**Revolusi Data untuk Respons Polisi Detroit: Pengoptimalan melalui Analisis Python**

Natasya Ega Lina Marbun1, Khusnun Nisa2, Presilia3, Irvan Alfaritzi4, Syalaisha Andini Putriansyah5

Program Studi Sains Data, Fakultas Sains, Institut Teknologi Sumatera.

*Email*:[*natasya.122450024@student.itera.ac.id*](mailto:natasya.122450024@student.itera.ac.id)*,* [*khusnun.122450078@student.itera.ac.id*](mailto:khusnun.122450078@student.itera.ac.id)*,* [*presilia.122450081@student.itera.ac.id*](mailto:presilia.122450081@student.itera.ac.id)*,* [*irvan.122450093@student.itera.ac.id*](mailto:irvan.122450093@student.itera.ac.id)*,* [*syalaisha.122450111@student.itera.ac.id*](mailto:syalaisha.122450111@student.itera.ac.id)

1. **Pendahuluan**

Detroit adalah kota terbesar di negara bagian Michigan, Amerika Serikat. Terletak di wilayah Midwest, Detroit memiliki sejarah yang kaya dan beragam, terutama dikenal sebagai pusat industri otomotif dunia. Namun kota ini juga terkenal dengan tingkat kriminalitasnya yang telah menjadi salah satu isu utama selama beberapa dekade. Kota ini dikenal memiliki tingkat kejahatan yang tinggi, terutama selama paruh kedua abad ke-20 dan awal abad ke-21, meskipun ada upaya yang signifikan untuk mengurangi angka kejahatan dalam beberapa tahun terakhir.

Kepolisian Detroit bertanggung jawab untuk menegakkan hukum dan menjaga ketertiban di kota. Badan tersebut memiliki berbagai unit, termasuk patroli, detektif, SWAT, dan unit khusus lainnya untuk menangani kejahatan serius dan terorganisir. Kemudian Kepolisian Detroit juga berkoordinasi dengan call center 911, yaitu pusat komunikasi darurat yang melayani masyarakat dalam situasi darurat yang membutuhkan respon cepat dari layanan penegakan hukum, medis, atau pemadam kebakaran. Di Amerika Serikat, sistem *call center* 911 adalah jalur darurat yang digunakan untuk menghubungi berbagai layanan darurat melalui satu nomor telepon yang mudah diingat.

Laporan ini akan membahas tentang penggunaan analisis python untuk pengoptimalisasian data, terhadap respon layanan *call center* 911 dalam tiga puluh hari terakhir. Pembahasan utamanya akan mengenai rata-rata waktu respon, rata-rata waktu pengiriman bantuan, dan rata-rata total waktu, berdasarkan area lingkungannya.

1. **Metode**
   1. **Lambda**

Fungsi lambda dalam Python adalah fungsi anonim (fungsi tanpa nama) yang didefinisikan menggunakan kata kunci lambda (J. Zelle, 2003). Fungsi ini bisa memiliki sejumlah argumen, tetapi hanya bisa memiliki satu ekspresi. Biasanya digunakan untuk tugas-tugas sederhana dan sering digunakan bersama fungsi seperti map(), filter(), dan reduce().

* 1. **Map**

Fungsi map menerapkan fungsi tertentu pada setiap item dari iterable (seperti list) dan mengembalikan peta objek (map object) (Ramalho, L., 2015). Fungsi ini berguna untuk melakukan operasi yang sama pada semua elemen dalam iterable.

* 1. **Filter**

Fungsi filter menerapkan fungsi tertentu pada setiap item dari iterable dan hanya mengembalikan item yang memenuhi kondisi tertentu (Beazley, D., 2019). Ini berguna untuk menyaring elemen-elemen dari iterable berdasarkan kriteria yang ditentukan.

* 1. **Reduce**

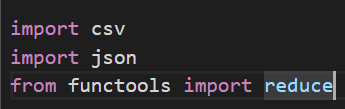
Fungsi reduce dari modul functools menerapkan fungsi tertentu secara berulang (cumulative) pada item-item dari iterable, dari kiri ke kanan, sehingga mengurangi iterable menjadi satu nilai tunggal (Bader, D., 2017). Ini berguna untuk operasi agregasi seperti penjumlahan atau perkalian semua elemen dalam iterable.

