TUGAS M10
KNOWLEDGE DISCOVERY
TEXT MINING STUDI KASUS

Ni Putu Devira AM
1120800012
S2 Elektro 2020

SOAL

Do iteratively for data1.txt until data50.txt (i=1-50)

1. data ← read text file for data-i
2. keywords ← Apply Tokenizing, Filtering and Stemming/Tagging for the data
3. scores ← Calculate TF and remove keywords those have TF lower than 50% from highest scores, and print it

4. querylist ← give query “pertumbuhan ekonomi, perkembangan pasar dan pergerakan harga saham” (with category “economy”), and apply Tokenizing, Filtering and Stemming/Tagging for the query
5. rankdocs ← apply searching from querylist for the scores of each data, and retrieve 10 data those have highest scores
6. label ← read label.csv
7. recall, precision ← calculate Recall and Precision from rankdocs started from rank 1 until rank 10, with comparing the category of rankdocs with the label of query category
8. Visualize graph of recall and precision

PERALATAN

1. Laptop/PC
2. Software Anaconda (Python)
3. Data set (Berita & Label)

LANGKAH 1 – Membaca Data 1-50

```
import os
import re
import string
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory

os.listdir()
list_file = os.listdir("D:/KULIAH\MASTER/S2 PENS/KULIAH/SEMESTER 2/Sistem Temu Pengetahuan - P Ali Ridho/P7-Studi Kasus Text Mini
for namafile in list_file:
    #if namafile.endswith(".txt"):
    print("\n",namafile,"\n")
    f = open("D:/KULIAH\MASTER/S2 PENS/KULIAH/SEMESTER 2/Sistem Temu Pengetahuan - P Ali Ridho/P7-Studi Kasus Text Mining/textmir
    text=f.read()
    f.close()
    print("\nText:\n----------------\n", text)
```

```
 data1.txt


Text:
----------------
 Harga emas batangan bersertifikat Antam keluaran Logam Mulia PT Aneka Tambang Tbk (ANTM) naik pada hari Selasa (26/5)

Mengutip situs Logam Mulia, harga pecahan satu gram emas Antam berada di Rp 917.000. Harga emas Antam ini naik Rp 1.000 dari h
arga Jumat (22/5) lalu di Rp 916.000.

Sementara, harga pembelian kembali atau buyback emas Antam juga turun Rp 1.000 dan berada di Rp 816.000.

Berikut harga emas batangan Antam dalam pecahan lainnya per hari ini dan belum termasuk pajak:

Harga emas 0,5 gram: Rp 488.500

Harga emas 1 gram: Rp 917.000
```

LANGKAH 2 – Melakukan Tokenizing, Filtering & Stemming Data

```
print("\nText:\n----------------\n", text)
text = text.lower()
text = re.sub(r"\d+", "", text)
text = text.translate(str.maketrans("","",string.punctuation))
text = text.strip()
tokens = word_tokenize(text)
print("\nTokenizing:\n----------------\n", tokens)

# Filtering dengan Sastrawi -------------------
factory = StopWordRemoverFactory()
stopword = factory.create_stop_word_remover()
text = stopword.remove(text)
print("\nSetelah filtering:\n----------------\n", text)

# Stemming dengan Sastrawi -----------------
factory = StemmerFactory()
stemmer = factory.create_stemmer()
text = stemmer.stem(text)
print("\nOutput stemming:\n----------------\n", text)
```

```
Tokenizing:
----------------
 ['harga', 'emas', 'batangan', 'b
m', 'naik', 'pada', 'hari', 'sela
m', 'berada', 'di', 'rp', 'harga'
ara', 'harga', 'pembelian', 'kemb
'berikut', 'harga', 'emas', 'bata
k', 'pajak', 'harga', 'emas', 'gr
am', 'rp', 'harga', 'emas', 'gram
m', 'rp', 'harga', 'emas', 'gram'
mas', 'dan', 'perak', 'batangan',
a', 'harga', 'per', 'gram', 'emas
di', 'karena', 'ada', 'biaya', 't
g', 'kecil', 'lebih', 'mahal', 'd
rga', 'per', 'gram', 'emas', 'bat
```

```
Setelah filtering:
----------------
 harga emas batangan bersertifikat antam keluaran logam mulia pt aneka tambang tbk antm naik hari selasa

mengutip situs logam mulia harga pecahan satu gram emas antam berada rp  harga emas antam naik rp  harga jumat  lalu rp

sementara harga pembelian

berikut harga emas batanga

harga emas  gram rp

harga emas  gram rp

harga emas  gram rp

harga emas  gram rp
```

```
Output stemming:
----------------
 harga emas batang sertifikat antam keluar logam mulia pt aneka tambang tbk antm naik hari selasa kutip situs logam mulia harg
a pecah satu gram emas antam ada rp harga emas antam naik rp harga jumat lalu rp sementara harga beli atau buyback emas antam
turun rp ada rp ikut harga emas batang antam pecah lain per hari masuk pajak harga emas gram rp harga emas gram rp harga emas
gram rp harga emas gram rp harga emas gram rp harga emas gram rp harga emas gram rp harga emas gram rp harga emas gram rp harg
a emas gram rp terang logam mulia antam jual emas perak batang beberapa ukur berat misal gram gram dan gram biasa harga per gr
am emas antam beda gantung berat batang beda jadi ada biaya tambah cetak harga per gram emas antam batang kecil lebih mahal ba
tang lebih besar harga ada sini harga per gram emas batang kilogram biasa jadi patok laku bisnis emas
```

