

Rancang Bangun Game Kumbang Kum-Oid Berbasis Android Menggunakan Algoritma A* (A Star)

Fadhli Almuiini Ahda¹ Martshena Gema²

STMIK Asia Malang

¹adhi3_22@yahoo.com

ABSTRAK. Video *game* merupakan salah satu media hiburan yang sangat populer untuk semua orang. Teknologi *game* telah mengalami kemajuan yang sangat pesat. Salah satunya adalah *game* puzzle yang memberi permainan. *Game* juga bisa diartikan sebagai arena keputusan dan aksi pemainnya. Permainan berbasis komputer, komputer dirancang dengan menerapkan *Artificial intelligence* (AI). Algoritma yang digunakan dalam *game* ini adalah algoritma A*. Dimana A* merupakan perbaikan dari BFS dengan memodifikasi fungsi heuristiknya yang akan meminimumkan total *cost* lintasan. Dengan konsep perbandingan terhadap fungsi $f(n)$ dari sebuah *node* dengan menentukan nilai $g(n)$ dan nilai $h(n)$, sehingga komputer dapat memilih langkah terpendek dengan mengambil nilai terkecil dalam setiap *state*. Sehingga pada kondisi yang tepat algoritma A* akan memberikan solusi terbaik dalam waktu yang optimal. Dari hasil pengujian yang dilakukan *game* simulasi mampu menemukan jalur dengan rute terpendek. Pencarian jalur akan berhasil menemukan solusi berupa rute terpendek, jika terdapat jalur yang bisa dilalui (ada jalur yang tidak tertutup penghalang) dan akan gagal dalam menemukan solusi jika tidak ada jalur yang bisa dilalui atau semua jalur yang ada tertutup oleh penghalang. Dari pengujian beta yang didapat, disimpulkan bahwa 78% orang menyatakan bahwa *game* dan simulasi kumbang kum-oid mampu memberi variasi dalam *game* puzzle, serta 60% orang menyatakan bahwa *game* dan simulasi kumbang kum-oid menarik untuk dimainkan.

Kata Kunci: *Artificial intelligence, Algoritma A*, Game*

1. PENDAHULUAN

Algoritma A* dapat dipakai untuk pencarian langkah pada berbagai macam *game maze*. Algoritma ini juga dapat digunakan untuk path finding atau pencarian jalan suatu karakter untuk sampai ke tujuan, dalam proses pencariannya akan dimodelkan dengan fungsi heuristic. A* (*A Star*) menggunakan fungsi heuristic yang sering juga disebut $f(n)$ yang merupakan penentuan urutan titik mana yang akan dikunjungi terlebih dahulu. Fungsi heuristic ini sebenarnya menyimpulkan seberapa baik / mungkin titik itu dikunjungi untuk mencapai titik tujuan.

A* juga akan meminimumkan total biaya lintasan, pada kondisi yang tepat A* akan memberikan solusi yang terbaik dalam waktu yang optimal. Algoritma A* adalah 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 maze*. Berdasarkan latar belakang masalah diatas penulis tertarik untuk mengambil pokok bahasan Tugas akhir dengan judul “Rancang Bangun Game Kumbang Kum-Oid Berbasis Android Menggunakan Algoritma A* (A Star)”.

Berdasarkan latar belakang masalah yang telah diuraikan diatas, yang menjadi pokok permasalahan adalah sebagai berikut “Bagaimana cara mengimplementasikan Algoritma A* dalam pencarian jalan terpendek pada *game* dan simulasi Kum-Oid berbasis android.

Dalam penelitian ini diberikan beberapa batasan masalah antara lain : [1] Bahasa Pemrograman yang digunakan adalah Action Script 3.0, [2] Algoritma A* (Star), [3] *Game* menggunakan *Meinhattan*, [4] Algoritma diterapkan pada musuh (NPC).

