

Aplikasi Algoritma Greedy pada Optimasi Pelaksanaan Misi dalam Permainan Assassins Creed : Revelations

Miftahul Mahfuzh – 13513017
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13513017@std.stei.itb.ac.id

Abstrak--Makalah ini berisi tentang aplikasi algoritma greedy untuk pemilihan misi pada permainan Assassin's Creed : Revelations. Pemilihan misi dalam permainan ini dipengaruhi oleh jarak pemain terhadap start point misi pada map, sehingga player harus memilih mana misi yang paling optimal untuk di jalankan terlebih dahulu. Di dalam makalah saya ini, saya akan membahas mengenai algoritma greedy sebagai implementasi pemilihan jalur optimal untuk seluruh misi yang sedang tersedia pada permainan ini

Kata kunci—greedy, efisiensi, assassins creed, misi

1. PENDAHULUAN

A) Latar Belakang



Assassins Creed : Revelations adalah sebuah game yang memiliki genre open world action

adventure. Permainan produksi Ubisoft Montreal ini pertama kali dirilis pada tahun 2002 dan termasuk game terlaris pada masanya.

Game ini bercerita tentang Ezio Auditore, seorang mentor assassins yang ingin menyingkap sebuah rahasia terpendam dalam sejarah persaudaraan assassins di masa lalu, yang berkaitan dengan nenek moyangnya yang juga seorang assassin, yang bernama Altair Ibnu La'had. Sehingga dia melakukan perjalanan ke Turki dan melakukan penyelidikan di sana. Game ini secara umum mengambil latar belakang Turki pada zaman kekhalifahan utsmaniyah (Ottoman), dan berkisah tentang kehidupan Ezio selama di Turki hingga pada akhirnya ia berhasil menyingkap rahasia yang dicarinya tersebut.

Pada kasus topik makalah ini, kita diberi berbagai macam pilihan misi saat permainan, sedangkan dari pilihan-pilihan tersebut kita dapat memilih satu misi yang akan dijalankan. Maka akan di implementasikan algoritma greedy untuk menentukan misi mana yang sebaiknya diambil sehingga total waktu yang dibutuhkan untuk menyelesaikan keseluruhan permainan menjadi lebih sedikit (pada kasus ini di asumsikan waktu yang dibutuhkan untuk menyelesaikan suatu misi adalah sama untuk semua misi yang ada) .

B) Tujuan

Tujuan dibuatnya makalah ini adalah untuk menyelesaikan salah satu tugas mata kuliah IF2211 Strategi Algoritma berupa makalah serta untuk mengimplementasikan algoritma greedy pada pemilihan rute misi di permainan Assassins Creed : Revelations, sehingga meningkatkan nilai efektifitas dalam permainan.



2. TEORI DASAR

A) Prinsip Algoritma Greedy

Prinsip dari algoritma Greedy kurang lebih sesuai dengan namanya -- greedy (rakus). Dengan algoritma ini, pemakai menyelesaikan suatu permasalahan dengan mengambil langkah berikutnya yang paling menguntungkan dari pilihan-pilihan yang ada pada langkah tersebut dan mengambilnya. Pada algoritma greedy, setelah mengambil keputusan pada suatu langkah, maka tidak dapat lagi kembali ke langkah-langkah yang sebelumnya. Pada setiap langkah, algoritma memilih

pilihan paling optimum. Pada langkah tersebut, pilihan yang dipilih merupakan pilihan optimum lokal. Harapan yang ingin dicapai dengan pemilihan sebuah optimum lokal adalah sisa dari langkah-langkahnya akan mencapai sebuah optimum global pada langkah terakhir. Karena prinsip tersebutlah maka algoritma ini disebut dengan algoritma rakus.

Elemen-elemen yang terdapat pada algoritma Greedy adalah sebagai berikut:

1. Himpunan kandidat, C

Himpunan yang berisi kandidat elemen-elemen yang dapat membentuk solusi.

2. Himpunan solusi, S

Himpunan yang berisi elemen-elemen solusi.

3. Fungsi seleksi (selection function)

Fungsi seleksi adalah sebuah fungsi yang membatasi serta memilih kandidat terbaik untuk dijadikan himpunan solusi.

4. Fungsi kelayakan (feasible)

Fungsi kelayakan menentukan apakah kandidat dapat memberikan solusi.

5. Fungsi solusi

Fungsi solusi memberikan sebuah nilai true apabila solusi sudah ditemukan dan false apabila solusi belum ditemukan.

