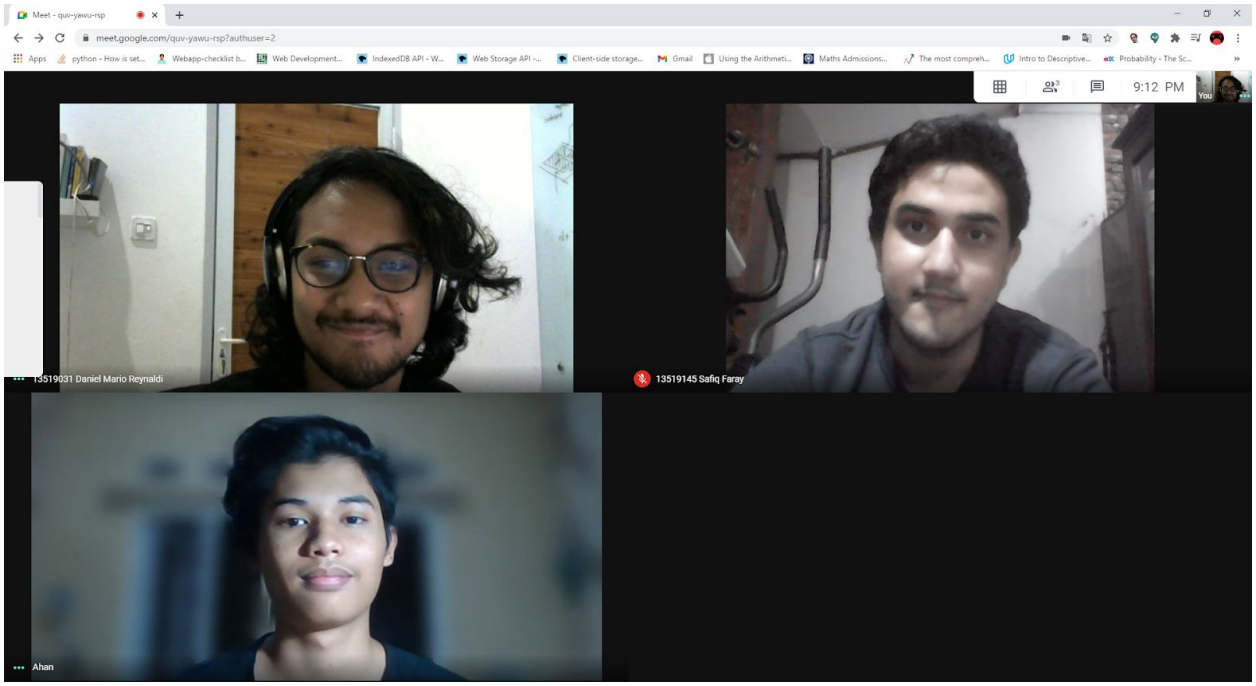
Laporan Tugas Besar 2
Mata Kuliah Aljabar Linier dan Geometri
IF2123 2020/2021



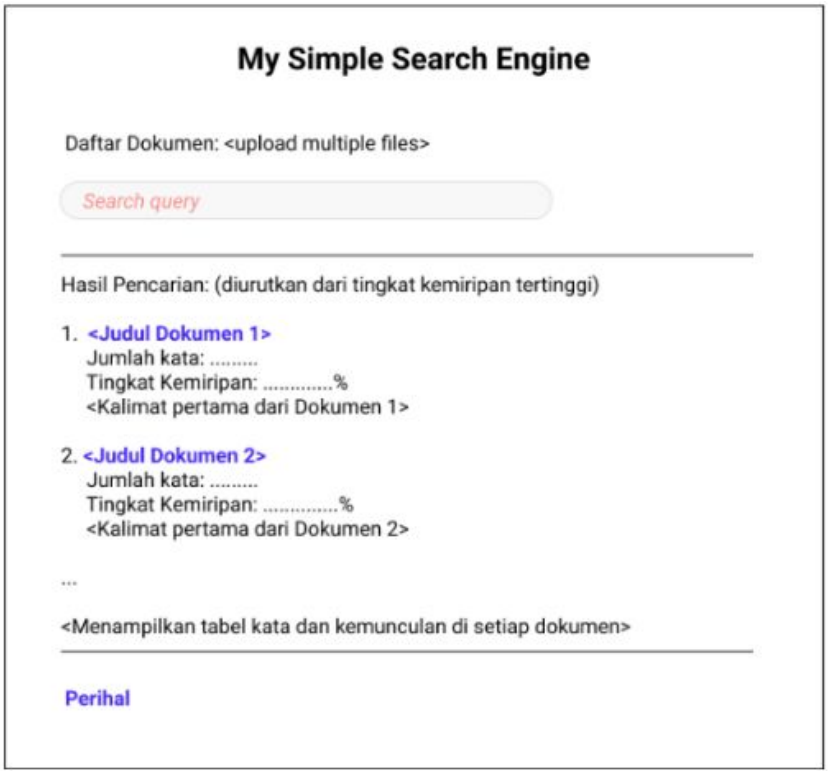
Daniel Mario Reynaldi/13519031
Farhan Nur Hidayat Denira/13519071
Safiq Faray/13519145

