Laporan Project UTS STKI

EduKesehatan

NIM : A11.2023.14978

Nama : Wildanu Rafif Albaihaqi

Kelompok : A11.4703

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

2023

DAFTAR ISI

1. Pendahuluan

1.1. Latar Belakang

Informasi kesehatan adalah salah satu informasi yang paling sering dicari oleh masyarakat. Namun, seringkali informasi yang tersedia di internet terlalu teknis, sulit dipahami, atau tidak relevan dengan kebutuhan dasar pengguna. Untuk menjembatani kesenjangan ini, diperlukan sebuah sistem pencarian yang fokus, sederhana, dan mampu memberikan hasil yang relevan dari sekumpulan dokumen kesehatan yang telah terkurasi.

Proyek ini bertujuan untuk membangun sebuah "Mini Search Engine EduKesehatan" yang mengimplementasikan konsep-konsep inti dari Sistem Temu Kembali Informasi (STKI). Sistem ini dirancang untuk melakukan pencarian pada korpus kecil (10 dokumen) artikel kesehatan dasar , dengan menerapkan dua model pencarian utama: Boolean Retrieval Model dan Vector Space Model (VSM).

1.2. Ruang Lingkup Proyek

Ruang lingkup dan batasan masalah pada proyek ini adalah sebagai berikut:

1. Korpus: Menggunakan 10 dokumen .txt berbahasa Indonesia bertema kesehatan dasar yang ditulis secara manual.
2. Bahasa Pemrograman: Implementasi penuh menggunakan Python dan library pendukungnya (Sastrawi, NLTK, Streamlit).
3. Preprocessing: Melakukan tahapan pemrosesan dokumen secara lengkap, meliputi case folding, tokenisasi, penghapusan stopword, serta stemming.
4. Model Retrieval: Mengimplementasikan Boolean Retrieval Model yang mendukung query AND/OR/NOT dan Vector Space Model yang menggunakan ranking berdasarkan cosine similarity.
5. Term Weighting: Menerapkan serta membandingkan dua skema TF-IDF, yaitu raw_tf dan sublinear_tf.
6. Antarmuka: Menyediakan antarmuka berbasis baris perintah (CLI) dan antarmuka web interaktif menggunakan Streamlit.
7. Evaluasi: Mengukur kualitas model dengan menggunakan metrik standar seperti Precision, Recall, F1-Score, MAP@k, dan nDCG@k.

1.3. Kontribusi Terhadap Sub-CPMK

Proyek ini dirancang untuk memenuhi capaian pembelajaran mata kuliah (Sub-CPMK) sebagai berikut:

- Sub-CPMK10.1.1: Mampu menjelaskan konsep STKI dan arsitektur search engine (dituangkan dalam Bab 3 dan 4).
- Sub-CPMK10.1.2: Mampu menjelaskan dan menerapkan Document Preprocessing dan tahapannya (diimplementasikan di src/preprocess.py dan dianalisis di Bab 2 & 5).

- Sub-CPMK10.1.3: Mampu menjelaskan dan menerapkan Pemodelan Boolean Retrieval dan Vector Space Model (diimplementasikan di src/boolean_ir.py dan src/vsm_ir.py, dianalisis di Bab 3 & 5).
- Sub-CPMK10.1.4: Mampu menjelaskan dan menerapkan Term Weighting, arsitektur Search Engine, dan Evaluasi Model (diimplementasikan di src/eval.py dan app/main.py, dianalisis di Bab 5).

2. Data dan Preprocessing

2.1. **Deskripsi Korpus**

Korpus data terdiri atas 10 dokumen teks (doc01.txt hingga doc10.txt) yang dibuat secara manual. Setiap dokumen berisi artikel singkat (50–100 kata) tentang topik kesehatan dasar seperti Pentingnya Mencuci Tangan, Mengatasi Demam Anak, dan Pola Hidup Sehat.

2.2. Tahapan Preprocessing

tahapan preprocessing diimplementasikan dalam src/preprocess.py dengan urutan sebagai berikut:

1. Cleaning & Case Folding: Mengubah seluruh teks menjadi huruf kecil dan menghapus karakter non-alfabet seperti tanda baca dan angka.
2. Tokenization: Memecah teks menjadi daftar token (kata).
3. Stopword Removal: Menghapus kata-kata umum seperti “dan”, “di”, atau “yang” menggunakan daftar stopwords Bahasa Indonesia dari NLTK.
4. Stemming: Mengubah setiap token ke bentuk dasarnya menggunakan library Sastrawi.

