

Aplikasi Program Dinamis untuk Pemilihan Jalur Terpendek pada Permainan GTA IV

Mahardiansyah Kartika / 13509041

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

mahardiansyahkartika@gmail.com

Abstrak—Grand Theft Auto IV (GTA IV) adalah sebuah permainan yang dibuat oleh Rockstar North dan dapat dimainkan pada PlayStation 3, XBOX 360, dan PC. Tujuan utama dari game ini adalah menyelesaikan semua misi yang disediakan.

Game ini berlatar belakang di sebuah kota yang bernama Liberty City. Sebuah kota baru yang memiliki bangunan tinggi dan megah, banyak pejalan kaki dan kendaraan bermotor yang membuat kota ini lebih terasa hidup. Pemain berperan sebagai seseorang yang bernama Nico Bellic. Dia datang ke Liberty City dengan suatu tujuan yang harus dia selesaikan.

Banyak sekali tempat yang dapat dikunjungi dalam GTA IV. Makalah ini membahas tentang aplikasi program dinamis untuk mendapat jalur terpendek pada peta GTA IV agar pemain hanya mengeluarkan sedikit waktu untuk menyelesaikan suatu misi.

Kata Kunci—GTA IV, Jalan Terpendek, Optimum, Program Dinamis.

I. PENDAHULUAN

Dalam permainan Grand Theft Auto IV (GTA IV) ini pemain dapat menjelajahi kota Liberty City. Keadaan di kota ini sangat padat, baik dikarenakan pejalan kaki maupun kendaraan bermotor yang beraneka ragam sehingga kota ini terasa begitu hidup.

Pemain pada game ini berperan sebagai Nico Bellic, pria bekas tentara Rusia yang sangat terlatih dalam hal bertarung dengan tangan kosong dan menggunakan berbagai macam senjata api. Cerita ini diawali ketika Nico datang ke Liberty City untuk mengadu nasib di kota tersebut. Dia memiliki sepupu yang telah cukup lama tinggal di Liberty City yang bernama Roman. Nico tertarik untuk mengadu nasib di Liberty City karena Roman telah mengirim e-mail kepada Nico tentang betapa mewahnya kehidupan dia di Liberty City. Namun ketika Nico telah sampai di Liberty City dia merasa ditipu oleh sepupunya.

Roman datang menjemput Nico dengan menggunakan sebuah mobil yang sangat usang. Kemudian Roman mengajak Nico ke apartemennya yang berukuran kecil dan kumuh. Ternyata kehidupan Roman tidak sesuai dengan yang dia tuliskan pada e-mail yang dia kirim.

Disinilah awal kehidupan Nico di Liberty City. Dia membantu pekerjaan sepupunya sebagai agen taxi. Masalah pun dimulai disini karena setelah sekian lama barulah terungkap bahwa Roman memiliki banyak hutang pada seseorang yang kemudian menimbulkan masalah yang besar. Nico pun akhirnya terlibat pada masalah Roman dan disinilah musuh bermunculan.



Gambar 1 : Cover GTA IV

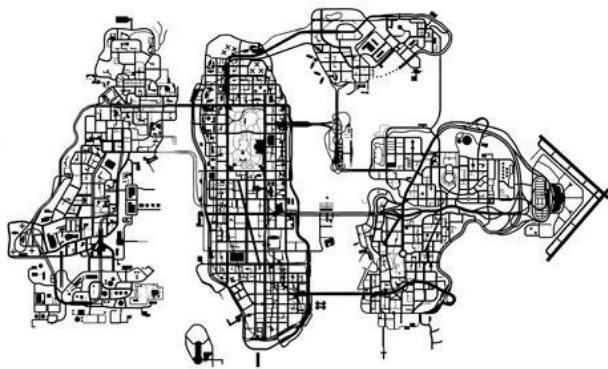
Pada game GTA IV ini pemain berperan sebagai seorang penjahat dan terlibat berbagai kasus kejahatan. Misi yang akan dijalankan akan sangat beraneka ragam, mulai dari meledakkan bangunan, mencuri mobil dan helicopter, menjadi sniper dari atap gedung, mengantarkan seseorang ke suatu tempat sambil melindunginya, membunuh orang-orang penting, hingga merampok bank dan pastinya dikejar-kejar polisi.

