

LAPORAN TUGAS BESAR
ALJABAR LINIER DAN GEOMETRI
IF2123



Naufal Yahya Kurnianto – 13519141 – K1

Alif Bhadrika Parikesit – 13519186 – K2

Rayhan Asadel – 13519196 – K4

BAB I

DESKRIPSI MASALAH

Search engine merupakan suatu kata yang sangat familiar pada zaman ini. Seluruh orang, mulai dari yang muda hingga yang tua tentu tidak bisa melepas diri dari search engine. Search engine merupakan program yang mencari sesuatu hal dari database dengan memasukkan query. Program akan mencari hal yang berkorelasi dengan query yang dimasukkan kemudian ditampilkan sehingga kita sebagai manusia dapat mendapatkan info atau ilmu baru. Search engine ini dibuat dengan memanfaatkan berbagai ilmu seperti temu-balik informasi yang ada pada materi vektor. Dengan mengubah search query (input pengguna) menjadi ruang vektor. Program dapat menghitung berapa banyak kemunculan kata pada query yang kemudian akan dibandingkan dengan ruang vektor dari dokumen yang ada. Jika terjadi kesamaan, maka dapat ditarik kesimpulan bahwa dokumen tersebut memiliki kesamaan dengan search query dan dapat ditampilkan ke pengguna.

Buatlah program mesin pencarian (search engine) 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

TEORI SINGKAT

2.1 Information Retrieval (Temu-Balik Informasi)

Temu-balik informasi digunakan untuk menemukan kembali informasi yang relevan terhadap kebutuhan pengguna dari suatu kumpulan informasi secara otomatis. Sistem ini umumnya digunakan pada pencarian informasi yang isinya tidak terstruktur. Oleh karena itu, suatu aplikasi umum temu-balik informasi adalah mesin pencarian (search engine) di internet.

Salah satu pemodelan untuk menggunakan temu-balik informasi adalah dengan menggunakan vektor. Misalkan terdapat n kata berbeda sebagai kamus kata atau indeks kata dari query maupun dokumen. Kata-kata berbeda tersebut dapat membentuk ruang vektor berdimensi n . Hasil vektor yang dapat menyatakan dokumen maupun query contohnya adalah sebagai berikut:

$$w = (w[1], w[2], w[3], \dots, w[n]); w[i] \text{ menyatakan frekuensi kemunculan kata}$$

Contohnya adalah, misalkan terdapat empat buah kata berbeda, dua buah dokumen dan sebuah query. Dapat dibentuk:

$$D1 = (1, 2, 3, 4), D2 = (2, 3, 4, 5), Q = (1, 1, 1, 1)$$

Angka dalam vektor melambangkan frekuensi kemunculan kata yang berbeda. Dapat disimpulkan misalnya bahwa pada dokumen pertama kata 1 muncul sekali; kata 2 muncul dua kali dan seterusnya.

2.2 Vektor

Vektor adalah representasi fisik untuk satuan yang memiliki besar dan arah. Vektor dilambangkan dengan huruf-huruf kecil dan dicetak tebal atau memakai tanda panah jika dituliskan tangan. Secara geometri, vektor dinyatakan dengan garis berarah. Setelah itu, ada istilah ruang vektor yang melambangkan ruang tempat vektor didefinisikan.

Vektor di R^n :

$$\mathbf{v} = (v_1, v_2, \dots, v_n) \text{ atau } \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

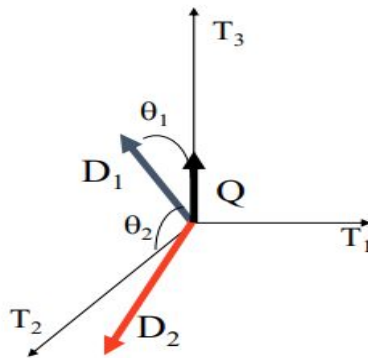
Vektor dapat mengaplikasikan berbagai aturan matematika seperti penjumlahan yang menggunakan kaidah parallelogram atau kaidah segitiga, pengurangan, dan perkalian. Perkalian juga dibagi dengan perkalian vektor dengan skalar dan sesama vektor. Vektor juga memiliki panjang atau *magnitude* yang dinamakan norma. Berbagai sifat-sifat aljabar juga berlaku untuk penjumlahan antar vektor.

2.2 Cosine Similarity

Untuk menentukan dokumen yang relevan dengan query, dapat dilakukan pengukuran kesamaan (similarity measure) antara query dengan dokumen. Semakin besar ukuran kesamaannya, maka semakin relevan dokumen tersebut dengan query. Pengukuran kesamaan ini dapat dilakukan dengan menggunakan vektor-vektor yang didapat dari query dan dokumen. Contohnya, untuk mengukur kesamaan antara dua vektor \mathbf{Q} dan \mathbf{D} , gunakan rumus cosine similarity. Rumusnya sebagai berikut:

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

Rumus demikian diperoleh karena konsep yang digunakan adalah menghitung cosinus, semakin besar cosinus hingga mendekati 1. Maka akan semakin kecil sudut yang memisahkan antara dua vektor tersebut. Dapat dilihat pada visualisasi berikut:



Jika sudut semakin kecil, maka nilai cosinus akan semakin besar. Karena sudut semakin kecil, vektor dokumen akan berjarak lebih dekat dengan vektor query sehingga dokumen akan lebih relevan dengan query. Setelah mendapatkan nilai cosinus antara dokumen pertama dengan query, dilanjutkan dengan menghitung tiap-tiap cosinus antara dokumen lain dengan query. Kemudian, dapat dilakukan pemeringkatan dari yang bernilai paling besar hingga paling kecil untuk menentukan dokumen mana yang paling atau lebih relevan dengan query.

