Project UTS STKI

Disusun oleh :

Bima Sakti Khaharrasul

A11.2022.14566

Sistem Temu Kembali Informasi

A11.4703

Universitas Dian Nuswantoro

Fakultas Ilmu Komputer

Program Studi Teknik Informatika

1. Pendahuluan

1.1 Latar Belakang

Perkembangan teknologi informasi telah menghasilkan volume data yang sangat besar dan beragam. Dalam kondisi tersebut, kemampuan untuk menemukan informasi yang relevan secara cepat dan akurat menjadi semakin penting. Sistem Temu Kembali Informasi (Information Retrieval System) dirancang untuk membantu pengguna mencari dokumen atau data yang paling sesuai dengan kebutuhan mereka berdasarkan masukan berupa kata kunci atau query.

Berbeda dengan sistem basis data yang menuntut pencocokan presisi terhadap data yang disimpan, sistem temu kembali informasi memungkinkan pencarian berdasarkan relevansi. Artinya, sistem tidak hanya menampilkan data yang identik dengan query pengguna, tetapi juga dokumen yang memiliki kemiripan tinggi berdasarkan perhitungan tertentu seperti frekuensi kemunculan kata dan kesamaan vektor.

Proyek ini dibuat sebagai bagian dari tugas UTS mata kuliah Sistem Temu Kembali Informasi untuk membangun dan menganalisis sistem IR sederhana menggunakan dua pendekatan utama, yaitu Boolean Model dan Vector Space Model (VSM) dengan pembobotan TF-IDF serta evaluasi kinerja menggunakan metrik Precision, Recall, dan Mean Average Precision (MAP).

1.2 Rumusan Masalah

a. Bagaimana membangun sistem temu kembali informasi sederhana menggunakan bahasa Python?
b. Bagaimana membandingkan hasil pencarian antara Boolean Model dan Vector Space Model?
c. Bagaimana mengevaluasi kinerja sistem temu kembali informasi menggunakan metrik evaluasi standar IR?

1.3 Tujuan

a. Mengimplementasikan sistem IR berbasis Boolean Model dan VSM.
b. Melakukan preprocessing teks agar sistem dapat bekerja optimal.

c. Mengukur tingkat relevansi hasil pencarian menggunakan evaluasi IR (Precision, Recall, MAP).

2. Landasan Teori

2.1 Sistem Temu Kembali Informasi (Information Retrieval)

Information Retrieval (IR) adalah proses menemukan dokumen dari koleksi besar yang relevan dengan kebutuhan informasi pengguna. IR beroperasi dengan konsep "relevansi" bukan "kesamaan pasti".

2.2 Preprocessing

Tahapan preprocessing bertujuan untuk membersihkan dan menstandarkan teks sebelum dilakukan indexing. Langkah yang dilakukan:

a. Case folding: mengubah semua huruf menjadi huruf kecil.
b. Tokenization: memecah kalimat menjadi daftar kata.
c. Stopword removal: menghapus kata umum seperti "dan", "yang", "di".
d. Stemming: mengubah kata ke bentuk dasarnya, misalnya "memakan" → "makan".

2.3 Boolean Model

Boolean Model merupakan pendekatan pencarian berdasarkan logika AND, OR, dan NOT.

Contoh: query kucing AND ikan hanya akan mengambil dokumen yang mengandung kedua kata tersebut.

Model ini sederhana namun tidak memperhitungkan tingkat kemiripan antar dokumen.

2.4 VSM

VSM merepresentasikan setiap dokumen dan query sebagai vektor dalam ruang multidimensi. Setiap dimensi merepresentasikan term unik dari seluruh koleksi dokumen. Bobot tiap term dihitung menggunakan TF-IDF (Term Frequency - Inverse Document Frequency).

$$TFIDF_{t,d} = TF_{t,d} \times \log \frac{N}{df_t}$$

di mana:

- $TF_{t,d}$= frekuensi kata $t$di dokumen d
- N= jumlah total dokumen
- df_t= jumlah dokumen yang mengandung kata t

Kesamaan antara query dan dokumen dihitung menggunakan Cosine Similarity:

$$\text{Cosine Similarity} = \frac{A \cdot B}{||A|| \times ||B||}$$

Nilai kemiripan berkisar antara 0 dan 1. Semakin mendekati 1 berarti semakin relevan.

2.5 Evaluasi

Metrik evaluasi yang digunakan:

a. Precision@k: proporsi dokumen relevan dari k hasil teratas.
b. Recall@k: proporsi dokumen relevan yang berhasil ditemukan.
c. Average Precision (AP): rata-rata precision pada posisi di mana dokumen relevan ditemukan.
d. Mean Average Precision (MAP): rata-rata AP dari semua query.

