UJIAN TENGAH SEMESTER

SISTEM TEMU KEMBALI INFORMASI

Disusun Oleh:

Sabrina Aska Amalina (A11.2023.15264)

TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS DIAN NUSWANTORO

DAFTAR ISI

BAB I PENDAHULUAN 3
1.1. Pendahuluan 3
1.2. Tujuan 3
1.3. Ruang Lingkup 4
1.4. Ruang Lingkup 4
1.5. Kontribusi Proyek terhadap Sub-CPMK 4
BAB II DATA DAN *PREPROCESSING* 5
2.1. Data 5
2.2. Tahapan *Preprocessing* 5
2.3. Contoh Hasil *Preprocessing* 5
BAB III Metode *Information Retrieval* (IR) 6
3.1. *Boolean Retrieval* Model 6
3.2. *Vector Space* Model (VSM) 6
3.3. Formula Pembobotan: TF, IDF, TF-IDF 6
3.4. *Cosine Similarity* 6
BAB IV Arsitektur, Eksperimen, dan Evaluasi 7
4.1. Arsitektur *Search Engine* 7
4.2. Eksperimen dan Evaluasi 7
BAB V PENUTUPAN 10
5.1. Diskusi (Kelebihan, Keterbatasan, dan Saran) 10
5.2. Kesimpulan (Capaian Sub-CPMK) 10

BAB I
PENDAHULUAN

1.1. Pendahuluan

Sistem Temu Kembali Informasi (STKI) merupakan cabang ilmu komputer yang berfokus pada pencarian dan pengambilan informasi relevan dari sekumpulan data atau dokumen berdasarkan permintaan (query) pengguna (Vika Ummu Hani, 2025). Dalam kehidupan sehari-hari, konsep STKI digunakan pada mesin pencari, sistem rekomendasi, dan berbagai aplikasi pencarian dokumen. Berbeda dari database retrieval yang mengandalkan kecocokan eksak (exact match), STKI bekerja lebih fleksibel dengan memanfaatkan kecocokan tingkat kemiripan (similarity matching), pembobotan istilah (term weighting), dan perankingan dokumen. Melalui pendekatan ini, sistem dapat menemukan dokumen yang paling mendekati kebutuhan pengguna meskipun tidak ada kecocokan kata yang identik (Applications, 2024).

Indexing adalah proses pembuatan struktur indeks yang memungkinkan sistem menemukan istilah dalam dokumen tanpa harus membaca seluruh isi dokumen, sehingga pencarian dapat dilakukan dengan jauh lebih cepat dan efisien (Skodev, 2024). Setelah dokumen yang memuat istilah kata kunci ditemukan melalui indeks, sistem melakukan ranking dengan menghitung skor relevansi menggunakan metode seperti TF–IDF, cosine similarity, atau BM25. Skor inilah yang digunakan untuk mengurutkan dokumen dari yang paling relevan hingga yang paling tidak relevan, sehingga hasil pencarian sesuai dengan kebutuhan pengguna.

1.2. Tujuan

Tujuan dari pengembangan mini Information Retrieval System ini adalah membangun sistem pencarian dokumen berbasis abstrak publikasi techno.com agar pengguna dapat menemukan informasi yang relevan dengan lebih efisien. Secara ringkas, proyek ini bertujuan untuk:

1. Membuat *pipeline preprocessing* Bahasa Indonesia (*cleaning*, tokenisasi, *stopword removal*, *stemming* Sastrawi) agar dokumen techno.com lebih seragam dan siap diproses.
2. Mengembangkan model *Boolean Retrieval* untuk pencarian berbasis operator AND, OR, dan NOT.
3. Menerapkan *Vector Space Model* (VSM) dengan TF–IDF untuk menghasilkan *ranking* dokumen berdasarkan kemiripan dengan *query*.

4. Mengimplementasikan skema pembobotan lanjutan seperti TF–IDF sublinear dan BM25 untuk membandingkan efektivitas model.
5. Membangun *mini search engine* yang mendukung *Boolean*, VSM, dan BM25.
6. Melakukan evaluasi sistem menggunakan *gold set* techno.com dengan metrik Precision@k, Recall@k, F1-score, MAP@k, dan nDCG@k.

