

APLIKASI DOT PRODUCT PADA SISTEM TEMU-BALIK INFORMASI

LAPORAN TUGAS BESAR

Diajukan Untuk Memenuhi Tugas IF 2123 Aljabar Linier dan Geometri
Semester I 2020/2021



Disusun oleh

Gde Anantha Priharsena	(13519026)
Reihan Andhika Putra	(13519043)
Reyhan Emyr Arrosyid	(13519167)

**TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG
2020**

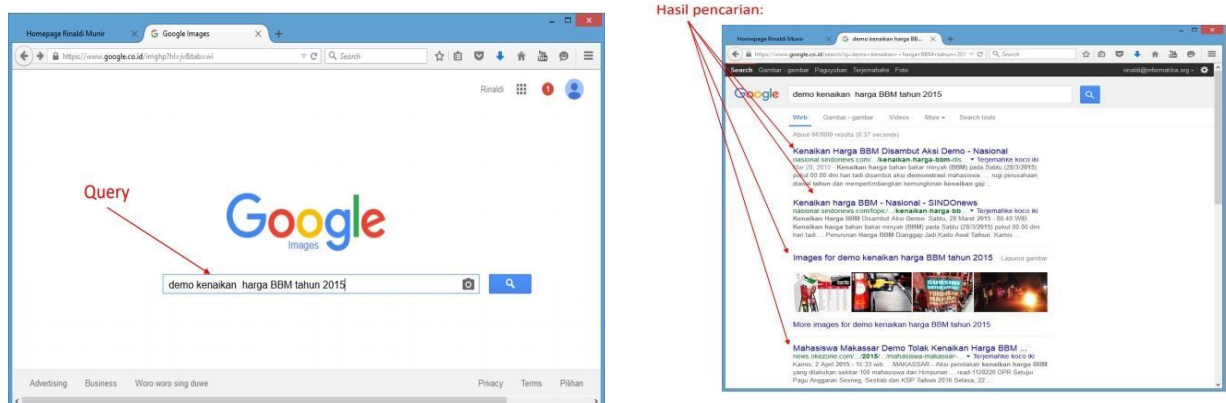
BAB I

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:

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

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

1.2 Penggunaan Program

Berikut ini adalah input yang akan dimasukkan pengguna untuk eksekusi program.

1. **Search query**, berisi kumpulan kata yang akan digunakan untuk melakukan pencarian
2. **Kumpulan dokumen**, dilakukan dengan cara mengunggah multiple file ke dalam web browser.

Tampilan layout dari aplikasi web yang akan dibangun adalah sebagai berikut.

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](#)

Gambar 2. Tampilan layout dari aplikasi web search engine yang dibangun.

Perihal: link ke halaman tentang program dan pembuatnya (Konsep singkat *search engine* yang dibuat, How to Use, About Us).

Catatan: 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*).

Anda dapat menambahkan menu lainnya, gambar, logo, dan sebagainya. Tampilan Front End dari website dibuat semenarik mungkin selama mencakup seluruh informasi pada layout yang diberikan di atas.

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 term dan banyak kemunculan term dalam setiap dokumen akan ditampilkan pada web browser dengan layout sebagai berikut.

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

Untuk menyederhanakan pembuatan search engine, terdapat hal-hal yang perlu diperhatikan dalam eksekusi program ini.

1. Silahkan lakukan stemming dan penghapusan *stopwords* pada setiap dokumen
2. Tidak perlu dibedakan antara huruf-huruf besar dan huruf-huruf kecil.
3. *Stemming* dan penghapusan stopword dilakukan saat **penyusunan vektor**, sehingga halaman yang berisi *full-text* terkait dokumen tetap seperti semula.
4. Penghapusan karakter-karakter yang tidak perlu untuk ditampilkan (jika menggunakan *web scraping* atau format dokumen berupa html)
5. Bahasa yang digunakan dalam dokumen adalah bahasa Inggris atau bahasa Indonesia (pilih salah satu)

Petunjuk: silahkan gunakan library sastrawi atau nltk untuk stemming kata dan penghapusan stopwords

1.3 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 (unsur kreativitas diperbolehkan/dianjurkan).
7. Program harus modular dan mengandung komentar yang jelas.
8. Dilarang menggunakan library cosine similarity yang sudah jadi.

BAB II

DASAR TEORI

2.1 Vektor

Vektor adalah objek geometri yang memiliki besaran dan memiliki arah. Setiap vektor dapat dinyatakan secara geometris sebagai segmen garis berarah pada bidang atau ruang. Vektor jika digambar dilambangkan dengan tanda panah (\rightarrow). Besar vektor proporsional dengan panjang panah dan arahnya bertepatan dengan arah panah. Vektor dapat melambangkan perpindahan dari titik A ke titik B . Vektor memiliki sifat-sifat sebagai berikut:

1. Vektor dikatakan sama jika memiliki besar dan arah yang sama.
2. Vektor harus memiliki unit yang sama agar dapat dijumlahkan atau dikurangkan.
3. Negatif dari suatu vektor memiliki besar yang sama namun berlawanan arah.
4. Pengurangan vektor dapat dilakukan dengan menjumlahkan dengan vektor negatif.
5. Perkalian atau pembagian vektor dengan skalar akan menghasilkan vektor.
6. Proyeksi dari suatu vektor di sepanjang sumbu koordinat disebut sebagai komponen vektor.
7. Menjumlahkan vektor dilakukan dengan menjumlahkan komponen-komponen yang bersesuaian.

2.2 Operasi Vektor

Vektor pun dapat dikenakan operasi aljabar seperti penjumlahan, pengurangan, dan perkalian. Perkalian vektor hanya dapat dilakukan jika kedua vektor berada pada ruang yang sama, yang terdiri dari:

1. Hasil kali titik (*dot product*)

Hasil kali titik akan menghasilkan besaran skalar. Misalnya a dan b berada pada vektor ruang yang sama, maka hasil kali titiknya akan didefinisikan sebagai berikut:

$$\vec{a} \cdot \vec{b} = \|\vec{a}\| \|\vec{b}\| \cos \alpha$$

Dimana $\|\vec{a}\|$ dan $\|\vec{b}\|$ masing – masing merupakan panjang vektor a dan b . Dan α adalah sudut yang dibentuk antara dua vektor tersebut.

2.3 Information Retrieval Dengan Model Ruang Vektor

Temu-balik informasi (*information retrieval*) adalah menemukan kembali (*retrieval*) informasi yang relevan terhadap kebutuhan pengguna dari suatu kumpulan informasi secara otomatis. IR tidak sama dengan pencarian di dalam basisdata (*database*). IR umumnya digunakan pada pencarian informasi yang isinya tidak terstruktur. Informasi terstruktur contohnya tabel-tabel di dalam basisdata (*database*). Informasi tak-terstruktur contohnya dokumen (isinya bergantung pembuatnya) dan laman web (*webpage*).

Salah satu model IR adalah **model ruang vektor**. Model ini menggunakan teori di dalam aljabar vector. Misalkan terdapat n kata berbeda sebagai kamus kata (*vocabulary*) atau indeks kata (*term index*). Maka dapat kita definisikan

1. Kata-kata tersebut membentuk ruang vektor berdimensi n . Setiap dokumen maupun *query* dinyatakan sebagai vektor $\mathbf{w} = (w_1, w_2, \dots, w_n)$ di dalam \mathbf{R}^n .
2. w_i = bobot setiap kata i di dalam *query* atau dokumen
3. Nilai w_i dapat menyatakan jumlah kemunculan kata tersebut dalam dokumen (*term frequency*)

Contoh: Misalkan terdapat tiga buah kata (T_1 , T_2 , dan T_3), dua buah dokumen (D_1 dan D_2) serta sebuah *query* Q . Masing-masing dinyatakan sebagai vector:

$$\mathbf{D}_1 = (2, 3, 5), \mathbf{D}_2 = (3, 7, 1), \mathbf{Q} = (0, 0, 2)$$

$\mathbf{D}_1 = (2, 3, 5)$ artinya dokumen D_1 mengandung 2 buah kata T_1 , 3 buah kata T_2 , dan 5 buah kata T_3 .

Contoh: Misalkan $T_1 = \text{Menteri}$, $T_2 = \text{minta}$, $T_3 = \text{Korupsi}$

$D_1 = \text{Menteri}$ olahraga *meminta* maaf atas perbuatan *korupsi*. *Menteri* tersebut terlibat *korupsi* anggaran. *Meminta-minta* komisi termasuk *korupsi*. *Korupsi* sudah mandarah daging di Indonesia. *Korupsi* sudah menjadi budaya.

