# LAPORAN PROYEK UJIAN TENGAH SEMESTER

## Implementasi dan Analisis Perbandingan Model Boolean dan Vector Space pada Sistem Temu Kembali Informasi Berbasis Teks


Disusun oleh:

| | |
|---|---|
| Nama | : Raditya Abdul Afeef |
| NIM | : A11.2022.14203 |
| Program Studi | : Teknik Informatika |

**FAKULTAS ILMU KOMPUTER**
**UNIVERSITAS DIAN NUSWANTORO**
**SEMARANG**

**2025**

# BAB I
# PENDAHULUAN

## A. Latar Belakang

Di era informasi digital, volume data tekstual yang dihasilkan setiap hari tumbuh secara eksponensial. Kemampuan untuk menyaring, menemukan, dan menyajikan informasi yang relevan dari koleksi data yang masif merupakan tantangan utama. Sistem Temu Kembali Informasi (STKI), atau yang lebih dikenal sebagai mesin pencari, adalah teknologi inti yang menjawab tantangan ini. Memahami prinsip-prinsip dasar yang mendasari cara kerja mesin pencari—mulai dari pemrosesan teks mentah hingga perankingan dokumen—adalah kompetensi esensial dalam ilmu komputer. Proyek ini dirancang sebagai implementasi praktis dari teori-teori STKI, membangun sebuah sistem fungsional dari awal untuk mendapatkan pemahaman yang mendalam dan aplikatif.

## B. Ruang Lingkup Proyek

Proyek ini memiliki ruang lingkup sebagai berikut::

1. Menggunakan korpus data teks berukuran kecil (10 dokumen) berbahasa Indonesia dengan tema "Hewan Langka di Indonesia" yang dikumpulkan secara manual dari Wikipedia.
2. Mengimplementasikan dua model pencarian fundamental: Boolean Retrieval Model untuk pencarian eksak dan Vector Space Model (VSM) untuk pencarian berbasis relevansi.
3. Melakukan evaluasi kuantitatif terhadap kedua model menggunakan metrik standar industri seperti Precision, Recall, dan Mean Average Precision (MAP).
4. Membangun antarmuka pengguna (UI) yang interaktif menggunakan framework Streamlit.

## C. Kontribusi Proyek vs. Sub-CPMK

Proyek ini secara langsung berkontribusi pada pencapaian beberapa Sub-Capaian Pembelajaran Mata Kuliah (Sub-CPMK), antara lain:

1. Sub-CPMK 10.1.2: Diimplementasikan melalui pembangunan modul preprocessing data teks.
2. Sub-CPMK 10.1.3: Diimplementasikan melalui pembangunan Boolean Model dan Vector Space Model lengkap dengan perhitungan TF-IDF dan Cosine Similarity.
3. Sub-CPMK 10.1.4: Diimplementasikan melalui penerapan skema pembobotan istilah (Term Weighting) dan evaluasi model menggunakan metrik yang relevan.

# BAB II
# METODOLOGI DAN IMPLEMENTASI

## A. Kerangka Kerja Proyek

Proyek ini dilaksanakan melalui beberapa tahapan yang sistematis, digambarkan sebagai berikut:

1. Pengumpulan Data: Mengidentifikasi tema dan mengumpulkan 10 dokumen teks dari sumber terpercaya (Wikipedia).
2. Preprocessing Data: Mengaplikasikan pipeline preprocessing pada seluruh korpus untuk menghasilkan token yang bersih.
3. Pembangunan Model: Mengimplementasikan logika untuk Model Boolean (dengan Inverted Index) dan VSM (dengan TF-IDF dan Cosine Similarity).
4. Evaluasi Model: Menyiapkan truth set (gold standard) dan skenario pengujian, kemudian menghitung metrik performa (Precision, Recall, MAP) untuk setiap model.
5. Pengembangan Antarmuka: Mengintegrasikan semua modul ke dalam sebuah aplikasi web interaktif menggunakan Streamlit.

## B. Data dan Preprocessing

1. Sumber dan Karakteristik Data

   Korpus data yang digunakan dalam proyek ini terdiri dari 10 dokumen teks yang berisi deskripsi singkat mengenai hewan-hewan langka di Indonesia. Sumber data diambil dari artikel Wikipedia Indonesia. Pemilihan tema yang spesifik ini bertujuan untuk menciptakan sebuah skenario pencarian yang realistis, di mana dokumen-dokumen memiliki kemiripan tema namun tetap memiliki entitas yang unik.

