

Tugas Besar 1 IF 2123 Aljabar Linier dan Geometri Aplikasi Dot Product  
pada Sistem Temu-balik Informasi  
Semester I Tahun 2020/2021



Kelompok bismillah kelar:

13519027 - Haikal Lazuardi Fadil

13519154 - Rafi Raihansyah Munandar

13519161 - Harith Fakhiri Setiawan

Asisten :

13518040 - Muhammad Farid Adilazuarda

## 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

### 1. Vektor

Vektor merupakan besaran yang memiliki besar dan arah. Sebuah vektor  $\vec{u}$  merupakan sebuah list dari angka angka

$$u = (u_1, u_2, \dots, u_n)$$

di mana  $u_i$  disebut sebagai komponen dari  $\vec{u}$ . Besar dari vektor  $\vec{u}$  dituliskan dengan

$$\|\vec{u}\| \equiv u.$$

Posisi, perpindahan, kecepatan dan percepatan merupakan contoh dari vektor. Vektor memiliki sifat-sifat sebagai berikut:

- a. Jika vektor memiliki ukuran dan arah yang sama, maka vektor tersebut dikatakan sama.
- b. Agar dapat dikurang dan ditambah, vektor harus memiliki dimensi atau ukuran yang sama.
- c. Jika vektor negatif dengan besaran yang sama, vektor tersebut berlawanan arah dengan positif.
- d. Pengurangan vektor dapat dilakukan dengan menjumlahkan vektor positif dengan negatif dan vektor negatif dengan vektor negatif.
- e. Hasil dari perkalian ataupun pembagian vektor adalah skalar.
- f. Proyeksi vektor sepanjang sumbu koordinat disebut komponen vektor.

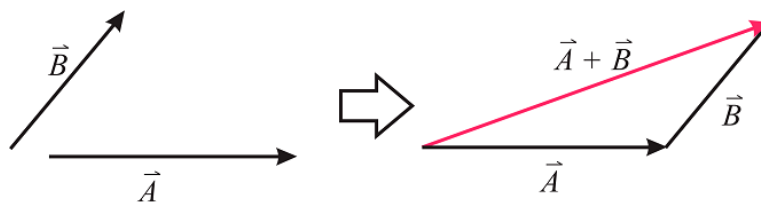
Jika ada vektor  $\mathbf{u}$  dengan komponen  $(u_1, u_2, u_3)$ , maka panjang vektor  $\mathbf{v}$  dihitung

$$\|\vec{u}\| = \sqrt{(u_1)^2 + (u_2)^2 + (u_3)^2}.$$

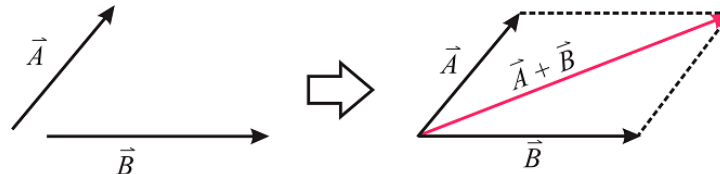
Jika ada juga vektor  $\mathbf{v}$   $(v_1, v_2, v_3)$

$$\mathbf{u} + \mathbf{v} = (u_1 + v_1, u_2 + v_2, u_3 + v_3)$$

Dengan ilustrasi sebagai berikut



Gambar 1. Ilustrasi pertambahan vektor metode poligon



Gambar 2. Ilustrasi pertambahan vektor metode jajar genjang

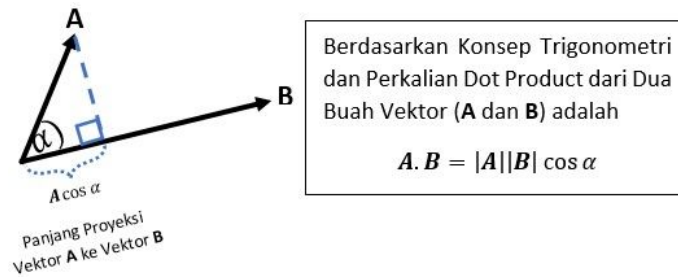
Sumber gambar 1 dan 2 :

<https://www.zenius.net/prologmateri/matematika/a/136/penjumlahan-vektor>

Jika vektor  $\mathbf{u}$  dan  $\mathbf{v}$  dioperasikan secara *dot product* maka,

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

$$\mathbf{u} \cdot \mathbf{v} = u_1 v_1 + u_2 v_2 + u_3 v_3$$



Gambar 3. Ilustrasi *dot product* vektor

Sumber gambar :

<https://www.gammafisblog.com/2019/04/konsep-perkalian-titik-dot-product-dari.html>

