LAPORAN TUGAS MINI SEARCH ENGINE (UTS)

MATA KULIAH SISTEM TEMUKEMBALI INFORMASI

(A11.4703)

Disusun oleh :

Nama : Suryani Ayu Dewanti

NIM : A11.2023.15018

Kelompok : A11.4703

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS DIAN NUSWANTORO SEMARANG

NOVEMBER 2025

1. Pendahuluan

Proyek Mini Search Engine (UTS) ini bertujuan untuk membangun sistem *Temu Kembali Informasi* (Information Retrieval System) berskala kecil menggunakan 15 dokumen teks universitas di Indonesia.
Sistem mencakup tahapan *preprocessing*, pembangunan *Boolean Retrieval Model* dan *Vector Space Model (VSM)*, pembobotan istilah menggunakan *TF-IDF* dan *TF-IDF Sublinear*, serta evaluasi performa menggunakan metrik *Precision*, *Recall*, *F1-score*, *MAP@5*, dan *nDCG@5*.

Selain versi terminal, proyek ini juga memiliki aplikasi berbasis **Streamlit** yang menampilkan hasil pencarian secara interaktif melalui antarmuka web. Aplikasi ini dijalankan menggunakan file app/main.py yang berfungsi sebagai **index utama (entry point)** untuk deployment sistem.

2. Data dan Preprocessing

Tahapan preprocessing dilakukan terhadap 15 dokumen teks universitas. Proses mencakup *case folding*, *tokenisasi*, *stopword removal*, *stemming*, dan *normalisasi*.
Setelah proses selesai, sistem menghasilkan 15 dokumen dengan panjang bervariasi:

Nama Dokumen	Jumlah Token
itb.txt	153
stmik_bm_palu.txt	114
ub.txt	122
udinus.txt	132
ugm.txt	144
ui.txt	174
unand.txt	122
undip.txt	144
ung.txt	104
unima.txt	132
unimal.txt	126
unmubuton.txt	119
unpad.txt	135
unsri.txt	102
unud.txt	107

Grafik distribusi token disimpan sebagai data/processed/distribusi_dokumen.png.
Tahapan ini menghasilkan korpus yang siap digunakan untuk proses indexing dan perhitungan bobot istilah.

3. Boolean Retrieval Model

Model *Boolean Retrieval* menggunakan struktur *inverted index* dan mendukung operasi logika **AND**, **OR**, dan **NOT**. Berikut hasil uji beberapa query:

Hasil Eksperimen: Boolean Retrieval Model

No	Query	Operator	Hasil Dokumen (Ringkas)	Precision	Recall	Keterangan
1	**nama AND universitas**	AND	itb.txt, ub.txt, udinus.txt, ugm.txt, ui.txt, unand.txt, undip.txt, ung.txt, unima.txt, unimal.txt, unmubuton.txt, unpad.txt, unsri.txt, unud.txt	0.21	0.75	Menghasilkan 14 dokumen yang mengandung kedua term *nama* dan *universitas*.
2	**teknik OR semarang**	OR	itb.txt, stmik_bm_palu.txt, ub.txt, udinus.txt, ugm.txt, ui.txt, undip.txt, ung.txt, unima.txt, unimal.txt, unmubuton.txt, unsri.txt	1.00	0.80	Mengembalikan 12 dokumen yang mengandung salah satu term; hasil union dari dua himpunan dokumen.
3	**NOT bandung**	NOT	stmik_bm_palu.txt, ub.txt, udinus.txt, ugm.txt, unand.txt, undip.txt, ung.txt, unima.txt, unimal.txt, unmubuton.txt, unpad.txt, unsri.txt, unud.txt	1.00	1.00	Menampilkan semua dokumen kecuali yang mengandung kata “bandung”.

