

# **APLIKASI DOT PRODUCT PADA SISTEM TEMU-BALIK INFORMASI: PROGRAM SEARCH ENGINE SEDERHANA**

## **LAPORAN TUGAS BESAR 2**

Diajukan sebagai salah satu tugas besar  
mata kuliah IF2123 Aljabar Linier dan Geometri  
pada Semester 1 Tahun Akademik 2020-2021

Oleh:

Rais Vaza Man Tazakka (13519060)

Siti Iedrania Azzariyat Akbar (13519137)

Karina Imani (13519166)



**PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG  
BANDUNG  
2020**

## **BAB 1**

### **DESKRIPSI MASALAH**

Harus dibuat sebuah mesin pencarian dengan sebuah website lokal sederhana dengan spesifikasi 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. Sedikitnya, terdapat 15 dokumen berbeda sebagai kandidat dokumen. Penggunaan web scraping untuk mengekstraksi dokumen dari website akan menjadi bonus.
3. Hasil pencarian yang terurut berdasarkan similaritas tertinggi dari hasil teratas hingga hasil terbawah berupa judul dokumen dan kalimat pertama dari dokumen tersebut. Nilai similaritas tiap dokumen disertakan.
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. Fitur fungsional lain yang menunjang program bisa ditambahkan.
7. Program harus modular dan mengandung komentar yang jelas.
8. Dilarang menggunakan library cosine similarity yang sudah jadi.

## BAB 2

### TEORI SINGKAT

#### I. Sistem Temu Balik Informasi

Temu balik informasi (*information retrieval*) adalah aktivitas menemukan kembali informasi-informasi yang relevan terhadap kebutuhan pengguna dari suatu kumpulan informasi. Temu balik informasi dibedakan dari pencarian dalam basis data (*database*) karena umumnya digunakan untuk mencari informasi yang tidak terstruktur, seperti dokumen dan laman web (*webpage*).

Sistem temu balik informasi (*information retrieval system*) melakukan *retrieval* secara otomatis berdasarkan *query* yang diberikan pengguna. Salah satu aplikasi umum dari sistem temu balik informasi adalah *search-engine* atau mesin pencarian yang mengembalikan alamat-alamat laman web terkait, dan salah satu implementasinya adalah dengan menggunakan *cosine similarity*.

#### II. Teori Dasar Vektor

Vektor adalah kuantitas fisik yang memiliki besar dan arah, ditulis dengan notasi:

$$\vec{v} = (v_1, v_2, \dots, v_n)$$

untuk suatu vektor dalam ruang  $R^n$ . Sebagai contoh, vektor dalam  $R^2$  dan  $R^3$  adalah:

- $R^2$  :  $(v_1, v_2)$
- $R^3$  :  $(v_1, v_2, v_3)$

Salah satu karakteristik vektor yang dimanfaatkan dalam sistem temu balik informasi menggunakan *cosine similarity* adalah sudut antara dua vektor. Sudut antara dua vektor dapat diperoleh menggunakan rumus sebagai berikut:

$$\cos \theta = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$$

dengan  $\mathbf{u}$  dan  $\mathbf{v}$  berupa vektor di ruang yang sama,  $\mathbf{u} \cdot \mathbf{v}$  *dot product* kedua vektor, dan  $\|\mathbf{u}\|$  dan  $\|\mathbf{v}\|$  *magnitude* atau panjang tiap vektor. Adapun, komponen-komponen rumus tersebut dapat dihitung dengan cara berikut:

Komponen	Rumus
$\mathbf{u} \cdot \mathbf{v}$ ( <i>dot product</i> )	$u \cdot v = u_1 v_1 + u_2 v_2 + \dots + u_n v_n$
$\ \mathbf{u}\ $ ( <i>magnitude</i> )	$\ u\  = \sqrt{u_1^2 + u_2^2 + \dots + u_n^2}$

Jika  $\mathbf{u}$  dan  $\mathbf{v}$  adalah vektor tidak-nol dan  $\theta$  adalah sudut antara kedua vektor, maka jika:

1.  $\mathbf{u} \cdot \mathbf{v} > 0$  :  $\theta$  adalah sudut lancip ( $0^\circ < \theta < 90^\circ$ )
2.  $\mathbf{u} \cdot \mathbf{v} = 0$  :  $\theta$  tepat  $90^\circ$
3.  $\mathbf{u} \cdot \mathbf{v} < 0$  :  $\theta$  adalah sudut tumpul ( $90^\circ < \theta < 180^\circ$ )

### III. Metode *Cosine Similarity*

Salah satu cara mencari hasil-hasil pencarian yang relevan dengan suatu query adalah mencari jumlah kata yang cocok antara query dan dokumen sebanyak mungkin. Namun, cara ini memiliki kelemahan, yaitu semakin besar ukuran dokumen dan semakin banyak jumlah katanya, mesin akan menganggap dokumen tersebut lebih cocok, meskipun topik dokumen tersebut malah lebih tidak relevan. Metode *cosine similarity* memperbaiki kelemahan metode ini.

Metode *cosine similarity* membandingkan kecocokan dokumen dengan query tanpa memperhatikan ukurannya. Metode ini memanfaatkan vektor untuk mencari sudut antara dua vektor di ruang  $R^n$ , dengan  $n$  sebagai jumlah *term* yang ada. Karena metode ini memanfaatkan ‘arah’ kedua vektor dan bukan ‘magnitudo’ mereka, metode ini cocok untuk menghitung kesamaan dokumen dengan query tanpa memperhatikan ukuran atau banyak kata masing-masing.

Adapun, rumus *cosine similarity* atau sim antara query  $Q$  dan dokumen  $D$  dapat dihitung menggunakan rumus:

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

Vektor dari query dan dokumen dapat ditentukan berdasarkan jumlah kata pada setiap term unik. Untuk mencari kemiripan dengan query, dapat digunakan jumlah term unik di query saja. Sebagai contoh:

**Q: Japanese feudal lords**

**D1:** Daimyo were powerful **Japanese feudal lords** who, from the 10th century to the early Meiji period in the middle 19th century, ruled most of **Japan** from their vast, hereditary land holdings. (<https://en.wikipedia.org/wiki/Daimyo>)

**D2:** The Sengoku period is a period in **Japanese** history of near-constant civil war, social upheaval, and political intrigue from 1467 to 1615. The Sengoku period was initiated by the Ōnin War in 1467 which collapsed the **feudal** system of **Japan** under the Ashikaga Shogunate. ([https://en.wikipedia.org/wiki/Sengoku\\_period](https://en.wikipedia.org/wiki/Sengoku_period))

Query yang diterima, dan dokumen yang ada, terlebih dahulu ‘dibersihkan’ dengan *stemming* dan penghapusan *stop words*:

- Stemming : mengubah kata menjadi bentuk dasar, misal *memakan* → *makan*.
- Stop words : kata-kata umum yang sering muncul, misal *dan, atau, tapi, akan*.

Setelah dibersihkan, dihitung jumlah kata dalam tiap dokumen yang sama dengan kata pada query. Contohnya, *term* query adalah **Japan**, **feud**, dan **lord**, maka jumlah tiap *term* yang cocok adalah:

Term	Q	D1	D2
Japan	1	2	2
feudal	1	1	1
lord	1	1	0

Berdasarkan tabel, diperoleh vektor sebagai berikut:

1. **Q** = (1, 1, 1)
2. **D1** = (2, 1, 1)
3. **D2** = (2, 1, 0)

Menggunakan rumus-rumus yang telah dijabarkan di atas, dihitung nilai sim antara query dengan masing-masing dokumen, yakni  $\text{sim}(\mathbf{Q}, \mathbf{D1})$  dan  $\text{sim}(\mathbf{Q}, \mathbf{D2})$ :

	D1	D2
<b>Q·D</b> ( <i>dot product</i> )	$2 + 1 + 1 = 4$	$2 + 1 = 3$
<b>  Q     D  </b> ( <i>magnitude</i> )	4.24	3.87
<b>sim(Q, D)</b> ( <i>cosine similarity</i> )	0.94	0.77

Karena nilai  $\cos \theta$  terkecil adalah 0 dan terbesar adalah 1 ( $0 \leq \cos \theta \leq 1$ ), maka nilai cosinus yang semakin besar mengindikasikan kecocokan kedua dokumen. Semakin berimpit dua vektor, semakin serupa dokumen-dokumen yang direpresentasikannya.

