

**Tugas Besar 2 IF 2123 Aljabar Linier dan Geometri
Semester I Tahun 2020/2021**

**Chika Engine
Aplikasi *Dot Product* pada Sistem Temu-Balik
Informasi**



13519019 Jason Stanley Yoman

13519083 Shaffira Alya Mevia

13519118 Cynthia Rusadi

**Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2020**

Bab I

Deskripsi Masalah

Shuchi'in Academy adalah sekolah yang sangat berprestasi dan bergengsi yang terletak di Tokyo, Jepang. Murid-murid dari sekolah tersebut berada di antara kalangan atas atau sangat berprestasi. Seperti contohnya adalah Shirogane Miyuki, berada di kalangan menengah ke bawah tetapi sangat berprestasi, dan Kaguya Shinomiya, berada di kalangan atas dan juga sangat berprestasi. Keduanya mempunyai jabatan di osis Shuchi'in Academy, dengan Shirogane Miyuki menjabat sebagai Ketua Osis dan Kaguya Shinomiya menjabat sebagai Wakil Ketua Osis. Selain itu, ada beberapa anggota osis juga, yaitu Chika Fujiwara, sebagai Sekretaris, Ishigami Yuu, sebagai Bendahara, dan Miiko Iino, sebagai Auditor.

Pada suatu hari, Chika ingin melakukan riset mengenai psikologi dan *search engine* yang biasa digunakan, yaitu G^ogle, sedang mengalami *maintenance* pada saat itu. Chika merasa sedih karena pada saat itu, Chika ingin melakukan riset segera untuk diberitahukan ke anggota-anggota osisnya. Di saat itu juga, Chika langsung terpikirkan sebuah ide, yaitu membuat *search engine*, tetapi Chika kurang mengerti cara membuat *search engine*, lalu Chika bertanya pada teman dekatnya, yaitu Kaguya, untuk menjelaskan cara kerja dari *search engine*.

Kaguya, yang mempunyai pengalaman dalam pembuatan *search engine*, langsung menjelaskan semuanya ke Chika dan Kaguya menjelaskan bahwa materi utama yang digunakan adalah materi vektor di ruang Euclidean, dengan topiknya adalah temu-balik informasi (*information retrieval*). Chika bertanya apakah itu temu-balik informasi dan Kaguya menjawab bahwa temu-balik informasi merupakan proses untuk menemukan kembali informasi yang relevan terhadap kebutuhan pengguna dari suatu kumpulan informasi yang tidak terstruktur secara otomatis. Setelah itu, Kaguya menjelaskan langkah-langkahnya dan Chika pun mencatatnya.

Langkah-langkah yang dicatat oleh Chika adalah, hal pertama yang harus dilakukan adalah mengubah *search query* menjadi ruang vektor, begitu juga dengan dokumen. Dokumen dan *query* yang telah diubah menjadi ruang vektor akan dikalkulasikan *term frequency*-nya dan *inverse document frequency*-nya dan keduanya dikalikan. Setelah kalkulasi tersebut dilakukan, perlu dilakukan penentuan dokumen mana yang relatif paling relevan dengan *search query*-nya dengan menggunakan *similarity measure* antara *query* dengan dokumen. *Similarity measure* tersebut dapat diukur dengan menggunakan *cosine similarity*. Chika merasa penjelasan Kaguya sangat berguna dan langsung bergegas untuk membuat *search query*-nya tersendiri.

Bab II

Teori Singkat

1. *Information Retrieval* (Temu-balik Informasi)

Menemukan kembali informasi yang relevan bergantung pada kebutuhan pengguna dari kumpulan-kumpulan informasi secara otomatis. IR (*Information Retrieval*) digunakan bukan untuk pencarian di dalam *database*, melainkan untuk pencarian informasi yang isinya tidak terstruktur, seperti dokumen, *webpage*, dan lain-lainnya. Konsep dasar dari IR adalah:

- Indexing*, pengindeksan dokumen yang mencakup proses pencatatan ciri-ciri dokumen, analisis dari isi dokumen tersebut, klasifikasi, dan pembuatan list entri yang bertujuan untuk mempermudah pencarian dokumen yang relevan dengan *query*-nya
- Searching*, proses untuk mencari dan menemukan kembali dokumen yang relevan dengan *query*
- Memberi *ranking* berdasarkan *query*-nya, proses pencarian data berdasarkan peringkat dokumen yang relevan dengan *query* dan akan ditampilkan berdasarkan tingkatan tersebut

Tahapan dari *indexing* tersebut dibagi menjadi 4, yaitu *parsing* (penghapusan tanda baca), *stopword removal* (penghapusan kata-kata yang relatif sering muncul dalam dokumen), *stemming* (penghapusan prefik dan sufik agar terbentuk kata dasar), dan TF-IDF (pengindeksan dokumen berdasarkan *query*).

