

Penerapan Algoritma A* Sebagai Sistem Pencarian Rute Pada NPC Game Labirin

1st Dema Ghifari Auliansyah

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

demaauliansyah@student.telkomuniversity.ac.id

2nd Randy Erfa Saputra

Fakultas Teknik elektro

Universitas Telkom

Bandung, Indonesia

resaputraa@telkomuniversity.ac.id

3rd Ratna Astuti Nugrahaeni

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

ratnaan@telkomuniversity.ac.id

Abstrak - Sarana hiburan yang paling digemari oleh banyak kalangan masyarakat hingga saat ini adalah permainan atau game pada media elektronik seperti *smartphone* ataupun *Personal Computer* (PC). Untuk menciptakan game yang menarik untuk dimainkan, game perlu dibuat serealistik mungkin dengan mengimplementasikan sistem *Artificial Intelligence* (AI) atau kecerdasan buatan pada NPC (*Non-Player Character*) sebagai sistem perilaku dan navigasi dalam game. Salah satu sistem yang umum diketahui adalah *pathfinding algorithm* atau algoritma pencarian rute. *Pathfinding algorithm* merupakan konsep dasar algoritma yang umum digunakan sebagai sistem navigasi karakter dalam sebuah game. Dengan penggunaan *pathfinding algorithm*, NPC dapat bergerak dengan sendirinya setelah diterapkan algoritma tersebut. Hal ini dapat membuat game yang diciptakan akan lebih menarik. Pada penelitian ini, didapatkan hasil perbandingan perhitungan manual dan program implementasi Algoritma A* pada NPC YOLO Maze Game sudah sesuai sehingga, dapat disimpulkan bahwa penerapan Algoritma A* untuk NPC dalam mencari jalur tercepat menemukan Player pada YOLO Maze Game berhasil dilakukan dan berdasarkan hasil kuesioner dengan nilai validitas 0.54 yang berarti valid dan reliabel dengan nilai reabilitas 0.77 terhadap pertanyaan nomor 5 "Bagaimana tingkat kesulitan NPC pada YOLO Maze Game?" didapat 17 responden dari 43 responden atau 39.5% mengatakan bahwa tingkat kesulitan NPC pada YOLO Maze Game sulit.

Kata Kunci: algoritma a*, ai, game, npc, pathfinding

I. PENDAHULUAN

Permainan video (*video game*) merupakan sarana hiburan dalam media elektronik seperti *Personal Computer* (PC) ataupun *smartphone* yang dibuat semenarik mungkin demi memenuhi kepuasan para pemainnya. *Video game* umumnya menyediakan sistem penghargaan seperti skor paling banyak dalam satu waktu yang ditentukan dengan memenuhi persyaratan tertentu yang ada dalam permainan. Jenis yang dihadirkan pun beragam, seperti *arcade*, *race*, *adventure* dan lainnya.

Salah satu genre yang terkenal yaitu *Maze* (Labirin) adalah tipe *game* yang mengharuskan pemain menemukan jalan keluar untuk memenangkan permainan atau pemain perlu mencapai poin tertentu untuk menyelesaikan *game*, adapun tipe *game* labirin yang mengharuskan pemain untuk mengalahkan NPC (*Non-Playable Character*) [1]. *PAC-MAN* yang dibuat oleh Toru Iwatani pada tahun 1980 merupakan salah satu *game* dengan genre labirin yang mengimplementasikan kecerdasan buatan pada NPCnya. Dalam *game* ini, pemain ditempatkan pada labirin yang berisi

titik-titik sebagai tujuan utama para pemain untuk memakan titik-titik tersebut. NPC dalam *game* ini dibuat dalam bentuk *ghost* atau hantu yang didesain menggunakan kecerdasan buatan berupa algoritma pencarian rute dengan tujuan untuk mengejar pemain menggunakan rute tercepat yang ada [2].

Dalam pembuatan *game*, perlu digunakan algoritma yang dapat berfungsi sebagai sistem alur permainan, bantuan, fitur ataupun sistem dalam permainan tersebut. Algoritma A* merupakan salah satu algoritma yang banyak digunakan dalam pembuatan *game* labirin. Dengan penggunaan algoritma tersebut, *game* ini dapat menentukan jalur tercepat yang dapat dilalui.