Cerita yang pada game ini cukup menarik karena disini pemain akan merasakan persahabatan, permusuhan, bahkan percintaan. Nico nantinya akan mengkhianati dan

dikhianati. Roman juga akan dikejar-kejar oleh para penagih hutang dan Nico harus melindunginya sehingga ia harus berjuang untuk menyelamatkan sepupunya tersebut dan terlibat dalam berbagai kasus pembunuhan.

Game ini menyediakan kebebasan bagi para pemainnya untuk mengunjungi setiap sudut kota Liberty City. Untuk melakukannya tentu saja dibutuhkan sebuah kendaraan. Dari mana mendapatkan kendaraan tersebut? Disini pemain tidak perlu membeli mobil karena bisa mendapatkannya dengan langsung mengambil kendaraan orang lain yang ada di jalan. Nico akan langsung membuka pintu mobil tersebut dan mengeluarkan pemilik kendaraan. Ada pemilik yang melawan dan ada juga pemilik yang lari karena ketakutan.

Banyak sekali tempat-tempat umum yang bisa dikunjungi di GTA IV. Ada tempat makan seperti Burgershot atau Cluckin Bell ataupun stand hotdog di pinggir jalan yang dapat mengisi kembali *health* pemain. Selain tempat makan, ada juga tempat bermain seperti bowling, dart, dan billiard. Tempat lainnya yang bisa dikunjungi juga ialah Strip Club dan Cabaret Show. Dalam game ini kita juga bisa pergi ke Internet Café untuk sekedar melihat email ataupun browsing ke situs-situs yang tersedia.



Gambar 2 : Peta GTA IV

Terdapat tiga pulau utama yang ada di Liberty City. Hanya satu pulau yang dapat diakses pada waktu pertama kali memulai permainan. Jalan menuju pulau kedua ditutup oleh polisi. Pulau pertama berisi oleh perumahan yang kumuh, pulau kedua ditempati oleh perumahan yang mewah, dan pulau ketiga merupakan kawasan industri yang memiliki banyak sekali pabrik. Terdapat jalan tol yang dapat digunakan dan tentunya pemain harus berhenti dan membayar pada *toolboth* yang tersedia.

Map yang disediakan memiliki legenda peta yang memudahkan pemain. Tempat-tempat umum memiliki *icon* masing-masing. Juga terdapat GPS yang akan menunjukkan jalan legal terdekat untuk mencapai tempat tujuan.

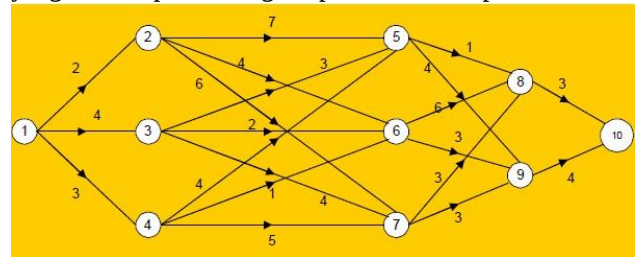
II. ALGORITMA PROGRAM DINAMIS

Program dinamis merupakan metode pemecahan masalah dengan cara menguraikan solusi menjadi

sekumpulan langkah (*step*) atau tahapan (*stage*), sedemikian sehingga solusi dari persoalan dapat dipandang dari serangkaian keputusan yang saling berkaitan. *Stage* (tahap) adalah bagian persoalan yang mengandung *decision variable*.

Program dinamis menyediakan prosedur sistematis untuk menentukan kombinasi keputusan yang optimal. Tujuan utama dari program dinamis adalah agar mempermudah penyelesaian persoalan optimasi yang memiliki karakteristik tertentu.

