

# Aplikasi Pengolah Data Statistik dengan Paradigma Pemrograman Berbasis Fungsi

*Cyntia Kristina Sidauruk (122450023)<sup>1</sup>, Patricia Leondrea Diajeng Putri (122450050)<sup>2</sup>, Berliana Enda Putri (122450065)<sup>3</sup>, Feryadi Yulius (122450087)<sup>4</sup>, Naufal Fakhri (122450089)<sup>5</sup>*

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

*Email: <sup>1</sup>[cyntia.122450023@student.itera.ac.id](mailto:cyntia.122450023@student.itera.ac.id),  
<sup>2</sup>[patricia.122450050@student.itera.ac.id](mailto:patricia.122450050@student.itera.ac.id),  
<sup>3</sup>[berliana.122450065@student.itera.ac.id](mailto:berliana.122450065@student.itera.ac.id),  
<sup>4</sup>[feryadi.122450087@student.itera.ac.id](mailto:feryadi.122450087@student.itera.ac.id), <sup>5</sup>[naufal.122450089@student.itera.ac.id](mailto:naufal.122450089@student.itera.ac.id)*

## 1. PENDAHULUAN

Artikel ini bertujuan untuk menunjukkan penerapan paradigma pemrograman berbasis fungsi dalam bahasa pemrograman Python pada data dengan keluaran nilai statistika deskriptif, seperti:

- Mean (rata-rata)
- Median (nilai tengah)
- Modus (nilai terbanyak)
- Total (jumlah)
- Produk (perkalian)

Mean merupakan teknik penjelasan kelompok yang berdasarkan pada nilai rata-rata dari semua anggota kelompok tersebut. Rata-rata ini diperoleh dengan menjumlahkan seluruh data individu dalam kelompok, kemudian membaginya dengan jumlah individu yang terdapat dalam kelompok tersebut. Istilah yang sering digunakan untuk menyebut ukuran ini adalah "mean", yang dihitung dengan cara membagi jumlah nilai data dengan jumlah observasi.

Median adalah nilai yang terletak di posisi tengah setelah data diatur dari yang terkecil hingga yang terbesar. Istilah "rata-rata posisi" sering dipakai untuk menggambarkan median karena nilai ini dipengaruhi oleh urutan data. Karena median menempati posisi tengah dalam urutan data, 50% dari total data akan berada di bawah median, dan 50% sisanya akan berada di atasnya.

Modus adalah nilai yang memiliki frekuensi paling tinggi dalam kumpulan data. Biasanya, ukuran ini digunakan untuk mengetahui seberapa sering terjadinya suatu peristiwa. Modus cocok digunakan untuk data yang berskala nominal. Dalam data yang tidak dikelompokkan, modus dapat ditentukan dengan menghitung frekuensi masing-masing nilai pengamatan, kemudian mencari nilai pengamatan yang memiliki frekuensi observasi paling banyak.

Lambda adalah ekspresi untuk membuat fungsi. Lambda memungkinkan Anda membuat fungsi tanpa nama. Dengan Lambda, anda tidak perlu menggunakan `def` dan `return`. Lambda digunakan untuk membuat fungsi dengan ekspresi satu baris. Lambda dapat memiliki beberapa argumen atau parameter, tetapi hanya satu ekspresi atau isi. Lambda adalah fungsi anonim, jadi Anda dapat menggunakan nama apa pun yang Anda sukai.

## 2. METODE

Pada era digital ini, pengolahan data menjadi salah satu aspek yang sangat penting dalam berbagai bidang, termasuk statistik. Dalam artikel ini, akan membahas beberapa metode untuk membangun sebuah aplikasi pengolahan data statistik. Fungsi Python yang digunakan yaitu `pandas`, `functools` (`reduce`), fungsi `lambda`, dan juga fungsi `map`.

### 1. Pandas

Pandas merupakan salah satu *library* Python yang cukup populer di kalangan pemrograman. Pandas dikenal sebagai pustaka python yang sangat kuat untuk manipulasi serta menganalisis data. Pada aplikasi pengolahan data ini, Pandas digunakan agar mempermudah kita untuk mengimpor, membersihkan, serta merangkum data statistik. Dengan menggunakan Pandas sebagai metode dalam program aplikasi ini, memberikan kemudahan agar mampu membaca, menyaring serta mengolah data dengan mudah. Hal ini dikarenakan Pandas memungkinkan kita untuk mengakses, menggabungkan, serta memanipulasi data dengan cara yang intuitif.

