

**PROJECT DESAIN ANALISIS ALGORITMA
COIN CHANGE DENGAN ALGORITMA *GREEDY***

Laporan Desain Analisis Algoritma
Disusun untuk Memenuhi Tugas Mata Kuliah Desain Analisis Algoritma

Dosen Pengampu:
Fajar Muslim S.T., M.T.



Disusun Oleh:

1. Lidya Khairunnisa (L0123075)
2. Lu'lu'a Lim'a Laila (L0123076)

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA
UNIVERSITAS SEBELAS MARET
SURAKARTA**

2024

BAB I

DESKRIPSI MASALAH

Algoritma *greedy* adalah salah satu algoritma yang dapat digunakan untuk mendapatkan solusi terbaik dan merupakan algoritma yang paling populer saat ini. Aplikasi algoritma *greedy* dapat digunakan dalam masalah seperti: mencari jalur terpendek, strategi permainan monopoli, masalah penukaran uang dan lain-lain. Algoritma *greedy* adalah algoritma yang memecahkan masalah langkah per langkah, pada setiap langkah yaitu mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan juga berharap bahwa dengan memilih optimum lokal pada setiap langkah akan berakhir dengan optimum global.

Sebuah algoritma dikatakan *greedy* apabila membuat pilihan optimum lokal pada setiap langkah dengan harapan akan menemukan solusi yang optimal. Untuk beberapa kasus, algoritma *greedy* mudah diimplementasikan dan memiliki kecepatan program yang cukup tinggi.

Salah satu dari masalah optimasi adalah persoalan penukaran uang (*coin change*). Dimana persoalan ini membahas tentang bagaimana cara agar dapat menukar uang dengan nilai yang sama namun dengan jumlah uang yang lebih sedikit dari berbagai satuan. Persoalan ini merupakan persoalan umum yang biasa terjadi dalam kehidupan sehari-hari umat manusia.

Contohnya ketika kita ingin menukarkan uang Rp. 100.000 dengan uang 50.000 dan 10.000, dalam hal ini banyak kombinasi dan kemungkinan yang bisa terjadi. Maka dari itu dibutuhkan suatu algoritma yang tepat. Dalam persoalan diatas, algoritma yang dicari adalah yang memiliki ketepatan solusi yang cukup optimal dan kecepatan yang cukup tinggi. Oleh sebab itu, Algoritma Greedy akan digunakan untuk menyelesaikan persoalan penukaran uang ini.

BAB II

TEORI SINGKAT

Teori yang digunakan pada penerapan strategi algoritma dalam coin change adalah algoritma *Greedy*.

A. Definisi Algoritma *Greedy*

Algoritma *Greedy* adalah salah satu metode yang digunakan untuk menyelesaikan masalah optimasi, yaitu masalah yang tidak hanya mencari solusi, tetapi mencari solusi terbaik. Prinsip dasar algoritma ini adalah memilih solusi terbaik yang bisa diambil pada setiap langkah, tanpa mempertimbangkan konsekuensi jangka panjang. Meskipun metode ini sering kali memberikan solusi yang baik, hasilnya tidak selalu optimal. Algoritma *Greedy* sering dibandingkan dengan metode pencarian *brute force* dan *exhaustive search*, yang juga berfokus pada pencarian solusi, tetapi *Greedy* lebih efisien dalam mengambil keputusan di setiap tahap proses.

B. Penerapan Algoritma *Greedy*

Greedy akan menyelesaikan dengan mengutamakan kecepatan untuk menemukan suatu solusi, yang dianggap optimal. Langkah langkah yang dapat dilakukan dalam memecahkan *coin change* dengan metode *Greedy* adalah sebagai berikut:

- Diberikan sebuah target jumlah V rupiah dan sebuah list satuan uang N koin. Kita mempunyai nilai $Koin_i$ (dalam mata uang rupiah) untuk tipe koin i $[0 \dots N-1]$, Berapa jumlah minimum koin yang dibutuhkan untuk memperoleh jumlah V jika diasumsikan kita memiliki banyak koin untuk tiap satuan.
- Langkah 1 : Pilih 6 buah koin 10 (Total = 60)
- Langkah 2 : Pilih 1 buah koin 5 (Total = 60 + 5 = 65)

Maka dengan menggunakan algoritma *greedy* akan ditemukan solusi minuman dari koin pecahan yaitu Jumlah koin minimum = 7 (solusi optimal!) dimana untuk mendapatkan jumlah yang cocok dari uang adalah = 10, 5, 2, 1. Sedangkan jumlah maksimum dari uang adalah = koin pecahan 1 sebanyak input tersedia. Dengan menggunakan algoritma *greedy* didapatkan solusi menggunakan semua nilai koin yang tersedia. Masalah penukaran uang dinyatakan sebagai himpunan : Nilai uang yang ditukar : V , Himpunan koin (multiset): $\{i_1, i_2, \dots, i_n\}$. Himpunan solusi: $X = \{x_1, x_2, \dots, x_n\}$, $x_i = 1$ jika i_i dipilih, $x_i = 0$ jika i_i tidak dipilih.

