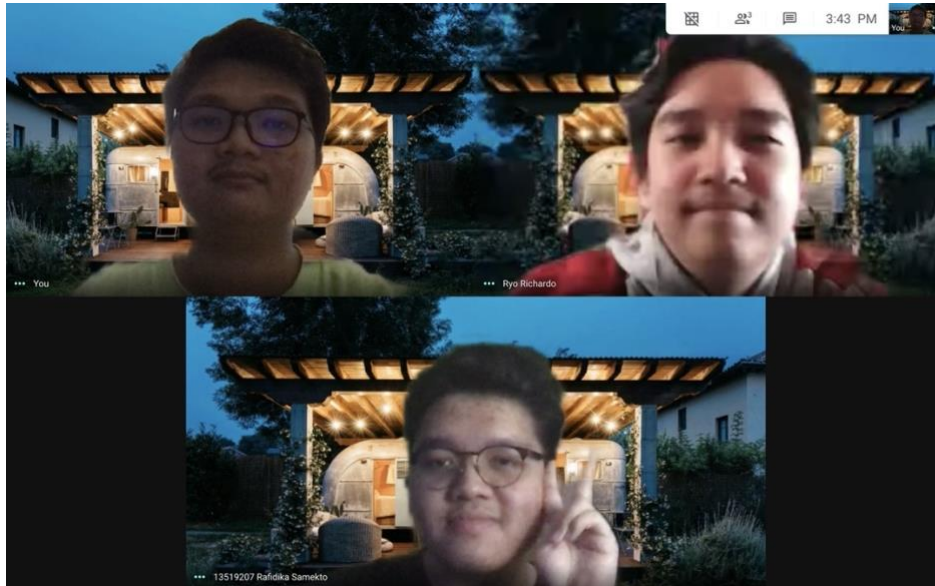


Laporan
Tugas Besar 2 IF2123 Aljabar Linier dan Geometri
Aplikasi Dot Product pada Sistem Temu-balik Informasi



oleh

Andres Jerriel Sinabutar	13519218
Ryo Richardo	13519193
Rafidika Samekto	13519207

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

2020

DAFTAR ISI

COVER	i
DAFTAR ISI	ii
BAB I DESKRIPSI MASALAH	1
1.1 Spesifikasi Tugas.....	1
1.2 Bahasa Program yang Digunakan dan Abstraksi Program.....	2
BAB II TEORI SINGKAT	3
2.1 Information Retrieval	3
2.2 Vektor	4
2.3 Cosine Similarity	5
BAB III IMPLEMENTASI PROGRAM	6
3.1 Front-End	6
3.1.1 HTML	7
3.1.2 CSS	8
3.2 Back-End.....	9
3.2.1 Driver/App	9
3.2.2 Cosine Similarity	10
3.2.3 Database Generator	11
BAB IV EKSPERIMEN	12
BAB V KESIMPULAN, SARAN, DAN REFLEKSI	17
5.1 Kesimpulan	17
5.2 Saran	17
5.3 Refleksi	18
DAFTAR REFERENSI	19

BAB I

DESKRIPSI MASALAH

1.1 Spesifikasi Tugas

Pada tugas besar ini, mahasiswa diminta untuk membuat aplikasi berbasis website yang dapat digunakan untuk melakukan pencarian terhadap kumpulan dokumen yang telah diunggah ke dalam web.

Spesifikasi program yang dibuat sebagai berikut:

1. Program mampu menerima search kueri. Search kueri 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.

1.2 Bahasa Program yang Digunakan dan Abstraksi Program

Pada pembuatan program ini, digunakan bahasa pemrograman Python. Untuk menjalankan program, pengguna perlu untuk mengunduh dan melakukan instalasi Python dan berbagai *library* yang digunakan. Ketika program dijalankan, program menampilkan dua isian kepada pengguna. Pertama, pengguna akan diminta untuk mengunggah dokumen-dokumen yang akan menjadi bahan pencarian kueri. Lalu, pengguna dapat mengetikkan kata kunci yang ingin dicari dalam kumpulan dokumen yang sudah diunggah. Selanjutnya program akan menjalankan pencarian sesuai dengan kueri pengguna dan akan menampilkan urutan dokumen yang paling sesuai dengan kueri pengguna.

BAB II

TEORI SINGKAT

2.1 Information Retrieval

Information Retrieval (IR) atau yang dalam Bahasa Indonesia disebut system temu-balik informasi adalah sistem yang dapat menemukan kembali informasi yang relevan terhadap kebutuhan pengguna dari kumpulan informasi secara otomatis. IR umumnya digunakan untuk mencari informasi dari sumber informasi yang bentuknya tidak terstruktur. Contoh sumber informasi yang tidak terstruktur adalah dokumen dan lama web (*webpage*), yang berbeda dengan informasi yang terstruktur seperti kumpulan data pada tabel yang tersimpan dalam basis data (*database*). Salah satu aplikasi IR yang paling terkenal saat ini adalah *search engine*.

Salah satu pemodelan IR adalah dengan memanfaatkan model ruang vektor. Model ini memanfaatkan teori dalam konsep aljabar vektor. Misalnya terdapat n kata berbeda yang dibangun dari setiap kata yang berada pada kueri maupun dokumen. Jumlah kata yang berbeda akan membentuk sebuah ruang vektor berdimensi n , dimana setiap dokumen maupun kueri akan membentuk vektor $w = (w_1, w_2, \dots, w_n)$ di dalam R^n . Nilai dari setiap w_i sendiri menyatakan jumlah kemunculan kata i pada dokumen atau kueri. Misalnya dalam sebuah kamus kata terbentuk ruang vektor berdimensi 3 dengan kata pertama adalah ayam, kata kedua adalah goreng, dan kata ketiga adalah sukabumi. Jika terdapat dokumen D_1 yang berisi “Sedia ayam goreng kalasan, ayam bakar madu, dan ayam geprek”, maka akan terbentuk suatu vektor D_1 yang bernilai $(3, 1, 0)$. Misalnya pengguna memasukkan kueri berupa kata kunci “ayam goreng sukabumi”, maka akan terbentuk suatu vektor Q yang bernilai $(1, 1, 1)$.

