“Implementasi Fungsi Map dan Lambda dalam Analisis Sentimen : Studi Kasus pada Penghitungan Jumlah kata Positif dan Negatif dalam Aplikasi Analisis Sentimen”

Kelompok 7

[1]Aditya Rahman
122450113

[2]Farahanum Afifah Ardiansyah
122450056

[3]Fayyaza Aqila Syafitri Achjar
122450131

[4]Azizah Kusumah Putri
122450068

[5]Eggi Satria
122450032

[6]Meira 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 : [1]*aditya.122450113@student.itera.ac.id*, [2]*farahanum.122450056@student.itera.ac.id*,
[3]*fayyaza.122450131@student.itera.ac.id*, [4]*azizah.122450068@student.itera.ac.id*,
[5]*eggi.122450032@student.itera.ac.id*, [6]*meira.122450011@student.itera.ac.id*

1. PENDAHULUAN

1.1 Latar Belakang

Analisis sentimen adalah proses mengidentifikasi, mengekstraksi atau mengidentifikasi sentimen yang terkandung dalam teks. Dengan meningkatnya penggunaan media sosial dan platform online lainnya, analisis sentimen menjadi semakin penting untuk memahami pandangan, opini, dan emosi pengguna terhadap suatu topik, produk, layanan, atau merek tertentu.

Salah satu pendekatan analisis sentimen adalah dengan menghitung jumlah kata positif dan negatif dalam teks. Hal ini dapat dilakukan dengan menggunakan algoritma yang memetakan kata-kata tertentu ke dalam kategori positif atau negatif dan kemudian menghitung jumlah kemunculan kata-kata tersebut dalam teks yang dianalisis..

1.2 Rumusan Masalah

a. Bagaimana mengimplementasikan fungsi map dan lambda dalam penghitungan jumlah kata positif dan negatif dalam sebuah teks pada aplikasi analisis sentimen?
b. Seberapa efektif dan efisien penggunaan fungsi map dan lambda dalam memproses dan mengklasifikasikan kata-kata positif dan negatif dalam analisis sentimen dibandingkan dengan metode lainnya

1.3 Tujuan

Mengimplementasikan fungsi map dan lambda dalam analisis Sentimen untuk mengetahui kata kata positif dan negatif dalam sebuah kalimat ataupun teks

2. METODE

2.1 Fungsi Lambda

Fungsi lambda adalah fungsi mini satu baris yang digunakan untuk menggantikan fungsi sederhana. Sayangnya, fungsi lambda hanya dapat memuat satu ekspresi saja dan tidak dapat digunakan untuk menggantikan fungsi yang kompleks dan rumit (Anam et al., n.d., #). Struktur fungsi lambda :

lambda parameter1, parameter2 : ekspresi

Penulisan lambda diawal adalah sebuah keharusan agar program mengetahui bahwa fungsi yang dimasukkan menggunakan fungsi lambda sedangkan parameter1 dan parameter2 dapat diubah sesuai kebutuhan asalkan tidak melanggar aturan penamaan variabel. Begitupun dengan ekspresi dapat diubah sesuai dengan kebutuhan user.

Misalkan, untuk menghitung luas persegi panjang menggunakan fungsi lambda maka bentuk fungsi nya adalah :

lambda panjang, lebar : panjang *lebar

fungsi lambda juga dapat dimasukkan ke dalam suatu variabel sehingga bentuknya menjadi :

luas_persegipanjang = lambda panjang, lebar : panjang *lebar

dengan ini maka ’luas_persegipanjang’ bukan lagi sebuah variabel melainkan sebuah fungsi yang dapat dipanggil dan memiliki parameter ’panjang’ dan ’lebar’.

2.2 Fungsi Map

Fungsi map merupakan fungsi *built-in* dalam bahasa pemrograman python yang digunakan untuk memproses dan mengubah setiap item yang ada ke dalam perulangan tanpa menggunakan *loop*. Cara kerja fungsi ini adalah dengan menerapkan fungsi ke dalam setiap item dan melakukan pembaruan pada variabel nya yang berbentuk *list* (Ramos, n.d.). Struktur fungsi map :

map(function, iterable)

’function’ dapat diubah menjadi fungsi apapun yang kemudian akan diterapkan ke dalam setiap item yang di-inisiasikan dalam parameter ’iterable’. Sedangkan ’iterable’ adalah data yang nantinya akan dilakukan perulangan di dalamnya.

Untuk menggunakan fungsi ini *user* harus menginisiasikan fungsi yang akan digunakan pada parameter ’function’ terlebih dahulu serta memuat data yang akan diulang dan nantinya akan diletakkan pada parameter ’iterable’. Adapun bentuk parameter ’iterable’ yang digunakan adalah list.

