

LAPORAN TUGAS BESAR II

IF2123 ALJABAR LINIER DAN GEOMETRI

Semester I Tahun 2020/2021



Disusun oleh:
Kelompok 20

13519021	Arjuna Marcelino
13519092	Sharon Bernadetha Marbun
13519120	Epata Tuah

Himpunan Mahasiswa Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2020

Daftar Isi

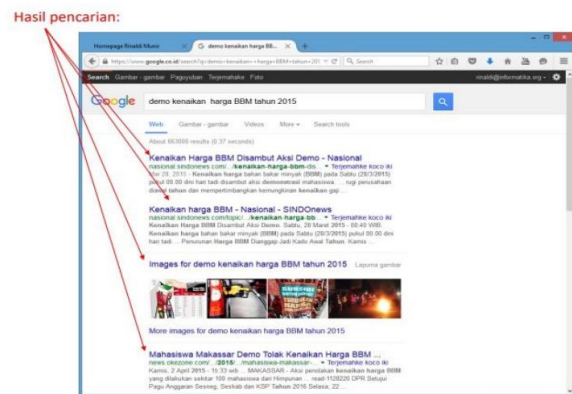
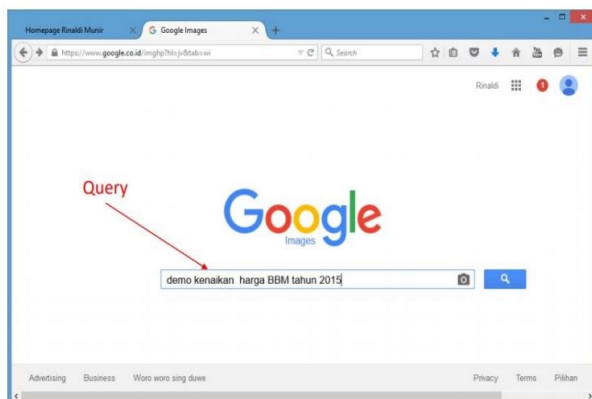
I. Deskripsi Masalah	2
1.1 Abstraksi	2
1.2 Penggunaan Program	3
1.3 Spesifikasi Tugas	4
II. Teori Singkat	6
III. Implementasi Program dalam Python	10
IV. Eksperimen	13
4.1 Tampilan Awal	13
4.2 Pencarian Query	15
4.3 Pengunggahan File	20
V. Kesimpulan, Saran, dan Refleksi	23
5.1 Kesimpulan	23
5.2 Saran	23
5.3 Refleksi	23
Referensi	24

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 vektor 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.



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:

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

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

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*, 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. Melakukan 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: dapat menggunakan library sastrawi atau nltk untuk stemming kata dan penghapusan stopwords.

1.3 Spesifikasi Tugas

Program mesin pencarian menggunakan 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. 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.

II. Teori Singkat

Temu-balik informasi (*information retrieval*) adalah menemukan kembali (*retrieval*) informasi yang relevan terhadap kebutuhan pengguna dari suatu kumpulan informasi secara otomatis.

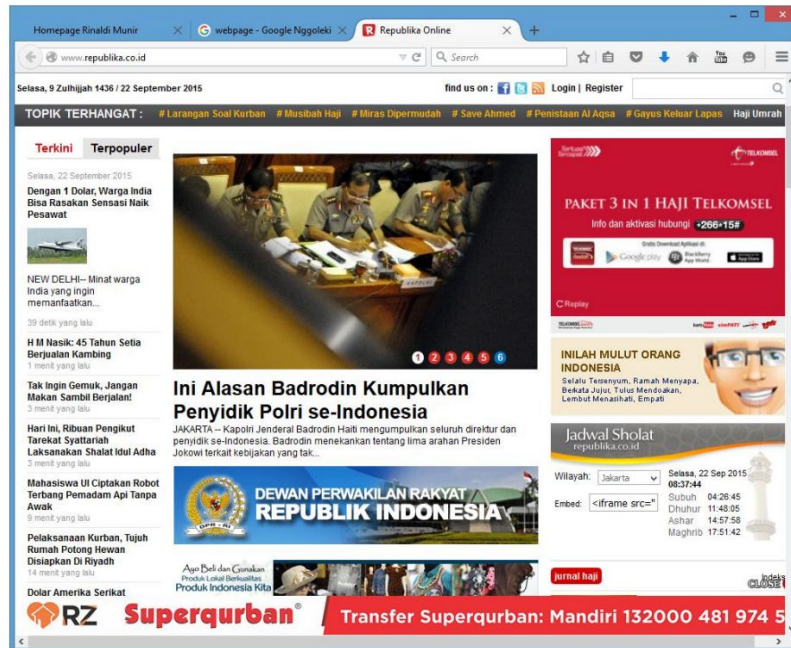


IR (*information retrieval*) tidak sama dengan pencarian di dalam basis data (*database*) dan umumnya digunakan pada pencarian informasi yang isinya tidak terstruktur.

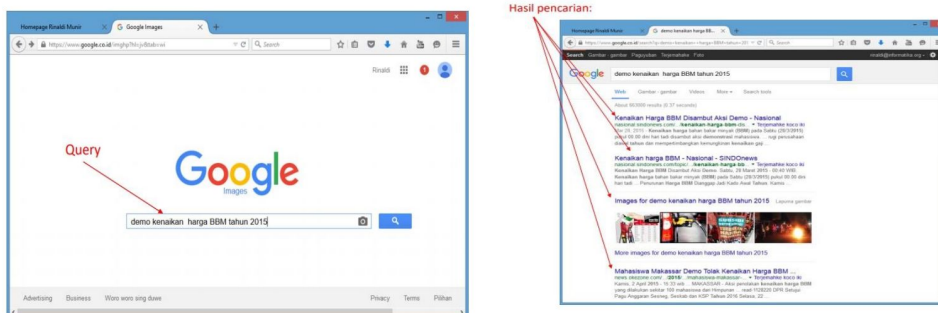
Contoh informasi terstruktur adalah tabel-tabel di dalam basis data (*database*).

Tabel mahasiswa						
NO	NAMA	NIM	JENIS KELAMIN	Umur	Tahun Lahir	Asal
1	Yusuf R	10018149	L	18	1992	Jogja
2	Lukman Reza	10018148	L	18	1992	Sulawesi
3	Aril	10018154	L	18	1992	Sumatra
4	Kifli	10018156	L	18	1992	Jogja
5	Khairuddin	10018151	L	18	1992	Papua
6	Angga	10018181	L	18	1992	Wonosobo
7	Nely	10018170	P	18	1992	Jogja
8	Reza	10018129	L	18	1992	Jogja
9	Ana	10017213	P	20	1990	Jogja
10	Nina	10012312	P	19	1991	Jogja

Contoh informasi tak-terstruktur adalah dokumen (isinya bergantung pembuatnya) dan laman web (*webpage*).



Contoh aplikasi dari IR adalah *search engine* yang ditunjukkan oleh gambar berikut.



Salah satu model IR adalah model ruang vektor. Model ini menggunakan teori di dalam aljabar vektor. Misalkan terdapat n kata berbeda sebagai kamus kata (*vocabulary*) atau indeks kata (*term index*). Kata-kata tersebut membentuk ruang vektor berdimensi n . Setiap dokumen maupun *query* dinyatakan sebagai vektor $w = (w_1, w_2, \dots, w_n)$ di dalam \mathbb{R}^n , dimana nilai w_i adalah bobot setiap kata i di dalam *query* atau dokumen. 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 vektor:

$$D_1 = (2, 3, 5), D_2 = (3, 7, 1), Q = (0, 0, 2)$$

$D_1 = (2, 3, 5)$ artinya dokumen D_1 mengandung 2 buah kata T_1 , 3 buah kata T_2 , dan 5 buah kata T_3 .