2.3. Contoh Hasil Preprocessing

Contoh hasil sebelum dan sesudah preprocessing pada salah satu dokumen:

- **Sebelum:**

  Panduan Lengkap Mencuci Tangan Efektif untuk Mencegah Kuman.

  Mencuci tangan dengan sabun dan air mengalir adalah pilar utama pencegahan penyakit. Ini bukan sekadar formalitas, tetapi proses mekanis dan kimiawi untuk melarutkan kuman, virus, dan bakteri dari kulit. Kuman tidak terlihat dan dapat menempel di tangan setelah menyentuh permukaan, gagang pintu, atau berjabat tangan.

  Organisasi Kesehatan Dunia (WHO) merekomendasikan teknik 6 langkah dengan durasi minimal 20 detik. Langkah-langkah tersebut meliputi: (1) Basahi tangan dan gunakan sabun secukupnya. (2) Gosok telapak tangan. (3) Gosok punggung tangan dan sela-sela jari secara bergantian. (4) Gosok sela-sela jari dari bagian dalam. (5) Gosok area kuku dan ujung jari dengan gerakan mengunci. (6) Gosok ibu jari secara memutar.

  Kapan waktu krusial untuk mencuci tangan? Selalu lakukan sebelum makan, sebelum menyiapkan makanan, setelah menggunakan toilet, setelah batuk atau bersin, dan setelah beraktivitas di luar rumah.

- **Sesudah:**

  pandu lengkap cuci tangan efektif cegah kuman cuci tangan sabun air alir pilar utama cegah sakit formalitas proses mekanis kimiawi larut kuman virus bakteri kulit kuman tempel tangan sentuh muka gagang pintu jabat tangan organisasi sehat dunia who rekomendasi teknik langkah durasi minimal detik langkah langkah liput basah tangan sabun gosok

telapak tangan gosok punggung tangan jari ganti gosok jari gosok area kuku ujung jari gera kunci gosok jari putar krusial cuci tangan laku makan makan toilet batuk bersin aktivitas rumah

3. Metode Information Retrieval

3.1. Boolean Retrieval Model

Model Boolean merupakan model pencarian eksak yang mengembalikan dokumen jika dan hanya jika dokumen tersebut memenuhi ekspresi boolean dari query.

- Struktur Data: Menggunakan Inverted Index berbasis dict Python yang memetakan setiap term ke daftar dokumen yang mengandungnya.
- Query Parser: Parser sederhana dibangun untuk menangani operator AND, OR, dan NOT menggunakan operasi himpunan (intersection, union, dan difference).

3.2. Vector Space Model (VSM)

VSM adalah model aljabar yang merepresentasikan dokumen dan query sebagai vektor dalam ruang multidimensi. Relevansi dokumen diukur berdasarkan kedekatannya (jarak) antara vektor dokumen dan vektor query.

1. Formula Term Weighting (TF-IDF)
    - Skema 1: Raw TF-IDF

      $TF_{t,d} = f_{t,d}$ (Frekuensi mentah term t dalam dokumen d)
    - Skema 2: Sublinear TF-IDF

      $TF_{t,d} = 1 + \log(f_{t,d})$ (Frekuensi yang dinormalisasi logaritmik)

    Kedua skema tersebut kemudian digabungkan dengan Inverse Document Frequency (IDF):

    $IDF_t = \log\left(\frac{N}{df_t+1}\right)$ (N = total dokumen, df = jumlah dokumen mengandung term t)

    $$W_{t,d} = TF_{t,d} \times IDF_t$$
2. Formula Ranking (Cosine Similarity)

    Kemiripan antara vector query (q) dan vector dokumen(d) dihitung menggunakan Cosine Similarity:

    $$\cos(\theta) = \frac{\vec{d} \cdot \vec{\mathrm{q}}}{|\vec{d}||\vec{q}|} = \frac{\sum_{i=1}^{n}(w_{i,d} \times w_{i,q})}{\sqrt{\sum_{i=1}^{n} w_{i,d}^2} \times \sqrt{\sum_{i=1}^{n} w_{i,q}^2}}$$