BAB III

IMPLEMENTASI PROGRAM

Program kami memiliki inti algoritma untuk mencari dokumen yang sesuai dengan query menggunakan konsep temu-balik informasi dengan rumus cosine similarity yang berbentuk vektor. Pertama, kami membangun referensi dokumen sebanyak 15 dokumen menggunakan web scraping. Web scraping kami menggunakan library beautiful soup untuk parsing html. Dengan mengambil versi html dari apnews.com (Associated Press) kami dapat membentuk file versi teks (.txt) yang berisi artikel dari laman web tersebut. Lakukan sebanyak 15 kali dengan artikel yang berbeda-beda, maka diperoleh 15 dokumen sebagai referensi search engine.

Aplikasi materi temu-balik informasi terdapat pada file data.py. Data.py ini menjadi library kami yang berisikan beberapa prosedur dan fungsi, antara lain:

1. Fungsi stemstring

Fungsi ini mengambil suatu line, digunakan untuk mengambil line query. Setelah itu, line yang telah diambil akan dilakukan stemming yang akan menghasilkan kalimat tanpa tanda baca dan stopwords yang tidak penting.

2. Fungsi stemfiles

Fungsi ini mengambil suatu file, digunakan untuk mengambil file dokumen. Setelah itu, seluruh kalimat dari file tersebut akan dilakukan stemming yang akan menghasilkan kumpulan kalimat tanpa tanda baca dan stopwords yang tidak penting.

3. Fungsi get_unique

Fungsi ini akan mengambil seluruh kalimat yang telah distem (Dokumen dan query) kemudian seluruh kalimat tersebut akan digabung menjadi sebuah list of string.

4. Fungsi `get_bow`

Fungsi ini akan membentuk bag of words dengan tipe data dictionary dari masing-masing dokumen. Menggunakan parameter fungsi seluruh kalimat dalam suatu dokumen dan list of string unique dari fungsi sebelumnya, dapat dibuat sebuah dictionary yang berisikan keys dan values masing-masing untuk tiap-tiap dokumen.

5. Fungsi `bow_query`

Fungsi ini akan membentuk bag of words dengan tipe data dictionary dari query. Menggunakan parameter fungsi kalimat query dan list of string unique dari fungsi sebelumnya, dapat dibuat sebuah dictionary yang berisikan keys dan values masing-masing untuk query.

6. Fungsi `sim`

Fungsi ini akan menerima 2 dictionary (query dan dokumen). Dictionary yang diterima akan dihitung kesamaannya menggunakan rumus cosine similarity.

7. Fungsi `get_result`

Fungsi ini akan menerima bag of words dokumen dan bag of words query. Fungsi ini mengaplikasikan penggunaan fungsi `sim` (menghitung kesamaan). Akan dikembalikan `sorted_result` yang berbentuk list berisi tuple.

8. Prosedur `show_result`

Prosedur ini akan menerima `sorted_result` dan string query yang dimasukkan. Setelah itu, `sorted_result` akan dibentuk menjadi file html agar bisa ditampilkan di search engine.

9. Prosedur show_term

Prosedur ini akan menerima kalimat query, bag of words query, dan bag of words dokumen. Setelah itu, akan dibentuk file html yang berisikan list query dan dokumen beserta nama dokumen masing-masing yang selanjutnya akan ditampilkan di search engine.

10. Fungsi get_term_table

Fungsi ini menerima kalimat query, bag of words query, dan bag of words dokumen. Setelah itu, fungsi akan membentuk sebuah tabel yang berisi frekuensi munculnya masing-masing unique words dari query di masing-masing dokumen.

Setelah memiliki library data, kami langsung mengaplikasikan berbagai fungsi dan prosedur tersebut pada app.py yang berisikan back-end dari website kami. Kami menggunakan framework flask. Terdapat beberapa path website yang tersedia di sini, antara lain:

1. @app.route('/') dan @app.route('/search')

Secara default akan menampilkan search.html (front-end webpage search). Terdapat judul search engine di tengah dan tempat input query di kiri. Selain itu, terdapat pula hyperlink yang mengarahkan ke halaman upload dan halaman about. Jika query terisi dengan sebuah string, akan menampilkan results.html yang berisi hasil pencarian terdiri dari judul yang merupakan hyperlink ke dokumen teks, jumlah kata dokumen tersebut, tingkat kemiripan dengan query, serta kalimat pertama dokumen tersebut. List dokumen yang memiliki tingkat kesamaan lebih dari 0% akan diurutkan dari yang paling mirip.

2. @app.route('/upload')

Menampilkan upload.html yang berisi tempat upload dokumen. Jika berhasil mengupload dokumen, akan menampilkan upload_success.html yang berisikan pesan dokumen berhasil diupload.

3. `@app.route('/about')`

Menampilkan about.html yang berisikan deskripsi simpel dan anggota kelompok.

4. `@app.route('/table')`, `@app.route('/hasil')`, dan `@app.route('/list')`

Menampilkan termtree.html; hasil.html; list.html yang akan dipakai pada laman results.html.