Walaupun dalam *game* tersebut terdapat rintangan berupa penghalang terhadap rute yang ada, algoritma A* akan mencari jalur tercepat dari titik awal ke titik akhir yang ditentukan [3]. Pada tugas akhir ini akan diimplementasi algoritma pencarian A* sebagai sistem navigasi NPC pada *game*. Penggunaan algoritma A* ini dapat memberikan kecerdasan buatan kepada NPC untuk mencari jalur tercepat 2 dari satu titik ke titik yang lain dengan harapan algoritma A* dapat bekerja dengan optimal.

II. KAJIAN TEORI

A. Artificial Intelligence

Untuk membangun sebuah perangkat yang dianggap cerdas, umum digunakan sistem kecerdasan buatan atau AI (*artificial intelligence*) dalam perangkat tersebut. Kecerdasan buatan merupakan ilmu tentang bagaimana membangun suatu sistem yang dapat menghasilkan kecerdasan yang menyerupai kecerdasan manusia. Sistem ini umumnya dikenal sebagai komputer. Kecerdasan diciptakan dan dimasukkan ke dalam sebuah mesin (komputer) agar dapat melakukan pekerjaan seperti layaknya yang dapat dikerjakan oleh manusia. Untuk *game* dengan kecerdasan buatan, mengacu pada teknik yang digunakan dalam permainan komputer dan video untuk menghasilkan kecerdasan pada perilaku *Non-Player Character* (NPC). Pada *game*, AI bertindak dalam mengendalikan komputer agar dapat bermain melawan pemain manusia [4].

B. NPC (Non-player Character)

Non-player character atau NPC, adalah sebuah objek dalam *game* yang berupa karakter yang tidak dapat dikendalikan oleh pemain, namun bertindak dan melakukan kegiatan yang seolah-olah dikendalikan oleh pemain. Dengan mengimplementasikan kecerdasan buatan pada NPC, maka sebuah *game* akan menjadi lebih menarik untuk dimainkan.

NPC merupakan karakter yang mampu berinteraksi dengan objek didalam dunia *game* dan tindakannya tidak dapat dikendalikan oleh pemain. Tindakan NPC yang tidak dapat diprediksi salah satunya adalah pergerakannya dari suatu tempat ke tempat lain. Dengan diterapkannya AI pada NPC, dapat membuat NPC mengotomasi rute pergerakannya didalam *game* [5]. NPC dari *game* dibuat seakan-akan NPC tersebut mempunyai kecerdasan dan pergerakan sealami mungkin. Keberadaan NPC sendiri dalam suatu *game* merupakan salah satu faktor dan komponen penting dalam sebuah *game* yang dapat menentukan permainan tersebut menjadi menarik atau tidak. Konsep AI merupakan salah satu model yang digunakan dalam membuat NPC. Sifat otonom dari AI merupakan keunggulan dalam memodelkan satu NPC pada *game* [6].

C. Path Finding

Path Finding merupakan sebuah proses untuk mencari jalur antar dua titik dalam sebuah area tertentu. *Pathfinding* dan variasi perkembangannya dari *pathfinding* sudah umum digunakan dalam AI *Game* dan merupakan metode yang paling banyak digunakan pada *game* [7]. Tujuan utama dari teori ini adalah untuk mencari jalur terpendek yang paling optimal untuk digunakan. Terdapat beberapa algoritma pencarian rute yang umum digunakan seperti, Algoritma Dijkstra, Algoritma A*, Algoritma D* dll [8]. Dalam video *game* biasanya terdapat NPC yang didesain untuk bergerak dari satu titik ke titik lain. Sistem navigasi pada NPC ini dapat diterapkan dengan berbagai cara. Salah satu yang umum digunakan yaitu menerapkan sistem pencarian rute atau *path finding* untuk mencari rute antar titik awal dan titik akhir pada NPC tersebut. Penelitian yang dilakukan oleh Rahman dan Wasum mengimplementasikan *pathfinding* pada karakter Hero dengan algoritma A* dalam *game Turn Based Strategy Perang Komando*. Dengan mengimplementasikan algoritma tersebut, salah satu karakter dalam *game* dapat menemukan jalur tercepat dengan *cost* atau biaya seminimal mungkin dari titik awal menuju titik akhir yang ditentukan [9].

