p-ISSN: 2746-9204
e-ISSN: 2746-9190
Volume 3, No. 2, 2023

PENERAPAN ALGORITMA A* UNTUK PENCARIAN RUTE TERPENDEK PADA APLIKASI GAME *SHOOT'EM UP*

Hasniati[*1], Syaiful Rahman[2], Suyono[3]

[1]Prodi Teknik Informatika, STMIK KHARISMA, Jl. Baji Ateka No. 20, Makassar, Indonesia, Kodepos: 90134. Email: hasniati@kharisma.ac.id
[2]Prodi Teknik Informatika, STMIK KHARISMA, Jl. Baji Ateka No. 20, Makassar, Indonesia, Kodepos: 90134. Email: syaifulrahman@kharisma.ac.id
[3]Prodi Teknik Informatika, STMIK KHARISMA, Jl. Baji Ateka No. 20, Makassar, Indonesia, Kodepos: 90134. Email: suyono_12@kharisma.ac.id

***Koresponden Author:** Hasniati, hasniati@kharisma.ac.id
Accepted: 07 21, 2023 ; Revised: 0 7 26, 2023; Published: 08 31, 2023

Abstrak

Algoritma A umumnya digunakan dalam pencarian jalur dan penjelajahan graf. Algoritma ini dikenal cukup populer karena tingkat kinerja dan akurasinya yang tinggi. Penelitian ini bertujuan mengimplementasikan algoritma A* untuk mencari jalan terpendek pada game shoot'em up. Dalam perancangan sistem, use case diagram digunakan untuk menjelaskan kebutuhan fungsional sistem, dan activity diagram untuk menggambarkan alur kerja sistem. Sedangkan dalam pembuatan aplikasi yang dapat digunakan pada desktop bersistem operasi Windows digunakan Unity. Metode pengujian, yang digunakan dalam penelitian ini adalah metode black box untuk memvalidasi game dapat bekerja sesuai dengan rancangan sistem, kemudian pengujian beta dan software quality untuk memverifikasi bahwa algoritma A* telah diimplementasikan dengan baik. Hasil dari penelitian ini adalah 1) hasil pengujian menggunakan metode black box yang telah dilakukan dapat disimpulkan bahwa semua tombol dan fitur-fitur yang telah dirancang pada aplikasi dapat berfungsi dengan baik dan benar; 2) hasil pengujian endurance yang dilakukan, dapat disimpulkan bahwa A* sangat akurat dan selalu menemukan rute terpendek dalam waktu yang cepat, dan juga waktu pencarian rute terpendek sangat bergantung pada jumlah simpul.*

Kata kunci*: Unity, Algoritma A*, Shoot'em up,Game*

Abstract

[APPLICATION OF THE A* ALGORITHM TO SEARCH FOR THE SHORTEST ROUTE IN THE SHOOT'EM UP GAME APPLICATION] *The A* algorithm is commonly used in pathfinding and graph traversal. This algorithm is known to be quite popular due to its high level of performance and accuracy. This research aims to implement the A* algorithm to find the shortest path in shoot'em up games. To design the system, use case diagram will be used to explain the functional requirements of the system, and the activity diagram to describe the system workflow. While in the application development that can be used in a desktop with a Windows operating system, Unity will be used. The testing methods used in this study are the black box method to validate that the game can run as designed, beta testing and software quality testing to verify that the A* algorithm was implemented well. The results of this study are 1) According to the results of black box testing, it can be concluded that all the buttons and features on the application manage to work as designed; 2) According to the result of endurance testing that have been done, it can be concluded that A* is very accurate and can find the shortest path in a short time, and the time used for the pathfinding is very dependant on the node count.*

Keywords*: Unity, A* algorithm, Shoot'em up, Game*

AGENTS
JOURNAL OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
p-ISSN: 2746-9204
e-ISSN: 2746-9190
Volume 3, No. 2, 2023

1. PENDAHULUAN

Dalam industri *game*, pengembang sering menghadapi tantangan untuk menciptakan pengalaman *gameplay* yang menarik dan menantang bagi para pemain. Salah satu aspek penting dalam pengembangan *game* adalah sistem pencarian rute terpendek, yang digunakan untuk mengatur pergerakan karakter dalam lingkungan permainan yang kompleks. Pencarian rute terpendek adalah proses menemukan jalan tercepat dari titik awal ke tujuan yang diinginkan.

