

LAPORAN TUGAS BESAR 2

Aljabar Linier dan Geometri



Renaldi Arlin 13519114

Farrell Abieza Zidan 13519182

Leonardus James Wang 13519189

IF 2123

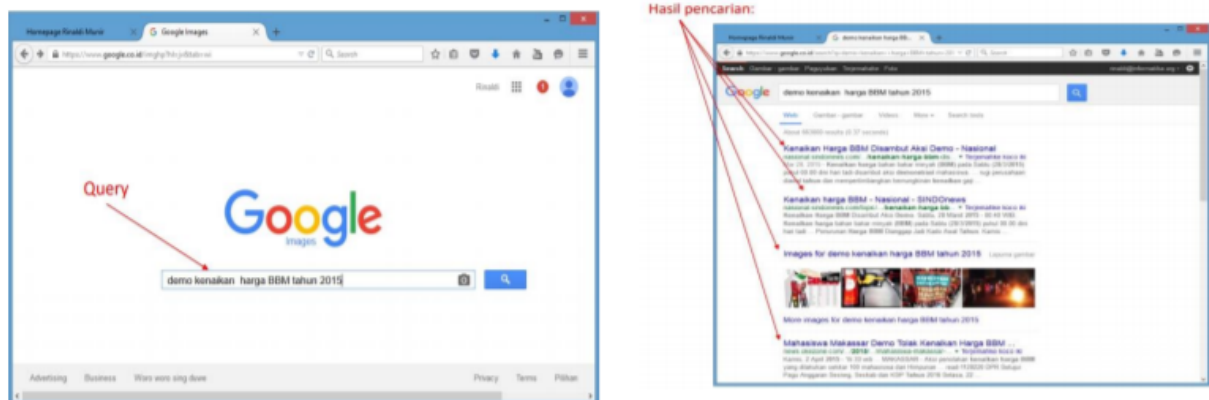
BAB 1

Deskripsi Masalah

1.1 Abstraksi

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.



Gambar 1. Contoh penerapan Sistem Temu-Balik pada mesin pencarian.
sumber: Aplikasi Dot Product pada Sistem Temu-balik Informasi by Rinaldi Munir

Ide utama dari sistem temu balik informasi adalah mengubah search query menjadi ruang vektor. Setiap dokumen maupun query dinyatakan sebagai vektor $w = (w_1, w_2, \dots, w_n)$ di dalam R^n , dimana nilai w_i 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:

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

Pada kesempatan ini, kalian ditantang untuk membuat sebuah search engine sederhana dengan model ruang vector dan memanfaatkan cosine similarity.

BAB 2

Teori Singkat

2.1 Retrieval Information

Retrieval Information: Menemukan kembali (retrieval) informasi yang relevan terhadap kebutuhan pengguna dari suatu kumpulan informasi secara otomatis.

IR tidak sama dengan pencarian di dalam basis data (database). IR umumnya digunakan pada pencarian informasi yang isinya tidak terstruktur seperti dokumen atau *webpage*.

2.2 IR dengan Model Ruang Vektor

Salah satu model IR adalah model ruang vektor. Model ini menggunakan teori di dalam aljabar vector, khususnya perkalian titik.

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 $w = (w_1, w_2, \dots, w_n)$ di dalam ruang vektor n , w_i = bobot setiap kata i di dalam query atau dokumen. Nilai w_i dapat menyatakan jumlah kemunculan kata tersebut dalam dokumen (term frequency)

Setelah cosine similarity dihitung (dijelaskan kemudian), hasil perhitungan di-ranking berdasarkan nilai cosinus dari besar ke kecil sebagai proses pemilihan dokumen yang yang “dekat” dengan query. Pe-ranking-an tersebut menyatakan dokumen yang paling relevan hingga yang kurang relevan dengan query. Nilai cosinus yang besar menyatakan dokumen yang relevan, nilai cosinus yang kecil menyatakan dokumen yang kurang relevan dengan query.

2.3 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 $Q = (q_1, q_2, \dots, q_n)$ dan $D = (d_1, d_2, \dots, d_n)$ diukur dengan rumus cosine(cosinus) similarity yang merupakan bagian dari rumus perkalian titik (dot product) dua buah vektor:

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

dengan $Q \cdot D$ adalah perkalian titik yang didefinisikan sebagai

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

Jika $\cos = 1$, berarti $= 0$, vektor Q dan D berimpit, yang berarti dokumen D sesuai dengan query Q. Jadi, nilai cosinus yang besar (mendekati 1) mengindikasikan bahwa dokumen cenderung sesuai dengan query

BAB 3

Implementasi

3.1 Implementasi front-end program

Front-end program pada proyek kali ini utamanya menggunakan library React.js dan Tailwind.css. Library React.js dimanfaatkan sebagai struktur utama dari proyek kali ini. Dalam implementasinya, code-style dari proyek kali ini adalah functional based component sehingga memanfaatkan React Hooks, ini digunakan agar code lebih terlihat rapi dan jelas. Untuk color theme dari web app kali ini adalah teal dan white untuk dapat memenuhi konsep minimalis. Background utama dari Web App menggunakan background yang terbentuk dari heropattern.

