

PENERAPAN HIGHER ORDER FUNCTION DALAM PENGOLAHAN DATA BERUPA EKSPLORASI DATA

Ahmad Sahidin Akbar¹, Nathanael Daniel Santoso², Safitri³, Ferdy Kevin Naibaho⁴, Elilya Octaviani⁵

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

Email: ahmad.122450044@student.itera.ac.id, nathanael.122450059@student.itera.ac.id,
safitri.122450071@student.itera.ac.id, ferdy.122450107@student.itera.ac.id, elilya.122450009@student.itera.ac.id

ABSTRAK

Penerapan *higher order function* pada pemrosesan data melalui eksplorasi data menggunakan proyek penelitian yang bertujuan untuk memperdalam pemahaman konsep *higher order function* dalam pemrosesan data. Eksplorasi data adalah langkah penting dalam analisis data yang memungkinkan untuk memperoleh pemahaman mendalam tentang struktur dan pola kumpulan data. Fungsi *higher order function* adalah sebuah konsep dalam pemrograman fungsional yang memungkinkan untuk memperlakukan fungsi sebagai nilai, termasuk kemampuan untuk menerima fungsi sebagai argumen dan mengembalikan fungsi sebagai hasil. Dalam konteks pemrosesan data, penerapan *higher order function* memberikan manfaat yang signifikan dengan mempercepat proses analisis, meningkatkan fleksibilitas, dan mengurangi *redudansi* kode.

Kata kunci : eksplorasi data, fleksibilitas, *higher order function*

PENDAHULUAN

Pemrosesan data adalah bagian penting dari banyak bidang seperti ilmu komputer, sains, dan teknik. Eksplorasi data merupakan langkah penting dalam proses pemrosesan data untuk memahami properti data, mengidentifikasi pola, dan menemukan anomali.

Di sisi lain, konsep *higher order function* dalam pemrograman fungsional telah mendapat perhatian dalam beberapa tahun terakhir, terutama dengan semakin populernya bahasa pemrograman seperti *haskell*, *scala*, dan *closure*.

Melalui eksplorasi data yang efektif, dapat membuat keputusan yang lebih tepat

berdasarkan informasi yang terdapat dalam data. Meskipun potensi penerapan fungsi tingkat tinggi dalam pemrosesan data telah diketahui, penelitian yang secara khusus membahas penerapan fungsi tingkat tinggi dalam eksplorasi data masih terbatas.

Oleh karena itu, tujuan penelitian ini adalah untuk mengisi kesenjangan ini dengan mengeksplorasi bagaimana fungsi *higher order function* dapat digunakan secara efektif dalam eksplorasi data. Melalui analisis mendetail dan studi kasus, penelitian ini diharapkan dapat memberikan wawasan baru mengenai kemungkinan dan manfaat penggunaan fungsi tingkat tinggi

untuk memfasilitasi eksplorasi data yang lebih efisien dan efektif.

METODE

1. Eksplorasi Data

Eksplorasi data merupakan salah satu cara untuk menganalisis suatu data dengan melakukan sebuah pendekatan dan membuat gambaran keseluruhan untuk memperoleh informasi dari sebuah data (Yunita, 2021). Biasanya data akan divisualisasikan berupa boxplot, scatterplot dan lain sebagainya tergantung jenis data.

2. Matplotlib

Matplotlib merupakan sebuah *library* dalam python yang komprehensif yang digunakan untuk membuat visualisasi data agar memudahkan sebuah individu dalam mengambil informasi penting penting dari sebuah data (Pintar , 2020).

3. Pandas

Pandas merupakan sebuah *library* dalam menganalisis data yang digunakan untuk membersihkan data yang akan digunakan dengan menyelaraskan data yang disebabkan oleh hilangnya sebuah nilai dalam data (Mutmainnah, 2019).

4. Reduce

Reduce merupakan salah satu fungsi dalam python yang digunakan untuk menggabungkan elemen dan nantinya dieksekusi menggunakan iterable menjadi satu nilai akhir (Nosuke, 2023).

5. Filter

Filter merupakan fungsi dalam python yang digunakan untuk menyaring elemen dari iterable dengan kondisi yang ditentukan. Fungsi ini mengembalikan elemen yang hanya memenuhi kriteria dari kondisi yang sudah ditentukan (Nosuke, 2023).