* 1. **List of Dictionaries**

List of dictionaries adalah struktur data dalam Python yang terdiri dari list yang berisi beberapa dictionaries (Zhang, H., & Chakraborty, D., 2020). Setiap dictionary adalah koleksi pasangan key-value, dan list dapat berisi banyak dictionary.

1. **Pembahasan**
   1. **Bagian I: Model Populasi Polisi Detroit**

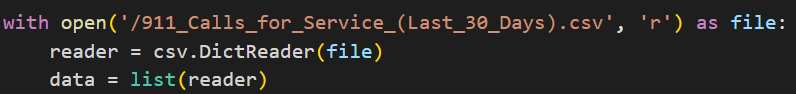
* **Impor Modul**

****

**Gambar 1. Impor Modul**

Langkah pertama dalam membuat program yaitu mengimpor modul. Dalam fungsi ini modul yang digunakan adalah csv, json, dan reduce.

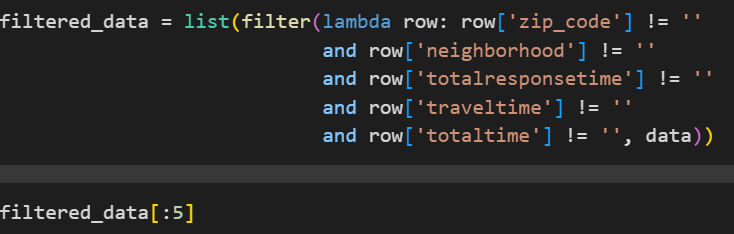
* **Impor Dataset**



**Gambar 2. Impor Dataset**

Pada gambar 2 kita mulai dengan ‘open’ digunakan untuk membuka file. Lalu kita masukkan path file yang menunjuk ke file csv.

* Mode ‘r’ menunjukkan bahwa file dibuka dalam mode baca.
* ‘With’ yaitu konteks yang memastikan file ditutup dengan benar setelah blok selesai dieksekusi, bahkan jika error.
* ‘csv.DictReacer(file), membaca file CSV dan setiap barisnya diubah menjadi dictionary, dimana kunci dictionary adalah nama kolom dari CSV dan nilai adalah data di baris tersebut.
* ‘list(reader)’ mengiterasi seluruh dari ‘reader’ dan mengubahnya menjadi list.
* ‘Data’ adalah list dari dictionary, dimana setiap dictionary merepresentasikan satu baris di file CSV.
* **Filter Data Kosong pada Kolom Zip Code atau Neighborhood**



**Gambar 3. Filter Data Kosong**

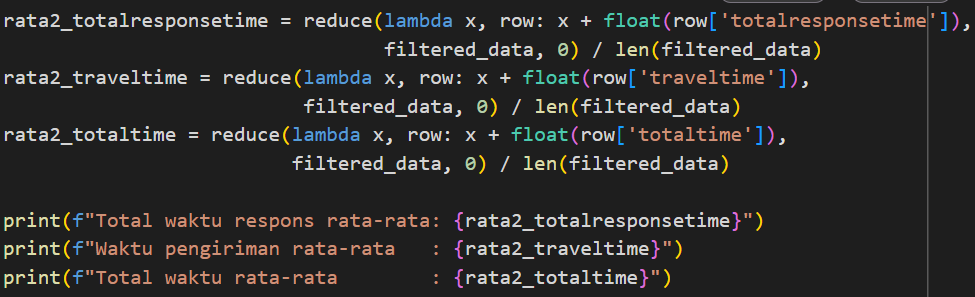
Kode ini digunakan untuk menyaring data CSV yang telah dibaca, dengan memastikan hanya menyertakan baris dimana semua kolom yang disebutkan (‘zip code’, ‘neighborhood’, ‘totalrespontime’, ‘traveltime’ dan ‘totaltime’ tidak kosong. Ini adalah langkah penting dalam memastikan integritas data sebelum analisis lebih lanjut. Berikut penjelasan kode pada gambar 3:

* ‘Filter’ yaitu fungsi bawaan python yang digunakan untuk menyaring elemen dalam iterable berdasarkan kondisi yang diberikan dan juga fungsi ini diterapkan pada setiap elemen dalam iterable, dan hanya elemen yang menghabiskan ‘true’ akan dimasukkan ke dalam hasil.
* ‘List’ mengubah hasil dari ‘filter’ menjadi list
* ‘Filtered\_data[:5]’ digunakan untuk mengambil 5 elemen pertama dari list yang berguna jika ingin melihat sebagian kecil dari data yang telah difilter. Elemen pertama dari objek filtered\_data dapat dilihat pada gambar 4.



