ANALISIS PENGARUH PENGGUNAAN NILAI HEURISTIK TERHADAP PERFORMANSI ALGORITMA A* PADA GAME PATHFINDING

Wina Witanti[1)], Dewi Nurul Rahayu[2)]
Kopertis Wilayah IV, dpk Jurusan Informatika,
Fakultas MIPA,Universitas Jenderal Achmad Yani
Jl. Terusan Jenderal Sudirman PO BOX 148, Cimahi, 40521
HP: +62 813 21902103
E-mail:wina.witanti@lecture.unjani.ac.id

Abstrak

Salah satu implementasi algoritma A adalah sebuah game pathfinding. Game pathfinding merupakan game untuk mencari jalan terpendek dari titik awal menuju titik tujuan pada sebuah peta.Algoritma ini digunakan untuk menentukan jalan terpendek menuju tujuan. Performansi yang akan dihasilkan algoritma A* pada game pathfinding yaitu waktu pencarian, jumlah langkah dari titik awal menuju titik tujuan dan simpul yang diperiksa. Algoritma A* merupakan perbaikan dari metode Best-First Search (BFS) dengan menggunakan fungsi heuristik. Fungsi heuristik sering jugadisebut f(n) yang merupakan penentuan urutan titik yang akan dikunjungi terlebih dahulu.Pada penelitian ini, dilakukan analisis terhadap penggunaan nilai heuristik yang berbeda untuk menentukan performansi algoritma A* pada sebuah game pathfinding dengan ordo maksimal 31x39. Hasil pengujian aplikasi pada penelitian ini,adalah selain didapatkan jalan menuju tujuan pada sebuah map dan pelacakan cabangnya, juga memberikan hasil pencarian jalan optimal yang merupakan jalan terpendek, dengan maksimal penghalang sebanyak 1207 penghalang, maka akan didapatkan maksimal sebanyak 38 jalan.*

***Kata Kunci**: pathfinding, A*, heuristik*

Abstract

One implementation of the A Algorithm is a pathfinding game.Pathfinding game is a game to find the shortest path from the starting point towards the destination point from a map. This algorithm is used to determine the choice of the shortest path to the destination point.A* Algorithm is an improvement of the best-first search (BFS) method by using the heuristic function. Heuristic function is also called f (n) which is the determination of the sequence point to be visited first. In this research will be analyzed the use of different heuristic value to determine the performance of the A* Algorithm in pathfinding game with a maximum order of 31x39. The test's result on the application of this research is obtained besides path to the destination on a map and tracking subsidiary, also provides search results optimal path which is the shortest path, with a maximum barrier as 1207, then will be get38 ways as a maximum way.*

***Keywords**: pathfinding, A*, heuristic*

1. PENDAHULUAN

Proses dan cara berpikir seperti manusia yaitu *artificial intelligence* (kecerdasan buatan) untuk berbagai keperluan komputasi telah banyak didopsi.Kecerdasan buatan dapat dipandang dari berbagai sudut pandang salah satunya sudut pandang penelitian. Domain yang sering dibahas oleh para peneliti salah satunya formal *task* yang diterapkan pada permainan atau *game*.Hal penting dalam menentukan keberhasilan suatu aplikasi berdasarkan kecerdasan buatan adalah kesuksesan dalam pencarian dan pencocokan. Pada dasarnya terdapat dua teknik pencarian dan pelacakan yang digunakan, yaitu pencarian buta (*blind search)* dan pencarian terbimbing (*heuristic search).* Dalam pencarian terbimbing ada beberapa algoritma yang dapat digunakansalah satunya algoritma *A**. Algoritma A*merupakan pengembangan dari *best-first search* (BFS) yang digunakan untuk mencari jalan terpendek *(shortest path)* yang sering dipakai dalam *game programming,* salah satunya akan diterapkan pada *game pathfinding.*Pada penelitian ini diimplementasikan algoritma A* dalam pencarian jalan terpendek pada *game pathfinding* dimana algoritma tersebut digunakan untuk mencari jalan terpendek pada *game pathfinding*. Tujuan dari penelitian ini adalah mengetahui performansi waktu pencarian, jarak dan simpul yang diperiksa dari titik awal menuju titik tujuan dengan algoritma A* *(A Star)* yang diterapkan dalam pencarian jalan terpendek pada *game pathfinding.*