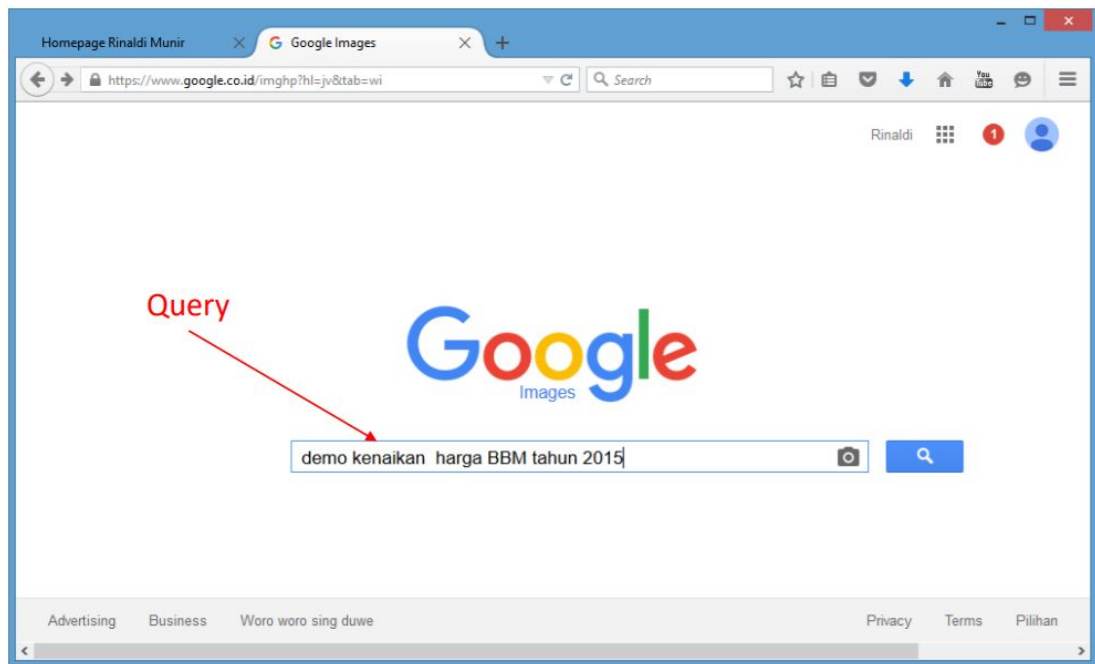
## 2. Information Retrieval

*Information Retrieval* pada dasarnya merupakan ilmu yang mempelajari tentang prosedur serta metode yang ditujukan untuk menemukan kembali informasi yang tersimpan dari berbagai sumber yang relevan atau koleksi sumber informasi yang dicari atau dibutuhkan. *Information Retrieval* ini biasa dikenal juga dengan “Temu Kembali Informasi” atau “Temu-balik Informasi”. Secara singkat, Temu-balik Informasi ini digunakan untuk menemukan kembali informasi sesuai dengan kebutuhan pengguna secara otomatis.

Proses dalam Temu-balik Informasi ini berupa *indexing*, *searching*, serta *recalling*. Pemrosesan data pada temu kembali informasi ini dapat berupa *text*, *table*, gambar, video, maupun audio. Terdapat beberapa poin penting yang perlu diketahui terkait Temu Kembali Informasi, diantaranya:

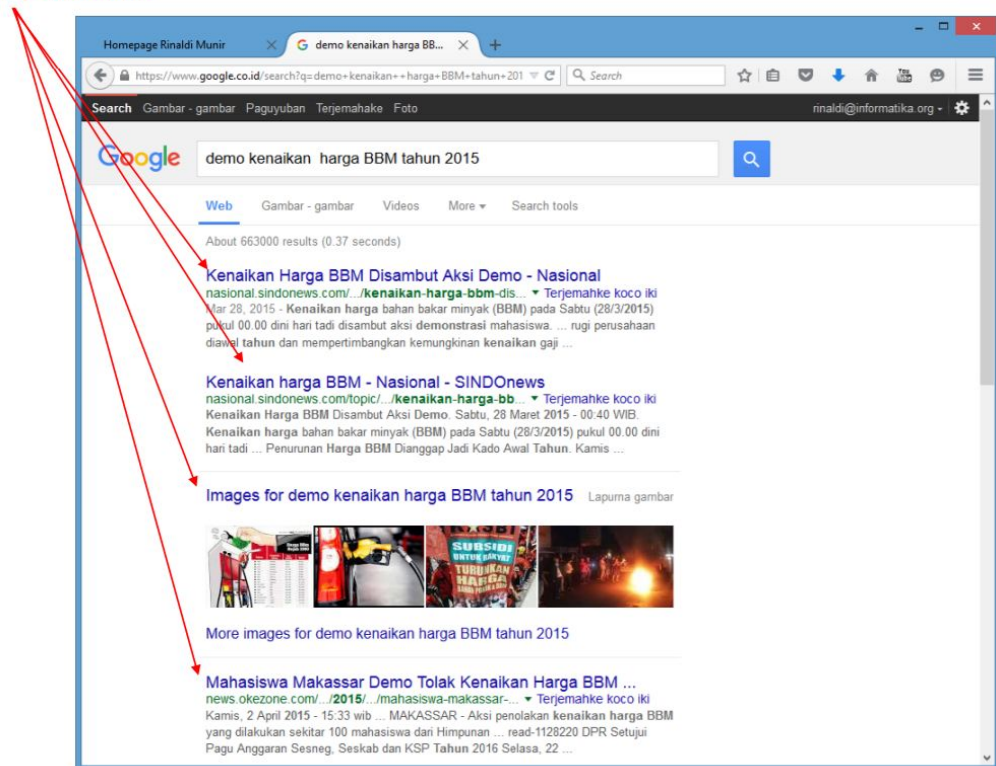
- Temu Kembali Informasi tidak sama dengan pencarian di dalam basis data (*database*)
- Pada umumnya, Temu Kembali Informasi digunakan untuk mencari informasi yang tidak terstruktur. Bentuk terstruktur contohnya adalah tabel tabel di dalam basis data. Sementara yang tidak terstruktur, umumnya berupa dokumen seperti artikel pada suatu laman web.
- Salah satu aplikasi dari Temu Kembali Informasi adalah *Search Engine*, atau mesin pencari, yang membutuhkan *query* yang digunakan sebagai parameter pencarian dari informasi yang dibutuhkan.

