JITE, 2 (2) Januari 2019 ISSN 2549-6247 (Print) ISSN 2549-6255 (Online)

JITE
(Journal of Informatics and Telecommunication Engineering)

Available online http://ojs.uma.ac.id/index.php/jite

PENERAPAN ALGORITMA DYNAMIC *PROGRAMMING* PADA PERGERAKAN LAWAN DALAM PERMAINAN *POLICE & THIEF*

IMPLEMENTATION OF DYNAMIC PROGRAMMING ALGORITHM FOR NPC MOVEMENT IN POLICE AND THIEF GAME

Insidini Fawwaz[1)*], Agus Winarta[1)], Selvianna[1)], Jevon Jose Ramli[1)], Lenny Marthalina Waruwu[1)]

1) Universitas Prima Indonesia , Indonesia

*Coresponding Email: insi.dini@gmail.com

Abstrak

Game memiliki arti dasar permainan yang merujuk pada pengertian kelincahan intelektual. Di dalam penerapannya, sebuah Game tentu memerlukan sebuah AI (Artificial Intelligence), dan AI yang digunakan dalam pembangunan Game police and thief ini adalah algoritma Dynamic *Programming* . Algoritma ini merupakan algoritma pencarian untuk menemukan rute terpendek, algoritma A* mencari rute terpendek dengan menjumlahkan jarak sebenarnya dengan jarak perkiraan sehingga membuatnya optimum dan complete. Police and Thief merupakan Game dimana pemain berusaha lari menjauh dari polisi. Genre dari game ini adalah arcade, dibangun dengan microsoft visual studio 2008, AI yang digunakan adalah algoritma Dynamic Programming yang digunakan untuk melakukan pencarian jalur guna untuk menyerang pemain. Hasil uji ini adalah polisi dalam game ini dari 10 percobaan yang dilakukan mencapai 90% keberhasilan berhasil mencari jalur terpendek dan optimal yang ditentukan oleh algoritma Dynamic Programming untuk menyerang pemain.

Kata Kunci: AI; game, Dynamic Programming, rute, terpendek.

Abstract

Games have the basic meaning of games, games in this case refer to the notion of intellectual agility. In its application, a Game certainly requires an AI (Artificial Intelligence), and the AI used in the construction of this police and thief game is the Dynamic Programming algorithm. This algorithm is a search algorithm to find the shortest route with the minimum cost, algorithm Dynamic Programming searches for the shortest route by adding the actual distance to the approximate distance so that it makes it optimum and complete. Police and thief is a game about a character who will try to run from police. The genre of this game is arcade, built with microsoft visual studio 2008, the AI used is the Dynamic Programming algorithm which is used to search the path to attack players. The results of this test from 10 tests 90% success rate for Dynamic Programming to help police in this game to find the closest path to attack players.

Keywords: AI, game, Dynamic Programming, path, shortest.

How to Cite: Fawwaz, I., Winarta, A., Selvianna, Ramli, J. J., & Waruwu, L. M. (2019). Penerapan Algoritma Dynamic Programming Pada Pergerakan Lawan Dalam Permainan Police & Thief. JITE *(Journal Of Informatics And Telecommunication Engineering)*. 2 (2):114-121

PENDAHULUAN

Game pada masa kini sudah berkembang sangat pesat. Dulunya *game* merupakan sarana hiburan bagi pemain. Tetapi pada era modern ini, *game* merupakan sarana pembelajaran, olahraga, maupun acara perlombaan yang dibuat untuk pemain. *Game* juga merupakan sebuah bagian dari media dikarenakan sifatnya yang dapat membantu dalam penyampaian pesan, berkomunikasi, dan dapat meningkatkan kinerja otak. *Game Police & Thief* disini merupakan salah satu *game* dengan tipe *maze* yang dapat bermanfaat untuk meningkatkan alur berpikir bagi pengguna di dalam menyelesaikan suatu permainan. *Police & Thief* adalah sebuah permainan yang dapat diselesaikan dengan cara pemain (*Thief*) mengambil kumpulan koin di labirin sambil menghindari beberapa 'polisi' yang bergerak secara acak (*random*) untuk menangkap pemain.