2. KAJIAN TEORI

2.1 Game

Video *game* merupakan salah satu media hiburan yang sangat populer untuk semua orang, baik itu tua maupun muda. Teknologi *game* telah mengalami kemajuan yang sangat pesat. Terbukti hal ini ditandai dengan berkembangnya jenis, produk dan alat yang beragam untuk berinteraksi dengan *game*. *Game* berasal dari kata bahasa Inggris yang memiliki arti dasar permainan. Permainan dalam hal ini merujuk pada pengertian “kelincahan intelektual” (*intellectual playability*). *Game* juga bisa diartikan sebagai arena keputusan dan aksi pemainnya. Ada target – target yang ingin dicapai pemainnya. (Surjana,2003:7).

2.2 Artificial Intelligence

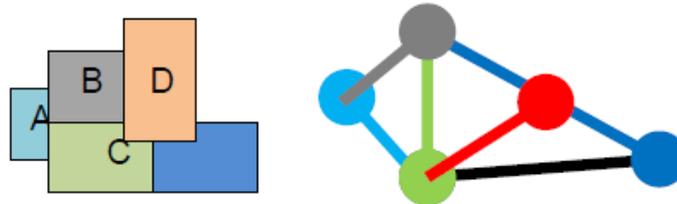
Artificial intelligence merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. (Kusumadewi,2003:1)

Lebih detailnya, pengertian kecerdasan buatan dapat dipandang dari berbagai sudut pandang, antara lain: [1] Kecerdasan akan membuat mesin menjadi cerdas (mampu berbuat seperti apa yang dilakukan oleh manusia) [2] Kecerdasan buatan suatu studi bagaimana membuat computer dapat melakukan sesuatu sebaik yang dikerjakan oleh manusia. [3] Kecerdasan buatan merupakan kumpulan peralatan yang sangat powerful dan metodologis dalam menyelesaikan masalah – masalah bisnis [4] Kecerdasan buatan meliputi studi tentang pemrograman simbolik, penyelesaian masalah (Problem Solving) dan pencarian (Searching).

2.3 Graph Keadaan

Graph terdiri dari *Node-Node* yang menunjukkan keadaan yaitu keadaan awal dan keadaan baru yang akan dicapai dengan menggunakan operator.*Node-Node* dalam graph keadaan saling dihubungkan dengan menggunakan arc (busur) yang diberi panah untuk menunjukkan arah dari suatu keadaan ke keadaan berikutnya. (Kusumadewi,2003:14)

Konsep graph ini pertama diperkenalkan oleh seorang matematikawan dari Swiss, Leonhard Euler pada abad 18. Pada representasi graph, vertex digambarkan dengan Node sedangkan edge digambarkan sebagai yang menghubungkan Node-Node tersebut. Contoh representasi graph terhadap suatu peta dapat dilihat pada Gambar 1. (Rahayu,2010:14)

