

# PENERAPAN ALGORITMA GREEDY DALAM OPTIMASI JUMLAH NILAI KARTU PADA PERMAINAN ANGELINA

Lea Angelina - 13506117

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung  
Jl. Ganesha no 10, Bandung  
e-mail: [if16117@students.itb.ac.id](mailto:if16117@students.itb.ac.id)

## ABSTRAK

Pada makalah ini, akan dibahas mengenai penerapan algoritma greedy pada permainan angelina.

Permainan angelina adalah permainan sederhana ciptaan penulis yang menggunakan dek kartu remi.

Objektif dalam permainan angelina ini adalah mendapatkan jumlah nilai kartu terdekat dengan  $V$ , tetapi tidak melebihi  $V$ .  $V$  adalah nilai yang sebelumnya ditetapkan melalui penawaran.

Algoritma greedy adalah suatu metode yang dipakai untuk masalah optimasi; ia mencari solusi optimal lokal dan mengharapkan kumpulan solusi-solusi optimal lokal akan mengarah kepada solusi optimal global.

**Kata kunci:** angelina, greedy, kartu, solusi, optimal, pilihan.

## I. PENDAHULUAN

Angelina adalah permainan kartu ciptaan penulis yang namanya diambil dari nama penulis.

Angelina dimainkan oleh lebih dari satu pemain. Aturan permainannya sangat sederhana, yaitu pemain boleh memilih untuk mengambil kartu dari tumpukan kartu tertutup atau tidak mengambil kartu. Bila memilih untuk tidak mengambil kartu, pemain tidak boleh mengambil kartu lagi pada langkah berikutnya. Pemenangnya adalah pemain yang mendapatkan total nilai kartu yang terdekat, namun tidak lebih dari  $V$ .

Penerapan algoritma greedy akan mencari optimal lokal pada setiap langkah dalam permainan ini, bertujuan untuk

mendapatkan optimal global, yaitu nilai kartu yang terdekat namun tidak lebih dari  $V$ .

## II. PERMAINAN ANGELINA

Permainan angelina sangat sederhana. Sesi pertama adalah sesi penawaran. Semua kartu dibagi kepada semua pemain sampai tidak ada lagi kartu yang tersisa, kemudian masing-masing pemain memilih salah satu kartu untuk dijadikan tawaran. Nilai total kartu yang dijadikan tawaran dijumlahkan, dan itulah nilai  $V$ .

Setelah itu, kartu dikembalikan dan diacak lagi, lalu ditumpuk dalam dek. Sesi kedua-yaitu sesi permainan-dimulai. Masing-masing pemain memiliki giliran, pada tiap giliran itu boleh memilih untuk mengambil kartu atau tidak. Pilihan tidak mengambil kartu akan mengakhiri giliran pemain tersebut. Setelah semua pemain selesai mengambil kartu, jumlah nilai kartu masing-masing pemain akan dibandingkan. Pemenangnya adalah pemain yang mendapatkan nilai kartu tertinggi (yang tidak lebih dari  $V$ ). Bila lebih dari  $V$ , akan dinyatakan burst. Jika ada lebih dari 1 pemain yang nilai kartunya sama, yang memiliki kartu tertinggi akan lebih tinggi peringkatnya. Nilai akhir kartu pemain ditentukan oleh rumus

$$\begin{aligned} \text{total} &= \sum \text{nilai final kartu}(i) \\ i &= 1 \text{ s/d } \text{kartu total pemain} \\ \text{nilai final kartu}(i) &= \text{nilai kartu}(i) + \text{nilai gambar} \\ \text{nilai gambar} &= \text{wajik:0; keriting:0.25; hati:0.5; sekop:0.75} \end{aligned}$$

Kartu angka memiliki nilai kartu yang sama dengan angkanya; King, queen, dan jack memiliki nilai kartu 10, sedangkan as memiliki nilai 1 atau 11, sesuai pilihan pemain.

### III. OPTIMALISASI

#### 3.1 Algoritma Greedy

Algoritma greedy membentuk solusi langkah demi langkah (step by step). Terdapat banyak pilihan yang perlu dieksplorasi pada setiap langkah solusi. Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Keputusan yang diambil pada suatu langkah tidak dapat diubah lagi pada langkah selanjutnya. Sebagai contoh, jika kita menggunakan algoritma greedy untuk menempatkan komponen di atas papan sirkuit, sekali sebuah komponen telah ditetapkan posisinya, komponen tersebut tidak dapat dipisahkan lagi. Pendekatan yang digunakan di dalam algoritma greedy adalah membuat pilihan yang “tampaknya” memberikan perolehan terbaik, yaitu dengan membuat pilihan optimum lokal pada setiap langkah dengan harapan bahwa sisanya mengarah ke solusi optimum global.