Dalam video game, kecerdasan buatan (*AI*) digunakan untuk menghasilkan perilaku yang responsif, adaptif, atau cerdas terutama pada karakter non-pemain (*NPC*) yang mirip dengan kecerdasan manusia [1]. Salah satu perilaku AI yang paling sering muncul dan paling penting yaitu pencarian rute terpendek (*pathfinding*). Pathfinding digunakan untuk mencari jalan terpendek dari satu titik ke titik lain. Di dalam *game*, *pathfinding* banyak digunakan untuk pergerakan karakter non-pemain (*NPC*), seperti pada *game* balap mobil [3], game edukasi [4] dan lain sebagainya.

Masalah yang paling umum dalam pencarian jalur dalam *video game* adalah bagaimana menghindari rintangan dengan cerdik dan mencari jalur yang paling efisien di medan yang berbeda[2]. Kejadian ini kemungkinan besar dengan AI yang dibuat akan mengambil jalan yang jauh atau bahkan bisa berjalan ke jalan buntu. Dengan melihat permasalahan tersebut, dibutuhkan sebuah algoritma untuk mencari jalan terpendek sehingga AI yang dibuat dapat mencari jalan terpendek.

Algoritma A* umumnya digunakan dalam pencarian jalur dan penjelajahan graf. Algoritma ini dikenal cukup populer karena tingkat kinerja dan akurasinya yang tinggi [1]. Algoritma A* mencari rute terlebih dahulu untuk melihat rute mana yang akan diambil, kemudian membandingkan rute tersebut dan mengambil jalur terpendek [5].

Sampai saat ini, salah satu genre game yang masih bertahan adalah genre shoot'em up. Walaupun cara bermainnya sederhana, permainan jenis *Shoot'em up* tetap menarik, terutama bagi pemain yang lebih suka permainan yang lebih ringan. *Shoot'em Up*, juga disebut "shmup" adalah jenis permainan di mana karakter utama menyerang pasukan lawan [6]. Tulisan ini menyajikan hasil penelitian terkait dengan pengembangan *Game Shoot'em Up* dengan menerapkan algoritma A* untuk pencarian rute terpendek.

Penerapan algoritma A* untuk mencari jalur terpendek pada permainan Shoot'em Up melibatkan beberapa langkah. Berikut ini tentang bagaimana kita dapat mengintegrasikan algoritma A* ke dalam game:

a) Tentukan Kotak Permainan:

Ubah lingkungan game menjadi kotak, di mana setiap sel mewakili area atau rintangan yang dapat dinavigasi. Grid ini akan digunakan oleh algoritma A* untuk menghitung jalur.

b) Buat Representasi Node:

Mewakili setiap sel pada grid sebagai node dalam algoritma A*. Setiap node harus menyimpan informasi seperti posisinya, tetangganya, dan apakah itu hambatan atau tidak.

c) Fungsi Heuristik:

Tentukan fungsi heuristik (biasanya jarak Manhattan atau jarak Euclidean) yang memperkirakan biaya dari sebuah node ke tujuan. Ini membantu memandu algoritma A* menuju tujuan secara efisien.

d) Daftar Terbuka dan Tertutup:

Pertahankan dua daftar: daftar terbuka dan daftar tertutup. Daftar terbuka berisi node-node yang akan dievaluasi, sedangkan daftar tertutup berisi node-node yang sudah dievaluasi.

e) Eksekusi Algoritma:

Mulailah dengan menambahkan node awal ke daftar terbuka. Meskipun daftar terbuka tidak kosong, lakukan hal berikut: Pilih node dengan total biaya terendah (biaya aktual + heuristik) dari daftar terbuka. Jika node yang dipilih adalah tujuannya, jalurnya telah ditemukan. Jika tidak, pindahkan node yang dipilih dari daftar terbuka ke daftar tertutup.

