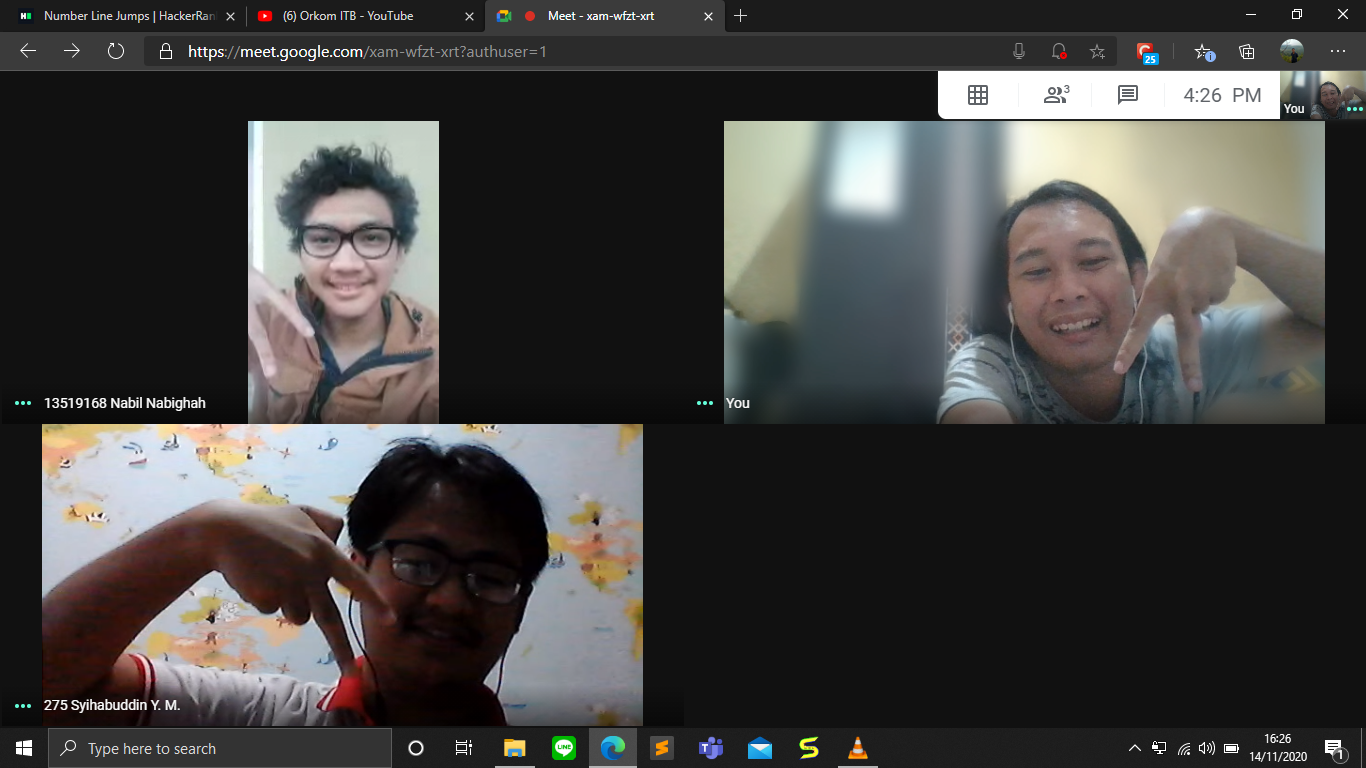
**LAPORAN TUGAS BESAR**

**IF2123**

**ALJABAR LINEAR DAN GEOMETRI**



**Disusun oleh:**

**Allief Nuriman - 13519221**

**Nabil Nabighah - 13519168**

**Syihabuddin Yahya Muhammad - 13519149**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

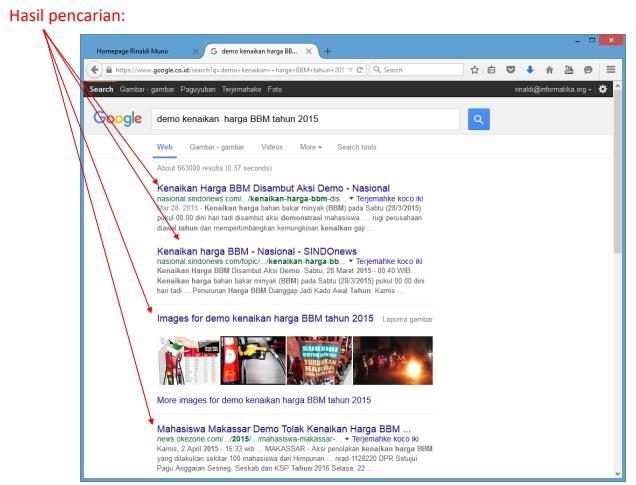
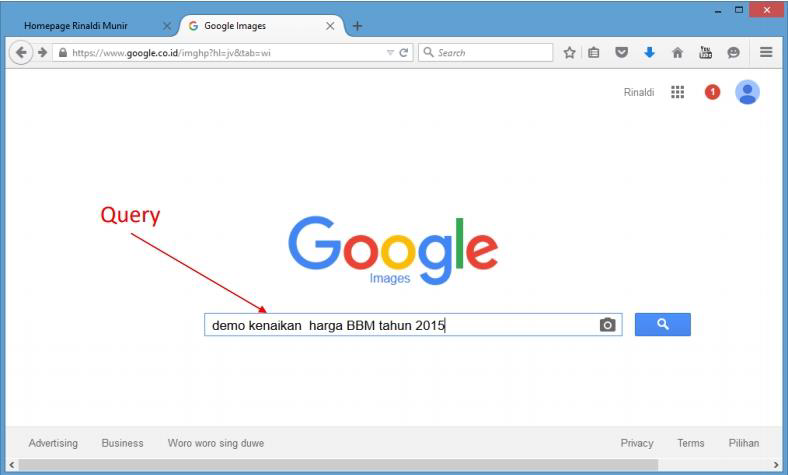
**INSTITUT TEKNOLOGI BANDUNG**

