LAPORAN TUGAS BESAR

Diajukan untuk memenuhi tugas besar

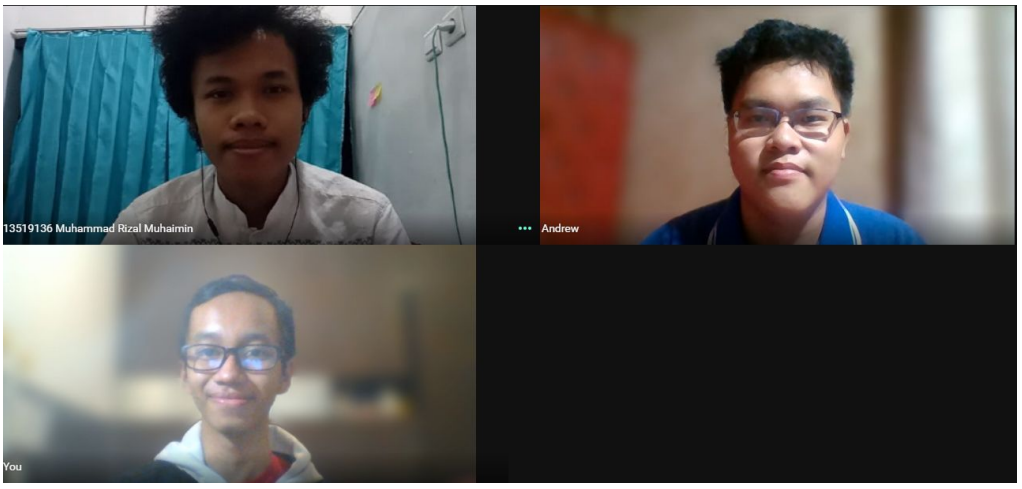
Matakuliah IF2123 Aljabar Linear dan Geometri

Oleh:

13519036 Andrew

13519076 Moses Ananta

13519136 Muhammad Rizal Muhaimin



PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2020

BAB I DESKRIPSI 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 maupun berimbuhan.
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

Information Retrieval adalah menemukan kembali informasi yang relevan terhadap kebutuhan pengguna dari suatu kumpulan informasi secara otomatis. Information Retrieval memiliki beberapa metode dalam mengambil data dan informasi, antara lain:

1. Inverted Index
   Inverted Index adalah sebuah struktur data index yang dibangun untuk memudahkan query pencarian yang memotong tiap kata (term) yang berbeda dari suatu daftar term dokumen. Inverted Index memiliki tujuan untuk meningkatkan kecepatan dan efisiensi dalam melakukan pencarian pada sekumpulan dokumen dan menemukan dokumen-dokumen yang memiliki query user.
2. Boolean Retrieval
   Boolean Retrieval merupakan proses pencarian informasi dari *query* yang menggunakan ekspresi Boolean. Dengan ekspresi boolean dengan menggunakan operator logika AND, OR dan NOT. Dalam menentukan hasil perhitungannya hanya berupa nilai binary (1 atau 0). Hasil boolean retrieval yang ada hanya dokumen relevan atau tidak sama sekali. Sehingga keunggulan dari boolean retrieval tidak menghasilkan dokumen yang sama.
3. Tokenization
   Tokenization adalah metode pemecah teks menjadi token-token yang berurutan. Tokenisasi secara garis besar memecah sekumpulan karakter dalam suatu teks ke dalam satuan kata, bagaimana membedakan karakter-karakter tertentu yang dapat diperlakukan sebagai pemisah kata atau bukan.
4. Stemming and Lemmatization
   Stemming adalah proses untuk mendapatkan kata dasar dengan cara menghapus imbuhan kata. Proses stem adalah bagian dari pra-pemrosesan dokumen teks dalam pencarian informasi. Stemming merupakan sebuah proses yang bertujuan untuk mereduksi jumlah variasi dalam representasi dari sebuah kata
   Lemmatization adalah proses menemukan bentuk dasar dari sebuah kata. *Lemmatization* adalah proses normalisasi pada teks/kata dengan berdasarkan pada

bentuk dasar yang merupakan bentuk lemma-nya. Normalisasi adalah mengidentifikasikan dan menghapus prefiks serta sufiks dari sebuah kata.

5. Dictionaries
   Dictionaries adalah penggunaan struktur data hash dan tree untuk melakukan pemrosesan query.
6. Wildcard Queries
   Wildcard Queries adalah penggunaan simbol "*" pada saat menuliskan query. Simbol "*" digunakan saat user tidak yakin dengan query yang ia tulis.
7. Vector Space Model
   Vector Space Model adalah teknik dasar dalam perolehan informasi yang dapat digunakan untuk penilaian relevansi dokumen terhadap kata kunci pencarian (query). Vector Space Model adalah representasi kumpulan dokumen sebagai vektor dalam sebuah ruang vektor.

Vektor didefinisikan sebagai sebuah kuantitas fisik yang memiliki besaran dan arah. Vektor dapat dilambangkan dengan huruf-huruf kecil yang dicetak tebal atau memakai tanda panah. Ruang vektor adalah ruang tempat vektor didefinisikan.

Salah satu model dari Information Retrieval adalah Vector Space Model. Sesuai dengan definisinya, semua kata yang terdapat di dalam suatu query atau dokumen akan direpresentasikan sebagai vektor di dalam sebuah ruang vektor. Misalkan sebuah query atau dokumen memiliki sejumlah n kata berbeda sebagai sebuah kamus kata atau indeks kata, kata-kata tersebut membentuk sebuah ruang vektor berdimensi n. Setiap query atau dokumen dinyatakan sebagai vektor w

$$w = (w_1,\ w_2,\ ...,\ w_n)$$

w_i adalah bobot atau jumlah kemunculan kata untuk setiap kata i di dalam query atau dokumen.

