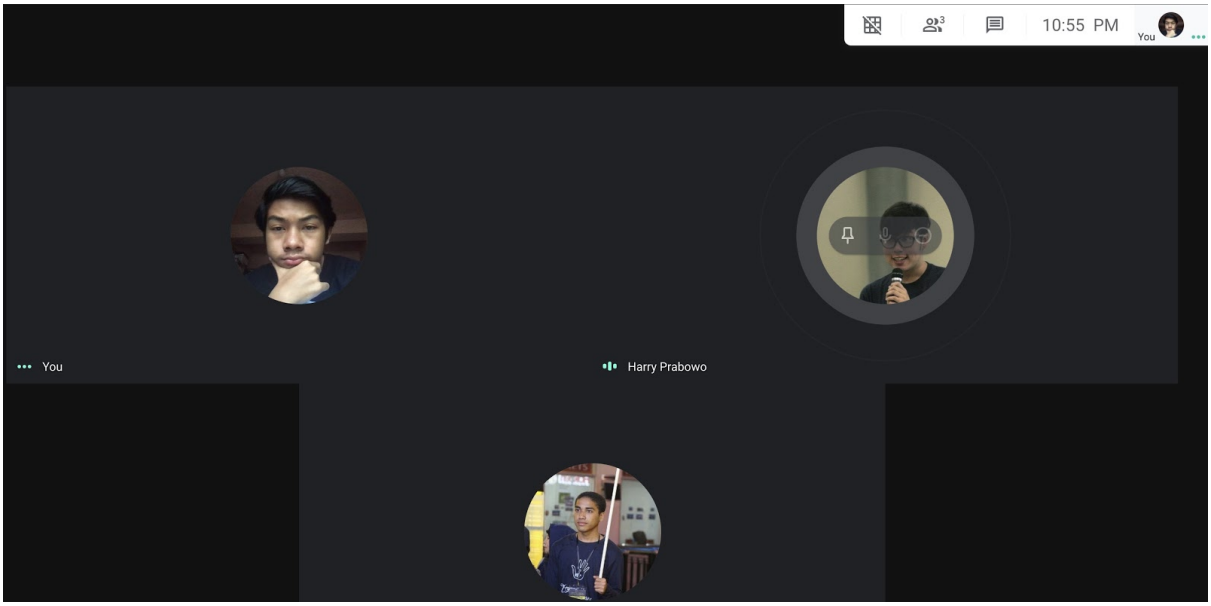
LAPORAN TUGAS BESAR 2 IF2123 ALJABAR LINIER DAN GEOMETRI
APLIKASI DOT PRODUCT PADA SISTEM TEMU BALIK INFORMASI
SEMESTER I TAHUN 2020/2021

Anggota Kelompok :

Harry Prabowo (13517094)
Alvin Rizqi Alfisyahrin (13519126)
La Ode Rajuh Emoko (13519170)



PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

2020

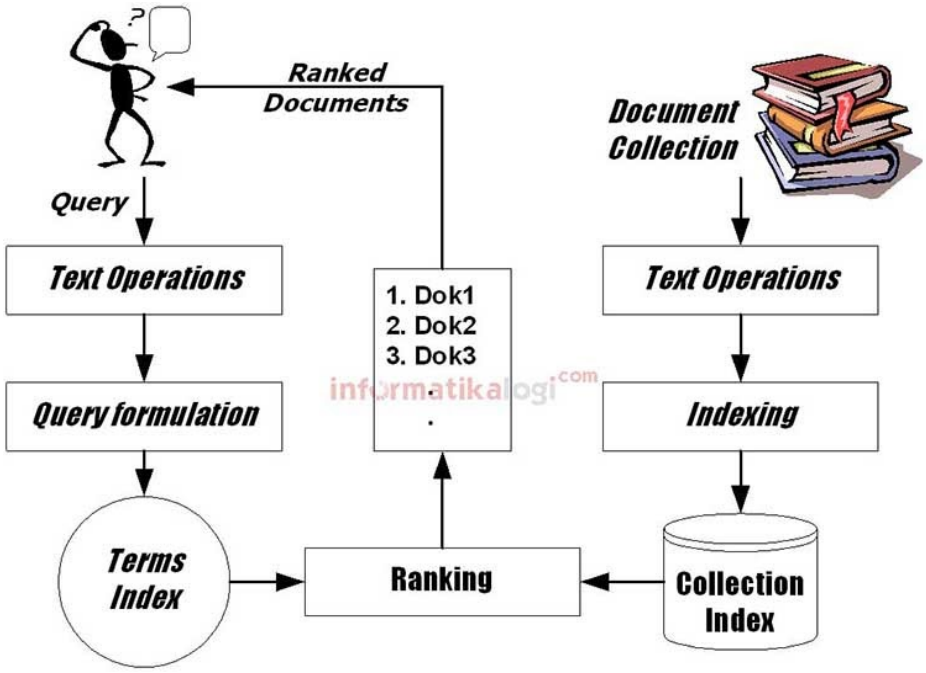
BAB 1 DESKRIPSI MASALAH

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 2 TEORI SINGKAT

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. Cara kerja Sistem Temu-Balik pada mesin pencarian

Ide utama dari sistem temu balik informasi adalah mengubah search query menjadi ruang vektor Setiap dokumen maupun query dinyatakan sebagai vektor w = (w1, w2,..., wn) di dalam Rn , dimana nilai wi 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:

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

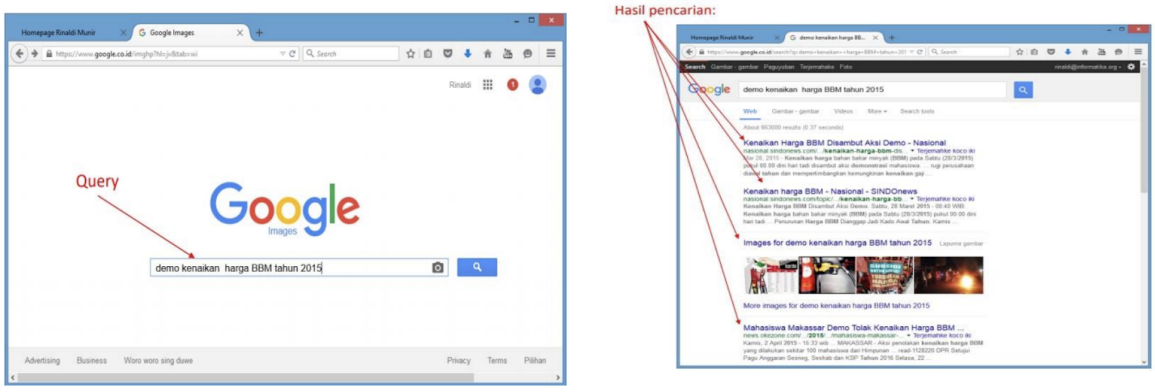
dengan **Q D** adalah perkalian titik yang didefinisikan sebagai

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

Rinaldi Munir - IF2123 Aljabar Geometri 12

Gambar 2. Persamaan *cosine similarity*

Jika cos theta = 1, berarti theta = 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. Setiap dokumen di dalam koleksi dokumen dihitung kesamaannya dengan query dengan rumus cosinus di atas. Selanjutnya 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. Berikut contoh aplikasi dari sistem temu balik informasi



Gambar 3. *Search engine Google*

BAB 3 IMPLEMENTASI PROGRAM

Pertama, ada bagian *pre-processing*, pada bagian ini kelompok kami melakukan perubahan huruf menjadi *lowercase*, menghapus angka, menghapus *whitespace*, menghapus tanda baca, menghapus *stopwords,* dan *stemming*. Setelah itu, semua kata dari semua dokumen diubah menjadi token. Setelah itu semua kata diubah menjadi *dictionary* dengan key nya adalah kata alias token, dan valuenya adalah weight dari kata tersebut. *Dictionary* dibuat per dokumen, lalu digabung dalam list of dokumen. Proses menentukan weight dari setiap kata ditentukan menggunakan metode tf*idf, dimana tf merupakan *term frequency* dan idf merupakan *inverse document frequency.*

$$\text{tf} = 1 + \log f_{t,d}$$

Term frequency diatas merupakan hasil yang sudah dinormalisasi, disini f(t,d) adalah jumlah kemunculan kata dalam suatu dokumen. Dilakukan normalisasi agar jika ada jumlah kata yang tidak seimbang, maka nilai kata lain tidak tertutupi. Contoh : kata “the” muncul 1000 kali, sedangkan kata “research” muncul hanya 15 kali, padahal lebih penting kata research, normalisasi ini dilakukan untuk menghindari hal tersebut.

$$\text{idf} = \log \frac{N}{n_t}$$

Inverse document frequency merupakan jumlah kemunculan kata pada tiap dokumen, pada persamaan ini dilambangkan sebagai nt, sedangkan N adalah jumlah dokumen tersebut.