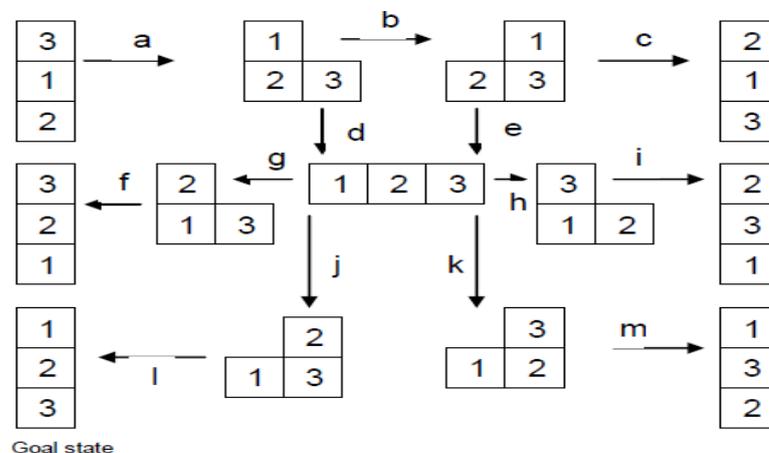


Gambar 1. Sebuah Peta dan representasi graphnya

2.4 Pohon Pencarian

Pelacakan adalah teknik pencarian sesuatu. Dalam pencarian ada dua kemungkinan hasil yang didapat yaitu menemukan dan tidak menemukan. Sehingga pencarian merupakan teknik yang penting dalam artificial intelligence. Hal ini penting dalam menentukan keberhasilan sistem bedasar kecerdasan adalah kesuksesan dalam pencarian dan pencocokan. Keberhasilan dan kualitas pencarian diukur dari empat cara yaitu [1] Kelengkapan, [2] Optimal, [3] Waktu yang digunakan, [4] Ruang yang dibutuhkan, [5] Memori yang digunakan.

Struktur pohon digunakan untuk menggambarkan keadaan secara hirarkis. Pohon juga terdiri dari beberapa node. Node yang terletak pada level-0 disebut juga “akar”. Node akar menunjukkan keadaan awal yang biasanya merupakan topik atau objek. Node akar ini terletak pada level-0. Node akar mempunyai beberapa percabangan yang terdiri atas beberapa node successor yang sering disebut dengan nama “anak” dan merupakan node – node perantara. Ruang keadaan ditunjukkan pada Gambar 2.



Gambar 2. Ruang Keadaan

2.5 Algoritma Pencarian

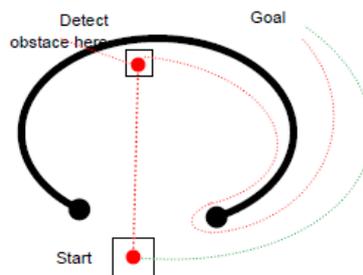
[1] Metode pencarian yang pertama adalah metode yang sederhana yang hanya berusaha mencari kemungkinan penyelesaian yang disebut juga pencarian buta. [2] Metode yang lebih kompleks yang akan mencari jejak terpendek. Metode ini adalah British Museum Procedure, Branch dan Bound, Dynamic Programming, Best first search, Greedy search, A* (A star) Search, dan Hill Climbing Search, metode – metode ini digunakan pada saat harga perjalanan untuk mencari kemungkinan menjadi perhitungan, beberapa procedure / metode yang kita terapkan saat berhadapan dengan musuh.

2.6 A* (A Star) Search

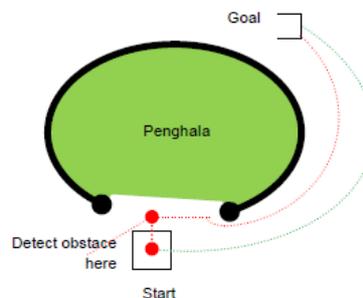
Algoritma A* merupakan perbaikan dari metode BFS dengan memodifikasi fungsi heuristicnya A* (A Star) akan meminimumkan total cost lintasan. Pada kondisi yang tepat A* akan memberikan solusi yang terbaik dalam waktu yang optimal.

2.7 Pathfinding

Pathfinding (pencarian jalan/rute) adalah salah satu bidang penerapan yang sering ditangani oleh kecerdasan khususnya dengan menggunakan algoritma pencarian. Penerapan yang dapat dilakukan dengan pathfinding antara lain adalah pencarian rute dalam suatu game dan pencarian jalan/ rute pada suatu peta. Algoritma pencarian yang dipakai harus mengenali jalan dan elemen pada peta yang tidak dapat dilewati. Penentuan rute tanpa pathfinding dan penentuan rute dengan pathfinding ditunjukkan pada Gambar 3 dan Gambar 4.