1.3. Ruang Lingkup

Ruang lingkup pengembangan sistem ini dibatasi pada pembangunan *mini Information Retrieval System* untuk dokumen publikasi UDINUS dari techno.com. Sistem hanya beroperasi pada korpus kecil (5–15 dokumen) dalam format teks dan berfokus pada proses inti IR, yaitu:

- preprocessing teks Bahasa Indonesia (*cleaning*, tokenisasi, *stopword removal*, *stemming*),
- pembuatan struktur indeks (*incidence matrix* dan *inverted index*),
- penerapan model *Boolean Retrieval* dan *Vector Space Model* (VSM) dengan TF–IDF,
- implementasi pembobotan TF–IDF *sublinear* dan BM25,
- pengembangan *search engine* sederhana (CLI dan Streamlit),
- evaluasi sistem menggunakan gold set terbatas melalui metrik Precision@k, Recall@k, MAP@k, dan nDCG@k.

1.4. Ruang Lingkup

- Dataset berupa 20 dokumen publikasi techno.com.
- Fokus pada preprocessing teks, indexing, vectorization, ranking, dan evaluasi.

1.5. Kontribusi Proyek terhadap Sub-CPMK

Tabel 1 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
DATA DAN *PREPROCESSING*

2.1. Data

Data yang digunakan dalam mini proyek Sistem Temu Kembali Informasi ini merupakan kumpulan dokumen publikasi UDINUS yaitu techno.com berisi abstrak yangdisimpan dalam format .txt dengan jumlah 20 teks. Dokumen ditempatkan dalam dua direktori utama yaitu :

- data/ berisi dokumen mentah publikasi techno.com
- data/processed/ berisi dokumen hasil preprocessing

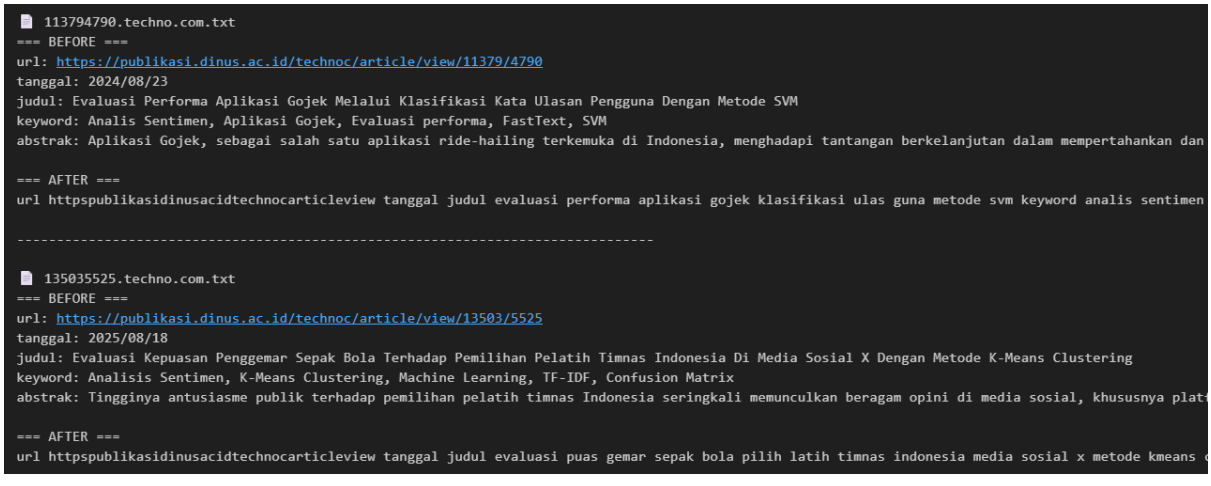
2.2. Tahapan *Preprocessing*

Preprocessing dilakukan menggunakan *script* preprocess.py yang yang terdiri dari empat tahap untuk menyiapkan dokumen sebelum dilakukan indexing dan perhitungan TF–IDF:

1. *Cleaning*: mengubah teks ke *lowercase*, menghapus angka, tanda baca, dan merapikan spasi.
2. Tokenisasi: memecah teks menjadi kata menggunakan *word_tokenize.*
3. *Stopword Removal*: menghapus kata umum Bahasa Indonesia menggunakan daftar *stopword* NLTK.
4. *Stemming*: mengubah kata menjadi bentuk dasar menggunakan stemmer Sastrawi.

