LAPORAN TUGAS MATA KULIAH
DESAIN & ANALISIS ALGORITMA

“COIN CHANGE PROBLEM”

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

Disusun Oleh :

Mutia Rahman (L0123101)

Oktavia Suci Rahmadhani (L0123110)

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA

UNIVERSITAS SEBELAS MARET

2024

A. Penjabaran Masalah

Program Coin change adalah sebuah program yang bertujuan untuk untuk menemukan kombinasi koin yang dapat menjumlahkan ke suatu nilai target tertentu dengan menggunakan jumlah koin sesedikit mungkin. Program ini mengambil pendekatan berbasis metode greedy, yang berarti program akan memilih kemungkinan terbaik pada saat itu, yaitu dengan mengambil koin dengan nilai tertinggi terlebih dahulu, sehingga lebih mungkin untuk mencapai solusi dengan jumlah koin lebih sedikit.

Masalah ini dikenal sebagai masalah Minimasi Jumlah Koin (Coin Change Problem). Ini adalah masalah klasik dalam optimasi yang dapat diselesaikan menggunakan beberapa pendekatan, termasuk algoritma greedy. Pada masalah ini, kita akan diberikan sejumlah koin dengan denominasi tertentu. Tugas kita adalah menemukan kombinasi koin dengan jumlah koin sesedikit mungkin untuk mencapai nilai uang tertentu yang ditentukan.

Masalah yang Diselesaikan: Tujuan dari program ini adalah mencari kombinasi koin sesedikit mungkin yang dapat mencapai nilai target. Jika solusi ditemukan, program akan menampilkannya, jika tidak, program akan memberi tahu bahwa tidak ada solusi yang ditemukan.

B. Implementasi Kode

Berikut adalah implementasi kode untuk program tersebut:

```
def get_minimum_coins(money, coins):
    result = {}
    count_coin = 0

    # Urutkan koin dari terbesar ke terkecil
    coins.sort(reverse=True)

    for coin in coins:
        if money <= 0:
            break
        count = money // coin  # Berapa banyak koin yang bisa
digunakan
        if count > 0:
            money -= coin * count  # Kurangi uang dengan
nilai koin yang digunakan
            result[coin] = count   # Simpan jumlah koin yang
digunakan
            count_coin += count    # Tambahkan jumlah koin ke
total

    if money > 0:
        return None, None  # Tidak ada solusi jika uang
tersisa tidak bisa dipecah dengan koin yang ada
    return result, count_coin
```

```
def main():
    try:
        money = int(input("Masukkan nominal uang = "))
        if money <= 0:
            print("Nominal uang harus lebih dari 0.")
            return

        # Meminta input semua nilai koin dalam satu baris
        coin_values = input("Masukkan nominal koin pecahan
yang diinginkan (pisahkan dengan spasi) = ")
        coins = list(map(int, coin_values.split()))  #
Mengonversi input menjadi list integer

        if not coins or any(c <= 0 for c in coins):
            print("Harap masukkan nilai koin yang valid.")
            return

        result, count_coin = get_minimum_coins(money, coins)

        if result is None:
            print("Tidak ada kombinasi koin yang dapat
memenuhi jumlah uang tersebut.")
        else:
            print(f"Banyak koin yang didapat = {count_coin}")
            print("Rincian = ")
            for coin, count in result.items():
                print(f"Koin {coin} = {count} keping")

    except ValueError:
        print("Input tidak valid. Masukkan angka yang
benar.")

if __name__ == "__main__":
    main()
```

C. Pengujian Program

- Uji coba program untuk input nominal uang bernilai nol

```
Masukkan nominal uang = 0
Nominal uang harus lebih dari 0.
```

- Uji coba program untuk kombinasi input yang tidak memiliki solusi

```
Masukkan nominal uang = 230
Masukkan nominal koin pecahan yang diinginkan (pisahkan dengan spasi) = 10 20 25 50
Tidak ada kombinasi koin yang dapat memenuhi jumlah uang tersebut.
```

- Uji coba program untuk kombinasi input yang memiliki solusi namun tidak optimal

```
Masukkan nominal uang = 15
Masukkan nominal koin pecahan yang diinginkan (pisahkan dengan spasi) = 10 7 1
Banyak koin yang didapat = 6
Rincian =
Koin 10 = 1 keping
Koin 1 = 5 keping
```

- Uji coba program untuk kombinasi input yang memiliki solusi dan optimal

```
Masukkan nominal uang = 30
Masukkan nominal koin pecahan yang diinginkan (pisahkan dengan spasi) = 1 5 10 25
Banyak koin yang didapat = 2
Rincian =
Koin 25 = 1 keping
Koin 5 = 1 keping
```

- Uji coba program untuk input yang tidak valid

```
Masukkan nominal uang = 20
Masukkan nominal koin pecahan yang diinginkan (pisahkan dengan spasi) = satu dua lima sepuluh
Input tidak valid. Masukkan angka yang benar.
```

	Ya	Tidak
Program berhasil dikompilasi tanpa kesalahan	☑	☐
Program berhasil running	☑	☐
Program dapat membaca input dan memberikan output	☑	☐
Solusi yang diberikan program memenuhi (jika tersedia)	☑	☐

D. Kesimpulan

Setelah dilakukan pengujian menyeluruh, program "Coin Change" ini menunjukkan performa yang cukup baik. Program mampu menampilkan pesan kesalahan dengan tepat ketika input yang dimasukkan tidak memiliki solusi, seperti saat pengguna memasukkan nilai koin negatif, nominal uang nol, atau input non-numerik. Dengan validasi yang kuat, pengguna akan mendapat umpan balik yang jelas jika terjadi kesalahan input. Program juga berhasil dikompilasi tanpa kendala dan berjalan dengan lancar di lingkungan eksekusi yang sesuai. Proses pembacaan input serta penampilan output dilakukan dengan benar, memberikan solusi jika tersedia.

Namun, seperti kebanyakan algoritma greedy, program ini tidak selalu memberikan solusi optimal untuk semua kombinasi koin. Dalam beberapa kasus, terutama saat koin-koin

bernilai besar tidak dapat secara efisien mencapai jumlah uang yang diminta, hasil yang diberikan mungkin tidak lengkap atau kurang akurat. Meskipun demikian, program ini tetap efektif untuk situasi di mana kombinasi koin dapat dipecahkan dengan pendekatan greedy, serta dilengkapi dengan validasi input yang memastikan bahwa pengguna memasukkan data yang benar.

E. Referensi

Munir, R. (n.d.). IF2211 Strategi Algoritma - Semester II Tahun 2023/2024. Informatika. Retrieved September 22, 2024, from https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2023-2024/stima23-24.htm

F. Pembagian Tugas

Nama	Tugas
Mutia Rahman	● Mencari referensi untuk laporan ● Mengembangkan dan mengimplementasikan kode ● Membuat laporan ● Membuat dan mengedit video demo
Oktavia Suci Rahmadhani	● Mencari referensi kode ● Mengembangkan dan mengimplementasikan kode ● Membuat laporan ● Membuat video demo