

LAPORAN TUGAS BESAR 2

Mata Kuliah Aljabar Linear dan Geometri IF 2123

Dosen Pengampu : Rinaldi Munir, Nugraha Priya Utama

Judhi Santoso , Rila Mandala



Disusun Oleh :

Alexander	13519090
Alvin Wilta	13519163
Josep Marcello	13519164

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2020

DAFTAR ISI

DAFTAR ISI	2
BAB I	
DESKRIPSI MASALAH	3
BAB II	
TEORI SINGKAT	6
Sistem Temu Balik Informasi (Retrieval Information)	6
Vektor (Vector)	7
Kesamaan Cosinus (Cosine Similarity)	8
BAB III	
IMPLEMENTASI PROGRAM	11
Backend (Server)	11
Frontend	14
BAB IV	
EKSPERIMEN	16
Mengupload File	16
Menghapus File, Ada Dokumen yang tidak memenuhi syarat	17
Melakukan Search	17
Membuka Dokumen hasil Pencarian	18
Tabel Kata	19
Melihat Perihal (About Us)	20
Web Scraper	20
BAB V	
KESIMPULAN DAN SARAN	22
Kesimpulan	22
Saran	22
Refleksi	22
REFERENSI	24

BAB I

DESKRIPSI MASALAH

Untuk mencari suatu dokumen atau data yang relevan dengan apa yang kita inginkan (*query*), dapat digunakan *search engine* untuk mempermudah dan mempercepat pencarian. Program dapat digunakan pada *website*, dalam keadaan *local*, tidak perlu melakukan *deploy* maka dari itu, Program ini akan membutuhkan 2 hal yaitu :

1. Search *Query*

Berisi kumpulan kata yang akan digunakan untuk melakukan pencarian, kata kata yang diinput akan dicari kemiripannya dengan dokumen dokumen yang tersedia.

2. Kumpulan Dokumen

Dokumen dokumen ini dapat diunggah ke dalam peramban situs, dan dapat ditampilkan jika memang dokumen tersebut memenuhi kesamaan dengan *query*.

Contoh tampilan peramban:

My Simple Search Engine

Daftar Dokumen: <upload multiple files>

Search query

Hasil Pencarian: (diurutkan dari tingkat kemiripan tertinggi)

1. <Judul Dokumen 1>
Jumlah kata:
Tingkat Kemiripan:%
<Kalimat pertama dari Dokumen 1>
2. <Judul Dokumen 2>
Jumlah kata:
Tingkat Kemiripan:%
<Kalimat pertama dari Dokumen 2>
- ...

<Menampilkan tabel kata dan kemunculan di setiap dokumen>

Perihal

Dari gambar tersebut dapat dilihat, teks yang diberikan warna biru merupakan *hyperlink* yang akan mengalihkan halaman ke halaman yang ingin dilihat. Apabila menekan *hyperlink* <Judul Dokumen 1>, maka akan diarahkan pada sebuah halaman yang berisi full-text terkait dokumen 1 tersebut (seperti Search Engine).

Data pada perihal merupakan link ke halaman tentang program dan pembuatnya (Konsep singkat search engine yang dibuat, *How to Use, About Us*).

Data uji berupa dokumen-dokumen yang akan diunggah ke dalam web browser. Format dan *extension* dokumen dibebaskan selama bisa dibaca oleh web browser (misalnya adalah dokumen dalam bentuk file txt atau file html). Minimal terdapat 15 dokumen berbeda.

Tabel data kata dan kemunculan di setiap dokumen dapat digambarkan/ditampilkan pada web browser dengan layout dibawah ini.

Term	Query	D1	D2	...	D3
Term1					
Term2					
...					
TermN					

Agar lebih sederhana, ada hal hal yang perlu diperhatikan dalam eksekusi program ini, yaitu :

1. Stemming

Merupakan kegiatan untuk memotong imbuhan-imbuhan pada setiap kata, sehingga kata kata dengan imbuhan akan kembali menjadi bentuk asalnya, yang mana dapat membantu proses pencarian kata yang memiliki imbuhan berbeda, namun memiliki kata asal yang sama.

2. Stopwords

Merupakan kegiatan untuk menghapus kata-kata yang merupakan konjungsi, maupun kata kata yang sangat umum digunakan (contohnya : apa,adalah,dll). Dengan adanya penghapusan stopwords ini, maka pencarian akan lebih akurat karena kata kata ini biasanya terdapat pada suatu dokumen dalam jumlah yang tidak sedikit.

3. Penghapusan karakter karakter yang tidak perlu untuk ditampilkan

Jika menggunakan data dengan tipe file .html maka akan ada sintaks yang tidak diperlukan seperti “
” dan lainnya, yang akan dilakukan proses *parsing* sehingga sintaks tersebut dapat dibuang, karena tidak relevan dan akan mengganggu dalam proses pencarian.

