

Implementasi Algoritma Dijkstra Pada Game Pacman Implementation Dijkstra Algorithm in The Pacman Game

Anggit Dwi Hartanto¹, Aji SuryaMandala^{*2}, Dimas Rio P.L.³, Sidiq Aminudin⁴, Andika Yudirianto⁵

*Penulis Korespondensi

^{1,2,3,4,5} Fakultas Ilmu Komputer, Universitas AMIKOM Yogyakarta

E-mail: ²aji.mandala@students.amikom.ac.id, ³dimas.0158@astudents.amikom.ac.id,
⁴sidiqaminudin@gmail.com, ⁵andika.yudirianto@students.amikom.com

Abstrak

Pacman adalah salah satu game berbentuk labirin dimana game ini telah menggunakan artificial intelligence atau kecerdasan buatan, kecerdasan buatan tersebut tersusun dari beberapa algoritma yang disisipkan dalam program dan Penerapan algoritma dijkstra sebagai salah satu metode penyelesaian masalah yaitu masalah rute minimum pada ghost pacman, dimana ghost berperan mengejar player. Algoritma dijkstra menggunakan prinsip yang hampir sama dengan algoritma greedy dimana dimulai dari titik pertama dan akan ketitik selanjutnya yang terhubung untuk sampai tujuan, cara membandingkan angka dimulai dari titik awal lalu melihat node selanjutnya jika terhubung maka akan mencocokkan 1 jalur dengan jalur lainnya. Dari hasil tahap pengujian didapatkan hasil bahwa algoritma dijkstra cukup baik melakukan pemecahan solusi rute minimum untuk mengejar player yaitu dengan mendapatkan nilai 13 sesuai dengan perhitungan manual.

Kata Kunci— algoritma dijkstra, pacman, pengujian

Abstract

Pacman is one of the labyrinth-shaped games where this game has used artificial intelligence, artificial intelligence is composed of several algorithms that are inserted in the program and Implementation of the dijkstra algorithm as a method of solving problems that is a minimum route problem on ghost pacman, where ghost plays a role chase player. The dijkstra algorithm uses a principle similar to the greedy algorithm where it starts from the first point and the next point is connected to get to the destination, how to compare numbers starting from the starting point and then see the next node if connected then matches one path with the path). From the results of the testing phase, it was found that the dijkstra algorithm is quite good at solving the minimum route solution to pursue the player, namely by getting a value of 13 according to manual calculations

Keywords— dijkstra algorithm, pacman, testing

1. PENDAHULUAN

Game dalam bahasa Indonesia yang berarti permainan yang dimainkan untuk mendapatkan kepuasan atau kesenangan saat memainkannya, dalam perkembangan dunia game sekarang sangat banyak menggunakan penerapan algoritma kepada karakter game tersebut untuk mempersulit pemain memenangkan permainan contoh saat kita bermain game pes dimana bot atau non player dapat bergerak mengambil bola dan setelah bola terambil maka akan langsung menyerang ke gawang kita, non player tersebut menggunakan sebuah algoritma yang tersimpan dalam sebuah system dan di era 90an terdapat game yang cukup populer game tersebut memiliki design yang biasa namun game tersebut cukup sulit untuk ditamatkan yaitu game pacman[1][2][7][9][10].

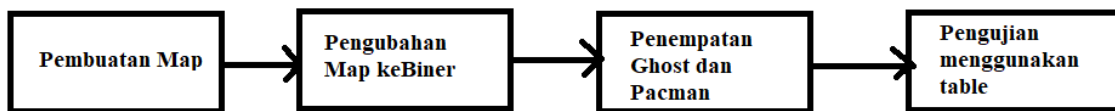
Pacman adalah salah satu game berbentuk labirin dimana game ini telah menggunakan artificial intelligence atau kecerdasan buatan, kecerdasan buatan tersebut tersusun dari beberappa algoritma yang disisipkan dalam progam. Game pacman ini dibuat untuk memberikan tujuan kepada player untuk mengambil semua titik dan menghindari ghost yang mengejar pemain, ghost pada game pacman dapat mengejar player dengan menggunakan penerapan algoritma, salah satu algoritma yang dipakai adalah dijkstra [7].

