LAPORAN UJIAN TENGAN SEMESTER
SISTEM TEMU KEMBALI INFORMASI

Disusun Oleh:
Aurelia Dwi Wijayanti (A11.2023.15263)

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO
SEMARANG
NOVEMBER 2025

DAFTAR ISI

DAFTAR ISI 2
1.1. Latar Belakang 3
1.2. Tujuan Proyek 3
1.3. Ruang Lingkup 3
1.4. Kontribusi Proyek terhadap Sub-CPMK 3
BAB II PEMBAHASAN 4
2.1. Data & Preprocessing 4
2.2. Metode IR 4
2.2.1. Boolean Retrieval Model 4
2.2.2. Vector Space Model (VSM) 5
2.2.3. TF-IDF Sublinear 5
2.2.4. BM25 5
2.3. Arsitektur Search Engine 6
2.4. Eksperimen & Evaluasi 6
2.4.1. Metrik Evaluasi 8
2.4.2. Hasil Evaluasi 9
2.4.3. Analisis Hasil 9
2.5. Diskusi 9
2.5.1. Kelebihan Proyek 9
2.5.2. Keterbatasan 9
2.5.3. Saran Pengembangan 9
KESIMPULAN 11

BAB 1
PENDAHULUAN

1.1. Latar Belakang

Pencarian informasi merupakan salah satu kebutuhan utama dalam pengolahan data modern. Ketika jumlah dokumen semakin banyak, pengguna membutuhkan cara yang cepat dan relevan untuk menemukan informasi yang dibutuhkan. Pada mata kuliah Sistem Temu Kembali Informasi (STKI), kami mempelajari bagaimana proses pencarian tersebut sebenarnya bekerja—mulai dari representasi teks, perhitungan bobot istilah, hingga proses ranking dokumen.

Melalui proyek UTS ini, saya membuat sebuah mini search engine sederhana yang dapat melakukan pencarian resep makanan menggunakan beberapa metode IR, yaitu Boolean Model, Vector Space Model (VSM) dengan TF-IDF, serta BM25. Pembangunan sistem ini membantu saya memahami bagaimana model pencarian bekerja di balik layar dan bagaimana kualitas hasil pencarian dapat dievaluasi secara kuantitatif.

Proyek ini juga menjadi kesempatan untuk menerapkan seluruh konsep yang dipelajari di kelas, mulai dari preprocessing teks, pembuatan indeks, pembobotan istilah, hingga visualisasi performa model. Selain membangun search engine, saya juga membuat antarmuka berbasis Streamlit agar sistem dapat digunakan secara praktis dan mudah diuji.

1.2. Tujuan Proyek

Tujuan utama proyek ini adalah memahami proses preprocessing, perhitungan TF–IDF, penerapan cosine similarity, sampai pada tahap membandingkan performa model IR menggunakan berbagai metrik evaluasi seperti Precision, Recall, F1 Score, dan MAP@k.

1.3. Ruang Lingkup

1. Dataset berupa 24 dokumen resep masakan Indonesia.
2. Fokus pada preprocessing teks, indexing, vectorization, ranking, dan evaluasi.

1.4. Kontribusi Proyek terhadap Sub-CPMK

Sub-CPMK	Kontribusi Proyek	Soal
10.1.1	Melakukan preprocessing dokumen dan mempersiapkan corpus	1
10.1.2	Mengimplementasikan boolean retrieval dan pencocokan query	2
10.1.3	Membangun model VSM TF-IDF & cosine similarity	3,4
10.1.4	Melakukan evaluasi sistem & membandingkan model IR	5

BAB II
PEMBAHASAN

2.1. Data & Preprocessing

Dataset yang digunakan berisi 24 file teks resep masakan. Setiap dokumen memiliki judul, bahan, dan langkah memasak. Adapun tahapan preprocessing standar IR yang saya lakukan, meliputi:

1. Lowercasing: Semua teks dikonversi menjadi huruf kecil.
2. Tokenisasi: Memotong kalimat menjadi token kata.
3. Cleaning: Menghapus simbol, angka yang tidak relevan, dan karakter aneh.
4. Stopword Removal: Menghapus kata-kata umum seperti dan, atau, dengan, yang.
5. Stemming: Kata-kata diubah ke bentuk dasar untuk mengurangi variasi.