**Gambar 4. List dari data yang difilter**

* **Menghitung Total Waktu Respons Rata-Rata, Waktu Pengiriman Rata-Rata, dan Total Waktu Rata-Rata untuk Kepolisian Detroit**

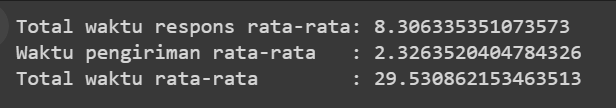
****

**Gambar 5. Menghitung waktu respon, waktu pengiriman dan waktu kepolisian detroit**

Pada gambar 5 dapat bahwa:

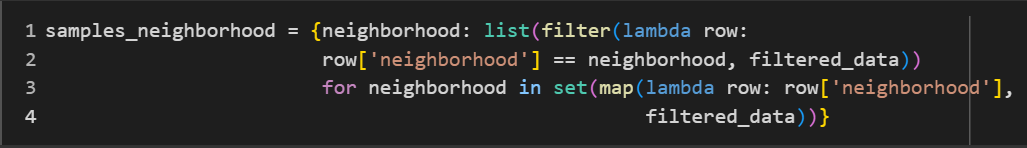
* Rata-rata\_totalresponsetime dibagian ‘lambda x, row: x + float[‘totalresponsetime’] yaitu fungsi lambda yang menambahkan nilai ‘totalresponsetime’ dari setiap baris ke akumulator(‘x’)
* ‘Filtered\_data’ yaitu iterable yang berisi data yang telah difilter.
* ‘0’ adalah nilai awal dari akumulator.
* Hasil penjumlahan dibagi dengan ‘len(filtered\_data)’ untuk mendapatkan rata-rata.
* ‘Rata2\_traveltime’ sama seperti diatas, tetapi untuk menghitung rata-rata.
* ‘Rata2\_totaltime’ sama seperti diatas, tetapi untuk menghitung rata-rata ‘totaltime.
* ‘Print’ digunakan untuk menampilkan hasil rata-rata yang telah dihitung.

Dari penjelasan tersebut, kode pada Gambar 4 bertujuan untuk menghitung total dari rata-rata waktu respon, pengiriman dan total waktu dalam data yang telah difilter menggunakan fungsi ‘reduce’. Hasil atau output pada kode tersebut dapat dilihat pada Gambar 5.



**Gambar 6. Hasil perhitungan dari objek filtered\_data**

* 1. **Bagian II: Modelkan Neighborhood Samples**
* **Memisahkan data berdasarkan Neighborhood**

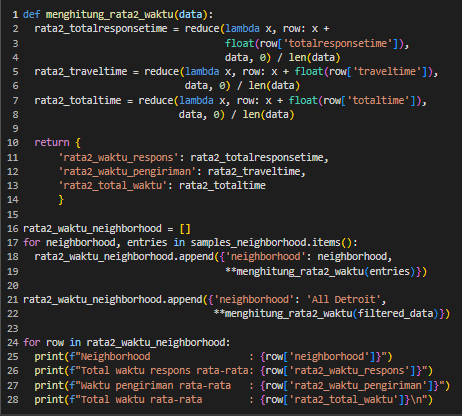
****

**Gambar 7. Pembuatan dictionary samples\_neighborhood**

Kode pada Gambar 7. berfungsi untuk membuat *dictionary* dengan kunci berupa setiap nilai unik pada kolom ‘Neighborhood’ dan nilainya berupa baris data yang memiliki ‘neighborhood’ tersebut. Berikut penjelasan langkah-langkah dari setiap baris kode:

1. Pembuatan set unik: membuat set yang berisi semua nilai unik dari kolom ‘neighborhood’ dalam objek filtered\_data.
2. Pengelompokkan data: memilih baris-baris dari filtered\_data yang memiliki nilai ‘neighborhood’ yang sama dengan nilai unik tersebut.
3. Pembuatan kamus: baris-baris yang terfilter untuk setiap ‘neighborhood’ dikonversi menjadi daftar dan ditetapkan sebagai nilai yang sesuai dengan kunci ‘neighborhood’ kemudian disimpan dalam objek samples\_neighborhood.

