol. 5, no. 2, pp. 198–207, 2020.
- [9] N. D. Wirasbawa, L. Benedict, B. G. Santoso, M. F. Farhan, and A. Kusnadi, “[9] Penerapan Konsep Non-Deterministic Finite Automata untuk Pembuatan Sereal Menggunakan Mesin Jual Otomatis dengan Dua Sistem Pembayaran,” *Simp. Nas. Ilm. Call Pap. Unindra*, vol. 0, no. 0, pp. 978–623, 2019, doi: 10.30998/simponi.v0i0.375.
- [10] M. R. Firdaus, P. D. Kusuma, and R. A. Nugraheni, “Perancangan Ai Pada Game Fighting Dengan Metode Monte Carlo Tree Search Ai Designing in Fighting Game With Monte Carlo Tree,” *eProceedings ...*, vol. 8, no. 5, pp. 6487–6495, 2021, [Online]. Available: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/16410>
- [11] A. Arban, “Implementasi Finite State Machine (FSM) pada Agent Permainan Game Lost Animal at Borneo berbasis Android,” *J. CoSciTech (Computer Sci. Inf. Technol.)*, vol. 3, no. 2, pp. 144–151, 2022, doi: 10.37859/coscitech.v3i2.3921.
- [12] R. Asrianto and M. Herwinanda, “Jurnal Computer Science and Information Technology (CoSciTech) algoritma support vector machine,” vol. 3, no. 3, pp. 431–440, 2022.