

## **BAB II**

### **KAJIAN PUSTAKA**

#### **A. Kajian Teoritis**

##### **A. Konsep Dasar Sistem**

Sistem dapat didefinisikan sebagai sekumpulan elemen atau komponen yang saling terkait dan bekerja sama secara harmonis untuk mencapai tujuan bersama. Berbagai ahli mendukung pandangan ini. Misalnya, (Maydianto & Rasid Ridho, 2021) menjelaskan bahwa sistem terbentuk dari gabungan elemen, komponen, atau variabel yang terhubung dan beroperasi selaras demi mencapai tujuan tertentu. Keterkaitan antar komponen ini menciptakan kesatuan yang utuh, yang kemudian dikenal sebagai sistem. Pendapat serupa diutarakan oleh (Effendi et al., 2023), yang mendefinisikan sistem sebagai gagasan dari elemen-elemen yang berinteraksi secara teratur untuk mencapai tujuan spesifik.

Berdasarkan pendapat-pendapat di atas, maka dapat disimpulkan bahwa sistem adalah entitas yang terdiri dari bagian-bagian yang saling berhubungan, berfungsi secara terkoordinasi, dan membentuk kesatuan yang kohesif demi mencapai tujuan yang telah ditetapkan.

##### **B. Sistem Rekomendasi**

Sistem rekomendasi merupakan sebuah program atau sistem yang tujuannya untuk menyaring informasi sebagai solusi atas permasalahan banyaknya informasi (*information overload*), dengan cara menyeleksi informasi yang dianggap penting dan relevan berdasarkan preferensi, minat,

serta kesukaan pengguna terhadap suatu konten atau item. (Fajriansyah et al., 2021). Pendapat lain yang dikemukakan oleh Faurina & Evlin, (2023). menjelaskan bahwa Sistem rekomendasi dapat didefinisikan sebagai alat atau sistem pendukung yang dapat membuat pengguna untuk menemukan atau mencari suatu informasi, produk, dan layanan dengan menggabungkan dan menganalisis preferensi dan saran dari pengguna lainnya.

Dari pendapat-pendapat di atas, dapat disimpulkan bahwa sistem rekomendasi merupakan suatu sistem cerdas yang dirancang untuk membantu pengguna dalam memilih informasi yang sesuai dengan kebutuhannya melalui proses penyaringan berbasis preferensi, minat, atau interaksi sebelumnya, serta dapat menggabungkan umpan balik dari pengguna lain untuk menghasilkan rekomendasi yang lebih akurat dan personal.

### *1.1. Collaborative Filtering*

*Collaborative Filtering* merupakan sebuah metode dalam membuat suatu prediksi dengan cara memfilter informasi item dari pendapat orang lain. Ide utama dari metode ini adalah memanfaatkan opini orang lain sebelumnya untuk memperkirakan item yang mungkin akan disukai atau diminati oleh seorang pengguna (Hartatik et al., 2021). (Prasatya & Bangsa, 2023) menyatakan bahwa Metode *Collaborative Filtering* ini berupa kumpulan dataset berdasarkan kemiripan karakteristik konsumen dan memberikan informasi berdasarkan kecenderungan yang serupa di antara kelompok konsumen,

sehingga memberikan informasi baru kepada konsumen. Perbedaan preferensi di antara anggota kelompok menjadi sumber informasi baru yang bermanfaat bagi anggota kelompok lainnya. Keunggulan teknik *collaborative filtering* adalah kemampuannya untuk menganalisis riwayat pilihan pengguna, sehingga meningkatkan akurasi rekomendasi yang dihasilkan.

$$S(x, y) = \sum_{z \in N_y} R(z, y) \times Si(x, z) / \sum_{z \in N_y} |Si(x, z)|$$

Sumber:(Billah et al., 2021).

Gambar 2. 1 Rumus *Collaborative Filtering*

Dimana:

$S(x,y)$  = Nilai rekomendasi *user* ( $x$ ) terhadap *item* ( $y$ )

$R(z,y)$  = Besar rating atau tingkat kesukaan *user* ( $z$ ) terhadap *item* ( $y$ )

$Si(x,z)$  = Nilai *similarity* antara *user* ( $x$ ) dan *user* ( $z$ )

Dari pernyataan di atas dapat disimpulkan bahwa metode *Collaborative Filtering* memanfaatkan informasi dari riwayat opini pengguna untuk memberikan prediksi terkait item yang mungkin disukai oleh pengguna lain dengan karakteristik serupa. Metode ini tidak hanya mengandalkan data individu, tetapi juga memperhatikan perbedaan preferensi dalam kelompok pengguna, yang memungkinkan peningkatan akurasi rekomendasi. Keunggulan utama dari teknik ini adalah kemampuannya untuk mengidentifikasi pola dari pilihan

pengguna, sehingga memberikan rekomendasi yang lebih relevan dan *personalized*.

### 1.2. *Content Based Filtering*

*Content Based Filtering* merupakan salah satu metode dalam sistem rekomendasi yang bekerja dengan menganalisis preferensi pengguna secara individual, tanpa melibatkan data atau aktivitas pengguna lain. Metode ini hanya berfokus pada data pengguna itu sendiri, seperti riwayat interaksi atau item yang pernah disukai, untuk menghasilkan rekomendasi item yang memiliki kemiripan karakteristik. (Larasati & Februariyanti, 2021). Putri et al., (2023) menjelaskan bahwa *Content Based Filtering* adalah metode rekomendasi yang bekerja dengan menganalisis atribut atau karakteristik dari suatu item, seperti spesifikasi atau deskripsi, untuk mengukur kesamaan antar item. Dengan pendekatan ini, sistem dapat merekomendasikan item yang mirip dengan apa yang pernah disukai atau dipilih oleh pengguna sebelumnya. Inti dari metode ini adalah fokus pada informasi dari item itu sendiri, bukan berdasarkan pendapat atau perilaku pengguna lain.