2. Vektor

Kuantitas fisik yang memiliki arah dan besar. Vektor dapat digambarkan dengan diberikan tanda panah pada titik ujungnya dengan diperhitungkan arah dan besarnya. Vektor dapat dinotasikan dengan memberikan tanda panah pada variabel. Besaran dari suatu vektor dapat ditulis dengan modulus $|AB|$ dan akan selalu bernilai positif, yang berarti arahnya tidak perlu diperhitungkan. Vektor juga tidak bergantung pada letak, yang berarti vektor satu dapat dikatakan sama dengan vektor lainnya jika memiliki panjang dan arah yang sama.

3. *Cosine Similarity*

Berfungsi untuk membandingkan kemiripan antar dokumen dan pada umumnya dokumen tersebut dibandingkan kemiripannya dengan *query*. Untuk mendapatkan nilai *cosine similarity*, perlu dilakukan perkalian skalar antara *query* dengan dokumen kemudian dijumlahkan terlebih dahulu. Setelah itu, dicari panjang dari masing-masing dokumen dan *query* dan dikalikan. Lalu, hasil perkalian skalar tersebut dibagi dengan hasil perkalian panjang dokumen dan *query*. Rumusnya adalah seperti berikut, dengan asumsi A adalah dokumen dan B adalah *query*:

$$\text{Cos } \alpha = \frac{\mathbf{A} \cdot \mathbf{B}}{|\mathbf{A}| |\mathbf{B}|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

Keterangan:

A = Vektor A, yang akan dibandingkan kemiripannya

B = Vektor B, yang akan dibandingkan kemiripannya

$A \cdot B$ = *dot product* antara vektor A dan vektor B

$|A|$ = panjang vektor A

$|B|$ = panjang vektor B

$|A||B|$ = cross product antara $|A|$ dan $|B|$

Bab III

Implementasi Program

1. get_15_word

Methods

Nama	Tipe	Parameter	Deskripsi
get_15_words_from_synopsis	Prosedur		Mendapatkan 15 kata dari sinopsis yang didapatkan berdasarkan <i>web scraping</i>
get_first_15_words	Fungsi	String	Mengembalikan 15 kata pertama dalam sebuah string

2. information_retrieval

Methods

Nama	Tipe	Parameter	Deskripsi
load_links	Fungsi		Mengembalikan link yang menuju ke hasil pencarian
load_titles	Fungsi		Mengembalikan judul dari dokumen
load_data	Fungsi		Mengembalikan data yang telah diproses
load_first_15_words	Fungsi		Mengembalikan 15 kata pertama yang berada di suatu dokumen
load_total_words	Fungsi		Mengembalikan jumlah kata yang berada di suatu dokumen
save_links	Prosedur	Array	Menyimpan link yang menuju ke hasil pencarian
save_titles	Prosedur	Array	Menyimpan judul dari dokumen
save_data	Prosedur	Array	Menyimpan data yang telah diproses
save_first_15_words	Prosedur	Array	Menyimpan 15 kata pertama yang berada di suatu dokumen

save_total_words	Prosedur	Array	Menyimpan jumlah kata yang berada di suatu dokumen
retrieve_information	Fungsi	Array	Melakukan kalkulasi sehingga mendapatkan urutan dokumen berdasarkan <i>query</i> -nya lalu mengembalikan hasil urutan tersebut
upload_file	Prosedur	File Object, String, String	Melakukan proses peng- <i>upload</i> -an dokumen dan disimpan ke folder test

3. preprocess

Attributes

Nama	Tipe	Deskripsi
tokenizer	nltk.tokenize.regexp.RegexpTokenizer	Menghapus <i>punctuation</i>
stop_words	Set	<i>List</i> dari <i>stop words</i>
porter	nltk.stem.porter.PorterStemmer	Untuk melakukan <i>stemming</i>

