



Penerapan Algoritma A* Pathfinding dan Behavior Tree Pada Perilaku Non-Playable Character (NPC) pada Game Labirin “Dungeon Escape”

Wildan Fadilah^{1*}, Diny Syarifah Sany²

¹⁻² Program Studi Teknik Informatika, Fakultas Teknik, Universitas Suryakencana, Indonesia

wildanfa62@gmail.com^{1*}, dny.sany@gmail.com²

Alamat: Jl. Pasirgede Raya, Bojongherang, Kec. Cianjur, Kabupaten Cianjur, Jawa Barat 43216

Korespondensi penulis: wildanfa62@gmail.com*

Abstract: This study focuses on the development of an intelligent enemy behavior system in the 2D maze game *Dungeon Escape* by combining the A* algorithm for pathfinding and a Behavior Tree (BT) structure for decision-making. The main objective is to enhance Non-Playable Character (NPC) intelligence, enabling more adaptive and realistic interactions with players. The A* algorithm is implemented to allow NPCs to pursue players through the shortest and most efficient paths while avoiding obstacles on a grid-based map. Meanwhile, the Behavior Tree framework is designed to manage NPC actions based on dynamic conditions, such as attacking when in close proximity, chasing players within a certain detection radius, and retreating to the original guard position when the player leaves the active zone. The research methodology involves a comprehensive literature review, system design, and implementation using the Unity game engine. Testing procedures consist of both white-box and black-box approaches to evaluate the correctness, functionality, and efficiency of the system. The results indicate that all major game features—including player navigation, combat mechanics, key collection, enemy pathfinding, and user interface interactions—operate smoothly as expected. Furthermore, NPCs exhibit adaptive behavior by dynamically switching between patrolling, chasing, and attacking modes depending on the player's location and proximity. Performance testing shows that the integrated A* and BT system runs efficiently without significant delays or instability, even in higher-level stages with more complex layouts. The final game prototype includes seven progressively challenging levels, offering players an engaging and dynamic gameplay experience. This study demonstrates that the combination of pathfinding and decision-making algorithms provides an effective solution for designing intelligent NPCs, improving both realism and entertainment value in 2D games. The findings are expected to serve as a useful reference for future research and development in AI-based game design, particularly in the context of Unity game projects.

Keywords: Artificial Intelligence, Behavior Tree, Game, NPC, Unity

Abstrak: Penelitian ini berfokus pada sektor Usaha Mikro, Kecil, dan Menengah (UMKM) yang memiliki peran strategis dalam perekonomian nasional. Namun, masih banyak pelaku UMKM yang belum secara optimal memanfaatkan informasi akuntansi serta belum mengintegrasikan pemasaran digital dalam aktivitas bisnis mereka. Oleh karena itu, penelitian ini dilakukan untuk mengetahui sejauh mana kedua variabel tersebut berkontribusi terhadap keberhasilan UMKM. Penelitian ini menggunakan pendekatan kuantitatif dengan metode pengujian hipotesis. Data primer dikumpulkan melalui penyebaran kuesioner kepada 125 pelaku UMKM yang dipilih menggunakan teknik purposive sampling. Analisis data dilakukan dengan regresi linear berganda menggunakan perangkat lunak SPSS versi terbaru. Hasil penelitian menunjukkan bahwa penggunaan informasi akuntansi dan pemasaran digital berpengaruh positif dan signifikan terhadap keberhasilan bisnis UMKM. Temuan ini menegaskan bahwa pengelolaan informasi akuntansi yang baik dapat membantu UMKM dalam mengambil keputusan bisnis yang lebih tepat, seperti pengendalian biaya, pengelolaan arus kas, serta perencanaan keuangan yang berkelanjutan. Di sisi lain, penerapan strategi pemasaran digital memberikan peluang bagi UMKM untuk memperluas jangkauan pasar, meningkatkan interaksi dengan konsumen, serta memperkuat daya saing di era transformasi digital. Secara keseluruhan, penelitian ini memberikan implikasi penting bagi pengembangan UMKM, baik dari sisi akademis maupun praktis. Dari sisi akademis, hasil penelitian ini memperkuat literatur terkait pentingnya informasi akuntansi dan pemasaran digital sebagai faktor penentu keberhasilan UMKM. Sementara dari sisi praktis, penelitian ini memberikan rekomendasi kepada pelaku UMKM untuk meningkatkan pemahaman dan penerapan akuntansi sederhana yang sesuai dengan kebutuhan usaha serta memaksimalkan pemanfaatan media digital dalam strategi pemasaran. Dengan demikian, penggabungan kedua aspek ini diharapkan dapat memperkuat daya saing dan keberlanjutan UMKM di tengah dinamika ekonomi yang semakin kompleks.

