

APLIKASI PENGURUTAN DATA

Oktavia Nurwinda Puspitasari¹, Mutiara Dian Pitaloka², Syadza Puspadari Azhar³, Kayla Amanda Sukma⁴, Muhammad Wafi Raihan⁵

Jurusan Sains Data, Fakultas Sains, Institut Teknologi Sumatera, Lampung Selatan, Indonesia

Email: oktavia.122450041@student.itera.ac.id mutiara.122450047@student.itera.ac.id

syadza.122450072@student.itera.ac.id kayla.122450086@student.itera.ac.id

muhammad.122450144@student.itera.ac.id

1. Pendahuluan

Dalam dunia pemrograman, pengurutan data adalah salah satu proses yang umum dan penting dalam mengelola suatu data dalam kondisi acak dapat menjadi urut dari data terkecil sampai data terbesar begitu sebaliknya, dari data terbesar sampai dengan data terkecil (Wahyudi, 2009). Dengan mengurutkan data, kita dapat menyusunnya menjadi urutan data yg terstruktur dan dapat diakses dengan cara yg lebih efisien. Aplikasi pengurutan data ini dapat mengurutkan data dengan cara manual, dan dapat juga mengurutkan data secara otomatis. Fitur untuk mengurutkan data secara otomatis dapat mengatur kriteria pengurutan secara otomatis seperti mengurutkan data berdasarkan nama, usia, dan lainnya.

Ada berbagai metode dan algoritma yang digunakan untuk mengurutkan data, termasuk penggunaan fungsi “map” dan “lambda”. Lambda bisa memiliki lebih dari satu argumen atau parameter, tapi hanya bisa memiliki satu ekspresi atau isi (Muhardian, 2019).

Fungsi map menerapkan suatu operasi atau fungsi pada setiap elemen dalam sebuah data (Alza, 2019). Biasanya lambda function digunakan pada fungsi yang parameternya adalah fungsi seperti map.

Penggunaan kombinasi fungsi “map” dan “lambda” dalam pengurutan data untuk menerapkan kriteria pengurutan yang spesifik tanpa perlu menuliskan fungsi pengurutan yang terpisah. Pengurutan data yang berbeda - beda tanpa perlu menuliskan ulang fungsi baru di setiap pengurutan data.

Dalam laporan ini kami akan membuat aplikasi pengurutan data untuk mengurutkan data secara otomatis.

2. Metode

a. Pemrograman Berbasis Fungsi

Pemrograman Berbasis Fungsi adalah suatu paradigma pemrograman yang menggunakan fungsi sebagai unit utama untuk menjalankan komputasi (Syafery, 2023). Dengan menggunakan *functional programming* dalam pengaplikasian pengurutan data kita dapat menekankan penggunaan fungsi sebagai elemen utama pada implementasi program, program lebih terfokus pada fungsi sebagai unit utama, meningkatkan fleksibilitas untuk pengurutan kriteria data, serta menggunakan *high-order function* untuk operasi pengurutan yang dinamis berdasarkan kriteria.

b. Fungsi Map

Fungsi map adalah *high-order function* yang dapat digunakan untuk menjalankan fungsi dengan banyak nilai masukan. Perintah map dapat memproses *iterable* dari suatu data satu per satu dan meletakkan hasilnya dalam *map object* (Alza, 2019). Dalam pengaplikasian pengurutan data, map digunakan untuk menerapkan fungsi di setiap elemen *iterable* dan mengembalikan *iterator* dari hasilnya. Dengan kata lain map akan menghasilkan list dari hasil pengurutan data berdasarkan nama dan usia.

c. Fungsi Lambda

Lambda atau dikenal sebagai *anonymous function* adalah suatu ekspresi untuk membuat fungsi. Lambda bisa terdiri dari atas lebih dari satu argumen/parameter tetapi hanya mempunyai satu ekspresi (Muhardian, 2019). Fungsi lambda juga tidak perlu didefinisikan dengan kata kunci def. Pada pengaplikasian pengurutan data fungsi lambda digunakan sebagai kunci untuk pengurutan data dan memberikan nilai pada atribut sesuai dengan kriteria dalam list data. Disini data akan diurutkan sesuai dengan abjad nama dan urutan umur.

3. Pembahasan

```
# Data yang digunakan
data = [
    {'Nama': 'Joko', 'Usia': 20},
    {'Nama': 'Kael', 'Usia': 23},
    {'Nama': 'Anwar', 'Usia': 25},
    {'Nama': 'Siti', 'Usia': 19},
    {'Nama': 'Anisa', 'Usia': 18},
]
```

Pada kode diatas, kita masukkan terlebih dahulu list data yang akan kita gunakan untuk diurutkan. Kami memasukkan 5 data yang terdiri dari nama dan usia agar dapat diurutkan berdasarkan 2 kriteria, yaitu diurutkan berdasarkan abjad dari nama, serta diurutkan berdasarkan umur dari yang lebih muda hingga yang paling tua.

```
# Kriteria yang akan diurutkan
kriteria_nama = 'Nama'
kriteria_usia = 'Usia'
```