Contoh :



Gambar 4. Contoh aplikasi Temu Kembali Informasi

Hasil pencarian:



Gambar 5. Contoh aplikasi Temu Kembali Informasi

Sumber gambar 4 dan 5 : [Aplikasi Dot Product pada Sistem Temu Kembali Informasi by Rinaldi Munir](#)

Pada tugas kali ini, metode yang digunakan adalah Model Ruang Vektor pada Temu-balik Informasi atau *Information Retrieval*.

a. Model Ruang Vektor

Misalkan terdapat tiga buah kata ( $T_1$ ,  $T_2$ ,  $T_3$ ) serta dokumen  $D_1$  dan  $D_2$  serta sebuah *query*  $Q$ .

$T_1$  = Manusia

$T_2$  = Jawa

$T_3$  = Indonesia

$Q$  = Mobil buatan Indonesia itu memiliki ciri khas Indonesia

$D_1$  = Manusia adalah makhluk sosial. Salah satu daerah Indonesia yang jumlah manusianya banyak ada di Jawa. Jawa merupakan salah satu pulau yang paling padat di Indonesia. Ibukota Indonesia pun terdapat di Provinsi Jawa Barat. Indonesia adalah negara kepulauan yang kaya akan budayanya. Tidak heran juga banyak negara yang ingin mengakuisisi kebudayaan Indonesia.

$D_2$  = Manusia adalah makhluk paling pintar di bumi. Selain pintar, manusia juga memiliki emosi yang dapat membuatnya merasa sedih, senang, dan marah. Salah satu daerah Indonesia yang jumlah manusianya banyak ada di Jawa. Pulau Jawa merupakan pulau yang sangat padat penduduk. Banyak sekali mata pencaharian penduduk yang cukup beragam di Pulau Jawa. Pulau Jawa sendiri dibagi menjadi beberapa provinsi seperti Jawa Barat, Jawa Timur, Jawa Tengah, DKI Jakarta, dan lain lain.

Berdasarkan pernyataan diatas, didapat bahwa  $Q = (0,0,2)$ ,  $D_1 = (2,3,5)$ , dan  $D_2 = (3,7,1)$  serta dapat direpresentasikan sebagai berikut.

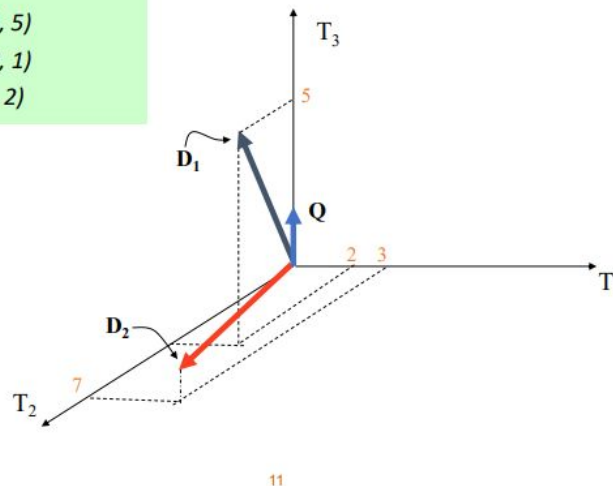
## Representasi grafik vektor

Contoh:

$D_1 = (2, 3, 5)$

$D_2 = (3, 7, 1)$

$Q = (0, 0, 2)$



Gambar 6. Data yang direpresentasikan sebagai grafik vektor

Sumber gambar : [Aplikasi Dot Product pada Sistem Temu Kembali Informasi by Rinaldi Munir](#)

### 3. Cosine Similarity

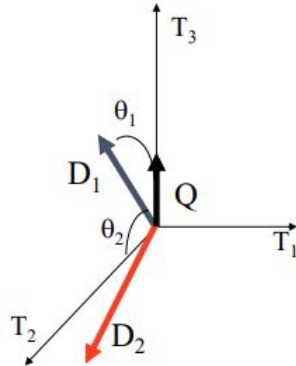
Cosine Similarity dapat didefinisikan sebagai metode yang digunakan untuk mencari kemiripan antara kedua buah vektor dengan menggunakan aplikasi *dot product* dari dua vektor yang bukan nol. Secara mendasar, kemiripan dari kedua buah vektor tersebut diukur dari sudut yang dibentuk dari kedua vektor tersebut. Semakin kecil sudut yang dibentuk dari kedua vektor tersebut, semakin besar similaritas dari kedua vektor tersebut.

*Dot Product* dari dua buah vektor dapat dicari menggunakan rumus:

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

$$\mathbf{Q} \cdot \mathbf{D} = q_1 d_1 + q_2 d_2 + \dots + q_n d_n$$

Pada kasus ini, Q dilambangkan sebagai vektor *Query*, atau kata kunci yang dibutuhkan pengguna. Dan D dilambangkan sebagai vektor dari kata unik dokumen yang terdapat pada data tidak terstruktur.