Methods

Nama	Tipe	Parameter	Deskripsi
preprocess_sentence	Fungsi	Array	Menghapus <i>stopwords</i> dan melakukan <i>stemming</i> sebuah kalimat
preprocess	Prosedur		Memproses setiap kalimat yang <i>stopwords</i> -nya sudah dihapus dan <i>stemming</i> telah dilakukan

4. total_words

Methods

Nama	Tipe	Parameter	Deskripsi
get_total_words_from_synopsis	Prosedur		Mendapatkan jumlah kata sinopsis yang didapatkan berdasarkan <i>web scraping</i>

get_total_words	Fungsi	String	Mengembalikan jumlah kata dalam sebuah string
-----------------	--------	--------	---

5. vectorize

Methods

Nama	Tipe	Parameter	Deskripsi
ratio	Fungsi	Array, Integer, Integer, Array	Menghitung rasio setiap kata yang berada di suatu dokumen
TF	Fungsi	Matrix, Array, Integer, Integer	Melakukan kalkulasi TF
IDF	Fungsi	Integer, Integer, Matrix, Array	Melakukan kalkulasi IDF
TF_IDF	Fungsi	Integer, Integer, Matrix, Array	Melakukan kalkulasi TF-IDF
vectorizer	Fungsi	Array, Array	Mengembalikan hasil kalkulasi dari TF-IDF dan <i>list</i> dari kata-kata yang berada di seluruh dokumen
dot	Fungsi	Array, Array	Mengembalikan kalkulasi <i>dot product</i> dari <i>array</i> 1 dengan yang lainnya
norm	Fungsi	Array	Mengembalikan kalkulasi panjang dari sebuah vektor