2.3. Contoh Hasil *Preprocessing*

Adapun contoh hasil *preprocessing* dapat dilihat pada Gambar 1 berikut:



Gambar 1 Hasil *Preprocessing*

BAB III
Metode *Information Retrieval* (IR)

3.1. *Boolean Retrieval* Model

Boolean Retrieval adalah model pencarian yang menggunakan operasi logika AND, OR, dan NOT untuk menentukan relevansi dokumen terhadap *query*, misalnya pada ekspresi "(data AND sistem) OR (informatika NOT jaringan)". Operasi AND mengambil dokumen yang mengandung semua *term*, OR mengambil dokumen yang memuat salah satu *term*, dan NOT mengambil dokumen yang tidak mengandung *term* tertentu. Kelebihan model ini adalah mudah dan cepat, tetapi dia tidak dapat mengukur tingkat kemiripan dokumen.

3.2. *Vector Space* Model (VSM)

VSM mengukur tingkat kemiripan (*similarity*) antara dokumen dan *query* berdasarkan bobot *term*, biasanya TF-IDF. Nilai $w_{t,i}$adalah bobot TF-IDF suatu term dalam dokumen.

$$d_i = (w_{1,i}, w_{2,i}, \dots, w_{|V|,i})$$

3.3. Formula Pembobotan: TF, IDF, TF-IDF

1. *Term Frequency* (TF), mengukur seberapa sering sebuah term muncul dalam dokumen.

$$TF_{t,d} = \frac{f_{t,d}}{\max(f_{k,d})}$$

2. *Document Frequency* (DF), jumlah dokumen yang mengandung term t:

$$DF_t = |\{d : t \in d\}|$$

3. *Inverse Document Frequency* (IDF), IDF mengurangi bobot term yang sangat umum.

$$IDF_t = \log\left(\frac{N}{DF_t}\right) + 1$$

4. TF-IDF, bobot ini digunakan untuk membentuk vektor dokumen dan query dalam VSM.

$$TFIDF_{t,d} = TF_{t,d} \times IDF_t$$

3.4. *Cosine Similarity*

Cosine similarity mengukur sudut antara vektor dokumen dan *query*:

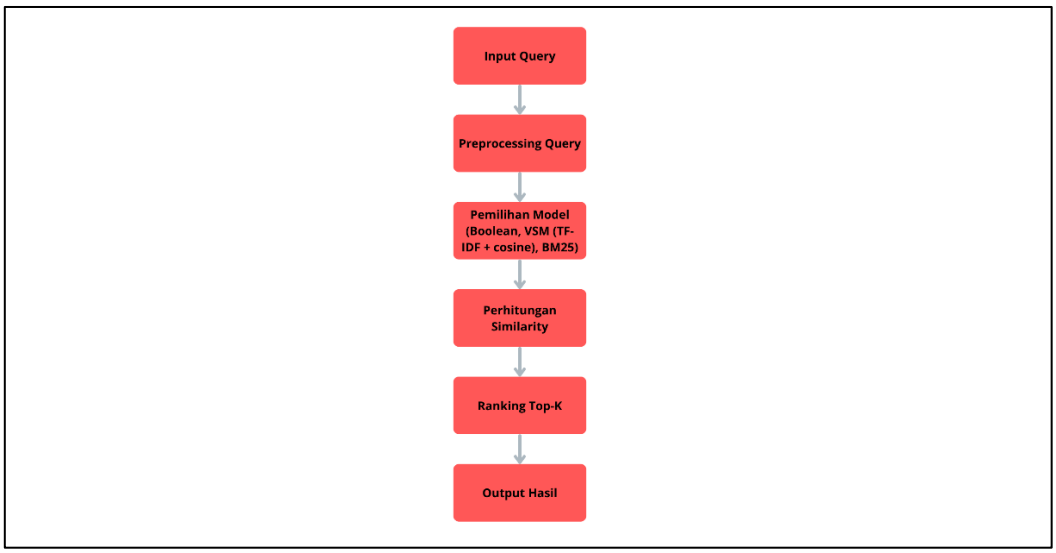
$$cosine(q,d) = \frac{\sum_t (w_{t,q} \cdot w_{t,d})}{\sqrt{\sum_t w_{t,q}^2} \cdot \sqrt{\sum_t w_{t,q}^2}}$$

Nilai berada pada rentang 0 ≤ cosine(q,d) ≤ 1, dengan 1 sangat mirip dan 0 tidak mirip.