Sebelum menggunakan Pandas dalam pemrograman, dilakukan `peng-import-in` terlebih dahulu, kemudian Pandas dapat bekerja. Dalam artikel ini, Pandas bekerja untuk membaca data dari file CSV menggunakan `'pd.read_csv (file_path)'` dan memanipulasi nya. Setelah itu, dengan menggunakan Pandas dapat mengambil kolom data tertentu menggunakan `'data [column_name]'`.

### 2. Functools (Reduce)

Functools merupakan modul Python yang menyediakan fungsi-fungsi tingkat tinggi untuk memanipulasi fungsi dan juga panggilan fungsi. Salah satu fungsi pada Functools adalah fungsi `'reduce'`. Fungsi `'reduce'` dapat diimplementasikan pada perhitungan matematika yaitu pelipatan atau reduksi. Fungsi `'reduce'` sendiri berguna untuk mengaplikasikan suatu fungsi kumulatif pada elemen-elemen iterable (objek yang dapat diubah). Sebelum menggunakan fungsi `'reduce'`, harus dilakukan `peng-import-an` terlebih dahulu. Kemudian, fungsi `'reduce'` dapat bekerja sesuai fungsinya.

Pada artikel ini, fungsi **'reduce'** digunakan untuk menghitung beberapa statistik dasar yang digunakan dalam artikel ini, seperti mean (**'hitung\_mean'**), median (**'hitung\_median'**), modus (**'hitung\_modus'**), total (**'hitung\_total'**), dan produk (**'hitung\_produk'**).

### 3. Fungsi Lambda

Fungsi lambda dikenal sebagai sebuah ekspresi untuk membuat fungsi. Fungsi lambda merupakan cara cepat untuk membuat fungsi anonim atau fungsi satu baris, sehingga kita dapat menggunakannya untuk operasi sederhana. Lambda bisa memiliki lebih dari satu argumen atau parameter dapat penggunaannya. Namun, lambda hanya bisa memiliki satu ekspresi atau isi.

Dalam artikel ini, fungsi lambda digunakan pada penggunaan fungsi **'reduce'** untuk menghitung mean, median, modus, total serta produk. Saat menghitung mean, fungsi lambda digunakan dalam fungsi **'reduce'** untuk menjumlahkan semua elemen dalam data. Pada menghitung median, fungsi lambda digunakan dalam fungsi **'reduce'** untuk menjumlahkan dua elemen pada perhitungan median. Pada menghitung modus, fungsi lambda digunakan dalam fungsi **'reduce'** untuk membandingkan dua kunci dalam *dictionary* frekuensi saat mencari nilai modus. Pada menghitung total dan produk, fungsi lambda digunakan dalam kedua fungsi untuk menjumlahkan dan mengalikan dua elemen dalam data.

### 4. Fungsi Map

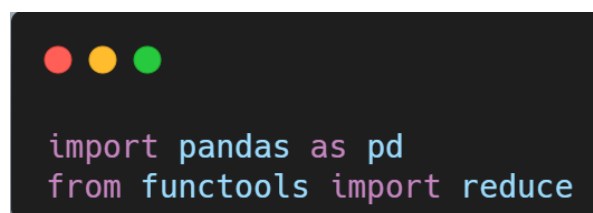
Fungsi map merupakan fungsi bawaan dalam Python. Fungsi map digunakan untuk menerapkan suatu fungsi ke setiap elemen dari suatu iterable seperti list ataupun series. Fungsi map tidak selalu diperlukan, terutama jika bekerja menggunakan DataFrame Pandas.

## 3. PEMBAHASAN

### 3.1 Dataset

Dataset yang digunakan dalam pembahasan kali ini diambil dari situs resmi kaggle bernama Walmart Dataset. Dataset berupa data dari 45 toko Walmart di Amerika Serikat selama tahun 2010-2011.

### 3.2 Import Package



```
import pandas as pd
from functools import reduce
```

Pandas merupakan library yang biasa digunakan untuk manipulasi data dalam python, Disini pandas dapat digunakan sebagai pd, yang memungkinkan kita untuk menggunakan pandas dengan lebih singkat setiap menggunakannya.

Functools merupakan library yang digunakan untuk membuat fungsi yang lebih efektif dan efisien. Kemudian akan diimport fungsi reduce dari library functools yang digunakan untuk menggabungkan dua atau lebih elemen dalam sebuah iterable hingga menjadi satu elemen.