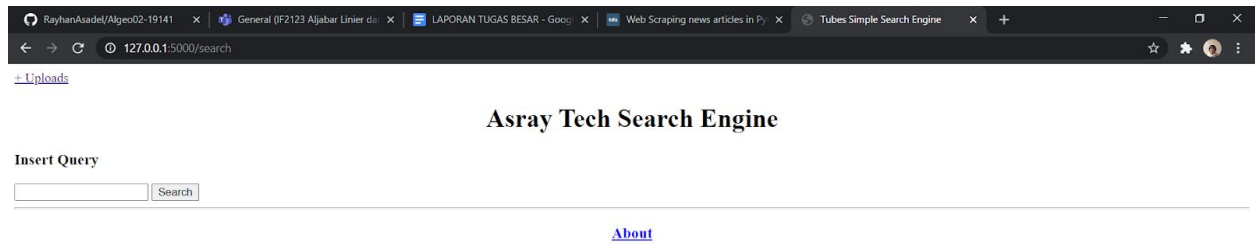
Adapun berbagai laman web berbasis html (berada di templates) yang kami miliki adalah sebagai berikut:

1. About.html
2. Hasil.html
3. List.html
4. Results.html
5. Search.html
6. Termtree.html
7. Upload.html
8. upload_success.html

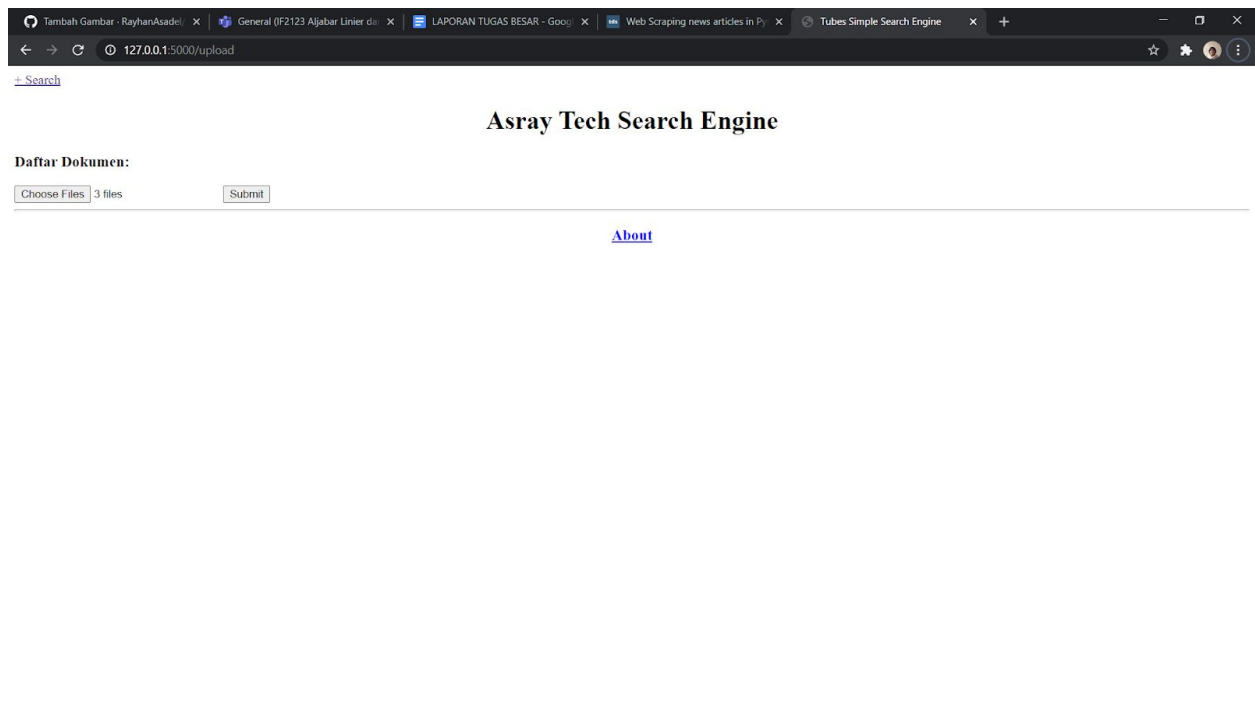
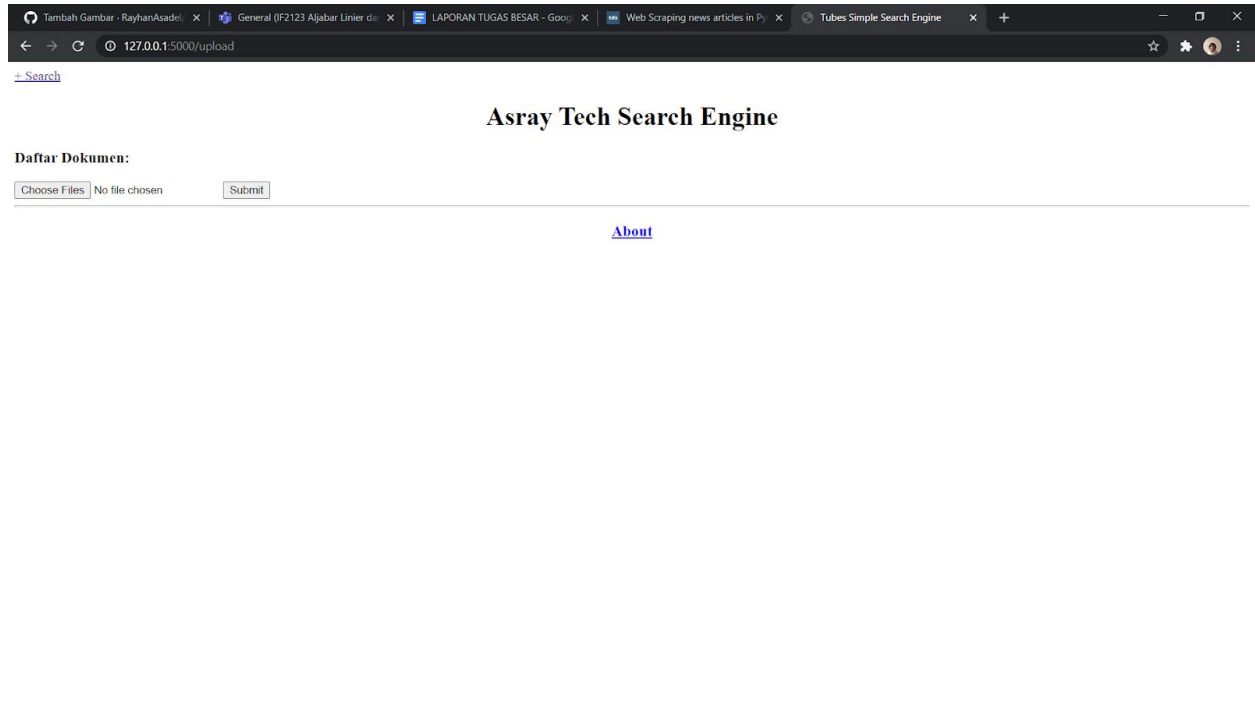
Selain itu, berbagai dokumen sebagai referensi berada pada di folder text yang berada di dalam static. Dokumen berbentuk teks (format .txt).

