
Implementasi Algoritma Greedy Pada Permainan Coin Exchange Problem

Muhammad Afrian Suwandi – L0123087¹, Muhammad Rafli Werizky – L0123097²

^{1,2} Fakultas Teknologi Informasi dan Sains Data, Universitas Sebelas Maret

¹ srian744@gmail.com; ² rafliwrzky@gmail.com

Abstrak

Coin exchange adalah permainan yang bertujuan untuk menemukan apakah nilai yang dimasukkan dapat dibagi habis dengan nilai atau angka koin yang ada atau disediakan. Penggunaan algoritma yang tepat dapat membantu untuk melakukan pembagian sesuai prioritas nilai koin tertinggi. Salah satu algoritma yang dapat dilakukan adalah menggunakan algoritma *Greedy*. Tulisan ini akan membahas prinsip algoritma *Greedy* dan penerapan dalam permainan *coin exchange* yang akan digunakan pada terminal.

Kata-kata kunci : coin exchange, greedy

Abstract

Coin exchange is a game that aims to find out whether the entered value can be divided evenly by the value or number of coins available or provided. Using the right algorithm can help to divide according to the highest coin value priority. One algorithm that can be done is using the Greedy algorithm. This article will discuss the principles of the Greedy algorithm and its application in the coin exchange game that will be used on the terminal.

Keywords : coin exchange, greedy

I. PENDAHULUAN

Coin Change adalah permainan mengasah otak dimana sejumlah nilai bilangan bulat akan dipecah menjadi serangkaian koin. Tujuan dari permainan ini adalah menggunakan koin-koin untuk mengumpulkan sejumlah nilai tertentu dengan menggunakan koin terbesar yang memiliki jumlah koin paling sedikit atau disebut juga paling optimal. Namun, peraturan dari permainan ini adalah jika nilai bilangan bulat dengan masukkan dari pengguna tidak bisa diselesaikan dengan koin yang disediakan, maka nilai tersebut tidak bisa dipecah atau nilai tersebut salah

II. DASAR TEORI

A. Algoritma Greedy

Algoritma greedy adalah algoritma yang memecahkan masalah optimasi dengan cara yang terlihat rakus dan pengambilan keputusan secara langsung atau sekarang dengan harapan setiap langkah akan membawa lebih dekat ke solusi akhir yang optimal. Dalam konteks greedy, akan selalu memilih opsi yang paling menguntungkan saat itu tanpa mempertimbangkan konsekuensi di masa depan. Prinsip utamanya yaitu dilakukan selagi bisa, atau melakukan yang bisa dilakukan sekarang. Algoritma *greedy* tidak mempertimbangkan, namun langsung dilakukan, sesuai dengan pemikiran logis yang muncul, tanpa memikirkan dampak dari masalah yang sedang dihadapi.

Kegunaan utama dari algoritma *greedy* adalah menemukan solusi optimal dalam persoalan optimasi

dengan cepat. Pendekatan ini sangat berguna dalam banyak kasus di mana seseorang perlu memaksimalkan atau meminimalkan sesuatu dengan cara yang efisien.

Kelebihan Algoritma *Greedy* :

1. Efektif untuk memecahkan masalah optimasi jika mengandung sub struktur yang optimal.
2. Efisien karena memberikan solusi yang cepat, meskipun tidak selalu yang terbaik.

Kekurangan Algoritma *Greedy* :

1. Algoritma greedy tidak beroperasi secara menyeluruh terhadap semua kemungkinan solusi yang ada. (tidak seperti metode exhaustive search).
2. Terdapat beberapa fungsi seleksi yang berbeda, sehingga kita harus memilih fungsi yang tepat jika kita ingin algoritma menghasilkan solusi optimal.

B. Permainan Coin Exchange

Permainan coin exchange adalah permainan matematika yang melibatkan pembagian dari suatu angka menggunakan operasi aritmatika dasar yaitu pembagian untuk menghabiskan suatu nilai yang dimasukkan oleh pengguna. Permainan ini akan diselesaikan jika nilai yang dimasukkan oleh pengguna dapat dibagi habis dengan koin yang disediakan.

Masalah ini mencerminkan situasi sehari-hari, seperti ketika seseorang ingin memberikan uang kembalian atau membagi uang dalam denominasi tertentu dengan cara yang paling efisien.

III. PEMBAHASAN

Berikut adalah hasil akhir beserta contoh-contoh outputnya dan pembahasan algoritma *greedy* yang dipakai. Untuk mengingat proyek ini dikhususkan bagi sistem operasi windows dikarenakan penggunaan `os.system("pause")` dan `os.system("cls")`. Bagi pengguna sistem operasi lain dikhawatirkan akan menimbulkan *error* atau output yang ditampilkan akan berantakan.

