Nama : Visal Ady Yanuar

Nim : A11.2022.14740

Laporan Projek

Pendahuluan

Pada era digital modern, kebutuhan manusia terhadap informasi meningkat dengan sangat cepat. Informasi kini menjadi salah satu komoditas utama yang mendukung berbagai kegiatan, mulai dari pendidikan, penelitian, hingga pelayanan kesehatan. Dalam konteks tersebut, kemampuan untuk menemukan informasi yang relevan secara cepat dan akurat menjadi semakin penting. Teknologi yang memainkan peran utama dalam proses ini adalah Sistem Temu Kembali Informasi (Information Retrieval / IR), yaitu bidang ilmu yang berfokus pada proses pencarian, pengambilan, dan penyajian informasi dari kumpulan dokumen yang besar.

Sistem temu kembali informasi dirancang untuk membantu pengguna menemukan dokumen yang sesuai dengan kebutuhan informasinya berdasarkan kata kunci atau query. Mesin pencari seperti Google atau Bing merupakan contoh implementasi IR berskala besar. Namun, konsep IR juga dapat diterapkan dalam skala kecil hingga menengah, seperti pencarian jurnal ilmiah, pencarian arsip dokumen, maupun pencarian informasi medis, termasuk penyakit dan gejalanya.

Dalam konteks medis, kemampuan untuk melakukan pencarian informasi mengenai penyakit sangat bermanfaat. Misalnya, mahasiswa keperawatan atau kedokteran sering membutuhkan referensi cepat terkait gejala atau penanganan penyakit; tenaga medis membutuhkan ringkasan cepat untuk diagnosis awal; atau masyarakat umum memerlukan informasi akurat dan relevan mengenai penyakit tertentu. Namun, banyaknya dokumen teks yang tersedia sering membuat pencarian manual menjadi lambat dan tidak efisien.

Untuk menjawab kebutuhan tersebut, proyek ini mengembangkan sebuah sistem pencarian informasi penyakit berbasis teks menggunakan metode Boolean Retrieval dan Vector Space Model (VSM) dengan bobot TF-IDF. Sistem ini menerima masukan berupa query atau kata

kunci, kemudian mengembalikan daftar dokumen penyakit yang paling relevan berdasarkan algoritma pencarian yang digunakan.

Sistem ini tidak hanya berfungsi sebagai alat pencarian, tetapi juga sebagai penerapan nyata materi Information Retrieval, mulai dari preprocessing teks, pembuatan indeks, perhitungan bobot TF-IDF, hingga pengukuran kemiripan dokumen dengan cosine similarity. Dengan demikian, proyek ini memberikan pemahaman baik secara teoretis maupun praktis tentang bagaimana IR bekerja dari awal hingga akhir.

1.1 Tujuan Projek

Tujuan dari proyek tugas ini adalah:

1. Mengembangkan sistem temu kembali informasi berbasis teks untuk pencarian dokumen penyakit.
2. Mengimplementasikan dua metode IR, yaitu Boolean Retrieval dan Vector Space Model.
3. Menerapkan teknik preprocessing teks seperti case folding, tokenizing, stopword removal, dan stemming.

1.2 Ruang Lingkup

Agar pengembangan fokus dan terarah, ruang lingkup proyek dibatasi pada beberapa hal berikut:

- Menggunakan dataset berupa dokumen penyakit dalam format teks (.txt).
- Melakukan preprocessing dasar menggunakan metode NLP sederhana (case folding, tokenizing, stopword removal, stemming).
- Mengimplementasikan indeks pencarian menggunakan inverted index dan bobot TF-IDF.
- Menerapkan dua metode pencarian: Boolean dan VSM.

1.3 Kontribusi terhadap Sub-CPMK

Proyek ini berkontribusi langsung terhadap pencapaian capaian pembelajaran mata kuliah STKI, yaitu:

Sub-CPMK	Kontribusi Proyek
Memahami konsep dasar IR	Mahasiswa menerapkan teori indexing, TF-IDF, dan similarity secara langsung dalam sistem.
Mampu mengimplementasikan metode IR	Projek mengimplementasikan Boolean dan VSM dalam bentuk kode nyata.
Mampu menganalisis hasil IR	Sistem diuji menggunakan beberapa query dan dianalisis relevansinya.