Kemudian, berdasarkan hasil yang diperoleh, dapat dilihat bahwa dokumen D1 lebih cocok kepada query dibandingkan dokumen D2, karena memiliki nilai sim yang lebih besar ( $\text{sim}(\mathbf{Q}, \mathbf{D1}) < \text{sim}(\mathbf{Q}, \mathbf{D2})$ ).

## BAB 3

### IMPLEMENTASI PROGRAM

#### 1. Web Scraping

Pada program *web scraping*, implementasi spesifik pada artikel-artikel yang terdapat pada situs *kompas.com*. Digunakan *library* *urlopen* yang terdapat pada *package* *urllib.request* untuk mengolah URL dan *library* *BeautifulSoup* dari *package* *bs4* untuk menyaring dan mengekstraksi konten dari *website*. Proses ekstraksi dibedakan untuk judul dan konten artikel. Judul artikel diperoleh dengan mengidentifikasi elemen html yang memiliki *class* “*read\_content*”. Konten artikel diperoleh dengan mengidentifikasi elemen html yang memiliki *class* “*read\_content*”. Dalam dokumen tersebut, dicari elemen-elemen yang memiliki *tag* “*p*”, “*h2*”, “*h3*”, “*ol*”, “*ul*”, dan “*table*”. Untuk menyaring isi konten dari iklan dan media (seperti foto), identifikasi konten dibuat lebih spesifik lagi dengan menghilangkan elemen-elemen yang memiliki *class* “*inner-link-baca-juga*” atau “*photo*”. Kemudian, judul dan konten disatukan dan disimpan sebagai dokumen dalam folder “*src/static/*” dengan ekstensi “.txt”.

#### 2. Stemming dan Remove Stop Words

Karena program yang dibuat adalah untuk artikel-artikel berbahasa Indonesia, untuk proses *stemming* dan penghapusan *stopwords* digunakan *library* *Sastrawi*. Masing-masing diimplementasikan dalam fungsi *stem(dokumen)* dan *removeStopWord(dokumen)* yang mengambil suatu argumen artikel berupa string dan mengembalikan string juga.

#### 3. Document Database

Document database memiliki 5 komponen, yaitu nama file dokumen, judul dokumen, isi dari dokumen, paragraf pertama dokumen, dan jumlah kata pada dokumen. Document database disimpan dan diolah dalam *database.py*. Di dalamnya terdapat tiga fungsi berikut.

1. *isiKontenDokumen(dokumen)*

Memasukkan body dokumen ke array *kontenDokumen*. Parameter berupa string hasil *readline*.

2. *isiJumlahKataDokumen(judulDokumen, konten)*

Memasukkan jumlah kata dokumen ke array *jumlahKataDokumen* dengan menambahkan kata yang terdapat di judul dan konten. Parameter berupa string.

3. *newFileIn(direktori, filename)*

Fungsi ini menambahkan data baru ke database setiap kali ada dokumen baru yang di-upload. Isinya berupa *append* hasil pembacaan baru ke array lama dan mengisi ulang array database. Parameter berupa string *directory* dan *filename*.

Selanjutnya, kelima array tersebut dihimpun menjadi satu array database.

#### 4. Term Database

Database diimplementasikan dalam bentuk matriks (*array of array*) yang berisi frekuensi kemunculan setiap term yang terdapat pada seluruh dokumen pada

masing-masing dokumen. Term database menerima input berupa document database (matriks) dan mengembalikan output berupa matriks.

Untuk pengisian dan penampilan tabel digunakan tiga fungsi, yaitu:

1. `tabelVektor(database)`

Menerima database yang mengandung isi dari dokumen, lalu membuat tabel yang berisi frekuensi kemunculan term di setiap dokumen. Di dalamnya menggunakan fungsi-fungsi lain untuk mendapatkan setiap term dari isi dokumen.

2. `getNamaJudul(database)`

Digunakan untuk membantu output tabel dengan mengambil nama dan judul dokumen dari database, agar nama dokumen yang panjang dapat dilihat dengan lebih baik.

3. `hapusHeader(tabel)`

Berfungsi untuk mengambil bagian body dari tabel, agar dapat dilakukan pemrosesan pada tabel untuk output.

## 5. Fungsi Cosine Similarity

Nilai similarity dihitung saat sudah mendapatkan query. Fungsi ini diimplementasikan dalam bentuk `tabelSimQuery(database, query)` yang menghitung kemunculan setiap term pada query di semua dokumen menggunakan pengulangan, dengan input berupa array of array (database dokumen) dan string query. Lalu akan dilakukan penghitungan sehingga didapatkan tabel yang berisi nilai similarity untuk setiap dokumen.

## 6. Fungsi Tabel Lainnya

Untuk membantu mengatur data-data dalam tabel, digunakan beberapa fungsi tambahan sebagai berikut.

1. `sortBySim(tabel)`

Parameter tabel berupa tabel data dengan array nilai sim sebagai elemen terakhir tabel. Mengurutkan seluruh tabel berdasarkan sim dari yang terbesar sampai terkecil.

2. `dataByQuery(database, tabel)`

Parameter database berupa tabel database yang berisi keterangan seperti nama file, judul dokumen, dan banyak kata. Parameter tabel berupa tabel jumlah term query dan semua dokumen, dengan sim sebagai elemen terakhir tabel. Fungsi ini membentuk tabel yang siap dicetak ke HTML.

3. `tabelDisplay(tabel)`

Digunakan untuk menampilkan bagian yang diperlukan dari tabel similarity, yaitu dengan menghilangkan baris terakhirnya. Input berupa array of array dan output sama.

4. `compactTable(tabel)`

Mengembalikan tabel yang telah dipangkas dari baris tabel yang kosong, yaitu yang seluruh anggotanya adalah bilangan nol. Input berupa array of array dan output sama.

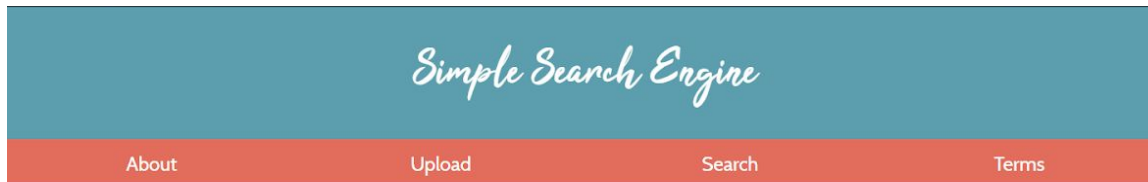
## 7. Implementasi Flask

Untuk menghubungkan antara Python sebagai back-end dan HTML sebagai front-end, digunakan framework Flask, yang memanfaatkan *Jinja* dan *Werkzeug WSGI toolkit*. Adapun, implementasi Flask adalah:

- method = 'GET'  
Berfungsi menerima data dari web server.
- method = 'POST'  
Berfungsi mengirim input ke web server.
- def home(), def about(), def()  
Berfungsi untuk me-render template HTML bersangkutan, yakni main.html dan about.html secara berturut-turut.
- def home\_post()  
Fungsi yang dijalankan ketika home menerima instruksi 'POST', yakni ketika input dimasukkan ke *search box*. Berfungsi untuk redirect ke halaman result/<query>.
- def result(q)  
Berfungsi menampilkan result berdasarkan query q yang menjadi parameter fungsi tersebut. Di akhir fungsi, akan me-render halaman result/<query> dengan isi yang akan ditampilkan Jinja sebagai list *result*, dan tabel kemunculan kata yang akan ditampilkan di bawahnya.
- def upload()  
Fungsi yang dijalankan ketika upload menerima instruksi 'POST', yakni ketika user mengunggah file .txt ke server. Memeriksa apakah nama file bukan string kosong dan apakah *extension* file .txt.
- def isAllowed(filename)  
Fungsi khusus untuk memeriksa apakah *extension* file diperbolehkan (termasuk dalam array *allowed file extensions*), dalam hal ini hanya .txt.
- def terms()  
Berfungsi menampilkan jumlah kemunculan kata setiap term unik di setiap dokumen yang ada di database.