$D_2 = (3, 7, 1)$ artinya dokumen D_2 mengandung 3 buah kata T_1 , 7 buah kata T_2 , dan satu buah kata T_3 .

$Q = (0, 0, 2)$ artinya query Q hanya mengandung 2 buah kata T_3 .

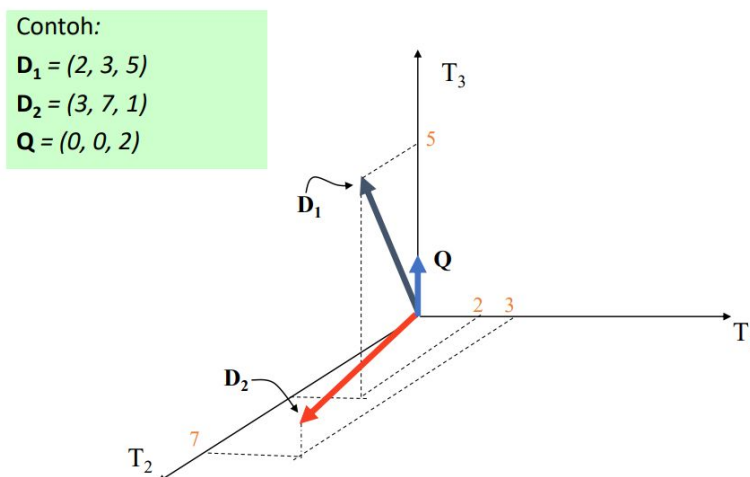
Contoh: Misalkan T_1 = Menteri, T_2 = minta, T_3 = Korupsi

D_1 = Menteri olahraga meminta maaf atas perbuatan korupsi. Menteri tersebut terlibat korupsi anggaran. Meminta-minta komisi termasuk korupsi. Korupsi sudah mendarah daging di Indonesia. Korupsi sudah menjadi budaya.

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.

Q = Korupsi besar atau kecil tetap saja korupsi.

Representasinya dalam bentuk grafik vektor adalah sebagai berikut.

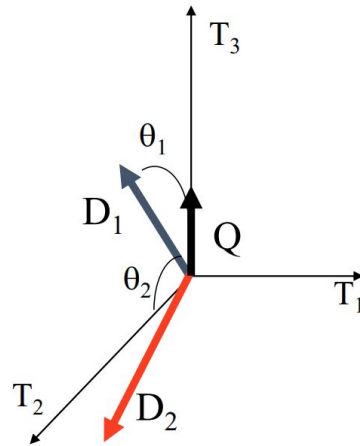


Penentuan dokumen mana yang relevan dengan *query* dipandang sebagai pengukuran kesamaan (*similarity measure*) antara *query* dengan dokumen. Semakin sama suatu vektor dokumen dengan vektor *query*, semakin relevan dokumen tersebut dengan *query*. Kesamaan (*sim*) antara dua vektor $Q = (q_1, q_2, \dots, q_n)$ dan $D = (d_1, d_2, \dots, d_n)$ diukur dengan rumus *cosine similarity* yang merupakan bagian dari rumus perkalian titik (*dot product*) dua buah vektor:

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

dengan $Q \cdot D$ adalah perkalian titik yang didefinisikan sebagai berikut.

$$Q \cdot D = q_1 d_1 + q_2 d_2 + \dots + q_n d_n$$



Jika $\cos(\theta) = 1$, berarti $\theta = 0$, vektor Q dan D berimpit, yang berarti dokumen D sesuai dengan *query* Q . Jadi, nilai *cosinus* yang besar (mendekati 1) mengindikasikan bahwa dokumen cenderung sesuai dengan *query*.

Setiap dokumen di dalam koleksi dokumen dihitung kesamaannya dengan *query* dengan rumus cosinus di atas. Selanjutnya hasil perhitungan di-*ranking* berdasarkan nilai cosinus dari besar ke kecil sebagai proses pemilihan dokumen yang yang “dekat” dengan *query*. Pe-*ranking*-an tersebut menyatakan dokumen yang paling relevan hingga yang kurang relevan dengan *query*. Nilai cosinus yang besar menyatakan dokumen yang relevan, nilai cosinus yang kecil menyatakan dokumen yang kurang relevan dengan *query*.

Pada contoh di atas:

$$Q \cdot D_1 = (2)(0) + (3)(0) + (5)(2) = 10$$

$$Q \cdot D_2 = (3)(0) + (7)(0) + (1)(2) = 2$$

$$\|Q\| = \sqrt{0^2 + 0^2 + 2^2} = \sqrt{4} = 2$$

$$\|D_1\| = \sqrt{2^2 + 3^2 + 5^2} = \sqrt{4 + 9 + 25} = \sqrt{38}$$

$$\|D_2\| = \sqrt{3^2 + 7^2 + 1^2} = \sqrt{9 + 49 + 1} = \sqrt{59}$$

$$\text{sim}(Q, D_1) = \cos \theta_1 = \frac{Q_1 \cdot D_1}{\|Q\| \|D_1\|} = \frac{10}{2\sqrt{38}} = 0.81$$

$$\text{sim}(Q, D_2) = \cos \theta_2 = \frac{Q_1 \cdot D_2}{\|Q\| \|D_2\|} = \frac{2}{2\sqrt{59}} = 0.13$$

Karena $0.81 > 0.13$, maka dokumen D_1 lebih sesuai dengan *query* Q dibandingkan dengan dokumen Q_2 .

III. Implementasi Program dalam Python

Program mesin pencarian ini dibagi ke dalam lima buah subprogram, yaitu

1. Input_File.py

Subprogram ini digunakan untuk menginput seluruh file yang berada di dalam folder test.

Atribut yang terdapat di dalam subprogram ini adalah sebagai berikut.

- a. nDok, menyimpan jumlah dokumen yang ada di dalam folder test
- b. file_list, array yang menyimpan nama file beserta ekstensinya
- c. d, array yang menyimpan dokumen asli
- d. judul, array yang menyimpan judul dokumen
- e. stop, array yang menyimpan hasil penghapusan stopwords
- f. clean, array yang menyimpan hasil pembersihan dokumen (*stopword + stemming*)
- g. s, array yang menyimpan kalimat pertama dokumen

2. Tab_Sim.py

Subprogram ini digunakan untuk membuat tabel yang berisi nilai similaritas dokumen terhadap *query* yang diinput oleh pengguna.

Method yang terdapat di dalam subprogram ini adalah sebagai berikut.

- a. Tab_Sim, fungsi dengan parameter nTerm (jumlah terms), tab_frekuensi (tabel frekuensi), dan nDok (jumlah dokumen) yang mengembalikan array similaritas.
- b. norm, fungsi dengan parameter d (indeks dokumen), nTerm (jumlah terms), dan tab_frekuensi (tabel frekuensi) yang mengembalikan nilai normal dari vektor dokumen ke-d.
- c. kalidot, fungsi dengan parameter d (indeks dokumen), nTerm (jumlah terms), dan tab_frekuensi (tabel frekuensi) yang mengembalikan hasil perkalian dot antara vektor dokumen ke-d dengan *query*

Atribut yang terdapat di dalam subprogram ini adalah sebagai berikut.

- a. sim, array yang berisi nilai similaritas dokumen dengan query pada method Tab_Sim
- b. sum, menyimpan nilai normal dari vektor dokumen pada method norm
- c. sum, menyimpan hasil jumlah perkalian dot antara vektor dokumen dan *query* pada method kalidot

3. Web_Scraping.py

Subprogram ini digunakan untuk melakukan *web scraping* link yang telah diinput oleh pengguna ke dalam folder test dalam bentuk file txt.

Method yang terdapat di dalam subprogram ini adalah sebagai berikut.

- a. web_scrap, prosedur dengan parameter input (masukan link oleh pengguna melalui *keyboard*) yang melakukan *web scraping* link ke dalam folder test dalam bentuk file txt.