**2020/2021**

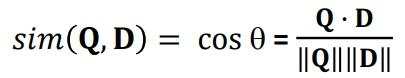
**BAB I**

**Deksripsi Masalah**

Hampir semua dari kita pernah menggunakan search engine, seperti google, bing dan yahoo! search. Setiap hari, bahkan untuk sesuatu yang sederhana kita menggunakan mesin pencarian Tapi, pernahkah kalian membayangkan bagaimana cara search engine tersebut mendapatkan semua dokumen kita berdasarkan apa yang ingin kita cari? Sebagaimana yang telah diajarkan di dalam kuliah pada materi vector di ruang Euclidean, temu-balik informasi (information retrieval) merupakan proses menemukan kembali (retrieval) informasi yang relevan terhadap kebutuhan pengguna dari suatu kumpulan informasi secara otomatis. Biasanya, sistem temu balik informasi ini digunakan untuk mencari informasi pada informasi yang tidak terstruktur, seperti laman web atau dokumen.



Ide utama dari sistem temu balik informasi adalah mengubah search query menjadi ruang vektor Setiap dokumen maupun query dinyatakan sebagai vektor w = (w1, w2 ,..., wn) di dalam Rn, dimana nilai wi 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:



**BAB II**

**Teori Singkat**

**Definisi Vektor**

Kuantitas fisik dapat direpresentasikan dalam dua jenis, yaitu skalar dan vektor. Skalar adalah kuantitas fisik yang hanya mengandung besar. Sedangkan vektor didefinisikan sebagai kuantitas fisik yang mengandung besar dan arah. Sebagai contoh kecepatan sebuah motor adalah 80 km/jam ke arah utara. Vektor dilambangkan dengan huruf kecil yang bercetak tebal atau huruf kecil yang memiliki tanda panah di atasnya, contohnya **u,v,w** dan ,. Ruang vektor atau ruang Euclidian didefinisikan di R2 , R3 , Rn. Vektor di ruang Rn adalah **v** = (v1, v2 ,..., vn).

Norma (*norm*) sebuah vektor adalah panjang (atau *magnitude*) sebuah vektor **u.** Norma vektor dilambangkan dengan ||**v**||. Norma sebuah vektor dinamakan juga norma Euclidean. Sebagai contoh norma vektor **v** = (v1, v2) di R2  adalah ||**v**|| = . Jika P1(x1,y1) danP2(x2,y2) adalah dua titik di R2 maka jarak (*d*) kedua titik tersebut adalah norma vektor yaitu .

**Operasi Vektor**

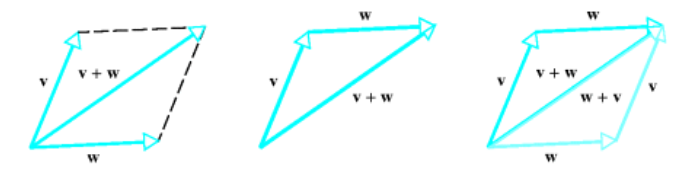
Vektor memiliki beberapa operasi, yaitu :

1. **Penjumlahan Vektor**

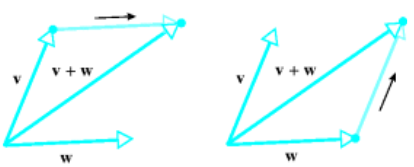
Jika vektor **u** = (u1, u2 ,..., un) dan vektor **v** = (v1, v2 ,..., vn), maka penjumlah antara vektor u dan vektor v adalah

**u + v** = (u1 + v1, u2 + v2 ,..., un+ vn)

Selain dengna menggunakan metode tersebut, penjumlahan vektor juga dapat menggunakan kaidah parallelogram, kaidah segitiga.



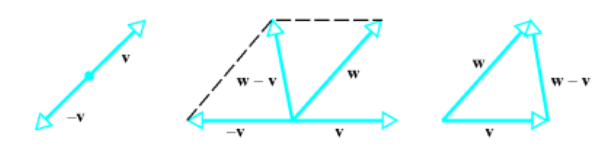
Gambar diatas adalah penjumlahan vektor dengan menggunakan kaidan parallelogram



Gambar diatas adalah penjumlahan vektor dengan menggunakan kaidah segitiga

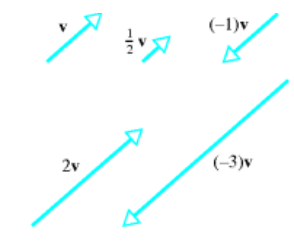
1. **Pengurangan Vektor**

Pengurangan vektor sama dengan penjumlahan vektor namun dengan menjumlahkan negatif dari vektor tersebut. Jika vektor **u** = (u1, u2 ,..., un) dan vektor **v** = (v1, v2 ,..., vn), maka pengurangan antara vektor **u** dan vektor **v** adalah

**u + (-v) = u - v** = (u1 - v1, u2 - v2 ,..., un - vn)

1. **Perkalian Vektor dengan Skalar**

Jika vektor **u** = (u1, u2 ,..., un), dan k adalah suatu skalar, maka k**u** adalah vektor yang panjangnya |k| kali panjang **u**



1. **Perkalian titik (*dot product*)**

Jika **u dan v** adalah vektor tidak nol di R2 atau R3, maka perkalian titik (*dot product*) dari **u** dan **v** adalah

****

adalah sudut yang dibentuk vektor **u** dan **v**

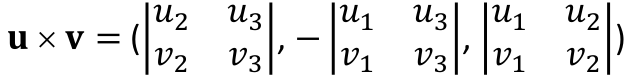
Selain cara diatas, dapat juga menggunakan kombinasi linear. Sebagai contoh

