

PERANCANGAN GAME 3D MAGGIE'S ODYSSEY DENGAN METODE PATHFINDING

Fadel Zufar Ardana, Ahmad Fahrudi Setiawan, Febriana Santi Wahyuni

Teknik Informatika, Institut Teknologi Nasional Malang
Jalan Raya Karanglo km 2 Malang, Indonesia
fadelardana08@gmail.com

ABSTRAK

Industri *game* semakin berkembang pesat dan semakin banyak peminatnya. Hal ini ditandai dengan semakin banyaknya *game* yang bermunculan dengan berbagai *genre* dan *platform* yang berbeda. Salah satu *genre game* yang populer adalah *game* petualangan (*adventure*), yang biasanya memiliki alur cerita yang menarik dan banyak tantangan yang harus dihadapi oleh pemain. Dalam penelitian ini, akan dibuat *game* petualangan 3D yang menarik dan menantang dengan memiliki alur cerita yang menarik dan tantangan yang berbeda pada setiap *level*-nya. Berbicara tentang pengembangan *game* 3D, salah satu aspek penting adalah kemampuan AI dalam menavigasi lingkungan *game* dengan cerdas dan efisien. Metode *pathfinding* merupakan salah satu teknik dalam pengembangan *game* yang digunakan untuk mencari jalur terbaik antara dua titik pada peta *game*. *Game* yang akan dirancang adalah *game* 3D berjudul "Maggie's Odyssey" dengan metode *pathfinding* sebagai fokus utama pengembangan. Dari simulasi pengujian *pathfinding* menunjukkan berhasil dengan rumus perhitungan dan sistem dapat menemukan jalur terbaik dengan memilih nilai $f(n)$ terkecil dari beberapa iterasi yang telah dilakukan sistem dengan hasil rute yang ditempuh. Hasil pengujian user didapat 24.2% menyatakan Sangat Baik, 46.7% menyatakan Baik, 22.5% menyatakan Cukup, 4.2% menyatakan Buruk dan 0.8% menyatakan Sangat Buruk.

Kata kunci : 3D, Game, Pathfinding

1. PENDAHULUAN

Perkembangan *game* masa kini semakin pesat dan semakin banyak peminatnya. Hal ini ditandai dengan semakin banyaknya *game* yang bermunculan dengan berbagai *genre* dan *platform* yang berbeda. Salah satu *genre game* yang populer adalah petualangan (*adventure*), yang biasanya mempunyai alur cerita yang menarik dan banyak tantangan yang harus dihadapi oleh pemain. Dalam *game* petualangan, pemain harus menyelesaikan berbagai rintangan, mengalahkan musuh dan mencapai tujuan akhir. Namun, untuk menciptakan *game* petualangan yang menarik dan menantang, diperlukan sebuah sistem yang mampu mengatur jalannya permainan dengan baik.

Dalam penelitian ini, akan dibuat *game* petualangan 3D yang menarik dan menantang dengan memiliki cerita yang menarik serta tingkat tantangan yang berbeda pada setiap *level*-nya. Berbicara tentang pengembangan *game* 3D, salah satu aspek penting adalah kemampuan AI dalam menavigasi lingkungan *game* dengan cerdas dan efisien. Salah satu metode yang dapat digunakan untuk memperbaiki kemampuan navigasi AI adalah metode *pathfinding*. Metode *pathfinding* merupakan salah satu teknik dalam pengembangan *game* yang digunakan untuk mencari jalur terbaik antara dua titik pada peta *game*. *Game* yang akan dirancang adalah *game* 3D berjudul "Maggie's Odyssey" dengan metode *pathfinding* sebagai fokus utama pengembangan. Metode *pathfinding* dipilih karena peneliti telah menyesuaikan atau aturan spesifik untuk menjadikan *pathfinding* efisien dalam konteks *game* yang akan dibuat dibandingkan menggunakan metode lainnya.

Oleh karena itu, dibuatnya *game* "Maggie's Odyssey" dengan metode *pathfinding* bukan hanya

tentang menciptakan permainan yang lebih efisien tetapi juga tentang berkontribusi pada pengembangan industri *game* dan teknologi AI yang lebih canggih. Selain itu, hasil penelitian ini juga diharapkan dapat memberikan *insight* bagi pengembang *game* awam dalam menggunakan teknik *pathfinding* pada *game* petualangan.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Pada studi yang berjudul "Implementasi Metode Pathfinding dengan Algoritma A* pada Game Rogue-like menggunakan Unity" yang bertujuan menerapkan kecerdasan buatan pada permainan untuk meningkatkan interaktivitas dan pengalaman pengguna. Algoritma A* digunakan untuk mencari jalur dengan cepat dan efisien, hasil dari studi ini adalah pergerakan unit dalam permainan Rogue-Like dapat melewati rintangan menuju tujuan akhir [1].

Pada studi yang berjudul "Implementasi Algoritma Collision Detection dan A*(A Star) pada Non Player Character Game World Of New Normal " menggunakan Algoritma A* untuk menemukan rute terpendek yang bisa dilalui oleh karakter *non-player* dalam mencapai targetnya. Hasil studi menunjukkan bahwa karakter *non-player* sukses menemukan rute terpendek menuju pemain dengan Algoritma A*. Algoritma *Collision Detection* digunakan untuk mendeteksi tumbukan antara pemain dan objek lain, serta musuh. Pengujian dengan melibatkan 50 responden dalam permainan menunjukkan tingkat kepuasan yang cukup memuaskan, mencapai persentase 81% [2].