Konsep dalam struktur navigasi pada proyek kali ini menggunakan React Router. Sehingga web app kali ini adalah Single Page App, dimana hanya mengganti route dari URL akan menggantikan komponen yang akan ditampilkan pada App.js.

Sedangkan styling dari web app kali ini menggunakan Tailwind.css, sebuah framework css yang memaksimalkan utilitas. Untuk file css yang terbentuk dari Tailwind terdapat pada main.css. Selain styling, animasi pada web app kali ini juga kebanyakan menggunakan Tailwind. Contohnya untuk animasi saat hover pada sebuah komponen atau focus. Namun, untuk animasi setiap re render menggunakan library React Spring.

Selain itu, interaksi Front-end dengan Back-end diterapkan dengan bantuan library axios. Response dari axios akan digunakan sebagai nilai state baru pada Front-end. State dari hasil request pada axios digunakan dalam memvisualisasikan hasil search query dengan cara dimapping tiap arraynya lalu mengembalikan sebuah komponen.

3.2 Implementasi back-end program

Back-end program yang bekerja pada file utama server.js mulanya membuat isi suatu dokumen menjadi sebuah vektor berdimensi R, dengan R adalah jumlah setiap kata berbeda yang ada di dalam dokumen tersebut. Lalu program juga akan membuat sebuah vektor lain yang mana merupakan sebuah Query dari inputan user di web search engine ini.

Program akan memanggil prosedur “readTxt (namaFile)”, secara garis besar prosedur ini akan membaca beberapa “file.txt” yang merupakan sebuah dokumen untuk setiap file. Isi dokumen akan diubah menjadi sebuah string, lalu diubah lagi menjadi array yang berisi kata-kata yang sudah di stemming (contohnya sebuah array of string yang tidak memiliki tanda baca lagi dan hanya berupa kata kata). Prosedur kemudian memanggil “afterReadTxt (namaFile)” untuk mengisi variable dataMatriks dengan array of string yang sudah di stemming tadi dengan bantuan fungsi “masukTabel(arr, brsMatriks)”.

Untuk pembuatan Query, program mengambil sebuah string masukan user yang didapat dari Front-end dari komponen input, kemudian value input tersebut dimasukkan sebagai parameter dalam GET request endpoint axios, lalu mengubahnya menjadi bentuk yang sama seperti vektor dokumen sebelumnya, prosesnya diubah di dalam fungsi "queryToArray(str)".

Variable dataMatriks akan berguna pada fungsi perhitungan similaritas cosine. Fungsi fungsi yang bertanggung jawab dalam pembentukan ranking dokumen diantaranya fungsi "similaritasVektor(tabelQuery)", fungsi "makeQueryTable(matriksSearch, dataMatriks)", fungsi "makeRealQueryTable(matriksSearch, dataMatriks)", dan fungsi "sortRank(listCosine, dataMatriks)". Fungsi makeQueryTable memiliki inputan matriksSearch, sebuah matriks $N \times 2$, dengan N merupakan kata kata yang berbeda satu sama lain dari inputan user yang sudah diubah menjadi array. masukan dataMatriks yang diisi oleh variable dataMatriks diperlukan untuk mengisi kolom dokumen yang ada didalam tabel query.

Fungsi makeQueryTable berbeda pada fungsi makeRealQueryTable. pada dasarnya jika fungsi makeQueryTable memberi keluaran sebuah matriks table yang isi termnya merupakan kata kata dari query saja(tabel ini sama persis seperti yang ditulis dalam spek tugas besar ini), sedangkan fungsi makeRealQueryTable menghasilkan sebuah tabel yang termnya berasal dari semua kata yang ada di setiap dokumen, dan juga kata kata yang berasal dari inputan user / Query.