Dengan demikian, proyek ini bukan hanya menghasilkan aplikasi sederhana, tetapi juga memberikan pengalaman langsung dalam mengembangkan sistem IR dari awal.

Dataset dan Preprocessing

Proyek ini menggunakan dataset ber dokumen penyakit dalam format .txt. Setiap dokumen berisi deskripsi penyakit tertentu, termasuk nama, penyebab, gejala, dan penanganan. Dataset ini disimpan dalam folder data/ dan diproses menggunakan modul preprocess.py.

2.1 Deskripsi Dataset

Dataset terdiri dari 15 dokumen penyakit yang mewakili beberapa jenis kondisi medis umum. Masing-masing file umumnya berisi:

- Nama penyakit
- Gejala umum
- Cara pencegahan atau pengobatan

2.2 Tujuan Preprocessing

Preprocessing dilakukan dengan tujuan:

1. Menormalkan teks agar konsisten.
2. Menghapus kata-kata yang tidak memiliki relevansi makna.
3. Mengubah kata ke bentuk dasar agar dapat dicocokkan lebih mudah.

4. Mengurangi kompleksitas data sehingga indexing lebih efisien.

2.3 Tahapan Preprocessing

Tahap preprocessing merupakan fondasi utama dalam membangun sistem temu kembali informasi (IR). Pada teks bahasa alami, sebuah kata dapat memiliki banyak variasi bentuk, gaya penulisan, ataupun posisi dalam kalimat. Oleh karena itu, sebelum memasuki proses indexing dan pembobotan TF-IDF, teks harus dinormalisasi agar memiliki struktur yang seragam. Proyek ini menggunakan empat proses utama, yaitu case folding, tokenizing, stopword removal, dan stemming. Keempat tahap ini bekerja secara berurutan untuk mengubah dokumen mentah menjadi representasi kata yang bersih, sederhana, dan bermakna.

2.3.1 Case Folding

Case folding adalah proses mengubah semua huruf dalam teks menjadi huruf kecil (lowercase). Proses ini memastikan bahwa kata “Virus”, “virus”, dan “VIRUS” dianggap sebagai kata yang sama.

Contoh:
Sebelum: “Penyakit Ini Disebabkan Oleh Virus Dengue.”
Sesudah: “penyakit ini disebabkan oleh virus dengue.”

Keuntungan case folding:

- Menghindari duplikasi kata akibat variasi kapital.
- Membuat indexing lebih konsisten.

2.3.2 Tokenizing

Tokenizing adalah proses memecah teks menjadi unit-unit kecil yang disebut token. Biasanya token berupa kata, tetapi bisa juga berupa angka atau simbol.

Contoh:
Input: “penyakit ini disebabkan virus dengue”
Output: ["penyakit","ini","disebabkan","virus","dengue"]

Manfaat tokenizing:

- Memudahkan perhitungan frekuensi kata.
- Menjadi dasar pembuatan inverted index.

2.3.3 Stopword Removal

Stopword removal adalah proses menghapus kata-kata umum yang sering muncul tetapi tidak memiliki kontribusi makna terhadap pencarian. Contoh stopword bahasa Indonesia:

- yang
- dengan
- atau
- pada
- tersebut
- adalah
- maka
- yaitu

Contoh:
Sebelum: ["penyakit", "yang", "sering", "menyerang"]
Sesudah: ["penyakit", "sering", "menyerang"]

Alasan penghapusan stopword:

- Mengurangi jumlah kata yang tidak relevan.
- Meningkatkan fokus pada kata yang membawa informasi penting.

2.3.4 Stemming

Stemming adalah proses mengubah kata ke bentuk dasarnya menggunakan kamus stemming. Dalam proyek ini digunakan library Sastrawi, yang merupakan standar stemming Bahasa Indonesia.