Adapun penelitian menggunakan algoritma *Dynamic Programming* dalam mencari solusi terbaik dalam pencarian jalur terdekat seperti metode Dijkstra, Floyd-Warshall, Bellman-Fort, A* (A Star), dan Algoritma Greedy. Algoritma *Dynamic* Programming merupakan jenis algoritma yang menggunakan pendekatan penyelesaian masalah dengan mencari nilai maksimum sementara pada setiap langkahnya.

Penelitian mengenai pencarian jalur terdekat telah banyak dilakukan oleh para peneliti sebelumnya seperti penelitian Alvin Juvianto (2017) pada penelitian dengan judul Implementasi Algoritma Greedy pada Pencarian Langkah Optimal Permainan Mahjong Solitaire menggunakan algoritma pencarian greedy untuk mencari langkah optimal bagi NPC pada permainan solitaire.

Kemudian pada penelitian yang dilakukan oleh Widyaiswara Madya (2015) pada penelitian dengan judul Penggunaan *Dynamic Programming* Untuk Memilih Jalur Transportasi Dengan Biaya Minimum dan Fathoni, M dengan Triprabowo (2014) dengan judul Pencarian Rute Terpendek denan Menggunakan *Dynamic Programming* dimana algoitma *Dynamic Programming* digunakan untuk mencari jalur terpendek dengan membandingkan jarak terpendek yang didapatkan apabila menggunakan *gmaps.*

Pada penelitian terdahulu yang dilakukan oleh Nurhidayati (2010) Penerapan pendekaan dynamic programming mampu menyelesaikan beberapa masalah seprtti alokasi, muatan, capital budgeting, pengawasan persediaan, penentuan jalur terpendek, dan lain-lain.

METODE PENELITIAN

Kecerdasan Buatan merupakan kecerdasan yang ditambahkan kepada suatu sistem dengan algoritma agar dapat bekerja atau berpikir seperti manusia.

Sistem AI sekarang ini sering digunakan dalam bidang ekonomi, obat-obatan, teknik dan militer, seperti yang telah dibangun dalam beberapa aplikasi perangkat lunak komputer rumah dan video *game*.

Kecerdasan buatan berasal dari bahasa Inggris "*Artificial Intelligence*" atau disingkat *AI*, yang terbagi atas dua kata yakni *intelligence* adalah kata sifat yang berarti cerdas, dan *artificial* artinya buatan.

Sutojo, et. al., (2011) Kecerdasan buatan yang dimaksud di sini merujuk pada mesin yang mampu berpikir, menimbang tindakan yang akan diambil, dan mampu mengambil keputusan seperti yang dilakukan oleh manusia.

Suyanto (2011) Sebagian kalangan menerjemahkan *Artificial Intelligence* sebagai kecerdasan buatan, kecerdasan artifisial, intelijensia artifisial, atau intelijensia buatan.

Menurut Suyanto (2011) Para ahli mendefinisikan AI secara berbeda-beda tergantung pada sudut pandang mereka masing-masing. Ada yang fokus pada logika berpikir manusia saja, tetapi ada juga yang mendefinisikan AI secara lebih luas pada tingkah laku manusia.

Jadi dapat disimpulkan bahwa, kecerdasan buatan (*artificial intelligence*) adalah kemampuan komputer untuk meniru pola pikir manusia dalam menyelesaikan suatu permasalahan tertentu. Algoritma *Dynamic Programming*, *dynamic programming* atau program dinamik adalah suatu teknik matematika yang digunakan untuk mengoptimalkan proses pengambilan keputusan secara bertahap-ganda contoh, jika kita menulis solusi rekursif sederhana untuk Angka Fibonacci, kita mendapatkan kompleksitas waktu yang eksponensial dan jika kita mengoptimalkannya dengan menyimpan solusi dari sub-masalah, kompleksitas waktu berkurang menjadi linear

Menurut Luknanto (2013) Tidak seperti program linier, Program Dinamik (PD) tidak mempunyai standar formulasi matematik. PD lebih merupakan suatu cara umum untuk melakukan optimasi dengan persamaan matematik yang cocok dengan masalah yang dihadapi.