D. Fungsi Heuristic

Pada *artificial intelligence*, teknik dasar pemecahan masalah dibagi kedalam beberapa bagian, yaitu *searching*, *reasoning*, *planning*, dan *learning*. Setiap teknik digunakan untuk memecahkan masalah-masalah tertentu misalnya, untuk mencari jalur dapat menggunakan teknik *searching*. Teknik *searching* sendiri dibagi menjadi 2, yaitu *blind search* dan *heuristic search*. Heuristik adalah sebuah teknik yang mengembangkan efisiensi dalam proses pencarian, namun dengan kemungkinan mengorbankan kelengkapan (*completeness*). Fungsi heuristik digunakan untuk mengevaluasi keadaan-keadaan problema individual dan menentukan seberapa jauh hal tersebut dapat digunakan untuk mendapatkan solusi yang diinginkan [4].

E. Manhattan Distance

Manhattan Distance merupakan salah satu fungsi heuristik yang paling sering digunakan dalam menghitung jarak terdekat dari dua titik [10]. Fungsi ini bekerja dengan menghitung selisih koordinat posisi titik awal dan titik tujuan saja. Fungsi Heuristik ini dapat dihitung berdasarkan selisih jarak koordinat yang diambil dari dua buah titik, yang dapat diformulasikan menggunakan rumus (1).

$$h(n) = abs(start.x-goal.x) + abs(start.y-goal.y) \quad (1)$$

Keterangan :

Start.x : Koordinat x titik awal

Start.y : Koordinat y titik awal

Goal.x : Koordinat x titik tujuan

Goal.y : Koordinat y titik tujuan

F. Algoritma A*(star)

Menurut Xiao Cui dan Hao Shi dalam penelitiannya dengan judul A*-Based Pathfinding in Modern Computer, algoritma A* merupakan algoritma pencarian rute yang umum digunakan untuk mencari solusi berbagai masalah. Algoritma A* bekerja dengan mencari titik akhir menggunakan jalur tercepat dengan cara menjelajahi secara berulang-ulang area yang tersedia yang kemudian memungkinkan jalurnya untuk digunakan [11]. Dalam pembuatan *game*, algoritma A* merupakan salah satu yang banyak digunakan pada NPC sebagai sistem pencarian rute tercepat. Seperti penelitian yang dilakukan oleh Athhariq dan Putra yang membahas penentuan pergerakan NPC menggunakan algoritma A* dalam sebuah *game*. Hasil yang didapatkan NPC mampu berjalan menuju pemain sesuai dengan perhitungan algoritma A* [5].

Algoritma A* membutuhkan dua antrean, yaitu OPEN dan CLOSED. OPEN adalah list yang digunakan untuk menyimpan simpul atau *node* yang pernah digunakan dan nilai heuristiknya telah dihitung tetapi belum dipilih sebagai simpul terbaik (*best node*), sedangkan CLOSED adalah list untuk menyimpan simpul-simpul yang tidak mungkin dipilih sebagai simpul terbaik [12].

Secara matematis, fungsi sebagai estimasi fungsi evaluasi terhadap node n dapat dituliskan dengan persamaan (2) berikut:

$$f(n) = g(n) + h(n) \quad (2)$$

f(n): Nilai Heuristic + biaya yang diperlukan.

g(n): Biaya yang diperlukan dari titik awal ke titik n.

h(n): Biaya estimasi dari titik n ke titik tujuan.

Nilai *f(n)* adalah *cost* perkiraan suatu *node* yang teridentifikasi. Fungsi *g(n)*, adalah banyaknya langkah yang diperlukan untuk menuju ke *node* tujuan. Setiap *node* harus memiliki informasi nilai *h(n)*, yaitu estimasi *node cost* yang terhitung dari *node* tujuan yang hasilnya menjadi nilai H. *Node* dengan nilai terendah merupakan solusi terbaik untuk diperiksa pertama kali pada *g(n)+h(n)*. Dengan fungsi heuristik yang memenuhi kondisi tersebut, maka pencarian dengan algoritma A* dapat optimal.