2. Tahapan Preprocessing

   Data mentah dari web tidak dapat langsung digunakan oleh model. Diperlukan serangkaian proses pembersihan yang disebut preprocessing untuk mengubah teks menjadi data yang terstruktur dan bersih. Tahapan yang dilakukan adalah sebagai berikut:

   a. Case Folding: Menyeragamkan teks dengan mengubah semua huruf menjadi huruf kecil.
   b. Cleaning: Menghilangkan *noise* atau karakter yang tidak relevan seperti tanda baca, angka, URL, dan tag HTML.
   c. Tokenization: Memecah kalimat atau teks menjadi unit-unit kata individual yang disebut token.
   d. Stopword Removal: Menghapus kata-kata umum yang sering muncul namun tidak memiliki makna signifikan dalam konteks pencarian (contoh: "yang", "di", "dan", "adalah").

e. Stemming: Mengubah setiap kata ke bentuk dasarnya (kata dasar) untuk mengurangi variasi kata dengan makna yang sama (contoh: "menemukan", "ditemukan" -> "temu"). Proses ini dilakukan menggunakan library Sastrawi yang dirancang khusus untuk Bahasa Indonesia.

## C. Metode Information Retrieval

1. Boolean Retrieval Model

   Model Boolean adalah model pencarian paling dasar yang bekerja berdasarkan logika himpunan (set) dan teori Boolean. Model ini memproses query yang mengandung operator logika seperti AND, OR, dan NOT.

   - AND: Mengembalikan dokumen yang mengandung semua term query (irisan/intersection).
   - OR: Mengembalikan dokumen yang mengandung salah satu dari term query (gabungan/union).
   - NOT: Mengembalikan dokumen yang tidak mengandung term query (komplemen).

   Tulang punggung dari model ini adalah struktur data Inverted Index, yang memetakan setiap kata unik ke daftar dokumen yang memuatnya, sehingga proses pencarian menjadi sangat efisien.

2. Vector Space Model (VSM)

   VSM adalah model yang lebih canggih yang merepresentasikan dokumen dan query sebagai vektor dalam sebuah ruang multidimensi, di mana setiap dimensi diwakili oleh sebuah kata unik dari korpus. Relevansi antara sebuah query dan sebuah dokumen diukur berdasarkan "kedekatan" vektor keduanya.

   - Pembobotan TF-IDF: Untuk menentukan nilai setiap komponen dalam vektor, digunakan skema pembobotan TF-IDF (Term Frequency-Inverse Document Frequency).
     - Term Frequency (TF): Mengukur seberapa sering sebuah kata muncul dalam satu dokumen.
     - Inverse Document Frequency (IDF): Mengukur seberapa langka atau unik sebuah kata di seluruh koleksi dokumen.
     - Rumus TF-IDF:

       $$W(t,d) = TF(t,d) \times IDF(t)$$

   - Cosine Similarity: Kedekatan antara vektor query (q) dan vektor dokumen (d) dihitung menggunakan Cosine Similarity, yang mengukur sudut kosinus di antara keduanya. Skor berkisar dari 0 (tidak mirip) hingga 1 (sangat mirip).

     Rumus Cosine Similarity:

     $$Similarity(q, d) = (q \cdot d) / (||q|| \times ||d||)$$

## D. Arsitektur Search Engine

Arsitektur aplikasi mesin pencari ini dirancang secara modular dan efisien, terdiri dari dua fase utama:

1. Fase Inisialisasi: Saat aplikasi pertama kali dijalankan, kelas SearchEngine akan diinisialisasi. Proses ini hanya terjadi satu kali dan mencakup semua perhitungan berat:
   - Memuat semua dokumen dari folder data/.
   - Melakukan preprocessing pada ke-10 dokumen.
   - Membangun Inverted Index untuk model Boolean.
   - Menghitung dan menyimpan matriks TF-IDF untuk model VSM.
   - Dengan melakukan ini di awal, pencarian selanjutnya dapat dilakukan dengan sangat cepat.