Kedua frekuensi ini dikalikan sehingga dihasilkan *weight* dari setiap *terms*. Semakin tinggi nilai *weight* maka terms tersebut semakin penting.

Untuk programnya sendiri, peserta membuat 3 class, yaitu words yang berarti kata dari seluruh dokumen, lalu document untuk mengakses 1 dokumen dari list of dokumen, dan documents untuk mengakses seluruh dokumen. Lalu, peserta juga membuat 3 fungsi utama pada preprocessing, yaitu *preprocess* yang berperan sebagai penghapusan hal-hal yang tidak dibutuhkan pada dokumen, lalu *weight* untuk menghitung *weight* dari setiap terms, dan yang terakhir adalah *make_dictionary* untuk membuat list yang berisi dictionary dari masing-masing dokumen. *Dictionary* ini memiliki nilai *key* semua kata yang ada, dengan nilai *value* adalah weight dari masing-masing kata di dokumen tersebut. Dengan kata lain, *dictionary* ini merupakan vektor dari dokumen yang siap diproses menggunakan *cosine similarity.*

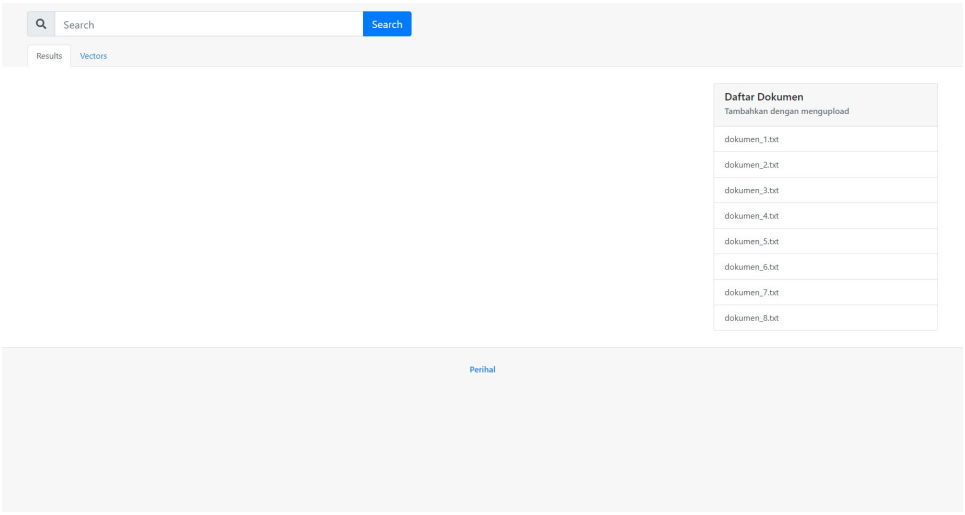
Pada perhitungan cosine similarity dibuat fungsi “panjang” untuk mencari panjang vektor, juga fungsi “cosine_sim” untuk mencari *similarity* dari masing masing vektor dengan vektor query. Semua hasil dari perhitungan dimasukkan ke *dictionary* dengan nilai *key* adalah nama file tersebut di list of documents yang kita punya, dan nilai *value* adalah similaritas dokumen dengan query. Setelah hasil *dictionary*-nya berhasil dibuat, dictionary

tersebut diurutkan sesuai nilai similaritasnya mulai dari yang paling besar sampai yang paling kecil.

