

# LAPORAN TUGAS BESAR 2



Nicholas Chen 13519029  
Jordan Daniel Joshua 13519098  
Naufal Alexander Suryasumingrat 13519135

IF 2123  
ALJABAR LINIER DAN GEOMETRI  
INSTITUT TEKNOLOGI BANDUNG  
2020/2021

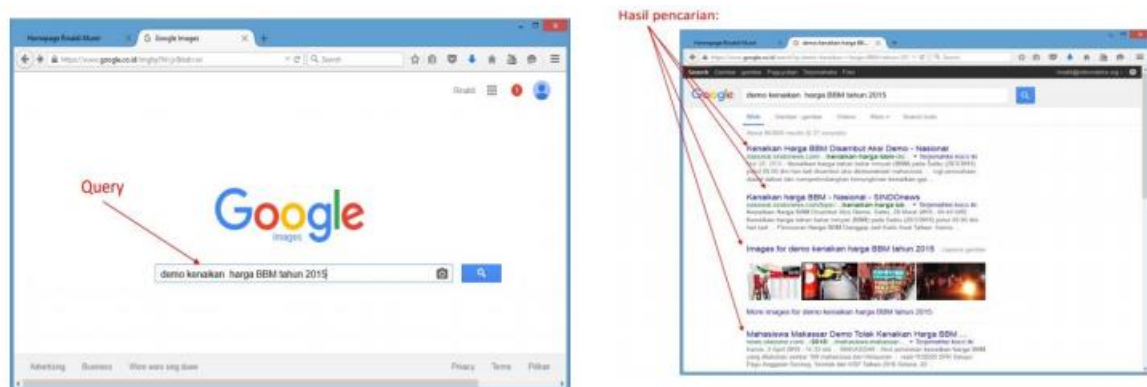
# BAB I

## 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 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.



*Gambar 1.1 Contoh penerapan Sistem Temu-Balik pada mesin pencarian*  
sumber: Aplikasi Dot Product pada Sistem Temu-balik Informasi by Rinaldi Munir

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](#)

Gambar 1.2 Tampilan layout dari aplikasi web search engine yang dibangun.

**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* <Judul Dokumen1>, maka akan diarahkan pada sebuah halaman yang berisi *full-text* terkait dokumen 1 tersebut (seperti *Search Engine*).

Data uji berupa dokumen-dokumen yang akan diunggah ke dalam web browser. Format dan extension dokumen dari file yang diunggah dalam bentuk html. Terdapat 15 dokumen yang telah disiapkan untuk uji coba.

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 beberapa hal-hal yang perlu dilakukan dalam eksekusi program ini.

1. Dilakukan stemming dan penghapusan stopwords pada setiap dokumen.
2. Tidak dibedakan antara huruf-huruf besar dan huruf-huruf kecil.
3. Stemming dan penghapusan stopwords dilakukan saat penyusunan vektor, sehingga halaman yang berisi full-text terkait dokumen tetap seperti semula.
4. Dihapus karakter-karakter yang tidak perlu ditampilkan (saat menggunakan web scraping atau format dokumen berupa html).
5. Bahasa yang digunakan dalam dokumen adalah Bahasa Indonesia.
6. Digunakan library nltk untuk stemming kata dan penghapusan stopwords.

## BAB II

### TEORI SINGKAT

#### 2.1 Retrieval Information

Temu-balik informasi (Retrieval Information) menemukan cara mendapatkan kembali (retrieval) informasi yang relevan terhadap kebutuhan pengguna dari suatu kumpulan informasi secara otomatis.



Sumber gambar: <https://sites.google.com/site/berbagiinformasidanekspresi/arsip/pengantar-temu-kembali-informasi-information-retrieval>

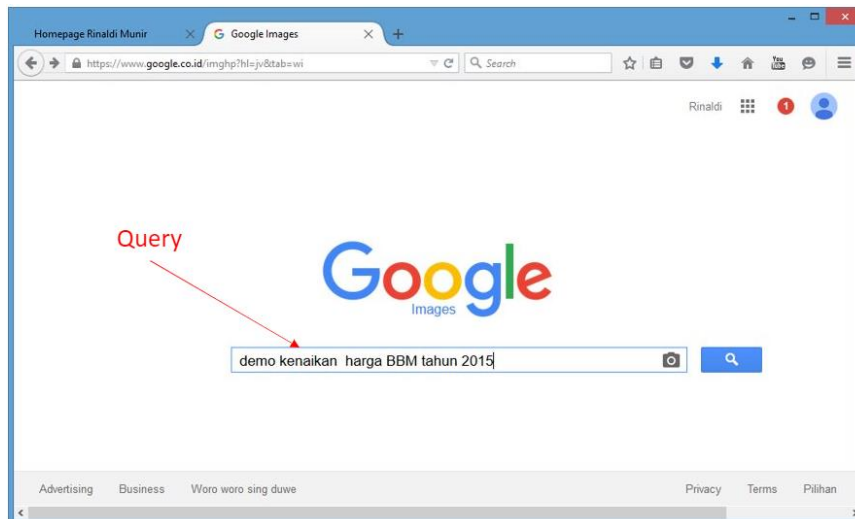
IR tidak sama dengan pencarian di dalam basisdata (database) yang sudah dalam keadaan terstruktur. IR umumnya digunakan pada pencarian informasi yang isinya acak seperti laman pada web, dokumen-dokumen digital, dan lainnya.

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

Gambar 2.1 Contoh data yang tersimpan pada database terstruktur

sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo12Aplikasi-dot-product-pada-IR.pdf>