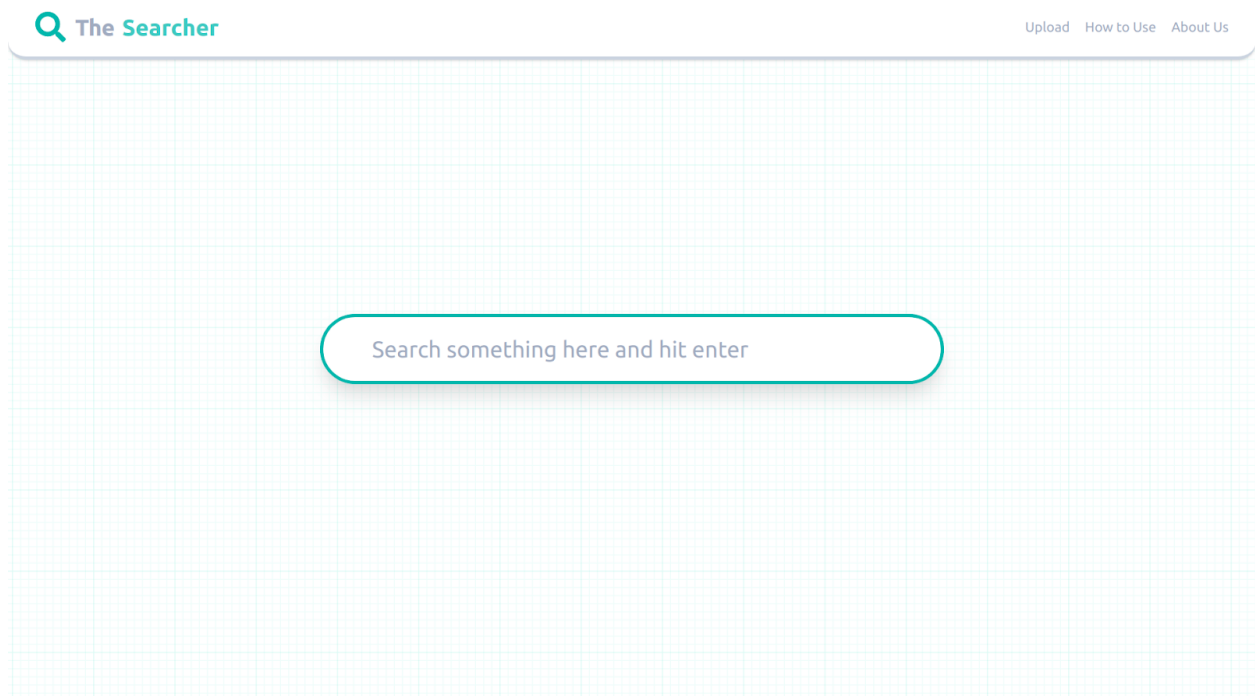
Fungsi similaritasVektor kemudian menggunakan hasil keluaran fungsi makeRealQueryTable sebagai masukan parameter tabelQuery. SimilaritasVektor merupakan fungsi yang mengimplementasi perhitungan vektor $A \cdot B$ yang dibagi dengan perkalian dari akar kuadrat A dengan akar kuadrat B. Nilai yang biasa disebut sebagai similaritas cosinus ini nantinya akan digunakan sebagai parameter seberapa tinggi dokumen tersebut memiliki kemiripan dengan query yang diinput oleh user. Fungsi similaritasVektor memiliki keluaran sebuah array float sebanyak N, dengan N merupakan jumlah dokumen yang dibaca oleh program.

Fungsi sortRank merupakan fungsi yang menghasilkan matriks objek, berisi nama Dokumen, presentase dokumen tersebut (nilai rank), dan juga isi dari dokumen itu sendiri. Fungsi sortRank menggunakan inputan listCosine (ini didapat dari kalkulasi fungsi similaritasVektor) dan nilainya dimasukkan kedalam presentase dokumen, dan inputan dataMatriks karena fungsi sortRank membutuhkan nama dokumen mana yang memiliki nilai rank tersebut.

Keluaran dari fungsi sortRank inilah yang kemudian akan di send balik kedalam Front End untuk dibaca oleh user, sekarang pengguna program mengetahui dokumen mana yang paling similar dengan inputan yang diberikannya.

BAB 4

Eksperimen



Ini adalah tampilan utama dan pertama saat menjalankan website. Terdapat 4 page di website ini, yaitu Home, Upload, How to Use dan About Us. Pada saat ini, baru terdapat 2 dokumen dalam database. Jika ingin melakukan upload, silahkan ke page upload dan pilih file agar dapat masuk ke dalam database. Jika tidak, bisa langsung mengetikkan query ke dalam search bar, lalu tekan enter untuk melakukan IR.

