

Dynamic Programming dalam Board game Power Grid

Samuel Enrico Wijaya / 13509059
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
kesatriabajaberkarat@gmail.com

Abstract— Power grid is a one of difficult board game. In this game, players demanded to develop a lot of strategies on each of game step. Person who wins is the person that success in managing his/her strategies so his effort from the beginning of the game will yield satisfactory result. Therefore author will try to implement one topic of discussion in algorithm strategy namely dynamic programming in playing power grid.

Index Terms—Power grid, dynamic programming.

I. PENDAHULUAN

Pada pelajaran Stratei algoritma, kami para mahasiswa diebri banyak sekali metode yang berguna dalam kehidupan sehari-hari. bukan hanya dalam pemrograman, tetapi juga dalam masalah yang umum seperti mencari jalan terpendek untuk tukang pos atau pengantar paket, pencarian jalur terefektif dalam perjalanan menuju kota lain, dan sebagainya. Karenanya dalam masalah kali ini penulis akan mengangkat tema yang tidak jauh juga dari kehidupan penulis, yaitu tentang game. Topic dan game yang diangkat pada masalah kali ini adalah penggunaan dynamic programming dalam permainan power grid.

II. DASAR TEORI

2.1 DYNAMIC PROGRAMMING

Dynamic programming merupakan metode pemecahan masalah dengan cara menguraikan solusi menjadi sekumpulan langkah (*step*) atau tahapan (*stage*). Dari pengertian di atas diharapkan solusi akhir yang diperoleh dari metode ini merupakan kumpulan solusi tiap tahapan yang saling berkaitan.

Dynamic programming awalnya digunakan pada tahun 1940-an oleh Richard Bellman untuk menjelaskan pemecahan suatu masalah dimana pemilihan solusi dilakukan satu demi satu. *Dynamic programming* adalah teknik optimasi masalah matematika dan computer programming. Pada kedua konteks, *dynamic programming* merujuk kepada pemecahan suatu masalah besar dengan cara memecah masalah besar tersebut ke

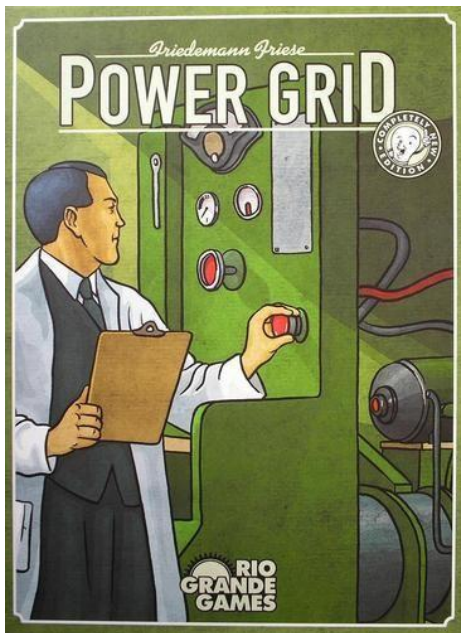
dalam sub-masalah yang lebih sederhana menggunakan rekursif.

Tahap penyelesaian masalah dengan *dynamic programming* ialah sebagai berikut :

1. Persoalan dapat dibagi menjadi beberapa tahap (*stage*), yang pada setiap tahap hanya diambil satu keputusan.
2. Masing-masing tahap terdiri dari sejumlah status (*state*) yang berhubungan dengan tahap tersebut. Secara umum, status merupakan bermacam kemungkinan masukan yang ada pada tahap tersebut.
3. Hasil dari keputusan yang diambil pada setiap tahap ditransformasikan dari status yang bersangkutan ke status berikutnya pada tahap berikutnya.
4. Ongkos (*cost*) pada suatu tahap meningkat secara teratur (*steadily*) dengan bertambahnya jumlah tahapan.
5. Ongkos pada suatu tahap bergantung pada ongkos tahap-tahap yang sudah berjalan dan ongkos pada tahap tersebut.
6. Keputusan terbaik pada suatu tahap bersifat independen terhadap keputusan yang dilakukan pada tahap sebelumnya.
7. Adanya hubungan rekursif yang mengidentifikasi keputusan terbaik untuk setiap status pada tahap k memberikan keputusan terbaik untuk setiap status pada tahap $k + 1$.
8. Prinsip optimalitas berlaku pada persoalan tersebut.