Gambar 3. Penentuan rute tanpa pathfinding



Gambar 4. Penentuan rute dengan pathfinding

3. PEMBAHASAN

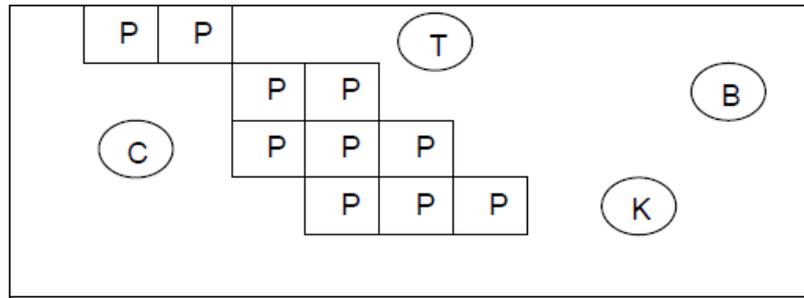
3.1 ANALISA PERANCANGAN SISTEM

3.1.1 Konsep Game dan Simulasi Kumbang Kum-Oid

Game dan simulasi kumbang kum-oid merupakan game yang didalamnya terdapat simulasi kumbang Kum-oid, kumbang yang didalam game diharuskan untuk mengumpulkan crystal pada keranjang yang disediakan. Di dalam game ini, player akan bermain sebagai kumbang pengumpul crystal yang bernama Clod. Tugas player didalam game adalah mengumpulkan crystal pada keranjang, player diberikan sebuah tas yang mampu menyimpan satu crystal, jika player menemukan crystal, maka crystal akan disimpan didalam tas terlebih dahulu. Setelah crystal disimpan didalam tas, tugas player selanjutnya adalah mencari keranjang untuk meleatakan crystal tersebut.

Game ini juga memiliki non player character (NPC) berupa kumbang Merah yang bernama Koid. Koid bertugas untuk mencuri crystal yang ada didalam tas player. Crystal yang telah di curi akan dibuang

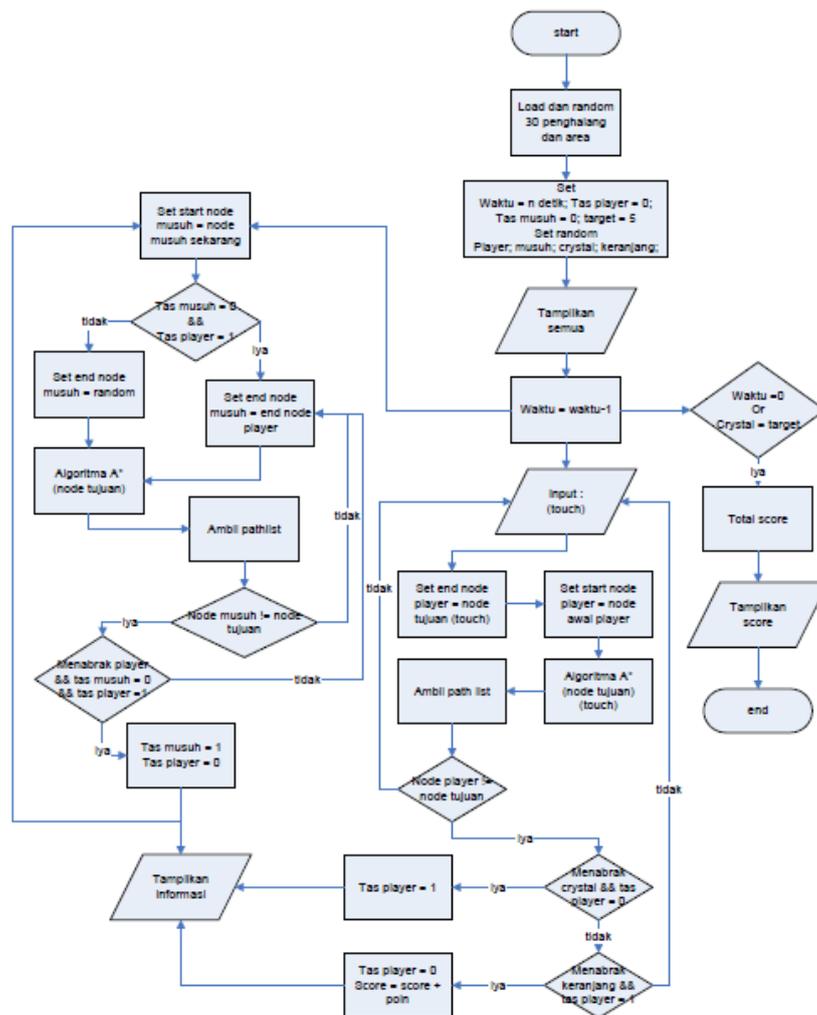
secara random. Untuk setiap crystal yang berhasil dtaruh pada keranjang akan diberi score sesuai warna crystal. Semua score akan diakumulasi jika waktu habis atau semua crystal terkumpul. Gambar simulasi game map kum-oid dapat dilihat pada Gambar 5.