Metode *Dynamic Programming* sesuai dengan namanya dalam menyelesaikan masalahnya *Dynamic Programming* dapat diaplikasikan dalam berbagai masalah pemograman matematik karena *Dynamic*

Programming cenderung lebih fleksibel dibandingkan algoritma optimasi yang lain.

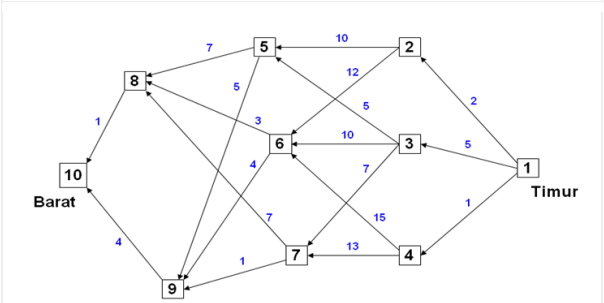
Metode Program Dinamik ini pertama kali dikembangkan oleh Richard E. Bellman pada tahun 1957. Dalam teknik ini, keputusan yang menyangkut suatu persoalan dioptimalkan secara bertahap dan bukan sekaligus. Menurut Munir (2008) Pada program dinamis, rangkaian keputusan yang optimal dibuat dengan menggunakan Prinsip Optimalitas

Jadi, inti dari teknik ini adalah membagi satu persoalan menjadi beberapa bagian persoalan yang dalam program dinamik disebut tahap, kemudian memecahkan tiap tahap dengan mengoptimalkan keputusan atas tahap sampai seluruh persoalan telah terpecahkan. Keputusan optimal atas seluruh persoalan ialah kumpulan dari sejumlah keputusan optimal atas seluruh tahap yang kemudian disebut sebagai kebijakan optimal.

Pendekatan Program Dinamik didasarkan pada prinsip optimasi yang mengatakan kurang lebih demikian, "Suatu kebijakan optimal mempunyai sifat bahwa apapun keadaan dan keputusan awal, keputusan berikutnya harus membentuk suatu kebijakan optimal dengan memperhatikan keadaan dari hasil keputusan pertama".

```
for all i, 1 <= i <= N

dp[i][0] = dp[i-1][0] + cost[i][0];

for all j, 1 <= j <= j

dp[0][j] = dp[0][j-1] + cost[0][j];

otherwise

dp[i][j] = max(dp[i-1][j], dp[i][j-1]) + cost[i][j];
```

Contoh penyelesaian algoritma *Dynamic* untuk masalah pencarian jarak terdekat dari kota timur ke barat:

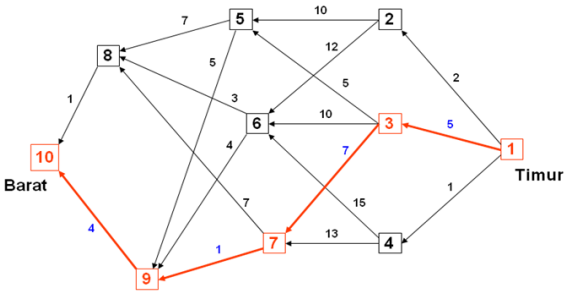


Gambar 1. Rute Perjalanan dari timur ke barat

Untuk mencari jarak terpendek dari Timur ke Barat, sebuah algoritma *Dynamic Programming* akan menjalankan langkah-langkah :

Kita menemukan bahwa jalur dengan biaya minimal adalah jalur dari kota 1 ke kota 3. Selanjutnya kita lihat kita bisa mengetahui bahwa dari kota 3, biaya yang paling minimum adalah kita melanjutkan perjalanan ke kota 7.