2.3 Modul ’re’

Salah satu modul *built-in* yang dimiliki oleh pemrograman python adalah modul ‘re’. Modul ini menyediakan operasi yang dapat digunakan untuk melakukan pencocokan (Python, n.d.). Dalam modul ’re’ terdapat 4 fungsi utama, yaitu :

- findall() : untuk mengembalikan kecocokan dalam bentuk *list*.
- search() : akan mengembalikan object ’Match’ jika ada kecocokan dalam *string* yang dimasukkan.

- split() : hasilnya adalah *list* yang telah membagi masukan *string* ke dalam tiap kecocokan.
- sub() : akan menggantikan kecocokan pada masukkan string.

2.4 Modul 'functools'

Sama seperti modul 're', modul 'functools' juga merupakan modul *built-in* python yang digunakan untuk bekerja pada fungsi-fungsi yang dapat digunakan sebagai objek (Das, 2023). Beberapa fungsi utama dalam modul 'functools' adalah :

- partial() : untuk membuat fungsi parsial dari parameter tertentu.
- partialmethod() : definisi metode dari fungsi yang didefinisikan untuk suatu parameter tertentu.
- reduce() : untuk menerapkan fungsi dengan dua parameter secara berulang kali sehingga urutan elemennya berkurang menjadi satu nilai saja.
- cache() : akan membuat 'pembungkus tipis' pada sekitaran pencarian *dictionary* untuk tiap parameter dalam fungsi.
- cached_property() : untuk mengubah metode *class* menjadi properti dengan nilai yang dihitung sekali kemudian di-*cache* sebagai atribut normal.

3. PEMBAHASAN

3.1 Text Preprocessing

Rangkaian pra-pemrosesan teks diawali dengan mengubah seluruh teks menjadi huruf kecil. Hal ini dilakukan untuk membuat teks menjadi konsisten dan tidak ada perbedaan antara huruf besar dan huruf kecil. Tahap berikutnya adalah cleansing, yaitu untuk menghilangkan semua karakter khusus dan tanda baca sehingga hanya tersisa huruf dan spasi. Langkah selanjutnya adalah tokenizing, yaitu membagi teks menjadi beberapa token berdasarkan spasi sebagai pemisah. Kemudian selanjutnya membuat daftar kata dengan kategori kata positif dan kata negatif. Setiap kata tersebut dikelompokan berdasarkan kategori yang sesuai. Berikut ini adalah tabel pengelompokan kata.

Tabel 1 Pengelompokan Kata

No	Kelas Kata		No	Kelas Kata	
	Positif	**Negatif**		**Positif**	**Negatif**
1	Bagus	Buruk	**9**	Menyenangkan	Menyebalkan
2	Mudah	Susah	**10**	Sukai	Sulit
3	Suka	Tidak	**11**	Asik	Sukar
4	Positif	Negatif	**12**	Terbaik	Benci
5	Keren	Jelek	**13**	Terfavorit	Tidak suka
6	Mantap	Wah	**14**	Favorit	Menyedihkan
7	Puas	Mengecewakan	**15**	Hebat	Jijik
8	Baik	Kecewa	**16**	Senang	Menjengkelkan

3.2 Menghitung sentimen

Gambar 1. Fungsi menghitung sentimen

```
# using lambda function and map to count positive and negative words in the tokens
check_positive = lambda kata: kata in positive_kata
check_negative = lambda kata: kata in negative_kata
count_sentiment = lambda fungsi, tokens: sum(map(fungsi, tokens))
return count_sentiment(check_positive, tokens), count_sentiment(
    check_negative, tokens
)
```

Fungsi **menghitung_sentiment** digunakan untuk melakukan penghitungan jumlah kata negatif maupun kata positif yang terdapat pada sebuah teks. Fungsi tersebut meliputi pendefinisian terhadap variabel, meliputi **positive_kata** sebagai kata positif dan **negative_kata** sebagai kata negatif. Selanjutnya terdapat pendefinisian terhadap fungsi, dimana fungsi **check_positive** merupakan fungsi lambda yang akan mengembalikan **True** apabila adanya kata dalam **positive_kata**, dan akan mengembalikan **False** apabila tidak. Lalu ada fungsi **check_negative** yang merupakan fungsi lambda akan mengembalikan True apabila adanya kata dalam **negative_kata** dan akan mengembalikan **False** jika tidak. Selanjutnya fungsi **count_sentiment** yang digunakan untuk menerima argumen berupa daftar token, dan menghitung jumlah kata yang memenuhi fungsi tersebut. Fungsi **menghitung_sentiment** ini menggunakan fungsi map yang berguna sebagai iterator untuk mengembalikan hasil ke fungsi **check_positive** dan **check_negative** pada setiap token yang terdapat dalam teks.