Gambar 5. Gambar simulasi game map kum-oid

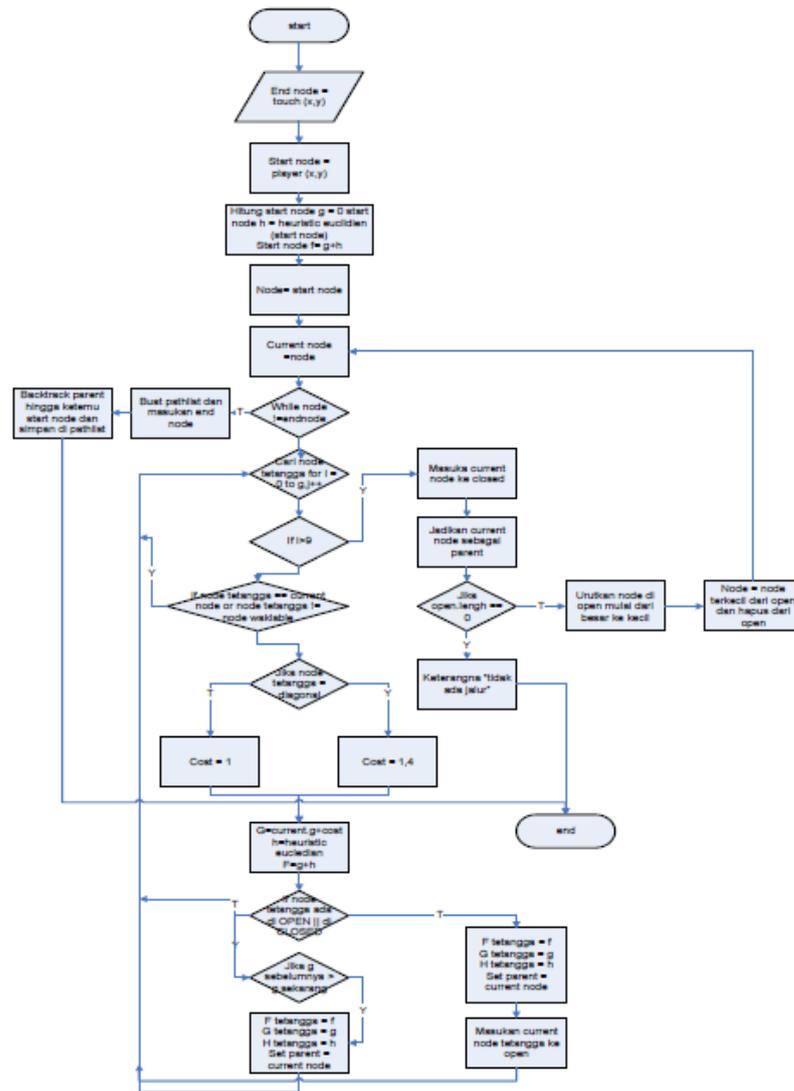
Keterangan:
 P: Penghalang
 K: Koid
 B: Crystal
 T: Tempat Crystal

