1. Definisi STKI, Beda Database Retrieval, Peran Index & Ranking

- **Definisi STKI (Sistem Temu Kembali Informasi):**
    - Jelaskan bahwa STKI adalah ilmu tentang pencarian **informasi tidak terstruktur** (seperti dokumen teks, halaman web) yang dapat **memenuhi kebutuhan informasi** pengguna.
    - Tujuannya bukan hanya mengambil data, tetapi menemukan materi yang **relevan** dengan kueri.
- **Perbedaan dengan Database Retrieval (DB):**
    - **DB:** Berurusan dengan **data terstruktur** (tabel, baris, kolom). Kueri bersifat *exact match* (misal, SELECT * FROM mahasiswa WHERE nim='A11.4703'). Hasilnya pasti (ditemukan atau tidak).
    - **STKI:** Berurusan dengan **data tidak terstruktur** (teks bebas). Kueri bersifat *partial match* dan ambigu (misal, "info pmb uin"). Hasilnya diukur dengan **relevansi** dan bersifat tidak pasti.
- **Peran Index (Indeks):**
    - Jelaskan bahwa index adalah inti dari kecepatan STKI.
    - Gunakan **Inverted Index** (yang kita bangun di **Soal 03**) sebagai contoh. Jelaskan bahwa ini adalah struktur data yang memetakan *kata* (term) ke *daftar dokumen* yang mengandung kata tersebut (term $\rightarrow$ [doc1, doc3, doc5]).
    - Tanpa index, sistem harus membaca *setiap* dokumen untuk *setiap* kueri (ini disebut *sequential scan*), yang sangat lambat.
- **Peran Ranking (Peringkat):**
    - Jelaskan bahwa setelah dokumen *ditemukan* (retrieved), dokumen tersebut harus *diurutkan* berdasarkan relevansinya.
    - Inilah peran **Ranking**. Model Boolean (**Soal 03**) hanya memberi hasil *set* (relevan/tidak), sedangkan **Vector Space Model (VSM)** (**Soal 04**) memberi *skor*.
    - Sebutkan **Cosine Similarity** sebagai metode untuk menghitung skor relevansi antara vektor kueri dan vektor dokumen.

---

2. Garis Besar Arsitektur Search Engine Klasik [4]

Jelaskan alur dari kueri hingga hasil[5]. Anda bisa gunakan alur yang kita implementasikan di **Soal 05 (Streamlit App)** sebagai contoh nyata.

1. **Query:** Pengguna mengetikkan kueri (misal: "UIN di Semarang") ke antarmuka (Streamlit).
2. **Preprocess:** Kueri tersebut dibersihkan menggunakan modul preprocess.py (case-folding, stopword removal, stemming) menjadi "uin semarang".
3. **Retrieve:** Sistem menggunakan *index* (Inverted Index atau Matriks TF-IDF) untuk mengambil dokumen yang mengandung *term* "uin" atau "semarang".

4. **Rank:** Sistem menghitung skor untuk setiap dokumen yang diambil. Dalam **VSM (Soal 04)**, ini adalah skor **Cosine Similarity** antara vektor kueri "uin semarang" dan vektor tiap dokumen.
5. **Present:** Sistem menampilkan **Top-K** (misal, k=3) dokumen dengan skor tertinggi kepada pengguna, biasanya dalam bentuk *snippet*.

3. Sketsa Arsitektur & Demo Screenshot

Ini adalah bagian "demo singkat (screenshot) arsitektur IR".

A. Sketsa Arsitektur (Anda bisa gambar ulang ini)

Jelaskan bahwa arsitektur STKI terbagi dua: Indexing Pipeline (dilakukan sekali, offline) dan Query Pipeline (dilakukan setiap ada kueri, online).

Indexing Pipeline (Offline - Persiapan Data)

[5 Dokumen .txt] $\rightarrow$ preprocess.py $\rightarrow$ [5 Dokumen Bersih] $\rightarrow$ vsm_ir.py $\rightarrow$ [Matriks TF-IDF (Index)]

Query Pipeline (Online - Aplikasi Streamlit)

[User Query] $\rightarrow$ preprocess.py $\rightarrow$ VSM.transform(query) $\rightarrow$ Hitung Cosine Sim $\rightarrow$ [Ranking (Skor)] $\rightarrow$ [Hasil Top-K]

B. Demo Singkat (Screenshot)

Untuk bagian ini:

1. Jalankan aplikasi **Streamlit** Anda dari **Soal 05**.
2. Lakukan sebuah pencarian (misal: "UIN Walisongo").
3. **Ambil screenshot** jendela browser yang menampilkan hasil pencarian di aplikasi Streamlit Anda.
4. Masukkan gambar tersebut ke dalam laporan Anda sebagai "Demo Arsitektur IR (Query Pipeline)".

---

4. Peta Materi ke RPS (Kaitan Soal 2 s.d. 5)

Ini adalah bagian penting untuk menunjukkan pemahaman Anda terhadap mata kuliah. Jelaskan bagaimana setiap soal yang Anda kerjakan terhubung dengan Sub-CPMK di Tabel 2.

- **Soal 02 (Document Preprocessing)** secara langsung mengimplementasikan **Sub-CPMK10.1.2** , yaitu "Mampu menjelaskan Document Preprocessing dan tahapannya"[13]. Kita membangun modul preprocess.py untuk tokenisasi, case-folding, stopword removal, dan stemming.
- **Soal 03 (Boolean Retrieval Model)** secara langsung mengimplementasikan **Sub-CPMK10.1.3** , yaitu "Mampu menjelaskan Pemodelan... Boolean Retrieval Model". Kita membangun inverted_index dan memproses kueri AND/OR/NOT.

- **Soal 04 (Vector Space Model & Ranking)** juga mengimplementasikan **Sub-CPMK10.1.3** , yaitu "Mampu menjelaskan Pemodelan... Vector Space Model". Kita membangun matriks TF-IDF dan menghitung ranking menggunakan Cosine Similarity.
- **Soal 05 (Term Weighting, Search Engine, dan Evaluasi)** mencakup **Sub-CPMK10.1.4** , yaitu "Mampu menjelaskan konsep Term Weighting... Search Engine dan Evaluasi Model". Kita membandingkan skema TF-IDF (standar vs sublinear), menyatukannya dalam aplikasi Streamlit (Search Engine), dan mengevaluasinya dengan Precision@k.