Penentuan dokumen yang relevan dengan kueri ditentukan dengan pengukuran kesamaan (*similarity measure*). Kesamaan antara vektor Q dengan vektor D₁ ditentukan dengan rumus *cosine similarity* yang berbunyi:

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

Jika nilai $\cos \theta = 1$, berarti $\theta = 0$ yang berarti vektor Q dan D berhimpit. Jadi, semakin besar nilai cosinus (semakin mendekati 1) menunjukkan dokumen semakin mirip dengan kueri.

2.2 Vektor

Vektor adalah kuantitas fisik yang memiliki besar dan arah. Penulisan vektor biasanya dilambangkan dengan huruf-huruf kecil yang dicetak tebal atau dengan anak panah di atasnya. Vektor didefinisikan dalam ruang vektor atau ruang Euclidean, yang dinotasikan sebagai \mathbb{R}^n . Sebagai contoh, vektor \vec{v} didefinisikan dalam \mathbb{R}^3 sebagai (v_1, v_2, v_3) . Vektor memiliki sifat-sifat yang mirip dengan bilangan biasa, seperti penjumlahan, pengurangan, perkalian dengan skalar.

Identitas sebuah vektor ditentukan panjang dan arahnya. Panjang sebuah vektor didefinisikan sebagai akar dari jumlah kuadrat setiap elemennya, sehingga vektor $\vec{v} = (v_1, v_2, v_3)$ memiliki panjang $\sqrt{v_1^2 + v_2^2 + v_3^2}$. Arah dari suatu vektor \vec{v} didefinisikan dengan \hat{v} (baca: v topi) yang bernilai $\frac{\vec{v}}{\|\vec{v}\|}$. Selain itu, pada ilmu vektor juga terdapat relasi antara dua vektor yaitu perkalian titik (*dot product*), yang berbunyi:

$$\vec{v} \cdot \vec{w} = \|\vec{v}\| \|\vec{w}\| \cos \theta$$

Dengan menggunakan perkalian titik kita dapat menentukan besar sudut yang terbentuk antara dua vektor yang berpotongan pada suatu titik.

2.3 Cosine Similarity

Seperti yang sudah dijelaskan di atas, rumus *cosine similarity* didapatkan dari relasi perkalian titik antara dua vektor. Rumus *cosine similarity* berbunyi sebagai berikut:

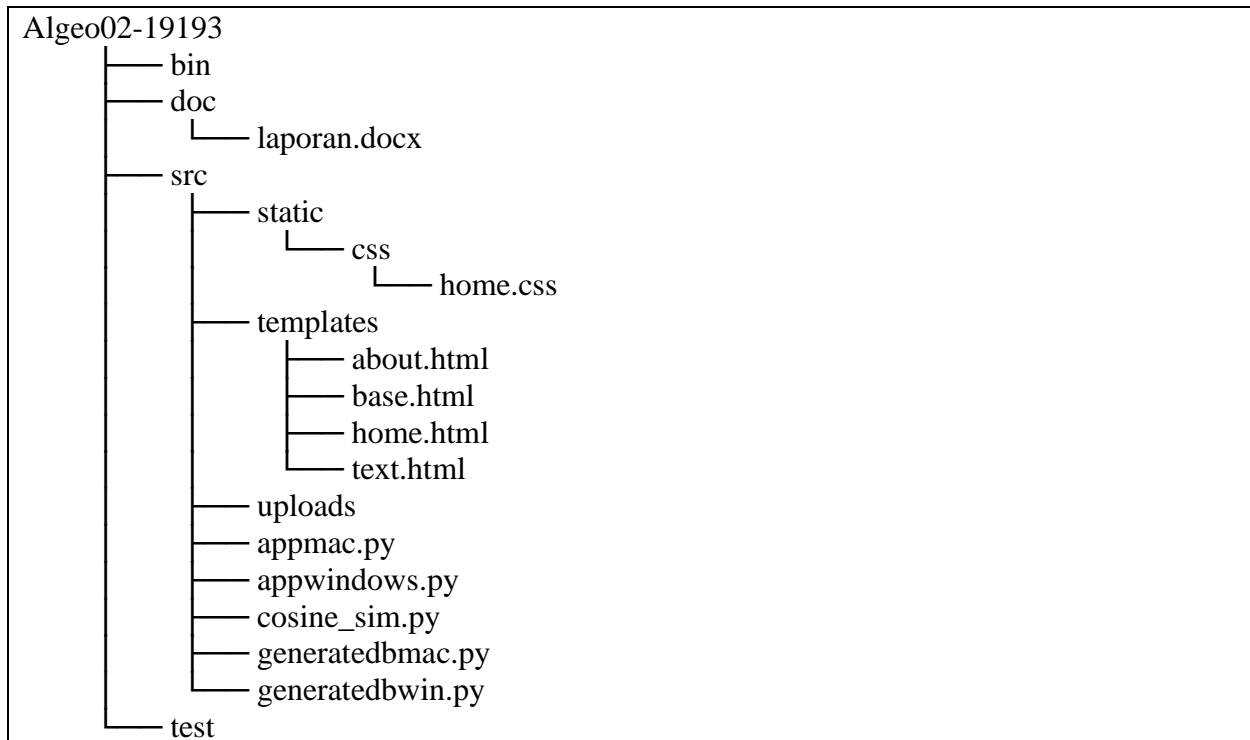
$$\text{sim}(A, B) = \cos \theta = \frac{A \cdot B}{\|A\| \|B\|}$$

Nilai similaritas akan semakin tinggi apabila nilai $\cos \theta$ semakin mendekati 1, atau nilai θ semakin mendekati 0. Maka, nilai similaritas akan mencapai puncak apabila vektor A dan vektor B berhimpit.

BAB III

IMPLEMENTASI PROGRAM

Secara garis besar, struktur dari program kami dapat dijelaskan dalam diagram berikut:



3.1 Front-End

Bagian *front-end* program berada di folder "static/css" dan folder "templates". Folder "templates" berisi seluruh file HTML yang digunakan untuk menampilkan fungsi utama program, seperti menampilkan dokumen-dokumen yang diunggah, mencari kata dalam dokumen, dan mengurutkan hasil-hasil pencarian dokumen berdasarkan similaritasnya dengan *query* yang dimasukkan. Folder "static/css" berisi file CSS untuk menghias file HTML agar terkesan "hidup". Kami tidak menggunakan JavaScript untuk bagian *front-end* karena kami merasa JavaScript tidak terlalu diperlukan untuk membuat sebuah *website* yang sederhana.