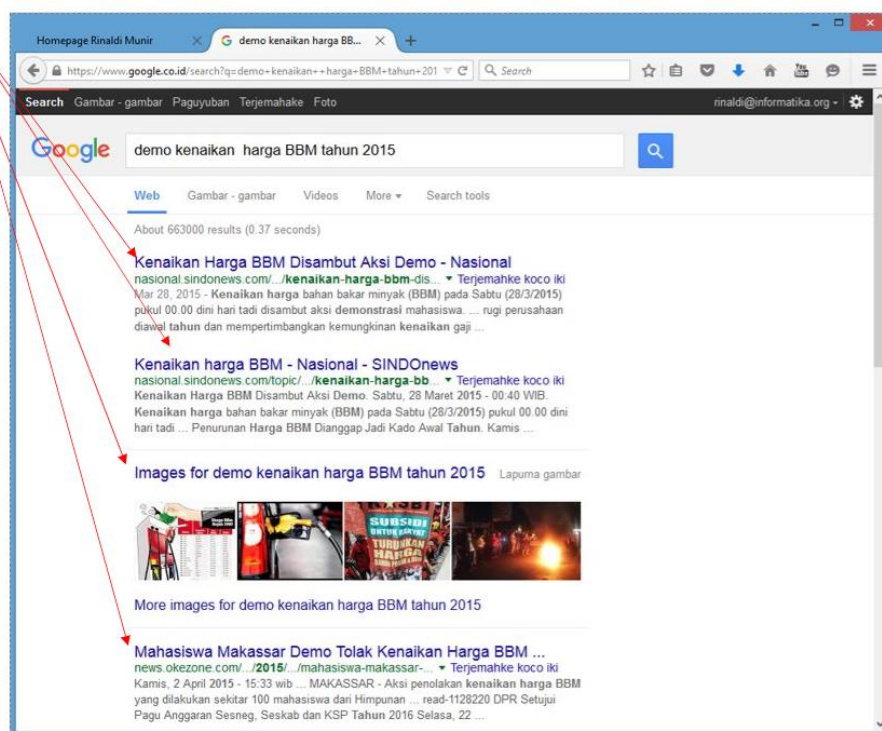
Salah satu contoh penggunaan sistem temu-balik informasi adalah *Search Engine* seperti *Google*, *Mozilla Firefox*, dan *Opera*. *Search Engine* akan menerima input query sebagai kata kunci pencarian, lalu menggunakan teknik pencarian seperti Vektor dan Cosine Similarity untuk mengurutkan ratusan juta dokumen yang ada dan menampilkannya dari dokumen yang paling relevan dengan query.



*Gambar 2.2 Contoh proses input query pada Search Engine*

sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo12Aplikasi-dot-product-pada-IR.pdf>

Hasil pencarian:

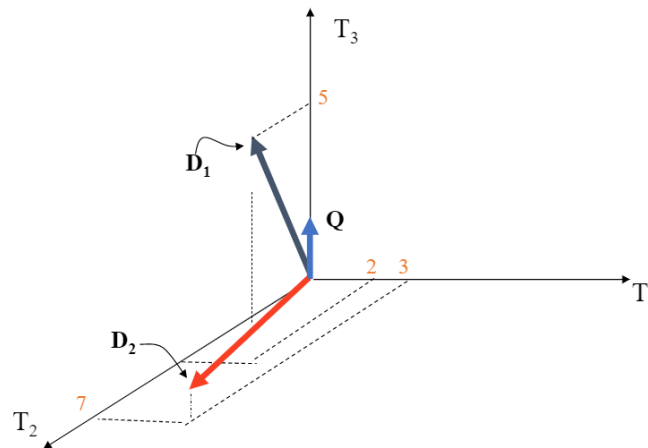


*Gambar 2.3 Contoh hasil pemrosesan Search Engine*

sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo12Aplikasi-dot-product-pada-IR.pdf>

IR dapat dimodel dalam bentuk ruang vektor. Model ini berkaitan erat dengan aljabar vektor. Misalkan terdapat  $n$  kata berbeda sebagai kamus kata (vocabulary) atau indeks kata (term index). Kata-kata tersebut membentuk ruang vektor berdimensi  $n$  dan setiap dokumen maupun query dinyatakan sebagai vektor  $w = (w_1, w_2, \dots, w_n)$  di dalam  $R_n$  dengan  $w_i$  adalah bobot setiap kata  $i$  di dalam query atau dokumen. Nilai  $w_i$  dapat menyatakan jumlah kemunculan kata tersebut dalam dokumen (term frequency).

Sebagai contoh misalkan pada query dan dokumen sampel terdapat 3 buah kata unik (T1, T2, dan T3). Untuk menyusun vektor query dan dokumen, cukup dihitung jumlah kemunculan tiap katanya. Jika query berisi 2 kata T3, dokumen 1 (D1) berisi 2 kata T1, 3 kata T2, dan 5 kata T3, sedangkan dokumen 2 berisi 3 kata T1, 7 kata T2, dan 1 kata T3, maka masing-masing dapat disusun menjadi  $D_1 = (2,3,5)$ ,  $D_2 = (3,7,1)$ ,  $Q = (0,0,2)$ .



Gambar 2.4 Representasi vektor dalam grafik 3 dimensi

sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo12Aplikasi-dot-product-pada-IR.pdf>

## 2.2 Vektor dan Cosine Similarity

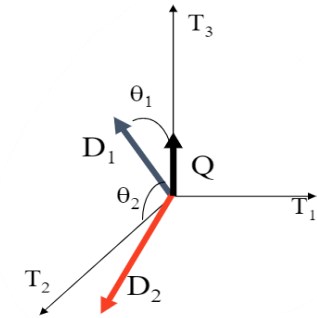
Setelah IR dimodelkan dalam bentuk ruang vektor, untuk menentukan kemiripan query dengan masing-masing dokumen dapat digunakan metode *Cosine Similarity*. Masing-masing dokumen akan dihitung dengan rumus berikut:

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

Gambar 2.5 Rumusan dari Cosine Similarity

sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo12Aplikasi-dot-product-pada-IR.pdf>

*Cosine Similarity* menentukan kemiripan kedua vektor berdasarkan besar sudut yang terbentuk antara kedua vektor. Jika sudut yang membentuk kedua vektor semakin kecil, maka tingkat kemiripan kedua vektor tersebut akan makin tinggi, berlaku pula sebaliknya.



*Gambar 2.6 Contoh Representasi ruang vektor dalam bentuk grafik*

sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo12Aplikasi-dot-product-pada-IR.pdf>

Setelah dihitung masing-masing *similarity* dari dokumen yang termasuk dalam pencarian, dilakukan pe-ranking-an dari 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.