Penentuan dokumen mana yang relevan dengan query dilihat sebagai pengukuran kesamaan antara query dan dokumen. Kesamaan antara vektor query dengan vektor dokumen diukur dengan rumus cosine similarity yang merupakan bagian dari rumus perkalian titik (dot product) antara dua vektor.

$$Q \bullet D = ||Q||\ ||D||\ cos\ \theta$$

$$CosineSimilarity(Q, D) = cos\ \theta\ = \frac{Q \bullet D}{||Q||\ ||D||}$$

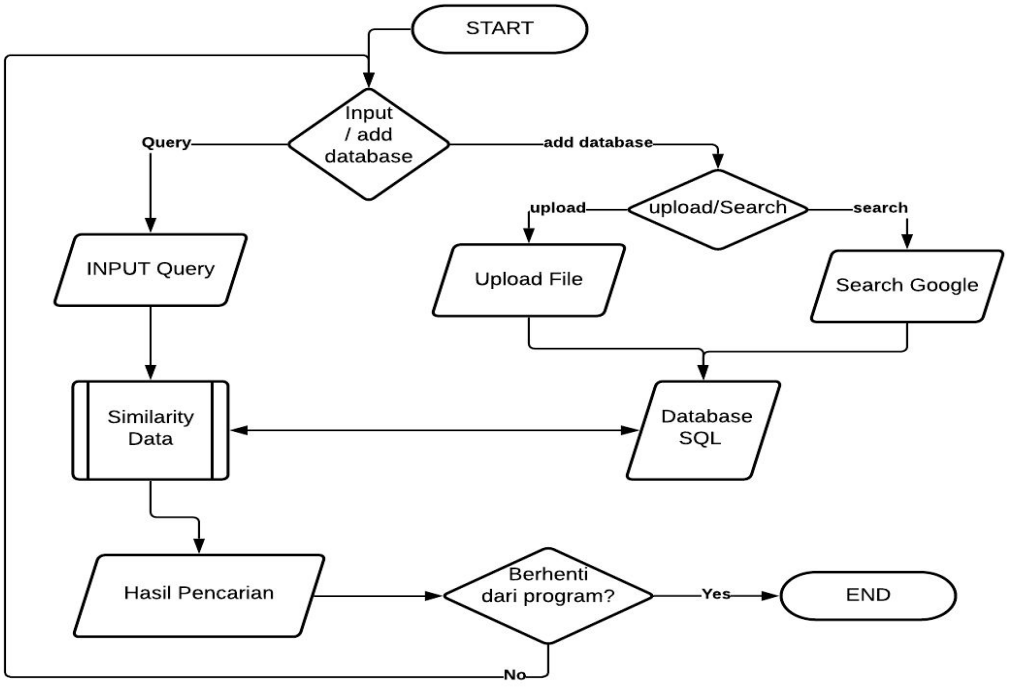
Jika $cos\ \theta = 1$, berarti $\theta = 0$, menandakan bahwa vektor query dan vektor dokumen berimpit, yang berarti dokumen sesuai dengan query. Semakin mendekati 1 nilai $cos\ \theta$, maka dokumen akan semakin relevan dengan query yang berkorespondensi. Selanjutnya, setiap dokumen akan diurut berdasarkan tingkat relevansi dengan query. Pengurutan setiap dokumen dilakukan terurut mengecil dari dokumen yang paling relevan menuju ke dokumen yang paling kurang relevan.

BAB III IMPLEMENTASI

3.1. Garis Besar Program

Program menggunakan HTML dan CSS sebagai Front-end, dan Flask Python sebagai Back-end. Program juga menggunakan SQLite sebagai tempat penyimpanan dokumen. Sebelum menggunakan search engine, user mengikuti instruksi install yang terdapat di dalam README.md. Setelah semua hal yang diperlukan untuk menjalankan program sudah di-install, user dapat menjalankan "flask run" untuk menggunakan search engine. User membuka localhost, lalu memasukkan query yang ingin di search. Setelah user menekan tombol "Mari kita cari!", program akan melakukan proses kalkulasi cosine similarity dari dokumen-dokumen yang terdapat di dalam database dengan query yang dimasukkan. Kemudian, program akan menampilkan urutan dokumen-dokumen berdasarkan tingkat kemiripannya terhadap query, terurut dari yang paling mirip sampai dengan yang paling tidak mirip. Jika similarity dari suatu dokumen dengan query yang dimasukkan tidak ada, atau $CosineSimilarity(Q, D) = 0$, maka dokumen tersebut tidak akan ditampilkan. Untuk menambahkan dokumen ke dalam database, user mengklik hyperlink "Tambahkan dokumen". User memiliki dua pilihan untuk menambahkan dokumen ke dalam database. Pilihan pertama yaitu menambahkan file .html atau .txt ke dalam tempat upload file yang telah disediakan. Pilihan kedua yaitu dengan menggunakan search engine google.

User juga dapat mengklik hyperlink "Tentang kami dan rip-off kami" untuk mengetahui secara singkat bagaimana program bekerja, bagaimana cara menggunakan program, dan anggota kelompok beserta tugas yang dilakukan oleh masing-masing anggota.