3.1.1 HTML

Demi menyelaraskan dengan *back-end*, kami membuat sebuah file HTML yang digunakan sebagai "template" untuk file HTML lain yang akan dibuat dan file tersebut kami namakan "base.html". File ini akan di-*extends* menjadi "home.html", "about.html", dan "text.html". "home.html" adalah file HTML utama karena disinilah tempat fungsi-fungsi utama program ditampilkan. Tampilan file bisa berubah bergantung pada interaksi apa yang dilakukan pengguna saat program dijalankan. "about.html" menampilkan konsep singkat *search engine* yang dibuat, profil singkat tiap anggota kelompok kami, dan cara menggunakan program kami. "text.html" digunakan untuk menampilkan dokumen yang diunggah.

Pada saat menjalankan program untuk pertama kalinya, pengguna akan diarahkan ke halaman utama dan diminta untuk mengunggah dokumen bertipe ".txt" atau memasukkan *link* Wikipedia. Sesaat setelah dokumen diunggah, pengguna dapat melihat tabel dengan 3 kolom. Kolom-kolom tersebut digunakan untuk menampilkan info mengenai dokumen yang diunggah, yaitu nama dokumen dan kapan dokumen diunggah. Kolom terakhir berfungsi untuk menampilkan aksi-aksi yang bisa dilakukan terhadap dokumen, yaitu membuka dan menghapus dokumen dari *database*. Pengguna kemudian dapat mencari kata yang ada pada dokumen dan mencarinya. Tampilan "home.html" akan berubah dan menampilkan dokumen-dokumen tadi serta mengurutkannya berdasarkan tingkat kesamaannya dengan *query* yang dimasukkan. Pada akhir tampilan (bagian paling bawah), terdapat *link* yang mengarahkan pengguna ke halaman "about.html".

File "about.html" dan "text.html" berfungsi dengan lebih sederhana ketimbang "home.html" karena keduanya tidak memerlukan "bantuan" *back-end* sebanyak halaman utama HTML kami. "about.html" hanya menampilkan info-info yang tidak perlu berubah sama

sekali, seperti cara kerja program, profil pembuat, dan konsep singkat prgram. "text.html" dibuat untuk menampilkan dokumen yang diunggah.

3.1.2 CSS

Pada program ini, kami hanya menggunakan satu file CSS bernama "home.css" yang mencakup seluruh keperluan *styling* untuk tiga file HTML kami. File ini terletak pada folder "static/css" dan bersama dengan program, terdapat dua gambar yang digunakan sebagai *icon* pada *title bar* (gambar pada *tab browser*) dan gambar *background* untuk keperluan styling file HTML kami. *Styling* yang kami lakukan tidak murni hanya ada pada file CSS saja. *Styling* lain adalah dengan meng-"include" sebuah font dari Google Fonts dan kami link pada "base.html" sehingga font tersebut dapat berlaku secara global pada program kami. Kami juga melakukan *linking* dengan gambar yang digunakan sebagai *icon* pada *title bar* di "base.html".

Tema yang kami gunakan disini adalah *game* Among Us. Oleh karena itu, kami memilih *background image* hitam dengan dot-dot putih yang mirip dengan gambar saat seorang *impostor* dikeluarkan dari *spaceship* The Skeld atau gambar di halaman utama *game*. Font yang kami pilih bernama Amatic SC yang mirip (atau bahkan sama) dengan font yang digunakan pada *game*.

Di bagian awal, kami melakukan styling untuk seluruh elemen HTML, yaitu dengan mengaplikasikan *background image* dan mengubah font. Secara umum, styling yang kami lakukan bertujuan agar sebagian besar elemen dapat diposisikan di tengah layar dan dapat dilihat dengan jelas. Oleh karena itu, banyak dari elemen HTML kami *style* dengan margin kiri dan kanan di-*set* sebagai *auto* dan warna elemen di-*set* sebagai putih (ini disebabkan *background image* yang kami gunakan berwarna hitam yang sangat dominan). Untuk bagian hasil pencarian, kami *style* sedemikian rupa sehingga mirip dengan berbagai *search engine* yang biasa ditemukan seperti

Google dan Bing, yaitu dengan menggeser seluruh elemen sedikit ke kanan (dari yang seharusnya rata kiri dengan layar).

3.2 Back-End

Pada bagian *back-end* program dibagi menjadi 3 bagian, *driver/app* (appmac.py dan appwindows.py), *cosine similarity* (cosine_sim.py), dan *database generator* (generatedbmac.py dan generatedbwin.py).

3.2.1 Driver/app

Pada bagian *driver* terdapat semua *class* dan *route* yang dijalankan sebagai program utama. Bagian *driver* dibedakan menjadi appmac.py dan appwindows.py karena terdapat perbedaan dalam sistem penamaan file pada sistem operasi MacOS dan Windows. Mula-mula *driver* perlu mengimport beberapa library, seperti os, requests, flask, werkzeug.utils (secure_filename), flask_sqlalchemy (SQLAlchemy), datetime, pathlib (Path), bs4 (BeautifulSoup), numpy, dan pandas. Pada *driver* terdapat *class* Document dengan atribut 'name' sebagai nama file dan 'date_uploaded' sebagai tanggal dokumen tersebut diunggah. Kemudian pada *driver* juga terdapat lima macam *route*, yaitu:

- a) Home. Mula-mula *route* ini akan memeriksa apakah ada file pada request.files. Jika tidak ada, maka *route* akan mencoba untuk meminta url dari form input url. Jika berhasil, maka *route* akan menggunakan library BeautifulSoup untuk mengkonversi artikel dalam html menjadi text, yang akan disimpan dalam file .txt dalam folder 'uploads'. *Route* kemudian membuat variabel dengan *class* Document yang berisikan nama file tersebut, yang akan diunggah dalam *database*. Percabangan berikutnya ketika *route* menemukan file pada request.files, maka *route* akan membuat variabel dengan *class* Document yang berisikan

nama file tersebut, dan kemudian diunggah ke *database*. *Route* juga akan menyimpan file yang diunggah ke dalam folder ‘uploads’. Terakhir, *route* akan mengurutkan file berdasarkan tanggal file tersebut diunggah, dan kembali ke laman home.html.