Kata kunci: Kecerdasan Buatan, NPC, Permainan, Pohon Perilaku, Unity

1. LATAR BELAKANG

Game dalam bahasa Indonesia merupakan permainan yang dimainkan untuk mendapatkan kesenangan dalam memainkannya *game* telah menjadi bagian dari budaya kuno mulai dari permainan papan tradisional hingga video *game* modern, dalam perkembangan dunia *game* sekarang sangat banyak menerapkan kecerdasan buatan pada *game*, Penerapan kecerdasan buatan pada *game* membuat *game* lebih interaktif dan dapat meningkatkan pengalaman bermain bagi pemain.

Labirin adalah sebuah puzzle dalam bentuk percabangan jalan yang kompleks dan memiliki banyak jalan buntu. Tujuan permainan ini adalah pemain harus menemukan jalan keluar dari sebuah pintu masuk ke satu atau lebih pintu keluar. Bisa juga kondisi pemain menang yaitu ketika dia mencapai suatu titik di dalam maze tersebut. Maze dalam dunia nyata banyak dibuat di taman atau ruangan-ruangan dengan pembatas berupa pagar tanaman, tembok atau pagar. Ukurannya bervariasi, tergantung ukuran ruangan atau taman tersebut. (Widodo & Ahmad, 2017)

Non playable character (NPC) adalah objek dinamis yang tidak dikontrol oleh pemain, dan dapat menentukan perilakunya sendiri dan beroperasi dalam ruang virtual dengan kecerdasan buatan (Fikriansyah et al., 2023). Kecerdasan buatan pada *game* seringkali diterapkan pada *game* untuk membuat karakter atau *npc* memiliki perilaku yang mampu bergerak maupun berfikir.

Pathfinding adalah proses untuk menentukan jalur terbaik dari satu titik ke titik lainnya dalam suatu lingkungan. Dalam pengembangan *game*, pathfinding sangat penting untuk mengatur pergerakan karakter non-pemain (NPC) agar dapat bergerak secara logis dan efisien. Salah satu algoritma yang umum digunakan adalah A* karena mampu menemukan jalur optimal dengan mempertimbangkan biaya jarak dan estimasi tujuan (Octavian & Hermawan, 2023).

Banyak penelitian terdahulu yang menerapkan Algoritma A* pada *non-playable character (NPC)* musuh untuk pencarian jalur pada pemain, namun itu membuat karakter musuh hanya berperilaku untuk mencari jalur ke pemain saja. Dalam kasus ini di *game* labirin *dungeon escape non-playable character (NPC)* tidak hanya menerapkan A* untuk pencarian jalur pada pemain tetapi menambahkan *Behavior Tree* untuk memberikan perilaku pada *non-playable character* agar tidak hanya sekedar mengejar saja, tetapi juga dapat memutuskan tindakan lain seperti menyerang saat dekat, kembali ke posisi semula jika pemain menjauh, atau tetap diam saat tidak mendeteksi pemain (Buckland, 2025).

Penggunaan algoritma A* dalam penelitian ini untuk memungkinkan NPC musuh mencari jalur terpendek menuju pemain secara efisien dalam lingkungan grid seperti labirin. A* menggabungkan biaya aktual dan estimasi, sehingga optimal dan cepat. Sementara itu, Behavior Tree digunakan sebagai sistem pengambilan keputusan yang modular, memungkinkan NPC untuk mengejar, menyerang, atau kembali ke posisi awal sesuai kondisi permainan. Integrasi antara A* dan Behavior Tree menjadikan NPC bertindak sesuai kondisi permainan.

Berdasarkan uraian diatas, penelitian ini akan menerapkan *pathfinding* menggunakan algoritma A* ke dalam game labirin dungeon escape dikarenakan Algoritma A* dapat menjamin untuk menemukan jalur terbaik bagi *non playable character* dan *Behavior tree* sebagai pengatur tindakan *npc* pada *game*.