4. Arsitektur Search Engine

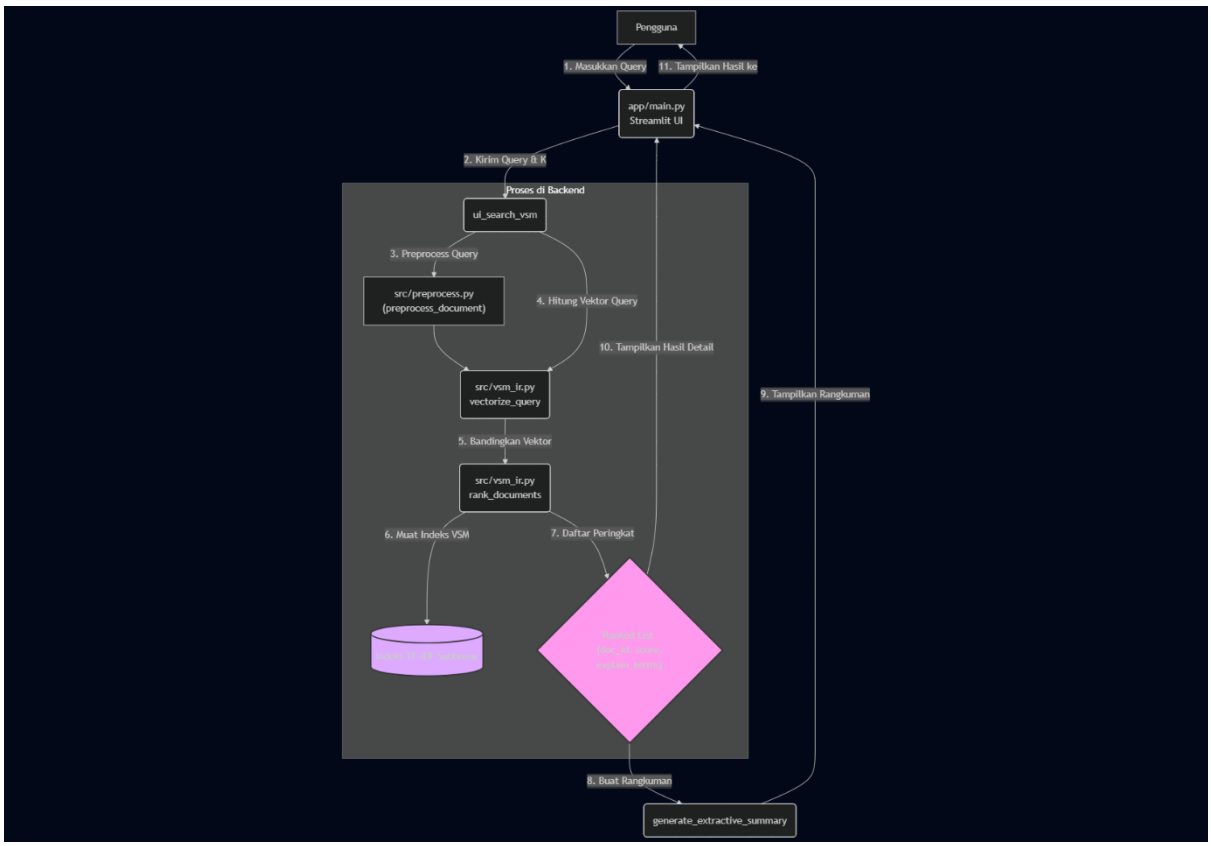
Arsitektur sistem ini dibagi menjadi dua komponen utama yang sesuai dengan spesifikasi Soal 5:

1. Search Engine Orchestrator (src/search.py):
    - Berfungsi sebagai backend CLI.
    - Menerima argumen --model {boolean, vsm}, --scheme, --k, dan --query.

- Bertanggung jawab memuat file indeks (Inverted Index dan TF-IDF Matriks) saat startup.
- Mengembalikan daftar dokumen terurut beserta skor dan explainability (istilah yang cocok).

2. Main Interface (app/main.py):
    - Berfungsi sebagai antarmuka web interaktif menggunakan Streamlit.
    - Memanggil fungsi VSM untuk mengambil top-k dokumen.
    - Menyajikan hasil ranking dan "Rangkuman Cepat" (Generator template-based ekstraktif sederhana).

