LAPORAN UJIAN TENGAH SEMESTER GANJIL 2025/2026

MATA KULIAH SISTEM TEMU KEMBALI INFORMASI (A11.4703)

Implementasi Mini Search Engine (Boolean & VSM) pada Korpus Teks UIN Jawa Tengah

Disusun oleh:

Nama: Silvan Ridho Pradana

NIM: A11.2022.14284

Kelompok: A11.4703

Dosen Pengampu:

Abu Salam, M.Kom

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS DIAN NUSWANTORO

SEMARANG

2025

BAB 1. PENDAHULUAN

1.1 Latar Belakang dan Definisi STKI

Sistem Temu Kembali Informasi (STKI) atau *Information Retrieval* (IR) adalah bidang ilmu yang berfokus pada pencarian, pengambilan, dan penyajian informasi dari koleksi data tidak terstruktur (seperti dokumen teks, halaman web, atau multimedia) yang relevan dengan kebutuhan informasi pengguna[1].

Perbedaan fundamental antara STKI dan sistem *database retrieval* (SQL) terletak pada sifat data dan kueri. Sistem database bekerja pada **data terstruktur** (tabel, skema) dengan kueri yang bersifat **presisi** dan *exact-match* (misal, SELECT * FROM mahasiswa WHERE nim='A11.12345'). Hasilnya bersifat biner (ditemukan atau tidak). Sebaliknya, STKI bekerja pada **data tidak terstruktur** (teks bebas), di mana kueri seringkali **ambigu** (misal, "info pmb uin"). Hasilnya tidak biner, melainkan diukur berdasarkan derajat **relevansi**[2].

1.2 Peran Index dan Ranking

Untuk menemukan informasi secara cepat dalam jutaan dokumen, STKI tidak melakukan pemindaian sekuensial. Inti dari kecepatan STKI adalah **Indeks (Index)**. Dalam proyek ini, kita mengimplementasikan **Inverted Index** (dijelaskan di Bab 3)[3], sebuah struktur data yang memetakan setiap kata unik (term) ke daftar dokumen yang mengandung kata tersebut.

Selain menemukan, STKI juga harus mengurutkan hasil. Inilah peran **Ranking**[4]. Model Boolean Retrieval (Soal 03) hanya membagi dokumen menjadi dua set (relevan atau tidak), sedangkan **Vector Space Model (VSM)** (Soal 04) memberikan **skor numerik** untuk setiap dokumen. Skor ini dihitung menggunakan **Cosine Similarity**[5], yang mengukur kesamaan sudut antara vektor kueri dan vektor dokumen.

1.3 Tujuan dan Ruang Lingkup Proyek

- **Tujuan:** Tujuan utama proyek ini adalah merancang dan mengimplementasikan sebuah *mini search engine* yang menerapkan konsep-konsep fundamental STKI pada korpus teks kecil.
- **Ruang Lingkup:**
    - **Korpus:** 5 dokumen .txt buatan sendiri mengenai profil UIN di Jawa Tengah.
    - **Model:** Boolean Retrieval Model dan Vector Space Model (VSM).
    - **Fitur:** Preprocessing (termasuk stemming Sastrawi), ranking Cosine Similarity, perbandingan skema bobot, dan antarmuka web interaktif menggunakan Streamlit.
    - **Bahasa:** Python 3, dengan pustaka utama NLTK, Sastrawi, Scikit-learn, dan Streamlit.

1.4 Keterkaitan Proyek dengan Capaian Pembelajaran (Sub-CPMK)

Proyek ini dirancang untuk memenuhi capaian pembelajaran mata kuliah (Sub-CPMK) sesuai RPS:

1. **Sub-CPMK10.1.1 (Konsep STKI):** Dipenuhi melalui esai pada **Bab 1** ini yang menjelaskan konsep dasar STKI, arsitektur, index, dan ranking.
2. **Sub-CPMK10.1.2 (Document Preprocessing):** Diimplementasikan secara praktis pada **Soal 02** (dijelaskan di **Bab 2**) melalui modul src/preprocess.py yang mencakup tokenisasi, case-folding, stopword removal, dan stemming.
3. **Sub-CPMK10.1.3 (Pemodelan):** Diimplementasikan pada **Soal 03** (Boolean Model) dan **Soal 04** (Vector Space Model). Kedua metode ini dijelaskan secara rinci di **Bab 3**.
4. **Sub-CPMK10.1.4 (Term Weighting & Evaluasi):** Dipenuhi pada **Soal 05** (dijelaskan di **Bab 5**) melalui eksperimen perbandingan dua skema *term weighting* (TF-IDF standar vs Sublinear) dan evaluasi model menggunakan metrik standar (Precision@k, MAP@k).