Untuk setiap tetangga dari node yang dipilih: Jika tetangga tidak dapat dilintasi atau berada dalam daftar tertutup, lewati saja.

p-ISSN: 2746-9204
e-ISSN: 2746-9190
Volume 3, No. 2, 2023

Jika tetangganya tidak ada dalam daftar terbuka, hitung biayanya dan tambahkan ke daftar terbuka.
Jika tetangga sudah ada dalam daftar terbuka, perbarui biayanya jika jalur baru tersebut lebih baik.

f) Rekonstruksi Jalur:
Setelah node tujuan tercapai, rekonstruksi jalur terpendek dengan melakukan backtracking dari node tujuan ke node awal menggunakan referensi induk yang disimpan di setiap node.

g) Integrasikan ke dalam Gameplay:
Gunakan jalur terpendek yang dihitung untuk memandu pergerakan musuh atau entitas apa pun yang dikendalikan AI di game Shoot'em Up. Perbarui posisi mereka berdasarkan hasil pencarian jalan untuk menciptakan perilaku musuh yang dinamis dan cerdas.

h) Pengoptimalan:
Bergantung pada ukuran grid dan kompleksitas permainan, kita mungkin perlu menerapkan pengoptimalan seperti pembaruan pencarian jalur secara berkala, bukan setiap frame.

Perlu pula diperhatikan bahwa pengintegrasian algoritma A* memerlukan pengkodean dan pengujian, dan spesifikasinya dapat bervariasi berdasarkan mesin dan arsitektur game. Penting juga untuk mencapai keseimbangan antara pergerakan musuh yang cerdas dan mempertahankan sifat permainan Shoot'em Up yang serba cepat dan menantang.

2. METODE

Penelitian dilaksanakan menggunakan model waterfall, adapun tahapannya adalah sebagai berikut :

a. Analisis dan definisi kebutuhan.

Pada tahap ini, peneliti mengumpulkan data serta menentukan kebutuhan fungsional dan non-fungsional aplikasi yang akan dibangun. Kegiatan ini bertujuan agar aplikasi yang dibangun sesuai dengan kebutuhan pengguna. Pengumpulan data yang dilakukan melalui studi literatur, yaitu mencari dan mempelajari karya – karya tulis yang relevan dengan penelitian untuk mendapatkan landasan teori dan konsep-konsep lainnya, baik melalui karya tulis fisik maupun hasil pencarian di internet. Data yang dikumpulkan diantaranya mengenai teori atau konsep tentang pencarian rute A*, media pembelajaran, dan perancangan game shooter yang akan digunakan dalam penelitian.

b. Perancangan Sistem dan Perangkat Lunak

Pada tahap ini, rancangan dari aplikasi dibuat berdasarkan hasil dari analisis. Perancangan sistem dan perangkat lunak ini bertujuan untuk membuat gambaran awal mengenai aplikasi. Pada tahap ini, peneliti melakukan perancangan interface (Perancangan input-output), pemodelan fungsi dengan menggunakan *Use Case Diagram*, serta pemodelan *behavior* dengan menggunakan *Activity Diagram*

c. Implementasi dan Pengujian Unit

Pada tahap ini, dilakukan implementasi dari perancangan sistem ke dalam kode program. Penulis menggunakan *Unity* v2017.1.0f3 sebagai *tools* dan C# sebagai bahasa pemrograman. Selama aplikasi dibuat, peneliti juga melakukan pengujian untuk memastikan bahwa aplikasi yang dibangun bebas dari kesalahan.

d. Integrasi dan Pengujian Sistem

Pada tahap ini, dilakukan pengujian pada aplikasi secara menyeluruh dengan menggunakan metode *black box testing*, dimana peneliti memastikan fungsi-fungsi, masukan, dan keluaran dari aplikasi sesuai dengan rancangan. Kemudian akan dilakukan *Endurance Testing* dalam bentuk uji performa *software*, khususnya kecepatan pencarian rute. *Endurance testing* juga akan melihat pengaruh jumlah node yang digunakan di dalam game terhadap kecepatan pencarian rute musuh.