Beberapa penelitian telah dilakukan dengan menerapkan algoritma A* pada sebuah *game* adalah pada tahun 2017 Imam Ahmad dan Wahyu Widodo melakukan penelitian dengan judul Penerapan Algoritma A* pada *Game* Petualangan Labirin Berbasis Android. Dalam penelitian tersebut dibuat sebuah *game* yang memiliki tujuan utama seekor kelinci melewati labirin dengan menemukan jalan tercepat menuju makanannya dengan waktu yang sudah disediakan. Apabila pemain mengalami kesulitan mencari jalur, maka pemain dapat menekan tombol *Help* dan sistem akan membantu pemain dalam mencari jalur tercepat menuju lokasi makanan [13].

Penelitian lain yang berkaitan juga dengan penerapan algoritma A* adalah oleh Rifky Fajar dan Salamun Rohman

pada tahun 2020 dengan judul Implementasi Algoritma A* (star) Dalam Penentuan Rute Terpendek Yang Dapat Dilalui Non-Player Character (NPC) pada Game Good Thief. Dalam penelitian tersebut dibuat sebuah game labirin yang bercerita tentang pencuri baik yang harus mencari uang dari setiap rumah yang ada untuk memenuhi kebutuhan di panti asuhannya. Pada game Good Thief terdapat NPC penjaga yang menerapkan algoritma A* untuk menemukan jalur terpendek yang dapat dilalui penjaga untuk menangkap pencuri baik [14].

III. METODE

Metode yang dilakukan pada penyusunan Tugas Akhir ini meliputi:

A. Studi Pustaka

Pengumpulan dan pemahaman literatur beberapa buku referensi, artikel, jurnal ilmiah, dan sumber lain yang mendukung pengerjaan Tugas Akhir ini.

B. Pemodelan sistem

Berdasarkan studi pustaka, dibuat perancangan aplikasi yang sesuai dengan kondisi yang diinginkan.

C. Implementasi perangkat

Membangun produk yang akan dibuat dan meneliti bagaimana hasil yang didapatkan.

D. Diskusi ilmiah

Berkonsultasi dengan dosen pembimbing, partner tugas akhir dan narasumber lain yang berkompeten untuk mendapatkan pemahaman materi dan teori-teori yang mendukung.

IV. HASIL DAN PEMBAHASAN

Pada pengujian alpha ini akan berfokus pada fungsionalitas aplikasi yang akan dibuat menggunakan teknik pengujian blackbox. Teknik pengujian ini dilakukan terhadap aplikasi untuk memastikan bahwa aplikasi dapat berjalan sesuai dengan kebutuhan dan tujuan yang diharapkan. Skenario pengujian alpha dengan metode pengujian blackbox dapat dilihat pada Tabel 4.1.

TABEL 4.1
TABEL PENGUJIAN ALPHA

Menu yang di uji	Fitur yang di uji	Detail pengujian	Jenis pengujian	Kesi mpulan
Main menu	Tombol Start/Score/Exit	Dapat mengalihkan tampilan kearah sesuai tombol	Blackbox	Berhasil
	Antarmuka	Menampilkan antarmuka	Blackbox	Berhasil
Scoreboard	Antarmuka	Menampilkan antarmuka	Blackbox	Berhasil
	Tombol back	Dapat mengalihkan Kembali tampilan ke main menu	Blackbox	Berhasil