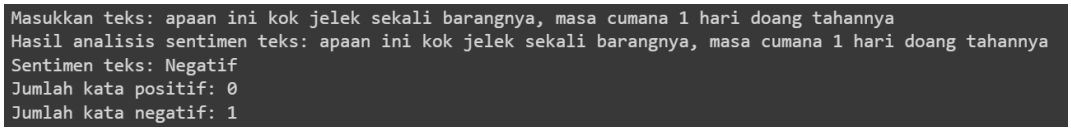
3.3 Analisis Sentimen

Gambar 2. Fungsi menganalisis sentimen

```
67
68 def analyze_sentiment(text):
69     """
70     Menganalisis sentimen teks dan mengembalikan hasilnya.
71     """
72
73     # Preprocessing
74     tokens = ft.reduce(lambda x, y: y(x), dataPreprosesing_list, text)
75
76     positive_count, negative_count = menghitung_sentiment(tokens)
77
78     if positive_count > negative_count:
79         return "Positif", positive_count, negative_count
80     elif negative_count > positive_count:
81         return "Negatif", positive_count, negative_count
82     else:
83         return "Netral", positive_count, negative_count
84
85
86 # Contoh penggunaan
87 text = input("Masukkan teks: ")
88 sentiment, positif_count, negatif_count = analyze_sentiment(text)
89 print(f"Hasil analisis sentimen teks: {text}")
90 print(f"Sentimen teks: {sentiment}")
91 print(f"Jumlah kata positif: {positif_count}")
92 print(f"Jumlah kata negatif: {negatif_count}")
```

Fungsi **analyze_sentiment** pada kode bertujuan untuk menerima teks sebagai input dengan maksud untuk melakukan preprocessing terhadap teks menggunakan serangkaian fungsi dalam **dataPreprocessing_list.** Setelah itu, fungsi menghitung jumlah token yang mengandung sentimen positif dan negatif. Jika jumlah token positif lebih besar dari jumlah token negatif, fungsi akan mengembalikan hasil “Positif” bersama dengan jumlah positif dan negatif. Sebaliknya, jika jumlah token negatif lebih besar dari jumlah token positif, hasilnya akan “Negatif”. Jika keduanya menghasilkan token yang sama, maka hasilnya akan “Netral”.

Gambar 3. Keluaran Output



Teks yang dimasukkan pengguna mungkin menggambarkan ketidakpuasan terhadap suatu barang atau pengalaman, dengan menggambarkan barang tersebut sebagai jelek dan memiliki masa pakai yang singkat. Dengan tidak adanya kata kata positif yang terdeteksi dalam teks, dan dengan adanya satu kata negatif yang teridentifikasi, dan menunjukan bahwa teks tersebut memiliki sentimen negatif yang dominan. Oleh karena itu, dapat disimpulkan bahwa penggunaan merasa kecewa atau tidak puas terhadap barang atau pengalaman yang diungkapkan dalam teks.

4. KESIMPULAN

Jika ditinjau dari pembahasan dan kode yang dibuat dapat disimpulkan bahwa, implementasi fungsi map dan lambda dalam analisis sentimen berhasil mengidentifikasikan kata-kata positif dan negatif dalam teks secara efektif. Fungsi map digunakan untuk mengiterasi melalui setiap token dalam teks serta mengaplikasikan fungsi yang ditentukan pada setiap elemen. Sedangkan, fungsi lambda digunakan untuk mendefinisikan fungsi tanpa nama secara langsung di dalam pemanggilan fungsi map, yang akan mengurangi kompleksitas kode. Dengan adanya implementasi ini, memungkinkan analisis sentimen yang efisien dan mudah dipahami, dengan kode yang ringkas dan dapat diterapkan secara cepat dalam berbagai aplikasi. Implikasi dari penelitian ini menunjukkan bahwa penggunaan fungsi map dan lambda dapat menjadi tambahan yang bernilai dalam aplikasi analisis sentimen.

DAFTAR PUSTAKA

Anam, S., Widhiatmoko, F., Fitriah, Z., Yanti, I., Sa'adah, U., & Guci, A. N. (n.d.). *Pengantar Algoritma dan Pemrograman dengan Python*. Universitas Brawijaya Press. https://play.google.com/books/reader?id=hnXoEAAAQBAJ&pg=GBS.PR4&hl=en

Das, M. (2023, June 7). *Functools: Fun With Functools Module in Python 📈 | by Manoj Das*. Medium. Retrieved March 6, 2024, from https://medium.com/@HeCanThink/functools-fun-with-functools-module-in-python-6316ca2a77ca

Python. (n.d.). *re — Regular expression operations — Python 3.12.2 documentation*. Python Docs. Retrieved March 6, 2024, from https://docs.python.org/3/library/re.html

Ramos, L. P. (n.d.). *Python's map(): Processing Iterables Without a Loop – Real Python*. Real Python. Retrieved March 6, 2024, from https://realpython.com/python-map-function/