

Implementation Of A* Algorithm In A Great Elephant Game With Unity 2D

Ahmad Zuhdi¹, Imam Ahmad², Ade Dwi Putra³

^{1,2}Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Teknokrat Indonesia

³Sistem Informasi, Fakultas Teknik dan Ilmu Komputer, Universitas Teknokrat Indonesia
Jl. H. Zainal Abidin Pagaram No.9-11, Bandar Lampung, Indonesia

e-mail: didicsextream11@gmail.com¹, imamahmad@teknokrat.ac.id², adedwiputra@teknokrat.ac.id³

Received : July, 2023

Accepted : August, 2023

Published : August, 2023

Abstract

Video game is a game play which related to interesting User Interface through a picture which processed and transmitted into a video. Pathfinding is a route search method. Google maps is one example of an application which using pathfinding method. A Algorithm through pathfinding method is the development of dijkstra algorithm using heuristic to produce best and optimum solution. So it is possible to discover fastest path without checking in every path. Thus, the objective of this research is to implement A* Algorithm to move the elephant in the game The Great Elephant. This algorithm can be implemented to the character for discovering nearest route and increase the score easily. Because of the route is in maze form, it will be time wasting to looking for any food. This algorithm help the character to not wasting time in looking for the food. The result of this research is an Android game which is succeeded in implementing A* Algorithm into main character to look for nearest route. The result of the test is using ISO 25010 in functionality and portability aspect which is generated 100% success. While, in the usability aspect, the score is 89,54% or very well.*

Keywords: *game, arcade, labirin, algoritma a*, pathfinding*

Abstrak

Video Game atau yang lebih sering dikenal dengan permainan video adalah sebuah permainan yang berhubungan dengan *User Interface* menarik melalui gambar yang diproses atau dipancarkan oleh video. *Pathfinding* adalah sebuah metode pencarian rute, contoh aplikasinya adalah *Google Maps*. Metode *pathfinding* dengan Algoritma A* adalah pengembangan dari algoritma dijkstra yang menggunakan heuristik dalam menghasilkan solusi yang baik dan optimal sehingga memungkinkannya menemukan jalur terpendek lebih cepat tanpa perlu memeriksa setiap jalur. Oleh Karena itu, tujuan dalam penelitian ini adalah implementasi Algoritma A* untuk menggerakkan karakter gajah pada *game great elephant*. Algoritma ini dapat diimplementasikan pada karakter untuk mencari rute terdekat sehingga memudahkan dalam pencapaian skornya, dikarenakan labirin yang menyusahkan mencari makanan, sehingga tidak banyak membuang waktu. Hasil penelitian ini berupa *game* pada *smartphone Android* yang berhasil mengimplementasikan Algoritma A* pada karakter pemain untuk mencari rute terdekat. Hasil pengujian menggunakan ISO 25010 pada aspek *functionality* dan *portability* menghasilkan 100% sukses dan pada aspek *usability* dengan skor 89,54% atau sangat baik.

Kata Kunci: *game, arcade, labirin, algoritma a*, pathfinding*

1. PENDAHULUAN

Video *Game* atau yang lebih sering dikenal dengan permainan video adalah sebuah permainan yang berhubungan dengan *User Interface* menarik melalui gambar yang diproses atau dipancarkan oleh video yang seakan hidup. *Great Elephant* merupakan *game* dengan genre *arcade* yang berbasis 2D (dua dimensi). *Arcade* adalah permainan yang tidak terfokus terhadap *storyline* dan biasanya *game* ini dituntut untuk memberikan kesenangan yang seakan memberikan tantangan sekaligus untuk mencapai angka kemenangan yang diinginkan.

Penggunaan *handphone* pintar pada era digitalisasi sangat tinggi, karena hampir semua komunikasi menggunakan *handphone* pintar. *Game* yang akan dibangun adalah *game* berbasis *Android*. *Android* adalah sistem operasi yang dikenalkan atau dirancang oleh Google. Pengembangan dengan *Android* pada *handphone* pintar yang memungkinkan kita sebagai pengembang dapat mengembangkan aplikasi dengan bebas[1].