Dari pendekatan yang disebutkan di atas, kita dapat mendefinisikan algoritma greedy sebagai berikut:

Algoritma greedy adalah algoritma yang memecahkan masalah langkah per langkah:

1. mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan (prinsip “take what you can get now!”)
2. berharap bahwa dengan memilih optimum lokal pada setiap langkah akan berakhir dengan 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 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 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 mengoptimalkan (memaksimumkan atau meminimumkan) nilai solusi.

#### 3.2 Penerapan Greedy Terhadap Optimalisasi Nilai Kartu pada Angelina

Algoritma greedy pada permainan angelina berfungsi untuk membuat keputusan pada setiap langkah yang berarti membuat keputusan terhadap pertimbangan untuk menambahkan kartu pada setiap langkah hingga memutuskan untuk tidak menambah kartu lagi.

Asosiasinya:

1. Himpunan Kandidat: Himpunan kartu yang masih ada di tumpukan (belum terambil).
2. Himpunan Solusi: Kartu yang ada di tangan pemain saat tidak ada lagi pengambilan kartu.
3. Fungsi Seleksi: Mengembalikan keputusan, apakah pemain akan mengambil kartu lagi atau tidak.
4. Fungsi Kelayakan: Mengembalikan apakah solusi layak (kurang dari sama dengan  $V$ ) atau tidak.
5. Fungsi Obyektif: Menghitung kemungkinan keberadaan kartu yang ‘tidak aman’ (yang jika ditambahkan pada kartu di tangan pemain, hasilnya melebihi  $V$ ).

Algoritma greedy akan membuat keputusan dari pilihan pertimbangan pengambilan kartu kandidat berdasarkan jumlah kartu di tangan pada saat itu. Fungsi seleksi akan memilih keputusan terbaik, apakah akan mengambil kartu atau tidak. Fungsi seleksi akan membuat keputusan berdasarkan fungsi obyektif. Jika kemungkinan calon kartu yang akan diambil ‘aman’ lebih besar, maka kartu akan diambil dan ditambahkan ke tangan pemain. Jika kemungkinan calon kartu itu ‘tidak aman’ lebih besar, maka pengambilan kartu tidak jadi dilakukan.

Jika ternyata himpunan solusi melanggar batasan, maka himpunan kandidat yang tersisa akan tidak dipertimbangkan lagi atau 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 terambil 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

```
KartuDek [1..52] of Boolean;  
{Himpunan kartu, KartuDek[i] (1<=i<=52) true  
bila kartu masih ada di dek (belum diambil)}  
TotalNilaiKartu  
[] of integer;  
{Jumlah nilai kartu yang ada di tangan}
```

```
PROCEDURE greedy()
```

#### Kamus

```
stop : boolean;  
nilaifinalkartu : real;
```

#### Algoritma

```
stop = false;  
Inisialisasi();  
Penawaran(); {Penawaran menginisialisasi nilai  
untuk V}  
while (not (stop)) do  
  if (isAmbil()) and (isLayak()) then  
    {isAmbil() adalah predikat seleksi,  
    bernilai true jika fungsi objektif  
    hitungKemungkinan() bernilai kurang dari  
    0.5 dari skala 0 sampai 1.  
    hitungKemungkinan() adalah fungsi dalam  
    predikat isAmbil() yang menghitung  
    kemungkinan keberadaan kartu yang  
    menyebabkan jumlah nilai kartu yang  
    akan dimiliki melebihi V; isLayak()  
    bernilai true jika kartu di tangan pemain  
    kurang dari V}  
    Ambil();  
  endif
```

### 3.3 Analisis

Prosedur greedy akan mengoptimasi tiap langkah dengan cara menambah kartu pada setiap langkah apabila kemungkinan adanya kartu dalam kandidat yang bila ditambahkan pada kartu pada pemain hasilnya tidak melanggar batasan lebih besar daripada kartu yang apabila ditambahkan, nilainya akan lebih dari V.

Penggunaan algoritma greedy pada permainan ini hanya mempertimbangkan kemungkinan jumlah nilai kartu pada pemain dan kemungkinan kartu yang belum terbuka. Hasilnya cukup memuaskan, karena 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 menyesatkan dan tidak mempertimbangkan kartu yang dimiliki oleh pemain lain.

## IV. KESIMPULAN

Algoritma greedy adalah algoritma yang sederhana, straight to the point, dan biasanya memberikan hasil yang cukup baik untuk menyelesaikan masalah optimasi. Dalam penerapannya terhadap permainan angeline, greedy membuat keputusan berdasarkan kartu yang belum terbuka pada setiap langkah.

Rancangan algoritma yang dibuat penulis, ketika diharapkan hasilnya cukup baik. Kekurangannya hanya satu, yaitu algoritma greedy tidak menjamin tercapainya optimum global.

## REFERENSI

- [1] Munir, Rinaldi. Strategi Algoritmik . ITB, 2007.
- [2] <http://en.wikipedia.org> Tanggal akses: 19 Mei 2008, pukul: 19.00.