TUGAS BESAR II

MESIN PENCARI

IF2123 – Aljabar Linear dan Geometri

K - 01, K - 03, K - 04

Oleh

Kelompok 14 :

Ariya Adinatha (13519048) K04

Wisnu Aditya Samiadji (13519093) K01

Hokki Suwanda (13519143) K03

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2020

BAB I

DESKRIPSI MASALAH

1.1 Spesifikasi Tugas

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.
7. Program harus modular dan mengandung komentar yang jelas.
8. Dilarang menggunakan *library cosine similarity* yang sudah jadi.

BAB II

LANDASAN TEORI

1. Information Retrieval

Information retrieval adalah menemukan kembali informasi yang relevan terhadap kebutuhan pengguna dari suatu kumpulan informasi secara otomatis. *Information retrieval* dapat dimodelkan dengan ruang vektor. Model ini menggunakan teori di dalam aljabar vektor. Misalkan terdapat n kata berbeda sebagai kamus kata atau indeks kata. Kata-kata tersebut membentuk ruang vektor berdimensi n. Setiap dokumen maupun query dinyatakan sebagai vektor $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ di dalam $\mathbf{R}^n$. Dengan w_i menyatakan jumlah kemunculan kata ke-*i* dalam dokumen.

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 antara dua vektor $\mathbf{Q} = (q_1, q_2, \ldots, q_n)$ dan $\mathbf{D} = (d_1, d_2, \ldots, d_n)$ diukur dengan :

$$sim(\boldsymbol{Q}, \boldsymbol{D}) = \boldsymbol{cos\theta} = \frac{Q.D}{||\boldsymbol{Q}||.||\boldsymbol{D}||}$$

Semakin besar nilai cosinus tersebut, semakin sesuai sebuah dokumen dengan *query*.

BAB III

IMPLEMENTASI

3.1 Struktur *modul*

Untuk pemrosesan *file*, kelompok penulis menggunakan bahasa pemrograman python. Terdapat empat buah modul yang digunakan

1. Modul *tokenize* (menggunakan nltk)
   Modul ini memisahkan tiap kata dan tanda baca dan menganggap hasil pemisahannya sebagai sebuah *list*.

2. Modul *similarity*
   Modul ini menghitung nilai cosinus *similarity*. Modul ini menerima *input query* dan namafile. Awalnya, melakukan *tokenize* pada *query* dan *file*. Kemudian menghilangkan *stopwords* dan melakukan *stemming* pada tiap *query* dan *file*. *Dot product* dihitung dengan menjumlahkan perkalian frekuensi kata yang ada di *query* dengan frekuensi kata tersebut di dokumen. Panjang dihitung dengan menjumlahkan kuadrat dari frekuensi tiap kata di dalam himpunan kata. Panjang dihitung untuk dokumen dan juga *query*. Kemudian mengalikan panjang kedua vektor yang menghasilkan nilai panjang kuadrat. Nilai cosinusnya adalah hasil *dot product* dibagi dengan akar dari panjang kuadrat tadi. Outputnya adalah nilai cosinus dengan dua angka di belakang koma.

3. Modul *kalimatPertama*
   Secara umum, modul ini memiliki *output* kalimat pertama di setiap *file* sebelum *stopwords* dihilangkan dan *stemming* dilakukan. Modul ini menerima *input* namafile, kemudian mencari indeks kemunculan pertama karakter titik (‘.’) di dalamnya, misalnya *i*. *Output* dari modul ini adalah *string* dari indeks 0 hingga indeks (*i* - 1).

4. Modul *hitungKata*
   Modul ini memiliki *output* jumlah kata (sesudah disaring) pada sebuah *file*. Modul ini menerima *input* sebuah namafile, melakukan *tokenize*, menghilangkan *stopwords* pada hasil *tokenize*, kemudian menghasilkan panjang *list* setelah *stopwords* dihilangkan.

5. Modul *countFrek*
   Modul ini menerima *input* berupa *query* dan namafile. Modul ini pada awalnya melakukan *tokenize* pada *query* dan *file* dan menghilangkan *stopwords* pada *query* dan file. Lalu melakukan *stemming* pada setiap kata di *query* dan file. Kemudian meng-*output list* semua katanya.

3.2 Struktur website lokal

Fitur yang tersedia di website lokal adalah :

1. Search
   Seperti search engine pada umumnya, fitur ini menerima masukan sebuah query. Jika tombol enter atau *submit* ditekan, pengguna akan diarahkan ke halaman hasil pencarian yang dokumennya diurutkan berdasarkan yang paling mirip hingga ke yang paling tidak mirip. Hasil pencarian menampilkan judul dokumen, persentase kemiripan, jumlah kata di dalam dokumen, kalimat pertama pada dokumen. Fitur ini juga menampilkan tabel daftar kata dan frekuensinya. Kata yang ditampilkan tidak mencakup *stopwords* dan sudah di-*stem*.