## 8. HTML dan CSS

Aplikasi berbasis web diimplementasikan menggunakan HTML, yang tampilan umumnya memiliki *title bar* dan *navigation bar* di sisi atas halaman.



Adapun, template isi website dibagi menjadi beberapa file HTML, yaitu:

1. main.html



Berisi *search box* yang dapat menerima input pengguna, dan *submit button* yang berfungsi mengirim input tersebut ke back end.

2. `about.html`

Berisi keterangan singkat *search engine* dan data diri (nama, NIM) pembuatnya.

3. `upload.html`

Berisi form *upload file* yang menerima file dengan *extension* .txt dan mengunduh file tersebut ke direktori yang telah ditentukan.

4. `result.html`

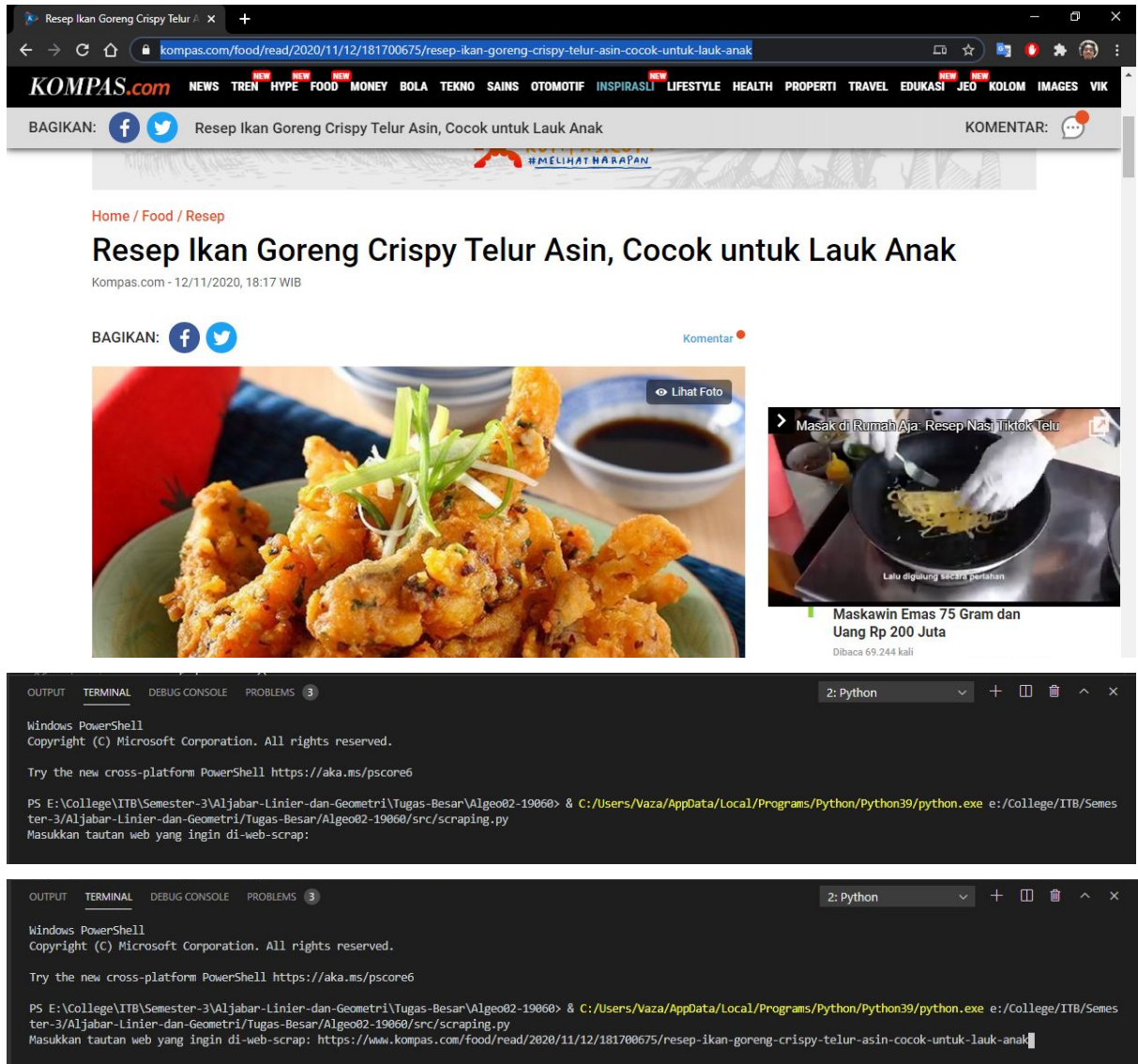
Halaman yang tidak dapat diakses dari *navigation bar*, selain dengan melakukan input query ke *search box*. Menampilkan hasil pencarian *query*

5. `terms.html`

Berisi tabel perhitungan jumlah semua term unik yang muncul di semua dokumen yang ada pada database.

## BAB 4 EKSPERIMEN

### 1. Web Scraping



The image displays a web browser window showing a recipe article on Kompas.com. The article title is "Resep Ikan Goreng Crispy Telur Asin, Cocok untuk Lauk Anak" (Recipe for Crispy Fried Fish with Salted Egg, Suitable for Children's Side Dish). The article is dated 12/11/2020, 18:17 WIB. Below the title, there is a social media share button for Facebook and Twitter, and a comment button. The main image shows a plate of fried fish with a garnish of green onions and a bowl of dipping sauce. To the right of the main image, there is a video thumbnail titled "Masak di Rumah Aja: Resep Nasi Tikiok Telur" (Cooking at Home: Recipe for Nasi Tikiok Telur) with a subtitle "Lalu digalung sedikit paku" (Then hang a little fern). Below the video thumbnail, there is a text overlay: "Maskawin Emas 75 Gram dan Uang Rp 200 Juta" (75 Gram Gold Maskawin and Rp 200 Million Cash) and "Dibaca 69.244 kali" (Read 69,244 times).