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

Bab 1
Deskripsi Permasalahan

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. Namun, 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.

Pada tugas besar kali ini, kami diminta untuk membuat sebuah search engine sederhana dengan menggunakan model ruang vektor dan memanfaatkan cosine similarity. Search engine yang akan dibangun diminta memiliki layout yang mirip dengan gambar berikut:



Search engine akan menerima input dari pengguna, yaitu Search query dan beberapa dokumen, dimana search engine akan melakukan pencarian search query pada dokumen-dokumen tersebut. Setelah menerima input search engine akan menampilkan hasil pencarian yang diurutkan berdasarkan tingkat kemiripan, mulai dari dokumen dengan tingkat kemiripan tertinggi.

Pada halaman pencarian, pengguna dapat menekan judul dokumen yang diinginkan, lalu program akan mengarahkan pengguna pada sebuah halaman yang berisi full-text terkait dokumen tersebut. Halaman pencarian juga menampilkan tabel kata dan kemunculan dari setiap dokumen, dengan tabel seperti berikut:

Term	Query	D1	D2	...	D3
Term1					
Term2					
...					
TermN					

Pembuatan search engine juga dibuat dengan memperhatikan hal-hal berikut:

1. Silahkan lakukan 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. Menggunakan bahasa indonesia atau bahasa inggris.

Search engine memiliki spesifikasi 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 2
Dasar Teori

Information Retrieval (IR) atau sering disebut “temu kembali informasi” adalah ilmu yang mempelajari prosedur-prosedur dan metode-metode untuk menemukan kembali informasi yang tersimpan dari berbagai sumber (resources) yang relevan atau koleksi sumber informasi yang dicari atau dibutuhkan. Dengan tindakan index (indexing), panggilan (searching), pemanggilan data kembali (recalling).

Dalam pencarian data, beberapa jenis data dapat ditemukan diantaranya texts, table, gambar (image), video, audio. Adapun tujuan dari Information Retrieval adalah untuk memenuhi informasi pengguna dengan cara meretrieve dokumen yang relevan atau mengurangi dokumen pencarian yang tidak relevan.

Vektor merupakan adalah objek geometri yang memiliki besar dan arah. Salah satu model dari Information Retrieval (IR) model ruang vektor, dimana model ini menggunakan teori di dalam aljabar vektor. 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 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:

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

Cosine similarity sendiri berfungsi untuk membandingkan kemiripan antar dokumen, dalam hal ini yang dibandingkan adalah query dengan dokumen latih. Dalam menghitung cosine similarity, pertama yang dilakukan yaitu melakukan perkalian skalar antara query dengan dokumen kemudian dijumlahkan, setelah itu melakukan perkalian antara panjang dokumen dengan panjang query yang telah dikuadratkan, setelah itu di hitung akar pangkat dua. Selanjutnya hasil perkalian skalar tersebut dibagi dengan hasil perkalian panjang dokumen dan query.

Bab 3
Implementasi Program

Algoritma Search Engine

```
@app.route('/search_result', methods=['GET','POST'])
def search_result():

    results = []
    querylib = []
    query_term = 0
    qvector = []

    if(request.method=='POST'):

        query = request.form.get("search")

        if(query):
            #Proses Pre Processing (stemming dan stop words removal) serta vektorisasi query
            query_processed = Preprocessing(query)
            querylib = Querylib(query_processed)
            qvector = Vectorizer(querylib,query_processed)
            query_term = len(querylib)

            results = []
            filepath = './static/'
            for filename in glob.glob(os.path.join(filepath, '*.html')):
                with open(os.path.join(os.getcwd(),filename), 'r') as f:
                    #Proses Pre Processing (stemming dan stop words removal) serta vektorisasi text dokumen
                    #Hasil Preprocessing dan vektorisasi disimpan didalam tuple berisi (tingkat kemiripan, vektor,judul,kalimat pertama,link, jumlah kata)
                    soup = BeautifulSoup(f, 'html.parser')
                    text = HtmlParser(soup)
                    title = HtmlTitleParse(soup)
                    text_processed = Preprocessing(text)
                    textlib = Querylib(text_processed)
                    tqvector = Vectorizer(querylib,text_processed)
                    tvector = Vectorizer(textlib,text_processed)
                    sim = cosine_similiarity(qvector,tqvector,tvector)
                    numofwords = Wcounter(text)
                    link = title + '.html'
                    first_sentence = getFirstSen(text)
                    res = (sim,tqvector,title,first_sentence,link,numofwords)
                    results.append(res)

                Sort(results)  #Diurutkan berdasarkan tingkat kemiripan paling besar ke paling kecil
    return render_template('search_result.html', query=query, results=results, query_term=query_term, querylib=querylib,qvector=qvector)
```