Atribut yang terdapat di dalam subprogram ini adalah sebagai berikut.

- a. link, array yang menyimpan link unik dari inputan pengguna
- b. nLink, menyimpan banyaknya jumlah link
- c. isiDokumen, array yang menyimpan teks hasil *web scraping*
- d. judulWeb, array yang menyimpan judul pertama yang ditemukan di website
- e. slash, menyimpan konversi ASCII dari simbol ‘\’
- f. old, menyimpan array simbol-simbol yang tidak bisa dijadikan bagian dari nama file
- g. new, menyimpan array spasi kosong untuk mengganti simbol-simbol pada array old
- h. url, melakukan *request access* terhadap link unik
- i. page, menyimpan data mentah dari link unik
- j. html, men-*decode* “utf-8” pada data mentah link unik dan membaca page
- k. soup, melakukan penghapusan kode-kode *html* pada link unik sehingga dihasilkan data yang bersih
- l. tmp, menyimpan judul yang terdeteksi di website dan mengganti simbol-simbol array old menjadi spasi agar judul dapat digunakan sebagai nama file
- m. dokumen, menyimpan isi link unik sebagai string
- n. fname, menyimpan nama file berformat .txt dalam folder test dengan penamaan sesuai judul yang ada di variabel judulWeb

4. app.py

Subprogram ini merupakan main dari program yang memadukan *back-end* dengan *front-end* sehingga dapat menampilkan website lokal yang sesuai dengan spesifikasi.

Atribut yang terdapat di dalam subprogram ini adalah sebagai berikut.

- a. nDok, menyimpan jumlah dokumen yang ada di dalam folder test yang dipanggil dari subprogram Input_File.py
- b. clean, array yang menyimpan hasil pembersihan dokumen (*stopword* + *stemming*) yang dipanggil dari subprogram Input_File.py
- c. judul, array yang menyimpan judul dokumen yang dipanggil dari subprogram Input_File.py
- d. query, menyimpan query yang diinput oleh pengguna melalui *keyboard*
- e. terms, array yang menyimpan *term query*
- f. nTerm, menyimpan banyaknya jumlah term *query*
- g. tab_frekuensi, tabel frekuensi yang menyimpan kemunculan kata pada dokumen dan query yang sesuai dengan term
- h. Tab_Sim, menyimpan nilai fungsi Tab_Sim yaitu tabel similaritas yang dipanggil dari subprogram Tab_Sim.py
- i. Index_SortedSim, array yang menyimpan indeks terurut mengecil dari Tab_Sim
- j. Tab_countKata, array yang menyimpan jumlah kata tiap dokumen yang sudah terurut

- k. Tab_sortedJudul, array yang menyimpan isi judul dokumen yang sudah terurut
- l. Tab_FirstSent, array yang menyimpan kalimat pertama dari dokumen yang sudah terurut
- m. tab_info, tabel gabungan dari array judul, Tab_countKata, Tab_Sim, dan Tab_FirstSent
- n. term_frekuensi, tabel gabungan dari array terms dan tabel tab_frekuensi
- o. files, array yang menyimpan teks hasil upload manual oleh pengguna

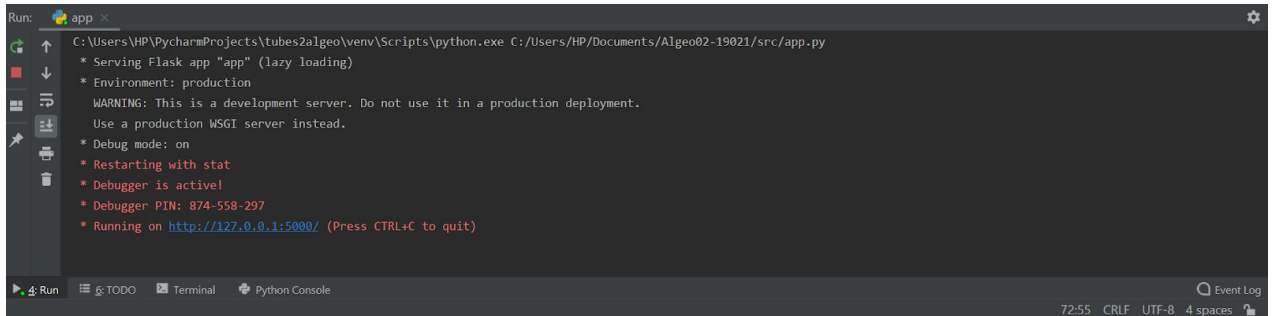
Method yang terdapat di dalam subprogram ini adalah sebagai berikut.

- a. allowed_file, fungsi dengan parameter filename (nama file yang akan kita unggah) yang mengembalikan file dengan ekstensi yang sesuai ke dalam folder penyimpanan.
- b. index, fungsi yang merupakan beranda yang menampilkan halaman utama dari *search engine* yang dilengkapi dengan *search bar* (route : /).
- c. search, fungsi yang akan jalan saat kita memberi masukan pada *search bar* pada halaman utama (index) dan akan melanjutkan program kepada fungsi *search_query* dan mendefinisikan query (parameter *search query*) adalah inputan user di *search bar* (route : /).
- d. search_query, fungsi dengan parameter query yang memproses query dan menentukan dokumen yang relevan dengan query dengan rumus pengukuran yang disebut *cosine similarity*. Lalu akan menampilkan hasil pencarian berupa judul dokumen, jumlah kata dokumen, tingkat kemiripan, kalimat pertama dari dokumen secara terurut dari yang paling relevan. Di akhir akan ditampilkan tabel term dan banyak kemunculan term dalam setiap dokumen (route : /<query>).
- e. perihal, fungsi yang menampilkan informasi tentang program dan pembuatnya (Konsep singkat search engine yang dibuat, How to Use, About Us) (route : /perihal).
- f. pranala, fungsi yang menampilkan halaman untuk menerima url yang akan diunduh atau di-*web scraping* (route : /pranala).
- g. ambil_pranala, fungsi yang menerima masukan pranala dari user yang akan diproses dengan *web scraping* kemudian diunduh/dicatat ke dalam folder (route : /pranala).
- h. daftar, fungsi yang menampilkan informasi daftar dokumen yang tersedia (route : /daftar-dokumen).
- i. buka_file, fungsi dengan parameter file (judul file) yang akan menampilkan isi dari file .txt atau .html yang dituju (route : /test/<file>).
- j. upload, fungsi yang menampilkan halaman untuk mengunggah dokumen (route: /unggah).
- k. upload_file, fungsi yang menerima dokumen hasil unggahan dan menampilkan keberhasilan pengunggahan (route : /unggah).

