

PENGGUNAAN ALGORITMA *GREEDY* DALAM PERMAINAN KARTU *BLACK JACK*

Dwitiyo Abhirama - 13505013

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl Ganesha 10, Bandung
e-mail: if15013@students.if.itb.ac.id

ABSTRAK

Makalah ini menjelaskan penggunaan algoritma *greedy* pada permainan kartu *black jack*. Permainan kartu *black jack* merupakan permainan yang dimainkan oleh lebih dari satu orang pemain. Tujuan permainan *black jack* adalah mengumpulkan jumlah nilai terdekat dengan 21 tetapi tidak lebih dari 21. Algoritma *greedy* digunakan untuk menentukan langkah yang akan diambil oleh seorang pemain.

Algoritma *greedy* merupakan algoritma yang akan mengambil solusi optimal pada setiap langkah. Dengan menggunakan algoritma *greedy* pada permainan kartu *black jack* seorang pemain akan menyeleksi kemungkinan untuk memutuskan akan menambah kartu atau tidak pada setiap langkah. Diharapkan dengan mengambil solusi optimal lokal akan mendapatkan hasil optimal pada permainan (keseluruhan langkah).

Kata Kunci: *greedy*, optimal, solusi, seleksi, kendala, kemungkinan, kartu, *black jack*.

1. PENDAHULUAN

Permainan kartu merupakan permainan yang cukup terkenal. Permainan kartu mempunyai banyak variasi permainan. Salah satu di antara berbagai variasi permainan kartu tersebut adalah permainan kartu *black jack* yang cukup banyak dimainkan.

Permainan *black jack* merupakan permainan kartu yang dimainkan oleh lebih dari seorang

pemain. Permainan *black jack* umumnya dimainkan oleh dua pemain saja. Algoritma *greedy* akan diterapkan pada permainan kartu *black jack* ini untuk mengoptimisasi langkah-langkah yang akan diambil oleh seorang pemain untuk memperoleh hasil optimal.

2. PERMAINAN KARTU *BLACK JACK*

Permainan kartu *black jack* dimulai dengan membagikan dua buah kartu secara acak (sebelum permainan dimulai, kartu yang akan digunakan terlebih dahulu dikocok) kepada masing-masing pemain. Satu kartu dari dua kartu yang dibagikan pada awal permainan kepada masing-masing pemain dibuka (dibalik) agar dapat terlihat oleh semua pemain.

Setelah semua pemain memperoleh pembagian kartu sebagaimana mestinya, permainan kartu *black jack* siap untuk dimulai. Tujuan dari permainan kartu *black jack* adalah mengumpulkan kartu sedemikian sehingga jumlah nilai kartu yang dimiliki pemain memiliki selisih paling kecil dengan angka 21 (dua puluh satu), tetapi jumlah nilai kartunya tidak boleh lebih dari angka 21. Seorang pemain dinyatakan sebagai pemenang jika jumlah nilai kartunya paling tinggi (tetapi tentu saja jumlah nilai kartunya tidak boleh lebih dari angka 21). Apabila jumlah nilai kartu setiap pemain lebih dari angka 21 maka tidak ada pemain yang dinyatakan sebagai pemenang.

Kartu-kartu pada permainan kartu *black jack* mempunyai nilai sebagaimana yang tercantum pada tabel 1 berikut ini.

Tabel 1. Tabel Nilai Kartu pada Permainan Kartu *Black Jack*

Kartu	Gambar / Jenis	Nilai
-------	----------------	-------

2	Semua gambar / jenis	2
3	Semua gambar / jenis	3
4	Semua gambar / jenis	4
5	Semua gambar / jenis	5
6	Semua gambar / jenis	6
7	Semua gambar / jenis	7
8	Semua gambar / jenis	8
9	Semua gambar / jenis	9
10	Semua gambar / jenis	10
Jack (J)	Semua gambar / jenis	10
Queen (Q)	Semua gambar / jenis	10
King (K)	Semua gambar / jenis	10
As (A)	Semua gambar / jenis	11

Dalam permainan kartu *black jack* semua gambar atau jenis kartu bernilai sama. Penjumlahan nilai kartu dapat dilakukan terhadap gambar atau jenis kartu yang berbeda. Hal ini berarti seorang pemain tidak harus mengumpulkan kartu dengan gambar atau jenis yang sama.

