Aplikasi Penghitung Bobot Kata

Happy Syahrul Ramadhan (122450013), Nurul Alfajar Gumel (122450127), Eli Dwi Berema(122450064), Dhafin Razaqa Luthfi (122450133), Fadhil Fitra Wijaya(122450082)

1. Pendahuluan

Di era informasi yang tersedia dalam jumlah besar dan dapat terus berkembang dengan cepat, sehingga mengelola dan menganalisis teks menjadi semakin penting. Aspek penting dalam analisis teks, salah satunya adalah pemahaman terhadap signifikansi relatif dari kata-kata yang ada pada teks tersebut. Dengan demikian, aplikasi pembobotan kata menjadi sebuah tools yang sangat berguna.

Aplikasi pembobotan kata menggunakan metode Term Frequency-Inverse Document Frequency (TF IDF) yang dapat memberikan penekanan atau bobot tertentu pada setiap kata pada suatu dokumen teks. Bobot tersebut menunjukkan tingkat pentingnya kata dalam konteks teks yang diberikan. Hal tersebut memungkinkan dilakukannya identifikasi pola, ekstraksi informasi penting dan pemahaman teks yang lebih baik.

Dengan demikian, aplikasi penghitung bobot kata dapat sangat berguna dalam menganalisis teks pada era informasi modern, yang memungkinkan kita untuk mendapatkan wawasan lebih luas dari data teks yang tersedia.

2. Metode

Pada Latihan praktikum Pemrograman Berbasis Fungsi di Modul 2 ini, digunakan metode Term Frequency-Inverse Document Frequency (TF IDF). Metode ini dibuat untuk memberikan pembobotan yang lebih tinggi kepada kata-kata yang lebih berisikan kalimat yang informatif atau penting didalam sebuah dokumen atau korpus.

```
1 from sklearn.feature_extraction.text import TfidfVectorizer
2
3
4 vectorize = TfidfVectorizer()
5 respon = vectorize.fit_transform(data)
6 print(respon)
```

Fungsi dari TF IDF sendiri dapat di import dengan menggunakan modul TfidfVectorizer dari library sklearn, untuk bisa mendapatkan nilai bobot dari setiap katanya yang berdasarkan frekuensi kata tersebut.

```
from functools import reduce

kamus_kata = {}
kamus_kata = reduce(lambda acc, kata: {**acc, kata: acc.get(kata, 0) + 1}, data_kata, {})
print(kamus_kata)
```

Selain menggunakan fungsi TF IDF, digunakan juga fungsi yang di import dari modul functools yaitu fungsi reduce dan fungsi lambda, untuk menentukan frekuensi dari setiap kemunculan kata.

```
import pandas as pd

df = pd.DataFrame(respon.todense().T,
                  index = stop_word,
                  columns=['Bobot Kata'])
df = df.reset_index().rename(columns={'index': 'Data Kata'})
print(df)
```

Library pandas untuk memudahkan kita dalam merepresentasikan matriks yang diubah dari bentuk ke bentuk agar menjadi lebih rapi dan dapat melakukan transpose matriks agar dapat mengubah keterangan dokumen atau keterangan kalimat menjadi kolom.

3. Pembahasan

3.1 *Library*

Dalam pembuatan aplikasi penghitung bobot kata, terdapat tiga library yang digunakan yaitu functools, sklearn, dan pandas. Pada library functools diambil fungsi reduce yang akan melakukan akumulasi data berdasarkan daftar yang diberikan yang kemudian dikembalikan menjadi suatu nilai. Sklearn merupakan library yang berisikan fungsi yang digunakan dalam *machine learning,* pembuatan model statistika, dan *dimensionality reduction* dari library sklearn diambil fungsi TfidfVectorizer yang merupakan formula statistika yang mengubah suatu teks menjadi suatu vektor berdasarkan kata-kata yang terdapat di dalamnya. Pandas merupakan library yang memiliki tujuan utama untuk melakukan analisis dan pengolahan data, pada kasus ini fungsi yang digunakan adalah *DataFrame,* bertujuan untuk merepresentasikan hasil dan melakukan transpose pada data yang dimiliki.