Below the browser window, there is a terminal window showing the execution of a web scraping script. The terminal output is as follows:

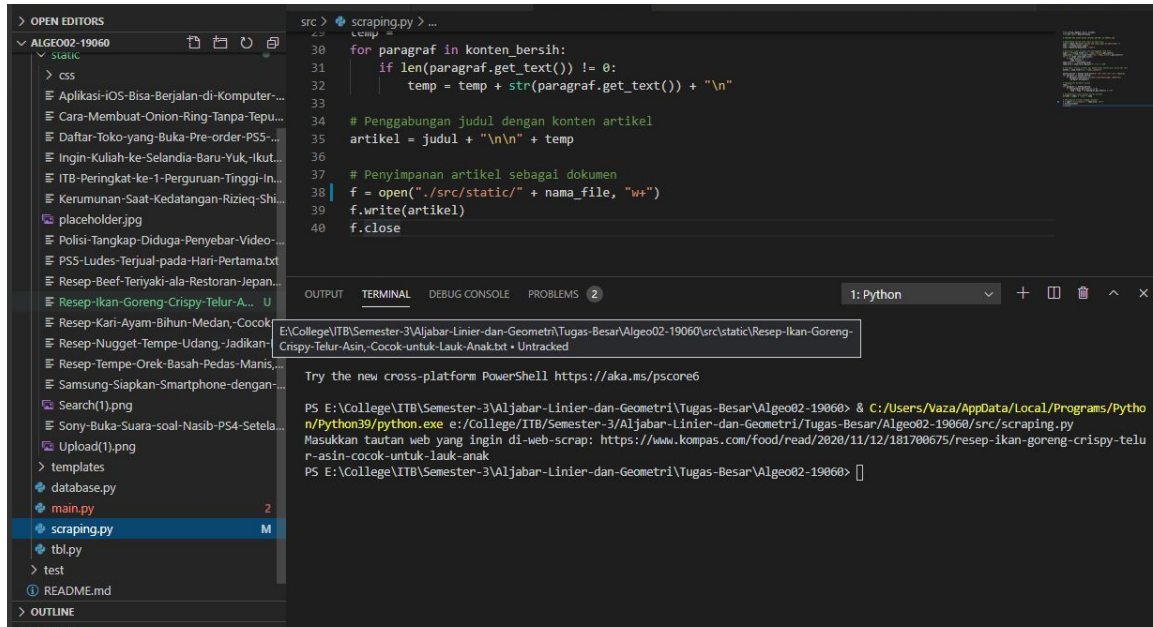
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\College\ITB\Semester-3\Aljabar-Linier-dan-Geometri\Tugas-Besar\Algeo02-19060> & C:/Users/Vaza/AppData/Local/Programs/Python/Python39/python.exe e:/College/ITB/Semes
ter-3/Aljabar-Linier-dan-Geometri/Tugas-Besar/Algeo02-19060/src/scraping.py
Masukkan tautan web yang ingin di-web-scrap:
```

The terminal window shows the execution of a Python script using the command prompt. The script is located at `e:/College/ITB/Semester-3/Aljabar-Linier-dan-Geometri/Tugas-Besar/Algeo02-19060/src/scraping.py`. The prompt indicates that the user should enter the URL of the website to be scraped.

File hasil *scraping* terdapat dalam folder “static” (di-highlight warna hijau)



```
src > scraping.py > ...
30 for paragraph in konten bersih:
31     if len(paragraf.get_text()) != 0:
32         temp = temp + str(paragraf.get_text()) + "\n"
33
34 # Penggabungan judul dengan konten artikel
35 artikel = judul + "\n\n" + temp
36
37 # Penyimpanan artikel sebagai dokumen
38 f = open("./src/static/" + nama_file, "w")
39 f.write(artikel)
40 f.close

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS 2 1: Python
E:\College\ITB\Semester-3\Aljabar-Linier-dan-Geometri\Tugas-Besar\Algeo02-19060\src\static\Resep-Ikan-Goreng-Crispy-Telur-Asin, Cocok-untuk-Lauk-Anak.txt • Untracked

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\College\ITB\Semester-3\Aljabar-Linier-dan-Geometri\Tugas-Besar\Algeo02-19060> & C:/Users/Vaza/AppData/Local/Programs/Python/Python39/python.exe e:/College/ITB/Semester-3/Aljabar-Linier-dan-Geometri/Tugas-Besar/Algeo02-19060/src/scraping.py
Masukkan tautan web yang ingin di-web-scrap: https://www.kompas.com/food/read/2020/11/12/181700675/resep-ikan-goreng-crispy-telur-asin-cocok-untuk-lauk-anak
PS E:\College\ITB\Semester-3\Aljabar-Linier-dan-Geometri\Tugas-Besar\Algeo02-19060> []
```

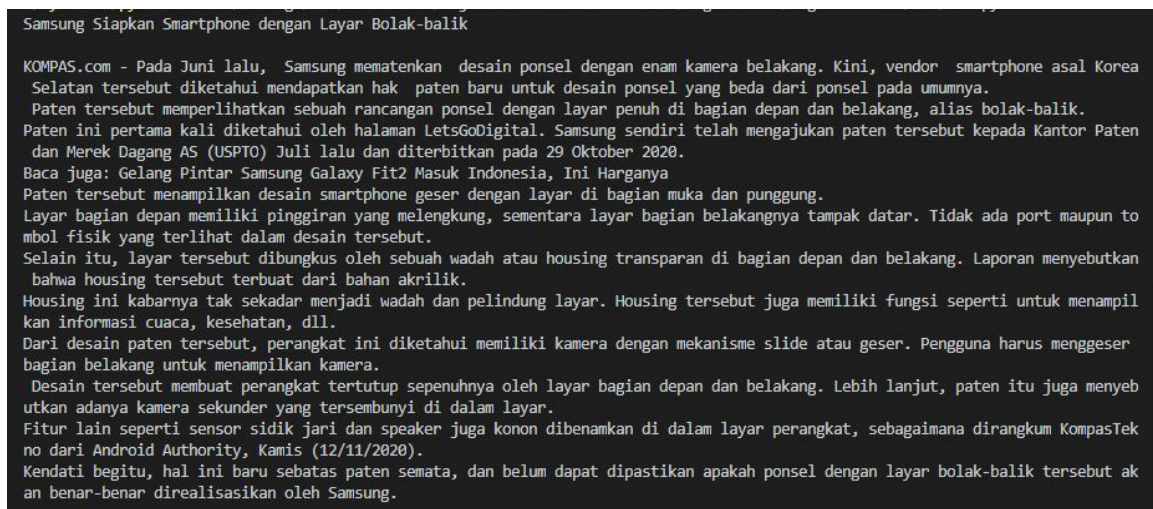
Scraping pada situs <https://www.kompas.com/food/read/2020/11/12/181700675/resep-ikan-goreng-crispy-telur-asin-cocok-untuk-lauk-anak> berhasil dilakukan. Dokumen hasil ekstraksi disimpan dalam folder “static” dan ter-highlight hijau seperti pada gambar di atas.

## 2. Stemming dan Remove Stop Words



```
293
294 f = open("./src/static/Samsung-Siapkan-Smartphone-dengan-Layar-Bolak-balik.txt", "r")
295 artikel = f.read()
296 print(artikel)
```

Artikel asli



Samsung Siapkan Smartphone dengan Layar Bolak-balik

KOMPAS.com - Pada Juni lalu, Samsung mematenkan desain ponsel dengan enam kamera belakang. Kini, vendor smartphone asal Korea Selatan tersebut diketahui mendapatkan hak paten baru untuk desain ponsel yang beda dari ponsel pada umumnya. Paten tersebut memperlihatkan sebuah rancangan ponsel dengan layar penuh di bagian depan dan belakang, alias bolak-balik. Paten ini pertama kali diketahui oleh halaman LetsGoDigital. Samsung sendiri telah mengajukan paten tersebut kepada Kantor Paten dan Merek Dagang AS (USPTO) Juli lalu dan diterbitkan pada 29 Oktober 2020.

Baca juga: Gelang Pintar Samsung Galaxy Fit2 Masuk Indonesia, Ini Harganya

Paten tersebut menampilkan desain smartphone geser dengan layar di bagian muka dan punggung. Layar bagian depan memiliki pinggirannya yang melengkung, sementara layar bagian belakangnya tampak datar. Tidak ada port maupun tombol fisik yang terlihat dalam desain tersebut.

Selain itu, layar tersebut dibungkus oleh sebuah wadah atau housing transparan di bagian depan dan belakang. Laporan menyebutkan bahwa housing tersebut terbuat dari bahan akrilik.

Housing ini kabarnya tak sekadar menjadi wadah dan pelindung layar. Housing tersebut juga memiliki fungsi seperti untuk menampilkan informasi cuaca, kesehatan, dll.

Dari desain paten tersebut, perangkat ini diketahui memiliki kamera dengan mekanisme slide atau geser. Pengguna harus menggeser bagian belakang untuk menampilkan kamera.

Desain tersebut membuat perangkat tertutup sepenuhnya oleh layar bagian depan dan belakang. Lebih lanjut, paten itu juga menyebutkan adanya kamera sekunder yang tersembunyi di dalam layar.

Fitur lain seperti sensor sidik jari dan speaker juga konon dibenamkan di dalam layar perangkat, sebagaimana dirangkum KompasTekno dari Android Authority, Kamis (12/11/2020).

Kendati begitu, hal ini baru sebatas paten semata, dan belum dapat dipastikan apakah ponsel dengan layar bolak-balik tersebut akan benar-benar direalisasikan oleh Samsung.

Proses stemming



