LAPORAN TUGAS BESAR
IF2123
ALJABAR LINEAR DAN GEOMETRI

Disusun oleh:

Ronggur Mahendra Widya Putra (13519008)
Muhammad Furqon(13519184)
Ahmad Saladin(13519187)

TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2020/2021

BAB I
Deksripsi Masalah

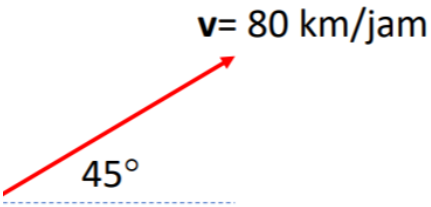
Buatlah program mesin pencarian dengan sebuah website lokal sederhana. Spesifikasi program adalah sebagai berikut:

1. Program mampu menerima search query. Search query dapat berupa kata dasar
2. Dokumen yang akan menjadi kandidat dibebaskan formatnya dan disiapkan secara manual. Minimal terdapat 15 dokumen berbeda sebagai kandidat dokumen. Bonus: Gunakan web scraping untuk mengekstraksi dokumen dari website.
3. Hasil pencarian yang terurut berdasarkan similaritas tertinggi dari hasil teratas hingga hasil terbawah berupa judul dokumen dan kalimat pertama dari dokumen tersebut. Sertakan juga nilai similaritas tiap dokumen.
4. Program disarankan untuk melakukan pembersihan dokumen terlebih dahulu sebelum diproses dalam perhitungan cosine similarity. Pembersihan dokumen bisa meliputi hal-hal berikut ini.
    a. Stemming dan Penghapusan stopwords dari isi dokumen.
    b. Penghapusan karakter-karakter yang tidak perlu.
5. Program dibuat dalam sebuah website lokal sederhana. Dibebaskan untuk menggunakan framework pemrograman website apapun. Salah satu framework website yang bisa dimanfaatkan adalah Flask (Python), ReactJS, dan PHP.
6. Kalian dapat menambahkan fitur fungsional lain yang menunjang program yang anda buat (unsur kreativitas diperbolehkan/dianjurkan).
7. Program harus modular dan mengandung komentar yang jelas.
8. Dilarang menggunakan library cosine similarity yang sudah jadi.

BAB II
Teori Singkat

Vektor

Vektor adalah suatu kuantitas fisik yang memiliki besar dan arah. Contoh: kecepatan (v) mobil 80 km/jam ke arah timur laut



gambar1:Contoh Vektor

Vektor dilambangkan dengan huruf-huruf kecil (dicetak tebal) atau memakai tanda panah (jika berupa tulisan tangan) Contoh, u, v, w, … atau u, v ђ , w, … a, b, c, …

Contoh, **u**, **v**, **w**, … atau $\vec{u}$, $\vec{v}$, $\vec{w}$, …

a, **b**, **c**, …

gambar2:Notasi Vektor

Ruang Vektor

Ruang vektor adalah ruang tempat vektor didefinisikan. Ruang vektor disebut juga sebagai ruang Euclidean. Ruang vektor berdimensi dua dinotasikan sebagai R^2. Ruang vektor berdimensi n dinotasikan R^n.

Vektor di R^n:

$\mathbf{v} = (v_1, v_2, \ldots, v_n)$ atau $\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$

gambar3:Vektor di R^n.

Dot Product

Jika u dan v adalah vektor tidak nol di R2 atau R 3 , maka perkalian titik (dot product), atau disebut juga Euclidean inner product, u dan v adalah

$$\mathbf{u} \cdot \mathbf{v} = \|\mathbf{u}\|\|\mathbf{v}\| \cos\theta$$

gambar4:Perkalian titik (dot product) dari vektor u dan v

yang dalam hal ini θ adalah sudut yang dibentuk oleh u dan v.

Hasil perkalian dari dot product adalah skalar.

Information Retrieval dengan Model Ruang Vektor

Salah satu model IR adalah model ruang vektor. Model ini menggunakan teori di dalam aljabar vector dengan cara menghitung kata. Misalkan terdapat n kata berbeda sebagai kamus kata (vocabulary) atau indeks kata (term index). Kata-kata tersebut membentuk ruang vektor berdimensi n.
Setiap dokumen maupun query dinyatakan sebagai vektor w = (w1, w2,..., wn) di dalam R^n
wi = bobot setiap kata i di dalam query atau dokumen
Nilai wi dapat menyatakan jumlah kemunculan kata tersebut dalam dokumen (term frequency)