Dari pernyataan di atas dapat disimpulkan bahwa *Content Based Filtering* merupakan metode rekomendasi yang bersifat personal, karena hanya mempertimbangkan data dan preferensi dari pengguna itu sendiri tanpa melibatkan informasi dari pengguna lain. Metode ini bekerja dengan menganalisis atribut atau karakteristik item yang pernah

disukai oleh pengguna, lalu mencari item lain yang memiliki kemiripan. Dengan demikian, sistem dapat memberikan rekomendasi yang relevan berdasarkan kesesuaian antara preferensi pengguna dan fitur dari item yang tersedia.

### 1.3. *Term Frequency - Inverse Document Frequency*

Sebagai teknik penting dalam pemrosesan teks dan pemodelan bahasa alami, *Term Frequency – Inverse Document Frequency* (TF-IDF) utamanya bertujuan untuk mengukur tingkat kepentingan sebuah kata (*term*) di dalam suatu dokumen, terutama saat bekerja dengan banyak dokumen. (Septiani & Isabela, 2022). Dalam metode TF-IDF nilai *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) dikalikan untuk memperoleh bobot atau nilai penting suatu kata dalam sebuah dokumen. Bobot ini menunjukkan seberapa signifikan kata tersebut dalam dokumen tertentu jika dibandingkan dengan kumpulan dokumen. Dengan kata lain, semakin sering kata muncul dalam satu dokumen namun jarang muncul di dokumen lain, maka bobotnya akan semakin tinggi, sehingga dianggap lebih penting dalam konteks dokumen tersebut.

$$tf = 0,5 + 0,5 \times \frac{tf}{\max(tf)}$$

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

$$W_{d.t} = tf_{d.t} \times idf_{d.t}$$

Sumber: (Septiani & Isabela, 2022).

Gambar 2. 2 Rumus *Term Frequency - Inverse Document Frequency*

Dalam perhitungan TF-IDF, terdapat beberapa komponen penting yang perlu dipahami. Simbol  $D$  mengacu pada dokumen ke- $d$  dalam kumpulan dokumen, sedangkan  $t$  menunjukkan *term* atau kata ke- $t$  yang terdapat dalam dokumen tersebut. Nilai  $W$  merupakan bobot dari term  $i$  terhadap dokumen  $D$ , yang menunjukkan tingkat kepentingan *term* tersebut dalam dokumen.  $tf$  (*term frequency*) adalah jumlah kemunculan suatu term dalam dokumen tertentu, yang mencerminkan seberapa sering kata tersebut digunakan dalam dokumen tersebut. Sementara itu,  $idf$  (*inverse document frequency*) merupakan ukuran yang menunjukkan seberapa jarang suatu term muncul di seluruh dokumen dalam koleksi. Komponen  $df$  (*document frequency*) adalah jumlah dokumen yang mengandung term tertentu. Semakin kecil nilai  $df$ , maka nilai  $idf$  semakin tinggi, yang berarti term tersebut lebih unik dan penting dalam konteks dokumen tertentu.

Nouvalina & Hati, (2024) menjelaskan bahwa metode TF-IDF merupakan pendekatan yang menggabungkan dua konsep utama, yaitu frekuensi kemunculan suatu kata dalam dokumen tertentu (*Term Frequency/TF*) dan kebalikannya, yakni seberapa jarang kata tersebut muncul di seluruh dokumen (*Inverse Document Frequency/IDF*). Gabungan kedua nilai ini digunakan untuk menghitung bobot kata, yang mencerminkan seberapa penting kata tersebut dalam konteks dokumen tertentu dibandingkan dengan kumpulan dokumen lainnya.

Dari pernyataan di atas dapat disimpulkan bahwa metode TF-IDF merupakan teknik penting dalam pengolahan teks yang bertujuan untuk menilai tingkat kepentingan suatu kata dalam sebuah dokumen dibandingkan dengan seluruh koleksi dokumen. Metode ini menggabungkan dua komponen, yaitu *Term Frequency* (TF) yang menunjukkan seberapa sering kata muncul dalam satu dokumen, dan *Inverse Document Frequency* (IDF) yang mengukur seberapa jarang kata tersebut muncul di keseluruhan dokumen. Hasil perkalian keduanya menghasilkan bobot kata yang tinggi untuk kata-kata yang sering muncul dalam satu dokumen namun jarang ditemukan di dokumen lain, sehingga dianggap lebih bermakna atau relevan dalam konteks dokumen tersebut. TF-IDF banyak digunakan dalam sistem rekomendasi berbasis konten karena mampu mengekstraksi informasi penting dari data teks secara efisien.

### C. Website

Website adalah salah satu bentuk kemajuan teknologi yang berfungsi sebagai media untuk menyajikan berbagai jenis informasi, seperti teks, gambar, suara, dokumen, hingga video. Salah satu keunggulan utama dari website adalah kemampuannya untuk saling menghubungkan antar halaman atau dokumen melalui *hyperlink*, sehingga memudahkan pengguna dalam mengakses informasi secara terstruktur dan interaktif. Website juga menjadi sarana penting dalam penyebaran informasi secara luas dan real-time di era digital saat ini.

### 3.1. Framework

Selama proses pengembangan web, alat bantu yang berbentuk kerangka kerja dikenal sebagai *framework*. Fokus utama penggunaan *framework* adalah untuk memudahkan pekerjaan pengembang web (*web developer*) dalam menulis kode program dengan cepat. Jika *framework* memiliki struktur dan aturan yang jelas, pengembang tidak perlu membangun aplikasi dari awal secara manual. Dengan menawarkan komponen dasar yang dapat digunakan kembali, *framework* membuat proses pembuatan web lebih efisien, terorganisir, dan konsisten. (Nasution et al., 2023).

*Framework* merupakan seperangkat kode program yang memiliki struktur kerangka kerja sistematis dan dirancang untuk membantu pengembang dalam membangun aplikasi, baik berbasis web, mobile, maupun desktop. Di dalamnya terdapat berbagai perintah dan fungsi dasar yang bersifat umum, yang dapat digunakan kembali dalam proses pengembangan aplikasi. Dengan memanfaatkan *framework*, proses pembuatan aplikasi menjadi lebih cepat, efisien, serta menghasilkan struktur kode yang rapi dan teratur. (Kansha et al., 2023).