3.2. Daftar file

a. rank.py

File yang berfungsi untuk memproses data yang berupa masukan dokumen dan query dan menentukan kata-kata unik data tersebut dan jumlahnya serta menentukan tingkat kemiripan setiap dokumen dengan query menggunakan cosine similarity.

b. routes.py

File yang berperan dalam memberikan fungsionalitas pada html yaitu menerima inputan data seperti dokumen dan query yang akan diproses menggunakan fungsi dari file python lainnya yang ada pada program untuk kemudian ditampilkan dokumen-dokumen tersebut secara berurut dan matriks yang berisi jumlah kemunculan kata-kata unik query pada dokumen analisis dan menentukan alur kerja program dan menghubungkan link-link yang ada pada html.

c. driverGetData.py

Merupakan file yang berfungsi sebagai file uji dari URLfromGoogle,py, saveDatatoSql.py, dan readdataUpload.py.

d. getURLfromGoogle.py

Terdapat fungsi UrlFromGoogle(Input) yang mencari alamat web yang sesuai dengan query (Input) yang mengembalikan hasil dalam bentuk list dan UrlInGoogle(List) yang menerima masukan list url untuk mendapatkan alamat url yang siap untuk diolah ke tahap selanjutnya dalam bentuk list.

e. makeFinalSql.py

Terdapat 4 modular sebagai berikut:

i. SaveFinalSql() merupakan prosedur yang bertujuan menggabungkan file uploads dengan data yang diperoleh dari google untuk menjadi database yang siap untuk diolah ke tahap penghitungan kesamaan dengan query.

ii. ReturnListSastra() adalah fungsi yang mengembalikan kalimat dalam Sastrawi bahasa Indonesia dalam bentuk list dari seluruh kalimat yang ada di database.

iii. SavePersen(list) merupakan prosedur yang menerima masukan list dari hasil perhitungan kesamaan dengan query yang ada kemudian hasil perhitungan dimasukkan dalam finalDataBase.sqlite kolom persen.

iv. ReturnRank() adalah fungsi yang mengembalikan URL atau file ".html" dari finalDataBase.sqlite dalam bentuk list yang sudah terurut mengecil berdasarkan nilai persentase kesamaan dengan query.

f. makeSastra.py

Terdapat fungsi Makesastra(input) yang menerima masukan kalimat dan mengembalikan dalam kalimat Sastrawi bahasa Indonesia.

g. saveDatatoSql.py

Terdapat prosedur urltoSQL(List) yang menerima masukan list of url yang didapatkan dari UrlInGoogle(List) yang terdapat di file getURLfromGoogle.py dimana berfungi mengolah semua data yang terdapat dalam html agar bisa diproses ke tahap selanjutnya dan disimpan dalam fileBaseGoogle.sqlite

h. index.html

File HTML utama yang berfungsi untuk menerima masukan query dari pengguna dan menampilkan dokumen-dokumen yang ada pada *database* , jumlah kata ,dan sejumlah

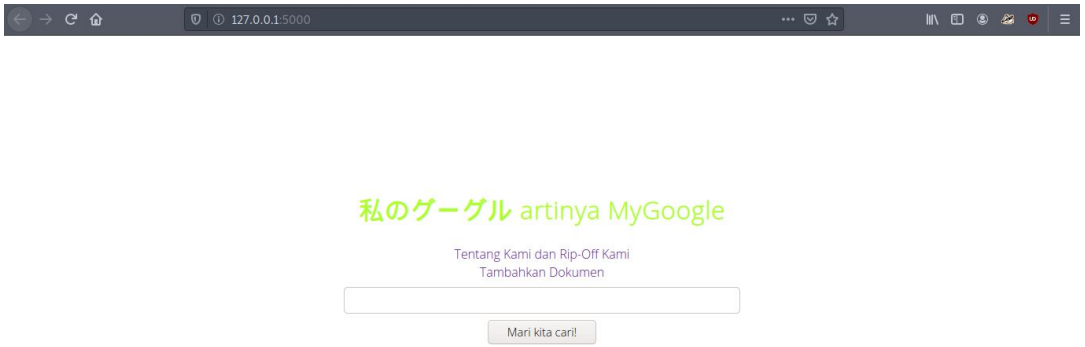
kalimat pada dokumen tersebut secara berurut berdasarkan tingkat kemiripannya dengan query dan menampilkan sebuah *data frame* berbentuk matriks yang berisikan kata-kata unik yang ada pada query dan jumlah kata-kata unik tersebut pada dokumen yang ada pada *database*. Pada file ini juga terdapat link menuju perihal.html dan adddata.html

i. perihal.html
   File yang menampilkan perihal pada laman web (konsep singkat search engine, How to Use, dan About Us) .
j. adddata.html
   File yang menampilkan pilihan penambahan dokumen ke dalam database.
k. upload.html
   File yang menyediakan tempat untuk meng-upload file dokumen yang akan dimasukkan ke dalam database.
l. googling.html
   File yang menyediakan tempat untuk mengambil dokumen dari Google berdasarkan query yang di-input.

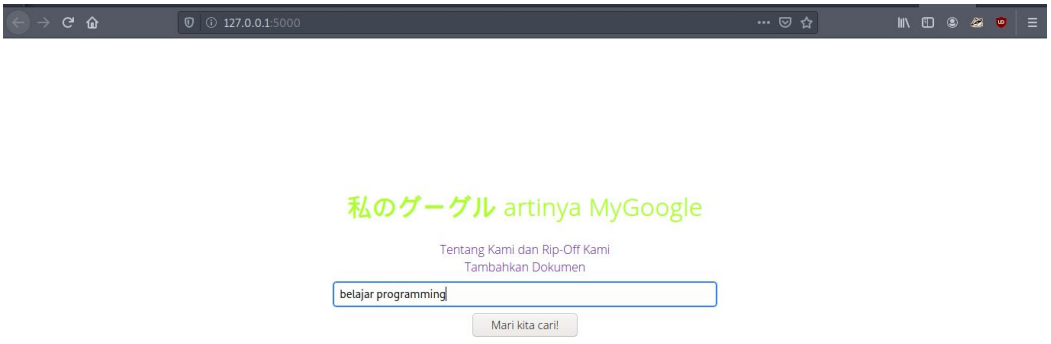
BAB IV EKSPERIMEN

4.1. Eksperimen

Setelah mengikuti langkah-langkah yang diperlukan untuk melakukan instalasi program sesuai yang tertera pada Github, program dijalankan dengan mengetikan perintah **flask run** pada command line**.** Setelah mengetikan perintah tersebut akan muncul sebuah alamat web yang jika dimasukan ke dalam web browser, akan memunculkan tampilan web seperti gambar yang di bawah ini.

