TUGAS BESAR 2
APLIKASI DOT PRODUCT PADA SISTEM TEMU-BALIK INFORMASI

IF 2123
ALJABAR LINIER DAN GEOMETRI
SEMESTER 1

Oleh

Kelompok 25 “gugel”

Dwianditya Hanif Raharjanto	**13519046**
Nizamixavier Rafif Lutvie	**13519085**
Muhammad Fawwaz Naabigh	**13519206**



PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2020/2021

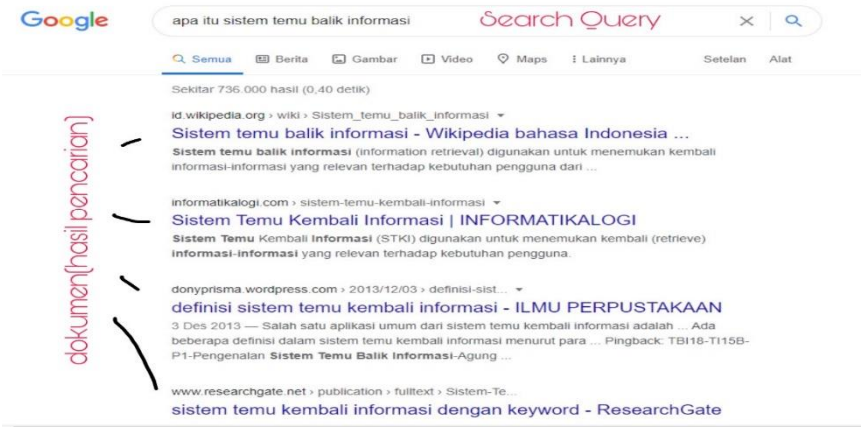
DAFTAR ISI

BAB 1
DESKRIPSI MASALAH

1.1 Abstraksi

Di zaman yang modern seperti sekarang sangatlah mudah untuk mengakses sesuatu hal seperti informasi dan lain sebagainya. Kita cukup memasukkan suatu kata kunci pada *search engine* kemudian kita akan mendapatkan apa yang kita mau. *Search engine* itu sendiri adalah suatu program yang difungsikan untuk membantu pengguna dalam mencari sesuatu yang tersimpan dalam suatu *database*. Contohnya seperti *google, yahoo! search, dan bing.* Namun, pernahkah terpikirkan bagaimana cara kerja *search engine* tersebut ?

Sesuai dengan apa yang kami pelajari dalam kuliah aljabar linier dan geometri pada materi vektor di ruang Euclidean, *search engine* ini dapat diaplikasikan dengan proses temu-balik informasi (*Information Retrieval)* dengan model ruang vektor yang merupakan suatu proses untuk menemukan kembali informasi yang relevan terhadap pengguna dari suatu kumpulan informasi secara otomatis. Umumnya digunakan untuk pencarian informasi yang isinya tidak terstruktur seperti dokumen (isinya bergantung dari pembuatnya) dan laman web.



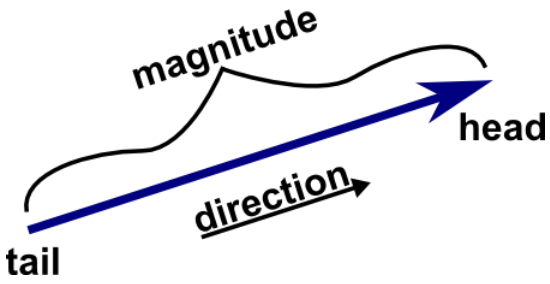
Secara singkat prinsip kerja temu-balik informasi dengan model ruang vektor ini mengubah *search query*, dokumen yang ada pada database menjadi sebuah vektor $w=(w_1,w_2,...,w_n)$ di dalam ruang R^n, dimana w_1 menyatakan jumlah kemunculan kata tersebut dalam dokumen (*term frequency).* Penentuan dokumen mana yang paling relevan dengan *search query* dapat ditemukan dengan mengukur kesamaan (*similarity measure)* antara *query* dan dokumen. Semakin tinggi kesamaannya, semakin relevan dokumen tersebut dengan *query*. Cara untuk mengukur kesamaan tersebut dengan menggunakan *cosine similarity.*

Pada kesempatan ini kami akan membuat program *search engine* sederhana menggunakan prinsip temu-balik informasi dengan model ruang vektor dan menggunakan *cosine similarity* untuk menghitung relevansi dokumen dengan *search query* yang ada.