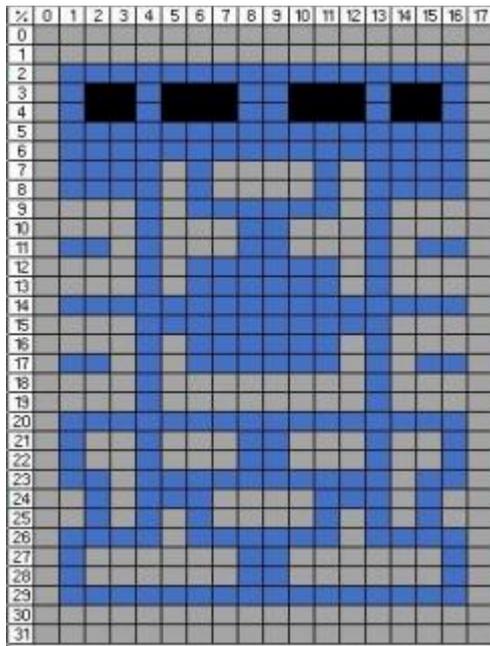
	Hasil score	Dapat menampilkan score	Blackbox	Berhasil
Game	Karakter pemain	Dapat berjalan sesuai input dari keyboard	Blackbox	Berhasil
	Karakter NPC	Dapat berjalan sesuai algoritma	Blackbox	berhasil
	Tampilan antarmuka menang permainan	Tampilan menang muncul saat pemain berhasil mengumpulkan poin	Blackbox	Berhasil
	Tombol replay dan menu di tampilan menang	Tombol replay dapat mengulang Kembali permainan dan tombol menu mengarahkan Kembali ke main menu	Blackbox	Berhasil

Berdasarkan hasil pengujian alpha yang telah didapatkan berdasarkan tabel diatas, dapat dihitung akurasi keberhasilan pengujian blackbox dengan persamaan (3) sebagai berikut.

$$\begin{aligned}
 \text{Hasil Pengujian} &= \frac{\text{Jumlah pengujian berhasil}}{\text{Total pengujian}} \times 100\% \quad (3) \\
 &= \frac{10}{10} \times 100\% \\
 &= 100\%
 \end{aligned}$$

B. Pengujian Algoritma A*(star)

Di bawah ini merupakan pengujian akurasi algoritma A* yang diterapkan pada game untuk menguji perangkat lunak yang sudah dibangun sudah berjalan seperti yang diharapkan atau belum. Pengujian berdasarkan algoritma A* yang telah diterapkan pada NPC dalam game sebagai sistem pencarian rute dengan pemain sebagai targetnya. 18 Untuk menghitung nilai akurasi sistem ini, dilakukan uji coba sebanyak 4 kali pada sistem yang telah dibangun, setiap uji coba akan dilakukan dengan penggunaan rute yang berbeda terhadap NPC dalam mengejar player. Perhitungan algoritma A*(dilampirkan) menggunakan Manhattan Distance untuk mencari jalur terpendek antara dua buah simpul horizontal dan vertikal.



GAMBAR 4.1
GRID YANG DIGUNAKAN PADA GAME.

Gambar 4.5 merupakan papan grid yang digunakan dalam game dengan ukuran 18x32 pixel dengan biaya dari titik awal menuju titik n adalah 1 pixel = 1 node ke arah vertikal atau horizontal. Nilai h(n) atau heuristic di dapat dari jarak perkiraan dari node awal ke node tujuan yang dihitung berdasarkan selisih jarak koordinat yang diambil dari dua buah titik, yang dapat di formulasikan menggunakan persamaan (2).

3			Sesuai
4			Sesuai

Dari hasil pengujian pada Tabel 4.2 di atas, dapat disimpulkan bahwa implementasi algoritma A* berhasil di implementasikan dan sesuai dengan teori perhitungan dari algoritma A*.

V. KESIMPULAN

Setelah dilakukan pengujian berupa pengujian perhitungan algoritma A* dan pengujian Beta berupa kuesioner kepada masyarakat umum dan demo game yang telah dibuat, maka dapat diambil kesimpulan sebagai berikut:

1. Berdasarkan hasil perbandingan perhitungan manual dan program implementasi Algoritma A* pada NPC YOLO Maze Game sudah sesuai sehingga, dapat disimpulkan bahwa penerapan Algoritma A* untuk NPC dalam mencari jalur tercepat menemukan player pada YOLO Maze Game berhasil dilakukan.
2. Dari hasil kuesioner dengan nilai validitas 0,54 yang berarti valid dan reliabel dengan nilai reliabilitas 0,77 terhadap pertanyaan nomor 5 “Bagaimana tingkat kesulitan NPC pada YOLO Maze Game ?” didapat 17 responden dari 43 responden atau 39.5% mengatakan bahwa tingkat kesulitan NPC pada YOLO Maze Game sulit.