Penyelesaian persoalan dengan menggunakan metode ini memiliki sejumlah berhingga pilihan yang mungkin dan solusi pada setiap tahap dibangun dari hasil solusi tahap sebelumnya serta menggunakan persyaratan optimasi dan kendala untuk membatasi sejumlah pilihan yang harus dipertimbangkan pada suatu tahap.



Gambar 3 : Graf Berbobot suatu lintasan

Pada program dinamis, rangkaian keputusan yang optimal dibuat dengan menggunakan Prinsip Optimalitas yaitu jika solusi total optimal, maka bagian solusi sampai tahap ke- k juga optimal.

Prinsip optimalitas berarti bahwa jika kita bekerja dari tahap k ke tahap $k+1$, kita dapat menggunakan hasil optimal dari tahap k tanpa harus kembali ke tahap awal. Prinsip optimalitas menjamin bahwa pengambilan keputusan pada suatu tahap adalah keputusan yang benar untuk tahap-tahap selanjutnya.

Program dinamis mempunyai karakteristik sebagai berikut :

1. Persoalan dapat dibagi menjadi beberapa tahap, yang pada setiap tahap hanya diambil satu keputusan yang optimal.
2. Masing-masing tahap terdiri atas sejumlah status yang berhubungan dengan tahap tersebut.
3. Hasil keputusan yang diambil pada setiap tahap ditransformasikan dari status yang bersangkutan ke status berikutnya pada tahap berikutnya.
4. Ongkos pada suatu tahap bergantung pada ongkos tahap-tahap sebelumnya dan meningkat secara teratur dengan bertambahnya jumlah tahapan.
5. Keputusan terbaik pada suatu tahap bersifat independen terhadap keputusan yang dilakukan tahap sebelumnya.
6. Adanya hubungan rekursif yang mengidentifikasi keputusan terbaik untuk setiap status pada tahap k memberikan keputusan terbaik untuk setiap status pada tahap $k+1$.
7. Prinsip optimalitas berlaku di persoalan tersebut.

Dalam menyelesaikan program dinamis terdapat dua pendekatan yaitu maju (*forward* atau *up-down*) dan mundur (*backward* atau *bottom-up*). Misalkan $x_1, x_2, x_3, \dots, x_n$ menyatakan peubah keputusan yang harus dibuat masing-masing untuk tahap 1, 2, 3, ..., n, maka

- Program dinamis maju. Program dinamis bergerak mulai dari tahap 1, terus maju ke tahap 2, 3, dan seterusnya sampai tahap n. Runtunan peubah keputusan adalah $x_1, x_2, x_3, \dots, x_n$.
- Program dinamis mundur. Program dinamis mulai dari tahap n, terus mundur sampai ke tahap n-1, n-2, sampai ke tahap 1. Runtunan peubah keputusan adalah $x_n, x_{n-1}, x_{n-2}, \dots, x_1$.

Terdapat empat langkah untuk mengembangkan algoritma program dinamis :

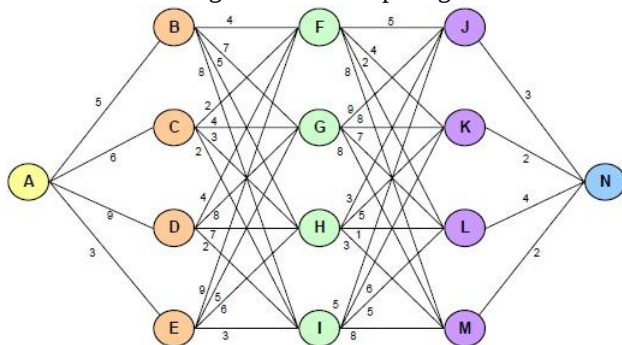
- Karakteristik struktur solusi optimal.
- Definisikan secara rekursif nilai solusi optimal.
- Hitung nilai solusi optimal secara maju atau mundur.
- Konstruksi solusi optimal.