2. Fase Pencarian: Ketika pengguna memasukkan query dan mengklik tombol "Cari":
   - Antarmuka Streamlit (app/main.py) menangkap input query dan pilihan model (Boolean/VSM).
   - Input tersebut diteruskan ke fungsi engine.search().
   - Fungsi orchestrator ini kemudian memanggil modul yang sesuai (boolean_ir.py atau vsm_ir.py) untuk memproses query.
   - Hasil (daftar dokumen atau daftar dokumen terurut) dikembalikan ke antarmuka untuk ditampilkan kepada pengguna.

# BAB III
# HASIL DAN ANALISIS

## A. Hasil Preprocessing

Pipeline preprocessing terbukti efektif dalam membersihkan dan mereduksi data. Perbandingan teks before vs after menunjukkan hilangnya noise dan normalisasi kata menjadi bentuk dasarnya.

```
#Testing pipeline sebelum dan sesudah
sample_doc_name = 'harimau_sumatra.txt'
sample_text = documents[sample_doc_name]

processed_tokens = preprocess_pipeline(sample_text)

print("--- SEBELUM PREPROCESSING ---")
print(sample_text)
print("\n" + "="*50 + "\n")
print("--- SETELAH PREPROCESSING ---")
print(processed_tokens)
```

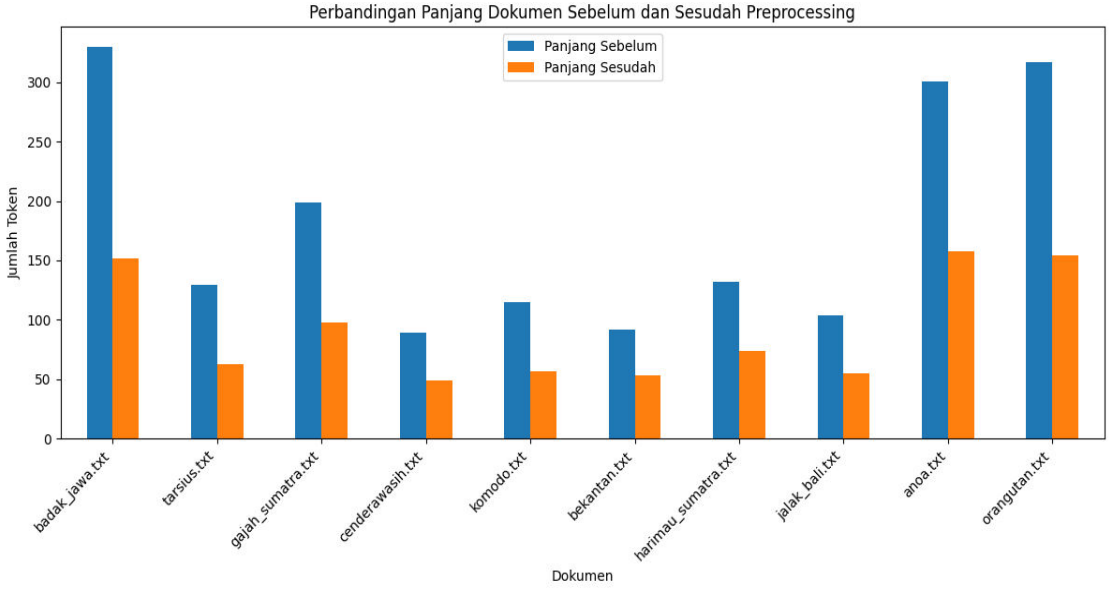
```
--- SEBELUM PREPROCESSING ---
Harimau sumatra adalah populasi Panthera tigris sondaica yang mendiami pulau Sumatra, Indonesia dan satu-satunya anggota subspesies harimau sunda

Penghancuran habitat merupakan ancaman terbesar terhadap populasi saat ini. Pembalakan tetap berlangsung bahkan di taman nasional yang seharusnya

==================================================

--- SETELAH PREPROCESSING ---
['harimau', 'sumatra', 'populasi', 'panthera', 'tigris', 'sondaica', 'ami', 'pulau', 'sumatra', 'indonesia', 'satusatunya', 'anggota', 'subspesie
```

Efektivitas ini juga terlihat dari grafik perbandingan jumlah token, di mana terjadi penurunan signifikan setelah preprocessing.