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

Gambar 7. Ilustrasi *Cosine similarity*

Sumber gambar : [Aplikasi Dot Product pada Sistem Temu Kembali Informasi by Rinaldi Munir](#)

## BAB III

### Implementasi Program

**clean\_txt.py** : File ini berisi fungsi-fungsi untuk membersihkan teks untuk mempermudah pemrosesan teks. Pada file ini terdapat dua fungsi yaitu:

- a. Fungsi `clean_file`  
Fungsi ini membuka dokumen dengan ekstensi `.txt` kemudian membacanya serta mengubah huruf kapital menjadi huruf kecil dan menghilangkan berbagai macam tanda baca menggunakan *regular expression*. Fungsi ini mengembalikan *array of sentences*.
- b. Fungsi `stem`  
Sesuai namanya, fungsi ini membuat kata-kata berimbuhan menjadi kata dasar terhadap kata-kata yang sudah dibersihkan. Fungsi ini menerima masukan berupa *array of sentences* dan mengembalikan *array of sentences* juga.
- c. Fungsi `cleanQuery`  
Fungsi ini menerima *string of sentence* lalu mengubah huruf kapital menjadi huruf kecil dan menghilangkan berbagai macam tanda baca menggunakan *regular expression*. Fungsi ini mengembalikan *string of sentence*.

**cosinesim.py** : File ini berisi fungsi untuk mencari nilai *Cosine Similarity*

- a. Fungsi `sim`  
Fungsi ini menerima masukan dua buah vektor berupa array yang berfungsi untuk menghitung similaritas dari dua *array of words*.

**textprocessing.py** : File ini berisi berbagai fungsi untuk memproses teks agar dapat dicari similaritasnya.

- a. Fungsi `inputKata`  
Fungsi ini menerima string setelah proses *cleaning* dan *stemming* dan mengembalikannya sebagai *array of words*.
- b. Fungsi `listToString`  
Fungsi ini mengubah elemen-elemen array menjadi string. Prosedur ini dibutuhkan setelah proses *data cleaning* dan *stemming*. Hal ini disebabkan setelah proses tersebut terbentuk *array of sentences* sedangkan untuk proses-proses selanjutnya membutuhkan *array of words*.
- c. Fungsi `gabungarray`  
Fungsi ini menggabungkan *array of words* dari *query* dengan *array of words* dari dokumen-dokumen yang ada. Fungsi ini akan mengembalikan hasil konkatenasi dari dua *array of words*.
- d. Fungsi `removeduplicatex`  
Fungsi ini menghapus kata-kata yang sama dari *array of words* dan mengembalikan *array of words* dengan elemen-elemen kata yang unik.
- e. Fungsi `jumlahKata`  
Fungsi ini menghitung jumlah masing-masing elemen *array of words* yang terdapat pada suatu dokumen dan suatu *query (count vectorizer)*.



**\_\_init.py\_\_** : File ini berfungsi untuk mengimport file dari directory lain.

**home.html** : File ini menampilkan *homepage* (laman yang pertama muncul). Kontennya berisi *file uploader*, *link to file converter*, dan *search query* awal.

**index.html** : File ini merupakan struktur awal untuk menampilkan hasil query.

**perihal.html** : File ini menampilkan konten perihal yang berisi seputar bagaimana program bekerja, serta pembuat program (anggota kelompok).

**search.html** : File ini merupakan *extension* dari index.html dan berfungsi untuk menampilkan hasil *query*.

**main.py** : File ini digunakan untuk menjalankan *live server* yang menampilkan web page.

*Framework* yang digunakan adalah Flask. Flask menghubungkan *frontend* dengan *backend* sehingga program yang diinput dari *frontend* dapat diproses di *backend* kemudian ditampilkan kembali hasilnya pada *web page*. Beberapa fungsi yang digunakan berupa:

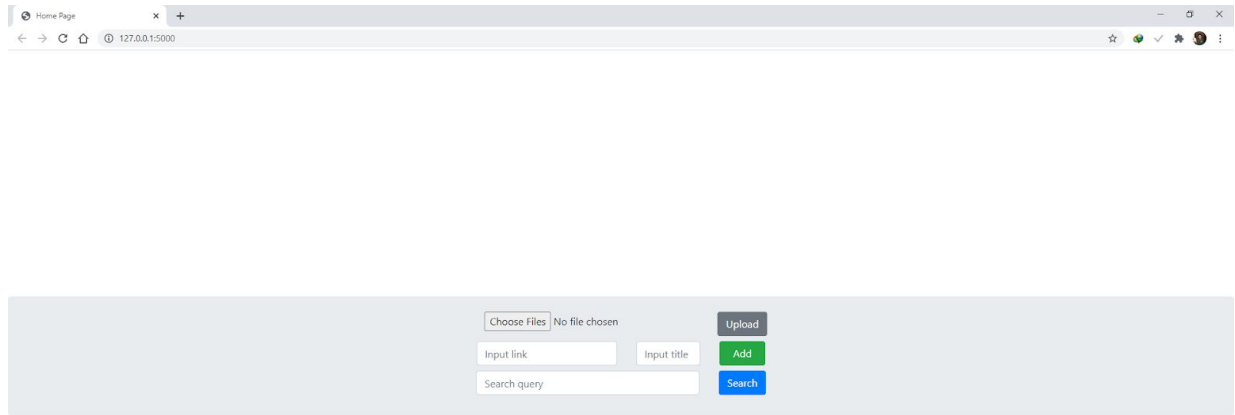
- a. Fungsi `allowed_filename`  
Menerima file kemudian mengembalikan apakah file tersebut memenuhi extension yang diterima (.txt) atau tidak.
- b. Fungsi `sort`  
Menerima array of tuple berupa nama file, tingkat similaritas, kemudian mengurutkannya berdasarkan nilai similaritas tertinggi (tuple berindex 0 memiliki nilai similaritas tertinggi).
- c. Fungsi `home`  
Menampilkan *home page*.
- d. Fungsi `upload`  
Menerima submit button kemudian menggunakan metode POST untuk mengirim input yang kemudian diproses pada *backend*.
- e. Fungsi `search`  
Memproses hasil input *query* dan dokumen kemudian menampilkan hasil *query* yang berturut dari nilai similaritas paling tinggi serta tabel jumlah kata *query* pada masing-masing file dokumen.
- f. Fungsi `perihal`  
Menampilkan laman perihal.