- b) Delete. *Route* ini akan meminta nama file yang ingin dihapus. Kemudian, *route* akan mencoba untuk menghapus file pada folder ‘uploads’ dan pada *database*. Jika berhasil maka akan kembali ke laman home.html.
- c) Uploads. *Route* ini akan membuka file yang dituju dan membaca teks di dalam file. *Route* akan menampilkan teks tersebut di laman text.html.
- d) Uploads. *Route* ini akan mengunggah file dari lokal ke folder ‘uploads’.
- e) Search. Tujuan dari *route* ini adalah membuat list bernama ‘array’ dengan elemen yang memiliki atribut ‘name’ (nama file), ‘path’ (alamat file), ‘first’ (kalimat pertama), ‘text’ (teks), ‘count’ (jumlah kata), dan ‘cos’ (nilai *cosine similarity* dalam persen). *Route* ini juga akan membuat tabel bernama ‘tftable’ yang berisi data kemunculan setiap kata unik dari kueri pada masing-masing file. *Route* ini mengimport fungsi-fungsi dari file cosine_sim.py untuk membantu dalam melakukan *case folding* dan *stemming*. List dan tabel tersebut nantinya akan diekspor ke home.html agar bisa disajikan dalam bentuk list dokumen dan tabel kemunculan kata kueri.
- f) About. *Route* ini akan aktif jika pengguna menekan tombol ‘Tentang’. *Route* ini akan mengarahkan pengguna ke laman about.html.

3.2.2 Cosine similarity

File cosine_sim.py berisi fungsi-fungsi yang berguna untuk melakukan *case folding* dan *stemming*. Untuk membantu merealisasikan fungsi tersebut, diperlukan beberapa library seperti Sastrawi, nltk, re, string, pathlib (Path), dan ntpath. Fungsi-fungsi tersebut adalah:

- a) Transpose; berfungsi untuk me-transpose tabel pada *route* 'Search' yang masih terbalik antara baris dan kolomnya. Fungsi transpose ini dibuat agar penambahan list jumlah kata ke dalam tabel 'tftable' dapat berjalan dengan lebih efisien.
- b) Path_leaf; berfungsi untuk mengambil nama file dari alamat tempat file itu disimpan.
- c) TakeCos; berfungsi untuk mengambil atribut 'cos' pada elemen.
- d) Take1sentence; berfungsi untuk mengambil kalimat pertama pada teks jika teks tersebut memiliki lebih dari 100 karakter. Satu kalimat didefinisikan sebagai rangkaian kata hingga mencapai tanda titik ('.') untuk pertama kali.
- e) Tokenize; berfungsi untuk mengubah teks menjadi token kata.
- f) Case_folding; berfungsi untuk mengubah teks menjadi huruf kecil, membuang angka yang tidak penting dan membuang tanda baca.
- g) Filtering; berfungsi untuk menghapus kata hubung (di, ke, dan, dari, dll).
- h) Stemming; berfungsi untuk melakukan *stemming*.
- i) Cosine_sim; berfungsi untuk menghitung nilai *cosine similarity* dan memberikan list jumlah kemunculan setiap elemen pada vektor pertama di vektor kedua.

3.2.3 Database generator

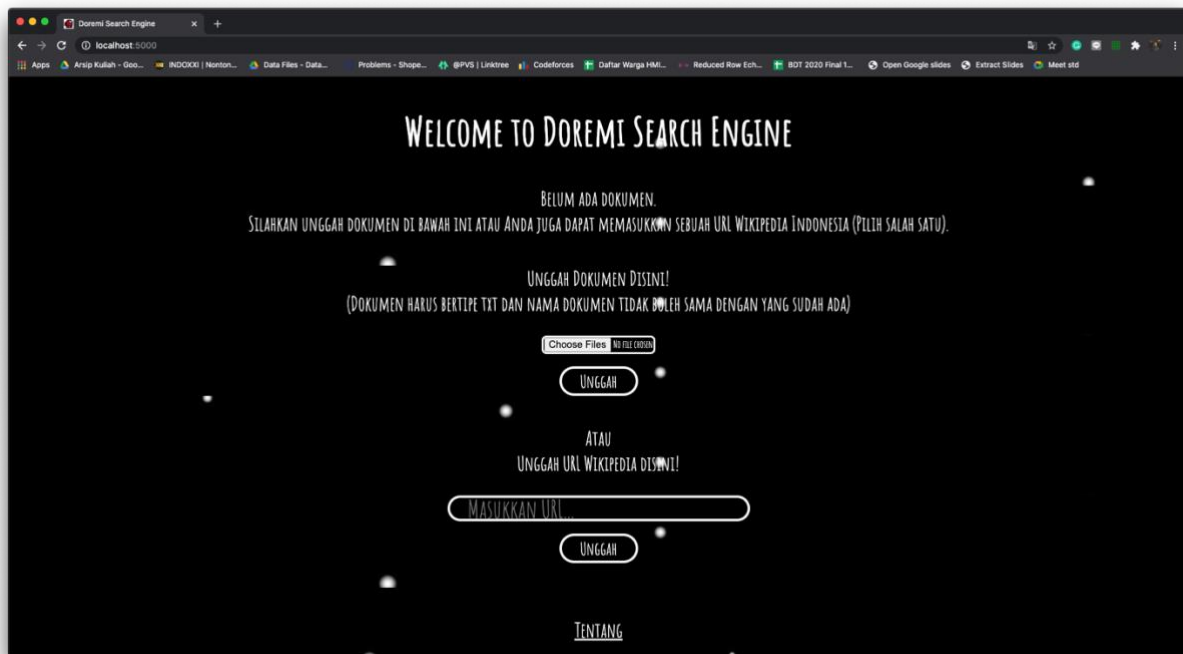
Database generator dibagi menjadi *generatedbmac.py* dan *generatedbwin.py* karena terdapat perbedaan versi *driver* untuk sistem operasi MacOS dan Windows. *Database generator* akan mengimport 'db' dari *driver*, lalu mengaktifkannya sebagai *database*.

