

# IMPLEMENTASI FUNGSI PENGURUTAN MENGGUNAKAN MAP DAN LAMBDA DENGAN KRITERIA DARI YANG TERKECIL HINGGA YANG TERBESAR

Eksanty F Sugma Islamiaty<sup>1</sup>, Najla Juwairia<sup>2</sup>, Kharisa Harvanny<sup>3</sup>,  
Danang Hilal Kurniawan<sup>4</sup>, Mirzan Yusuf Rabbani<sup>5</sup>

Program Studi Sains Data, Fakultas Sains, Institut Teknologi Sumatera, Lampung Selatan,  
Indonesia

E-mail: [eksanty.122450001@student.itera.ac.id](mailto:eksanty.122450001@student.itera.ac.id)<sup>1</sup>, [najla.122450037@student.itera.ac.id](mailto:najla.122450037@student.itera.ac.id)<sup>2</sup>,  
[kharisa.122450061@student.itera.ac.id](mailto:kharisa.122450061@student.itera.ac.id)<sup>3</sup>, [danang.122450085@student.itera.ac.id](mailto:danang.122450085@student.itera.ac.id)<sup>4</sup>,  
[mirzan.122450118@student.itera.ac.id](mailto:mirzan.122450118@student.itera.ac.id)<sup>5</sup>

---

## Abstrak

Pembuatan aplikasi pengurutan data ini bertujuan untuk menyederhanakan proses pengurutan data dari nilai terkecil hingga nilai terbesar dari angka acak yang tersedia dengan menggunakan kode yang lebih ringkas dan mudah dipahami sehingga aplikasi ini dapat dikembangkan lebih lanjut. Metode yang digunakan adalah fungsi yang menggunakan map, lambda, dan fungsi pengurutan bawaan dari python (*sorted*) untuk proses pengurutan data. Fungsi tersebut digunakan agar program dapat berjalan secara efektif dan efisien. Output yang diberikan oleh program ini adalah tampilan data sebelum pengurutan dan tampilan data setelah pengurutan. Aplikasi ini dapat dikembangkan lebih lanjut untuk digunakan dalam mengelola berbagai bidang, seperti analisis kerja bisnis.

**Kata kunci:** Aplikasi, Fungsi, Map, Lambda, Pengurutan.

## PENDAHULUAN

Dalam pemrograman fungsional, konsepnya berpusat pada fungsi. Paradigma ini merujuk pada hubungan antara fungsi-fungsi dalam program. Dasar konsep ini bersumber dari matematika, khususnya dari gagasan tentang fungsi. Di dalam banyak bahasa pemrograman fungsional, ada penekanan pada fungsi murni, yaitu fungsi yang akan memberikan hasil yang sama persis ketika diberikan input yang sama. Dari pemrograman fungsional ini dapat dibuat sebuah aplikasi pengurutan data menggunakan Map dan Lambda.

Pengurutan merupakan proses dalam pemrograman untuk menata elemen-elemen suatu data secara teratur. Untuk bahasa pemrograman dapat digunakan fungsi 'sorted()' yang digunakan untuk mengurutkan list. Dalam mengimplementasikannya dapat digunakan fungsi 'map' dan fungsi 'lambda' untuk

mengurutkan list dari yang terkecil hingga terbesar. Pada fungsi lambda dapat digunakan untuk membandingkan elemen dalam list dan untuk fungsi map sendiri dapat digunakan untuk melakukan perbandingan dalam list pada setiap pasangan elemen secara berulang hingga dapat hasilnya yaitu list yang berurutan.

## METODE PENELITIAN

### 1. Lambda

Lambda adalah fungsi yang tidak memiliki nama. Lambda umumnya disimpan ke suatu variabel atau dieksekusi langsung.

### 2. Fungsi map

Fungsi map() (yang merupakan fungsi bawaan dalam Python) digunakan untuk menerapkan fungsi ke setiap item dalam iterable (seperti daftar atau kamus Python). Ini mengembalikan iterable

baru ( objek peta) yang dapat Kita gunakan di bagian lain kode Kita (Dsouza, 2021).