Definisikan pengurutan data. Pada kode di atas, kami mendefinisikan 2 kriteria yaitu kriteria nama dan kriteria usia.

```
# Mendefinisikan fungsi untuk pengurutan data
def urutkan_data(data, kriteria):
    return sorted(data, key = lambda x: x[kriteria])
```

Setelah mendefinisikan kriteria yang akan diurutkan, kami membuat fungsi definisi dengan parameter data dan kriteria yang akan mengembalikan nilai yang telah diurutkan dengan fungsi *sort* dengan data yang digunakan adalah data yang kita masukkan di awal. Setelah itu menetapkan key fungsi lambda yang akan mengembalikan nilai x berdasarkan indeks yang ditentukan oleh variabel 'kriteria'.

```
# Menggunakan map serta lambda untuk pengaplikasian pengurutan data
hasil_urutan_nama = list(map(lambda x: urutkan_data(data, x),
[kriteria_nama]))
hasil_urutan_usia = list(map(lambda x: urutkan_data(data, x),
[kriteria_usia]))
```

Dengan fungsi di atas, kami membuat variabel hasil dari urutan dari data yang telah dibuat dengan nama variabel "hasil_urutan_nama" dan "hasil_urutan_usia" dengan menggunakan fungsi map dengan argumen fungsi adalah fungsi urutkan_data yang telah kami definisikan sebelumnya dengan parameter data dan x, setelah itu menentukan ulangan list yang akan kita gunakan. Pada "hasil_urutan_nama", kami menggunakan "kriteria_nama" untuk menerapkan fungsi "urutkan_data" pada setiap kriteria dalam list "kriteria_nama", dimana seperti yang telah kami definisikan di awal, "kriteria_nama" menyimpan list "nama" dari list "data" yang telah dimasukkan sebelumnya. Hal yang sama berlaku untuk "hasil_urutan_usia" yang akan mengurutkan berdasarkan usia.

```
# Menampilkan hasil output
print("Hasil urutan berdasarkan nama:", hasil_urutan_nama[0])
print("Hasil urutan berdasarkan usia:", hasil_urutan_usia[0])
```

Untuk menampilkan hasil dari data yang telah diurutkan dengan fungsi yang telah dibuat sebelumnya dengan menggunakan print yang akan memberikan output hasil urutan nama dan hasil urutan usia yang dimulai dari index[0].

4. Kesimpulan

Dalam laporan ini, kami membuat aplikasi pengurutan data dengan menggunakan paradigma pemrograman berbasis fungsi. Pendekatan ini memanfaatkan fungsi sebagai faktor utama dalam menjalankan program, meningkatkan fleksibilitas dalam menentukan kriteria pengurutan data, dan menggunakan *high-order function* seperti map dan lambda untuk operasi pengurutan yang dinamis.

Metode pengurutan yang digunakan dalam aplikasi ini mencakup penggunaan fungsi lambda untuk mengurutkan data berdasarkan kriteria tertentu, seperti nama dan usia. Fungsi

map digunakan untuk menerapkan fungsi pengurutan pada setiap elemen *iterable*, dan menghasilkan list dari hasil pengurutan data.

Penggunaan kombinasi fungsi lambda dan map memungkinkan pengurutan data yang berbeda-beda tanpa perlu menuliskan ulang fungsi baru di setiap pengurutan data. Dengan fungsi ini pula kita dapat melakukan fungsi pengurutan dengan menggunakan sedikit memori sehingga akan menghemat penyimpanan. Kami mengimplementasikan pengurutan berdasarkan abjad nama dan urutan usia terkecil, namun metode ini dapat dengan mudah disesuaikan dengan kriteria lain menyesuaikan dengan data yang ada.

Melalui implementasi dengan data yang telah dimasukkan di awal sebagai contoh, aplikasi ini akan menghasilkan output yang menampilkan data yang diurutkan berdasarkan nama dan usia. Dengan demikian, aplikasi pengurutan data ini dapat menjadi landasan untuk pengembangan lebih lanjut dalam pemrograman berbasis fungsi dan aplikasi pengelolaan data.

Daftar Pustaka

- Alza. (2019, June 21). *Menggunakan fungsi map - Belajar koding untuk pemula*. Retrieved March 6, 2024 from Belajar koding untuk pemula: <https://koding.alza.web.id/menggunakan-perintah-map/>
- Muhardian, A. (2019, December 18). *Belajar Python: Membuat Fungsi dengan Lambda Expression*. Retrieved March 6, 2024 from Petani Kode: <https://www.petanikode.com/python-lambda/>
- Syaferi, F. (2023, November 2). *Memahami Konsep Functional Programming di Python*. Retrieved March 6, 2024 from Universitas Mahakarya Asia: <https://blog.unmaha.ac.id/memahami-konsep-functional-programming-di-python>
- Wahyudi, E. N. (2009, January 1). *Algoritma Sederhana dalam Memahami Proses Pengurutan Data*. Retrieved March 6, 2024 from Neliti: <https://www.neliti.com/publications/245915/algoritma-sederhana-dalam-memahami-proses-pengurutan-data>