BAB IV

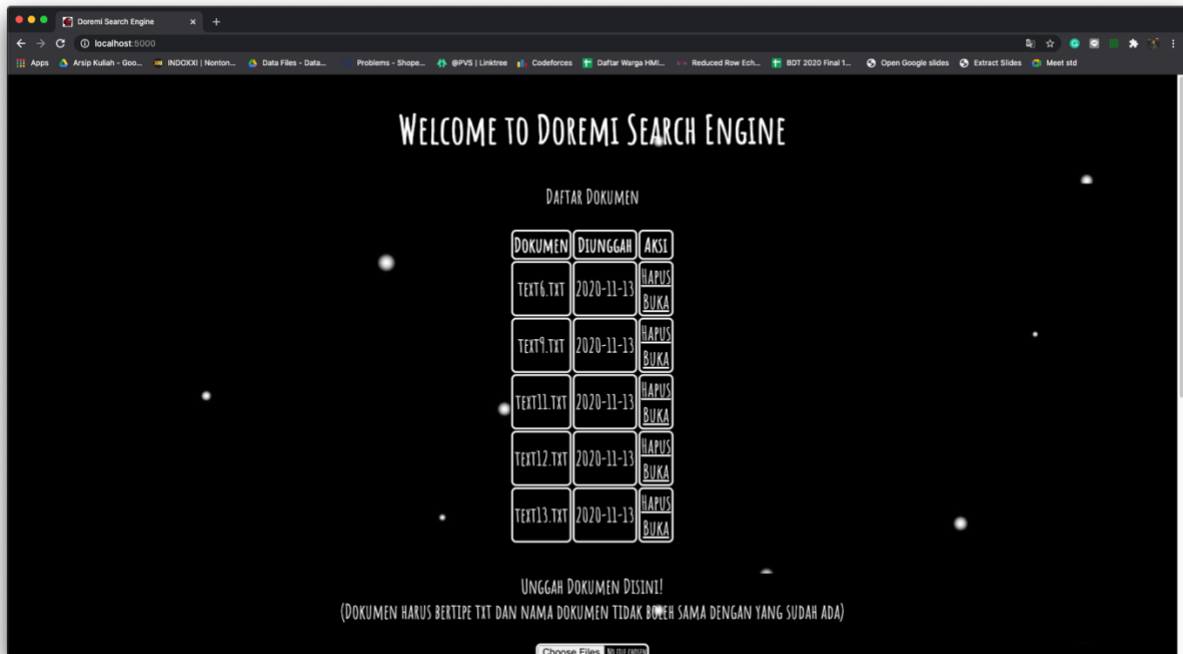
EKSPERIMEN

Berikut adalah implementasi dari program yang kami buat:

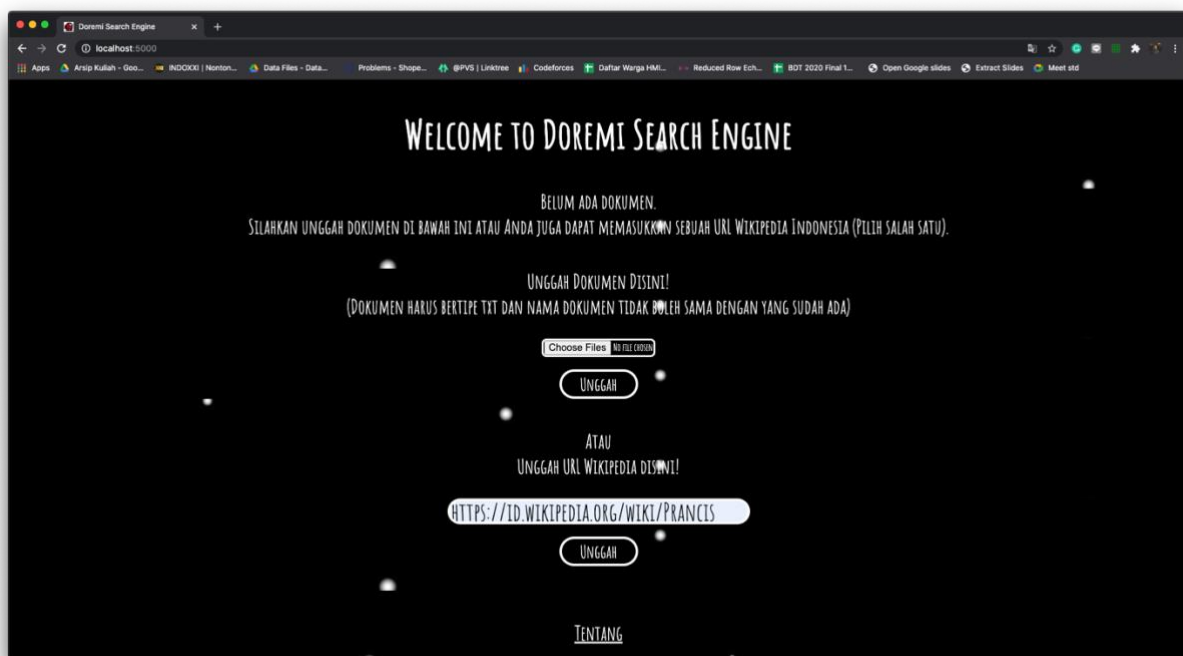
1. Ini adalah tampilan halaman utama dari web search engine