Karakter yang digunakan dalam permainan ini adalah hewan gajah, sama seperti dengan nama permainannya *Great Elephant*. Karakter akan berada di dalam sebuah labirin dan harus memakan benda yang akan muncul secara acak di dalam labirin. Ada waktu yang akan membatasi pemain dalam memakan benda di labirin yang berliku-liku untuk menuju benda tersebut.

Pathfinding adalah sebuah metode pencarian rute, contoh aplikasinya adalah *Google Maps*. Metode *pathfinding* dengan Algoritma *A** adalah pencarian rute dengan *heuristik* dalam menghasilkan solusi yang baik dan optimal sehingga memungkinkannya menemukan jalur terpendek lebih cepat tanpa perlu memeriksa setiap jalur [2].

Yusra Fernando dkk. melakukan penelitian yang menerapkan Algoritma *A** untuk mencari lokasi foto mendapat hasil yang sukses[3]. Lalu penelitian yang dilakukan Ahmad Wildan Rizky Ramadhan dkk. yang membandingkan Algoritma *A** dan Dijkstra pada *game* menghasilkan bahwa Algoritma *A** lebih baik dari pada Dijkstra dari segi waktu, score dan *health*[4]. Dan penelitian pada *game* rpg yang dilakukan oleh Andy Pramono dengan

menerapkan Algoritma *A** terbukti sangat efisien sebesar 86%[5]. Tujuan dari penelitian ini untuk mengimplementasikan *pathfinding* dengan menggunakan *algoritma A** pada *game* 2D *Great Elephant*.

1.1 Algoritma A*

Algoritma *A** pertama kali dikemukakan oleh Peter Hart, Nils Nilsson dan Bertram Raphael pada tahun 1968. Algoritma *A** menggunakan pendekatan *heuristic* untuk menyelesaikan masalah lebih cepat[6]. Untuk mempersingkat jalur yang dituju, algoritma *A** menggunakan fungsi:

$$f(n) = g(n) + h(n),$$

dimana:

$$f(n) = \text{biaya estimasi terendah,}$$

$$g(n) = \text{biaya dari node awal ke node } n,$$

$h(n)$ = estimasi biaya dari node n untuk mencapai node akhir.

1.2 Pathfinding

Pathfinding adalah suatu pencarian jalur tercepat dari titik awal ke titik yang dituju agar terhindar dari *obstacle* disepanjang jalur yang dilewati. *Pathfinding* dikategorikan menjadi dua yaitu *undirected* dan *directed*. *Undirected* dianalogikan dengan seekor semut yang terjebak didalam labirin yang berkeliaran membabi buta mencoba mencari jalan keluar, terdapat dua pendekatan dalam penyelesaian masalah pada *pathfinding undirected* yaitu *Breadth-first search* dan *Depth-first search*. *Directed* tidak melewati labirin secara membabi buta, terdapat beberapa metode untuk menentukan semua node yang berdekatan sebelum ditentukan mana yang terbaik, menggunakan pendekatan *Uniform cost search* $g(n)$ dan *Heuristic search* $h(n)$ [7].

1.3 Game (Permainan)

Menurut Clark *game* adalah suatu aktivitas yang diputuskan oleh pemain, untuk mencapai *goals* yang terbatas oleh ruang lingkup tertentu. Pemain dituntut untuk bisa mencapai *game over* atau permainan dinyatakan selesai dengan kondisi menang ataupun kalah, sehingga pemain harus berpikir dan membuat strategi untuk memenangkan permainan dengan waktu atau poin yang ingin dicapai[8].

1.4 Unity

Unity (banyak dikenal sebagai *Unity3D*) adalah *game engine* atau mesin permainan dan juga sebuah *integrated development environment* (IDE) sebagai pembuat media yang menarik, seperti *video games*. Unity diciptakan oleh *Unity Technologies* kali pertama pada bulan Juni 2005 di *Apple Inc. Apple Worldwide Developers Conference*. Unity dapat digunakan untuk membuat *game* dua dimensi maupun tiga dimensi. Unity sangat populer dalam pengembangan *game*, selain itu *unity* juga dapat melakukan *rendering*[9].