$\mathbf{D}_2 = (3, 7, 1)$ artinya dokumen D_2 mengandung 3 buah kata T_1 , 7 buah kata T_2 , dan satu buah kata T_3 .

$D_2 =$ Gubernur Jabar *meminta* waktu ketemu *Menteri* Sosial. Dia *meminta* Pak *Menteri* mengunjungi panti. *Permintaan* yang wajar. Sekretaris Gubernur mengirim surat *permintaan* kepada *Menteri* tersebut. Apakah *meminta-minta* termasuk perbuatan *korupsi*? Tidak selalu, bukan? *Meminta* waktu saja.

$\mathbf{Q} = (0, 0, 2)$ artinya *query* Q hanya mengandung 2 buah kata T_3 .

Contoh: $Q = \text{Korupsi besar atau kecil tetap saja korupsi}$.

2.4 Cosine Similarity

Cosine Similarity digunakan untuk melihat kemiripan antar dokumen teks. Kemiripan dalam VSM ini ditemukan oleh vektor dari dokumen pembanding dan vektor dari dokumen uji. *Cosine Similarity* akan menghasilkan sebuah matriks yang saling berelasi antara dokumen-dokumen dengan melihat besar sudutnya. Cosinus sering digunakan untuk membandingkan dokumen – dokumen. Dapat dirumuskan sebagai berikut:

$$\cos \theta = \frac{D_1 \cdot D_2}{\|D_1\| \cdot \|D_2\|}$$

Keterangan:

D_1 = dokumen pembanding

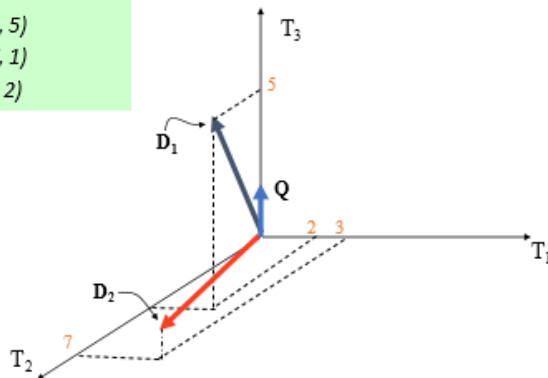
D_2 = dokumen uji

Contoh:

$D_1 = (2, 3, 5)$

$D_2 = (3, 7, 1)$

$Q = (0, 0, 2)$



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 “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*.

2.5 Text Preprocessing

Pada *natural language processing* (NLP), informasi yang akan digali berisi data-data yang strukturnya “sembarang” atau tidak terstruktur. Oleh karena itu, diperlukan proses perubahan bentuk menjadi data yang terstruktur untuk kebutuhan lebih lanjut (*sentiment analysis, topic modelling, dll*).

2.5.1. Library

Adapun library yang digunakan dalam melakukan *natural language processing* adalah

1. Natural Language Toolkit (NLTK)

Natural Language Toolkit adalah *library* python untuk bekerja dengan permodelan teks. NLTK menyediakan alat yang baik mempersiapkan teks sebelum digunakan pada *machine learning* atau algoritma *deep learning*. NLTK dapat diinstal melalui “pip”.

2. Python Sastrawi

Python Sastrawi adalah pengembangan dari proyek [PHP Sastrawi](#). Python Sastrawi merupakan library sederhana yang dapat mengubah kata berimbuhan bahasa Indonesia menjadi bentuk dasarnya. Sastrawi juga dapat diinstal melalui “pip”.

2.5.2. Case Folding

Case folding adalah salah satu bentuk *text preprocessing* yang paling sederhana dan efektif meskipun sering diabaikan. Tujuan dari *case folding* untuk mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf ‘a’ sampai ‘z’ yang diterima. Karakter selain huruf dihilangkan dan dianggap *delimiter*. Pada tahap ini tidak menggunakan *external library* apapun, kita bisa memanfaatkan modul yang tersedia di python. Ada beberapa cara yang dapat digunakan dalam tahap *case folding*, anda dapat menggunakan beberapa atau menggunakan semuanya, tergantung pada tugas yang diberikan.

1. Mengubah Text Menjadi Lowercase

Salah satu contoh pentingnya penggunaan *lower case* adalah untuk mesin pencarian. Bayangkan anda sedang mencari dokumen yang mengandung “indonesia” namun tidak ada hasil yang muncul karena “indonesia” di indeks sebagai “INDONESIA”. Contoh dibawah menunjukan bagaimana python mengubah teks menjadi *lowercase* :

```
kalimat = "Berikut ini adalah 5 negara dengan pendidikan  
terbaik di dunia adalah Korea Selatan, Jepang, Singapura,  
Hong Kong, dan Finlandia."lower_case = kalimat.lower()  
print(lower_case) # output  
# berikut ini adalah 5 negara dengan pendidikan terbaik di  
dunia adalah korea selatan, jepang, singapura, hong kong,  
dan finlandia.
```

2. Menghapus Angka

Hapuslah angka jika tidak relevan dengan apa yang akan anda analisa, contohnya seperti nomor rumah, nomor telepon, dll. *Regular expression (regex)* dapat digunakan untuk menghapus karakter angka. Python memiliki modulre untuk melakukan hal – hal yang berkaitan dengan *regex*. Contoh dibawah menunjukan bagaimana python menghapus angka dalam sebuah kalimat :

```
import re # impor modul regular expression
kalimat = "Berikut ini adalah 5 negara dengan pendidikan
terbaik di dunia adalah Korea Selatan, Jepang, Singapura,
Hong Kong, dan Finlandia."
hasil = re.sub(r"\d+", "", kalimat)
print(hasil) # ouput
# Berikut ini adalah negara dengan pendidikan terbaik di
dunia adalah Korea Selatan, Jepang, Singapura, Hong Kong,
dan Finlandia.
```

3. Menghapus Tanda Baca

Sama halnya dengan angka, tanda baca dalam kalimat tidak memiliki pengaruh pada *text preprocessing*. Menghapus tanda baca seperti `[!'#$%&'()*+,-./:;<=>?@[\\^`{|}~]` dapat dilakukan di pyhton seperti dibawah ini :

```
kalimat = "Ini &adalah [contoh] kalimat? {dengan} tanda.
baca?!!"
hasil =
kalimat.translate(str.maketrans("", "", string.punctuation))
print(hasil) # output
# Ini adalah contoh kalimat dengan tanda baca
```

4. Menghapus whitespace (karakter kosong)

Untuk menghapus spasi di awal dan akhir, anda dapat menggunakan fungsi `strip()` pada pyhton. Perhatikan kode dibawah ini :

```
kalimat = " \t ini kalimat contoh\t "
hasil = kalimat.strip()
print(hasil) # output
# ini kalimat contoh
```

2.5.3. Tokenizing

Tokenizing adalah proses pemisahan teks menjadi potongan-potongan yang disebut sebagai token untuk kemudian di analisa. Kata, angka, simbol, tanda baca dan entitas penting lainnya dapat dianggap sebagai token. Didalam NLP, token diartikan sebagai “kata” meskipun *tokenize* juga dapat dilakukan pada paragraf maupun kalimat.

Sebuah kalimat atau data dapat dipisah menjadi kata-kata dengan kelas `word_tokenize()` pada modul NLTK.

```
# impor word_tokenize dari modul nltk
from nltk.tokenize import word_tokenize
kalimat = "Andi kerap melakukan transaksi rutin secara daring atau online."
tokens = nltk.tokenize.word_tokenize(kalimat)
print(tokens) # ouput
# ['Andi', 'kerap', 'melakukan', 'transaksi', 'rutin', 'secara', 'daring', 'atau', 'online', '.']
```

Dari *output* kode diatas terdapat kemunculan tanda baca titik(.) dan koma (,) serta token “Andi” yang masih menggunakan huruf besar pada awal kata. Hal tersebut nantinya dapat mengganggu proses perhitungan dalam penerapan algoritma. Jadi, sebaiknya teks telah melewati tahap *case folding* sebelum di *tokenize* agar menghasilkan hasil yang lebih konsisten.

Prinsip yang sama dapat diterapkan untuk memisahkan kalimat pada paragraf. Anda dapat menggunakan kelas `sent_tokenize()` pada modul NLTK. Saya telah menambahkan kalimat pada contoh seperti dibawah ini :

```
# impor sent_tokenize dari modul nltk
from nltk.tokenize import sent_tokenize
kalimat = "Andi kerap melakukan transaksi rutin secara daring atau online. Menurut Andi belanja online lebih praktis & murah."
tokens = nltk.tokenize.sent_tokenize(kalimat)
print(tokens) # ouput
# ['Andi kerap melakukan transaksi rutin secara daring atau online.', 'Menurut Andi belanja online lebih praktis & murah.']
```

2.5.4. Filtering (Remove Stopword)

Filtering adalah tahap mengambil kata-kata penting dari hasil token dengan menggunakan algoritma *stoplist* (membuang kata kurang penting) atau *wordlist* (menyimpan kata penting).

