APLIKASI DOT PRODUCT PADA SISTEM TEMU-BALIK INFORMASI

Laporan Tugas Besar 2 IF2123 Aljabar Linier dan Geometri
Disusun sebagai syarat tugas besar mata kuliah Aljabar Linier dan Geometri IF2123
Semester I Tahun 2020/2021

Kelompok Auto A

Ferdy Irawan Firdaus	**13519030**
Fakhri Nail W.	**13519035**
Fabian Savero Diaz P.	**13519140**

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO & INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2020

BAB I

DESKRIPSI MASALAH

Hampir semua dari kita pernah menggunakan *search engine*, seperti google, bing dan yahoo! search. Setiap hari, bahkan untuk sesuatu yang sederhana kita menggunakan mesin pencarian Tapi, pernahkah kalian membayangkan bagaimana cara *search engine* tersebut mendapatkan semua dokumen kita berdasarkan apa yang ingin kita cari?

Sebagaimana yang telah diajarkan di dalam kuliah pada materi *vector* di ruang Euclidean, temu-balik informasi (*information retrieval*) merupakan proses menemukan kembali (*retrieval*) informasi yang relevan terhadap kebutuhan pengguna dari suatu kumpulan informasi secara otomatis. Biasanya, sistem temu balik informasi ini digunakan untuk mencari informasi pada informasi yang tidak terstruktur, seperti laman web atau dokumen.

Ide utama dari sistem temu balik informasi adalah mengubah *search query* menjadi ruang vektor Setiap dokumen maupun *query* dinyatakan sebagai vektor w = (w1, w2,..., wn) di dalam Rn, dimana nilai wi dapat menyatakan jumlah kemunculan kata tersebut dalam dokumen (*term frequency*). Penentuan dokumen mana yang relevan dengan *search query* dipandang sebagai pengukuran kesamaan (*similarity measure*) antara query dengan dokumen. Semakin sama suatu vektor dokumen dengan vektor *query*, semakin relevan dokumen tersebut dengan *query*. Kesamaan tersebut dapat diukur dengan *cosine similarity* dengan rumus:

$$sim(Q, D) = cos\ cos\ \theta\ = \frac{Q\ \cdot D}{\|Q\|\|D\|}$$

Pada kesempatan ini penulis berkesempatan membuat sebuah search engine sederhana dengan model ruang *vector* dan memanfaatkan *cosine similarity*. Program ini mampu menerima search query. *Search query* berupa kata dasar maupun berimbuhan. Terdapat minimal 15 dokumen berbeda yang akan menjadi kandidat perankingan disisipkan manual. Dokumen-dokumen ini dan *query* dilakukan stemming dan stopword terlebih dahulu kemudian hasil pencarian terurut berdasarkan similaritas tertinggi di paling atas sampai posisi paling bawah adalah dokumen dengan similaritas terendah. Nilai similaritas dari setiap dokumen juga ditampilkan.

BAB II

TEORI DASAR

2.1 *Information Retrieval* (IR) atau Temu-Balik Informasi

2.1.1 Pengertian *Information Retrieval* (IR) atau Temu-Balik Informasi

Information Retrieval (IR) atau sering disebut "temu kembali informasi" adalah ilmu yang mempelajari prosedur-prosedur dan metode-metode untuk menemukan kembali informasi yang tersimpan dari berbagai sumber (*resources*) yang relevan atau koleksi sumber informasi yang dicari atau dibutuhkan. Dengan tindakan indeks (*indexing*), panggilan (*searching*), pemanggilan data kembali (*recalling*).

Dalam pencarian data, beberapa jenis data dapat ditemukan diantaranya *texts, table*, gambar (*image*), video, audio. Adapun tujuan dari *Information Retrieval* adalah untuk memenuhi informasi pengguna dengan cara meretrieve dokumen yang relevan atau mengurangi dokumen pencarian yang tidak relevan.

2.1.2 Definisi *Information Retrieval* (IR) atau Temu-Balik Informasi

Secara Konsep sederhana IR merupakan proses mencari, dan kemudian mendapatkan apa yang dicari. Jika kita titik beratkan kepada prosesnya maka di dalamnya akan terungkap bagaimana perjalanan informasi yang diminta, menjadi informasi yang diberikan. Menurut beberapa ahli IR didefinisikan sebagai berikut:

1. Menurut Kowalaski

   Informasi Retrieval adalah konsep sederhana dalam pencarian yang dilakukan oleh seseorang. seperti contoh ketika user akan mencari informasi yang dia butuhkan, maka sistem menerjemahkan kepada bentuk *statement* yang kemudian dieksekusi oleh sistem pencari.

2. William Hersh Menyatakan:

   Information Retrieval adalah "bidang di persimpangan ilmu informasi dan ilmu komputer. Berkutat dengan pengindeksan dan pengambilan informasi dari sumber informasi heterogen dan sebagian besar-tekstual. Istilah ini diciptakan oleh Mooers pada tahun 1951, yang menganjurkan bahwa diterapkan ke "aspek intelektual" deskripsi informasi dan sistem untuk pencarian (Mooers, 1951). "

3. Kutipan Dari Wikipedia

   Wikipedia menjelaskan *Information Retrieval* (IR) adalah seni dan ilmu mencari informasi dalam dokumen, mencari dokumen itu sendiri, mencari metadata

yang menjelaskan dokumen, atau mencari dalam *database*, apakah relasional *database* itu berdiri sendiri atau *database hypertext* jaringan seperti Internet atau intranet, untuk teks, suara, gambar, video atau data.

Dari ketiga rujukan definisi diatas, sudah jelas dimaksudkan bahwa *information retrieval* adalah bidang keilmuan dalam Teknologi informasi yang menjelaskan tentang "Pencarian dan Pengambilan Kembali Informasi".

Keilmuan ini mengungkapkan bagaimana metode metode pencarian informasi yang dilakukan oleh end user dari gudang gudang penyimpanan yang berskala besar, contoh sederhananya adalah media penyimpanan kita sendiri.