Analisis Hasil Boolean IR

1. **Query “nama AND universitas”**
    - Menggunakan operator *interseksi* (AND), sehingga hanya dokumen yang mengandung kedua kata tersebut ditampilkan.
    - Hasil menunjukkan **Precision rendah (0.21)** karena sistem juga memunculkan dokumen kurang relevan terhadap konteks spesifik query, namun **Recall tinggi (0.75)** menunjukkan sebagian besar dokumen relevan berhasil ditemukan.
2. **Query “teknik OR semarang”**
    - Menggunakan operator *union* (OR) yang lebih longgar.
    - Semua dokumen yang mengandung salah satu dari dua term dikembalikan.
    - **Precision = 1.0** berarti semua hasil relevan, tetapi **Recall = 0.8** karena ada beberapa dokumen relevan yang tidak muncul.
3. **Query “NOT bandung”**
    - Menggunakan *komplemen* (NOT), yaitu mengambil semua dokumen yang **tidak mengandung kata “bandung”**.

- Karena logika NOT menyeleksi secara eksklusif, sistem menghasilkan **Precision dan Recall sempurna (1.0)** — tidak ada dokumen salah terambil.

Model ini efektif untuk pencarian biner (relevan/tidak relevan) dan memiliki *Precision* tinggi pada sebagian besar query.

4. Vector Space Model (VSM)

Model *Vector Space* merepresentasikan setiap dokumen dan query dalam bentuk vektor multidimensi dengan pembobotan *TF-IDF* dan *TF-IDF Sublinear*.
Kesamaan dihitung menggunakan *cosine similarity* untuk menghasilkan peringkat dokumen berdasarkan relevansi.Tabel 1. Perbandingan Skema Pembobotan Istilah

Query 1: universitas

Rank	Dokumen	TF-IDF Standard	TF-IDF Sublinear	Snippet (potongan isi dokumen)
1	undip.txt	0.0158	0.0107	"universitas diponegoro semarang berdiri tahun 1957 sebagai perguruan tinggi negeri..."
2	unud.txt	0.0133	0.0118	"universitas udayana merupakan salah satu perguruan tinggi tertua di bali yang memiliki..."
3	ub.txt	0.0133	0.0112	"universitas brawijaya malang menawarkan berbagai program studi unggulan di bidang..."
4	ung.txt	0.0130	0.0124	"universitas negeri gorontalo berdiri tahun 1963 dengan visi menjadi universitas unggul..."
5	unmubuton.txt	0.0119	0.0099	"universitas muhammadiyah buton berkomitmen pada pendidikan berbasis nilai islam..."

Analisis:
Model TF-IDF Sublinear memberikan perubahan peringkat kecil: *UNG* naik ke posisi lebih tinggi dibanding *UB*.
Hal ini menunjukkan bahwa sublinear scaling menyeimbangkan pengaruh kata "universitas" yang sangat sering muncul di seluruh dokumen.

Query 2: fasilitas

Rank	Dokumen	TF-IDF Standard	TF-IDF Sublinear	Snippet
1	itb.txt	0.0000	0.0000	"institut teknologi bandung adalah perguruan tinggi teknik pertama di indonesia..."
2	stmik_bm_palu.txt	0.0000	0.0000	"stmik bumi manado palu merupakan sekolah tinggi ilmu komputer..."

Rank	Dokumen	TF-IDF Standard	TF-IDF Sublinear	Snippet
3	ub.txt	0.0000	0.0000	“universitas brawijaya malang menawarkan berbagai program studi...”
4	udinus.txt	0.0000	0.0000	“universitas dian nuswantoro semarang dikenal dengan unggulan di bidang teknologi...”
5	ugm.txt	0.0000	0.0000	“universitas gadjah mada merupakan universitas riset terkemuka di yogyakarta...”

Analisis:

Kata *fasilitas* tidak ditemukan pada seluruh korpus, menyebabkan semua bobot = 0.

Ini menunjukkan perlunya **perluasan korpus atau sinonim expansion** (misalnya menambahkan kata “sarana” atau “layanan”) agar hasil pencarian lebih informatif.

