

BAB II TINJAUAN PUSTAKA

A. Penelitian Terkait

Penelitian terkait merupakan kajian dari penelitian sebelumnya yang dapat digunakan sebagai inspirasi dan referensi yang berkaitan dengan latar belakang. Berikut ini penelitian sebelumnya yang relevan dengan penelitian adalah :

Tabel 1 Daftar Penelitian

No	Peneliti	Keterangan Penelitian
(1)	(2)	(3)
1	Selvia Ferdiana Kusuma, Tessy Badriyah, Prasetyo Wibowo, Rosiyah Faradisa, Solichul Huda, 2023[4]	Judul : Optimalisasi Fitur Pencarian Pada E-Catalog Menggunakan Query Expansion dan Algoritma TF-IDF Metode : Query Expansion dan Algoritma TF-IDF Software : Hasil Penelitian : Penelitian ini menghasilkan sebuah <i>e-catalog</i> yang mengoptimalkan fitur pencarian menggunakan query expansion dan algoritma TF-IDF. Hasil efektifitas penggunaan query expansion dan pembobotan TF-IDF menunjukkan hasil yang positif. Proses pencarian menghasilkan produk yang lebih spesifik dibandingkan tanpa menggunakan query expansion dan pembobotan TF-IDF.
2	Salsabila Nida Cahyani, Galuh Wilujeng Saraswati, 2023[5]	Judul : Implementation Of Support Vector Machine Method In Classifying School Library Books With Combination Of TF-IDF And Word2Vec Metode : Support Vector Machine Software : Python Hasil Penelitian : Setelah dilakukan pengujian algoritma Support Vector Machine (SVM)

		sebanyak 3 kali dengan menggunakan TF-IDF saja, Word2Vecsaja dan kombinasi TF-IDF dan Word2Vec, dapat disimpulkan bahwa algoritma SVM cocok untuk digunakan pada sistem klasifikasi judul buku otomatis. Nilai akurasi tertinggi yaitu 73% terdapat pada metode klasifikasi SVM dengan menggunakan kombinasi ekstraksi fitur TF-IDF dan Word2Vec dibandingkan dengan metode ekstraksi fitur lainnya, dimana memiliki nilai precision83%, recall72%, dan F1-Scoresebesar 76%.
3	Roni Ardianzah; Husni Thamrin, S.T., M.T., Ph.D. , 2024[6]	Judul : Pengembangan Sistem Pencarian Pada Aplikasi Skripsi Untuk Meningkatkan Hasil Pencarian Judul Metode : Waterfall Software : framework Django, PostGreSQL Hasil Penelitian : Sistem mampu menampilkan hasil kemiripan judul antara judul mahasiswa dengan data judul. Pengujian dengan black box dan System Usability Scale (SUS) menunjukkan nilai rata-rata sebesar 77,17 membuktikan bahwa sistem layak diterima oleh pengguna.
4	Dian Asmarajati, 2020[8]	Judul : Analisis Perbandingan Algoritma Tf-Idf Dengan Sql Query Untuk Kasus Pencarian Pada Sistem Informasi Dokumentasi Arsip (SIDOKAR) Metode : SQL Query dan TF-IDF Software : Hasil Penelitian : Hasil yang diperoleh dari penelitian ini adalah dalam hal kecepatan proses atau running time SQL query lebih efisien dari pada Algoritma TF-IDF sedangkan dalam hal