Jika vektor **u** = (u1, u2 ,..., un) dan vektor **v** = (v1, v2 ,..., vn), maka perkalian titik (*dot product*) dari **u** dan **v** adalah

****

1. **Perkalian silang (*cross product*)**

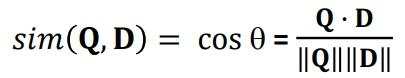
Jika vektor **u** = (u1, u2 ,..., un) dan vektor **v** = (v1, v2 ,..., vn) yang berada di R3, maka perkalian silang (*cross product*) antara vektor **u** dan vektor **v** adalah

****

**Aplikasi Vektor**

Salah satu aplikasi dari operasi perkalian titik (*dot product*) ada pada bidang sistem Temu-balik Informasi (*Information retrieval*). *Information retrieval* (selanjutnya disingkat IR) tidaklah sama dengan pencarian dalam database. IR biasanya digunakan untuk pencarian informasi dengan isi informasi yang tidak terstruktur. Contoh dari informasi tak-terstruktur adalah dokumen dan laman web. Contoh dari aplikasi IR adalah *search engine* atau mesin pencari.

Salah satu model dari IR adalah model ruang vektor. Misalkan terdapat *n* kata berbeda sebagai kamus data atau (*vocabulary*). Kata tersebut dapat membentuk sebuah ruang vektor berdimensi *n*. Setiap dokumen ataupun *query* dinyatakan sebagai vektor **w** = (w1, w2 ,..., wn)di dalam Rn, wi adalah jumlah dari setiap kata *i* dalam *query* ataupun dokumen. Penentuan dokumen yang relevan dengan *query* dipandang sebagai pengukuran kesamaan atau *similarity measure* antara *query* dan dokumen. Semakin sama vektor antara *query* dan dokumen maka akan semakin relevan dokumen tersebut dengan *query*nya. Kesamaan antara *query* dengan dokumen bisa dihitung dengan perkalian dot atau *dot product* dari vektor *query* dan vektor dokumen sebagaimana persamaan dibawah.



**BAB III**

**Implementasi**

**3.1 Garis Besar**

Dalam pembuatan search engine ini, penulis menggunakan framework flask beserta library nltk untuk stemming kata dan library dataclass untuk membuat ADT.

Atribut file yang diupload pada server dan query akan disimpan pada ADT dokumen yang berisikan judul dokumen(string), isi dokumen(string), frekuensi kata yang telah di stemming(dictionary), tingkat similarity(float), jumlah kata(integer), jumlah karakter(integer), dan 30 karakter pertama(string) dari dokumen tersebut.

Ketika dokumen dan query telah masuk, maka akan dibandingkan similarity antara frekuensi kata yang telah di stemming pada ADT file upload dan query, dimana hasilnya disimpan pada similarity. Perhitungan similarity dilakukan dengan menggunakan metode vektor. Kemudian array yang menyimpan atribut file akan diurutkan berdasarkan atribut similaritynya dan ditampilkan pada laman hasil pencarian.

Jika pengguna menekan judul dokumen yang ditampilkan, pengguna aka dilempar pada halaman yang berisi isi dari sokumen tersebut. Selain hasil pencarian ditampilkan juga tabel yang berisikan kata yang telah di stemming dari query beserta frekuensinya pada query dan file.

**3.2 Fungsi**

Dalam pembuatan search engine ini penulis berbagai fungsi yaitu,

1. sim

Fungsi sim adalah fungsi yang menerima argumen sebuah vektor *query* dan vektor *dokumen* kemudian akan menghitung similarity dari *query* dan dokumen tersebut. Kemudian mengembalikan hasilnya.

1. dotprod

Fungsi dotprod adalah fungsi yang menerima argumen sebuah vektor *query* dan vektor dokumen kemudian akan menghitung perkalian titik (*dot product*) dari *query* dan dokumen tersebut. Lalu mengembalikan hasilnya

1. magvec

Fungsi magvec adalah fungsi yang menerima suatu vektor *query* kemudian menghitung besar dari vektor *query* tersebut dan mengembalikan besarnya.

1. sort

Prosedur sort adalah prosedur untuk menyortir sebuah *query* dan dokumen sehingga terurut membesar

