

Implementasi Fungsi Map, Filter, dan Reduce dalam Pengolahan Data Laporan Polisi Detroit

Kelompok 7

¹Aditya Rahman
122450113

²Farahanum Afifah Ardiansyah
122450056

³Fayyaza Aqila Syafitri Achjar
122450131

⁴Azizah Kusumah Putri
122450068

⁵Eggi Satria
122450032

⁶Meira Listyaningrum
122450011

Program Studi Sains Data, Fakultas Sains, Institut Teknologi Sumatera
Jl. Terusan Ryacudu, Way Huwi, Kec. Jati Agung, Kabupaten Lampung Selatan, Lampung 35365
Kontak : ¹aditya.122450113@student.itera.ac.id, ²farahanum.122450056@student.itera.ac.id,
³fayyaza.122450131@student.itera.ac.id, ⁴azizah.122450068@student.itera.ac.id,
⁵eggi.122450032@student.itera.ac.id, ⁶meira.122450011@student.itera.ac.id

1. PENDAHULUAN

1.1 Latar Belakang

Seiring dengan berkembangnya teknologi digital, data yang diperoleh menjadi sangat banyak sehingga pengolahan data menjadi sangat penting. Python, sebagai bahasa pemrograman yang populer juga menawarkan kemudahan pengolahan data menggunakan fungsi-fungsi *built-in* seperti fungsi map, filter dan reduce. Ketiganya merupakan fungsi yang digunakan untuk data-data berbentuk *list* atau *dictionary*.

Fungsi map diterapkan untuk mengubah elemen-elemen dalam *list* sesuai dengan fungsi yang dipasangkan dengannya. Hasilnya adalah *list* yang berisi nilai-nilai yang telah dikenakan fungsi di dalamnya. Selanjutnya, fungsi filter digunakan untuk menyeleksi elemen-elemen dalam *list* yang memenuhi kondisi yang diterapkan ke dalamnya. Terakhir adalah fungsi reduce, fungsi ini tersedia dalam modul *functools* dan digunakan untuk menerapkan fungsi yang dimasukkan secara berurutan dan hasilnya hanya memiliki satu output.

Ketiga fungsi ini dapat digunakan untuk melakukan pemrosesan data dengan mengubah struktur data mentah yang digunakan menjadi struktur yang lebih kompleks dalam bahasa pemrograman python.

1.2 Rumusan Masalah

- Bagaimana cara mengubah data menjadi list atau dictionary?
- Bagaimana penggunaan fungsi map, filter, dan reduce dalam melakukan pemrosesan data?
- Bagaimana cara melakukan penambahan item pada dictionary?

1.3 Tujuan

Mengimplementasikan fungsi map, filter, dan reduce dalam melakukan pemrosesan data.

2. METODE

2.1 Map

Map merupakan fungsi yang digunakan untuk menerapkan suatu fungsi tertentu dalam tiap elemen di sebuah iterasi, contohnya list tuple, dan array. Map akan mengembalikan hasil dalam bentuk iterator. Fungsi yang diterapkan pada map dapat berupa fungsi bawaan atau fungsi yang didefinisikan oleh pengguna. (Mitchell. D, 2015)

2.2 Filter

Filter merupakan fungsi bawaan dari python yang digunakan untuk memfilter atau menyaring elemen-elemen dalam suatu iterasi berdasarkan kondisi yang telah diberikan. Dengan filter akan memungkinkan untuk memilih subset dari data yang memenuhi suatu kriteria tertentu. (Mitchell. D, 2015)

2.3 Reduce

Reduce merupakan fungsi dalam modul functools yang digunakan untuk menggabungkan seluruh elemen dalam suatu iterasi menjadi nilai dengan menggunakan fungsi tertentu. Dengan reduce memungkinkan pengguna untuk membuat pemrograman lebih efisien dalam melakukan operasi, seperti penjumlahan, pengurangan, perkalian dan operasi lainnya pada seluruh elemen suatu struktur data. (VanderPlas J., 2016)