BAB IV

EKSPERIMEN



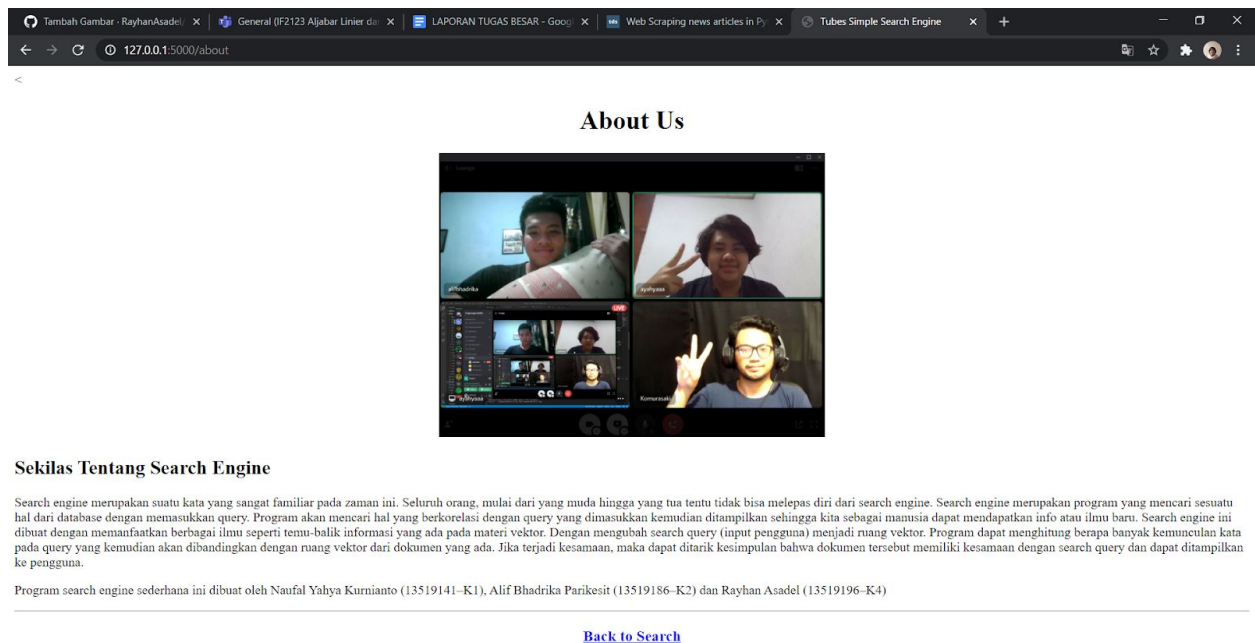
Halaman awal search engine. Terdiri dari hyperlink yang mengarahkan ke halaman upload. Selain itu, ada judul search engine yang berada di tengah dan tempat memasukkan query di bagian kiri, serta hyperlink ke halaman about.



Halaman upload dokumen. Terdapat hyperlink yang akan mengembalikan pengguna ke halaman awal. Selain itu ada tempat memasukkan beberapa dokumen di bagian kiri. Perhatikan ada perbedaan di gambar kedua, yaitu tulisan 3 files yang bermakna 3 file siap diupload.



Halaman upload dokumen setelah melakukan upload. Akan diberikan pesan bahwa file berhasil diupload, ditambah dengan hyperlink “OKE” yang akan mengarahkan pengguna kembali ke halaman upload awal.