Dari pernyataan di atas dapat disimpulkan bahwa *framework* merupakan alat bantu berupa perangkat lunak yang menyediakan struktur kerja terorganisir dan komponen-komponen dasar yang dapat digunakan kembali dalam pengembangan aplikasi. *Framework* tidak hanya mempercepat proses pembuatan aplikasi, tetapi juga memastikan

bahwa kode yang ditulis oleh pengembang menjadi lebih rapi, terstruktur, dan mudah dipelihara. Baik dalam pengembangan aplikasi web, mobile, maupun desktop, *framework* berperan penting dalam meningkatkan efisiensi kerja dan konsistensi sistem yang dibangun. Dengan adanya aturan dan pola yang sudah ditetapkan di dalam *framework*, pengembang tidak perlu memulai dari nol, sehingga waktu dan tenaga dapat lebih dihemat serta risiko kesalahan dalam pengkodean dapat diminimalkan.

### 3.2. Laravel

Laravel merupakan *framework* web yang bersifat open-source dan dibangun menggunakan Bahasa Pemrograman PHP. *Framework* ini dikembangkan oleh Taylor Otwell dengan tujuan untuk memudahkan proses pengembangan aplikasi website yang rapi, efisien, dan terstruktur. Laravel menerapkan arsitektur *Model-View-Controller* (MVC), yang memisahkan logika aplikasi, antarmuka pengguna, dan pengelolaan data, sehingga memudahkan pengembang dalam membangun serta mengelola aplikasi berskala kecil maupun besar. (Yuniarti et al., 2022).

Sweetania & Herawati, (2022) menyebutkan bahwa Laravel merupakan *framework* yang dirancang untuk mengoptimalkan penggunaan bahasa pemrograman PHP dalam pengembangan aplikasi website. Laravel mengikuti pola arsitektur *Model-View-Controller* (MVC), yang memisahkan logika aplikasi dari antarmuka pengguna.

Pendekatan ini membuat aplikasi menjadi tiga komponen utama: Model yang menangani manipulasi data, *View* sebagai antarmuka pengguna, dan *Controller* yang mengatur alur kerja aplikasi. Dengan struktur ini, proses pengembangan menjadi lebih terorganisir, efisien, dan mudah untuk dilakukan pemeliharaan maupun pengembangan lebih lanjut.

Dari pernyataan di atas dapat disimpulkan bahwa Laravel merupakan *framework open-source* berbasis PHP yang dirancang untuk mempermudah dan mempercepat proses pengembangan aplikasi web. Dengan menerapkan arsitektur *Model-View-Controller* (MVC), Laravel memisahkan logika bisnis, tampilan, dan pengelolaan data, sehingga menciptakan struktur kode yang lebih rapi dan mudah dipelihara. Selain arsitektur yang mendukung pengembangan aplikasi berskala kecil hingga besar, Laravel juga dilengkapi berbagai fitur bawaan seperti *routing*, *blade templating*, *middleware*, dan *Eloquent ORM* yang menjadikan proses pengembangan lebih efisien, terstruktur, dan sesuai dengan standar pengembangan modern.

### 3.3. Laragon

Laragon adalah sebuah lingkungan pengembangan (development environment) yang dirancang untuk mempermudah dan mempercepat proses pembuatan website. Dengan sifatnya yang portabel, Laragon dapat dijalankan di berbagai perangkat tanpa instalasi ulang, serta menyediakan isolasi lingkungan agar setiap proyek tidak saling mengganggu. Selain itu, Laragon mendukung berbagai bahasa

pemrograman seperti PHP, Node.js, dan Python, serta menawarkan antarmuka yang intuitif, sehingga sangat memudahkan para pengembang dalam mengelola dan menjalankan proyek web secara efisien. (Kholid et al., 2025).

### 3.4. PHP

PHP merupakan bahasa pemrograman berbasis skrip yang digunakan di sisi server dan dirancang khusus untuk pengembangan web. Umumnya pengembang menggunakan dalam Pemrograman web. Bahasa ini pertama kali dikembangkan oleh Rasmus Lerdorf pada tahun 1994. PHP memiliki beberapa perbedaan dari lisensi *open source* populer lainnya seperti GNU *General Public License* (GPL). (Noviana, 2022).

Menurut Uzayr, (2023) dalam bukunya, PHP awalnya diciptakan oleh Rasmus Lerdorf pada tahun 1994 sebagai alat untuk melacak pengunjung situs pribadinya, dengan nama *Personal Home Page*. Seiring waktu, PHP berkembang dari sekadar kumpulan alat menjadi bahasa pemrograman yang utuh. Pada pertengahan 1990-an, kode PHP mulai dirilis ke publik, dan pada tahun 1997, Andi Gutmans dan Zeev Suraski melakukan pengembangan besar dengan merilis PHP 3.0, yang memperkenalkan fitur-fitur penting yang kini umum digunakan, terutama untuk pengembangan situs *e-commerce*.

Dari pendapat-pendapat di atas dapat disimpulkan bahwa PHP merupakan bahasa pemrograman berbasis skrip yang awalnya

dikembangkan untuk keperluan pribadi, namun kemudian berkembang menjadi bahasa pemrograman *server-side* yang kuat dan banyak digunakan dalam pengembangan web. Diciptakan oleh Rasmus Lerdorf pada tahun 1994, PHP awalnya dikenal sebagai *Personal Home Page* dan digunakan untuk melacak aktivitas pengunjung di situs pribadinya. Seiring perkembangannya, dengan kontribusi dari Andi Gutmans dan Zeev Suraski, PHP berevolusi menjadi bahasa pemrograman yang mendukung pengembangan aplikasi web berskala besar, termasuk situs *e-commerce*, dengan fitur-fitur modern seperti dukungan terhadap berbagai database, pemrosesan data dinamis, dan arsitektur modular. PHP bersifat *open source*, gratis digunakan, dan terus mengalami pengembangan aktif, menjadikannya salah satu teknologi paling populer dalam pengembangan web saat ini.