2.4 JSON

JSON (JavaScript Object Notation) merupakan suatu format pertukaran data yang mudah dibaca atau ringan, serta ditulis oleh manusia. JSON umumnya akan digunakan untuk mentransmisikan data terstruktur antara sebuah server dan sebuah aplikasi web sebagai pengganti XML. Di dalam python modul JSON ini digunakan untuk membaca data serta menulis data JSON. (VanderPlas J., 2016)

3. PEMBAHASAN

Dalam bab ini, kami melakukan analisis terhadap fungsi map, filter dan reduce dalam beberapa metode. Kami mengimplementasikan penerapan map, filter dan reduce menggunakan bahasa pemrograman python dengan membuat penerapan call service, kode yang kami buat dan gunakan untuk analisis ini dapat diakses pada link berikut :

https://colab.research.google.com/drive/10v60SobbCAGN_wD9N1MoMRbQdKvH0HT3?usp=sharing

3.1 Penerapan *map*, *filter* dan *reduce*

Pada penerapan fungsi map, filter dan reduce yang kami lakukan memiliki tugas yang berbeda. Pertama yaitu fungsi *Filter* yang digunakan untuk menyaring file dictionary yang tidak memiliki nilai pada kolom Zip, Neighborhood, totalresponstime, time_on_scene, atau totaltime, dan memastikan hanya data lengkap yang akan diproses lebih lanjut lagi. Fungsi *map* diterapkan untuk mengambil nilai spesifik (seperti totalresponstime, time_on_scene dan totaltime) dari setiap dictionary dalam list, mengkonversinya ke tipe yang sesuai dan menggantikan nilai yang hilang dengan 0. Kemudian *reduce*

digunakan untuk menjumlahkan semua nilai yang telah diekstraksi dan dihitung oleh *map* menghasilkan total yang kemudian dibagi dengan jumlah elemen untuk mendapatkan rata-rata *totalresponstime*, *time_on_scene* dan *totaltime* untuk keseluruhan dataset serta untuk *neighborhood* secara individual

3.2 Model Populasi Polisi Detroit

Pada model populasi polisi detroit code yang diberikan akan membaca data dari file CSV Laporan Polisi Detroit serta memfilter data yang hilang dan akan menghitung total waktu respons rata-rata, waktu pengiriman rata-rata, dan total waktu rata-rata untuk kepolisian detroit.

Code `list_dict_output = []` akan membuat list kosong yang akan menampung data yang telah diproses. Selanjutnya data dari file CSV akan diubah menjadi list dictionary menggunakan fungsi `csv.DictReader()` dan data akan disimpan dalam variabel baru yaitu `data`. Selanjutnya akan dilakukan pemfilteran data dengan menggunakan fungsi `list_dict_output = list(filter(lambda x: ..., data))` dan hasil filter akan disimpan dalam `list_dict_output`. Langkah selanjutnya yaitu menghitung total waktu respons rata-rata dengan menggunakan fungsi `reduce` dan `map` yang dimana `reduce` akan menjumlahkan semua nilai “*totalresponsetime*” dalam `list_dict_output` dan fungsi `map` digunakan untuk mengubah nilai “*totalresponsetime*” menjadi bilangan bulat. Hasil penjumlahan dibagi dengan jumlah baris data yang valid dengan fungsi `(len(list_dict_output))`. Selanjutnya akan dilakukan penghitungan total waktu rata-rata yang juga menggunakan fungsi `reduce` untuk menghitung total waktu rata-rata. Dan langkah terakhir yang dilakukan yaitu mencetak hasil yang dimana `print("Total Response Time Average: ", total_response_time)` akan menampilkan total waktu respons rata-rata dengan format keterangan, `print("Delivery Time Average: ", delivery_time)` akan menampilkan waktu pengiriman rata-rata dengan format keterangan, dan `print("Total Time Average: ", total_time)` akan menampilkan total waktu rata-rata dengan format keterangan.