BAB II

TEORI SINGKAT

1. Sistem Temu Balik Informasi (*Retrieval Information*)

Sistem Temu Balik Informasi atau yang biasa dikenal dengan Retrieval Information merupakan sistem yang digunakan untuk menemukan informasi yang relevan dengan kebutuhan dari penggunaannya, dengan menerapkan sistem tersebut, permasalahan pencarian informasi pada dokumen dapat memberikan hasil yang relevan dan tepat sesuai dengan input/ kebutuhan pengguna.

Sistem Temu Balik Informasi terbagi menjadi 2 proses, yaitu :

1. Proses *Index*

Proses Index merupakan proses untuk memberikan bobot pada query yang diberikan oleh pengguna, dimana bobot bobot pada query ini akan disimpan pada suatu list, yang biasanya merupakan list kemunculan kata-kata pada query. Sebelum pembobotan dilakukan, ada beberapa hal yang harus dilakukan, agar pencarian bisa menjadi lebih relevan dan akurat

a. Parsing

Parsing merupakan suatu proses untuk menguraikan dokumen yang ada, ketika dokumen yang kita gunakan bukan merupakan bentuk plaintext, bisa saja bentuk html, sehingga akan ada perbedaan formatting yang akan mengganggu proses pencarian dan dapat dihilangkan dengan metode parsing ini dengan mengubah format tersebut menjadi bentuk plaintext

b. Stemming

Pada tahap ini, query dan dokumen akan dibersihkan dari tanda baca dan menyederhanakan kata kata yang memiliki imbuhan kembali menjadi kata dasarnya.

c. Stopword

Selanjutnya query dan dokumen akan dibersihkan dari kata kata yang umum digunakan, biasanya merupakan konjungsi (contohnya : adalah, apa, dimana). Hal ini dilakukan agar pencarian dokumen dapat lebih relevan terhadap inti dari query yang ingin dicari.

2. Proses *Retrieval*

Proses retrieval merupakan proses untuk menghitung kemiripan antara dokumen dengan query dengan bantuan Cosine Similarity dimana proses, dan informasi yang sesuai akan dikembalikan kepada pengguna secara berurutan dari yang paling relevan berdasarkan penghitungan similaritas terkait..

2. Vektor (*Vector*)

Vektor merupakan sebuah elemen pada ruang vektor yang memiliki arah, dimana ruang vektor merupakan suatu struktur matematika yang dibentuk oleh sekumpulan vektor, yaitu objek yang dapat dikalikan serta dijumlahkan dengan vektor lainnya, dan juga dapat dikalikan dengan suatu skalar. Pada sistem temu balik informasi Vektor digunakan pada Vector Space Model, dimana sebuah model aljabar untuk menggambarkan dokumen maupun query dengan term yang berisi kata. Yang dapat dituliskan sebagai berikut

$$D = [T1, T2, T3,Tn]$$

$$Q = [Q1, Q2, Q3,Qn]$$

D = Vektor dokumen yang memuat term-term yang didapat dari query

T = Term yang berupa kata-kata dasar yang didapat pada query

Q = Vektor query yang memuat penggalan kata-kata dasar

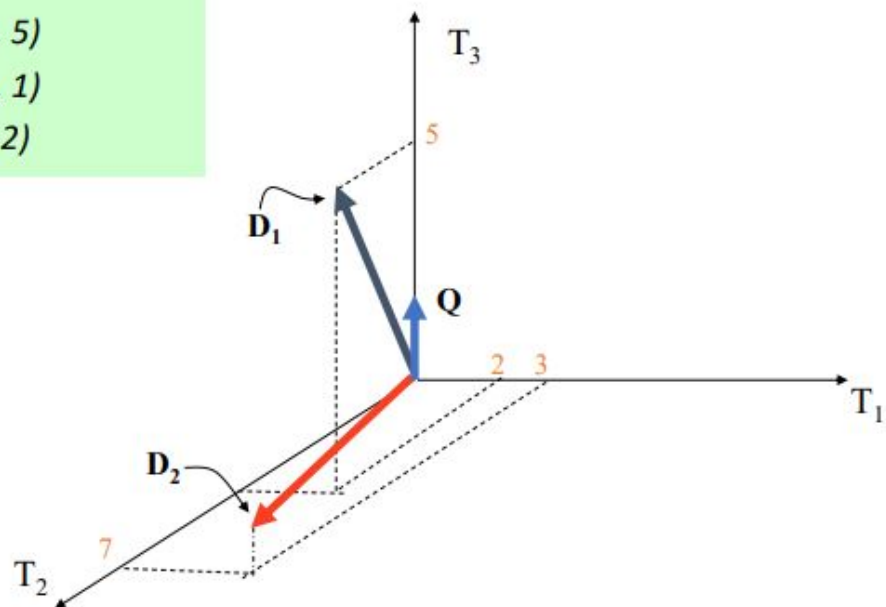
Term frequency akan dimasukkan ke dalam array. jika direpresentasikan pada grafik vektor, akan membentuk