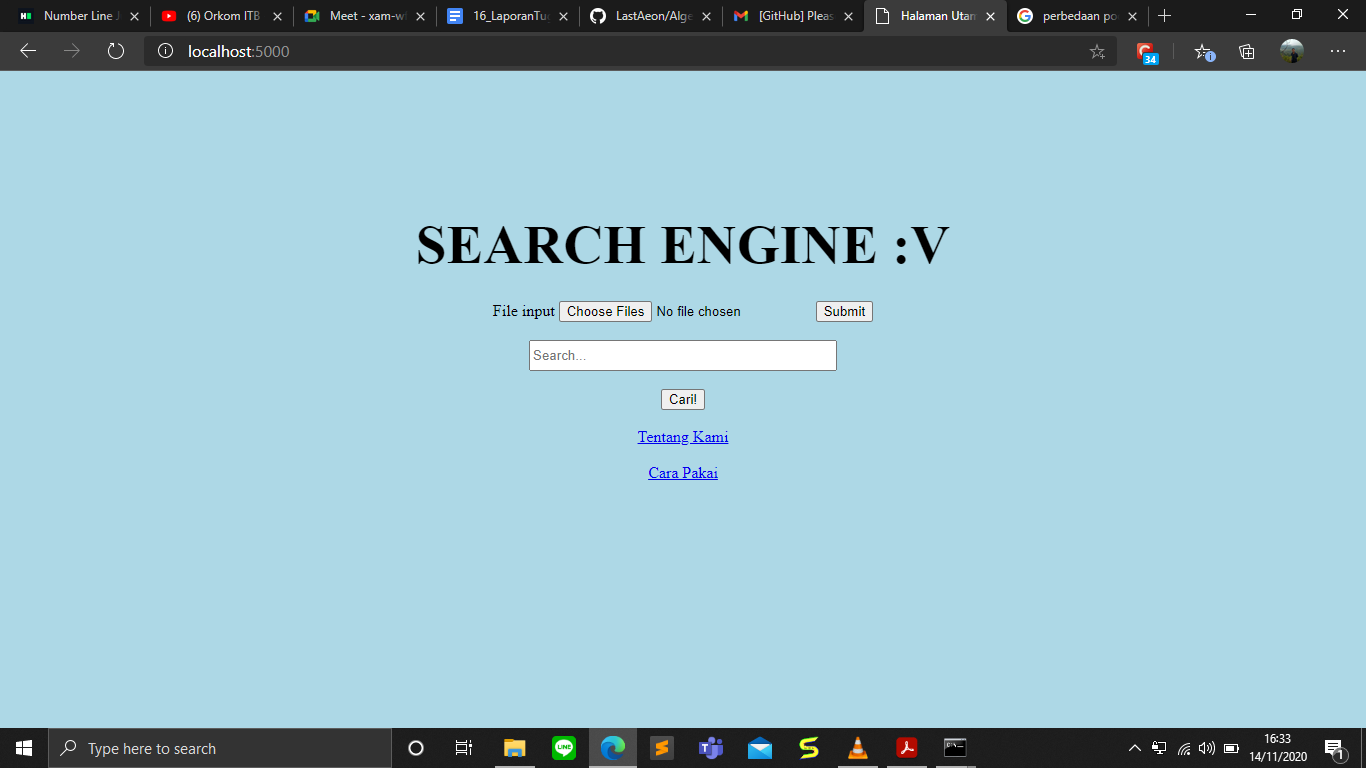
1. sortD

Prosderu sorD adalah prosedur untuk menyortir ADT dokumen berdasarkan atribut similarity menjadi terurut mengecil