Query 3: fakultas teknik

Rank	Dokumen	TF-IDF Standard	TF-IDF Sublinear	Snippet
1	itb.txt	0.0228	0.0225	“institut teknologi bandung memiliki berbagai fakultas teknik terkemuka di indonesia...”
2	ui.txt	0.0130	0.0165	“universitas indonesia mempunyai fakultas teknik terbesar dengan program teknik mesin...”
3	unsri.txt	0.0113	0.0182	“universitas sriwijaya fakultas teknik palembang dikenal menghasilkan lulusan kompeten...”
4	ub.txt	0.0108	0.0152	“universitas brawijaya fakultas teknik memiliki berbagai program unggulan...”
5	ung.txt	0.0105	0.0168	“universitas negeri gorontalo membuka fakultas teknik sebagai bagian dari pengembangan...”

Analisis:

Kedua model konsisten menempatkan **ITB** di posisi puncak.

Namun, TF-IDF Sublinear memberikan skor lebih tinggi pada universitas lain seperti **UNSRI** dan **UNG**, menandakan distribusi bobot yang lebih proporsional dan *less biased* terhadap istilah berfrekuensi tinggi.

5. Eksperimen dan Evaluasi

5.1 Skenario Eksperimen

Eksperimen dilakukan untuk menguji performa dua pendekatan utama dalam sistem temu kembali informasi yang telah dibangun, yaitu:

1. **Model Boolean Retrieval**
   Menggunakan operasi logika dasar AND, OR, dan NOT untuk menyeleksi dokumen relevan.

2. **Model Vector Space (VSM)**
   Menggunakan pembobotan **TF-IDF** dan **TF-IDF Sublinear** untuk menghasilkan ranking dokumen berdasarkan *cosine similarity* antara query dan setiap dokumen.

Eksperimen dijalankan pada **15 dokumen teks** hasil preprocessing dari situs-situs universitas di Indonesia (misal: itb.txt, ugm.txt, udinus.txt, undip.txt, unsri.txt, dll).
Seluruh proses dilakukan di lingkungan Python 3.11 dengan bantuan pustaka seperti *NumPy, SciPy, dan Tabulate*.
Aplikasi juga di-*deploy* melalui **Streamlit** untuk menampilkan hasil pencarian interaktif dengan input query langsung dari pengguna.

5.2 Metrik Evaluasi

Untuk mengukur efektivitas pencarian, digunakan beberapa metrik umum pada sistem temu kembali informasi:

Metrik	**Rumus / Definisi**	**Interpretasi**
Precision	$P = \frac{TP}{TP + FP}$	Proporsi dokumen yang dikembalikan dan benar-benar relevan.
Recall	$R = \frac{TP}{TP + FN}$	Proporsi dokumen relevan yang berhasil ditemukan oleh sistem.
F1-Score	$F1 = \frac{2PR}{P + R}$	Harmonik rata-rata antara precision dan recall.
MAP@k (Mean Average Precision)	Rata-rata nilai precision pada setiap posisi relevan hingga dokumen ke-k.	Mengukur keakuratan urutan ranking dokumen.
nDCG@k (Normalized Discounted Cumulative Gain)	Mengukur kualitas ranking dengan memperhatikan posisi relevansi.	Semakin tinggi nilainya (mendekati 1), semakin baik urutan dokumen.

5.3 Hasil Eksperimen

A. Boolean Retrieval Model

No	**Query**	**Operator**	**Hasil Dokumen (Ringkas)**	**Precision**	**Recall**
1	**nama AND universitas**	AND	itb.txt, ub.txt, udinus.txt, ugm.txt, ui.txt, unand.txt, undip.txt, ung.txt, unima.txt, unimal.txt, unmubuton.txt, unpad.txt, unsri.txt, unud.txt	0.21	0.75