Contoh:

$$D_1 = (2, 3, 5)$$

$$D_2 = (3, 7, 1)$$

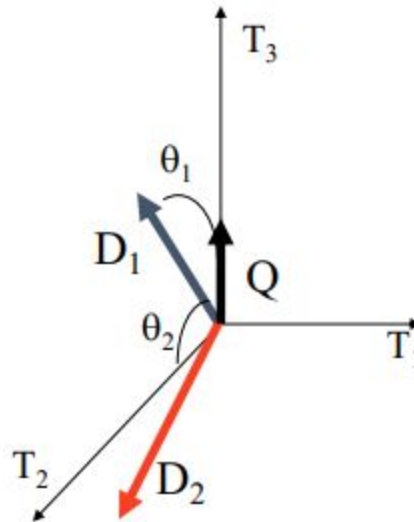
$$Q = (0, 0, 2)$$



3. Kesamaan Cosinus (*Cosine Similarity*)

Kesamaan Cosinus yang biasa dikenal dengan Cosine Similarity merupakan metode yang digunakan untuk mencari relevansi dari dokumen dan query yang dicari pengguna, metode ini menggunakan vektor yang telah disediakan pada bagian 2, ada 2 vektor yang menyatakan jumlah kata pada masing masing dokumen dan query dengan term yang sama (kata yang dibandingkan antara query dan term sama).

Untuk mencari kesamaan antara query dan dokumen, kita bisa melihat sudut antara vektor-vektor tersebut, semakin kecil sudut yang dihasilkan maka semakin mirip dokumen tersebut dengan query.



Jika $\cos \theta$ semakin dekat dengan 1 maka hal itu menandakan Query dan Dokumen itu memiliki vektor yang berhimpitan, yang mengimplikasi bahwa Dokumen tersebut mirip dengan query. Setiap dokumen akan dihitung dan diurutkan sesuai dengan nilai Cosinusnya. Semakin kecil nilai Cosinus yang didapat maka semakin tidak relevan dokumen tersebut.

Nilai Cosine dari query dan dokumen dapat dicari dengan menggunakan rumus

$$\cos \theta = \frac{Q \cdot D}{||Q|| \times ||D||}$$

Dimana \cdot merupakan perkalian dot vektor (Penjumlahan dari perkalian setiap elemen dari kedua vektor) dan $||x||$ melambangkan panjang dari suatu vektor yang didapat dengan mencari akar kuadrat dari penjumlahan setiap elemen yang dikuadratkan pada vektor tersebut.

$$D = [D_1, D_2, D_3, \dots, D_n]$$

$$Q = [Q_1, Q_2, Q_3, \dots, Q_n]$$

Dimana nilai Dot Vektor dituliskan sebagai

$$D \cdot Q = Q_1 \times D_1 + Q_2 \times D_2 + \dots + Q_n \times D_n$$

Dan Panjang vektor dituliskan sebagai

$$||Q|| = \sqrt{Q_1^2 + Q_2^2 + \dots + Q_n^2}$$

BAB III

IMPLEMENTASI PROGRAM

Secara garis besar, program kami terbagi menjadi 2 bagian, yaitu *frontend* dan *backend*. *frontend* berfungsi untuk mengendalikan tampilan situs, sedangkan *backend* berfungsi untuk melakukan pengolahan data. Untuk *backend*, kami menggunakan bantuan *framework* Express, sementara pada Frontend kami menggunakan bantuan *framework* Vue. Pengerjaan tugas ini dilakukan pada git sebagai version control, live share sebagai sarana kolaborasi *coding*, dan discord sebagai sarana komunikasi verbal maupun visual secara *daring*.

1. Backend (Server)

a. Main.js

Main.js merupakan file utama dari server dengan file file lainnya merupakan file bantuan yang berisi fungsi fungsi yang digunakan pada Server. Main.js menerima data dan request dari Frontend yang mana akan diproses sesuai permintaan request.