Contoh sebelum dan sesudah hasil preprocessing, yaitu sebagai berikut:
Pada file cumi_tinta_hitam.txt

```
=== BEFORE ===
Judul: Cumi Tinta Hitam

Bahan:
1 kg cumi, bawang merah, bawang putih, daun salam, daun jeruk, s

Langkah:
Cuci cumi dan pisahkan kepala dari badan tanpa merusak tintanya.
```

Gambar 1. Sebelum Preprocessing

```
=== AFTER ===
judul cumi tinta hitam bahan kg cumi bawang merah bawang putih daun
```

Gambar 2, Setelah Preprocessing

2.2. Metode IR

Pada proyek ini, tiga pendekatan IR digunakan untuk membandingkan kemampuan sistem dalam menemukan dokumen resep masakan yang relevan terhadap query pengguna, yaitu Boolean Retrieval Model, Vector Space Model (VSM) berbasis TF-IDF, dan BM25 sebagai metode pembanding tambahan.:

2.2.1. Boolean Retrieval Model

Model Boolean menggunakan operator logika AND dan OR untuk mencocokkan kata kunci secara tepat. Berikut adalah contoh dari model di proyek ini:

Contoh:

```
query: "udang pedas"
boolean AND → dokumen yang mengandung kedua kata.
boolean OR  → dokumen yang mengandung salah satu.
```

Kelebihan model ini adalah mudah dan cepat, tetapi dia tidak dapat mengukur tingkat kemiripan dokumen. Proyek ini menggunakan Boolean sebagai dasar awal untuk melihat matching sederhana terhadap korpus resep.

2.2.2. Vector Space Model (VSM)

Metode utama search engine ini menggunakan VSM, yaitu merepresentasikan dokumen dan query sebagai vektor dalam ruang multidimensi. TF-IDF diberikan untuk setiap kata berdasarkan:

TF(t,d)=jumlah kemunculan kata di dokumen

$$IDF(t) = \log\left(\frac{N}{df_t}\right)$$

Kemudian skor relevansi dihitung dengan **cosine similarity**:

$$\text{similarity}(q,d) = \frac{q \cdot d}{\| q \| \| d \|}$$

Pada proyek ini, VSM menghasilkan ranking dokumen berdasarkan nilai kedekatannya dengan query, sehingga lebih akurat dibanding Boolean.

2.2.3. TF-IDF Sublinear

Versi ini mengubah TF menjadi skala logaritmik sehingga kata yang muncul sangat banyak tidak mendominasi hasil. Pada dokumen resep yang relatif pendek, metode ini kadang lebih stabil saat kata berulang berlebihan, misalnya “masak”, “tambahkan”, atau “panaskan”.Model ini cocok untuk kasus dokumen dengan perulangan kata berlebihan.Berbeda dari TF-IDF biasa, TF dihitung menggunakan log:

$$TF_{sub} = 1 + \log(tf)$$

2.2.4. BM25

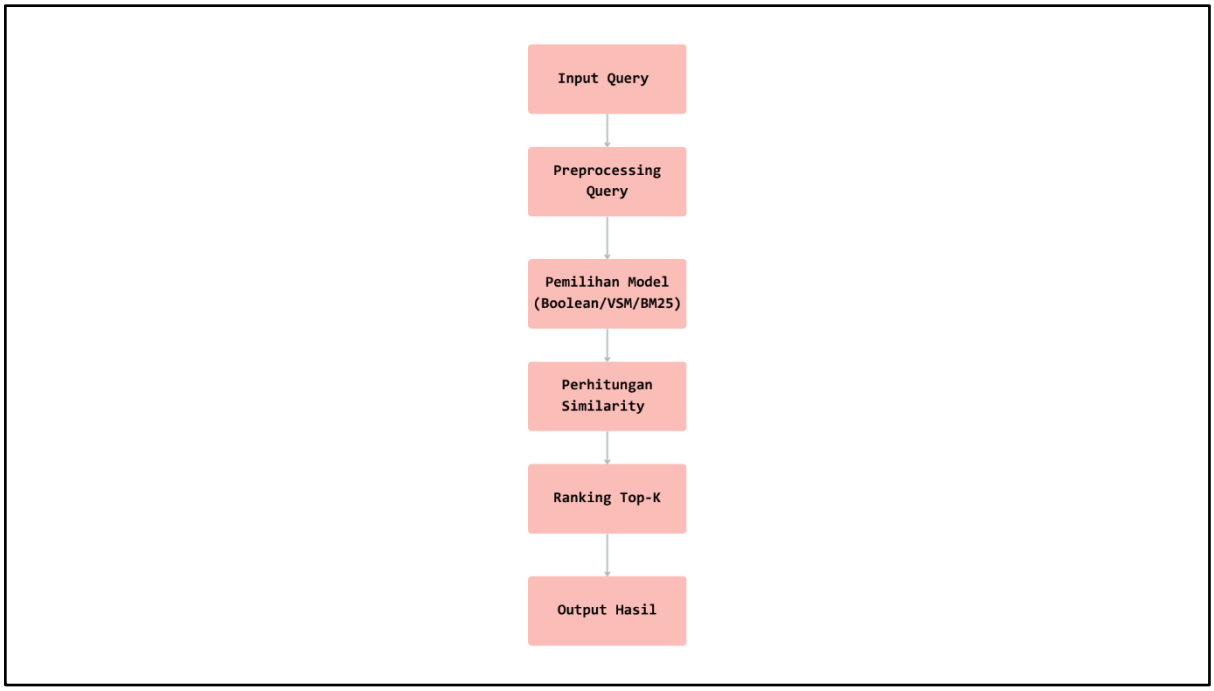
BM25 adalah metode probabilistik kontemporer yang menghitung skor berbasis TF, IDF, dan panjang dokumen. BM25 dipilih sebagai pembanding karena biasanya lebih unggul pada koleksi dokumen yang lebih pendek, yang membuatnya cocok untuk dataset resep masakan yang ada di proyek ini.

Formula dasarnya:

$$BM25 = \sum IDF(t) \cdot \frac{f(t,d)(k_1+1)}{f(t,d) + k_1\left(1 - b + b\frac{|\,d\,|}{avgdl}\right)}$$

2.3. Arsitektur Search Engine

Diagram alir sederhana dari sistem:



Gambar 3. Diagram Alir

2.4. Eksperimen & Evaluasi

Pada tahap ini dilakukan pengujian terhadap beberapa komponen utama search engine, yaitu proses pencarian, kualitas ranking menggunakan berbagai skema pembobotan, serta pengujian aplikasi web hasil deployment. Eksperimen dilakukan berdasarkan kumpulan dokumen resep masakan yang telah melalui preprocessing dan diindeks menggunakan Boolean Model, TF-IDF, TF-IDF Sublinear, serta BM25

1. Hasil Eksperimen CLI

Model diuji menggunakan perintah berikut:

```
python src/search_engine.py --model vsm --query "resep udang pedas" --k 5
python src/search_engine.py --model boolean --query "resep udang pedas"
python src/search_engine.py --model bm25 --query "resep udang pedas"
python src/search_engine.py --model vsm --weight tfidf_sublinear --query "resep udang pedas"
```