Algoritma dijkstra adalah salah satu algoritma yang sering digunakan untuk memecahkan masalah search problem, Algoritma dijkstra menggunakan prinsip yang hampir sama dengan algoritma greedy dimana dimulai dari titik pertama dan akan ketitik selanjutnya yang terhubung untuk sampai ketujuan, cara membandingkan angka dimulai dari titik awal lalu melihat node selanjutnya jika terhubung maka akan mencocokkan 1 jalur dengan jalur lainnya mana yang bernilai lebih kecil dan jika node tidak terhubung langsung dari titik start(yang dimulai), maka titik akan diberi symbol infinity ∞ dan jika jalur yang dilalui lebih pendek maka dijkstra dapat melakukan perubahan angka atau update angka untuk mendapatkan nilai jarak yang minimum. Kelebihan algoritma dijkstra dari pada lainnya adalah sering digunakan untuk menyelesaikan permasalahan rute terpendek[4][2][8].

Permasalahan rute terpendek dalam game pacman tentunya digunakan untuk memecahkan masalah jalur yaitu dengan membuat ghost dapat menentukan rute dengan jalur terpendek yang memungkinkan untuk dilewati ghost untuk mengejar player dengan jarak yang efektif dengan begitu terciptalah solusi terbaik kepada masalah jalur yang diberikan sehingga membuat ghost dapat menangkap player. Player dituntut menghindari ghost serta melalui map labirin yang berkelok kelok sembari mengambil semua titik.[3].

2. METODE PENELITIAN

Berikut adalah alur penerapan algoritma dengan pembuatan map, perubahan map ke biner, penempatan ghost dan pacman, pengujian menggunakan table



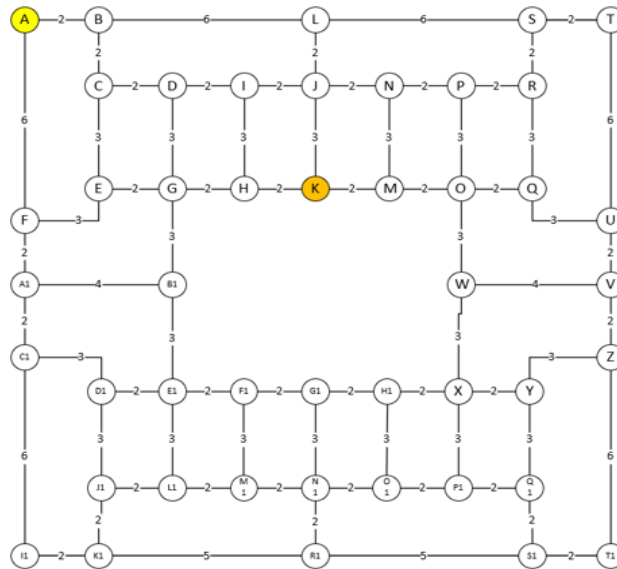
Gambar 1. Metode dalam pencarian rute pacman

2.1. Pembuatan Map

Tahap pertama adalah pembuatan map pacman, map adalah sebuah peta atau tempat labirin yang digunakan sebagai tempat ghost dan pacman berada. Pembuatan map dapat dilakukan dengan mendesign map menggunakan aplikasi seperti photoshop atau corel.