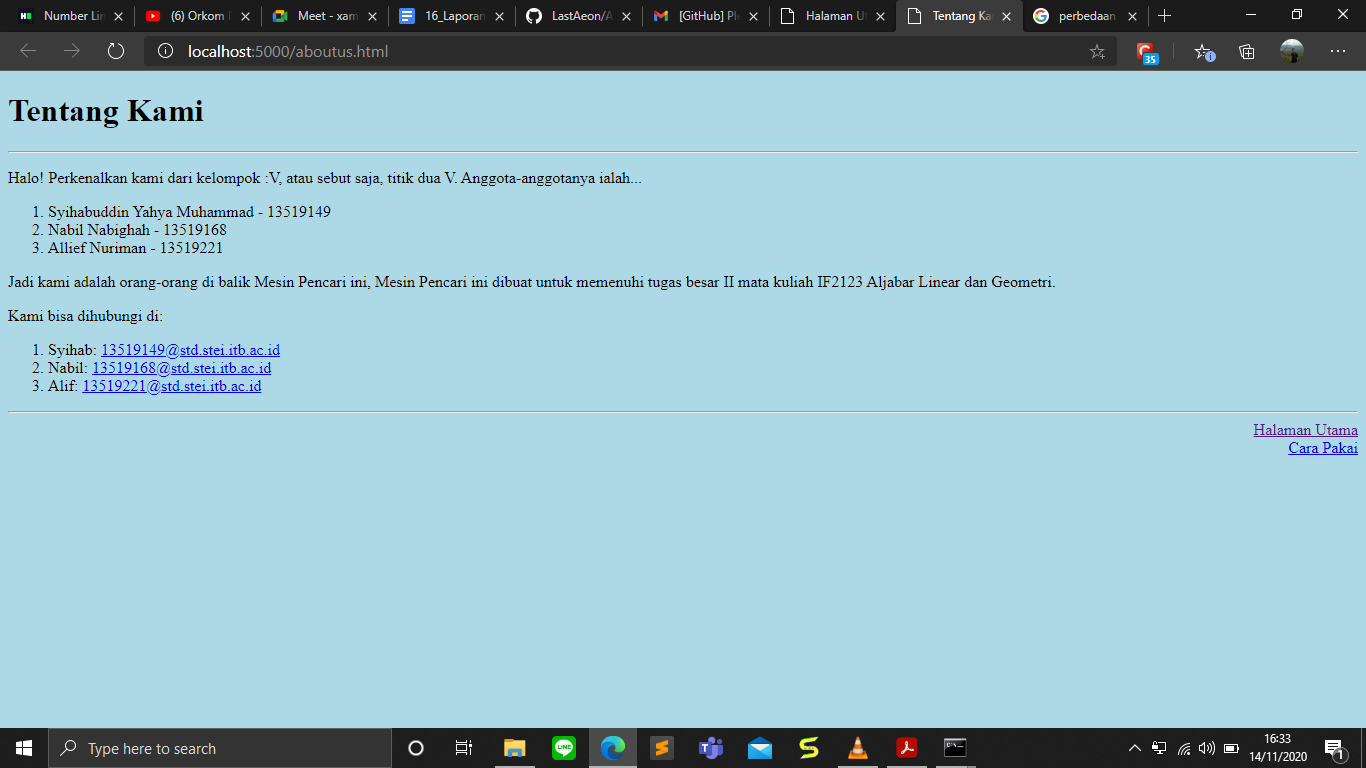
**BAB IV**

**Eksperimen**

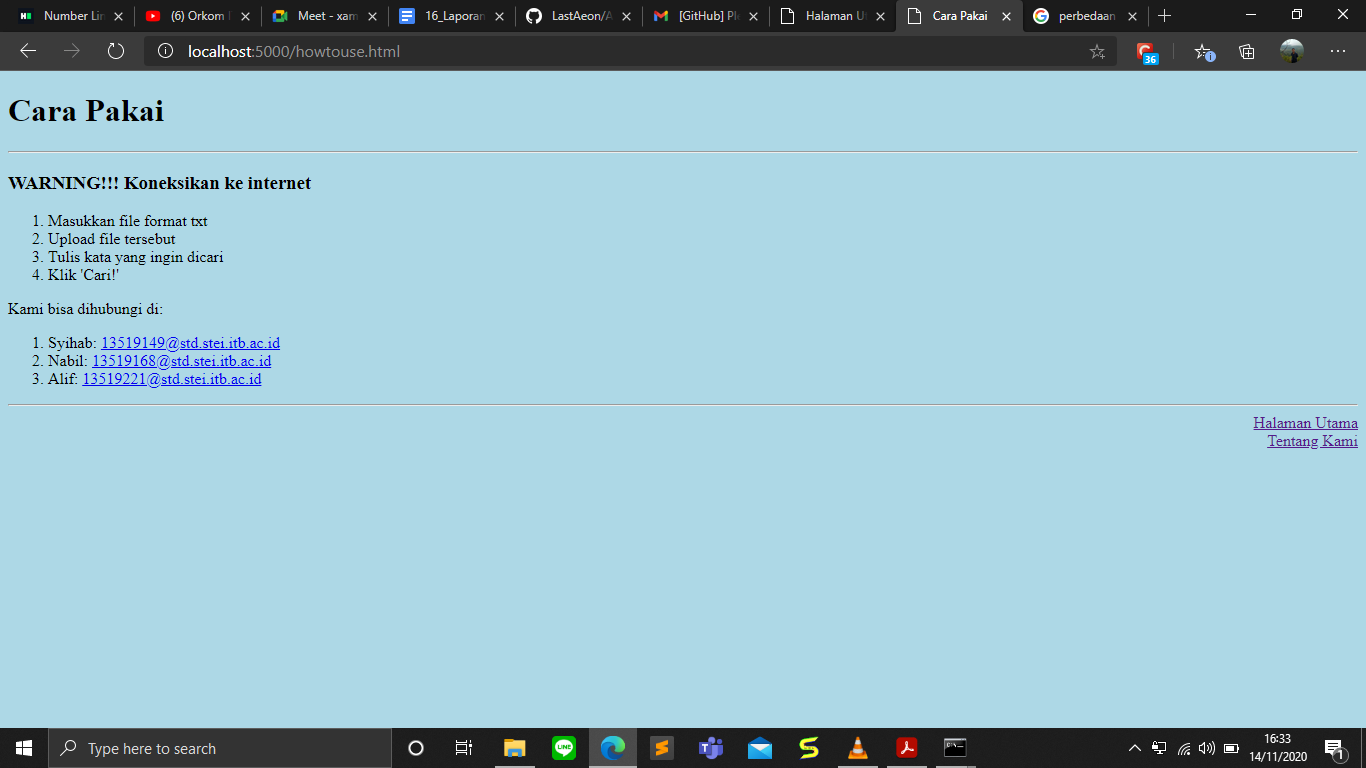
**4.1 Tampilan Website**



Halaman utama

