

Penerapan fungsi map, filter dan reduce pada pengolahan data Layanan Panggilan Darurat 911

Dearnı Monica Manik, Nabila Anilda Zahrah, Abit Ahmad Oktarian,
Allya Nurul Islami Pasha, Yohana Manik, David Bobby C.Nainggolan

Program Studi Sains Data Institut Teknologi Sumatera

Jl. Terusan Ryacudu, Way Huwi, Kec. Jatiagung, Kabupaten Lampung Selatan,
Lampung 35365

Email:

nabila.122450063@student.itera.ac.id dearni.122450075@student.itera.ac.id
abit.122450042@student.itera.ac.id allya.122450033@student.itera.ac.id
david.122450048@student.itera.ac.id yohana.122450126@student.itera.ac.id

Pendahuluan

Layanan panggilan darurat 911 adalah salah satu hal penting dalam setiap komunitas untuk memberikan bantuan cepat dalam situasi darurat. Pengelolaan data yang efisien dalam konteks ini menjadi krusial untuk memastikan respon yang tepat waktu dan efektif terhadap situasi darurat yang muncul. Dalam upaya untuk meningkatkan efisiensi pengolahan data pada layanan panggilan darurat 911, penerapan fungsi map, filter, dan reduce menjadi relevan. Fungsi-fungsi ini adalah bagian integral dari pemrograman fungsional yang memungkinkan manipulasi data dengan cara yang efisien dan efektif.

Dalam laporan ini, kami akan mengeksplorasi konsep dan penerapan fungsi map, filter, dan reduce dalam pengolahan data Layanan Panggilan Darurat 911. Kami akan membahas bagaimana fungsi-fungsi ini dapat diterapkan untuk menyederhanakan proses pengolahan data, memungkinkan analisis yang lebih cepat, dan meningkatkan responsibilitas sistem secara keseluruhan.

Langkah-langkah praktis dalam penerapan fungsi-fungsi ini akan diuraikan, bersama dengan contoh-contoh konkret tentang bagaimana mereka dapat diterapkan dalam pengelolaan data panggilan darurat. Diharapkan laporan ini akan memberikan wawasan yang berharga tentang cara memanfaatkan fungsi map, filter, dan reduce untuk meningkatkan efisiensi dan responsibilitas sistem Layanan Panggilan Darurat 911.

Metode

1. Pengumpulan Data

Data untuk Layanan Panggilan Darurat 911 dikumpulkan dari berbagai sumber, termasuk panggilan telepon, teks, dan data lokasi. Informasi yang terkumpul mencakup waktu panggilan, jenis keadaan darurat, lokasi, dan rincian tambahan yang relevan.

2. Pemrosesan Awal Data

Data yang terkumpul kemudian diproses untuk membersihkan dan memformatnya sesuai dengan kebutuhan analisis. Langkah ini melibatkan penghapusan data yang tidak relevan, penanganan data yang hilang atau tidak lengkap, serta konversi format data ke dalam bentuk yang dapat diolah lebih lanjut.

3. Penerapan Fungsi Map

Fungsi map digunakan untuk menerapkan transformasi ke setiap elemen data dalam rangkaian data panggilan darurat. Misalnya, fungsi map dapat digunakan untuk mengekstrak informasi tertentu dari setiap panggilan, seperti jenis kejadian atau waktu panggilan.

4. Penerapan Fungsi Filter

Fungsi filter digunakan untuk menyaring data berdasarkan kriteria tertentu. Contohnya, kita dapat menggunakan fungsi filter untuk menghapus panggilan yang bukan merupakan kejadian darurat yang sebenarnya, atau untuk menyaring panggilan berdasarkan jenis kejadian tertentu. Pada program yang dibuat, fungsi filter digunakan untuk menghapus baris yang memiliki data yang hilang di kolom 'zip_code' atau 'neighborhood'. Ini dilakukan dengan menggunakan metode dropna pada DataFrame dataset dengan parameter subset=['zip_code', 'neighborhood'], yang menghasilkan filtered_dataset.