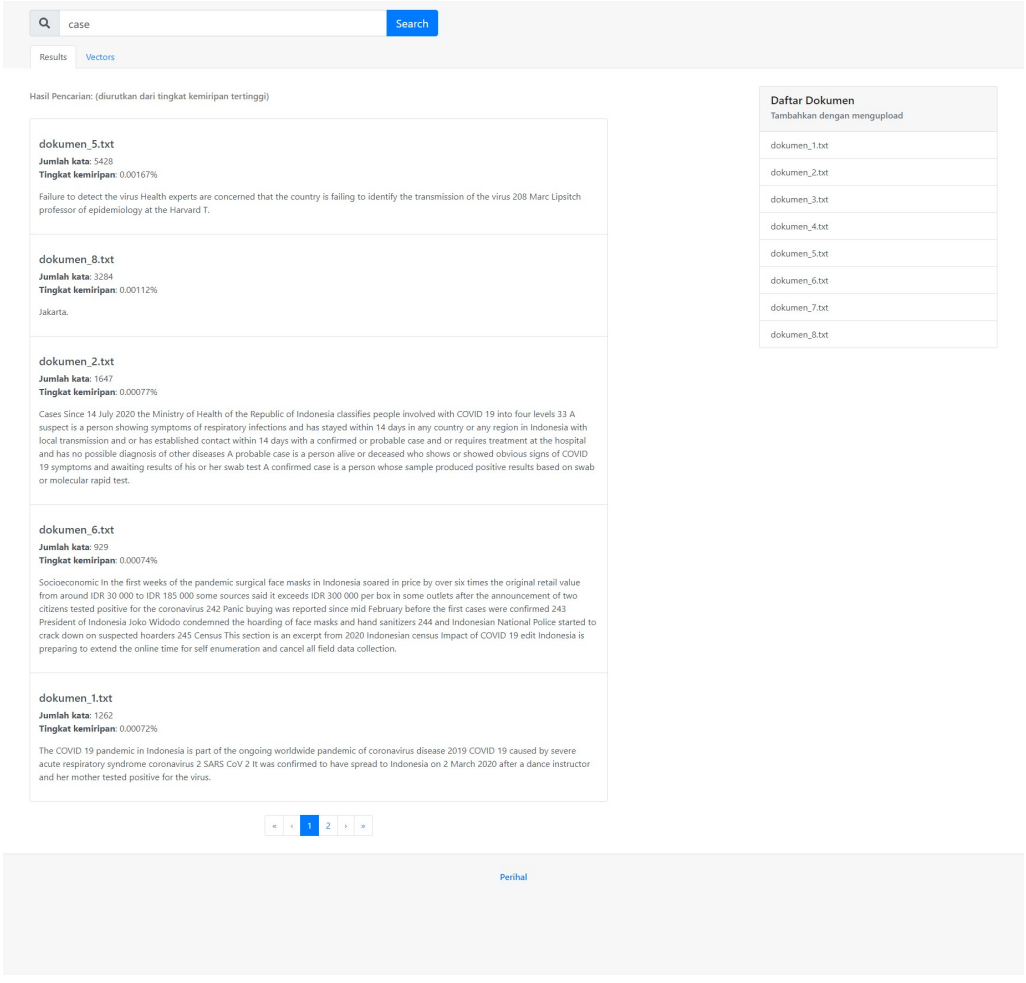
Proses yang sudah dijelaskan sebelumnya disebut *backend* atau proses dibelakang web yang tidak ditampilkan dalam web. Proses selanjutnya adalah *frontend* yang berarti program untuk menampilkan proses-proses yang sudah dilakukan di *backend*. Pada *frontend* kami menggunakan React sebagai framework. Lalu, ada file yarn.lock dan package.json di frontend sebagai basic approach, pada file public berisi desain dari web seperti logo, pada file src terdapat code yang membangun website, dimulai dari controller, helper, maupun pages.