## BAB III

### IMPLEMENTASI

#### 3.1 Web Scrapping

Untuk melakukan web scrapping, diimport modul beautiful soup dari internet. Pertama, diambil berita-berita dari link yang diberikan, disini diambil contoh dari link <https://bola.kompas.com/liga-inggris>. Dari link ini terdapat 20 berita dan semua link berita dimasukkan kedalam sebuah array bernama list\_url.

```
list_url = get_link("https://bola.kompas.com/liga-inggris")
```

*Gambar 3.1 kode untuk mengambil berita*

Kebanyakan berita yang terdapat diinternet memiliki jumlah halaman lebih dari satu. Untuk mengambil semua halaman berita, digunakan fungsi berikut.

```
add_link(list_url, "?page=all#page2")
```

*Gambar 3.2 kode untuk mengambil keseluruhan halaman berita*

Setelah itu, dilakukan pengekstrakan berita dari judul berita dan isi berita. Pada berita pada laman Kompas, isi artikel diletakkan pada class bernama "read\_\_content". Digunakan fungsi BeautifulSoup untuk mengambil isi dari class tersebut dan dilakukan pembersihan syntax-syntax html pada dokumen. Setelah itu isi dari berita tersebut dimasukkan kedalam sebuah variabel. Judul artikel dari laman Kompas diletakkan pada laman "read\_\_title" dan dilakukan hal serupa untuk pengekstrakannya. Setelah itu variabel-variabel hasil tersebut dimasukkan kedalam sebuah dictionary agar lebih mudah diakses.

Setelah dictionary terbentuk, dibuat file html baru untuk masing-masing berita dengan fungsi toHTML. Fungsi ini meminta input dictionary hasil scrape untuk selanjutnya men-generate masing-masing html.

```
def toHTML(scrape_result):  
    for scrape in range(len(scrape_result)):  
        savedHTML = open("test/" + str(scrape + 1) + ".html", "w")  
        toSave = "<html><head><title>"  
        toSave += scrape_result[scrape].get("title")  
        toSave += "</title></head><body><h1>"  
        toSave += scrape_result[scrape].get("title")  
        toSave += "</h1><p>"  
        toSave += scrape_result[scrape].get("content")  
        toSave += "</p></body></html>"  
        savedHTML.write(toSave)  
        savedHTML.close()
```

*Gambar 3.3 kode untuk mengenerate html*

### 3.2 Data Processing dari file lokal atau file yang sudah diupload pengguna

Pada tahap ini prosedurnya mirip seperti web scarping. Data diambil dari file lokal atau hasil upload pengguna. Jenis file yang diproses pada tahap ini harus berbentuk html. Pertama, dilakukan pengekstrakan judul artikel dan isi artikel menggunakan fungsi dari BeautifulSoup dan hasilnya disusun kedalam sebuah dictionary.

```
# Dictionary yang akan digunakan
scrape_result = [{
    "link"      : val[0],
    "title"     : val[1],
    "content"   : val[2],
    "words"     : val[3],
    "fsentence" : val[4],
    "similarity": val[5],
    "count"     : val[6]}]
```

*Gambar 3.4 dictionary tempat menyimpan hasil scraping*

Pada dictionary juga dimasukkan hyperlink ("link") untuk masing-masing file yang sedang diproses, tuple (words) yang terdiri dari setiap kata pada setiap file beserta jumlahnya, kalimat pertama ("fsentence") pada isi artikel, hasil similarity, dan jumlah kata keseluruhan pada isi artikel ("count").

### 3.3 Fungsi Similarity

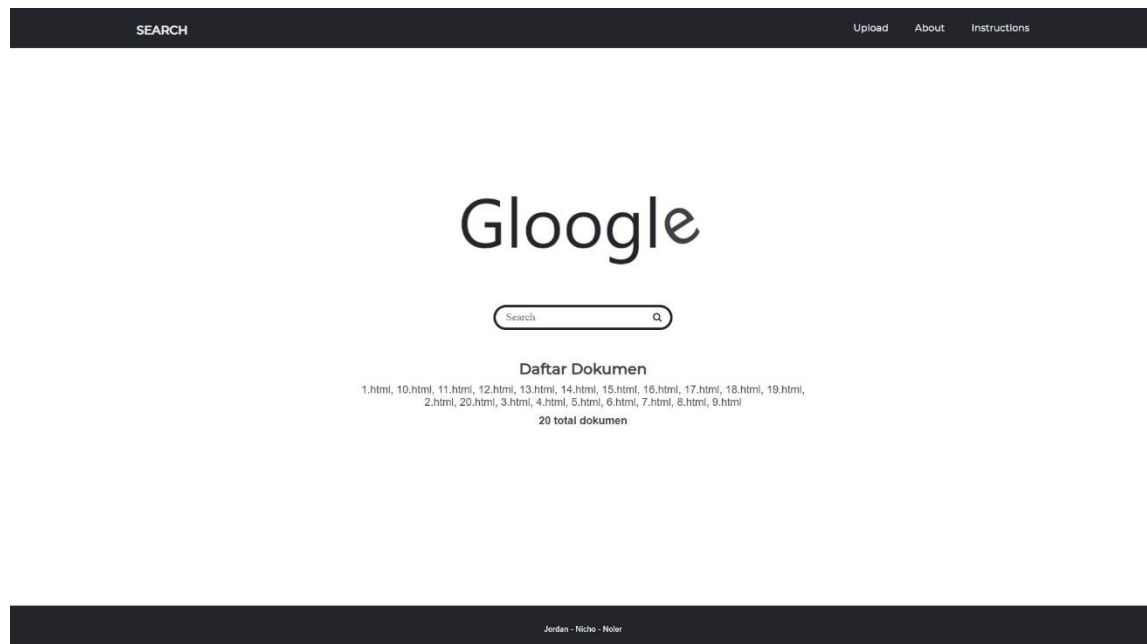
Fungsi ini digunakan untuk menghitung similarity untuk mengisi data dictionary pada tahap Data Processing. Fungsi ini menerima input array query dan array tuple yang terdapat pada "words" dictionary. Untuk setiap kata pada query dilakukan pencarian kata pada array tuple, lalu jumlah katanya disimpan kedalam sebuah variabel. Setelah itu dihitung pula panjang dari vektor query dan panjang dari vektor dokumen. Kemudian besar similarity dihitung dengan rumus dot product vektor dan fungsi mengembalikan hasil cosinusnya. Semakin besar angka similarity, maka semakin mirip dokumen dengan input query.

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

*Gambar 3.5 rumus similarity dengan perhitungan cosine dalam vektor ruang*

### 3.4 Front-end User Interface

Untuk bagian front-end, digunakan Flask, css, dan Jinja. Dengan Flask, dibuat 5 route page yang setiap bagian atas lamannya terdapat navigation bar untuk berpindah ke page lainnya.