LANGKAH 3 – Menghitung skor Term Frequency (TF)

```
tokens = word_tokenize(text)
print("\nTokenizing:\n----------------\n", tokens)

tf = FreqDist(tokens)
print("\nTerm Frequency:\n----------------\n", tf.most_common())

word, frequency=tf.most_common()[0]
print("\nKeyword yang paling banyak muncul:\n----------------\n", word,"=", frequency , "\n")
threshold=0.5*frequency
print("\nKeseluruhan keywords:\n----------------\n")
for word, frequency in tf.most_common():
    print(word, ":", frequency)
tf.plot(cumulative=False)
plt.show()

print("\nTHRESHOLD\n")
print("threshold = ",threshold)
for word, frequency in tf.most_common():
    if frequency > threshold:
        print(word, ":", frequency)
```

```
Term Frequency:
----------------
 [('harga', 20), ('emas', 20), ('gram', 17), ('rp', 15), ('antam', 8), ('batang', 7), ('ada', 4), ('per', 4), ('logam', 3),
('mulia', 3), ('naik', 2), ('hari', 2), ('pecah', 2), ('berat', 2), ('biasa', 2), ('beda', 2), ('jadi', 2), ('lebih', 2), ('se
rtifikat', 1), ('keluar', 1), ('pt', 1), ('aneka', 1), ('tambang', 1), ('tbk', 1), ('antm', 1), ('selasa', 1), ('kutip', 1),
('situs', 1), ('satu', 1), ('jumat', 1), ('lalu', 1), ('sementara', 1), ('beli', 1), ('atau', 1), ('buyback', 1), ('turun',
1), ('ikut', 1), ('lain', 1), ('masuk', 1), ('pajak', 1), ('terang', 1), ('jual', 1), ('perak', 1), ('beberapa', 1), ('ukur',
1), ('misal', 1), ('dan', 1), ('gantung', 1), ('biaya', 1), ('tambah', 1), ('cetak', 1), ('kecil', 1), ('mahal', 1), ('besar',
1), ('sini', 1), ('kilogram', 1), ('patok', 1), ('laku', 1), ('bisnis', 1)]

Keyword yang paling banyak muncul:
----------------
 harga = 20
```

Dengan Threshold TF = 50%*Freq Tertinggi

```
Keseluruhan keywords:
----------------

harga : 20
emas : 20
gram : 17
rp : 15
antam : 8
batang : 7
ada : 4
per : 4
logam : 3
mulia : 3
naik : 2
hari : 2
pecah : 2
berat : 2
biasa : 2
```

```
THRESHOLD

threshold =  10.0
harga : 20
emas : 20
gram : 17
rp : 15
```

LANGKAH 4 – Memberi Query

```
#QUERYLIST
print("-----------------------------------------------------------------------------------------------------------------------")
f = open("D:/KULIAH/MASTER/S2 PENS/KULIAH/SEMESTER 2/Sistem Temu Pengetahuan - P Ali Ridho/P7-Studi Kasus Text Mining/querylist.t
textt=f.read()
f.close()
textt = textt.lower()
textt = re.sub(r"\d+", "", textt)
textt = textt.translate(str.maketrans("","",string.punctuation))
textt= textt.strip()

tokens = word_tokenize(textt)
print("\nTokenizing:\n----------------\n", tokens)

# Filtering dengan Sastrawi -------------------
factory = StopWordRemoverFactory()
stopword = factory.create_stop_word_remover()
textt = stopword.remove(textt)
print("\nSetelah filtering:\n----------------\n", textt)

# Stemming dengan Sastrawi -----------------
factory = StemmerFactory()
stemmer = factory.create_stemmer()
textt = stemmer.stem(textt)
print("\nOutput stemming:\n----------------\n", textt)
```

```
Tokenizing:
----------------
 ['pertumbuhan', 'ekonomi', 'perkembangan', 'pasar', 'dan', 'pergerakan', 'harga', 'saham']

Setelah filtering:
----------------
 pertumbuhan ekonomi perkembangan pasar pergerakan harga saham

Output stemming:
----------------
 tumbuh ekonomi kembang pasar gera harga saham
```