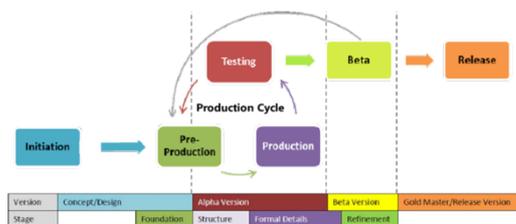
Gambar 1.1 Logo Unity

1.5 Android

Perusahaan *Google* merancang *operation system* (OS) bernama *Android* yang berbasis dari *kernel Linux* pada telepon pintar[10]. *Android* merupakan *operation system* yang menggunakan *java* sebagai bahasa pemrograman[11]. Peluncuran *Android* pada tahun 2007 sudah banyak perkembangan versi, salah satunya adalah *Android 10* pada 2019. Unikny *Android* menggunakan makanan dalam penamaan versi sistem operasi mereka[12].

1.6 Metode Pengembangan Sistem

Pembuatan atau pembangunan *game Great Elephant* menggunakan *Game Development Life Cycle* (GDLC) sebagai metode pengembangan sistem. *Game* banyak menggunakan metode ini dalam pembangunannya, adapun tahapan *Initiation, Pre-production, Production, Testing, Beta, Release*[13].



Gambar 1.2 Metode *Game Development Life Cycle* (GDLC)

1.7 Skala Likert

Skala ini adalah untuk mengukur sikap atau perilaku yang dimiliki responden, selain itu juga untuk memahami persepsi atau pendapat orang ataupun banyak orang, sehingga mendapatkan hasil yang sesuai dengan fenomena sosial yang diteliti. Untuk dapat memberikan hasil atau jawaban yang sesuai maka harus diadakan pengukuran, pada penelitian ini menggunakan kuesioner sebagai instrumen penelitian. Setelah jawaban dikumpulkan dan dianalisis, kemudian akan didapatkan hasil yang sesuai dengan fenomena yang diteliti[14].

1.8 ISO 25010

ISO (*International Organization for Standardization*) 25010 adalah standar yang memperbarui model ISO 9126 (2007). Ada delapan *sub-fitur* dan karakteristik utama. Berbasis ISO 9126, yang tujuan utamanya adalah untuk memandu dalam pengembangan produk perangkat lunak dalam hal spesifikasi dan evaluasi kebutuhan sistem, ilustrasi model ISO 25010[15].

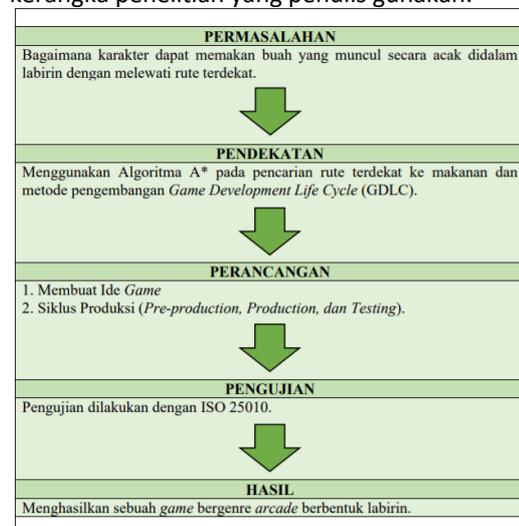


Gambar 1.3 ISO 25010

2. METODE PENELITIAN

2.1 Kerangka Penelitian

Kerangka penelitian adalah bentuk tahapan pola pemikiran yang berdasarkan rumusan masalah dan tujuan penelitian, berikut adalah kerangka penelitian yang penulis gunakan:



Gambar 2.1 Kerangka Penelitian

Pada permasalahan yang ada adalah bagaimana karakter dapat berjalan menuju buah untuk memakan buah tersebut didalam labirin. Disini penulis melakukan pendekatan pertama yaitu melalui simulasi dengan aplikasi *excel* menggunakan rumus Algoritma A*. Lalu melakukan perancangan membuat ide maupun latar. Lalu mengimplementasikan Algoritma A* dan juga menggunakan GDLC sebagai metode pengembangan *game* yang akan dibangun. Lalu *game* diuji menggunakan ISO 25010.