Pada Program ini kami menggunakan bantuan dari package multer, path, fs, yang mana digunakan untuk memproses upload file, file yang diterima merupakan file dengan extension html dan txt.

b. Parsedoc.js

i. createDoc

Fungsi yang menerima suatu string dan akan mengembalikan menjadi bentuk objek, dengan isi yaitu konten, konten original dan nama file.

ii. parseDoc

Fungsi yang akan melakukan parsing terhadap konten dari suatu dokumen antara HTML atau txt, dimana dokumen tersebut akan diformat, diambil datanya, dan dibuat menjadi objek javascript dengan menggunakan bantuan fungsi **createDoc**.

iii. readAllDoc

Fungsi yang menerima direktori dari folder dan melakukan parsing terhadap semua file txt dan html pada 1 directory sekaligus dan mengembalikan list object.

c. Process.js

i. stemString

Fungsi ini menerima sebuah string dan melakukan stemming dengan bantuan *package* sastrawi, dan akan mengembalikan string dengan elemennya yang sudah dilakukan proses stemming.

ii. removeStopwords

Fungsi ini menerima sebuah string dan melakukan penghapusan stopwords, dan akan mengembalikan string dengan elemennya yang sudah bersih dari stopwords yang ada sesuai dengan list pada **stopwords.js**.

iii. cleanString

Fungsi ini merupakan fungsi gabungan dari **stemString** dan **removeStopwords**, dimana fungsi akan menerima sebuah string dan akan mengembalikan lagi string dengan elemen yang sudah di stem dan stopwords yang sudah dihapus.

iv. removeEscapeChr

Fungsi ini merupakan fungsi untuk menerima sebuah string dan menghapus karakter karakter spesial pada string tersebut, contohnya (\n, \s, \t, \r). Dan akan mengembalikan string yang sudah dibersihkan dari karakter spesialnya.

v. addToTermDict

Fungsi ini menerima sebuah string dan akan memasukan setiap elemen yang dipisahkan oleh spasi (kata) dan memasukkan terhadap **termDictionary** yang tersedia, ketika ada kata yang sudah ada pada **termDictionary**, maka tidak akan dimasukkan lagi.

vi. toObj

Fungsi ini menerima query dan akan dikembalikan lagi sebagai objek, yang berisi kemunculan kata pada masing masing query.

vii. createDocQueryObj

Fungsi ini menerima dokumen yang akan dikembalikan menjadi objek yang berisi kemunculan kata pada dokumen, dan kata acuan yang digunakan merupakan kata kata pada query.

viii. sortSimilaritiesDsc

Fungsi ini digunakan untuk melakukan sorting terhadap suatu array Objek. Digunakan untuk mengurutkan hasil pencarian agar sesuai dengan nilai *Cosine Similarity* dari setiap dokumennya.

ix. makeFirstSentence

Fungsi ini digunakan untuk memunculkan kalimat pertama pada dokumen untuk diperlihatkan pada search engine, dengan syarat jika belum menemukan titik sampai 100 karakter, maka akan diambil hanya 100 karakter saja.

x. countDotProd

Fungsi ini menerima 2 buah vektor dan akan menghasilkan sebuah angka yang merupakan nilai dot product dari kedua vektor tersebut.

xi. cosineSim

Fungsi ini menerima nilai query dan dokumen dalam bentuk vektor dan akan mencari nilai *Cosine Similarity* dengan menggunakan operasi vektor dari kedua vektor tersebut, dan akan mengembalikan sebuah angka yang merupakan nilai Cos dari selisih sudut kedua vektor tersebut yang bernilai 0-1.

xii. toVector

Fungsi ini menerima objek dan akan mengembalikan sebuah vektor dari objek tersebut, dimana vektor tersebut merupakan sebuah list dengan elemen berupa kemunculan kata dari query maupun dokumen.

xiii. containsFile

Fungsi ini mencari apakah file yang akan diupload pengguna sudah ada pada list atau belum, jika belum maka dokumen tersebut akan dimasukan kedalam list Global, jika sudah maka dokumen tersebut tidak akan dimasukkan lagi, sehingga tidak akan mengalami pengulangan hasil pencarian.

xiv. testProcess

Fungsi proses yang menguji algoritma pencarian saat development, menerima query dari *frontend* dan mengembalikan object query kembali ke *frontend*.

xv. mainProcess

Fungsi proses algoritma pencarian utama dengan menerima query dari frontend dan mengembalikan object query kembali ke frontend.

d. Scraper.js

i. extractHTML

Menerima 3 parameter berupa *url*, *filter*, dan *filtermethod* untuk menentukan scraping apa yang diinginkan. Parameter *url* mengambil HTML dari url yang diberikan, parameter *filtermethod* menentukan jenis tag apa yang ingin di filter, dan filter untuk menentukan detail dari tag tersebut.

e. stopWord.js

Berisi list *stopword* dalam bahasa indonesia yang digunakan pada proses pembersihan query dan dokumen.

