Penggunaaan Algoritma *Greedy* Dalam Aplikasi *Vending Machine*

Aryo Nugroho/13505063

Email : if15063@students.if.itb.ac.id

ABSTRAK

Menggunakan teori algoritma *Greedy* penulis ingin mencoba untuk membuat aplikasi dalam mesin penjual otomatis. Pada mesin penjual otomatis uang kembalian yang akan diterima akan di optimasi jumlah pecahannya. Hal ini karena pada tempat penyimpanan uang dalm mesin tersebut jumlahnya terbatas. Jadi jika uang kembalian yang diberikan lebih sedikit maka transaksi bisa semakin banyak. *Greedy* ang akan digunakan adalah terhadap besarnya nilai nominal dari uang tersebut.

Kata kunci: *Greedy*, optimasi, pecahan.

1. PENDAHULUAN

Melihat banyaknya perkembangan dibidang otomatisasi. Penulis ingin membahas salah satu aplikasi otomatis di bidang industri penjualan. Yang penulis ambil adalah tentang mesin penjual otomatis. Mesin ini memang belum banyak terdapat di Indonesia., akan tetapi sudah banyak terdapat di Negara berkembang lainnya.

Mesin penjual otomatis adalah mesin yang memungkinkan terjadinya transaksi pembelian barang secara otomatis(tanpa adanya manusia yang menjual barangnya). Biasanya barang yang dijual adalah minuman kaleng atau rokok. Mesin ini biasanya menerima masukan uang kertas dan uang kembaliannya adalah uang logam.

Yang ingin dibahas oleh penulis adalah tentang bagaimana mesin dapat memberikan kembalian secara optimal. Yang dimaksud optimal adalah uang kembalian mempunyai jumlah koin paling sedikit diantara semua kemungkinan uang kembalian.

Untuk itu program dari mesin akan mengaplikasikan algoritma *Greedy* untuk menentukan pecahan berapa saja yang bisa jadi kembaliannya. Diharapkan dengan menggunakan algoritma *Greedy* pembeli akan mendapatkan uang kembalian dengan jumlah koin sesedikit mungkin.

2. METODE

1. Dasar Teori Algoritma *Greedy*.

Algoritma *Greedy* adalah algoritma yang memecahkan masalah dengan membentuk solusi langkah per langkah. Untuk setiap langkah solusi akan dieksplorasi, oleh karena itu setiap langkah harus dibuat keputusan yang terbaik. Keputusan yang telah diambil tidak dapat diubah lagi pada langkah selanjutnya.

Pendekatan yang digunakan membuat pilihan tampak memberikan solusi terbaik yaitu optimum local. Sehingga diharapkan pada akhirnya memberikan solusi optimum global.

2. Skema umum Algoritma *Greedy*

Persoalan optimasi dalam konteks algoritma *Greedy* disusun oleh beberapa komponen sebagai berikut:

1. **Himpunan kandidat C**

   Himpunan ini berisi elemen-elemen pembentuk solusi persoalan. Contohnya adalah himpunan koin yang akan menjadi bahasan makalah ini. Pada setiap langkah,

satu buah kandidat diambil dari himpunannya.

2. Himpunan solusi S

Berisi kandidat-kandidat yang terpilih sebagai solusi persoalan. Dengan kata lain solusi adalah himpunan bagian dari himpunan kandidat.

3. **Fungsi Seleksi**

Fungsi yang pada setiap langkah memilih kandidat yang paling memungkinkan mencapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya.

4. Fungsi Kelayakan

Fungsi yang memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala yang ada. Kandidiat yang layak dimasukan kedalam kandidat solusi, sedangkan kandidat yang tidak layak dibuang.

5. **Fungsi objektif**

Fungsi yang memaksimalkan atau meminimalkan solusi.

3. Permasalahan yang dihadapi

Permasalahan yang dihadapi oleh penulis adalah tentang bagaimana sebuah mesin penjual otomatis dapat memberikan kembalian dalam pecahan yang optimal.

Untuk itu kita akan menggunakan prinsip *Greedy*.

Greedy yang akan dilakukan adalah *Greedy* dengan terhadap nilai nominal koin. Nilai nominal yang akan diambil adalah nilai nominal terbesar. Dengan harapan bahwa nilai yang terbesar diambil agar jumlah koin yang nilainya kecil tidak terlalu banyak. Misal kembalian 500 jiaka ada satu koin lima ratus lebih optimal daripada 5 buah koin 100.

Di Indonesia uang koin logam ada pada pecahan 100, 200, 500 dan 1000. Jadi mesin akan menyimpan uang kembalian sejumlah yang ditentukan dalam 4 buah pecahan. Untuk lebih jelasnya kita akan langsung ke contoh.

4. Solusi permasalahan

Contoh misalkan dari 4 buah pecahan mata uang yang telah disebutkan diatas, yaitu 100, 200, 500, dan 1000. Kita membeli berbagai barang yang totalnya mencapai 7200 rupiah. Pembeli membayar ke mesin dengan uang sebesar 10000 rupiah. Maka kembalian yang akan didapat adalah sebesar 2800 rupiah.