```

294 f = open("./src/static/Samsung-Siapkan-Smartphone-dengan-Layar-Bolak-balik.txt", "r")
295 artikel = f.read()
296 artikel = stem(artikel)
297 print(artikel)

```

samsung siap smartphone dengan layar bolak-balik kompas com - pada juni lalu samsung paten desain ponsel dengan enam kamera belakang kini vendor smartphone asal korea selatan sebut tahu dapat hak paten baru untuk desain ponsel yang beda dari ponsel pada umumnya paten sebut lihat buah rancang ponsel dengan layar penuh di bagian depan dan belakang alias bolak-balik paten ini pertama kali tahu oleh halaman letsdigital samsung sendiri telah ajukan paten sebut kepada kantor paten dan merek dagang as uspto juli lalu dan terbit pada 29 oktober 2020 baca juga gelang pintar samsung galaxy fit2 masuk indonesia ini harga paten sebut tampil desain smartphone geser dengan layar di bagian muka dan punggung layar bagian depan milik pinggir yang lengkung sementara layar bagian belakang tampak datar tidak ada port maupun tombol fisik yang lihat dalam desain sebut selain itu layar sebut bungkus oleh buah wadah atau housing transparan di bagian depan dan belakang laporan sebut bahwa housing sebut buat dari bahan akrilik housing ini kabar tak sekadar jadi wadah dan lindung layar housing sebut juga milik fungsi seperti untuk tampil informasi cuaca sehat dll dari desain paten sebut perangkat ini tahu milik kamera dengan mekanisme slide atau geser guna harus geser bagian belakang untuk tampil kamera desain sebut buat perangkat tutup sepenuhnya oleh layar bagian depan dan belakang lebih lanjut paten itu juga sebut ada kamera sekunder yang sembunyi di dalam layar fitur lain seperti sensor sidik jari dan speaker juga konon benam di dalam layar perangkat bagian mana rangkum kompastekno dari android authority Kamis 12 11 2020 kendati begitu hal ini baru batas paten semata dan belum dapat pasti apakah ponsel dengan layar bolak-balik sebut akan benar realisasi oleh samsung

### Proses penghapusan stopwords

```

294 f = open("./src/static/Samsung-Siapkan-Smartphone-dengan-Layar-Bolak-balik.txt", "r")
295 artikel = f.read()
296 artikel = stem(artikel)
297 artikel = removeStopWord(artikel)
298 print(artikel)

```

samsung smartphone layar bolak-balik kompas com - juni samsung paten desain ponsel enam kamera kini vendor smartphone korea selatan tahu hak paten desain ponsel beda ponsel umum paten lihat buah rancang ponsel layar penuh alias bolak-balik paten pertama tahu halaman letsdigital samsung telah ajukan paten kepada kantor paten merek dagang as uspto juli terbit 29 oktober 2020 baca gelang pintar samsung galaxy fit2 indonesia harga paten tampil desain smartphone geser layar muka punggung layar milik pinggir lengkung layar datar ada port tombol fisik lihat desain selain layar bungkus buah wadah housing transparan bagian depan belakang laporan bahwa housing buat bahan akrilik housing kabar sekadar wadah dan lindung layar housing milik fungsi untuk tampil informasi cuaca sehat dll desain paten perangkat tahu milik kamera mekanisme slide geser harus geser bagian belakang untuk tampil kamera desain buat perangkat tutup sepenuhnya layar bagian depan dan belakang lanjut paten juga ada kamera sekunder sembunyi dalam layar fitur seperti sensor sidik jari dan speaker juga konon benam dalam layar perangkat rangkum kompastekno android authority Kamis 12 11 2020 kendati hal baru batas paten dan dapat apakah ponsel layar bolak-balik akan realisasi samsung

Stemming dan penghapusan stopwords pada artikel di atas berhasil dilakukan. Stemming mengubah kata-kata ke bentuk dasarnya. Penghapusan stopwords menghapus kata-kata yang dianggap tidak signifikan untuk perhitungan.

### 3. Document Database

```
>>> data = database.database
>>> for i in range (len(data)):
    print(data[i][0])
```

Aplikasi-iOS-Bisa-Berjalan-di-Komputer-Mac-Terbaru.txt  
Aplikasi iOS Bisa Berjalan di Komputer Mac Terbaru

KOMPAS.com - Apple resmi meluncurkan tiga perangkat Mac terbarunya yang dite-  
nagai chip M1 berbasis ARM yakni MacBook Air, MacBook Pro 13 inci, dan Mac M  
ini. Dalam acara yang sama, Apple juga memastikan tanggal rilis sistem operas-  
i MacOS Big Sur.

Sistem operasi tersebut bisa mulai diunduh pengguna perangkat Mac mulai 12 No-  
vember atau esok hari. Apple mengatakan, sistem operasi Big Sur dalam perangk-  
at Mac terbarunya, akan mendukung universal app. Artinya, aplikasi iPhone dan  
iPad bisa dijalankan melalui perangkat Mac.

Dalam acara peluncuran tersebut, Apple pun memamerkan kemampuan baru ini mela-  
lui cuplikan video. Video tersebut menampilkan aplikasi iPhone dan iPad seper-  
ti HBO Max dan Among Us, dapat berjalan di ketiga perangkat Mac terbarunya.  
Aplikasi tersebut dihadirkan di Mac App Store bersama dengan jajaran aplikasi  
macOS lainnya.

Apple menjelaskan, teknologi baru yang disebut Rosetta 2 membantu chip M1 yan-  
g menjadi otak dari perangkat Mac terbaru untuk menjalankan aplikasi yang dib-  
uat untuk Mac berbasis Intel.

<bagian isi artikel dipotong untuk mempermudah pembacaan>

Aplikasi universal itu merupakan besutan dari dua perusahaan pengembangan sof-  
tware yakni Omni Group dan Adobe.

Nantinya, Adobe akan menawarkan versi universal untuk aplikasi Lightroom pada  
Desember 2020 dan Photoshop di awal 2021 mendatang.

Sebagaimana dihimpun KompasTekno dari Tech Crunch, Rabu (11/11/2020), aplikas-  
i universal ini dapat diunduh melalui toko App Store baik di smartphone maupu-  
n di web.

KOMPAS.com - Apple resmi meluncurkan tiga perangkat Mac terbarunya yang dite-  
nagai chip M1 berbasis ARM yakni MacBook Air, MacBook Pro 13 inci, dan Mac M  
ini. Dalam acara yang sama, Apple juga memastikan tanggal rilis sistem operas-  
i MacOS Big Sur.