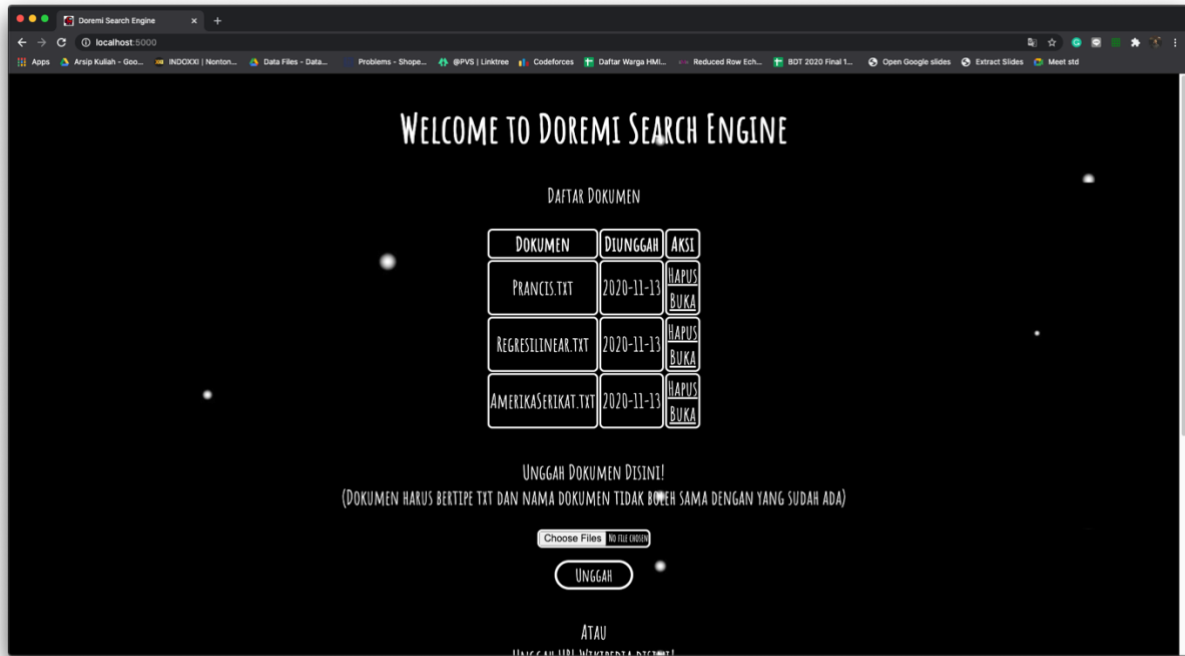
2. Beberapa dokumen berbentuk .txt diunggah ke database



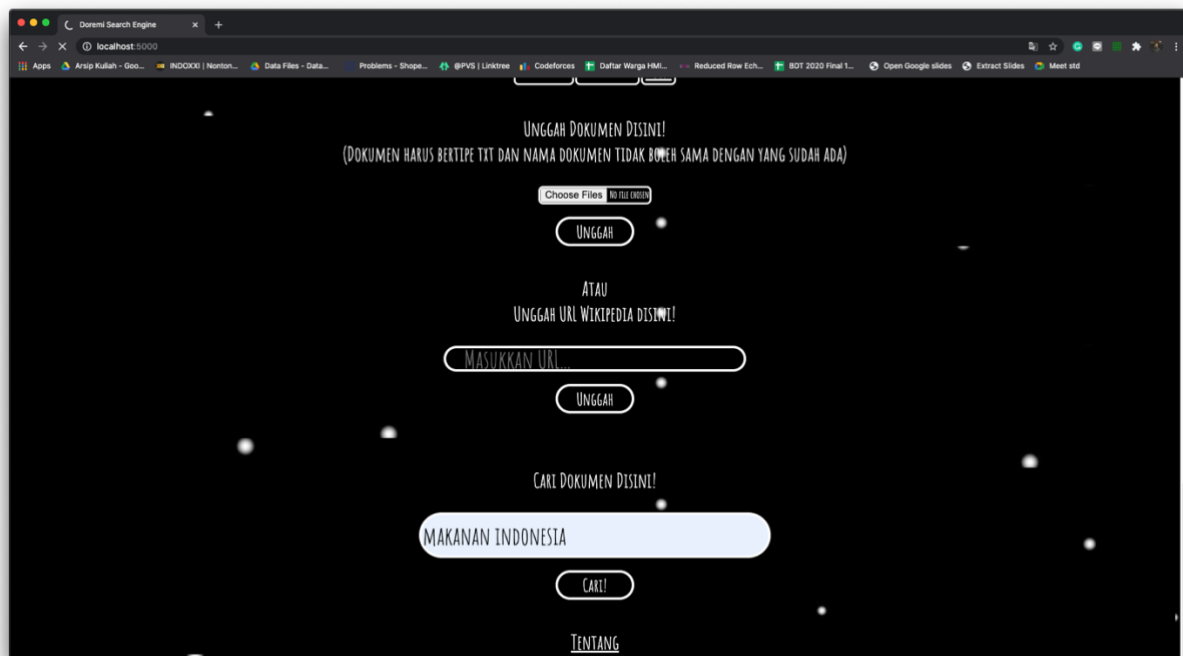
3. Kita juga dapat mengunggah sebuah situs Wikipedia Indonesia sebagai sumber dokumen untuk bahan pencarian