Pada studi yang berjudul "Implementasi Algoritma A* (A Star) Pada Game 3D Kebudayaan Suku Minahasa", yang bertujuan merancang *game* 3D

budaya Minahasa dengan menerapkan algoritma A* yang diterapkan pada NPC musuh. Hasil yang diperoleh dari algoritma A* (Star) yang diimplementasikan pada NPC dan hasil pengujian penggunaan *game* yaitu 77,69 masuk dalam kategori Baik. Hasil pengujian dengan 7 responden yang terdiri dari orang yang bermain atau terbiasa bermain sebesar 83,99% dengan kategori Sangat Baik [3].

2.2. Game

Game adalah suatu produk dari teknologi informasi yang sangat populer saat ini, *game* adalah bentuk aplikasi yang interaktif. Dengan kata lain, *game* bersifat menghibur dan dapat memberikan berbagai tantangan kepada pemainnya [4].

Game terdiri dari beberapa *genre* diantaranya :

1. Adventure Game

Genre Adventure game adalah gabungan dari komponen *game action* dengan *game adventure*, yang biasanya terdiri dari rintangan jangka panjang yang harus dilewati dengan alat atau *item* sebagai pendukung dalam melewati rintangan, dengan rintangan-rintangan kecil yang akan terus ada [5].

2. Strategy Game

Genre strategy game adalah *game* yang memerlukan strategi dan perhitungan. Mayoritas *game* strategi adalah *game* bernuansa peperangan. Contoh: Warcraft [6].

3. Simulation Game

Genre Simulation game, menggambarkan simulasi dunia di dalamnya yang semirip mungkin dengan gambaran dunia nyata dengan mempertimbangkan detail dari berbagai aspek. Contoh: The Sims, Taxi Simulator dan Bus Simulator [7].

4. Fighting Game

Genre fighting game adalah *game* dengan mekanik bertarung satu dengan lain untuk mengalahkan musuh dengan cara membuat HP bar musuh habis dengan pukulan, tendangan, jurus, dan sebagainya. *Game fighting* dimainkan secara 1 lawan 1 di sebuah arena yang terbilang kecil. Contoh *game* jenis ini adalah seri Tekken, Soul Calibur, BlazeBlue, seri Street Fighter, Dragon Ball Fighter Z [3].

5. Shooting Game

Genre shooting game adalah *game* aksi dengan karakter yang diberikan senjata dan biasanya berguna untuk bertahan hidup dari serangan musuh. Inti dari *game* ini adalah menembak. Inti dari *game shooting* ini adalah menembak. Contoh: GTA, dan Crysis [3].

6. Puzzle Game

Genre puzzle game ini adalah permainan teka-teki dengan pemain yang mengharuskan memecahkan berbagai teka-teki yang disediakan pada *game*. Contoh: Tetris, Minesweeper dan Bejeweled [3].

7. Sport Game

Genre sport game ini merupakan adaptasi olahraga nyata yang memerlukan kelincahan serta strategi dalam memainkannya. Contoh: Winning Eleven, NBA, dan FIFA Street [3].

2.3. Kecerdasan Buatan

Kecerdasan buatan atau *Artificial Intelligence* adalah satu dari banyak ilmu komputer yang mempelajari mekanisme komputer yang dapat melakukan pekerjaan selayak dan sebaik manusia atau bahkan bisa lebih baik dari yang manusia lakukan [8]. Kecerdasan buatan diantaranya adalah :

1. Pathfinding

Pathfinding banyak diimplementasikan dalam pembuatan *game AI (Artificial Intelligence)* misalnya AI diimplementasikan pada NPC (*Non-Player Character*) yang dapat mengejar *player* secara efisien dengan dapat menghindari halangan yang terdapat dalam *game* [3].

Pathfinding terdiri dari beberapa jenis yaitu :

a. A*

Algoritma A* (A Star) digunakan untuk menentukan rute terpendek yang dapat dilalui oleh NPC untuk menuju target dengan mempertimbangkan harga terkecil yang harus diambil [2].

Notasi Metode Algoritma A*:

$$F(n) = G(n) + H(n) \quad (1)$$

keterangan :

$F(n)$ = nilai untuk membandingkan dan memilih jalur terbaik yang diperlukan

$G(n)$ = nilai jarak sebenarnya dari *node* awal ke *node* ke n

$H(n)$ = nilai perkiraan biaya (*cost*) dari *node* n menuju *goal node* (titik target)

b. BFS (Breadth-First Search)

Algoritma BFS menjelaskan proses pemecahan masalah yang harus mengambil setiap keputusan terbaik pada setiap langkah dengan mengunjungi setiap *node* dari pohon yang mungkin menjadi solusi optimal penyelesaian. Alternatifnya, dengan kata lain, pencarian dilakukan dengan mengunjungi *node* per *level* hingga status tujuan terdeteksi. Jika ada solusi, pencarian BFS memastikan penemuan penyelesaian dengan lintasan terpendek [9].

c. DFS (Depth-First Search)

DFS adalah proses yang memastikan bahwa suatu solusi, jika ada, yang mempunyai jumlah cabang pohon terbatas berhingga. Jika solusi ditemukan, maka terdapat jalur yang panjangnya berhingga dari keadaan awal ke keadaan tujuan [10].

d. Dijkstra

Algoritma Dijkstra digunakan untuk menentukan jalur terpendek (*Shortest Path Problem*) pada suatu graf. Algoritma Dijkstra dapat menentukan jarak tercepat dari 2 titik dalam *node* yang berada dalam graf yang memiliki berat yang membedakan jarak antara *node* satu dengan *node* yang lainnya [6].