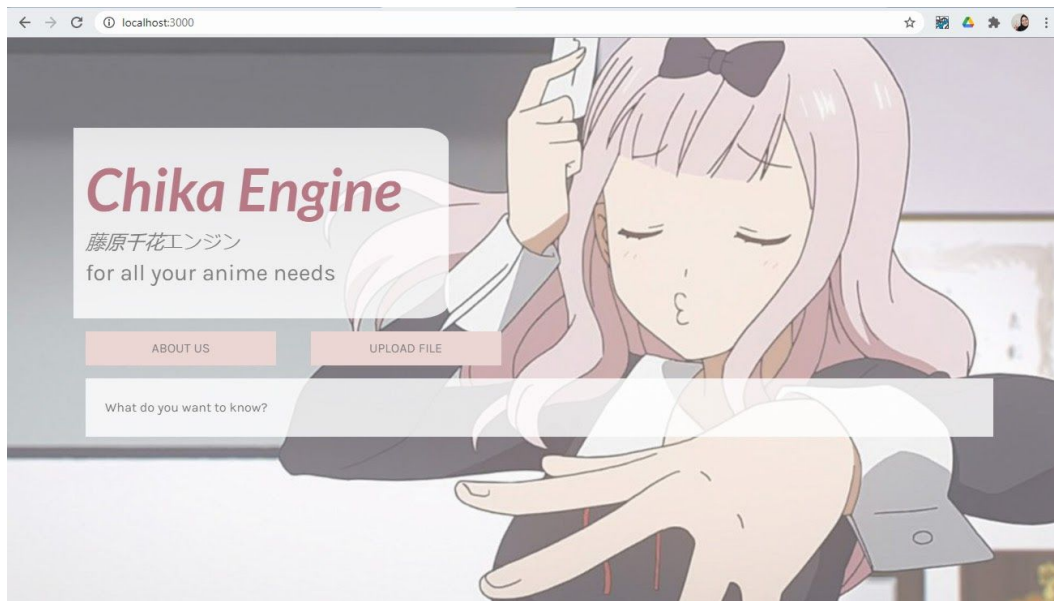
6. web-scrapper

Methods

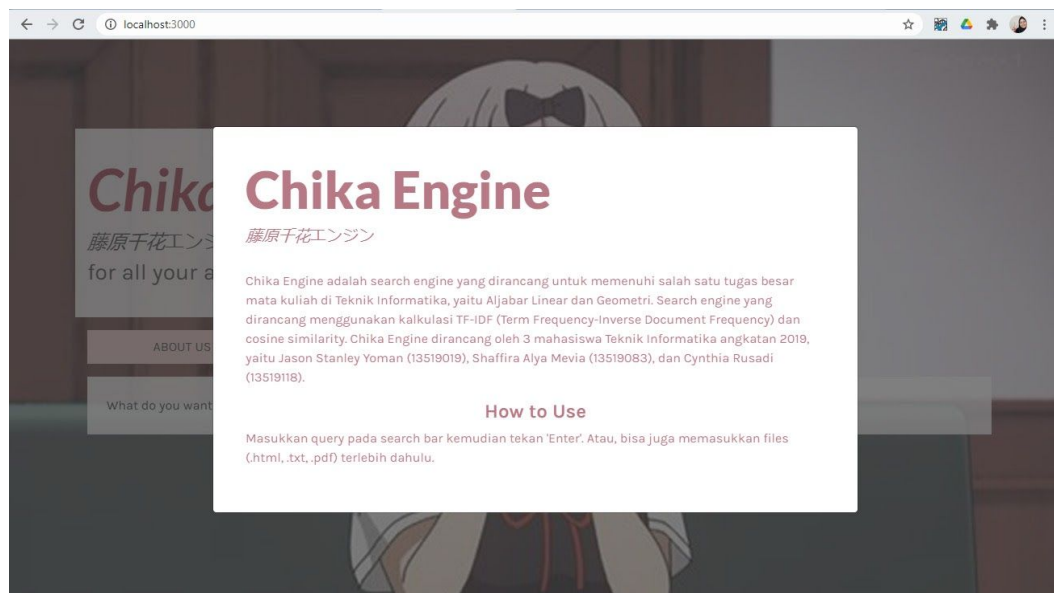
Nama	Tipe	Parameter	Deskripsi
start	Prosedur		Melakukan <i>web scraping</i> , yang pada program kali ini, <i>website</i> yang digunakan adalah https://myanimelist.net/topanime.php?limit=0

Bab IV Eksperimen

Hasil *screenshot* di bawah ini merupakan tampilan dari Chika Engine. Adanya 3 fitur yang dapat digunakan, yaitu ABOUT US, UPLOAD FILE, dan *search bar*. Tampilan ABOUT US mencakup deskripsi dari Chika Engine, cara penggunaan, dan kontak dari pembuatnya, yaitu Jason Stanley Yoman, Shaffira Alya Mevia, dan Cynthia Rusadi.