2.2. Pengubahan Map ke Biner

Tahap kedua adalah Pengubahan map ke biner, pengubahan dilakukan setelah membuat map dan menentukan siku putaran seperti siku A, B, C dan seterusnya dari map yang kita design lagi kita ubah kebiner dengan angka 0 dan 1. Didalam pacman terdapat sebuah map yang seperti labirin, map tersebut dibuat berkelok kelok yang fungsinya memberikan hambatan ke player. Map yang didesign akan dirubah keangka biner dimana angka 1 berarti dapat dilalui atau jalur untuk jalan sedangkan 0 adalah tembok dimana penghalang atau tembok harus dihindari oleh pemain ataupun ghost. Untuk mempermudah proses perhitungan maka map tersebut diubah kedalam bentuk jalur Gambar ini menjelaskan posisi pacman dan ghost, pada gambar ini pacman berada dititik A dan ghost berada di titik K. Ghost dari titik K akan menangkap pacman yang berada dititik A dengan menggunakan sebuah algoritma yaitu algoritma dijkstra.



Gambar 2. Map pacman hitung

2.3. Penempatan Ghost dan Pacman

Tahap ketiga adalah penempatan ghost dan pacman ditempat awal atau dititik 0 penempatan bisa ditaruh dimanapun asal masih berada dalam map. Dari gambar 2 pacman berada dititik A dan Ghost berada dititik K.

2.4. Pengujian Menggunakan Table

Tahap keempat adalah pengujian menggunakan table yang digunakan ghost sebagai pencarian rute terpendek untuk mengejar pacman. Berikut beberapa rule algoritma dijkstra yang digunakan untuk menyelesaikan masalah rute terpendek tersebut:

1. Titik awal bernilai 0 / nol
2. Titik Lainnya = tidak diketahui/tak terhingga (∞). Maksud titik lainnya adalah titik yang tidak terhubung langsung dengan visited contoh saja node K dengan N, node K ke N diberi tanda (∞) karena tidak langsung terhubung dengan node K melainkan terhubung dengan node M.
3. Pilih titik dengan jarak paling optimal dari titik awal. Jarak optimum adalah jarak terbaik yang didapat dari visited pilih jarak dengan nilai yang paling kecil.
4. Titik paling dekat yang belum dikunjungi pada titik terakhir kita lakukan perhitungan untuk menentukan jaraknya dari titik awal.
5. Jika jarak yang dihitung lebih kecil dari jarak yang sudah diketahui maka kita tentukan bahwa jarak paling dekat adalah dari titik yang baru kita dihitung.
6. Lalu kita update titik sebelumnya dengan titik yang baru
7. Jika sudah selesai , pada titik akhir yang sudah optimal kita masukan ke daftar visited agar tidak dilakukan perhitungan lagi.

3. HASIL DAN PEMBAHASAN

Dari Gambar 2 kita akan menguji algoritma dijkstra terhadap ghost untukmengejar pacman, dari gambar tersebut dapat dilihat pacman berada pada posisi titik A dan ghost berada pada posisi awal yaitu K

Table 1. *Proses 1 dan 2*

A			B		
Titik	Jarak dari K	Titik sebelumnya	Titik	Jarak dari K	Titik sebelumnya
A	=		A	=	
B	=		B	=	
C	=		C	=	
D	=		D	=	
E	=		E	=	
F	=		F	=	
G	=		G	4	H
H	2	K	H	2	K
I	=		I	5	H
J	3	K	J	3	K
K	0	-	K	0	
L	=		L	=	
M	2	K	M	2	K
N	=		N	=	
O	=		O	=	
P	=		P	=	
Q	=		Q	=	
R	=		R	=	
S	=		S	=	
T	=		T	=	
U	=		U	=	
V	=		V	=	
W	=		W	=	
X	=		X	=	
Y	=		Y	=	
Z	=		Z	=	
R1	=		R1	=	
Q1	=		Q1	=	
P1	=		P1	=	
M1	=		M1	=	
O1	=		O1	=	

Table 1 A adalah langkah pertama ghost menentukan rute yang akan dilalui untuk mengejar pacman, untuk melihat rute dapat dilihat pada Gambar 2.ghost yang berada di titik K akan mencari rute selanjutnya, karena K sudah ditempati maka K dimasukan kedalam visited dan terdapat 2 titik selanjutnya yaitu H dan M. Nilai terkecil dari H memiliki 2 nilai dan M memiliki 2 nilai kedua node memiliki nilai yang sama, ghost mengambil node H karena pada kasus ini pacman berada disebelah kiri maka ghost mengambil node H untuk dicek tapi pada dasarnya jika menemukan ke dua node bernilai sama maka ke dua node harus dicek, trik ini dilakukan untuk menekan jumlah table dan mempersingkat waktu.