BAB 2
TEORI SINGKAT

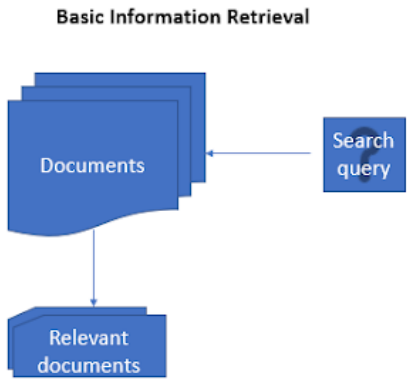
2.1 Vektor

Vektor adalah sebuah objek yang memiliki *magnitude* atau besaran dan arah. Secara geometri dapat digambarkan dengan garis panah yang mana panjang dari panah itu adalah *magnitude*-nya dan arah dari vektor itu adalah dari bagian belakang vektor menuju bagian depan (*from tail to head)*.



2.2 Retrieval Information (Temu-Balik Informasi)

Temu-balik informasi (*Retrieval Information)* merupakan suatu sistem yang bertujuan untuk menemukan kembali informasi yang relevan terhadap pengguna dari suatu kumpulan informasi secara otomatis. Umumnya sistem ini digunakan untuk pencarian informasi yang tidak terstruktur seperti dokumen (bergantung dari pembuatnya) dan laman web.



Aplikasi dari system temu-balik informasi ini adalah pembuatan *search engine*. Model temu-balik informasi yang kami gunakan adalah model ruang vektor yang berprinsip pada teori aljabar vektor.

Penjelasan lebih lanjutnya adalah misalkan ada n kata berbeda sebagai "*database"* atau indeks kata (*term index)*. Kata-kata yang menjadi indeks kata ini membentuk ruang vektor berdimensi n. Kemudian setiap dokumen maupun *query* dinyatakan pula sebagai vektor w = (w_1,w_2,..,w_n) dalam R^n untuk w_i adalah bobot setiap kata *i* di dalam *query* atau dokumen.

Contohnya adalah misalkan ada tiga buah kata (***T*₁,*T*₂,** dan ***T*₃**), dua dokumen (***D*₁**, ***D*₂**), dan *query* ***Q***. Dalam bentuk vektor nya bisa kita tuliskan sebagai berikut :

$$\boldsymbol{D_1} = (2,3,5);\ \boldsymbol{D_2}=(3,7,1);\ \boldsymbol{Q}=(0,0,2)$$

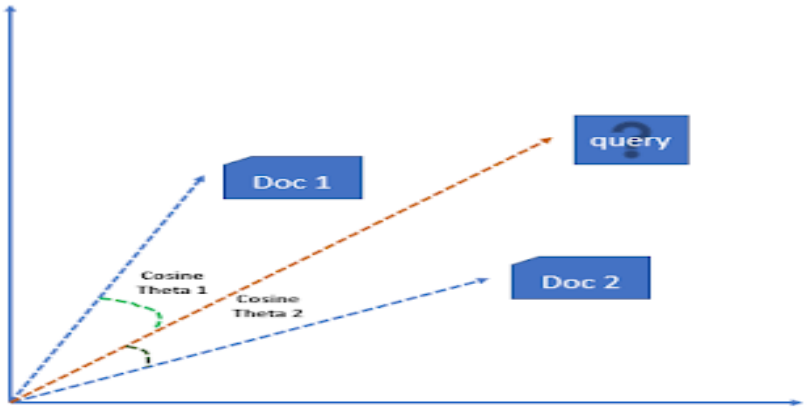
D_1 = (2,3,5) memiliki arti bahwa ***D*₁** mengandung dua buah kata ***T*₁** , tiga buah kata ***T*₂**, lima buah kata ***T*₃**. Misalkan ***T*₁** = Menteri, ***T*₂** = minta, ***T*₃** = Korupsi.
Misalkan isi dari D_1 adalah *"**Menteri** olahraga me**minta** maaf atas perbuatan **korupsi**. **Menteri** tersebut terlibat **korupsi.** Me**minta-minta** komisi termasuk **korupsi**. **Korupsi** sudah menjadi budaya. **Korupsi** sudah mendarah daging."*.

2.3 Cosine Similarity

Cosine Similarity adalah suatu cara untuk mengukur relevansi antara *search query* dengan dokumen. Semakin tinggi hasil *cosine similarity* maka semakin relevan pula dokumen dengan *query*.

Prinsip penghitungan *cosine similarity* sebenarnya adalah perkalian dot dua buah vektor. Namun, yang kita cari bukan hasil dari perkalian dot dua buah vektor, tetapi hasil dari cos α nya.
Contoh ada 2 buah vektor ***Q*** = (q_1,q_2,..,q_n) dan ***D*** = (d_1,d_2,…,d_n) diukur dengan rumus *cosine similarity* atau perkalian titik dua buah vektor :

$$\mathbf{Q \bullet D = ||Q||\ ||D||\ \cos\alpha \rightarrow sim(Q,D) = \cos\alpha = \frac{Q.D}{||Q||\,||D||}}$$



Setelah menghitung *similarity* setiap dokumen dengan *query*, hasil perhitungan kita urutkan dari besar ke kecil. Urutan pertama lah yang paling relevan antara dokumen dengan *query* dan urutan terakhir relevansi antara dokumen dengan *query* lebih sedikit dari urutan-urutan sebelumnya.