```
65
>>> |
```

Contoh isi dari database dokumen, yaitu data dari dokumen 1 (atas ke bawah: nama,  
judul, isi, paragraf pertama, jumlah kata).

Disini dimanfaatkan fungsi built-in readline dan readlines untuk membaca bagian-bagian  
dari dokumen, lalu data di-append ke array yang bersesuaian.

#### 4. Term Database

```

>>> for i in range (len(data)):
        print(data[i])

['dokumen1.txt', 'dokumen2.txt']
['Judul Dokumen 1', 'Judul Dokumen 2']
['Judul Dokumen 1 Ini adalah kalimat pertama dokumen 1. Ini adalah kalimat
kedua dokumen 1.', 'Judul Dokumen 1 Ini adalah kalimat pertama dokumen 2.
Ini adalah kalimat kedua dokumen 2.']
['Ini adalah kalimat pertama dokumen 1.', 'Ini adalah kalimat pertama doku
men 2.']
[42, 69]
>>> hasil = tabelVektor(data)
>>> for i in range (len(hasil)):
        print(hasil[i])

['Term', 'dokumen1.txt', 'dokumen2.txt']
['judul', 1, 1]
['dokumen', 3, 3]
['1', 3, 1]
['adalah', 2, 2]
['kalimat', 2, 2]
['pertama', 1, 1]
['2', 0, 2]
>>> |

```

Contoh pembuatan tabel kemunculan term dengan menggunakan database mini.

Waktu yang dibutuhkan untuk membuat tabel bertambah seiring dengan bertambahnya jumlah dokumen.

## 5. Fungsi Cosine Similarity



```

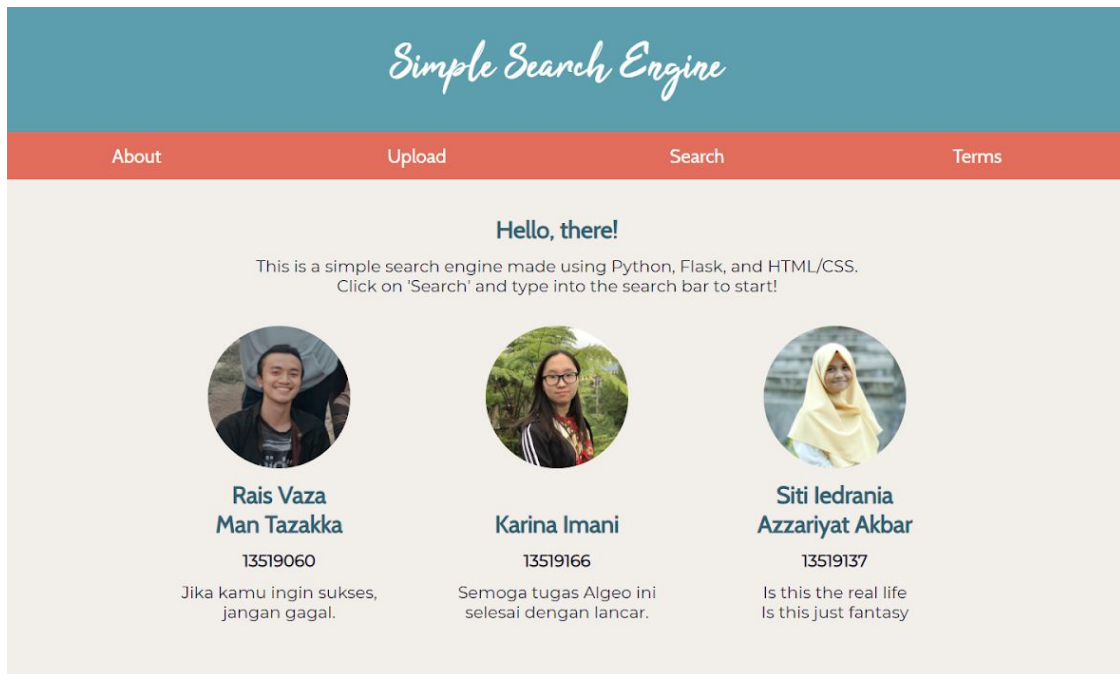
>>> data = database.database
>>> query = "onion korek iris bawang"
>>> tabel = tabelSimQuery(data, query)
>>> for i in range (len(tabel)):
    print(tabel[i])

['Term', 'Query', 'Aplikasi-iOS-Bisa-Berjalan-di-Komputer-Mac-Terbaru.txt', '
Cara-Membuat-Onion-Ring-Tanpa-Tepung-Roti,-Pakai-Terigu-Serbaguna-Saja.txt',
'Daftar-Toko-yang-Buka-Pre-order-PS5-di-Indonesia.txt', 'Ingin-Kuliah-ke-Sela
ndia-Baru-Yuk,-Ikut-Pameran-Pendidikan-Virtual-Ini.txt', 'ITB-Peringkat-ke-1-
Perguruan-Tinggi-Inovatif-2020-Versi-Kemenristek.txt', 'Kerumunan-Saat-Kedata
ngan-Rizieq-Shihab-Bisa-Jadi-Bahan-Evaluasi-Izin-Reuni-PA-212-di-Monas.txt',
'Polisi-Tangkap-Diduga-Penyebar-Video-Syur-Mirip-Artis-GA.txt', 'PS5-Ludes-Te
rjual-pada-Hari-Pertama.txt', 'Resep-Beef-Teriyaki-ala-Restoran-Jepang,-Bikin
-Sendiri-di-Rumah.txt', 'Resep-Kari-Ayam-Bihun-Medan,-Cocok-Disantap-Bersama-
Keluarga.txt', 'Resep-Nugget-Tempe-Udang,-Jadikan-Lauk-Makan-Malam.txt', 'Res
ep-Tempe-Orek-Basah-Pedas-Manis,-Lauk-ala-Warteg.txt', 'Samsung-Siapkan-Smart
phone-dengan-Layar-Bolak-balik.txt', 'Sony-Buka-Suara-soal-Nasib-PS4-Setelah-
PS5-Meluncur.txt']
['onion', 1, 0, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
['korek', 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
['iris', 1, 0, 2, 0, 0, 0, 0, 0, 0, 2, 0, 0, 4, 0]
['bawang', 1, 0, 3, 0, 0, 0, 0, 0, 0, 2, 3, 4, 3, 0]
[0, 0, 0, 0.7857142857142857, 0, 0, 0, 0, 0, 0, 0, 0.7071067811865475, 0.5, 0.5,
0.7844645405527362, 0, 0]
>>> |