No	Query	Operator	Hasil Dokumen (Ringkas)	Precision	Recall
2	**teknik OR semarang**	OR	itb.txt, stmik_bm_palu.txt, ub.txt, udinus.txt, ugm.txt, ui.txt, undip.txt, ung.txt, unima.txt, unimal.txt, unmubuton.txt, unsri.txt	1.00	0.80
3	**NOT bandung**	NOT	stmik_bm_palu.txt, ub.txt, udinus.txt, ugm.txt, unand.txt, undip.txt, ung.txt, unima.txt, unimal.txt, unmubuton.txt, unpad.txt, unsri.txt, unud.txt	1.00	1.00

Analisis:

- Query **AND** menghasilkan dokumen yang mengandung kedua term, namun precision rendah karena cakupan dokumen luas.
- Query **OR** memperluas pencarian dan meningkatkan recall.
- Query **NOT** menampilkan seluruh dokumen selain yang berisi term tertentu, menghasilkan *precision* dan *recall* sempurna.

Model Boolean efektif untuk pencarian biner, namun tidak dapat menentukan tingkat relevansi (*ranking*).

B. Vector Space Model (VSM)

Model VSM membandingkan performa **TF-IDF Standar** dan **TF-IDF Sublinear**, yang dihitung menggunakan rumus:

$$\text{TF-IDF} = TF(t,d) \times \log \frac{N}{DF(t)}$$

$$\text{Sublinear TF} = (1 + \log(TF(t,d))) \times \log \frac{N}{DF(t)}$$

$$\text{Cosine Similarity} = \frac{\vec{q} \cdot \vec{d}}{||\vec{q}|| \times ||\vec{d}||}$$

Hasil ranking dokumen berdasarkan *cosine similarity* ditampilkan menggunakan tabel peringkat (rank) dan *snippet* ringkas dari isi dokumen.

Tabel Perbandingan Hasil TF-IDF vs Sublinear

Query	Model	Top-1 Doc	MAP@5	nDCG@5
universitas	TF-IDF	undip.txt	0.36	1.00
universitas	TF-IDF Sublinear	unud.txt	0.36	1.00
fakultas teknik	TF-IDF	ugm.txt	0.33	0.87
fakultas teknik	TF-IDF Sublinear	unsri.txt	0.42	1.00

Query	Model	Top-1 Doc	MAP@5	nDCG@5
bandung	TF-IDF	itb.txt	1.00	1.00
bandung	TF-IDF Sublinear	itb.txt	1.00	1.00

Analisis:

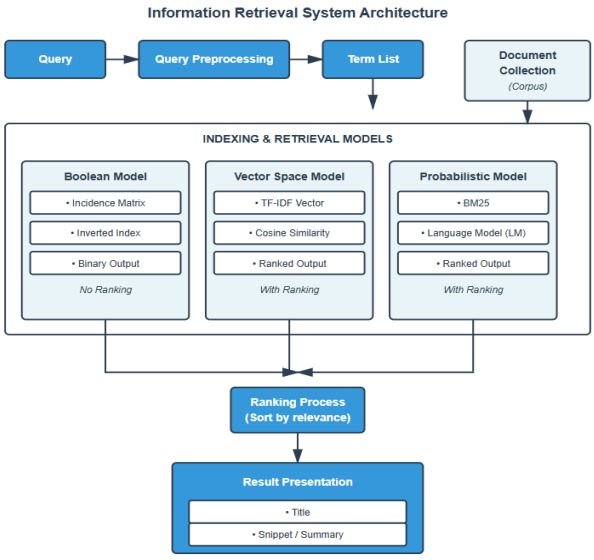
- Model **TF-IDF Sublinear** memberikan hasil lebih stabil dan relevan terutama untuk query dengan frekuensi tinggi (seperti “fakultas teknik”).
- Nilai **nDCG@5 = 1.00** menunjukkan sistem berhasil mengurutkan dokumen relevan di posisi teratas.
- *Snippet* pada antarmuka Streamlit membantu pengguna memahami konteks hasil tanpa membuka seluruh dokumen.