Halaman About us. Berisikan review singkat tentang search engine, anggota kelompok tugas besar, dan foto anggota kelompok. Terdapat hyperlink yang akan mengarahkan pengguna kembali ke halaman awal yaitu halaman search.

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000/?s=ricciardo+renault'. The page title is 'Asray Tech Search Engine'. Below the title, there is a section titled 'Result for ricciardo renaul :'. It lists three search results with their respective titles, word counts, and snippets.

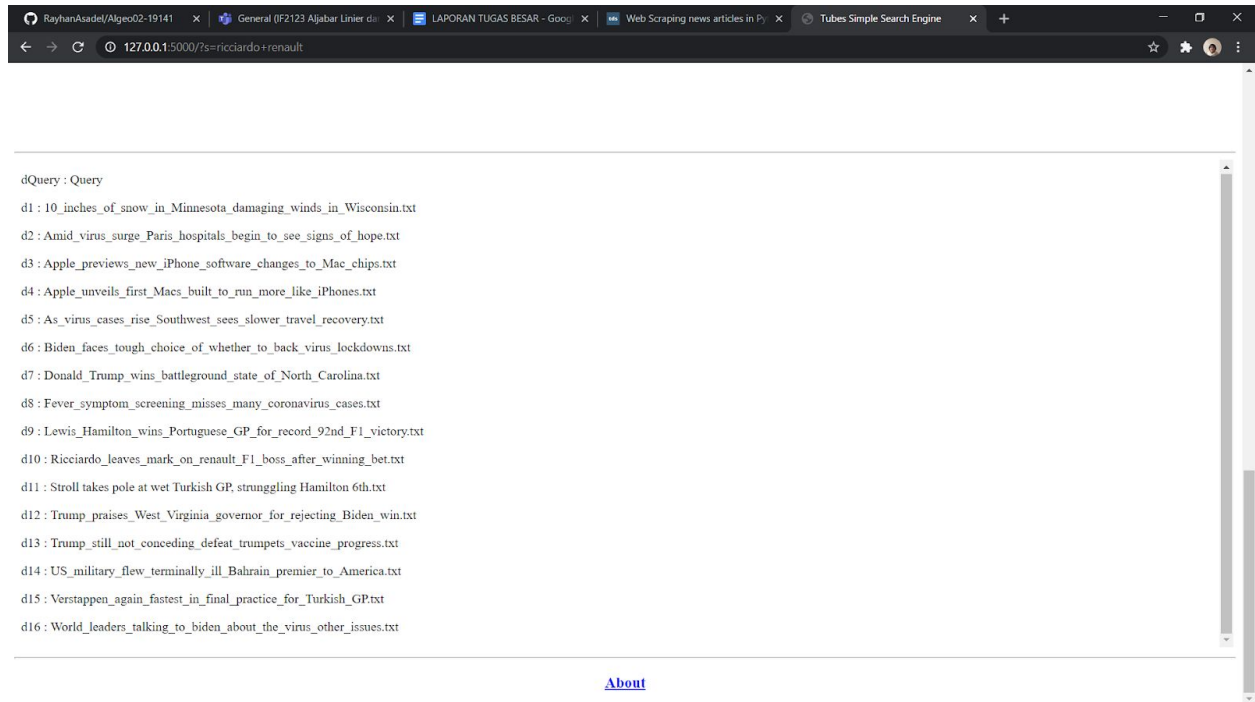
Result for ricciardo renaul :

- [Ricciardo leaves mark on renaul F1 boss after winning bet.txt](#)
Jumlah kata: 331
Tingkat Kemiripan: 59.44 %
NÜRBURG, Germany (AP) — Daniel Ricciardo's first podium finish for Renault comes with a special bonus. He gets to choose a tattoo for his boss.
- [Verstappen again fastest in final practice for Turkish GP.txt](#)
Jumlah kata: 268
Tingkat Kemiripan: 4.48 %
ISTANBUL (AP) — Red Bull driver Max Verstappen was again fastest in a cold and rain-soaked final practice for the Turkish Grand Prix ahead of qualifying later Saturday.
- [Stroll takes pole at wet Turkish GP, struggling Hamilton 6th.txt](#)
Jumlah kata: 877
Tingkat Kemiripan: 4.38 %
A rare sight in Formula One qualifying saw record-breaking Lewis Hamilton struggle and Lance Stroll tame a treacherous track to claim his first pole position on Saturday.

The screenshot shows the same web browser window with the address bar displaying '127.0.0.1:5000/?s=ricciardo+renault'. The page title is 'Asray Tech Search Engine'. Below the title, there is a section titled 'dQuery : Query'. It lists 15 search results with their respective titles, word counts, and snippets.

