222112058_Feza Raffa Arnanda_Penugasan Praktikum 3

September 6, 2023

0.1 Penugasan Praktikum Pertemuan 3

Nama : Feza Raffa Arnanda
NIM : 222112058
Kelas : 3SD2

A. Inverted Index

[ ]:

```
doc1_term = ["pengembangan", "sistem", "informasi", "penjadwalan"]
doc2_term = ["pengembangan", "model", "analisis", "sentimen", "berita"]
doc3_term = ["analisis", "sistem", "input", "output"]
corpus_term = [doc1_term, doc2_term, doc3_term]
```

[ ]:

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

stemmer_factory = StemmerFactory()
stemmer = stemmer_factory.create_stemmer()

# stemmer = PorterStemmer()

inverted_index = {}
for i in range(len(corpus_term)):
    for item in corpus_term[i]:
        item = stemmer.stem(item)
        if item not in inverted_index:
            inverted_index[item] = []
        if (item in inverted_index) and ((i+1) not in inverted_index[item]):
            inverted_index[item].append(i+1)
print(inverted_index)
```

```
{'kembang': [1, 2], 'sistem': [1, 3], 'informasi': [1], 'jadwal': [1], 'model':
[2], 'analisis': [2, 3], 'sentimen': [2], 'berita': [2], 'input': [3], 'output':
[3]}
```

B. Boolean Retrieval

[ ]:

```
def AND(posting1, posting2):
    p1 = 0
    p2 = 0
    result = list()
    while p1 < len(posting1) and p2 < len(posting2):
        if posting1[p1] == posting2[p2]:
            result.append(posting1[p1])
            p1 += 1
            p2 += 1
        elif posting1[p1] > posting2[p2]:
            p2 += 1
        else:
            p1 += 1
    return result
```

[ ]:

```
AND(inverted_index['sistem'],inverted_index['analisis'])
```

[ ]: [3]

[ ]:

```
AND(inverted_index['informasi'],inverted_index['jadwal'])
```

[ ]: [1]

[ ]:

```
sistem_informasi = AND(inverted_index['sistem'],inverted_index['informasi'])

AND(sistem_informasi, inverted_index['sistem'],)
```

[ ]: [1]

[ ]:

```
def OR(posting1, posting2):
    p1 = 0
    p2 = 0
    result = list()
    while p1 < len(posting1) and p2 < len(posting2):
        if posting1[p1] == posting2[p2]:
            result.append(posting1[p1])
            p1 += 1
            p2 += 1
        elif posting1[p1] > posting2[p2]:
            result.append(posting2[p2])
            p2 += 1
        else:
            result.append(posting1[p1])
            p1 += 1
    while p1 < len(posting1):
        result.append(posting1[p1])
        p1 += 1
    while p2 < len(posting2):
        result.append(posting2[p2])
        p2 += 1
    return result
```

[ ]:

```
OR(inverted_index['sistem'],inverted_index['analisis'])
```

```
[ ]: [1, 2, 3]
```

```
[ ]: OR(inverted_index['sistem'],inverted_index['informasi'])
```

```
[ ]: [1, 3]
```

```
[ ]: def NOT(posting):
         result = list()
         i = 1
         for item in posting:
             while i < item:
                 result.append(i)
                 i += 1
             i += 1

         while i <= len(corpus_term):
             result.append(i)
             i += 1
         return result
```

```
[ ]: NOT(inverted_index['jadwal'])
```

```
[ ]: [2, 3]
```

```
[ ]: NOT(inverted_index['sentimen'])
```

```
[ ]: [1, 3]
```

0.2 Penugasan

1. Menggunakan sekumpulan dokumen pada folder ”berita”, setelah dilakukan preprocessing pada penugasan Modul 2, tambahkan kode untuk menghasilkan inverted index dengan output berupa term dan daftar lokasinya (posting lists) Pada program di bawah ini, digunakan library Sastrawi untuk melakukan stemming. Sastrawi digunakan karena alasan hasil stemming yang lebih baik dibanding library lainnnya seperti NLTK. Program inverted index melanjutkan dari program pada praktikum dimana akan dilakukan looping ke semua kata yang sudah dilakukan preprocessing. Step :

1. Pengecekan apakah sudah terdapat term dalam inverted_index
2. Jika belum ada, maka term tersebut akan terbentuk key baru dalam dictionary inverted_index dan akan memasukkan nama file ke valuenya
3. Jika term sudah ada di inverted_index, maka file yang terdapat term tersebut akan masuk ke value dari key term tersebut.

