

# PEMFILTERAN DATA TEKS DENGAN HIGHER ORDER FUNCTION

Febiya Jomy Pratiwi, Alvia Asrinda Br.Ginting  
Farrel Julio Akbar, Akmal Faiz Abdillah  
Daffa Ahmad Naufal

Program Studi Sains Data Institut Teknologi Sumatera  
Jl. Terusan Ryacudu, Way Huwi, Kec. Jatiagung, Kabupaten Lampung Selatan  
Lampung 35365

Email:

[febiya.122450074@student.itera.ac.id](mailto:febiya.122450074@student.itera.ac.id), [alvia.122450077@student.itera.ac.id](mailto:alvia.122450077@student.itera.ac.id)  
[farrel.122450110@student.itera.ac.id](mailto:farrel.122450110@student.itera.ac.id), [akmal.122450114@student.itera.ac.id](mailto:akmal.122450114@student.itera.ac.id)  
[daffa.122450137@student.itera.ac.id](mailto:daffa.122450137@student.itera.ac.id)

## I. Pendahuluan

### A. Latar Belakang

Proses pemfilteran data teks adalah bagian penting dari pengolahan data teks, yang diperlukan dalam berbagai analisis teks dan dalam memproses bahasa alami. Lebih sering daripada tidak, pada tahap pemuatan data, kita ingin menyaring data sesuai dengan kriteria tertentu, seperti data yang memiliki kata kunci, kata dengan panjang tertentu, atau karakteristik lain. pemfilteran sering digunakan untuk membersihkan data set sebelum mengirimkannya ke lebih banyak analisis atau hanya untuk membersihkan noise. Terdapat berbagai metode sofistikasi pemfilteran data text. Salah satu pendekatan untuk mengimplementasi proses ini adalah menggunakan higher-order functions. Higher-order functions dalam pemrograman fungsional memungkinkan kami untuk menyimpan fungsi dalam variabel sebagai argumen dan nilai balik. Dengan cara ini, kita dapat membuat alturasi menjadi lebih fleksibel dan mudah terbaca.

Pada konteks pemrograman Python, fungsi `filter()` adalah contoh lain dari higher-order function yang sangat berguna untuk melakukan pemfilteran data. Fungsi ini memungkinkan kita untuk menerapkan fungsi `filter` ke setiap elemen dalam suatu iterable, misalnya list, yang berarti bahwa hanya elemen-elemen di mana fungsi `filter()` bernilai `True` yang akan disertakan dalam output. Oleh karena itu, dengan menggabungkan memberikan fungsi `filter()` dengan fungsi-fungsi pengujian kriteria higher-order, kita spin the wheel proses yang sangat elegant dan efektif untuk merujuk pemfilteran data teks.

Dalam artikel ini, kami akan menjelaskan konsep dan implementasi dari pemfilteran data teks menggunakan higher-order function, dengan fokus pada

penggunaan fungsi `filter()` dalam bahasa pemrograman Python. Kami akan memperkenalkan metode untuk mendefinisikan kriteria pemfilteran dalam bentuk fungsi-fungsi yang bersifat higher-order, serta menunjukkan bagaimana menggunakan fungsi `filter()` untuk menerapkan kriteria-kriteria ini pada data teks yang diberikan. Dengan demikian, pembaca akan mendapatkan pemahaman yang lebih baik tentang konsep ini dan bagaimana mengimplementasikannya dalam praktik pemrograman sehari-hari.

## B. Tujuan

Adapun tujuan dari laporan ini adalah sebagai berikut, yaitu:

1. Memahami konsep pemfilteran data teks
2. Mengenal penggunaan Higher Order Functions
3. Mengimplementasikan pemfilteran data teks

## II. Metode

### A. Fungsi `Len()`

Fungsi `len()` memerlukan satu parameter utama yang berupa urutan atau koleksi, yang dapat berisi bytes, string, list, tuple, range, dictionary, set, dan frozenset. Fungsi `len()` adalah fungsi yang tersedia di Python yang berguna untuk menghitung jumlah karakter dalam parameter utama. [1]

### B. List

List adalah tipe data kolektif yang digunakan untuk menyimpan berbagai jenis data dalam satu variabel. List memiliki karakteristik utama yaitu memiliki tanda kurung siku (`[]`) dan bersifat terurut serta dapat diubah nilainya. List juga memiliki index yang memungkinkan kita untuk mengakses dan memanipulasi data di dalamnya. Selain itu, list juga merupakan salah satu dari empat jenis tipe data bawaan (built-in) dalam bahasa pemrograman Python, selain tuples, set, dan dictionaries. [2]

### C. Fungsi `Filter()`

Fungsi `Filter()` digunakan untuk menyaring data berdasarkan inisial nama kolom. Fungsi ini memungkinkan pengguna untuk mengembalikan rentang data yang memenuhi tinggi atau lebar yang sama seperti larik sumber. Fungsi `filter()` dalam Python mengambil fungsi dan daftar sebagai argumen, memberikan cara yang elegan untuk menyaring elemen dari urutan "order" yang mengembalikan nilai true. [3]

#### D. Fungsi List()

fungsi `list()` digunakan untuk membuat daftar yang berisi berbagai tipe data seperti string, integer, float, boolean, dll. Dengan `list()`, Anda dapat mengelola dan mengakses data-data tersebut dalam satu struktur data terorganisir. Misalnya, Anda dapat membuat daftar yang berisi nama, umur, dan status pekerjaan seseorang dalam satu list. Ini memudahkan akses dan manipulasi data sesuai kebutuhan Anda [4]

### III. Pembahasan

#### A. Analisis Program

Penggunaan fungsi dalam HOF (Higher Order Function) memungkinkan pemilihan elemen data teks secara selektif yang memenuhi kriteria tertentu, yang disebut dengan pemfilteran konten teks. Dalam kasus kode yang diberikan, penggunaan fungsi `filter()` adalah contoh penggunaan HOF.