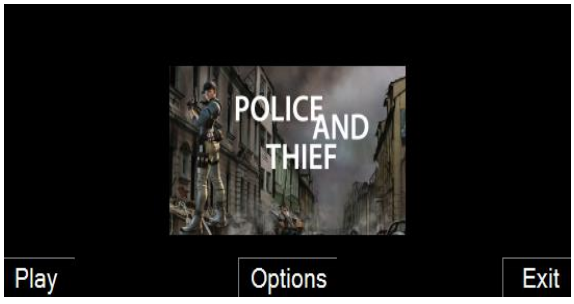
Kita bisa melihat bahwa ketika kita berangkat dari kota 7, jalur yang membutuhkan biaya minimum adalah menuju ke kota 9. Dan dari kota 9 kita mengakhiri perjalanan di 10. Sebagai kesimpulan, jalur perjalanan yang membutuhkan biaya yang minimum adalah jalur dari kota 1 ke kota 3, kemudian ke kota ke 7, selanjutnya ke kota 9, dan berakhir di kota 10. Dengan total biaya 5 + 7 + 1 + 4 = 17. Jalur transportasi yang mempunyai biaya minimum tersebut bisa kita lihat pada gambar 2.



Gambar 2. Rute Perjalanan dari timur ke barat

HASIL DAN PEMBAHASAN

Ketika *game* dijalankan, *form* yang pertama kali muncul adalah *form* 'Main' seperti terlihat pada gambar berikut:

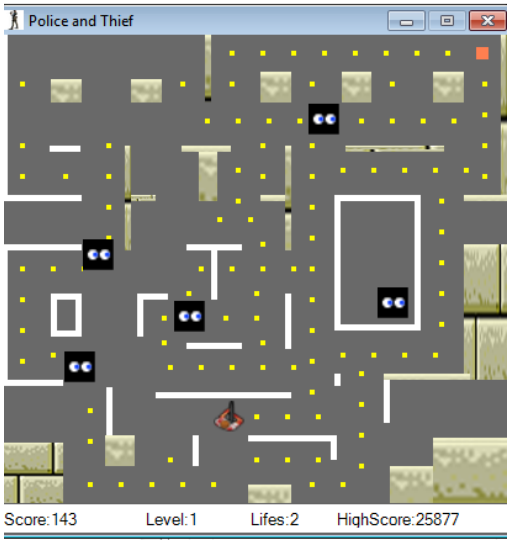


Gambar 3 Form Main

Pada form ini terdapat link "*Play Game*" dengan menekan link "*Play Game*" atau menekan latar dari tampilan *form Main user* kemudian akan dibawa ke tampilan *game* seperti berikut ini :

Gambar 4 Form Game

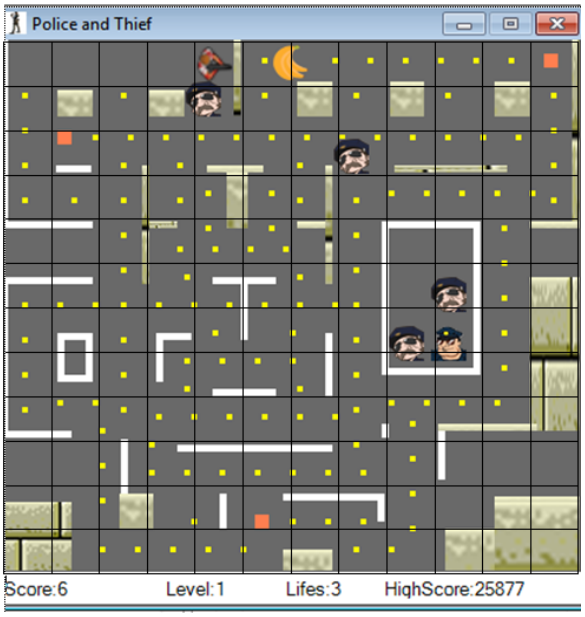
Pada permainan *Police and Thief* ini, pemain akan diberikan 3 kesempatan untuk bermain dan memiliki 3 buah powerup yang dapat membuat polisi menjadi takut dan berjalan mundur dari pemain. Tampilan polisi menjadi taktut dapat dilihat pada gambar 6 sebagai berikut :