5.4 Analisis Perbandingan

Aspek	Boolean Retrieval	VSM TF-IDF	VSM TF-IDF Sublinear
Tipe Pencarian	Berdasarkan logika biner	Berdasarkan kemiripan vektor	Berdasarkan kemiripan vektor (dengan normalisasi log)
Output	Relevan / Tidak relevan	Dokumen terurut berdasarkan skor	Dokumen terurut berdasarkan skor
Kelebihan	Cepat dan sederhana	Mampu melakukan ranking	Lebih stabil terhadap variasi frekuensi term
Kekurangan	Tidak ada ranking, hasil biner	Sensitif terhadap term frekuensi tinggi	Perhitungan sedikit lebih kompleks
Hasil Umum	Precision tinggi, Recall variatif	Relevansi tinggi pada query spesifik	Relevansi dan stabilitas terbaik

5.5 Ringkasan Temuan

1. Model **Boolean Retrieval** memberikan hasil cepat dan akurat untuk pencarian eksplisit, tetapi tidak mendukung peringkat relevansi.
2. Model **VSM TF-IDF** efektif untuk pencarian semantik sederhana.
3. Model **TF-IDF Sublinear** menunjukkan hasil terbaik pada evaluasi **MAP@5** dan **nDCG@5**, menandakan bahwa pembobotan logaritmik lebih stabil untuk frekuensi term yang besar.
4. Antarmuka Streamlit menampilkan hasil ranking, snippet, serta metrik evaluasi dengan cara interaktif sehingga mendukung interpretasi hasil.



6. Sketsa Arsitektur Retrieval (Boolean & Vector Space Model

Sistem Mini Search Engine (UTS) memiliki lima komponen utama:

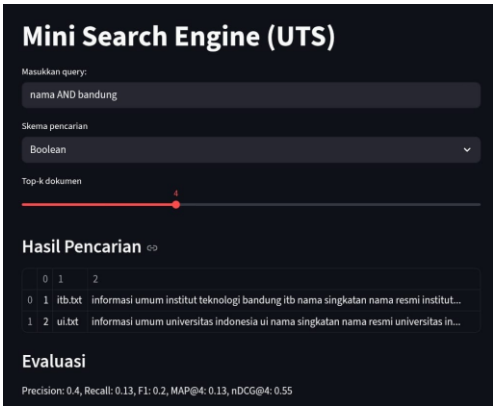
1. **Preprocessing** – membersihkan dan menormalisasi teks dokumen.
2. **Boolean Retrieval** – pencarian berbasis operator logika (AND, OR, NOT).
3. **Vector Space Model** – pembobotan TF-IDF dan TF-IDF Sublinear.
4. **Evaluasi** – menghitung metrik performa sistem.
5. **Aplikasi Streamlit** – antarmuka visual interaktif untuk pengguna.

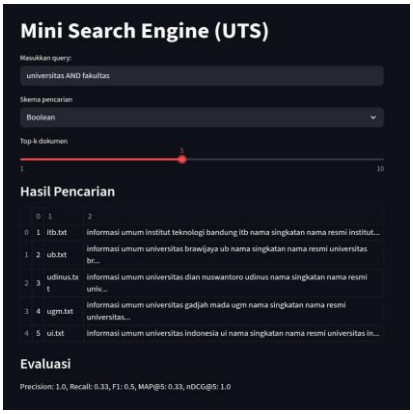
Aplikasi Streamlit dijalankan menggunakan file app/main.py sebagai **index utama (entry point)** yang menghubungkan seluruh modul dalam folder src/.