5. Penerapan Fungsi Reduce

Fungsi reduce digunakan untuk merangkum atau mengagregasi data. Misalnya, kita dapat menggunakan fungsi reduce untuk menghitung jumlah panggilan darurat yang diterima dalam rentang waktu tertentu atau untuk menghitung statistik keseluruhan tentang jenis kejadian yang paling umum. Pada program yang dibuat, fungsi reduce digunakan untuk menghitung total waktu respons rata-rata, waktu pengiriman rata-rata, dan total waktu rata-rata untuk setiap kuartal (neighborhood). Langkah-langkahnya adalah sebagai berikut:

- Data yang di filter di organisir ke dalam neighborhood_data, sebuah kamus (dictionary) di mana kunci adalah nama kuartal dan nilai adalah daftar data yang sesuai.
- Untuk setiap kuartal, menggunakan reduce untuk menghitung total waktu respons, waktu pengiriman, dan total waktu.
- Nilai-nilai ini kemudian digunakan untuk menghitung rata-rata waktu respons, waktu pengiriman, dan total waktu untuk setiap kuartal.

6. Pembuatan List Dictionary

Hasil perhitungan disimpan dalam list dictionary `neighborhood_statistics`. Setiap elemen dalam list ini mewakili statistik untuk satu kuartal, termasuk rata-rata waktu response, waktu pengiriman, total waktu, dan populasi kuartal tersebut.

7. Penulisan ke File JSON

List dictionary `neighborhood_statistics` kemudian ditulis ke dalam file JSON menggunakan modul `json`. File ini berisi statistik untuk setiap kuartal dalam format yang mudah dibaca.

8. Unduh File JSON

Akhirnya, file JSON yang telah dibuat dapat diunduh menggunakan `files.download` dari Colab, sehingga hasil perhitungan dapat disimpan dan diakses untuk analisis lebih lanjut. Melalui metode ini, data diproses, statistik dihitung, dan hasilnya disimpan dalam format yang dapat digunakan dan dipertahankan untuk analisis lebih lanjut.

9. Evaluasi dan Analisis

Setelah penerapan fungsi `map`, `filter`, dan `reduce`, data yang dihasilkan dievaluasi dan dianalisis untuk mendapatkan wawasan yang berguna tentang tren, pola, dan statistik terkait dengan layanan panggilan darurat. Analisis ini dapat digunakan untuk meningkatkan efisiensi sistem dan responsibilitas layanan.

10. Iterasi dan Pembaruan

Proses ini merupakan iteratif, dan hasil analisis digunakan untuk memperbaiki dan memperbarui proses pengolahan data secara terus-menerus. Ini termasuk penyesuaian kriteria `filter`, transformasi data, dan strategi reduksi untuk memastikan bahwa sistem tetap responsif terhadap perubahan kebutuhan dan kondisi.

Dengan mengikuti langkah-langkah di atas, penerapan fungsi `map`, `filter`, dan `reduce` dapat membantu meningkatkan efisiensi dan responsibilitas sistem dalam pengolahan data Layanan Panggilan Darurat 911.

Pembahasan

```
import pandas as pd

# Membaca dataset
file_path = '1.csv' # Ganti 'nama_file.csv' dengan nama file CSV Anda
dataset = pd.read_csv(file_path)

# Menghapus baris yang memiliki data yang hilang di kolom Zip atau Neighborhood
filtered_dataset = dataset.dropna(subset=['zip_code', 'neighborhood'])

# Menghitung total waktu respons rata-rata, waktu pengiriman rata-rata, dan total waktu rata-rata
total_time_respond = filtered_dataset['traveltime'].mean()
total_time_arrival = filtered_dataset['totalresponsetime'].mean()
total_response_time = filtered_dataset['totaltime'].mean()

print("Total Average Time to Respond:", total_time_respond)
print("Total Average Time to Arrival:", total_time_arrival)
print("Total Average Response Time:", total_response_time)
```

Gambar 1

Pada gambar 1 kode yang digunakan menggunakan fungsi `filter` untuk menghapus baris yang memiliki data yang hilang di kolom 'zip_code' atau 'neighborhood'. Dalam contoh ini, `dropna` digunakan untuk melakukan filter terhadap dataset berdasarkan kolom tertentu, yang pada akhirnya menghasilkan dataset baru yang telah difilter.

```
# Filter baris yang memiliki data yang hilang di kolom Zip atau Neighborhood
filtered_data = list(filter(lambda x: x['zip_code'] != '' and x['neighborhood'] != '', data))

# Hitung total waktu respons rata-rata, waktu pengiriman rata-rata, dan total waktu rata-rata untuk kepolisian Detroit
total_time_respond = reduce(lambda acc, val: acc + val['traveltime'], filtered_data, 0)
total_time_arrival = reduce(lambda acc, val: acc + val['totalresponsetime'], filtered_data, 0)
total_response_time = reduce(lambda acc, val: acc + val['totaltime'], filtered_data, 0)

average_time_respond = total_time_respond / len(filtered_data)
average_time_arrival = total_time_arrival / len(filtered_data)
average_response_time = total_response_time / len(filtered_data)

print("Total Average Time to Respond:", average_time_respond)
print("Total Average Time to Arrival:", average_time_arrival)
print("Total Average Response Time:", average_response_time)
```