2. Finite State Machine

Finite State Machine (FSM) merupakan metodologi perancangan sistem kendali yang menunjukkan perilaku sistem dalam menggunakan tiga hal yaitu : *State*(Keadaan), *event* (kejadian) dan *action* (aksi). Dalam periode yang signifikan, sistem akan berada dalam satu *state* yang aktif. Sistem dapat berpindah menuju *state* lainnya jika mendapatkan suatu *event* tertentu yang berasal dari luar atau komponen dalam sistemnya tersebut [11].

3. Rule-Based System (RBS)

Rule-Based System adalah metode pengambilan keputusan yang bergantung pada aturan tertentu dan dapat digunakan dalam agen kecerdasan buatan dalam bentuk virtual yang mampu melakukan tindakan tertentu. Aturan ini mencakup apa yang menyebabkan tindakan tersebut, proses yang terlibat dalam tindakan tersebut, dan hasil dari tindakan tersebut. Karena aturan-aturan ini bersifat tetap, RBS tidak memungkinkan variasi keputusan untuk kasus yang sama. Aturan yang diciptakan untuk satu tindakan mungkin tidak berlaku untuk tindakan lain. Untuk menambahkan jenis tindakan baru, perlu dibuat seperangkat aturan baru, yang dapat meningkatkan kompleksitas dan berpotensi menciptakan konflik antar aturan yang dapat mengakibatkan kegagalan RBS[12].

4. Decision Tree

Decision Tree merupakan sebuah bagan pohon hirarki yang diterapkan menggunakan *Nested if-else*, di mana pada masing-masing kondisi akan diperiksa dan akan diambil keputusan berdasarkan kondisi tertentu. *Decision Tree* sendiri adalah implementasi dari teknik pengelompokan yang merupakan proses penentuan suatu fungsi tujuan yang memetakan masing-masing himpunan *behaviour* dalam satu kelompok dari kelas yang telah didefinisikan sebelumnya. *Decision tree* dapat menemukan relasi antara nilai masukan dengan nilai keluaran[13].

5. Behaviour Tree

Behavior tree difungsikan untuk mengatur perilaku dalam bagan berupa pohon. *Behavior tree* memungkinkan untuk mengontrol perilaku karakter pada *game* yang dapat menjelaskan tingkatan keputusan dan aksi, sehingga dapat disimpulkan bahwa *behavior tree* adalah sebuah Pemodelan yang memiliki struktur berbentuk pohon yang dapat digunakan mengatur perilaku NPC pada sebuah *game*[14].

3. METODE PENELITIAN

3.1. Analisis Pengguna

Analisis pengguna diperlukan untuk mengetahui ketentuan pengguna untuk dapat memainkan *game* "Maggie's Odyssey", diantaranya :

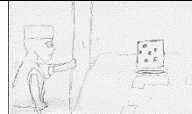
1. Prioritas pengguna berumur minimal 10 tahun sampai 20 tahun karena pada usia tersebut, pengguna terbilang cukup untuk mengoperasikan komputer.
2. Pengguna memiliki perangkat *desktop* atau komputer dan dapat mengoperasikan aplikasi.

3.2. Storyboard

Storyboard ditunjukkan pada tabel 1.

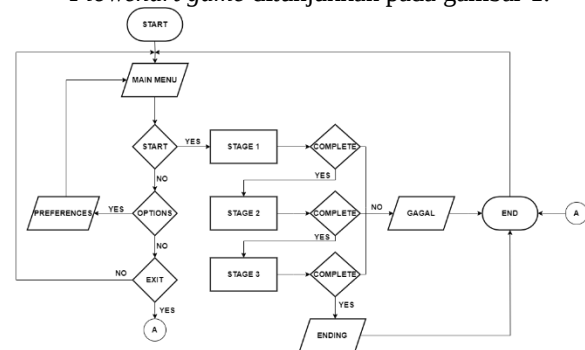
Tabel 1. *Storyboard*

No	Gambar	Keterangan
1		Maggie menemukan sebuah peta kuno yang menunjukkan lokasi harta karun legendaris yang tersembunyi di suatu tempat.

No	Gambar	Keterangan
2		Maggie mengikuti jalan di kota, disepanjang jalan terdapat kaleng untuk membuatnya bertambah kuat
3		Di sepanjang jalan kota, Maggie bertemu dengan robot-robot aneh
4		Maggie melewati Gurun yang penuh bahaya
5		Maggie menghadapi monster yang mengikuti dan menghalangi jalannya
6		Maggie' akhirnya tiba di Gua, yang berbahaya, dengan hati-hati Maggie' berusaha untuk mencapai harta karun yang diincar
7		Maggie' berhasil mendapatkan harta karun berupa patung emas
8		Tiba-tiba Maggie' berada di tempat tidurnya yang mana selama ini sedang bermimpi tentang semua petualangannya tersebut.

3.3. Flowchart Game

Flowchart game ditunjukkan pada gambar 1.