## B. Evaluasi Model Boolean

Model Boolean dievaluasi menggunakan metrik Precision dan Recall pada 3 query uji dengan truth set yang telah ditentukan sebelumnya.

```
--- UJI WAJIB BOOLEAN RETRIEVAL ---

Query: 'pulau AND sulawesi'
 -> Gold Standard: ['anoa.txt']
 -> Hasil Sistem: ['anoa.txt']
   - True Positives: 1
   - False Positives: 0
   - False Negatives: 0
   - Precision: 1.00
   - Recall: 1.00

Query: 'kera OR monyet'
 -> Gold Standard: ['bekantan.txt', 'orangutan.txt']
 -> Hasil Sistem: ['bekantan.txt', 'orangutan.txt']
   - True Positives: 2
   - False Positives: 0
   - False Negatives: 0
   - Precision: 1.00
   - Recall: 1.00

Query: 'punah AND kritis'
 -> Gold Standard: ['harimau_sumatra.txt']
 -> Hasil Sistem: ['badak_jawa.txt', 'harimau_sumatra.txt']
   - True Positives: 1
   - False Positives: 1
   - False Negatives: 0
   - Precision: 0.50
   - Recall: 1.00
```

Model ini menunjukkan kinerja sempurna (Precision=1.00, Recall=1.00) untuk query yang term-nya spesifik. Namun, sebuah hasil menarik terlihat pada query punah AND kritis. Model ini menghasilkan Recall 1.00 tetapi Precision hanya 0.50. Ini adalah contoh klasik yang menyoroti kelemahan Model Boolean dimana ia menemukan kedua kata kunci di dokumen badak_jawa.txt dan harimau_sumatra.txt dan menganggap keduanya sama-sama relevan, tanpa mempertimbangkan konteks atau bobot kata kunci tersebut di masing-masing dokumen.

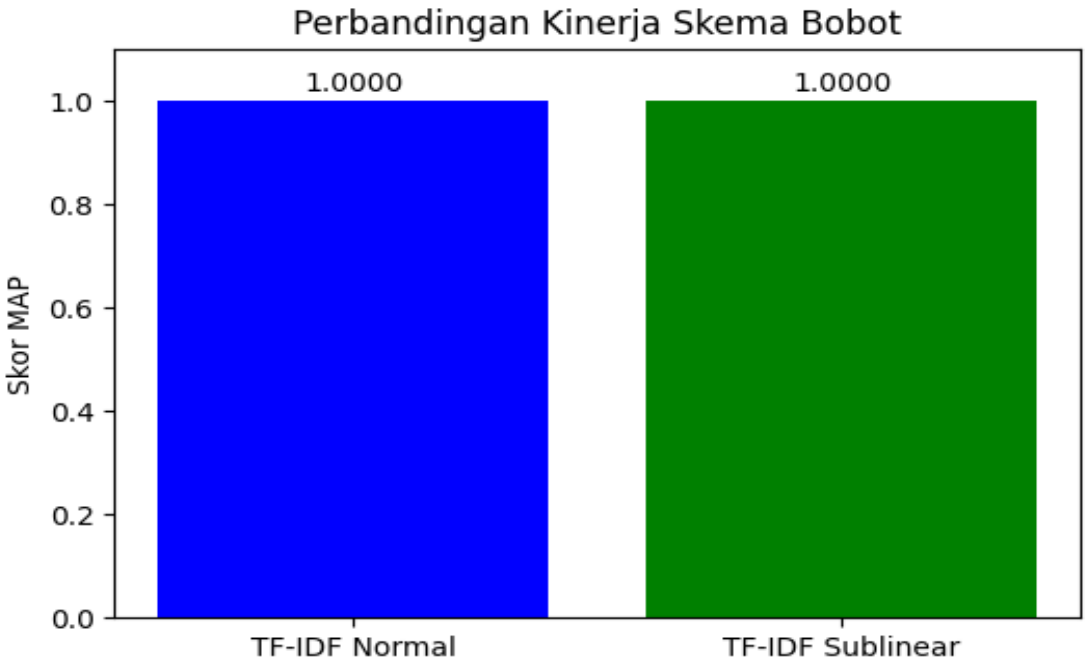
## C. Evaluasi Model VSM & Perbandingan Skema Bobot