BAB 2. DATA & PREPROCESSING (SOAL 02)

2.1 Deskripsi Korpus Data

Korpus yang digunakan dalam proyek ini adalah 5 dokumen teks (.txt) yang dibuat secara manual. Konten dokumen berisi informasi ringkas mengenai profil beberapa Universitas Islam Negeri (UIN) yang berlokasi di provinsi Jawa Tengah.

Kelima dokumen tersebut adalah:

- doc1.txt: Profil UIN Walisongo Semarang.
- doc2.txt: Profil UIN Raden Mas Said Surakarta.
- doc3.txt: Profil UIN K.H. Abdurrahman Wahid Pekalongan.
- doc4.txt: Profil UIN Salatiga.
- doc5.txt: Informasi umum Penerimaan Mahasiswa Baru (PMB) UIN.

2.2 Tahapan Document Preprocessing

Preprocessing adalah langkah krusial untuk mengubah teks tidak terstruktur menjadi format yang bersih dan terstruktur untuk pengindeksan[8888]. Kami membuat modul src/preprocess.py yang mengimplementasikan tahapan berikut:

1. **Cleaning & Case Folding:** Mengubah seluruh teks menjadi huruf kecil (lower()), menghapus angka, dan menghapus seluruh tanda baca (string.punctuation).
2. **Tokenization:** Memecah teks yang sudah bersih menjadi daftar kata (tokens) menggunakan nltk.word_tokenize.
3. **Stopword Removal:** Menghapus kata-kata umum dalam Bahasa Indonesia (seperti 'dan', 'di', 'yang', 'adalah') menggunakan daftar stopwords.words('indonesian') dari NLTK.
4. **Stemming:** Mengubah setiap token ke bentuk kata dasarnya (misal: 'universitas', 'negeri', 'mahasiswa') menggunakan StemmerFactory dari pustaka Sastrawi.

2.3 Contoh Hasil Preprocessing

Sesuai permintaan soal, berikut adalah perbandingan *before* (sebelum) dan *after* (sesudah) preprocessing pada dua dokumen sampel.

[SISIPKAN SCREENSHOT TABEL/HASIL 'BEFORE/AFTER' DARI NOTEBOOK ANDA DI SINI]

(Contoh format di bawah, ganti dengan screenshot Anda)

| Doc ID | Before | After |

| :--- | :--- | :--- |

| doc1.txt | Universitas Islam Negeri Walisongo (disingkat UIN Walisongo) adalah sebuah perguruan... | universitas islam negeri walisongo uin walisongo buah tinggi negeri kota semarang provinsi jawa... |

| doc2.txt | Universitas Islam Negeri Raden Mas Said Surakarta (disingkat UIN RMS) adalah... | universitas islam negeri raden mas said surakarta uin rms tinggi islam negeri indonesia... |

2.4 Analisis Token (Uji)

Sebagai bagian dari uji, kami menganalisis frekuensi token di seluruh korpus yang telah diproses. Berikut adalah 10 token yang paling sering muncul:

[SISIPKAN SCREENSHOT TABEL 10 TOKEN TERATAS DARI NOTEBOOK ANDA DI SINI]

(Contoh format di bawah, ganti dengan screenshot Anda)

| Token | Frekuensi |

| :--- | :--- |

| uin | 9 |

| negeri | 6 |

| islam | 5 |

| salatiga | 5 |

| walisongo | 5 |

| semarang | 4 |

| jawa | 4 |

| tengah | 4 |

| iain | 4 |

| kampus | 4 |

BAB 3. METODE INFORMATION RETRIEVAL (SOAL 03 & 04)

3.1 Boolean Retrieval Model (Soal 03)

Model Boolean adalah model STKI klasik yang mengambil dokumen berdasarkan kriteria biner (benar/salah) menggunakan operator logika[11].