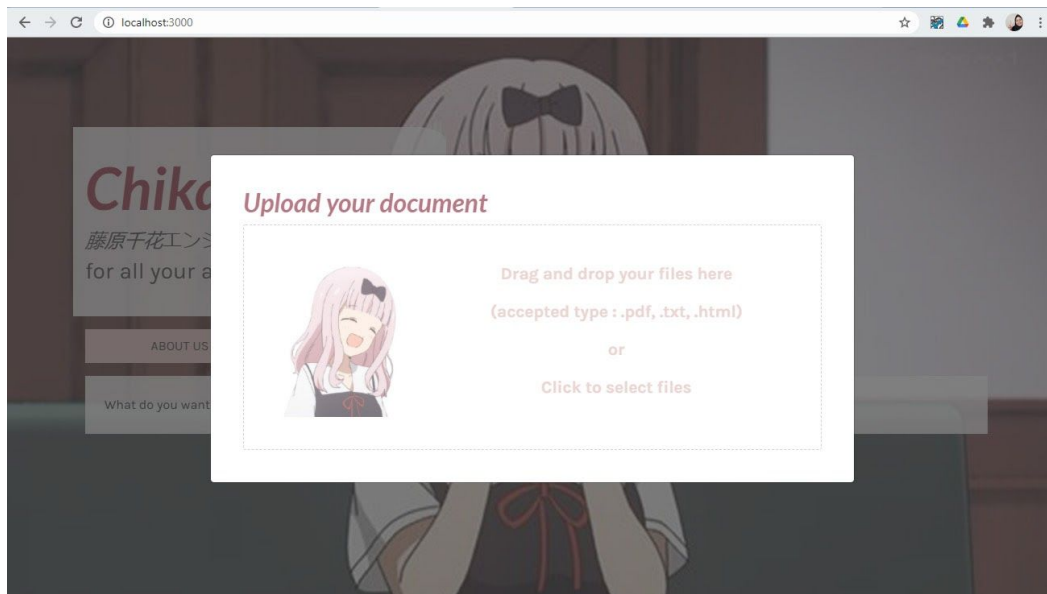


Gambar 4.1 Landing Page Chika Engine

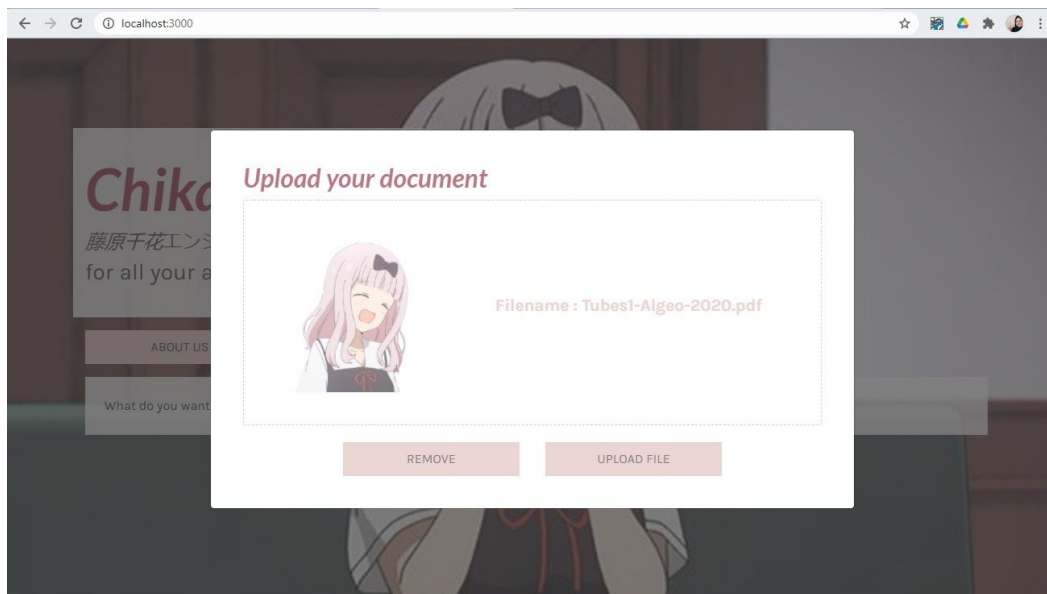


Gambar 4.2 Modal About Us

Kemudian, dua *screenshot* di bawah ini adalah tampilan saat akan melakukan *upload* dokumen. Jenis file dokumen yang dapat diterima adalah .pdf, .txt, dan .html.