BAB 4 EKSPERIMEN

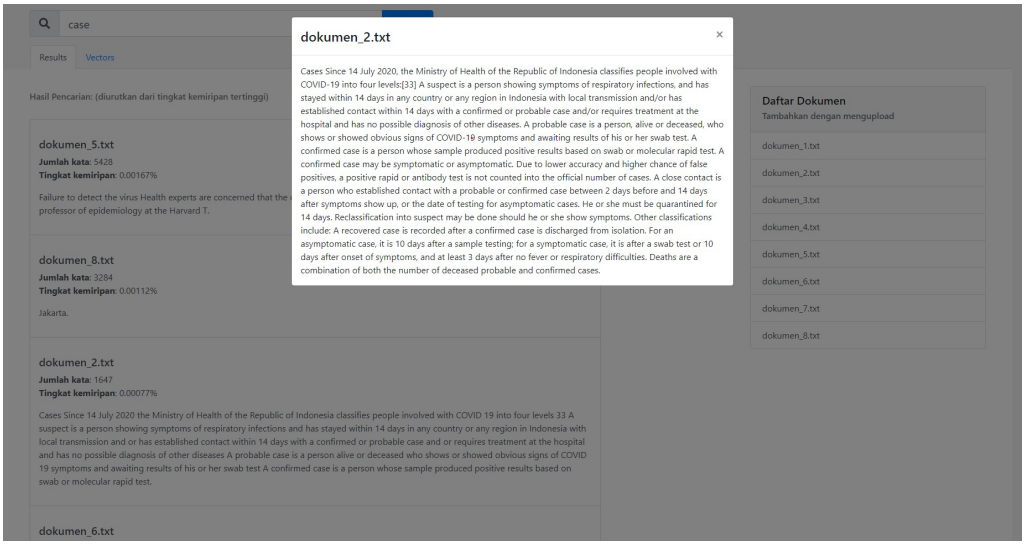
1. Tampilan Awal



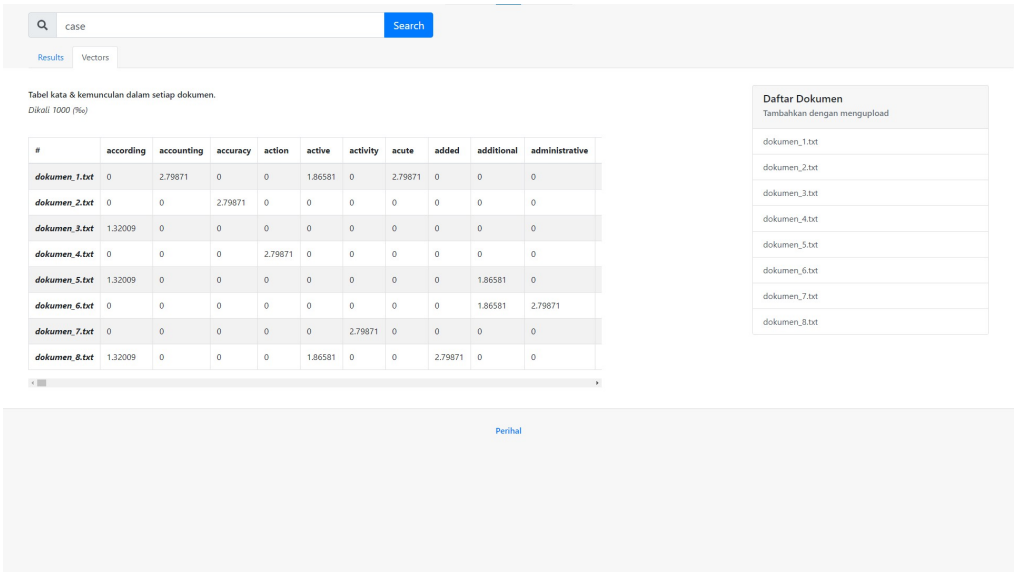
2. Test Pencarian



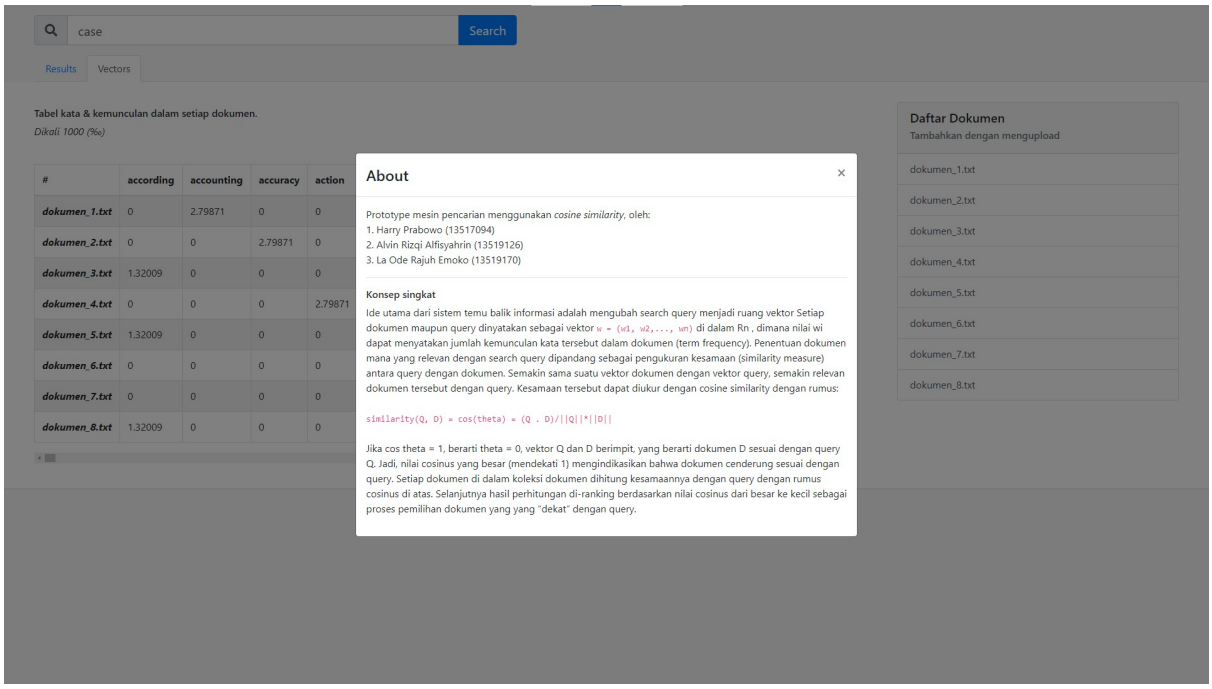
3. Buka File



4. Melihat Vektor



5. Perihal


case
Search
Results
Vectors
Tabel kata & kemunculan dalam setiap dokumen.
Dikali 1000 (‰)
according accounting accuracy action
dokumen_1.txt 0 2.79871 0 0
dokumen_2.txt 0 0 2.79871 0
dokumen_3.txt 1.32009 0 0 0
dokumen_4.txt 0 0 0 2.79871
dokumen_5.txt 1.32009 0 0 0
dokumen_6.txt 0 0 0 0
dokumen_7.txt 0 0 0 0
dokumen_8.txt 1.32009 0 0 0
About
Prototype mesin pencarian menggunakan cosine similarity, oleh:
1. Harry Prabowo (13517094)
2. Alvin Rizqi Alfisyahrin (13519126)
3. La Ode Rajuh Emoko (13519170)
Konsep singkat
Ide utama dari sistem temu balik informasi adalah mengubah search query menjadi ruang vektor Setiap dokumen maupun query dinyatakan sebagai vektor w = (w1, w2,..., wn) di dalam Rn , dimana nilai wi 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:
similarity(Q, D) = cos(theta) = (Q . D)/||Q||*||D||
Jika cos theta = 1, berarti theta = 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. Setiap dokumen di dalam koleksi dokumen dihitung kesamaannya dengan query dengan rumus cosinus di atas. Selanjutnya hasil perhitungan di-ranking berdasarkan nilai cosinus dari besar ke kecil sebagai proses pemilihan dokumen yang yang "dekat" dengan query.