Terkadang ketika semakin banyak data yang kita simpan dalam sebuah media penyimpanan tak jarang kita akan lupa dimana kita meletakan data yang kita simpan tadi, sehingga kita melakukan proses pencarian data yang kita lupa tadi, bisa dengan menggunakan *tools* pencarian atau bisa dengan memeriksa satu persatu tempat penyimpanan data kita.

Dalam studi kasus yang lebih kompleks, penerapan IR adalah *Search Engine* (Mesin Pencari) seperti google, yahoo, bing dll. SE merupakan implementasi yang sangat kompleks dari IR.

2.1.3 Peranan *Information Retrieval* (IR) atau Temu-Balik Informasi

Information Retrieval (IR) memiliki kegunaan yang banyak untuk user. Kita bisa melihat fungsinya di mesin pencari untuk mencari informasi, atau di perpustakaan, di apotik dan lain sebagainya. Itu semua adalah karena jasa IR. *Information Retrieval* mempunyai peran untuk:

1. Menganalisis isi sumber informasi dan pertanyaan pengguna.
2. Mempertemukan pertanyaan pengguna dengan sumber informasi untuk mendapatkan dokumen yang relevan.

2.1.4 Contoh Penerapan *Information Retrieval* (IR) atau Temu-Balik Informasi

1. *Searching Text* melalui *Web Search Engine*

   Keyword dimasukkan oleh user untuk pencarian informasi yang diinginkan pada *Search Engine*, yang mana informasi yang didapatkan mengandung relevansi/keterkaitan dengan yang diharapkan.

2. Information retrieval di Perpustakaan

   Perpustakaan adalah salah satu institusi pertama yang mengadopsi sistem IR untuk mendapatkan informasi. Pada umumnya, sistem yang digunakan di perpustakaan pada awalnya dikembangkan oleh institusi akademis dan kemudian oleh produsen

komersil. Pada generasi pertama, sistem pada dasarnya terdiri dari suatu otomatisasi dari teknologi sebelumnya (seperti kartu katalog) dan memungkinkan pencarian berdasar judul dan nama pengarang. Pada generasi kedua, kemampuan pencarian ditambahkan dengan pencarian berdasarkan pokok utama, dengan kata kunci, dan tambahan lagi fasilitas *query* kompleks. Pada generasi ketiga, yang sekarang ini yang sedang menyebar, fokusnya adalah meningkatkan antarmuka grafis, format elektronik, fitur *hypertext*, dan sistem arsitektur terbuka.

3. CBIR (*Content Based Image Retrieval*) *Technology*

   Retrieval berdasarkan kategori konten dan warna. Dimana *user* mendeskripsikan image apa yang akan dicari dengan cara memilih kategori misalnya jenis *image*, negara, tahun pembuatan dan sebagainya.

2.2 Vektor

2.2.1 Definisi Vektor

Vektor merupakan sebuah besaran yang memiliki arah. Vektor digambarkan sebagai panah dengan yang menunjukan arah vektor dan panjang garisnya disebut besar vektor.

2.2.2 Notasi Vektor

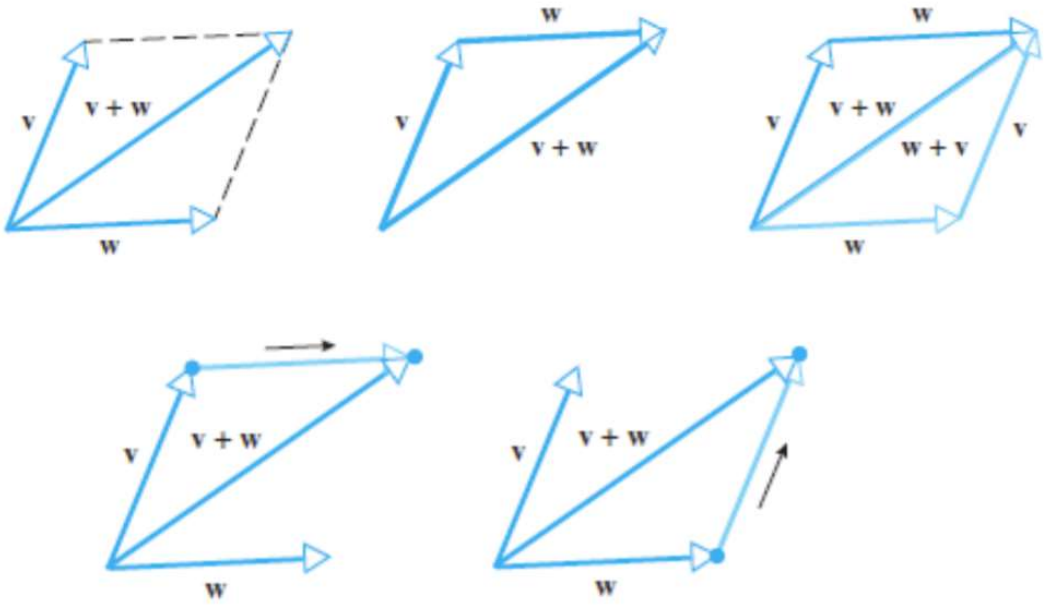
Dalam penulisannya, jika vektor berawal dari titik A dan berakhir di titik B bisa ditulis dengan sebuah huruf kecil yang diatasnya ada tanda garis/panah seperti $\vec{v}$ atau $\bar{v}$ atau juga $\vec{AB}$.

2.2.3 Ruang Vektor (Ruang Euclidean)

Suatu ruang vektor adalah suatu himpunan objek yang dapat dijumlahkan satu sama lain dan dikalikan dengan suatu bilangan, yang masing-masing menghasilkan anggota lain dalam himpunan itu. Vektor di $R^n : \boldsymbol{v} = (v_1, v_2, v_3, \ldots, v_n)$.