Setiap pemain akan mendapatkan kesempatan untuk menambah kartu (mengambil kartu dari tumpukan kartu yang tersisa) secara bergiliran. Pada kesempatan tersebut seorang pemain berhak untuk memutuskan apakah ingin menambah kartu atau tidak. Jika seorang pemain memutuskan untuk menambah kartu, maka pemain tersebut akan mengambil kartu dari tumpukan kartu yang tersisa. Seorang pemain hanya dapat menambah satu kartu pada setiap kesempatan. Kartu yang diambil pada kesempatan tersebut tetap tertutup (tidak dibalik sehingga hanya dapat dilihat oleh pemain tersebut). Jika seorang pemain memutuskan untuk tidak menambah kartu pada suatu kesempatan, maka pemain tersebut tidak dapat memutuskan untuk menambah kartu lagi pada kesempatan berikutnya.

Permainan dapat berakhir jika semua pemain telah memutuskan untuk tidak menambah kartu lagi. Kemudian semua kartu yang dimiliki oleh masing-masing pemain dibuka (dibalik sehingga dapat dilihat oleh semua pemain). Kemudian jumlah nilai kartu yang dimiliki oleh masing-masing pemain dihitung. Pemenang permainan dapat ditentukan.

3. PROSES OPTIMALISASI PERMAINAN

3.1. ALGORITMA *GREEDY*

Secara harafiah *greedy* mempunyai arti rakus atau tamak. Dalam hal ini *greedy* berarti berusaha mendapatkan hasil sebanyak-banyaknya (optimum).

Algoritma *greedy* merupakan algoritma yang cukup sederhana dan populer dalam persoalan optimasi. Persoalan optimasi merupakan masalah untuk menemukan solusi yang optimum dari berbagai kemungkinan yang ada.

Persoalan optimasi mempunyai sekumpulan kendala dan fungsi optimasi. Penyelesaian persoalan optimasi adalah dengan menemukan solusi optimum yaitu solusi yang memenuhi semua kendala (disebut juga solusi layak) yang mengoptimalkan fungsi optimasi.

Algoritma *greedy* membentuk solusi langkah per langkah. Terdapat banyak pilihan atau kemungkinan yang perlu dipertimbangkan. Oleh karena itu, pada setiap langkah atau kesempatan perlu dibuat keputusan yang terbaik (optimum lokal) dalam menentukan pilihan. Keputusan yang telah diambil pada suatu langkah tidak dapat diubah pada langkah-langkah berikutnya.

Pendekatan yang digunakan di dalam algoritma *greedy* adalah dengan membuat keputusan yang seakan-akan akan memberikan hasil yang terbaik. Hal ini dilakukan dengan memilih keputusan yang terbaik pada setiap kesempatan dengan harapan akan mendapatkan solusi optimum secara keseluruhan (optimum global).

Untuk mempermudah penyelesaian persoalan optimasi dengan algoritma *greedy*, disusun elemen-elemen sebagai berikut.

