

# Implementasi *Higher Order Function* pada Sistem Validasi Data Positif

Arafi Ramadhan Maulana (122450002)<sup>1)</sup>, Dwi Ratna Anggraeni (122450008)<sup>2)</sup>, Raid Muhammad Naufal (122450027)<sup>3)</sup>, Rayan Koemi Karuby (122450038)<sup>4)</sup>, Muhammad Deriansyah Okutra (122450101)<sup>5)</sup>

Program Studi Sains Data, Fakultas Sains, Institut Teknologi Sumatera

Email :

[arafi.122450002@student.itera.ac.id](mailto:arafi.122450002@student.itera.ac.id)<sup>1</sup>, [dwi.122450008@student.itera.ac.id](mailto:dwi.122450008@student.itera.ac.id)<sup>2</sup>,  
[raid.122450027@student.itera.ac.id](mailto:raid.122450027@student.itera.ac.id)<sup>3</sup>, [rayan.122450038@student.itera.ac.id](mailto:rayan.122450038@student.itera.ac.id)<sup>4</sup>,  
[mderiansyah.122450101@student.itera.ac.id](mailto:mderiansyah.122450101@student.itera.ac.id)<sup>5</sup>

## 1. Pendahuluan

Validasi data adalah sebuah proses pemeriksaan data yang memenuhi syarat dengan membandingkan dengan beberapa aturan, skema, atau batasan yang sebelumnya ditetapkan untuk tiap atribut. Dalam beberapa kasus, kita perlu memastikan data yang digunakan sesuai dengan kriteria tertentu untuk mencegah kesalah-kesalahan yang tidak terduga agar dapat lanjut ke tahapan berikutnya. Pada pembahasan berikut ini, kita melakukan validasi terhadap data positif yang memastikan bahwa semua nilai dalam dataset memiliki nilai lebih besar dari nol (Juhari, 2021).

Dalam melakukan validasi terhadap data positif kami menggunakan penerapan *Higher-Order Functions* (HOF). HOF merupakan fungsi yang menerima fungsi lain sebagai parameternya dan mengembalikan fungsi lain sebagai hasilnya. Implementasi HOF ini menggunakan bahasa pemrograman *Python* memungkinkan kita dapat secara efisien memfilter data sesuai dengan kriteria yang ditentukan (Priambodo, 2017).

Hal pertama yang dilakukan adalah menghasilkan dataset dengan menggunakan fungsi `random.randint` untuk menghasilkan angka secara acak dalam rentang yang ditentukan. kemudian mendefinisikan fungsi *higher-order* yang digunakan sebagai argumen untuk fungsi filter. Fungsi tersebut dapat memfilter sebuah iterable sesuai kondisi yang ditentukan. Kemudian menganalisis dimensi dataset, penganalisisan dimensi dilakukan pada *dataset* sebelum dan sesudah validasi dilakukan. Hal tersebut akan memberikan pemahaman yang baik mengenai efektivitas validasi data positif dengan HOF.

## 2. Metode

### 2.1. *Random*

*Random* merupakan salah satu modul dalam *python* yang mempunyai fitur untuk menyediakan bilangan acak untuk sebuah program. *Random* dapat digunakan dengan cara melakukan *import* modul. *Random* dapat menambah fungsionalitas pembangkitan bilangan acak ke kode *python* pengguna.

#### 2.1.1. *randint()*

*randint()* adalah sebuah fungsi dalam modul *Random* yang dapat menghasilkan bilangan bulat acak di antara dua bilangan yang sudah didefinisikan oleh pengguna, dan biasa disimbolkan dengan 'a' dan 'b'. Dengan kata lain hasilnya berada di antara 'a' dan 'b' dan kedua ujungnya bisa termasuk ke dalam hasil pengacakan angka.

## 2.2. angka\_positif()

angka\_positif() menjadi fungsi *user defined* dalam program ini. Fungsi tersebut menggunakan ekspresi *lambda* untuk menambah efisiensi pada kode program. angka\_positif() memiliki aturan bahwa jika 'x' lebih besar dari 0, maka angka 'x' akan dikembalikan, dan jika sebaliknya maka None.

## 2.3. range()

range() merupakan fungsi *built-in python* yang memiliki fitur untuk membuat urutan bilangan bulat secara dinamis dengan rentangnya. range() dapat digunakan dalam 'for' loop, pembuatan *list*, dan konteks lain.

## 2.4. len()

len() berguna untuk mengetahui berapa banyak elemen yang ada di dalam *string*, *list*, dan *tuple*, terutama saat pengguna menggunakan data yang dinamis (berubah-ubah). len() mengembalikan nilai jumlah elemen dengan cara memasukkan nama urutan yang digunakan.

## 2.5. list()

list() digunakan ketika pengguna perlu mendefinisikan suatu daftar untuk elemen yang dapat diubah-ubah (*mutable*) sehingga tetap masih bisa untuk menambah, menghapus, atau mengubah elemennya. List yang sudah didefinisikan menjadi suatu fungsi yang hampir selalu digunakan dalam setiap kode.

## 2.6. filter()

filter() adalah salah satu fungsi *built-in*, yang berfungsi untuk menyaring elemen dari suatu *sequence* dengan suatu persyaratan tertentu. filter() menerima dua argumen yaitu fungsi yang digunakan sebagai persyaratan/kriteria dan *list*/urutan yang akan disaring.