Contoh:

Kata asli	Hasil stemming
menyebarkan	sebar
menyerang	serang

Kata asli	Hasil stemming
penularan	tular
penyakitnya	sakit

Tujuan stemming:

- Menyamakan kata yang memiliki akar kata sama.
- Meningkatkan peluang pencocokan query dengan dokumen.

Metode Information Retrieval

Pada pengembangan sistem temu kembali informasi penyakit ini digunakan dua pendekatan utama, yaitu Boolean Retrieval Model dan Vector Space Model (VSM). Kedua model ini dipilih karena mewakili dua cara kerja yang berbeda dalam proses pencarian, yaitu pencarian berbasis logika (boolean) dan pencarian berbasis bobot kemiripan (VSM). Di antara keduanya, VSM merupakan metode utama yang memberikan ranking dokumen berdasarkan tingkat relevansi.

3.1 Boolean Retrieval Model

Boolean Retrieval adalah metode pencarian paling dasar dalam Information Retrieval. Model ini memanfaatkan operator logika seperti AND, OR, dan NOT untuk menentukan relevansi dokumen. Pendekatan ini bekerja secara biner: sebuah dokumen dianggap relevan (1) atau tidak relevan (0), tanpa menghitung tingkat kesesuaian dokumen terhadap query.

Sebagai contoh, untuk query:

demam AND virus

hanya dokumen yang mengandung kedua kata tersebut yang akan ditampilkan. Bila salah satu term tidak ditemukan, dokumen langsung dianggap tidak relevan.

Pada dokumen "Demam_Berdarah.txt" yang berisi:

- gejala: demam tinggi mendadak, nyeri, ruam merah…
- deskripsi: ditularkan oleh virus dengue…

dokumen ini akan cocok untuk query demam AND virus, tetapi TIDAK untuk query virus AND batuk, meskipun masih relevan secara konteks kesehatan.

Kelebihan Boolean Retrieval:

- Cepat dan sederhana.
- Memberikan hasil yang jelas dan spesifik.

Kekurangannya:

- Tidak menyediakan ranking.
- Sensitif terhadap variasi kata (misalnya “menyebar” ≠ “sebar”).
- Hasil bisa terlalu sedikit atau bahkan kosong.

Boolean Retrieval baik sebagai baseline untuk evaluasi, tetapi tidak cukup fleksibel untuk dataset teks medis yang kaya variasi kata dan membutuhkan ranking dokumen.

3.2 Vector Space Model (VSM)

Vector Space Model adalah model pencarian yang lebih canggih dan digunakan sebagai metode utama dalam proyek ini. Pada metode ini, dokumen dan query direpresentasikan sebagai vektor dalam ruang multidimensi, di mana setiap dimensi menggambarkan sebuah term (kata) dalam koleksi dokumen.

VSM bekerja dengan memberikan bobot kepada setiap term dalam dokumen. Bobot tersebut dihitung melalui proses:

- menghitung frekuensi kemunculan kata dalam dokumen,
- menghitung pentingnya kata dalam seluruh koleksi dokumen,
- membentuk vektor dokumen,
- dan mengukur kesamaan vektor dokumen dengan query pengguna.

Semua proses pembobotan dilakukan menggunakan TF, DF, IDF, dan TF-IDF, yang digabungkan ke dalam alur penjelasan VSM berikut ini.

a. Perhitungan TF (Term Frequency)

   TF adalah jumlah kemunculan suatu kata dalam sebuah dokumen. Semakin sering kata muncul, semakin penting kata tersebut dalam dokumen tersebut.