1. Himpunan Kandidat, C.
Himpunan ini merupakan elemen-elemen pembentuk solusi. Hal ini berarti solusi diambil dari anggota himpunan kandidat.
2. Himpunan Solusi, S.
Himpunan solusi berisi kandidat-kandidat yang telah terpilih sebagai solusi persoalan. Jadi, himpunan solusi merupakan himpunan bagian dari himpunan kandidat.
3. Fungsi (Predikat) Seleksi
Fungsi seleksi merupakan fungsi yang memutuskan pilihan pada setiap langkah. Fungsi seleksi akan mempengaruhi pengambilan kandidat pada setiap langkah. Kandidat yang telah diambil tidak pernah diubah pada langkah berikutnya.
4. Fungsi (Predikat) Kelayakan.
Fungsi kelayakan akan memeriksa apakah pengambilan suatu kandidat bersama-sama dengan himpunan solusi yang telah terbentuk tidak melanggar kendala. Ketidaklayakan dapat mengakibatkan pembuangan kandidat (kandidat tidak dipertimbangkan lagi).
5. Fungsi Obyektif.
Fungsi obyektif merupakan fungsi yang akan memoptimalkan (memaksimalkan atau meminimumkan) nilai solusi.

3.2. OPTIMALISASI PADA PERMAINAN KARTU *BLACK JACK*

Penggunaan algoritma *greedy* pada permainan kartu *black jack* oleh seorang pemain terutama adalah untuk membuat keputusan pada setiap langkah hingga permainan berakhir. Hal ini berarti akan memutuskan pertimbangan kesempatan penambahan kartu pada setiap langkah

hingga suatu saat memutuskan untuk tidak menambah kartu lagi.

Dalam permainan ini, himpunan kartu yang belum terambil oleh salah satu di antara semua pemain merupakan himpunan kandidat. Setiap pemain dapat mengambil sebuah kartu dari himpunan kartu kandidat tersebut pada setiap kesempatan yang dimilikinya untuk digabungkan dengan kartu yang telah dimiliki sebelumnya. Semua kartu yang dimiliki oleh seorang pemain hingga suatu saat merupakan himpunan solusi sampai sejauh saat itu. Himpunan solusi akhir merupakan semua kartu yang dimiliki oleh pemain tersebut ketika permainan berakhir.

Jumlah nilai kartu yang dipunyai oleh seorang pemain merupakan sesuatu yang akan dioptimasi oleh algoritma *greedy* dalam permainan ini. Fungsi obyektif dalam permainan ini berkaitan dengan memaksimalkan jumlah nilai kartu yang dipunyai oleh pemain tersebut.

Kendala atau batasan dalam permainan ini adalah jumlah nilai kartu yang dimiliki oleh seorang pemain tidak boleh lebih dari angka 21 (dua puluh satu). Selain itu, setiap pemain hanya dapat mengambil satu kartu dalam setiap kesempatan (tentu saja bila pemain tersebut memutuskan untuk menambah kartu) hingga pemain tersebut memutuskan untuk tidak menambah kartu lagi.

Algoritma *greedy* akan menyeleksi pilihan pertimbangan pengambilan kartu kandidat berdasarkan peluang atau kemungkinan jumlah kartu pada saat itu. Fungsi seleksi akan memutuskan apakah pengambilan kartu merupakan keputusan yang terbaik pada kesempatan tersebut. Fungsi seleksi akan menghitung kemungkinan keberadaan kartu pada himpunan kartu kandidat yang dapat menaikkan jumlah nilai kartu yang dimiliki tetapi tidak sampai melebihi angka 21. Bila kemungkinan keberadaan kartu tersebut lebih besar daripada kemungkinan keberadaan kartu yang menyebabkan jumlah nilai kartu yang dimiliki melebihi angka 21, maka fungsi (predikat) seleksi akan

memutuskan untuk mengambil sebuah kartu dari himpunan kandidat yang tersisa.

Algoritma *greedy* akan mempertimbangkan kelayakan kartu (kandidat) yang diambil bersama-sama dengan himpunan solusi yang telah terbentuk (kartu yang telah dimiliki) apakah melebihi angka 21 atau tidak (melanggar batasan atau tidak). Jika melanggar batasan, maka himpunan kandidat yang tersisa akan tidak dipertimbangkan lagi. Hal ini berarti tidak akan terjadi penambahan kartu lagi.

Penggunaan algoritma di atas terhadap seorang pemain hanya akan memperhatikan kartu yang dimiliki oleh pemain tersebut saja dan kartu yang masih mungkin diambil oleh pemain tersebut serta tidak memperhatikan kartu yang dimiliki pemain lain.

