

Meningkatkan Efisiensi Pencarian Jalur Pada Game Berbasis Tile Menggunakan Algoritma A* Dengan Penambahan Parameter Heuristik

Improving Path Search Efficiency In Tile-Based Games Using The A Algorithm With The Addition Of Heuristic Parameters*

Zaidan Rahadian Huda
Universitas Dian Nuswantoro Semarang
E-mail: 111202012899@mhs.dinus.ac.id

Abstrak

Algoritma A* merupakan algoritma populer yang digunakan dalam pencarian jalur karena keefektifannya dalam menemukan rute terpendek. Dalam lingkungan seperti game terutama game berbasis *tile*, keberadaan rintangan atau jebakan dapat memengaruhi efisiensi jalur yang diambil. Penelitian ini mengusulkan modifikasi pada fungsi heuristik A* dengan menambahkan nilai penalti ketika algoritma mendeteksi adanya rintangan atau jebakan di sekitar jalur. Tujuan dari modifikasi ini adalah untuk meningkatkan efisiensi serta menghasilkan jalur yang lebih aman dan realistis bagi karakter dalam permainan. Pengujian dilakukan pada delapan jenis lingkungan berbeda yang memiliki tingkat kerumitan dan rintangan bervariasi. Hasil penelitian menunjukkan bahwa modifikasi heuristik pada algoritma A* mampu menghasilkan jalur yang lebih aman, serta meningkatkan efisiensi dalam sebagian besar skenario. Meskipun pada satu lingkungan hasil yang diperoleh serupa dengan algoritma A* standar, algoritma yang dimodifikasi tetap menunjukkan keunggulan dari segi pemilihan jalur yang lebih aman dan sesuai konteks permainan. Dengan demikian, pendekatan ini dapat menjadi alternatif yang efektif untuk implementasi *pathfinding* dalam pengembangan game berbasis *tile*.

Kata kunci: Algoritma A*, Heuristik, Pencarian Jalur, Modifikasi Heuristik, Game Berbasis Tile

Abstract

The A algorithm is a popular pathfinding algorithm known for its effectiveness in finding the shortest route. In environments such as games especially tile-based games the presence of obstacles or traps can affect the efficiency of the chosen path. This study proposes a modification to the A* heuristic function by adding a penalty value when the algorithm detects obstacles or traps near the path. The goal of this modification is to improve efficiency and generate safer and more realistic routes for in-game characters. Testing was conducted on eight different types of environments with varying levels of complexity and obstacles. The results of the study show that the heuristic modification to the A* algorithm is capable of producing safer paths and improving efficiency in most scenarios. Although in one environment the results were similar to the standard A* algorithm, the modified version still demonstrated advantages in selecting safer and more contextually appropriate routes. Therefore, this approach can serve as an effective alternative for implementing pathfinding in tile-based game development.*

Keywords: A* Algorithm, Heuristic, Pathfinding, Heuristic Modification, Tile-Based Game

1. PENDAHULUAN

Game berbasis *tile* merupakan jenis permainan yang memodelkan dunia permainan dalam bentuk grid atau petak-petak kecil yang disebut *tile*. Setiap *tile* merepresentasikan bagian dari lingkungan seperti tanah, rumput, atau tembok. Sistem ini telah banyak digunakan pada game 2D

maupun 3D karena memberikan kemudahan dalam perancangan serta optimalisasi performa[1]. Beberapa game populer seperti *Pokémon*, *Terraria*, *Stardew Valley*, dan *Undertale* menerapkan sistem ini. Namun, kelemahan dari sistem berbasis tile adalah gerakan karakter yang kaku dan terbatas hanya pada satuan tile, sehingga terlihat kurang alami[2].

Pathfinding merupakan proses pencarian rute terbaik antara dua titik, yang bertujuan untuk membantu karakter dalam game bergerak secara logis dan efisien[3]. Game seperti *Warcraft III*, *Age of Empires*, dan *The Sims* menggunakan sistem pathfinding untuk mengatur pergerakan karakter. Meski demikian, pendekatan pathfinding konvensional sering kali hanya mempertimbangkan jarak, tanpa memperhitungkan medan sulit atau area berbahaya, yang dapat menyebabkan jalur tidak optimal atau terjadinya kemacetan dalam permainan.

Algoritma A* adalah salah satu algoritma pencarian jalur yang umum digunakan karena menggabungkan pencarian graf dengan estimasi heuristik untuk menemukan jalur terpendek secara efisien[4,5]. Dibandingkan dengan algoritma seperti Dijkstra, A* dikenal lebih efisien[6]. Penelitian sebelumnya menunjukkan bahwa modifikasi pada fungsi heuristik dapat meningkatkan efisiensi algoritma sebesar 10–13% dan mempersingkat jalur hingga 2,5–3%[7]. A* juga mampu beradaptasi dengan elemen dinamis dalam permainan seperti jebakan atau rintangan[8], sehingga dapat mengarahkan karakter melalui jalur yang lebih aman.