3.1.2 Perancangan Game



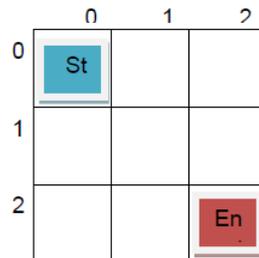
Gambar 6. Flowchart Level Game

3.1.2 Konsep Algoritma A* (A Star)



Gambar 7. Flowchart A*

Diasumsikan posisi random clod berada dikordinat $x : 0$ dan $y : 0$ sedangkan crystal menempati $x : 2$ dan $y : 2$. Maka set start node = (0,0) dan endNode = (2,2). Seperti pada Gambar 8.



Gambar 8. Set start dan end node

	0	1	2
0	G=2.8 h=2.8 F=5.6	G=2.4 H=2.4 F=4.8	g=2.8 H=2 F=4.8
1	G=2.4 H=2.4 F=4.8	Best node	G=2.4 H=1 F=3.4
2	G=2.8 H=2 F=4.8	G=2.4 H=1 F=3.4	End Node

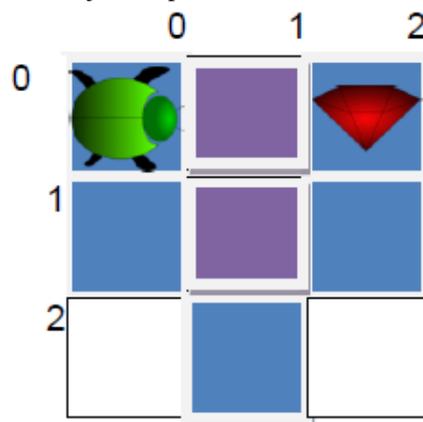
Gambar 9. Nilai g,hdan f tetangga node (1,1)

Berdasarkan Gambar 9, karena $g(0,0), (0,1), (1,0)$ yang lama $<$ $g(0,0), (0,1), (1,0)$ yang baru, parent node $(0,0), (0,1), (1,0)$ tidak perlu diganti. Kemudian set parent node $(0,2), (1,2), (2,0), (2,1), (2,2)$ ke node $(1,1)$. Seperti Tabel 1.

Tabel 1. Set parent $(0,2), (1,2), (2,0), (2,1), (2,2)$ ke $(1,1)$

Node	Node parent
0,1	0,0
1,0	0,0
1,1	0,0
0,2	1,1
1,2	1,1
2,0	1,1
2,1	1,1
2,2	1,1

Dengan demikian PATH LIST berisi jalur yang terbaik, yaitu : $(0,0), (0,1), (1,2), (2,1), (2,0)$. Jalur ini adalah jalur terpendek. Jadi algoritma A* adalah optimal, tanpa ad batasan waktu dan memori, A* adalah complete. Hasil pencarian jalan dengan algoritma A* ditunjukkan pada Gambar 10.

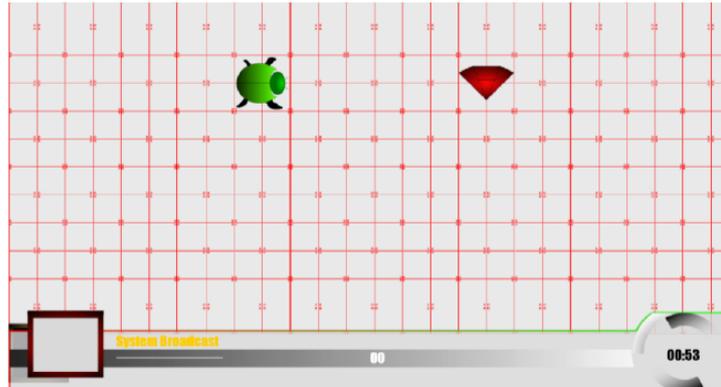


Gambar 10. Hasil Pencarian jalan dengan algoritma A*

3.2 PENGUJIAN SISTEM

3.2.1 Pengujian Tanpa Penghalang

Berikut ini adalah pengujian A* pada Game dan simulasi. Kumbang kum-oid dengan map 13x 6 yang tidak diberi penghalang. Diasumsikan Clod berada di posisi 2,2 dan berlian posisi 12,2. Hasil pengujian tanpa penghalang ditunjukkan pada Gambar 11. Sedangkan path listnya ditunjukkan pada Tabel 2.