### 3.5. React JS

React JS merupakan *library* JavaScript bersifat open-source yang umum digunakan untuk membangun *User Interface* (UI) pada aplikasi berbasis satu halaman. *Library* ini dikembangkan pada tahun 2013 dan berperan dalam pengelolaan tampilan, baik untuk aplikasi versi desktop maupun mobile. Salah satu keunggulan React adalah kemampuannya dalam membentuk *reusable components* yang dapat digunakan kembali tanpa perlu membangun dari awal (Sulistiyorini et al., 2022).

#### D. Basis Data

Dalam bukunya, (Abdul Kadir, 2020) mengatakan bahwa *database* adalah kumpulan data yang saling berhubungan dan dikelola dengan baik sehingga membuat pengguna mudah dalam menggunakan data dengan cepat. Lebih lanjut, istilah ini merujuk pada sistem manajemen basis data, juga dikenal sebagai DBMS. DBMS adalah perangkat lunak yang dimaksudkan untuk mengelola dan mengorganisir jumlah besar data dan memberikan akses kepada pengguna. Selain itu, DBMS menyediakan mekanisme untuk memastikan integritas, keamanan, dan konsistensi data. Dengan adanya DBMS, pengguna tidak perlu lagi menangani data secara manual. Berbagai fungsi otomatisasi sistem mempermudah pengolahan data dalam skala kecil maupun besar.

Database berasal dari gabungan dua kata, yaitu “basis” dan “data”, di mana “basis” berarti tempat pusat, markas, atau gudang sebagai tempat berkumpul, sedangkan “data” merupakan fakta atau bukti yang merepresentasikan suatu objek seperti individu atau benda, yang disimpan dalam berbagai bentuk seperti angka, huruf, simbol, teks, suara, atau gambar. Secara keseluruhan, basis data atau *database* adalah kumpulan data yang tersusun secara sistematis dalam komputer dan dapat diakses serta dikelola melalui program komputer untuk memperoleh informasi yang dibutuhkan (Ramadhani et al., 2023).

Sehingga dapat disimpulkan bahwa basis data merupakan kumpulan data yang saling berhubungan dan tersusun secara sistematis, yang

dirancang untuk memenuhi kebutuhan informasi, khususnya dalam konteks organisasi. Pengelolaan basis data dilakukan melalui sistem manajemen basis data (DBMS), yaitu perangkat lunak yang memungkinkan pengorganisasian, pengendalian, dan pengaksesan data secara efisien. Dengan adanya DBMS, pengguna dapat memproses dan mengelola data dalam jumlah besar secara otomatis, tanpa harus menangani secara manual, serta memastikan bahwa data yang disimpan tetap terjaga integritas, keamanan, dan konsistensinya. Maka, basis data bukan hanya sekadar tempat penyimpanan informasi, melainkan juga sistem terstruktur yang mendukung pengambilan keputusan dan operasional suatu organisasi.

#### 4.1. MySQL

MySQL merupakan salah satu DBMS *open source* yang menyediakan dua jenis lisensi, yaitu *Free Software* dan *Shareware*. Melalui lisensi GNU *General Public License* (GPL), MySQL dapat digunakan secara gratis, baik untuk keperluan pribadi maupun komersial, tanpa harus membayar biaya lisensi. Hal ini menjadikan MySQL sebagai pilihan populer dalam pengembangan aplikasi karena selain bebas digunakan, juga memiliki dukungan komunitas yang luas dan dokumentasi yang lengkap. (Siregar et al., 2024).


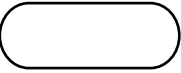
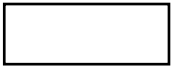
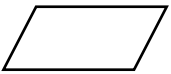
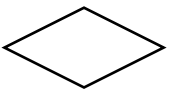

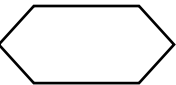
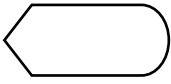

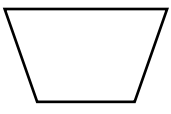
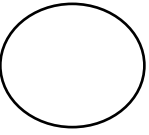
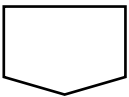
#### 6. Flowchart

Flowchart merupakan suatu diagram berbentuk grafik yang digunakan untuk menggambarkan alur, tahapan, atau langkah-langkah dalam suatu proses secara sistematis dan terstruktur. Diagram ini biasanya

digunakan dalam pengembangan sistem atau program untuk menunjukkan urutan proses yang harus dilakukan dari awal hingga akhir. Dengan adanya flowchart, proses kerja suatu sistem menjadi lebih mudah dipahami karena setiap langkah divisualisasikan secara jelas dan logis. Selain itu, flowchart juga berfungsi sebagai alat bantu dalam kegiatan analisis, perancangan sistem, serta proses pengkodean (*coding*), karena dapat membantu menguraikan permasalahan menjadi bagian-bagian kecil yang lebih rinci. Hal ini memungkinkan pengembang atau analis untuk melihat potensi kesalahan, alur proses yang tidak efisien, atau kebutuhan perbaikan lainnya dalam sistem operasional yang sedang dirancang atau dijalankan. Flowchart sering digunakan tidak hanya dalam bidang pemrograman, tetapi juga dalam berbagai bidang lainnya yang membutuhkan pemetaan proses kerja yang efektif dan terorganisir. (Listyoningrum et al., 2023).

Flowchart adalah representasi grafis yang menjelaskan urutan langkah-langkah atau prosedur dalam suatu program secara sistematis. Dalam konteks sistem, flowchart menggambarkan alur proses lengkap dengan media input, output, serta jenis penyimpanan data yang digunakan selama pengolahan. Sementara itu, flowchart program lebih fokus pada pemetaan detail dari tiap proses atau instruksi menggunakan simbol-simbol khusus, serta menunjukkan keterkaitan antar proses tersebut dalam satu rangkaian logika pemrograman (Zalukhu et al., 2023).