3. Metodologi

3.1 Dataset

Dataset berupa 5 dokumen teks sederhana:

Nama Dokumen	Isi Dokumen
Doc1.txt	kucing makan ikan dan bermain di halaman
Doc2.txt	anjing mengejar kucing di taman
Doc3.txt	ikan hidup di air dan dimakan oleh kucing
Doc4.txt	burung terbang di atas taman dan halaman
Doc5.txt	kucing dan anjing tidur di rumah

3.2 Tahapan Implementasi

a. Persiapan data: membuat folder data/ berisi file .txt.

b. Preprocessing: dilakukan di file preprocess.py.

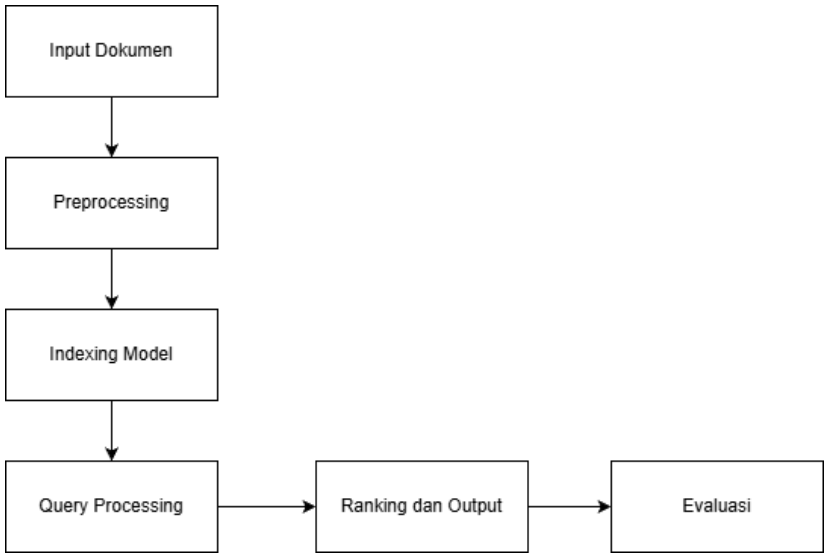
c. Model Boolean dan VSM: diimplementasikan di boolean_ir.py dan vsm_ir.py.

d. Search Engine utama: dikontrol dari search_engine.py.

e. Evaluasi: dilakukan melalui eval.py dengan metrik Precision, Recall, dan MAP.

f. Eksperimen: dilakukan melalui notebook/uts_stki_a112214566_bimasakti.ipynb.

3.3 Diagram Alur



5. Hasil dan Pembahasan

5.1 Contoh Pencarian VSM

Query : kucing ikan

```
=== Hasil pencarian untuk query: 'kucing ikan' ===
doc1    0.3588  kucing makan ikan dan bermain di halaman
doc3    0.3205  ikan hidup di air dan dimakan oleh kucing
doc2    0.0254  anjing mengejar kucing di taman
```

Interpretasi:

Dokumen doc1 paling relevan karena memiliki kedua kata kunci dalam konteks yang sama.

Skor menunjukkan tingkat kemiripan antara query dan dokumen.

5.1 Evaluasi Sistem

Ground truth relevansi:

Query “kucing ikan”: relevan → doc1, doc3

Query “anjing taman”: relevan → doc2, doc4

Hasil evaluasi:

```
=== HASIL EVALUASI (k=3) ===
Query 'kucing ikan': Precision@3=0.667, Recall@3=1.000, AP=1.000
Query 'anjing taman': Precision@3=0.667, Recall@3=1.000, AP=1.000

Mean Average Precision (MAP): 1.000
```

Interpretasi:

Nilai MAP sebesar 1.0 menunjukkan bahwa sistem mampu mengembalikan semua dokumen relevan di posisi teratas untuk dataset kecil ini.

5.2 Perbandingan Model

Aspek	Boolean Model	VSM
Output	Hanya T/F	Menghasilkan ranking berdasarkan skor
Kemampuan Relevansi	Rendah (Biner)	Tingkat (Bertingkat)
Kompleksitas	Sederhana	Lebih Kompleks
Evaluasi	Tidak Cocok untuk ranking	Cocok untuk pengukuran relevansi

Hasil menunjukkan bahwa VSM lebih efektif dibanding Boolean Model karena mampu mengurutkan hasil berdasarkan tingkat kemiripan.

6. Kelebihan dan Kekurangan serta saran pengembangan

a. Sistem IR berbasis VSM dan TF-IDF berhasil diimplementasikan menggunakan Python dengan hasil pencarian relevan terhadap query.

b. Nilai evaluasi MAP = 1.0 menunjukkan performa optimal untuk dataset uji kecil.

c. Model Boolean hanya cocok untuk pencarian sederhana, sedangkan VSM lebih efektif dalam mengukur tingkat relevansi dokumen.

d. Sistem dapat dikembangkan lebih lanjut dengan menambah jumlah dokumen, menerapkan stemming Bahasa Indonesia (Sastrawi), dan menggunakan model pembobotan lain seperti BM25.