3.2 Pengolahan Data Awal

Aplikasi ini bertujuan untuk memberikan bobot nilai pada kata dalam suatu teks. Sebelum melakukan pembobotan teks yang akan dilakukan pembobotan perlu diolah terlebih dahulu. Dalam hal ini *map function* dan *reduce function* memiliki peranan utama. Berdasarkan data teks yang dimiliki akan dilakukan penyamaan besar dan kecilnya huruf. Hal

ini dilakukan karena bahasa pemrograman python merupakan suatu bahasa pemrograman yang sensitif, sehingga setiap perbedaan kata berdasarkan huruf kapital dapat memiliki nilai yang berbeda. Berikut kode pemrograman berdasarkan *map function:*

```
data = ["""Saat ini di Bandar Lampung sering turun hujan dan banjir sampai mengakibatkan banjir
        beberapa daerah Bandar Lampung banjir terseBut mengakibatkan banyak kerusakan yang disebabkan oleh banjir"""]

data_lower = list(map(str.lower, data)) # menormalisasi data text dengan mengubah semua menjadi lowercase atau huruf kecil semua
data_lower
```

Kemudian hasil teks yang telah dinormalisasikan dari *map function* akan dilakukan pemisahan kata, setiap kata dalam teks akan menjadi suatu data tersendiri yang kemudian akan diakumulasi menggunakan *reduce function.* Sehingga banyak suatu kata yang sama dalam teks tersebut dapat diketahui. Berikut kode pemrograman pemisahan teks menjadi per kata dan perhitungan banyak kata menggunakan *reduce function:*

```
data_kata = [kata for kalimat in data_lower for kata in kalimat.split()]
print(data_kata)
```

```
from functools import reduce

kamus_kata = {}
kamus_kata = reduce(lambda acc, kata: {**acc, kata: acc.get(kata, 0) + 1}, data_kata, {})
print(kamus_kata)
```

Berdasarkan kedua fungsi ini didapatkan hasil kata bahwa terdapat 20 kata dan kata banjir merupakan kata yang paling sering digunakan sebanyak 4.

3.3 Pembobotan Kata

Setelah didapatkan banyak kata dan banyak kata yang digunakan dalam teks, aplikasi ini akan melakukan pemberian bobot dan membuat ranking berdasarkan nilai bobot yang diberikan. Pemberian bobot pada setiap kata yang telah dilakukan pemisahan dapat menggunakan fungsi TfidfVectorizer dari sklearn. Fungsi ini akan merubah nilai dari suatu kata dan seringnya kata tersebut muncul ke dalam vektor. Berikut kode pemrograman TfidfVectorizer:

```
from sklearn.feature_extraction.text import TfidfVectorizer


vectorize = TfidfVectorizer()
respon = vectorize.fit_transform(data)
print(respon)
```

berdasarkan hasil vektorisasi didapatkan suatu nilai yang dapat menjadi acuan dalam melakukan ranking, dengan begitu dapat dilakukan representasi hasil dan mengurutkan

berdasarkan tingkat prioritas data kata tersebut pada teks. Dari hasil pemrograman didapatkan bahwa kata “banjir”, “bandar”, “lampung”, dan “mengakibatkan” merupakan kata yang paling sering muncul dalam teks, dan “banjir” merupakan kata yang paling sering mucul sebanyak 4 kali. Berikut hasil akhir dari pemrograman :

	Data Kata	Bobot Kata
1	banjir	0.603023
0	bandar	0.301511
12	mengakibatkan	0.301511
11	lampung	0.301511
18	turun	0.150756
17	tersebut	0.150756

4. Kesimpulan

Metode Term Frequency-Inverse Document Frequency(TF-IDF) yang digunakan dalam aplikasi ini membuat pengguna dapat menganalisis teks dengan memberikan bobot pada setiap kata,mengidentifikasi pola lalu memperoleh informasi.Dalam penggunaan metode ini terdapat functools utama yang digunakan yaitu:

1.Fungsi lambda, yang berfungsi untuk menentukan frekuensi dari setiap kemunculan kata.

2.Map function dan reduce function yang berfungsi untuk penyamaan besar dan kecilnya huruf.

3.TfidfVectorizer yang berfungsi merubah nilai dari suatu kata berdasarkan tingkat kemunculan suatu kata ke dalam bentuk vektor.

Link google collab :
https://colab.research.google.com/drive/1Pt-jSnc8StD1oqtLdlWdfUAuyfd30CK0?usp=sharing