Contoh: Misalkan terdapat tiga buah kata (T1, T2, dan T3), dua buah dokumen (D1 dan D2) serta sebuah query Q. Masing-masing dinyatakan sebagai vektor:
D1 = (2, 3, 5), D2 = (3, 7, 1), Q = (0, 0, 2)
D1 = (2, 3, 5) artinya dokumen D1 mengandung 2 buah kata T1, 3 buah kata T2, dan 5 buah kata T3
.
Contoh: Misalkan T1 = Menteri, T2 = minta, T3 = Korupsi
D1= Menteri olahraga meminta maaf atas perbuatan korupsi. Menteri tersebut terlibat korupsi anggaran. Meminta-minta komisi termasuk korupsi. Korupsi sudah mandarah daging di Indonesia. Korupsi sudah menjadi budaya.

D2 = (3, 7, 1) artinya dokumen D2 mengandung 3 buah kata T1, 7 buah kata T2, dan satu buah kata T3.
Contoh: D2= Gubernur Jabar meminta waktu ketemu Menteri Sosial. Dia meminta Pak Menteri mengunjungi panti. Permintaan yang wajar. Sekretaris Gubernur mengirim surat permintaan kepada Menteri tersebut. Apakah meminta-minta termasuk perbuatan korupsi? Tidak selalu, bukan? Meminta waktu saja.
Q = (0, 0, 2) artinya query Q hanya mengandung 2 buah kata T3
Contoh: Q = Korupsi besar atau kecil tetap saja korupsi.

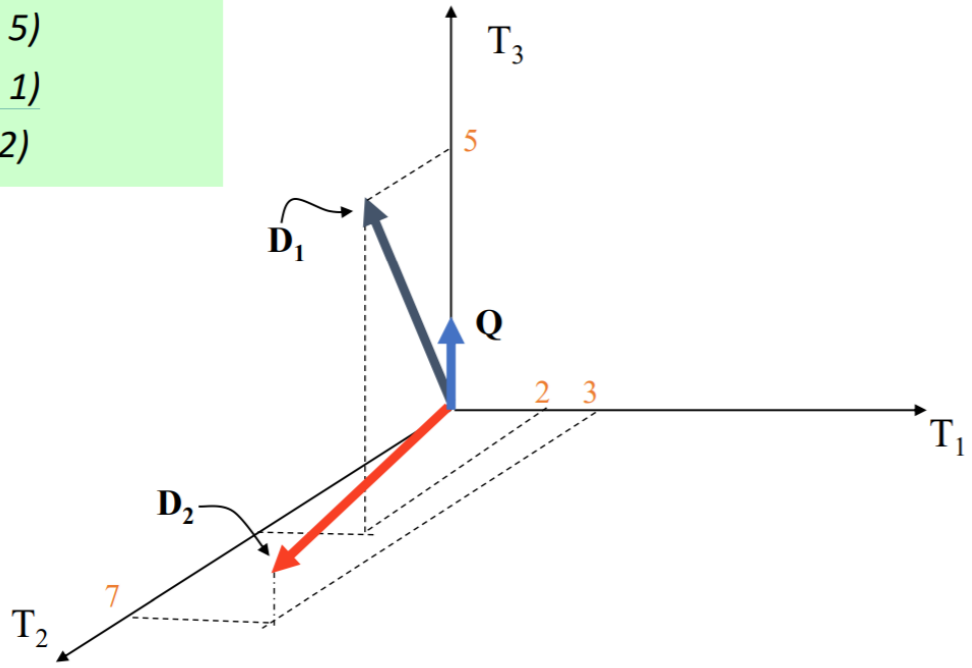
Representasi grafik vektor dari D1,D2, dan Q dapat dilihat pada gambar di bawah

Contoh:

$D_1 = (2, 3, 5)$

$D_2 = (3, 7, 1)$

$Q = (0, 0, 2)$



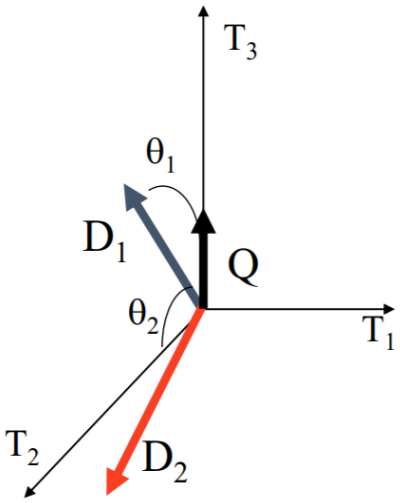
gambar4:Representasi grafik vektor dari D1,D2, dan Q