A. Algoritma Greedy

Pada laporan ini, aturan permainan coin exchange yang digunakan adalah langsung memecahkan masalah yang sudah ada. Namun ditambahkan menu untuk mengganti nilai koin dan juga mengganti nilai target koin.

a. Implementasi Greedy

Algoritma greedy yang diimplementasikan pada laporan ini, memprioritaskan penggunaan koin terbesar. Tujuan utamanya adalah meminimalkan jumlah koin yang digunakan. Berikut adalah kodenya

```
coin = [1, 2, 5, 10] # Coin awal
totalTargetCoin = 33 # Target total coin awal
```

Gambar 3.1. Koin awal
Sumber: Penulis

Di atas adalah koin awal dan target koin awal yang disediakan. Fungsinya sebagai ketetapan awal jika pengguna tidak ingin mengubah coin dan target jumlah coin.

```
def solveCEP():
    solusiCoin = []
    coinTersedia = coin[:]

    os.system("cls")
    while sum(solusiCoin) != totalTargetCoin:
        if not coinTersedia:
            break

        maks = max(coinTersedia)
        if sum(solusiCoin) + maks <= totalTargetCoin:
            solusiCoin.append(maks)
        else:
            coinTersedia.remove(maks)

    if sum(solusiCoin) == totalTargetCoin:
        solusi_dict = {}
        for koin in solusiCoin:
            if koin in solusi_dict:
                solusi_dict[koin] += 1
            else:
                solusi_dict[koin] = 1

        print("Solusi Coin:")
        for koin, jumlah in solusi_dict.items():
            print(f"{koin} : {jumlah} keping")
    else:
        print("Tidak ada solusi yang ditemukan")

    print("")
    os.system("pause")
    menu()
```

Gambar 3.2. Greedy
Sumber: Penulis

Di atas adalah penggunaan algoritma greedy dan sebagai kode utama dari program coin exchange yang telah dibuat. Pertama-tama inialisasi solusi dengan array kosong dan juga satu variabel yang digunakan untuk sebagai *dummy* dari algoritma yang akan menghapus nilai maksimum di dalam array jika telah melewati nilai atau sama dengan target jumlah coin.

Algoritma akan *looping* selama jumlah dari solusi coin tidak sama dengan solusi coin. Pengecekan untuk fungsi solusi dengan mengambil nilai maximum dari coin yang tersedia. Jika jumlah nilai di dalam solusi coin ditambah nilai maksimum dari array yang diambil sebelumnya kurang dari sama dengan target maka nilai maksimum tersebut akan diambil untuk menjadi solusi coin pertama. Begitu pun seterusnya sampai jumlah solusi coin ditambah nilai maksimum lebih dari target coin. Jika telah melebihi target maka nilai maksimum tersebut akan dihapus dari list array *dummy* coin. Setelah target dicapai maka hasil akan diprint dan dihitung jumlah berapa kali coin tersebut muncul.

B. Program

Penyelesaian permainan *coin exchange problem* dengan algoritma greedy diimplementasikan menggunakan bahasa pemrograman Python. Dalam permainan *coin exchange problem*, kita menambahkan menu untuk membantu pengguna dan sebagai fitur tambahan.

a. Main Menu

```
# Dictionary untuk opsi menu
opsiMenu = {
    "1": solveCEP,
    "2": updateCoin,
    "3": updateTargetCoin,
    "4": about,
    "5": exitProgram,
}

def menu():
    os.system("cls")
    judul()

    # Menampilkan nilai awal dan target coin
    print(f"Nilai Coin yang tersedia: {coin}")
    print(f"Target Jumlah Coin: {totalTargetCoin}")

    print("\n1. Solve Problem")
    print("\n2. Update Coin Values")
    print("\n3. Update Total Target Coin")
    print("\n4. About")
    print("\n5. Exit")

    pilihan = input("\nMasukkan pilihan (1 - 5): ")
    action = opsiMenu.get(pilihan)
    if action:
        action()
    else:
        print("Pilihan Tidak Valid")
        menu()

menu()
```

Gambar 3.3 Main menu
Sumber: Penulis

Fungsi `menu()` dipertujukan untuk interaksi dengan pengguna dengan menggunakan *dictionary* pada python dengan pencegahan *input* selain opsi yang ada di menu.

b. Menu Solve

```
def solveCEP():
    solusiCoin = []
    coinTersedia = coin[:]

    os.system("cls")
    while sum(solusiCoin) != totalTargetCoin:
        if not coinTersedia:
            break

        maks = max(coinTersedia)
        if sum(solusiCoin) + maks <= totalTargetCoin:
            solusiCoin.append(maks)
        else:
            coinTersedia.remove(maks)

    if sum(solusiCoin) == totalTargetCoin:
        solusi_dict = {}
        for koin in solusiCoin:
            if koin in solusi_dict:
                solusi_dict[koin] += 1
            else:
                solusi_dict[koin] = 1

        print("Solusi Coin:")
        for koin, jumlah in solusi_dict.items():
            print(f"{koin} : {jumlah} keping")
        else:
            print("Tidak ada solusi yang ditemukan")

    print("")
    os.system("pause")
    menu()
```