Gambar 2

Pada gambar 2 diatas digunakan fungsi `reduce` untuk menghitung total waktu respons rata-rata, waktu pengiriman rata-rata, dan total waktu rata-rata untuk setiap kuartal (neighborhood). Fungsi `reduce` digunakan untuk mengumpulkan nilai-nilai dalam suatu iterable (dalam hal ini, daftar data yang disortir berdasarkan kuartal) menjadi satu nilai tunggal.

Meskipun tidak secara langsung ditunjukkan dalam kode yang diberikan, `map` bisa diterapkan dalam beberapa kasus di mana kita ingin menerapkan suatu fungsi ke setiap elemen dalam suatu iterable. Misalnya, jika ada kebutuhan untuk mengubah format atau menghitung

nilai-nilai tertentu untuk setiap entri dalam dataset, `map` bisa digunakan untuk menerapkan fungsi tersebut ke setiap elemen. Penggunaan fungsi `map`, `filter`, dan `reduce` di sini mencerminkan efisiensi dalam manipulasi data. Mereka membantu mengelola dataset dengan cara yang terstruktur dan efisien, mengurangi kerumitan dan jumlah baris kode yang diperlukan untuk mencapai tujuan yang sama. Penggunaan fungsi-fungsi ini juga meningkatkan keterbacaan kode. Mereka menyediakan metode yang terstruktur dan baku untuk melakukan operasi-operasi umum pada data, membuatnya lebih mudah bagi pembaca untuk memahami logika di balik manipulasi data tersebut.

Melalui penerapan fungsi-fungsi ini, data dari layanan panggilan darurat 911 diolah dengan baik. Misalnya, dengan menggunakan `filter`, baris-baris yang tidak lengkap atau tidak relevan dapat dihapus dari dataset, dan dengan `reduce`, statistik yang signifikan seperti rata-rata waktu respons dan waktu kedatangan dapat dihitung untuk setiap kuartal, yang dapat memberikan wawasan yang berharga dalam analisis data lebih lanjut.

Kesimpulan

1. Penggunaan fungsi map memungkinkan transformasi data yang cepat dan efisien, sementara fungsi filter memungkinkan penyaringan data yang relevan. Ini mengurangi beban kerja dan meningkatkan kecepatan pemrosesan data.
2. Dengan menggunakan fungsi filter, kita dapat menyaring panggilan darurat berdasarkan kriteria tertentu, memastikan bahwa data yang dianalisis adalah yang paling relevan. Hal ini meningkatkan akurasi analisis dan responsibilitas sistem.
3. Fungsi reduce memungkinkan pengurangan kompleksitas data dengan mengagregasi informasi menjadi bentuk yang lebih sederhana dan dapat dipahami. Ini memudahkan untuk mengidentifikasi tren dan pola penting dalam data panggilan darurat.
4. Dengan menerapkan fungsi-fungsi ini, sistem Layanan Panggilan Darurat 911 dapat menjadi lebih responsif terhadap situasi darurat. Analisis yang lebih cepat dan akurat memungkinkan respons yang lebih tepat waktu dan efektif terhadap panggilan darurat.
5. Proses iteratif dalam penerapan fungsi map, filter, dan reduce memungkinkan untuk penyesuaian dan perbaikan yang berkelanjutan. Ini memungkinkan sistem untuk tetap responsif terhadap perubahan kebutuhan dan kondisi.

6. Penerapan fungsi map, filter, dan reduce membawa manfaat yang signifikan dalam pengolahan data Layanan Panggilan Darurat 911, meningkatkan efisiensi, akurasi, dan responsibilitas sistem secara keseluruhan. Diharapkan penggunaan yang terus-menerus dari konsep ini akan membawa dampak positif yang berkelanjutan dalam layanan darurat masyarakat.
7. Penerapan fungsi map, filter, dan reduce dalam manipulasi data layanan panggilan darurat 911 membantu mengoptimalkan pengolahan data, meningkatkan keterbacaan kode, dan menghasilkan informasi yang berguna dalam pemahaman dan analisis lebih lanjut tentang layanan tersebut.