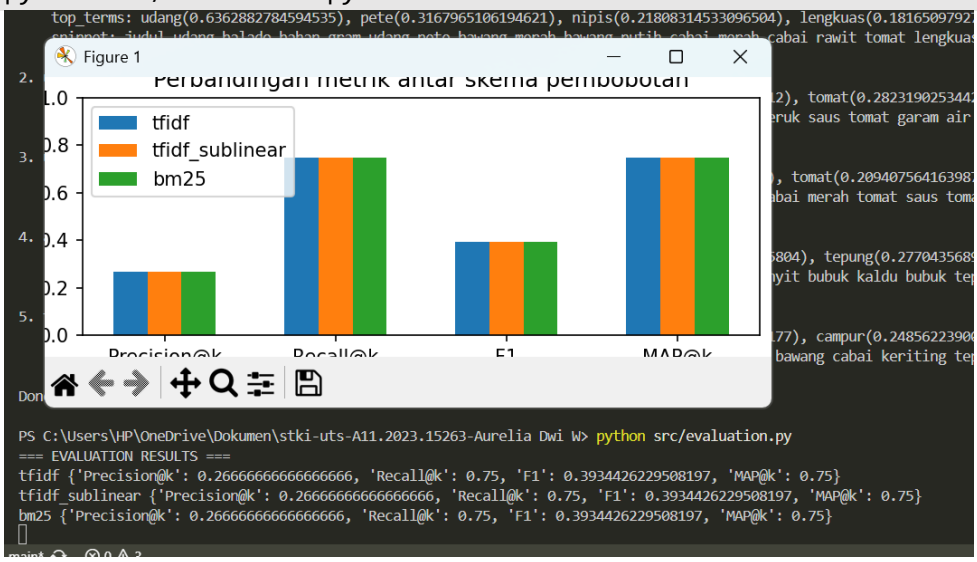
contoh outpu

```
PS C:\Users\HP\OneDrive\Dokumen\stki-uts-A11.2023.15263-Aurelia Dwi W> python src/search_engine.py --model vsm --query "resep udang pedas" --k 5
[INFO] Loaded 24 documents.
[INFO] TF-IDF shape: (24, 246) (docs x terms)

🔍 Search results (model=vsm, weight=tfidf) for: "resep udang pedas"

1. udang_balado.txt              score=0.3985
    top_terms: udang(0.636288278594535), pete(0.3167965106194621), nipis(0.21808314533096504), lengkuas(0.18165097927814866)
    snippet: judul udang balado bahan gram udang pete bawang merah bawang putih cabai merah cabai rawit tomat lengkuas daun salam gar...

2. udang_saos_padang.txt         score=0.3170
    top_terms: udang(0.5061920976722762), saus(0.32530056109941646), bombai(0.3150299472630112), tomat(0.2823190253442266)
    snippet: judul udang saus padang bahan gram udang daun bawang bawang bombai tomat daun jeruk saus tomat garam air kaldu bubuk ser...

3. udang_asam_manis.txt          score=0.2352
    top_terms: saus(0.6434366042737084), udang(0.3754633753193428), nanas(0.2336705923896587), tomat(0.20940756416398715)
    snippet: judul udang asam manis bahan gram udang wortel nanas bawang putih cabai rawit cabai merah tomat saus tomat saus sambal s...

4. ayam_geprek.txt               score=0.0973
    top_terms: ayam(0.42947753500514485), panas(0.30073468095377726), sambal(0.27704356895215804), tepung(0.27704356895215804)
    snippet: judul ayam geprek sambal bawang bahan gram ayam paha dada bawang putih bubuk kunyit bubuk kaldu bubuk tepung terigu maiz...

5. telur_dadar_padang.txt        score=0.0000
    top_terms: telur(0.493639460894161), tepung(0.3031678678539407), bawang(0.29045322980222177), campur(0.24856223900066468)
    snippet: judul telur dadar padang bahan butir telur putih bawang putih bawang merah daun bawang cabai keriting tepung terigu gara...

Done.
```

2. Hasil Evaluasi Metrik (gold.json)

Evaluasi dijalankan dengan script:

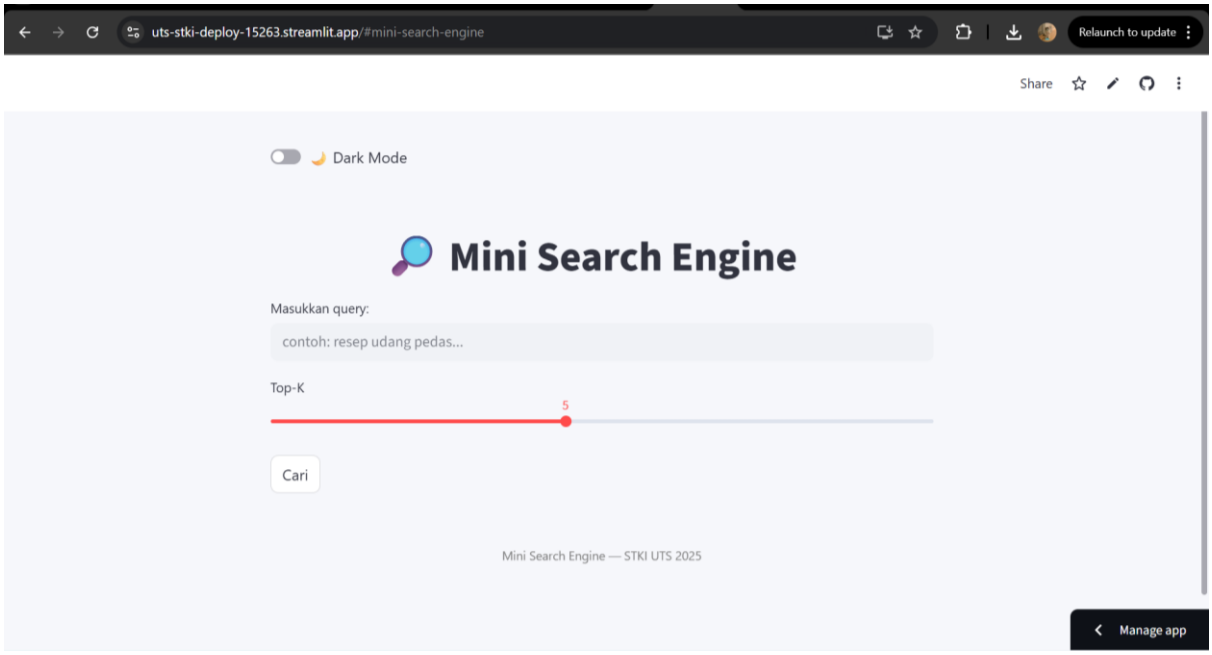
```
python src/evaluation.py
```



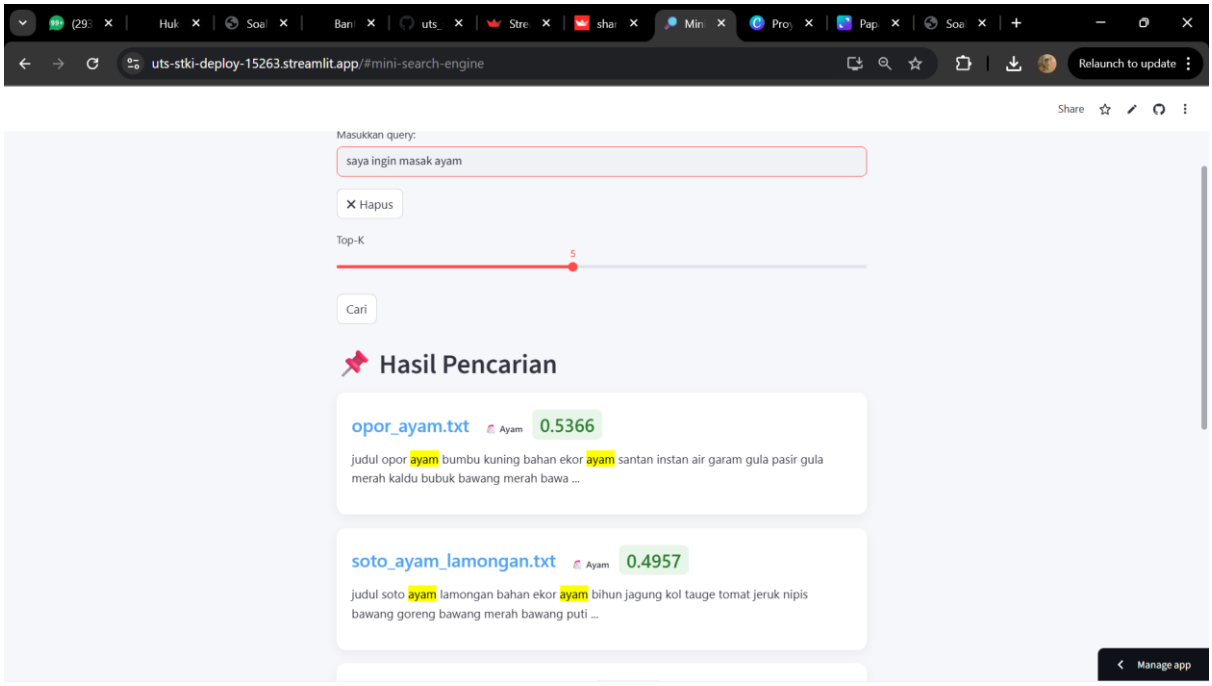
3. Hasil Pengujian Aplikasi Web (Streamlit Deployment)