**webScrapping.py** : File ini berfungsi untuk membuat dokumen dengan format .txt yang diambil dari sebuah tautan.

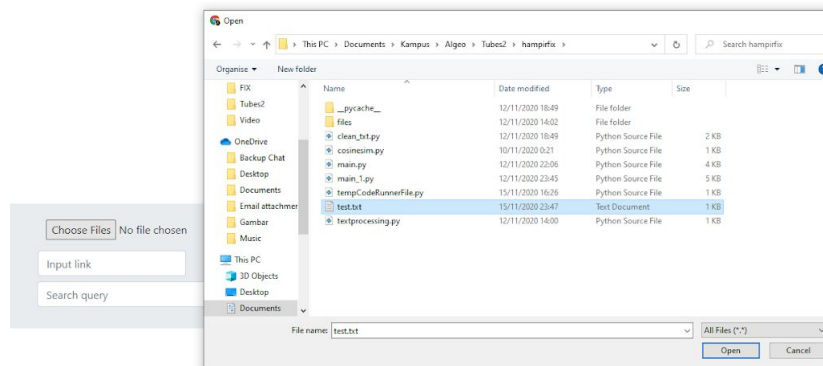
- a. Fungsi `webScrapping`  
Fungsi ini menerima input berupa tautan yang berisikan sebuah artikel, kemudian membuat file dengan format .txt dari artikel pada tautan terkait. Tautan terkait hanya dapat diambil dari "kompas.com".