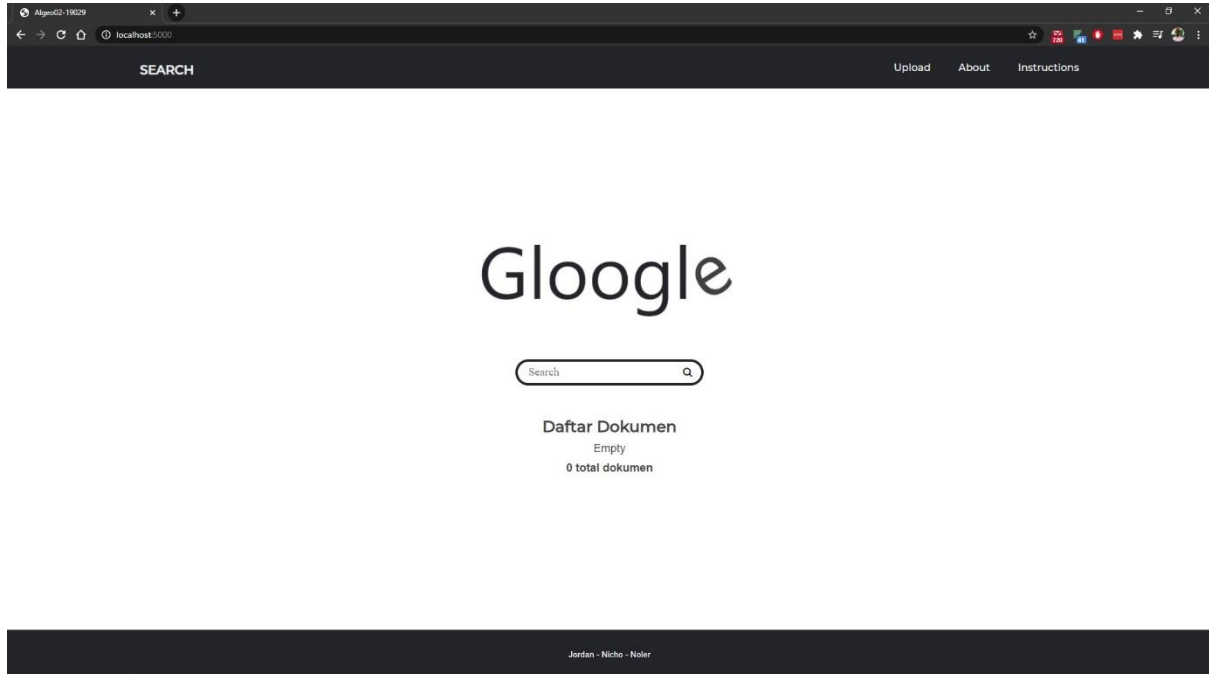
*Gambar 3.6 tampilan laman search awal*

Pada page pertama, terdapat sebuah search box untuk menerima input query dari user dan diberi route ("/"). Sebelum memasukkan query, user dapat meng-upload file-file yang ingin diproses pada laman upload. Laman ini memiliki route ("/upload"). Setelah user menekan tombol pencarian, user akan berpindah ke laman list hasil pencarian yang sudah terurut berdasarkan similaritas beserta hyperlink ke masing masing dokumennya. Pada laman ini user bisa langsung melakukan search ulang dengan query yang berbeda dan user juga bisa melihat data statistik mengenai pencarian dokumen. Berikutnya ada laman instruction yang berisi instruksi singkat tentang cara penggunaan search engine dan ada laman about us yang berisi penjelasan trivia mengenai program ini dan pembuatnya.

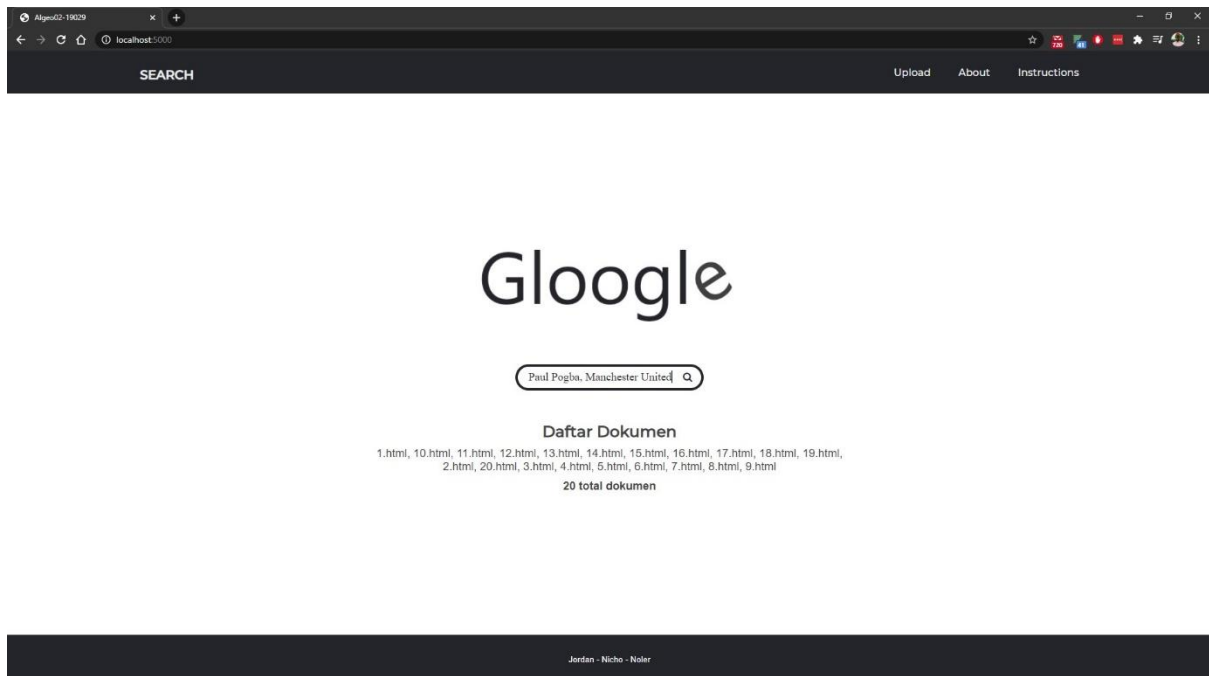
## BAB IV

### EKSPERIMEN

#### 4.1 Laman Search Awal



*Gambar 4.1 tampilan laman search awal sebelum ada file yang diupload*



*Gambar 4.2 tampilan laman search awal sesudah ada file yang diupload*

## 4.2 Laman Hasil Search

SEARCH

UploadAboutInstructions

Daftar Dokumen

1.html, 10.html, 11.html, 12.html, 13.html, 14.html, 15.html, 16.html, 17.html, 18.html, 19.html, 2.html, 20.html, 3.html, 4.html, 5.html, 6.html, 7.html, 8.html, 9.html

20 total dokumen

Search

Showing results for "Paul Pogba, Manchester United"

Paul Pogba Hanya Dapat Rating 2,5 dari 10 Saat Bela Timnas Perancis

Jumlah kata : 203  
Similarity : 54.35%  
Kalimat pertama : KOMPAS.com - Peruntungan Paul Pogba tak kunjung membaik musim ini.