Visited : K.

Unvisited : A B C D E F G H I J L M N O P Q R S T U V W X Y Z.

Table 1 B adalah langkah ke 2 ghost menentukan rute yang akan dilalui untuk mengejar pacman, untuk melihat rute dapat dilihat pada Gambar 2. Ghost yang awalnya berada pada node K berpindah atau berjalan menjadi ke node H karena proses pertama, karena H sudah ditempati maka H dimasukan kedalam visited. Lalu dari H menghasilkan 2 titik calon visited yaitu G dan I dimana G memiliki nilai 2 dan I memiliki nilai 3, Sistem pun melakukan pengecekan kepada 2 node tersebut berbeda dengan manusia yang langsung tau bahwa G adalah nilai terkecil, tetapi tidak dfengan sistem ia mengecek ke dua node tersebut lalu membandingkanya darin hasil perbandingan atau pengecekan tersebut didapatkan hasil bahwa nilai node G lebih kecil dibandingkan dengan nilai node I dengan nilai 2.

Visited : K H

Unvisited : A B C D E F G I J L M N O P Q R S T U V W X Y Z

Tabel 2. Proses 3 dan 4

A			B		
Titik	Jarak dari K	Titik sebelumnya	Titik	Jarak dari K	Titik sebelumnya
A	=		A	=	
B	=		B	=	
C	=		C	9	E
D	7	G	D	7	G
E	6	G	E	6	G
F	=		F	9	E
G	4	H	G	4	H
H	2	K	H	2	K
I	5	H	I	5	H
J	3	K	J	3	K
K	0		K	0	
L	=		L	=	
M	2	K	M	2	K
N	=		N	=	
O	=		O	=	
P	=		P	=	
Q	=		Q	=	
R	=		R	=	
S	=		S	=	
T	=		T	=	
U	=		U	=	
V	=		V	=	
W	=		W	=	
X	=		X	=	
Y	=		Y	=	
Z	=		Z	=	
R1	=		R1	=	
Q1	=		Q1	=	
P1	=		P1	=	
M1	=		M1	=	
O1	=		O1	=	

Table 2 A adalah langkah ke 3 ghost menentukan rute yang akan dilalui untuk mengejar pacman, untuk melihat rute dapat dilihat pada Gambar 2. Ghost yang awalnya berada pada node H berpindah atau berjalan menjadi ke node G karena proses kedua, karena G sudah ditempati maka G dimasukan kedalam visited. Lalu dari node H menghasilkan 2 jalur yaitu E dan D, dimana E memiliki nilai 2 dan D memiliki nilai 3, sama seperti proses kedua yaitu sistem akan membandingkan ke 2 node tersebut untuk melihat node dengan jalur terpendek setelah dilakukan pengecekan maka didapatkan hasil bahwa node E lebih memiliki jalur dengan rute terpendek sehingga sistem mengambil jalur E, jalur E adalah jalur yang bernilai 2 sedangkan jalur D bernilai 3. Pengecekan pun selesai yaitu dengan memilih E