2. Frontend

a. App.vue

Merupakan home dan pemersatu dari keempat file vue lainnya, digunakan untuk menampung komponen vue dari search, scrape dan About Us.

b. Search.vue

Berisi tentang tampilan untuk menampilkan hasil pencarian berdasarkan urutan nilai *Cosine Similarity*, dengan setiap dokumen yang dapat diklik untuk melihat isi konten original dari file tersebut, serta juga tabel yang berisi kemunculan term pada setiap dokumen.

c. uploadedFile.vue

Berisi mengenai tampilan untuk mengunggah file, menampilkan file dan memperlihatkan file yang tidak sesuai dengan syarat syarat tertentu, dan juga dapat menghapus file yang tidak jadi diunggah.

d. webSraper.vue

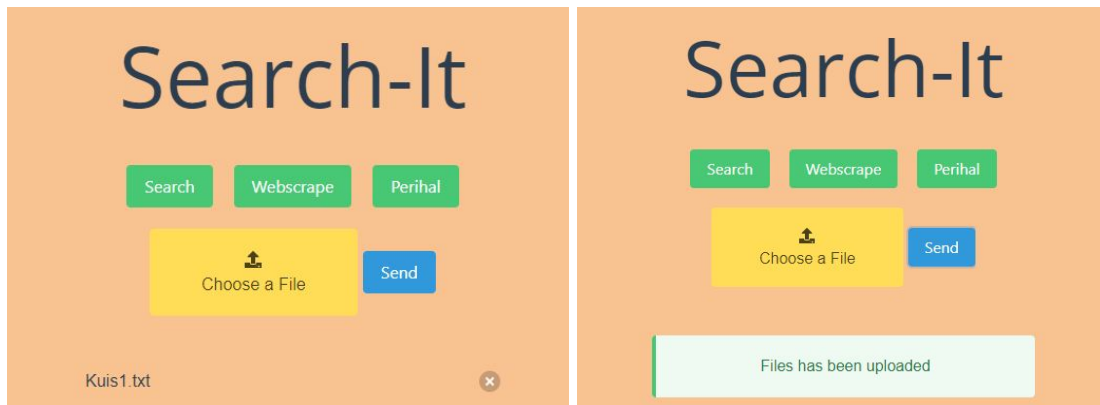
Berisi tampilan untuk menampilkan pilihan url, metode, dan filter untuk *web scraping*, kemudian menampilkan hasil tulisan dari *web scraping* tersebut.