Gambar 3.4. Menu solve

Sumber: Penulis

Fungsi `SolveCEP()` merupakan algoritma utama untuk memecahkan permasalahan *coin exchange* dengan menggunakan algoritma *greedy* sesuai dengan subbab sebelumnya.

c. Menu Update Coin

```
def updateCoin():
    global coin
    os.system("cls")
    while True:
        coin_temp = input("Masukkan Coin yang Tersedia (Pisahkan dengan spasi, contoh: 1 2 5 10): ")
        try:
            coin_list = list(map(int, coin_temp.split()))
            if any(koin <= 0 for koin in coin_list):
                print("Nilai koin harus bilangan positif. Tidak ada nilai negatif atau nol yang diizinkan.")
            else:
                coin = coin_list
                os.system("cls")
                print(f"Coin berhasil diperbarui menjadi: {coin}")
                break
        except ValueError:
            print("Input tidak valid. Pastikan Anda hanya memasukkan angka. Silakan coba lagi.")

    print("")
    os.system("pause")
    menu()
```

Gambar 3.5. Menu update coin

Sumber: Penulis

Fungsi `updateCoin()` diperuntukan untuk memperbarui nilai coin-coin untuk menjadi sesuai dengan kemauan pengguna. Untuk memperbarui memiliki syarat menambahkan spasi (" ") untuk di setiap angka yang pengguna masukkan.

Fungsinya juga memiliki fitur untuk menyaring *input* dari pengguna hanya bilangan positif. Jika tidak memenuhi maka akan *looping* sampai *input* benar.

d. Menu Update Target Coin

```
def updateTargetCoin():
    global totalTargetCoin
    os.system("cls")
    while True:
        try:
            targetCoin_temp = int(input("Masukkan Target Coin: "))
            if targetCoin_temp <= 0:
                print("Target Coin harus lebih besar dari 0. Silakan coba lagi.")
            else:
                totalTargetCoin = targetCoin_temp
                os.system("cls")
                print(f"Target Coin diperbarui menjadi: {totalTargetCoin}")
                break
        except ValueError:
            print("Input tidak valid. Silakan masukkan angka yang benar.")

    print("")
    os.system("pause")
    menu()
```

Gambar 3.6. Menu update target coin

Sumber: Penulis

`UpdateTargetCoin()` juga merupakan opsi menu agar pengguna bebas menentukan target jumlah coin yang diinginkan. Fungsi ini juga memiliki fitur menyaring *input* yang tidak diinginkan seperti negatif atau 0.

e. Menu About

```
def about():
    os.system("cls")
    print('About: ...')
    • Muhamad Afrian Suwandi - L0123087 - Sebelas Maret University
    • Muhammad Rafli Werizky - L0123097 - Sebelas Maret University
    print('')

    os.system("pause")
    menu()
```

Gambar 3.6. Menu about

Sumber: Penulis

Fungsi `about()` merupakan fitur tambahan untuk menampilkan pembuat code.

f. Menu Exit

```
def exitProgram():
    os.system("cls")
    print("Terima kasih!! Semoga Selamat Dunia Akhirat")
    exit()
```

Gambar 3.7. Menu exit

Sumber: Penulis

Fungsi `exitProgram()` juga merupakan fitur tambahan agar pengguna dapat keluar dari program jika telah selesai menggunakan program

IV. HASIL PENELITIAN

Dibawah ini merupakan hasil *output* sampel dari berbagai kombinasi koin dan target koin, kemudian akan mengeluarkan kemungkinan sub optimal dari angka-angka tersebut dalam bentuk terminal.

Target	Koin	Output
33	1, 2, 4, 10	<pre>Solusi Coin: 10 : 3 keping 2 : 1 keping 1 : 1 keping Press any key to continue . . .</pre>
8	1, 3, 4, 5	<pre>Solusi Coin: 5 : 1 keping 3 : 1 keping Press any key to continue . . .</pre>

Target	Koin	Output
7	1, 3, 4, 5	Solusi Coin: 5 : 1 keping 1 : 2 keping Press any key to continue . . .
13	3, 4, 5, 6	Tidak ada solusi yang ditemukan Press any key to continue . . .