```
[ ]: import os
     from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
     from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
```

```python
from nltk.tokenize import word_tokenize


# Inisialisasi Stemmer dan Stopword Remover
stemmer_factory = StemmerFactory()
stemmer = stemmer_factory.create_stemmer()

stopword_factory = StopWordRemoverFactory()
stopword_remover = stopword_factory.create_stop_word_remover()

# Inisialisasi inverted index dan dokumen yang sudah ditemukan
inverted_index = {}
documents_found = set()

# Path ke folder "berita"
path = "C:/Users/FEZA/My Drive/00. Drive PC/1.STIS/5. Semester 5/Information Retrieval [IR] P/Pertemuan 3/berita"

# Iterasi ke semua file di folder berita
for file in os.listdir(path):
    if os.path.isfile(os.path.join(path, file)):
        with open(os.path.join(path, file), 'r', encoding='utf-8') as f:
            content = f.read().lower()  # Case folding

            # Tokenization
            words = word_tokenize(content)

            # Menghilangkan stopword dan dilakukan stemming
            filtered_words = [stemmer.stem(stopword_remover.remove(word)) for word in words]

            # Membangun inverted index
            for term in filtered_words:
                if term not in inverted_index:
                    inverted_index[term] = [file]
                elif file not in inverted_index[term]:
                    inverted_index[term].append(file)

# Print hasil inverted index
for term, doc_list in inverted_index.items():
    print(f"Term: {term}")
    print(f"Documents: {doc_list}")
```

```
Term: wilayah
Documents: ['berita1.txt', 'berita5.txt']
Term: kamu
Documents: ['berita1.txt']
```

```
Term: 
Documents: ['berita1.txt', 'berita2.txt', 'berita3.txt', 'berita4.txt',
'berita5.txt', 'desktop.ini']
Term: bebas
Documents: ['berita1.txt']
Term: covid-19
Documents: ['berita1.txt', 'berita2.txt', 'berita3.txt', 'berita4.txt',
'berita5.txt']
Term: cek
Documents: ['berita1.txt']
Term: 34
Documents: ['berita1.txt']
Term: kab kota
Documents: ['berita1.txt']
Term: zona
Documents: ['berita1.txt']
Term: hijau
Documents: ['berita1.txt']
Term: baru
Documents: ['berita1.txt', 'berita3.txt', 'berita4.txt']
Term: jakarta
Documents: ['berita1.txt', 'berita2.txt', 'berita3.txt', 'berita4.txt',
'berita5.txt']
Term: -
Documents: ['berita1.txt', 'berita2.txt', 'berita3.txt', 'berita4.txt',
'berita5.txt']
Term: perintah
Documents: ['berita1.txt']
Term: rencana
Documents: ['berita1.txt', 'berita2.txt']
Term: bakal
Documents: ['berita1.txt', 'berita2.txt', 'berita3.txt']
Term: terap
Documents: ['berita1.txt']
Term: laku
Documents: ['berita1.txt', 'berita2.txt']
Term: batas
Documents: ['berita1.txt']
Term: giat
Documents: ['berita1.txt']
Term: masyarakat
Documents: ['berita1.txt']
Term: ppkm
Documents: ['berita1.txt']
Term: level
Documents: ['berita1.txt']
Term: 3
Documents: ['berita1.txt']
```