BAB IV
Arsitektur, Eksperimen, dan Evaluasi

4.1. Arsitektur *Search Engine*

Diagram alir *search engine* dari sistem adalah sebagai berikut:



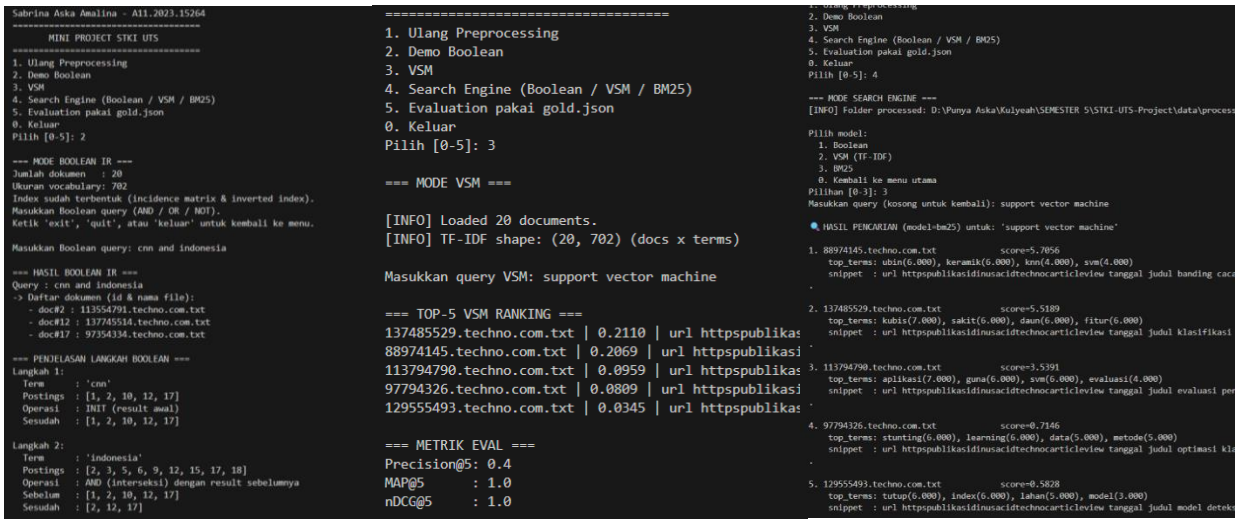
Gambar 2 Diagram Alir *Search Engine*

4.2. Eksperimen dan Evaluasi

Eksperimen meliputi pengujian preprocessing, Boolean Retrieval, Vector Space Model (VSM), dan perbandingan skema pembobotan (TF-IDF, Sublinear TF-IDF, dan BM25).