Stemming dan Stop word removal

```
def Preprocessing(text):
#{Fungsi yang menerima text dan melakukan stemming serta stop word removal}
#{Mengembalikan list berisi kata-kata yang telah diproses}
    stop_words = set(stopwords.words('english'))
    ps = PorterStemmer()

    token = word_tokenize(text)
    filtered = []
    for word in token:
        if word not in stop_words:
            filtered.append(ps.stem(word))
    return filtered
```

Vectorizer

```
def Vectorizer(querylib,bag_of_words):
#{Mengubah array berisi kata-kata menjadi vektor yang dapat dihitung dalam cosine similarity}
#{Parameter fungsi berupa library kata-kata dan array yang berisi kata kata yang telah di preprocessing}
    n = len(querylib)
    vector = [0 for i in range(n)]

    for i in range(n):
        for word in bag_of_words:
            if querylib[i] == word:
                vector[i] += 1
    return vector
```

Cosine Similarity

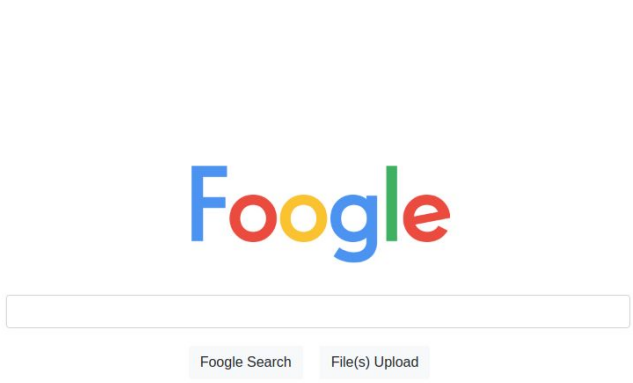
```
def cosine_similiarity(vector1,vector2):
#{Menghitung cosine similarity dari 2 buah vector}
    n = len(vector1)
    result = 0
    for i in range(n):
        result += (vector1[i]*vector2[i])
    if(magnitude(vector1)==0 or magnitude(vector2)==0):
        return 0
    else:
        return (result/(magnitude(vector1)*magnitude(vector3)))
```

Langkah-langkah Algoritma Search Engine:

