**Membuat Implementasi Penghitungan Skor**

Disusun untuk Memenuhi Tugas Mata Kuliah

Temu Kembali Informasi

Dosen Pengampu: **Arif Nurwidyantoro, M.Cs**

****

Disusun oleh Kelompok 4:

Muhammad Faqih Husaen (19/442478/PA/19227)

Fajar Merah Diwangkara (20/459264/PA/19925)

Ryan Novianno (20/462191/PA/20163)

**PROGRAM STUDI S1 ILMU KOMPUTER**

**DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

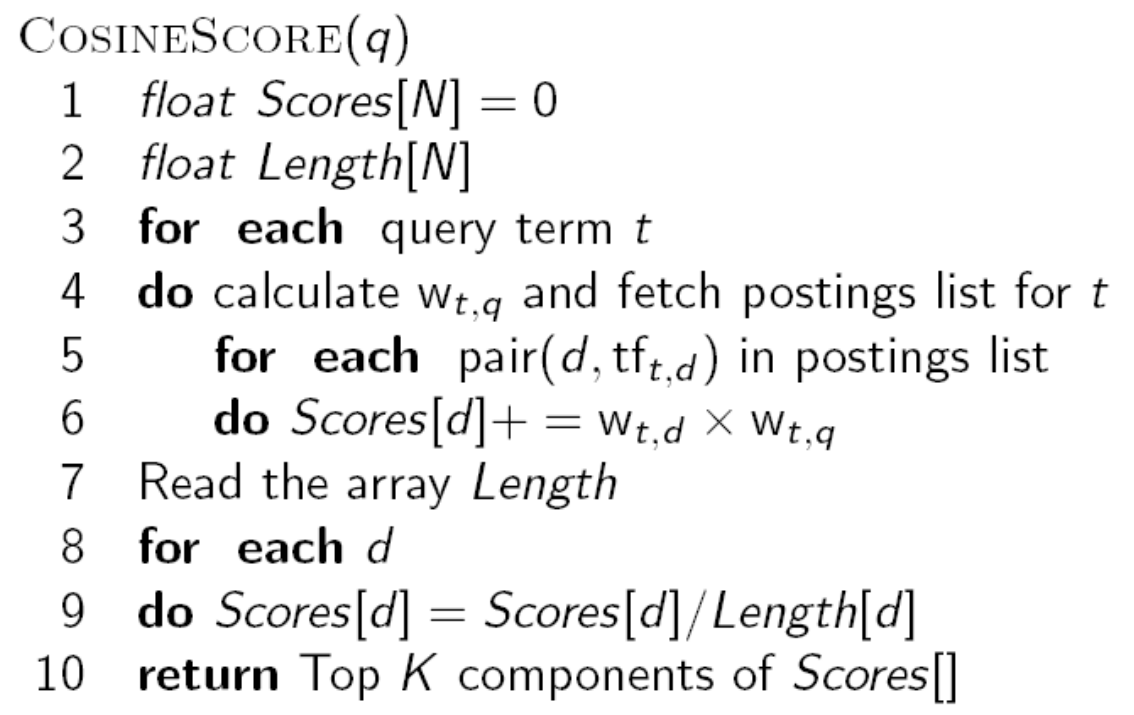
**UNIVERSITAS GADJAH MADA**

**2022**

Link google colab: <https://colab.research.google.com/drive/1EDSq9F-1Bg_JafSX_2c7PIwrfgrI8yL8?usp=sharing#scrollTo=slZ_v6uadmLm>

Metode Penghitungan Skor

Metode yang digunakan untuk menghitung skor pencarian adalah metode cosine. Metode cosine similarity merupakan metode untuk menghitung kesamaan antara dua buah objek yang dinyatakan dalam dua buah vector dengan menggunakan keywords (kata kunci) dari sebuah dokumen sebagai ukuran. Cosine similarity memiliki cara kerja sebagai berikut :



Implementasi Penghitungan Skor

Fungsi unique() digunakan untuk mendapatkan daftar term tanpa duplikat dari sebuah daftar term.

def unique(x):

list\_set = set(x)

unique\_list = (list(list\_set))

return unique\_list

Fungsi get\_idf() berfungsi untuk mendapatkan nilai idf dari masing-masing term. Proses perhitungan dilakukan dengan pertama kali menghitung kemunculan terms pada keseluruhan dokumen terlebih dahulu dan kemudian dilakukan perhitungan idf dengan rumus log((1+N) /(df+1))+1.