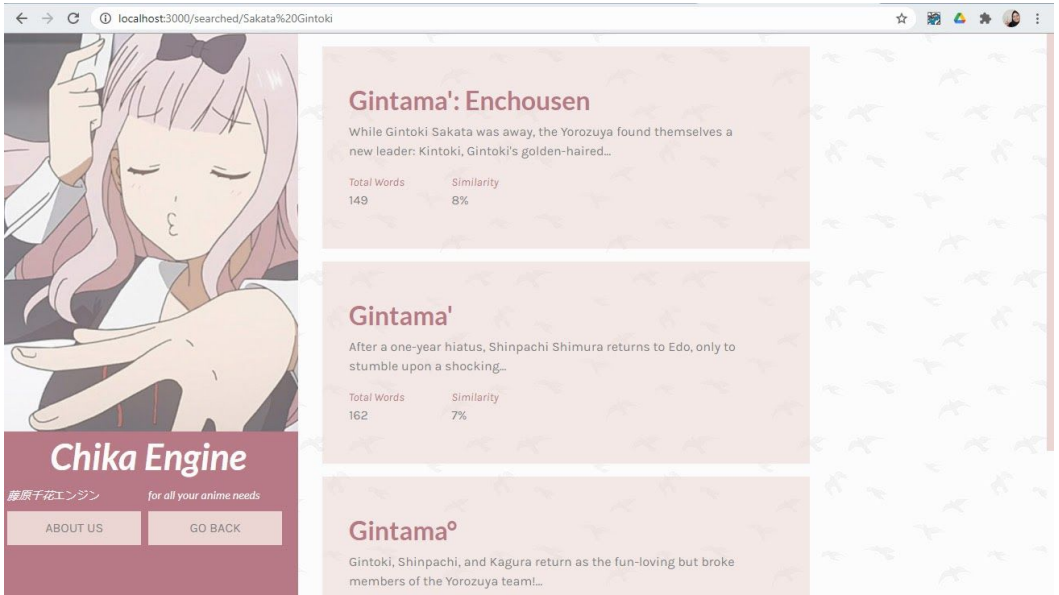


Gambar 4.3 Modal Upload Dokumen



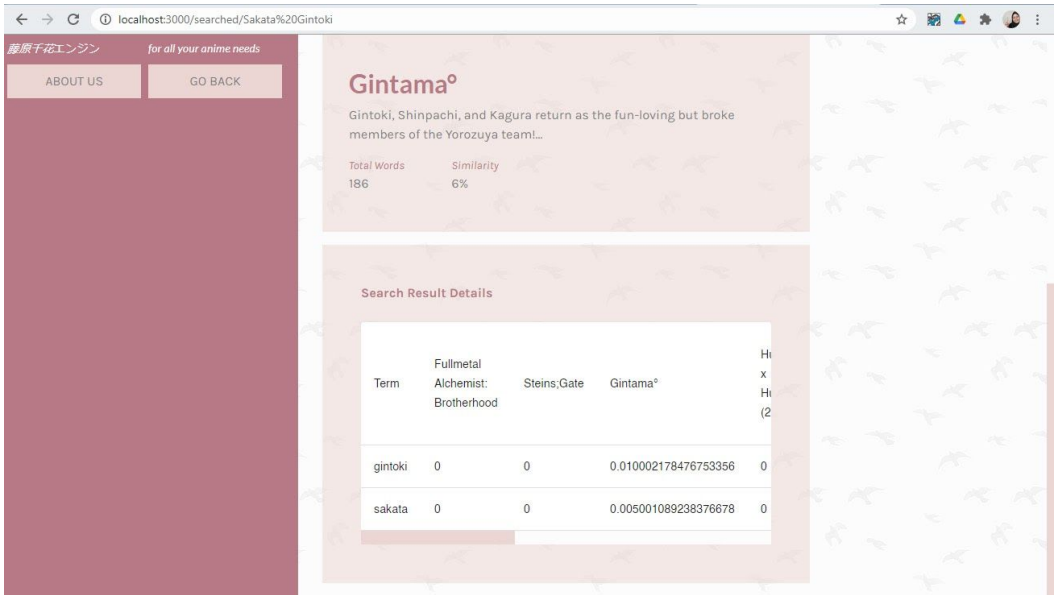
Gambar 4.4 Dokumen berhasil diterima

Screenshot di bawah ini merupakan salah satu contoh eksperimen. Query yang diterima adalah ‘Sakata Gintoki’ dan berikut adalah hasil *ranking* berdasarkan *query*-nya. Data dokumennya berasal dari hasil *web-scraping* <https://myanimelist.net/topanime.php?limit=0>.