6. Fungsi obyektif

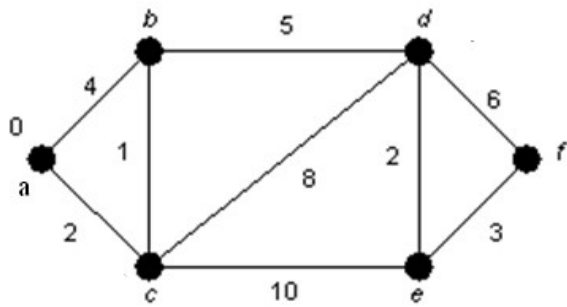
Fungsi obyektif mengindikasikan apabila solusi sudah lengkap atau belum lengkap.

Dengan elemen-elemen tersebut, cara kerja algoritma Greedy yaitu melibatkan pencarian sebuah himpunan bagian, S, dari himpunan kandidat, C. Himpunan solusi S harus memenuhi beberapa kriteria, yaitu menyatakan suatu solusi dan S dioptimisasi oleh fungsi obyektif.

Pada algoritma Greedy sebuah solusi optimum global belum tentu merupakan solusi terbaik, namun merupakan sub-optimum atau pseudo-optimum. Hal ini terjadi karena algoritma greedy tidak bekerja secara keseluruhan, karena hanya mengambil satu path, sedangkan kemungkinan lainnya tidak dipertimbangkan. Selain itu juga terdapat fungsi seleksi yang berbeda. Untuk mencapai sebuah solusi optimal harus digunakan sebuah fungsi seleksi yang tepat. Karena itu, pemakaian algoritma greedy tidak selalu berhasil memberikan solusi optimal, namun untuk permasalahan yang tidak harus memberikan jawaban terbaik mutlak, maka algoritma greedy dapat digunakan untuk memberikan sebuah solusi hampiran (approximation). Pada masalah seperti itu, penggunaan algoritma greedy lebih baik daripada menggunakan algoritma lain yang lebih rumit untuk memberikan solusi eksak.

Berikut dibahas contoh penggunaan algoritma greedy dalam persoalan shortest path dengan algoritma Dijkstra:

graf tertera pada Gambar 3

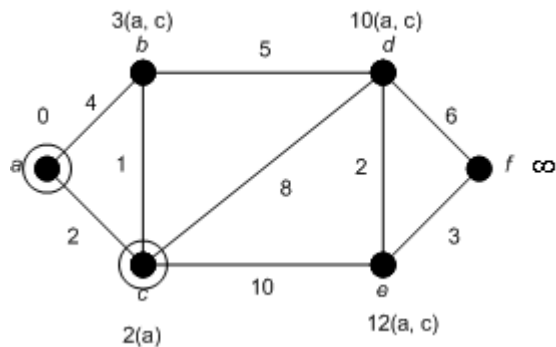


Gambar 3 : Graf G

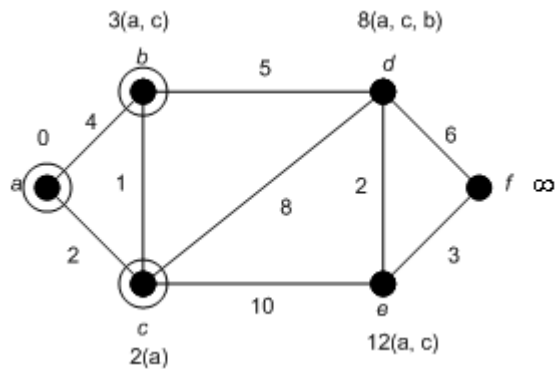
Pada gambar di atas, graf G menggambarkan rute-rute yang akan dipilih. Lintasan terpendek dari sebuah node ke node yang lain dapat dicari dengan strategi greedy sebagai berikut:

- Pada setiap langkah, ambil sisi yang berbobot minimum yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul lain yang belum terpilih.
- Lintasan dari simpul asal ke simpul yang baru haruslah merupakan lintasan yang terpendek di antara semua lintasannya ke simpul-simpul yang belum terpilih.