2.2 Simulasi Algoritma A*

Simulasi Algoritma A* adalah bagaimana cara kerja Algoritma A* menentukan manakah rute terpendek untuk mencapai tujuan, dimana menggunakan rumus sebagai berikut:

$$f(n) = g(n) + h(n),$$

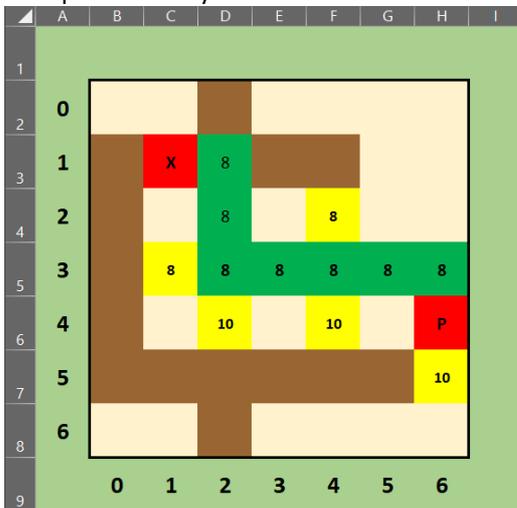
dimana

$f(n)$ = biaya estimasi terendah,

$g(n)$ = biaya dari *node* awal ke *node* n ,

$h(n)$ = estimasi biaya dari *node* n untuk mencapai *node* akhir.

Berikut adalah gambar yang menampilkan kotak berukuran 7x7 dan terdapat dua titik yaitu P dan X:



Gambar 2.2 Simulasi Algoritma A*

Berikut adalah penjelasan pada gambar 2.2 melalui tabel 3.1.

Tabel 3.1 Perhitungan Algoritma A*

No.	Langkah	$g(n)$	$h(n)$	$f(n)$
1	Pemain berjalan ke atas	1	7	8
2	Pemain tidak berjalan ke bawah	1	9	10
3	Pemain berjalan ke kiri	2	6	8
3	Pemain berjalan ke kiri	3	5	8
4	Pemain tidak berjalan ke atas	4	4	8
5	Pemain tidak berjalan ke bawah	4	6	10
6	Pemain berjalan ke kiri	4	4	8
7	Pemain berjalan ke kiri	5	3	8
8	Pemain berjalan ke atas	6	2	8
9	Pemain tidak berjalan ke kiri	6	2	8

10	Pemain tidak berjalan ke bawah	6	4	10
11	Pemain berjalan ke atas	7	1	8

Berdasarkan gambar 2.2, pada warna merah menandakan pemain dan tujuan, lalu warna hijau adalah jalur yang dengan biaya terendah, dan kuning adalah jalur yang tidak dilalui. Posisi pemain pada langkah pertama berjalan keatas karena nilai $f(n) = 8$ lebih kecil dibanding nilai $f(n) = 10$, begitu seterusnya sampai mencapai tujuan yaitu X.

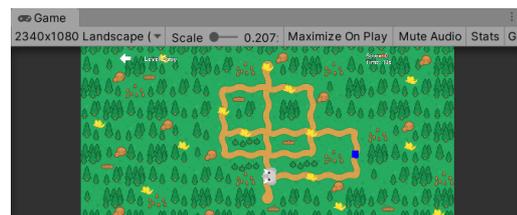
3. HASIL DAN PEMBAHASAN

3.1 Hasil Penelitian

Hasil penelitian adalah berhasilnya dibuat *game great elephant* ini dengan mengimplementasikan Algoritma A* dan dibangun dengan *Unity*. Dimana *game* di *export* ke dalam bentuk apk dan dijalankan di *handphone* pintar berbasis *Android*. Kemudian akan dilakukan pengujian pada *game* yang diuji oleh penguji ahli dan pengguna dengan menggunakan ISO 25010.

3.2 Pembahasan

Pada tahapan ini adalah pembahasan mengenai implementasi Algoritma A* pada *game great elephant*, peneliti melakukan *debug* pada *unity* untuk melihat apakah algoritma berjalan dengan maksimal atau tidak. *Debug* yang dilakukan adalah mengenai posisi pemain dari titik biru gambar 3.1, kemudian nilai $g(n)$, nilai $h(n)$ dan yang terakhir didapatkan hasil akhir yaitu $f(n)$. Hasil *debug* pada karakter ada pada gambar 3.3.