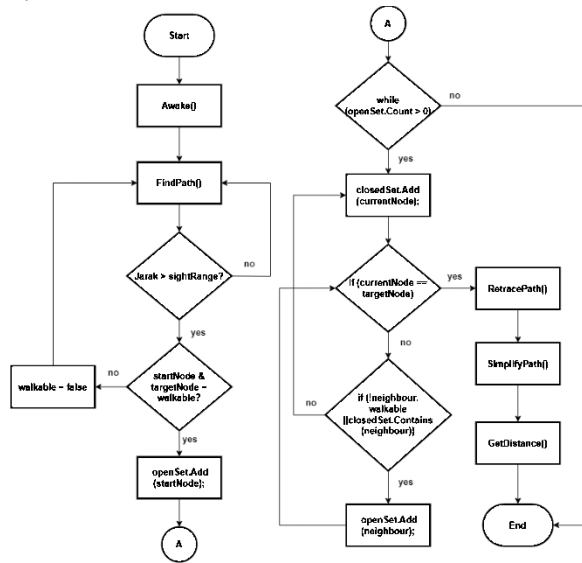
Gambar 1. *Flowchart game*

Flowchart dimulai dengan sistem yang menampilkan tampilan menu utama *game* dengan 3 pilihan. Jika pemain memilih "Start" sistem akan memuat *level* pertama, jika *level* pertama sukses, maka dapat lanjut ke *level* selanjutnya, sampai *level* ketiga. Jika setiap *level* gagal maka sistem memberitahu pemain bahwa *game* selesai karena mati menunjukkan pemain gagal dan *flowchart* selesai, jika kondisi semua *level* sukses, maka sistem akan memberitahu pemain bahwa *game* selesai dengan *ending* yang menunjukkan pemain telah sukses melewati semua *level*. Kembali ke opsi menu terdapat "Options" dan sistem akan menampilkan preferensi *game*. Opsi terakhir adalah "Exit" jika, pemain keluar *game* maka proses selesai dan jika tidak maka akan kembali ke menu utama.

3.4. Flowchart Pathfinding A*

Flowchart pathfinding ditunjukkan pada gambar

2.

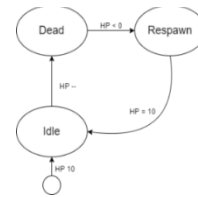


Gambar 2. Flowchart Pathfinding A*

Flowchart dimulai dengan fungsi Awake() yang berfungsi untuk mengambil komponen yang dibutuhkan. Dilanjutkan dengan proses FindPath() yang menerima permintaan pencarian jalur dan fungsi lainnya. Pada FindPath() dilanjutkan kondisi untuk memeriksa apakah jarak antara dua titik melebihi “sightRange”. Pada kondisi no sistem akan terus melakukan proses FindPath(), jika kondisi yes, proses akan berlanjut ke kondisi selanjutnya yaitu apakah startNode & targetNode = walkable, jika kondisi no maka node tersebut tidak dapat dilewati dan harus mencari node lain didekatnya, jika kondisi yes sistem akan melanjutkan proses untuk menambahkan node ke openSet. Lalu terdapat kondisi ketika openSet > 0, jika kondisi no yaitu openSet kosong maka proses akan berhenti karena tidak menemukan node yang akan dilalui, jika kondisi yes atau openSet > 0 terpenuhi maka sistem akan melanjutkan ke node selanjutnya dan menambahkan node sebelumnya ke closedSet. Lalu terdapat kondisi yaitu apakah currentNode adalah targetNode, jika kondisi yes terpenuhi dan node selanjutnya adalah targetNode maka sistem akan melanjutkan proses ke fungsi RetracePath(), SimplifyPath(), GetDistance() dan sistem akan mengakhiri prosesnya setelah rute targetNode telah ditemukan beserta jarak yang ditempuh, jika kondisi no terpenuhi maka akan berlanjut ke kondisi selanjutnya yaitu apakah node tetangga walkable atau sudah ada pada closedSet, kondisi yes terpenuhi maka sistem akan melanjutkan pencarian jalan dan menambahkan node tetangga yang akan dilalui ke openSet, jika kondisi no terpenuhi maka proses akan kembali ke closedSet untuk mengecek node tetangga lainnya.

3.5. Diagram Finite State Machine (FSM)

Diagram FSM karakter ditunjukkan pada gambar 3.



Gambar 3. Diagram FSM karakter utama

Pada diagram FSM di atas merupakan diagram state karakter utama, yang mana state pertama dari karakter adalah Idle dengan HP = 10, jika dari karakter terkena damage dari objek, HP dari karakter akan berkurang hingga menjadi state “Dead” (mati), setelah karakter mati, karakter akan menuju state “Respawn” dan kembali di titik awal dan akan kembali ke state Idle dengan HP = 10.

Diagram FSM enemy ditunjukkan pada gambar 4.

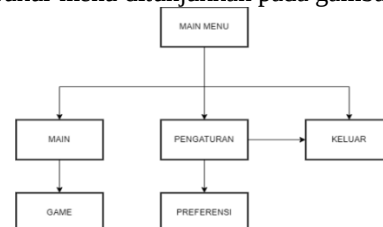


Gambar 4. Diagram FSM enemy

Pada diagram FSM di atas merupakan diagram state NPC yang diberi fungsi pathfinding dan pergerakan yang acak, dengan state pertama dari NPC adalah Patrol/random movement, jika terdapat target (player) dalam jangkauan 10 unit NPC akan mengejar target dengan fungsi pathfinding dan akan menyerang Player, yang akan mengurangi HP karakter player. Setelah pada state Serang, target tidak ada pada jangkauan NPC maka state akan kembali ke Patrol.

3.6. Struktur Menu

Struktur menu ditunjukkan pada gambar 5.



Gambar 5. Struktur Menu

Pada struktur menu, sistem akan menampilkan menu yang terdiri dari menu untuk tampilan awal UI game, menu untuk memulai game yaitu “Main” yang berlanjut ke game, lalu menu pengaturan untuk mengatur beberapa preferensi permainan dan menu keluar untuk menutup game.

4. HASIL DAN PEMBAHASAN

4.1. Pembuatan Menu Utama

Menu utama ditunjukkan pada gambar 6.



Gambar 6 Menu Utama

Pada gambar 6 terdapat tampilan menu utama yang menggambarkan seorang Maggie' yang berada di depan goa melihat harta karun yang dicari. Terdapat menu Main untuk menuju ke pilih *level game*, lalu menu pengaturan yang terdapat preferensi untuk mengatur audio, menghapus data tersimpan serta tentang pengembangan *game*, serta menu keluar dari *game*.