3.3 Memodelkan Neighborhood Samples

Untuk memodelkan neighborhood samples langkah pertama yang dilakukan yaitu mencari neighborhood unik dengan code `neighborhoods = list(set(map(lambda x: x["neighborhood"], data)))` untuk menemukan semua neighborhood unik dalam data, `set()` untuk menghapus nilai duplikat, `map()` untuk mengubah setiap baris data menjadi nama neighborhoodnya serta `list()` untuk mengubah hasil menjadi list. Langkah selanjutnya yaitu membuat dictionary neighborhood menggunakan code `neighborhood_dict = {}` untuk membuat dictionary kosong untuk menyimpan data neighborhood. Selanjutnya mengelompokkan data berdasarkan neighborhood yang menggunakan fungsi `filter()` untuk menyaring data berdasarkan kondisi yang diberikan. Selanjutnya menghitung statistik waktu rata-rata yang menggunakan fungsi `reduce()` untuk menjumlahkan semua nilai “*totalresponsetime*” dalam data neighborhood, dan fungsi `map()` untuk mengubah nilai “*totalresponsetime*” menjadi bilangan bulat. Fungsi `delivery_time` akan menghitung waktu pengiriman rata-rata dan `total_time` akan menghitung total waktu rata-rata dengan cara yang sama seperti `total_response_time`. Selanjutnya menambahkan data populasi detroit dengan code `neighborhood_dict["All Detroit"] = { ... }`. Dan code ini akan menghasilkan dictionary `neighborhood_dict`.

3.4. Membuat file output JSON

Pada code ini hal pertama yang dilakukan yaitu membuat file JSON dengan syntax with open("output.json", "w") as f: yang akan membuka file bernama "output.json". Selanjutnya menyimpan data ke file JSON dengan menggunakan syntax json.dump(neighborhood_dict, f) untuk mengubah data neighborhood_dict menjadi format JSON. Selanjutnya membaca file JSON dengan syntax with open("output.json", "r") as f: yang akan membuka file "output.json" dan menyimpan file tersebut dalam variabel f. Langkah selanjutnya memuat data dari file JSON dengan syntax data = json.load(f) yang menggunakan model json untuk membaca isi dari file JSON dan akan mengubah kembali menjadi format dictionary yang disimpan dalam variabel data. Selanjutnya mencetak pesan dengan code print("Output JSON created").

4. KESIMPULAN

Berdasarkan dari pembahasan yang ada, artikel ini membahas bagaimana penerapan fungsi map, filter, dan reduce untuk melakukan analisis pada data laporan polisi Detroit. Fungsi filter digunakan untuk menyaring data yang tidak lengkap, kemudian fungsi map untuk mengonversi dan menggantikan nilai yang hilang dengan nilai 0 dari setiap dictionary dalam list. Fungsi reduce digunakan untuk menjumlahkan nilai yang telah diekstraksi untuk menghitung rata-rata totalresponsetime, time_on_scene, dan totaltime baik untuk keseluruhan dataset maupun per neighborhood. Pada proses memodelkan sampel neighborhood, kita harus menemukan neighborhood yang unik kemudian mengelompokkan data berdasarkan neighborhood untuk menghitung statistik waktu rata-rata. Pada proses pembuatan file output JSON, data dari neighborhood_dict disimpan dalam file JSON, setelah itu file JSON dibaca kembali dan isinya dimuat ke dalam variabel data dengan mengubah kembali menjadi format dictionary.

5. DAFTAR PUSTAKA

Kadir, A. (2019). *Logika Pemrograman Python*. Elex Media Komputindo.

Mertz, D. (2015). *Functional Programming in Python*. O'Reilly Media, Inc.

<https://www.oreilly.com/library/view/functional-programming-in/9781492048633/>

Mitchell, D. (2015). The Python Standard Library by Example. *Addison-Wesley Professional*.

Python. (n.d.). *Sorted*. Python. Retrieved May 1, 2024, from

<https://docs.python.org/3/library/functions.html#sorted>

Sestoft, P. (2012). *Programming Language Concepts*. Springer. 10.1007/978-1-4471-4156-3_5

VanderPlas J. (2016). Python Data Science Handbook. *Essential Tools for Working with Data*.