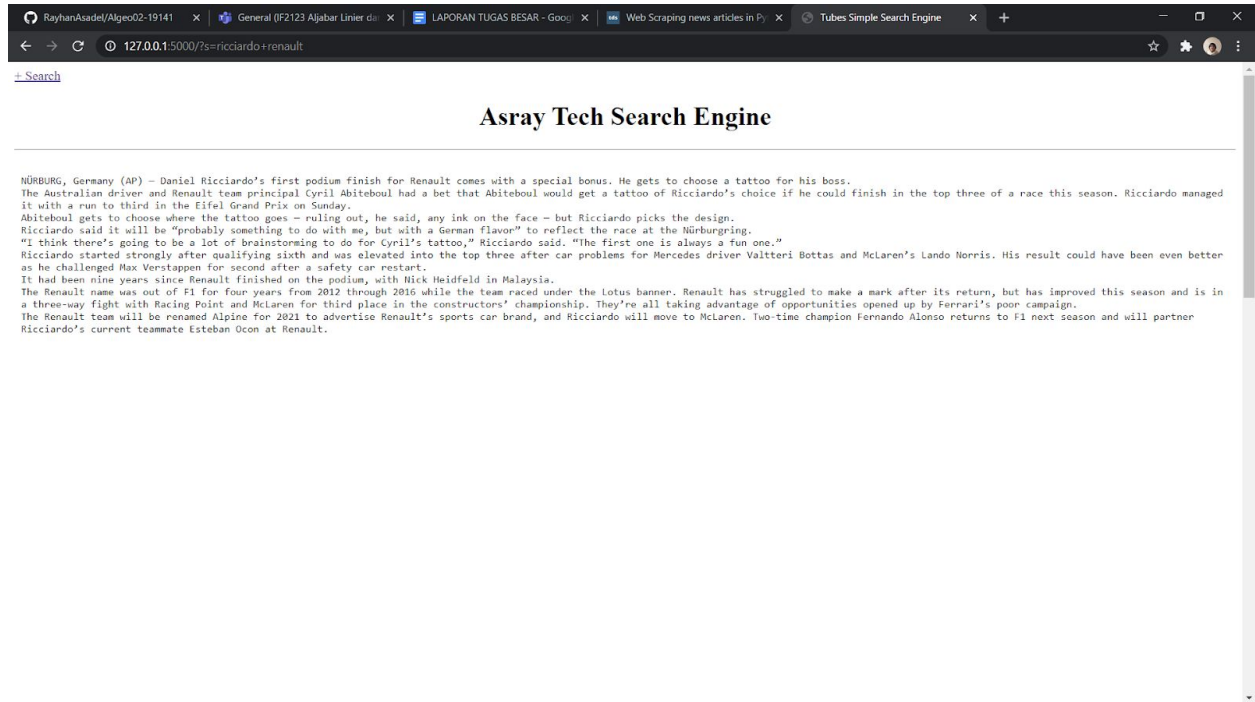
dQuery : Query

- d1 : 10_inches_of_snow_in_Minnesota_damaging_winds_in_Wisconsin.txt
- d2 : Amid_virus_surge_Paris_hospitals_begin_to_see_signs_of_hope.txt
- d3 : Apple_previews_new_iPhone_software_changes_to_Mac_chips.txt
- d4 : Apple_unveils_first_Macs_built_to_run_more_like_iPhones.txt
- d5 : As_virus_cases_rise_Southwest_sees_slower_travel_recovery.txt
- d6 : Biden_faces_tough_choice_of_whether_to_back_virus_lockdowns.txt
- d7 : Donald_Trump_wins_battleground_state_of_North_Carolina.txt
- d8 : Fever_symptom_screening_misses_many_coronavirus_cases.txt
- d9 : Lewis_Hamilton_wins_Portuguese_GP_for_record_92nd_F1_victory.txt
- d10 : Ricciardo_leaves_mark_on_renaul_F1_boss_after_winning_bet.txt
- d11 : Stroll_takes_pole_at_wet_Turkish_GP_struggling_Hamilton_6th.txt
- d12 : Trump_praises_West_Virginia_governor_for_rejecting_Biden_win.txt
- d13 : Trump_still_not_conceding_defeat_trumpets_vaccine_progress.txt
- d14 : US_military_flew_terminally_ill_Bahrain_premier_to_America.txt
- d15 : Verstappen_again_fastest_in_final_practice_for_Turkish_GP.txt