Visited : K H G

Unvisited : A B C D E F H I J L M N O P Q R S T U V W X Y Z

Table 2 B adalah langkah ke 4 ghost menentukan rute yang akan dilalui untuk mengejar pacman, untuk melihat rute dapat dilihat pada Gambar 2. Ghost yang awalnya berada pada node G berpindah atau berjalan menjadi ke node E karena proses ketiga, karena E sudah ditempati maka E dimasukan kedalam visited. Lalu setelah menempati node E maka system melihat jalur berikutnya yang akan dilalui dari hasil melihat jalur didapat 2 calon visited yaitu C dan F dimana C bernilai 3 dan F bernilai 3, karena ke 2 jalur bernilai sama, maka karena pacman berada posisi atas maka system mengambil node C yang bernilai 3, kegunaan visited adalah mencegah system mengambil jalur yang sudah ditempati sehingga mencegah ghost hanya berputar putar atau looping, pada kenyataanya jika lebih dari 1 node bernilai sama maka ke2 node tersebut disimpan dalam visited.

Visited : K H G E

Unvisited : A B C D F H I J L M N O P Q R S T U V W X Y Z

Tabel 3 Proses 5 dan 6

A			B		
Titik	Jarak dari K	Titik sebelumnya	Titik	Jarak dari K	Titik sebelumnya
A	=		A	13	B
B	11	C	B	11	C
C	9	E	C	9	E
D	7	G	D	7	G
E	6	G	E	6	G
F	9	E	F	9	E
G	4	H	G	4	H
H	2	K	H	2	K
I	5	H	I	5	H
J	3	K	J	3	K
K	0		K	0	
L	=		L	17	B
M	2	K	M	2	K
N	=		N	=	
O	=		O	=	
P	=		P	=	
Q	=		Q	=	
R	=		R	=	
S	=		S	=	
T	=		T	=	
U	=		U	=	
V	=		V	=	
W	=		W	=	
X	=		X	=	
Y	=		Y	=	
Z	=		Z	=	
R1	=		R1	=	
Q1	=		Q1	=	
P1	=		P1	=	
M1	=		M1	=	
O1	=		O1	=	

Table 3 A adalah langkah ke 5 ghost menentukan rute yang akan dilalui untuk mengejar pacman, untuk melihat rute dapat dilihat pada Gambar 2. Ghost yang awalnya berada pada node E berpindah atau berjalan menjadi ke node C karena proses keempat, karena C sudah ditempati maka C dimasukkan kedalam visited.

Lalu dari node C sistem akan melihat mana jalur yang akan menjadi calon visited setelah dilakukan pengecekan didapatkan B dan D, dimana B dengan nilai 2 dan D dengan nilai 2 karena nilai sama maka sistem melihat lokasi pacman karena pacman berada di posisi atas kiri maka sistem memprioritaskan B dengan nilai 2, tapi pada kenyataannya jika terdapat 2 jalur node dengan nilai yang sama maka ke 2 jalur tersebut dimasukkan kedalam visited trik ini dilakukan agar mempersingkat jumlah table dan mempersingkat waktu.

Visited : K H G E C

Unvisited : A B C D F H I J L M N O P Q R S T U V W X Y Z

Table 3 B adalah langkah ke 6 ghost menentukan rute yang akan dilalui untuk mengejar pacman, untuk melihat rute dapat dilihat pada Gambar 2. Ghost yang awalnya berada pada node C berpindah atau berjalan menjadi ke node B karena proses kelima, karena B sudah ditempati maka B dimasukkan kedalam visited dan karena node B langsung bertemu dengan node A tujuan maka node A juga dimasukkan ke visited sebagai hasil pencarian.

Visited : K H G E C B A

Unvisited : A B C D F H I J L M N O P Q R S T U V W X Y Z

Pengejaran pacman dengan rute terpendek menggunakan algoritma dijkstra telah selesai yaitu dengan melalui rute K H G E C B A dengan jumlah jarak 13 hasil ini membuktikan bahwa hasil algoritma dijkstra dengan perhitungan manual yaitu berjarak 13 menghasilkan nilai sama, jadi dapat disimpulkan bahwa algoritma dijkstra cukup bagus menangani kasus pencarian pada tipe map seperti diatas. Karena tidak semua algoritma cocok diterapkan pada semua kasus pencarian terutama jika ada perbedaan dalam mendesign map maka kemungkinan juga kita perlu memodifikasi algoritma dijkstra agar mendukung dengan model map yang kita buat.

