1. Data Collection

- **Sumber Dokumen:** PDF publik (Wikipedia bahasa Indonesia)
    - *Ekonomi Indonesia - Wikipedia bahasa Indonesia.pdf*
    - *Indonesia - Wikipedia bahasa Indonesia.pdf*
    - *Sejarah Indonesia - Wikipedia bahasa Indonesia.pdf*

2. Load & Extract Text

- **Script:** `load_docs.py`
- **Proses:**
    1. Baca semua PDF di folder `data/`.
    2. Ekstrak teks per halaman menggunakan PyPDF2.
    3. Simpan sebagai dictionary `{filename: content}`.

3. Preprocessing

- **Script:** `process.py`
- **Langkah-langkah:**
    1. **Cleaning & Normalization**
        - Lowercase
        - Hilangkan angka & karakter spesial
        - Hilangkan whitespace berlebih
    2. **Tokenization & Lemmatization**
        - Tokenisasi teks (nltk)
        - Hilangkan stopwords (bahasa Indonesia + Inggris)
        - Lemmatization
    3. **Chunking**
        - Membagi teks menjadi potongan (chunk) 300–500 kata
        - Overlap antar chunk 50 kata
        - Gunakan `LangChain RecursiveCharacterTextSplitter`
    4. **Simpan** hasil chunk ke folder `processed/`

4. Embedding & Indexing

- **Script:** `build_index.py`
- **Model Embedding:** `sentence-transformers/all-MiniLM-L6-v2`
- **Proses:**
    1. Ambil semua chunk dari dokumen.
    2. Encode setiap chunk menjadi vector embedding.
    3. Buat index FAISS (`IndexFlatL2`) untuk pencarian vektor cepat.
    4. Simpan index (`docs.index`) dan metadata (`metadata.pkl`) di folder `index/`.

5. RAG Chatbot

- **Script:** `rag_chat.py`
- **LLM:** `google/flan-t5-base` (seq2seq)
- **Proses:**
    1. **Retrieve (FAISS)**

    - Encode query user menjadi vector embedding
    - Cari top-k chunk terdekat di index FAISS
2. **Generate (LLM)**
    - Gabungkan teks chunk sebagai konteks
    - Buat prompt
    - Berikan ke LLM (`text2text-generation`) untuk menghasilkan jawaban
3. Output: jawaban + referensi (dokumen asal chunk)