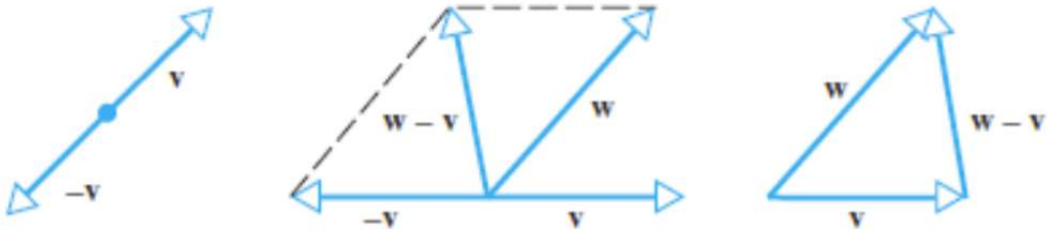
2.2.4 Penjumlahan Dua Vektor

Menggunakan kaidah parallelogram atau kaidah segitiga. Jika $\mathbf{v} = (v_1, v_2, v_3, \ldots, v_n)$ dan $\mathbf{w} = (w_1, w_2, w_3, \ldots, w_n)$, maka $\mathbf{v} + \mathbf{w} = (v_1 + w_1, v_2 + w_2, \ldots, v_n + w_n)$.



2.2.5 Pengurangan Dua Vektor

Jika **v** = $(v_1, v_2, v_3, \ldots, v_n)$ dan **w** = $(w_1, w_2, w_3, \ldots, w_n)$, maka **v - w** = $(v_1 - w_1, v_2 - w_2, \ldots, v_n - w_n)$.



2.2.6 Perkalian Vektor dengan Skalar

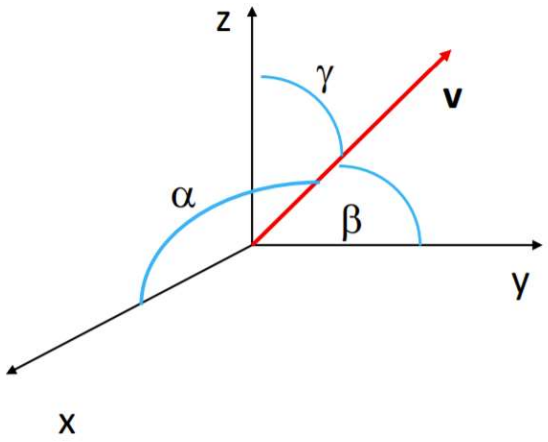
kv = vektor yang panjangnya |k| kali Panjang **v.** Jika v = $(v_1, v_2, v_3, \ldots, v_n)$ maka **kv** = $(kv_1, kv_2, kv_3, \ldots, kv_n)$.

2.2.7 Norma Sebuah Vektor

Panjang (atau magnitude) sebuah vektor **v** dinamakan norma (norm) **v**. Norma vektor **v** dilambangkan dengan ||**v**|| . Norma sebuah vektor dinamakan juga norma Euclidean. Norma vektor **v** = $(v_1, v_2, v_3, \ldots, v_n)$ di R^n adalah ||**v**|| = $\sqrt{v_1^2 + v_2^2 + v_3^2 + \ldots + v_n^2}$

2.2.8 Arah Sebuah Vektor

Misalkan **v** = (v_1, v_2, v_3) adalah vektor di R^3maka arah vektor v adalah$cos(\alpha) = \frac{v_1}{\|v\|}$; $cos(\beta) = \frac{v_2}{\|v\|}$; $cos(\gamma) = \frac{v_3}{\|v\|}$



2.2.9 Vektor Satuan

Vektor satuan (unit vector) adalah vektor dengan panjang = 1. Vektor satuan dilambangkan dengan **u**. Jika **v** adalah vektor di R^ndan **v** ≠ 0 maka **u** = $\frac{1}{\|v\|}$ $v$atau **u** = $\frac{v}{\|v\|}$.Vektor **u** memilik arah yang sama dengan **v.**

2.2.10 Vektor Satuan Standar

Vektor satuan standard di R^n adalah $e_1, e_2, e_3, \ldots, e_n$, e_1= (1,0,0,...,0), e_2 = (0,1,0,...,0), …, dan e_n= (0,0,0,...,1).

2.2.11 Perkalian Titik (*Dot Product*).

Jika **u** dan **v** adalah vektor tidak nol di R^2 atau R^3, maka perkalian titik (*dot product*), atau disebut juga *Euclidean inner product*, **u** dan **v** adalah $u \cdot v = ||u||\,||v||\,cos\,\theta$ dalam hal ini θadalah sudut antara **u** dan **v.** **u** · **v** = ($u_1v_1 + u_2v_2 + \ldots + u_nv_n$).

2.2.12 Perkalian Silang (*Cross Product*)

Cross Product adalah bentuk perkalian antara 2 vektor yang akan menghasilkan vektor yang tegak lurus dengan kedua vektor itu di dalam dimensi 3, yang didefinisikan dalam rumus:

$u \times v = ||u||\,||v||\,sin\,\theta$

2.3 *Cosine Similarity*

2.3.1 *Pengertian Cosine Similarity*

Cosine similarity merupakan salah satu metode yang berfungsi untuk membandingkan kemiripan antar dokumen, dalam hal ini yang dibandingkan adalah query dengan dokumen latih. Dalam menghitung *cosine similarity*, pertama yang dilakukan yaitu melakukan perkalian skalar antara query dengan dokumen kemudian dijumlahkan, setelah itu melakukan perkalian antara panjang dokumen dengan panjang query yang telah dikuadratkan, setelah itu di hitung

akar pangkat dua. Selanjutnya hasil perkalian skalar tersebut dibagi dengan hasil perkalian panjang dokumen dan query.