Cosine Similarity

Penentuan dokumen mana yang relevan dengan 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 (sim) antara dua vektor Q = (q1 , q2 , …, qn) dan D = (d1 , d2 , …, dn) diukur dengan rumus cosinus similarity yang merupakan bagian dari rumus perkalian titik (dot product) dua buah vektor:

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

gambar5:Cosine similarity

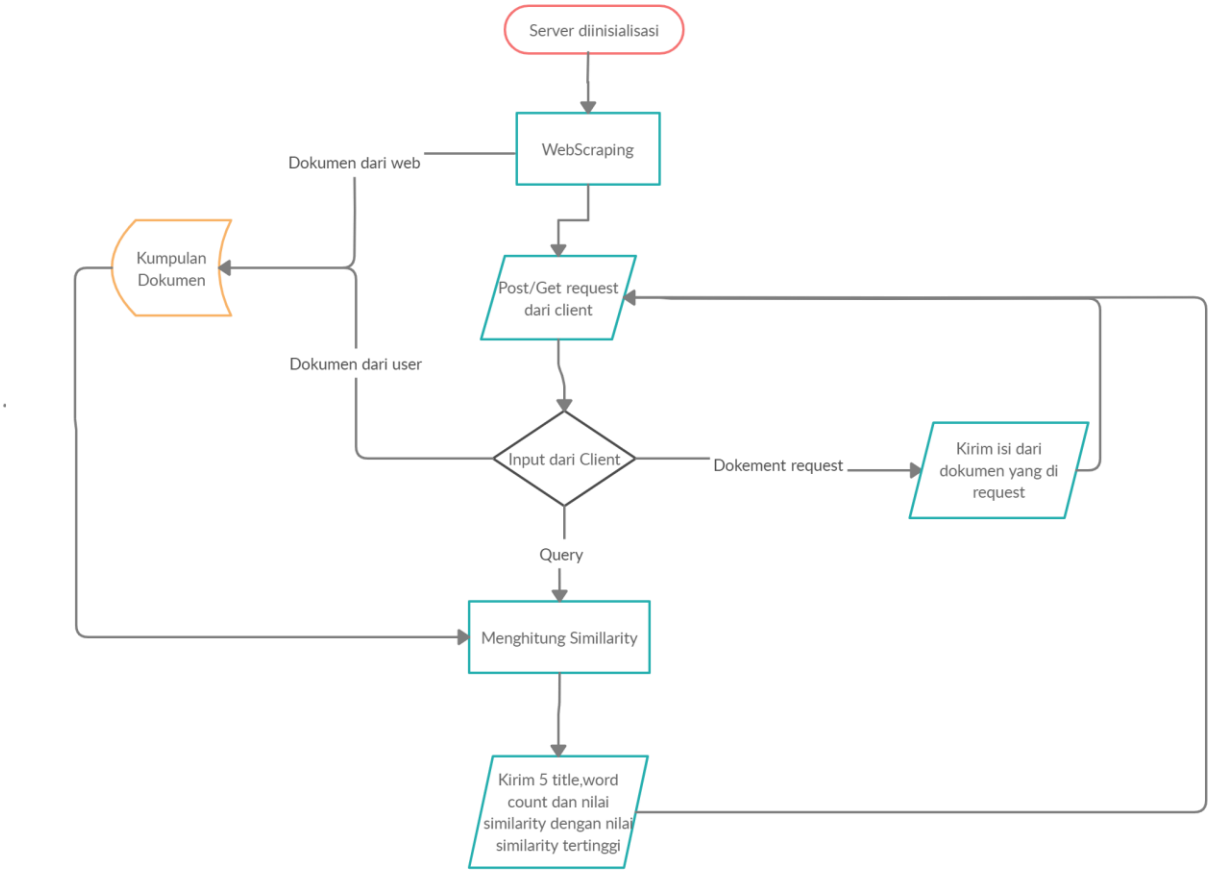


$$sim(\mathbf{Q}, \mathbf{D}) = \cos\theta = \frac{\mathbf{Q} \cdot \mathbf{D}}{\|\mathbf{Q}\|\|\mathbf{D}\|}$$

gambar6:Cosine similarity

BAB III
Implementasi

Kami membuat suatu aplikasi yang terdiri dari server dan client. Aplikasi kami menerima input dari user melalui client lalu mengirimkan data tersebut ke server. Data tersebut bisa berupa *query*, dokumen dari *user*, dan *request* untuk dokumen. Aplikasi kami bekerja seperti diagram dibawah ini :



Kami memanfaatkan *framework* ReactJS sebagai frontend (atau *client-side*) dengan bahasa javascript dan Flask sebagai backend*(atau server-side)* dengan bahasa python. Keduanya berada dalam sebuah React Flask App dalam sebuah direktori react-flask-app sebagai berikut :

React Flask App
Frontend:direktori src/src

1.App.js

Bagian utama frontend yang berhubungan dengan backend. Query yang diterima akan dikirimkan ke bagian backend untuk dilakukan kalkulasi. Hasil dari backend akan dikirim ke frontend.