BAB 3
IMPLEMENTASI PROGRAM

3.1 Struktur Program

Pada program yang kami buat, kami menggunakan bahasa pemrograman *python* lebih tepatnya menggunakan *framework Flask* untuk *backend* dan html untuk *frontend*. Kami mendefinisikan beberapa fungsi penunjang dan *method* untuk program kami. Berikut akan kami jelaskan mengenai masing-masing fungssi dan *method* yang kami buat.
Berikut merupakan fungsi-fungsi yang ada pada *framework Flask* :

a) dotProduct(v1,v2)
   Sebuah fungsi yang nantinya akan memproses untuk menghasilkan hasil perhitungan perkalian titik dua buah vektor v1 dan v2.
b) normaVektor(v)
   Sebuah fungsi yang nantinya akan memproses untuk menghasilkan hasil perhitungan norma suatu vektor v.
c) removePunctuation(sentence)
   Sebuah fungsi yang bertujuan untuk menghilangkan pungtuasi pada suatu kalimat sentence.
d) removeStopwords(sentence)
   Sebuah fungsi yang bertujuan untuk menghilangkan *stopwords* pada suatu kalimat sentence. Arti dari *stopwords* itu sendiri adalah kata umum yang biasanya tidak memiliki makna.
e) stemSentence(sentence)
   Sebuah fungsi yang bertujuan untuk menguraikan suatu kata sentence menjadi bentuk kata dasarnya.
f) main()
   Sebuah fungsi yang bertujuan untuk memproses inputan *query*. Jika tidak menginput apa-apa akan tetap pada laman yang sama, tetapi jika sudah menginput sesuatu akan mendapatkan hasil pencarian dari inputan tersebut.
g) upload()
   Fungsi upload ini berfungsi agar kami dapat mengupload dokumen-dokumen yang nantinya akan menjadi referensi dokumen untuk pengguna setelah pengguna memasukkan *query*.
h) result()
   Fungsi result ini berfungsi untuk menghitung dan menampilkan hasil masukan *query* pengguna berupa judul dokumen, jumlah kata dalam dokumen, tingkat kemiripan dokumen dengan *query* dan kalimat pertama pada dokumen.
i) about()
   Fungsi about ini berfungsi untuk me-*render* laman web “about” kami.

Kemudian beranjak pada html. Pada html ini kita memiliki dua *method* yaitu “POST” dan “GET”. *Method* “POST” sesuai dengan artinya yaitu memberikan sesuatu, jadi *method* ini berfungsi untuk seluruh kegiatan yang berhubungan dengan memberikan sesuatu pada program, contoh mengupload file, menginput *query*.
Method “GET” adalah kebalikan dari post, yaitu kita yang mendapatkan sesuatu dari program. Contoh seperti hasil pencarian dari inputan *query* kita.

Selain *method* kami juga ada beberapa fungsi pada html, yaitu sebagai berikut :

a) about.html
   Program about.html ini memiliki fungsi untuk mengolah laman web bagian “about”. Yaitu laman web yang berisi tentang konsep singkat *search engine* kami, bagaimana cara menggunakannya, dan tentang kami.
b) file.html
   Program file.html ini memiliki fungsi untuk mengolah laman web “upload file”. Yaitu laman web yang berisi tombol untuk meng-*upload* dokumen.
c) result.html
   Program result.html ini memiliki fungsi untuk mengolah laman web utama kami yaitu laman web untuk mencari suatu dokumen dengan inputan kata kunci tertentu. Pada laman ini kami juga sudah menampilkan hasil dari pencarian.
d) index.html
   Program result.html ini memiliki fungsi untuk mengolah laman web utama kami yaitu laman web untuk mencari suatu dokumen. Namun, pada laman web kami belum ada inputan yang masuk dan belum ada hasil dari pencariannya.

Dari setiap kode html kami kurang lebih strukturnya sama, ada *body, sidebar, page content* secara garis besar dan yang membedakan hanya pada *page content*-nya. Pada bagian *sidebar* ada dua fungsi yang bernama “*w3_open()*” dan “*w3_close()*” yang memiliki bertujuan agar *sidebar* kami dapat muncul dan menghilang saat kita klik.

Kemudian dalam kasus bonus yaitu membuat *web scrapping* kami menggunakan library *bs4* dan meng-*import BeautifulSoup*. Kami mendefinisikan fungsi *removePunctuation* untuk menghilangkan pungtuasi pada judul artikel, karena judul artikel akan dijadikan nama suatu file dan nama suatu file tidak boleh ada sebuah pungtuasi. Kemudian ada variabel url yang berfungsi untuk menyimpan url web yang ingin kita proses, kemudian variabel soup yang berfungsi untuk mendapatkan konten dari html. Yang terakhir ada variabel title yang berfungsi untuk memproses judul, dan variabel paragraphs untuk memproses paragraf. Kemudian setelah selesai memproses suatu url, kode kami akan memproses untuk menyimpan hasil proses tersebut kedalam folder statis.