III. METODE PEMECAHAN MASALAH

A. Aplikasi Program Dinamis untuk mencari Lintasan Terpendek (Shortes Path)

Masalah untuk pencarian jalur pada suatu graf dapat dilakukan dengan menggunakan program dinamis. Untuk mempermudah hal tersebut pendekatan yang digunakan adalah program dinamik mundur.

Diberikan suatu graf berbobot seperti gambar di bawah.



Gambar 4 : Graf Berbobot

Program dinamis akan digunakan untuk mencari jarak terdekat dari kota A ke kota N dan memperlihatkan lintasan yang dilewati.

Titik	Jarak	Titik	Jarak	Titik	Jarak	Titik	Jarak	Titik
A	5	B	4	F	5	J	3	N
	6	C	7		4	K	2	
	9	D	5		2	L	4	
	3	E	8		8	M	2	
			2	G	9			
			4		8			
			3		7			
			2		8			
			4	H	3			
			8		5			
			7		1			
			2		3			
			9	I	5			
			5		6			
			8		5			
			3		8			

Gambar 5 : Tabel dari graf

Untuk menyelesaikan kasus tersebut dengan program dinamis mundur. Misalkan x_1, x_2, x_3, x_4 adalah simpul-simpul yang dikunjungi pada tahap k ($k = 1, 2, 3, 4$). Maka rute yang dilalui adalah $x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4$.

Tahap (k) merupakan proses memilih simpul tujuan berikutnya (ada 4 tahap). Status (s) yang berhubungan dengan masing-masing tahap adalah simpul-simpul dalam graf.

Algoritma program dinamis menggunakan relasi rekurens. Berikut ini merupakan relasi rekurens yang menyatakan lintasan terpendek dari status s ke x_4 pada tahap k :

$$f_4(s) = c_{sx_4} \quad (\text{basis})$$

$$f_k(s) = \min \{c_{sx_k} + f_{k+1}(x_k)\}, \quad (\text{rekurens})$$

$$k = 1, 2, 3$$

- x_k : peubah keputusan pada tahap k
- c_{sx_k} : bobot sisi dari s ke x_k
- $f_k(s, x_k)$: total bobot lintasan dari s ke x_k
- $f_k(s)$: nilai minimum dari $f_k(s, x_k)$

Dengan menggunakan program dinamis mundur maka untuk mendapatkan $f_1(1)$ dengan cara mencari $f_4(s), f_3(s), f_2(s)$ terlebih dahulu. Berikut merupakan tahapan-tahapan yang dilakukan untuk mendapatkan solusi optimum.

Tahap 4

s	$f_4^*(s)$	x_4^*
J	3	N
K	2	N
L	4	N
M	2	N

Tahap 3

s	$f_3 = c_s + f_4^*$				$f_3^*(s)$	x_3^*
	J	K	L	M		
F	8	6	6	10	6	KL
G	12	10	11	10	10	KM
H	6	7	5	5	5	LM
I	8	8	9	10	8	JK

Tahap 2

s	$f_2 = c_s + f_3^*$				$f_2^*(s)$	x_2^*
	F	G	H	I		
B	10	17	10	16	10	FH
C	8	14	8	10	8	FH
D	10	18	12	10	10	FI
E	15	15	11	11	11	HI

Tahap 1

s	$f_1 = c_s + f_2^*$				$f_1^*(s)$	x_1^*
	B	C	D	E		
A	15	14	19	14	14	CE

Gambar 6 : Tahapan Pencarian Solusi

Solusi optimum dapat dibaca pada tabel di bawah ini.

	x_1	x_2	x_3	x_4	Panjang Lintasan Terpendek
A	C	F	K	N	14
			L	N	14
		H	L	N	14
			M	N	14
	E	H	L	N	14
			M	N	14
		I	J	N	14
			K	N	14