REFERENSI

[1] F. Badri dan M. F. Al Habib, “IMPLEMENTASI ALGORITMA A* (A Star) PADA NPC (NON-PLAYABLE CHARACTER) GAME PACMAN MENGGUNAKAN GAME ENGINE UNITY 5 BERBASIS ANDROID,” Teknik: Engineering and Sains Journal, pp. 49-56, 2020.

[2] F. H. Putra, S. M. Nasution, S. T., M. T. and R. A. Nugrahaeni, S. T., M. T., "COMPARISON OF A* ALGORITHM AND TIME BOUNDED A* ALGORITHM ON MAZE CHASE GAME NPC," International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), pp. 79- 48, 2019.

Nomor uji	Jalur pada game	Jalur berdasarkan teori perhitungan A*	Hasil
1			Sesuai
2			Sesuai

- [3] B. T. D. Irianto, S. Andryana and A. Gunaryati, "PENERAPAN ALGORITMA A-STAR DALAM MENCARI JALUR TERCEPAT DAN PERGERAKAN NONPLAYER CHARACTER PADA GAME PETUALANGAN LABIRIN TECH-EDU," JURNAL MEDIA INFORMATIKA BUDIDARMA, vol. 5, no. 3, pp. 953-962, 2021.
- [4] E. Junanto, A. B. Osmond, S.T., M.T. and A. S. R. Ansori S.T., M.T., "MEMBUAT PERGERAKAN NON-PLAYER CHARACTER (NPC) MENGGUNAKAN METODE A STAR," e-Proceeding of Engineering, vol. 7, p. 1491, 2020.
- [5] A. and D. A. Putra, "Penentuan Pergerakan Non-Player Character Menggunakan Algoritma a* Pada Game Action Role-Playing Game," Jurnal Infomedia, vol. 2, pp. 35-40, 2017.
- [6] J. Pasaribu, A. B. Osmond and R. E. Saputra, "PENGEMBANGAN PERILAKU KARAKTER LALAT PADA GAME," e-Proceeding of Engineering, vol. 6, no. 2, pp. 5715-5722, 2019.
- [7] P. Cowling, M. Buro, M. Bida, A. Botea, B. Bouzy, M. Butz, P. Hingston, H. Muñoz-Avila, D. Nau and M. Sipper, "Search in Real-Time Video Games," Dagstuhl Research Online Publication Server, vol. 6, pp. 1-19, 2013.
- [8] I. Zarembo and S. Kodors, "Pathfinding Algorithm Efficiency Analysis in 2D Grid," in Proceedings of the 9th International Scientific and Practical Conference., Rēzekne, 2013. 29
- [9] B. Rahman and W. Wasum, "Implementasi Pathfinding pada karakter Hero dengan Algoritma A* dalam Game TBS Perang Komando," Seminar Nasional FST, vol. 2, pp. 40-44, 2019. [10] S. J. Russel and P. Norvig, Artificial Intelligence : A Modern Approach Second Edition, New Jersey: Pearson Education, Inc., 2003.
- [11] X. Cui and H. Shi, "A*-based Pathfinding in Modern Computer Games," IJCSNS International Journal of Computer Science and Network Security, vol. 11, no. 1, pp. 125-130, 2011. [12] A. Setyawan, P. Harsadi and S. Siswanti, "Penerapan Pathfinding Menggunakan Algoritma A* Pada Non Player," Jurnal Ilmiah Sinus (JIS), vol. 17, no. 2, pp. 39-50, 2019.
- [13] I. Ahmad dan W. Widodo, "Penerapan Algoritma A*(star) pada Game Petualangan Labirin Berbasis Android," Khazanah Informatika, vol. 3, no. 2, pp. 57-63, 2017.
- [14] R. F. Oktanugraha and S. R. Nudin, "Implementasi Algoritma A*(star) Dalam Penentuan Rute Terpendek Yang Dapat Dilalui Non-Player Character Pada Game Good Thief," JINACS, vol. 2, no. 1, pp. 74-85, 2020.