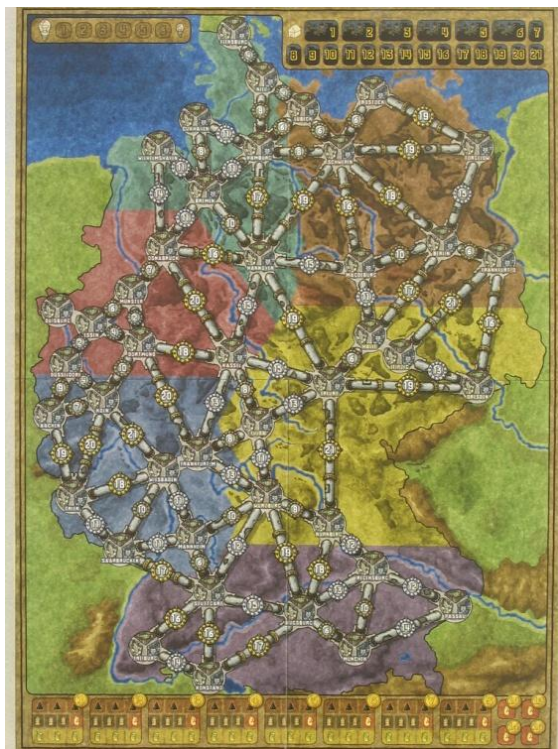
2.2 POWER GRID

Power grid adalah edisi inggris dari *board game* bahasa Jerman Funkenschlag yang dirancang oleh Friedemann Friese. Dalam permainan, setiap pemain mewakili sebuah perusahaan yang berusaha membangun pembangkit listrik untuk mengaliri kota yang akan mereka bangun juga. Pemain yang menang adalah pemain yang memiliki kota yang mampu ia pasok listrik terbanyak pada akhir permainan.

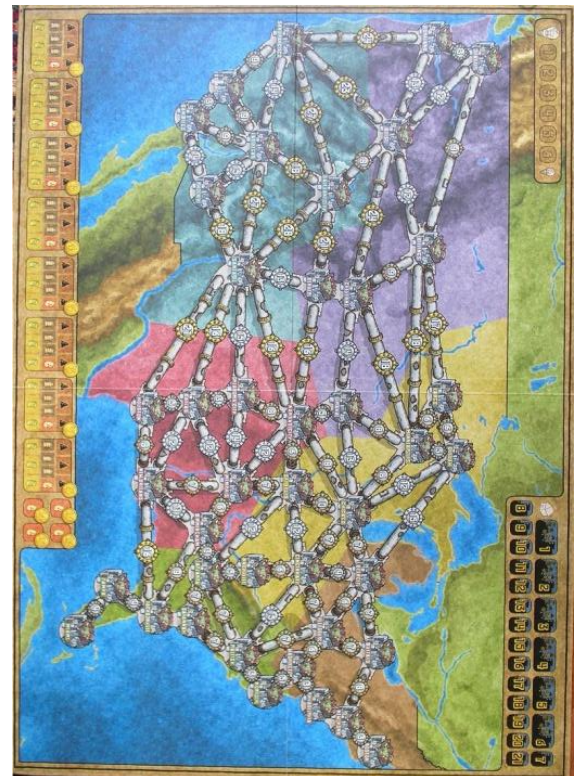


Gambar 1 : cover power grid

Papan permainan terdiri dari dua permukaan, yang masing-masing permukaannya memiliki wilayah yang berbeda. Wilayah pertama adalah Amerika Serikat dan wilayah kedua adalah Jerman. Tiap Negara terdiri dari enam wilayah dan masing-masing wilayah memiliki beberapa kota. Kota ini nantinya akan terhubung dengan kota dengan biaya yang berbeda-beda. Inilah strategi kunci untuk memenangkan permainan ini.



Gambar 2 : peta Jerman



Gambar 2: peta Amerika serikat

Satu siklus permainan ini terdiri dari lima tahap yang akan diulang-ulang yaitu :

1. Menentukan orde
 Dalam tahap ini para pemain akan menentukan giliran main pada tiap siklus dengan dua cara yaitu : pertama urutan akan ditentukan dari banyaknya rumah yang dimiliki pemain. Pemain yang memiliki rumah tebanya akan mendapat giliran pertama. Kedua, bila ada pemain yang memiliki jumlah rumah yang sama maka urutan akan ditentukan dengan mahalnnya pembangkit listrik yang merekamiliki. Pemain dengan pembangkit listrik yang lebih mahal akan berhak main terlebih dahulu.
2. Lelang pembangkit listrik
 Pembangkit listrik adalah sarana yang kita gunakan untuk mengalirkan energi ke dalam kota yang nantiya akan kita bangun. Karenanya kita juga harus pintar dalam memilih pembangkit listrik yang tepat. Lelang akan dimulai dari pemain pertama, ada 4 pembangkit listrik yang dapat pemain pilih untuk dilelang tiap kali. Setelah ada pembangkit listrik yang tebeli, para pemain mengambil 1 lembar pembangkit listrik lagi dari *deck* dan mengurutkannya dari harga termurah sampai harga termahal. Saat lelang para pemain boleh menawar ataupun melwati giliran lelanghnya, tetapi jika pemain memilih melewati

giliran lelangnya, maka ia tidak berhak melesan lagi pada tahap lelang siklus tersebut.