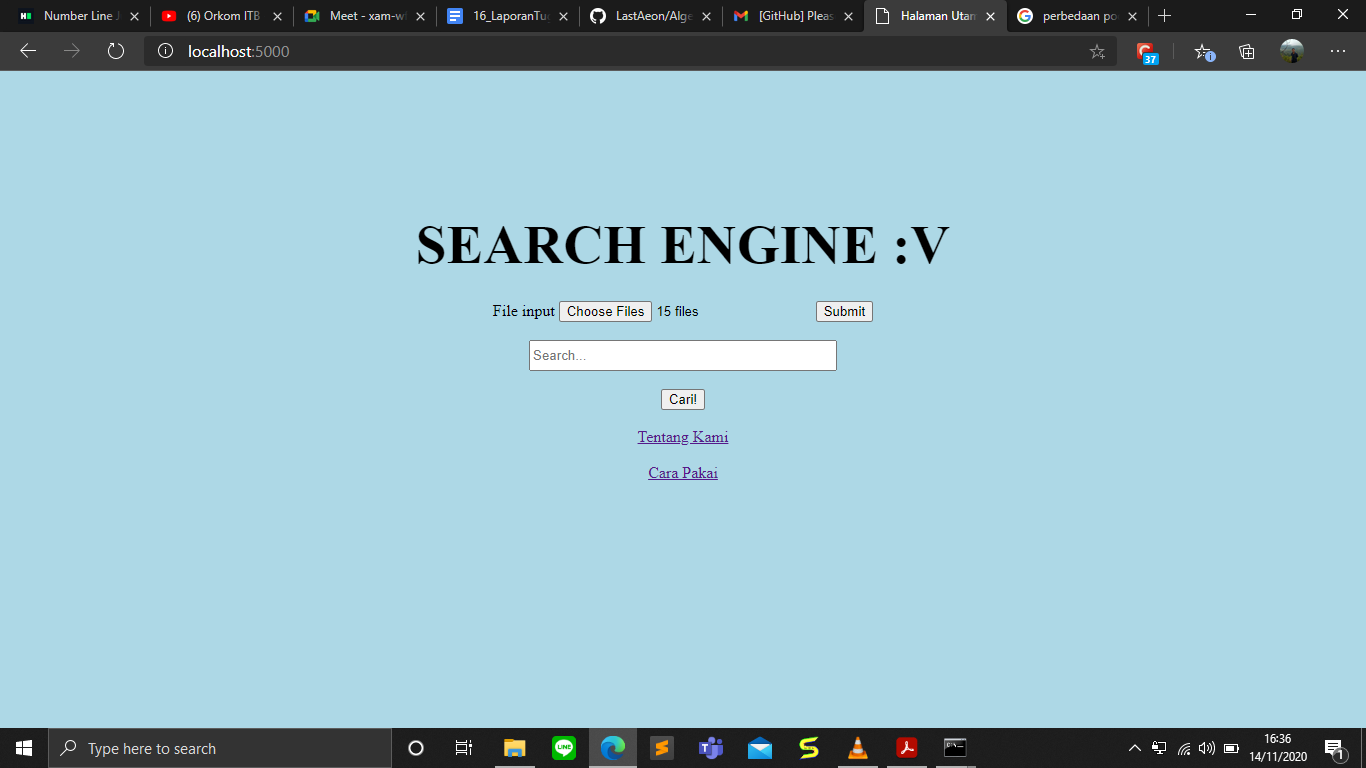


Laman tentang kami

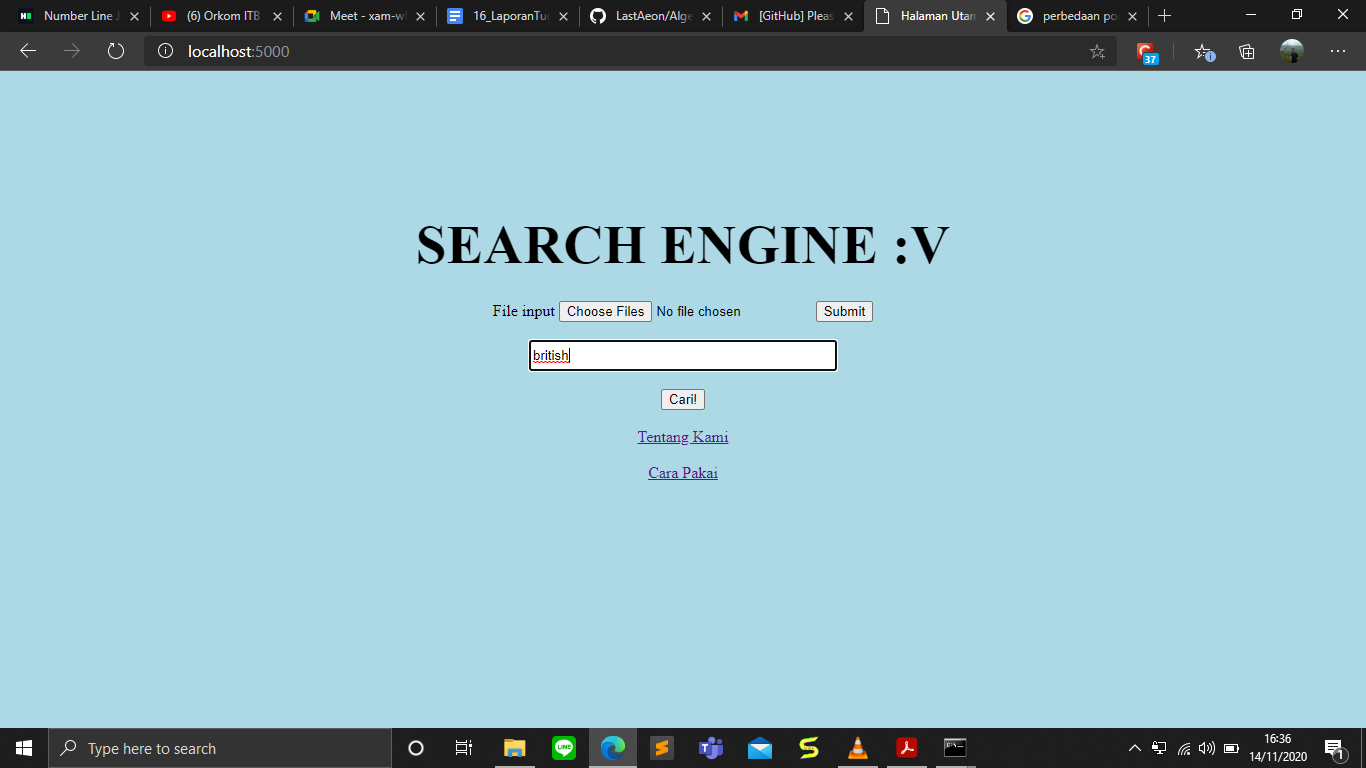


Laman cara pakai

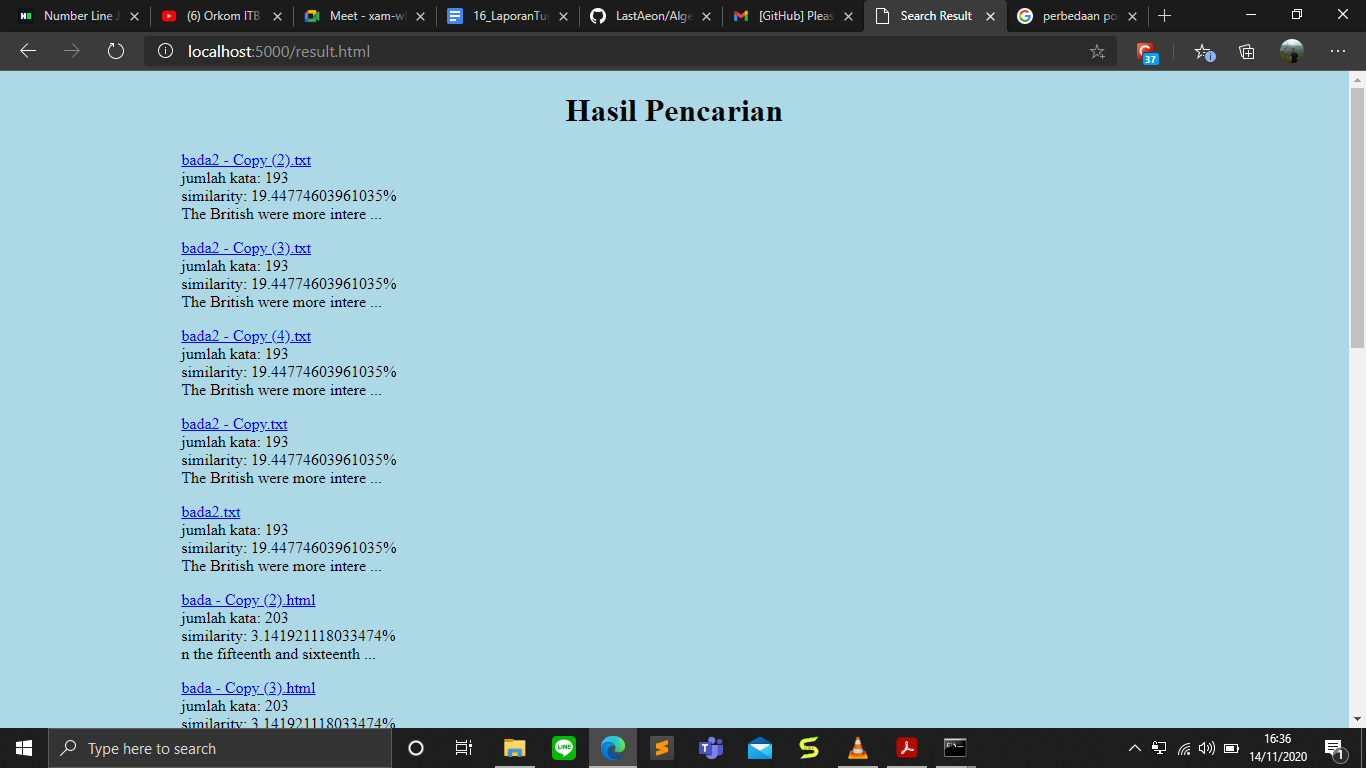