2. TINJAUAN PUSTAKA

2.1 Algoritma A*

Algoritma A* (*A star*) adalah algoritma *pathfinding* pengembangan dari Algoritma BFS (*Best First Search*). Seperti halnya pada BFS, untuk menemukan solusi, A* juga 'dituntun' oleh fungsi heuristik[1,2,3]. Perbedaan cara kerja A* dengan BFS adalah selain memperhitungkan *cost* dari *currentstate* ke tujuan dengan fungsi *heuristic* (seperti BFS), algoritma ini juga mempertimbangkan *cost* yang telah ditempuh selama ini dari *initial state* menuju ke *current state*. Jadi, apabila jalan yang ditempuh sudah terlalu panjang dan ada jalan lain yang lebih kecil *cost*-nya namun memberikan solusi yang sama apabila dilihat dari *goal*, maka jalan baru yang lebih pendek itulah yang akan dipilih [2,5]. Notasi yang dipakai oleh Algoritma A* adalah sebagai berikut:

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

Pada notasi standar yang dipakai untuk algoritma A* tersebut, digunakan *g(n)* untuk mewakili *cost* rute dari *node* awal ke *node* n, lalu *h(n)* mewakili perkiraan *cost* dari *node* awal ke *nodegoal*, yang dihitung dengan fungsi *heuristic*. A* menyeimbangkan kedua nilai ini dalam mencari jalan dari *node* awal ke *nodegoal*. Dalam menentukan *node* yang akan dikembangkan, algoritma ini akan memilih *node* dengan nilai *f(n)* yang paling kecil. Sehubungan denganAlgoritma A* yang memberikan prioritas berdasarkan harga *f(n)*, maka jika keadaan sepadan terjadi, terdapat lebih dari satu *node* dengan prioritas sama. Akibatnya adalah *node-node* tersebut akan diperiksa terlebih dahulu, yang mungkin *node* tersebut terletak berjauhan dengan *node* tujuan. Hal ini berakibat turunnya kinerja algoritma A*[4].

2.2 Fungsi Heuristik

Algoritma A* sebagai algoritma pencarian yang menggunakan fungsi heuristikuntuk 'menuntun' pencarian rute, khususnya dalam hal pengembangan dan pemeriksaan *node-node* pada peta [5]. Terdapat beberapa fungsi heuristikumum yang bisa dipakai untuk algoritma A* ini. Salah satunya adalah yang dikenal dengan istilah '*Manhattan Distance*'. Fungsi heuristikini digunakan untuk kasus dimana pergerakan pada peta hanya lurus (horizontal atau vertikal), tidak diperbolehkan pergerakan diagonal[6]. Perhitungan nilai heuristikuntuk *node* ke-n menggunakan *Manhattan Distance* adalah sebagai berikut:

$$h(n) = (abs(n.x - goal.x) + abs(n.y - goal.y)) \quad (2)$$

Dimana h(n) adalah nilai heuristikuntuk *node* n, dan *goal* adalah *node* tujuan. Jika pergerakan diagonal pada peta diperbolehkan, maka digunakan fungsi heuristikselain *Manhattan Distance*.