* **Menghitung total waktu respon, waktu pengiriman rata-rata dan total waktu rata-rata untuk setiap neighborhood**

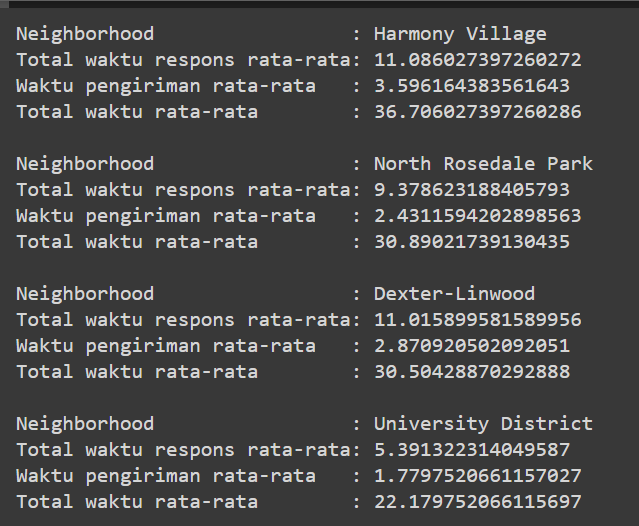
****

**Gambar 8. Kode operasi perhitungan setiap neighborhood**

Kode pada Gambar 8 akan melakukan perhitungan rata-rata waktu respons, waktu pengiriman, dan total waktu untuk setiap ‘neighborhood’ dan juga untuk seluruh data. Berikut penjelasan langkah-langkah dari setiap baris kode:

1. Fungsi menghitung\_rata2\_waktu(data): menghitung rata-rata waktu respons, waktu pengiriman, dan total waktu dari data yang diberikan dengan menjumlahkan nilai-nilai dari kolom tertentu dan dibagi dengan jumlah baris data.
2. Pengelompokan data: pada setiap ‘neighborhood’ dalam samples\_neighborhood, fungsi menghitung\_rata2\_waktu(entries) dipanggil untuk menghitung rata-rata waktu untuk setiap ‘neighborhood’. Hasilnya ditambahkan ke daftar rata2\_waktu\_neighborhood.
3. Perhitungan rata-rata untuk seluruh data: memanggil fungsi menghitung\_rata2\_waktu(filtered data). Hasil dari pemanggilan fungsi tersebut disimpan kedalam rata2\_waktu\_neighborhood dengan ‘neighborhood’ ditetapkan sebagai ‘All Detroit’.
4. Manampilkan hasil: mencetak ‘neighborhood’ dan rata-rata waktu respons, waktu pengiriman, dan total waktu.

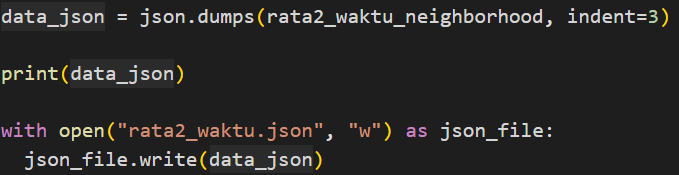
Berikut output kode dari Gambar 8:



**Gambar 9. Hasil Perhitungan setiap neighborhood**

Setiap ‘neighborhood’ memiliki tiga nilai yang menunjukkan rata-rata waktu respons, waktu pengiriman, dan total waktu. Nilai-nilai ini dihitung dari data yang terkait dengan masing-masing ‘neighborhood’ dan juga untuk keseluruhan data yang ditandai sebagai ‘All Detroit’. Perhitungan ini berfungsi untuk membandingkan performa waktu antar ‘neighborhood’ atau untuk melihat gambaran keseluruhan performa waktu di semua area.