Menu pilih *level* ditunjukkan pada gambar 7.



Gambar 7. Menu Pilih Level

Pada gambar 7 terdapat tampilan menu pilih *level* yang terdiri dari 3 *level*, saat awal memulai *game*, player hanya dapat memainkan *level 1* yang mana *level* lainnya bisa dimainkan setelah *level 1* selesai.

4.2. Pembuatan Karakter

Desain Karakter ditunjukkan pada tabel 2.

Tabel 2. Desain Karakter

No	Gambar	Keterangan
1		Desain karakter Maggie, dengan desain memakai peci, bersetelan hijau, dan bersepatu coklat.
2		Desain karakter monster Olaf, dibuat dengan <i>lowpoly</i> , berkepala plontos, berbaju oranye, dan memiliki, bercak hijau di beberapa bagian tubuh.
3		Desain karakter monster Munn, berwujud mirip babi, memiliki tanduk kecil, berbadan merah muda.
4		Desain karakter monster Elm, berwujud aneh, memiliki tanduk kecil, berbadan oranye, lucu namun cukup berbahaya.
5		Desain karakter robot Gluz, berbentuk kubus, memiliki mata di beberapa sisi dan terdapat sesuatu berbentuk halo untuk menopang tubuhnya yang melayang.

4.3. Pembuatan Level

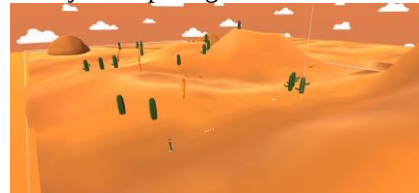
Level 1 ditunjukkan pada gambar 8.



Gambar 8. Level 1

Level 1 pada gambar 13 disusun dengan konsep perkotaan sederhana yang terlupakan. dengan latar waktu siang hari. Terdapat jalan yang menuntun *player* menuju objektif *game*.

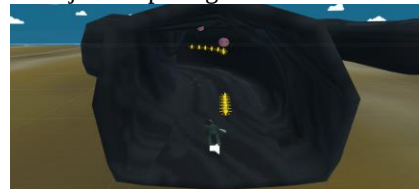
Level 2 ditunjukkan pada gambar 9.



Gambar 9. Level 2

Level 2 pada gambar 14 disusun dengan konsep padang pasir/gurun, yang terlihat luas dan berbukit. Terdapat jamu kaleng di beberapa titik, lalu monster yang disebut Elm di beberapa titik gurun juga batuan besar, serta kaktus dan pohon kering yang menggambarkan ekosistem padang pasir, juga portal yang digunakan untuk berpindah *level*.

Level 3 ditunjukkan pada gambar 10.

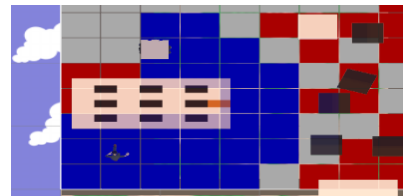


Gambar 10. Level 3

Pembuatan *level 3* pada gambar 4.61 ini disusun dengan konsep goa yang bercabang dan berbahaya, terdapat jamu kaleng, lalu batuan lancip di sekitar permukaan goa, monster kecil yang disebut Munn, dan monster besar Olaf yang menghalangi karakter utama untuk mendapatkan objektif utama yaitu harta karun.

4.4. Pengujian Simulasi Pathfinding

Pengujian *pathfinding* adalah pengujian pencarian rute oleh sistem. Dalam pengujian ini dibuat simulasi jalur yang berbentuk *grid* yang akan dijalankan pada pengujian, *grid* 100 x 100 pixel dengan radius dibuat dekat agar sistem lebih cepat dalam melakukan pencarian. Pengaturan titik awal musuh (x dan y) berada di titik (x=2, y=5) dengan target yaitu player pada titik (x=1, y=1). Musuh / sistem akan melakukan pencarian jalur atau rute yang akan dilalui untuk menuju ke target yaitu *player*.



Gambar 11. Pengujian Simulasi Pathfinding A* Oleh Sistem

Pengujian *pathfinding* oleh sistem dengan visualisasi tetangga yang telah dihitung (*grid* berwarna

biru), dan visualisasi *grid* yang tidak bisa dilewati karena terdapat halangan (*grid* berwarna merah).



Gambar 12. Hasil Rute Terdekat Dari Pengujian

Pengujian simulasi sistem *pathfinding* A* pada NPC ditunjukkan pada tabel 3.

Tabel 3. Pengujian Simulasi Sistem *Pathfinding* A*

ITERASI 1					
No	Node Awal	Tetangga	G(n)	H(n)	F(n)
1	(2,5)	(1,5)	10	40	50
2	(2,5)	(1,6)	14	50	64
3	(2,5)	(2,6)	10	54	64
4	(2,5)	(3,5)	10	48	58
5	(2,5)	(3,6)	14	58	72
ITERASI 2					
1	(1,5)	(0,5)	20	44	64
2	(1,5)	(0,6)	24	54	78
ITERASI 3					
1	(3,5)	(4,5)	20	52	72
2	(3,5)	(4,6)	24	62	86
ITERASI 4					
1	(1,6)	(0,7)	28	64	92
2	(1,6)	(1,7)	24	60	84
3	(1,6)	(2,7)	28	64	92
ITERASI 5					
1	(2,6)	(2,7)	20	64	84
2	(2,6)	(3,7)	24	68	92
ITERASI 6					
1	(4,5)	(5,4)	34	52	86
2	(4,5)	(5,5)	30	56	86
3	(4,5)	(5,6)	34	66	100
ITERASI 7					
1	(3,6)	(4,7)	28	72	100
ITERASI 8					
1	(1,7)	(0,8)	38	74	112
2	(1,7)	(1,8)	34	70	104
3	(1,7)	(2,8)	38	74	112
ITERASI 9					
1	(2,7)	(2,8)	30	74	104
2	(2,7)	(3,8)	34	78	112
ITERASI 10					
1	(5,4)	(5,3)	44	48	92
2	(5,4)	(6,4)	44	62	106
3	(5,4)	(6,5)	48	66	114
ITERASI 11					
1	(5,5)	(6,5)	40	66	106
2	(5,5)	(6,6)	44	70	114
ITERASI 12					
1	(4,6)	(5,7)	38	76	114
ITERASI 13					
1	(5,3)	(5,2)	54	44	98
2	(5,3)	(6,2)	58	54	112
ITERASI 14					
1	(5,2)	(4,1)	68	30	98