```
Term: hitung
Documents: ['berita1.txt']
Term: 24
Documents: ['berita1.txt']
Term: desember
Documents: ['berita1.txt']
Term: 2021
Documents: ['berita1.txt']
Term: hingga
Documents: ['berita1.txt', 'berita2.txt', 'berita4.txt']
Term: 2
Documents: ['berita1.txt', 'berita3.txt']
Term: januari
Documents: ['berita1.txt', 'berita2.txt']
Term: menteri
Documents: ['berita1.txt']
Term: sehat
Documents: ['berita1.txt']
Term: ri
Documents: ['berita1.txt', 'berita3.txt', 'berita4.txt']
Term: pasti
Documents: ['berita1.txt', 'berita2.txt']
Term: bijak
Documents: ['berita1.txt']
Term: tahap
Documents: ['berita1.txt']
Term: kaji
Documents: ['berita1.txt']
Term: direktur
Documents: ['berita1.txt']
Term: cegah
Documents: ['berita1.txt']
Term: kendali
Documents: ['berita1.txt']
Term: sakit
Documents: ['berita1.txt']
Term: tular
Documents: ['berita1.txt']
Term: langsung
Documents: ['berita1.txt', 'berita2.txt', 'berita3.txt']
Term: p2pml
Documents: ['berita1.txt']
Term: kemenkes
Documents: ['berita1.txt', 'berita4.txt']
Term: dr
Documents: ['berita1.txt', 'berita5.txt']
Term: siti
Documents: ['berita1.txt']
```

```
Term: nadia
Documents: ['berita1.txt']
Term: tarmizi
Documents: ['berita1.txt']
Term: kasus
Documents: ['berita1.txt', 'berita4.txt', 'berita5.txt']
Term: naik
Documents: ['berita1.txt', 'berita5.txt']
Term: signifikan
Documents: ['berita1.txt', 'berita4.txt']
Term: umum
Documents: ['berita1.txt']
Term: picu
Documents: ['berita1.txt']
Term: tingkat
Documents: ['berita1.txt', 'berita4.txt']
Term: mobilitas
Documents: ['berita1.txt']
Term: longgar
Documents: ['berita1.txt']
Term: protokol
Documents: ['berita1.txt']
Term: https
Documents: ['berita1.txt', 'berita2.txt', 'berita3.txt', 'berita4.txt',
'berita5.txt']
Term: health detik com berita-detikhealth d-5816690 wilayah-kamu-sudah-bebas-
covid-19-cek-34-kabkota-zona-hijau-terbaru
Documents: ['berita1.txt']
Term: vaksin
Documents: ['berita2.txt', 'berita3.txt']
Term: rutin
Documents: ['berita2.txt']
Term: tahun
Documents: ['berita2.txt']
Term: gantung
Documents: ['berita2.txt']
Term: jelas
Documents: ['berita2.txt']
Term: beri
Documents: ['berita2.txt']
Term: booster
Documents: ['berita2.txt', 'berita3.txt']
Term: dosis
Documents: ['berita2.txt', 'berita3.txt']
Term: tiga
Documents: ['berita2.txt', 'berita3.txt']
Term: indonesia
Documents: ['berita2.txt', 'berita3.txt', 'berita4.txt']
```

```
Term: 2022
Documents: ['berita2.txt', 'berita3.txt']
Term: lantas
Documents: ['berita2.txt']
Term: ada
Documents: ['berita2.txt', 'berita4.txt']
Term: mungkin
Documents: ['berita2.txt']
Term: vaksinasi
Documents: ['berita2.txt']
Term: influenza
Documents: ['berita2.txt']
Term: ketua
Documents: ['berita2.txt', 'berita3.txt']
Term: satgas
Documents: ['berita2.txt', 'berita3.txt']
Term: ikat
Documents: ['berita2.txt', 'berita3.txt']
Term: dokter
Documents: ['berita2.txt', 'berita3.txt']
Term: idi
Documents: ['berita2.txt', 'berita3.txt']
Term: prof
Documents: ['berita2.txt', 'berita3.txt']
Term: zubairi
Documents: ['berita2.txt', 'berita3.txt']
Term: djoerban
Documents: ['berita2.txt', 'berita3.txt']
Term: kini
Documents: ['berita2.txt', 'berita5.txt']
Term: kait
Documents: ['berita2.txt', 'berita3.txt']
Term: sebut
Documents: ['berita2.txt', 'berita3.txt', 'berita4.txt', 'berita5.txt']
Term: turut
Documents: ['berita2.txt']
Term: cukup
Documents: ['berita2.txt']
Term: sekali
Documents: ['berita2.txt']
Term: kemudian
Documents: ['berita2.txt']
Term: perlu
Documents: ['berita2.txt']
Term: health detik com berita-detikhealth d-5816582 vaksin-covid-19-bakal-rutin-
setiap-tahun-tergantung-ini-penjelasannya
Documents: ['berita2.txt']
Term: mulai
```