Pogba di Bawah Performa, Deschamps Sebut Itu Dampak Situasi Buruk Klub

Jumlah kata : 230  
Similarity : 45.32%  
Kalimat pertama : KOMPAS.com - Pelatih tim nasional Perancis, Didier Deschamps, angkat bicara soal Paul Pogba yang tampil di bawah performa saat berusia Finlandia. Selasa (10/11/2020).

Jangan Khawatir, Bos MU di Belakang Solskjaer!

Jumlah kata : 208  
Similarity : 28.85%  
Kalimat pertama : MANCHESTER, KOMPAS.com - Meski memulai musim 2020-2021 dengan capaian terburuk, Manchester United bakal tetap didampingi pelatih Ole Gunnar Solskjaer.

Man United Tampil Kurang Menjanjikan, Setan Merah Kekeh Pertahankan Solskjaer

Jumlah kata : 208  
Similarity : 27.24%  
Kalimat pertama : KOMPAS.com - Manchester United tegaskan tetap pertahankan Ole Gunnar Solskjaer setelah pelatih asal Norwegia itu dikabarkan terancam dipecat.

Masih Keteteran di Urutan Medioker, MU Pun Terlanda Urusan Kocek

Jumlah kata : 211  
Similarity : 26.59%  
Kalimat pertama : MANCHESTER, KOMPAS.com - Hingga jelang pekan kesembilan Liga Primer, Manchester United masih keteteran di urutan medioke alias papan tengah.

Van de Beek Cadangan, Rafael van der Vaart Sindir Pemain Man United

Jumlah kata : 209  
Similarity : 19.76%  
Kalimat pertama : KOMPAS.com - Mantan gelandang timnas Belanda, Rafael van der Vaart, tidak terima dengan keputusan Ole Gunnar Solskjaer yang mencadangkan junior dan komplotniya, Donny van de Beek.

Pelan-pelan Castore Gantikan Adidas di Wolves

Jumlah kata : 148  
Similarity : 16.62%  
Kalimat pertama : MIDLAND, KOMPAS.com - Perusahaan perlengkapan olahraga asal Inggris, Castore, pelan-pelan menggantikan Adidas dalam kerja sama dengan klub Liga Primer, Wolverhampton Wanderers.

Van de Beek Cetak Gol Beruntun, Man United Didesak Penggemar

Jumlah kata : 301  
Similarity : 12.79%  
Kalimat pertama : KOMPAS.com - Manchester United mendapat desakan dari para penggemar setelah Donny van de Beek kembali tampil gemilang bersama tim nasional Belanda.

Lini Serang Arsenal Melempem, Kalah Sangar dari Bek Man United Harry Maguire

Jumlah kata : 172  
Similarity : 12.17%  
Kalimat pertama : KOMPAS.com - Bek termahal dunia, Harry Maguire, tercatat tampil lebih agresif ketimbang dua penyerang Arsenal yakni Pierre-Emerick Aubameyang dan Alexandre Lacazette di Premier League musim ini.

Sangar di Lapangan Tengah, Fernandes Ternyata Awali Karier sebagai Bek

Jumlah kata : 168  
Similarity : 11.41%  
Kalimat pertama : KOMPAS.com - Bruno Fernandes menjelma sebagai salah satu gelandang terbaik.

Melwood, Ikon Liverpool setelah "Ruang Sepatu" di Anfield

Jumlah kata : 212  
Similarity : 3.46%  
Kalimat pertama : LIVERPOOL, KOMPAS.com - Pusat latihan di Melwood akan menjadi ikon Liverpool setelah "Ruang Sepatu" atau Boot Room di Stadion Anfield.

Penjelasan Southgate Soal Cedera Joe Gomez di Latihan Timnas Inggris

Jumlah kata : 298  
Similarity : 3.05%  
Kalimat pertama : KOMPAS.com - Liverpool kembali mendapat pukulan seputar kebugaran pemain dengan kabar cederanya Joe Gomez saat berlatih bersama Timnas Inggris pada Rabu (11/11/2020).

Crystal Palace Tambah Mitra Baru di Jersey Latihan

Jumlah kata : 161  
Similarity : 2.14%  
Kalimat pertama : LONDON, KOMPAS.com - Klub Liga Primer Crystal Palace menambah mitra baru di jersey latihannya.

Curhat Nicolas Pepe yang Ingin Tersenyum Lagi bersama Arsenal

Jumlah kata : 180  
Similarity : 2.02%  
Kalimat pertama : KOMPAS.com - Winger Arsenal, Nicolas Pepe, mengungkapkan unek-uneknya selama tampil di bawah kepemimpinan Mikel Arteta.

Terkait Cedera Joe Gomez, Southgate Ajak Bicara Juergen Klopp

Jumlah kata : 208  
Similarity : 1.77%  
Kalimat pertama : KOMPAS.com - Pelatih tim nasional Inggris, Gareth Southgate, berbicara kepada Juergen Klopp, menyusul cedera yang menimpa salah satu pemain Liverpool, Joe Gomez.

Inggris Vs Irlandia, Kans The Three Lions Akhiri "Kutukan"

Jumlah kata : 215  
Similarity : 0.00%  
Kalimat pertama : KOMPAS.com - Tim nasional Inggris akan menjamu Republik Irlandia pada pertandingan persahabatan jelang lanjutan UEFA Nations League 2020-2021.

Mo Salah Positif Covid-19, Bukti Video dan Pendirian Wali Kota Berseberangan

Jumlah kata : 274  
Similarity : 0.00%  
Kalimat pertama : KOMPAS.com - Wali kota Nagrig, Maher Shtiyah, kukuh dalam pendapatnya bahwa penyerang Liverpool, Mohamed Salah, tetap menjaga jarak fisik dan mematuhi protokol kesehatan saat menghadiri pernikahan adiknya, Nasr Salah.