2	(5,2)	(5,1)	64	40	104
3	(5,2)	(6,1)	68	50	118
ITERASI 15					
1	(4,1)	(3,0)	82	24	106
2	(4,1)	(3,1)	78	20	98
3	(4,1)	(4,0)	78	34	112
4	(4,1)	(5,0)	82	44	126
ITERASI 16					
1	(3,1)	(2,0)	92	14	106
2	(3,1)	(2,1)	88	10	98
ITERASI 17					
1	(2,1)	(1,0)	102	10	112
Node Awal, Tetangga, G(n), H(n), F(n)					
2	(2,1)	(1,1)	98	0	98

Keterangan :

X = titik koordinat X pada grid

Y = titik koordinat Y pada grid

$G(n)$ = nilai jarak sebenarnya dari *node* awal ke *node n* (dengan asumsi pergerakan horizontal dan vertikal memiliki biaya 10, dan pergerakan diagonal memiliki biaya 14)

$H(n)$ = nilai perkiraan biaya (cost) dari *node n* menuju *goal node* (titik target), dengan rumus :

$$\text{Sumbu } X = |X(\text{nodeA}) - X(\text{nodeB})| \quad (2)$$

$$\text{Sumbu } Y = |Y(\text{nodeA}) - Y(\text{nodeB})| \quad (3)$$

dengan kondisi perhitungan berdasarkan kode :

jika (hasil *Sumbu X* > hasil *Sumbu Y*), maka perhitungan lanjutan adalah =

$$14 * (\text{hasil sumbu } Y) + 10 * (\text{hasil sumbu } X - \text{hasil sumbu } Y) \quad (4)$$

jika (hasil *Sumbu X* < hasil *Sumbu Y*), maka perhitungan lanjutan adalah =

$$14 * (\text{hasil sumbu } Y) + 10 * (\text{hasil sumbu } Y - \text{hasil sumbu } X) \quad (5)$$

$F(n)$ = nilai untuk membandingkan dan memilih jalur terbaik yang diperlukan (dengan kolom berwarna kuning yang menunjukkan jalur terpendek yang telah dihitung oleh sistem)

Pada iterasi 1 diketahui bahwa *node* awal adalah titik dari unit yang akan mencari target, tetangga terdekat dari *node* awal (2, 5) adalah *node* (1, 5), (1, 6), (2, 6), (3, 5), (3, 6) dengan $f(n)$ masing-masing adalah 50, 64, 64, 58, 72. Pada Iterasi 2, tetangga dari *node* dengan $f(n)$ terkecil adalah halangan, jadi sistem tidak memilih rute ini. Pada iterasi 3, didapat nilai terkecil dari tetangga *node* (3, 5) yaitu *node* (4, 5). Pada iterasi 4 nilai $f(n)$ dari *node* terkecil tidak lebih kecil dari *node* (4, 5), sehingga jalan ke *node* pada iterasi 4 tidak dipilih demikian juga pada iterasi 5. Pada iterasi ke 6 *node* (4, 5) memiliki tetangga terdekat dengan nilai $f(n)$ terkecil yaitu *node* (5, 5) dan *node* (5, 4). Pada iterasi 7, 8, dan 9 *node* terkecil tidak dipilih karena tidak lebih kecil dari tetangga lainnya yaitu *node* (4, 5). Pada iterasi 10 didapat nilai terkecil dari tetangga yaitu *node* (5, 3). Pada iterasi 11 dan 12 *node* terkecil tidak dipilih karena nilai $f(n)$ tidak lebih kecil dari tetangga lainnya dari *node* (4, 5) yaitu (5, 3). Pada iterasi 13 didapat *node* dengan $f(n)$ terkecil yaitu *node* (5, 2). Pada iterasi 14 didapat *node* dengan $f(n)$ terkecil yaitu *node* (4, 1). Pada iterasi ke 15 didapat *node* dengan $f(n)$ terkecil yaitu *node* (3, 1). Pada iterasi 16 didapat *node* dengan $f(n)$ terkecil yaitu *node* (2, 1).

Pada iterasi 17 didapat *node* dengan $f(n)$ terkecil yaitu *node* (1, 1) yang merupakan *node* dari target. Dengan kesimpulan hasil rute yang ditempuh yaitu titik (3, 5), (4, 5), (5, 4), (5, 3), (5, 2), (4, 1), (3, 1), dan (2, 1) dengan titik (1, 1) sebagai titik/*node* target.

4.5. Pengujian Control Player

Pengujian *control player* adalah pengujian setiap fungsi dari *keyboard* dan *mouse* yang sudah ditetapkan untuk mengerjakan karakter utama pada *game*. Hasil pengujian *control player* ditunjukkan pada tabel 4.