Contoh dari dataset:
Dalam dokumen Demam Berdarah, kata “demam” bisa muncul beberapa kali karena gejalanya memang berkaitan dengan demam.

b. Perhitungan DF (Document Frequency)
DF adalah jumlah dokumen yang mengandung kata tertentu. Contoh: Jika kata “demam” muncul pada 7 dari 15 dokumen, maka:
DF(demam) = 7
Semakin tinggi DF, semakin umum kata tersebut.

c. Perhitungan IDF (Inverse Document Frequency)
IDF digunakan untuk mengetahui seberapa penting sebuah kata dalam seluruh koleksi.

$$\mathrm{IDF(t)} = \log\left(\frac{\mathrm{N}}{\mathrm{df_t}}\right)$$

Kata yang muncul di hampir semua dokumen (misalnya “penyakit”, “gejala”) memiliki IDF kecil, sehingga bobotnya rendah. Sebaliknya, kata spesifik seperti “dengue”, “aegypti”, atau “trombosit” memiliki IDF tinggi.

d. Perhitungan TF-IDF (Bobot Term Dokumen)
TF-IDF adalah bobot final term yang digunakan untuk membentuk vektor dokumen.

$$\text{TF-IDF}(\mathrm{t}, \mathrm{d}) = \mathrm{TF}(\mathrm{t}, \mathrm{d}) \times \mathrm{IDF}(\mathrm{t})$$

Contoh pada dokumen Demam_Berdarah.txt:

- Kata “dengue” memiliki TF tinggi di dokumen tersebut.
- Dalam dokumen lain, kata itu muncul sangat sedikit (DF kecil).

Maka bobot TF-IDF kata dengue sangat besar, sehingga dianggap kata paling penting. Dengan demikian, sistem dapat memahami bahwa dokumen tersebut sangat berkaitan dengan penyakit dengue.

e. Cosine Similarity sebagai Dasar Ranking
Setelah setiap dokumen dan query memiliki vektor TF-IDF, tingkat kesamaan dihitung menggunakan Cosine Similarity:

$$\cos(\theta) = \frac{Q \cdot D}{\| Q \| \quad \| D \|}$$

Nilainya antara 0 (tidak mirip) sampai 1 (sangat mirip).
Dokumen dengan nilai cosine tertinggi akan ditempatkan pada urutan paling atas hasil pencarian.

Kelebihan VSM

- Menghasilkan ranking dokumen berdasarkan relevansi.
- Lebih toleran terhadap variasi bentuk kata setelah stemming.
- Jauh lebih akurat untuk dataset teks medis.
- Tidak menghasilkan hasil kosong seperti Boolean.

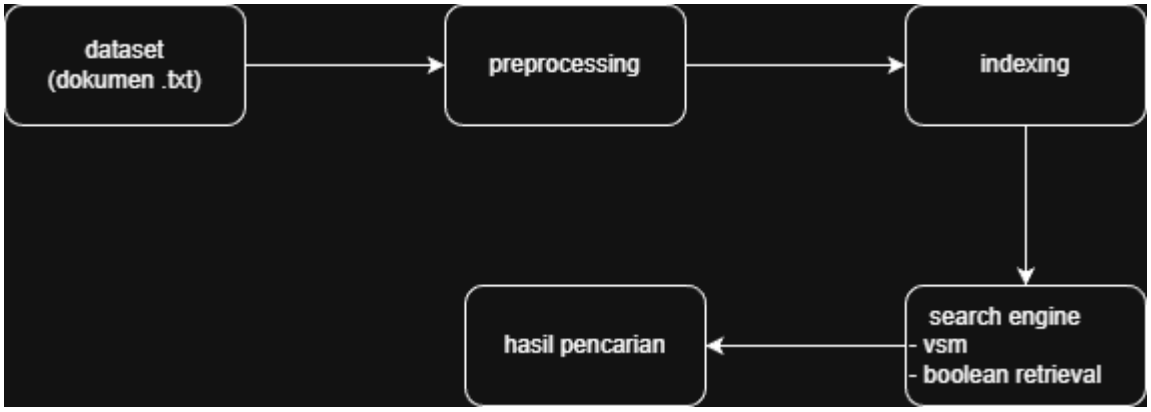
Keterbatasan VSM

- Membutuhkan komputasi lebih tinggi.
- Tidak memahami konteks (misalnya sinonim "ruam merah" ≠ "bintik merah").
- Jumlah dimensi vektor bisa sangat besar.