BAB IV

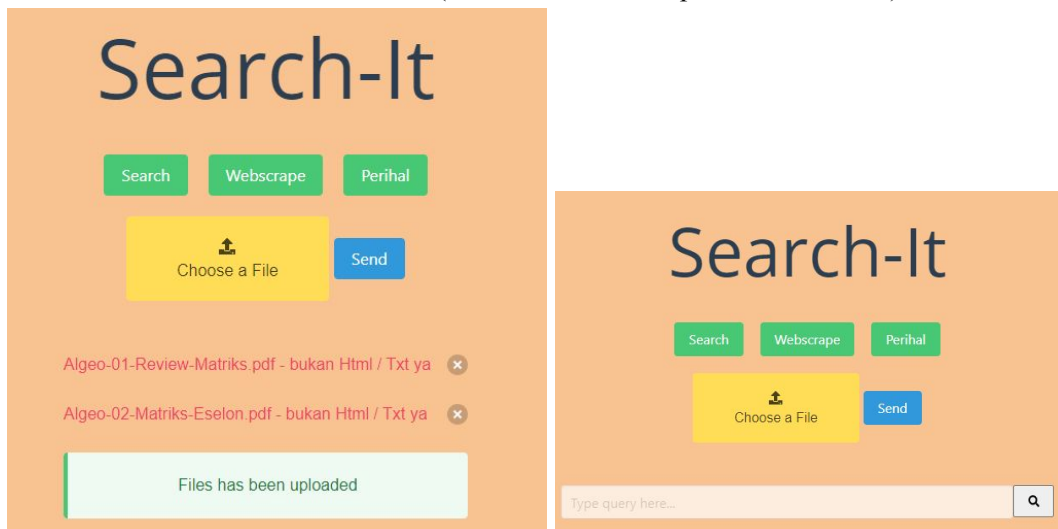
EKSPERIMEN

1. Mengupload File

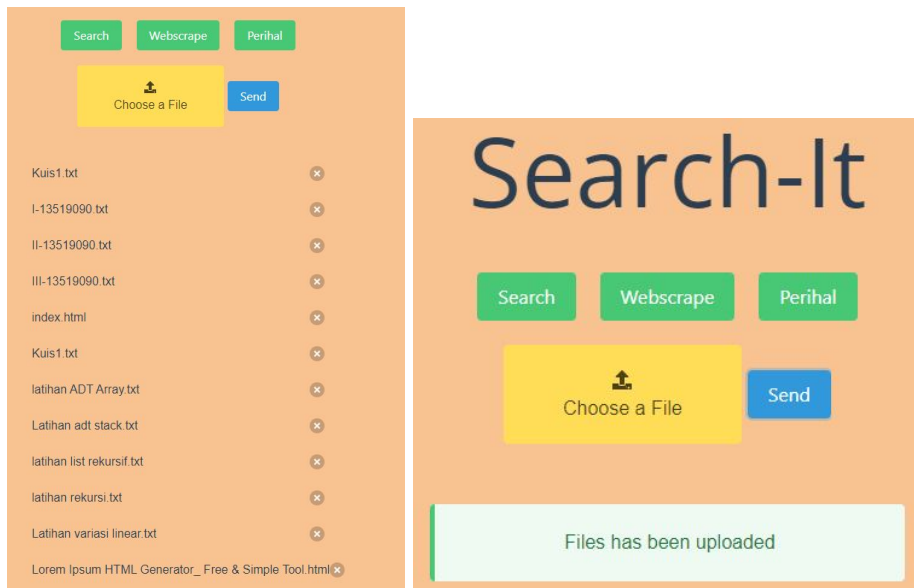
Percobaan 1: Upload 1 file



Percobaan 2: Penolakan Format File (File tidak akan ter-upload ke *backend*)

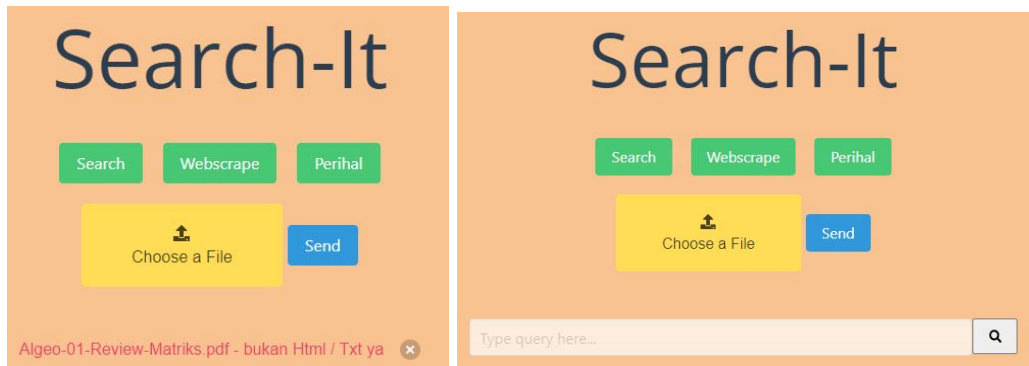


Percobaan 3: Upload file campuran(html dan txt)