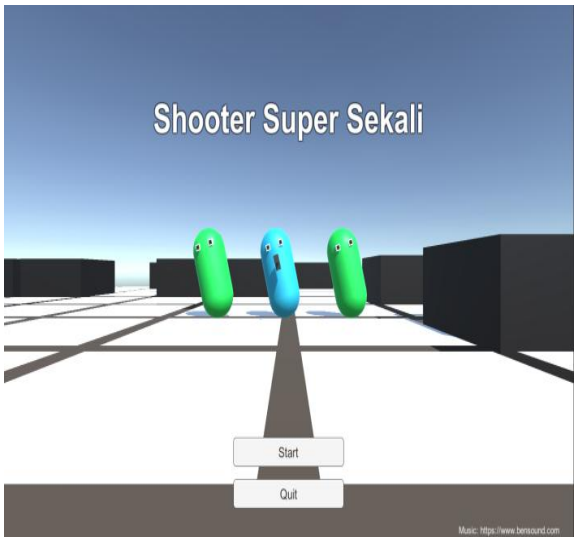
3. HASIL DAN PEMBAHASAN

1. Pengujian Black Box

Aplikasi telah berhasil dibangun dan dilakukan pengujian dengan metode *black box testing*, dimana aplikasi diuji dengan cara memberikan *input* dan mengamati apakah *output* yang dihasilkan sesuai dengan spesifikasi kebutuhan fungsi.

p-ISSN: 2746-9204
e-ISSN: 2746-9190
Volume 3, No. 2, 2023

Berikut sebagian *output* aplikasi ditampilkan pada Gambar 1 sampai dengan Gambar 4 :



Gambar 1 Main Menu

Gambar 2 Tampilan Game saat Memulai Permainan Baru

Gambar 3 Tampilan *Game* saat Musuh Menemukan Rute Terpendek

Pengembangan Game Shoot'em Up menjadi semakin menarik dengan penerapan algoritma A* untuk pencarian rute terpendek. Algoritma A* dapat memberikan tantangan yang dinamis bagi pemain dengan menghasilkan rute musuh yang cerdas dan taktis dalam menghadapi serangan.

Dengan mengintegrasikan algoritma A* ke dalam Game Shoot'em Up, pemain akan mengalami pengalaman bermain yang lebih mendalam karena musuh akan bergerak dengan lebih cerdas dan terkoordinasi, menciptakan permainan yang penuh strategi dan tuntutan refleks tinggi.

Gambar 4 Tampilan *Game Over*

Setelah pemain kalah, layar dapat berubah menjadi latar belakang yang gelap dengan teks besar "Game Over" yang terpusat di tengah atas.

2. Pengujian *Endurance*

Pengaruh jumlah simpul dengan kecepatan algoritma A* dengan menggunakan *obstacle* yang sama, posisi awal dan target yang sama ditunjukkan pada Tabel 1 dan Gambar 1.

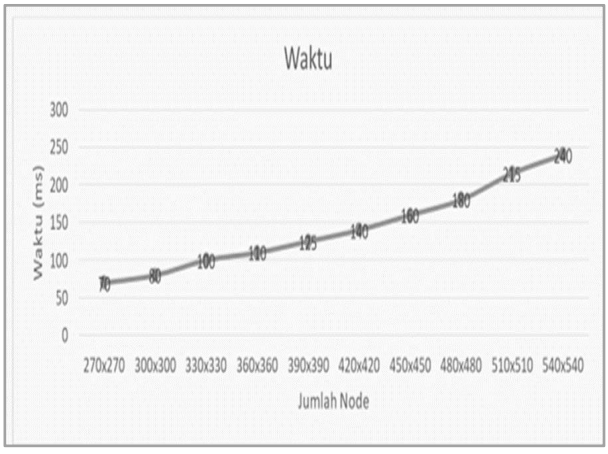
Tabel 1 Pengaruh Jumlah Node Terhadap Waktu Pencarian