2.App.css

Berfungsi untuk mengatur penampilan dari halaman web.

3.Index.js

Berfungsi untuk me render UI dari aplikasi kami, dan routing dari page awal ke page dokumen.

4.Doc1.js

Berfungsi untuk melihat dokumen dengan similarity tertinggi.

5.Doc2.js

Berfungsi untuk melihat dokumen dengan similarity kedua tertinggi.

6.Doc3.js

Berfungsi untuk melihat dokumen dengan similarity ketiga tertinggi.

7.Doc4.js

Berfungsi untuk melihat dokumen dengan similarity keempat tertinggi.

8.Doc5.js

Berfungsi untuk melihat dokumen dengan similarity kelima tertinggi.

9.perihal.js

Berfungsi untuk menampilkan perihal dari dari aplikasi kami, seperti konsep singkat, cara menggunakan, dan identitas kami.

Backend:direktori src/api

Backend berfungsi untuk mengkalkulasikan hasil dari query yang diinput oleh user serta melakukan *web-scraping*, yaitu mengambil dari artikel di internet. Artikel yang diambil dari dua sumber artikel berbahasa Indonesia. Artikel bersumber dari Kompas.com dan Tribunnews.com, yang diambil adalah artikel yang masuk ke dalam daftar artikel populer.

1.api.py

Berfungsi sebagai program utama backend. Menerima query dari pengguna, lalu menghitung similarity menggunakan fungsi dari dua file lainnya. Hasil dari perhitungan akan dikirimkan ke frontend.