		kesesuaian hasil pencarian data relevan Algoritma TF-IDF lebih efektif dari pada SQL query.
5	Aditya Irvandani, 2020[10]	Judul : Sistem Rekomendasi Pemilihan Fotografer Dengan Metode Haversine Dan Tf-Idf Di Malang Raya Metode : Haversine, TF-IDF Software : HTML5, CSS, PHP versi 4, MySQL dan CodeIgniter3. Hasil Penelitian : Didapatkan hasil yang sama antara perhitungan manual dengan perhitungan pada system. .Pada pengujian <i>user</i> dapat disimpulkan bahwa system rekomendasi pemilihan fotografer mampu di implementasikan dan digunakan oleh masyarakat umum dengan prosentase menyatakan baik 78%, prosentase menyatakan cukup 21% dan prosentase menyatakan kurang 1%.
6	Nur Ani, Desi Yosephine Sinaga, Nickolas Junior, Muhamad Doni Munggaran, 2023[11]	Judul : Penerapan Algoritma Term Frequency-Inverse Document Frequency(TF IDF) Untuk Fitur Pencarian Dokumen Standar Nasional Indonesia Metode : Prototype, TF-IDF Software : Python Hasil Penelitian : Website pencarian SNI memudahkan proses pencarian dengan pemanfaatan algoritma TF-IDF memberikan manfaat dimana <i>user</i> dapat melakukan pencarian dengan kata kunci tanpa harus mengingat detail dari SNI yang akan dicari. Website pencarian SNI dibuat dengan fitur baru yaitu menu berita, dimana <i>user</i> dapat memperbaharui informasi mereka dari sumber berita yang terpercaya. Penggunaan kata kunci yang merujuk pada standar yang akan dicari

		lebih efektif penggunaannya karena kemampuan manusia dalam mengikat kata lebih mudah dibandingkan dengan angka. Kata kunci yang menggambarkan apa yang dicari oleh user juga akan menampilkan hasil <i>score</i> TF-IDF dimana semakin tinggi skor TF-IDFnya maka semakin sesuai dengan apa yang dicari oleh pengguna.
7	Karina Djunaidi, Rahma Farah Ningrum, Dine Tiara Kusuma, Khesya Medhyna Saradiba, 2024[12]	Judul : Pencarian Fungsional Obat Menggunakan Algoritma Tf-Idf dan Cosine Similarity Metode : Tf-Idf dan Cosine Similarity Software : Hasil Penelitian : Penelitian ini menggunakan data sejumlah 425 data obat yang diperoleh dari web-scraping pada website Ai-Care. Dari sejumlah data tersebut data yang di uji cobakan hanya 10 data indikasi obat yang diproses melalui text preprocessing. Setelah diperoleh hasil dari text preprocessing kemudian diolah dengan TF/IDF dan akan di evaluasi menggunakan cosine similarity
8	Herlina Sari, Guidio Leonardo Ginting, Taronishoki Zebua, Mesran, 2021[13]	Judul : Penerapan Algoritma Text Mining dan TF-IDF Untuk Pengelompokkan Topik Skripsi Pada Aplikasi Repository STIMIK Budi Darma Metode : Text Mining, TF-IDF, dan Cosine Similarity Software : Hasil Penelitian : Dari berbagai penjelasan yang telah diuraikan dalam laporan, maka dapat disimpulkan berbagai hal Algoritma text mining dan TF-IDF mampu melakukan klasifikasi skripsi. Adanya proses cosine similarity dikarenakan dibutuhkan untuk membandingkan antar abstrak. Semakin banyak keyword yang terdapat pada

		abstrak maka semakin baik untuk mengelompokkan.
9	Novi Prisma Yunita, 2023[14]	Judul : Aplikasi Pencarian Hadis Menggunakan Vector Space Model Dengan Pembobotan TF-IDF Dan Confix-Stripping Stemmer Metode : TF-IDF Software : Hasil Penelitian : Penelitian ini bertujuan membangun aplikasi pencarian hadis yang dapat digunakan untuk menemukan hasil pencarian yang relevan dengan <i>keyword</i> pencarian. Aplikasi pencarian dibangun menggunakan metode Information Retrieval antara lain VSM dan TF-IDF untuk membangun ruang vector dan pembobotan term, dan stemming menggunakan algoritma Confix-Stripping Stemmer (CS Stemmer)

Dari referensi jurnal ilmiah yang dikumpulkan ada beberapa hal yang sama pada penelitian sebelumnya yaitu pada penelitian ini menggunakan Algoritma Term Frequency-Inverse Document Frequency (TF-IDF) untuk menghitung bobot kata kunci sehingga menampilkan hasil yang relevan. Pada penelitian ini hal yang membedakan pada penelitian terdahulu yaitu menggunakan metode waterfall dan evaluasi menggunakan akurasi, presisi dan recall.