2. Upload File
   Tidak seperti search engine pada umumnya, fitur ini berfungsi untuk menambahkan dokumen yang akan dicari isinya. Fitur ini memungkinkan pengguna untuk memasukkan format file apapun, tetapi file yang akan digunakan pada website lokal adalah file berformat *.txt sehingga file yang tidak berformat *.txt akan diabaikan oleh mesin pencari.

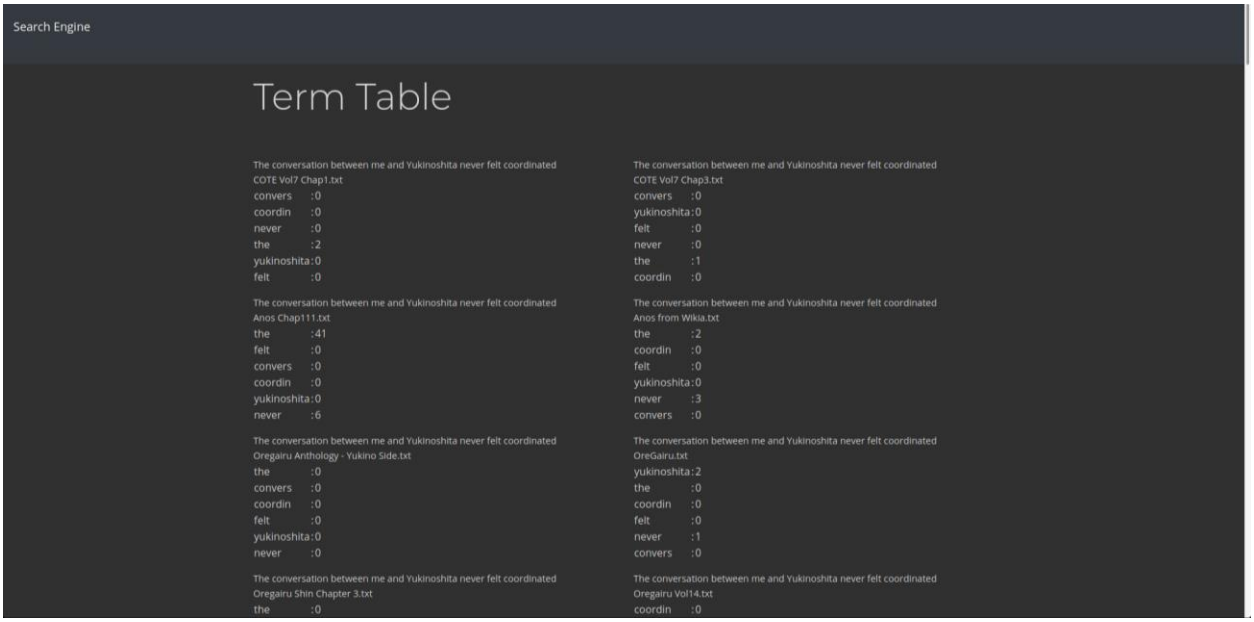
3. About Us
   Fitur ini akan mengarahkan pengunjung ke laman yang berisi profil pembuat yang terdiri dari tiga orang, dengan anggota seperti yang tertera pada halaman sampul laporan ini

BAB IV

EKSPERIMEN

4.1 Hasil Pengujian

Untuk pengujian, dokumen yang digunakan adalah dokumen *light novel* dari Jepang dalam bahasa Inggris.



BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

Mesin pencari merupakan salah satu aplikasi dari ruang vektor yang menggunakan nilai kemiripan dalam bentuk cosinus. Nilai kemiripan diukur antara vektor *query* dengan vektor *dokumen* yang komponen tiap vektornya berarti frekuensi munculnya suatu kata dasar. Semakin besar nilai cosinusnya, semakin mirip suatu dokumen dengan permintaan dari pengguna (*query*).

5.2 Refleksi

- Tugas ini cukup susah karena perlu eksplorasi dan belajar *from scratch*. Tetapi jika sudah menguasai backend engineering, tugas ini tidak begitu sulit.
- Sempat terjadi perubahan penggunaan bahasa karena kurangnya pemahaman terhadap spesifikasi tugas
- Ternyata menggunakan PHP lebih sulit daripada menggunakan JavaScript

5.3 Saran

- Untuk kelompok penulis, sebaiknya memahami spesifikasi dengan lebih cermat
- Mungkin memberi tugas untuk eksplorasi adalah hal yang baik, tetapi rasanya agak sulit dilakukan karena banyaknya tugas besar dari mata kuliah lain yang mengikuti.

DAFTAR PUSTAKA

IF2123 Aljabar Geometri - Semester I Tahun 2020/2021 diakses pada 11 November 2020.

https://pythonprogramming.net/stemming-nltk-tutorial/ diakses pada 13 November 2020

Removing stop words with NLTK in Python diakses pada 14 November 2020

https://www.nltk.org/book/ch01.html diakses pada 14 November 2020