Untuk mendekati kenyataan, *cost* untuk perpindahan *node* secara diagonal dan orthogonal dibedakan. *Cost* diagonal adalah 1,4 kali *cost* perpindahan secara orthogonal, maka fungsi heuristikyang digunakan adalah sebagai berikut:

$$\left.\begin{aligned} &h\_diagonal(n) = \min(abs(n.x - goal.x) + abs(n.y - goal.y)) \\ &h\_orthogonal(n) = (abs(n.x - goal.x) + abs(n.y - goal.y)) \\ &h(n) = h\_diagonal(n) + (h\_orthogonal(n) - (2 * h\_diagonal(n))) \end{aligned}\right\} \quad (3)$$

Dimana h_diagonal(n) adalah banyaknya langkah diagonal yang bisa diambil untuk mencapai *goal* dari *node* n. h_orthogonal adalah banyaknya langkah lurus yang bisa diambil untuk mencapai *goal* dari *node* n. Nilai *heuristic* kemudian diperoleh dari h_diagonal(n) ditambah dengan selisih h_orthogonal(n) dengan dua kali h_diagonal(n). Dengan kata lain, jumlah langkah diagonal kali *cost* diagonal ditambah jumlah langkah lurus yang masih bisa diambil dikali *cost* pergerakan lurus[8].

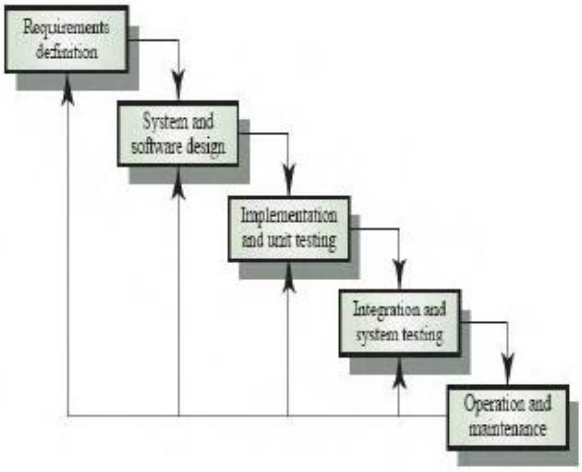
2.3 Algoritma A* pada Game

Setiap *game* memiliki aturan main. Hal ini mempermudah upaya menghasilkan ruang pencarian dan memberikan kebebasan pada para peneliti dari bermacam-macam ambisi dan kompleksitas sifat serta kurangnya struktur permasalahan. Papan konfigurasi yang digunakan untuk memainkan *game* ini mudah direpresentasikan pada komputer dan tidak memerlukan bentuk yang kompleks. *Game* dapat menghasilkan sejumlah besar pencarian ruang. Hal ini cukup besar dan kompleks sehingga membutuhkan suatu teknik yang tangguh untuk menentukan alternatif pengeksplorasian ruang permasalahan. Teknik ini

dikenal dengan nama heuristikdan merupakan area utama dari penelitian tentang kecerdasan buatan. Banyak hal yang biasanya dikenal sebagai kecerdasan tampaknya berada dalam heuristikyang digunakan oleh manusia untuk menyelesaikan permasalahannya[6].

3. ANALISIS, PERANCANGAN DAN IMPLEMENTASI

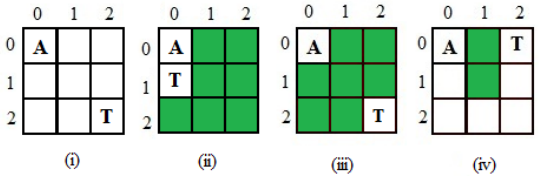
Analisis merupakan bagian terpenting dalam pembangunan sebuah *game*. *Game* yang akan dibangun adalah *game pathfinding* merupakansebuah *game* simulasipencarian jalan untuk sampai ke tujuan,dalam proses pencariannya akan dimodelkan dengan fungsi heuristik. Perancangan dalam pembangunan *game pathfinding*, digunakan model Waterfall [7], seperti pada Gambar 1.



Gambar 1.Model Waterfall

3.1 Analisis

Deskripsi masalah adalah suatu gambaran masalah yang diangkat dalam penelitian ini, yangakan berupa kotak-kotak dengan ordo minimal 3x3 dan maksimal 31x39. Gambar 2 merupakan contoh empat kondisi yang diimplementasikan.