Stopword adalah kata umum yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna. Contoh *stopword* dalam bahasa Indonesia adalah “yang”, “dan”, “di”, “dari”, dll. Makna di balik penggunaan *stopword* yaitu dengan menghapus kata-kata yang memiliki informasi rendah dari sebuah teks, kita dapat fokus pada kata-kata penting sebagai gantinya.

Contoh penggunaan *filtering* dapat kita temukan pada konteks mesin pencarian. Jika permintaan pencarian anda adalah “apa itu pengertian manajemen?” tentunya anda ingin sistem pencarian fokus pada memunculkan dokumen dengan topik tentang “pengertian manajemen” di atas dokumen dengan topik “apa itu”. Hal ini dapat dilakukan dengan mencegah kata dari daftar *stopword* dianalisa.

Selain untuk *stemming*, library Sastrawi juga mendukung proses *filtering*. Kita dapat menggunakan `stopWordRemoverFactory` dari modul sastrawi. Untuk melihat daftar *stopword* yang telah didefinisikan dalam library Sastrawi dapat menggunakan kode berikut :

```
from Sastrawi.StopWordRemover.StopWordRemoverFactory import
StopWordRemoverFactory
factory = StopWordRemoverFactory()
stopwords = factory.get_stop_words()
print(stopwords)
```

Kode diatas akan menampilkan *stopword* yang tersedia di library Sastrawi. Proses *filtering* pada Sastrawi dapat dilihat pada baris kode dibawah :

```
from Sastrawi.StopWordRemover.StopWordRemoverFactory import
StopWordRemoverFactory
from nltk.tokenize import word_tokenize
factory = StopWordRemoverFactory()
stopword = factory.create_stop_word_remover()
kalimat = "Andi kerap melakukan transaksi rutin secara daring atau online. Menurut Andi belanja online lebih praktis & murah."
kalimat =
kalimat.translate(str.maketrans('', '', string.punctuation)).lower()
stop = stopword.remove(kalimat)
```

```
tokens = nltk.tokenize.word_tokenize(stop)
print(tokens) # output
# ['andi', 'kerap', 'transaksi', 'rutin', 'daring', 'online',
'andi', 'belanja', 'online', 'praktis', 'murah']
```

Kita dapat menambah atau mengurangi kata pada daftar *stopword* sesuai dengan kebutuhan analisa. Pada dasarnya daftar *stopword* pada library Sastrawi tersimpan di dalam list yang anda lihat [disini](#). Jadi sebenarnya kita tinggal mengubah daftar pada list tersebut. Tetapi hal tersebut bisa menjadi permasalahan apabila pada suatu kasus kita diharuskan menambahkan *stopword* secara dinamis. Library Sastrawi dapat mengatasi permasalahan tersebut, perhatikan kode dibawah ini :

```
from Sastrawi.StopWordRemover.StopWordRemoverFactory import
StopWordRemoverFactory, StopWordRemover, ArrayDictionary
from nltk.tokenize import word_tokenize
stop_factory = StopWordRemoverFactory().get_stop_words() #load
default stopwords
more_stopword = ['daring', 'online'] #menambahkan stopwords
kalimat =
"andi kerap melakukan transaksi rutin secara daring atau online.
Menurut andi belanja online lebih praktis & murah."
kalimat =
kalimat.translate(str.maketrans('', '', string.punctuation)).lower()
data = stop_factory + more_stopword #menggabungkan stopwords
dictionary = ArrayDictionary(data)
str = StopWordRemover(dictionary)
tokens = nltk.tokenize.word_tokenize(str.remove(kalimat))

print(tokens) # output
# ['andi', 'kerap', 'transaksi', 'rutin', 'andi', 'belanja',
'praktis', 'murah']
```

2.5.5. Stemming

Stemming adalah proses menghilangkan [infleksi](#) kata ke bentuk dasarnya, namun bentuk dasar tersebut tidak berarti sama dengan akar kata (*root word*). Misalnya kata “mendengarkan”, “dengarkan”, “didengarkan” akan ditransformasi menjadi kata “dengar”.

Idenya adalah ketika anda mencari dokumen “cara membuka lemari”, anda juga ingin melihat dokumen yang menyebutkan “cara terbuka lemari” atau

“cara dibuka lemari” meskipun terdengar tidak enak. Tentunya anda ingin mencocokkan semua variasi kata untuk memunculkan dokumen yang paling relevan.

Proses *stemming* antara satu bahasa dengan bahasa yang lain tentu berbeda. Contohnya pada teks berbahasa inggris, proses yang diperlukan hanya proses menghilangkan sufiks. Sedangkan pada teks berbahasa Indonesia semua kata imbuhan baik itu sufiks dan prefiks juga dihilangkan.

Untuk melakukan *stemming* bahasa Indonesia kita dapat menggunakan library Python Sastrawi yang sudah kita siapkan di awal. *Library* Sastrawi menerapkan Algoritma Nazief dan Adriani dalam melakukan *stemming* bahasa Indonesia.

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
factory = StemmerFactory()
stemmer = factory.create_stemmer()

kalimat = "Andi kerap melakukan transaksi rutin secara daring atau
online. Menurut Andi belanja online lebih praktis & murah."
hasil = stemmer.stem(kalimat)
print(hasil)