Gambar 3.1 Posisi Pemain

Pada posisi awal pemain, pemain memiliki dua pilihan, ke atas atau kebawah, Algoritma A* menghitung dan mendapatkan hasil yang baik yaitu dengan kearah atas.ditandai dengan garis merah pada gambar 3.2.



Gambar 3.2 Penentuan Rute



Gambar 3.3 Debug

Berikut adalah proses pemain berjalan kearah mana:

Tabel 3.1 Penjelasan Debug

No.	Langkah	$g(n)$	$h(n)$	$f(n)$
1	Pemain tidak berjalan ke atas	1	6	7
2	Pemain berjalan ke bawah	1	4	5
3	Pemain berjalan ke kiri	2	3	5
3	Pemain berjalan ke kiri	3	2	5
4	Pemain berjalan ke kiri	4	1	5
5	Pemain berjalan ke kiri	5	0	5

Berdasarkan gambar 3.2, pada titik biru adalah titik awal dari pemain, diberikanlah opsi oleh labirin dengan arah atas bergaris kuning atau bawah bergaris merah, melalui perhitungan menggunakan Algoritma A* atau garis merah maka, pada langkah pertama, pemain tidak berjalan ke atas, dapat dilihat pada nilai $f(n) = 7$ yang lebih besar dibandingkan pada langkah kedua jika pemain berjalan ke arah bawah yaitu $f(n) = 5$, sampai pada langkah kelima yaitu dengan yaitu *goal* atau tujuan.

3.3 Pengujian

Berikut adalah hasil pengujian secara keseluruhan pada aplikasi pada tabel.

Tabel 3.2 Hasil Keseluruhan Pengujian

Aspek	Skor	Skor Max	%Skor	Hasil
Functionality	46	46	100	Sukses
Portability	22	22	100	Sukses
Usability	591	660	89,54	Sangat Baik

Hasil dari pengujian ISO 25010, terhadap tiga aspek yaitu *functionality* dengan mengaitkan satu orang ahli, dan mengaitkan sebelas pengguna untuk pengujian *portability* dan *usability*. Pengujian *functionality* untuk melihat apakah fungsi sudah sesuai atau belum, pada pengujian *portability* untuk melihat apakah *game* dapat dipasang dan digunakan pada *smartphone* dengan versi yang telah ditargetkan. Lalu pada pengujian *usability* untuk melihat apakah *game* dapat dimainkan oleh pengguna.

Pengujian *functionality* dan *usability* menghasilkan skor 100% atau sukses. Sedangkan pada pengujian *usability* mendapatkan skor 89,54% atau sangat baik. Skor pada pengujian *usability* didapat dari perhitungan skor maksimal yaitu dua belas buah pertanyaan, dengan satu bobot maksimal pada satu pertanyaan adalah 55 dikalikan dua belas maka didapatkan hasil 660. Skor yang didapat setelah dilakukan pengujian *usability* yaitu $(591/660) \times 100\% = 89,54\%$ nilai ini sudah menunjukkan sangat baik.

4. KESIMPULAN

Sebagaimana yang telah dinyatakan dalam hasil pengujian dan pembahasan pada 3 aspek pengujian. Implementasi Algoritma A* Pada *Game Great Elephant* Dengan *Unity 2D*, dapat disimpulkan sebagai berikut:

1. *Game Great Elephant* berhasil dibangun berdasarkan model yang telah dijelaskan, dengan hasil pengujian yang diperoleh *functionality* 100%, *portability* 100% dan *Usability* 89,54%.
2. Algoritma A* berhasil diimplementasikan untuk mencari rute terdekat pemain ke tujuan.