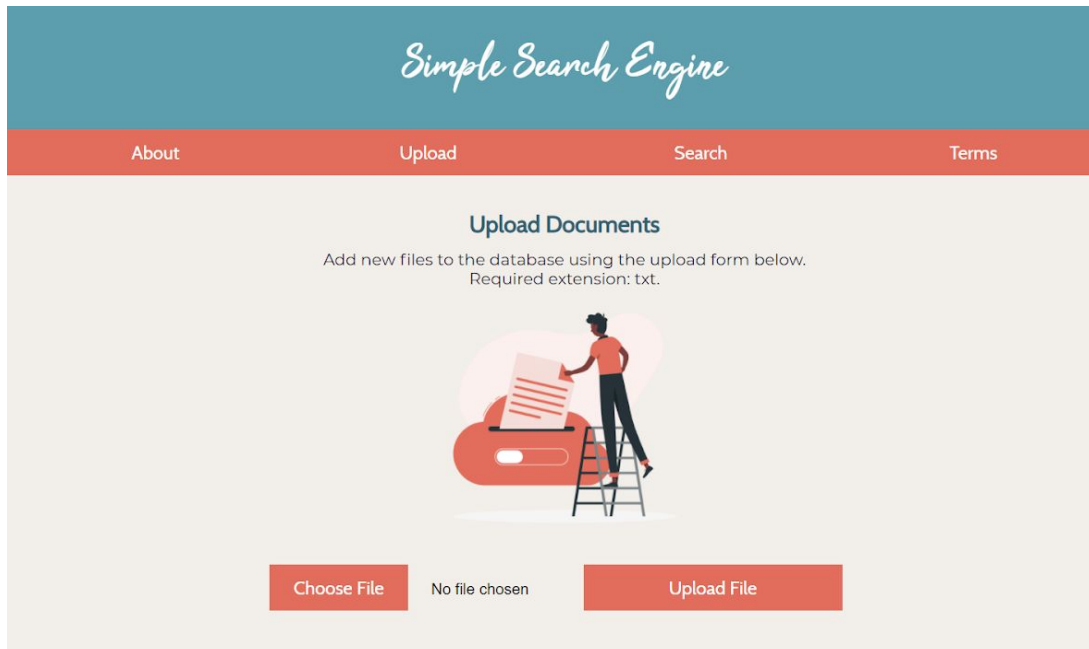
```

Contoh hasil dari fungsi `tabelSimQuery(tabel)`. Dapat dilihat bahwa kemunculan masing-masing term di dokumen didata dan dihitung nilai similaritasnya, lalu ditempatkan di bagian paling bawah array yang bersesuaian dengan masing-masing dokumen.

## 6. Webpage

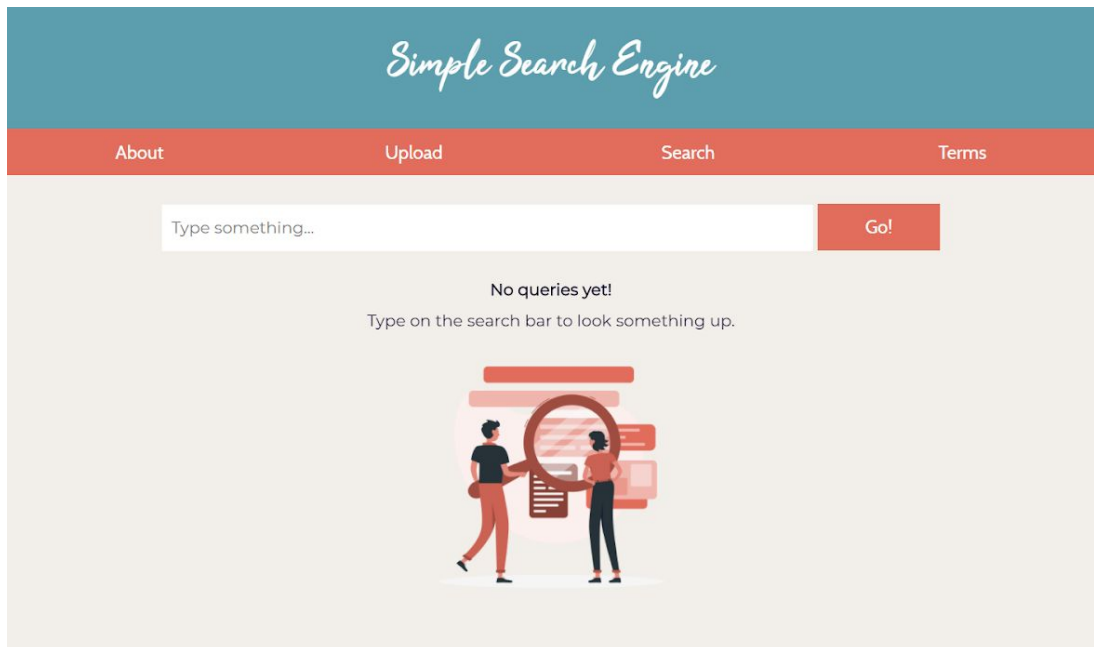


Layout halaman `about.html` (menu “About” pada *navigation bar*).



Layout halaman `upload.html` (menu “Upload” pada *navigation bar*).



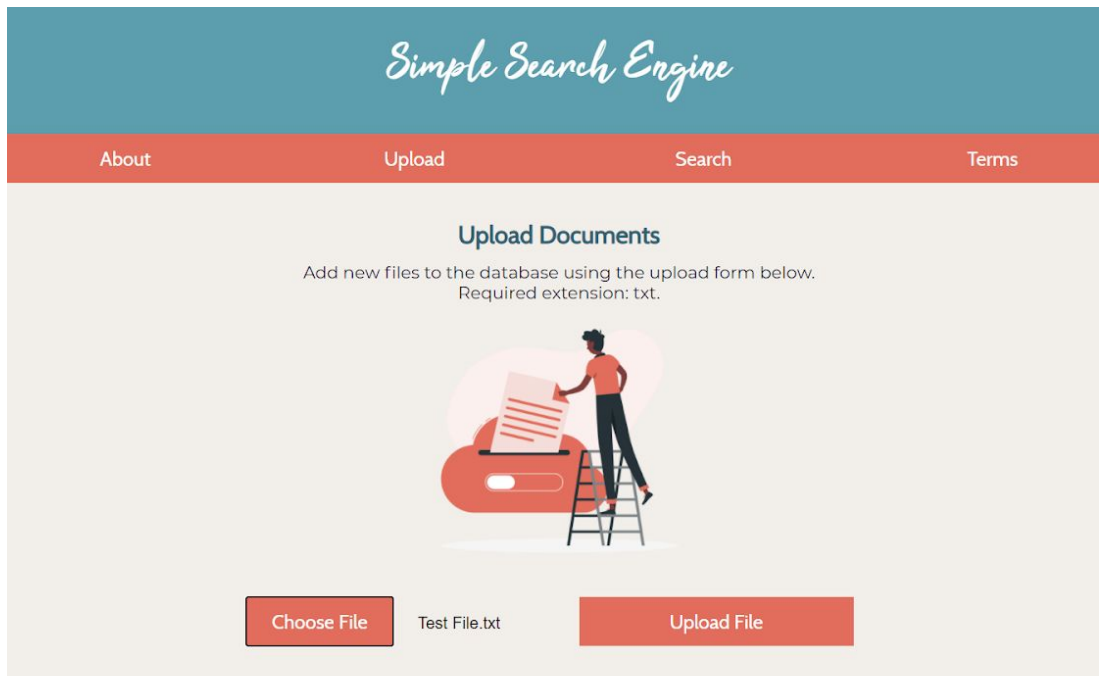


Layout halaman main.html (menu “Search” pada *navigation bar*).

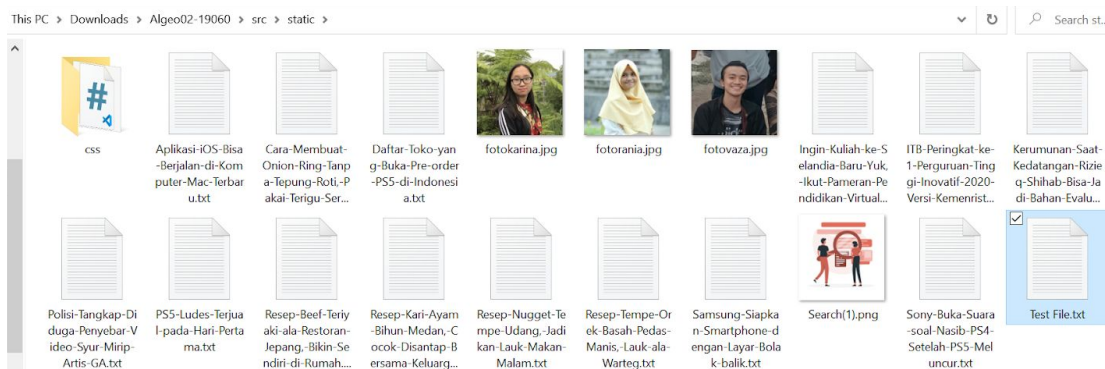
Term	D1	D2	D3	D4	D5	D6	D7
kompas	1	1	1	1	1	1	1
com	1	1	1	2	1	1	1
-	1	1	3	2	1	1	1
apple	8	0	0	1	0	0	0
resmi	1	0	3	0	0	0	0
luncur	2	0	1	0	0	0	0
tiga	2	0	0	1	0	1	0

Layout halaman terms.html (menu “Terms” pada *navigation bar*).

## 7. Upload File



Proses upload 'Test File.txt' ke direktori src/static yang dispesifikasi pada program.



File 'Test File.txt' sudah berada di direktori src/static.

