

MENYARING KATA DENGAN SYNTACTIC SUGAR

Febiya Jomy Pratiwi, Alvia Asrinda Br.Ginting
Farrel Julio Akbar, Akmal Faiz Abdillah
Daffa Ahmad Naufal

Program Studi Sains Data Institut Teknologi Sumatera
Jl. Terusan Ryacudu, Way Huwi, Kec. Jatiagung, Kabupaten Lampung Selatan
Lampung 35365

Email:

febiya.122450074@student.itera.ac.id, alvia.122450077@student.itera.ac.id
farrel.122450110@student.itera.ac.id, akmal.122450114@student.itera.ac.id
daffa.122450137@student.itera.ac.id

I. Pendahuluan

A. Latar Belakang

Penyaringan kata merupakan prosedur standar dalam pengolahan teks. Ketika kita mempunyai daftar kata-kata yang ingin kita hilangkan dari teks, misalnya kata kata yang tidak relevan. Proses ini seperti memisahkan kulit beras dari beras saat proses penggilingan. Kita hanya akan menyisakan informasi yang benar-benar diperlukan saja.

Pada zaman digital seperti saat ini, dengan pengembangan perangkat lunak kita bisa menyaring kata dengan menggunakan teknik-teknik yang mengandalkan penulisan kode yang kompleks. Seperti, penggunaan regular expression, yaitu pola string yang digunakan untuk mencocokkan dan mengidentifikasi kata kata tertentu. Akan tetapi, penggunaan regular expression akan rumit bagi seseorang yang baru belajar.

Disini kita akan membahas penggunaan syntactic sugar, yaitu sebuah gagasan dalam dunia pemrograman yang bisa mengurangi kompleksitas dan akan menjadi lebih mudah bagi seseorang yang baru belajar. Gula sintaksis merupakan aspek dalam bahasa pemrograman yang diciptakan untuk mempermudah penulisan kode dan membuatnya lebih mudah dipahami tanpa mengurangi kemampuan bahasa tersebut. Dengan memanfaatkan konsep syntactic sugar, proses penyaringan kata dapat menjadi lebih sederhana dan lebih mudah dipahami, bahkan oleh seseorang yang baru belajar.

B. Tujuan

Adapun tujuan dari laporan ini adalah sebagai berikut, yaitu:

1. Memperkenalkan konsep syntactic sugar dalam penyaringan kata

2. Menyerderhanakan proses penyaringan kata-kata sehingga lebih mudah dimengerti oleh pengembang
3. Meningkatkan efesiensi penulisan kode dalam penyaringan kata
4. Menguraikan tahapan penyaringan kata menggunakan syntactic sugar pada pemrograman

II Metode

A. Fungsi Lambda

Ekspresi lambda, juga dikenal sebagai fungsi anonim, adalah sebuah fungsi tanpa nama. Ekspresi lambda digunakan untuk membuat fungsi kecil yang hanya terdiri dari satu baris kode. Salah satu karakteristik dari ekspresi lambda adalah kemampuannya untuk mengembalikan nilai. Untuk informasi lebih lanjut mengenai penggunaan fungsi di Python, silakan baca panduan cara menggunakan fungsi di Python. [1]

B. Fungsi Split

Fungsi string split digunakan untuk memisahkan string atau baris menjadi daftar kata dengan menggunakan string pembatas seperti karakter koma (.). Split string umum digunakan dalam pengolahan data yang dihasilkan oleh berbagai alat database. Selain itu, fungsi split string juga dapat digunakan untuk membuat substring berdasarkan pembatas, mengubah huruf menjadi huruf kecil atau besar, melakukan pengecekan konten string (baik angka maupun huruf), serta menghapus spasi di ujung string. [2]

C. Fungsi Join

Metode join() digunakan untuk menggabungkan elemen-elemen iterable dengan string yang ditentukan, menghasilkan string baru. Selain itu, metode ini juga dapat digunakan untuk menggabungkan dua atau lebih tabel dengan mencocokkan nilai dari masing-masing tabelnya. [3]