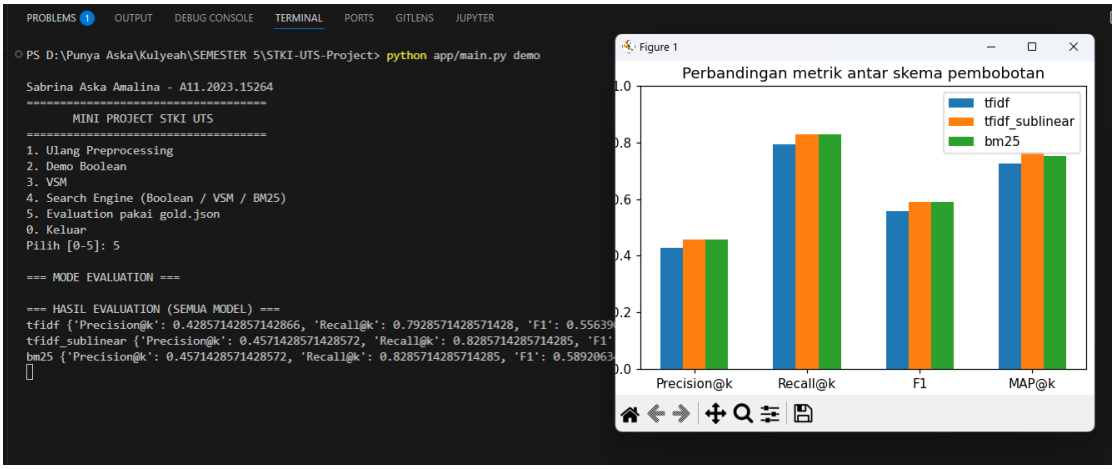
1. Eksperimen CLI

   Model diuji dengan menjalankan perintah `python app/main.py demo`



Gambar 3 Eksperimen CLI

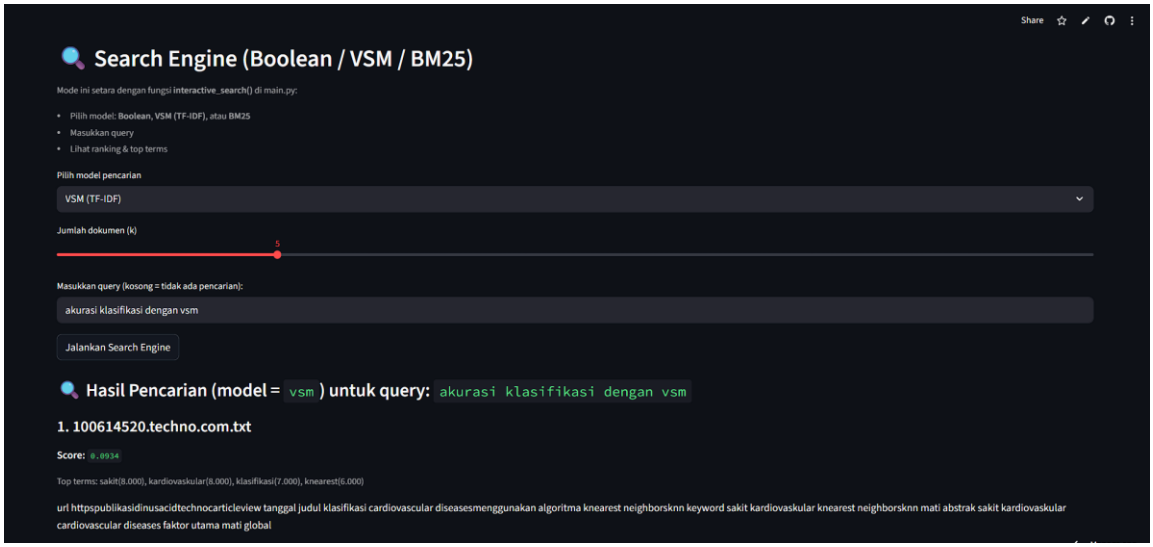
2. Hasil Evaluasi Metrik (gold.json)



Gambar 4 Hasil Evaluasi Metrik

3. Hasil Pengujian Aplikasi Web (Streamlit Deployment)

Link: https://sabrinaskaa-ir-uts.streamlit.app/

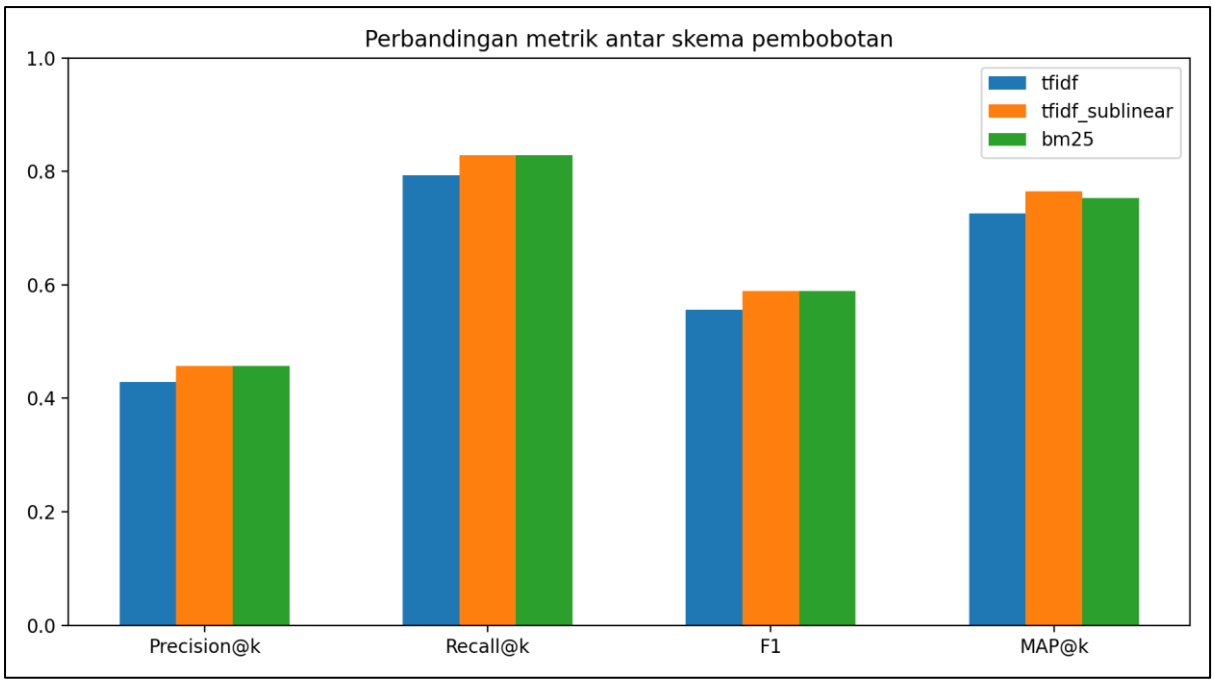


Gambar 5 *Search Engine*

4. Diagram Metrik Evaluasi

Pada tahap ini dilakukan pengukuran performa sistem menggunakan metrik:

- Precision – ketepatan dokumen relevan.
- Recall – kelengkapan dokumen relevan yang ditemukan.
- F1-score – keseimbangan precision & recall.
- Precision@k – ketepatan pada k dokumen teratas.
- MAP@k – rata-rata presisi kumulatif.
- nDCG@k – kualitas urutan ranking.



Gambar 6 Diagram Perbandingan Metriks

5. Hasil Evaluasi

Tabel 2 Hasil Metriks

Model	Precision@k	Recall@k	F1	MAP@k
TF-IDF	0.4286	0.7929	0.5564	0.7264
TF-IDF Sublinear	0.4571	0.8286	0.5892	0.7655
BM25	0.4571	0.8286	0.5892	0.7530

Adapun analisis hasil yakni sebagai berikut:

- TF-IDF standar menunjukkan kinerja terendah karena tidak menormalkan frekuensi kata dan tidak mempertimbangkan panjang dokumen, sehingga banyak dokumen relevan gagal muncul di posisi atas.
- TF-IDF Sublinear memberikan hasil terbaik di semua metrik (Precision, Recall, F1, MAP). Log-scaling membuat bobot kata lebih seimbang, sehingga model lebih efektif mengurutkan dokumen relevan di posisi tinggi.
- BM25 memiliki performa sangat baik dan hampir setara dengan Sublinear, terutama pada Precision dan Recall. Sedikit tertinggal pada MAP karena ranking-nya sedikit kurang stabil dibanding Sublinear.
- Secara keseluruhan, model yang menerapkan normalisasi frekuensi kata dan panjang dokumen (Sublinear dan BM25) unggul jauh dibanding TF-IDF standar.

BAB V
PENUTUPAN

5.1. Diskusi (Kelebihan, Keterbatasan, dan Saran)

Sistem temu kembali informasi yang dibangun memiliki beberapa kelebihan utama. Pipeline preprocessing sudah lengkap dan efektif, sehingga dokumen lebih seragam untuk proses indexing. Sistem juga mendukung dua pendekatan pencarian, yaitu Boolean Retrieval dan VSM dengan berbagai skema pembobotan (TF-IDF, Sublinear TF-IDF, dan BM25), sehingga memungkinkan perbandingan performa yang jelas. Evaluasi menggunakan metrik standar IR membuat kualitas model dapat diukur secara objektif, dan hasilnya menunjukkan bahwa Sublinear TF-IDF dan BM25 mampu memberikan relevansi dan ranking dokumen yang lebih baik.

Adapun keterbatasan sistem ini antara lain ukuran dataset yang kecil, belum adanya pencarian frasa (phrase query), serta belum mendukung pemahaman konteks atau semantic search. Tampilan antarmuka Streamlit juga masih minimalis dan dapat dikembangkan lebih informatif. Untuk pengembangan selanjutnya, sistem dapat ditingkatkan dengan menambah korpus yang lebih besar, membangun positional index, menerapkan word embeddings atau model semantic retrieval, serta memperkaya fitur antarmuka seperti highlight kata kunci dan tampilan detail dokumen.

5.2. Kesimpulan (Capaian Sub-CPMK)

1. Sub-CPMK 10.1.1 – Preprocessing Dokumen
    - Berhasil melakukan cleaning, tokenisasi, stopword removal, dan stemming.
    - Menghasilkan korpus yang seragam dan siap untuk proses indexing.
2. Sub-CPMK 10.1.2 – Boolean Retrieval
    - Membangun incidence matrix dan inverted index.
    - Mendukung pencarian dengan AND, OR, NOT beserta hasil dan eksekusinya.
3. Sub-CPMK 10.1.3 – VSM TF-IDF & Cosine Similarity
    - Menghitung bobot TF-IDF dan mengubah dokumen menjadi vektor.
    - Menghasilkan ranking dokumen berdasarkan cosine similarity.
4. Sub-CPMK 10.1.4 – Evaluasi Model IR
    - Melakukan evaluasi dengan Precision@k, Recall@k, F1, MAP@k, dan nDCG.
    - Membandingkan performa TF-IDF, Sublinear TF-IDF, dan BM25.