The Searcher

google

Upload How to Use About Us

photos.txt
 Similarity: 26.74%
 Content: Google Photos

enviroment_blame.txt
 Similarity: 0.00%
 Content: Fact-checking the US

banyak1.txt
 Similarity: 0.00%
 Content: Harmonizer plumbeous

banyak2.txt
 Similarity: 0.00%
 Content: Census nondesigned

contoh.txt
 Similarity: 0.00%
 Content: Ini dari contoh.txt

contoh2.txt
 Similarity: 0.00%
 Content: This considerations

contoh3.txt
 Similarity: 0.00%
 Content: This is a very

contoh4.txt
 Similarity: 0.00%
 Content: This is a very

alexa.txt
 Similarity: 0.00%
 Content: Alexa to start

covid.txt
 Similarity: 0.00%
 Content: Vaccine alliance

food.txt
 Similarity: 0.00%
 Content: 21 Must Eat Local

hongkong.txt
 Similarity: 0.00%
 Content: Can Hong Kong's

Ini adalah tampilan setelah beberapa dokumen lainnya di upload dan juga query google dimasukkan. Nanti terdapat beberapa beberapa nama dokumen yang muncul dengan similarity, jumlah kata dan beberapa kata pertama. Tampilan dokumen akan di pertama ke terakhir (kiri ke kanan, atas ke bawah) berdasarkan similaritas. Jika di click salah satu dokumennya, maka anda bisa melihat isinya

The Searcher

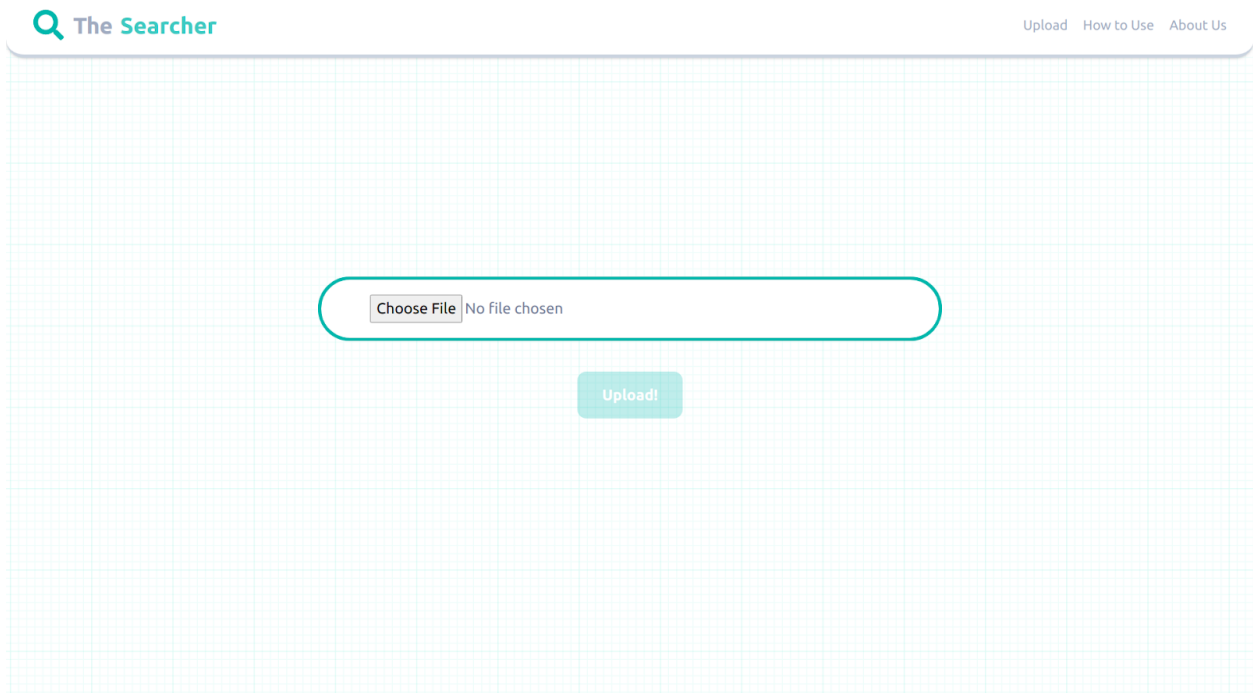
google

Upload How to Use About Us

Term	googl
Query	1
PSS.txt	0
alexa.txt	0
banyak1.txt	0
banyak2.txt	0
contoh.txt	0
contoh2.txt	0
contoh3.txt	0
contoh4.txt	0
enviroment_blame.txt	0
covid.txt	0
food.txt	0
hongkong.txt	0
pakistan.txt	0
photos.txt	12
rainforest.txt	0
scam.txt	0