3.2 Garis Besar Program

Pada bagian ini kami akan menjelaskan bagaimana garis besar program kami berjalan atau bisa dikatakan alur kerja dari program kami.

Pertama kita meng-*impor't library* pendukung dan *framework* untuk kode kami. Disini kami menggunakan *framework* “Flask” dan *library* “nltk” dan “pandas”. Kemudian mendefinisikan variabel awal untuk mempermudah kami dalam menulis kode. Kemudian masuk pada fungsi-fungsi yang sudah dituliskan pada struktur program sesuai dengan tujuannya masing-masing.

Mungkin alur jalannya program yang ingin kami *highlight* mulai dari pendefinisian variabel “dataList”. Variabel tersebut bertujuan untuk menampung nama dokumen, jumlah kata, kemiripan, lokasi, baris pertama dokumen yang nantinya akan menjadi hasil dari pencarian.

Kemudian ada “df_query” yang bertujuan untuk menampung tiap *terms* pada *query*. Dan “doclist” sebuah list yang menampung seluruh dokumen yang ada.

Sebelum beranjak ke alur kerja fungsi “main()” kami ingin menjelaskan apa itu *@app.route*. Jadi *@app.route* itu akan selalu diikuti fungsi khusus seperti fungsi “main()” “result()” dan yang lainnya yang berhubungan dengan tampilan web. Nah, kegunaannya adalah semisal di kode kita tulis @app.route(“/def_function”) dan kita memiliki domain web “gugel.com” maka cara mengakses “def_function” pada web kami adalah “gugel.com/def_function”.

Mari beranjak pada fungsi-fungsi utama pada program *search engine* sederhana. Pertama fungsi “main()”, ketika *method* yang ingin dilakukan adalah “POST” makan program akan memproses inputan dari *query* kemudian akan menampilkan hasil dari pencariannya tersebut. Selain hal di atas makan tidak akan terjadi apa-apa.

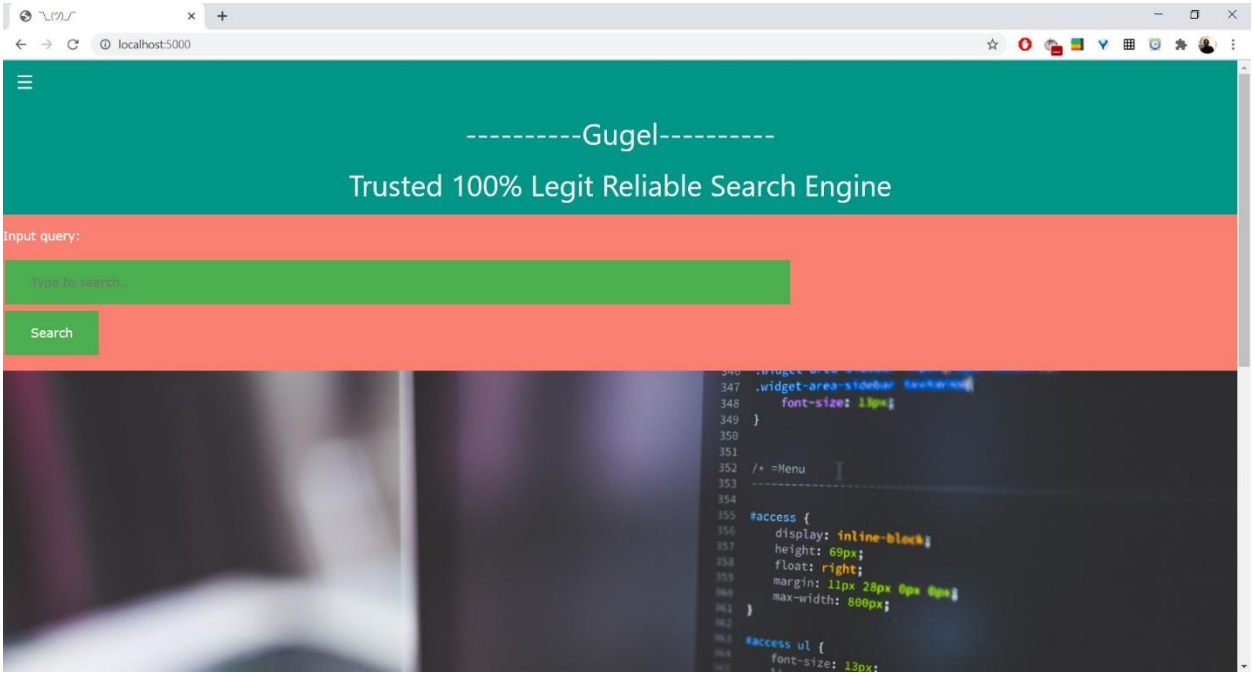
Kedua adalah fungsi “upload()”, ketika *method* yang ingin dilakukan adala “POST” program akan mengecek apakah *directory file* sudah ada atau belum, jika belum maka akan dibuat terlebih dahulu agar bisa melanjutkan proses berikutnya. Jika sudah ada *directory file* program akan mengecek apakah *file* yang diupload bernama atau tidak, jika bernama maka *file* tersebut akan berhasil diupload menuju “database” kami. Jika ada kasus nama *file* sudah ada dalam “database” maka akan secara otomatis tersimpan dengan format nama namafile_02.ekstensi. Jika tadi *file* yang ingin diupload tidak bernama, berarti tidak ada file yang ingin diupload sehingga ketika memencet tombol *submit* tidak terjadi apa-apa.