3. Membeli sumber daya

Sumber daya adalah sarana yang dibutuhkan pemain untuk dapat menyalakan pembangkit listrik. Ada empat sumber daya yang ditawarkan pada permainan ini, yaitu batu bara, minyak, sampah, dan nuklir. Masing-masing pembangkit listrik memiliki bahan yang berbeda-beda yang dibutuhkan untuk membangkitkannya. Ada pembangkit listrik yang membutuhkan tiga buah sampah untuk membangkitkan listrik satu rumah, ada pembangkit listrik yang hanya memerlukan satu buah nuklir untuk membangkitkan listrik lima buah rumah.

4. Membangun rumah

Setelah semua itu barulah pemain membangun rumah. Para pemain dari urutan terakhir ke pertama dapat membeli sebuah rumah dan meletakkannya di sebuah kota pada peta. Tiap rumah berharga 10, rumah pertama yang pemain letakkan akan menentukan rute pemain itu selanjutnya, karena tiap kota memiliki penghubung dengan cost nya masing-masing, kota kedua yang pemain letakkan akan berharga 10+ cost dari kota terdekat ke kota dimana rumah tersebut ingin dibangun.

5. Birokrasi

Para pemain akan menerima sejumlah uang dari bank sesuai dengan jumlah rumah yang dapat ia aliri listrik pada akhir siklus,

Setelah 5 tahap dalam 1 siklus itu selesai dilanglah siklus itu uteras menerus sampai kartu pembangkit habis atau ada pemain yang sudah memiliki limabelas rumah. Pemain yang menang adalah pemain yang dapat menyaakan rumah terbanyak saat permainan selesai (walaupun pemain memiliki paling banyak rumah, tetepi jika tidak dapat menyalakan rumahnya maka tidak akan dihitung sebagai poing kemenangan.)

III. PENYELESAIAN MASALAH

3.1 ANALISIS MASALAH

Seperti yang dapat kita lihat, permainan ini bukanlah permainan yang mudah. Banyak strategi yang harus diterapkan, bahkan dari awal permainan. Akan tetapi yang akan penulis bahas dalam makalah kali ini hanyalah tahap pembangunan rumah. Sebuah Negara dalam permainan ini beserta kota-kotanya dapat dijadikan matriks hubungan jarak. Dari matriks itu dapat kita ambil kota terbaik dan jalur teroptimal dalam membangun rumah. Walau cara itu tidak 100% berhasil karena ada pemain

lain dalam permainan itu yang dapat menghalangi jalur yang hendak kita lewati.

3.2 SOLUSI

Algoritma yang akan diterapkan adalah seperti berikut : Awalnya akan dipilih satu kota, setelah itu akan dilihat jalur secara acak yang memiliki cost tersedikit sampai rumah tersebut mencapai lima belas buah.

Untuk mendapatkan hal itu, terlebih dahulu kita akan membuat matriks jarak antara satu kota dengan kota lainnya. Jarak yang saya maksud adalah biaya yang harus dibayarkan saat seorang pemain ingin mengembangkan wilayahnya (membangun rumahnya) dari kota satu ke kota lainnya

	1	2	3	4	5	6	7
1				6			
2			19	6			
3		19		19		15	
4	6	6	19		16	18	
5				16		10	
6			15	18	10		6
7						6	
8							
9	4						
10	6			8			
11				19	15		
12					11	17	
13							21
14							16

8	9	10	11	12	13	14
	4	6				
		8	19			
			15	11		
				17		
					21	16
	4					
4		8				
	8		17			
		17				
						13

Gambar 4 : matriks jarak area coklat dan sekitarnya

Keterangan :

1. Lubick
2. Rostock
3. Forgelow
4. Schwerin
5. Macdeburc
6. Berlin
7. Frankfurter
8. Finsburg
9. Kill
10. Hamburg
11. Mannover
12. Mavie
13. Leipzic
14. Cresoen

Matriks jarak di atas menggambarkan hubungan jarak (harga yang harus dibayar saat ingin membangun rumah melewati jalur tersebut) antar kota. Angka-angka dalam index matriks merepresentasikan sebuah kota yang bisa dilihat dalam daftar di atas. Misal perpotongan kolom 4 dengan baris 1 tertulis angka 6, itu berarti harga yang harus dibayar jika kita ingin mengekspansi wilayah kita dari kota 1 ke kota 4 adalah sebesar 6.