BAB III

KODE DAN PENGUJIAN

A. Kode Program

1. Kode Program dalam *Python*

```
7 def bisaTambahKoin(totalTerkumpul, koin, totalUang):  
8     return (totalTerkumpul + koin) <= totalUang
```

Gambar 2.1 Fungsi ini memeriksa apakah koin saat ini bisa ditambahkan ke total yang sudah terkumpul tanpa melebihi jumlah uang yang harus ditukarkan.

```
10 def tukarUang(daftarKoin, totalUang):  
11     koinTerpakai = []  
12     jumlahTiapKoin = {koin: 0 for koin in daftarKoin}  
13     totalTerkumpul = 0  
14  
15     daftarKoin.sort(reverse=True)  
16  
17     while totalTerkumpul != totalUang:  
18         for koin in daftarKoin:  
19             while bisaTambahKoin(totalTerkumpul, koin, totalUang):  
20                 totalTerkumpul += koin  
21                 koinTerpakai.append(koin)  
22                 jumlahTiapKoin[koin] += 1  
23  
24     return koinTerpakai, jumlahTiapKoin
```

Gambar 2.2 Fungsi ini berperan sebagai inti dari algoritma greedy untuk melakukan penukaran uang dengan menggunakan koin dari daftarKoin.

```
26 if __name__ == "__main__":  
27  
28     print("="*40)  
29     print(f'{"PROGRAM TUKAR UANG":^40}')  
30     print("="*40)  
31  
32     totalUang = int(input("Masukkan jumlah uang yang ingin ditukar: "))  
33     daftarKoin = [10, 5, 2, 1]  
34  
35     daftarKoinTerpakai, jumlahTiapKoin = tukarUang(daftarKoin, totalUang)  
36  
37     print(f"Uang Yang Ditukar : {totalUang}")  
38     print(f"Himpunan Koin      : {daftarKoin}")  
39     print(f"Solusi              : {daftarKoinTerpakai}")  
40     print("="*40)  
41     print("Jumlah setiap koin yang dipakai:")  
42     for koin, jumlah in jumlahTiapKoin.items():  
43         print(f"Koin {koin:<3}: {jumlah} buah")  
44     print("="*40)  
45
```

Gambar 2.3 Header saat program dijalankan, input pengguna, pemanggilan fungsi dan output yang akan dihasilkan.

B. Testing dan Hasil Tangkap Layar Input Output

1. Input Output *Python*

```
=====
                        PROGRAM TUKAR UANG
=====
Masukkan jumlah uang yang ingin ditukar:
```

Gambar 2.6 Output awal saat kode dijalankan

```
=====
                        PROGRAM TUKAR UANG
=====
Masukkan jumlah uang yang ingin ditukar: 65
```

Gambar 2.7 Saat pengguna mencoba memasukkan angka 65

```
=====
                        PROGRAM TUKAR UANG
=====
Masukkan jumlah uang yang ingin ditukar: 65
Uang Yang Ditukar : 65
Himpunan Koin      : [10, 5, 2, 1]
Solusi              : [10, 10, 10, 10, 10, 10, 5]
=====
Jumlah setiap koin yang dipakai:
Koin 10 : 6 buah
Koin 5  : 1 buah
Koin 2  : 0 buah
Koin 1  : 0 buah
=====
PS D:\DARI C\Documents\KULIAH\sem 3\DAA>
```

Gambar 2.8 Hasil dari input yang dimasukkan

BAB IV

KESIMPULAN

Algoritma *greedy* untuk masalah penukaran uang memungkinkan kita menemukan solusi secara langsung tanpa mempertimbangkan solusi lain yang mungkin lebih optimal. Algoritma ini mudah diimplementasikan dan memiliki kompleksitas yang efisien, baik dari segi waktu maupun penggunaan ruang (space). Namun, meskipun demikian, hasil yang diperoleh tidak selalu merupakan solusi yang paling optimal.

Di dalam proyek Desain Analisis Algoritma ini, kami berhasil membuat sebuah *source code* dalam bahasa *python* dan kami berhasil melakukan penyelesaian *coin change* dengan *algoritma greedy* dengan cukup baik.

REFERENSI

- Coin Changing Revisited. (n.d.). <https://condor.depaul.edu/~rjohnson/algorithm/coins.pdf>
- Harianja, A.P. dan Sitepu, P. B.. (2018). IMPLEMENTASI PENUKARAN UANG RUPIAH DENGAN MENGGUNAKAN ALGORITMA GREEDY. Jurnal Teknik Informatika Unika St. Thomas (JTIUST), 03 (02), 152-157.
- Ihsan, I., & Aziz, R. (2019). Solusi Optimal Coin Change Problem dengan Algoritma Greedy dan Dynamic Programming. Diakses dari https://www.researchgate.net/publication/332440755_Solusi_Optimal_Coin_Change_Problem_dengan_Algoritma_Greedy_dan_Dynamic_Programming pada 30 September 2024.