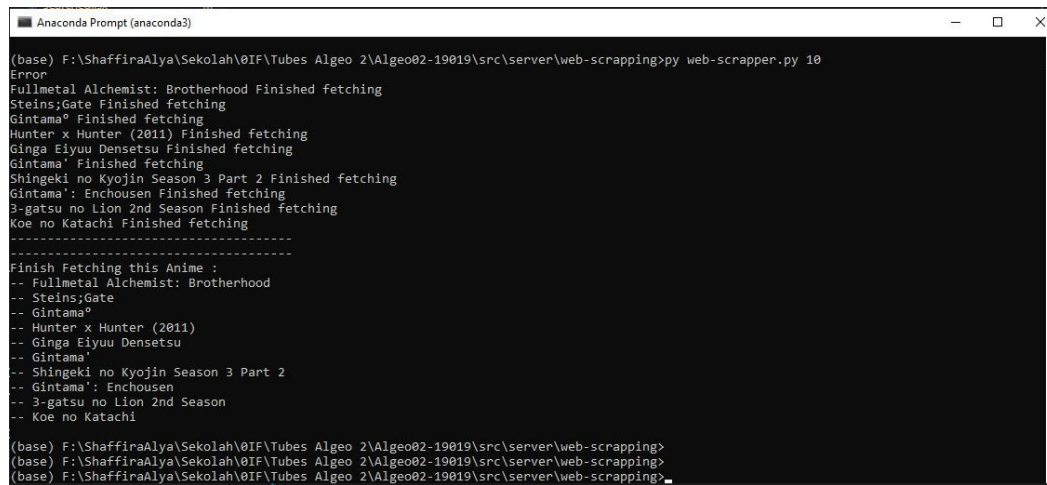
Gambar 4.5 Search Feed

Dan yang di bawah ini adalah tabel *term* dengan banyak kemunculan *term*.



Gambar 4.6 Detail hasil pencarian

Berikut adalah *screenshot* dari terminal saat melakukan *web scraping*. *Web scraping* yang digunakan pada contoh di bawah ini hanya mengambil beberapa judul, untuk efisiensi pengerjaan. *Output* dari *web scraping* tersebut adalah judul-judul yang berada di [website https://myanimelist.net/topanime.php?limit=0](https://myanimelist.net/topanime.php?limit=0).



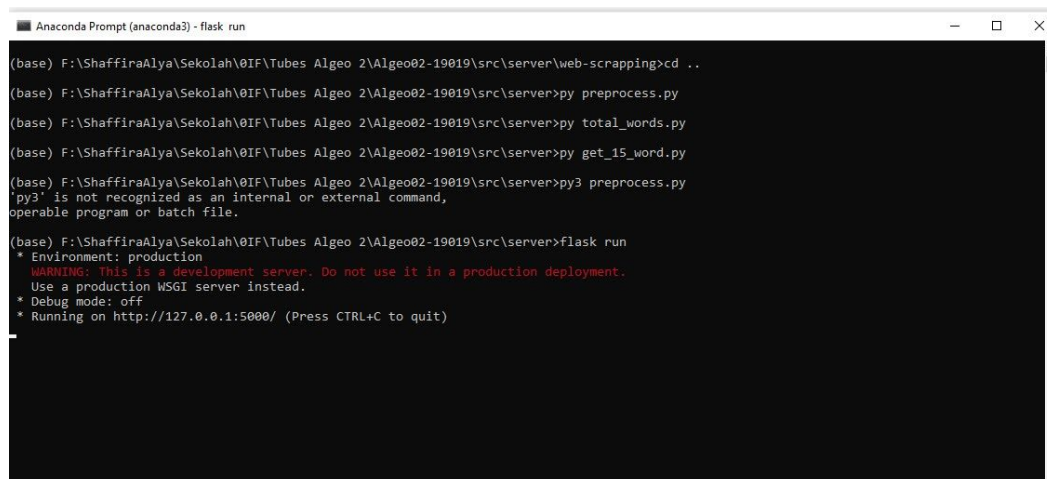
```

Anaconda Prompt (anaconda3)
(base) F:\ShaffiraAlya\Sekolah\0IF\Tubes Algeo 2\Algeo02-19019\src\server\web-scrapping>py web-scrapper.py 10
Error
Fullmetal Alchemist: Brotherhood Finished fetching
Steins;Gate Finished fetching
Gintama° Finished fetching
Hunter x Hunter (2011) Finished fetching
Ginga Eiyuu Densetsu Finished fetching
Gintama' Finished fetching
Shingeki no Kyojin Season 3 Part 2 Finished fetching
Gintama': Enchousen Finished fetching
3-gatsu no Lion 2nd Season Finished fetching
Koe no Katachi Finished fetching
-----
Finish Fetching this Anime :
-- Fullmetal Alchemist: Brotherhood
-- Steins;Gate
-- Gintama°
-- Hunter x Hunter (2011)
-- Ginga Eiyuu Densetsu
-- Gintama'
-- Shingeki no Kyojin Season 3 Part 2
-- Gintama': Enchousen
-- 3-gatsu no Lion 2nd Season
-- Koe no Katachi
(base) F:\ShaffiraAlya\Sekolah\0IF\Tubes Algeo 2\Algeo02-19019\src\server\web-scrapping>
(base) F:\ShaffiraAlya\Sekolah\0IF\Tubes Algeo 2\Algeo02-19019\src\server\web-scrapping>
(base) F:\ShaffiraAlya\Sekolah\0IF\Tubes Algeo 2\Algeo02-19019\src\server\web-scrapping>