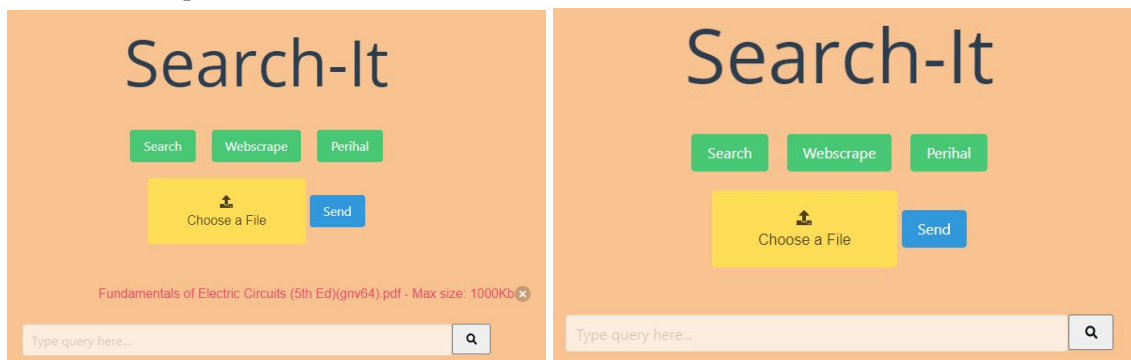


2. Menghapus File, Ada Dokumen yang tidak memenuhi syarat

Percobaan 1: Upload file pdf

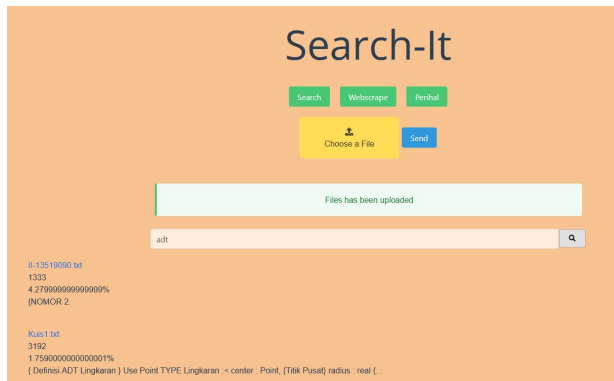


Percobaan 2: Upload file lebih dari 1MB



3. Melakukan Search

Percobaan 1: Search query dengan 1 kata



Percobaan 2: Search query dengan 4 kata



4. Membuka Dokumen hasil Pencarian

Percobaan 1: Membuka file txt



Percobaan 2: Membuka file html



5. Tabel Kata

Percobaan 1: Tabel hasil search query dengan 1 kata

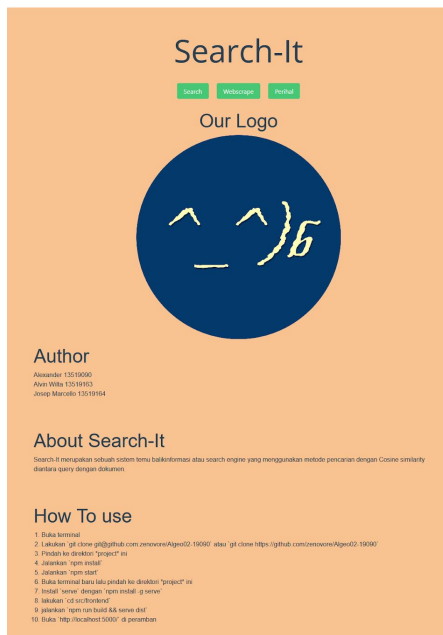
Term	Query	II-13519090.txt	Kuis1.txt	latihan ADT Array.txt	index.html	I-13519090.txt	III-13519090.txt	Latihan adt stack.txt	latihan list rekursif.txt	latihan rekursi.txt	Latihan variasi linear.txt	Lorem Ipsum HTML Generator_ Free & Simple Tool.html
adt	1	1	1	0	0	0	0	0	0	0	0	0

Percobaan 2: Tabel hasil search query dengan 4 kata

Term	Query	I-13519090.txt	Kuis1.txt	latihan ADT Array.txt	latihan list rekursif.txt	II-13519090.txt	Latihan adt stack.txt	Latihan variasi linear.txt	index.html	latihan rekursi.txt	Lorem Ipsum HTML Generator_ Free & Simple Tool.html	III-13519090.txt
matriks	1	4	0	0	0	0	0	0	0	0	0	0
segitiga	2	1	0	0	0	0	0	0	0	0	0	0
bawah	1	1	0	0	0	0	1	0	0	0	0	0
true	1	2	9	9	4	1	0	1	0	0	0	0

6. Melihat Perihal (About Us)

Percobaan 1: Membuka halaman About Us dari navboard

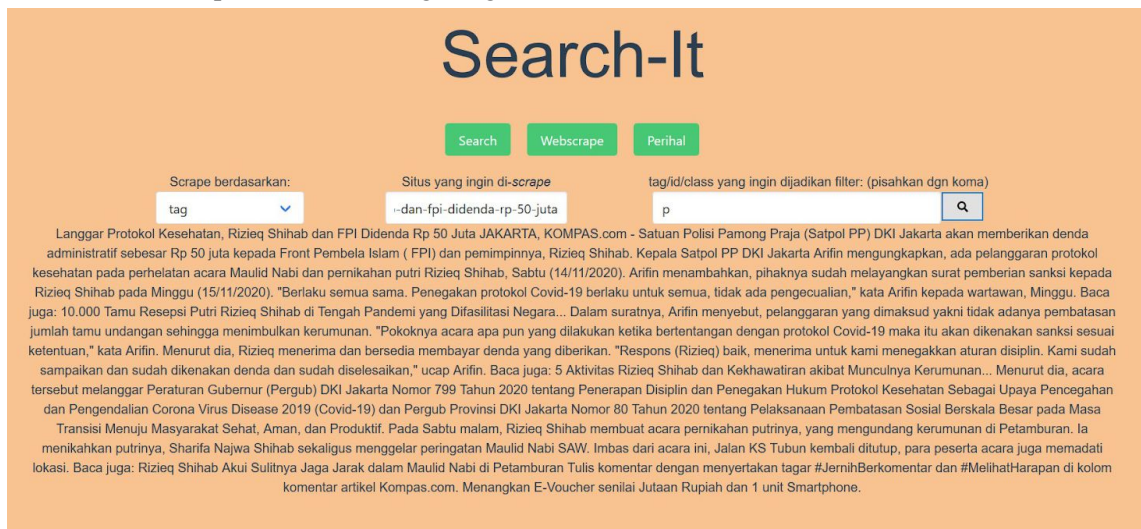


7. Web Scraper

Percobaan 1: Scrape berdasarkan class



Percobaan 2: Scrape berdasarkan tag dengan filter



Percobaan 3: Scrape berdasarkan id dengan filter

Search-It

Search

Webscrape

Perihal

Scrape berdasarkan:

Situs yang ingin di-*scrape*

tag/id/class yang ingin dijadikan filter: (pisahkan dgn koma)

id

3m/jspmarc/Algeo01-19116/

readme

README.md Tubes Pertama IF2123 Aljabar Linier dan Geometri Algeo01-19116 Kelompok 59 Nama NIM Jeane Mikha Erwansyah 13519116 Josep Marcello 13519164 David Owen Adiwiguna 13519169 SOP Pembagian tugas NIM Tugas 13519116 Metode Gauss, Gauss-Jordan, Pembuat program utama 13519164 Determinan matriks metode ekspansi kofaktor dan reduksi baris, interpolasi polinom 13519169 Matriks balikan, Kaidah cramer, regresi linier berganda Spesifikasi Tugas Tugas Besar 1 IF 2123 Aljabar Linier dan Geometri Menghitung solusi SPL dengan metode eliminasi Gauss, Gauss-Jordan, matriks balikan, dan kaidah Cramer Kaidah Cramer khusus untuk SPL dengan \$n\$ peubah dan \$n\$ variabel Menyelesaikan persoalan interpolasi polinom dan regresi linier berganda Menghitung matriks balikan Menghitung determinan matriks dengan metode reduksi baris dan ekspansi kofaktor Petunjuk Instalasi Jalankan perintah make jar. Tidak punya make? Jalankan: git clone https://github.com/jspmarc/Algeo01-19116 javac -g -d bin src/Matriks.java src/Main.java jar -cvfe bin/tubes.jar Main -C bin/Matriks.class -C bin/Main.class Petunjuk Penggunaan jalankan perintah make Tidak punya make? Pastikan Anda sudah melalui proses instalasi java -jar bin/tubes.jar

BAB V

KESIMPULAN DAN SARAN

1. Kesimpulan

Pembuatan *search engine* berbasis web dengan memanfaatkan penggunaan *cosine similarity* sebagai aplikasi dari penggunaan vektor memiliki komponen yang berbeda-beda. Pengerjaan *search engine* tersebut dibagi menjadi bagian *frontend* dan *backend*, serta dibantu dengan menggunakan 2 *framework* utama yaitu *express* untuk *frontend* dan *vue* untuk *backend*. Pada bagian *frontend*, terdapat fitur untuk mengupload file, mencari kemunculan kata pada dokumen yang diupload, menampilkan tabel term, dan menampilkan halaman tentang kami. Pada bagian *backend* memuat fitur parsing, *stemming*, *stopwords*, *vectoring* dokumen, penghitungan *cosine similarity*, pengurutan kemunculan dokumen, dan pengolahan data pada tabel.

Search engine ini dibuat oleh kelompok penulis dengan menggunakan LINE dan Discord sebagai media komunikasi verbal dan visual, git sebagai *version control*, dan live share sebagai media kolaborasi. Pengerjaan tugas ini memberikan pengalaman kepada setiap anggota kelompok kami dalam pengembangan website yang fungsional dan dilengkapi dengan fitur *search engine*, *file uploading*, *term counting*, dan *web scraping*.

2. Saran

Berdasarkan proses pengerjaan yang telah kami lalui, berikut ini adalah saran dari kami untuk pengerjaan berikutnya.

1. Memahami dengan jelas apa yang ditangkap/dikeluarkan dari *frontend* maupun *backend*.
2. Baca Spesifikasi dengan jelas dan bertanya jika kurang jelas.
3. Mencari referensi menggunakan *package* ataupun *framework* yang bisa mempermudah pekerjaan dan mempelajarinya dari awal.

3. Refleksi

- 1) Mengerjakan tabel merupakan pekerjaan sulit yang tertunda. Sehingga melakukan pekerjaan dekat dengan deadline kurang baik.
- 2) Penggunaan beberapa laptop tertentu dapat mempengaruhi server yang di run maupun di start, pada beberapa laptop/komputer run bisa dibilang memakan

waktu yang cukup lama dan tidak dapat auto-update seiring dokumennya berubah, namun di beberapa laptop/komputer lain waktu berjalannya sangat cepat.

- 3) Browser yang memiliki keberagaman, program dapat dijalankan dengan lancar di salah satu *browser*, namun tidak lancar pada *browser* lain.
- 4) Memperhatikan struktur dari data agar tidak mempersulit dalam pengaplikasiannya.
- 5) Selalu testing setelah mengerjakan satu bagian agar tidak berantakan saat di akhir.

REFERENSI

Putung, K. D., Lumenta, A. S. M., & Jacobus, A. (2016). Penerapan Sistem Temu Kembali Informasi Pada Kumpulan Dokumen Skripsi. Jurnal Teknik Informatika, 8(1).

<https://doi.org/10.35793/jti.8.1.2016.12227>

“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>

Robbani, Hanif Amal. "Sastrawi". Pypi, 2020, <https://pypi.org/project/Sastrawi/>. Accessed 16 Nov 2020.

You, Evan. "Introduction — Vue.js". Vuejs.Org, 2020, <https://vuejs.org/v2/guide/>. Accessed 16 Nov 2020.