Tabel 1 : Tabel Solusi Optimum

Jadi terdapat delapan lintasan terpendek dari kota A ke kota N, yaitu

$A \rightarrow C \rightarrow F \rightarrow K \rightarrow N$
 $A \rightarrow C \rightarrow F \rightarrow L \rightarrow N$
 $A \rightarrow C \rightarrow H \rightarrow L \rightarrow N$
 $A \rightarrow C \rightarrow H \rightarrow M \rightarrow N$
 $A \rightarrow C \rightarrow H \rightarrow L \rightarrow N$
 $A \rightarrow C \rightarrow H \rightarrow M \rightarrow N$
 $A \rightarrow C \rightarrow I \rightarrow J \rightarrow N$
 $A \rightarrow C \rightarrow I \rightarrow K \rightarrow N$

Panjang kedelapan lintasan tersebut sama, yaitu 14.

B. Aplikasi Program Dinamis pada Game GTA IV

Gambar di bawah ini ialah potongan peta pada game.



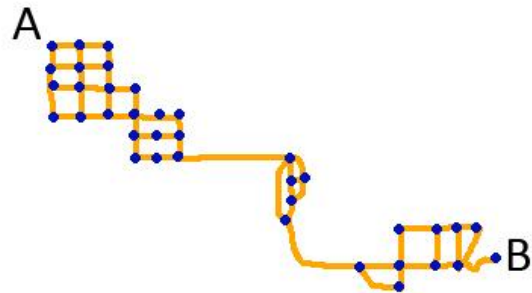
Gambar 7 : Potongan Peta

Sekarang kita konversi terlebih dahulu peta ini menjadi suatu graf. Caranya dengan menjadikan jalan sebagai suatu busur dan persimpangan jalan menjadi suatu simpul.



Gambar 8 : Konversi Peta ke Graf

Busur yang merepresentasikan jalan diberi warna kuning dan simpul yang merepresentasikan persimpangan diberi warna biru. Hasil dari konversi tersebut dapat dilihat seperti gambar dibawah ini.

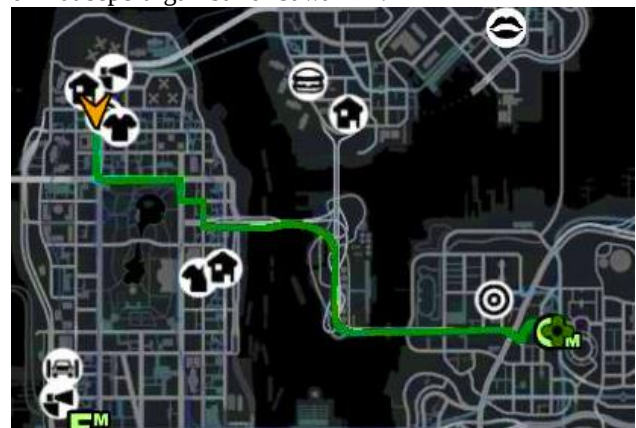


Gambar 9 : Graf dari peta

Kita akan menghitung jarak terpendek dari titik A ke titik B. Untuk mengukur jarak antar persimpangan penulis menggunakan fitur penghitung jarak yang telah disediakan oleh game GTA IV.

Setelah didapatkan tabel jarak antar setiap persimpangan maka tinggal dilakukan cara yang sama seperti yang telah di bahas pada bagian A. Pada graf kali ini lebih kompleks dibandingkan graf sebelumnya yang hanya memiliki 4 langkah. Graf kali ini memiliki 19 langkah pengerjaan untuk mendapatkan solusi optimum. Cara menghitung untuk mendapatkan jarak terpendek persis sama dengan yang telah ditulis di bagian A sehingga penulis tidak perlu menjabarkan secara detail *step-by-step* penggunaan algoritma program dinamis mundur untuk mendapatkan rute terbaik.

Setelah dilakukan pencarian menggunakan algoritma program dinamis mundur didapatkan suatu jalur yang optimum dari titik A menuju titik B. Jalur tersebut dapat dilihat seperti gambar di bawah ini.