Ketiga adalah fungsi “result()”, jika dijelaskan secara detail akan memakan banyak halaman karena fungsi ini merupakan inti dari pemrosesan *search engine.* Jika bisa saya rangkum alur kerjanya adalah akan me-*reset* isi dari variabel “dataList”, “docList”, dan “df_query” supaya setiap penginputan *query* yang berbeda, hasil yang dihasilkan tetap relevan denga *query* tersebut, jadi hasil pencarian *query* sebelumnya tidak menumpuk. Kemudian memproses kalimat-kalimat yang ada pada *query* dan dokumen-dokumen yang ada menjadi *terms* yang bentuknya kata-kata yang bermakna dan kata dalam bentuk kata dasar. Setelah mendapatkan *terms*-nya kita ubah data-data yang sudah ada menjadi bentuk vektor agar bisa kita hitung kemiripannya. Setelah mendapatkan tingkat kemiripannya kita tampilkan kepada pengguna berupa nama dokumen, jumlah kata, tingkat kemiripan, dan kalimat pertama pada dokumen untuk setiap dokumen yang ada dengan urutan yang tingkat kemiripannya paling tinggi ditampilkan paling atas. Kemudian kita tampilkan juga tabel yang berisi *terms* dan kemuncul *terms* pada *query* dan dokumen-dokumen yang ada.

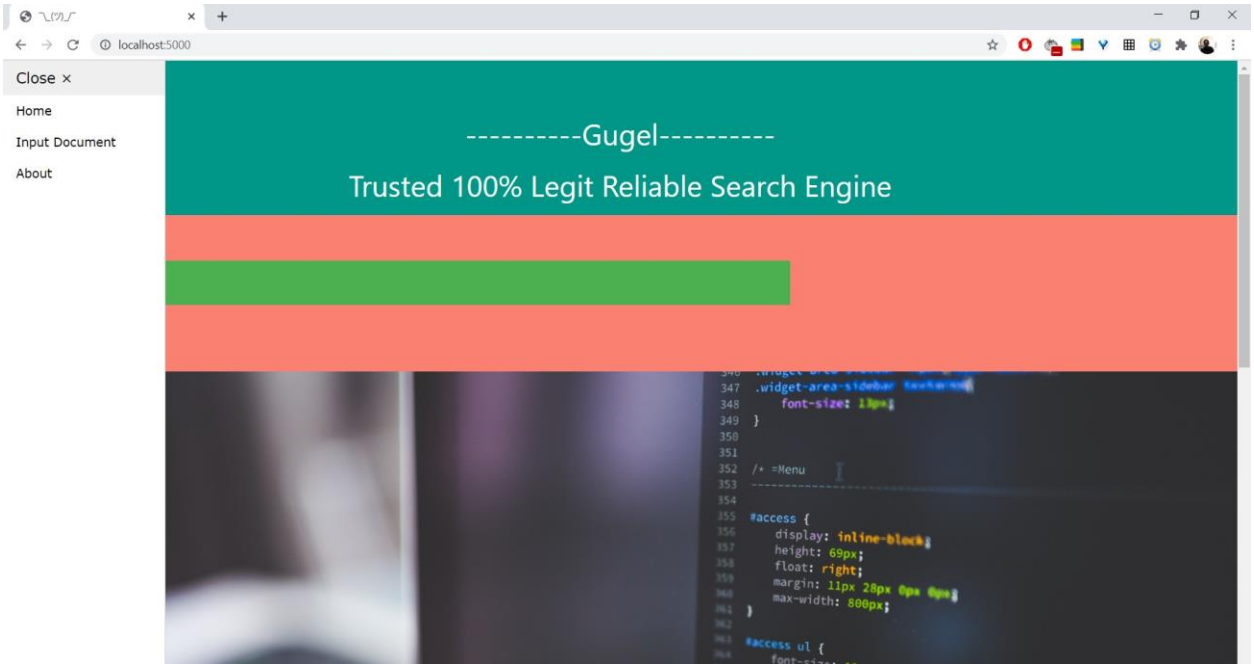
Terakhir adalah fungsi “about()”, fungsi ini sebenarnya tidak memiliki alur kerja karena dalam fungsi ini hanya bertugas untuk me-*redirect* pengguna ke laman “about us”.