**4.2 Penggunaan Website dengan Query 1 Kata**



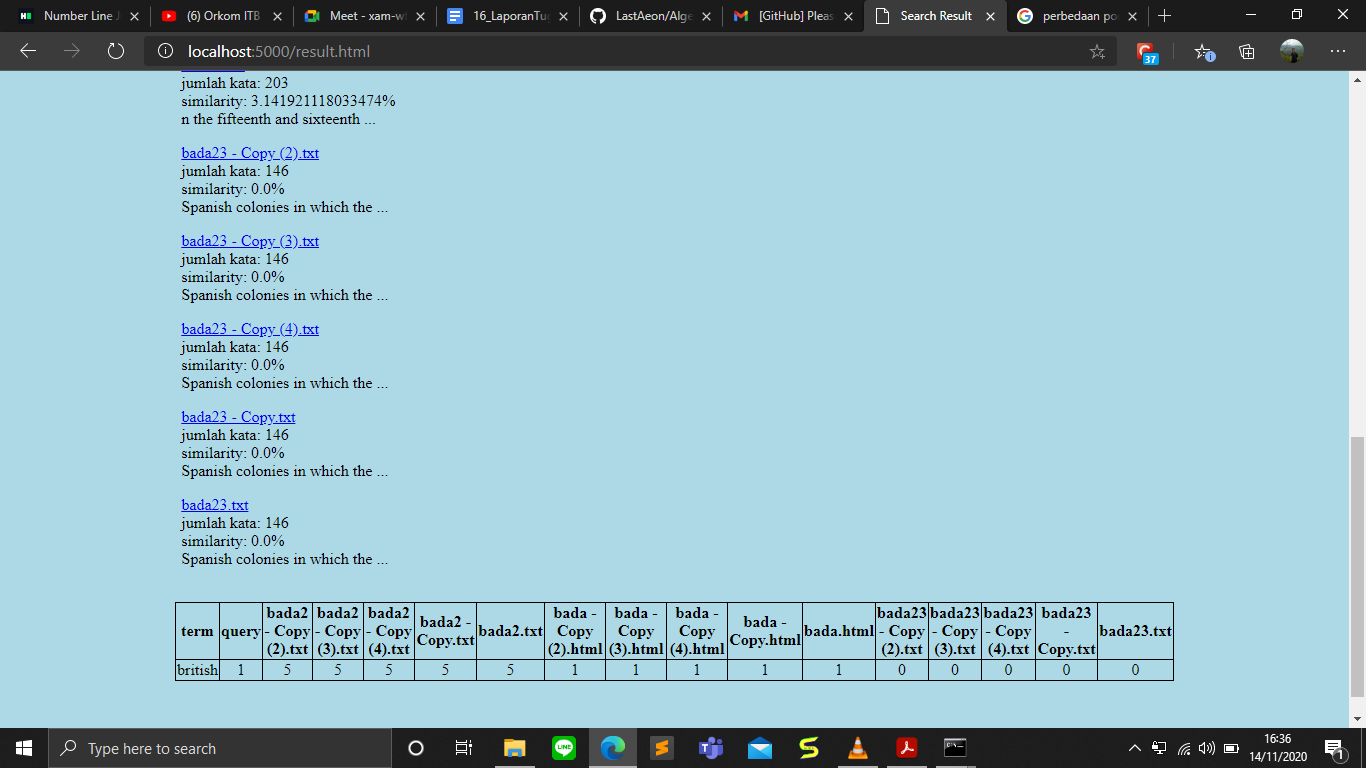
Melakukan penggunggahan file ke dalam program untuk disimpan ke dalam ADT dokumen