Konsep & Implementasi:

Kami mengimplementasikan model ini di src/boolean_ir.py. Langkah utamanya adalah membangun Inverted Index, sebuah struktur data dict Python yang memetakan setiap term ke sebuah set ID dokumen yang mengandung term tersebut.

Contoh: {'semarang': {'doc1.txt', 'doc2.txt', 'doc4.txt'}, ...}

Parser kueri sederhana kemudian diimplementasikan untuk menangani tiga operator utama:

- term1 AND term2: Melakukan operasi **irisan (intersection)** pada *set* dokumen.
- term1 OR term2: Melakukan operasi **gabungan (union)** pada *set* dokumen.
- term1 AND NOT term2: Melakukan operasi **selisih (difference)** pada *set* dokumen.

3.2 Vector Space Model (Soal 04)

VSM adalah model aljabar yang merepresentasikan dokumen dan kueri sebagai vektor dalam ruang vektor multidimensi. Setiap dimensi dalam ruang ini merepresentasikan satu *term* unik dari *vocabulary*.

Pembobotan Istilah (Term Weighting):

Bobot setiap term dalam vektor dihitung menggunakan TF-IDF (Term Frequency-Inverse Document Frequency)1313. Kami menggunakan implementasi efisien TfidfVectorizer dari scikit-learn.

1. Term Frequency (TF): $tf(t,d)$

Menghitung seberapa sering term t muncul di dokumen d.

2. **Inverse Document Frequency (IDF):**

$$idf(t,D) = \log\left(\frac{N}{|\{d \in D : t \in d\}|}\right)$$

Menghitung "keistimewaan" sebuah term. N adalah jumlah total dokumen, dan bagian penyebut adalah jumlah dokumen yang mengandung term t.

3. TF-IDF Weight: $w_{t,d} = tf(t,d) \times idf(t,D)$

Bobot akhir adalah perkalian antara TF dan IDF.

Perankingan (Ranking):

Relevansi antara kueri (q) dan dokumen (d) dihitung dengan mengukur kesamaan (kedekatan) antara vektor kueri dan vektor dokumen. Metode yang paling umum digunakan adalah Cosine Similarity14141414, yang menghitung kosinus sudut (θ) di antara dua vektor. Skor yang lebih tinggi (mendekati 1.0) berarti lebih relevan.

$$similarity(q, d) = \cos(\theta) = \frac{\vec{q} \cdot \vec{d}}{|\vec{q}| |\vec{d}|}$$

Hasil pencarian kemudian diurutkan berdasarkan skor Cosine Similarity ini dari yang tertinggi ke terendah.

BAB 4. ARSITEKTUR SEARCH ENGINE (SOAL 05)

4.1 Diagram Alir Sistem

Arsitektur *search engine* ini terbagi menjadi dua alur utama: *Indexing Pipeline* (dijalankan *offline* untuk persiapan) dan *Query Pipeline* (dijalankan *online* saat ada pencarian)[16].

1. Indexing Pipeline (Offline):

[5 Dokumen .txt] $\rightarrow$ preprocess.py (Clean, Tokenize, Stop, Stem) $\rightarrow$ [Korpus Bersih] $\rightarrow$ vsm_ir.py (.fit_transform()) $\rightarrow$ [Matriks TF-IDF (Indeks VSM)]

2. Query Pipeline (Online / Real-time):

[Kueri Pengguna] $\rightarrow$ preprocess.py (Clean, Tokenize, Stop, Stem) $\rightarrow$ [Kueri Bersih] $\rightarrow$ vsm_ir.py (.transform()) $\rightarrow$ [Vektor Kueri] $\rightarrow$ cosine_similarity() (vs Indeks VSM) $\rightarrow$ [Daftar Skor] $\rightarrow$ [Hasil Top-K Terurut]

4.2 Implementasi Antarmuka

Proyek ini memiliki dua *entrypoint* untuk mengakses *search engine*:

1. CLI Orchestrator (src/search.py):

Sesuai permintaan Soal 05 (Langkah 2)17, sebuah skrip command-line dibuat untuk menjalankan pencarian. Skrip ini dapat menerima argumen seperti --model {boolean, vsm}, --query "...", dan --k K18.