Dengan menggunakan strategi di atas pada setiap langkah atau kesempatan, diharapkan pada akhir permainan diperoleh jumlah nilai kartu yang optimum sehingga dapat menjadi pemenang dalam permainan ini.

Berikut ini rancangan skema algoritma di atas.

```
Kamus Global
ArrKartu [1..52] of Boolean;
{Himpunan kartu, ArrKartu[i]
(1<=i<=52) true bila kartu yang
bersesuaian masih mungkin diambil
(belum terbuka)}
MyKartu [] of integer; {Jumlah nilai
kartu yang dimiliki}
```

```
PROCEDURE greedy()
```

```
Kamus
stop : boolean;
nilai : integer;
```

```
Algoritma
stop = false;
AwalMain(); {Menginisialisasi
permainan, membagi kartu secara acak}
while (not (stop)) do
  if (isAmbil()) then
    {predikat (fungsi) seleksi,
true jika kemungkinan
keberadaan kartu yang
menyebabkan jumlah nilai
kartu yang akan dimiliki
melebihi angka 21 lebih kecil
daripada yang tidak, jika
true maka ambil sebuah kartu}
    Ambil();
  endif
```

```
  if (not (isLayak())) then
    {predikat (fungsi) kelayakan,
false jika kartu yang diambil
bersama-sama dengan kartu
yang telah dimiliki jumlah
nilainya melebihi angka 21,
jika tidak layak maka
himpunan kandidat kartu yang
tersisa akan diabaikan karena
memutuskan untuk tidak akan
pernah mengambil kartu lagi}
    stop = true;
  endif
endwhile
nilai = solusi();
output(nilai); {mengeluarkan nilai
solusi}
endprocedure
```

3.3. ANALISIS HASIL

Seorang pemain akan melakukan optimasi pada setiap langkah dengan menggunakan prosedur *greedy*. Optimasi yang dilakukan adalah hanya akan menambah kartu pada setiap langkah apabila kemungkinan keberadaan kartu dalam kandidat sehingga tidak melanggar batasan cukup besar (lebih besar daripada yang melanggar batasan).

Penggunaan algoritma *greedy* pada permainan ini cukup sederhana, mudah dan cepat karena hanya mempertimbangkan kemungkinan jumlah nilai kartu yang ada pada diri sendiri dan kemungkinan kartu yang belum terbuka. Pengambilan keputusan dapat dilakukan dengan cukup cepat dan baik.

Berdasarkan kriteria optimasi yang dilakukan hasilnya cukup memuaskan. Hal ini dikarenakan penambahan kartu dilakukan jika kemungkinan masih cukup besar sehingga hasil akhir dari serangkaian permainan cukup memuaskan (optimum).

Meskipun demikian, algoritma *greedy* tidak dapat menjamin akan selalu memberikan hasil yang terbaik dalam setiap permainan. Hal ini karena algoritma *greedy* pada suatu langkah dapat menyisahkan dan tidak mempertimbangkan kartu yang dimiliki oleh pemain lain.

4. KESIMPULAN

Algoritma *greedy* merupakan algoritma yang cukup sederhana dan baik dalam menyelesaikan persoalan optimasi. Inti penggunaan algoritma *greedy* pada permainan kartu *black jack* adalah pembuatan keputusan pengambilan kartu berdasarkan kemungkinan kartu yang blum terbuka pada setiap langkah. Pengoptimalisasian setiap langkah diharapkan akan memberikan hasil akhir yang memuaskan (optimum). Namun, perlu disadari bahwa algoritma *greedy* tidak pasti akan selalu memberikan hasil yang terbaik.

REFERENSI

- [1] Munir, Rinaldi. "Strategi Algoritmik", ITB, 2007.
- [2] <http://en.wikipedia.org> Tanggal akses: 20 Mei 2007, pukul: 08.00.

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.