### 3.4 Fungsi Menghitung Mean

```
def hitung_mean(data):  
    return reduce(lambda x, y: x + y, data) / len(data)
```

Dalam fungsi hitung\_mean, fungsi reduce dari library functools digunakan untuk menghitung rata-rata dari data yang telah di import yaitu data walmart\_sales. Fungsi ini menggabungkan dua elemen dalam iterable menjadi satu elemen dengan menggunakan fungsi lambda yang kita buat. Fungsi ini menerima parameter yaitu sebuah data.

Untuk menggabungkan dua elemen dalam iterable data dalam contoh di atas, kita menggunakan fungsi lambda  $\lambda x, y: x + y$  kemudian akan dibagi dengan panjang dari data. Fungsi reduce akan mengulangi iterable data dan menggunakan fungsi lambda untuk menggabungkan setiap dua elemen, hingga menjadi satu elemen yang diberikan sebagai output.

### 3.5 Fungsi Mencari Median

```
def hitung_median(data):  
    sorted_data = sorted(data)  
    n = len(sorted_data)  
  
    if n % 2 == 1:  
        return sorted_data[n // 2]  
    else:  
        return reduce(lambda x, y: x + y, sorted_data[n // 2 - 1:n // 2 + 1]) / 2
```

Fungsi dari def hitung\_median adalah untuk menghitung nilai median dari data walmart\_sales. sorted(data) akan membuat salinan dan akan mengurutkan data walmart\_sales kemudian akan disimpan pada variabel sorted\_data. n akan menyimpan panjang elemen dari data yang telah diurutkan sebelumnya.

Kode `if n % 2 == 1`, dibuat untuk mengecek apakah `len(panjang)` dari `sorted_data` adalah ganjil. Jika benar, maka nilai median merupakan nilai tengah dari data terurut dengan kode `return sorted_data[n // 2]`. Untuk mencari nilai tengah kita hanya membagi `n` dengan 2. Jika panjang data bukan ganjil melainkan genap, maka rata-rata dari nilai tengah merupakan nilai median. Dengan penggunaan `lambda` untuk menjumlahkan nilai tengah kemudian dibagi dengan 2 dan dengan memakai fungsi `reduce`, maka didapat hasil dari nilai tengah saat data genap.

### 3. 6 Fungsi Mencari Modus

```
def hitung_modus(data):  
    frequency = {}  
    for value in data:  
        if value in frequency:  
            frequency[value] += 1  
        else:  
            frequency[value] = 1  
  
    max_freq = reduce(lambda x, y: x if frequency[x] > frequency[y] else y, frequency.keys())  
  
    return max_freq
```

Fungsi dari `def hitung_modus` adalah untuk menghitung nilai modus dari data `walmart_sales`. `frequency` dengan kamus kosong yang artinya akan menyimpan frekuensi masing-masing nilai yang muncul dalam daftar `walmart_sales`. Kemudian akan mengiterasi setiap nilai dalam data `walmart_sales`. kemudian pada baris `if value in frequency:` disini akan memeriksa apakah nilai ada di dalam kamus `frequency` yang telah dibuat sebelumnya. Jika nilai ada, maka frekuensi dari nilai tersebut ditambahkan 1, jika nilai tidak ada maka, menambahkan value ke dalam `frequency` dan ditetapkan sebagai `frequency` awal 1.

Pada penggunaan fungsi `reduce` dan `lambda x, y : x if frequency[x] > frequency[y]`, ini akan membandingkan frekuensi dari kunci `x` dan `y` dengan frekuensi terbesar `x` maupun `y`, frekuensi yang paling tinggi di dalam kamus `frequency` sehingga nantinya akan menjadi nilai modus. Kode `return max_freq` akan mengembalikan nilai modus yang telah di dapat.