IV. Eksperimen

4.1 Tampilan Awal

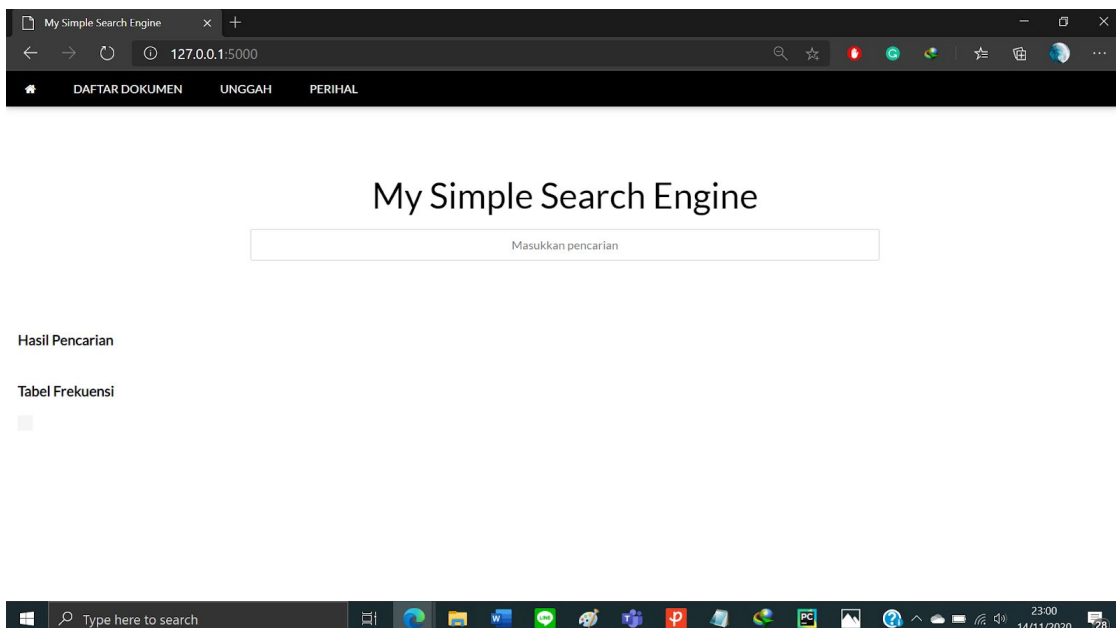
Setelah `app.py` di-*run*, diberikan sebuah link server <http://127.0.0.1:5000/> pada terminal sebagai berikut.