# ouput
# andi kerap laku transaksi rutin cara daring atau online turut
andi belanja online lebih praktis murah
```

BAB III

IMPLEMENTASI PROGRAM

Bahasa Pemrograman : Python

Framework :

1. Frontend : Flask, Bootstrap
2. Backend : Flask

Database : SQLAlchemy

Model :

Document adalah satu-satunya model yang akan berinteraksi dengan database yang dimiliki oleh sistem. Document akan berkorelasi dengan tabel 'Document'. Document sendiri berisi beberapa kolom yaitu

1. id sebagai primary key dengan type data integer + primary key
2. date_created sebagai tanggal dibuatnya file dengan type data datetime
3. name sebagai nama dokumen dengan type data string dan boleh kosong
4. url sebagai link dokumen apabila menggunakan web scraping dengan type data string dan boleh kosong
5. wordcnt sebagai banyaknya kata dalam dokumen sebelum preprocessing dengan type data integer
6. first_sentence sebagai kalimat pertama dalam dokumen dengan type data string
7. sim sebagai similaritas dokumen dengan query dengan tipe data float dan boleh kosong

View :

1. base.html
 - Digunakan sebagai dasar dari file html yang lain, Di dalam base.html, didefinisikan script dan style css yang digunakan oleh semua file lain
- a. index.html
 - Digunakan sebagai tampilan utama dari web, di file ini ditampilkan semua daftar dokumen, tempat input file baru, dan term table
- b. aboutus.html
 - Digunakan untuk menampilkan penjelasan dari program ini dan juga pembuatnya
- c. howtouse.html
 - Digunakan untuk menampilkan cara menggunakan program ini

2. viewfile.html

→ Digunakan untuk menampilkan isi dari dokumen yang berupa file .txt

Controller :

Dikarenakan hanya terdapat satu model pada program kami yaitu model Document maka controller yang ada juga hanya satu yaitu controller Document. Adapun method yang dimiliki oleh controller tersebut adalah

1. index():

→ Route : '/'

→ Allowed method : POST, GET

→ Digunakan untuk merender template halaman utama (index.html). Apabila user tidak menginput 'query' untuk di cari similaritasnya maka fungsi akan melakukan Query ke database untuk mengambil data document yang akan ditampilkan di halaman utama. Apabila user menginput 'query' untuk dicari similaritas maka fungsi akan melakukan perhitungan cosine similarity dari input query terhadap semua dokumen yang ada di database ataupun di web scraping lalu menghitung similaritasnya dan mengupdate nilai similaritas di database lalu kembali merender template halaman utama dengan data dokumen yang diurut berdasar similaritas dan data term table yang bisa ditampilkan di halaman utama.

2. upload():

→ Route : '/upload

→ Allowed method : POST, GET

→ Method upload digunakan apabila user melakukan upload beberapa file. Ekstensi file yang diupload akan divalidasi. Setelah itu untuk tiap file yang diunggah akan disimpan dalam folder './static/namafile.txt', lalu di database diupdate nama dari file, jumlah kata yang ada di file tanpa punctuation, kalimat pertama di file, dan similaritas awal yaitu 0. Apabila ada tahap yang gagal (validasi atau proses penambahan ke database) maka file yang tersimpan akan dihapus dan user dilempar ke halaman utama dengan pesan kesalahan. Apabila berhasil maka user akan dilempar ke halaman utama dengan pesan success.

3. getUrl()



















→ Route : '/get-from-url'

- Allowed method : POST, GET
 - Method `getUrl` digunakan untuk mengambil data dokumen yang dimasukkan oleh pengguna dari website. Setelah menerima website, isi dari website tersebut disimpan ke dalam file static dengan terlebih dahulu membuang tag html, lalu di database di-update nama dari file, jumlah kata yang ada di file tanpa punctuation, kalimat pertama di file, dan similaritas awal yaitu 0. Apabila ada tahap yang gagal (validasi atau proses penambahan ke database) maka file yang tersimpan akan dihapus dan user dilempar ke halaman utama dengan pesan kesalahan. Apabila berhasil maka user akan dilempar ke halaman utama dengan pesan success.
4. `delete(id)`
- Route : `'/delete/<id>'`
 - Allowed method : GET
 - Digunakan untuk menghapus dokumen dari list sekaligus database yang ada. Setelah pengguna menekan tombol delete pada elemen yang diinginkan, elemen dengan id tersebut akan dihapus dari database dan juga file statik yang menyimpan isi dokumen tersebut dihapus.
5. `view(id)`
- Route : `'/view/<id>'`
 - Allowed method : GET
 - Digunakan untuk melihat isi dari dokumen yang diinginkan jika dokumen tersebut berupa file .txt

BAB IV

EKSPERIMEN

4.1 Txt File

Percobaan 1																																												
Penguji : Reihan Andhika Putra	Jumlah Dokumen : 6	Panjang Dokumen : 1 Kalimat																																										
tc1-1_reihan.txt	tc1-2_reihan.txt	tc1-3_reihan.txt																																										
Reihan sedang bermain sepak bola dengan Doni.	Aku makam malam bersama keluarga Reihan.	Reihan sedang mengerjakan soal UTS Kalkulus di kampus.																																										
tc1-4_reihan.txt	tc1-5_reihan.txt	tc1-6_reihan.txt																																										
Andi sedang belajar untuk mengerjakan UTS Kalkulus.	Andi suka bermain sepak bola di kampus.	Doni, Andi, dan Reihan sedang makan malam.																																										
Query : Andi belajar untuk UTS																																												
Hasil Query																																												
<div><div>Data Table</div><table><tr><th>Filename</th><th>Added</th><th>First line</th><th>Word length</th><th>Similarity</th><th>Actions</th></tr><tr><td>tc1-4_reihan.txt</td><td>2020-11-09</td><td>Andi sedang belajar untuk mengerjakan UTS Kalkulus.</td><td>7</td><td>70.71%</td><td></td></tr><tr><td>tc1-5_reihan.txt</td><td>2020-11-09</td><td>Andi suka bermain sepak bola di kampus.</td><td>7</td><td>23.57%</td><td></td></tr><tr><td>tc1-6_reihan.txt</td><td>2020-11-09</td><td>Doni, Andi, dan Reihan sedang makan malam.</td><td>7</td><td>23.57%</td><td></td></tr><tr><td>tc1-3_reihan.txt</td><td>2020-11-09</td><td>Reihan sedang mengerjakan soal UTS Kalkulus di kampus.</td><td>8</td><td>21.82%</td><td></td></tr><tr><td>tc1-1_reihan.txt</td><td>2020-11-09</td><td>Reihan sedang bermain sepak bola dengan Doni.</td><td>7</td><td>0.0%</td><td></td></tr><tr><td>tc1-2_reihan.txt</td><td>2020-11-09</td><td>Aku makam malam bersama keluarga Reihan.</td><td>6</td><td>0.0%</td><td></td></tr></table></div>			Filename	Added	First line	Word length	Similarity	Actions	tc1-4_reihan.txt	2020-11-09	Andi sedang belajar untuk mengerjakan UTS Kalkulus.	7	70.71%		tc1-5_reihan.txt	2020-11-09	Andi suka bermain sepak bola di kampus.	7	23.57%		tc1-6_reihan.txt	2020-11-09	Doni, Andi, dan Reihan sedang makan malam.	7	23.57%		tc1-3_reihan.txt	2020-11-09	Reihan sedang mengerjakan soal UTS Kalkulus di kampus.	8	21.82%		tc1-1_reihan.txt	2020-11-09	Reihan sedang bermain sepak bola dengan Doni.	7	0.0%		tc1-2_reihan.txt	2020-11-09	Aku makam malam bersama keluarga Reihan.	6	0.0%	
Filename	Added	First line	Word length	Similarity	Actions																																							
tc1-4_reihan.txt	2020-11-09	Andi sedang belajar untuk mengerjakan UTS Kalkulus.	7	70.71%																																								
tc1-5_reihan.txt	2020-11-09	Andi suka bermain sepak bola di kampus.	7	23.57%																																								
tc1-6_reihan.txt	2020-11-09	Doni, Andi, dan Reihan sedang makan malam.	7	23.57%																																								