D. Fungsi Lower

Metode lower() berfungsi untuk mengubah kata atau string menjadi huruf kecil guna menghilangkan kapitalisasi. Proses ini tidak memodifikasi string atau kata asli, melainkan mengembalikan string atau kata baru dengan seluruh karakter dalam huruf kecil. [4]

III. Pembahasan

A. Analisis Program

Memfilter kata menggunakan "syantatic sugar" mengacu pada penggunaan fitur atau sintaksis dalam bahasa pemrograman yang menyediakan cara yang lebih ringkas atau mudah dibaca dalam melakukan tugas tertentu, seperti memfilter kata dalam kalimat.

Dalam konteks Python, penggunaan "syntatic sugar" dapat merujuk pada pendekatan penulisan kode yang lebih ringkas atau elegan, terutama menggunakan fitur seperti ekspresi lambda, buku pemahaman nama, atau fitur lain yang membuat kode lebih mudah atau lebih mudah untuk digunakan.

```
1 def filter_words(words):
2     return lambda x: ' '.join(word for word in x.split() if word.lower() not in words)
3
4 filter_out = filter_words(['ini', 'itu', 'dan'])
5 print(filter_out("Ini kalimat untuk diuji")) # Output: kalimat untuk diuji
```

1. Lambda Expression

- Dengan menggunakan ekspresi lambda (*lambda x: ' '.join(word for word in x.split() if word.lower() not in words)*), kita dapat langsung menghasilkan fungsi anonim tanpa mengungkapkan nama fungsinya.
- Kode dibuat lebih pendek dan berfokus pada tugas-tugas yang perlu Anda lakukan di dalamnya.

2. List Comprehension

- Hasil akhir dari bagian ini *' '.join(word for word in x.split() if word.lower() not in words)* adalah pemahaman daftar yang menghasilkan daftar yang disusun.
- Sebuah metode yang disebut Pemahaman Daftar dapat digunakan untuk meringkas hasil operasi yang diterapkan pada elemen lain, dengan fitur yang memungkinkan pemfilteran satu kata dari kata-kata berdasarkan kriteria tertentu.

3. Metode String

- Dengan menggunakan *x.split()*, Anda dapat membagi string (x) menjadi daftar kata yang dipisahkan dengan spasi.

- Gunakan metode string seperti *.lower()* untuk memastikan kata-kata dengan huruf kecil dibandingkan sehingga tidak ada perbedaan ukuran huruf besar-kecil saat memfilter.

IV. Kesimpulan

Penggunaan “syntactic sugar” dalam Python memungkinkan filtrasi kata-kata yang lebih mudah dan lebih efisien, seperti filtrasi dalam daftar. Menggunakan ekspresi lambda, seperti `.join` (kata untuk kata dalam `x.split()` jika `word.lower()` tidak dalam kata) ', memungkinkan fungsi anonim tanpa mendefinisikan nama mereka. Ini membuat kode lebih ringkas dan efektif. Pemahaman daftar melibatkan menggunakan fungsi lambda untuk membandingkan nilai kata berdasarkan kriteria tertentu. Menggunakan metode string seperti `.lower()`, kode dapat membuat daftar dengan ukuran yang lebih kecil, memastikan filter yang lebih akurat.

References

- [1] J. Surya and Efitra, "DASAR-DASAR PEMROGRAMAN DENGAN PYTHON," jambi, PT. Sonpedia Publishing Indonesia, 2023, p. 113.
- [2] M. Lutz, "Learning Python," sebastopol, O'Reilly Media, 2007, p. 147.
- [3] D. . A. B. Prasetyo and Y. A. Susetyo, "Implementasi Information Schema Database Pada Postgre SQL Untuk Pembuatan Tabel Informasi Dengan Menggunakan Python Di PT XYZ," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 9, no. 3, p. 1967, 2022.
- [4] A. K. Fauziyyah and D. H. Gautama , "Analisis Sentimen Pandemi Covid-19 Pada Streaming Twitter," *Jurnal Ilmiah SINUS (JIS)* , vol. 18, no. 2, p. 34, 2020.