Gambar 5 Tampilan polisi menjadi takut dan mundur

Dari percobaan yang dilakukan pada *game Police and Thief* algoritma *Dynamic Programming* dapat melakukan pencarian jarak terdekat menuju ke node pemain, sehingga *police* dapat digerakkan untuk mengejar dan mengalahkan pemain.

Contoh penyelesaian algoritma *Dynamic* pada permainan *police* and *thief* :



Gambar 6 Tampilan *game* apabila menampakkan path

Pada gambar 1 diatas tampak *game police* and *thief* apabila ditampilkan dalam bentuk path dimana path ini merupakan langkah yang akan dilalui oleh polisi atau *user*. Ketika *user* atau polisi bergerak, maka *user* atau polisi akan mengisi path yang tersedia. Sehingga algoritma *Dynamic Programming* berusaha mencari lokasi *user* melalui path. Contoh implementasinya dalam path 6x6 sebagai berikut :

Gambar 7 Hasil perhitungan *Dynamic Programming* untuk langkah polisi

Pada gambar diatas polisi yang berada pada posisi 36 harus menuju ke posisi satu dimana pada posisi 9, 11, dan 19 merupakan jalan yang tidak bisa dilalui sama sekali sehingga langkah yang akan coba dilalui oleh polisi adalah sebagai berikut :

1. 36-30-24-23-22-16-10-4-3-2-8-14-13-7-1
2. 36-30-24-23-22-16-17-18-12-6-5-4-3-2-8-14-13-7-1
3. 36-30-24-23-22-16-15-14-13-7-1 (A)
4. 36-30-29-23-22-16-15-14-13-7-1 (B)
5. 36-35-29-23-22-16-15-14-13-7-1 (C)

Pencarian jarak dilakukan dengan mencari jarak terdekat terlebih dahulu. Kemudian mencari segala kemungkinan jalan dari arah kiri-kanan-atas-bawah dengan syarat jalan yang dilalui tidak terhalang *obstacle.* Dari kemungkinan diatas pada pencarian jarak ke 3,4,dan5 memiliki jarak kotak terkecil yakni 11 kotak.

Dari ketiga jarak yang ditemukan, program akan mengambil jarak terpendek pertama. Sehingga dihasilkan jarak yang dilalui adalah sebagai berikut :

Gambar 8 Hasil perhitungan *Dynamic Programming* untuk langkah polisi

Selanjutnya dilakukan percobaan permainan *game* sambil melakukan proses *debugging*. Percobaan dilakukan dengan menjalankan aplikasi permainan *police & thief* sebanyak 10 kali. Kemudian penulis akan memantau jalan yang dilalui oleh polisi apakah jalan dilalui sudah efisien atau belum.

Tabel 1. Tabel percobaan *case*

No	Percobaan	Hasil	
1	Percobaan pertama pemain tidak bergerak	Jalur yang dilalui polisi adalah jalur yang paling efisien	Berhasil
2	Percobaan berikutnya pemain bergerak maju ke kotak selanjutnya	Jalur yang dilalui polisi adalah jalur yang paling efisien	Berhasil
3	Percobaan berikutnya pemain bergerak maju ke kotak selanjutnya	Jalur yang dilalui polisi adalah jalur yang paling efisien	Berhasil
4	Percobaan berikutnya pemain maju 2 kotak kemudian mundur sekali	Jalur yang dilalui polisi adalah jalur yang paling efisien	Berhasil
5	Percobaan berikutnya pemain maju ke sembarang arah	Jalur yang dilalui polisi adalah jalur yang paling efisien	Berhasil
6	Percobaan berikutnya pemain bermain seperti biasa berusaha mengumpulkan koin	Jalur yang dilalui oleh polisi untuk mengejar penjahat adalah jalur yang paling efisien	Berhasil
7	Percobaan berikutnya pemain memakan powerup, Polisi mulai berjalan	Jalur yang dilalui oleh polisi untuk mengejar penjahat adalah jalur yang paling efisien	Berhasil