* 1. **Bagian III: Buat File Output JSON**
* Menggunakan modul json, format list dictionary sebagai json

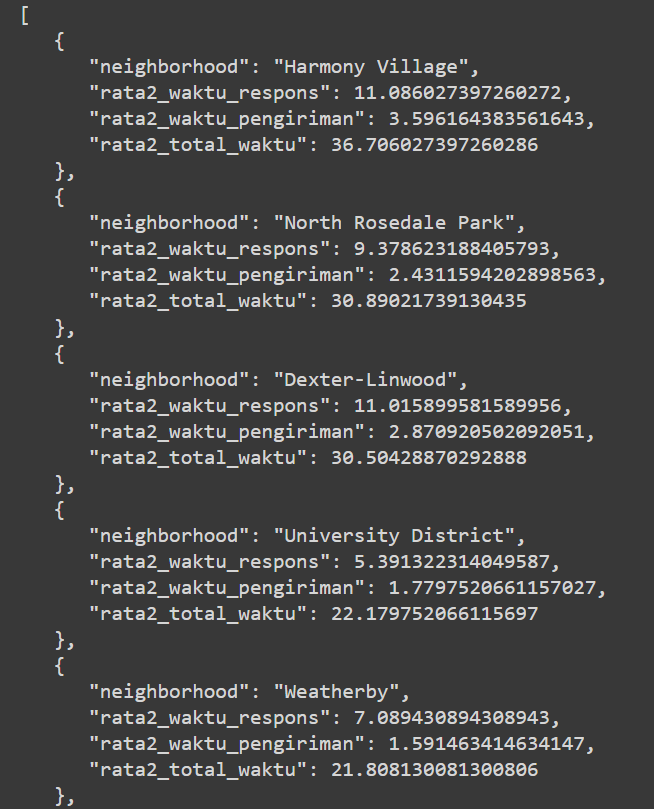
****

**Gambar 10. Menggunakan modul json**

Pada gambar dapat dijelaskan bahwa:

* ‘Json.dumps’ mengonversi objek python menjadi string JSON.
* ‘indent=3’ menambahkan indentasi sebesar 3 spasi untuk membuat output JSON lebih mudah dibaca.
* ‘Print’ digunakan untuk menampilkan string JSON yang telah dihasilkan ke console yang berguna untuk melihat hasil konversi sebelum menyimpannya ke file.
* ‘With open……as json\_file:’ membuka file bernama ‘rata2\_waktu.json’. jika file tersebut tidak ada, maka akan dibuat; jika file sudah ada isinya, akan ditimpa.
* File ‘rata2\_waktu.json’ akan berisi data yang sama dengan output.

Berikut output kode dari Gambar 10.

  
**Gambar 11. Hasil konversi File**

Output merupakan objek data\_json yang berisi hasil konversi dari setiap atribut pada objek rata2\_waktu\_neighborhood. Konversi data ke format JSON dilakukan untuk memudahkan pertukaran data antar aplikasi yang berbeda.

1. **Kesimpulan**

Dengan demikian, dapat disimpulkan bahwa Analisis data panggilan darurat 911 di Detroit selama tiga puluh hari terakhir menggunakan Python bertujuan untuk memahami kinerja layanan respon. Dengan menggunakan fungsi-fungsi Python, seperti lambda, map, filter, dan reduce, data diproses untuk mengevaluasi rata-rata waktu respons, waktu pengiriman bantuan, dan total waktu respon berdasarkan lingkungan. Hasil analisis disajikan dalam struktur data yang memungkinkan identifikasi area-area yang memerlukan perbaikan untuk meningkatkan efisiensi respons dalam situasi darurat di Detroit.

1. **Daftar Pustaka**

Bader, D. (2017). Python Tricks: A Buffet of Awesome Python Features. Dan Bader Press.

Beazley, D. (2019). Python Essential Reference (4th ed.). Addison-Wesley Professional.

J. Zelle (2003). "Python Programming: An Introduction to Computer Science." 1st ed. Franklin, Beedle & Associates Inc.,Generator Expressions. Journal of Computational Methods in Science and Engineering, 20(3), 321-334.

Ramalho, L. (2015). Fluent Python: Clear, Concise, and Effective Programming (1st ed.). O'Reilly Media.

Zhang, H., & Chakraborty, D. (2020). Efficiently Generating Time Series Data with Python