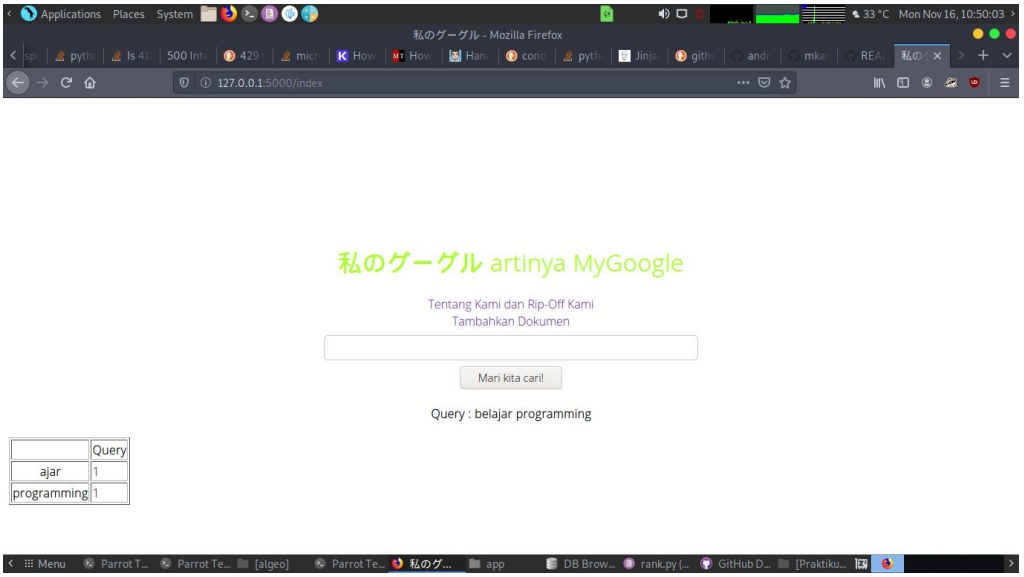


Dapat dilihat bahwa pada halaman tersebut terdapat link menuju halaman perihal dan halaman untuk menambahkan dokumen untuk dianalisis dan sebuah form untuk menerima masukan query dari pengguna dan sebuah submit button. Pada eksperimen ini, pada form akan dimasukan kalimat berupa “belajar programming” seperti di bawah ini.

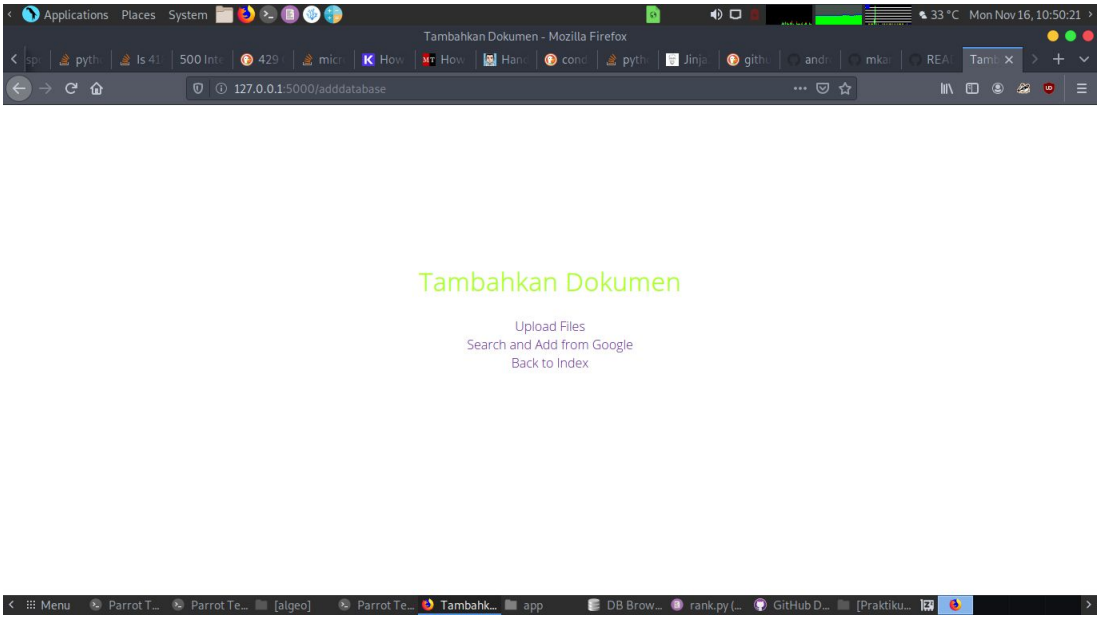


Hasilnya dapat dilihat seperti gambar di bawah ini. Dikarenakan tidak ada file dokumen yang diupload ke dalam database, maka tidak ada dokumen yang akan dianalisis sehingga hanya muncul tulisan yang muncul pada masuk query dari pengguna dan matriks yang berisi jumlah