Bendtner Kecanduan Judi, Sempat Rugi Rp 7,4 Miliar dalam 1 Malam

Jumlah kata : 166  
Similarity : 0.00%  
Kalimat pertama : KOMPAS.com - Mantan striker Denmark yang pernah bermain untuk Arsenal, Nicklas Bendtner, mengaku pernah kecanduan judi.

Bersama Jose Mourinho, Harry Kane Optimis Tottenham Hotspur Bisa Juara

Jumlah kata : 196  
Similarity : 0.00%  
Kalimat pertama : KOMPAS.com - Bomber Tottenham Hotspur, Harry Kane, mengaku punya keyakinan timnya bisa meraih gelar juara pada musim ini.

James Rodríguez Moncer, Ancelotti Ingin Bajak Bintang Real Madrid Lagi

Jumlah kata : 210  
Similarity : 0.00%  
Kalimat pertama : KOMPAS.com - Pelatih Everton, Carlo Ancelotti, dikabarkan ingin membajak bintang Real Madrid lagi setelah proyek James Rodríguez berhasil awal musim ini.

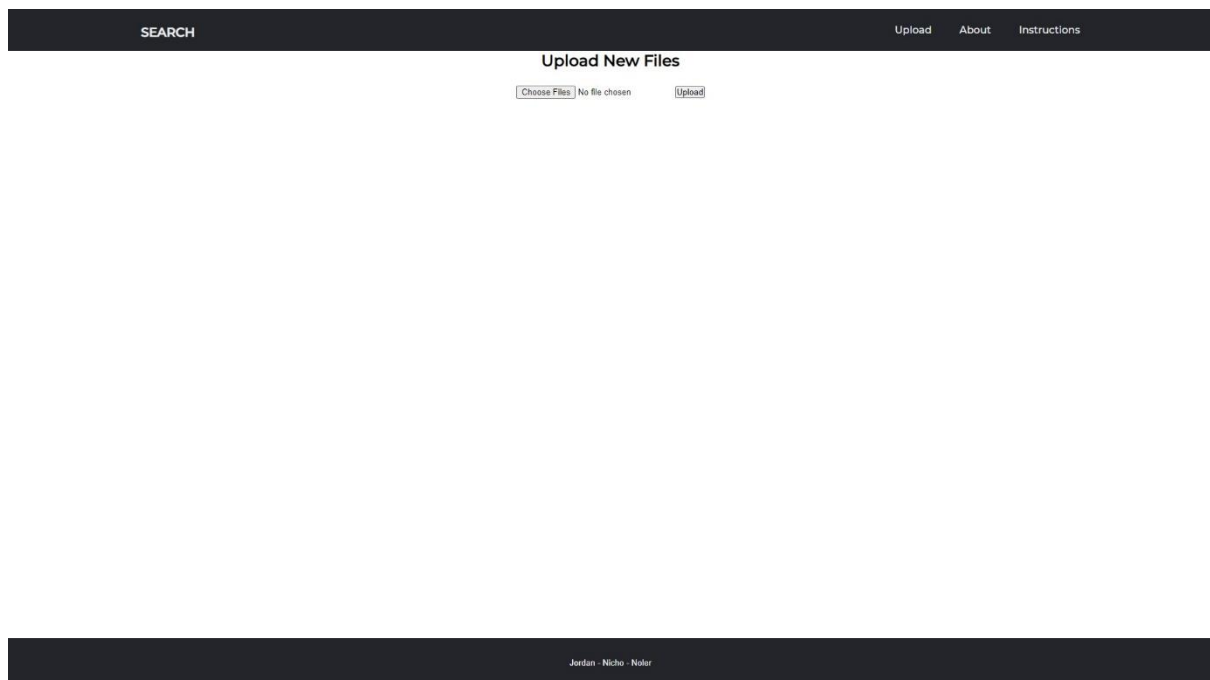
TABEL

Gambar 4.3 tampilan laman hasil search

TUGAS BESAR 2 ALJABAR LINIER DAN GEOMETRI 2020/2021

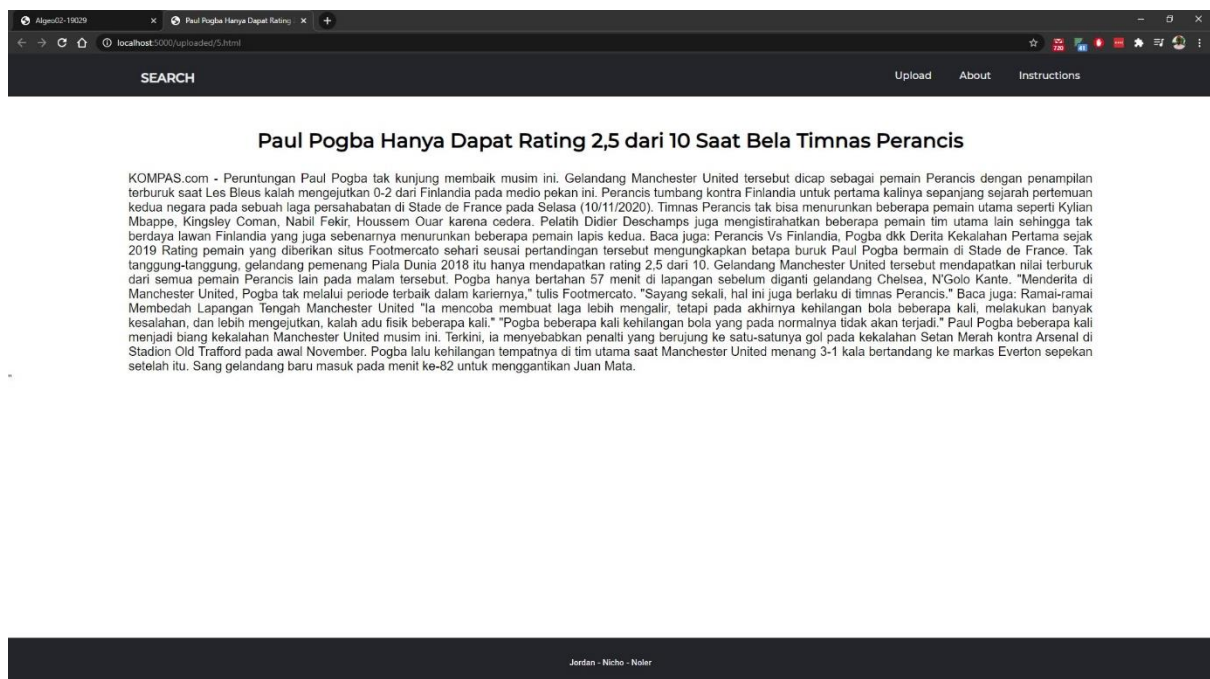
12

## 4.3 Laman Upload



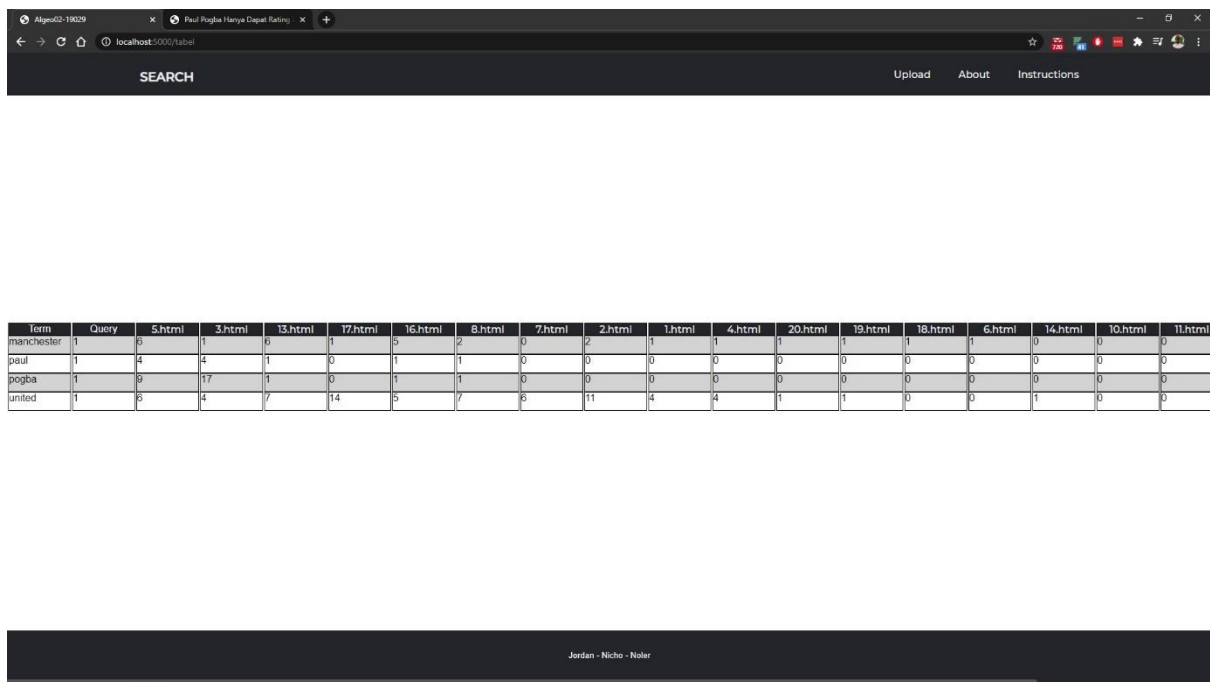
Gambar 4.4 tampilan laman untuk upload file html

## 4.4 Contoh Dokumen HTML



Gambar 4.5 tampilan laman untuk lihat plain html file

## 4.5 Laman Statistik Hasil Pencarian

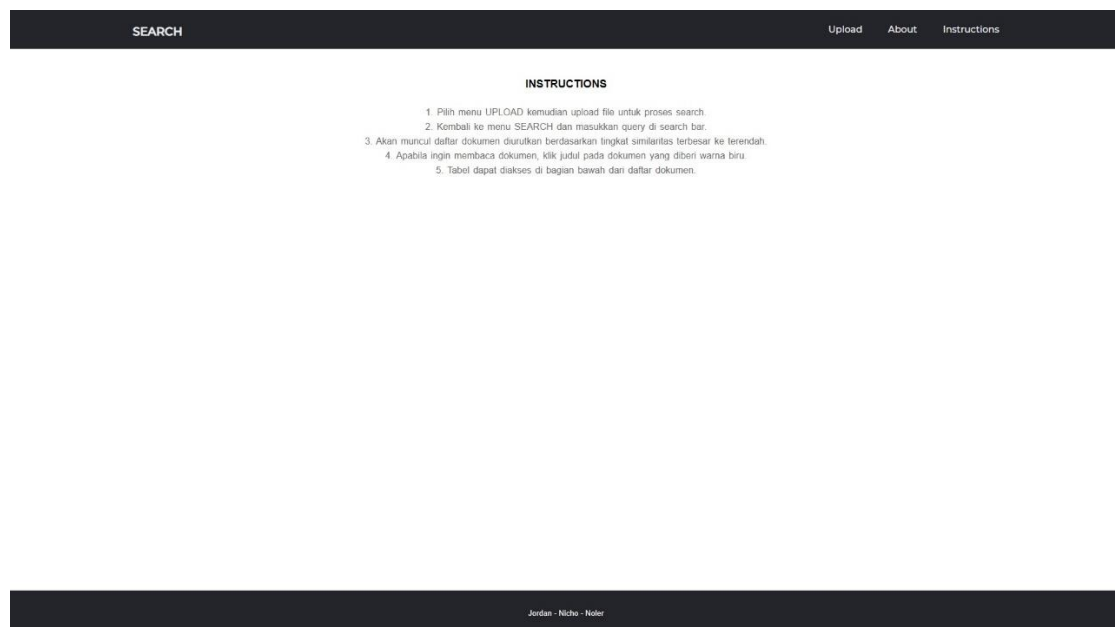


The screenshot shows a web browser window with the address bar displaying 'localhost:5000/tabel'. The page has a dark header with 'SEARCH' on the left and 'Upload', 'About', and 'Instructions' on the right. The main content area contains a table with search results. The footer of the page reads 'Jordan - Nicho - Noler'.