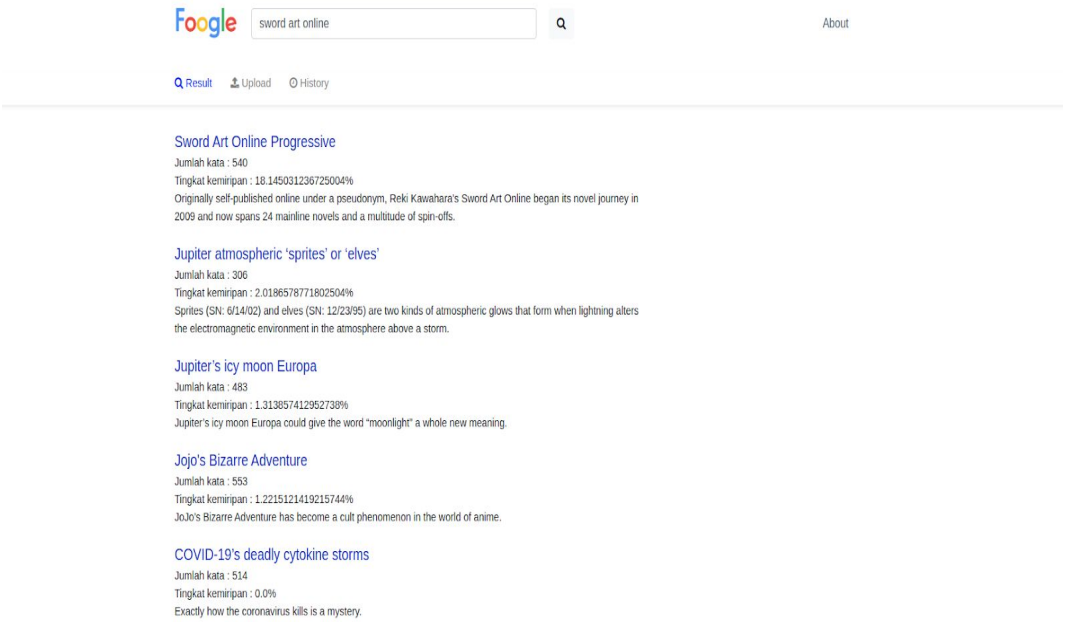
1. Parsing dokumen html menjadi string yang dapat diproses oleh Python dengan library BeautifulSoup.
2. PreProcessing string text dengan fungsi preprocessing. Preprocessing meliputi mengubah string menjadi list of word/bag of words lalu stop word removal dan stemming.
3. Setelah preprocessing akan dibuat library yaitu list kata kata pada bag of words, tujuannya adalah mempermudah vektorisasi. Setiap kata menempati indeks tertentu pada list.
4. Vektorisasi adalah proses mengubah bag of words menjadi vektor yang dapat diproses dengan cosine similarity. Vektorisasi dilakukan dengan mencocokan kata kata yang ada pada library dengan kata-kata pada bag of words.
5. Setelah vektorisasi hasilnya dilakukan cosine similarity antara vektor query dengan vektor text/dokumen.
6. Hasil cosine similarity dimasukan ke dalam array bersama hasil teks lainnya dan kemudian diurutkan terurut mengecil.

Bab 4
Eksperimen

Hasil Front-End



Hasil Search Engine



Term	Query	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
sword	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
art	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
onlin	1	5	1	1	1	0	0	0	0	0	0	0	0	0	0	0

Bab 5
Kesimpulan, Saran, dan Refleksi

5.1. Kesimpulan

- Materi Ruang Vektor memiliki banyak sekali pemanfaatan pada kehidupan, salah satunya adalah pemanfaatan ruang vektor dalam Information Information Retrieval (IR) atau “Sistem Temu Balik Informasi”.
- Information Retrieval (IR) atau “Sistem Temu Balik Informasi” dengan model ruang vektor dapat dibuat dengan menyatakan *search query* dan jumlah kemunculan kata sebagai vektor pada suatu ruang vektor.
- *Similarity measure* antara *query* dan dokumen dapat ditentukan dengan persamaan berikut:

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

  Dimana vektor **Q** adalah query dan vektor **D** adalah jumlah kemunculan kata pada dokumen. Semakin besar nilai *similarity measure,* maka semakin relevan dan sesuai antara *search query* dan dokumen tersebut.
- Python merupakan bahasa pemrograman yang memiliki kapabilitas sangat luas, salah satunya adalah python dapat digunakan untuk membuat website, dengan memanfaatkan salah satu *library* yang tersedia yaitu “Flask”.
- *Stemming* kata, penghapusan *stopwords,* dan *web-scraping* dapat dilakukan untuk membuat *search engine* menampilkan hasil yang lebih relevan.

5.2 Saran

- Sebaiknya diberikan sedikit pembekalan ilmu mengenai *web development* sebelumnya, agar mahasiswa setidaknya memiliki bayangan mengenai pembuatan website.

5.3 Refleksi

- Mahasiswa menjadi paham mengenai pemanfaatan ruang vektor dalam “Sistem temu balik informasi”.
- Mahasiswa menjadi paham mengenai pembuatan website dan *web development.*
- Mahasiswa memahami cara membuat sebuah *search engine* sederhana dengan memanfaatkan model ruang vektor.
- Mahasiswa memahami penggunaan bahasa pemrograman *Python* lebih dalam.
- Mahasiswa memahami penggunaan *Github* lebih dalam.
- Mahasiswa memahami cara bekerjasama dan pembagian tugas antar anggota.
- Mahasiswa memahami *time management* yang penting dalam pelaksanaan tugas berkelompok.

5.4. Referensi

- https://www.payahtidur.com/project/cosine-similarity
- https://ligiaprapta17.wordpress.com/2015/03/03/pengertian-information-retrieval-ir-peranan-ir-dan-contoh-contoh-ir/
- https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-12-Aplikasi-dot-product-pada-IR.pdf