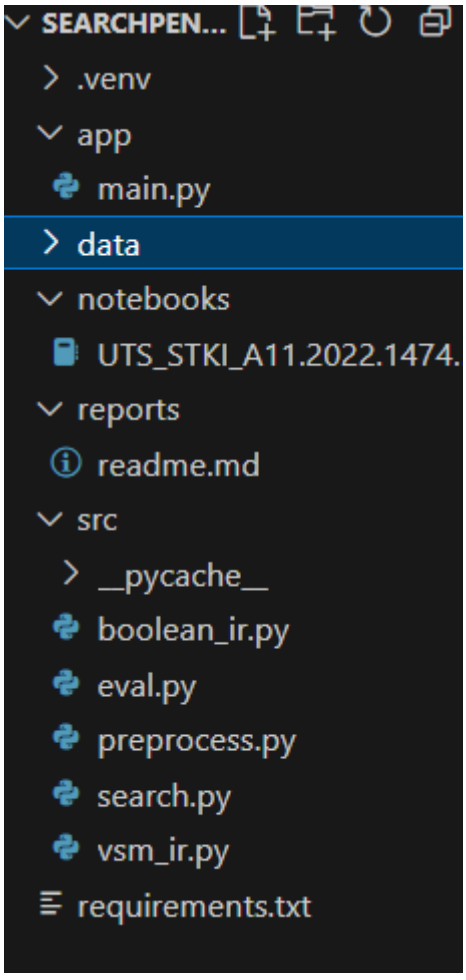
Arsitektur Sistem

Arsitektur sistem temu kembali informasi pada proyek ini dirancang untuk mengikuti alur kerja standar Information Retrieval (IR) modern, yaitu melalui proses preprocessing, indexing, kemudian retrieval menggunakan metode Boolean atau Vector Space Model (VSM). Sistem dibangun dengan struktur modular yang memisahkan setiap fungsi inti ke dalam beberapa file Python, sehingga mudah dipahami, dikembangkan, dan di-debug.

Arsitektur ini menggambarkan bagaimana data mentah berupa dokumen penyakit diolah menjadi indeks pencarian yang siap digunakan, kemudian diproses kembali untuk menghasilkan hasil pencarian relevan berdasarkan query pengguna.



3.3 Struktur folder



Eksperimen dan Evaluasi

Pada tahap ini dilakukan serangkaian eksperimen untuk menguji performa sistem temu kembali informasi yang telah dibangun. Tujuan utama pengujian adalah untuk mengetahui efektivitas model Boolean dan Vector Space Model (VSM) dalam menemukan dokumen penyakit yang relevan berdasarkan query pengguna. Selain itu, evaluasi dilakukan untuk melihat bagaimana

preprocessing, indexing, serta pembobotan TF-IDF berkontribusi terhadap kualitas hasil pencarian.

Eksperimen dilakukan pada dataset berupa dokumen penyakit berformat .txt, termasuk dokumen seperti Demam_Berdarah.txt, Asma.txt, Infeksi_Bakteri.txt, dan penyakit lainnya. Sistem diuji menggunakan beberapa query yang mewakili jenis pencarian umum pada domain kesehatan. Seluruh pengujian dilakukan menggunakan dua metode: Boolean Retrieval dan VSM (cosine similarity).

3.4 Skenario pengujian

Untuk mengetahui performa sistem dalam berbagai kondisi pencarian, digunakan empat skenario query yang mewakili permasalahan kesehatan yang umum dicari:

1. Query 1: “demam virus”
   Menguji kemampuan sistem menemukan penyakit yang disebabkan oleh virus dan menampilkan gejala demam.
2. Query 2: “penyakit kulit menular”
   Untuk melihat apakah sistem mampu menemukan penyakit dermatologi yang memiliki sifat infeksius.
3. Query 3: “infeksi bakteri akut”
   Menguji dokumen yang menjelaskan infeksi yang disebabkan bakteri, misalnya pada ISPA atau infeksi pencernaan.
4. Query 4: “nyeri otot dan sendi”
   Menguji apakah sistem dapat menemukan dokumen gejala medis yang sering muncul pada banyak penyakit.