Term	Query	5.html	3.html	13.html	17.html	16.html	8.html	7.html	2.html	1.html	4.html	20.html	19.html	18.html	6.html	14.html	10.html	11.html
manchester	1	3	1	0	1	0	2	0	2	1	1	1	1	1	0	0	0	0
paul	1	4	4	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0
pogba	1	9	17	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0
united	1	8	4	7	14	5	7	8	11	4	4	1	1	0	0	1	0	0

Gambar 4.6 tampilan laman data statistik dari hasil pencarian

## 4.6 Laman Instruction



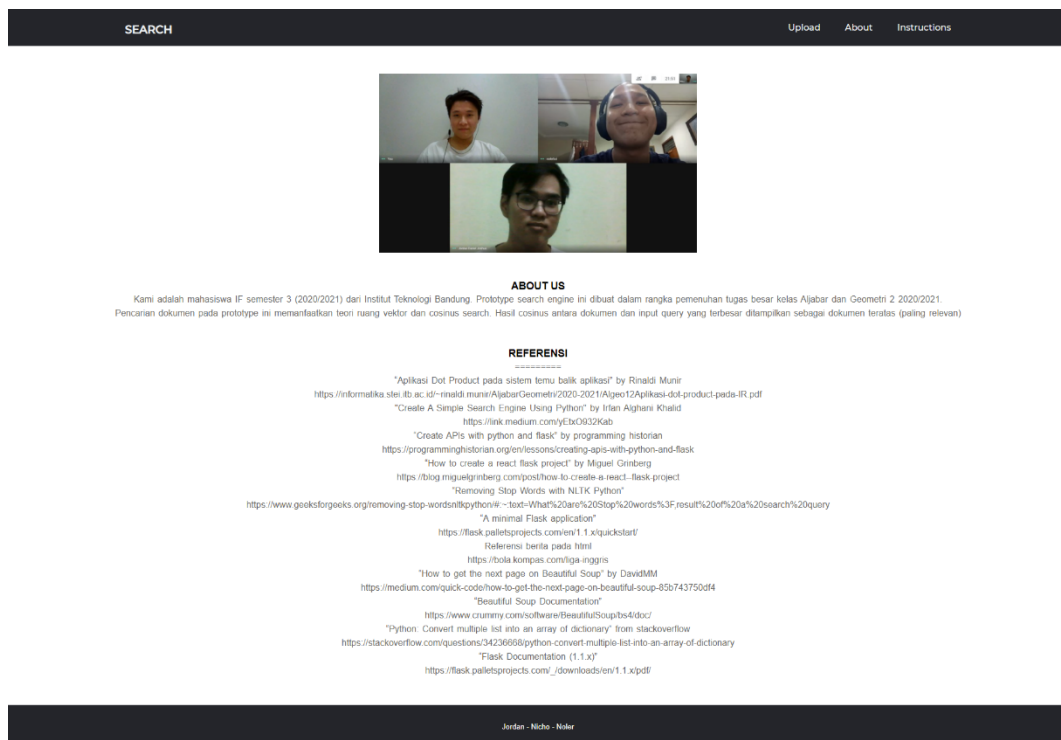
The screenshot shows a web browser window with the address bar displaying 'localhost:5000/tabel'. The page has a dark header with 'SEARCH' on the left and 'Upload', 'About', and 'Instructions' on the right. The main content area contains a section titled 'INSTRUCTIONS' with a list of five steps. The footer of the page reads 'Jordan - Nicho - Noler'.

**INSTRUCTIONS**

1. Pilih menu UPLOAD kemudian upload file untuk proses search.
2. Kembali ke menu SEARCH dan masukkan query di search bar.
3. Akan muncul daftar dokumen diurutkan berdasarkan tingkat similaritas terbesar ke terendah.
4. Apabila ingin membaca dokumen, klik judul pada dokumen yang diberi warna biru.
5. Tabel dapat diakses di bagian bawah dari daftar dokumen.

Gambar 4.7 tampilan laman instruksi

## 4.6 Laman About Us



*Gambar 4.8 tampilan laman about us*



## **BAB V**

### **KESIMPULAN, SARAN dan REFLEKSI**

#### **5.1 Kesimpulan**

Dengan program ini, pengguna dapat memasukkan sekumpulan file-file html kedalam search engine ini untuk melakukan pencarian dengan query tertentu. Hasil dari pencarian akan ditampilkan terurut dari atas sebagai yang paling relevan ke bawah. Metode pencarian yang digunakan pada search engine ini adalah hasil implementasi teori vektor ruang.

#### **5.2 Saran**

Metode ini dapat digunakan secara efektif apabila jumlah dokumen relatif sedikit. Untuk penanganan dokumen dalam jumlah besar seperti google search engine harus menggunakan metode yang lain. Kelemahan lain yang dapat diperbaiki pada search engine ini antara lain; urutan kata pada query tidak menentukan hasil searching, hanya jumlah katanya saja yang menentukan, kata-kata dasar seperti kata ganti orang (aku,kamu,dia) dan imbuhan (sejak, dan, atau, jikalau) tidak dapat digunakan dalam pencarian (karena metode stemming dengan library nltk).

#### **5.3 Refleksi**

Dari pengerjaan tugas ini, kami banyak belajar mengenai html dan konstruksi web sederhana. Selain itu, kami juga belajar bahwa harus teliti dalam mencari kesalahan-kesalahan seperti kesalahan hasil perhitungan similarity yang terlalu besar, kesalahan perhitungan jumlah kata pada berita karena variabel yang digunakan sama untuk setiap file dan lupa direset menjadi nol, serta kesalahan dan kelalaian minor lainnya yang dapat merusak keberjalanannya program.

## DAFTAR REFERENSI

"Aplikasi Dot Product pada sistem temu balik aplikasi" by Rinaldi Munir

<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo12Aplikasi-dot-product-pada-IR.pdf>

"Create A Simple Search Engine Using Python" by Irfan Alghani Khalid

<https://link.medium.com/yEtXO932Kab>

"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-wordsnltkpython/#:~:text=What%20are%20Stop%20words%3F,result%20of%20a%20search%20query>

"A minimal Flask application"

<https://flask.palletsprojects.com/en/1.1.x/quickstart/>

Referensi berita pada html

<https://bola.kompas.com/liga-inggris>

"How to get the next page on BeautifulSoup" by DavidMM

<https://medium.com/quick-code/how-to-get-the-next-page-on-beautiful-soup-85b743750df4>

"Beautiful Soup Documentation"

<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

"Python: Convert multiple list into an array of dictionary" from stackoverflow

<https://stackoverflow.com/questions/34236668/python-convert-multiple-list-into-an-array-of-dictionary>

"Flask Documentation (1.1.x)"

<https://flask.palletsprojects.com/ /downloads/en/1.1.x/pdf/>