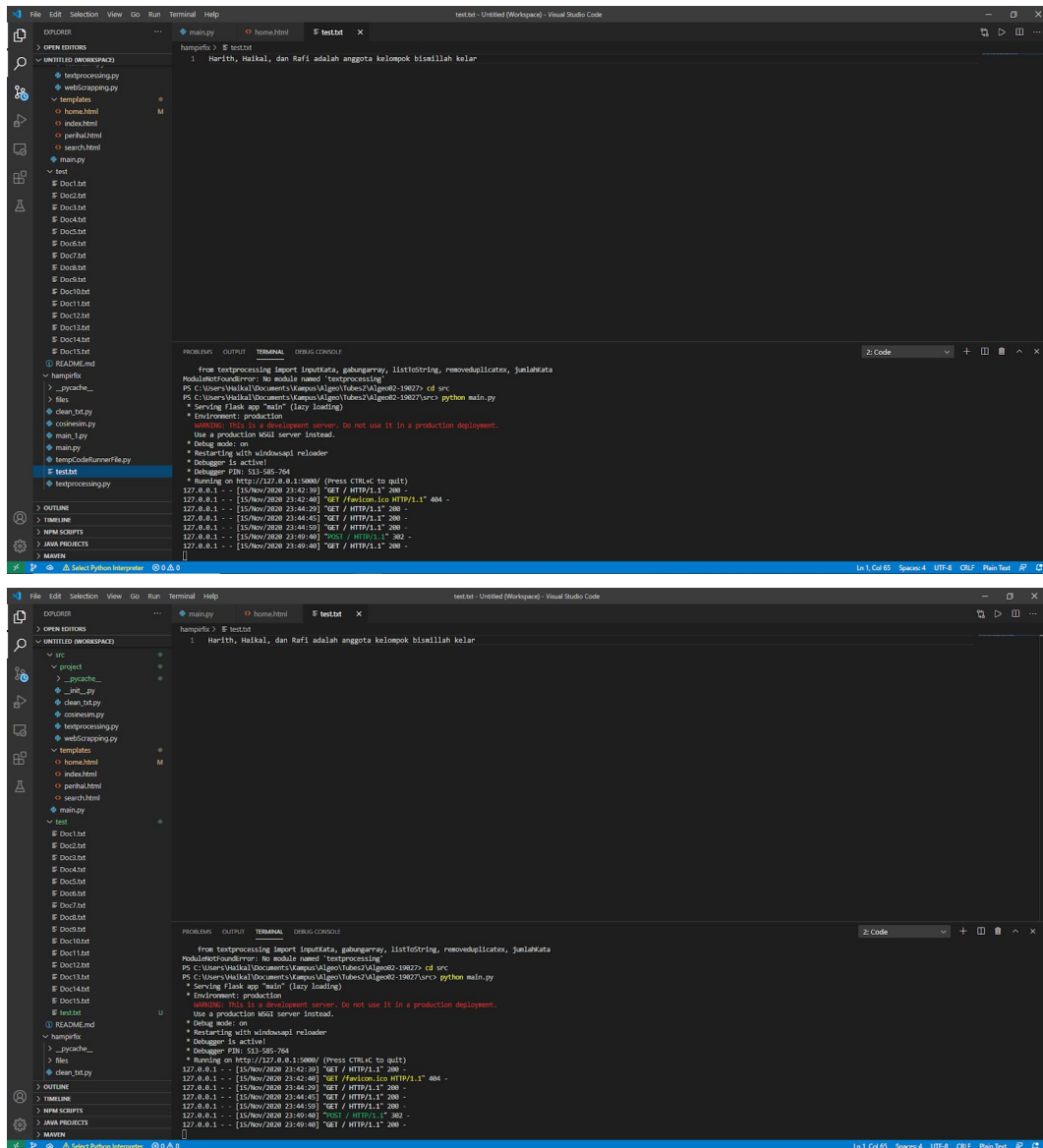
## BAB IV Eksperimen

### 1. Tampilan Home Page.



2. File dapat di-*upload* dengan cara menekan tombol 'choose file', kemudian akan muncul window untuk memilih file yang akan di-*upload*. Setelah itu, tekan tombol 'upload' untuk meng-*upload* file yang sudah dipilih.

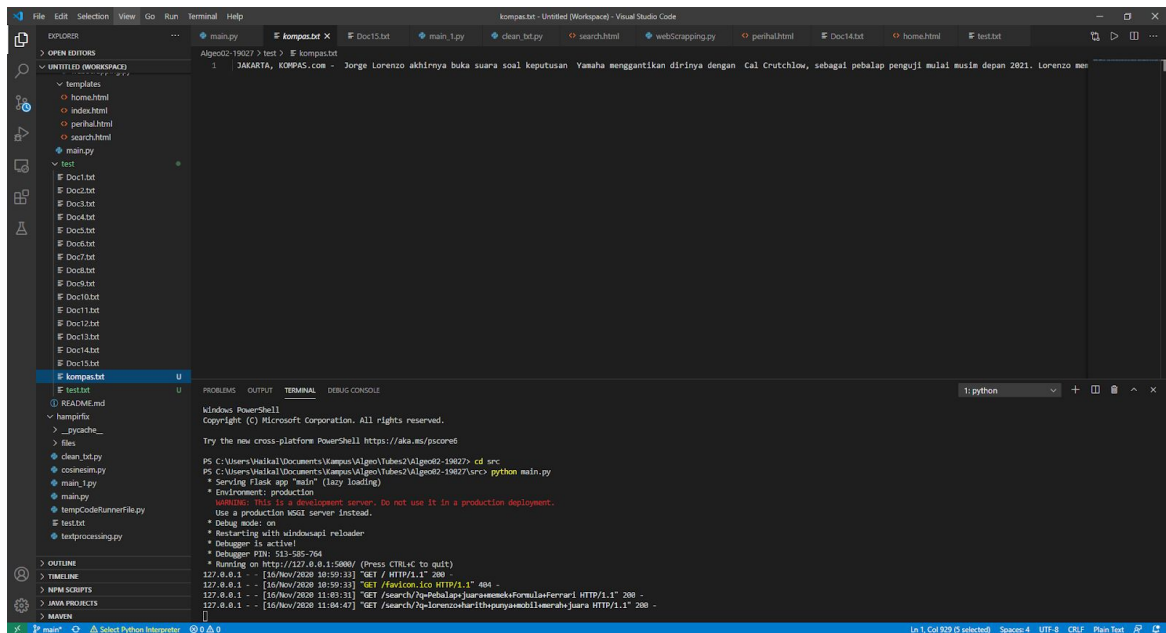




4. Selain itu, kita juga dapat mengambil artikel dari internet, kemudian menyimpannya dalam bentuk file .txt. Kali ini, kami merancang agar program kami dapat mengambil semua artikel dari kompas.com dengan cara menyalin tautan dari artikel tersebut kemudian menempelkannya di form link yang sudah kami buat. Setelah itu, kita dapat menentukan nama file yang akan menampung artikel yang sudah kita pilih. Jika sudah, klik tombol 'add'.