## 3. Pembahasan

Pada aplikasi ini, kami menggunakan data bilangan acak yang telah dibangkitkan dengan menggunakan bantuan modul *Random* dengan data berbentuk *list* dan tipe data *integer* dalam rentang -50 sampai 50.

```
[1] import random
```

Gambar 1. Import Library

Pada Gambar 1. *library* yang digunakan aplikasi ini adalah *library random*. *Random* digunakan untuk membangkitkan bilangan acak *list* data yang akan digunakan.

```
[2] angka_positif = lambda x: x if x > 0 else None
```

Gambar 2. Fungsi Angka Positif

Selanjutnya, buatlah fungsi untuk mengecek nilai pada *list* data apakah bernilai lebih dari nol (positif) atau kurang dari sama dengan nol (negatif). Fungsi "angka\_positif" menerima parameter berupa x atau nilai setiap elemen pada *list* dan akan mengembalikan x atau nilai setiap elemen pada *list* jika memenuhi kondisi (positif) dan tidak mengembalikan nilai apapun jika tidak memenuhi kondisi (negatif).

```
[3] data = [random.randint(-50, 50) for i in range(50)]
```

**Gambar 3.** Membangkitkan *List* Data dengan Bilangan Acak

Setelah membuat fungsi “angka\_positif”, langkah selanjutnya adalah membangkitkan *list* data dengan bilangan acak dari -50 sampai 50 sebanyak 50 elemen pada *list* data dengan menggunakan bantuan modul *random*.

```
[4] print(data)
```

```
[5, -6, 19, -33, 30, -26, 12, 17, -21, 1, -42, -50, -12, -26, -37, -2, 42, 8, 41,
```

**Gambar 4.** *List* Data

Pada Gambar 4. tersebut menampilkan elemen-elemen pada *list* data yang telah dibangkitkan dengan menggunakan perintah “*print(data)*”.

```
[5] len(data)
```

```
50
```

**Gambar 5.** Panjang *List* Data

Selanjutnya, untuk menampilkan panjang *list* data yang telah dibangkitkan dapat dengan menggunakan perintah “*len(data)*”. Pada Gambar 5. panjang *list* data yang telah dibangkitkan adalah 50 elemen data.

```
[6] data_valid = list(filter(angka_positif, data))
```

**Gambar 6.** Penerapan Validasi Data Positif dengan HOF

Setelah itu, langkah yang dilakukan adalah menginisialisasi *list* baru sebagai “*data\_valid*” dengan menggunakan fungsi *list()* dan kemudian memfilter data yang telah dibangkitkan menggunakan fungsi *filter()* dengan argumen berupa fungsi “angka\_positif” yang merupakan fungsi untuk mengecek apakah elemen bernilai positif atau negatif. Pada perintah di atas, elemen-elemen yang memenuhi kondisi (positif) maka akan tersimpan di dalam *list* “*data\_valid*”.

```
[7] print(data_valid)
```

```
[5, 19, 30, 12, 17, 1, 42, 8, 41, 31, 11, 8, 42, 9, 6, 36, 48, 22, 27, 28]
```

**Gambar 7.** *List* Data setelah Divalidasi

Pada Gambar 7. tersebut menampilkan elemen-elemen pada *list* “*data\_valid*” yang telah divalidasi dengan menggunakan perintah “*print(data\_valid)*”.

```
[8] len(data_valid)
```

```
20
```

**Gambar 8.** Panjang *List* Data setelah Divalidasi

Selanjutnya, untuk menampilkan panjang *list* “*data\_valid*” dapat dengan menggunakan perintah “*len(data\_valid)*”. Pada Gambar 8. panjang *list* “*data\_valid*” adalah 20 elemen data. Dengan kata

lain, dari 50 elemen data yang telah dibangkitkan sebelumnya, terdapat 30 elemen data dengan nilai negatif atau nol dan 20 elemen data dengan nilai positif.

#### 4. Kesimpulan

Dari aplikasi yang disediakan menghasilkan 50 bilangan acak dalam rentang -50 sampai 50, lalu memfilter hanya nilai dengan bilangan positif yang disertakan dalam *list* “*data\_valid*”, dengan demikian dari 50 bilangan acak yang terdiri dari nilai dengan bilangan negatif dan positif hanya terdiri dari 20 nilai dengan bilangan positif pada bilangan acak kali ini. Untuk bilangan yang dihasilkan dari *random.randint()* selalu bilangan acak, dapat dikatakan setiap kali kita menjalankan kodenya maka akan menghasilkan output yang berbeda-beda. Fungsi *filter* ini biasanya digunakan untuk fokus pada analisis bilangan positif, yang relevan dalam berbagai analisis data, oleh karena itu memfilter bilangan positif ini adalah langkah penting dalam mempersiapkan analisis lebih lanjut.

#### 5. Daftar Pustaka

- Juhari, T. (2021, October 24). *Validasi Tipedata Pada Python*. <https://hirazone.medium.com/>. Retrieved May 1, 2024, from <https://hirazone.medium.com/validasi-tipedata-pada-python-626488bcfdd3>
- Priambodo, B. (2017, July 20). *Higher-order Function — Paradigma Fungsional Praktis, Part 4* | by Bobby Priambodo | *Paradigma Fungsional*. Medium.com. Retrieved May 1, 2024, from <https://medium.com/paradigma-fungsional/higher-order-function-paradigma-fungsional-praktis-part-4-c836bd23a82>