2.3.2 Rumus *Cosine Similarity*

$$sim(Q, D) = cos\ cos\ \theta = \frac{Q \cdot D}{\|Q\|\|D\|} = \frac{\sum_{i=1}^{n} Q_i \times D_i}{\sqrt{\sum_{i=1}^{n} (Q_i)^2} \times \sqrt{\sum_{i=1}^{n} (D_i)^2}}$$

Keterangan:

Q = Vektor Q (Query), yang akan dibandingkan kemiripannya

D = Vektor D (Dokumen), yang akan dibandingkan kemiripannya

$Q \cdot D$ = *dot product* antara vektor Q dan vektor D

$\|Q\|$ = Panjang vector Q

$\|D\|$ = Panjang vector D

BAB III

IMPLEMENTASI PROGRAM

3.1 stemming.py

File ini bertugas untuk menangani *stemming* dari tiap dokumen dan query. Berisi tiga fungsi, yaitu:

1. hanya_huruf : berfungsi untuk menyeleksi hanya huruf saja (baik huruf biasa maupun kapital) atau untuk menghilangkan semua simbol atau karakter selain huruf seperti koma “,”, titik “.”, dan lain-lain.
2. stemming_file : berfungsi untuk men-*stemming* dan stopword dokumen yang dimasukkan
3. stemming_query : berfungsi untuk men-*stemming* dan stopword query

3.2 prosesquery.py

File ini bertugas untuk menangani tiap dokumen dan query yang masuk lalu memprosesnya. File ini mempunyai satu *class,* yaitu Document dengan beberapa atribut dan method.

1. Atribut
    a. judul : untuk menyimpan judul tiap file
    b. url : untuk menyimpan url tiap file
    c. jml_kata : untuk menyimpan jumlah kata tiap file
    d. kata : untuk menyimpan kata yang ada setelah di *stemming*
    e. firstline : untuk menyimpan kalimat pertama tiap file
    f. similarity : untuk menyimpan hasil perhitungan similarity
    g. dict : untuk menyimpan kata yang ada di *database* dan jumlah kemunculannya di tiap file

2. Method

a. createDict : membuat *dictionary* tiap file
b. createSimilarity : menghitung similarity tiap file
c. getDict : mengambil *dictionary*
d. getURL : mengambil url
e. getJudul : mengambil judul
f. getKata : mengambil list kata
g. getFirstLine : mengambil kalimat pertama
h. getJmlKata : mengambil jumlah kata
i. getSimilarity : mengambil nilai perhitungan *similarity*

Selain *class* Document ada juga variabel dan fungsi global yang berfungsi untuk menangani query dan menghitung atribut class *Document* sebelum dimasukkan.

1. Variabel
    a. database : menyimpan semua kata yang pernah muncul di tiap file
    b. listOfDocuments : menyimpan *object Document* yang ada

2. Fungsi

a. dotProduct : menghitung perkalian dot antara dua vektor
b. lengthVector : menghitung panjang vektor
c. addToDatabase : mengirim hasil *stemming* tiap file ke dalam database
d. createDictQuery : membuat *dictionary* untuk query
e. createVecQuery : membuat vektor untuk query
f. sortSimilarity : mengurutkan *object Document* yang ada di listOfDocuments berdasarkan *similarity*-nya
g. calculateJmlKata : menghitung jumlah kata yang ada di tiap file
h. extraceFirstLine : mengambil kalimat pertama file

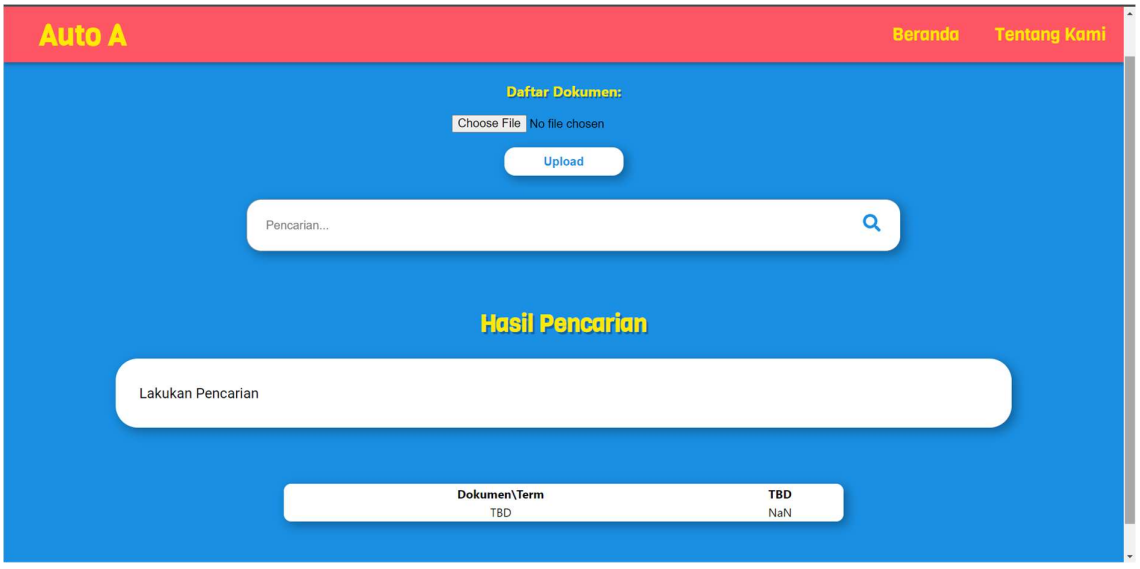
3.3 api.py

File ini bertugas untuk menggabungkan semua file di atas dan menghubungkan antara *front-end* dan *back-end.* File berisi beberapa fungsi, yaitu:

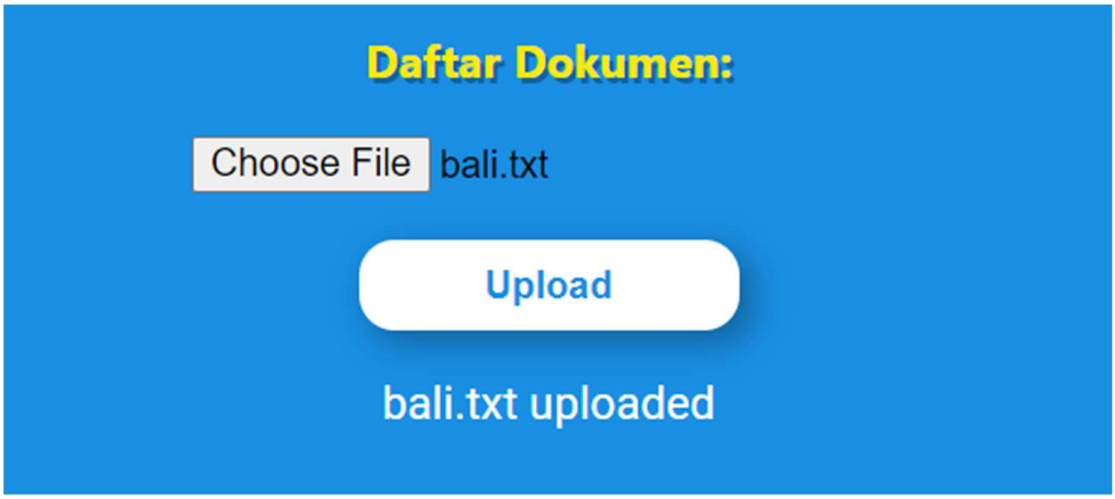
1. upload_file : berfungsi untuk mengupload file, memanggil fungsi *stemming* dan membuat *object* baru
2. query : berfungsi untuk mengambil input query dari user dan mengolahnya
3. getfile : berfungsi untuk men-*download* file yang ditampilkan di web

3.4 Garis Besar Program

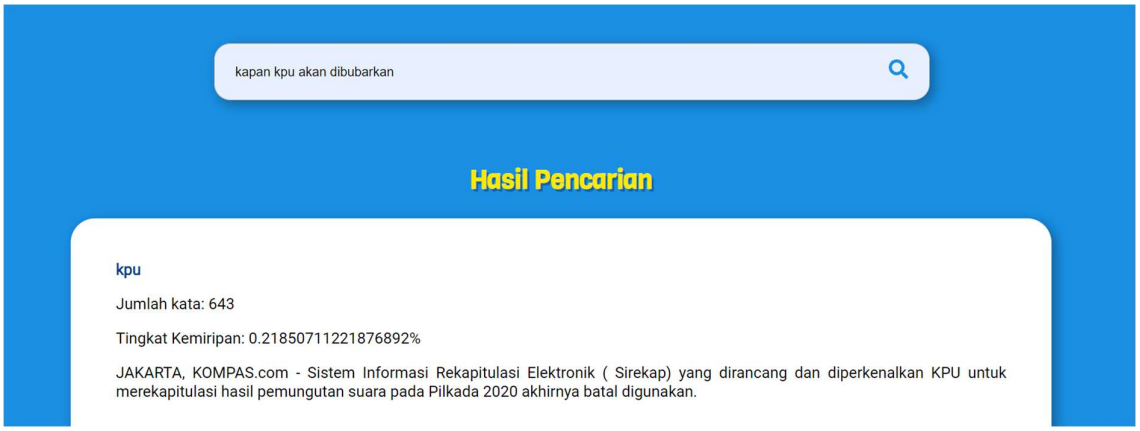
Setelah menjalankan flask dan node akan muncul web dengan tampilan seperti ini



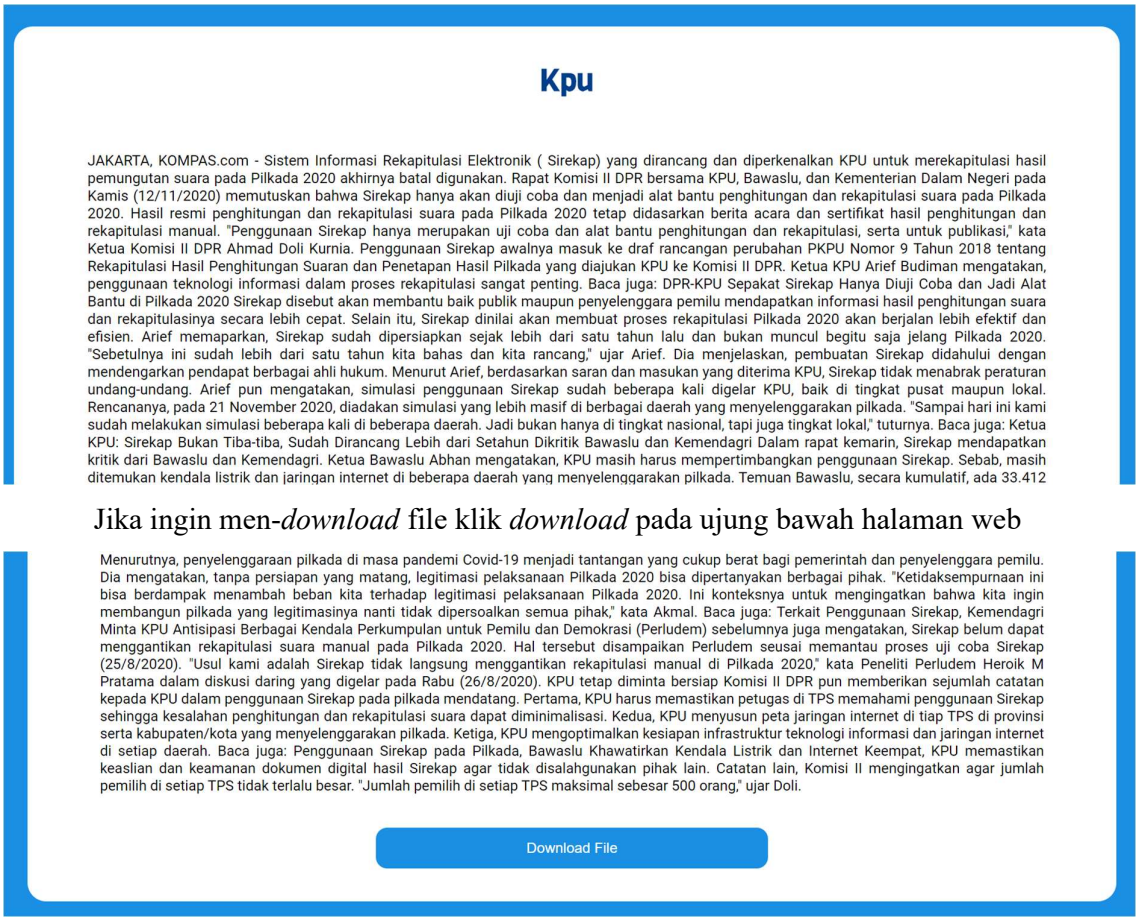
User meng-*upload* file-file yang ingin dipakai dengan cara mengklik tombol choose file lalu pilih file yang ingin di-*upload* setelah itu klik *upload*. Lakukan ini satu persatu tiap file sampai semua file yang ingin dipakai ter-*upload*. Ketika file sudah ter-*upload*, maka akan langsung di-*stemming* oleh program dan dimasukkan datanya ke kamus kata. Karena itu, jika ingin melakukan search harus terlebih dahulu melakukan *upload*, jika tidak maka tidak akan ada kamus kata.