def get\_idf(corpus):

idf\_dict=defaultdict(lambda: 0)

terms\_only = []

for n in corpus:

temp = corpus[n]

for j in range(0, len(temp)):

terms\_only.append(temp[j])

unique\_word = unique(terms\_only)

for word in unique\_word :

count = 0

for n in terms\_only:

if word == n:

count += 1

idf\_dict[word] = (math.log((1+len(corpus))/(count+1)))+1

return idf\_dict

Fungsi Term\_frequency() berfungsi untuk menghitung kemunculan term pada suatu dokumen.

def Term\_frequency(term, unique\_word):

term\_frequency = defaultdict(lambda: 0)

for word in unique\_word :

for i in range(len(term)):

if word == term[i]:

term\_frequency[word] += 1

return term\_frequency

Fungsi tf\_idf berfungsi untuk menghitung nilai weight tf-idf dengan menggunakan nilai term frequency (tf) dan inverse document frequency (idf).

def tf\_idf(tf, idf):

tf\_idf = (math.log(1+tf))\*idf

return tf\_idf

Fungsi cosine berfungsi untuk mendapatkan skor cosine untuk masing-masing dokumen. Proses perhitungan dimulai dengan mengambil setiap kata unik dari query dan dilanjutkan menghitung Term frequency (tf) dari query. Kemudian untuk setiap dokumen dilakukan perhitungan Term frequency (tf) dan tf idf. Langkah terakhir adalah menghitung score menggunakan nilai tf idf yang sudah didapatkan.

def cosine(document\_corpus, query\_corpus, N, idf):

tf\_idf\_document = defaultdict(lambda: 0)

tf\_idf\_query = defaultdict(lambda: 0)

score = {}

query\_unique\_terms = unique(query\_corpus)

query\_tf = Term\_frequency(query\_corpus, query\_unique\_terms)

for n in document\_corpus:

product = []

document\_terms = document\_corpus[n]

document\_tf = Term\_frequency(document\_terms, query\_unique\_terms)

for w in query\_unique\_terms:

weight\_document = tf\_idf(document\_tf[w], idf[w])

weight\_query = tf\_idf(query\_tf[w], idf[w])

product.append(weight\_query\*weight\_document)

score[n] = sum(product)/len(document\_terms)

return score

Contoh penggunaan kode penghitungan skor

1. Pertama definisikan query

query = ["A", "man", "is", "so", "small", "in", "the", "wilderness", "believe", "me", "The", "way", "how",

"people", "is", "now", "we", "are", "not", "tailored", "to", "live", "there", "So", "when", "Stew",

"say", "he", "stumble", "across", "a", "house", "in", "the", "middle", "of", "the", "mountain",

"my", "ears", "prick", "up"]

1. Definisikan term dan dokumen. Untuk di contoh yang kami pakai, kami menggunakan term dan dokumen yang berasal dari inverted index hasil indexing. Fungsi crawl\_and\_scrape berfungsi untuk melakukan indexing lalu menghasilkan inverted index, lalu menghasilkan data yang dibutuhkan untuk pengukuran skor dokumen

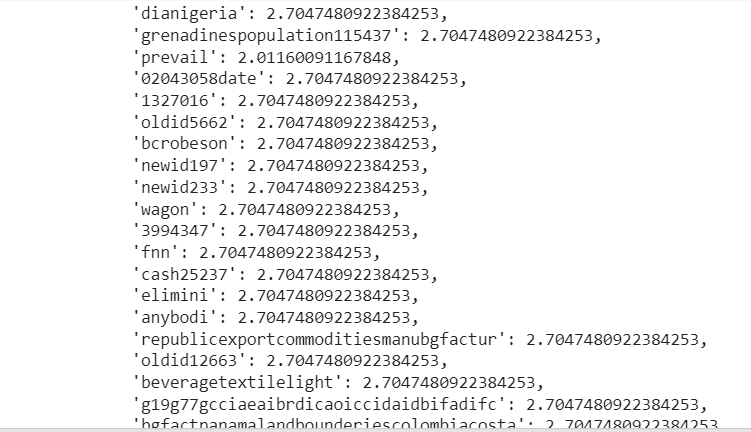
document\_terms, index = crawl\_and\_scrape("foldersatu", "hasil\_ scraping")

1. Hitung nilai idf dari masing-masing term

idf = get\_idf(document\_terms)

idf

Contoh hasil penghitungan nilai idf masing-masing term



1. Hitung skor cosine masing-masing dokumen

rangking = cosine(satu, query, len(satu), idf)

rangking

Contoh hasil penghitungan skor cosine masing-masing dokumen