HASIL DAN PEMBAHASAN

```
import pandas as pd
import matplotlib.pyplot as plt
from functools import reduce
```

Pertama kita mengimport *library-library* yang diperlukan yaitu pandas, matplotlib dan fungsi *functools* dari *library reduce*.

```
# Baca dataset dari Kaggle
data =
pd.read_csv('Salary_Data.csv') #
Sesuaikan dengan lokasi file
dataset Anda
```

Selanjutnya kita akan membaca *dataset* (Salary_Data.csv) dengan fungsi pandas *read_csv*.

```
# Eksplorasi data
print("Informasi Ringkas tentang
Dataset:")
print(data.info())
print("\nStatistik Deskriptif:")
print(data.describe())
```

Kode selanjutnya kita akan melakukan eksplorasi data dengan mencetak informasi ringkas tentang dataset, menampilkan informasi mengenai dataset menggunakan metode *info()* dari Pandas *DataFrame*, dan menampilkan statistik deskriptif tentang dataset menggunakan metode *describe()* dari Pandas *DataFrame*.

```
# Menghitung gaji berdasarkan umur
```

```
def hitung_gaji(umur):
    return umur
```

Kode selanjutnya untuk mendefinisikan fungsi `hitung_gaji` yang sementara ini hanya mengembalikan nilai umur. Namun, fungsi ini bisa dimodifikasi untuk melakukan perhitungan gaji sesuai kebutuhan.

```
# Membuat kolom baru 'Gaji'
# berdasarkan umur dengan
# higher-order function
data['Gaji'] =
data['Age'].apply(hitung_gaji)
```

Selanjutnya kita akan menambahkan kolom baru 'Gaji' ke *DataFrame* `data` dengan mengaplikasikan fungsi `hitung_gaji` ke kolom 'Age'.

```
# Filter kolom yang diperlukan
filtered_data = data[['Age',
'Salary', 'YearsExperience']]
```

Selanjutnya kita membuat *DataFrame* baru yang hanya berisi kolom 'Age', 'Salary', dan 'YearsExperience'.

```
# Mendefinisikan fungsi lambda
# untuk menggabungkan kolom 'Age',
# 'YearsExperience', dan 'Salary'
# dalam satu string
gabung_data = lambda age, exp,
salary: f"Pengalaman kerja {exp}
dengan umur {age} mendapatkan gaji
{salary}"
```

Kode ini untuk mendefinisikan fungsi `lambda` `gabung_data` yang akan menggabungkan nilai dari kolom 'Age', 'YearsExperience', dan 'Salary' dalam satu string.

```
# Menggunakan reduce untuk
# menggabungkan nilai-nilai dari
# setiap baris
```

```
hasil_reduce = reduce(lambda a, b:
f"{a}\n{b}",
filtered_data.apply(lambda row:
gabung_data(row['Age'],
row['YearsExperience'],
row['Salary']), axis=1))
```

Selanjutnya kita menggunakan fungsi `reduce` untuk menggabungkan nilai-nilai dari setiap baris *DataFrame* `filtered_data` menjadi satu string.

```
# Cetak hasil reduce
print("\nHasil Pengolahan Data:")
print(hasil_reduce)
```

Kode selanjutnya untuk mencetak label untuk hasil pengolahan data dan mencetak hasil pengolahan data yang telah digabungkan.

```
# Scatterplot
plt.figure(figsize=(10, 6))
plt.scatter(data['YearsExperience'],
data['Age'], c=data['Salary'],
cmap='viridis', alpha=0.7)
plt.colorbar(label='Salary')
plt.xlabel('Years of Experience')
plt.ylabel('Age')
plt.title('Scatterplot of Salary
by Years of Experience and Age')
plt.grid(True)
plt.show()
```

Kode yang terakhir untuk melakukan visualisasi dengan membuat *scatter plot* dengan 'YearsExperience' sebagai sumbu x, 'Age' sebagai sumbu y, dan 'Salary' sebagai warna titik.

Untuk hasil outputnya didapat seperti berikut:

```
↳ Informasi Ringkas tentang Dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 3 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   YearsExperience  30 non-null    float64
1   Age              30 non-null    float64
2   Salary           30 non-null    int64
dtypes: float64(2), int64(1)
memory usage: 848.0 bytes
None
```

```
Statistik Deskriptif:
      YearsExperience      Age      Salary
count      30.000000    30.000000    30.000000
mean         5.313333    27.216667    76003.000000
std          2.837888     5.161267    27414.429785
min           1.100000    21.000000    37731.000000
25%           3.200000    23.300000    56720.750000
50%           4.700000    25.000000    65237.000000
75%           7.700000    30.750000    100544.750000
max          10.500000    38.000000    122391.000000
```

Hasil Pengolahan Data:

Pengalaman kerja 1.1 dengan umur 21.0 mendapatkan gaji 39343.0
Pengalaman kerja 1.3 dengan umur 21.5 mendapatkan gaji 46205.0
Pengalaman kerja 1.5 dengan umur 21.7 mendapatkan gaji 37731.0
Pengalaman kerja 2.0 dengan umur 22.0 mendapatkan gaji 43525.0
Pengalaman kerja 2.2 dengan umur 22.2 mendapatkan gaji 39891.0
Pengalaman kerja 2.9 dengan umur 23.0 mendapatkan gaji 56642.0
Pengalaman kerja 3.0 dengan umur 23.0 mendapatkan gaji 60150.0
Pengalaman kerja 3.2 dengan umur 23.3 mendapatkan gaji 54445.0
Pengalaman kerja 3.2 dengan umur 23.3 mendapatkan gaji 64445.0
Pengalaman kerja 3.7 dengan umur 23.6 mendapatkan gaji 57189.0
Pengalaman kerja 3.9 dengan umur 23.9 mendapatkan gaji 63218.0
Pengalaman kerja 4.0 dengan umur 24.0 mendapatkan gaji 55794.0
Pengalaman kerja 4.0 dengan umur 24.0 mendapatkan gaji 56957.0
Pengalaman kerja 4.1 dengan umur 24.0 mendapatkan gaji 57081.0
Pengalaman kerja 4.5 dengan umur 25.0 mendapatkan gaji 61111.0
Pengalaman kerja 4.9 dengan umur 25.0 mendapatkan gaji 67938.0
Pengalaman kerja 5.1 dengan umur 26.0 mendapatkan gaji 66029.0
Pengalaman kerja 5.3 dengan umur 27.0 mendapatkan gaji 83088.0
Pengalaman kerja 5.9 dengan umur 28.0 mendapatkan gaji 81363.0
Pengalaman kerja 6.0 dengan umur 29.0 mendapatkan gaji 93940.0
Pengalaman kerja 6.8 dengan umur 30.0 mendapatkan gaji 91738.0

Pengalaman kerja 7.1 dengan umur 30.0 mendapatkan gaji 98273.0
Pengalaman kerja 7.9 dengan umur 31.0 mendapatkan gaji 101302.0
Pengalaman kerja 8.2 dengan umur 32.0 mendapatkan gaji 113812.0
Pengalaman kerja 8.7 dengan umur 33.0 mendapatkan gaji 109431.0
Pengalaman kerja 9.0 dengan umur 34.0 mendapatkan gaji 105582.0
Pengalaman kerja 9.5 dengan umur 35.0 mendapatkan gaji 116969.0
Pengalaman kerja 9.6 dengan umur 36.0 mendapatkan gaji 112635.0
Pengalaman kerja 10.3 dengan umur 37.0 mendapatkan gaji 122391.0
Pengalaman kerja 10.5 dengan umur 38.0 mendapatkan gaji 121872.0



KESIMPULAN

Laporan ini menggunakan konsep fungsi yang beroperasi pada level yang lebih abstrak dalam pemrograman, yang disebut fungsi *higher-order*, dalam konteks eksplorasi data. Melalui penggunaan library seperti Pandas dan Matplotlib, serta penerapan fungsi *higher-order* seperti *'reduce'* dan *'filter'*, penelitian ini menunjukkan bagaimana pendekatan ini dapat diterapkan untuk menganalisis data dengan lebih efektif. Dengan memanfaatkan statistik deskriptif dan visualisasi grafis, laporan ini memberikan pandangan yang jelas tentang manfaat praktis dari penggunaan fungsi *higher-order* dalam proses eksplorasi data.

DAFTAR PUSTAKA

- Mutmainnah, I. (2019). *Mengenal Pandas Dalam Python*. Iin Mutmainnah. Retrieved April 29, 2024, from <https://medium.com/@16611092/mengenal-pandas-dalam-python-cc66d0c5ea40>
- Nosuke. (2023). *Tutorial Python 46: Belajar Python Map, Filter, Reduce*. Nosuke. <https://codekey.id/python/tutorial-python-46-belajar-python-map-filter-reduce/>

Pintar, G. (2020). *Mengenal Matplotlib Python dan Contoh Penggunaannya*. Gubuk Pintar. Retrieved April 29, 2024, from <https://www.gubukpintar.com/2020/11/matplotlib-adalah.html>

Yunita. (2021). *MENGENAL EXPLORATORY DATA ANALYSIS*. YUNITA. Retrieved April 29, 2024,

from <https://eksplorasidata.mipa.ugm.ac.id/2021/08/16/mengenal-exploratory-data-analysis/>