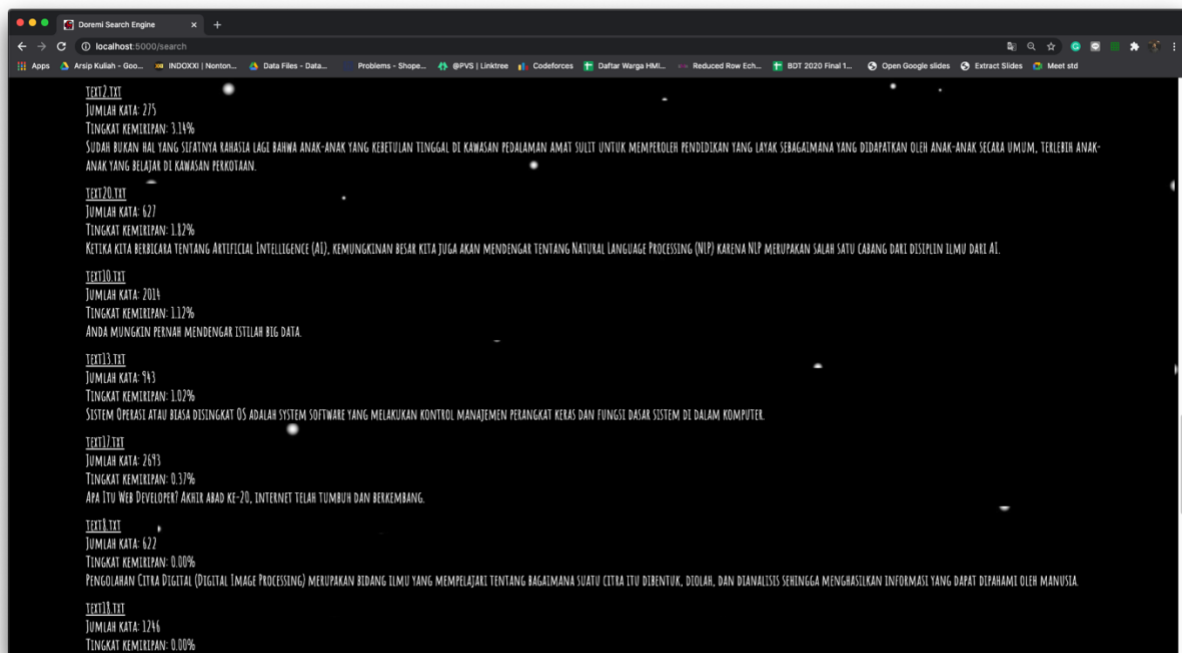
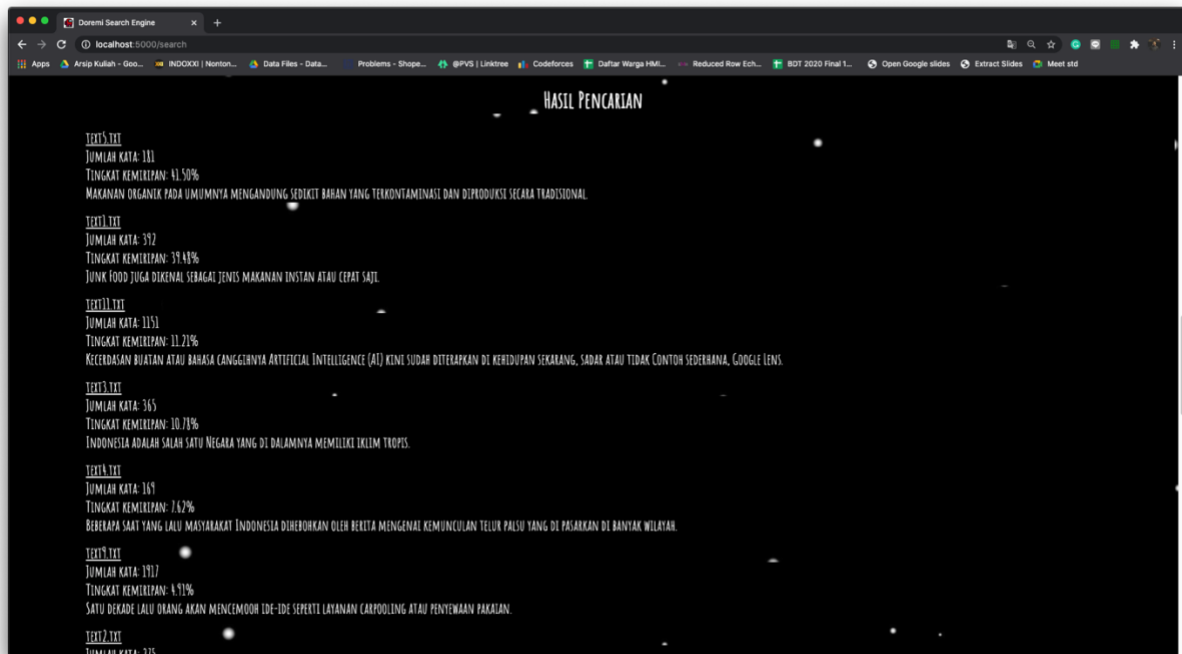
4. Web Wikipedia yang telah berhasil diunggah akan ditampilkan dalam daftar dokumen

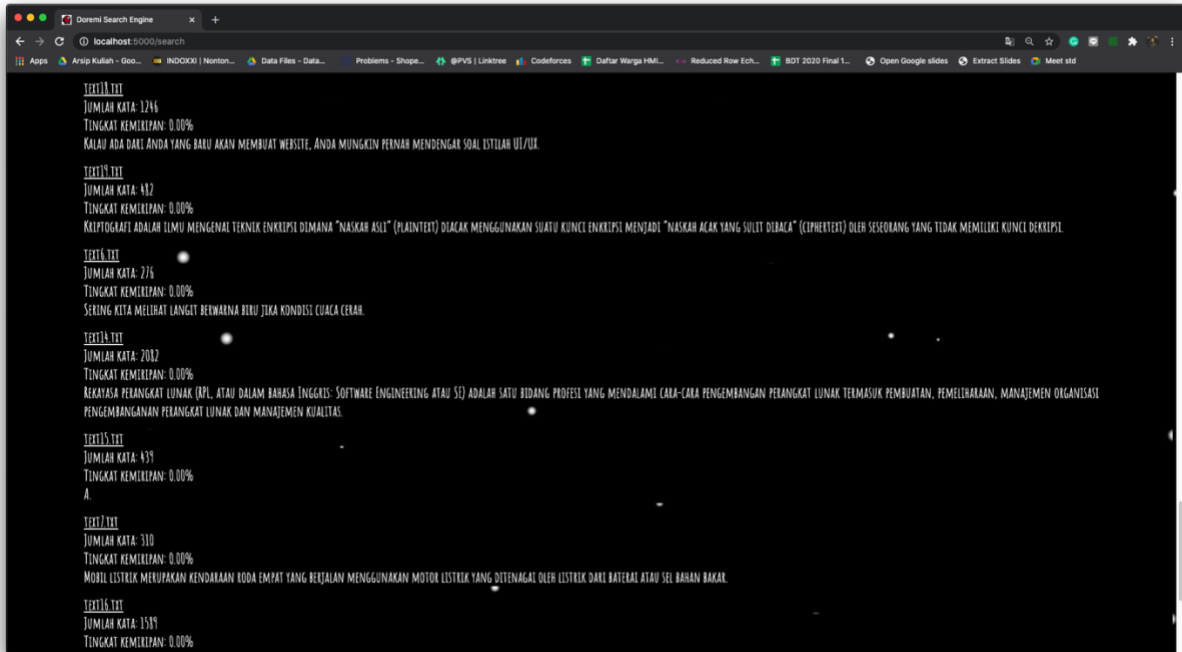


5. Selanjutnya, setelah semua dokumen telah diunggah, masukkan kueri yang ingin dicari, lalu klik tombol Cari.



6. Tunggu beberapa saat, hasil pencarian dari dokumen yang telah diunggah akan muncul.





7. Selain itu, tabel frekuensi dari setiap term yang dimasukkan dalam kueri akan ditampilkan.

Doremi Search Engine

x

+

localhost/search

Apps

Arjip Kullah - Geo...