```
127.0.0.1 - - [16/Nov/2020 09:25:41] "GET /upload/ HTTP/1.1" 200 -
['Aplikasi-iOS-Bisa-Berjalan-di-Komputer-Mac-Terbaru.txt', 'Cara-Membuat-Onion-Ring-Tanpa-Tepung-Roti-Pakai-Terigu-Serb
aguna-Saja.txt', 'Daftar-Toko-yang-Buka-Pre-order-PS5-di-Indonesia.txt', 'Ingin-Kuliah-ke-Selandia-Baru-Yuk-Ikut-Pamera
n-Pendidikan-Virtual-Ini.txt', 'ITB-Peringkat-ke-1-Perguruan-Tinggi-Inovatif-2020-Versi-Kemendiknas.txt', 'Kerumunan-Saa
t-Kedatangan-Rizieq-Shihab-Bisa-Jadi-Bahan-Evaluasi-Izin-Reuni-PA-212-di-Monas.txt', 'Polisi-Tangkap-Diduga-Penyebar-Vid
eo-Syur-Mirip-Artis-GA.txt', 'PS5-Ludes-Terjual-pada-Hari-Pertama.txt', 'Resep-Beef-Teriyaki-ala-Restoran-Jepang-Bikin
Sendiri-di-Rumah.txt', 'Resep-Kari-Ayam-Bihun-Medan-Cocok-Disantap-Bersama-Keluarga.txt', 'Resep-Nugget-Tempe-Udang-Ja
dikan-Lauk-Makan-Malam.txt', 'Resep-Tempe-Orek-Basah-Pedas-Manis-Lauk-ala-Warteg.txt', 'Samsung-Siapkan-Smartphone-deng
an-Layar-Bolak-balik.txt', 'Sony-Buka-Suara-soal-Nasib-PS4-Setelah-PS5-Meluncur.txt', 'Test File.txt']
127.0.0.1 - - [16/Nov/2020 09:25:48] "POST /upload/ HTTP/1.1" 302 -
```

Ini adalah hasil print database[0] setelah di-upload file test bernama "Test File.txt".

## 8. Query Result

# Simple Search Engine

[About](#)[Upload](#)[Search](#)[Terms](#)

## Search Results

Displaying results according to the highest sim. Click Search to input a new query.

[Cara Membuat Onion Ring Tanpa Tepung Roti, Pakai Terigu Serbaguna Saja](#)

92 words | Similarity: 78.57%

KOMPAS.com - Bawang bombai biasanya dijadikan bumbu masakan. Namun, kamu bisa berkreasi dengan membuat onion ring alias bombai goreng balut tepung.

[Resep Tempe Orek Basah Pedas Manis, Lauk ala Warteg](#)

75 words | Similarity: 78.45%

KOMPAS.com - Tempe bisa diolah dengan berbagai bahan makanan. Olahan tempe juga bisa jadi penyelamat saat tanggal tua. Coba saja bikin tempe orek basah.

[Resep Beef Teriyaki ala Restoran Jepang, Bikin Sendiri di Rumah](#)

88 words | Similarity: 70.71%

Result untuk *query* “onion korek iris bawang”. Dapat dilihat bahwa dokumen langsung ditampilkan secara terurut sesuai dengan nilai similaritasnya dan ditampilkan informasi tentang dokumen yang bersesuaian.

## **BAB 5**

### **KESIMPULAN**

#### **1. Kesimpulan**

Salah satu aplikasi dari konsep dot product pada vektor adalah mesin pencarian (*search engine*) dengan menghitung *cosine similarity* antara *query* dengan dokumen-dokumen yang tersedia pada basis data (*database*) mesin pencarian tersebut.

Search engine berbasis *local server* dapat dibuat dengan memanfaatkan Python sebagai *back-end*, HTML dan CSS sebagai *front-end*, dan dihubungkan dengan Flask, Jinja2, dan Werkzeug. Library Sastrawi dapat digunakan untuk melakukan *stemming* dan penghapusan *stopwords* dalam Bahasa Indonesia, dan database dokumen dapat diperoleh menggunakan *web scraping*.

#### **2. Saran**

Penulis menemukan bahwa program ini masih dapat dikembangkan dalam berbagai aspek, seperti pengolahan data dengan cepat, penambahan fitur-fitur, dan penampilan informasi. Penulis berharap hal ini dapat menjadi pertimbangan untuk pengembangan kedepannya.

Penulis berharap program ini dapat dimanfaatkan sebesar-besarnya untuk kepentingan ilmu pengetahuan dan teknologi. Penulis juga berharap program ini dapat dikembangkan dan/atau dijadikan dasar untuk pembuatan program yang dapat membantu masyarakat.

#### **3. Refleksi**

Sebagai evaluasi, penulis bertekad untuk:

1. Eksplorasi berbagai komponen tugas lebih jauh, seperti *stemming* dan penghapusan *stopwords*, serta Flask sebagai salah satu framework penghubung *back-end* dan *front-end* yang bermanfaat.
2. Lebih memperhatikan kerapian program, penamaan fungsi dan variabel, serta pembagian modul-modul fungsi.
3. Mempelajari lebih jauh mengenai proses pendataan yang efisien bagi data dengan jumlah yang sangat banyak.
4. Lebih memahirkan diri dalam debugging dan memastikan tidak ada test case yang terlupakan dalam mengembangkan suatu program.

## DAFTAR PUSTAKA

- Tim Mata Kuliah IF1213 Aljabar Linier dan Geometri. (2020). *Spesifikasi Tugas Besar 2*. Diakses pada 15 November 2020, dari: Microsoft Teams IF1213 Aljabar Linier dan Geometri, Tubes2-Algeo-2020-update (1).pdf.
- Geeks for Geeks. (2020). *What is Information Retrieval?* Diakses pada 15 November 2020, dari: <https://www.geeksforgeeks.org/what-is-information-retrieval/>
- Pérez-Montoro, Mario, dan Lluís Codina. (2017). *Essentials of Search Engine Optimization*. Diakses pada 15 November 2020, dari: <https://www.sciencedirect.com/topics/computer-science/information-retrieval-systems>
- Munir, Rinaldi. (2020). *Vektor di Ruang Euclidean (bagian 2)*. Diakses pada 15 November 2020, dari: <http://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-11-Vektor-di-Ruang-Euclidean-Bag2.pdf>
- Munir, Rinaldi. (2020). *Aplikasi Dot Product dalam Sistem Temu-Balik Informasi (Information Retrieval System)*. Diakses pada 15 November 2020, dari: <http://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-12-Aplikasi-dot-product-pada-IR.pdf>
- Prabhakaran, Selva. (2020). *Cosine Similarity – Understanding the Math and How it Works (with Python Codes)*. Diakses pada 15 November 2020, dari: <https://www.machinelearningplus.com/nlp/cosine-similarity/>
- Stanford NLP Group. (2009). *Stemming and Lemmatization*. Cambridge University Press. Diakses pada 15 November 2020, dari: <https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html>
- Yulio, Anggri. (2017). *Stopword Removal Bahasa Indonesia dengan Python Sastrawi*. Diakses pada 15 November 2020, dari: <https://devtrik.com/python/stopword-removal-bahasa-indonesia-python-sastrawi/>