Setelah itu, lakukan pencarian dengan mengisi query sesuai yang ingin dicari. Disini query akan di-*stemming* dan dibuat vektornya, setelah itu akan dihitung similarity antar dokumen dengan menggunakan rumus yang ada.



Lalu akan tampil hasil pencarian query terhadap file-file yang telah di-upload, klik judul pada file jika ingin membuka halaman file.

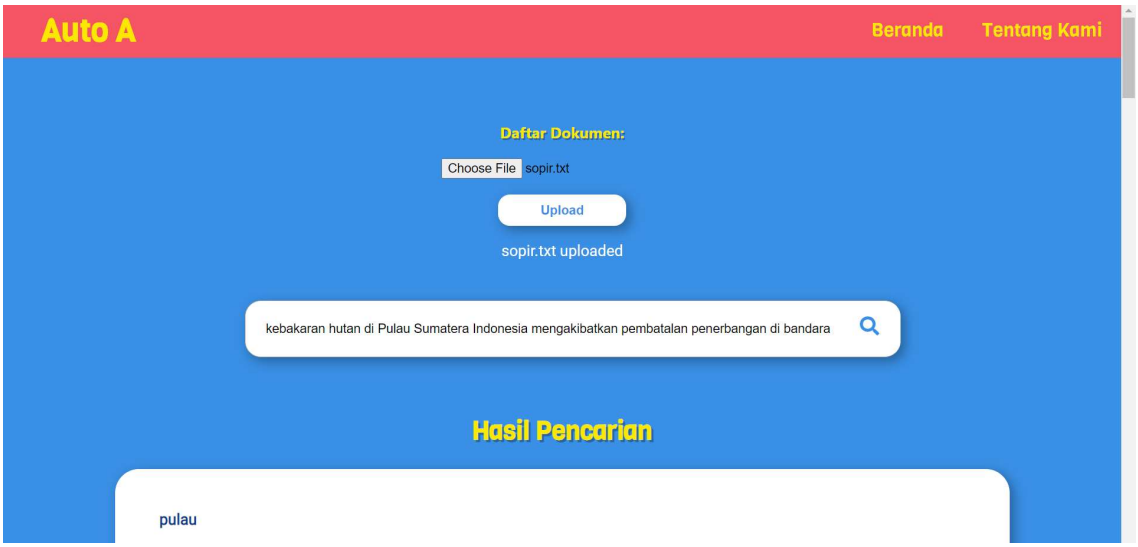


Jika ingin men-*download* file klik *download* pada ujung bawah halaman web

BAB IV

EKSPERIMEN

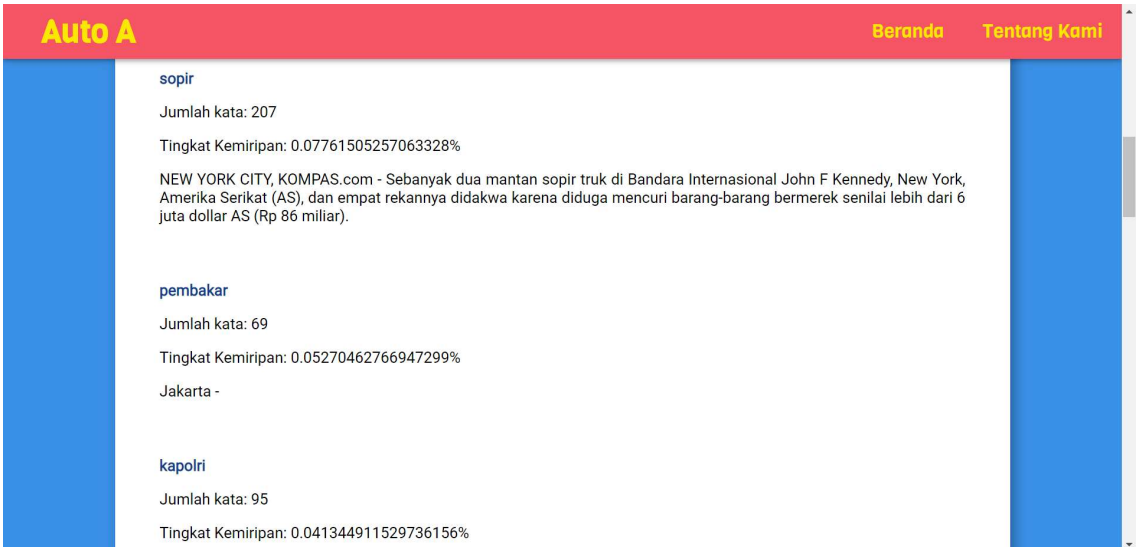
Di sini kami memasukkan *query* yaitu “kebakaran hutan di Pulau Sumatera Indonesia mengakibatkan pembatalan penerbangan di bandara”, *query* tersebut memiliki beberapa kata yang terdapat huruf kapital yaitu Pulau, Sumatera dan Indonesia, sehingga di *backend* diproses terlebih dahulu untuk semua simbol (koma “,”, titik “.”, dan sebagainya) dihilangkan/dihapus lalu huruf kapital dijadikan huruf biasa menjadi pulau, sumatera dan indonesia. Selanjutnya *query* tersebut memiliki sebelas kata (termasuk kata depan) kemudian *query* ini dilakukan *stopword* atau menghilangkan kata-kata umum atau tidak bermakna seperti aku, kamu, yang, lalu, dan lain-lain. Kemudian dilakukan *stemming* (menghapus kata depan, imbuhan, awalan, akhiran, dan sebagainya atau menjadikan kata dasar seperti yang terdapat di dalam *library* Sastrawi). Sehingga *query* tersebut tersisa sembilan kata yaitu kebakaran, hutan, pulau, sumatera, indonesia, akibat, batal, terbang, dan bandara.