Tabel 4. Pengujian Control Player

No	Tombol	Aksi	Hasil
1	A	Karakter berjalan ke arah kiri	✓
2	W	Karakter berjalan ke arah kanan	✓
3	S	Karakter berjalan ke arah depan	✓
4	D	Karakter berjalan ke arah belakang	✓
5	Mouse Kiri	Karakter melakukan serangan	✓
6	Spasi	Karakter melompat	✓

Keterangan :

✓ = Berhasil

Total pengujian = 6

Persentase keberhasilan = $(6/6) * 100\% = 100\%$

Pada pengujian *control player* didapatkan hasil bahwa 6 dari 6 tombol pada kontrol karakter telah berfungsi dengan baik dengan persentase keberhasilan 100%.

4.6. Pengujian Fungsional Game

Pengujian fungsional *game* adalah pengujian setiap fungsi dari karakter dan skenario yang telah direncanakan berjalan dengan baik atau tidak. Hasil pengujian fungsional *game* ditunjukkan pada tabel 5.

Tabel 5. Pengujian Fungsional Game

No	Aksi	Hasil
1	Karakter musuh Gluz mengejar <i>player</i> ketika <i>player</i> ada pada jarak tertentu	✓
2	Karakter musuh Elm mengejar <i>player</i> ketika <i>player</i> ada pada jarak tertentu	✓
3	Karakter musuh Munn akan jatuh dari langit-langit goa setelah <i>player</i> mendekat pada jarak tertentu	✓
4	Karakter musuh Olaf mengejar <i>player</i> ketika <i>player</i> ada pada jarak tertentu	✓
5	Karakter Olaf dapat tumbang setelah menginjak jebakan	✓
6	<i>Player</i> menjalankan karakter ke arah yang diinginkan	✓
7	Karakter Maggie' dapat menyerang ke musuh hingga HP berkurang atau habis	✓
8	Tombol UI bekerja dengan normal	✓
9	Karakter Maggie' bisa <i>respawn</i> setelah kehabisan nyawa	✓
10	Tampilan <i>Game Over</i> akan tampil setelah Karakter Maggie' kehabisan nyawa selama 3 kali	✓
11	Item Jamu Kaleng yang bisa meregenerasi beberapa HP <i>player</i> setelah 10 item didapatkan	✓
12	Fungsi hapus data berfungsi dengan baik	X
13	Fungsi volume audio berfungsi dengan baik	✓

Keterangan :

✓ = Berhasil

X = Gagal

Total pengujian = 13

Persentase keberhasilan = $(12/13) * 100\% = 92.31\%$.

Pengujian fungsional dari *game* menunjukkan semua berjalan dengan baik, yaitu 12 dari 13 pengujian pada fungsi hapus data, terdapat kendala pada hal tersebut dan menunjukkan persentase keberhasilan 92.31%.

4.7. Pengujian Device

Pengujian *device* dilakukan untuk mengetahui spesifikasi *device* yang dapat menjalankan *game* Maggie's Odyssey. Hasil pengujian *device* ditunjukkan pada tabel 6.

Tabel 6. Pengujian Device

No	Prosesor	RAM	VGA	Spesifikasi	Hasil
1	Intel Core i3	6 GB	Intel(R) HD Graphics 4000	Rendah	✓
2	Intel(R) Core(TM) i7-6500U	8 GB	Intel(R) HD Graphics 520	Rendah	✓
3	AMD A9-9425 RADEON R5	12 GB	AMD Radeon(TM) R5 Graphics	Rendah	✓

Keterangan :

✓ = Berhasil

Total pengujian = 3

Persentase keberhasilan = $(3/3) * 100\% = 100\%$.

Pengujian *device* menunjukkan bahwa *game* berjalan dengan lancar pada spesifikasi rendah dengan persentase 100%, terdapat keterbatasan objek pengujian pada *device* spesifikasi tinggi sehingga tidak ada pengujian untuk *device* berspesifikasi tinggi.

4.8. Pengujian User

Pengujian *user* dilakukan untuk mengetahui sistem sudah berjalan dengan sesuai atau sebaliknya. Pengujian dilakukan terhadap 20 orang. Hasil pengujian *user* ditunjukkan pada tabel 7.

Tabel 7 Pengujian User

No	Aspek Penilaian	Jawaban				
		Sangat Buruk	Buruk	Cukup	Baik	Sangat Baik
1	Kesan Pertama	0	0	6	7	7
2	Desain Karakter	0	2	3	10	5
3	Desain Grafis	1	1	4	9	5
4	Kualitas Grafik	0	2	5	10	3
5	Musik Latar	0	0	6	10	4
6	Efek Suara	0	0	3	10	5
	Total	1	5	27	56	29

Jumlah pertanyaan : 6

Jumlah user : 20

Faktor pembagi : $6 \times 20 = 120$

Persentase responden ditunjukkan pada tabel 8

Tabel 8 Persentase Responden Pada Pengujian User

No	Persentase	Nilai
1	Persentase user memilih Sangat Baik	$(29 / 120 \times 100\%) = 24.2\%$
2	Persentase user memilih Baik	$(56 / 120 \times 100\%) = 46.7\%$
3	Persentase user memilih Cukup	$(27 / 120 \times 100\%) = 22.5\%$
4	Persentase user memilih Buruk	$(5 / 120 \times 100\%) = 4.2\%$
5	Persentase user memilih Sangat Buruk	$(1 / 120 \times 100\%) = 0.8\%$