4. KESIMPULAN

Berdasarkan dari hasil penelitian dari awal hingga akhir didapatkan beberapa kesimpulan, yaitu :

1. Berdasarkan dari hasil pengujian terhadap algoritma dijkstra, algoritma dijkstra cukup baik melakukan penanganan melakukan pencarian rute terpendek yaitu dengan mendapatkan nilai 13 sesuai dengan perhitungan manual mencari rute terpendek.
2. Algoritma dijkstra hampir sama dengan algoritma greedy dimana dia akan mencocokkan 1 node dengan node lainnya.
3. Node yang tidak terhubung langsung dari titik visited akan diberi tanda ∞ dan node yang terhubung dengan node visited akan ditulis angka sesuai jarak yang dimiliki.
4. dibutuhkan analisis dan juga perhitungan matematika untuk memecahkan masalah rute terpendek tersebut yang akan di implementasikan kedalam game pacman.

5. SARAN

Saran yang dapat diberikan jika ingin mengembangkan jurnal ini adalah pihak peneliti selanjutnya harus dapat memodifikasi algoritma dijkstra untuk menambal kelemahan algoritma tersebut agar mendapatkan hasil yang sempurna.

UCAPAN TERIMA KASIH

Pertama tama saya berterimakasih kepada Tuhan yang maha kuasa atas kehendaknya kami dapat membuat jurnal ini dengan lancar dann saya juga mengucapkan terimakasih kepada bapak Anggit sebagai dosen pembimbing pembuat jurnal ini, sekaligus juga saya mengucapkan terimakasih kepada rekan rekan saya yang membantu dalam pembuatan jurnal

DAFTAR PUSTAKA

- [1] Kresno Muly M. Hanna Alif, 2012, Pembuatan Game Labirint Dengan Menggunakan Blender 3d, Vol. 13 No. 2 JUNI 2012
- [2] Febri Anjar. Desi Purwanti, 2013, Algoritma Dijkstra Pada Aplikasi Searching Hotel Dikota Semarang
- [3] Eko Budi Hartono, 2016, Penerapan Algoritma Dijkstra untuk System Pendukung Keputusan bagi Penentuan Jalur Terpendek Pengiriman Paket Barang Travel, Vol 1, No 1 (2016)
- [4] Ferdiansyah. Ahmad Rizal, 2013, Penerapan Algoritma Dijkstra Untuk Menentukan Rute Terpendek Pembacaan Water Mater Induk PDAM Tirta Kerta Raharja Kabupaten Tanggerang, M Vol.2 No.1 September 2013
- [5] M. Yuliana. Purnomo, 2011, Implementasi Algoritma Greedy Pada Layanan Taksi Wisata.
- [6] D. Rahmawati. Candra, 2013, Implementasi Algoritma Greedy Untuk Menyelesaikan Masalah Knapsack Problem, Vol. 12, No. 3, September 2013
- [7] Fathirul Ilmi. Septi Maisari, 2014, Penerapan Algoritma Greedy Pada Permainan Matematika
- [8] Windi Eka Y. R. Dwiretno Istidiati, 2015, Pencarian SPBU Terdekat dan Penentuan Jarak Terpendek Menggunakan Algoritma DIJKSTRA, Vol 4, No 1
- [9] Vandi Putrandika, 2007, Analisis Penerapan Algoritma Greedy Pada Permainan Capsa, 2007.
- [10] Alim Julianto. Alim Agung, 2017, Implementasi Algoritma Greedy pada Pencarian Langkah Optimal Permainan Mahjong Solitaire, Vol. 1 No. 3 2017