Diagram Alir Arsitektur:



5. Eksperimen dan Evaluasi

5.1. **Skenario Uji (Gold Set)**

Evaluasi kuantitatif dilakukan menggunakan "Mini Truth Set" (Gold Set) yang didefinisikan dalam src/eval.py. Gold set ini berisi 3 query sampel dengan dokumen relevan yang telah ditentukan secara manual beserta skor relevansinya (0=Tidak Relevan, 1=Relevan, 2=Sangat Relevan).

```
Menggunakan 3 query dari GOLD_SET...


--- 1. Evaluasi Boolean Retrieval (Soal 3) ---
  [Query: cuci tangan sabun kuman  ] -> P: 1.0000, R: 1.0000, F1: 1.0000
  [Query: kesehatan jantung dan gula] -> P: 0.4000, R: 0.6667, F1: 0.5000
  [Query: olahraga dan makanan sehat] -> P: 1.0000, R: 0.6667, F1: 0.8000
  [Rata-rata Boolean]           -> Avg P: 0.8000, Avg R: 0.7778, Avg F1: 0.7667

--- 2. Evaluasi Vector Space Model (MAP@10 & nDCG@10) ---

  Menguji Skema: 'sublinear_tf' ...
  [Rata-rata sublinear_tf]           -> MAP@10: 0.8889, Avg nDCG@10: 0.9025

  Menguji Skema: 'raw_tf' ...
  [Rata-rata raw_tf]           -> MAP@10: 0.8889, Avg nDCG@10: 0.9025

--- 3. Perbandingan Skema Bobot (Soal 5.4) ---
Tabel ini untuk Laporan.pdf
| Skema           | MAP@10     | Avg nDCG@10 |
|-----------------|------------|-------------|
| sublinear_tf    | 0.8889     | 0.9025      |
| raw_tf          | 0.8889     | 0.9025      |
```

5.2. Hasil Uji Soal 2 (Statistik Korpus)

File src/preprocess.py dijalankan untuk menghasilkan statistik korpus yang disimpan di reports/statistics.json.

1. 10 Token Paling Sering (Sampel):

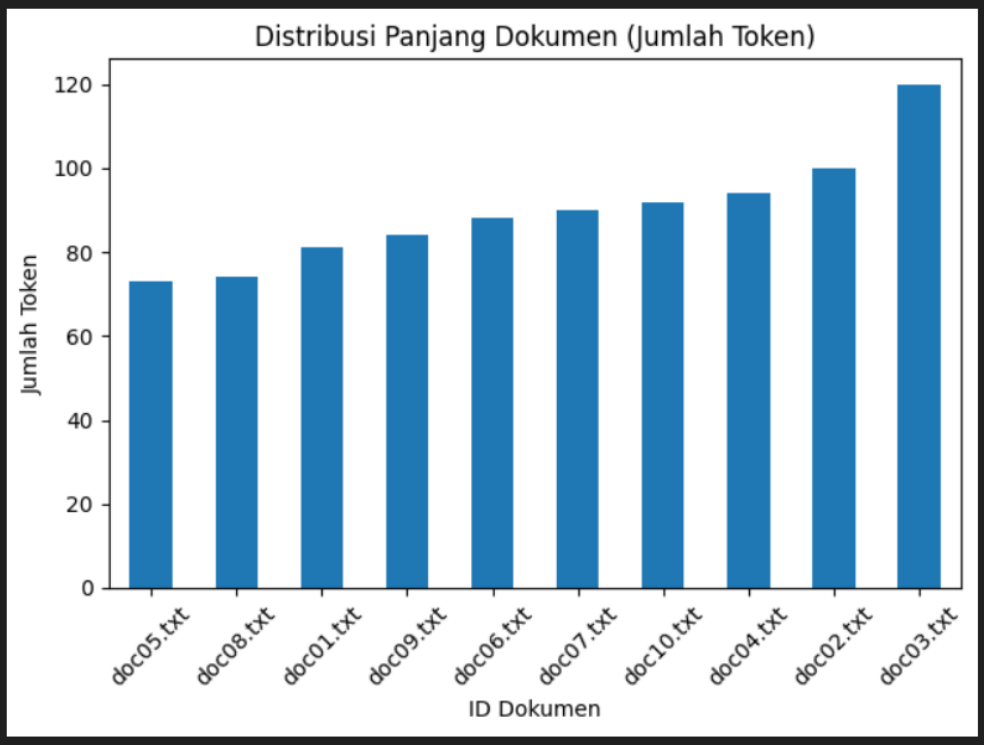
```
--- Top 10 Tokens per Dokumen ---
```