### 3. 8 Fungsi Menghitung Total

```
def hitung_total(data):  
    return reduce(lambda x, y: x + y, data)
```

menjumlahkan semua elemen yang terdapat pada data walmart\_sales. Hasil penjumlahan akan dikembalikan menjadi nilai total

### 3. 9 Fungsi Menghitung Produk

```
def hitung_produk(data):  
    return reduce(lambda x, y: x * y, data)
```

menjumlahkan semua elemen yang terdapat pada data walmart\_sales. Hasil penjumlahan akan dikembalikan menjadi nilai total

### 3. 10 Fungsi Menghitung Statistika Deskriptif

```
def hitung_statistik(file_path, column_name):
```

Fungsi untuk menghitung dan menampilkan data statistik dasar pada sebuah kolom dalam file csv

```
data = pd.read_csv(file_path)  
column_data = data[column_name]  
  
mean = hitung_mean(column_data)  
median = hitung_median(column_data)  
modus = hitung_modus(column_data)  
total = hitung_total(column_data)  
produk = hitung_produk(column_data)  
  
print(f"Mean: {mean},\nMedian: {median},\nModus: {modus},\nTotal: {total},\nProduk: {produk}\n")
```

Pada file\_path alamat file csv yang ingin kita baca, column\_name nama kolom pada file csv yang ingin kita analisa, selanjutnya kita menggunakan column\_data = data[column\_name] mengambil kolom tertentu dari data frame yang kita punya yang sesuai dengan column\_name, setelah kita mendefinisikan column\_data kita bisa mencari nilai mean = hitung\_mean(column\_data) menggunakan fungsi hitung\_mean untuk mean dengan menghitung nilai rata-rata pada column\_data, median = hitung\_median(column\_data) menggunakan fungsi hitung\_median untuk

menghitung nilai tengah pada column\_data. `modus = hitung_modus(column_data)` menggunakan fungsi `hitung_modus` untuk menghitung nilai terbanyak pada column\_data, `total = hitung_total(column_data)` menggunakan fungsi `hitung_total` untuk menghitung nilai total dari jumlah data pada column\_data, `produk = hitung_produk(column_data)` menggunakan fungsi `hitung_produk` untuk menghitung nilai produk yang didapat dengan perkalian data pada column\_data. Setelah kita menghitung nilai `mean`, `median`, `modus`, `total`, `produk`. Kita melakukan print untuk menampilkan hasilnya dengan menggunakan:

```
print(f"Mean: {mean},\nMedian: {median},\nModus: {modus},\nTotal: {total},\nProduk: {produk}\n")
```

### 3. 11 Pengaplikasian

```
file_path = "/content/Walmart_sales.csv"  
column_name = "Weekly_Sales"  
hitung_statistik(file_path, column_name)
```

```
file_path = "/content/Walmart_sales.csv"
```

Dalam bentuk csv menggunakan library pandas

```
column_name = "Weekly_Sales"
```

Pada `column_name` kita isi dengan nama kolom yang ingin kita cari nilai rata-ratanya

```
hitung_statistik(file_path, column_name)
```

Digunakan untuk menghitung dan menampilkan statistik dasar dari sebuah kolom pada file CSV yang kita punya

```
Mean: 1046964.8775617732,  
Median: 960746.04,  
Modus: 760281.43,  
Total: 6737218987.11001,  
Produk: inf
```

Disini Kita bisa lihat untuk mean nya kita dapat 1046964.87 hasil ini didapat dari penjumlahan semua data dibagi dengan banyaknya data. Hasil nilai dari nilai median 960746.04 didapat dari mencari nilai tengah dari data set yang kita gunakan. Hasil nilai Modus 760281.43 didapat dari nilai yang paling banyak muncul dari data set yang kita gunakan. Hasil nilai Total 6737218987.11001 didapat dari seluruh hasil total yang kita gunakan. Hasil ini kita dapat dari analisis perhitungan pada produk inf yang artinya tak terhingga.

## **4. KESIMPULAN**

Penggunaan paradigma pemrograman berbasis fungsi dalam pengolahan data statistik dengan bahasa pemrograman Python membawa sejumlah keuntungan. Dalam aplikasi ini, paradigma ini diterapkan dengan menggunakan library pandas, functools (reduce), fungsi lambda, dan fungsi map. Penerapan pada dataset Walmart Sales menunjukkan bahwa program dapat menghasilkan nilai statistika deskriptif dengan akurat. Hasil mean, median, modus, total, dan produk dapat diakses dengan mudah melalui fungsi yang telah diimplementasikan.

Dapat diambil kesimpulan kalau paradigma pemrograman berbasis fungsi dengan Python memberikan solusi yang efektif dan efisien dalam pengolahan data statistik. Program ini dapat diterapkan dalam berbagai konteks, termasuk eksplorasi data, analisis, dan penelitian. Dengan menggunakan beberapa fungsi dasar, kita dapat menghasilkan informasi yang bermanfaat dari data numerik dengan cepat dan akurat