VSM mampu memberikan hasil yang terurut berdasarkan relevansi. Untuk query "hewan kera besar pemakan buah dari kalimantan", sistem berhasil menempatkan bekantan.txt dan orangutan.txt di peringkat teratas dengan skor relevansi yang tinggi.

```
Hasil pencarian teratas untuk query: 'hewan kera besar pemakan buah dari kalimantan'

Skor: 0.1459 - Dokumen: bekantan.txt
   Snippet: Bekantan, kahau, kera belanda atau monyet belanda (Nasalis larvatus) adalah jenis monyet berhidung panjang dengan
--------------------------------------------------
Skor: 0.1409 - Dokumen: orangutan.txt
   Snippet: Orang utan (bentuk tidak baku: orangutan) atau mawas adalah kera besar yang berasal dari hutan hujan Indonesia dan
--------------------------------------------------
Skor: 0.0273 - Dokumen: gajah_sumatra.txt
   Snippet: Gajah sumatera (bahasa Latin: Elephas maximus sumatranus) adalah subspesies dari gajah asia yang hanya berhabitat
--------------------------------------------------
```

Untuk analisis lebih lanjut, dilakukan perbandingan kinerja antara skema bobot TF-IDF Normal dan Sublinear menggunakan metrik MAP.




Hasil evaluasi menunjukkan skor MAP yang identik (1.0000) untuk kedua skema. Ini mengindikasikan bahwa pada korpus data yang kecil dan memiliki topik yang cukup terpisah, perubahan cara pembobotan oleh skema Sublinear belum cukup kuat untuk mengubah urutan dokumen teratas yang relevansinya sudah sangat jelas. Fenomena ini mungkin akan berbeda pada korpus yang lebih besar dan kompleks

# BAB IV
# DISKUSI DAN KESIMPULAN

## A. Diskusi

1. Analisis Perbandingan Model
   Berdasarkan implementasi dan evaluasi, kedua model menunjukkan karakteristik yang berbeda:
   - Boolean Model: Sangat cepat dan presisi untuk query yang membutuhkan hasil eksak. Namun, model ini kaku, tidak memiliki konsep relevansi, dan sering kali mengembalikan terlalu banyak atau terlalu sedikit hasil.
   - Vector Space Model: Jauh lebih fleksibel dan intuitif bagi pengguna. Kemampuannya untuk me-ranking dokumen berdasarkan relevansi adalah keunggulan utamanya. Namun, VSM lebih kompleks secara komputasi.

2. Keterbatasan Proyek
   Proyek ini memiliki beberapa keterbatasan, antara lain:
   - Ukuran Korpus: Korpus yang sangat kecil membuat beberapa fenomena (seperti perbedaan signifikan antar skema bobot) tidak terlihat.
   - Penanganan Bahasa: Tidak ada penanganan untuk sinonim (misalnya, "kera" dan "monyet" dianggap kata yang sama sekali berbeda).
   - Parser Query Sederhana: Parser untuk model Boolean masih sederhana dan tidak mendukung query yang kompleks dengan tanda kurung.

## B. Kesimpulan

Proyek ini telah berhasil mencapai semua tujuannya. Sebuah mesin pencari mini fungsional telah berhasil dibangun dari awal, mengimplementasikan Model Boolean dan Vector Space Model. Proses preprocessing terbukti esensial dalam menyiapkan data untuk kedua model. Evaluasi kuantitatif menunjukkan bahwa VSM, dengan kemampuannya me-ranking hasil berdasarkan relevansi, lebih cocok untuk kebutuhan pencarian umum, sementara Model Boolean tetap relevan untuk kasus penggunaan yang memerlukan pencocokan eksak. Proyek ini secara efektif memenuhi semua tujuan pembelajaran yang ditetapkan dalam Sub-CPMK mata kuliah Sistem Temu Kembali Informasi.

## C. Saran Pengembangan

Sistem ini memiliki potensi untuk dikembangkan lebih lanjut, antara lain dengan:

1. Penerapan Model BM25: Mengganti skema TF-IDF dengan Okapi BM25, yang merupakan standar industri dan sering kali memberikan hasil yang lebih baik.
2. Ekspansi Korpus: Menguji sistem pada korpus yang lebih besar untuk menganalisis tantangan skalabilitas dan melihat dampak skema pembobotan secara lebih jelas.
3. Penanganan Sinonim: Mengimplementasikan teknik seperti Query Expansion atau word embeddings untuk memungkinkan sistem memahami hubungan semantik antar kata (misalnya, "kera" dan "monyet").