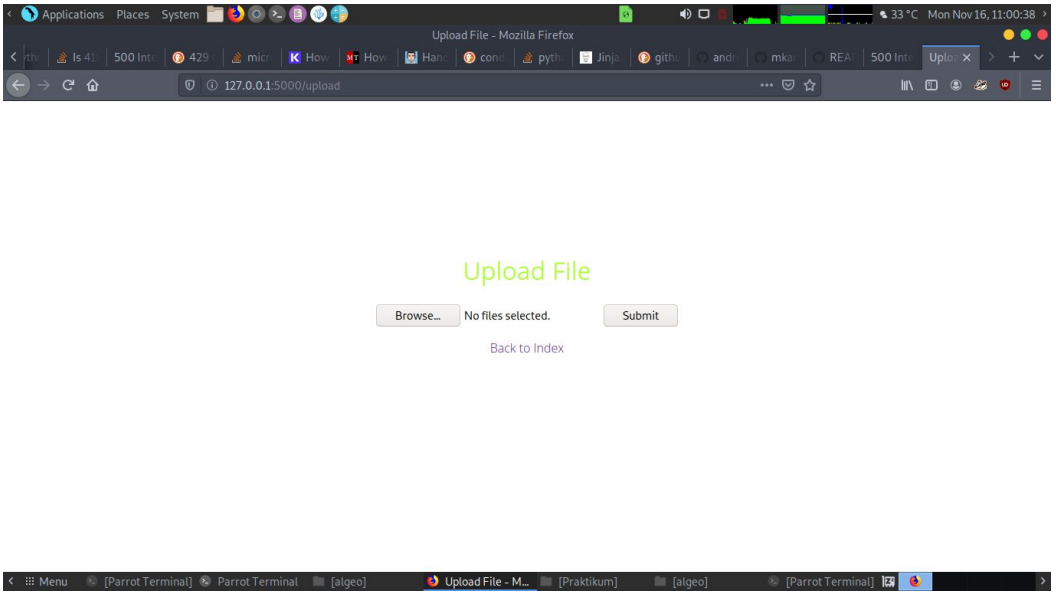
kemunculan kata-kata unik pada query. Perhatikan bahwa kata-kata unik pada pada matriks sudah dilakukan proses *stemming* dan *lemmatization* yaitu merubah kata-kata yang ada pada query menjadi kata dasarnya.



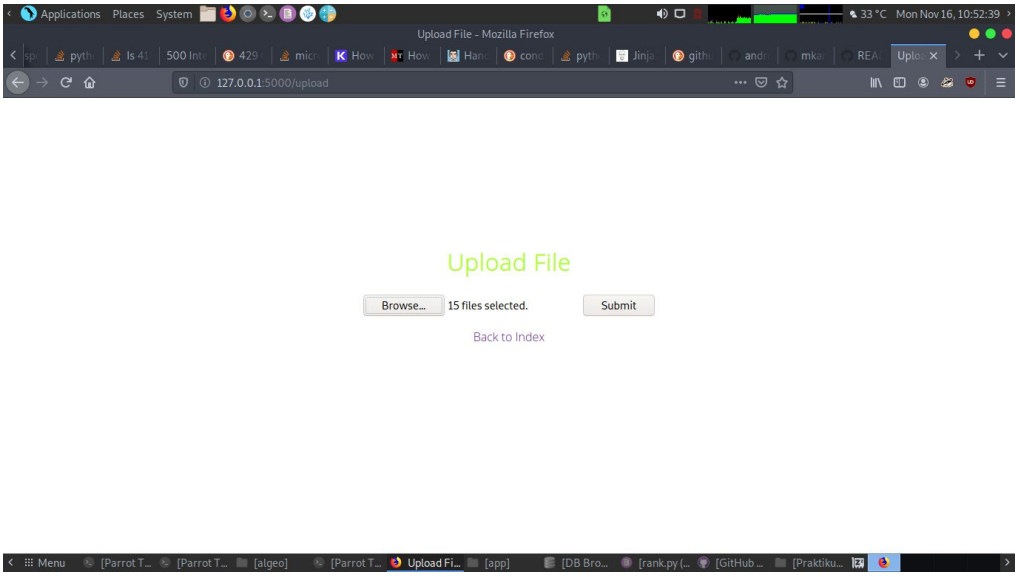
Dengan mengklik link yang ada pada tulisan "Tambahkan Dokumen", maka pengguna akan masuk pada halaman berikut yang berisi link menuju upload files, search and add from google, dan back to index yang mana akan membawa pengguna ke halaman awal.



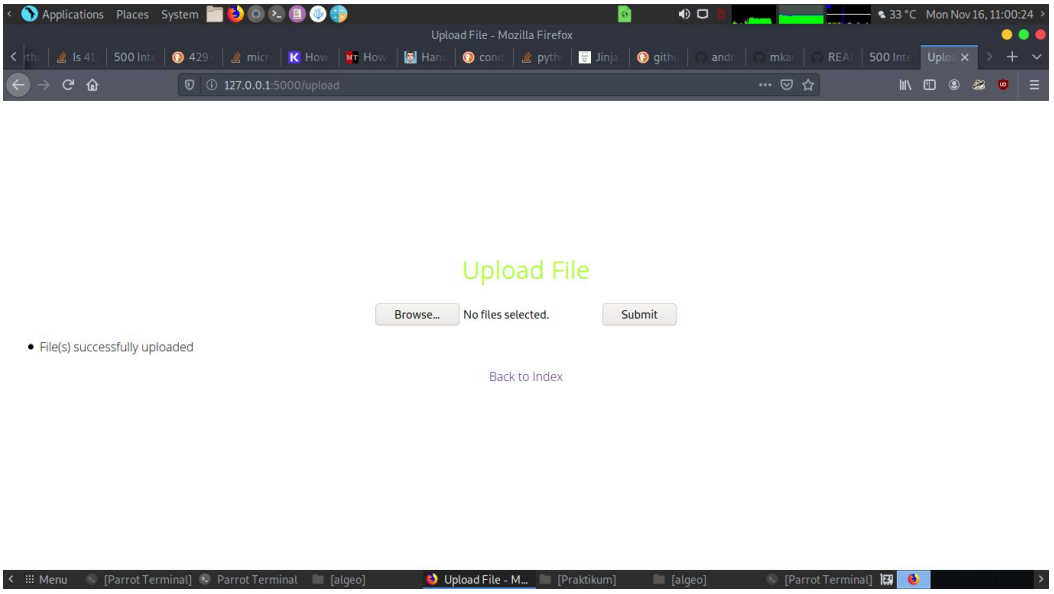
Pada halaman upload files, terdapat sebuah form di mana pengguna dapat mengupload dokumen yang akan dianalisis dan submit buttonnya. Juga terdapat link "back to index" yang jika diklik akan membawa pengguna ke halaman awal.