INDOXXI | Norton...

Data Files - Data...

Problems - Shope...

@PVS | Linktree

Codeforces

Daftar Warga HML...

Reduced Row Ech...

BDT 2020 Final 1...

Open Google slides

Extract Slides

Meet stl

Tingkat kemiripan: 0.00%

REKAYASA PERANGKAT LUNAK (RPL, atau dalam bahasa Inggris: Software Engineering atau SE) adalah satu bidang profesi yang mendalami cara-cara pengembangan perangkat lunak termasuk pembuatan, pemeliharaan, manajemen organisasi pengembangan perangkat lunak dan manajemen kualitas.

[TEXT15.TXT](#)

Jumlah kata: 439

Tingkat kemiripan: 0.00%

A.

[TEXT7.TXT](#)

Jumlah kata: 310

Tingkat kemiripan: 0.00%

Mobil listrik merupakan kendaraan roda empat yang berjalan menggunakan motor listrik yang ditenagai oleh listrik dari baterai atau sel bahan bakar.

[TEXT16.TXT](#)

Jumlah kata: 1589

Tingkat kemiripan: 0.00%

1.

[TEXT17.TXT](#)

Jumlah kata: 651

Tingkat kemiripan: 0.00%

Pemrograman Berorientasi Object atau dalam bahasa Inggris lebih dikenal dengan Object Oriented Programming (OOP) adalah sebuah paradigma dalam pemrograman yang menyelesaikan masalah program dengan menyediakan objek-objek (terdiri dari beberapa attribute dan method) yang saling berkaitan dan disusun kedalam satu kelompok atau yang disebut dengan kelas.

TABEL QUERY

TERM	QUERY	TEXT20.TXT	TEXT1.TXT	TEXT8.TXT	TEXT18.TXT	TEXT11.TXT	TEXT6.TXT	TEXT14.TXT	TEXT15.TXT	TEXT7.TXT	TEXT5.TXT	TEXT17.TXT	TEXT16.TXT	TEXT4.TXT	TEXT12.TXT	TEXT1.TXT	TEXT13.TXT	TEXT11.TXT	TEXT3.TXT	TEXT2.TXT	TEXT10.TXT
MAKAN	1	1	0	0	0	0	0	0	0	12	0	0	1	0	20	1	0	0	0	0	
INDONESIA	1	0	6	0	0	0	0	0	0	0	0	1	0	1	0	1	0	12	4	1	2

TENTANG

BAB V

KESIMPULAN, SARAN, DAN REFLEKSI

5.1 Kesimpulan

Dari proses pembuatan program dan eksperimen yang dilakukan, penulis mendapat beberapa kesimpulan, yaitu:

1. Bahasa pemrograman Python dengan *library* Flask sangat berguna dalam pembuatan *website* terlebih *website* yang membutuhkan manajemen *database* dan sinkronisasi antara *backend* (program utama) dan *frontend* (tampilan dan interaksi program dengan pengguna).
2. Aplikasi ilmu Aljabar Linier dan Geometri terutama vektor dan fungsi *cosine similarity* sangat berguna dan dapat membantu membantu manusia terutama dalam implementasinya sebagai sistem temu-balik informasi.

5.2 Saran

Tim penulis memiliki beberapa saran mengenai pengembangan program ini, yaitu

1. Algoritma yang dijelaskan dalam laporan tugas akhir ini masih memiliki banyak kemungkinan untuk dilakukan efisiensi. Oleh karena itu, dalam pengembangan program ini, masih bisa dilakukan efisiensi kinerja.
2. Program ini bisa dikembangkan lebih lanjut dengan menyempurnakan antarmuka grafis sehingga menjadi lebih fungsional dan lebih memperhatikan *user experience*.
3. Menimbang diperlukannya program ini dalam banyak mata kuliah di ITB, sebaiknya program ini juga dipublikasikan, setidaknya kepada khalayak ITB supaya program ini memiliki nilai kebermanfaatan yang lebih besar.

5.3 Refleksi

Dari proses pembuatan program dan eksperimen yang dilakukan, penulis merasa bahwa:

1. Pembagian tugas di awal dan linimasa pengerjaan sudah ditetapkan secara rinci dan jelas dan pada pelaksanaannya, linimasa yang telah ditetapkan dapat diikuti sebagaimana yang telah direncanakan.
2. Komunikasi antar anggota kelompok berjalan dengan baik, sehingga tidak terjadi miskomunikasi atau kesalahpahaman selama pengerjaan.
3. Selama proses pembuatan program, ketika ditemukan ketidaksesuaian saat proses eksperimen, temuan langsung dikomunikasikan kepada anggota kelompok lainnya dan bersama-sama mencari solusi.
4. Program telah berhasil dibuat sesuai dengan spesifikasi dan telah berhasil dilakukan uji coba dengan beberapa *testcase*.

DAFTAR REFERENSI

- Anton, Howard dan Chris Rorres. 2010. *Elementary Linear Algebra: Application Version*. USA: John Wiley & Sons
- Lyawati. 2020. *Aplikasi Aljabar Linear pada Kriptografi [Skripsi]*. Yogyakarta (ID): Universitas Sanata Dharma.
- Marsudi, Marjono. 2012. *Aljabar Linear*. Malang: Universitas Brawijaya Press.
- Munir, Rinaldi. 2020. *Aplikasi Dot Product pada Sistem Temu-balik Informasi*.
- Sibaroni, Yuliant. 2002. *Buku Ajar Aljabar Linear*. Bandung: Sekolah Tinggi Teknologi Telkom.