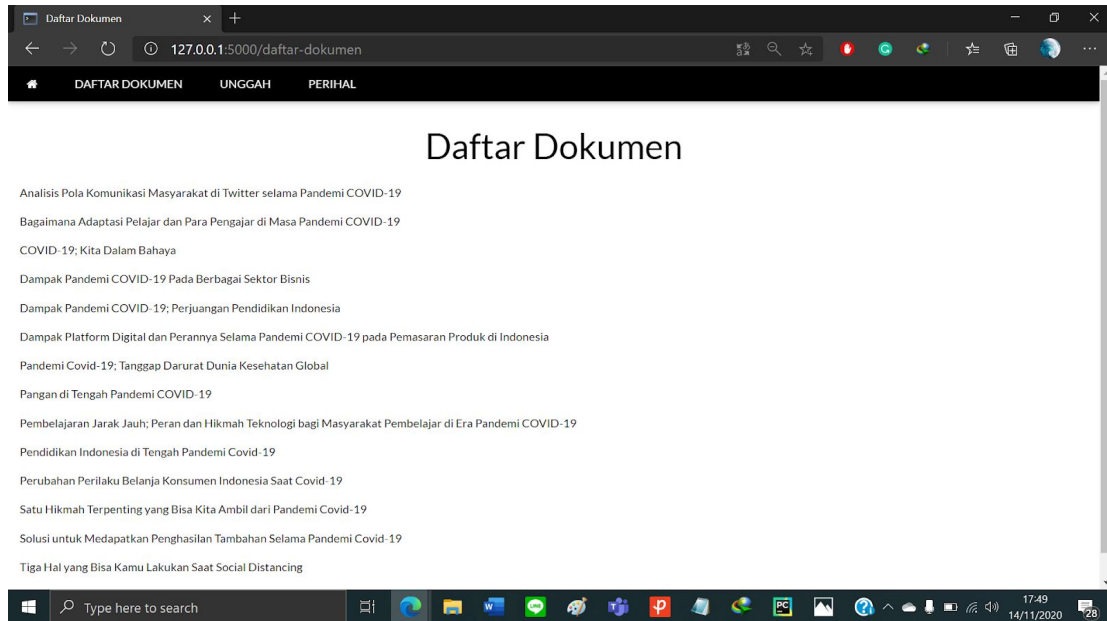
```
Run: app
C:\Users\HP\PycharmProjects\tubes2algeo\venv\Scripts\python.exe C:/Users/HP/Documents/Algeo02-19021/src/app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 874-558-297
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Link tersebut mengarahkan pengguna ke *website* lokal “My Simple Search Engine”. Pada halaman *website* terdapat empat buah tab menu, yaitu *home* (gambar rumah), daftar dokumen, unggah, dan perihal.

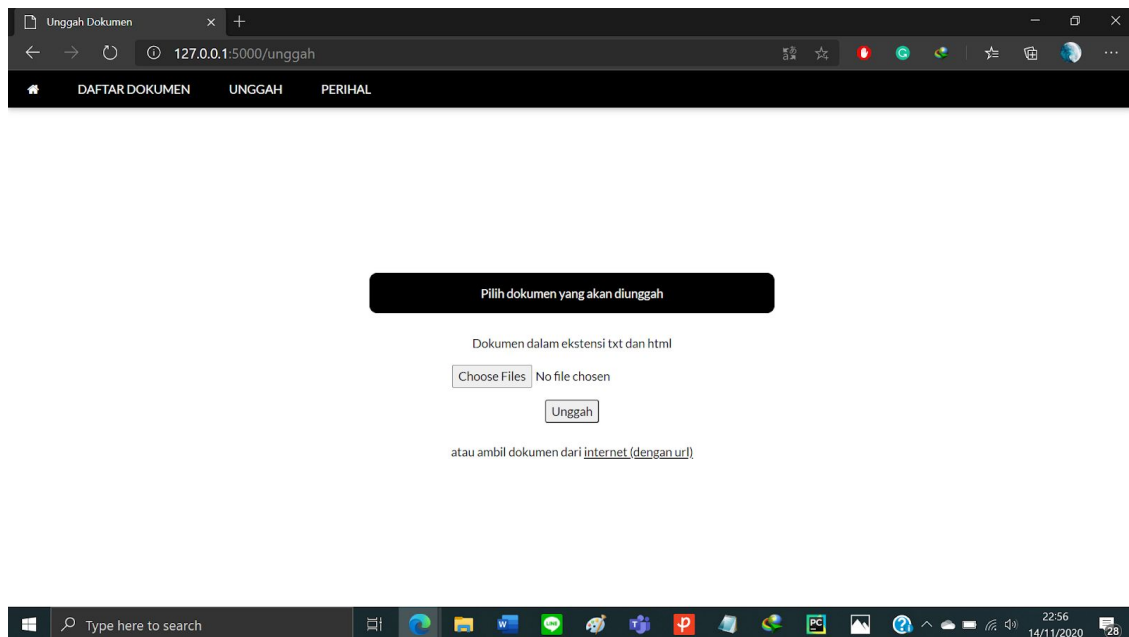
Tampilan tab *home* adalah sebagai berikut.



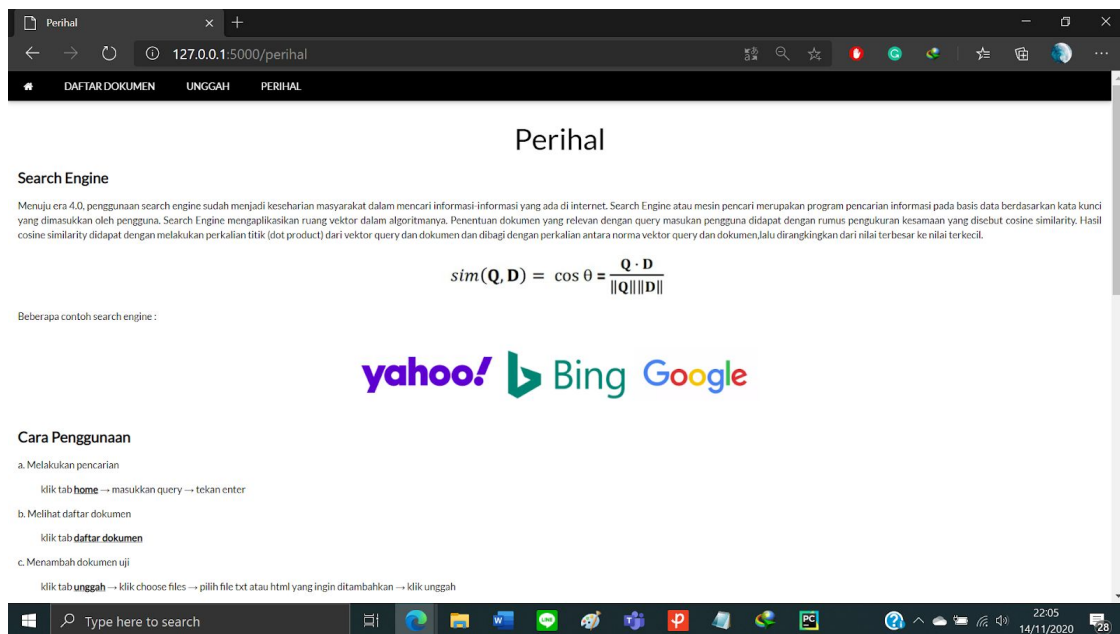
Tampilan tab daftar dokumen adalah sebagai berikut.



Tampilan tab unggah adalah sebagai berikut.

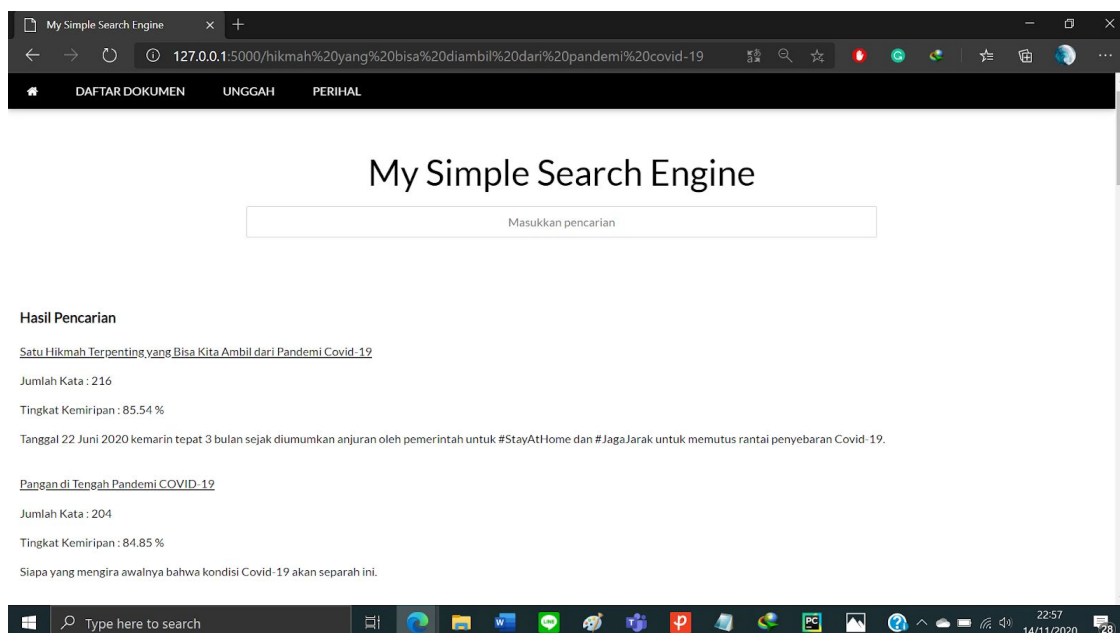


Tampilan tab perihal adalah sebagai berikut.



4.2 Pencarian Query

Dimasukkan *query* “hikmah yang bisa diambil dari pandemi covid-19” pada bar pencarian, hasil pencariannya ditampilkan sebagai berikut.



Hasil pencarian lebih lengkap terhadap *query* tersebut adalah sebagai berikut.

1) [Satu Hikmah Terpenting yang Bisa Kita Ambil dari Pandemi Covid-19](#)

Jumlah Kata : 216

Tingkat Kemiripan : 85.54 %

Tanggal 22 Juni 2020 kemarin tepat 3 bulan sejak diumumkan anjuran oleh pemerintah untuk #StayAtHome dan #JagaJarak untuk memutus rantai penyebaran Covid-19.

2) [Pangan di Tengah Pandemi COVID-19](#)

Jumlah Kata : 204

Tingkat Kemiripan : 84.85 %

Siapa yang mengira awalnya bahwa kondisi Covid-19 akan separah ini.

3) [Tiga Hal yang Bisa Kamu Lakukan Saat Social Distancing](#)

Jumlah Kata : 318

Tingkat Kemiripan : 77.46 %

Perkembangan penyebaran kasus COVID — 19 telah mencapai Indonesia dan terus berkembang.

4) [Dampak Pandemi COVID-19 Pada Berbagai Sektor Bisnis](#)

Jumlah Kata : 246

Tingkat Kemiripan : 76.28 %

Tahun 2020 baru berjalan tiga bulan namun goncangan ekonomi telah terjadi begitu hebatnya.

5) [Solusi untuk Mendapatkan Penghasilan Tambahan Selama Pandemi Covid-19](#)

Jumlah Kata : 368

Tingkat Kemiripan : 74.74 %

Pada saat sekarang ini, ekonomi dunia sedang terpuruk yang disebabkan oleh wabah virus corona atau dengan kata lain “Covid-19”.

6) [Perubahan Perilaku Belanja Konsumen Indonesia Saat Covid-19](#)

Jumlah Kata : 225

Tingkat Kemiripan : 73.03 %

Dampak ekonomi dari pandemi COVID-19 begitu luas dan begitu masif, sehingga benar-benar mengubah tatanan permainan bisnis hampir secara keseluruhan.

7) [Bagaimana Adaptasi Pelajar dan Para Pengajar di Masa Pandemi COVID-19](#)

Jumlah Kata : 262

Tingkat Kemiripan : 71.71 %

Pendidikan adalah kegiatan dimana kita semua dapat berbagi ilmu dan membangun kedisiplinan serta, membangun rasa sosial kita untuk mempersiapkan diri menghadapi dunia yang luas ini.

8) [Pembelajaran Jarak Jauh; Peran dan Hikmah Teknologi bagi Masyarakat Pembelajar di Era Pandemi COVID-19](#)

Jumlah Kata : 183

Tingkat Kemiripan : 71.71 %

