

## Laporan praktikum daa

Nama : Yohanes Yeningga

Nim : 20220047

Matkul : prak.daa latihan 1

A. Fungsi `give_change` memiliki dua parameter, yaitu `amount` dan `coins`.

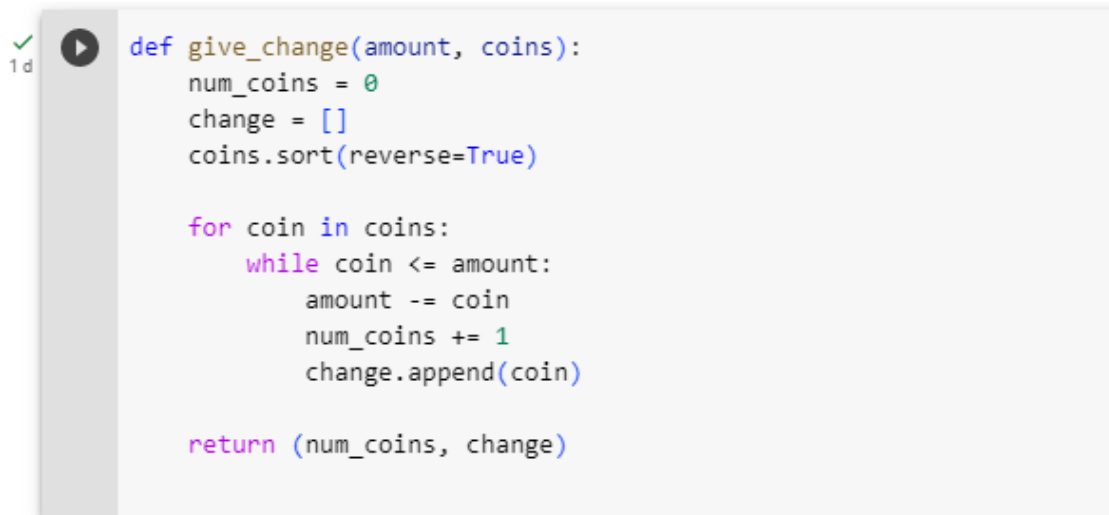
Parameter `amount` adalah jumlah uang yang harus diberikan kembali, sedangkan parameter `coins` adalah daftar koin yang tersedia untuk memberikan kembali. Pada baris ke-2, `num_coins` dan `change` diinisialisasi dengan nilai awal 0 dan daftar kosong. `num_coins` akan menyimpan jumlah koin yang digunakan untuk memberikan kembali uang, sedangkan `change` akan menyimpan daftar koin yang digunakan untuk memberikan kembali uang. Pada baris ke-3, koin-koin diurutkan secara menurun menggunakan metode `.sort(reverse=True)` untuk memudahkan algoritma memilih koin yang paling besar terlebih dahulu.

Berikut adalah contoh kode Python untuk algoritma greedy dalam memilih koin-koin untuk

memberikan kembalian uang dengan jumlah paling sedikit dari sejumlah uang yang diterima:

Kode yang diberikan tampaknya merupakan fungsi yang disebut `give_change` yang menghitung jumlah koin minimum yang diperlukan untuk melakukan perubahan untuk jumlah tertentu menggunakan sekumpulan koin tertentu. Ini mengembalikan jumlah koin yang digunakan dan daftar koin yang digunakan untuk melakukan perubahan.

Mari kita pecahkan kodenya:



```

def give_change(amount, coins):
    num_coins = 0
    change = []
    coins.sort(reverse=True)


    for coin in coins:
        while coin <= amount:
            amount -= coin
            num_coins += 1
            change.append(coin)

    return (num_coins, change)

```

- Fungsi ini mengambil dua parameter: jumlah (jumlah total yang perlu diberikan kembalian) dan koin (daftar denominasi koin yang tersedia).
- Ini menginisialisasi num\_coins ke 0, yang akan digunakan untuk melacak jumlah koin yang digunakan.
- Ini menginisialisasi perubahan daftar kosong untuk menyimpan koin yang digunakan untuk melakukan perubahan.
- Koin daftar diurutkan dalam urutan menurun menggunakan metode pengurutan. Ini dilakukan untuk memastikan bahwa denominasi yang lebih besar dipertimbangkan terlebih dahulu untuk perubahan yang optimal.
- Kode kemudian beralih melalui setiap koin dalam daftar koin yang diurutkan.
- Di dalam loop, while loop digunakan untuk berulang kali mengurangi jumlah koin saat ini selama koin kurang dari atau sama dengan jumlah yang tersisa.
- Untuk setiap pengurangan, penghitung num\_coins bertambah 1, dan koin ditambahkan ke daftar perubahan.
- Terakhir, fungsi mengembalikan tuple (num\_coins, change) yang mewakili jumlah total koin yang digunakan dan daftar koin yang digunakan untuk melakukan perubahan.

Namun, ada masalah dengan lekukan pada kode yang diberikan. Perulangan while tidak diindentasi dengan benar. Untuk memperbaikinya, Anda harus membuat indentasi blok kode di dalam while loop. Berikut kode yang diperbaiki:



```
def give_change(amount, coins):  
    num_coins = 0  
    change = []  
    coins.sort(reverse=True)  
  
    for coin in coins:  
        while coin <= amount:  
            amount -= coin  
            num_coins += 1  
            change.append(coin)  
  
    return (num_coins, change)
```

Sekarang fungsinya harus bekerja dengan benar untuk menghitung jumlah koin minimum yang diperlukan untuk melakukan perubahan untuk jumlah tertentu menggunakan denominasi koin yang disediakan.