Sintaks umum untuk ini adalah:

```
map(function, iterable, [iterable1, iterable2, ...])
```

Gambar 1 Sintaks Umum Fungsi Map

### 3. Lambda expression

Lambda expression di Python adalah sebuah ekspresi untuk membuat fungsi. Lambda sendiri berasal dari teori kalkulus, yakni Lambda Calculus yang dikenalkan oleh Alonzo Church di tahun 1930. Dengan lambda, kita bisa membuat fungsi tanpa nama atau dikenal juga dengan anonymous function.

```
lambda x,y : x**2 + y**2
```

Gambar 2 Contoh Penggunaan Lambda

## HASIL DAN PEMBAHASAN

### A. Metode Implementasi Fungsi Pengurutan

Fungsi lambda Python menyediakan cara yang fleksibel untuk menentukan kriteria pengurutan khusus untuk berbagai struktur data, dari daftar sederhana hingga objek kompleks. Dengan memanfaatkan fungsi lambda bersama dengan fungsi `sorted()`, Anda dapat secara efisien mengatasi berbagai tantangan pengurutan dengan Python.

### B. Teknik Pengurutan yang Digunakan

```
1 import random
2
3 def sort_data(data, sorting_criteria):
4     sorted_data = list(map(lambda x: x[1],
5                             sorted(map(lambda x: (sorting_criteria(x[1]), x[1]),
6                                         enumerate(data)), key=lambda x: x[0])))
7     return sorted_data
8
9 data_acak = [random.randint(1, 100) for _ in range(50)] #membuat angka acak
10 sorting_criteria = lambda x: x**2 #fungsi lambda untuk pengurutan
11 data_terurut = sort_data(data_acak, sorting_criteria) #pengurutan data dengan fungsi sort_data
12
13 print("Data acak sebelum diurutkan:", data_acak)
14 print("Data acak setelah diurutkan:", data_terurut)
```

Gambar 3 Teknik Pengurutan

Teknik pengurutan yang diterapkan dalam percobaan ini menggunakan pendekatan pemrograman berbasis fungsi dengan memanfaatkan fungsi map, lambda, dan fungsi pengurutan bawaan Python (sorted). Pertama, setiap

elemen data diberi indeks menggunakan fungsi enumerate, lalu kriteria pengurutan yang ditentukan oleh fungsi lambda diterapkan pada nilai setiap elemen. Selanjutnya, pasangan (kriteria, nilai) diurutkan berdasarkan kriteria pengurutan, dan nilai sebenarnya dari setiap elemen diambil setelah proses pengurutan. Hasil akhir dari pengurutan disimpan dalam bentuk list. Dengan menggunakan fungsi sort\_data sebagai pengurut eksternal, fleksibilitas dalam penggunaan kriteria pengurutan yang berbeda dapat diterapkan pada data acak yang telah dibuat.

### 1. Pembentukan Pasangan dan Indeks Nilai

```
1 import random
2
3 def sort_data(data, sorting_criteria):
4     sorted_data = list(map(lambda x: x[1],
5                             sorted(map(lambda x: (sorting_criteria(x[1]), x[1]),
6                                         enumerate(data)), key=lambda x: x[0])))
7     return sorted_data
```

Gambar 4 Pembentukan Pasangan dan Indeks Nilai

Fungsi enumerate(data) digunakan untuk memberikan indeks pada setiap elemen dalam data. Ini menghasilkan pasangan (indeks, nilai) untuk setiap elemen.

### 2. Penerapan Pengurutan

```
3 def sort_data(data, sorting_criteria):
4     sorted_data = list(map(lambda x: x[1],
5                             sorted(map(lambda x: (sorting_criteria(x[1]), x[1]),
6                                         enumerate(data)), key=lambda x: x[0])))
7     return sorted_data
```

Gambar 5 Penerapan Pengurutan

Melalui fungsi map pertama, kriteria pengurutan yang ditentukan oleh fungsi lambda sorting\_criteria diterapkan pada nilai setiap elemen dalam pasangan (indeks, nilai). Hasilnya adalah pasangan (kriteria, nilai).