BAB 4
HASIL EKSEKUSI PROGRAM

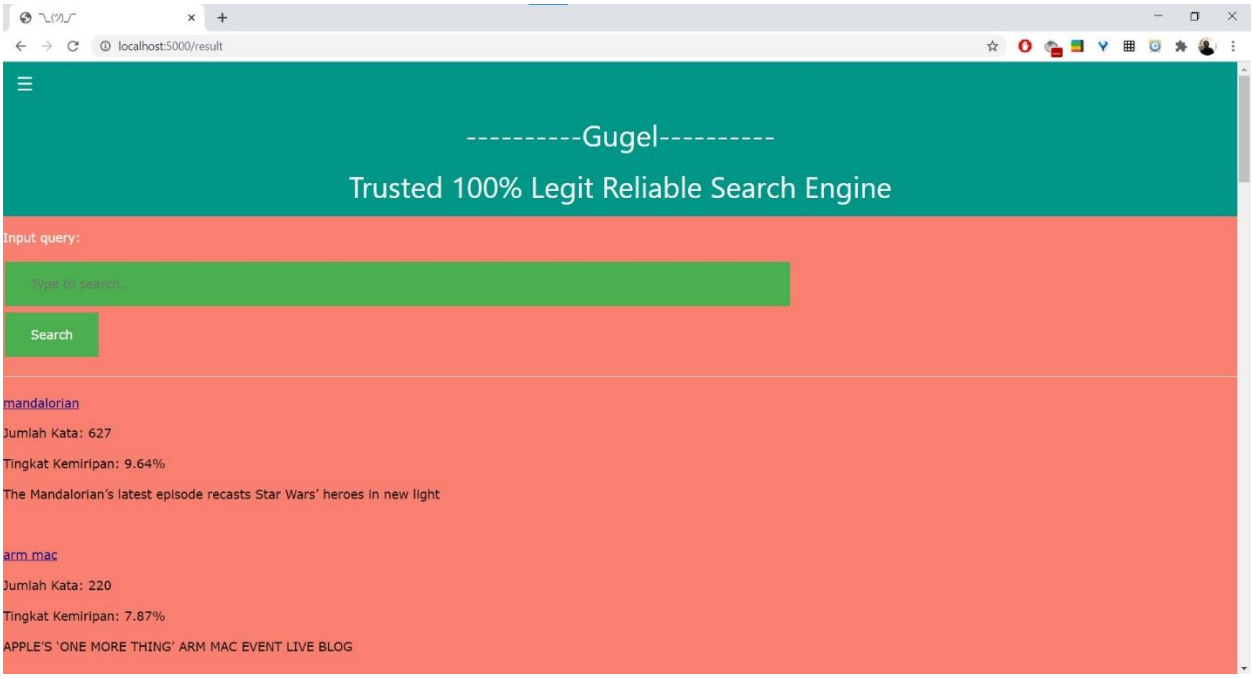
4.1 Screenshot Eksekusi Program



Gambar di atas merupakan penampakan halaman utama web kami. Untuk menjelajahi laman web lain dapat memencet tombol *sidebar* di pojok kiri atas. Berikut penampilan menu *sidebar* kami.