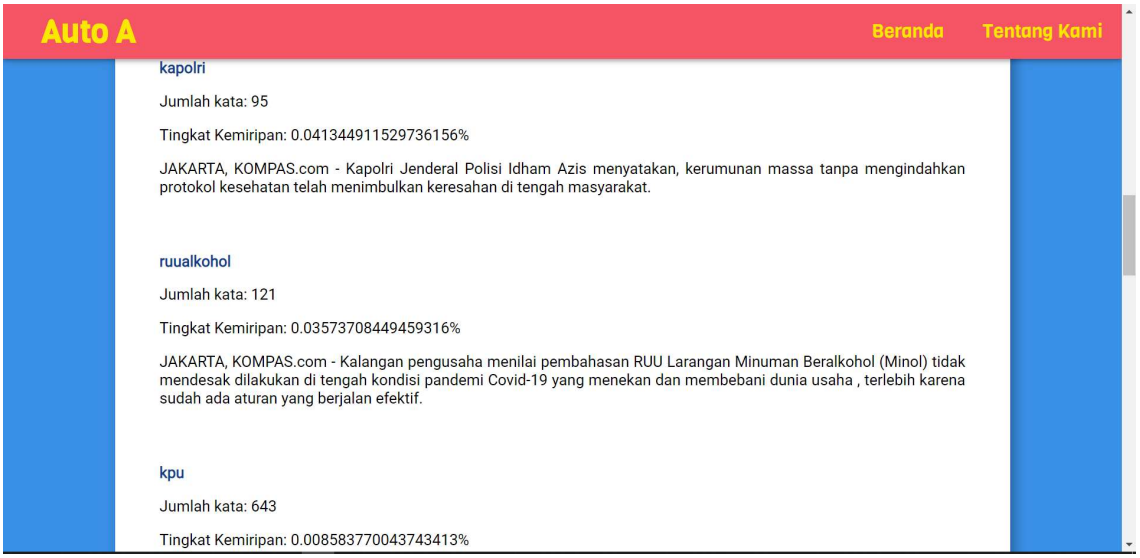
Setelah melakukan pencarian maka akan terdapat dokumen-dokumen yang sudah terurut berdasarkan urutan dari yang memiliki kemiripan paling besar sampai yang paling kecil. Setiap dokumen masing-masing menampilkan nama dokumen, jumlah kata dalam dokumen, kemiripan, dan kalimat awal.

Dokumen pulau terdapat di posisi paling atas artinya dokumen ini memiliki kemiripan yang paling besar dengan *query* dibandingkan dokumen lainnya. Dokumen pulau memiliki jumlah kata yaitu 201 kata dan kemiripan 0.15286%.

Di posisi kedua terdapat dokumen bali. Dokumen bali memiliki jumlah kata yaitu 343 kata dan kemiripan 0.12654%.

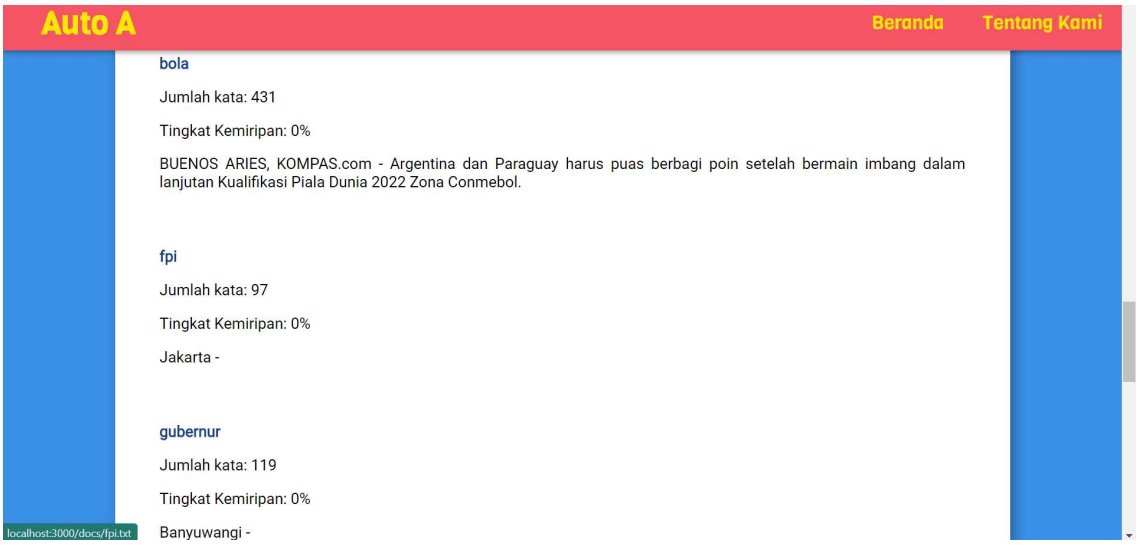


Posisi ketiga terdapat dokumen sopir. Dokumen sopir memiliki jumlah kata yaitu 207 kata dan kemiripan 0.0776%. Kemudian terdapat dokumen pembakar yang memiliki jumlah kata yaitu 69 kata dan kemiripan 0.0527%.