Cepatnya perubahan zaman pada abad ke-21 terjadi karena adanya perkembangan teknologi yang semakin cepat.

9) [Tujuh Hal yang Bisa Dilakukan Brand Lewat Konten di Saat Krisis Covid-19](#)

Jumlah Kata : 349

Tingkat Kemiripan : 68.03 %

Krisis COVID-19 membuat dunia bisnis di seluruh dunia porak poranda.

10) [Dampak Platform Digital dan Perannya Selama Pandemi COVID-19 pada Pemasaran Produk di Indonesia](#)

Jumlah Kata : 236

Tingkat Kemiripan : 63.25 %

Pandemi virus corona berdampak besar terhadap seluruh aspek kehidupan khususnya bidang perekonomian di Indonesia sehingga memerlukan upaya memasarkan produk yang harus terus berjalan lancar.

11) [Pandemi Covid-19: Tanggap Darurat Dunia Kesehatan Global](#)

Jumlah Kata : 274

Tingkat Kemiripan : 62.99 %

Covid-19 adalah penyakit yang ditimbulkan oleh infeksi virus corona baru atau SARS-CoV-2 yang berasal dari keluarga corona.

12) [Dampak Pandemi COVID-19: Perjuangan Pendidikan Indonesia](#)

Jumlah Kata : 238

Tingkat Kemiripan : 62.61 %

Suatu saat, perkembangan zaman akan menuntut perubahan peradaban pendidikan.

13) [Pendidikan Indonesia di Tengah Pandemi Covid-19](#)

Jumlah Kata : 261

Tingkat Kemiripan : 62.61 %

Kita tahu bahwa pandemi yang sedang terjadi telah merubah seluruh tatanan hidup masyarakat.

14) [Analisis Pola Komunikasi Masyarakat di Twitter selama Pandemi COVID-19](#)

Jumlah Kata : 323

Tingkat Kemiripan : 60.25 %

Pandemi COVID-19 belum berakhir.

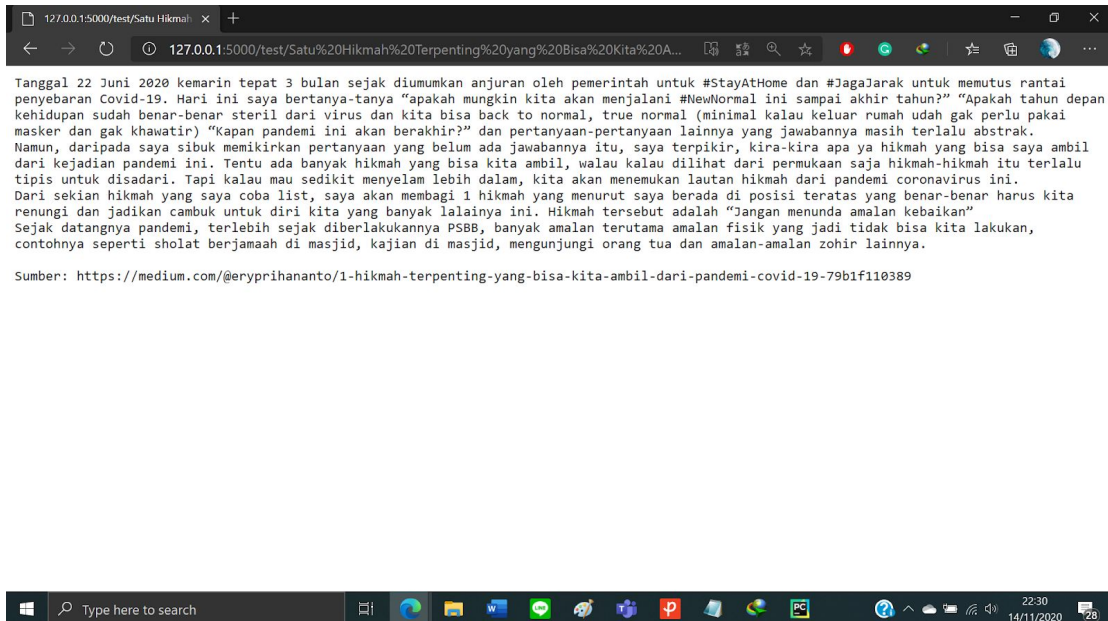
15) [COVID-19: Kita Dalam Bahaya](#)

Jumlah Kata : 259

Tingkat Kemiripan : 54.23 %

Hampir seluruh dunia mengalami dampak buruk yang disebabkan oleh wabah COVID-19 ini.

Setelah judul dokumen [Satu Hikmah Terpenting yang Bisa Kita Ambil dari Pandemi Covid-19](#) di-klik, ditampilkan isi dokumen sebagai berikut.



Selain itu, pada hasil pencarian ditampilkan juga tabel frekuensi kemunculan *term* pada *query* dan dokumen uji sebagai berikut.

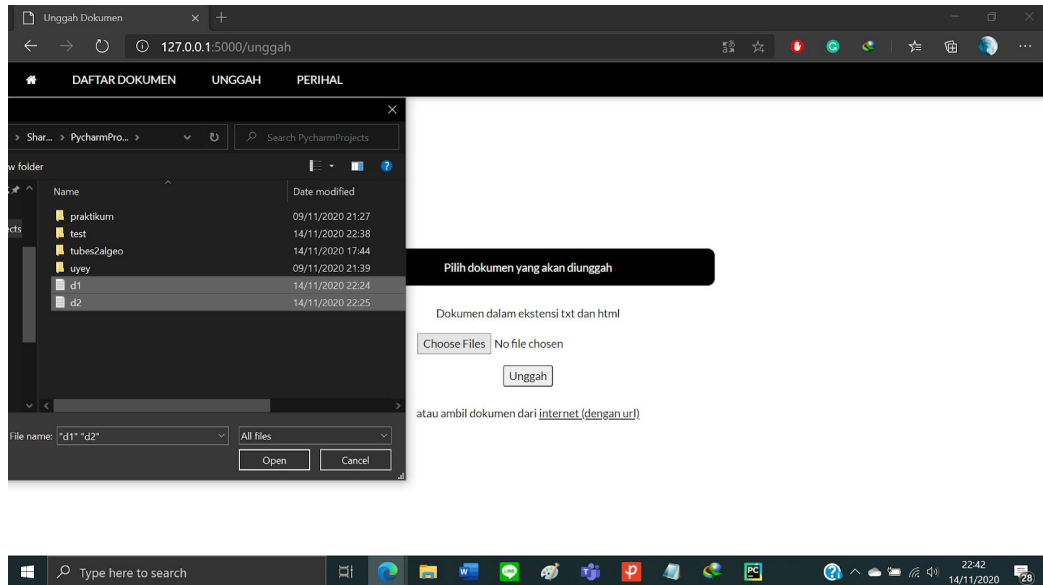
Tabel Frekuensi

Term	Query	Analisis Pola Komunikasi Masyarakat di Twitter selama Pandemi COVID-19	Bagaimana Adaptasi Pelajar dan Para Pengajar di Masa Pandemi COVID-19	COVID-19: Kita Dalam Bahaya	Dampak Pandemi COVID-19 Pada Berbagai Sektor Bisnis	Dampak Pandemi COVID-19: Perjuangan Pendidikan Indonesia	Dampak Platform Digital dan Perannya Selama Pandemi COVID-19 pada Pemasaran Produk di Indonesia	Pandemi Covid-19: Tanggapan Darurat Dunia Kesehatan Global	Pangan di Tengah Pandemi COVID-19	Pembelajaran Jarak Jauh: Peran dan Hikmah Teknologi bagi Masyarakat Pembelajar di Era Pandemi COVID-19	Pendidikan Indonesia di Tengah Pandemi Covid-19	Perubahan Perilaku Belanja Konsumen Indonesia Saat Covid-19	Satu Hikmah Terpenting yang Bisa Kita Ambil dari Pandemi Covid-19	Solusi untuk Mendapatkan Penghasilan Tambahan Selama Pandemi Covid-19	Tiga Hal yang Bisa Kamu Lakukan Saat Social Distancing	Tujuh Hal yang Bisa Dilakukan Brand Lewat Konten di Saat Krisis Covid-19