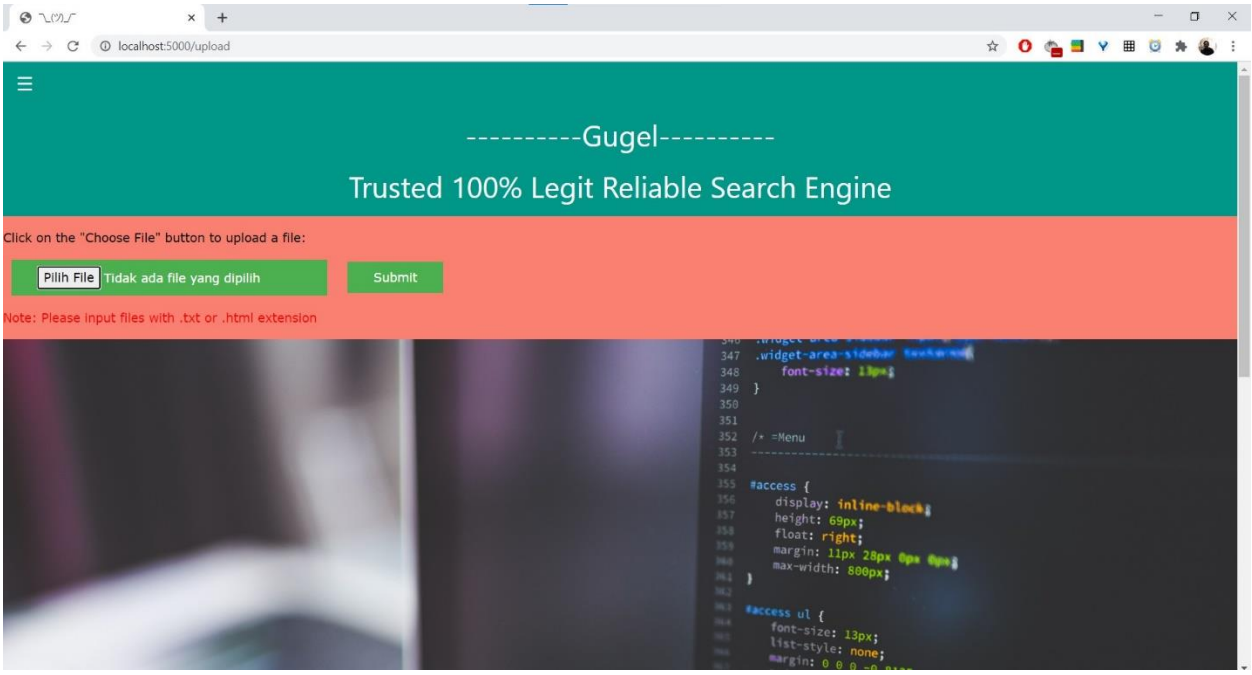


Gambar di atas merupakan tampilan *sidebar* kami. Ketika memencet *close* makan tampilan akan kembali seperti tampilan utama pada gambar pertama.

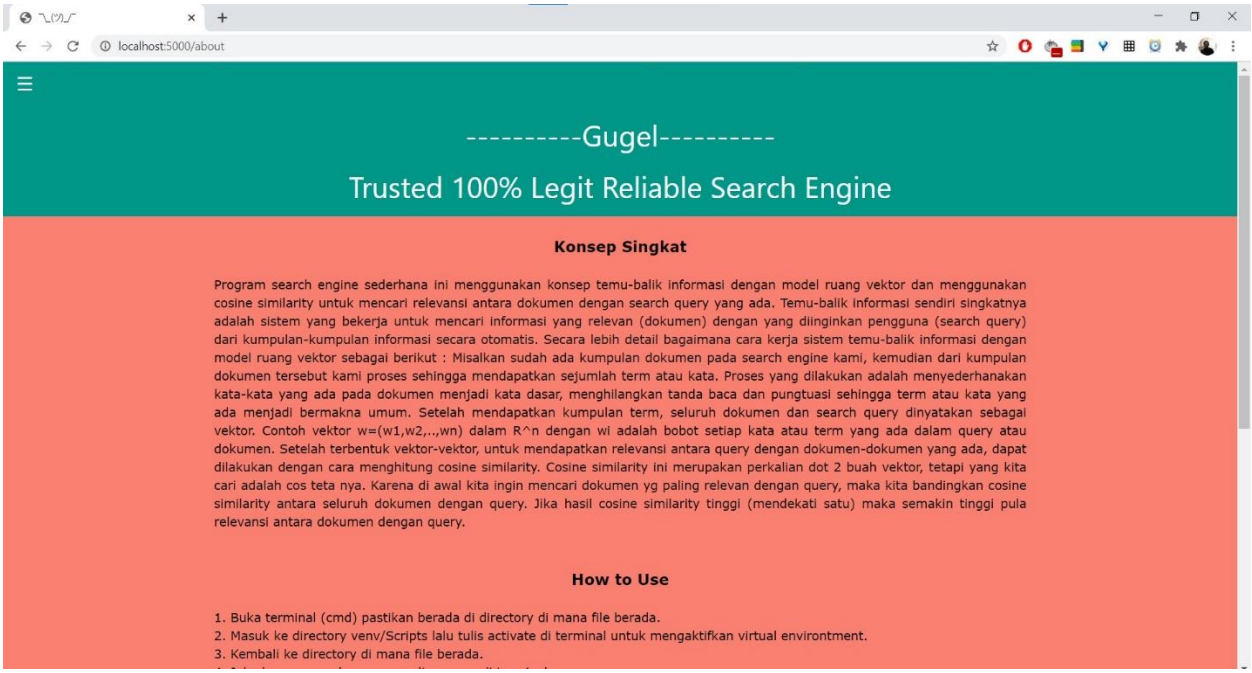


Term	Query	Amazon's Alexa	arm mac	BMW's	boring co	Facebook extends political ad ban another month as Trump refuses to concede - The Verge	Facebook	halo4	iHeartMedia	leica	mandalorian	Moscow's FR	Neural Net	nokias	Planet	PS5	psa macos is a little broken this morning with many nonapple apps hanging at launch	Zoom
new	1	5	3	3	1	4	0	0	2	1	7	0	1	3	1	2	2	0
phone	2	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0
old	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