Jumlah Node	Waktu
270x270	70ms
300x300	80ms

p-ISSN: 2746-9204
e-ISSN: 2746-9190
Volume 3, No. 2, 2023

330x330	100ms
360x360	110ms
390x390	125ms
420x420	140ms
450x450	160ms
480x480	180ms
510x510	215ms
540x540	240ms

Waktu pencarian rute terpendek sangat bergantung pada jumlah simpul. Dengan kecepatan 70 ms pada ruang pencarian dengan luas 270x270 *node*, dan 240 ms pada ruang pencarian 540x540 *node* yang dapat dilihat pada Tabel 1. dan pada grafik yang ditunjukkan pada Gambar 5.



Gambar 5. Pengaruh Jumlah Node Terhadap Waktu Pencarian

4. KESIMPULAN

Dari hasil keseluruhan penelitian, dapat disimpulkan bahwa:

1. Dari hasil pengujian menggunakan metode black box yang telah dilakukan dapat disimpulkan bahwa semua tombol dan fitur-fitur yang telah dirancang pada aplikasi dapat berfungsi dengan baik dan benar, terlihat dari karakter musuh berjalan mendekati karakter utama sehingga dapat dikatakan metode pathfinding A* berfungsi dengan baik, dan dapat diimplementasikan dalam game.
2. Dari hasil pengujian endurance yang dilakukan, dapat disimpulkan bahwa A* akurat dan dapat menemukan rute terpendek dalam waktu yang cepat, dan juga waktu pencarian rute terpendek sangat bergantung pada jumlah simpul.

5. DAFTAR PUSTAKA

[1] M. Ranjitha, N. Kazaka, dan J. Lincy, "*Artificial Intelligence Algorithms and Techniques in the computation of PlayerAdaptive Games*", Journal of Physics: Conference Series, Vol. 1427, 012006, 2020.

[2] C. Xiao, dan S. Hao , "*A*-based Pathfinding in Modern Computer Games*", International Journal of Computer Science and Network Security, Vol.11 No.1, 2011.

[3] Y. Sazaki, H. Satria and M. Syahroyni, "*Comparison of A and dynamic pathfinding algorithm with dynamic pathfinding algorithm for NPC on car racing game*," in 11th International Conference on Telecommunication Systems Services and Applications (TSSA), 2017.

[4] D. Kurniadi, A. Mulyani and R. S. Maolani, "*Implementation of Pathfinding Algorithm in Sundanese Land History Educational Game*", 2nd International Conference on Innovative and Creative Information Technology (ICITech), 2021.

[5] P. Sara Lutami, dkk, "*A Review of Pathfinding in Game Development*", CEPAT Journal of Computer Engineering: Progress, Application and Technology, Vol. 1 No. 1, hal 46-55, 2022.

[6] P. Adhitama, U. Muhammad, dan P. Novianti, "*Pengembangan Aplikasi Game Shoot'em Up Star Assault Dengan Game Maker Studio*", Prosiding Seminar Ilmu Komputer dan Teknologi Informasi, Vol. 2 No. 1, hal 321-324, 2017.

[7] Suyanto, "*Artificial Intelligence*", Informatika, Bandung, 2014.

[8] S. Yeni, Falahah, S. Hermi. "Penerapan

p-ISSN: 2746-9204
e-ISSN: 2746-9190
Volume 3, No. 2, 2023

Algoritma A* (Star) Untuk Mencari Rute Tercepat Dengan Hambatan", Seminar Nasional Telekomunikasi dan Informatika (SELISIK), 2016.