tc1-3_reihan.txt	2020-11-09	Reihan sedang mengerjakan soal UTS Kalkulus di kampus.	8	21.82%																																								
tc1-1_reihan.txt	2020-11-09	Reihan sedang bermain sepak bola dengan Doni.	7	0.0%																																								
tc1-2_reihan.txt	2020-11-09	Aku makam malam bersama keluarga Reihan.	6	0.0%																																								
Term Table		Perhitungan Manual																																										

Terms	Query	tc1-4_reihan.txt	tc1-5_reihan.txt	tc1-6_reihan.txt	tc1-3_reihan.txt	tc1-1_reihan.txt	tc1-2_reihan.txt
andi	1	1	1	1	0	0	0
ajar	1	1	0	0	0	0	0
uts	1	1	0	0	1	0	0
reihan	0	0	0	1	1	1	1
sedang	0	1	0	1	1	1	0
main	0	0	1	0	0	1	0
sepak	0	0	1	0	0	1	0
bola	0	0	1	0	0	1	0
doni	0	0	0	1	0	1	0
aku	0	0	0	0	0	0	1
makam	0	0	0	0	0	0	1
malam	0	0	0	1	0	0	1
sama	0	0	0	0	0	0	1
keluarga	0	0	0	0	0	0	1
kerja	0	1	0	0	1	0	0
soal	0	0	0	0	1	0	0
kalkulus	0	1	0	0	1	0	0
kampus	0	0	1	0	1	0	0
suka	0	0	1	0	0	0	0
makan	0	0	0	1	0	0	0

Sim doc 1 : 0 (tidak ada kata yang mirip)

Sim doc 2 : 0 (tidak ada kata yang mirip)

Sim doc 3 :

$$1 / (3^{0.5} \times 7^{0.5}) \times 100\% = 21,82 \%$$

Sim doc 4 :

$$(1+1+1)/(3^{0.5} \times 6^{0.5}) \times 100\% = 70,71\%$$

Sim doc 5 :

$$(1)/(3^{0.5} \times 6^{0.5}) \times 100\% = 23,57\%$$

Sim doc 6

$$(1)/(3^{0.5} \times 6^{0.5}) \times 100\% = 23,57\%$$

Perhitungan secara manual sudah sama dengan di program. Hasilnya ada beda sedikit koma saja

Percobaan 2

Penguji : Reihan Andhika Putra	Jumlah Dokumen : 5	Panjang Dokumen : 1 Kalimat
tc2-1_reihan.txt	tc2-2_reihan.txt	tc2-3_reihan.txt
Aku ingin begini, aku ingin begitu, ini itu banyak sekali.	Aku ingin begini, aku ingin begitu, ini itu banyak sekali.	Aku ingin kesana kemari ke angkasa.
tc2-4_reihan.txt		tc2-5_reihan.txt
Doraemon melukis dengan alat ajaib.		Baling baling bambu doraemon membawanya ke banyak tempat.
Query : Doraemon terbang ke angkasa		
Hasil Query		

Data Table

Filename	Added	First line	Word length	Similarity	Actions
tc2-2_reihan.txt	2020-11-09	Aku ingin terbang bebas, di angkasa.	6	57.74%	
tc2-3_reihan.txt	2020-11-09	Aku ingin kesana kemari ke angkasa.	6	28.87%	
tc2-4_reihan.txt	2020-11-09	Doraemon melukis dengan alat ajaib.	5	28.87%	
tc2-5_reihan.txt	2020-11-09	Baling baling bambu doraemon membawanya ke banyak tempat.	8	19.25%	
tc2-1_reihan.txt	2020-11-09	Aku ingin begini, aku ingin begitu, ini itu banyak sekali.	10	0.0%	

Term Table

Full Terms Table

Terms	Query	tc2-2_reihan.txt	tc2-3_reihan.txt	tc2-4_reihan.txt	tc2-5_reihan.txt	tc2-1_reihan.txt
doraemon	1	0	0	1	1	0
terbang	1	1	0	0	0	0
angkasa	1	1	1	0	0	0
aku	0	1	1	0	0	2
begini	0	0	0	0	0	1
begitu	0	0	0	0	0	1
itu	0	0	0	0	0	1
banyak	0	0	0	0	1	1
sekali	0	0	0	0	0	1
bebas	0	1	0	0	0	0
kesana	0	0	1	0	0	0
kemari	0	0	1	0	0	0
luk	0	0	0	1	0	0
alat	0	0	0	1	0	0
ajaib	0	0	0	1	0	0
baling	0	0	0	0	2	0
bambu	0	0	0	0	1	0
bawa	0	0	0	0	1	0
tempat	0	0	0	0	1	0

Perhitungan Manual

Sim doc 1 : 0 (tidak ada kata yang mirip)

Sim doc 2 :

$$(2)/(3^{0.5} \times 4^{0.5}) \times 100\% = 57,73\%$$

Sim doc 3 :

$$(2)/(3^{0.5} \times 4^{0.5}) \times 100\% = 28,86\%$$

Sim doc 4 :

$$2)/(3^{0.5} \times 4^{0.5}) \times 100\% = 28,86\%$$

Sim doc 5 :

$$(1)/(3^{0.5} \times 9^{0.5}) = 19,24\%$$

Perhitungan secara manual sudah sama dengan di program. Hasilnya ada beda sedikit koma saja

Percobaan 3

Penguji : Reyhan Emyr Arrosyid

Jumlah Dokumen : 4

Panjang Dokumen : 2 kalimat

tc3-1_emyr.txt

tc3-2_emyr.txt

Para mahasiswa baru saja diberi tugas besar ke-5. Mereka langsung sibuk mengerjakannya.

Joni dan Budi sedang berjalan-jalan. Mereka bertemu dengan Siti di taman.

tc3-3_emyr.txtx





tc3-4_emyr.txt

Budi sedang mengerjakan tugas besar Aljabar Linier dan Geometri. Budi merasa sangat senang.

Paman Anto sedang makan bakso di pinggir jalan. Ia tidak menghabiskannya karena sudah kenyang.

Query : Budi belajar aljabar dengan melihat tugasnya

Hasil Query

Filename	Added	First line	Word length	Similarity	Actions
tc3-3_emyr.txt	2020-11-11	Budi sedang mengerjakan tugas besar Aljabar Linier dan Geometri.	13	47.81%	
tc3-2_emyr.txt	2020-11-11	Joni dan Budi sedang berjalan-jalan.	12	16.9%	
tc3-1_emyr.txt	2020-11-11	Para mahasiswa baru saja diberi tugas besar ke-5.	13	14.14%	
tc3-4_emyr.txt	2020-11-11	Paman Anto sedang makan bakso di pinggir jalan.	14	0.0%	

Term Table

Perhitungan Manual

Terms	Query	tc3-3_emyr.txt	tc3-2_emyr.txt	tc3-1_emyr.txt	tc3-4_emyr.txt
budi	1	2	1	0	0
ajar	1	0	0	0	0
aljabar	1	1	0	0	0
lihat	1	0	0	0	0
tugas	1	1	0	1	0
mahasiswa	0	0	0	1	0
baru	0	0	0	1	0
diberi	0	0	0	1	0
besar	0	1	0	1	0
ke	0	0	0	1	0
5	0	0	0	1	0
langsung	0	0	0	1	0
sibuk	0	0	0	1	0
kerja	0	1	0	1	0
joni	0	0	1	0	0
sedang	0	1	1	0	1
jalan	0	0	1	0	1
temu	0	0	1	0	0
siti	0	0	1	0	0
taman	0	0	1	0	0
linier	0	1	0	0	0
geometri	0	1	0	0	0
rasa	0	1	0	0	0
sangat	0	1	0	0	0
senang	0	1	0	0	0
paman	0	0	0	0	1
anto	0	0	0	0	1
makan	0	0	0	0	1
bakso	0	0	0	0	1
pinggir	0	0	0	0	1
tidak	0	0	0	0	1
habis	0	0	0	0	1
sudah	0	0	0	0	1
kenyang	0	0	0	0	1

Sim doc 1:

$$\frac{1}{\sqrt{5}\sqrt{10}} \times 100\% = 14.14\%$$

Sim doc 2:

$$\frac{1}{\sqrt{5}\sqrt{7}} \times 100\% = 16.90\%$$

Sim doc 3:










$$\frac{4}{\sqrt{5}\sqrt{14}} \times 100\% = 47.809\%$$

Sim doc 4:

$$\frac{0}{\sqrt{5}\sqrt{11}} \times 100\% = 0\%$$

Komentar : Perhitungan secara manual sudah sama dengan di program.

Percobaan 4

Penguji : Reyhan Emyr Arrosyid	Jumlah Dokumen : 3	Panjang Dokumen : 3 kalimat																								
tc4-1_emyr.txt		tc4-2_emyr.txt																								
Di wilayah Sumatera hiduplah seorang petani yang sangat rajin bekerja. Ia hidup sendiri sebatang kara. Setiap hari ia bekerja menggarap lading dan mencari ikan dengan tidak mengenal lelah.		Setelah beberapa saat memandangi ikan hasil tangkapannya, petani itu sangat terkejut. Ternyata ikan yang ditangkapnya itu bisa berbicara. "Tolong aku jangan dimakan Pak!! Biarkan aku hidup", teriak ikan itu.																								
tc4-3_emyr.txt																										
Pada jaman dahulu di daerah jawa barat ada seorang lelaki yang sangat kaya. Seluruh sawah dan ladang di desanya menjadi miliknya. Penduduk desa hanya menjadi buruh tani penggarap sawah dan ladang lelaki kaya itu.																										
Query : Petani itu makan ikan di sumatera																										
Hasil Query																										
<div><div><div>Data Table</div><table><tr><th>Filename</th><th>Added</th><th>First line</th><th>Word length</th><th>Similarity</th><th>Actions</th></tr><tr><td>tc4-2_emyr.txt</td><td>2020-11-11</td><td>Setelah beberapa saat memandangi ikan hasil tangkapannya, petani itu sangat terkejut.</td><td>29</td><td>44.19%</td><td></td></tr><tr><td>tc4-1_emyr.txt</td><td>2020-11-11</td><td>Di wilayah Sumatera hiduplah seorang petani yang sangat rajin bekerja.</td><td>28</td><td>29.42%</td><td></td></tr><tr><td>tc4-3_emyr.txt</td><td>2020-11-11</td><td>Pada jaman dahulu di daerah jawa barat ada seorang lelaki yang sangat kaya.</td><td>34</td><td>8.22%</td><td></td></tr></table></div></div>			Filename	Added	First line	Word length	Similarity	Actions	tc4-2_emyr.txt	2020-11-11	Setelah beberapa saat memandangi ikan hasil tangkapannya, petani itu sangat terkejut.	29	44.19%		tc4-1_emyr.txt	2020-11-11	Di wilayah Sumatera hiduplah seorang petani yang sangat rajin bekerja.	28	29.42%		tc4-3_emyr.txt	2020-11-11	Pada jaman dahulu di daerah jawa barat ada seorang lelaki yang sangat kaya.	34	8.22%	