Daftar Dokumen
Tambahkan dengan mengupload
dokumen_1.txt
dokumen_2.txt
dokumen_3.txt
dokumen_4.txt
dokumen_5.txt
dokumen_6.txt
dokumen_7.txt
dokumen_8.txt

BAB 5 KESIMPULAN DAN SARAN

Kesimpulan

Pada Tugas Besar kali ini, peserta kelas Aljabar Linier dan Geometri 2020/2021 sudah menyelesaikan pengerjaan program mengikuti spesifikasi. Peserta juga bisa mengimplementasikan ilmu yang diajarkan di kelas menjadi sebuah *code*. Selain itu, peserta juga sudah membagi tugas dan memahami hasil code masing-masing agar lebih paham.

Saran

Saran untuk peserta kedepannya agar belajar lebih dalam tentang text processing, juga tentang frontend dan backend dalam membuat web.

Saran untuk tubes, mungkin kedepannya bisa diadakan tutorial membuat web dengan front end dan backend-nya, agar peserta dapat memahami dan tidak terlalu kesulitan dalam membuatnya. Hal ini bisa dilakukan karena fokus dari tubes ini adalah di pemanfaatan vektor dalam information retrieval, bukan pada frontend websitenya, sekaligus agar peserta dapat mempelajarinya.

BAB 6 REFERENSI

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

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