Tabel 2. 1 Tabel Flowchart

<b>SIMBOL</b>	<b>NAMA</b>	<b>FUNGSI</b>
	<i>Flow</i>	Arah aliran program
	<i>Terminator</i>	Awal atau akhir program
	<i>Process</i>	Menyatakan suatu proses
	<i>Input / Output</i>	Input / Output data
	<i>Decision</i>	Menyatakan suatu kondisi
	<i>Predefine Process</i>	Menyatakan sub-program
	<i>Preparation</i>	Inisialisasi / pemberian nilai awal
	<i>Display</i>	Menyatakan alat output
	<i>Document</i>	Input / Output Berbentuk Dokumen
	<i>Manual Operation</i>	Proses Manual
	<i>On Page Connector</i>	Penghubung bagian dalam satu halaman
	<i>Off page Connector</i>	Penghubung bagian berbeda halaman



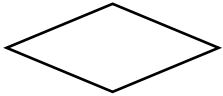

Sumber : Sari & Siregar, (2021)

## 7. Entity Relationship Diagram (ERD).

Entity-Relationship Diagram (ERD) merupakan gambaran konseptual yang digunakan untuk menjelaskan hubungan antar data dalam suatu sistem. ERD secara visual menunjukkan bagaimana struktur dari basis data tersusun, meliputi komponen seperti entitas (misalnya tabel), atribut (seperti kolom data), serta relasi antar entitas tersebut. Diagram ini merepresentasikan bentuk logis dari basis data, menjelaskan bagaimana data disimpan dalam tabel-tabel dan bagaimana keterkaitan antar tabel tersebut terbentuk (Suwada et al., 2025).

Entitas Relationship Diagram (ERD) dirancang untuk memudahkan proses analisis serta tahap-tahap perancangan berikutnya. Penyusunan ERD dilakukan dengan menampilkan seluruh hubungan antar entitas, termasuk relasi antara dua entitas, guna memberikan gambaran menyeluruh mengenai struktur hubungan dalam sistem (Sayuti et al., 2023).

Tabel 2. 2 Tabel Entity Relationship Diagram

Simbol	Arti Penggunaan
	Mewakili entitas, yang dapat berupa objek konkret atau ide abstrak, di mana informasi atau data akan disimpan.
	Elemen visual yang menandakan koneksi antara entitas dan relasi, sekaligus antara entitas dan atribut-atribut spesifiknya.
	Relasi berfungsi untuk memperlihatkan interaksi atau hubungan di antara sepasang entitas.
	Ciri khas atau hakikat sebuah entitas sebagian besar direpresentasikan melalui atributnya.

Sumber : Sari & Siregar, (2021)

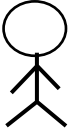
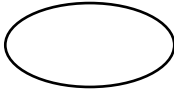


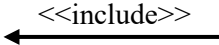
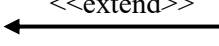
## 8. Unified Modeling Language (UML)

*Unified Modeling Language* (UML) adalah suatu standar pemodelan yang berfungsi sebagai cetak biru dalam menggambarkan proses bisnis serta mendefinisikan kelas-kelas dalam bahasa pemrograman tertentu. UML terdiri dari berbagai jenis diagram yang umum digunakan untuk membantu proses pengembangan sistem secara sistematis dan terstruktur (Ramdany et al., 2024). Menurut Saputra et al., (2023), *Unified Modeling Language* (UML) merupakan bahasa pemodelan berbasis gambar yang digunakan untuk menjelaskan, menggambarkan, mendefinisikan, membangun, dan mendokumentasikan sistem dalam pengembangan perangkat lunak berbasis objek (*Object Oriented*).

### 8.1. Use case diagram

Menurut Setiyani, (2021), *Use case* adalah gambaran atau deskripsi mengenai bagaimana suatu sistem bekerja, dilihat dari sudut pandang pengguna yang akan menggunakan sistem tersebut. Use case menjelaskan fungsi-fungsi utama yang harus dilakukan oleh sistem untuk memenuhi kebutuhan pengguna, termasuk interaksi antara pengguna dengan sistem serta proses-proses yang terjadi di dalamnya. Dengan kata lain, use case merinci apa saja yang akan dikerjakan oleh sistem dan bagaimana tiap komponen sistem terlibat dalam proses tersebut.

Tabel 2. 3 Tabel Use Case Diagram



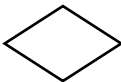


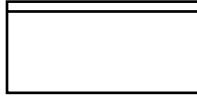
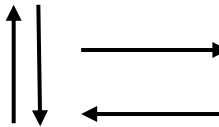
Simbol	Keterangan
	Aktor : Merepresentasikan peran yang dimainkan oleh individu, sistem eksternal, atau perangkat saat berinteraksi dengan sebuah <i>use case</i> .
	<i>Use Case</i> : Menggambarkan abstraksi dan interaksi yang terjadi antara sistem dengan aktor.
	<i>Associaton</i> : Representasi abstrak dari hubungan yang terjalin antara aktor dan sebuah <i>use case</i> .
	<i>Generalisasi</i> : Menggambarkan bagaimana seorang aktor dapat memiliki spesialisasi untuk berinteraksi dengan sebuah <i>use case</i> .
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.
	Dipergunakan untuk menunjukkan adanya penambahan fungsionalitas pada <i>use case</i> yang sudah ada, dengan syarat kondisi tertentu terpenuhi.

Sumber : Rasiban et al.(2024).

## 8.2. Activity diagram

Menurut (Sabarudin et al., 2022), Activity diagram digunakan untuk memvisualisasikan alur kerja atau aktivitas dalam suatu sistem maupun proses bisnis. Diagram ini telah mengalami perkembangan signifikan dari versi sebelumnya. Pada UML versi 1.X, penggunaannya lebih terfokus untuk menggambarkan proses komputasi sistem seperti login, dan tampilannya menyerupai flowchart. Namun, sejak UML versi 2.X, batasan tersebut dihapus, sehingga activity diagram kini dapat dimanfaatkan secara lebih luas untuk merepresentasikan alur kerja maupun proses bisnis secara umum.