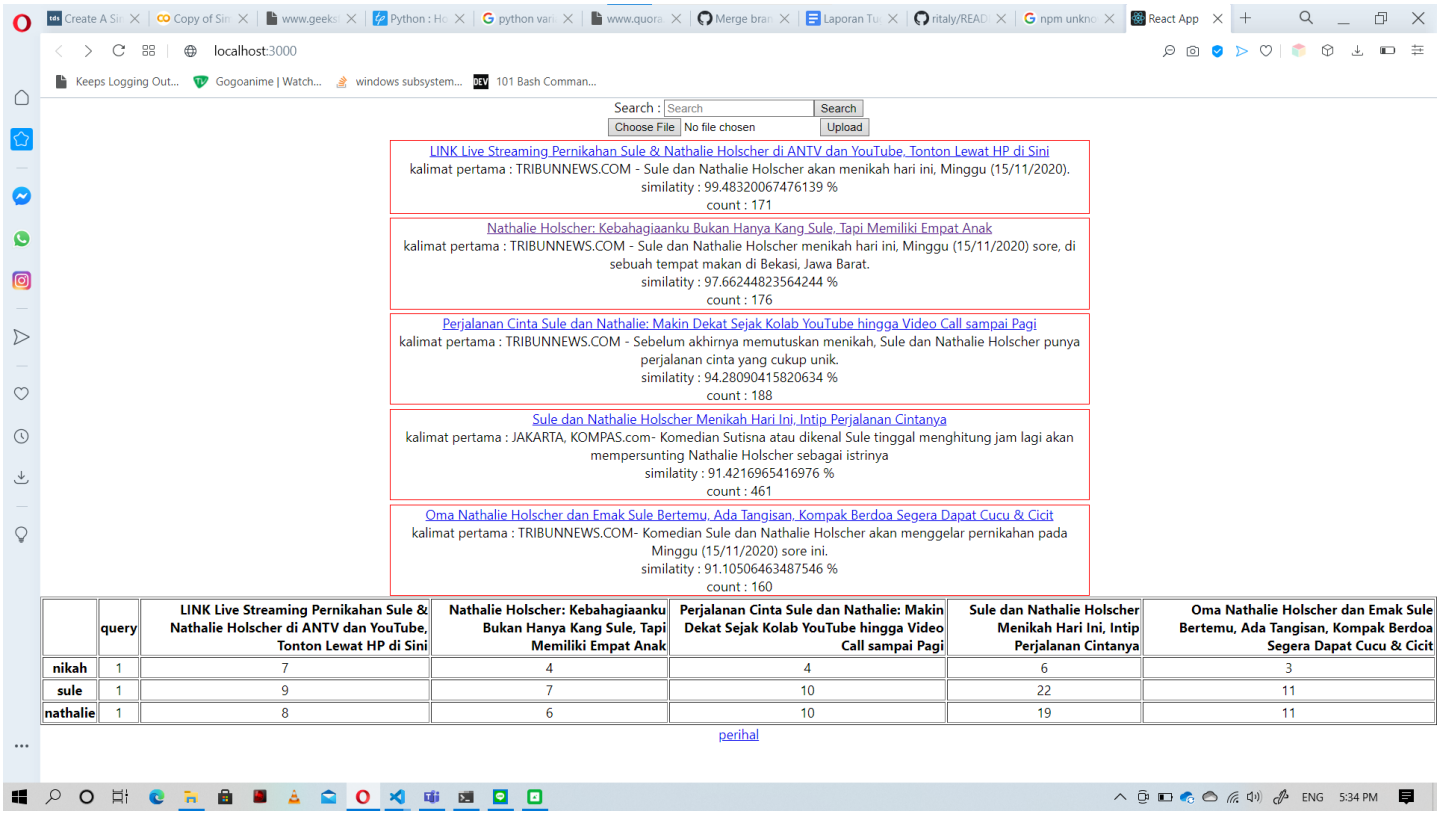
2.bacafile.py

Berfungsi untuk membaca file yang telah diupload sehingga dimasukkan ke dalam perhitungan similarity.

3.search.py

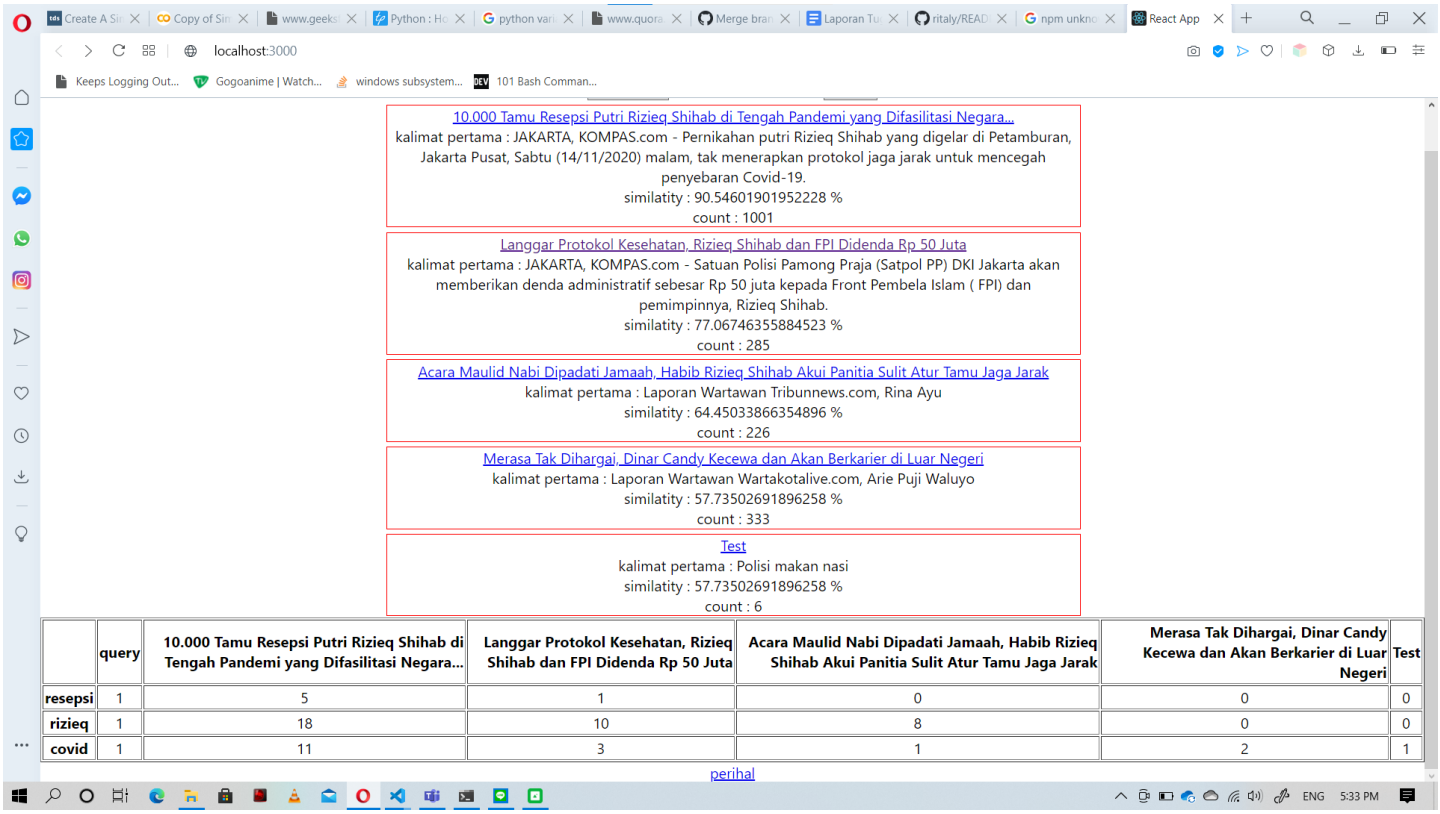
Berfungsi untuk menghitung similarity dengan menggunakan ruang vektor. Di dalam file ini juga terdapat fungsi webscraping yang mengambil artikel paling populer dari dua sumber. Sumber pertama adalah Kompas.com dan sumber yang kedua adalah Tribunnews.com. Selain itu terdapat juga fungsi untuk membersihkan dokumen dan melakukan stemming supaya perhitungan similarity lebih akurat.

