Penerapan Algoritma *Greedy* Dalam Aplikasi Pembayaran Tiket Parkir Otomatis

Agung Dwi Lambang Gito Santosa 13508086
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
Email : if18086@students.if.itb.ac.id

***Abstract*—Menggunakan teori algoritma *Greedy* penulis ingin mencoba untuk membuat aplikasi dalam mesin pembayaran tiket parkir otomatis. Pada mesin pembayaran tiket otomatis uang kembaian yang akan diterima akan di optimasi jumlah pecahannya. Hal ini karena pada tempat penyimpanan uang dalam mesin tersebut jumlahnya terbatas. Jadi jika uang kembalian yang di berikan lebih sedikit maka transaksi bisa semakin optimal. *Greedy* yang akan digunakan adalah terhadap besarnya nilai nominal dari uang tersebut.**

***Kata kunci—Greedy,*optimum,pecahan.**

I. PENDAHULUAN

Melihat banyaknya perkembangan dibidang otomatisasi. Penulis ingin membahas salah satu aplikasi otomatis di bidang pelayanan. Yang penulis ambil adalah tentang mesin pembayaran tiket parkir otomatis. Mesin ini memang belum ada di Indonesia., akan mesin ini akan sangat membantu jika diaplikasikan.

Mesin pembayaran tiket parkir otomatis adalah mesin yang memungkinkan terjadinya transaksi pembayaraan secara otomatis.. 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* kita akan mendapatkan uang kembalian dengan jumlah koin sesedikit mungkin.

II. 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 Algoritma *Greedy*

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

- **Himpunan *Candidate***
  Adalah himpunan yang berisi elemen pembentuk dari solusi persoalan. Contohnya adalah himpunan koin kembalia, pada setiap langkah akan diambil satu buah anggota dari himpunan *candidate.*
- **Himpunan *Solution***
  Adalah anggota dari himpunan *candidate* yang merupakan solusi dari persoalan yang ada. Himpunan *solution* adalah himpunan bagian dari himpunan *candidate.*
- **Fungsi *Selection***
  Fungsi yang pada setiap langkah memilih *candidate* yang paling memungkinkan mencapai solusi optimal. *Candidate* yang sudah dipilih pada suatu langkah tidak akan

diperiksa ulang pada langkah berikutnya.

- **Fungsi Kelayakan**
  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.
- **Fungsi Objektif**
  Adalah fungsi yang memaksimalkan atau meminimalkan solusi masalah.

3. **Permasalahan yang dihadapi**

Permasalahan yang dihadapi oleh penulis adalah bagaimana mesin pembayaran tiket parkir otomatis dapat memberikan kembalian dalam pecahan yang optimal dengan menggunakan algoritma *greedy.*

Greedy yang akan digunakan adalah *Greedy* dengan terhadap nilai nominal koin. Nilai nominal yang akan diambil adalah nilai nominal terbesar. Agar jumlah koin yang di keluarkan optimal.

4. **Solusi permasalahan**

Contoh misalkan tarif parkir 1 jam pertama adalah 1000, dan tiap10 menit selanjutnya adalah 100. Dan dari 4 buah pecahan mata uang, yaitu 100, 200, 400,dan 1000. Kita parkir selama 1 jam 30 menit yang totalnya mencapai 1300 . Kita membayar ke mesin dengan uang sebesar 5000 . Maka kembalian yang akan didapat adalah sebesar 3700 .