Filename	Added	First line	Word length	Similarity	Actions																					
tc4-2_emyr.txt	2020-11-11	Setelah beberapa saat memandangi ikan hasil tangkapannya, petani itu sangat terkejut.	29	44.19%																						
tc4-1_emyr.txt	2020-11-11	Di wilayah Sumatera hiduplah seorang petani yang sangat rajin bekerja.	28	29.42%																						
tc4-3_emyr.txt	2020-11-11	Pada jaman dahulu di daerah jawa barat ada seorang lelaki yang sangat kaya.	34	8.22%																						
Term Table		Perhitungan Manual																								

Full Terms Table

Terms	Query	tc4-2_emyr.txt	tc4-1_emyr.txt	tc4-3_emyr.txt
tani	1	1	1	1
makan	1	1	0	0
ikan	1	3	1	0
sumatera	1	0	1	0
wilayah	0	0	1	0
hidup	0	1	2	0
orang	0	0	1	1
sangat	0	1	1	1
rajin	0	0	1	0
kerja	0	0	2	0
sendiri	0	0	1	0
batang	0	0	1	0
kara	0	0	1	0
tiap	0	0	1	0
hari	0	0	1	0
garap	0	0	1	1
lading	0	0	1	0
cari	0	0	1	0
tidak	0	0	1	0
kenal	0	0	1	0
lelah	0	0	1	0
beberapa	0	1	0	0
pandang	0	1	0	0
hasil	0	1	0	0
tangkap	0	2	0	0
kejut	0	1	0	0
nyata	0	1	0	0
bisa	0	1	0	0
bicara	0	1	0	0
aku	0	2	0	0
jangan	0	1	0	0
pak	0	1	0	0
biar	0	1	0	0
teriak	0	1	0	0
jaman	0	0	0	1
daerah	0	0	0	1
jawa	0	0	0	1
barat	0	0	0	1
lelaki	0	0	0	2
kaya	0	0	0	2
seluruh	0	0	0	1
sawah	0	0	0	2
ladang	0	0	0	2
di	0	0	0	1
desa	0	0	0	2
jadi	0	0	0	2
milik	0	0	0	1
duduk	0	0	0	1
buruh	0	0	0	1

Sim doc 1:

$$\frac{3}{\sqrt{4}\sqrt{26}} \times 100\% = 29.417\%$$

Sim doc 2:

$$\frac{5}{\sqrt{4}\sqrt{32}} \times 100\% = 44.19\%$$

Sim doc 3:

$$\frac{1}{\sqrt{4}\sqrt{37}} \times 100\% = 8.219\%$$

Komentar : Perhitungan secara manual sudah sama dengan di program.

Percobaan 5

Penguji : Reyhan Emyr Arrosyid

Jumlah Dokumen : 2

Panjang Dokumen : 2
paragraf

tc5-1_emyr.txt

Di sebuah hutan, Singa adalah penguasa hutan. semua binatang sangat menghormatinya. Ia adalah Raja yang sangat bijak dan perkasa. Sehingga, tidak ada binatang yang berani kepadanya. Suatu hari, sang Raja hutan mendapatkan sebuah kiriman daging dari hutan sebelah. daging tersebut sangat lezat dan Raja pun melahapnya. Namun, setelah selesai makan. Ia merasa mulutnya sangat bau dan ternyata daging tersebut dicampuri dengan petai.

Sang Raja pun terdiam di dalam rumah dan tidak pergi kemana pun karena merasa sangat malu. Namun, tidak lama kemudian. Ia pun mendapatkan sebuah ide dan ingin menguji seluruh rakyatnya tentang bagaimana ia memimpin selama ini. Akhirnya, ia pun memanggil tiga binatang dari hutan yaitu, Katak, Kuda, dan Kancil.

tc5-2_emyr.txt



Sudah berbulan-bulan lamanya musim kemarau panjang datang. sementara itu hujan belum menampilkan tanda-tanda akan turun. Siapapun pasti akan tersiksa. terutama warga rawa. Lompatan Kodi Kodok jadi tak selincah biasanya. Cica si Cacing juga setengah mati menggali tanah. semua lesu, dan yang nampak paoing tersiksa adalah Bidi si Badak! karena kulitnya yang tebal harus direndam didalam air agar suhu tubuhnya tidak kepanasan.

Meskipun begitu, mereka tidak ada yang mengeluh. Karena semua sama-sama memahami, yang lain pasti sama tersiksanya. Sebagai pimpinan di rawa, Bidi Badak mengkhawatirkan nasib teman-temannya. Makanya, Bidi Badak mulai gelisah mencari kolam baru.

Query : Binatang itu datang dari hutan

Hasil Query

Data Table

Filename	Added	First line	Word length	Similarity	Actions
tc5-1_emyr.txt	2020-11-11	Di sebuah hutan, Singa adalah penguasa hutan.	111	35.96%	
tc5-2_emyr.txt	2020-11-11	Sudah berbulan-bulan lamanya musim kemarau panjang datang.	98	5.8%	

Term Table

Query Terms Table

Terms	Query	tc5-1_emyr.txt	tc5-2_emyr.txt
binatang	1	3	0
datang	1	0	1
hutan	1	5	0

Perhitungan Manual

Sim doc 1:

$$\frac{8}{\sqrt{3}\sqrt{135}} \times 100\% = 35.957\%$$

Sim doc 2:

$$\frac{1}{\sqrt{3}\sqrt{99}} \times 100\% = 5.80\%$$

Komentar : Perhitungan secara manual sudah sama dengan di program.

4.2 Web Scraping

Percobaan 1			
Penguji: Gde Anantha Priharsena	Jumlah Dokumen: 1	Jumlah URL: 2	
tc1_web_nantha.txt			
Pemain baru barcelona yang baru dipromosikan dari barcelona B bernama Riqui Puig belum dimainkan hingga saat ini oleh pelatih baru barcelona, Ronald Koeman.			
Link 1: https://www.bolasport.com/read/312420966/jika-barcelona-mampu-wujudkan-2-hal-lionel-messi-siap-perpanjang-kontrak			
Link 2: https://www.bolasport.com/read/312421574/sikap-tak-ramah-lionel-messi-bikin-antoine-griezmann-hancur-di-barcelona			
Query : Lionel messi adalah pemain bola barcelona			
Hasil Query			
	Data Table		
	Filename	Added First line Word length Similarity Actions	
	tc1_web_nantha.txt	2020-11-12 Pemain baru barcelona yang baru dipromosikan dari barcelona B 23 39.53%	
	Jika Barcelona Mampu Wujudkan 2 Hal, Lionel Messi Siap Perpanjang Kontrak - Bolasport.com	2020-11-12 Jika Barcelona Mampu Wujudkan 2 Hal, Lionel Messi Siap Perpanjang Kontrak - Bolasport.com 1052 38.7%	
Sikap Tak Ramah Lionel Messi Bikin Antoine Griezmann Hancur di Barcelona? - Bolasport.com	2020-11-12 Sikap Tak Ramah Lionel Messi Bikin Antoine Griezmann Hancur di Barcelona? 1042 33.89%		
Term Table		Perhitungan Manual	
Query Terms Table		Sim Doc 1 :	
		$\frac{5}{\sqrt{5}\sqrt{32}} \times 100\% = 39.53\%$	
		Sim Link 1 :	
		$\frac{57}{\sqrt{5}\sqrt{4339}} \times 100\% = 38.703\%$	
		Sim Link 2 :	
		$\frac{49}{\sqrt{5}\sqrt{4181}} \times 100\% = 33.888\%$	
Komentar : Dokumen yang diambil dari link lumayan panjang sehingga membutuhkan waktu agak lama saat mengirim query			

Percobaan 2																																															
Penguji : Gde Anantha Priharsena			Jumlah Dokumen : 0		Jumlah URL : 4																																										
Link 1 : https://bola.okezone.com/read/2020/11/10/46/2307224/mantan-agen-griezmann-lionel-messi-bikin-rusak-barcelona																																															
Link 2 : https://sport.detik.com/sepakbola/liga-spainvol/d-5249462/griezmann-mandek-di-barcelona-gara-gara-messi																																															
Link 3 : https://www.bolasport.com/read/312421574/sikap-tak-ramah-lionel-messi-bikin-antoine-griezmann-hancur-di-barcelona																																															
Link 4 : https://www.bolasport.com/read/312420966/jika-barcelona-mampu-wujudkan-2-hal-lionel-messi-siap-perpanjang-kontrak																																															
Query : Griezmann dan Lionel Messi bermain bersama di Barcelona																																															
Hasil Query																																															
Data Table																																															
<table><tr><th>Filename</th><th>Added</th><th>First line</th><th>Word length</th><th>Similarity</th><th>Actions</th></tr><tr><td>Mantan Agen Griezmann: Lionel Messi Bikin Rusak Barcelona! : Okezone Bola</td><td>2020-11-10</td><td>Mantan Agen Griezmann: Lionel Messi Bikin Rusak Barcelona!</td><td>795</td><td>65.32%</td><td></td></tr><tr><td>Griezmann Mandek di Barcelona Gara-gara Messi?</td><td>2020-11-10</td><td>Griezmann Mandek di Barcelona Gara-gara Messi?</td><td>548</td><td>63.43%</td><td></td></tr><tr><td>Sikap Tak Ramah Lionel Messi Bikin Antoine Griezmann Hancur di Barcelona? - Bolasport.com</td><td>2020-11-10</td><td>Sikap Tak Ramah Lionel Messi Bikin Antoine Griezmann Hancur di Barcelona?</td><td>1019</td><td>33.96%</td><td></td></tr><tr><td>Jika Barcelona Mampu Wujudkan 2 Hal, Lionel Messi Siap Perpanjang Kontrak - Bolasport.com</td><td>2020-11-10</td><td>Jika Barcelona Mampu Wujudkan 2 Hal, Lionel Messi Siap Perpanjang Kontrak - Bolasport.com</td><td>1029</td><td>31.64%</td><td></td></tr></table>						Filename	Added	First line	Word length	Similarity	Actions	Mantan Agen Griezmann: Lionel Messi Bikin Rusak Barcelona! : Okezone Bola	2020-11-10	Mantan Agen Griezmann: Lionel Messi Bikin Rusak Barcelona!	795	65.32%		Griezmann Mandek di Barcelona Gara-gara Messi?	2020-11-10	Griezmann Mandek di Barcelona Gara-gara Messi?	548	63.43%		Sikap Tak Ramah Lionel Messi Bikin Antoine Griezmann Hancur di Barcelona? - Bolasport.com	2020-11-10	Sikap Tak Ramah Lionel Messi Bikin Antoine Griezmann Hancur di Barcelona?	1019	33.96%		Jika Barcelona Mampu Wujudkan 2 Hal, Lionel Messi Siap Perpanjang Kontrak - Bolasport.com	2020-11-10	Jika Barcelona Mampu Wujudkan 2 Hal, Lionel Messi Siap Perpanjang Kontrak - Bolasport.com	1029	31.64%													
