Laporan UTS STKI
NAMA : BISONO PRIYAMBODO M

NIM : A11.2023.15217

1. Pendahuluan

Proyek ini dibuat untuk memenuhi UTS mata kuliah Sistem Temu Kembali Informasi (STKI),

dengan tujuan merancang dan membangun sebuah mini search engine yang dapat melakukan proses pencarian dokumen menggunakan dua pendekatan utama: Boolean Retrieval dan Vector Space Model(VSM).

Secara umum, search engine modern bekerja dengan tiga tahapan besar: preprocessing teks, indexing, dan retrieval. Pada proyek ini, seluruh tahapan tersebut direalisasikan dalam bentuk modul Python yang saling terhubung. Meskipun bersifat sederhana, sistem ini mengimplementasikan konsep inti Information Retrieval sebagaimana diterapkan pada mesin pencari skala besar seperti Google, Elasticsearch, dan Lucene.

Ruang lingkup proyek mencakup:

- Pembersihan teks (lowercasing, tokenisasi, stopword removal).
- Implementasi inverted index untuk Boolean Retrieval.
- Perhitungan bobot TF, DF, IDF.
- Pembentukan matriks TF-IDF & VSM.
- Pencarian menggunakan cosine similarity.
- Evaluasi retrieval menggunakan Precision, Recall, F1, MAP@5, dan nDCG@5.

Kontribusi proyek terhadap Sub-CPMK:

- Sub-CPMK 10.1.1 → Mampu melakukan preprocessing dan representasi dokumen.
- Sub-CPMK 10.1.2 → Mampu membangun inverted index dan melakukan Boolean Retrieval.
- Sub-CPMK 10.1.3 → Mampu mengimplementasikan VSM, TF-IDF, dan cosine similarity.
- Sub-CPMK 10.1.4 → Mampu melakukan evaluasi dan analisis hasil sistem temu kembali.

Proyek ini bukan hanya implementasi teknis, tetapi juga latihan bagaimana

mengubah konsep IR menjadi sistem nyata yang bisa dieksekusi dan diuji.

2. Dataset dan Tahap Preprocessing

Dataset proyek terdiri dari 15 dokumen teks (.txt) yang berisi informasi mengenai game Counter-Strike. Setiap dokumen memiliki gaya penulisan berbeda: ada yang berbentuk paragraf naratif, artikel pendek, hingga bullet point. Karena karakteristik data seperti ini biasanya berisik, preprocessing menjadi sangat penting.

3.

Tahapan preprocessing yang dilakukan:

1. Lowercasing
4. Semua teks diubah menjadi huruf kecil agar "Counter" dan "counter" dianggap sama.
2. Tokenisasi
5. Kalimat dipotong menjadi token menggunakan regex sederhana berbasis huruf dan angka.
3. Stopword Removal
6. Stopword Bahasa Indonesia seperti "yang", "dan", "atau", "sebagai" dihapus agar model fokus pada kata bermakna.
4. Stemming
7. Menggunakan stemming sederhana berbasis rule, seperti menghapus akhiran "-kan", "-nya", "-lah".
8. Tidak menggunakan stemmer library untuk menjaga konsistensi manual IR.

Contoh Before/After:

Before:

"Counter-Strike pertama kali lahir sebagai mod Half-Life buatan Minh Le."

After preprocessing:

"counter strike pertama kali lahir mod half life minh le"

Tahap preprocessing ini menghasilkan dokumen yang jauh lebih bersih dan siap diproses pada indexing dan

model VSM. Semua hasil preprocessing otomatis tersimpan pada folder:

`data/processed/`

3. Metode Information Retrieval

Proyek ini mengimplementasikan dua pendekatan IR yang umum digunakan: Boolean Model dan Vector Space Model.

A. Boolean Retrieval

Metode ini menggunakan struktur data inverted index, di mana setiap term menyimpan daftar dokumen yang mengandungnya. Query dapat menggunakan operator:

- AND
- OR
- NOT

Contoh query:

"counter AND strike" → mengambil dokumen yang mengandung kedua term.

Kelebihan Boolean Model:

- Cepat dan sederhana.
- Mendukung operasi logika.

Kelemahan:

- Tidak ada ranking.

- Hasil terlalu biner (relevan atau tidak).

B. Vector Space Model (VSM)

Setiap dokumen direpresentasikan sebagai vektor berdimensi |V| (jumlah vocabulary). Bobot utama menggunakan TF-IDF.