2. METODE PENELITIAN

Penelitian ini dilakukan melalui serangkaian tahapan yang terstruktur, yang dirancang untuk mencapai tujuan pengembangan NPC dalam game berbasis grid. Proses dimulai dengan studi literatur yang bertujuan mengumpulkan dan menelaah referensi terkait algoritma A*, struktur Behavior Tree, perilaku NPC, serta penerapannya dalam konteks game. Selanjutnya dilakukan analisis terhadap permasalahan penelitian, mencakup kebutuhan NPC dan pemain, serta justifikasi penggunaan algoritma A* untuk pencarian jalur dan Behavior Tree sebagai sistem pengambilan keputusan. Tahapan berikutnya adalah desain NPC yang mencakup perancangan struktur perilaku menggunakan Behavior Tree dan penentuan skenario pemanfaatan A*. Tahap ini termasuk dalam bagian Game Design dalam siklus GDLC (Game Development Life Cycle). Setelah itu, dilakukan desain game secara menyeluruh, mulai dari level, mekanik, visual, hingga interaksi pemain-NPC, yang termasuk dalam tahap Game Design & System Design. Hasil desain tersebut kemudian diimplementasikan dalam bentuk game, termasuk integrasi struktur Behavior Tree dan algoritma A*, yang merupakan bagian dari fase pengembangan (Development) dalam GDLC. Tahap pengujian dilakukan dengan metode white-box untuk memverifikasi logika internal dan simulasi perilaku NPC dalam berbagai situasi gameplay. Setelah pengujian, dilakukan evaluasi terhadap efektivitas NPC dan integrasi sistem AI. Akhirnya, penelitian ini disimpulkan dengan menyajikan hasil secara menyeluruh, termasuk kelebihan dan kekurangan pendekatan yang digunakan.

3. HASIL DAN PEMBAHASAN

Implementasi

Bagian ini merupakan implementasi dari perancangan game yang telah dibuat sebelumnya dan disertai dengan pembahasan dari setiap proses yang dilakukan dan hasil yang didapat.

Pembuatan Game

Pembuatan game adalah implementasi dari rancangan game desain yang telah dibuat pada bab sebelumnya, sehingga pada bagian ini akan menjelaskan hasil dari rancangan game yang telah dibuat pada game desain (Chamandard, 2023), adapun dalam proses pembuatan game ini, digunakan beberapa alat dan perangkat lunak yang mendukung berikut merupakan tabel alat dan perangkat lunak yang digunakan:

Tabel 1. Kebutuhan software

Alat / Software	Fungsi
Unity Game Engine Versi 2022.3.30f1	Game engine untuk membangun dan menjalankan game.
C#	Bahasa pemrograman untuk sistem kontrol, gameplay, pathfinding, dan behavior tree.
Visual Studio Code	Untuk menulis dan menguji skrip C# secara terintegrasi dengan unity.
Krita	Mendesain aset visual seperti tile, button dan game objek lain.
Itch.io	Sumber asset gratis yang digunakan.

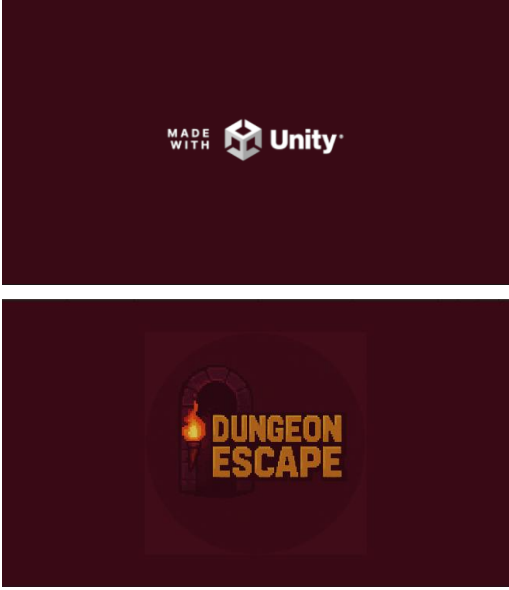
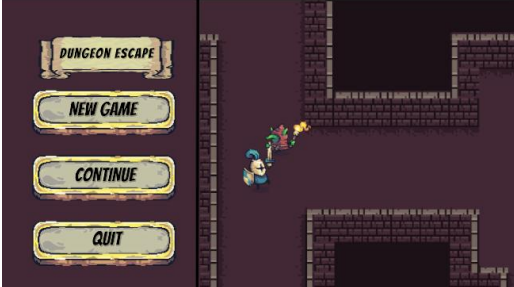
Tabel 2. Kebutuhan hardware

Hardware/Komponen	Spesifikasi
Processor (CPU)	Intel Core i3-10100f
RAM	8GB
Penyimpanan	SSD 256GB
Kartu Grafis (GPU)	Nvidia GeForce 1660 s
Monitor	Resolusi 1920 x 1080
Sistem Operasi	Window 10 (64-bit)

Implementasi Antar Muka

Tujuan dari perancangan antarmuka ini adalah untuk memberikan navigasi yang jelas dan tampilan visual yang mendukung kenyamanan user selama bermain (Chamandard, 2022), berikut merupakan table implementasi antarmuka:

Tabel 3. Implementasi Antar Muka

Antarmuka	Visual
1. Splash screen	
2. Menu utama	
3. Pause	
4. Retry	

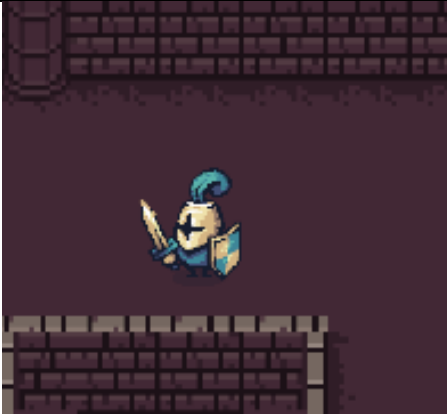
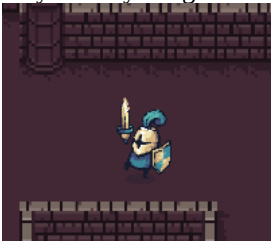
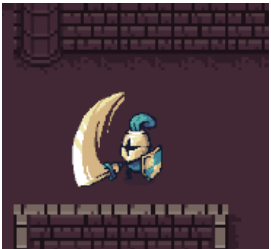
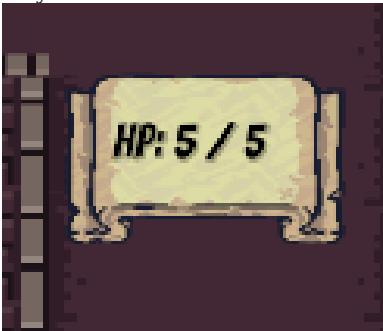
5. Level Selesai	
6. Permainan Selesai	

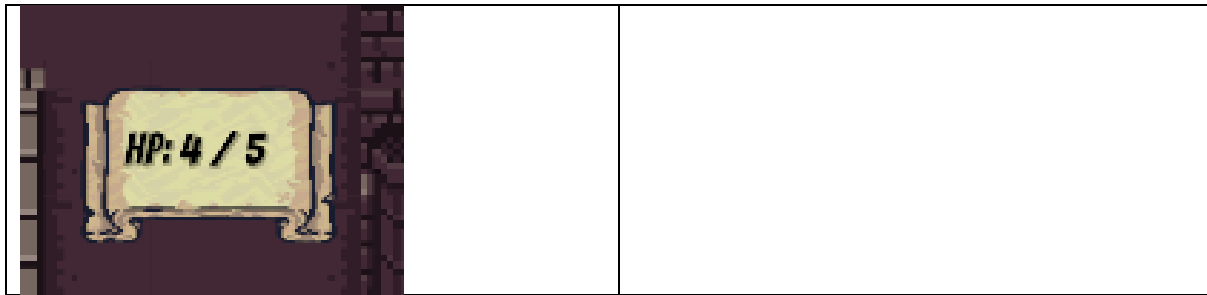
Implementasi Karakter pemain

Pemain merupakan aktor utama dalam game dan menjadi pusat dari sistem permainan (Isla, 2025). Implementasi pemain meliputi beberapa yaitu pergerakan, menyerang, menghindar, dan kesehatan (HP)

Tabel 4. Implementasi Player

Visual	Deskripsi
Pergerakan player 	Pemain bergerak dalam empat arah (atas, kiri, bawah, kanan) menggunakan input dari keyboard w, a, s, d atau arah panah.

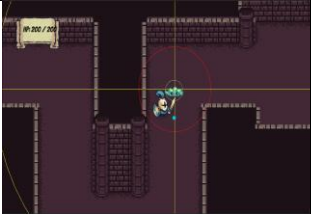
		
Player menyerang  	Pemain dapat menyerang musuh jika berada dalam jarak dekat. Serangan dilakukan menggunakan input tombol space dan ditampilkan melalui animasi.	
Player menghindar 	Pemain dapat menghindari serangan musuh, serangan dilakukan dengan menekan tombol kiri shift	
Player Health 	Kesehatan pemain ditampilkan sebagai indikator berapa jumlah nyawa yang dimiliki pemain, kesehatan akan berkurang seiring terkena serangan dari musuh	



Implementasi NPC

Implementasi NPC dalam game ini terdiri dari tiga bagian utama, yaitu pengambilan keputusan menggunakan Behavior Tree, mengejar menggunakan A*, menyerang menggunakan event animasi, dan kembali ke posisi awal (Menouar, 2017).