Melakukan pencarian kata british

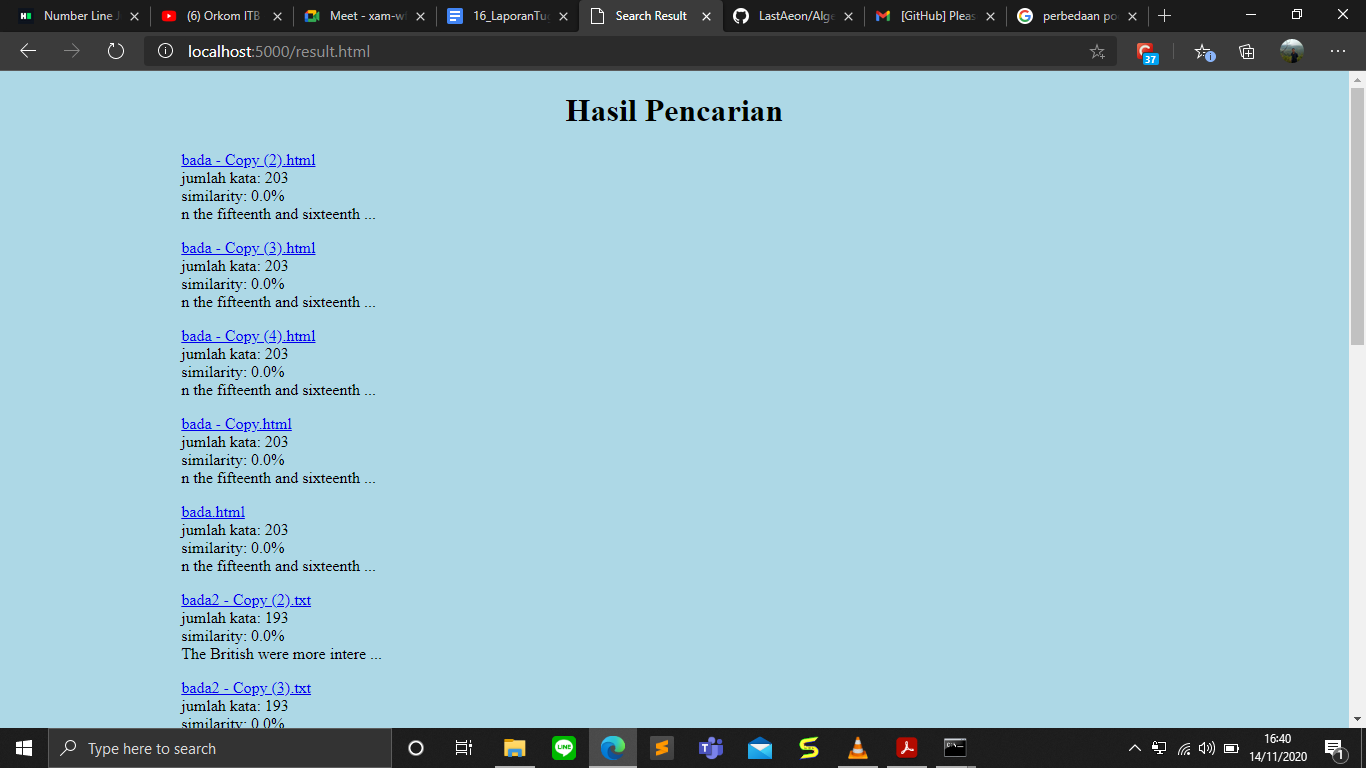


Hasil pencarian. Nampak bahwa hasil pencarian terurut berdasarkan similarity



Pada bagian paling bawah, terdapat tabel term, query, dan kemunculannya pada setiap dokumen.

**4.3 Penggunaan Website dengan Input Sembarang**



Hasil Pencarian dengan input “eofeofefef”



Nampak bahwa input “eofeofefef” tidak muncul dimanapun.

**BAB V**

**Kesimpulan, Saran, Refleksi**

**5. 1 Kesimpulan**

Search engine berhasil dibuat sesuai spesifikasi dan sesuai dengan penggunaan *cosine similarity*. Selain itu, web berhasil dibuat dengan memanfaatkan *framework* *flask*.

**5. 2 Saran**

Untuk kedepannya mungkin dapat dikembangkan cara menyimpan query yang lain agar penyimpanan dan pemrosesan dokumen dapat dilakukan dengan lebih efisien.

**5.3 Refleksi**

Ternyata kami perlu untuk mempelajari penggunaan *flask* dan membaca referensi dulu sebelum memulai pengerjaan agar meningkatkan efisiensi pengerjaan. Selain itu, kami juga mulai menyadari bahwa tugas besar di IF ITB membutuhkan eksplorasi lebih agar hasilnya memuaskan. Kami juga belajar untuk menggunakan HTML sebelum bertemu mata kuliah IF3110 Pengembangan Aplikasi Berbasis Web.

**DAFTAR PUSTAKA**

* "Create A Simple Search Engine Using Python" by Irfan Alghani Khalid <https://link.medium.com/yEtxO932Kab>
* "Implementing the TF-IDF Search Engine" by Kartheek Akella <https://link.medium.com/UcdqCt92Kab>
* “Create APIs with python and flask” by programming historian <https://programminghistorian.org/en/lessons/creating-apis-with-python-and-flask>
* “How to create a react flask project” by Miguel Grinberg <https://blog.miguelgrinberg.com/post/how-to-create-a-react--flask-project>
* “Removing Stop Words with NLTK Python” <https://www.geeksforgeeks.org/removing-stop-words-nltk-python/#:~:text=What%20are%20Stop%20words%3F,result%20of%20a%20search%20query>
* “Sastrawi Library” <https://pypi.org/project/Sastrawi/>
* “Print values of array of arrays in flask” by alessandro buffoli <https://stackoverflow.com/questions/51932277/print-values-of-array-of-arrays-in-flask/51932604>
* Howard Anton & Chris Rores, *Elementary LinearAlgebra, 10th Edition*.