```
Documents: ['berita3.txt']
Term: suntik
Documents: ['berita3.txt']
Term: masih
Documents: ['berita3.txt']
Term: ampuh
Documents: ['berita3.txt']
Term: lawan
Documents: ['berita3.txt']
Term: varian
Documents: ['berita3.txt', 'berita4.txt', 'berita5.txt']
Term: delta
Documents: ['berita3.txt', 'berita4.txt', 'berita5.txt']
Term: cs
Documents: ['berita3.txt']
Term: pakar
Documents: ['berita3.txt']
Term: aku
Documents: ['berita3.txt']
Term: guna
Documents: ['berita3.txt']
Term: 1-2
Documents: ['berita3.txt']
Term: memang
Documents: ['berita3.txt']
Term: alami
Documents: ['berita3.txt', 'berita4.txt']
Term: turun
Documents: ['berita3.txt', 'berita5.txt']
Term: efektivitas
Documents: ['berita3.txt']
Term: corona
Documents: ['berita3.txt', 'berita5.txt']
Term: ingat
Documents: ['berita3.txt']
Term: awal
Documents: ['berita3.txt']
Term: jenis
Documents: ['berita3.txt']
Term: ikut
Documents: ['berita3.txt']
Term: strain
Documents: ['berita3.txt']
Term: virus
Documents: ['berita3.txt']
Term: jawab
Documents: ['berita3.txt']
Term: tanya
```

```
Documents: ['berita3.txt']
Term: singgung
Documents: ['berita3.txt']
Term: riset
Documents: ['berita3.txt']
Term: 1
Documents: ['berita3.txt']
Term: dasar
Documents: ['berita3.txt']
Term: jauh
Documents: ['berita3.txt']
Term: pfizer
Documents: ['berita3.txt']
Term: moderna
Documents: ['berita3.txt']
Term: bukti
Documents: ['berita3.txt']
Term: health detik com berita-detikhealth d-5816534 ri-mulai-suntikkan-booster-di-2022-masihkah-ampuh-lawan-varian-delta-cs
Documents: ['berita3.txt']
Term: alert
Documents: ['berita4.txt']
Term: dki
Documents: ['berita4.txt']
Term: data
Documents: ['berita4.txt']
Term: balitbangkes
Documents: ['berita4.txt']
Term: per
Documents: ['berita4.txt']
Term: 13
Documents: ['berita4.txt']
Term: november
Documents: ['berita4.txt']
Term: tunjuk
Documents: ['berita4.txt']
Term: tambah
Documents: ['berita4.txt']
Term: jadi
Documents: ['berita4.txt']
Term: jawa
Documents: ['berita4.txt']
Term: barat
Documents: ['berita4.txt', 'berita5.txt']
Term: 165
Documents: ['berita4.txt']
Term: 90
Documents: ['berita4.txt']
```

```
Term: sulawesi
Documents: ['berita4.txt']
Term: utara
Documents: ['berita4.txt']
Term: 86
Documents: ['berita4.txt']
Term: satu
Documents: ['berita4.txt']
Term: bulan
Documents: ['berita4.txt']
Term: akhir
Documents: ['berita4.txt']
Term: alpha
Documents: ['berita4.txt']
Term: beta
Documents: ['berita4.txt']
Term: banyak
Documents: ['berita4.txt']
Term: asal
Documents: ['berita4.txt']
Term: total
Documents: ['berita4.txt']
Term: 1 327
Documents: ['berita4.txt']
Term: health detik com berita-detikhealth d-5812940 alert-kasus-varian-delta-
covid-19-di-dki-meningkat
Documents: ['berita4.txt']
Term: as
Documents: ['berita5.txt']
Term: dadak
Documents: ['berita5.txt']
Term: usai
Documents: ['berita5.txt']
Term: serang
Documents: ['berita5.txt']
Term: sempat
Documents: ['berita5.txt']
Term: reda
Documents: ['berita5.txt']
Term: jumlah
Documents: ['berita5.txt']
Term: amerika
Documents: ['berita5.txt']
Term: serikat
Documents: ['berita5.txt']
Term: padahal
Documents: ['berita5.txt']
Term: tahu
```