No file chosen

5. Berikut hasilnya. Seperti yang kita lihat, file tersebut sudah terbentuk dalam *directory* kami.

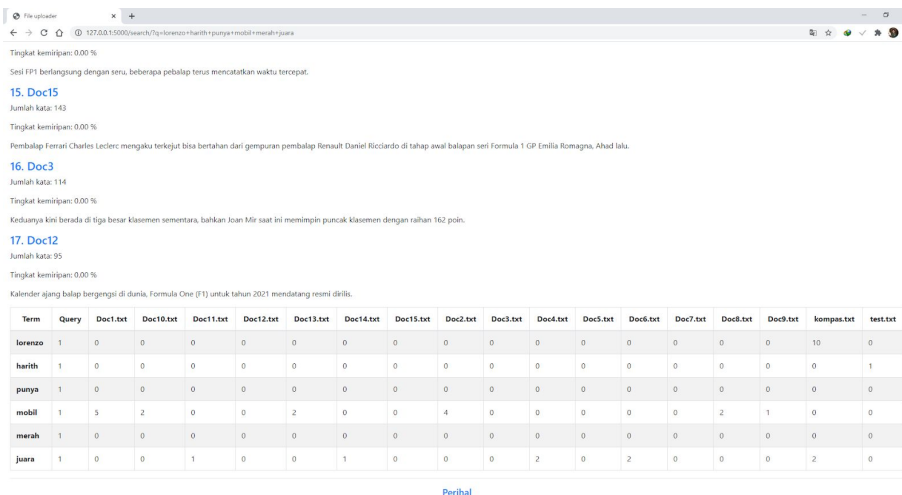
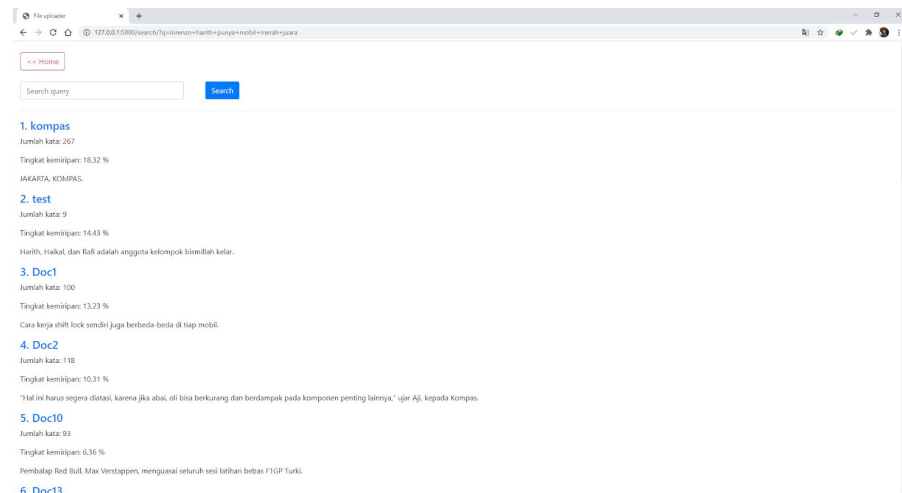


6. Untuk menjalankan program utama, ketik suatu kata atau kalimat dalam form *query* yang sudah disediakan. Setelah itu, klik 'search'.

The screenshot shows a web application interface with the following elements:

- A 'Choose Files' button with the text 'No file chosen'.
- An 'Input link' field containing the text 'lorenzoharithpunyamobilmerahjuara'.
- An 'Input title' field.
- A 'Search' button.
- A 'Perihal' label at the bottom.

7. Setelah melakukan langkah tersebut, akan ditampilkan hasil dari *searching* yang kita lakukan. Pada page tersebut, terdapat nama dokumen, jumlah kata, tingkat kemiripan dokumen dengan *search query*, dan kalimat pertama dari dokumen tersebut. Selain itu, pada bagian bawah halaman, ada tabel jumlah kata unik yang terdapat pada dokumen yang sesuai dengan *search query*.



8. Jika judul dokumen tersebut diklik, akan muncul isi dari dokumen tersebut.

File uploader

127.0.0.1:5000/search/?q=loranop+harith+puanya+mobil+mERAH+juara

<< Home

Search query

Search

1. [kompas](#)

Jumlah kata: 267

Tingkat kemiripan: 18.32 %

JAKARTA, KOMPAS.

2. [test](#)

Jumlah kata: 9

Tingkat kemiripan: 14.43 %

Harith, Haikal, dan Rafi adalah anggota kelompok bioteknik kelas.

3. [Doc1](#)

Jumlah kata: 100

Tingkat kemiripan: 13.23 %

Cara kerja shift lock sendiri juga berbeda-beda di tiap mobil.

4. [Doc2](#)

Jumlah kata: 118

Tingkat kemiripan: 10.31 %

"Hal ini harus segera diatasi, karena jika abai, oli bisa berkurang dan berdampak pada komponen penting lainnya," ujar Aji, kepada Kompas.

5. [Doc10](#)

Jumlah kata: 93

Tingkat kemiripan: 6.36 %

Pembalap Red Bull, Max Verstappen, menguasai seluruh sesi latihan bebas F1GP Turki.

127.0.0.1:5000/hes/kompas.txt

127.0.0.1:5000/hes/Doc14.txt

127.0.0.1:5000/hes/Doc14.txt

Ada cerita lucu ketika juara dunia Formula 1 Lewis Hamilton mencoba gaya selebrasi rekannya, pembalap Australia Daniel Ricciardo. Mereka berdua menyapang sampung dari sepatu. Ricciardo, memang sudah sering merayakan kemenangannya di podium dengan selebrasi ikonik ini. Ia sudah melakukannya sejak 2010, dan baru-baru ini ia juga melakukannya setelah finis di urutan ketiga di Eifel Grand Prix Jerman. Saat pembalap kelahiran Perth itu menungkan sampung ke sepatunya, Hamilton - yang finis di posisi pertama - memutuskan untuk bergabung dalam selebrasi itu. Ricciardo mengatakan dia terkejut ketika Hamilton tiba-tiba bersebelah, mengingat dia telah melihat saingan selebrasi "shony" itu di podium sebelumnya.

