# KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI PROGRAM STUDI ILMU KOMPUTER DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM UNIVERSITAS GADJAH MADA

# TEMU KEMBALI INFORMASI

# Tugas 8 Metode Perhitungan Skor Pencarian



### **DISUSUN OLEH:**

**ADAM YOGISYAH PUTRA** 20/455439/PA/19654

MUHAMMAD ARSYA PUTRA 20/462186/PA/20158

HIZKYA FIRSTADIPA HARTOKO 20/455447/PA/19662

#### DOSEN:

Dr. Lukman Heryawan, S.T., M.T.

### Metode Perhitungan Skor Pencarian

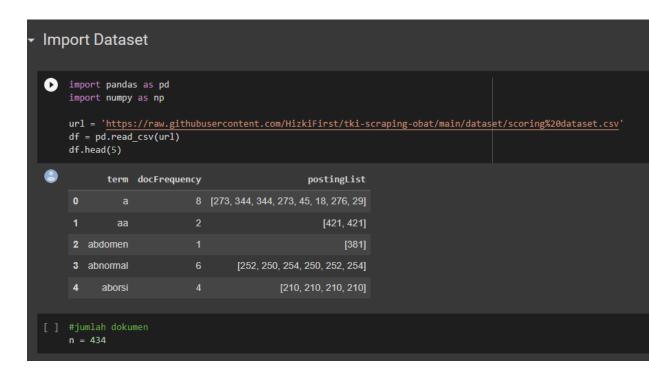
#### I. Metode

Metode yang digunakan pada saat melakukan perhitungan skor sistem temu kembali informasi adalah dengan menggunakan TF-IDF. Metode ini mempertimbangkan kemunculan suatu term dalam dokumen, yaitu sebagai Term Frequency, dan jumlah dokumen yang di dalamnya terdapat term tersebut yaitu Document Frequency yang di Invert menjadi Inverted Document Frequency.

## **II.** Proses Scoring

## A. Import Dataset

Melakukan import dataset yang akan digunakan untuk melakukan scoring. Dataset merupakan Document Inverted Index yang berisi term-term beserta frekuensi kemunculan term dalam dokumen dan posting list (daftar index dokumen yang memuat term).



# **B.** Proses Definisi Query Term

Mendefinisikan query yang digunakan untuk uji skoring dan melakukan tokenisasi terhadap query tersebut agar memperoleh query terms.

# C. Perhitungan TF-IDF

Menghitung skor dengan metode TF-IDF

Untuk tiap query term dilakukan perhitungan IDF dengan rumus:

$$IDF = log_{10} \frac{n}{DF}$$

Lalu dilakukan perhitungan TF dari query term tersebut pada tiap document .

Skor diperoleh dengan mengalikan hasil TF dengan IDF.

```
Perhitungan TF-IDF
     import math
     import operator as op
     skor = [0]*n
     docindex = []
     df_skor = pd.DataFrame()
     for qterm in query:
       i= df.index[df['term']==qterm]
       docindex.append(i[0])
       doclist = df.iloc[i[0]]["postingList"]
       doclist = doclist.strip('][').split(", ")
       docFreq = len(set(doclist))
       docUnique = set(doclist)
       docUnique = [eval(i) for i in docUnique]
       docUniqueList = sorted(list(docUnique))
       # Perhitungan IDF
       idf = math.log(n/docFreq,10)
       # Perhitungan TF
       for doc in docUniqueList:
         tf = op.countOf(doclist, str(doc))
         skor[doc] = tf*idf
```

## D. Pengurutan Hasil Skor

Hasil skor diurutkan dari terbesar ke terkecil

## III. Hasil

Hasil yang diperoleh adalah Skor dan document ID yang sudah terurut berdasarkan perhitungan TF-IDF

Link dataset

https://github.com/HizkiFirst/tki-scraping-obat/blob/main/dataset/scoring%20dataset.csv

### IV. Lampiran

Repository: <a href="https://github.com/HizkiFirst/tki-scraping-obat">https://github.com/HizkiFirst/tki-scraping-obat</a>

# Source code:

https://colab.research.google.com/drive/1-hC3LDkF1oVQd\_QCgd7LveIvqB7TeRtE?usp =sharing