Tabel 2. 4 Tabel Activity Diagram

Simbol	Nama	Keterangan
	Status Awal	Menggambarkan suatu diagram <i>activity</i> memiliki sebuah status awal.
	Aktivitas	Simbol aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	<i>Decision</i>	Simbol percabangan yang merepresentasikan adanya lebih dari satu alternatif aktivitas atau pilihan.
	<i>Join</i>	Merepresentasikan titik di mana berbagai jalur aktivitas menyatu atau digabungkan menjadi satu aliran tunggal.
	Status Akhir	Menggambarkan status final dalam diagram aktivitas, yang menunjukkan sistem telah menyelesaikan tugas atau alurnya.
	<i>Swimlane</i>	Mengidentifikasi dan memisahkan unit-unit organisasi yang memegang tanggung jawab atas pelaksanaan berbagai aktivitas.
	<i>Line Connector</i>	Digunakan sebagai penghubung antara satu simbol dengan simbol lainnya.

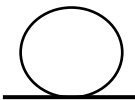
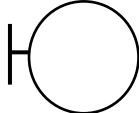
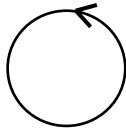
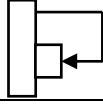


Sumber : Rasiban et al., (2024)

### 8.3. Sequence diagram

Mahardika et al., (2023) menjelaskan bahwa Sequence diagram digunakan untuk menggambarkan interaksi antar objek dalam suatu sistem berdasarkan urutan waktu. Diagram ini menunjukkan bagaimana objek-objek yang terlibat dalam sebuah *use case* saling berkomunikasi, serta urutan pesan yang dikirimkan antar objek tersebut. Dengan kata lain, sequence diagram membantu memvisualisasikan alur komunikasi

dan interaksi yang terjadi, yang mengikuti urutan waktu, antara aktor (pengguna) dan sistem atau antar objek dalam sistem itu sendiri.

Tabel 2. 5 Tabel Sequence Diagram

Simbol	Nama	Keterangan
	<i>Entity Class</i>	Simbol yang mempresentasikan sistem sebagai fondasi utama dalam perancangan basis data.
	<i>Boundary Class</i>	Simbol yang berfungsi untuk merepresentasikan titik komunikasi atau antarmuka antar lingkungan di dalam sistem.
	<i>Control Class</i>	Simbol yang melambangkan kelas objek, di mana seluruh logika terkait dikapsulasi.
	<i>Recursive</i>	Simbol yang merepresentasikan pesan yang di kirim oleh objek kepada dirinya sendiri.
	<i>Activation</i>	Simbol ini mewakili periode waktu suatu aktivitas berlangsung dalam alur proses sistem.
	<i>Life Line</i>	Sebuah simbol yang mewakili proses atau aktivitas yang melekat pada suatu objek.

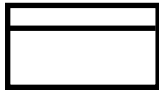
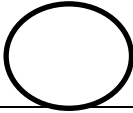





Sumber :Rasiban et al., (2024)

#### 8.4. Class diagram

Menurut Saputra et al., (2024), Class diagram adalah diagram yang digunakan untuk menggambarkan berbagai kelas dan paket yang ada dalam suatu sistem atau perangkat lunak. Diagram ini memberikan gambaran struktural yang bersifat statis tentang sistem, menunjukkan bagaimana kelas-kelas tersebut saling berhubungan, baik melalui asosiasi, pewarisan, atau dependensi. Dengan demikian, diagram kelas

membantu dalam memvisualisasikan struktur dasar sistem dan hubungan antar elemen-elemen penting dalam perangkat lunak tersebut.

Tabel 2. 6 Tabel Class Diagram

Simbol	Nama	Keterangan
	<i>Class</i>	Komponen kelas dalam struktur sistem.
	<i>Interface</i>	Memiliki kemiripan dengan ide interface dalam konteks pemrograman berorientasi objek.
	<i>Association</i>	Hubungan antar kelas secara generik dikenal sebagai asosiasi, dan sering kali disertai dengan indikasi multiplicit.
	<i>Direct Association</i>	Asosiasi merepresentasikan hubungan antar kelas, yang mengindikasikan bahwa satu kelas berfungsi atau dimanfaatkan oleh kelas lain, dan biasanya detail multiplicity juga disajikan.
	Generalisasi	Menggambarkan hubungan antara kelas-kelas sebagai generalisasi dan spesialisasi (dari yang lebih umum ke yang lebih spesifik).
	<i>Dependency</i>	Menggambarkan hubungan antara kelas-kelas sebagai bentuk kebergantungan, di mana satu kelas bergantung pada kelas lain.
	<i>Aggregation</i>	Hubungan antar kelas yang merepresentasikan konsep bagian dari keseluruhan (whole-part).

Sumber : Rasiban et al., (2024)

## B. Kajian Empiris

Penelitian ini disusun dengan mengacu pada sejumlah referensi jurnal yang relevan dengan topik mengenai sistem rekomendasi pemilihan laptop menggunakan metode hybrid *Collaborative Filtering–Content-Based Filtering*

(CF-CBF) dengan pendekatan TF-IDF. Referensi tersebut dijadikan sebagai dasar acuan yang disusun secara sistematis guna mendukung proses pengembangan sistem. Studi literatur dari penelitian sebelumnya dimanfaatkan sebagai landasan untuk memperkuat serta menyempurnakan rancangan sistem yang dibangun, melalui proses analisis perbandingan dan evaluasi terhadap sistem-sistem serupa yang telah dikembangkan pada penelitian terdahulu.

Dalam penelitian yang dilakukan oleh Salam et al., (2024), dikembangkan sebuah sistem rekomendasi topik tugas akhir menggunakan pendekatan *hybrid* yang menggabungkan metode *Collaborative Filtering* (CF) dan *Content-Based Filtering* (CBF). Pendekatan ini bertujuan meningkatkan akurasi dan relevansi rekomendasi secara personal. Penelitian ini melibatkan 38 mahasiswa dan menggunakan uji reliabilitas *Cronbach's alpha* ( $\alpha = 0,85$ ) untuk memastikan konsistensi instrumen. Hasil evaluasi sistem dengan metrik *Precision*, *Recall*, dan *F1-Score* menunjukkan bahwa metode CF memberikan hasil terbaik, dengan nilai *Precision* sebesar 0,85, *Recall* 0,92, dan *F1-Score* 0,88. Sistem ini diharapkan dapat membantu mahasiswa dalam memilih topik tugas akhir yang paling sesuai dengan preferensi akademik mereka.