Tabel 1. Implementasi NPC

No	NPC mengejar	NPC menyerang	NPC kembali keposisi awal
1			

Berdasarkan tabel di atas dapat dijelaskan, untuk lingkaran berwarna kuning merupakan zona radius NPC untuk mengejar pemain, lingkaran berwarna merah merupakan zona radius NPC menyerang pemain, dan garis biru merupakan *path* yang dipilih NPC menggunakan algoritma A*

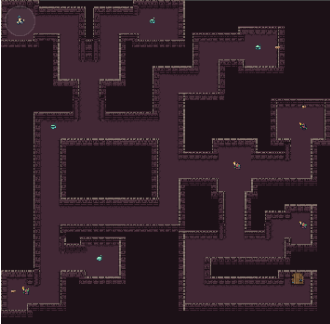
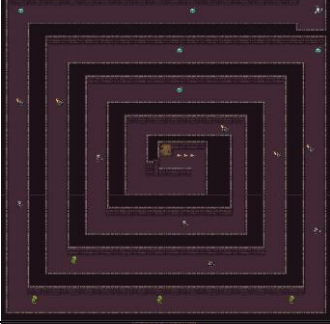
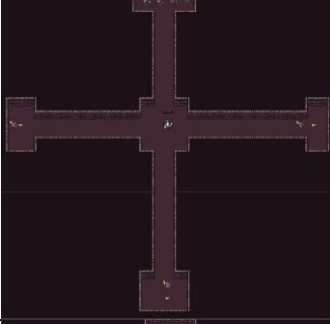
Implementasi Peta atau Map level permainan

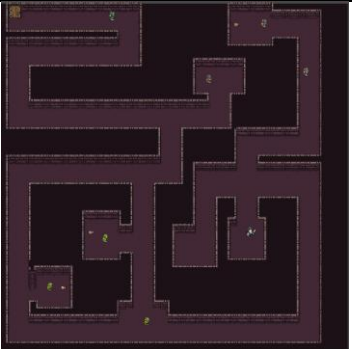
Peta permainan di implementasikan menggunakan fitur Tilemap 2D dari Unity untuk memudahkan pembangunan area permainan berbasis grid. Setiap peta mewakili satu level yang memiliki tata letak berbeda (Millington, 2016). Total terdapat 7 peta yang dibuat, masing-masing dengan jumlah musuh yang berbeda. Setiap peta memiliki lapisan (layer) seperti berikut:

1. Grid, sebagai pondasi dasar peta permainan yakni dengan ukuran x dan y 50x50, dan juga sebagai pondasi dasar grid untuk pathfinding.
2. Floor – menampilkan lantai atau jalan yang bisa dilalui.
3. Wall – berisi dinding atau batas yang tidak bisa dilalui atau unwalkable node, dilengkapi dengan tilemapcollider (Pinter, 2020).

Berikut merupakan tabel peta permainan

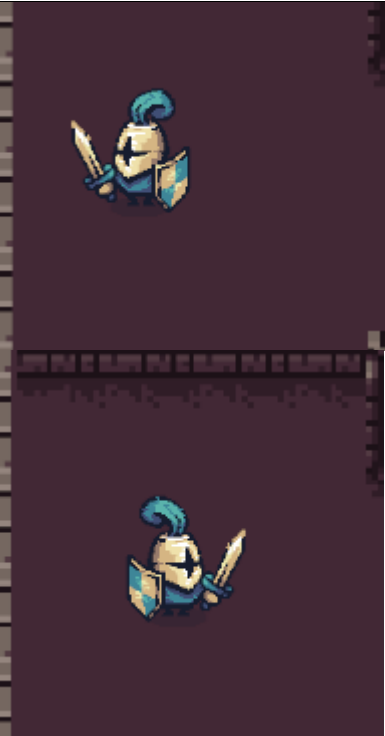
Tabel 2. Implementasi peta permainan

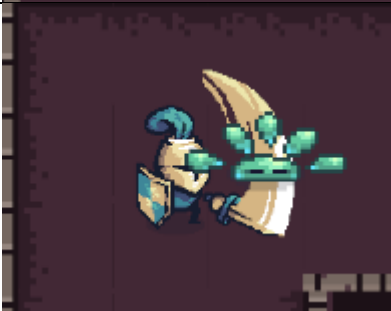
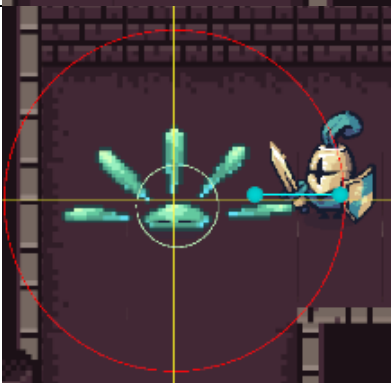
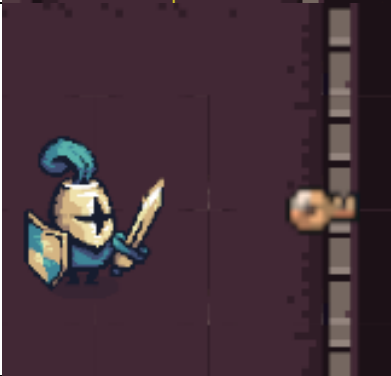
No	Gambar peta permainan	Ukuran Grid	Jumlah musuh	Jumlah Kunci
1		50 x 50	8	3
2		50 x 50	7	3
3		50 x 50	19	3
4		50 x 50	7	3
5		50 x 50	7	3