```
Documents: ['berita5.txt']
Term: catat
Documents: ['berita5.txt']
Term: stabil
Documents: ['berita5.txt']
Term: pasca
Documents: ['berita5.txt']
Term: musim
Documents: ['berita5.txt']
Term: panas
Documents: ['berita5.txt']
Term: apa
Documents: ['berita5.txt']
Term: sampai
Documents: ['berita5.txt']
Term: kepala
Documents: ['berita5.txt']
Term: nasihat
Documents: ['berita5.txt']
Term: medis
Documents: ['berita5.txt']
Term: gedung
Documents: ['berita5.txt']
Term: putih
Documents: ['berita5.txt']
Term: anthony
Documents: ['berita5.txt']
Term: fauci
Documents: ['berita5.txt']
Term: senin
Documents: ['berita5.txt']
Term: 15 11 2021
Documents: ['berita5.txt']
Term: nasional
Documents: ['berita5.txt']
Term: 57
Documents: ['berita5.txt']
Term: persen
Documents: ['berita5.txt']
Term: minggu
Documents: ['berita5.txt']
Term: lalu
Documents: ['berita5.txt']
Term: puncak
Documents: ['berita5.txt']
Term: gelombang
Documents: ['berita5.txt']
Term: pasien
```

```
Documents: ['berita5.txt']
Term: area
Documents: ['berita5.txt']
Term: tengah
Documents: ['berita5.txt']
Term: timur
Documents: ['berita5.txt']
Term: laut
Documents: ['berita5.txt']
Term: health detik com berita-detikhealth d-5813949 corona-di-as-mendadak-naik-
lagi-usai-serangan-delta-sempat-mereda
Documents: ['berita5.txt']
Term: shellclassinfo
Documents: ['desktop.ini']
Term: iconresource c
Documents: ['desktop.ini']
Term: program
Documents: ['desktop.ini']
Term: files google drive
Documents: ['desktop.ini']
Term: file
Documents: ['desktop.ini']
Term: stream 80 0 1 0 googledrivefs exe 23
Documents: ['desktop.ini']
```

2. Kemudian tambahkan kode untuk melakukan boolean retrieval dari inverted index pada Penugasan 1. Perhatikan daftar dokumen yang dikembalikan ketika menuliskan query berikut.

1. corona
2. covid
3. vaksin
4. corona OR covid
5. vaksin AND corona
6. vaksin AND corona AND pfizer
7. NOT vaksin

Pada kode dibawah, kita menggunakan string method yaitu .get(), dimana kita bisa mengambil atau menambahkan suatu key dan value ke dalam suatu objek. Objek disini adalah inverted_index yang berbentuk dictionary. Langkah-langkah :

1. Inisialisasi variabel query
2. Inisialisasi variabel resut berdasarkan query. Jika query termasuk boolean retrieval query, maka kita gunakan function boolean yang sudah dilakukan pada sesi praktikum untuk mereturn file apa saja yang memenuhi boolean query.
3. Print hasilnya

[ ]:
```python
# Query 1: corona
query1 = "corona"
```

```
result1 = inverted_index.get(query1, [])
print(f"Query: {query1}")
print(f"Documents: {result1}")

# Query 2: covid
query2 = "covid-19"
result2 = inverted_index.get(query2, [])
print(f"Query: {query2}")
print(f"Documents: {result2}")

# Query 3: vaksin
query3 = "vaksin"
result3 = inverted_index.get(query3, [])
print(f"Query: {query3}")
print(f"Documents: {result3}")

# Query 4: corona OR covid
query4 = "corona OR covid"
query4_terms = query4.split(" OR ")
result4 = OR(inverted_index.get(query4_terms[0], []), inverted_index.
 ↪get(query4_terms[1], []))
print(f"Query: {query4}")
print(f"Documents: {result4}")