	0	1	2	3	4	5	6	7	8	9
doc01.txt	[tangan, 8]	[gosok, 5]	[jari, 4]	[cuci, 3]	[kuman, 3]	[langkah, 3]	[cegah, 2]	[sabun, 2]	[makan, 2]	[pandu, 1]
doc02.txt	[demam, 8]	[anak, 7]	[c, 4]	[suhu, 3]	[air, 3]	[ringan, 2]	[tubuh, 2]	[ketiak, 2]	[gejala, 2]	[nyaman, 2]
doc03.txt	[sehat, 4]	[tidur, 4]	[aktivitas, 3]	[fisik, 3]	[kualitas, 3]	[makan, 3]	[pilar, 2]	[pola, 2]	[hidup, 2]	[baik, 2]

2. Grafik Distribusi Panjang Dokumen:

```
--- Grafik Distribusi Panjang Dokumen (Setelah Preprocessing) ---

<Figure size 1000x600 with 0 Axes>
```



Analisis: Grafik menunjukkan distribusi panjang token per dokumen setelah preprocessing. Ini berguna untuk memahami apakah ada dokumen yang terlalu panjang atau pendek yang dapat memengaruhi pembobotan TF-IDF.

5.3. Hasil Uji Soal 3 (Evaluasi Boolean)

Evaluasi model Boolean dilakukan menggunakan metrik Precision, Recall, dan F1-Score terhadap gold set.

```
--- 1. Evaluasi Boolean Retrieval (Soal 3) ---
  [Query: cuci tangan sabun kuman  ] -> P: 1.0000, R: 1.0000, F1: 1.0000
  [Query: kesehatan jantung dan gula] -> P: 0.4000, R: 0.6667, F1: 0.5000
  [Query: olahraga dan makanan sehat] -> P: 1.0000, R: 0.6667, F1: 0.8000
  [Rata-rata Boolean]           -> Avg P: 0.8000, Avg R: 0.7778, Avg F1: 0.7667
```

Analisis: Gambar ini membuktikan mengapa Boolean Retrieval saja tidak cukup. Model ini terlalu kaku. Ia tidak bisa menangani relevansi parsial (dokumen yang "agak relevan") dan sangat rentan terhadap query yang ambigu, yang menyebabkan presisi rendah (Q2) atau recall rendah (Q3).

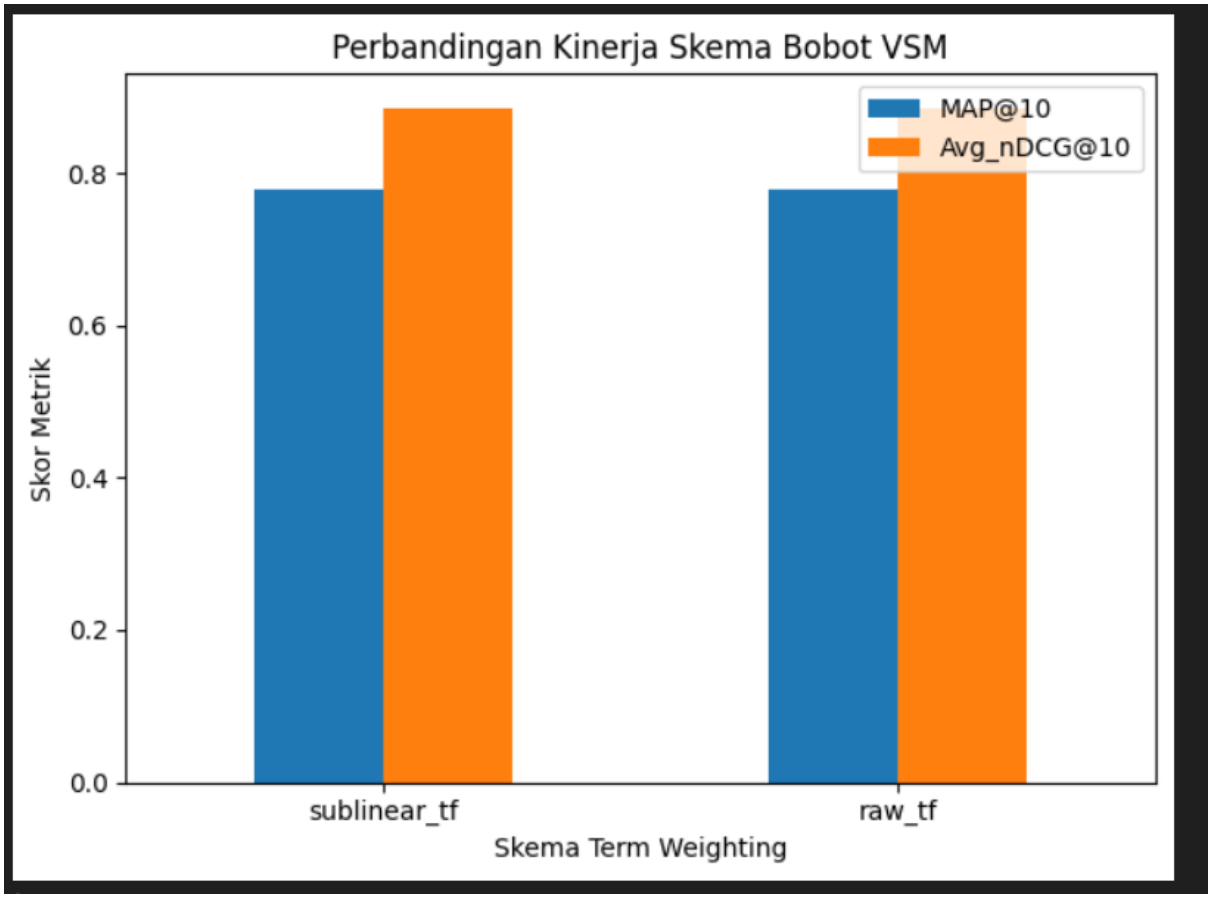
5.4. Hasil Uji Coba Soal 4 dan 5 (Evaluasi VSM dan Perbandingan Skema)