Diagram *mind map* dicontohkan pada gambar berikut ini :



Gambar 1 Mind Map

B. Landasan Teori

1. Katalog Elektronik

Katalog Elektronik merupakan sebuah sistem informasi yang memuat produk atau jasa, katalog elektronik digunakan untuk mempublikasikan dan mempromosikan sebuah produk atau layanan secara online kepada berbagai pihak[4]. Dengan adanya katalog elektronik calon pelanggan dapat mengenal dan mempelajari sebuah produk atau jasa. Katalog Elektronik memungkinkan semua orang dapat mengakses kapan saja dan dimana saja.

2. Algoritma TF-IDF

Algoritma TF-IDF (Term Frequency- Inverse Document Frequency) merupakan metode statistik numerik yang menciptakan seberapa pentingnya sebuah kata dalam sebuah dokumen korpus. Algoritma TF-IDF menggabungkan dua konsep, yaitu Term Frequency (TF) dan Inverse Document Frequency (IDF). TF mengukur frekuensi kemunculan sebuah kata dalam sebuah dokumen, semakin sering frekuensi kata muncul maka nilai semakin besar, sedangkan IDF mengukur kebalikan frekuensi dokumen yang mengandung kata tersebut[15]. Algoritma ini memiliki tujuan untuk mengidentifikasi kata kunci yang paling relevan dalam sebuah kalimat

Berikut ini perhitungan TF-IDF :

$$tf_{t,d} = \frac{\text{jumlah kemunculan kata}}{\text{jumlah kata dalam dokumen}}$$

$$idf_t = \log\left(\frac{D}{df_t}\right)$$

$$W_{d,t} = tf_{t,d} \times idf_t$$

Keterangan :

$W_{d,t}$	= Nilai bobot dokumen ke-d terhadap kata ke-t
$tf_{t,d}$	= frekuensi kemunculan kata ke-t pada dokumen ke-d
D	= jumlah keseluruhan dokumen
df_t	= jumlah dokumen yang mengandung kata ke-t
idf_t	= nilai <i>inverse document frequency</i> kata ke-t

3. React JS

React JS adalah library javascript yang digunakan untuk membangun desain antarmuka atau user interface yang interaktif. React JS dikembangkan oleh facebook yang di buat oleh Jordan Walke pada tahun 2013, disebut react karena fitur utamanya adalah reactive yang berarti memiliki kemampuan untuk bereaksi atau merespon secara dinamis ketika ada interaksi atau perubahan data. React memungkinkan pengembang untuk membuat komponen UI yang dapat digunakan kembali, sehingga lebih mudah untuk mengelola dan memelihara aplikasi web yang kompleks. Berikut ini fitur fitur dalam React JS:

a. Javascript XML (JSX)

Fitur ini merupakan ekstensi syntax JavaScript yang digunakan dalam pembuatan elemen React. Developer menggunakan untuk menyematkan kode HTML pada objek JavaScript, JSX memungkinkan pengembang untuk menggabungkan logika JavaScript dengan tampilan HTML secara lebih mudah dan intuitif

b. DOM (Document Object Model) Virtual

Fitur ini berfungsi untuk menyajikan halaman web dalam tampilan struktur data , React JS menyimpan struktur data DOM Virtual dalam memorinya, sehingga jika terdapat perubahan pada bagian tertentu dalam struktur data tersebut tidak perlu merender ulang semua datanya.