DAFTAR PUSTAKA

- [1] R. R. Setiawan and S. Nita, "Perancangan Aplikasi Pembelajaran Qur'an Edu Berbasis Android," *Pros.*

- Semin. Nas. Teknol. Inf. dan Komun.*, vol. 2, no. 1, pp. 225–228, 2019.
- [2] L. Safira, P. Harsadi, and S. Harjanto, “Penerapan Navmesh Dengan Algoritma A Star Pathfinding Pada Game Edukasi 3d Go Green,” *J. Teknol. Inf. dan Komun.*, vol. 9, no. 1, p. 17, 2021, doi: 10.30646/tikomsin.v9i1.540.
- [3] Y. Fernando, M. A. Mustaqov, and D. A. Megawaty, “Penerapan Algoritma a-Star Pada Aplikasi Pencarian Lokasi Fotografi Di Bandar Lampung Berbasis Android,” *J. Teknoinfo*, vol. 14, no. 1, p. 27, 2020, doi: 10.33365/jti.v14i1.509.
- [4] A. W. R. Ramadhan and D. Udjulawa, “Perbandingan Algoritma Dijkstra dan Algoritma A Star pada permainan Pac-Man,” *J. Algoritm.*, vol. 1, no. 1, pp. 12–20, 2020, doi: 10.35957/algoritme.v1i1.411.
- [5] A. Pramono, “Algoritma Pathfinding A* Pada Game RPG Tanaman Higienis,” *J. Edukasi dan Penelit. Inform.*, vol. 1, no. 2, 2015, doi: 10.26418/jp.v1i2.12517.
- [6] H. Kevin, “Analisis Perbandingan Algoritma Dijkstra dan Algoritma A* (A-Star) dalam Pencarian Rute Terpendek (Studi Kasus: SPBU di Kota Medan),” *Repos. Institusi Univ. Sumatera Utara*, 2020, [Online]. Available: repositori.usu.ac.id/handle/123456789/27965.
- [7] U. A. Shahrin Iskandar, N. Mat Diah, and M. Ismail, “Identifying Artificial Intelligence Pathfinding Algorithms for Platformer Games,” *IEEE Int. Conf. Autom. Control Intell. Syst.*, pp. 74–80, 2020, doi: 10.1109/I2CACIS49202.2020.9140177.
- [8] N. S. Azizah, “IMPLEMENTASI PATHFINDING DENGAN ALGORITMA A* MENGGUNAKAN UNITY 3D PADA GAME SHOOTER,” *Sinus Repository*, 2018. eprints.sinus.ac.id/377.
- [9] R. J. Iskandar, Antonius, and Edwinyo, “Penggunaan Unity Engine Pada Perancangan Game the Cient Dengan Navigation Mesh,” *InTekSis*, vol. 8, no. 2, pp. 61–72, 2019.
- [10] L. Safitri and S. Basuki, “Analisa dan Perancangan Sistem Informasi Text Chatting Berbasis Android Web View,” *J. IPSIKOM*, vol. 8, 2020.
- [11] M. H. Romadhon, Y. Yudhistira, and M. Mukrodin, “Sistem Informasi Rental Mobil Berbasis Android Dan Website Menggunakan Framework Codeigniter 3 Studi Kasus : CV Kopja Mandiri,” *J. Sist. Inf. dan Teknol. Perad.*, vol. 2, no. 1, pp. 30–36, 2021.
- [12] A. Y. Permana and P. Romadlon, “Perancangan Sistem Informasi Penjualan Perumahan Menggunakan Metode SDLC pada PT. Mandiri Land Prosperous Berbasis Mobile,” *SIGMA - J. Teknol. Pelita Bangsa*, vol. 10, no. 9, pp. 153–167, 2019.
- [13] Parjito, R. I. Kuncoro, and D. Pasha, “APLIKASI PEMBELAJARAN PENGENALAN KATA BAHASA LAMPUNG MENGGUNAKAN METODE GDLC BERBASIS ANDROID,” *Sci. Comput. Sci. Informatics J.*, vol. 6, no. 1, pp. 13–26, 2023.
- [14] Sugiyono, “Materi Skala Likert Lengkap dengan Contohnya,” *Wiki Elektronika*, 2021. <https://wikielektronika.com/skala-likert/> (accessed Sep. 11, 2021).
- [15] E. Peters and G. K. Aggrey, “An ISO 25010 based quality model for ERP systems,” *Adv. Sci. Technol. Eng. Syst.*, vol. 5, no. 2, pp. 578–583, 2020, doi: 10.25046/aj050272.