Gambar 10 : Jarak optimum

Garis yang berwarna hijau merupakan jalur yang optimum untuk pemain berpindah dari titik A menuju titik B. Sebenarnya banyak ditemukan jalur alternatif yang sama optimumnya dengan jalur yang ditampilkan diatas tetapi penulis hanya menampilkan satu jalur saja.

IV. ANALISIS

Aplikasi algoritma program dinamis sangat membantu dalam pencarian jalan terpendek dari suatu graf yang tersedia. Dengan pengaplikasian algoritma ini, proses komputasi untuk melakukan pencarian akan jauh berkurang jika dibandingkan dengan cara pencarian *brute force*.

Algoritma program dinamis dapat diaplikasikan apabila peningkatan ongkos secara linear dan diskrit sehingga optimasi parsial dapat dilakukan. Pendekatan program dinamis mundur bergerak dari tahap n , terus mundur ke tahap $n-1$, $n-2$, ..., 2 , 1 . Urutan variable keputusan adalah x_n , x_{n-1} , ..., x_2 , x_1 . Adapun kompleksitas waktu pada algoritma ini adalah $O(|s_1|*|s_2|)$, jika panjang kedua string adalah ' n '. Kompleksitas ruangnya juga sama jika seluruh matriks disimpan untuk merunut balik untuk mencari optimal optimal *alignment*. Jika nilai *edit distance* dibutuhkan, hanya dua baris dari matriks yang dialokasi dan matriks tersebut dapat mengalami *recycle* dan kompleksitas ruangnya berkurang menjadi $O(n)$.

Pengaplikasian algoritma Program Dinamis terbukti memiliki kelebihan dibandingkan pencarian dengan cara biasa, yaitu waktu komputasi yang jauh lebih cepat. Dalam permainan yang kompleks seperti GTA IV sangat diperlukan algoritma-algoritma yang sangat efisien dikarenakan banyak sekali hal yang harus diproses pada game ini. Setiap masalah harus diselesaikan oleh algoritma yang mangkus agar waktu komputasi yang dibutuhkan menjadi sangat cepat.

V. SIMPULAN

Kecepatan komputasi dalam permainan seperti Grand Theft Auto IV (GTA IV) sangatlah penting. Untuk menunjang kecepatan proses komputasi dalam suatu permainan maka diperlukan algoritma yang tepat untuk setiap persoalan yang dihadapi.

Algoritma Program Dinamis merupakan salah satu paradigma desain algoritma yang sudah biasa digunakan untuk persoalan optimasi. Algoritma ini dapat diaplikasikan apabila peningkatan ongkos secara linear dan diskrit sehingga optimasi parsial dapat dilakukan.

Pengaplikasian algoritma program dinamis mundur pada permainan GTA IV ini terdapat pada proses pencarian jalur terpendek, algoritma ini sudah cukup baik untuk menyelesaikan persoalan pemilihan jalur terpendek karena persoalan telah dibagi menjadi beberapa tahap.

REFERENSI

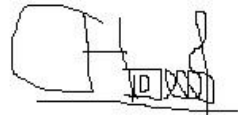
- [1] Munir, Rinaldi, Strategi Algoritmik, Teknik Informatika ITB. Bandung. 2007.
- [2] http://id.wikipedia.org/wiki/Grand_Theft_Auto_IV
Tanggal akses 7 Desember 2011.
- [3] <http://indonesiaindonesia.com/f/36141-review-grand-theft-auto-4-a/>
Tanggal akses 8 Desember 2011.

- [4] <http://www.informatika.org/~rinaldi/>
Tanggal akses 8 Desember 2011.
- [5] <http://luk.staff.ugm.ac.id/optimasi/pdf/Tahap4-00.pdf>
Tanggal akses 8 Desember 2011.
- [6] <http://www.scribd.com/doc/48414882/ALGORITMA-PROGRAM-DINAMIS>
Tanggal akses 8 Desember 2011.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 8 Desember 2011



Mahardiansyah Kartika
13509041