Dengan algoritma *Greedy* kita akan menguraikan uang tersebut dengan langkah-langkah sebagai berikut:

1. pilih 1 buah koin 1000 rupiah sebagai nilai tertinggi yang masuk kedalam fungsi kelayakan. Jadi kembalian tinggal 1800.
2. Pilih lagi 1000, hingga kembalian kuarng dariu 1000 rupiah. Jadi kita sekarang mempunyai 2 buah koin 1000 dan kembalian tinggal 800.
3. Pilih pecahan yang paling besar tetapi kurang dari kembalian yang tersisa. Pecahan 500 yang kita ambil.
4. Lakukan hal yang samauntuk setiap pecahan koin.

2800 = 1000 + 1000 + 500 + 200 + 100 (1)

Hasilnya kita akan mendapat 2 buah koin 1000, 1 buah koin 500, 1 buah koin 200, dan 1 buah koin 100. Pembeli akan mendapatkan 5 buah koin. Jika dicoba dengan algoritmabrute force maka akan didapatkan hasil yaitu 5 buah koin.

Artinya algoritma *Greedy* memberikan solusi yang optimal.

Kita coba lagi dengan kembalian contoh lain

Misalnya kita membeli barang dengan harga 3600 rupiah. Pembeli membayar denagn uang sebesar 5000 rupiah. Kita akan mencoba dengan cara yang sama. Seharusnya pembeli menerima kembalian sebesar 1400 rupiah

Maka didapat

1400= 1000+200+200. (2)

Jadi kita punya 1 koin 1000 dan 2 koin 200.

Jumlah koin kembalian adalah 3 koin.

Dan ternyata jika dengan algoritma brute force kita mendapat solusi optimal juga 3 koin. Itu artinya dengan *Greedy* mendapatkan solusi optimal.

Apakah dengan begitu kita pasti mendapat solusi optimal?

Bagai mana jika pecahanya lain?

Mari kita coba dengan pecahan yang berbeda. Misal nilai pecahannya 100 , 400, 600 , 1000.

Pembeli membeli barang seharga 18800 dan membayar dengan 20000 maka pembeli menerima kembalian 1200.

Jika menggunakan algoritma *Greedy* kita akan mengambil pecahan 1000 rupiah lalu pecahan 100 rupiah, lau 100 rupiah lagi. Jadi pecahan yang diambil

1200 = 1000 + 100 + 100 (3)

Pembeli mendapatkan 3 koin.

Jika dengan brute force kita mendapatkan solusi optimal dengan dua koin yaitu 2 buah koin 600.

Berarti dengan grredy tidak mendapatkan solusi optimal.

Artinya *Greedy* tak bisa menjamin pasti optimal.

Memang untuk kasus ketiga memang tidak optimal. Tetapi, tunggu dulu setelah diteliti lebih lanjut kasus ketiga tidak terjadi optimal karena pecahan koinnya. Pecahan koin memenuhi syarat agar selalu optimal adalah jika koin dikali 2 tidak akan lebih besar dari koin sesudahnya.

Contoh koin 200 jika ditambah koin 200 lebih kecil dari koin 500. Ini adalah contoh pecahan yang memenuhi syarat.

Contoh lainnya 400 jika ditambahkan 400 maka lebih besar dari 600.

Jadi jika koin yang digunakan pecahannya memenuhi syarat terbukti bahwa akan selalu optimal pada semua kasus.

Apakah mata uang Indonesia memenuhi syarat? Ternyata memenuhi syarat.

Jadi pada mesin penjual otomatis algoritma ini bisa diterapkan.

Pseudo-code untuk algoritma *Greedy*

```
Function change (input C :himpunan pecahan
koin, A: integer)-> himpunan koin

Deklarasi

S: himpunan koin

X: koin

Algoritma

S<-{}

While ((nilai semua koin di dalam S) != A)
and (C!= {}) do

X<-koin dengan nilai terbesar

C<- C-{X}

If ((nilai semua koin didalam S)+nilai koin
x<A) then

S<- S U {X}

Endif

Endwhile

If (nilai semua koin didalam S ==A) then

Return S

Else

Write ('tak ada solusi')

Endif
```

Dengan algoritma ini kompleksitas waktunya turun dari eksponensial menjadi kuadrat. Kompleksitasnya menjadi $O(n^2)$.

3. KESIMPULAN

Algoritma grredy bisa diterapkan sebagai salah satu cara untuk menyelesaikan permasalahan uang kembalian. Dengan pecahan mata uang Indonesia khususnya untuk uang logam, dapat dipastikan bahwa algoritma *Greedy* pasti mempunyai solusi optimal. Optimal disini adalah jumlah koin yang didapat berjumlah paling sedikit dari semua kemungkinan yang ada.

Akan tetapi algoritma *Greedy* tidak dapat menjamin pasri mendapatkan solusi optimal untuk semua kasus mata uang. Karena kepastian optimasi tergantung dari himpunan pecahan koin dari mata uang tersebut. Yang bisa dijamin optimasinya hanya jika pecahan yang sebelumnya dikali 2 hasilnya tidak akan lebih besar dari nilai pecahan sesudahnya.

Penulis mengharapkan algoritma ini bisa diterapkan pada mesin penjual otomatis agar pembeli tidak

terlalu banyak menerima kembalian dan mesin tersebut bisa menghemat menyimpan koin.

REFERENSI

Cormen, Thomas H. *Introduvtion to Algorithms,* The MIT Press,1989.

Munir, Rinaldi. *Diktat Kuliah IF2251 Strategi Algoritmik* , ITB..