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

1. Pilih 1 buah koin 1000 sebagai nilai tertinggi yang masuk kedalam fungsi kelayakan. Jadi kembalian tinggal 2700.
2. Pilih 1 lagi 1000, hingga kembalian kuarng dari 1000. Jadi kita sekarang mempunyai 2 buah koin 1000 dan kembalian tinggal 1700.
3. Pilih 1 lagi 1000, hingga kembalian kuarng dari 1000. Jadi kita sekarang mempunyai 2 buah koin 1000 dan kembalian tinggal 700.
4. Pilih pecahan yang paling besar tetapi kurang atau sama dengan dari kembalian yang tersisa. Pecahan 400 yang kita ambil. Jadi kembalian tinggal 300.
5. Pilih pecahan yang paling besar tetapi kurang atau sama dengan dari kembalian yang tersisa. Pecahan 200 yang kita ambil. Jadi kembalian tinggal 100.
6. Pilih pecahan yang paling besar tetapi kurang atau sama dengan kembalian yang tersisa. Pecahan 100 yang kita ambil. Jadi kembalian tinggal 300.

3700 = 1000 + 1000 + 1000 + 400 + 200 + 100 (1)

Hasilnya kita akan mendapat 3 buah koin 1000, 1 buah koin 400, 1 buah koin 200, dan 1 buah koin 100. Kita akan mendapatkan 6 buah koin. Jika dicoba dengan algoritma *brute force* maka akan didapatkan hasil yaitu 6 buah koin.

Artinya algoritma *Greedy* memberikan solusi yang optimal.

Kita coba lagi dengan kembalian contoh lain. Misalnya total pembayaraan tiket adalah 3600. Kita membayar dengan uang sebesar 5000. Kita akan mencoba dengan cara yang sama. Seharusnya Kita menerima kembalian sebesar 1400.

1400= 1000+400. (2)

Jadi kita punya 1 koin 1000 dan 1 koin 400. Jumlah koin kembalian adalah 2 koin. Dan ternyata jika dengan algoritma *brute force* kita mendapat solusi optimal juga 2 koin. Itu artinya dengan *Greedy* mendapatkan solusi optimal.

Pertanyaanya adalah, 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.Total yang harus kita bayar untuk parkir adalah1800 dan kita membayar dengan 3000 maka kita menerima kembalian 1200.

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

1200 = 1000 + 100 + 100 (3)

Kita mendapatkan 3 koin. Jika dengan *brute force* kita mendapatkan solusi optimal dengan dua koin yaitu 2 buah koin 600. Berarti dengan *Greedy* 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.**

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 pembayaran tiket otomatis bisa diterapkan.

Pseudo-code dari algoritma *Greedy*

```
Function change (input K : himpunan pecahan koin, A: integer )

Deklarasi

Sol : himpunan koin

M : koin

Algoritma

Sol<-()

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

 M<-koin dengan nilai terbesar
 K<- K-{M}
 If   ((nilai   semua   koin   didalam Sol)+nilai koin x < A) then
   Sol<- Sol U {M}
 Endif
Endwhile
If (nilai semua koin didalam Sol == A) then
  Return Sol
Else
  Write ('tidak ada solusi')
Endif
```

Dengan algoritma *Greedy* kompleksitas waktu menjadi $O(n^2)$. Kompleksitas waktu ini lebih cepat jika kita menggunakan algoritma *Brute Force*.

III. 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 berarti jumlah koin yang didapat berjumlah paling sedikit dari semua kemungkinan yang ada.

Akan tetapi algoritma *Greedy* tidak dapat menjamin mendapatkan solusi optimal untuk semua kasus mata uang. Karena tergantung dari himpunan pecahan koin dari mata uang tersebut.Algoritma *Greedy* akan optimal jika **pecahan yang sebelumnya dikali 2 hasilnya tidak akan lebih besar dari nilai pecahan sesudahnya**.

Penulis mengharapkan algoritma ini bisa diterapkan pada mesin pembayaran tiket parkir otomatis agar kita tidak terlalu banyak menerima kembalian dan mesin tersebut bisa menghemat menyimpan koin.

REFERENCES

[1] Munir, Rinaldi. Diktat Kuliah IF2251 Strategi Algoritmik ,

[2] www.wikipedia.com, tanggal akses 08-12-2010 jam 20.00

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, 9 Desember 2010

Agung Dwi Lambang Gito Santosa (13508086)