### 3. Pengurutan Berdasarkan Kriteria

```
3 def sort_data(data, sorting_criteria):
4     sorted_data = list(map(lambda x: x[1],
5                             sorted(map(lambda x: (sorting_criteria(x[1]), x[1]),
6                                         enumerate(data)), key=lambda x: x[0])))
7     return sorted_data
```

Gambar 6 Pengurutan Berdasarkan Kriteria

Fungsi `sorted` kemudian digunakan untuk mengurutkan pasangan (kriteria, nilai) berdasarkan kriteria pengurutan. Parameter `key=lambda x: x[0]` menentukan bahwa pengurutan dilakukan berdasarkan elemen pertama dari setiap pasangan, yaitu kriteria pengurutan.

#### 4. Ekstraksi Nilai Setelah Pengurutan

```
list(map(lambda x: x[1],
```

Gambar 7 Nilai Setelah Pengurutan

Melalui fungsi `map` kedua, nilai sebenarnya dari setiap elemen data diambil setelah proses pengurutan. Fungsi `lambda x: x[1]` digunakan untuk mengambil nilai (elemen kedua) dari setiap pasangan.

#### 5. Fungsi sebagai Pengurut Eksternal

```
3 def sort_data(data, sorting_criteria):
```

Gambar 8 Pengurutan Eksternal

Fungsi `sort_data` sebagai fungsi pengurut eksternal menerima data dan kriteria pengurutan sebagai parameter. Ini memberikan fleksibilitas dalam penggunaan fungsi ini untuk pengurutan berdasarkan kriteria yang berbeda.

#### 6. Membuat Data Acak

```
9 data_acak = [random.randint(1, 100) for _ in range(50)]
10 sorting_criteria = lambda x: x**2 #fungsi lambda untuk
11 data_terurut = sort_data(data_acak, sorting_criteria)
```

Gambar 9 Membuat Data Acak

#### 7. Fungsi Lambda untuk Kriteria Pengurutan

```
10 sorting_criteria = lambda x: x**2 #fungsi lambda untuk pengurutan
```

Gambar 10 Kriteria Pengurutan dengan Lambda

Fungsi `lambda sorting_criteria` digunakan untuk memberikan kriteria pengurutan. Dalam contoh ini, kriteria pengurutan adalah kuadrat dari setiap elemen data.

#### 8. Penggunaan Fungsi `sort_data`

```
11 data_terurut = sort_data(data_acak, sorting_criteria)
```

Gambar 11 Fungsi Sort Data

Fungsi `sort_data` dipanggil dengan memberikan data acak dan fungsi `lambda` sebagai argumen untuk melakukan pengurutan.

#### 9. Menampilkan Hasil

```
13 print("Data acak sebelum diurutkan:", data_acak)
14 print("Data acak setelah diurutkan:", data_terurut)
```

Gambar 12 Tampilan Hasil

Kode Python di atas menunjukkan sebuah implementasi pengurutan data menggunakan konsep pemrograman berbasis fungsi. Pertama-tama, kita menciptakan sebuah list `data_acak` yang berisi 50 angka acak antara 1 hingga 100. Tujuan utama dari kode ini adalah mengurutkan data acak ini berdasarkan suatu kriteria tertentu.

Fungsi `sort_data` berperan sebagai pengatur utama dalam proses pengurutan. Dalam konteks ini, penggunaan fungsi `lambda (sorting_criteria)` memungkinkan pengguna untuk dengan mudah mengubah kriteria pengurutan tanpa harus menulis fungsi terpisah. Pada kasus ini, kriteria pengurutan adalah kuadrat dari setiap elemen data. Proses pengurutan dimulai dengan membentuk pasangan (indeks, nilai) menggunakan fungsi `enumerate(data)`. Setelah itu, kriteria pengurutan diterapkan pada setiap nilai melalui fungsi pemetaan pertama, menghasilkan pasangan (kriteria, nilai). Selanjutnya, data diurutkan berdasarkan kriteria pengurutan menggunakan fungsi `sorted` dengan menggunakan elemen pertama dari setiap pasangan sebagai kunci pengurutan.