Tabel 4.1. Hasil Pengujian

Sumber : Penulisan

Di atas adalah output dari nilai dan target nilai yang ada. Dalam laporan ini, dibuat kasus yang dapat diselesaikan, kasus yang tidak dapat diselesaikan, dan kasus yang tidak optimal. Contoh kasus yang dapat diselesaikan yaitu coin 1, 2, 4, 10 dengan target coin 33 akan memunculkan output 10 sebanyak 3 keping, 2 sebanyak 1 keping, dan 1 sebanyak 1 keping. Contoh kasus yang dapat diselesaikan yaitu coin 1, 3, 4, 5 dengan target coin 8 akan memunculkan output 5 sebanyak 1 keping dan 3 sebanyak 1 keping. Contoh kasus yang tidak dapat diselesaikan yaitu coin 3, 4, 5, 6 dengan target coin 13, outputnya tidak ada karena algoritma greedy akan membagi 13 dengan 6 yang selanjutnya tidak bisa diselesaikan dengan dibagi coin yang lebih kecil dari 6. Contoh kasus yang dapat diselesaikan dengan tidak optimal yaitu 1, 3, 4, 5 dengan target coin 7, output yang dihasilkan adalah 5 sebanyak 1 keping dan 1 sebanyak 2 keping, hasil optimalnya adalah 4 sebanyak 2 keping.

V. KESIMPULAN DAN SARAN

Dari hasil penelitian dan analisis yang telah dilakukan, didapatkan bahwa algoritma greedy bisa digunakan untuk memperoleh solusi dengan memaksimalkan atau meminimalkan sesuatu dengan cara yang efisien. Permainan ini masih tergolong mudah dan sederhana untuk diterapkan algoritma greedy. Algoritma greedy dalam coin exchange problem biasanya akan menghasilkan solusi yang optimal. Namun, dalam beberapa kasus, algoritma greedy akan menemukan solusi sub-optimal, atau bisa dibilang bukan yang paling optimal namun mendekati yang paling optimal.

VI. PENUTUP

Puji dan syukur kami panjatkan kepada Tuhan Yang Maha Esa atas berkat rahmat dan karunia-Nya, kami dapat menyelesaikan laporan ini dengan lancar. Kami mengucapkan terima kasih kepada Bapak Fajar Muslim, S.T., M.T., atas bimbingannya selama masa perkuliahan sehingga kami dapat menyelesaikan laporan ini dengan baik. Kami juga ingin mengucapkan terimakasih atas materi dan referensi yang telah diberikan. Semoga dengan kemurahan hati Bapak, kami diberikan nilai A untuk Mata Kuliah Design dan Analisis Algoritma.

VII. LAMAN TERKAIT

Link repository GitHub :

<https://github.com/Riansuwandi/Coin-Exchange-Problem>

VIII. DAFTAR PUSTAKA

- [1] "MUNIR, RINALDI. "ALGORITMA GREEDY BAGIAN 1, " URL: [https://informatika.stel.itb.ac.id/~rinaldi.munir/stmik/2020-2021/ALGORITMA-GREEDY-\(2021\)-BAG1.PDF](https://informatika.stel.itb.ac.id/~rinaldi.munir/stmik/2020-2021/ALGORITMA-GREEDY-(2021)-BAG1.PDF) DIAKSES PADA 23 SEPTEMBER 2024
- [2] "MUNIR, RINALDI. "ALGORITMA GREEDY BAGIAN 2, " URL: [https://informatika.stel.itb.ac.id/~rinaldi.munir/stmik/2020-2021/ALGORITMA-GREEDY-\(2021\)-BAG2.PDF](https://informatika.stel.itb.ac.id/~rinaldi.munir/stmik/2020-2021/ALGORITMA-GREEDY-(2021)-BAG2.PDF) DIAKSES PADA 24 SEPTEMBER 2024
- [3] "MUNIR, RINALDI. "ALGORITMA GREEDY BAGIAN 3, " URL: [https://informatika.stel.itb.ac.id/~rinaldi.munir/stmik/2021-2022/ALGORITMA-GREEDY-\(2022\)-BAG3.PDF](https://informatika.stel.itb.ac.id/~rinaldi.munir/stmik/2021-2022/ALGORITMA-GREEDY-(2022)-BAG3.PDF) DIAKSES PADA 25 SEPTEMBER 2024
- [4] "ALGORITMA GREEDY: PENGERTIAN, JENIS DAN CONTOH PROGRAM URL: <https://fiktlumsu.ac.id/ALGORITMA-GREEDY-PENGERTIAN-JENIS-DAN-CONTOH-PROGRAM/> DIAKSES PADA 25 SEPTEMBER 2024

IX. DAFTAR TUGAS

Nama	Tugas
Muhammad Afrian Suwandi L0123087	<ul style="list-style-type: none"> Membuat Algoritma Greedy Menggunakan Python Membuat Menu Tambahan Membuat Laporan Membuat Video Demo
Muhammad Rafli Werizky L0123097	<ul style="list-style-type: none"> Membuat Algoritma Greedy Menggunakan Python Membuat Menu Tambahan Membuat Laporan Membuat Video Demo