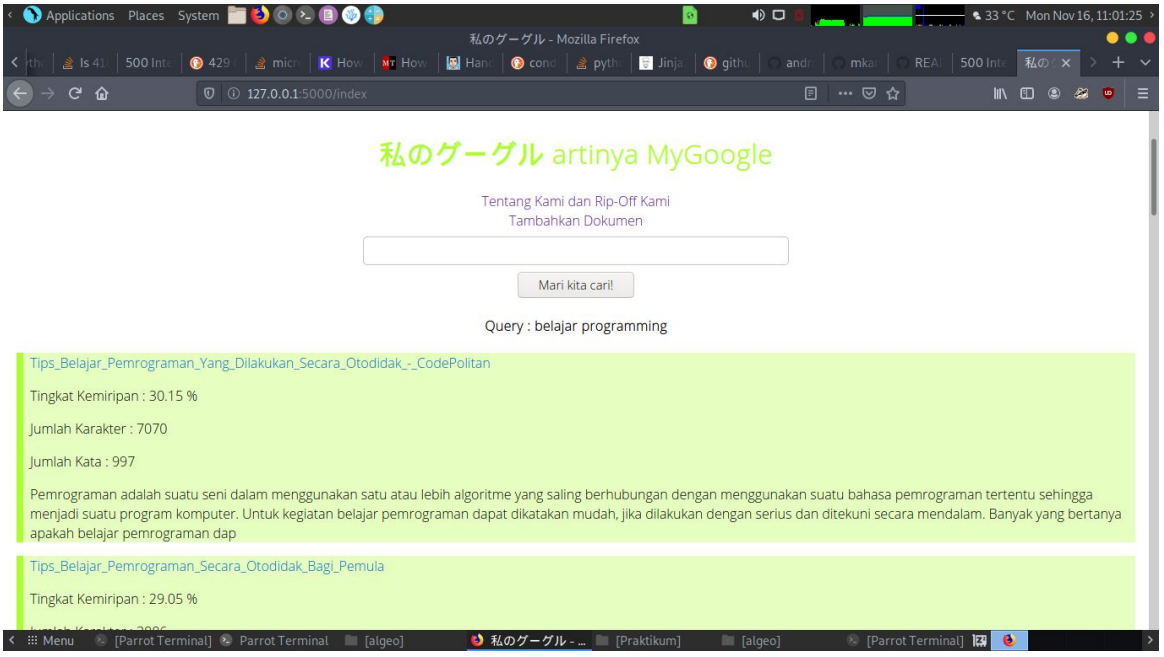
Pada eksperimen kali ini akan dimasukan 15 dokumen.



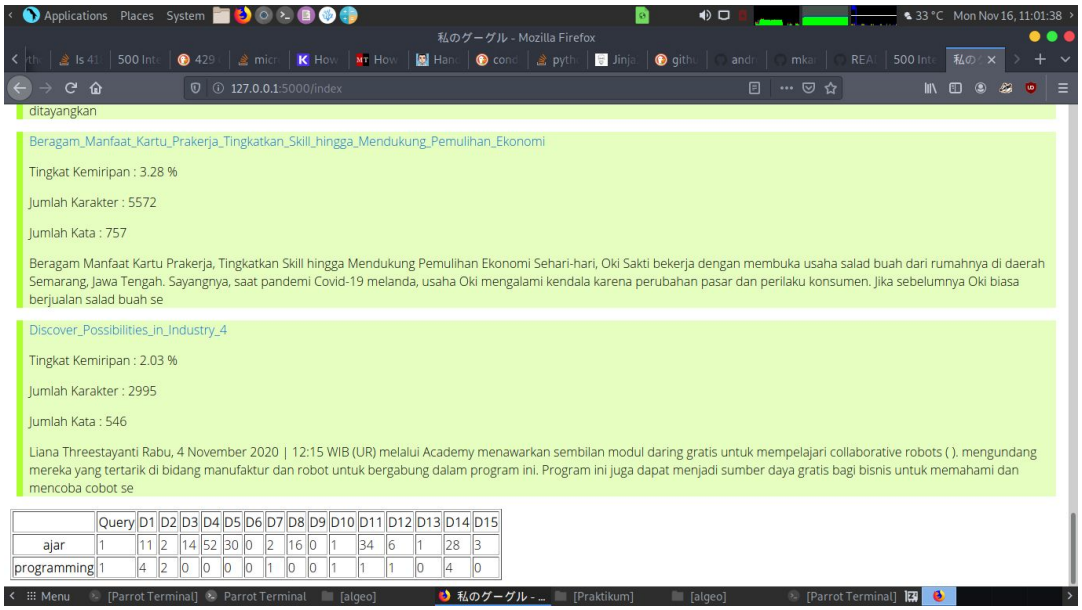
Setelah menunggu beberapa saat untuk dokumen diproses ke dalam database, akan muncul pesan seperti berikut.