Dengan menggunakan HOF, tindakan penerapan fungsi pada data teks setiap elemen memungkinkan pemilihan elemen yang memenuhi kriteria tertentu, yang pada gilirannya meningkatkan abstraksi dan fleksibilitas kode.

```
1 # Fungsi untuk memeriksa apakah sebuah kata memiliki panjang lebih dari 5 karakter
2 def panjang_lebih_dari_5(kata):
3     return len(kata) > 5
4
5 # Data teks yang akan difilter
6 data_teks = ["apel", "jeruk", "nanas", "semangka", "pisang", "stroberi"]
7
8 # Menggunakan fungsi filter() untuk memfilter data teks berdasarkan kriteria yang didefinisikan
9 hasil_filter = filter(panjang_lebih_dari_5, data_teks)
10
11 # Mengonversi hasil filter menjadi list untuk kemudahan pembacaan
12 hasil_filter_list = list(hasil_filter)
13
14 # Menampilkan hasil filter
15 print(hasil_filter_list)
16
```

### 1. Fungsi panjang\_lebih\_dari5(kata)

Fungsi Boolean digunakan untuk menentukan panjang sebuah kata, dan mengembalikan `panjang_lebih_dari_5(kata)` dengan 0 untuk setiap argumen (True jika 5 karakter atau False jika tidak). Pemfilteran data teks dimungkinkan dengan penggunaan fungsi perbandingan (fungsi predikat).

### 2. Data teks

Variabel `data_text` adalah daftar yang berisi banyak kata.

### 3. Penggunaan fungsi filter()

Berikut cara menggunakan fungsi `filter()`. Fungsi ini memfilter elemen `text_data` dengan menerapkan fungsi `panjang_lebih_dari_5()` ke setiap elemen. Hasilnya adalah sebuah iterator berisi elemen yang memenuhi kriteria memiliki lebih dari 5 karakter.

### 4. Konversikan output filter ke format daftar.

Hasil filter diubah menjadi daftar dengan baris ini sebagai iterator. Untuk memudahkan membaca dan menampilkan hasil penyaringan, hal ini dilakukan dengan sangat mudah.

### 5. Tampilkan hasil filter.

Ini menghasilkan daftar kata yang lebih dari 5 karakter pada hasil filter. Dengan mencetak keluaran filter sebagai daftar kata yang lebih dari 5 karakter.

Memfilter data teks menggunakan `filter()` dan fungsi yang ditentukan (`panjang_lebih_dari_5()`) dapat dilakukan dengan mempertimbangkan beberapa faktor, termasuk panjang kata. HOF digunakan untuk membedakan antara logika filter dan implementasi utama fungsi, misalnya dengan menggunakan fungsi dengan argumen arbitrer (seperti `panjang_lebih_dari_5()`), sehingga meningkatkan kejelasan dan ketidakberpihakan modul kodenya.

#### IV. Kesimpulan

HOF (Higher Order Function) fungsi memungkinkan pemilihan elemen data teks secara selektif yang memenuhi kriteria tertentu, yang disebut dengan pemfilteran konten teks. Fungsi `filter()` adalah contoh penggunaan HOF, tindakan penerapan fungsi pada data teks setiap elemen memilih elemen yang memenuhi kriteria tertentu, yang pada gilirannya meningkatkan abstraksi dan fleksibilitas kode. Fungsi `panjang_lebih_dari5(kata)` digunakan untuk menentukan panjang sebuah kata dan mengembalikan `panjang_lebih_dari_5(kata)` dengan 0 untuk setiap argumen.

# References

- [1] A. Kadir, "Langkah Mudah Pemrograman OpenCV & Python," jakarta, Elex Media Komputindo, 2019, p. 56.
  
- [2] Kahlil, M. . R. Munggaran, L. Kurnianggoro, A. Mahendra, N. Zarima, F. Noviantika and . A. Febriana, "Computer Vison Berbasis Deep Learning untuk Aplikasi Pertanian: Teori dan Praktik," Banda Aceh, Syiah Kuala University Press, 2023, p. 23.
  
- [3] J. Surya and Efitra , "DASAR-DASAR PEMROGRAMAN DENGAN PYTHON," Jambi, PT. Sonpedia Publishing Indonesia, 2023, p. 119.
  
- [4] M. R. faisal, D. Kartini, A. . R. Arrahimi and T. . H. Saragih, "Belajar Data Science: Text Mining Untuk Pemula I," BanjarBaru, Scripta Cendekia, 2023, p. 47.