Rumus digunakan:

TF = frekuensi term

DF = jumlah dokumen yang mengandung term

IDF = log(N / DF)

TF-IDF = TF × IDF

Model menghitung cosine similarity:

sim(Q,D) = (Q·D) / (||Q|| × ||D||)

Kami membandingkan dua skema pembobotan:

1. TF-IDF standar
2. TF-IDF Sublinear (logarithmic TF)

Sublinear TF mengurangi dominasi kata yang muncul terlalu sering (overweight).

Hasil eksperimen menunjukkan perbedaan keduanya cukup signifikan pada MAP dan nDCG.

4. Arsitektur Sistem Search Engine

Struktur folder utama proyek:

stki-uts-A112315217/

├─ data/
│ ├─ raw/ → dokumen asli
│ └─ processed/ → hasil preprocessing
├─ src/
│ ├─ preprocess.py
│ ├─ boolean_ir.py
│ ├─ vsm.py
│ ├─ eval.py
│ └─ search.py → orchestrator backend

├─ app/

│ └─ main.py → interface CLI

└─ notebooks/

└─ UTS_STKI.ipynb

Deskripsi modul:

- preprocess.py

  Membersihkan teks & menyiapkan dokumen untuk indexing.

- boolean_ir.py

  Membangun inverted index dan melakukan Boolean search.

- vsm.py

  Menghitung TF, DF, IDF, TF-IDF, membangun matriks TF-IDF, dan melakukan ranking.

- eval.py

  Berisi fungsi metrik evaluasi seperti precision, recall, F1, MAP, dan nDCG.

- main.py

  Menjalankan seluruh pipeline mulai preprocessing → boolean → VSM → evaluasi → mode interactive.

Alur sistem (flow sederhana):

Raw Files → Preprocessing → Indexing → Query → Boolean/VSM → Ranking → Evaluasi → Output

5. Eksperimen, Evaluasi, dan Analisis

Tiga query uji yang digunakan:

1. "counter strike"

2. "smoke flash"

3. "awp recoil"

A. Evaluasi Boolean Model

Boolean model cenderung menghasilkan recall tinggi, namun precision rendah karena tidak adanya ranking.

Contoh hasil (singkat):

- Precision = rendah (0.2 - 0.3)

- Recall = selalu tinggi untuk query sederhana

- F1 = cukup rendah

Query	precision	recall	f1
counter strike	0.25	1	0.4000
smoke flash	0.5556	0.7143	0.6250
awp recoil	0.6667	0.6667	0.6667

B. Evaluasi VSM – TF-IDF Standard dan Sublinear

Metrik yang digunakan:

- Precision@5
- MAP@5
- nDCG@5

Table standard

Metirc	Score
Average P@5	0.466
Average MAP@5	1.134
Average nDCG@5	0.786

Table Sublinear

Metirc	Score
Average P@5	0.533
Average MAP@5	1.201
Average nDCG@5	0.812

Ringkasan pengamatan:

- TF-IDF Standard → memberikan skor tinggi pada dokumen dengan frekuensi term tinggi.
- Sublinear → lebih stabil dan tidak terlalu overweight satu dokumen saja.

Secara umum Sublinear TF memenangkan MAP@5 dan nDCG@5.

Contoh perbandingan:

- Query "smoke flash" → TF-IDF Standard: P@5 = 0.6 | Sublinear = 1.0
- nDCG@5 Sublinear hampir selalu lebih baik.

C. Interpretasi Hasil

- Boolean berguna untuk filtering awal.
- VSM lebih baik untuk ranking nyata.
- Dataset kecil membuat metrik sensitif, namun pola tetap terlihat.
-

6. Diskusi, Kelebihan, Keterbatasan, dan Saran

Kelebihan Proyek:

- Pipeline IR lengkap sudah terimplementasi.
- Arsitektur modular, mudah dikembangkan ulang.
- Mendukung dua skema term weighting & membandingkannya.
- Tersedia interactive mode layaknya search console.

Keterbatasan:

- Dataset sangat kecil (15 dokumen).
- Preprocessing masih sederhana (stemming rule-based).
- Belum mendukung phrase retrieval atau proximity search.
- Model belum mendukung BM25.

Saran Pengembangan:

1. Menambah dataset lebih besar agar evaluasi lebih stabil.
2. Mengimplementasikan BM25 untuk pembobotan yang lebih modern.
3. Membuat antarmuka web menggunakan Flask/React.
4. Menambahkan fitur query expansion berbasis Word2Vec/BERT.
5. Menyimpan model TF-IDF dalam bentuk pickle/JSON untuk produksi.

Dengan adanya proyek ini, mahasiswa mampu memahami implementasi IR secara praktis dan melihat langsung bagaimana performa model berubah karena pemilihan skema TF-IDF.