BAB IV
Eksperimen



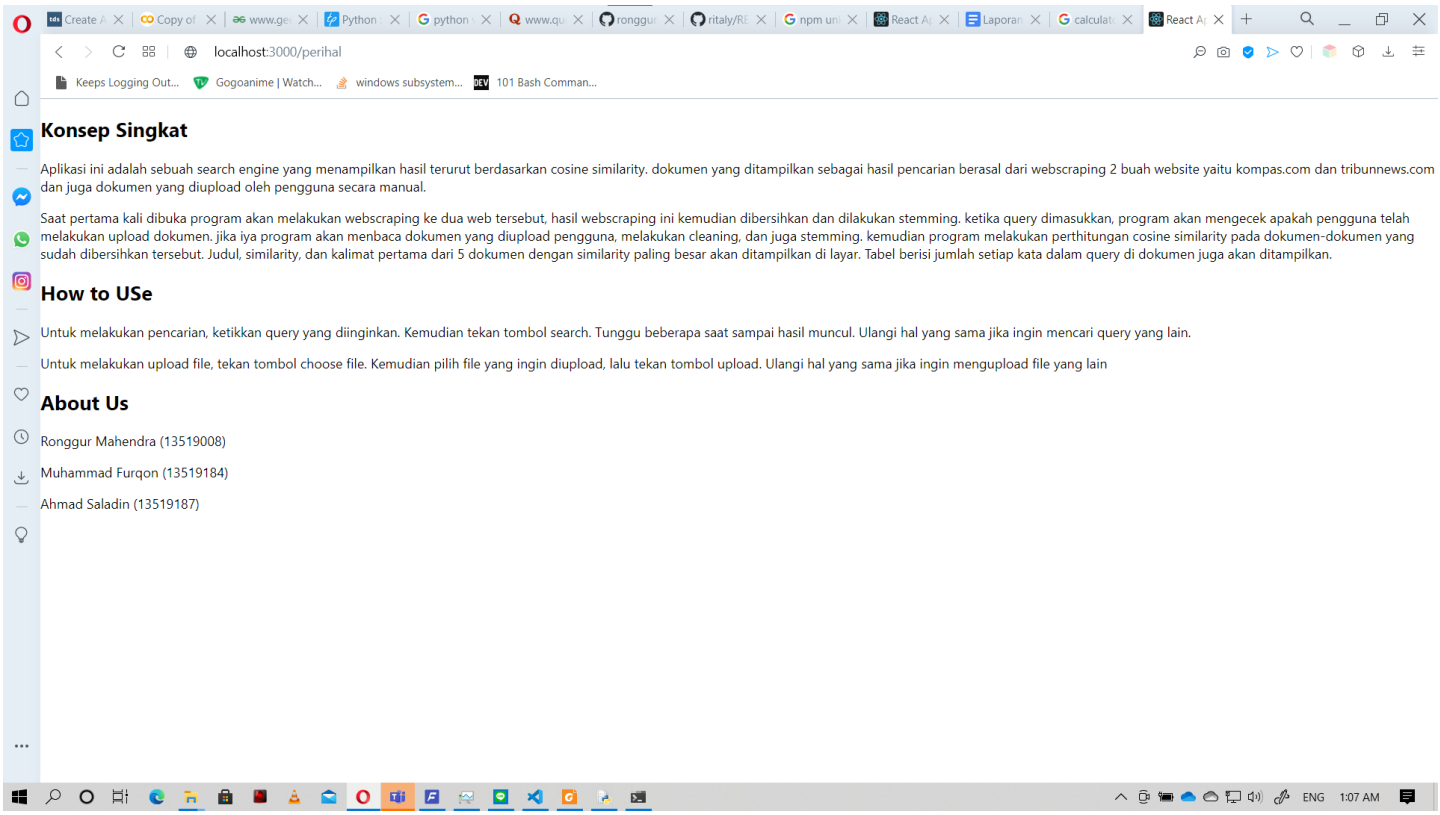
	query	LINK Live Streaming Pernikahan Sule & Nathalie Holscher di ANTV dan YouTube, Tonton Lewat HP di Sini	Nathalie Holscher: Kebahagiaanku Bukan Hanya Kang Sule, Tapi Memiliki Empat Anak	Perjalanan Cinta Sule dan Nathalie: Makin Dekat Sejak Kolab YouTube hingga Video Call sampai Pagi	Sule dan Nathalie Holscher Menikah Hari Ini, Intip Perjalanan Cintanya	Oma Nathalie Holscher dan Emak Sule Bertemu, Ada Tangisan, Kompak Berdoa Segera Dapat Cucu & Cicit
nikah	1	7	4	4	6	3
sule	1	9	7	10	22	11
nathalie	1	8	6	10	19	11