Penelitian ini mengusulkan modifikasi fungsi heuristik pada algoritma A Star dengan menambahkan nilai penalti pada node yang mengandung bahaya, seperti jebakan, dalam lingkungan berbasis tile. Tujuan dari modifikasi ini adalah untuk meningkatkan efisiensi dan keselamatan jalur pergerakan karakter pemain, sehingga menciptakan pengalaman bermain yang lebih cerdas dan realistis.

2. METODE PENELITIAN

2.1 Analisis Masalah

Permasalahan utama dalam pencarian jalur pada lingkungan permainan berbasis tile adalah bagaimana menentukan jalur yang tidak hanya optimal dari segi jarak, namun juga mempertimbangkan aspek keamanan seperti jebakan atau bahaya di sepanjang rute. Algoritma A* banyak digunakan karena efisiensinya, tetapi dalam bentuk standarnya, tidak mempertimbangkan risiko atau penalti dari tile berbahaya.

Penelitian ini memodifikasi fungsi heuristik dari algoritma A* dengan menambahkan penalti pada node yang bersifat berbahaya. Tujuan modifikasi ini adalah menghasilkan jalur yang cenderung lebih aman bagi karakter, meskipun berpotensi lebih panjang dibanding jalur terpendek.

2.2 Instrumen penelitian

Eksperimen dilakukan pada perangkat komputer dengan sistem operasi Windows 10, prosesor Intel Core i7-10700F, RAM 32 GB, dan GPU NVIDIA GeForce GTX 1660 SUPER. Pengembangan sistem menggunakan Unity versi 6000.0.39f dengan bahasa pemrograman C# melalui Visual Studio Code. Lingkungan pengujian dibangun menggunakan Unity 2D Tilemap, dan algoritma yang digunakan adalah A* dengan modifikasi pada heuristik berdasarkan keberadaan tile jebakan. Visualisasi jalur ditampilkan secara langsung melalui tilemap untuk mempermudah analisis.

2.3 Arsitektur Sistem

Sistem dibangun menggunakan game engine Unity untuk mensimulasikan lingkungan berbasis tile. Dengan memanfaatkan Unity 2D Tilemap, desain lingkungan dapat dibuat secara efisien dan fleksibel. Komponen utama dalam sistem terdiri dari beberapa elemen penting: (1) Tilemap sebagai representasi peta grid, di mana setiap tile memiliki status tertentu seperti titik awal (start), jalan, jebakan, tembok, dan tujuan; (2) Karakter pemain, yaitu entitas yang bergerak

dari titik awal menuju tujuan berdasarkan jalur yang dihasilkan; dan (3) Sistem pathfinding yang mengimplementasikan algoritma A* standar serta versi yang telah dimodifikasi.

2.4 Perumusan algoritma

2.4.1 Algoritma A* Standar

Rumus pada algoritma A* menggunakan fungsi sebagai berikut:

$$f(n) = g(n) + h(n) \quad (1)$$

$g(n)$ adalah biaya dari node awal ke node n . $h(n)$ adalah estimasi biaya dari node n ke tujuan (heuristik).

Untuk tile grid 8 arah, digunakan rumus *Diagonal Distance* atau *Octile Distance*:

$$h(n) = D(dx + dy) + (D2 - 2 \times D) \times \min(dx, dy) \quad (2)$$

Untuk menghitung nilai heuristik $h(n)$ pada lingkungan grid 8 arah, digunakan rumus *Octile Distance* yang memperhitungkan gerakan ortogonal dan diagonal. Nilai dx dan dy masing-masing merepresentasikan selisih absolut antara posisi saat ini dengan posisi tujuan pada sumbu x dan y , atau ditulis sebagai $dx = |x_{\text{current}} - x_{\text{goal}}|$ dan $dy = |y_{\text{current}} - y_{\text{goal}}|$. Nilai D merupakan bobot untuk gerakan ortogonal, yang dalam penelitian ini ditetapkan sebesar 10, sedangkan $D2$ adalah bobot untuk gerakan diagonal yang bernilai 14. Rumus ini memungkinkan estimasi jarak yang lebih realistis dibandingkan Manhattan Distance dalam lingkungan 8 arah, karena mempertimbangkan kemungkinan pergerakan diagonal secara efisien.