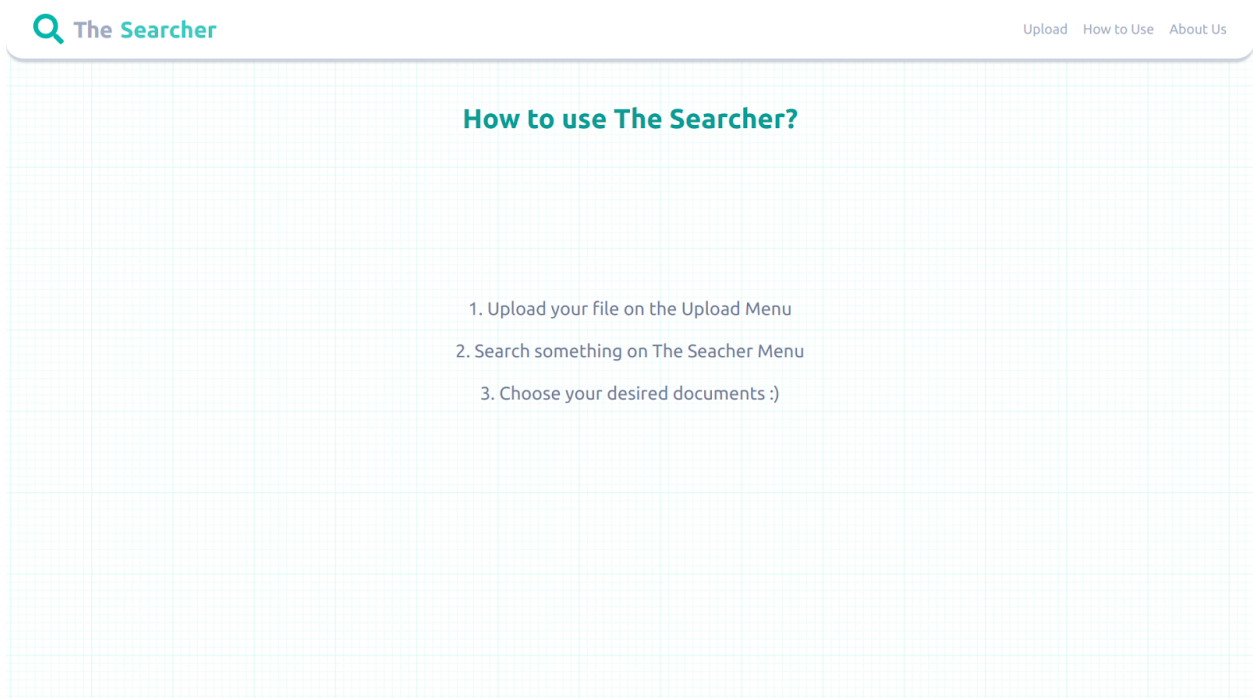
Jika scroll ke bawah, terdapat tabel yang sudah dipotong berisikan kata yang ada query setelah di stemming dan jumlah kata-kata yang ada per file dokumen.

Upload:




Di sini terdapat page untuk upload file-file yang ada. File harus dimasukkan lalu pencet upload agar masuk ke dalam database. File harus dimasukkan satu-satu.

How to use:




Ini adalah page how to use, berisikan cara pakainya.

 The Searcher

UploadHow to UseAbout Us


Get to Know Us!

The Pikirkan Nanti Saja




Renaldi Arlin
13519114

Here's something about me
IF = Begadang tiap Malam :)



Farrel Abieza Zidan
13519182

Wanna know about me?
I'm a Dedlener :)



Leonardus James Wang
13519189

This is my thought right now
Semoga Tubes cepat selesai :)

Dan ini adalah about us, page yang berisikan data-data member grup ini.

BAB V

Kesimpulan, saran, dan refleksi

Kesimpulan dari tugas ini adalah kita dapat mengaplikasikan pengetahuan matematika tentang vektor dalam dunia komputer. Aplikasi dot vektor setelah dikembangkan menjadi salah satu cara mendapat informasi yang mudah di zaman sekarang.

Saran kami adalah untuk mengembangkan lagi teknik stemmingnya agar dokumen-dokumen lebih mirip dengan query. Juga mungkin menggunakan teknik web scraping, dan juga handle file-file yang lebih besar kata-katanya.

Refleksi dari pengerjaan tugas ini adalah perlu komunikasi yang lebih baik antar orang agar input yang diminta dan data yang akan digunakan pas. Juga agar tidak melakukan meet yang sia-sia.

Referensi

<http://informatika.stei.itb.ac.id/~rinaldi.munir/>

<https://www.heropatterns.com/>

<https://reactjs.org/>

<https://tailwindcss.com/>

slide kuliah