Dari perhitungan pada tabel 4.6 yang ditujukan kepada 20 user didapatkan hasil persentase yang menunjukkan bahwa 24.2% menyatakan Sangat Baik, 46.7% menyatakan Baik, 22.5% menyatakan Cukup, 4.2% menyatakan Buruk dan 0.8% menyatakan Sangat Buruk.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil dari perancangan dan implementasi pada game 3D Maggie's Odyssey, dapat diambil kesimpulan diantaranya, dari hasil pengujian simulasi didapat hasil bahwa sistem dapat menemukan jalur terbaik dengan memilih nilai $f(n)$ terkecil dari beberapa iterasi yang telah dilakukan seperti dan Tabel 3 dengan hasil rute yang ditempuh yaitu titik (3, 5), (4, 5), (5, 4), (5, 3), (5, 2), (4, 1), (3, 1), dan (2, 1) dengan titik (1, 1) sebagai titik/node target. Pada pengujian control player didapatkan hasil bahwa 6 dari 6 tombol pada kontrol karakter telah berfungsi dengan baik dengan persentase keberhasilan 100%. Pengujian fungsional dari game menunjukkan semua berjalan dengan baik, yaitu 12 dari 13 pengujian pada fungsi hapus data, terdapat kendala pada hal tersebut dan menunjukkan persentase keberhasilan 92.31%. Pengujian device menunjukkan bahwa game berjalan dengan lancar pada spesifikasi rendah dengan persentase 100%, terdapat keterbatasan objek pengujian pada device spesifikasi tinggi sehingga tidak ada pengujian untuk device berspesifikasi tinggi. Hasil pengujian user didapat 24.2% menyatakan Sangat Baik, 46.7% menyatakan Baik, 22.5% menyatakan Cukup, 4.2% menyatakan Buruk dan 0.8% menyatakan Sangat Buruk. Saran yang dapat diambil diantaranya adalah perbaikan performa game, salah satunya adalah render asset pada saat game dimainkan pada kondisi fitur bayangan di aktifkan dan perbaikan pathfinding pada unit NPC di area map tidak rata, dengan cara membuat map dengan permukaan rata.

DAFTAR PUSTAKA

- [1] E. G. Agung, D. Eridani, and A. Fauzi, "Implementasi Metode Pathfinding dengan Algoritma A* pada Game Rogue-like menggunakan Unity," *Indones. J. Comput.*, vol. 7, no. December, pp. 81–94, 2022, doi: 10.34818/indoic.2022.7.3.677.
- [2] D. N. Yohanes and N. Rochmawati, "Implementasi Algoritma Collision Detection dan A* (A Star) pada Non Player Character Game World Of New Normal," *J. Informatics Comput. Sci.*, vol. 3, no. 03, pp. 322–333, 2022, doi: 10.26740/jinacs.v3n03.p322-333.
- [3] K. C. Lamia et al., "Implementasi Algoritma A* (A Star) Pada Game 3D Kebudayaan Suku Minahasa," 2022.
- [4] K. Asyram Naufal, F. Santi Wahyuni, and M. Orisa, "Perancangan Game Duma 3D Menggunakan Metode Finite State Machine," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 7, no. 1, pp. 957–963, 2023, doi: 10.36040/jati.v7i1.6176.
- [5] R. Fantoni, "PENERAPAN METODE FUZZY LOGIC UNTUK PEMBENTUKAN PERILAKU NON PLAYER CHARACTER PADA GAME ADVENTURE LOKI INVATION," 2018.
- [6] M. S. Amin, P. Subarkah, R. Umma, and E. B. Prasetya, "Implementasi Algoritma Dijkstra pada Game Strategi RPG Berbasis Web dengan Framework Javascript P5," *J. Ilm. IT CIDA*, vol. 8, no. 1, p. 41, 2022, doi: 10.55635/jic.v8i1.157.
- [7] A. D. Diasmara, A. W. Mahastama, and A. R. Chrismanto, "Sistem Cerdas Permainan Papan The Battle Of Honor dengan Decision Making dan Machine Learning," *J. Buana Inform.*, vol. 12, no. 2, pp. 136–145, 2021, doi: 10.24002/jbi.v12i2.4905.
- [8] U. Kalsum, "Pengenalan Kecerdasan Buatan (Artificial Intelligence) Kepada Para Remaja," *Procedia Comput. Sci.*, vol. 166, pp. 310–314, 2022, [Online]. Available: <https://www.binadarma.ac.id>
- [9] R. Rahim et al., "Congklak, a traditional game solution approach with breadth first search," *MATEC Web Conf.*, vol. 197, pp. 1–4, 2018, doi: 10.1051/mateconf/201819703007.
- [10] H. D. Hutahaean, "Penerapan Metode Best First Search Pada Permainan Tic Tac Toe," *MEANS (Media Inf. Anal. dan Sist.)*, vol. 2, no. 2, pp. 1–6, 2017, doi: 10.54367/means.v2i1.14.
- [11] A. L. Lie, "PENGEMBANGAN GAME 'THE LAST SAMURAI' DENGAN MENGGUNAKAN METODE FSM DAN FUZZY LOGIC MAMDANI," *J. Mhs. Tek. Inform.*, vol. 2, no. 2, 2018.
- [12] C. Nordiansyah, M. Hariadi, and M. Supeno, "Implementasi Perilaku Agen Berbasis Rule Menggunakan Blender Game Engine," no. 1.
- [13] B. Sugianto and G. P. Utama, "Implementasi Algoritma Pathfinding Dan Decision Tree Dalam Pembuatan Video Game Bergenre Third Person Shooter," *Skanika*, vol. 4, no. 2, pp. 7–14, 2021, doi: 10.36080/skanika.v4i2.1825.
- [14] A. Junaidi, A. Yunus, and A. S. Wiguna, "Implementasi Behavior Tree Pada Perilaku Npc Di Game Sidescroller," *Kurawal - J. Teknol. Inf. dan Ind.*, vol. 4, no. 2, pp. 92–103, 2021, doi: 10.33479/kurawal.v4i2.459.