# Query 5: vaksin AND corona
query5 = "vaksin AND corona"
query5_terms = query5.split(" AND ")
result5 = AND(inverted_index.get(query5_terms[0], []), inverted_index.
 ↪get(query5_terms[1], []))
print(f"Query: {query5}")
print(f"Documents: {result5}")

# Query 6: vaksin AND corona AND pfizer
query6 = "vaksin AND corona AND pfizer"
query6_terms = query6.split(" AND ")
result6 = AND(inverted_index.get(query6_terms[0], []), inverted_index.
 ↪get(query6_terms[1], []))
result6 = AND(result6, inverted_index.get(query6_terms[2], []))
print(f"Query: {query6}")
print(f"Documents: {result6}")

# Query 7: NOT vaksin
query7 = "NOT vaksin"
query7_terms = query7.split(" NOT ")
result7 = NOT(inverted_index.get(query7_terms[0], []))
print(f"Query: {query7}")
print(f"Documents: {result7}")
```

```
Query: corona
Documents: ['berita3.txt', 'berita5.txt']
Query: covid-19
Documents: ['berita1.txt', 'berita2.txt', 'berita3.txt', 'berita4.txt',
'berita5.txt']
Query: vaksin
Documents: ['berita2.txt', 'berita3.txt']
Query: corona OR covid
Documents: ['berita3.txt', 'berita5.txt']
Query: vaksin AND corona
Documents: ['berita3.txt']
Query: vaksin AND corona AND pfizer
Documents: ['berita3.txt']
Query: NOT vaksin
Documents: [1, 2, 3]
```

3. Modifikasi kode fungsi AND sehingga dapat melakukan optimasi query untuk list postings berikut: Pengembangan function AND disini menangkap parameter postings dimana parameter postings merupakan variabel yang terdiri dari banyak inverted_index yang akan dilakukan boolean retrieval. Algoritma :

1. Jika parameter bukan merupakan banyak dokumen, maka hasilnya akan kosong. Hal ini sesuai dengan boolean AND yang mengharuskan dua macam dokumen untuk dibandingkan
2. Untuk mempermudah dan mempercepat program untuk jalan, kita akan sort posting terpendek untuk menjadi base posting yang akan dilakukan query boolean AND
3. Base posting akan dilakukan query AND dengan posting lainnya secara terutur dan dilakukan looping untuk mengecek dokumen mana yang memenuhi boolean
4. Hasil akan menunjukkan hasil boolean retrieval dari inverted index yang sudah terinisialisasi

[ ]:

```python
def AND_optimized(postings):
    # Jika tidak ada posting list, maka hasilnya adalah list kosong
    if not postings:
        return []

    # Mengurutkan posting lists berdasarkan panjangnya (posting list terpendek␣
 ↪akan diolah pertama kali), sudah dijelaskan di kelas
    postings.sort(key=len)

    # Menggunakan posting list terpendek sebagai dasar untuk pencarian
    base_posting = postings[0]

    # Menghapus posting list terpendek dari daftar posting
    # Kode dibawah artinya postings[1] itu postingan terpendek kedua setelah␣
 ↪base_posting; memilih seluruh posting dimulai dari posting terpendek kedua
    postings = postings[1:]

    result = []
    for doc_id in base_posting:
        doc_id_found_in_all = True
        for posting in postings:
            # Jika doc_id tidak ada di salah satu posting list, hentikan␣
 ↪pencarian
            if doc_id not in posting:
                doc_id_found_in_all = False
                break
        # Jika doc_id ditemukan di semua posting list, tambahkan ke hasil
        if doc_id_found_in_all:
            result.append(doc_id)

    return result
```

[ ]:

```
vaksin_posting = inverted_index.get("vaksin", [])
corona_posting = inverted_index.get("corona", [])
pfizer_posting = inverted_index.get("pfizer", [])

# and_queries = AND_optimized(inverted_index.get(query_terms,[]))
# Menggabungkan keseluruhannya
query_result = AND_optimized([vaksin_posting, corona_posting, pfizer_posting])



# Print hasil optimized query boolean retrieval
query = "vaksin AND corona AND pfizer"
print(f"Query: {query}")
print(f"Documents: {query_result}")
```

```
['berita2.txt', 'berita3.txt']
Query: vaksin AND corona AND pfizer
Documents: ['berita3.txt']
```