Pada eksperimen pertama query yang dimasukkan adalah “Pernikahan sule nathalie”. Program mengembalikan 5 buah dokumen dengan similarity paling besar. Vektor dokumen-dokumen ini berturut-turut adalah (7,9,8), (4,7,6), (4,10,10), (6,22,19), dan (3,11,11). Jika dihitung secara manual menggunakan data vektor tersebut akan didapat similarity berturut 0.9948, 0.9766, 0.9428, 0.9142, 0.9110. Jadi dapat disimpulkan bahwa algoritma untuk menghitung cosine similarity yang digunakan sudah benar.

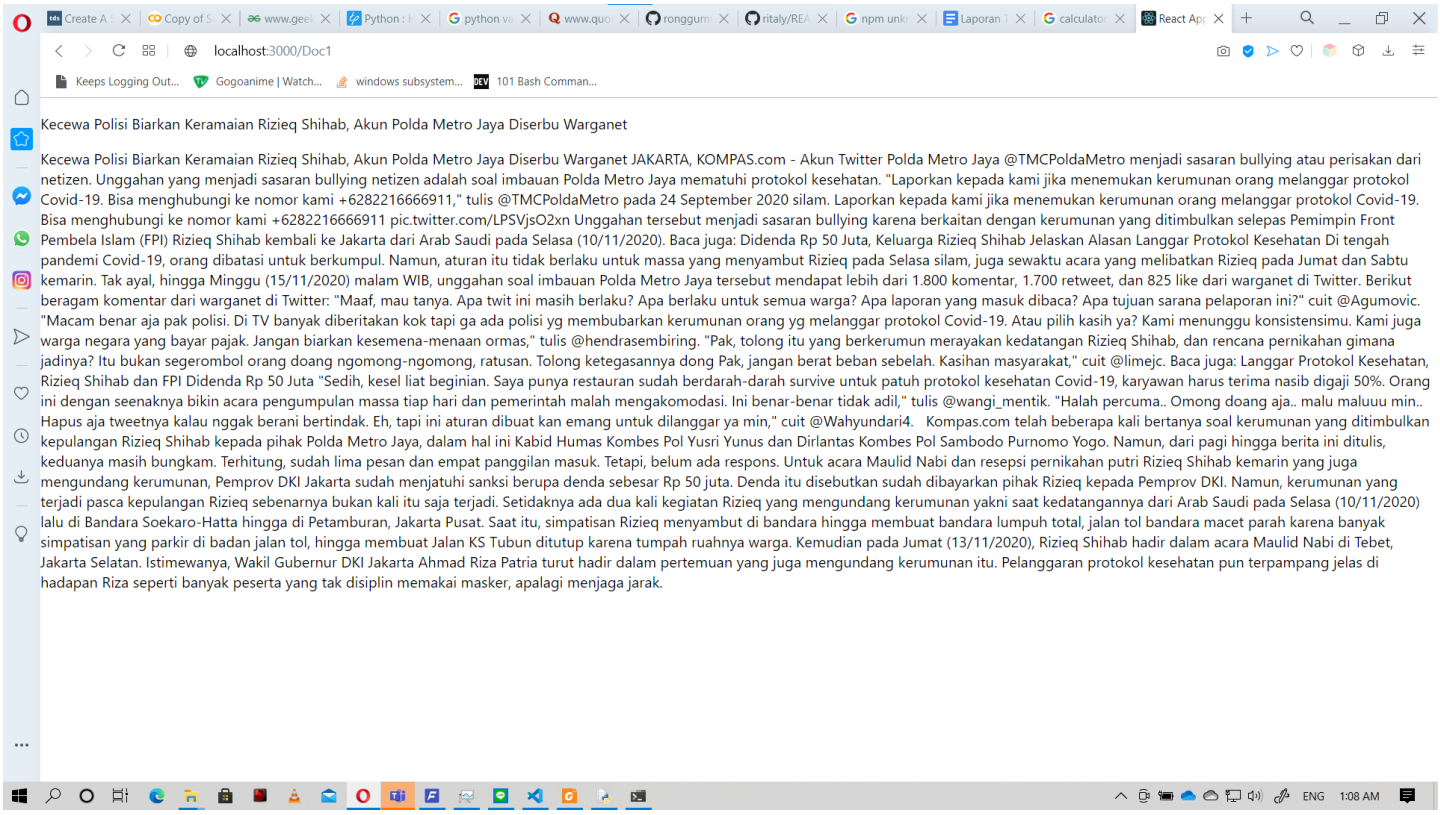


	query	10.000 Tamu Resepsi Putri Rizieq Shihab di Tengah Pandemi yang Difasilitasi Negara...	Langgar Protokol Kesehatan, Rizieq Shihab dan FPI Didenda Rp 50 Juta	Acara Maulid Nabi Dipadati Jamaah, Habib Rizieq Shihab Akui Panitia Sulit Atur Tamu Jaga Jarak	Merasa Tak Dihargai, Dinar Candy Kecewa dan Akan Berkarier di Luar Negeri	Test
resepsi	1	5	1	0	0	0
rizieq	1	18	10	8	0	0
covid	1	11	3	1	2	1

Pada eksperimen ini query yang digunakan adalah “resepsi riqieq covid”. Sama seperti eksperimen sebelumnya, hasil perhitungan cosine similarity juga sudah sesuai dengan hasil perhitungan manual.



Gambar di atas adalah tampilan dari halaman perihal yang ada di aplikasi ini.



Gambar di atas adalah contoh salah satu dokumen yang akan ditampilkan jika link judul pada halaman utama ditekan.

BAB V
Kesimpulan, Saran, Refleksi

Kesimpulan

Permasalahan membuat *search engine* atau sebuah mesin pencarian berdasarkan input query dari pengguna dari file yang telah diunggah dan artikel dari web yang diperoleh dengan metode *web-scraping*, dengan membersihkan dokumen atau artikel, menghitung kata, melakukan *word stemming*, dan menghasilkan keluaran hasil berupa lima hasil teratas, dapat diselesaikan dengan program dalam bentuk web lokal yang telah dibuat.

Saran

Sebagai saran pengembangan, program yang telah dibuat dapat diberikan tampilan yang lebih *user friendly*.

Refleksi

Dari tugas ini kami mendapatkan banyak ilmu mengenai bahasa pemrograman Python dan javascript, pentingnya kerja sama, komunikasi yang baik dengan sesama anggota, ilmu mengenai *web development*.

Daftar Pustaka

http://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Tubes2-Algeo-2020.pdf
http://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-12-Aplikasi-dot-product-pada-IR.pdf
http://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-10-Vektor-di-Ruang-Euclidean-Bag1.pdf

React-Flask:
https://blog.miguelgrinberg.com/post/how-to-create-a-react--flask-project

UploadFiles:
https://programmingwithmosh.com/javascript/react-file-upload-proper-server-side-nodejs-easy/
https://medium.com/excited-developers/file-upload-with-react-flask-e115e6f2bf99

ReadFiles:
https://www.w3schools.com/python/python_file_open.asp

SearchEngine:
https://towardsdatascience.com/create-a-simple-search-engine-using-python-412587619f5

CosineSimilarity:
https://masongallo.github.io/machine/learning,/python/2016/07/29/cosine-similarity.html

DF to HTML:
https://stackoverflow.com/questions/22180993/pandas-dataframe-display-on-a-webpage

Stemming:
https://pypi.org/project/Sastrawi/

Webscraping:
https://www.crummy.com/software/BeautifulSoup/bs4/doc/