Karena terlalu banyak kota dan terlalu besar bila matriks beban nya dimasukkan ke sini, maka saya akan memberi contoh satu matriks daerah pada jerman yaitu daerah yang berada di daerah kanan atas pada peta jerman (yang berwarna coklat dan sekitarnya).

```

while(z < 15) do
  while(r<15) do
    i interval [1..7]{
      if((m[k][i]<a)and(!ada(i,j)))
        a=m[k][i]
        b = b+a
        k = i
        j[c]=i
      c=c+1
      r=r+1
    }
    z=z+1
  }

```

Gambar 5 : contoh algoritma untuk mencari jalur teroptimal

Dengan asumsi bahwa kota yang telah kita lewati tidak

dapat kita lewati kembali, maka algoritma di atas akan menghasilkan sebuah jalur yang optimal.

3.3 ANALISIS ALGORITMA

Integer a adalah jarak yang akan dibandingkan dengan jarak selanjutnya, integer b adalah integer yang menyimpan total harga yang harus dibayarkan, integer c adalah counter untuk mengisi array kota (agar terurut), integer r adalah jumlah kota sekarang ini, array of integer j adalah array yang menampung jalur mana saja yang telah dilewati. Prosedur ada digunakan untuk mengecek apakah kota yang akan dituju sudah pernah dilewati atau belum. Integer z untuk menandai apakah semua kota pernah dijadikan kota pertama atau belum.

Awalnya kita akan memilih kota 1 sebagai kota yang kita dirikan rumah pertama kali. Setelah itu algoritma akan mulai mencari kemana kota selanjutnya yang dapat dibangun dengan biaya terkecil, dengan cara mengiterasi hubungan dari kota 1 ke kota lainnya dan mengambil jarak terpendek. Jika jarak itu telah didapat maka jarak itu akan disimpan ke dalam sebuah integer untuk dibandingkan lagi dengan jarak berikutnya. Jarak itu juga akan disimpan ke dalam sebuah variable yang menampung total jarak (agar kita dapat mengetahui berapa biaya yang harus kita simpan untuk dapat membangun semua rumah tersebut). Jika biaya telah kita dapat, maka kota dimana biaya itu berlaku akan disimpan ke dalam k yang kemudian akan diiterasi lagi seperti di atas sampai jalur tersebut melewati limabelas kota. Kota tersebut juga akan kita simpan ke dalam sebuah array yang akan menjadi hasil yang kita cari yaitu jalur terpendek. Semua langkah di atas kemudian akan diulang dengan menambah k sejumlah 1 (memilih kota awal sehingga semua kota pernah menjadi kota awal) sampai z mencapai 15.

Algoritma itu dapat diterapkan di contoh tabel tentunya dengan mengganti jumlah kota menjadi kurang dari atau sama dengan lima belas (karena kota yang ada di dalam contoh hanya empat belas buah).

IV. KESIMPULAN

Pemakaian topik bahasan strategi algoritma ternyata dapat kita terapkan dalam kehidupan sehari-hari. Dar sesuatu yang telah lama ada (pengantaran paket) sampai ke sesuatu yang baru, misal *board game* ini.

Power grid termasuk permainan yang rumit, oleh karenanya banyak algoritma yang dapat kita terapkan di dalamnya. Pemrograman dinamis adalah salah satunya, dengan menggunakannya seperti pada makalah ini, kita dapat mengakali permainan ini sehingga dapat menghasilkan hasil yang optimal. Akan tetapi algoritma tersebut tidak dapat menjamin kemenangan pemain. Banyak hal-hal lain yang harus dipertimbangkan juga,

seperti saan pelepasan, pembelian bahan bakar, maupun giliran main sang pemain. Algoritma ini saya buat dengan mengesampingkan itu semua. Dengan menganggap pemain dapat manage hal-hal lain tersebut, pemain dapat membangun rumahnya dengan biaya yang minimum untuk mencapai hasil yang maksimum.

V. SARAN

Penulis memberi saran pemain agar memikirkan langkahnya dengan hati-hati. Percuma saja pemain telah mendapat jalur yang maksimal tetapi digagalkan karena kecerobohan pemain sendiri,masl pemain lupa menyisakan sebagian uangnya untuk dapat membangun rumah di akhir permainan, lalu beberapa kota yang berada di jalur optimal telah diambil pemain lain, maka pemain mau tidak mau harus memikirkan jalur optimal lain. Hal itu tentu saja amat disayangkan karena menghambat permainan dan menyia-yiakan jalur yang telah anda cari sebelumnya.

VI. REFERENCES

- [1] Bahan kuliah IF2251 Strategi Algoritma, Program Dinamis, oleh Rinaldi Munir.
- [2] Dynamic programming - Wikipedia, the free encyclopedia diakses pada 7 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



Samuel Enico Wijaya / 13509059