2.4.1 Algoritma A* Modifikasi

Dalam penelitian ini, modifikasi dilakukan dengan menambahkan penalti ke dalam perhitungan $g(n)$ apabila node tetangga merupakan tile jebakan.

$$f(n) = (g(n) + P) + h(n) \quad (3)$$

P adalah nilai penalti jika node n merupakan tile jebakan. Dengan cara ini, jalur yang dihasilkan akan cenderung menjauhi tile yang memiliki risiko tinggi meskipun jaraknya lebih jauh. Jenis penalti yang diterapkan dibedakan berdasarkan tingkat bahaya jebakannya. Jebakan 1 penambahan biaya sebesar 80. Sedangkan jebakan 2 penambahan biaya sebesar 100.

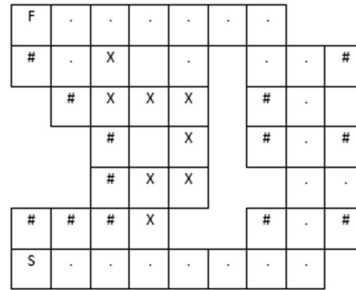
2.5 Implementasi Sistem

Implementasi dilakukan menggunakan Unity Engine dan bahasa pemrograman C#. Proses pencarian jalur dilakukan setiap kali pemain memilih tujuan di peta. Sistem *pathfinding* akan melakukan langkah-langkah sebagai berikut: mengidentifikasi node awal dan tujuan, mengeksekusi algoritma A* standar dan A* modifikasi secara terpisah, menghasilkan visualisasi jalur di tilemap, serta mencatat metrik evaluasi berupa jumlah langkah, dan jumlah jebakan yang dilalui.

2.6 Lingkungan Pengujian

Pengujian dilakukan pada delapan jenis lingkungan berbeda yang dibangun menggunakan 2D *tilemap*. Setiap lingkungan dirancang dengan tingkat kesulitan yang meningkat secara bertahap, mulai dari yang paling sederhana hingga yang paling kompleks. Peningkatan kesulitan tersebut ditandai dengan penambahan jumlah jebakan serta jalur yang semakin rumit untuk dilalui oleh karakter.

Salah satu contoh denah pada lingkungan dapat dilihat seperti berikut.



Gambar 1 Denah Lingkungan

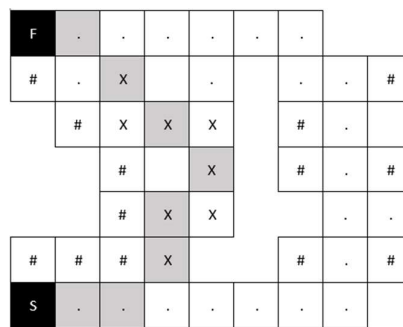
S menunjukkan posisi *Start* (awal), dan F menunjukkan *Finish* (tujuan). Karakter titik (.) merepresentasikan jalur yang dapat dilalui, sedangkan # adalah tembok atau rintangan yang tidak bisa dilewati. Simbol X digunakan untuk menandai tile jebakan yang memberikan penalti terhadap jalur jika dilewati oleh karakter.

2.7 Parameter Evaluasi

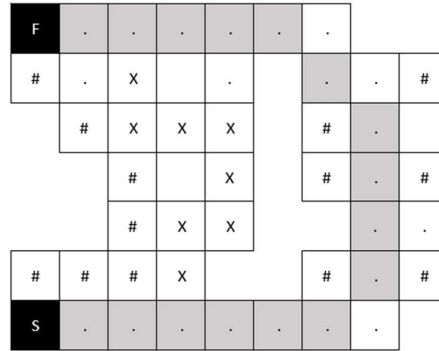
Parameter yang digunakan untuk mengevaluasi performa algoritma terdiri dari dua aspek utama. Pertama, panjang jalur yang dihasilkan, yaitu total jumlah langkah yang harus ditempuh dari titik awal hingga titik tujuan. Kedua, jumlah jebakan yang dilewati dalam jalur tersebut, yaitu seberapa banyak tile berbahaya yang termasuk dalam rute akhir. Kedua parameter ini digunakan untuk menilai tidak hanya efisiensi jalur secara jarak, tetapi juga tingkat keamanannya terhadap potensi bahaya di lingkungan permainan.

3. HASIL DAN PEMBAHASAN

Pengujian dilakukan pada delapan jenis lingkungan berbeda yang dibangun menggunakan 2D Tilemap. Setiap lingkungan memiliki tingkat kesulitan yang meningkat, ditandai dengan penambahan jumlah jebakan serta meningkatnya kompleksitas jalur yang dilalui. Salah satu lingkungan yang dipilih sebagai sampel adalah Map 2, dengan visualisasi jalur hasil pencarian oleh kedua algoritma ditunjukkan pada Gambar 2 dan Gambar 3.