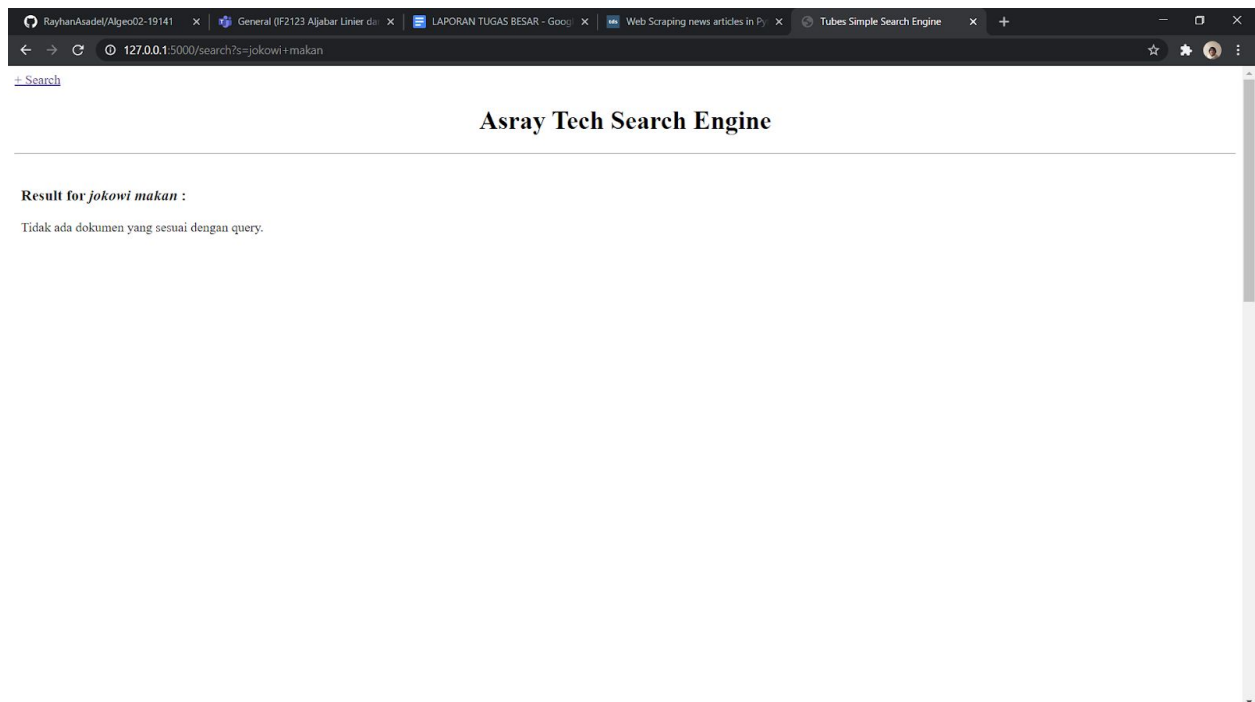


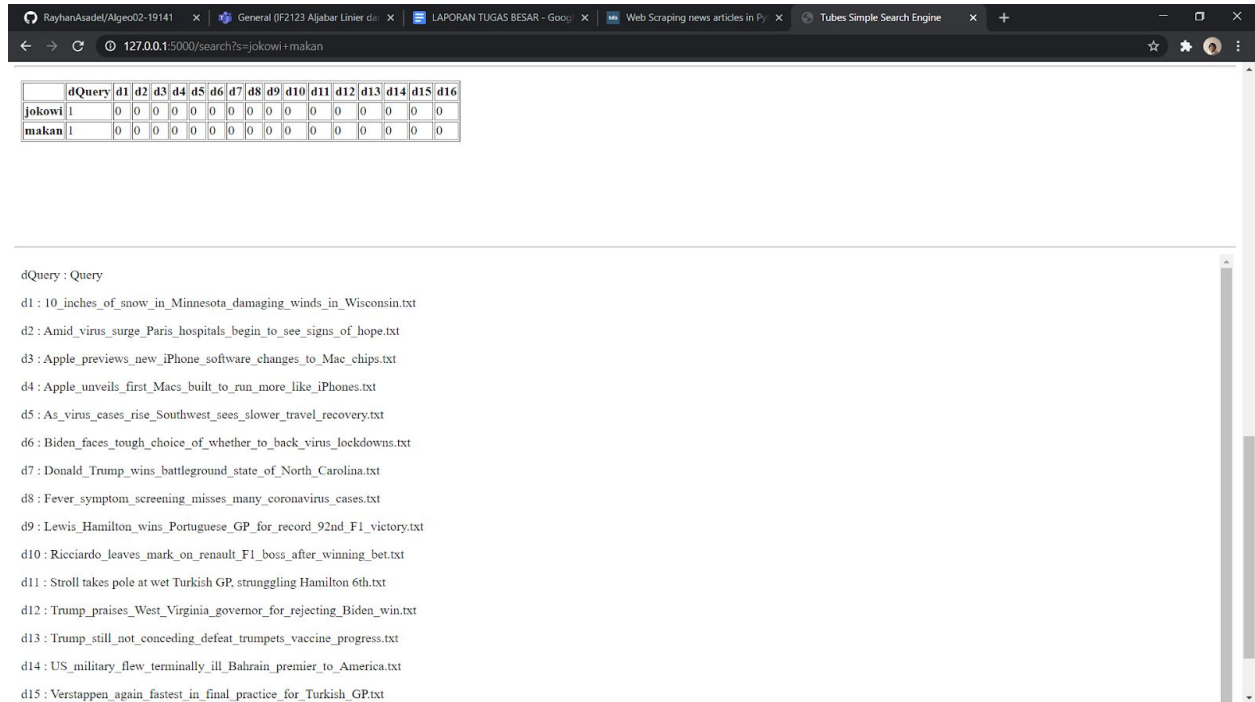
Halaman hasil pencarian. Menggunakan contoh query “ricciardo renauld”, diperoleh 3 dokumen yang memiliki tingkat kesamaan lebih dari 0%. 3 Dokumen tersebut ditampilkan terurut dari yang paling relevan dengan query. Perhatikan bahwa di tiap-tiap judul dokumen terdapat hyperlink yang akan mengarahkan pengguna ke dokumen terkait. Di bagian bawah terdapat tabel untuk term query. Ditampilkan berapa kali masing-masing term query muncul di tiap-tiap dokumen. Selain itu, ada pula list dokumen di bawahnya.