	mundur		
8	Percobaan dilakukan kembali untuk permainan berikutnya melakukan *case* yang sama	Jalur yang dilalui oleh polisi untuk mengejar penjahat adalah jalur yang paling efisien	Berhasil
9	Percobaan dilakukan kembali untuk permainan berikunya melakukan *case* yang sama lagi	Jalur yang dilalui oleh polisi untuk mengejar penjahat adalah jalur yang paling efisien	Berhasil
10	Percobaan yang dilakukan kembali kali ini dengan bergerak maju mundur dan bergerak ke sembarang arah berdekatan dengan jalan yang diblok	Polisi kesulitan bergerak menuju kearah penjahat	Gagal

Hasil dari percobaan yang dilakukan algoritma *Dynamic Programming* dapat bekerja dengan baik dalam mencari jalur yang paling efisien, namun ada kesalahan algoritma terjadi ketika penjahat bergerak bolak balik didepan jalan yang terblok.

SIMPULAN

Dari implementasi algoritma *Dynamic Programming* yang dilakukan. Algoritma yang bekerja ketika *timer* untuk menjalankan polisi diaktifkan. Polisi kemudian akan meyimpan posisi *user* dan akan menghitung jarak terdekat menuju ke arah *user* dengan menggunakan algoritma *Dynamic Programming*.

Dari 10 percobaan yang dilakuakan dengan mencoba beberapa *case*, kesalahan hanya terjadi ketika *user* bergerak bolak balik disekitar blok. Namun secara keseluruhan dari percobaan yang dilakukan, 90% keberhasilan Algoritma *Dynamic Programming* dalam mencari jarak terpendek untuk polisi.

DAFTAR PUSTAKA

Fathoni, M. Dan Triprabowo, (2014). Pencarian Rute Terpendek dengan Menggunakan *Dynamic Programming*, Universitas Airlangga, Surabaya.

Juvianto Alvin, Halim Agung, (2017). Algoritma Greedy pada Pencarian Langkah Optimal Permainan Mahjong Solitaire menggunakan algoritma pencarian greed. Teknik Informatika, Fakultas Teknologi dan Desain, Universitas Bunda Mulia.

Luknanto, D., (2013), Program Dinamik, Jurusan TeknikSipil, Fakultas Teknik, Universitas Gadjah Mada, Yogyakarta.

Munir, R., (2008), Program Dinamis, Bahan Kuliah Strategi Algoritma, Jurusan Teknik Informatika, Sekolah Informatika dan Elektro, Institut Teknologi Bandung.

Nurhidayati, F., U., (2010), Penggunaan Program Dinamik untuk Menentukan Total Biaya Minumum pada Perencanaan Produksi dan Pengendalian Persediaan, Skripsi, Jurusan Matematika, Fakultas Sains danTeknologi, Universitas Islam Negeri Maula Malik Ibrahim, Malang.

Suyanto. (2011). *Artificial Intelligence*. Bandung: Informatika Bandung.

T. Sutojo, S.Si., M.Kom., Edy Mulyanto, S.Si., M.Kom., Dr. Vincent Suhartono, (2011), *Kecerdasaan Buatan*, , Yogyakarta, Andi

Widyaiswara Madya. (2015). Penggunaan *Dynamic Programming* Untuk Memilih Jalur Transportasi Dengan Biaya Minimum. Malang.