Gambar 2 Denah Map 2 dengan Algoritma Standar



Gambar 3 Denah Map 2 dengan Algoritma Modifikasi

Berdasarkan Gambar 2 dan Gambar 3, terlihat bahwa pada denah lingkungan Map 2, algoritma A* standar mampu mencapai titik tujuan (finish) dengan jarak tempuh yang lebih pendek, namun harus melewati lebih banyak tile jebakan dibandingkan dengan algoritma A* yang telah dimodifikasi. Hal ini menunjukkan bahwa modifikasi heuristik berhasil mengarahkan pencarian jalur ke arah yang lebih aman, meskipun harus menempuh jarak yang lebih jauh.

Perbandingan lebih lengkap dapat dilihat pada Tabel 1, yang mencantumkan hasil pengujian pada tiga lingkungan berbeda (Map 2, Map 5, dan Map 8).

Tabel 1 Perbandingan Algoritma A* standar dengan Algoritma A* modifikasi

Lingkungan	Algoritma	Langkah	Jebakan Dilewati
Map 2	A* Standar	9	5
	A* Modifikasi	17	0
Map 5	A* Standar	14	7
	A* Modifikasi	26	2
Map 8	A* Standar	28	11
	A* Modifikasi	36	6

Dari Tabel 1, dapat disimpulkan bahwa algoritma A* modifikasi secara konsisten mampu menghasilkan jalur yang lebih aman dengan jumlah jebakan yang lebih sedikit, walaupun harus menempuh lebih banyak langkah dibandingkan algoritma A* standar. Hal ini menunjukkan keberhasilan modifikasi heuristik dalam mengarahkan pencarian jalur untuk menghindari bahaya.

4. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian, modifikasi pada heuristik algoritma A* terbukti mampu meningkatkan efisiensi pencarian jalur dalam lingkungan berbasis tile. Algoritma A* modifikasi menghasilkan jalur yang lebih aman karena melewati lebih sedikit tile jebakan dibandingkan dengan algoritma A* standar, meskipun langkah yang ditempuh lebih panjang. Hal ini menunjukkan bahwa modifikasi heuristik dapat mengarahkan pencarian ke rute yang lebih optimal dari segi keamanan. Untuk pengembangan selanjutnya, sistem dapat diperluas dengan menambahkan elemen dinamis seperti musuh atau jebakan bergerak, sehingga tantangan meningkat dan algoritma diuji dalam kondisi yang lebih kompleks dan realistis.

DAFTAR PUSTAKA

- [1] S. Nasution, A. H. Nasution, and A. L. Hakim, "Pembuatan Plugin Tile-Based Game Pada Unity 3D," *IT J. Res. Dev.*, vol. 4, no. 1, pp. 46–60, Aug. 2019, doi: 10.25299/itjrd.2019.vol4(1).3517.

- [2] P. Yap, "New Ideas in Pathfinding," in *Artificial Intelligence and Interactive Entertainment II*, Stanford, CA: AAAI, 2002, pp. 95–97. [Online]. Available: <https://cdn.aaai.org/Symposia/Spring/2002/SS-02-01/SS02-01-018.pdf>
- [3] X. Cui and H. Shi, "A*-based Pathfinding in Modern Computer Games," vol. 11, no. 1, pp. 125–130, 2011.
- [4] E. Sumantri and S. Hidayatullah, "Penerapan Algoritma A* pada Pencarian Rute Terpendek pada Rute Angkot Di Kota Sukabumi," vol. 8, no. 1, pp. 21–35, 2020, doi: <https://doi.org/10.55338/saintek.v5i2.1432>.
- [5] H. Hermawan and H. Setiyani, "IMPLEMENTASI ALGORITMA A-STAR PADA PERMAINAN KOMPUTER ROGUELIKE BERBASIS UNITY," *J. Algoritma Log. Dan Komputasi*, vol. 2, no. 1, May 2019, doi: [10.30813/j-alu.v2i1.1571](https://doi.org/10.30813/j-alu.v2i1.1571).
- [6] D. Foad, A. Ghifari, M. B. Kusuma, N. Hanafiah, and E. Gunawan, "A Systematic Literature Review of A* Pathfinding," *Procedia Comput. Sci.*, vol. 179, pp. 507–514, Jan. 2021, doi: [10.1016/j.procs.2021.01.034](https://doi.org/10.1016/j.procs.2021.01.034).
- [7] C. Niu, A. Li, X. Huang, W. Li, and C. Xu, "Research on Global Dynamic Path Planning Method Based on Improved A * Algorithm," *Math. Probl. Eng.*, vol. 2021, no. 1, p. 4977041, 2021, doi: [10.1155/2021/4977041](https://doi.org/10.1155/2021/4977041).
- [8] Z. Li and H. Li, "Study of the Interception Scheme Based on A* Path Finding Algorithm in Computer Game," *J. Comput. Commun.*, vol. 08, no. 07, pp. 32–49, 2020, doi: [10.4236/jcc.2020.87004](https://doi.org/10.4236/jcc.2020.87004).