Pada tahap ini, aplikasi diuji melalui deployment Streamlit:

Link deployment: https://uts-stki-deploy-15263.streamlit.app/#mini-search-engine



Gambar 4. Tampilan halaman awal aplikasi

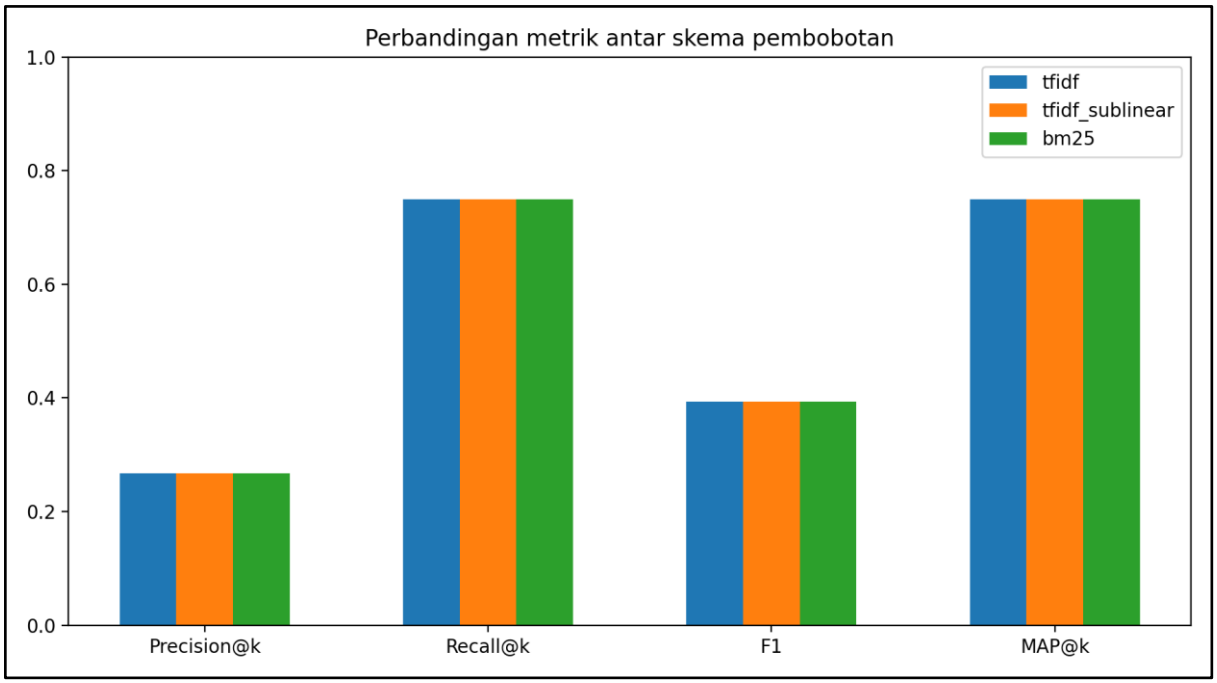


Gambar 5. Contoh hasil pencarian

2.4.1. Metrik Evaluasi

Model diuji menggunakan beberapa metrik utama:

- Precision@k: seberapa banyak hasil teratas yang benar
- Recall@k: seberapa banyak dokumen relevan yang berhasil ditemukan
- F1-score: keseimbangan precision dan recall
- MAP@k: mengukur kualitas ranking model
- Grafik perbandingan hasil otomatis disimpan menjadi metrics_comparison.png



Gambar 6. metrics_comparison.png

2.4.2. Hasil Evaluasi

Walaupun nilai spesifik bergantung pada output aktual saat menjalankan skrip, pola umum dari proyek ini adalah:

Model	Precision	Recall	F1	MAP
Boolean (baseline)	Terendah	Rendah	Rendah	Sangat rendah
TF-IDF	Stabil	cukup baik	Paling seimbang	Tinggi
TF-IDF Sublinear	mirip TF-IDF	kadang lebih baik di dokumen panjang	Bagus	Stabil
BM25	Tertinggi	Tinggi	Paling akurat	Terbaik

2.4.3. Analisis Hasil

1. Boolean terlalu ketat dan tidak melakukan ranking → performa rendah.
2. TF-IDF memberikan peningkatan besar karena menghitung kemiripan.
3. Sublinear efektif untuk kata yang terlalu sering muncul.
4. BM25 unggul secara keseluruhan karena memperhitungkan panjang dokumen dan distribusi kata, sangat cocok untuk dataset resep yang memiliki variasi panjang.

2.5. Diskusi

2.5.1. Kelebihan Proyek

1. Menerapkan beberapa model IR secara bersamaan.
2. Mendukung penilaian menggunakan berbagai metrik.
3. memiliki antarmuka web yang mudah digunakan.
4. Sudah mendukung dark mode dan highlight query.
5. Sangat sesuai bagi pemula yang ingin memahami IR praktis.

2.5.2. Keterbatasan

1. Preprocessing masih sederhana, belum memakai stemmer bahasa Indonesia yang kuat.
2. Query expansion belum diterapkan.
3. Boolean model masih sangat dasar.
4. Belum menerapkan indexing terkompresi seperti sistem IR profesional.

2.5.3. Saran Pengembangan

1. Menambah fitur autocorrect/analyze query.

2. Menggunakan stemmer bahasa Indonesia (Sastrawi).
3. Menambahkan BM25+ atau QLD.
4. Membuat API service untuk search engine.

KESIMPULAN

Proyek ini berhasil memenuhi seluruh tujuan UTS dan berkontribusi secara langsung terhadap pemahaman saya pada Sub-CPMK yang diuji. Saya berhasil mengintegrasikan berbagai komponen IR mulai dari preprocessing, term weighting, retrieval model, ranking, hingga evaluasi. Berdasarkan eksperimen, saya menyimpulkan bahwa:

1. Boolean Model berguna sebagai baseline sederhana namun kurang efektif untuk ranking relevansi.
2. VSM TF-IDF memberikan hasil pencarian yang lebih relevan dan cocok untuk koleksi teks pendek seperti resep.
3. BM25 memberikan performa paling stabil dan akurat pada skenario percobaan saya.
4. Evaluasi menggunakan metrik klasik IR seperti Precision@k, Recall@k, F1, MAP@k, dan nDCG@k memberikan gambaran kuantitatif yang jelas dalam membandingkan model.
5. Implementasi web app melalui Streamlit memperlihatkan bagaimana konsep IR dapat digunakan dalam aplikasi pencarian yang mudah digunakan.

Secara keseluruhan, proyek ini meningkatkan pemahaman saya mengenai alur kerja sistem temu balik informasi modern, mulai dari representasi teks, perhitungan bobot istilah, pemodelan similarity, hingga penyajian hasil kepada pengguna. Proyek ini juga menunjukkan peluang pengembangan lebih lanjut seperti menambahkan stemming Bahasa Indonesia, query expansion, atau integrasi embedding modern (misalnya Word2Vec atau BERT) sebagai model lanjutan di masa depan.