

LAPORAN DESAIN ANALISIS ALGORITMA
PROGRAM COIN GREEDY



Disusun Oleh :

- | | |
|-------------------------------|------------|
| 1. Nadzira Rifqi Amin Rinawan | (L0123104) |
| 2. Muhammad Hafiz Dicky Putra | (L0123096) |

Dosen Pengajar :

Fajar Muslim S.T., M.T.

FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA
UNIVERSITAS SEBELAS MARET
2024

Permasalahan

Greedy Coin merupakan sebuah program yang mengimplementasikan *Greedy Algorithm* atau algoritma rakus untuk menyelesaikan masalah **coin change**. Algoritma greedy pada prinsipnya mengambil keputusan terbaik di setiap langkah dengan harapan solusi akhirnya adalah yang terbaik secara keseluruhan. Bertujuan untuk menentukan jumlah minimum koin dari koin yang tersedia (1,2,5,10) yang diperlukan untuk mencapai sejumlah nilai tertentu.

Implementasi

Langkah-langkah menjalankan program :

1. Memasukan Nilai uang yang ingin dicari.
2. Tekan “Enter” untuk menjalankan program.

Pengujian Program

```
def check(amount):
    coin_list = [10,5,2,1]
    coin_names = ["coin 10", "coin 5", "coin 2", "coin 1"]
    count = {
        "coin 10":0,
        "coin 5":0,
        "coin 2":0,
        "coin 1":0
    }
```

- coin_list berfungsi untuk menyimpan nilai-nilai koin yang akan digunakan untuk membuat pecahan dari uang target atau disebut juga sebagai Himpunan Kandidat.
- coin_names adalah representasi yang akan muncul pada output untuk nilai yang terdapat pada coin_list.
- count adalah tempat untuk menyimpan jumlah dari koin-koin yang dihitung.

```
for i in range(len(coin_list)):
    while amount >= coin_list[i]:
        amount -= coin_list[i]
        count[coin_names[i]] += 1
```

- for i in range(len(coin_list)) : berfungsi untuk memeriksa setiap koin yang dapat digunakan untuk menyesuaikan dengan nilai 'amount' atau nilai target.
- while amount >= coin_list[i] : berfungsi untuk memeriksa apakah nilai 'amount' masih bisa dikurangi dengan nilai koin yang terdapat pada 'coin_list'.
- amount -= coin_list[i] : berfungsi untuk mengurangi nilai 'amount' dengan nilai koin yang telah dipilih.
- count[coin_names[i]] += 1 : berfungsi untuk menambah jumlah koin yang telah dipilih.

```
if amount == 0:
    print("The amount of coin change :")
    for category in count:
        print(f"{category} = {count[category]}")
else:
    print("Ini gabisa")
```

- Kode diatas berarti jika nilai 'amount' telah habis maka program akan menampilkan <"The amount of coin change :">
print(f"{category} = {count[category]}") berfungsi menampilkan daftar koin yang digunakan untuk menghitung uang pada nilai 'amount'
- jika user salah memasukkan nilai target atau 'amount' maka akan menampilkan "Ini gabisa"

```
total_money = float(input("Please insert the money : "))
check(total_money)
```

- Bagian ini meminta user untuk memasukkan jumlah uang. User akan mengetikkan angka, dan angka tersebut akan dibaca oleh program sebagai float, tetapi walaupun disini tertulis float karena nilai Himpunan Kandidatnya bernilai bilangan bulat maka program tidak dapat mengitungnya.
- check(total_money) berfungsi untuk menghitung berapa banyak koin yang dibutuhkan untuk memberikan kembalian senilai 'total_money'

Tampilan Program

- Tampilan jika program menemukan solusi

```
Please insert the money : 19
The amount of coin change :
coin 10 = 1
coin 5 = 1
coin 2 = 2
coin 1 = 0
```

- Tampilan jika program tidak menemukan hasil dari perhitungan

```
Please insert the money : 1.2
Ini gabisa
```

Peran Anggota Kelompok

1. Nadzira Rifqi Amin Rinawan : Membuat algoritma perhitungan menggunakan python, dan membantu membuat laporan dokumentasi.
2. Muhammad Hafiz Dicky Putra : Membantu membuat algoritma dan membuat laporan dokumentasi.