6			50 x 50	7	3
7			50 x 50	12	3

Pengujian Black Box

Tabel 7. Pengujian Black Box

No	Fitur	Input	Visual	Output	Hasil
1	Pergerakan Pemain	Menekan (WASD) atau (↑↓←→)		Pemain Bergerak ke arah yang ditekan	Berhasil

					
2	Pemain menyerang	Menekan space		Pemain menyerang NPC	Berhasil
3	Npc Menyerang pemain	Pemain masuk radius 2 < tile		NPC menyerang pemain	Berhasil
4	Pengambilan Kunci	Pemain menyentuh objek kunci		Objek kunci menghilang	Berhasil

5	Pintu keluar setelah 3 kunci didapat	Pemain menyentuh pintu keluar		Menampilkan Panel level complete	Berhasil
6	Tombol pause	Menekan ESC saat bermain		Menampilkan panel pause	Berhasil
7	Game Over	Saat HP pemain habis		Menampilkan panel retry	Berhasil

Pengujian White Box

Pengujian pada Perilaku NPC

Tabel 3. Pengujian White box perilaku NPC

No	Fungsi yang di uji	Kondisi	Logika yang diharapkan	Hasil
1	CheckRadius()	Pemain berada di jarak < 2f	Mengembalikan status Succes ke node Attack	Berhasil
2	ChasePlayer.Update()	Pemain berada di jarak 2-10	Pathfinding dijalankan, NPC mengejar	Berhasil
3	ReturnToOrigin()	Pemain tidak terdeteksi (diluar radius deteksi NPC)	NPC kembali ke posisi awal	Berhasil
4	DealDamage()	Saat pemain berada dalam radius deteksi < 2f NPC dan animasi serang aktif	Pemain kehilangan HP	Berhasil
5	ExitGate.OnTriggerEnter	Pemain memiliki 3 kunci	Panel Level complete muncul	Berhasil

Pengujian pada Pencarian Jalur NPC

Tabel 4. Pengujian White Box pada pathfinding

No	Fungsi yang diuji	Kondisi yang diuji	Logika yang diharapkan	Hasil
1	FindPath(Start,goal)	NPC dan pemain berada di posisi berbeda	Fungsi dijalankan, node start & goal diambil dari grid	Berhasil

2	GetNeighbours(node)	Node berada di tengah grid	Mengembalikan 4 atau 8 tangga tergantung mode grid	Berhasil
3	GetDistance(a, b)	Node A dan B berada 3 tile berjauhan	Mengembalikan nilai jarak yang sesuai	Berhasil
4	RetracePath()	Path ditemukan di tile tembok	Jalur disusun mundur dan dibalik, tidak kosong	Berhasil
5	Node unwalkable	NPC atau target di tile tembok	Path gagal, fungsi beri peringatan / mengembalikan null	Berhasil

Pengujian Perilaku NPC dengan Simulasi

Pengujian dilakukan dengan menyusun beberapa skenario simulasi untuk mengevaluasi bagaimana NPC merespons kehadiran pemain dalam radius tertentu (Rabin, 2015). Tujuannya adalah memastikan bahwa struktur behavior tree NPC berjalan sesuai desain yang meliputi: menyerang, mengejar, dan kembali ke posisi awal.

Tabel 5. Pengujian Perilaku NPC dengan Simulasi

No	Kondisi Simulasi	Diharapkan NPC Melakukan	Hasil
1	Pemain diam pada jarak 5 tile dari NPC	NPC mengejar pemain (masuk radius kejar 2-10)	Sesuai
2	Pemain sangat dekat (dalam jarak < 2 tile)	NPC menyerang pemain	Sesuai
3	Pemain menjauh lebih dari 10 tile	NPC kembali ke posisi awal	Sesuai
4	Pemain masuk radius serang saat NPC sedang kembali ke posisi awal	NPC berhenti kembali ke posisi awal, dan langsung menyerang	Sesuai

Pengujian Pathfinding NPC dengan Simulasi

Pengujian ini bertujuan untuk mengevaluasi kemampuan NPC dalam mencari dan menavigasi jalur dari posisi awal menuju target menggunakan algoritma A* berbasis grid, pengujian dilakukan dengan menyusun beberapa skenario simulasi, di mana posisi pemain dan halangan diatur sedemikian rupa untuk mengamati apakah NPC dapat menemukan jalur yang benar, dan optimal (Russel et al., 2020).