covid-19	1	1	2	4	3	3	2	6	2	2	4	1	1	3	1	3
ambil	1	0	1	0	0	0	0	0	0	0	0	0	2	0	0	0
pandemi	1	5	3	0	3	4	2	5	2	3	3	1	4	2	1	1
hikmah	1	0	0	0	0	0	0	0	1	0	0	0	7	0	0	0
bisa	1	1	0	1	2	0	0	0	1	1	0	2	3	4	1	5

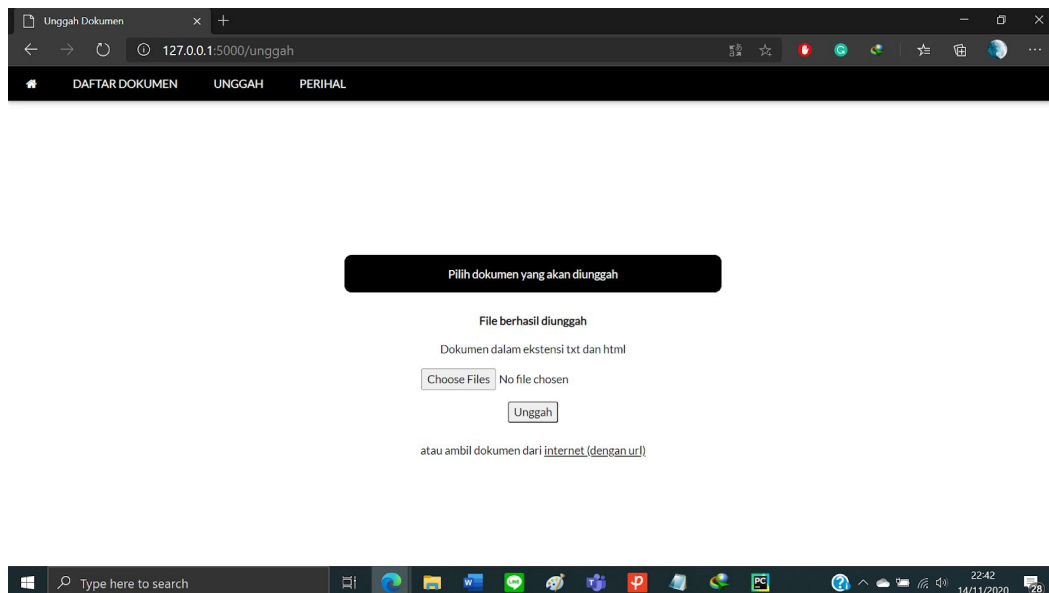
4.3 Pengunggahan File

a. File dari perangkat pengguna

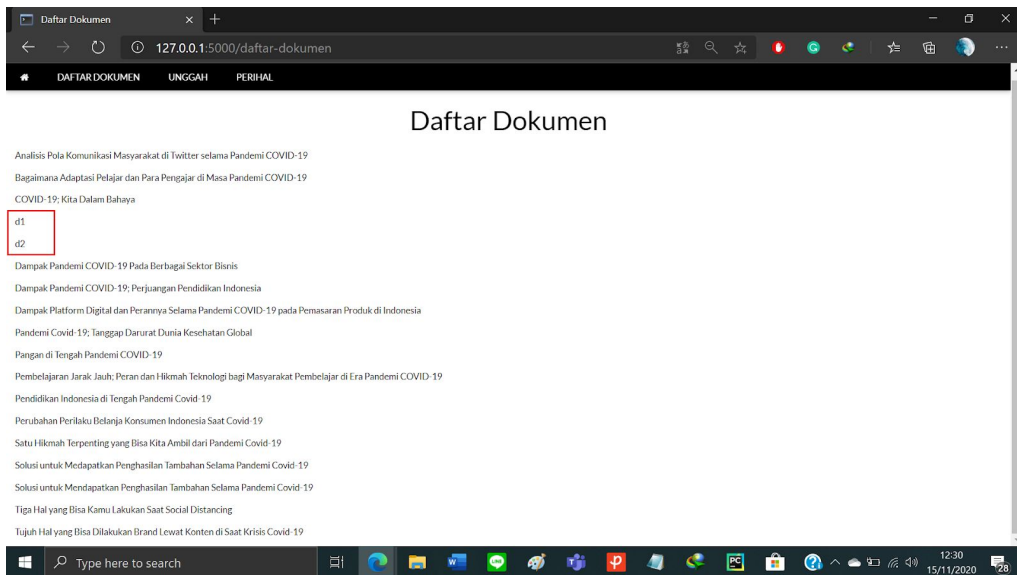
Pada tab unggah, pengguna meng-klik *Choose Files*, kemudian ditampilkan jendela *file explorer* sebagai berikut.



Dimasukkan file d1.txt dan d2.txt dan diklik *Unggah*, kemudian ditampilkan pesan berhasil sebagai berikut.

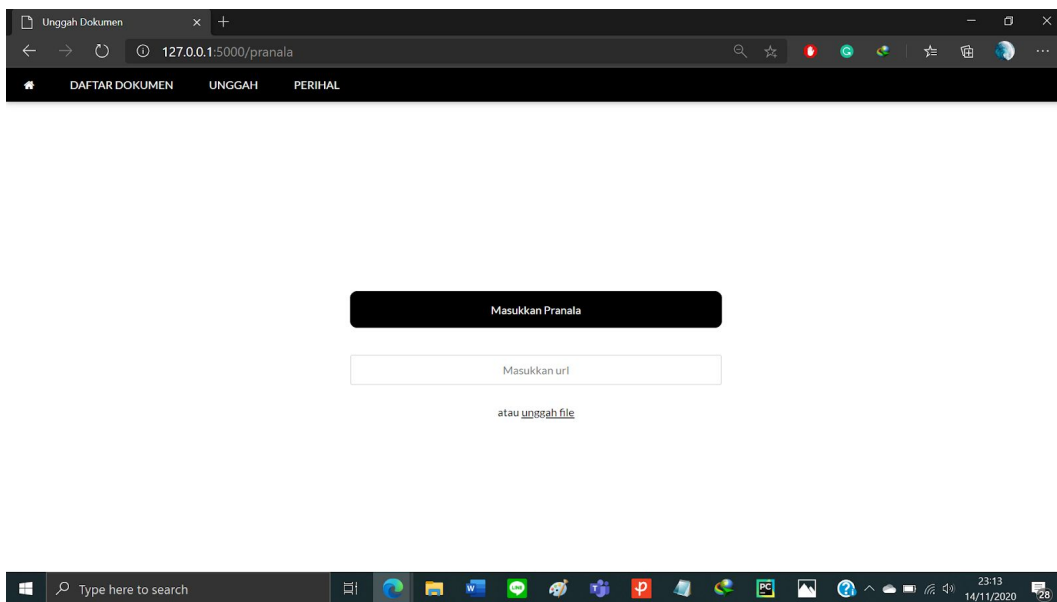


Daftar dokumen diperbaharui menjadi sebagai berikut (perhatikan yang ditandai merah).

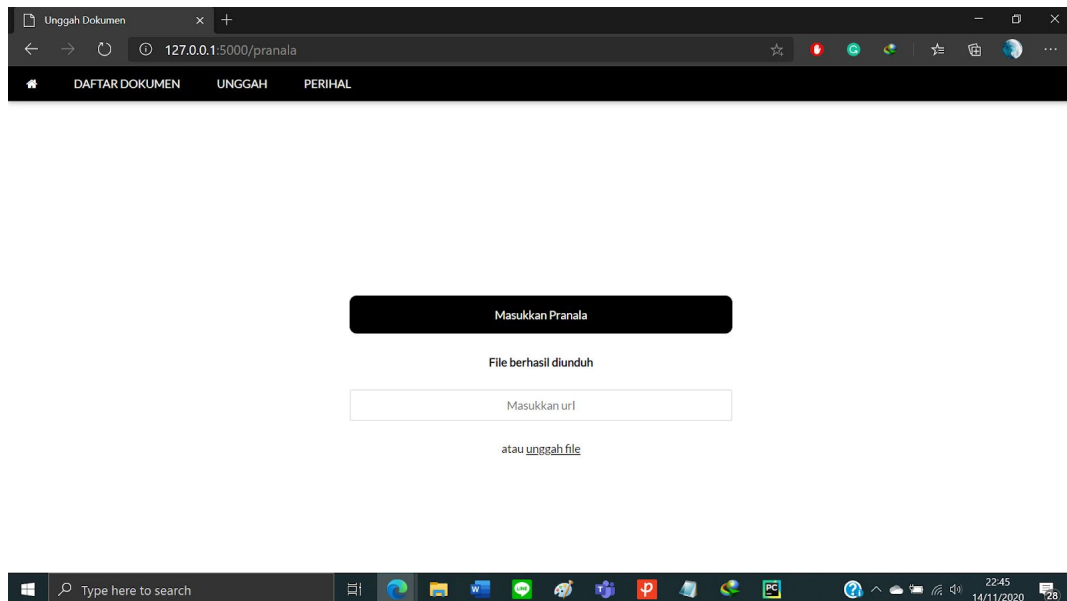


b. Web scraping

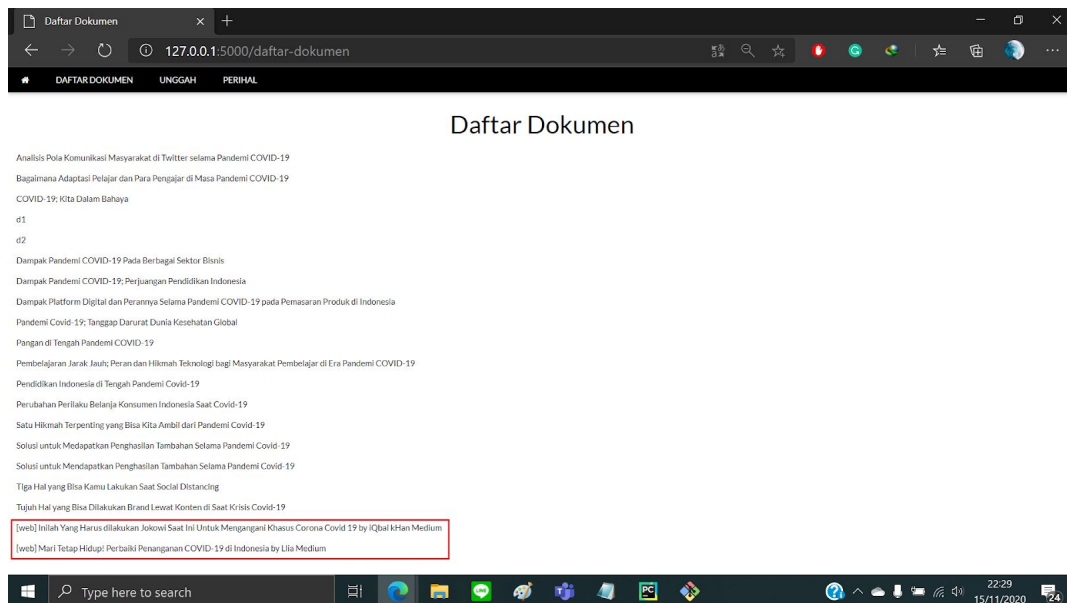
Pada tab unggah, pengguna mengklik internet (dengan url), kemudian tampilan *website* berubah menjadi sebagai berikut.



Pada bar yang tersedia dimasukkan dua buah link url “<https://medium.com/@salsabeela/mari-tetap-hidup-perbaiki-penanganan-covid-19-di-indonesia-997a5144114> <https://medium.com/@rasarab/inilah-yang-harus-dilakukan-jokowi-saat-ini-untuk-mengangani-khusus-corona-covid-19-bff50bf01842>” dan ditekan enter, kemudian ditampilkan pesan berhasil sebagai berikut.



Daftar dokumen diperbaharui menjadi sebagai berikut.



V. Kesimpulan, Saran, dan Refleksi

5.1 Kesimpulan

Program mesin pencarian ini memanfaatkan aplikasi ruang vektor di dalam algoritmanya. Pengurutan dokumen yang relevan dengan *query* dilakukan dengan memanfaatkan rumus *cosine similarity*. Nilai *cosine similarity* diperoleh dengan membagikan hasil perkalian titik vektor *query* dan vektor dokumen dengan hasil perkalian antara norma vektor *query* dan norma vektor dokumen. Nilai similaritas tersebut kemudian di-*ranking* dari besar ke kecil dengan nilai similaritas yang lebih besar dimiliki oleh dokumen yang lebih relevan dengan *query*.

Program ini akan membandingkan *query* dengan dokumen uji yang berada di dalam folder test. Selain mengambil dari folder test, dokumen uji juga dapat ditambahkan melalui *web scraping* atau pengunggahan file txt/html. Pada hasil pencarian, dokumen uji akan terurut secara menurun berdasarkan similaritasnya terhadap *query*. Disertakan juga beberapa info terkait dokumen tersebut berupa judul dokumen, jumlah kata, tingkat kemiripan, dan kalimat pertama dokumen. Selain itu, ditampilkan juga tabel frekuensi kemunculan *term* pada *query* dan dokumen uji.

5.2 Saran

Saran untuk pengembangan program, karena program yang kami buat masih berjalan dengan lambat, disarankan untuk mencari tahu algoritma alternatif yang lebih efektif dan efisien. Selain itu, *website* ini juga masih sederhana dan bersifat lokal sehingga program ini masih bisa dikembangkan supaya menjadi lebih baik lagi dan lebih ramah pengguna.