2. Antarmuka Web (app/main.py):

Sesuai permintaan Soal 05 (Langkah 3) 19 dan permintaan tambahan pengguna, sebuah antarmuka web interaktif dibuat menggunakan Streamlit. Aplikasi ini memuat model VSM yang telah dilatih dan menyajikan hasil pencarian (snippet dan skor) secara real-time.

4.3 Demo Antarmuka Streamlit

Berikut adalah screenshot dari aplikasi Streamlit (app/main.py) yang sedang berjalan, menampilkan hasil pencarian untuk kueri "pmb walisongo".

[SISIPKAN 1-2 SCREENSHOT APLIKASI STREAMLIT ANDA YANG BERJALAN DI SINI]

BAB 5. EKSPERIMEN & EVALUASI (SOAL 05)

5.1 Skenario Eksperimen

Kami melakukan dua skenario evaluasi untuk mengukur kinerja model:

1. **Evaluasi Model Boolean (Uji Wajib Soal 03):** Menggunakan 3 kueri Boolean dan *gold set* (jawaban relevan manual) untuk menghitung **Precision** dan **Recall**[20].
2. **Perbandingan Skema Bobot VSM (Soal 05):** Membandingkan performa dua skema pembobotan VSM:
    - **Model 1:** TF-IDF Standar (sublinear_tf=False)
    - **Model 2:** TF-IDF Sublinear (sublinear_tf=True) Perbandingan ini menggunakan *gold set* yang sama dan dievaluasi dengan metrik **Precision@k** dan **MAP@k** (Mean Average Precision)[22].

5.2 Metrik Evaluasi

- Precision (Presisi): $\frac{|Relevan \cap Diambil|}{|Diambil|}$

Proporsi dokumen yang diambil yang benar-benar relevan.

- Recall (Perolehan): $\frac{|Relevan \cap Diambil|}{|Relevan|}$

Proporsi dokumen relevan di koleksi yang berhasil diambil.

- Precision@k (P@k): $\frac{|Relevan \cap Diambil \ di \ Top \ K|}{K}$

Presisi pada K dokumen hasil teratas. Ini sangat penting untuk search engine web.

- **MAP@k (Mean Average Precision):** Rata-rata dari skor *Average Precision* (AP) untuk setiap kueri. Ini adalah metrik tunggal yang baik untuk mengukur performa ranking di seluruh set kueri.

5.3 Hasil dan Analisis

Hasil 5.3.1: Evaluasi Model Boolean

Tabel berikut menunjukkan hasil evaluasi Precision dan Recall untuk 3 kueri Boolean.

[SISIPKAN SCREENSHOT TABEL EVALUASI BOOLEAN DARI NOTEBOOK SOAL 03 ANDA DI SINI]

(Contoh format di bawah, ganti dengan screenshot Anda)

Query	Retrieved (Sistem)	Relevant (Gold)	TP	Precision	Recall
semarang	{'doc1.txt', 'doc2.txt', 'doc4.txt'}	{'doc1.txt', 'doc2.txt', 'doc4.txt'}	3	1.00	1.00
salatiga OR pekalongan	{'doc3.txt', 'doc4.txt', 'doc5.txt'}	{'doc3.txt', 'doc4.txt', 'doc5.txt'}	3	1.00	1.00
walisongo AND NOT salatiga	{'doc1.txt', 'doc2.txt'}	{'doc1.txt', 'doc2.txt'}	2	1.00	1.00

Analisis: Model Boolean mendapatkan skor 1.00 pada semua kueri. Ini wajar karena korpus sangat kecil dan *gold set* dibuat berdasarkan *match* eksak, yang merupakan keahlian model Boolean.

Hasil 5.3.2: Perbandingan Skema Bobot VSM (Soal 05)

Tabel berikut membandingkan performa P@3 dan MAP@3 antara TF-IDF standar dan TF-IDF Sublinear.

[SISIPKAN SCREENSHOT TABEL PERBANDINGAN VSM DARI NOTEBOOK SOAL 05 ANDA DI SINI]

(Contoh format di bawah, ganti dengan screenshot Anda)

| Query | Standard TF-IDF (P@3) | Sublinear TF-IDF (P@3) |

| :--- | :--- | :--- |

| semarang | 1.00 | 1.00 |

| salatiga atau pekalongan | 1.00 | 1.00 |

| walisongo | 1.00 | 1.00 |

| --- MAP@k --- | 1.00 | 1.00 |

Analisis: Hasil perbandingan menunjukkan bahwa **tidak ada perbedaan performa** antara skema bobot TF-IDF standar dan Sublinear pada korpus ini. Keduanya menghasilkan MAP@3 yang sempurna (1.00).