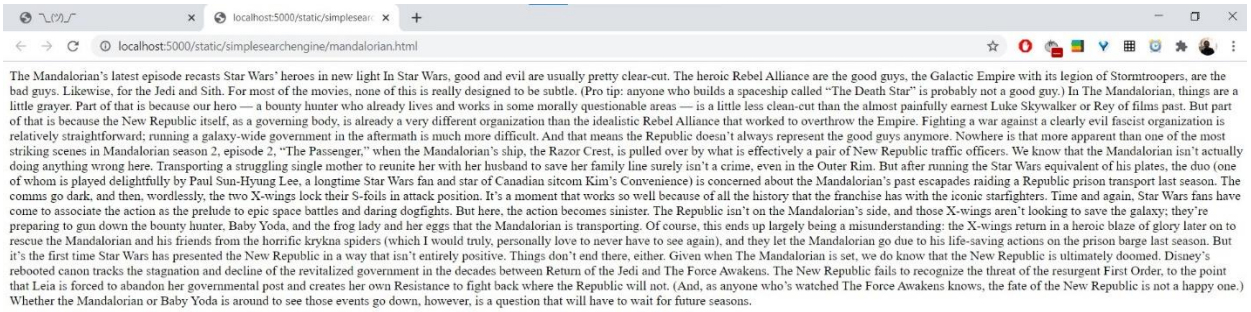
Dua gambar di atas merupakan tampilan ketika *search engine* kami menampilkan hasil pencarian terhadap suatu *query*.



Gambar di atas merupakan tampilan laman *upload* kami.



Gambar di atas merupakan tampilan laman *about* kami.



Gambar di atas merupakan tampilan jika kita memencet *hyperlink* dari hasil pencarian.

BAB 5
KESIMPULAN, SARAN, DAN REFLEKSI

5.1 Kesimpulan

Dengan melihat hasil yang telah kami capai, maka dapat kami tarik beberapa kesimpulan sebagai berikut :

1) Terdapat banyak *library* yang sudah tersedia untuk membantu dalam membuat program *search engine* sederhana ini.
2) Konsep vektor sebenarnya sangat bermanfaat untuk banyak hal seperti contohnya konsep vektor ternyata menjadi dasar untuk prinsip kerja sistem temu-balik informasi yang dapat diaplikasikan menjadi *search engine* sederhana.
3) Program yang kami buat sudah memenuhi spesifikasi yang diberikan.
4) *Natural Language Processing* (NLP) benar-benar memberikan banyak manfaat untuk manusia dalam kehidupan masa kini. Seperti contohnya *search engine* sederhana ini banyak memanfaatkan NLP.

5.2 Saran

Dari proses pengerjaan yang telah kami lalui, kami mendapatkan beberapa saran untuk kami maupun pembaca laporan kami sebagai berikut :

1) Memperbanyak diskusi agar masalah-masalah yang dihadapi dapat lebih cepat terselesaikan.
2) Mencari lebih banyak referensi terlebih dahulu agar dalam proses pengerjaan menjadi lebih optimal
3) Pembagian tugas dilakukan dengan jelas.
4) Selalu mengecek ulang program untuk meminimalisir adanya minus-minus yang minor.

5.3 Refleksi

Dengan selesainya proses pengerjaan dan menuai hasil akhir dalam mengerjakan tugas besar ini, ada beberapa hal yang dapat kami refleksikan. Yaitu sebagai berikut :

1) Tugas ini membuka pertemanan dan tali persaudaraan yang baru.
2) Bonding async.
3) Menambah wawasan baru mengenai pembuatan web.
4) Ketika bingung, rajin-rajin mencari jawaban dari kebingungan pada internet.

DAFTAR REFERENSI

Munir, Rinaldi. 2020. Sistem persamaan linier (Aplikasi dot product pada system temu-balik informasi (information retrieval). https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/algeo20-21.htm (diakses pada 11 November 2020).

Frank D and Nykamp DQ, "An introduction to vectors." From *Math Insight*. http://mathinsight.org/vector_introduction

Frank, D and Nykamp, DQ. "An introduction to vectors.". https://mathinsight.org/vector_introduction (diakses pada 11 November 2020).

Suhartono, Derwin. "Vector Space Model dalam Pengolahan Teks.". https://socs.binus.ac.id/2018/11/29/vector-space-model-dalam-pengolahan-teks/. (diakses pada 11 November 2020).

dataperspective. 2017. "Information retrieval document search using vector space model in R.". https://www.datasciencecentral.com/profiles/blogs/information-retrieval-document-search-using-vector-space-model-in (diakses pada 11 November 2020).