LANGKAH 5 – Memilih beberapa data dari jumlah yg masuk Querylist terbanyak

```
print("\nTHRESHOLD\n")
print("threshold = ",threshold)
for word, frequency in tf.most_common():
    if frequency >= threshold:
        print(word, ":", frequency)
        if (word=="tumbuh"):
            tumbuh=frequency
        elif (word=="ekonomi"):
            ekonomi=frequency
        elif (word=="kembang"):
            kembang=frequency
        elif (word=="pasar"):
            pasar=frequency
        elif (word=="gerak"):
            gerak=frequency
        elif (word=="harga"):
            harga=frequency
        elif (word=="saham"):
            saham=frequency
t=tumbuh
e=ekonomi
k=kembang
p=pasar
g=gerak
h=harga
s=saham
jumlah=t+e+k+p+g+h+s
```

```
t 0
e 0
k 0
p 0
g 3
h 0
s 5
jumlah 8
```

LANGKAH 6 – Menginputkan label

```
Text:
----------------
 data1,economy
data2,economy
data3,economy
data4,economy
data5,economy
data6,economy
data7,economy
data8,economy
data9,economy
data10,economy
data11,soccer
data12,soccer
data13,soccer
data14,soccer
data15,soccer
data16,soccer
data17,soccer
data18,soccer
data19,soccer
data20,soccer
data21,automotive
data22,automotive
data23,automotive
data24,automotive
data25,automotive
data26,automotive
data27,automotive
data28,automotive
data29,automotive
data30,automotive
data31,phone
data32,phone
```

```
f = open("D:/KULIAH\MASTER/S2 PENS/KULIAH/SEMESTER 2/Sistem Temu Pengetahuan - P Ali Ridho/P7-Studi Kasus Text Mining/label.csv",
text=f.read()
f.close()
print("\nText:\n----------------\n", text)
```

LANGKAH 7 – Menghitung Recall & Precision

```
recall= jumlah / 8
precision=jumlah / 8
print("recall",recall)
print("precision", precision)
```

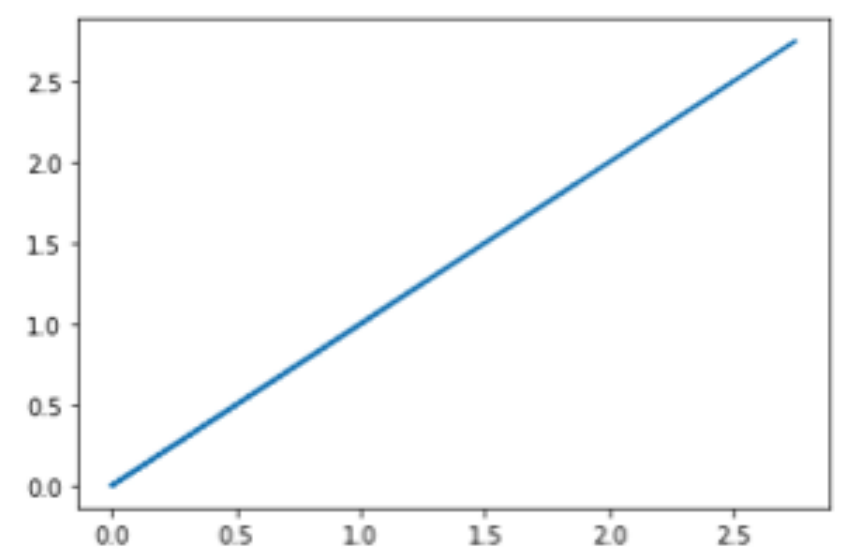
```
rcall.append(recall)
presisi.append(precision)
```

Contoh salah satu hasil perhitungan:

```
jumlah 8
recall 1.0
precision 1.0
```

LANGKAH 7 – Memplot Grafik Recall & Precision

```
plt.plot(rcall,presisi)
plt.show()
```



ANALISA

- Penggunaan Text Mining dapat membantu pencarian kata dalam dokumen, seperti yang diaplikasikan secara sederhana dalam praktikum ini. Proses dalam penyelesaiannya meliputi Tokenizing, Filtering, dan Stemming.
- Untuk melihat kinerja dari algoritma yang sudah dibuat dalam praktikum Text Mining, digunakan perhitungan Recall dan Precision. Dimana Recall adalah Tingkat keberhasilan sistem dalam menemukan Kembali informasi, sedangkan Precision adalah tingkat ketepatan antara informasi yang diminta pengguna.
- Untuk testing, dapat diiberikan sebuat querylist. Dalam kasus pada praktikum ini, diberikan querylist yaitu “pertumbuhan ekonomi perkembangan pasar pergerakan harga saham”
- Bahasan dalam praktikum kali ini yaitu Text Mining pada pengaplikasian Pencarian kata dalam dokumen, memiliki tahapan yatu proses mengatur dan Menyusun data dengan cara tertentu sehingga dapat menjadi sasaran analisis. Melakukannya dengan melibtakan penggunaan teknologi natural language processing, dan disini menggunakan beberapa library untuk menyelesaikan tugasnya, seperti sastrawi untuk Menyusun kata dalam Bahasa Indonesia sehingga menjadi yang diharapkan dan lain sebagainya.