Setelah itu ketika kita kembali lagi ke halaman index dan mengetikan kembali “belajar programming” pada form query, akan muncul tampilan seperti berikut yaitu daftar link judul dokumen serta tingkat kemiripannya dengan masukan query, jumlah karakter dan katanya, dan sebagian dari kalimatnya yang ditampilkan secara terurut dari yang tingkat kemiripan terbesar menuju terkecil. Perhatikan bahwa pada program ini, dokumen yang memiliki tingkat kemiripan 0% tidak ditampilkan pada web.

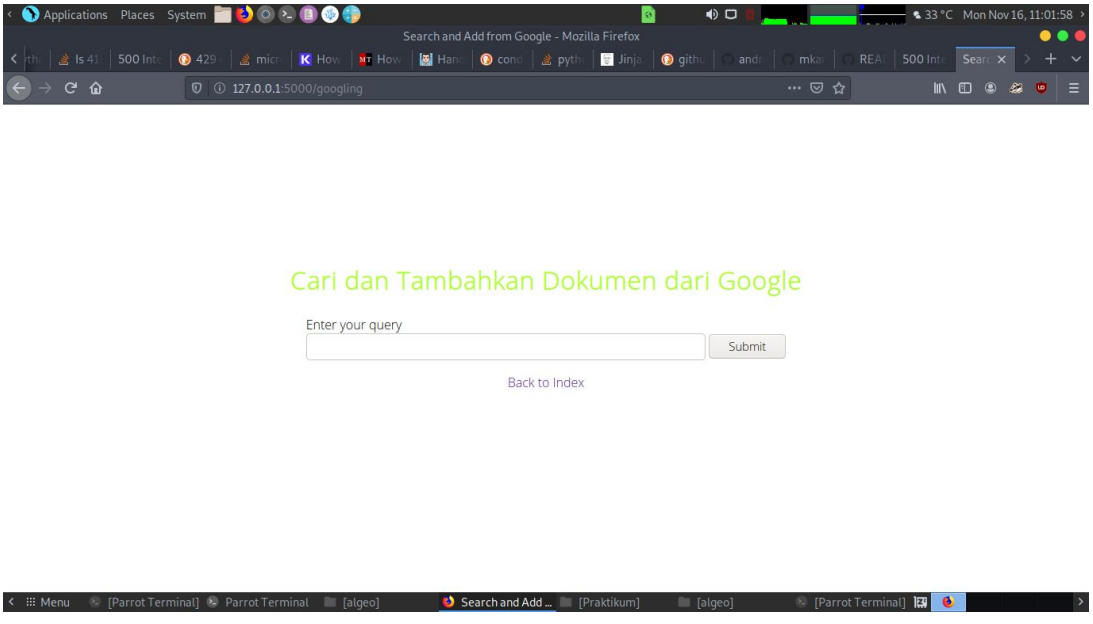


Dan pada bagian bawah index, terdapat dataframe matriks yang berisi kalimat unik query yang sudah diproses serta jumlah kemunculannya pada dokumen-dokumennya.

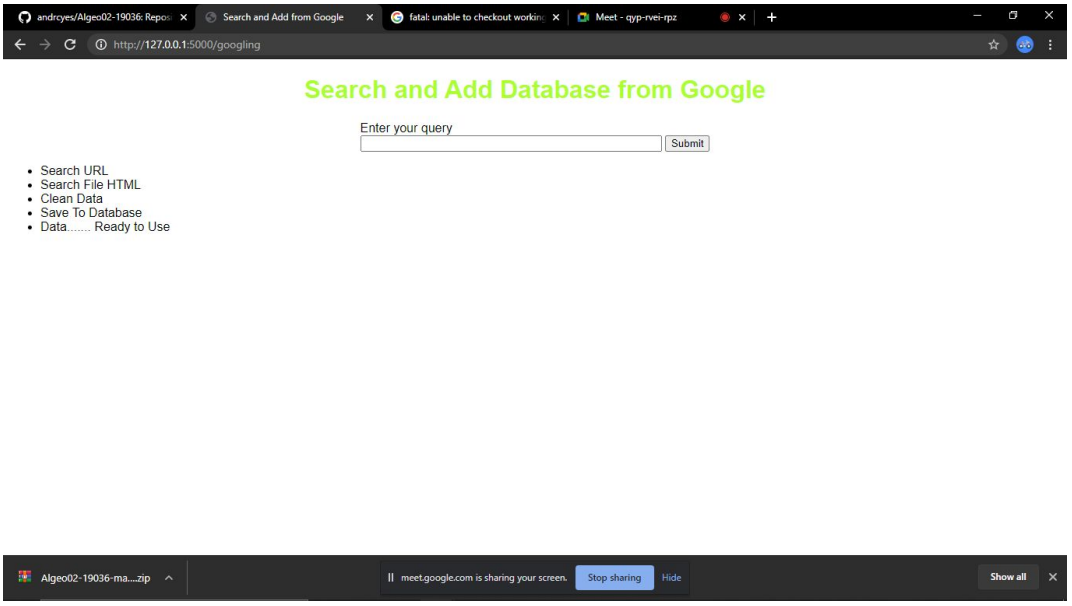


	Query	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
ajar	1	11	2	14	52	30	0	2	16	0	1	34	6	1	28	3
programming	1	4	2	0	0	0	0	1	0	0	1	1	1	0	4	0