*Gambar 2. Contoh kondisi ruang map yang akan dihitungdengan A**

A: Titik awal; T: Titik tujuan; Warna hijau/gelap: Penghalang

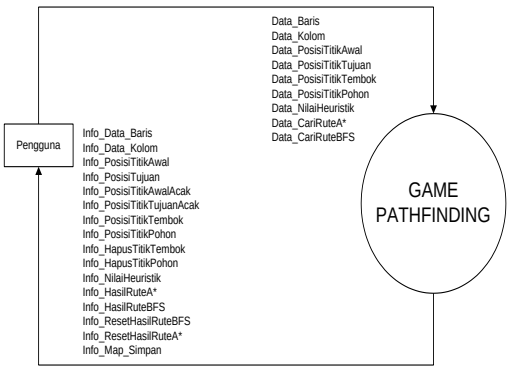
Pada analisis non-fungsional yang merupakan analisis untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan.

1) Analisis masukan
   Analisis masukan dilakukan untuk mengetahui masukan data apa saja yang diperlukan oleh sistem. Masukan data yang diperlukan oleh sistem yaitu titik awal, titik tujuan, baris kotak, kolom kotak, penghalang maupun tanpa penghalang dan nilai heuristik.
2) Analisis keluaran
   Analisis keluaran dilakukan untuk mengetahui keluaran dari sistem yang dibutuhkan oleh *user* yaitu berupa jalur terpendek dan pelacakan pencarian dari titik awalke titik tujuan.
3) Analisis kebutuhan perangkat keras *(hardware)*
   Agar aplikasi dapat berjalan dengan baik, maka dibutuhkan perangkat keras yang sesuai dengan kebutuhan aplikasi, lihat Tabel 1.

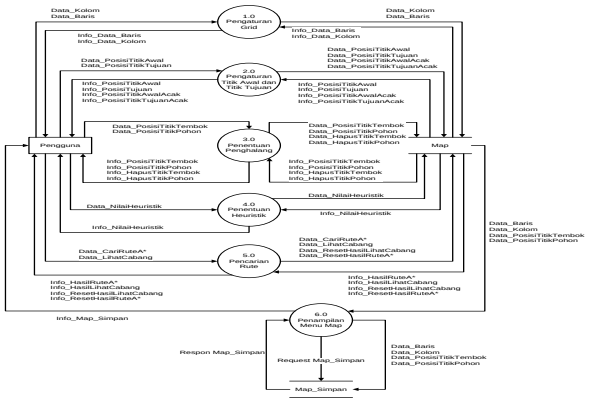
Tabel 1. Spesifikasi perangkat keras untuk implementasi

Komputer	Spesifikasi perangkat keras
Prosesor	Processor 1.6 Ghz
Layar	Resolusi 1366 x 768 pixel
Monitor	LCD 14 inc
Memori	Memori 128Mb
Harddisk	Harddisk 20 GB
Keyboard dan mouse	-

Diagram konteks atau disebut juga dengan model aplikasi fundamental yang merepresentasikan seluruh elemen aplikasi sebagai sebuah *bubble* tunggal dengan data masukan-keluaran. Pada *game pathfinding* ini dijelaskan dengan diagram konteks seperti pada Gambar 3, DFD level 0-nya dijelaskan pada Gambar 4.



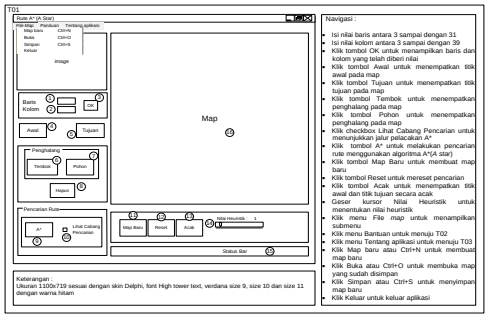
Gambar 3. Diagram konteks game pathfinding