Gambar 11. Pengujian Tanpa Penghalang

Tabel 2. Path list pada map tanpa penghalang

No	Path list
1	8,1
2	7,1
3	6,1
4	5,1

Dari pengujian yang telah dilakukan maka dapat diketahui jalur terpendek yang ditemukan yaitu : (5,1),(6,1),(7,1),(8,1). Hasil pencarian yang dilakukan algoritma A*, jalur terpendek yang telah ditemukan diberi gambar lingkaran dan daerah yang pernah dikunjungi di beri gambar persegi. Berikut pada Tabel 3 merupakan hasil pengujian A* tanpa penghalang.

Tabel 3. Pengujian A* tanpa penghalang

Data masukan	Yang diharapkan	Kesimpulan
Mencari jalur terpendek	Clod mampu mencari jalur menggunakan A*	[√] sesuai [] tidak sesuai
Mencapai tujuan	Clod mampu mencapai tujuan menggunakan A*	[√] sesuai [] tidak sesuai

3.2.2 Pengujian Dengan Penghalang

Berikut ini adalah pengujian A* pada Game kumbang kum-oid dengan map 13x6 yang diberi penghalang. Diasumsikan clod berada diposisi 0,0 dan berlian berada di posisi 0,2 kemudian diberi penghalang pada posisi (1,0),(1,1). Tampilan pengujian dengan penghalang dapat dilihat pada Gambar 12, sedangkan path list mapnya bisa dilihat pada Tabel 4.



Gambar 12. Pengujian dengan penghalang

Tabel 4. Path List map dengan penghalang

No	Path list
1	2,1
2	1,2
3	0,1

Dari pengujian yang telah dilakukan dapat diketahui jalur terpendek yang ditemukan yaitu : (0, 1), (1, 2), (2, 1). Hasil pencarian yang dilakukan algoritma A*, jalur terpendek yang telah ditemukan diberi gambar lingkaran dan daerah yang pernah dikunjungi diberi gambar persegi. Hasil pengujian A* dengan penghalang disimpulkan pada Tabel 5.

Tabel 5. Pengujian A* dengan penghalang

Data masukan	Yang diharapkan	Kesimpulan
Mencari jalur terpendek	Clod mampu mencari jalur menggunakan A*	[✓] sesuai [] tidak sesuai
Mencapai tujuan	Clod mampu mencapai tujuan menggunakan A*	[✓] sesuai [] tidak sesuai

4. KESIMPULAN

- a. Pencarian jalan pada Game dan simulasi kumbang kum-oid menggunakan algoritma A* akan berhasil menemukan solusi jika terdapat jalur yang bisa di lalui.
- b. Secara fungsionalitas, game dan simulasi kumbang kum-oid berjalan dengan baik.
- c. Game dapat diimplementasikan pada device smartphone andromax dengan OS android.
- d. Secara jenis, game dapat memberi variasi game dalam jenis game puzzle

DAFTAR PUSTAKA

- [1] Kusumadewi,Sri, Artificial Intelegence. Yogjakarta. Graha Ilmu. 2003
- [2] Rahayu, Dewi Nurul, Penerapan Algoritma A* (A Star) Dalam Pencarian Jalan Terpendek Pada Game Pathfinding. Bandung. Unikom. 2010
- [3] Surjana, Dadang, Rancang Bangun Aplikasi Game Mobile Crazy Becak. Bandung. Unikom. 2013