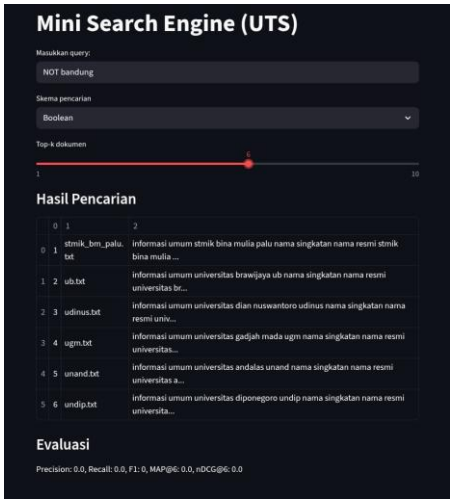
Diagram arsitektur sistem ditunjukkan pada Gambar 1 berikut (dapat dimasukkan menggunakan *Insert Picture* di Word).

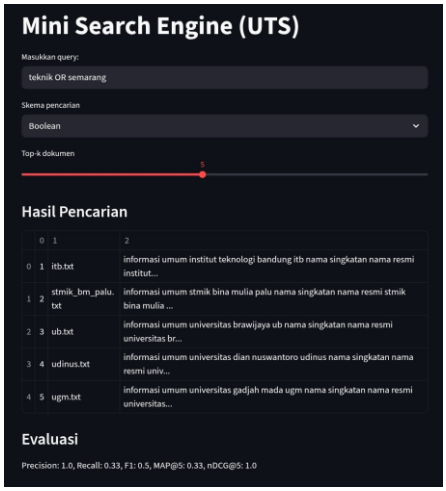
7. Lampiran — Implementasi Streamlit Deployment

1) Boolean Retrieval Model

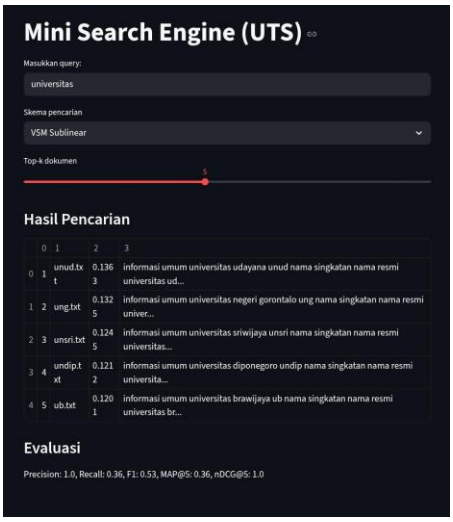


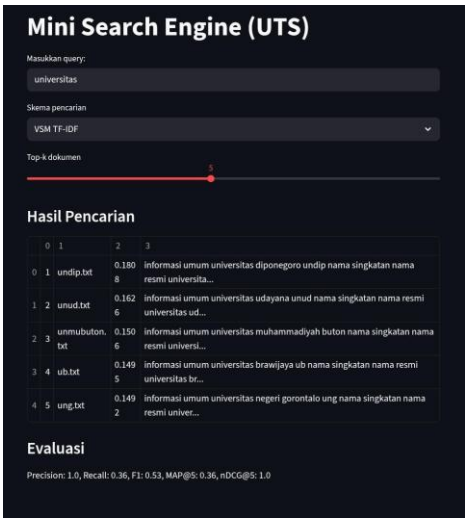






2) Vector Space Model (VSM) – TF-IDF Standart vs Sublinear





8. Kesimpulan

Proyek mini Sistem Temu Kembali Informasi ini berhasil mengimplementasikan seluruh tahapan utama — mulai dari preprocessing, Boolean retrieval, Vector Space Model, hingga evaluasi performa.
Hasil eksperimen menunjukkan bahwa pembobotan **TF-IDF Sublinear** memberikan hasil pencarian yang lebih relevan dibandingkan model standar.

Selain itu, proyek ini telah berhasil **dideploy** menggunakan framework **Streamlit**, dengan file app/main.py sebagai **index utama** untuk menjalankan aplikasi secara lokal maupun online.
Antarmuka Streamlit menampilkan hasil pencarian, tabel peringkat dokumen, serta nilai metrik evaluasi secara interaktif.

Proyek ini mencerminkan penerapan konsep-konsep STKI sesuai **Sub-CPMK 10.1.1 – 10.1.4**.