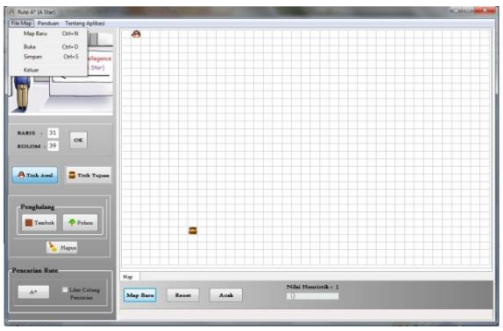
Gambar 4. DFD Level 0 game pathfinding

3.2 Perancangan dan Implementasi

Perancangan antarmuka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun sehingga memudahkan pembuatannya. Dalam aplikasi yang dibangun akan dirancang sebuah *game pathfinding* yang menggunakan Algoritma A*, dimana ordo yang siapkan minimal 3x3 dengan maksimal 7 penghalang dan maksimal ordo yang dimungkinkan untuk mengetahui jalur terpendek dari titik awal sampai dengan titik tujuan adalah 31x39 dengan maksimal penghalang sebanyak 1207 buah. Penghalang akan berupa tembok dan pohon yang pada saat implementasi, penggunaan penghalang ini dapat dikombinasikan. Penentuan jarak terpendek pada penelitian ini sangat tergantung pada pemilihan nilai heuristik, waktu, dan jumlah simpul yang diperiksa, sehingga akan menghasilkan jalan terpendeknya, dengan adanya penghalang atau tidak. Gambar 5 menjelaskan mengenai rancangan antarmuka menu utama dan Gambar 6 menjelaskan mengenai implementasi tampilan utamanya.



Gambar 5. Rancangan antarmuka menu utama



Gambar 6. Implementasi menu utama

4. HASIL DAN DISKUSI

Sebagai hasil analisis, maka dilakukan uji performansi. Performansi yang akan diuji pada aplikasi*game pathfing* yaitu waktu pencarian, simpul yang diperiksa dan langkah yang dihasilkan dengan parameter nilai heuristkantara rentang 0,1 sampai 3. Pengujian performansi dengan ordo maksimal 31x39 tanpa penghalang dengan nilai heuristik1 dilakukan dengan hasil pengujian performansi dengan parameter nilai heuristik antara

rentang 0,1 sampai 3 dengan titik awal pada simpul (0,0) dan titik tujuan pada simpul (31,39) seperti terlihat pada Tabel 2.

Tabel 2. Hasil pengujian ordo 31x39 tanpa penghalang

No.	Nilai heuristik	Waktu (ms)	Jumlah simpul yang diperiksa	Jalan yang dihasilkan
1	0,1	36785	1178	37
2	0,5	34788	1114	37
3	1	30764	985	37
4	1,5	24321	778	37
5	2	15413	493	37
6	2,5	3136	99	37
7	3	1264	39	37

Pengujian performansi pada ordo maksimal 31x39 dengan penghalang dan nilai heuristik1 juga dilakukan dengan hasil pengujian performansi pada dengan parameter nilai heuristik antara rentang 0,1 sampai 3 dengan titik awal pada simpul (0,0) dan titik tujuan pada simpul (31,39) seperti pada Tabel 3.

Tabel 3. Hasil pengujian ordo 31x39 dengan penghalang

No.	Nilai heuristik	Waktu (ms)	Jumlah simpul yang diperiksa	Jalan yang dihasilkan
1	0,1	16536	529	38
2	0,5	14789	473	38
3	1	11980	383	38
4	1,5	10078	322	38
5	2	7207	230	38
6	2,5	2824	89	38
7	3	1701	53	38

Berdasarkan pengujian alpha yang telah dilakukan terhadap aplikasi, maka jalan yang dihasilkan merupakan jalan terpendek karena simpul yang diperiksa relatif banyak dan memerlukan waktu pencarian yang relatif lama pula. Parameter dengan nilai heuristikpaling kecil akan menghasilkan waktu dan simpul yang diperiksa besar sebaliknya nilai heuristikpaling besar akan menghasilkan waktu dan simpul yang diperiksa kecil.

5. SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan hasil pengujian guna mengetahui performansi waktu pencarian, jarak dan simpul yang diperiksa dari titik awal menuju titik tujuan dengan Algoritma A* *(A Star)* yang diterapkan dalam pencarian jalan terpendek pada *game pathfinding*, maka didapatkanlah untuk minimum jumlah simpul yang diperiksa dengan waktu 1264 ms yaitu sebanyak 39 simpul dengan menggunakan nilai heuristik3 dan maksimal jumlah simpul yang diperiksa dengan waktu 36785 ms yaitu sebanyak 1178 simpul dengan menggunakan nilai heuristik 0,1 didapatkanlah 37 jalan yang dihasilkan untuk ordo 31x39 tanpa penghalang. Namun didapatkan sebanyak 38 jalan yang dihasilkan untuk ordo 31x39 dengan penghalang, yaitu untuk minimum jumlah simpul yang diperiksa dengan waktu 1701 ms yaitu sebanyak 53 simpul dengan menggunakan nilai heuristik 3 dan maksimal jumlah simpul yang diperiksa dengan waktu 16536 ms yaitu sebanyak 59 simpul dengan menggunakan nilai heuristik 0,1. Jadi, lamanya proses pencarian dan banyaknya simpul yang diperiksa untuk pencarian jalan tergantung pada jarak antara titik awal dan titik tujuan.Hasil Algoritma A*memberikan hasil pencarian jalan yang optimal.

5.2 Saran

Berikut adalah beberapa saran untuk pengembangan lebih lanjut, karenaAlgoritma A* mempunyai beberapa kelemahan diantaranya adalah membutuhkan komputasi yang relatif lama, maka A* dapat dikombinasikan dengan algoritma lain untuk meningkatkan performansi pencariannya. Ordo yang digunakan untuk *game pathfinding* juga dapat ditambah secara otomatis sesuai keinginan *user* (lebih dinamis) dan secara antarmuka dapat dikembangkan dengan *icon* yang lebih *user friendly*.

6. DAFTAR RUJUKAN

[1] A Star Pathfinding for Beginners, 2004. *Policyalnac.*Available at: <http://www.policyalmanac.org/games>. [Accesed 21 April 2011]

[2] AI Game Programming,2003. *Amit's Game Programming Site*.Avaliable at: <http://theory.stanford.edu/~amitp/GameProgramming>. [Accesed 9 Oktober 2010]
[3] Dechter, Rina; Judea Pearl. *Generalized Best-First Search Strategies and the Optimality of A**. Journal of the ACM32 (3): pp.505-536.1985
[4] Riftadi, Mohammad. *Variasi Penggunaan Fungsi Heuristik dalam Pengaplikasian Algoritma A**, MakalahIF2251, Teknik Informatika ITB, Bandung. Avaliable at: <http://www.informatika.org/~rinaldi/Stmik/2006-2007/Makalah_2007/MakalahSTMIK2007-069.pdf,> [Accesed 5 Juni 2012]
[5] Russel, Stuart J and Peter Novig, 2003. New Jersey: Prentice Hall. *Artificial Intelligence*.
[6] Sutanto, Arnold Nugroho.(2009), Penerapan Algoritma A* dalam Pencarian Jalan untuk Permainan *Lose Your Marble*, MakalahIF3051-059, Program Studi Teknik Informatika ITB, Bandung. Available at:<http://www.informatika.org/~rinaldi/Stmik/2009-2010/Makalah2009/MakalahIF3051_2009-059.pdf> [Accesed 6 Juni 2012]
[7] Sommervile, Ian. 2004. Prentice Hall. *Software Engineering.*
[8] Zou, Huilai., Qu, Zening., Qu, Youtian. *Optimized Application and Practice of A* Algorithm in Game Map Path-Finding*.Avaliable at: <http://multicore.zju.edu.cn/iwcsei/iwcsei2010/4108c138.pdf>[Accesed 28 Mei 2012]