Penyebabnya hampir pasti adalah **ukuran korpus yang sangat kecil** (hanya 5 dokumen). Pembobotan sublinear ($\log(tf)+1$) dirancang untuk mengurangi dampak dari *term* yang muncul sangat sering dalam *satu* dokumen (misal, 50 kali). Pada dokumen pendek kami, frekuensi *term* tidak cukup bervariasi untuk menunjukkan perbedaan antara tf dan $\log(tf)$. Meskipun demikian, eksperimen ini telah berhasil memenuhi syarat Soal 05 untuk membandingkan dua skema bobot.

BAB 6. DISKUSI

6.1 Kelebihan dan Keterbatasan

Kelebihan:

- **Implementasi Lengkap:** Proyek ini berhasil mengimplementasikan alur kerja STKI *end-to-end*, dari teks mentah hingga antarmuka web interaktif.
- **Model Ganda:** Mengimplementasikan dua model retrieval (Boolean dan VSM), memungkinkan perbandingan konsep.
- **Pemrosesan Teks Lokal:** Penggunaan Sastrawi untuk *stemming* Bahasa Indonesia meningkatkan kualitas pemrosesan teks.

Keterbatasan:

- **Ukuran Korpus:** Keterbatasan utama adalah ukuran korpus yang hanya 5 dokumen. Ini membuat evaluasi statistik (seperti perbandingan skema bobot) menjadi kurang signifikan.

- **Parser Kueri:** Parser Boolean di src/boolean_ir.py masih sederhana dan belum mendukung kueri kompleks (misal, tanda kurung atau prioritas operator).
- **Gold Set Subjektif:** *Gold set* yang digunakan untuk evaluasi dibuat secara manual dan subjektif oleh pengembang.

6.2 Saran Pengembangan

- **Korpus Lebih Besar:** Menggunakan korpus yang lebih besar dan standar (misal, koleksi artikel berita) untuk mendapatkan hasil evaluasi yang lebih bermakna.
- **Model Ranking Lanjutan:** Mengimplementasikan model ranking yang lebih canggih seperti **BM25 (Okapi)**, yang seringkali mengungguli TF-IDF murni.
- **Evaluasi nDCG:** Menambahkan metrik evaluasi **nDCG@k** untuk menangani relevansi bergradasi (bukan hanya relevan/tidak relevan).

BAB 7. KESIMPULAN

Proyek Ujian Tengah Semester ini telah berhasil mengimplementasikan sebuah *mini search engine* yang memenuhi semua persyaratan fungsional dan teoritis yang ditetapkan dalam soal.

Semua Sub-CPMK yang ditargetkan telah dicapai:

- **Sub-CPMK10.1.1 (Konsep STKI):** Dicapai melalui penjelasan konsep, arsitektur, dan perbandingan STKI vs DB di **Bab 1** dan **Bab 4**.
- **Sub-CPMK10.1.2 (Preprocessing):** Dicapai melalui implementasi praktis di src/preprocess.py dan didemonstrasikan di **Bab 2**.
- **Sub-CPMK10.1.3 (Pemodelan):** Dicapai dengan mengimplementasikan src/boolean_ir.py (Boolean Model) dan src/vsm_ir.py (Vector Space Model), yang dijelaskan di **Bab 3**.
- **Sub-CPMK10.1.4 (Weighting & Evaluasi):** Dicapai melalui eksperimen perbandingan skema *term weighting* dan analisis metrik evaluasi (Precision, Recall, MAP@k) di **Bab 5**.

Proyek ini memberikan pemahaman praktis yang solid tentang bagaimana sebuah *search engine* bekerja, dari pemrosesan teks mentah hingga penyajian hasil yang terurut.

DAFTAR PUSTAKA

- Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media. (Untuk NLTK)
- F. Pedregosa, G. Varoquaux, et al. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research. (Untuk Scikit-learn)
- Harimurti, A. et al. (2017). *Sastrawi: A Simple Python Library for Indonesian Language*. (Untuk Sastrawi)

- Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.