Fungsi pemetaan kedua kemudian digunakan untuk mengambil nilai (elemen kedua) dari setiap pasangan setelah proses pengurutan. Hasil akhirnya adalah list `sorted_data` yang berisi nilai-nilai data yang telah diurutkan berdasarkan kriteria yang telah ditentukan. Hasil pengurutan kemudian dicetak menggunakan pernyataan `print`. Dua

baris pertama menampilkan data acak sebelum diurutkan, sementara dua baris terakhir menampilkan data setelah proses pengurutan.

Secara keseluruhan, kode ini memberikan contoh konkret bagaimana konsep pemrograman berbasis fungsi, seperti fungsi lambda dan pemetaan, dapat digunakan untuk membuat proses pengurutan yang efisien dan mudah dimengerti. Selain itu, fleksibilitas dalam menentukan kriteria pengurutan memberikan daya adaptasi yang tinggi, memungkinkan penggunaan kode ini dalam berbagai konteks dengan kriteria yang berbeda.

## SIMPULAN DAN SARAN

Implementasi pengurutan data dapat menggunakan konsep pemrograman berbasis fungsi. Dengan pembuatan list data\_acak yang berisi angka acak dengan tujuan utama kode untuk mengurutkan data berdasarkan kriteria tertentu. Proses pengurutan melibatkan pembentukan pasangan (indeks, nilai) melalui enumerate(data), pengurutan data berdasarkan kriteria menggunakan fungsi sorted. Fungsi pemetaan menghasilkan list sorted\_data dengan nilai-nilai data yang terurut. Hasil akhir dari pengurutan menampilkan data acak sebelum dan sesudah proses pengurutan.

Pemograman aplikasi pengurutan data dengan menggunakan fungsi dapat dikatakan lebih efektif dan efisien dengan blok kode yang lebih sedikit daripada tidak menggunakan fungsi, sehingga penerapan fungsi pada program ini dapat dinyatakan lebih unggul dan relevan. Pengaplikasian pengurutan data ini dapat digunakan diberbagai macam bidang, misalnya pada e-

commerce untuk pengurutan produk berdasarkan popularitas, harga, dan ulasan konsumen, pada bidang keuangan yaitu guna menganalisis pergerakan saham suatu perusahaan, dan masih banyak biang lainnya. Pengembangan aplikasi atau program pengurutan data ini masih perlu dilakukan jika program atau aplikasi ini dipergunakan untuk kepentingan lebih lanjut.

## DAFTAR PUSTAKA

- Algoritma Selection Sort di Python*. (n.d.). BINUS UNIVERSITY. Retrieved March 12, 2024, from <https://binus.ac.id/bandung/2023/10/algoritma-selection-sort-di-python/>
- Dr. Joseph Teguh Santoso, S.Kom, M.Kom. (n.d.). proyek coding dengan python. 4.
- Germeç, M., & Germec, M. (2022, June 14). (PDF) 17. Lambda Functions in Python. ResearchGate. Retrieved March 12, 2024, from [https://www.researchgate.net/publication/361276559\\_17\\_Lambda\\_Functions\\_in\\_Python](https://www.researchgate.net/publication/361276559_17_Lambda_Functions_in_Python)
- Jain, S. (2023, November 17). *Python Lambda Functions*. GeeksforGeeks. Retrieved March 12, 2024, from <https://www.geeksforgeeks.org/python-lambda-anonymous-functions-filter-map-reduce/>
- Puthut, T. W., & Kadier, R. E. A. ((2012)). Desain dan Impleetasi Kontroler Prediktif Logika Fuzzy untuk Pengaturan Injeksi Bahan Bakar Ignition Engine. *Jurnal Teknik ITS*, 1(1), A164-A169