

Higher-Order Function 'reduce' untuk Menghitung Total Belanjaan

Asa Do'a Uyi (122450005), Try Yani Rizki Nur Rohmah (122450020), Marleta Cornelia Leander (122450092), Sofyan Fauzi Dzaki Arif (122450116), Amalia Melani Putri (122450122)

1. Pendahuluan

Higher-Order Function (HOF) merupakan fungsi yang beroperasi di dalam fungsi lain dan menggunakan berbagai argumen untuk mengembalikan nilainya. Salah satu pengembangan metode ini adalah fungsi *reduce* yang memiliki fungsi tingkat tinggi untuk mengurangi daftar nilai menjadi suatu nilai tunggal yang berulang kali serta menerapkan fungsi biner ke daftar nilai yang akan dibutuhkan. Pada laporan ini, kami menerapkan fungsi *reduce* kedalam *Higher-Order Function* dengan menerapkan perhitungan total belanjaan.

2. Metode

Terdapat beberapa metode yang digunakan dalam *Higher-Order Function (HOF)* dan pada metode kali ini yang digunakan adalah metode *reduce* dan *lambda* penjelasannya sebagai berikut:

2.1. Reduce:


Dalam menggunakan fungsi dari *reduce* ada beberapa cara yang harus digunakan, dan metode ini digunakan adalah *functools* yang ada di dalam fungsi python itu sendiri, penggunaanya sendiri menerapkan fungsi tertentu secara berulang pada dua elemen sekaligus dari sebuah iterable, dan menghubungkannya kembali menjadi satu nilai. Python Documentation - `functools.reduce`

2.2. Lambda:

Dan dalam *lambda* mendefinisikan suatu fungsi dengan menggunakan *lambda* secara singkat tanpa menggunakan fungsi *def* untuk mendefinisikan digunakan sekali dalam konteks yang sangat spesifik dan dalam penggunaan *reduce* biasanya kita juga menggunakan fungsi dari *lambda* itu sendiri dalam kasus penerapan fungsi-fungsi dari higher order function. Python Documentation - *Lambda Expressions*

3. Pembahasan

3.1 Mengimpor fungsi 'reduce' dari modul 'functools'

```
 from functools import reduce
```

Fungsi reduce akan digunakan nanti dalam program untuk menjumlahkan semua elemen dalam list.

3.2 Mendefinisikan fungsi 'hitung_total_belanjaan'

```
def hitung_total_belanjaan(belanjaan):  
    return reduce(lambda x, y: x + y, belanjaan)
```

Fungsi ini akan menerima sebuah list belanjaan dan mengembalikan hasil penjumlahan semua elemen di dalamnya menggunakan fungsi reduce. Fungsi reduce akan menjalankan fungsi lambda yang menjumlahkan dua elemen secara berurutan hingga hanya menyisakan satu nilai.

3.3 Meminta input jumlah item belanjaan dan harga

```
# Meminta input dari pengguna untuk daftar belanjaan  
belanjaan = []  
jumlah_item = int(input("Masukkan jumlah item belanjaan: "))  
  
for i in range(jumlah_item):  
    belanjaan.append(float(input(f"Masukkan harga item ke-{i+1}: ")))
```

Program meminta pengguna untuk memasukkan jumlah item belanjaan, kemudian menggunakan loop for untuk meminta harga dari setiap item dan menambahkannya ke dalam list belanjaan.

3.4 Menghitung total belanjaan

```
total_belanjaan = hitung_total_belanjaan(belanjaan)  
print("Total belanjaan:", total_belanjaan)
```

Program menggunakan fungsi **hitung_total_belanjaan** untuk menghitung total belanjaan dari list belanjaan yang telah dimasukkan oleh pengguna. Hasil total belanjaan kemudian dicetak ke layar.

Output Code :

Contoh pemakaian dengan menginput jumlah item sebesar 5, harga item ke-1 sebesar Rp20.000, harga item ke-2 sebesar Rp30.000, harga item ke-3 sebesar Rp10.000, harga item ke-4 sebesar Rp4.000, dan harga item ke-5 sebesar Rp5.000

```
Masukkan jumlah item belanjaan: 5
Masukkan harga item ke-1: 20000
Masukkan harga item ke-2: 30000
Masukkan harga item ke-3: 10000
Masukkan harga item ke-4: 4000
Masukkan harga item ke-5: 5000
Total belanjaan: 69000.0
```

4. Kesimpulan

Dari hasil yang telah dapat disimpulkan bahwa membuat program python dengan Higher-order Function yang menggunakan metode reduce dapat membantu dan memberikan kemudahan kita dalam menghitung total belanja untuk melakukan perhitungan dengan cepat dan akurat.

Link Colab: <https://colab.research.google.com/drive/1tC6U14dNMX-dJREMLmKhsTaudgua92Mw?usp=sharing>

Daftar Pustaka

[1] Kohli, Pushmeet, M. Pawan Kumar, and Philip HS Torr. "p³ & beyond: Move making algorithms for solving higher order functions." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.9 (2008): 1645-1656.