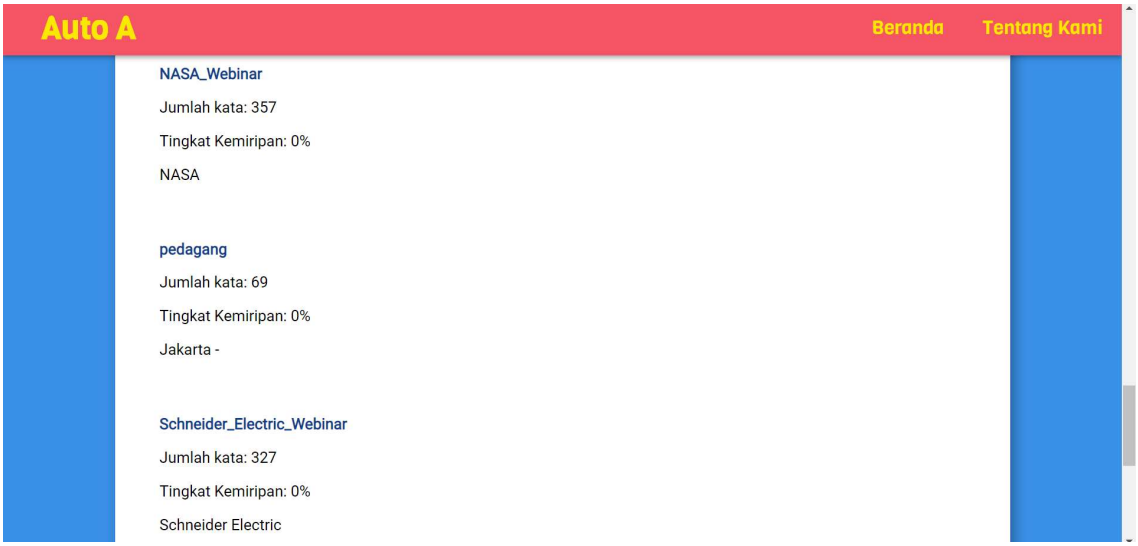


Dokumen kapolri memiliki jumlah kata yaitu 95 kata dan kemiripan 0.041345%. Dokumen ruualkohol memiliki jumlah kata yaitu 121 kata dan kemiripan 0.035737%.

Dokumen kpu memiliki jumlah kata yaitu 643 kata dan kemiripan 0.00858377%. Dokumen apple memiliki jumlah kata yaitu 1251 kata dan kemiripan 0% yang artinya tidak ada satupun kata dalam query yang mirip dengan dokumen.



Begitu juga dengan dokumen bola memiliki jumlah kata yaitu 431 kata dan kemiripan 0%. Dokumen fpi memiliki jumlah kata yaitu 97 kata dan kemiripan 0%. Dokumen gubernur memiliki jumlah kata yaitu 119 kata dan kemiripan 0%.



Dokumen NASA_Webinar memiliki jumlah kata yaitu 357 kata dan kemiripan 0%. Dokumen pedagang memiliki jumlah kata yaitu 69 kata dan kemiripan 0%. Dokumen Schneider_Electric_Webinar memiliki jumlah kata yaitu 327 kata dan kemiripan 0%.

Dokumen\Term	akibat	bakar	bandara	batal	hutan	indonesia	pulau	sumatera	terbang
query	1	1	1	1	1	1	1	1	1
pulau	0	0	0	0	0	0	7	0	0
bali	0	0	2	0	0	0	1	0	4
pembakar	0	1	0	0	0	0	0	1	0
sopir	0	0	3	0	0	0	0	0	0
kapolri	0	0	0	0	0	1	0	0	0
ruualkohol	0	0	0	0	0	1	0	0	0
kpu	0	0	0	1	0	0	0	0	0
apple	0	0	0	0	0	0	0	0	0
bola	0	0	0	0	0	0	0	0	0
fpi	0	0	0	0	0	0	0	0	0
gubernur	0	0	0	0	0	0	0	0	0
NASA_Webinar	0	0	0	0	0	0	0	0	0
pedagang	0	0	0	0	0	0	0	0	0
Schneider_Electric_Webinar	0	0	0	0	0	0	0	0	0

Dari sembilan kata/*term* diurutkan dari abjad yaitu akibat, bakar, bandara, batal, hutan, indonesia, pulau, sumatera dan terbang. Dari setiap dokumen juga ada jumlah masing-masing *term* atau kata. Di dalam dokumen pulau terdapat tujuh kata pulau, dalam dokumen bali terdapat dua kata bandara, 1 kata pulau, dan empat kata terbang dan seterusnya.

BAB V

PENUTUP

5.1 Kesimpulan

Kami berhasil membuat program mengenai Aplikasi *Dot Product* pada Sistem Temu-balik Informasi bernama Auto A. Dalam program kami dapat meng-*upload* dokumen yang diinginkan jadi dokumen bisa ditambah terus-menerus (dinamis) dan dapat menerima input *query* dari *user* yang mana hasil dari pencarian menampilkan daftar dokumen dari *similarity* atau kemiripan paling banyak berada di paling atas hingga kemiripan paling kecil berada di paling bawah.

5.1 Saran

Memakai dataframe atau sql untuk meng-*handle* kamus kata sehingga bisa mempercepat pemrosesan.

5.1 Refleksi

Proses pembuatan kamus dapat dipercepat.

REFERENSI

“Aplikasi Dot Product pada sistem temu balik aplikasi” by Rinaldi Munir”
https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-12-Aplikasi-dot-product-pada-IR.pdf

“Sastrawi Library”
https://pypi.org/project/Sastrawi/

“Cosine Similarity”
https://www.payahtidur.com/project/cosine-similarity