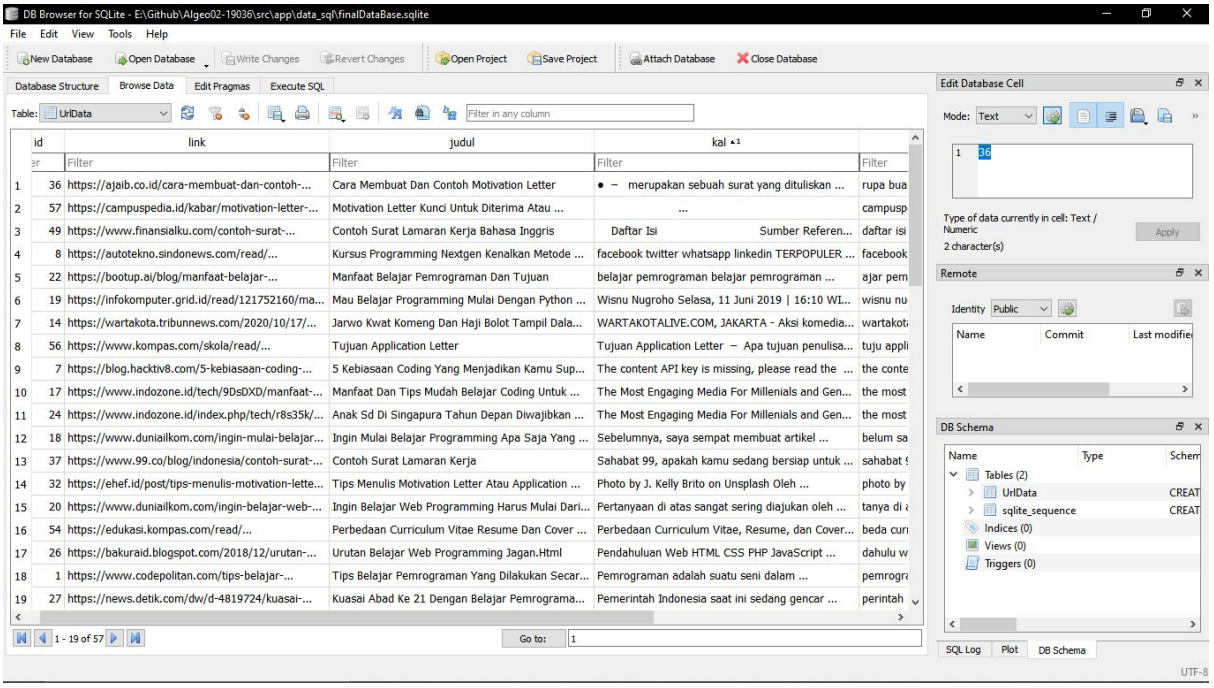
Dibawah ini merupakan tampilan halaman yang menerima masukan sebuah query dimana akan mencari dokumen yang sesuai dengan query di google serta mengolah data sampai siap diolah dan menjadi data *final* dari databese yang ada.



Dibawah ini merupakan tampilan notifikasi setiap proses yang terjadi dan menjadi acuan jika jika data didapatkan dan siap diolah. ‘Data ….. Ready To Use’ adalah notifikasi akhir bahwa data siap digunakan, jika tidak ada notifikasi dapat dipastikan proses error dalam notifikasi terakhir.



Ini merupakan gambaran bagaimana data yang diupload ataupun data dari google diolah dan dimasukkan dalam database yang berformat sqlite. Data yang disimpan adalah id, link, judul, kalimat, sastra(kalimat dasar bahasa indonesia), jumlah karakter, jumlah kata dan persen yang menyatakan kemiripan dengan input query.



BAB V KESIMPULAN, SARAN, DAN REFLEKSI

5.1. Kesimpulan

Program dapat berjalan dengan lancar. Dengan bantuan seperti library pandas, sqlite, dan lainnya, program ini dapat memproses masukan data dari pengguna lalu menyimpannya dalam database dan menampilkannya hasil analisisnya dalam bentuk data frame.Selain dari dokumen yang dimasukan oleh pengguna, program ini juga dapat mengambil data dari internet dengan menggunakan metode web scraping.

5.2. Saran

Untuk program, mungkin dapat dikembangkan lebih baik terutama dalam hal mempercepat analisis data dan pengambilan data melalui metode web scraping. Selain itu, program juga dapat ditingkatkan akurasi dengan menggunakan algoritma lainnya selain bag of words(algoritma program ini) seperti TF-IDF(Term Frequency–Inverse Document Frequency) dan lain-lainnya.

5.3. Refleksi

Tugas ini sangat bermanfaat bagi kami dikarenakan kami dapat diperkenalkan dengan materi Web Development yang di dalamnya kami dapat mengeksplor banyak hal terkait frontend dan backend. Sayangnya dengan kemampuan dan jumlah waktu yang kami perlukan untuk mengerjakan proyek ini, kami tidak bisa mencoba framework lainnya seperti bootstrap dan react.Selain itu kami pun mendapat sedikit kesulitan yang dikarenakan pengerjaan proyek ini tidak dilakukan pada operating system yang sama sehingga terdapat masalah pada pembuatan dan penggunaan virtual environment. Diharapkan pengalaman yang kami dapatkan dari proyek ini dapat membantu kami untuk meningkatkan koordinasi dari kerja sama kami dan efisiensi dari kerja tim serta membantu memberikan sebuah insight lebih mengenai dunia web development.

DAFTAR PUSTAKA

Kuliah Sambil Kerja, BINUS ONLINE Learning solusinya. (2020). Metode-Metode Information Retrieval. [online] diambil dari: https://onlinelearning.binus.ac.id/computer-science/post/metode-metode-information-retrieval/ .

“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 .

Exponea. (n.d.). Basic Syntax of Jinja. [online] diambil dari: https://docs.exponea.com/docs/jinja-syntax .

OverIQ.com. (2020). Form Handling in Flask. [online] diambil dari: https://overiq.com/flask-101/form-handling-in-flask/ .

Venkat, S. (2020). Implementing CountVectorizer from Scratch in Python Exclusive. [online] Medium. diambil dari: https://medium.com/@saivenkat_/implementing-countvectorizer-from-scratch-in-python-exclusive-d6d8063ace22 .