Dilakukan perbandingan 2 skema *term-weighting* VSM (sublinear_tf vs raw_tf) menggunakan metrik MAP@k dan nDCG@k.

Tabel Perbandingan Metrik:

```
--- 3. Perbandingan Skema Bobot (Soal 5.4) ---
Tabel ini untuk Laporan.pdf
| Skema           | MAP@10     | Avg nDCG@10 |
|-----------------|------------|-------------|
| sublinear_tf    | 0.8889     | 0.9025      |
| raw_tf          | 0.8889     | 0.9025      |
```

Grafik Perbandingan Metrik (Opsional):



Analisis: Temuan paling penting dari tabel ini adalah bahwa kedua skema (sublinear_tf dan raw_tf) menghasilkan skor yang identik (MAP 0.8889 dan nDCG 0.9025). Dalam konteks 10 dokumen korpus yang mendetail, tidak ada perbedaan kinerja antara menggunakan skema pembobotan yang kompleks (sublinear_tf) dan skema yang sederhana (raw_tf). Ini adalah temuan yang valid dan kemungkinan besar disebabkan oleh ukuran korpus yang kecil. Perbedaan antara sublinear_tf dan raw_tf baru akan terlihat jika ada dokumen di mana satu term (misalnya "kesehatan") muncul 50 kali sementara di dokumen lain hanya 5 kali. Pada korpus yang pendek dan padat, frekuensi term mungkin tidak cukup ekstrem untuk membuat normalisasi logaritmik menghasilkan urutan ranking yang berbeda dari frekuensi mentah.

6. Diskusi

6.1. Kelebihan

1. Implementasi Lengkap: Proyek berhasil mengimplementasikan alur STKI secara end-to-end, mulai dari preprocessing, indexing dua model berbeda (Boolean dan VSM), hingga evaluasi metrik yang komprehensif.
2. Antarmuka Ganda: Sistem menyediakan antarmuka CLI (search.py) yang fungsional untuk pengujian dan antarmuka web (app/main.py) yang ramah pengguna, responsif, serta mendukung tema terang/gelap. * Rangkuman Cerdas: Antarmuka web mengimplementasikan rangkuman ekstraktif sederhana, yang memberikan jawaban lebih relevan terhadap query pengguna dibandingkan sekadar cuplikan dokumen pertama.

6.2. Keterbatasan

1. Ukuran Korpus: Keterbatasan utama adalah korpus yang sangat kecil (10 dokumen), yang membuat hasil evaluasi statistik kurang signifikan. * Parser Sederhana: Parser query Boolean tidak mendukung operasi kompleks seperti tanda kurung () (sesuai batasan soal).