IF 2123 Aljabar Linier dan Geometri



Halaman yang memuat teks dokumen. Di halaman ini, yang merupakan satu halaman dengan halaman hasil pencarian, terdapat hyperlink yang mengarahkan pengguna ke halaman pencarian awal dan halaman about.





	dQuery	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12	d13	d14	d15	d16
jokowi	l	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
makan	l	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

dQuery : Query

d1 : 10_inches_of_snow_in_Minnesota_damaging_winds_in_Wisconsin.txt

d2 : Amid_virus_surge_Paris_hospitals_begin_to_see_signs_of_hope.txt

d3 : Apple_previews_new_iPhone_software_changes_to_Mac_chips.txt

d4 : Apple_unveils_first_Macs_built_to_run_more_like_iPhones.txt

d5 : As_virus_cases_rise_Southwest_sees_slower_travel_recovery.txt

d6 : Biden_faces_tough_choice_of_whether_to_back_virus_lockdowns.txt

d7 : Donald_Trump_wins_battleground_state_of_North_Carolina.txt

d8 : Fever_symptom_screening_misses_many_coronavirus_cases.txt

d9 : Lewis_Hamilton_wins_Portuguese_GP_for_record_92nd_F1_victory.txt

d10 : Ricciardo_leaves_mark_on_renault_F1_boss_after_winning_bet.txt

d11 : Stroll_takes_pole_at_wet_Turkish_GP_struggling_Hamilton_6th.txt

d12 : Trump_praises_West_Virginia_governor_for_rejecting_Biden_win.txt

d13 : Trump_still_not_conceding_defeat_trumpets_vaccine_progress.txt

d14 : US_military_flew_terminally_ill_Bahrain_premier_to_America.txt

d15 : Verstappen_again_fastest_in_final_practice_for_Turkish_GP.txt

Halaman ini merupakan hasil pencarian, jika term di query tidak ada yang terdapat di dokumen manapun. Perhatikan pada tabel bahwa term query hanya muncul di query, dan bernilai 0 di seluruh dokumen. Sama seperti halaman hasil pencarian normal, terdapat hyperlink yang mengarahkan ke halaman pencarian awal dan halaman about.

BAB V

KESIMPULAN, SARAN, DAN REFLEKSI

5.1 Kesimpulan

Search Engine atau sebuah Mesin Pencari, adalah suatu mesin/program yang dapat memanfaatkan sistem temu-balik informasi (information retrieval) untuk mengumpulkan kebutuhan informasi yang relevan, dengan keinginan pengguna. Konsep utama dari Mesin Pencari adalah dengan mengubah search query menjadi vector, dan membandingkan kemunculan kata-kata pada Query terhadap kata pada dokumen, untuk ditentukan dokumen mana yang paling relevan. Untuk menghitung korelevanan suatu dokumen dengan search query, digunakan metode cosine similarity.

5.2 Saran

Saran yang dapat kami berikan untuk pengembangan mesin pencari sederhana ini adalah dengan menggunakan algoritma pencarian yang lebih akurat, seperti TF-IDF. Kemudian, tampilan *web* dapat dikembangkan lagi dari sisi UI dan UX. Selain itu, metode pembuatan halaman hasil pencarian juga dapat dikembangkan agar tidak terkesan mengulangi manual tiap-tiap halaman html.

5.3 Refleksi

Dalam pengerjaan Tugas Besar ini pada awalnya terkesan susah dan tidak mengerti sama sekali, namun pada akhirnya jika mau berusaha dan mau belajar, akhirnya pun bisa juga baik mengerti dan mengerjakan, dan ternyata tidak sesusah atau serumit yang dipikirkan di awal pengerjaan. Selain itu, dalam waktu pengerjaan juga bisa dimulai lebih cepat lagi, namun karena kesibukan dan memerlukan waktu untuk belajar memahami materi Tugas, membuat pengerjaan tugas tidak bisa dilakukan langsung setelah tugas diumumkan. Hal lain yang kami dapatkan adalah bahwa kerja sama merupakan kunci. Tidak diharuskan semuanya menjadi benar-benar ahli, karena jika dilakukan bersama-sama kelemahan masing-masing orang dapat diisi oleh orang lainnya. Hal terakhir yang paling penting adalah jangan meremehkan tugas apapun dan anggota kelompok sendiri. Percayalah pada teman-teman bahwa mereka bisa dan akan membantu.

REFERENSI

Styling HTML Text Without CSS. Diakses 11 November 2020, dari

<https://stackoverflow.com/questions/21949198/styling-html-text-without-css>

TF-IDF and Cosine Similarity. Diakses 9 November 2020, dari

<https://janav.wordpress.com/2013/10/27/tf-idf-and-cosine-similarity/>

Tutorialspoint: HTML. Diakses 9 November 2020, dari

<https://www.tutorialspoint.com/html/index.htm>

Tutorialspoint: Flask. Diakses 9 November 2020, dari

<https://www.tutorialspoint.com/flask/index.htm>

Upload a File with Python Flask. Diakses 11 November 2020, dari

<https://pythonbasics.org/flask-upload-file/#:~:text=It%20is%20very%20simple%20to,it%20to%20the%20required%20location.>

How to scrape websites with Python and BeautifulSoup. Diakses 14 November 2020, dari

<https://www.freecodecamp.org/news/how-to-scrape-websites-with-python-and-beautifulsoup-5946935d93fe/>

Web Scraping news articles in Python. Diakses 14 November 2020, dari

<https://towardsdatascience.com/web-scraping-news-articles-in-python-9dd605799558>

Basic Tutorial

<https://www.geeksforgeeks.org/>

Reference Forum

<https://stackoverflow.com/>

NLTK Documentation

<https://www.nltk.org/>

Materi Kuliah IF2123 “Aplikasi Dot Product Pada Sistem Temu Balik Informasi”

<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-12-Aplikasi-dot-product-pada-IR.pdf>