```

Gambar 4.6 Proses *web-scraping*

Screenshot terminal di bawah ini adalah gambaran saat menyalakan server. Setelah melakukan *web scraping*, *command* yang harus digunakan adalah `py preprocess.py`, `py total_words.py`, `py get_15_word.py`, `py preprocess.py`, dan `flask run`. Jika OS yang digunakan bukan Windows, maka cobalah untuk menggantikan `command py` menjadi `python3`.



```

Anaconda Prompt (anaconda3) - flask run
(base) F:\ShaffiraAlya\Sekolah\0IF\Tubes Algeo 2\Algeo02-19019\src\server\web-scrapping>cd ..
(base) F:\ShaffiraAlya\Sekolah\0IF\Tubes Algeo 2\Algeo02-19019\src\server>py preprocess.py
(base) F:\ShaffiraAlya\Sekolah\0IF\Tubes Algeo 2\Algeo02-19019\src\server>py total_words.py
(base) F:\ShaffiraAlya\Sekolah\0IF\Tubes Algeo 2\Algeo02-19019\src\server>py get_15_word.py
(base) F:\ShaffiraAlya\Sekolah\0IF\Tubes Algeo 2\Algeo02-19019\src\server>py3 preprocess.py
'py3' is not recognized as an internal or external command,
operable program or batch file.
(base) F:\ShaffiraAlya\Sekolah\0IF\Tubes Algeo 2\Algeo02-19019\src\server>flask run
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

```

Gambar 4.7 Memproses hasil *web-scraping* dan menyalakan server

Bab V

Kesimpulan, Saran, dan Refleksi

1. Kesimpulan

Berdasarkan eksperimen yang telah dilakukan, dari *query* yang dimasukkan akan program sudah berhasil mencocokkan hasil dan mengeluarkan *output* berupa *search feed* dan dilengkapi juga dengan tabel detail pencarian. Untuk melakukan pengecekan *similarity*, ada 2 alternatif yang dapat digunakan, yaitu *upload* dokumen dan *web-scraping*.

2. Saran

Saran yang dapat kami berikan adalah untuk menggunakan *web-scraping* karena *upload* dokumen membutuhkan waktu yang lebih lama, yaitu harus melakukan *upload* satu per satu untuk setiap dokumennya. Dengan menggunakan *web-scraping*, akan mempermudah proses pengecekan.

3. Refleksi

Refleksi kami terhadap tugas besar ini adalah untuk membaca FAQ lebih awal agar dapat mengetahui *library* apa saja yang tidak diperbolehkan untuk digunakan. Karena pada sebelumnya, kalkulasi TF-IDF yang sebelumnya dikalkulasikan menggunakan *library* yang sudah tersedia, sedangkan *library* tersebut dilarang dan harus diimplementasikan secara manual.

Referensi

- <http://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-10-Vektor-di-Ruang-Euclidean-Bag1.pdf> (Diakses pada 12 November 2020; 22:59)
- <http://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-12-Aplikasi-dot-product-pada-IR.pdf> (Diakses pada 12 November 2020; 23:00)
- <https://garudacyber.co.id/artikel/1436-pengertian-dan-konsep-information-retrieval> (Diakses pada 12 November 2020; 22:44)
- <https://janav.wordpress.com/2013/10/27/tf-idf-and-cosine-similarity/> (Diakses pada 11 November 2020; 13:16)
- <https://journal.unnes.ac.id/nju/index.php/jte/article/download/10955/6659> (Diakses pada 12 November 2020; 23:10)
- <https://www.mejakita.com/materi/index/1157/apa-itu-vektor#:~:text=Vektor%20adalah%20suatu%20besaran%20yang,kecepatan%20%20percepatan%20%20dan%20gaya.&text=Vektor%20dapat%20digambarkan%20dengan%20memberi,dengan%20memperhitungkan%20besar%20dan%20arahnya.> (Diakses pada 12 November 2020; 22:55)
- <https://www.payahtidur.com/project/cosine-similarity> (Diakses pada 12 November 2020; 23:11)