Tabel 6. Tabel pengujian Pathfinding NPC dengan Simulasi

No	Map	Posisi Pemain	Posisi NPC	Jalur pathfinding	Jumlah Langkah
1	Level 1	{3,4}	{5,6}	{5,6} → {4,6} → {3,6} → {3,5} → {3,4} = 5	5
2	Level 2	{2,2}	{5,3}	{5,3} → {4,3} → {3,3} → {3,2} → {2,2} = 5	5
3	Level 3	{7,8}	{7,14}	{7,14} → {7,13} → {7,12} → {7,11} →	7

				$\{7,10\} \rightarrow \{7,9\} \rightarrow \{7,8\}$ $= 7$	
4	Level 4	$\{25,8\}$	$\{25,10\}$	$\{25,10\} \rightarrow \{25,9\} \rightarrow \{25,8\} \rightarrow \{25,7\} \rightarrow \{25,8\} = 5$	5
5	Level 5	$\{6,3\}$	$\{3,5\}$	$\{3,5\} \rightarrow \{3,4\} \rightarrow \{3,3\} \rightarrow \{4,3\} \rightarrow \{5,3\} \rightarrow \{6,3\}$	6
6	Level 6	$\{9,3\}$	$\{13,3\}$	$\{13,3\} \rightarrow \{12,3\} \rightarrow \{11,3\} \rightarrow \{10,3\} \rightarrow \{9,3\} = 5$	5
7	Level 7	$\{6,3\}$	$\{11,3\}$	$\{11,3\} \rightarrow \{10,3\} \rightarrow \{9,3\} \rightarrow \{8,3\} \rightarrow \{7,3\} \rightarrow \{6,3\} = 6$	6

Pengujian dilakukan dengan menempatkan NPC dan pemain pada berbagai posisi di peta grid yang telah dibangun. Pathfinding dihitung menggunakan algoritma A* yang diterapkan pada sistem NPC. Hasil menunjukkan bahwa jalur yang diambil NPC selalu memilih rute terdekat, NPC menghindari node unwalkable, jika posisi NPC dan pemain sama, maka NPC tidak bergerak

Evaluasi

Berdasarkan hasil pengujian black box dan white box yang telah dilakukan, seluruh fitur utama dalam game telah berfungsi sesuai dengan skenario yang dirancang. Fitur pergerakan pemain, serangan musuh, pengumpulan kunci, sistem pause, dan penyelesaian level telah berhasil

Penerapan integrasi Algoritma A* dan Behavior Tree pada musuh menunjukkan perilaku yang adaptif, mulai dari mengejar, menyerang, hingga kembali ke posisi awal, sesuai dengan radius interaksi yang ditentukan (Sharma et al., 2022).

Dengan demikian, dapat disimpulkan bahwa game ini telah berhasil memenuhi tujuan utama dari pengembangan, yaitu menciptakan game labirin dungeon escape dengan musuh yang dapat berperilaku cerdas dan tidak hanya mengejar tetapi kembali ke tempat semula.

Hasil Penelitian

Hasil penelitian ini membuktikan bahwa penggunaan behavior tree dan algoritma A* dapat meningkatkan perilaku NPC agar lebih responsif, tidak acak, dan menantang. Sistem yang dibangun berjalan stabil, dan pengalaman bermain menjadi lebih adaptif dalam berperilaku (Togelius et al., 2016).

Hasil Implementasi

Penelitian ini menghasilkan sebuah permainan 2D berjenis labirin dungeon escape, di mana pemain harus mengumpulkan kunci dan keluar dari labirin sambil menghindari atau menghadapi NPC musuh. NPC dikendalikan oleh sistem kecerdasan buatan yang menggunakan kombinasi algoritma A* untuk pathfinding dan behavior tree untuk mengatur perilaku (Tozour, 2024).

Berikut adalah hasil implementasi utama

- a. NPC mampu memilih jalur optimal menuju pemain
- b. Perilaku NPC berubah secara dinamis berdasarkan radius
- c. Struktur kondisi (if, sequence, selector) bekerja sesuai harapan

Simulasi pengujian juga menunjukkan bahwa pathfinding NPC dapat menyesuaikan dengan struktur peta, termasuk kondisi:

1. Jalan tertutup
2. Pemain berpindah cepat
3. NPC berada di posisi sama dengan pemain

Hasil Pengujian

Berdasarkan pengujian black box, semua fitur utama seperti pergerakan, interaksi UI, serangan, sistem kunci, dan transisi level telah berjalan sesuai dengan rancangan. Tidak ditemukan error kritis selama proses pengujian.