Saran untuk asisten, terdapat beberapa kerancuan informasi di dalam spesifikasi tugas besar sehingga terkadang kami perlu menunggu dahulu klarifikasi dari asisten melalui FAQ. Oleh karena itu, kami menyarankan supaya spesifikasi tugas besar ke depannya dipersiapkan dengan lebih rinci untuk menghindari adanya kesulitan pemahaman.

5.3 Refleksi

Kami mempelajari banyak hal dari pengerjaan tugas besar ini. Pemahaman dan pengalaman kami tentang bahasa pemrograman Python semakin luas dikarenakan selama pengerjaan tugas besar ini, terdapat banyak pemanfaatan *syntax* dan *library* Python yang baru kami pelajari. Selain itu, kami juga berkesempatan untuk memanfaatkan kreativitas kami dalam pembuatan desain *website* dari *Search Engine*-nya. Selain itu, dalam hal koordinasi, kami belajar bahwa komunikasi dan jadwal *meeting* yang konsisten adalah hal yang penting dalam pekerjaan kelompok sehingga tujuan dapat tercapai dengan baik.

Referensi

- A Practical Introduction to Web Scraping in Python*. Diakses pada tanggal 12 November 2020, dari <https://realpython.com/python-web-scraping-practical-introduction/>
- Count unique words in a text file (Python)*. Diakses pada tanggal 11 November 2020, dari <https://stackoverflow.com/questions/53271669/count-unique-words-in-a-text-file-python>
- CSS Tutorial*. Diakses pada tanggal 11 November 2020, dari <https://www.w3schools.com/css/default.asp>
- Finding the index of the sorted elements in python array*. Diakses pada tanggal 12 November 2020, dari <https://stackoverflow.com/questions/20668786/finding-the-index-of-sorted-elements-in-python-array>
- How to get filename without extension in python*. Diakses tanggal 12 November 2020, dari <https://www.codegrepper.com/code-examples/delphi/how+to+get+filename+without+extension+in+python>
- HTML Tutorial*. Diakses pada tanggal 11 November 2020, dari <https://www.w3schools.com/html/default.asp>
- Munir, R. 2020. *Aplikasi Dot Product pada sistem temu balik aplikasi*. Diakses pada tanggal 9 November 2020, dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-12-Aplikasi-dot-product-pada-IR.pdf>
- Sastrawi Library*. Diakses pada tanggal 9 November 2020, dari <https://pypi.org/project/Sastrawi/>
- Starter Template for Bootstrap*. Diakses pada tanggal 10 November 2020, dari <https://getbootstrap.com/docs/3.3/examples/starter-template/>
- Stopword Removal Bahasa Indonesia dengan Python Sastrawi*. Diakses pada tanggal 10 November 2020, dari <https://devtrik.com/python/stopword-removal-bahasa-indonesia-python-sastrawi/>
- Welcome to Flask - Flask Documentation*. Diakses pada tanggal 10 November 2020, dari <https://flask.palletsprojects.com/en/1.1.x/>
- Writing Multiple Files*. Diakses pada tanggal 12 November 2020, dari <https://stackoverflow.com/questions/39886038/writing-to-multiple-files>