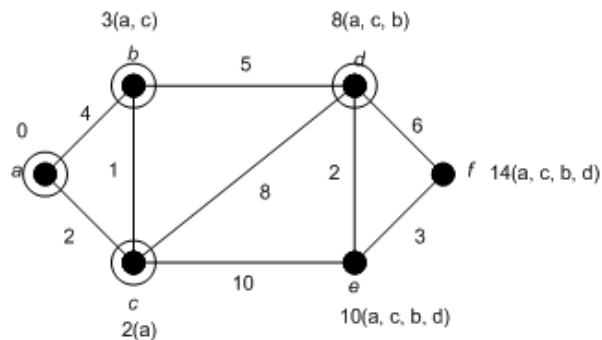
Gambar 5: mulai dari a



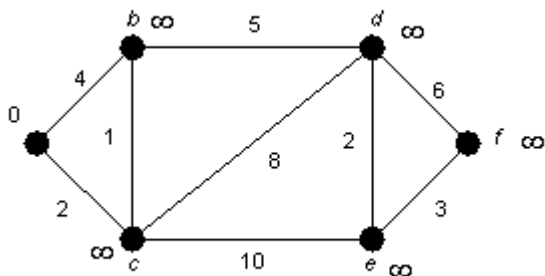
Gambar 6 : ambil node c



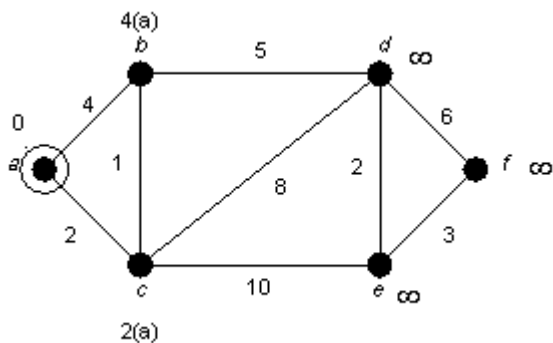
Gambar 7 : ambil node b

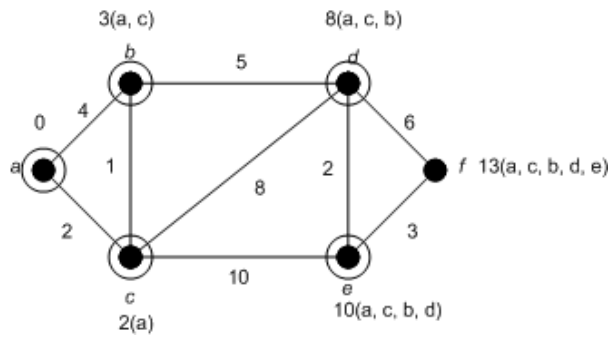


Gambar 8 : ambil node d

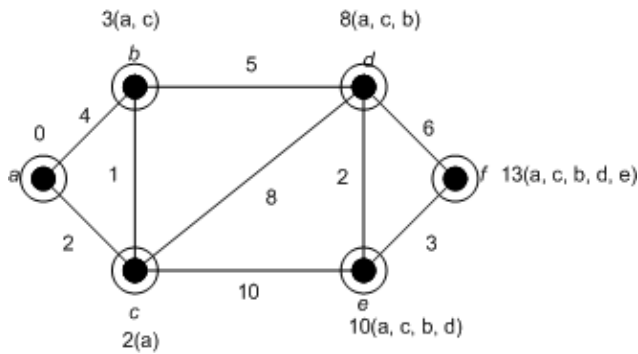


Gambar 4: inisialisasi graph





Gambar 9: ambil node e



Gambar 10 : rute terbentuk

3. ALGORITMA GREEDY PADA PEMILIHAN MISI UNTUK RUTE PERMAINAN OPTIMAL

Pada pemilihan berdasarkan rute atau panjang perjalanan, algoritma tidak memikirkan berapa biaya yang harus dikeluarkan untuk melewati rute tersebut. Algoritma greedy rute cukup sederhana dan bekerja seperti algoritma greedy untuk menyelesaikan permasalahan shortest path.

Algoritma greedy rute mencari lintasan terpendek dari simpul awal ke simpul berikutnya. Pada algoritma ini tidak digunakan algoritma djikstra, akan tetapi algoritma greedy murni berdasarkan pemilihan jarak terkecil antara calon-calon simpul. Dengan demikian akan diperoleh waktu komputasi yang sangat cepat.

Lintasan yang digunakan adalah map seperti yang tertera pada gambar Gambar 4 berikut:



Gambar 4 : Jerussalem Map

lintasan diatas beserta jarak antar titik yang diberi nomor direpresentasikan kembali dalam bentuk matriks dibawah ini :

	1	2	3	4	5	6	7	8	9	10	11
1	-	30	40	-	58	-	60	76	-	99	99
2	30	-	20	40	-	-	88	56	-	-	-
3	40	20	-	34	40	-	-	-	66	-	-
4	-	40	34	-	20	40	25	30	-	-	-
5	58	-	40	20	-	34	30	-	66	-	-
6	-	-	-	40	34	-	25	40	-	-	-
7	60	88	-	25	30	25	-	20	35	-	-
8	76	56	-	30	-	40	20	-	34	40	-
9	-	-	66	-	66	-	35	34	-	32	40
10	99	-	-	-	-	-	-	40	32	-	40
11	99	-	-	-	-	-	-	40	40	-	-

Table 1 : matriks Jerussalem map

Matriks pada tabel 1 menyatakan panjang lintasan dari sebuah simpul ke simpul yang lain. Untuk simpul yang mengarah ke dirinya sendiri dan simpul yang tidak memiliki lintasan ke sebuah simpul lain diberi dengan tanda -.