Pengujian white box pada logika behavior tree dan algoritma A* menunjukkan bahwa:

- a. NPC mampu memilih jalur optimal menuju pemain
- b. Perilaku NPC berubah secara dinamis berdasarkan radius
- c. Struktur kondisi (if, sequence, selector) bekerja sesuai harapan

Simulasi pengujian juga menunjukkan bahwa pathfinding NPC dapat menyesuaikan dengan struktur peta, termasuk kondisi:

1. Jalan tertutup
2. Pemain berpindah cepat
3. NPC berada di posisi sama dengan pemain

4. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, dapat disimpulkan bahwa game *top-down dungeon escape* berhasil dikembangkan menggunakan Unity dengan sistem musuh (NPC) yang mampu berperilaku cerdas dan tidak

acak. Implementasi algoritma A* untuk *pathfinding* memungkinkan NPC menemukan jalur tercepat menuju pemain dengan menghindari rintangan (*unwalkable tile*). Selain itu, penggunaan *behavior tree* berhasil memisahkan logika NPC menjadi beberapa perilaku, yaitu menyerang, mengejar, dan kembali ke posisi awal sesuai dengan radius jarak terhadap pemain. Fitur-fitur permainan seperti pergerakan pemain, serangan, antarmuka pengguna (UI) seperti *main menu*, *pause*, dan *level complete*, serta sistem pengumpulan kunci telah diuji dan berfungsi sesuai dengan rencana. Hasil pengujian, baik secara *black box* maupun *white box*, menunjukkan bahwa sistem bekerja secara stabil, adaptif, dan sesuai dengan tujuan penelitian. Game ini juga menyediakan 7 level dengan tingkat kesulitan yang meningkat, dan seluruh mekanisme berhasil berjalan dengan baik pada setiap level.

UCAPAN TERIMA KASIH

Terima kasih kepada dosen pembimbing yang telah memberikan arahan dan masukan berharga selama proses penelitian ini. Terima kasih juga kepada rekan-rekan Program Studi Teknik Informatika Universitas Suryakencana atas dukungan dan kerjasama yang diberikan. Semoga hasil dari penelitian ini dapat memberikan manfaat bagi pengembangan teknologi game berbasis kecerdasan buatan di masa mendatang.

DAFTAR REFERENSI

- Buckland, M. (2025). *Programming Game AI by Example*. Jones & Bartlett Learning.
- Champanand, A. J. (2022). *AI game development: Synthetic creatures with learning and reactive behaviors*. New Riders.
- Champanand, A. J. (2023). Understanding behavior trees. AiGameDev.com. Retrieved from <https://aigamedev.com>
- Fikriansyah, M., Wahyu Wiriasto, G., & Rachman, A. S. (2023). Non-player character in fire fighter games using genetic algorithm. *Jurnal Mnemonic*, 6(1), 11-19. <https://doi.org/10.36040/mnemonic.v6i1.5757>
- Isla, D. (2025). Handling complexity in the Halo 2 AI. Game Developers Conference. Retrieved from <https://www.gdcvault.com>
- Menouar, H., Guvenc, I., Akkaya, K., Uluagac, A. S., Kadri, A., & Tuncer, A. (2017). UAV-enabled intelligent transportation systems for the smart city: Applications and challenges. *IEEE Communications Magazine*, 55(3), 22-28. <https://doi.org/10.1109/MCOM.2017.1600238CM>
- Millington, I., & Funge, J. (2016). *Artificial intelligence for games* (2nd ed.). CRC Press.

- Octavian, F., & Hermawan, L. (2023). Penerapan algoritma pathfinding A* dalam game Dual Legacy berbasis Android. *Jurnal Buana Informatika*, 14(01), 20-29. <https://doi.org/10.24002/jbi.v14i01.6928>
- Pinter, D. (2020). Implementation of behavior tree in Unity 3D game engine. *International Journal of Game Development*, 6(2), 45-52.
- Rabin, S. (2015). *Game AI Pro 2: Collected wisdom of game AI professionals*. CRC Press. <https://doi.org/10.1201/b18373>
- Russell, S. J., & Norvig, P. (2020). *Artificial intelligence: A modern approach* (4th ed.). Pearson.
- Sharma, M., Ontañón, S., Mehta, M., & Ram, A. (2022). Drama management and player modeling for interactive fiction games. *Computational Intelligence*, 22(2), 123-134. <https://doi.org/10.1111/j.1467-8640.2006.00204.x>
- Togelius, J., & Yannakakis, G. N. (2016). *Artificial intelligence and games*. Springer.
- Tozour, P. (2024). Search algorithms for AI. In S. Rabin (Ed.), *AI game programming wisdom 2* (pp. 123-133). Charles River Media.
- Widodo, W., & Ahmad, I. (2017). Penerapan algoritma A Star (A*) pada game petualangan labirin berbasis Android. *Khazanah Informatika: Jurnal Ilmu Komputer dan Informatika*, 3(2), 57-63. <https://doi.org/10.23917/khif.v3i2.5221>