## BAB V

### Kesimpulan, Saran, dan Refleksi

#### 1. Kesimpulan

Berdasarkan proses pengerjaan yang sudah dilakukan dan hasil eksperimen yang didapat, penulis memperoleh kesimpulan yang dapat diambil dari Tugas Besar 2 IF 2123 Aljabar Linier dan Geometri tentang Aplikasi *Dot Product* pada sistem Temu-balik Informasi Semester I Tahun 2020/2021 sebagai berikut:

- Vektor merupakan besaran yang memiliki besar dan arah atau biasa dikenal dengan skalar. Vektor juga dapat dioperasikan dengan berbagai cara, seperti penjumlahan dan pengurangan, perkalian atau *dot product*, serta *cross product*.
- *Information Retrieval* atau Temu-balik Informasi merupakan ilmu yang mempelajari tentang prosedur yang bertujuan untuk menemukan informasi yang tersimpan dari berbagai sumber yang relevan atau koleksi sumber informasi yang dicari atau dibutuhkan. Umumnya, Temu-balik Informasi digunakan untuk mencari informasi yang tidak terstruktur. Selain itu, salah satu aplikasi dari Temu-balik Informasi adalah *Search Engine*, atau mesin pencari, yang membutuhkan *query* yang digunakan sebagai parameter pencarian dari informasi yang dibutuhkan. Pada tugas kali ini, metode yang digunakan adalah Model Ruang Vektor pada Temu Kembali Informasi atau *Information Retrieval*.
- Model ruang vektor pada *Information Retrieval* atau Temu-balik Informasi digunakan dengan cara membuat sejumlah vektor, yang terdiri dari 1 vektor yang terkait dengan *query*, dan beberapa vektor terkait dokumen yang ingin dicari similaritasnya. Elemen dari vektor ini dicari dengan cara menghitung jumlah setiap kata unik yang terdapat pada *query* serta dokumen terkait.
- *Cosine Similarity* merupakan metode yang digunakan untuk mencari kemiripan antara kedua buah vektor dengan menggunakan aplikasi *dot product* dari dua vektor bukan nol. *Cosine Similarity* ini dihitung dengan mencari sudut dari kedua vektor yang akan dibandingkan kemiripannya. Semakin kecil sudut yang dibentuk dari kedua vektor tersebut, semakin besar similaritas dari kedua vektor tersebut.
- Framework Flask digunakan untuk menyambungkan *frontend* (berupa tampilan pada html) dengan *backend* (pemrosesan input menggunakan fungsi-fungsi dan library pada python).

#### 2. Saran

Berdasarkan proses pengerjaan dan hasil eksperimen, tim penulis ingin memberi saran agar program yang sudah dibuat terus dikembangkan agar tidak sia-sia dan tidak hanya dikerjakan sebagai pemenuhan kewajiban untuk mengerjakan tugas besar. Menurut tim penulis, program dengan model ruang vektor seperti ini bisa diaplikasikan kepada hal lain. Oleh karena itu, program terbaik dari angkatan penulis sebaiknya dibagikan kepada rekan-rekan seangkatan penulis sehingga bisa dijadikan pelajaran bagi rekan-rekan penulis yang belum menyelesaikan tugas besarnya dengan baik.

#### 3. Refleksi

Dalam proses pengerjaan tugas ini, penulis mendapatkan banyak pelajaran berharga. Penulis dapat mengenal dan memahami bahasa pemrograman Python lebih lanjut. Selain itu, penulis juga dapat menguasai operasi-operasi pada vektor, khususnya *dot product* serta dapat mengaplikasikan *Information Retrieval* berupa *search engine* dengan metode ruang vektor. Penulis dapat lebih mengenal dan memahami pembuatan *frontend* dengan memanfaatkan html, css, bootstrap. Selain itu, penulis juga mempelajari cara menggabungkan *frontend* dengan *backend* melalui *framework* python yaitu Flask.

Penulis juga lebih mengenal cara bekerja dalam tim yang baik, saling memahami antar anggota, makna saling membantu satu sama lain, menghormati pendapat orang lain, dan tentunya bersosialisasi. Penulis berharap bisa mengerjakan tugas yang akan datang dengan lebih baik mengingat sudah mendapatkan pengalaman dalam mengerjakan tugas besar.



## REFERENSI

<https://ligiaprpta17.wordpress.com/2015/03/03/pengertian-information-retrieval-ir-peranan-ir-dan-contoh-contoh-ir/>

<https://www.geeksforgeeks.org/reading-writing-text-files-python/>

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

<https://www.youtube.com/watch?v=yN2K3tQjZ6c&t=4s>

<https://www.youtube.com/watch?v=xvqsFTUsOmc&t=2932s>

<https://www.youtube.com/watch?v=9ivlcQuRMdY>

<https://pypi.org/project/Sastrawi/>

<https://stackoverflow.com/questions/18424228/cosine-similarity-between-2-number-lists>

<https://medium.com/@ksnugroho/dasar-text-preprocessing-dengan-python-a4fa52608ffe>

<https://www.youtube.com/watch?v=0EgSo7hsRWM>

<https://stackoverflow.com/questions/16566268/remove-all-line-breaks-from-a-long-string-of-text>

<https://stackoverflow.com/questions/275018/how-can-i-remove-a-trailing-newline>

<https://www.kite.com/python/answers/how-to-remove-all-line-breaks-from-a-string-in-python>

<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2015-2016/Makalah-2015/Makalah-IF2123-2015-095.pdf>

<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-12-Aplikasi-dot-product-pada-IR.pdf>

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

<https://programminghistorian.org/en/lessons/creating-apis-with-python-and-flask>

<https://flask.palletsprojects.com/en/1.1.x/patterns/fileuploads/>