Pseudo code untuk algoritma greedy rute adalah sebagai berikut:

```

procedure greedyroute (input a: integer, b:
integer)
deklarasi:
R : array
algoritma :
R ← {} {menginisialisasi himpunan S}
while (b belum ada di R) {
hidupkan simpul a
masukkan simpul a ke himpunan solusi

```

```

do {
  if (matriks[a][next] != -
      and next belum hidup)
    {
      hidupkan simpul selanjutnya
    } else { do nothing }
} while (simpul belum habis)
Pilih simpul lintasan terpendek
Masukkan simpul ke R
a <-- simpul current
}

```

kasus 1:

Rute 1 ke 4

Pada lintasan di Map kita akan mencari lintasan terpendek dari node 1 ke node 4. Menggunakan algoritma greedy rute, berikut alur pemilihan rutenya:

- Simpul 1 hidup
- Simpul 1 menghidupkan ketujuh simpul yang terhubung dengan simpul tsb, yaitu simpul 2, 3, 5, 7, 8, 10 dan 11.
- Memasukkan simpul 1 ke dalam himpunan solusi.
- Memilih simpul dengan lintasan terpendek dari ketujuh simpul yang dihidupkan tadi, dalam kasus ini adalah simpul 2 dengan panjang lintasan 30.
- Memasukkan simpul 2 ke dalam himpunan solusi.
- Menghidupkan simpul 3, 4, 7 dan simpul 8
- Mengambil simpul 3 dan memasukkan ke dalam himpunan solusi
- Memasukkan simpul 4 ke dalam himpunan solusi
- Himpunan solusi selesai terbentuk

Pada pemecahan masalah di atas, rute yang terbentuk adalah 1 — 2 — 3 — 4 dengan panjang lintasan $(30+20+34) = 84$. Terlihat bahwa solusi memberikan hasil optimal. Pada kasus ini jika ada dua simpul yang panjangnya sama, maka algoritma akan memilih simpul yang lebih dahulu dihidupkan.

kasus 2:

Rute : 8 ke 3

Dengan menggunakan alur yang sama berikut proses pemilihan jalur untuk rute 8 sampai 11 :

- Simpul 8 hidup, masukkan ke himpunan solusi
- Menghidupkan simpul 1, 2, 4, 6, 7, 9, dan

simpul 10

- Memilih simpul 7
- Memasukkan simpul 7 ke dalam himpunan solusi
- Menghidupkan simpul 1, 2, 4, 5, 6 dan 9
- Memilih simpul 4
- Memasukkan simpul 4 ke dalam himpunan solusi
- Memasukkan simpul 3 ke dalam solusi
- selesai

Rute yang dibuat dari 8 ke 3 dengan algoritma greedy

rute adalah sebagai berikut: 8 - 7 - 4 - 3 dengan panjang lintasan total $(20 + 25 + 34) = 79$. Solusi yang diberikan adalah optimal.

kasus 3:

Rute 2 ke 6

- Menghidupkan simpul 2 dan memasukkan simpul 2 ke dalam himpunan solusi
- Menghidupkan simpul 1, 3, 4, 7 dan 8
- Memilih simpul 3
- Memasukkan simpul 3 ke dalam himpunan solusi
- Menghidupkan simpul 1, 2, 4, 5 dan 9
- Memilih simpul 4
- Memasukkan simpul 4 ke dalam himpunan solusi,
- Memasukkan simpul 6 ke dalam himpunan solusi
- Selesai

Pada pemecahan masalah di atas, rute yang terbentuk adalah 2 — 3 — 4 — 6 dengan panjang lintasan $(20 + 34 + 40) = 94$. Terlihat bahwa solusi tidak memberikan hasil optimal. Lintasan optimal adalah (2 - 4 - 6), dengan total jarak tempuh $40 + 40 = 80$

Pada kasus ini jika ada dua simpul yang panjangnya sama, maka algoritma akan memilih simpul yang lebih dahulu dihidupkan.

4. KESIMPULAN

Dari beberapa problem dan solusi yang di ambil dan di selesaikan pada makalah ini, terlihat bahwa algoritma greedy route tidak selalu memberikan hasil yang optimal, karena hanya mengandalkan perbandingan dengan jarak terpendeknya saja.

5. SARAN

Untuk kedepannya diharapkan algoritma greedy yang digunakan untuk menganalisis jalur misi dalam permainan Assassins Creed : Revelations lebih ditingkatkan lagi kemangkusannya. Sebagai contoh, dengan menerapkan algoritma djikstra pada pemilihan jalur misi tersebut.

REFERENSI

Munir, Rinaldi. (2007). Diktat Kuliah Strategi Algoritma, Departemen Teknik Informatika, Institut Teknologi Bandung.

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, 5 Mei 2015



Miftahul Mahfuzh (13513017)