Dalam penelitian yang dilakukan oleh Putra & Rachman, (2024), dilakukan perbandingan antara tiga pendekatan utama dalam sistem rekomendasi lagu, yaitu *Collaborative Filtering* (CF), *Content-Based Filtering* (CBF), dan *Hybrid Filtering*. Penelitian ini menggunakan dataset yang berisi 68.330 entri, dan mengevaluasi performa masing-masing metode berdasarkan nilai presisi. Metode CF dan CBF diuji secara terpisah, kemudian digabungkan

dalam dua pendekatan hybrid: CF-CBF dan CBF-CF. Hasilnya menunjukkan bahwa pendekatan hybrid CF-CBF memberikan presisi tertinggi sebesar 50,7%, mengungguli CF (49,9%) dan CBF (39,5%). Sebaliknya, pendekatan *hybrid* CBF-CF justru menghasilkan presisi terendah, yaitu 38,4%. Temuan ini mengindikasikan bahwa urutan penggabungan metode dalam pendekatan hybrid sangat memengaruhi efektivitas rekomendasi. Dengan demikian, *hybrid* CF-CBF dinilai paling optimal dalam memberikan rekomendasi lagu yang relevan sesuai preferensi pengguna.

Dalam penelitian yang dilakukan oleh (Ihsan & Setiawan, 2024), dikembangkan sebuah sistem rekomendasi tempat wisata yang dipersonalisasi sebagai upaya mendukung pemulihan sektor pariwisata pasca-pandemi, khususnya di Kota Bandung. Sistem ini dirancang menggunakan pendekatan *hybrid* yang mengintegrasikan *Content-Based Filtering* (CBF) dengan *Simple Recurrent Neural Network* (RNN), guna meningkatkan akurasi dalam memprediksi preferensi pengguna. Penelitian ini menggabungkan teknik TF-IDF dan *word embedding* berbasis *RoBERTa* untuk mengidentifikasi kesamaan antar destinasi wisata berdasarkan konten deskriptifnya. Selain itu, pola perilaku pengguna dianalisis dari waktu ke waktu menggunakan Simple RNN, yang kemudian dioptimalkan dengan algoritma *Particle Swarm Optimization* (PSO). Hasil eksperimen menunjukkan bahwa model RNN yang dioptimasi dengan PSO mampu mencapai akurasi sebesar 94,37%, mengungguli teknik optimasi lainnya seperti Adam dan SGD. Temuan ini menunjukkan bahwa kombinasi pendekatan konten, pembelajaran sekuensial, dan optimasi cerdas

dapat secara signifikan meningkatkan personalisasi dalam sistem rekomendasi pariwisata.

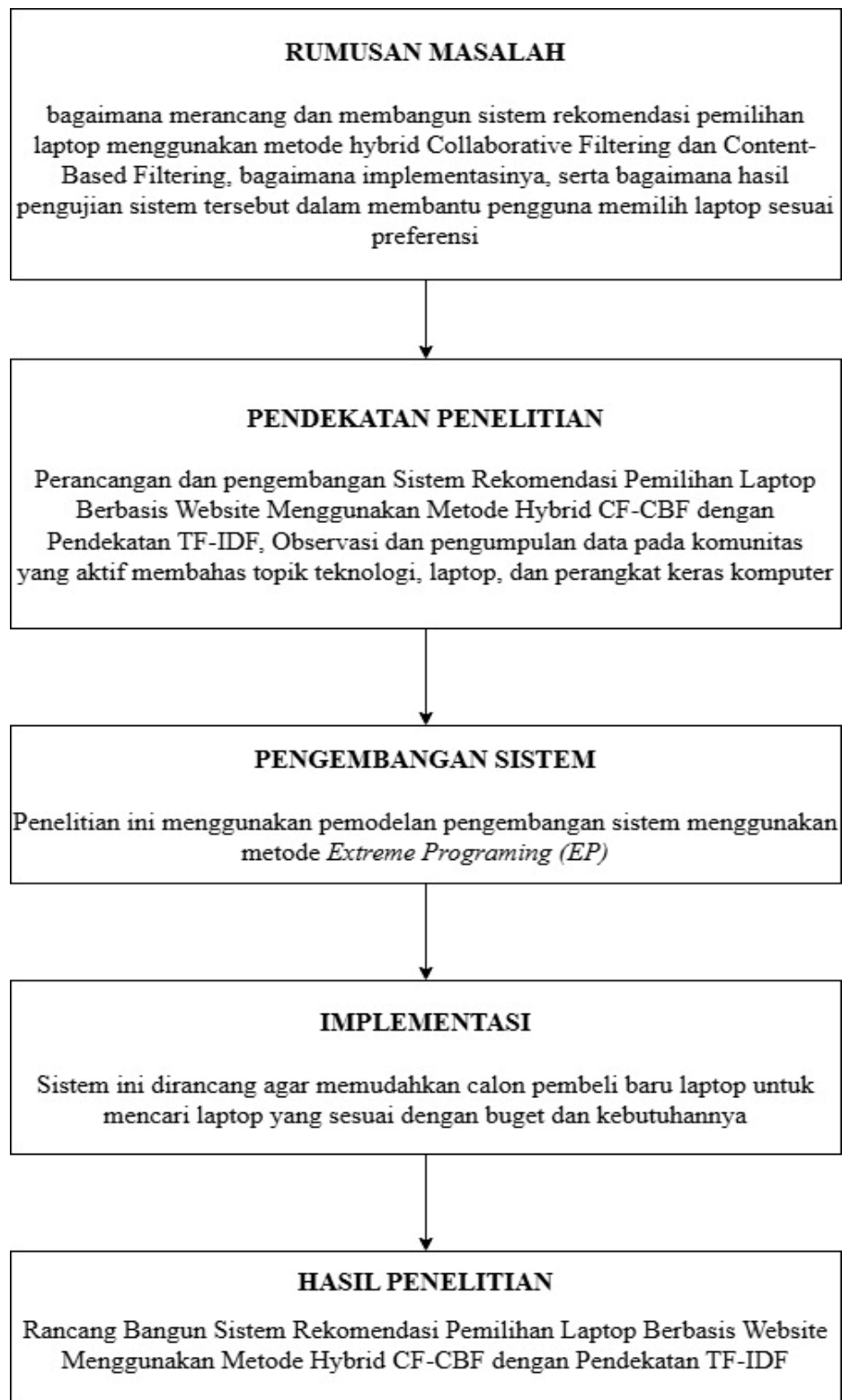
Dalam penelitian yang dilakukan oleh Sibuea & Widodo, (2024), dikembangkan dan dievaluasi sebuah sistem rekomendasi produk berbasis *Hybrid Filtering* yang mengintegrasikan *Collaborative Filtering* (CF) dan *Content-Based Filtering* (CBF). Hasil evaluasi menunjukkan bahwa pendekatan hybrid ini secara signifikan meningkatkan akurasi dan relevansi rekomendasi, dengan capaian precision sebesar 82% dan *recall* sebesar 78%, melebihi performa pendekatan individu. Penelitian ini juga menyoroti keunggulan *Hybrid Filtering* dalam mengatasi permasalahan cold start, yaitu situasi ketika sistem harus merekomendasikan item atau produk kepada pengguna atau item baru yang belum memiliki cukup riwayat interaksi. Dengan memanfaatkan atribut produk melalui pendekatan CBF, sistem masih dapat memberikan rekomendasi yang cukup akurat, sementara CF memperkuat akurasi untuk pengguna aktif. Lebih jauh, kombinasi antara analisis perilaku pengguna (CF) dan karakteristik produk (CBF) memungkinkan sistem menghasilkan rekomendasi yang lebih personal dan relevan, terutama bagi pengguna dengan preferensi kompleks. Meskipun pendekatan *hybrid* membutuhkan komputasi yang lebih kompleks dan waktu pemrosesan yang lebih lama, penelitian ini menyimpulkan bahwa dengan optimasi lanjutan, model *hybrid* memiliki potensi besar untuk diterapkan secara luas di platform *e-commerce*, guna meningkatkan pengalaman pengguna, konversi penjualan, serta loyalitas pelanggan.