Filename	Added	First line	Word length	Similarity	Actions																																										
Mantan Agen Griezmann: Lionel Messi Bikin Rusak Barcelona! : Okezone Bola	2020-11-10	Mantan Agen Griezmann: Lionel Messi Bikin Rusak Barcelona!	795	65.32%																																											
Griezmann Mandek di Barcelona Gara-gara Messi?	2020-11-10	Griezmann Mandek di Barcelona Gara-gara Messi?	548	63.43%																																											
Sikap Tak Ramah Lionel Messi Bikin Antoine Griezmann Hancur di Barcelona? - Bolasport.com	2020-11-10	Sikap Tak Ramah Lionel Messi Bikin Antoine Griezmann Hancur di Barcelona?	1019	33.96%																																											
Jika Barcelona Mampu Wujudkan 2 Hal, Lionel Messi Siap Perpanjang Kontrak - Bolasport.com	2020-11-10	Jika Barcelona Mampu Wujudkan 2 Hal, Lionel Messi Siap Perpanjang Kontrak - Bolasport.com	1029	31.64%																																											
Term Table				Perhitungan Manual																																											
Query Terms Table				Sim Link 1 : $\frac{99}{\sqrt{6}\sqrt{3828}} \times 100\% = 65.32\%$ Sim Link 2 : $\frac{64}{\sqrt{6}\sqrt{1697}} \times 100\% = 63.43\%$ Sim Link 3 : $\frac{54}{\sqrt{6}\sqrt{4214}} \times 100\% = 33.96\%$ Sim Link 4 : $\frac{41}{\sqrt{5}\sqrt{3358}} \times 100\% = 31.64\%$																																											
<table><tr><th>Terms</th><th>Query</th><th>Mantan Agen Griezmann: Lionel Messi Bikin Rusak Barcelona! : Okezone Bola</th><th>Griezmann Mandek di Barcelona Gara-gara Messi?</th><th>Sikap Tak Ramah Lionel Messi Bikin Antoine Griezmann Hancur di Barcelona? - Bolasport.com</th><th>Jika Barcelona Mampu Wujudkan 2 Hal, Lionel Messi Siap Perpanjang Kontrak - Bolasport.com</th></tr><tr><td>griezmann</td><td>1</td><td>17</td><td>17</td><td>12</td><td>1</td></tr><tr><td>lionel</td><td>1</td><td>12</td><td>7</td><td>6</td><td>8</td></tr><tr><td>messi</td><td>1</td><td>20</td><td>15</td><td>10</td><td>13</td></tr><tr><td>main</td><td>1</td><td>11</td><td>4</td><td>11</td><td>9</td></tr><tr><td>sama</td><td>1</td><td>7</td><td>2</td><td>2</td><td>1</td></tr><tr><td>barcelona</td><td>1</td><td>32</td><td>19</td><td>13</td><td>19</td></tr></table>				Terms	Query	Mantan Agen Griezmann: Lionel Messi Bikin Rusak Barcelona! : Okezone Bola	Griezmann Mandek di Barcelona Gara-gara Messi?	Sikap Tak Ramah Lionel Messi Bikin Antoine Griezmann Hancur di Barcelona? - Bolasport.com	Jika Barcelona Mampu Wujudkan 2 Hal, Lionel Messi Siap Perpanjang Kontrak - Bolasport.com	griezmann	1	17	17	12	1	lionel	1	12	7	6	8	messi	1	20	15	10	13	main	1	11	4	11	9	sama	1	7	2	2	1	barcelona	1	32	19	13	19		
Terms	Query	Mantan Agen Griezmann: Lionel Messi Bikin Rusak Barcelona! : Okezone Bola	Griezmann Mandek di Barcelona Gara-gara Messi?	Sikap Tak Ramah Lionel Messi Bikin Antoine Griezmann Hancur di Barcelona? - Bolasport.com	Jika Barcelona Mampu Wujudkan 2 Hal, Lionel Messi Siap Perpanjang Kontrak - Bolasport.com																																										
griezmann	1	17	17	12	1																																										
lionel	1	12	7	6	8																																										
messi	1	20	15	10	13																																										
main	1	11	4	11	9																																										
sama	1	7	2	2	1																																										
barcelona	1	32	19	13	19																																										

Komentar: Dokumen yang diambil dari link lumayan panjang sehingga membutuhkan waktu agak lama saat mengirim query.

Percobaan Final		
Penguji : Gde Anantha Priharsena	Jumlah Dokumen : 3	Jumlah URL : 3
tc1_web_nantha.txt		
Pemain baru barcelona yang baru dipromosikan dari barcelona B bernama Riqui Puig belum dimainkan hingga saat ini oleh pelatih baru barcelona, Ronald Koeman.		
tc2_web_nantha.txt	tc3_web_nantha.txt	
Dembele membuka keunggulan Barcelona di menit ke-22. Antonio Sanabria menyamakan skor di injury time babak pertama. Barulah Messi bermain di babak kedua. Antoine Griezmann mencetak gol yang membawa Barcelona unggul di menit ke-49, lalu Messi membuat dua gol lagi untuk membawa timnya unggul 4-2. Betis sempat memangkas skor lewat Loren Moron.	Barcelona yang bermain menyerang sejak menit awal mendapatkan beberapa kali peluang lewat Ansu Fati dan Griezmann, tapi gawang Claudio Bravo belum bisa dibobol. Gawang Bravo akhirnya bisa dijebol pada menit ke-22 lewat Dembele. Griezmann mengoper bola ke Dembele dan dengan kaki kirinya, pemain asal Prancis itu melepaskan sepakan keras ke pojok atas gawang.	
Link 1 : https://sport.detik.com/sepakbola/liga-spanyol/d-5249462/griezmann-mandek-di-barcelona-gara-gara-messi		
Link 2 : https://bola.okezone.com/read/2020/11/10/46/2307224/mantan-agen-griezmann-lionel-messi-bikin-rusak-barcelona		
Link 3 : https://www.bolasport.com/read/312421574/sikap-tak-ramah-lionel-messi-bikin-antoine-griezmann-hancur-di-barcelona		
Query : Pemain barcelona yaitu messi, griezmann, dembele dan pedri.		
Hasil Query		

Data Table						
Filename	Added	First line	Word length	Similarity	Actions	
Griezmann Mandek di Barcelona Gara-gara Messi?	2020-11-12	Griezmann Mandek di Barcelona Gara-gara Messi?	544	57.43%		
Mantan Agen Griezmann: Lionel Messi Bikin Rusak Barcelona! : Okezone Bola	2020-11-12	Mantan Agen Griezmann: Lionel Messi Bikin Rusak Barcelona!	814	50.58%		
tc3_web_nantha.txt	2020-11-12	Barcelona yang bermain menyerang sejak menit awal mendapatkan	54	36.59%		
tc1_web_nantha.txt	2020-11-12	Pemain baru barcelona yang baru dipromosikan dari barcelona B	23	36.08%		
tc2_web_nantha.txt	2020-11-12	Dembele membuka keunggulan Barcelona di menit ke-22.	54	34.66%		
Sikap Tak Ramah Lionel Messi Bikin Antoine Griezmann Hancur di Barcelona? - Bolasport.com	2020-11-12	Sikap Tak Ramah Lionel Messi Bikin Antoine Griezmann Hancur di Barcelona?	1042	32.2%		

Term Table							Perhitungan Manual																																																								
Query Terms Table							Sim Doc 1 :																																																								