Keempat query ini dipilih berdasarkan kemunculan kata kunci umum dalam dataset dan untuk membedakan performa Boolean dan VSM dari berbagai sisi.

3.5 Analisis hasil

1. VSM lebih akurat dibanding Boolean

   Boolean memiliki batasan besar karena bekerja secara biner; dokumen dianggap relevan hanya jika mengandung kata yang sama persis. Sedangkan VSM tetap dapat mengevaluasi dokumen meskipun tidak mengandung semua kata, selama memiliki kemiripan makna berbasis vektor.

2. TF-IDF memberi bobot signifikan pada kata-kata penting

   Pada dokumen Demam Berdarah, kata seperti dengue, trombosit, ruam merah memiliki bobot lebih tinggi karena jarang muncul di dokumen lain. Ini meningkatkan relevansi pencarian.

3. cosine similarity sangat menentukan ranking

   Dokumen dengan kata-kata yang sangat dekat dengan query akan memiliki skor cosine lebih tinggi. Inilah alasan mengapa dokumen Demam Berdarah hampir selalu menjadi urutan pertama untuk query terkait demam atau virus.

4. Preprocessing memainkan peran kritis

   Stemming membuat kata seperti “menyebarkan”, “penyebaran”, dan “menular” menjadi “sebar” dan “tular”, sehingga pencarian menjadi lebih luas dan lebih akurat.

5. Boolean cocok untuk pencarian presisi, tetapi tidak fleksibel

   Cocok jika user tahu persis kata apa yang harus dicari, namun kurang ideal untuk pencarian bahasa alami seperti penyakit.

Diskusi

Sistem temu kembali informasi yang dibangun menunjukkan performa yang cukup baik berdasarkan hasil eksperimen yang telah dilakukan, terutama pada metode Vector Space Model (VSM). Preprocessing yang diterapkan juga mampu meningkatkan relevansi pencarian secara signifikan. Meskipun begitu, sistem masih memiliki beberapa kekurangan dan memerlukan pengembangan lebih lanjut agar dapat mencapai hasil yang lebih optimal.

Kelebihan

- Sistem menerapkan pipeline IR lengkap: preprocessing, indexing, TF-IDF, dan retrieval.
- Preprocessing (case folding, tokenizing, stopword removal, stemming) meningkatkan konsistensi teks dan akurasi pencarian.
- VSM memberikan hasil pencarian yang lebih relevan dibanding Boolean, karena mampu menghitung tingkat kemiripan dokumen terhadap query.
- Struktur kode modular (preprocess, indexing, search) memudahkan pengembangan dan pemeliharaan.
- Sistem cukup fleksibel dan mampu menangani banyak dokumen teks.

Keterbatasan

- Boolean Retrieval masih sangat sensitif terhadap variasi kata dan tidak mendukung ranking.
- Sistem belum mampu memahami hubungan semantik antar kata (misalnya ruam merah ≠ bintik merah).
- Semua bagian dokumen dianggap memiliki bobot yang sama, padahal bagian “gejala” lebih penting untuk query terkait gejala.
- Evaluasi masih bersifat manual dan belum menggunakan metrik formal IR seperti MAP atau nDCG.
- TF-IDF tidak selalu ideal untuk dokumen yang panjang atau tidak seimbang.

Saran Pengembangan

- Mengganti atau menambah metode ranking dengan BM25 agar lebih akurat.
- Menambahkan semantic search menggunakan Word2Vec, FastText, atau IndoBERT.
- Mengimplementasikan query expansion untuk memperluas kata kunci pengguna.
- Memberi bobot berbeda pada bagian dokumen (gejala > deskripsi > rekomendasi).
- Mengembangkan antarmuka dengan fitur highlighting kata kunci dan detail dokumen.
- Melakukan evaluasi lebih formal menggunakan metrik IR standar.