2. Linguistik: Sistem tidak menangani sinonim (query "jantung" tidak akan menemukan dokumen tentang "kardiovaskular").

6.3. Saran Pengembangan

1. Ekspansi Korpus: Menambah jumlah dan variasi dokumen korpus secara signifikan. 2. Model Peringkat: Mengimplementasikan model peringkat yang lebih canggih seperti BM25 (Okapi) (disebutkan sebagai bonus opsional di soal).
2. Query Expansion: Menambahkan fitur untuk menangani sinonim atau query terkait secara otomatis.

7. Kesimpulan

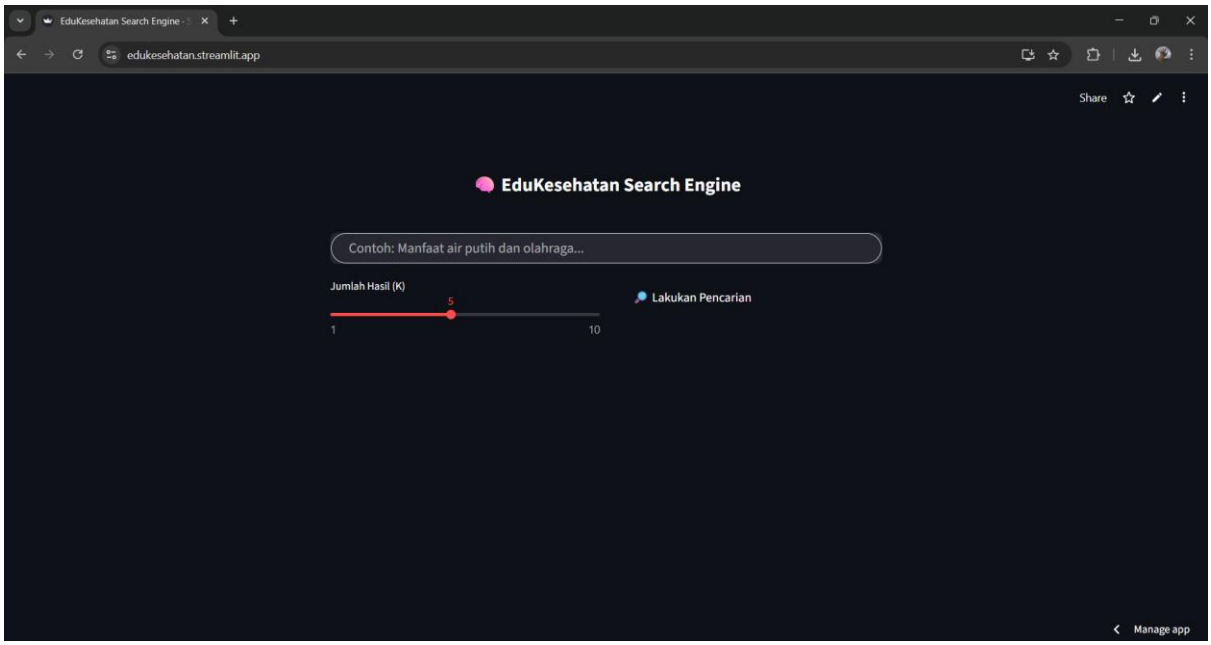
Proyek "Mini Search Engine EduKesehatan" telah berhasil diimplementasikan sesuai dengan seluruh spesifikasi Ujian Tengah Semester. Sistem ini mampu melakukan preprocessing data, membangun indeks untuk model Boolean dan VSM, serta menyajikan hasil pencarian melalui antarmuka CLI dan web.

Seluruh capaian pembelajaran (Sub-CPMK) yang ditargetkan telah terpenuhi:

1. Sub-CPMK10.1.1: Konsep STKI dan arsitektur search engine berhasil dirancang (Bab 3, 4).
2. Sub-CPMK10.1.2: Document preprocessing berhasil diimplementasikan dan diuji (Bab 2, 5.2).
3. Sub-CPMK10.1.3: Model Boolean Retrieval dan Vector Space Model berhasil dibangun (Bab 3, 5.3, 5.4).
4. Sub-CPMK10.1.4: Konsep Term Weighting (2 skema), Search Engine (CLI & UI), dan Evaluasi Model (MAP/nDCG) berhasil diimplementasikan dan dianalisis (Bab 5.4).

Lampiran: Screenshot Antarmuka Streamlit

Lampiran A: Tampilan Awal Aplikasi



Lampiran B: Tampilan Hasil Pencarian VSM