<table><tr><th>Terms</th><th>Query</th><th>Griezmann Mandek di Barcelona Gara-gara Messi?</th><th>Mantan Agen Griezmann: Lionel Messi Bikin Rusak Barcelona! : Okezone Bola</th><th>tc3_web_nantha.txt</th><th>tc1_web_nantha.txt</th><th>tc2_web_nantha.txt</th><th>Sikap Tak Ramah Lionel Messi Bikin Antoine Griezmann Hancur di Barcelona? - Bolasport.com</th></tr><tr><td>main</td><td>1</td><td>7</td><td>9</td><td>2</td><td>2</td><td>1</td><td>14</td></tr><tr><td>barcelona</td><td>1</td><td>16</td><td>30</td><td>1</td><td>3</td><td>2</td><td>14</td></tr><tr><td>messi</td><td>1</td><td>18</td><td>22</td><td>0</td><td>0</td><td>2</td><td>12</td></tr><tr><td>griezmann</td><td>1</td><td>19</td><td>13</td><td>2</td><td>0</td><td>1</td><td>11</td></tr><tr><td>dembele</td><td>1</td><td>0</td><td>1</td><td>2</td><td>0</td><td>1</td><td>0</td></tr><tr><td>pedri</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>							Terms	Query	Griezmann Mandek di Barcelona Gara-gara Messi?	Mantan Agen Griezmann: Lionel Messi Bikin Rusak Barcelona! : Okezone Bola	tc3_web_nantha.txt	tc1_web_nantha.txt	tc2_web_nantha.txt	Sikap Tak Ramah Lionel Messi Bikin Antoine Griezmann Hancur di Barcelona? - Bolasport.com	main	1	7	9	2	2	1	14	barcelona	1	16	30	1	3	2	14	messi	1	18	22	0	0	2	12	griezmann	1	19	13	2	0	1	11	dembele	1	0	1	2	0	1	0	pedri	1	0	0	0	0	0	0	$\frac{5}{\sqrt{6}\sqrt{32}} \times 100\% = 36.08\%$
Terms	Query	Griezmann Mandek di Barcelona Gara-gara Messi?	Mantan Agen Griezmann: Lionel Messi Bikin Rusak Barcelona! : Okezone Bola	tc3_web_nantha.txt	tc1_web_nantha.txt	tc2_web_nantha.txt	Sikap Tak Ramah Lionel Messi Bikin Antoine Griezmann Hancur di Barcelona? - Bolasport.com																																																								
main	1	7	9	2	2	1	14																																																								
barcelona	1	16	30	1	3	2	14																																																								
messi	1	18	22	0	0	2	12																																																								
griezmann	1	19	13	2	0	1	11																																																								
dembele	1	0	1	2	0	1	0																																																								
pedri	1	0	0	0	0	0	0																																																								
							Sim Doc 2 :																																																								
							$\frac{7}{\sqrt{6}\sqrt{68}} \times 100\% = 34.66\%$																																																								
							Sim Doc 3 :																																																								
							$\frac{7}{\sqrt{6}\sqrt{61}} \times 100\% = 36.59\%$																																																								
							Sim Link 1 :																																																								
							$\frac{60}{\sqrt{6}\sqrt{1819}} \times 100\% = 57.432\%$																																																								
							Sim Link 2 :																																																								
							$\frac{75}{\sqrt{6}\sqrt{3664}} \times 100\% = 50.583\%$																																																								
							Sim Link 3 :																																																								
							$\frac{51}{\sqrt{6}\sqrt{4181}} \times 100\% = 32.199\%$																																																								

Komentar : Dokumen yang diambil dari link lumayan panjang sehingga membutuhkan waktu agak lama saat mengirim query

BAB V

PENUTUP

1. Kesimpulan

Dari tugas besar IF 2123 Aljabar Linier dan Geometri semester I 2020/2021 berjudul “Aplikasi Dot Product Pada Sistem Temu Balik Informasi”, kami berhasil membuat sebuah program mesin pencarian (*Search Engine*) yang berbasis sistem temu-balik informasi (*information retrieval*) pada sebuah *website* pada server lokal. Program ini dapat menambahkan koleksi dokumen melalui dua metode, yaitu meng-*upload* file dengan ekstensi .txt dan metode *web scraping* (mengambil teks dari pranala *website*). Apabila user melakukan *input* pada *search query* maka program akan secara otomatis menghitung kesamaan antara query dengan dokumen yang ada di koleksi dokumen. Semakin sama suatu vektor dokumen dengan vektor query, semakin relevan dokumen tersebut dengan query. Kesamaan tersebut dihitung dengan *cosine similarity*. Setelah itu, Program akan menampilkan tingkat kesamaan dari setiap dokumennya dengan query berikut dengan vektor tiap dokumen dan vektor query dalam tabel term.

2. Saran

Saran-saran yang dapat kami berikan untuk tugas besar IF 2123 Aljabar Linier dan Geometri semester I 2020/2021 adalah:

- Algoritma yang digunakan pada Tugas Besar ini masih memiliki banyak kekurangan sehingga sangat memungkinkan untuk dilakukan efisiensi, misalnya dengan penggunaan beberapa library yang tersedia. Oleh karena itu, dalam pengembangan program ini, masih bisa dilakukan efisiensi kinerja.
- Program ini dapat dikembangkan lebih lanjut baik dari segi UI/UX supaya semakin *user-friendly* atau dari segi fungsionalitas program yang dapat dikembangkan untuk cek plagiarisme.
- Memperjelas spesifikasi dan batasan-batasan setiap program pada file tugas besar untuk mencegah adanya multitafsir dan kesalahpahaman pada proses pembuatan program.

- d. Menimbang fungsionalitas dari Program pada Tugas Besar ini, sebaiknya program ini bisa dipublikasikan setelah dikembangkan lebih lanjut. Supaya program ini memiliki kebermanfaatan yang lebih luas.

3. Refleksi

Setelah menyelesaikan tugas besar IF 2123 Aljabar Linier dan Geometri semester I 2020/2021, kami dapat merefleksikan beberapa hal, yaitu:

- a. Komunikasi antar anggota kelompok berjalan dengan baik, sehingga tidak terjadi miskomunikasi atau kesalahpahaman selama pengerjaan.
- b. Selama pengerjaan Tugas Besar, telah dibuat beberapa *milestone* untuk setiap orangnya dengan *deadline* yang bervariasi dan semua terselesaikan tepat waktu.
- c. Selama proses pembuatan program, ketika ditemukan ketidaksesuaian saat proses eksperimen, temuan langsung dikomunikasikan kepada anggota kelompok lainnya dan bersama-sama mencari solusi.
- d. Perlunya untuk mempelajari *web development* agar kedepannya dapat membuat sebuah website yang lebih fungsional dan lebih *user-friendly* dari segi UI/UX.
- e. Lebih merapikan *source code* program karena ada beberapa fungsi yang didefinisikan secara tidak modular.
- f. Mengerjakan tugas besar dengan perasaan gembira, karena *it's not worth it if you're not have fun*.

DAFTAR PUSTAKA

Welcome to Flask. (n.d.). Retrieved November 3, 2020, from <https://flask.palletsprojects.com/>

(Referensi penggunaan Flask).

Nugroho, K. S. (2020, June 04). Dasar Text Preprocessing dengan Python. Retrieved November 6, 2020, from <https://medium.com/@ksnugroho/dasar-text-preprocessing-dengan-python-a4fa52608ffe>

(Referensi penggunaan dan syntax NLTK dan Bab 2).

Munir, R. (n.d.). Aljabar Geometri. Retrieved November 3, 2020, from <http://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/algeo.htm>

(Referensi penggunaan Bab 2).

Real Python. (2020, November 07). Flask by Example – Text Processing with Requests, BeautifulSoup, and NLTK. Retrieved November 9, 2020, from <https://realpython.com/flask-by-example-part-3-text-processing-with-requests-beautifulsoup-nltk/>

(Referensi aplikasi dan syntax Web Scraping)

Akbar, J. (2019, January 14). Jumadilakbar/Recomender-Cosine-Similarity-tfidf-python-flask. Retrieved November 6, 2020, from <https://github.com/jumadilakbar/Recomender-Cosine-Similarity-tfidf-python-flask>

(Referensi Cosine Similarity namun dia pakai module)

Rabbani, H. A. (2018, September 24). PySastrawi. Retrieved November 10, 2020, from <https://github.com/har07/PySastrawi>

(Referensi penggunaan dan syntax Sastrawi)

Yulio, A. (2018, October 07). Stopword Removal Bahasa Indonesia dengan Python Sastrawi. Retrieved November 10, 2020, from <https://devtrik.com/python/stopword-removal-bahasa-indonesia-python-sastrawi/>

(Referensi aplikasi Sastrawi).

Joshi, R. (202, May 22). Word-count-flask. Retrieved November 11, 2020, from <https://github.com/raghavjoshi789/word-count-flask>

(Referensi aplikasi Web Scraping)