c. Komponen dan Properti React

React JS memungkinkan pengembang untuk memisahkan antarmuka pengguna menjadi komponen-komponen kecil yang dapat digunakan kembali. Setiap komponen memiliki logika dan tampilan tersendiri, sehingga memudahkan pengembangan dan pemeliharaan kode. Elemen React yang dihasilkan akan menentukan bagaimana tampilan antarmuka pengguna yang dilihat oleh user.

d. Manajemen State

State atau keadaan merupakan objek JavaScript yang mewakili satu bagian dari sebuah komponen. Library manajemen state memiliki peran penting dalam memfasilitasi komunikasi dan berbagi data antara komponen-komponen React.

e. Navigasi Terprogram

Navigasi terprogram adalah suatu situasi di mana serangkaian kode menghasilkan sebuah Tindakan yang mengarahkan pengguna ke area tertentu.

4. Waterfall

Metode Waterfall adalah suatu kerangka kerja yang digunakan untuk menstrukturkan, merencanakan, dan mengendalikan proses pengembangan suatu sistem informasi. Disebut waterfall karena model pengembangannya dianalogikan seperti air terjun, dimana tiap tahapannya dikerjakan secara berurutan dari atas ke bawah[16]. Tahapan pada metode waterfall yaitu :

a. Analisis Kebutuhan

Tahap ini adalah tahapan untuk menganalisa kebutuhan perangkat lunak sehingga dapat dipahami kebutuhan pengguna.

b. Desain dan Perancangan

Pada tahap ini penulis merancang desain dan pembuatan program dengan UML (Unified Modeling Language)

c. Kode Program

Desain ini harus diubah ke dalam bentuk program perangkat lunak. Hasil dari tahapan ini diharapkan program sesuai dengan desain yang sudah dibuat.

d. Pengujian (Testing)

Pengujian fokus pada perangkat lunak dari segi logic dan fungsional serta memastikan semua bagian sudah di uji sehingga keluaran yang dihasilkan sesuai dengan yang diharapkan.

Metode waterfall ini memiliki kelebihan dan kekurangan, berikut ini kelebihan yang dimilikinya :

- a. Workflow yang jelas
- b. Hasil dokumentasi yang baik
- c. Dapat menghemat biaya
- d. Digunakan untuk pengembangan software berskala besar

Adapun kelemahan dari metode waterfall ini adalah sebagai berikut :

- a. Membutuhkan tim yang solid
- b. Masih kurang fleksibilitas
- c. Tidak dapat melihat gambaran sistem dengan jelas
- d. Butuh waktu yang lebih lama

5. Black Box Testing

Black Box Testing merupakan metode pengujian perangkat lunak yang menguji fungsionalitas dari sebuah software . Jenis pengujian ini tidak memerlukan pengetahuan khusus terkait kode program dari software atau aplikasi yang sedang di uji. Black Box Testing memiliki beberapa kelebihan, seperti :

- a. Untuk menguji suatu aplikasi, penguji tidak perlu memiliki pemahaman terkait pemrograman
- b. Penguji tidak perlu memeriksa kode program
- c. Memungkinkan penguji dan developer bekerja sama secara independent tanpa mengganggu proses kerja satu sama lain
- d. Guna menemukan inkonsistensi program, pengujian dilakukan berdasarkan sudut pandang pengguna
- e. Dapat mengidentifikasi kekurangan atau bug program pada tahap awal pengujian

Adapun kekurangan dari pengujian ini adalah sebagai berikut :

- a. Penguji yang tidak memiliki pengetahuan teknis terkait pemrograman memiliki kemungkinan melewatkan skenario pengujian tertentu yang dapat menyebabkan bug tidak teridentifikasi
- b. Adanya kemungkinan pengujian tidak dilakukan dengan teliti karena deadline pengerjaan yang sangat singkat
- c. Tidak memungkinkan pengujian secara menyeluruh pada aplikasi skala besar dan kompleks