Dalam penelitian yang dilakukan oleh Huda et al., (2022), dikembangkan sistem rekomendasi artikel berbasis *Content-Based Filtering* untuk membantu mahasiswa aktif Program Studi Informatika Universitas Amikom Yogyakarta dalam menemukan artikel yang relevan dengan minat mereka, khususnya dalam konteks pemilihan konsentrasi. Permasalahan utama yang dihadapi adalah tingginya volume artikel yang ditawarkan, yang justru menyulitkan mahasiswa dalam memilih artikel yang sesuai dengan kebutuhan mereka. Sistem rekomendasi yang dibangun menggunakan pendekatan TF-IDF *Vector Similarity* dan memanfaatkan dataset dari portal berita Kabar Informatika. Evaluasi menunjukkan bahwa sistem mampu memberikan hasil *Recall@5* sebesar 73% dan *Recall@10* sebesar 80%, menandakan bahwa sistem cukup efektif dalam merekomendasikan artikel yang relevan. Meskipun jumlah item dalam dataset terbatas, kelengkapan fitur yang tersedia tetap memberikan kontribusi besar terhadap performa sistem. Penelitian ini juga menunjukkan bahwa mesin rekomendasi yang dikembangkan tidak memiliki dependensi tinggi terhadap dataset tertentu, sehingga memungkinkan untuk melakukan penggantian atau penambahan data secara fleksibel. Beberapa aspek teknis seperti proses tokenisasi, terutama dalam menangani kata-kata dari potongan kode program dalam artikel, disebut masih dapat ditingkatkan untuk lebih mengoptimalkan hasil rekomendasi. Penelitian ini memberikan kontribusi penting dalam pengembangan sistem rekomendasi berbasis konten untuk lingkungan akademik yang dinamis dan penuh informasi.

Berdasarkan sejumlah penelitian, pendekatan *Hybrid Filtering* yang menggabungkan *Collaborative Filtering* (CF) dan *Content-Based Filtering* (CBF) terbukti memberikan hasil rekomendasi yang lebih akurat dan relevan dibandingkan metode tunggal. Hybrid mampu mengatasi kekurangan masing-masing pendekatan, seperti masalah cold start yang biasa terjadi pada CF. Dalam beberapa studi, CF menunjukkan performa tinggi dalam hal presisi dan recall saat data interaksi pengguna cukup banyak, sementara CBF efektif ketika item atau pengguna masih baru, karena menggunakan informasi atribut atau konten. Selain itu, pendekatan berbasis TF-IDF masih banyak digunakan untuk menganalisis kemiripan konten, khususnya pada sistem rekomendasi artikel atau deskripsi item. Namun, TF-IDF memiliki keterbatasan dalam memahami konteks semantik, sehingga beberapa penelitian menggabungkannya dengan teknik lanjutan seperti word embedding atau model RNN yang dioptimalkan.

Menariknya, urutan penggabungan CF dan CBF juga berpengaruh terhadap hasil rekomendasi. Pendekatan CF-CBF cenderung memberikan hasil yang lebih baik daripada CBF-CF, karena perilaku pengguna yang dianalisis terlebih dahulu memberikan dasar yang kuat sebelum diperkaya oleh informasi konten. Secara umum, penelitian-penelitian ini menunjukkan bahwa pendekatan *hybrid* yang tepat dapat meningkatkan personalisasi, akurasi, dan efektivitas sistem rekomendasi di berbagai bidang.

### C. Kerangka berfikir



Gambar 2. 3 Gambar Kerangka berfikir dalam pembuatan sistem