BAB II
KAJIAN LITERATUR

2.1. Konsep Sistem Informasi

2.1.1 Sistem

Sistem adalah sekumpulan elemen-elemen yang saling bekerja sama untuk menangani, mengatur, serta mengkoordinasikan suatu kegiatan yang rutin terjadi. Dengan menjalankan suatu sistem dengan benar dan teratur maka hal ini dapat membantu kelancaran semua kegiatan yang dilakukan perusahaan sehingga tujuan perusahaan dapat tercapai. Adapun karakteristik atau sifat–sifat tertentu dari suatu sistem, yaitu:

1. Komponen sistem yang terdiri dari sejumlah komponen yang saling berinteraksi dan saling bekerja sama untuk membentuk suatu kesatuan.
2. Batasan sistem merupakan daerah yang membatasi suatu sistem dengan sistem yang lain atau lingkungan luarnya.
3. Lingkungan luar sistem adalah apapun di luar batasan sistem yang mempengaruhi operasi sistem.
4. Penghubung sistem merupakan media penghubung antara komponen sistem. Melalui penghubung ini, sumber daya dapat mengalir dari satu komponen ke komponen lainnya
5. Masukan sistem (input) adalah hasil dari perintah yang dimasukkan ke sistem. Masukan dapat berupa masukan perawatan dan masukan sinyal.
6. Keluaran sistem (output) adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan keluaran dapat menjadi masukan untuk komponen sistem lainnya.
7. Pengolah sistem yaitu suatu bagian sistem yang akan mengubah masukan menjadi keluaran.
8. Sasaran sistem yaitu sasaran atau tujuan yang sangat menentukan masukan yang dibutuhkan dan keluaran yang dihasilkan [6].

2.1.2 Informasi

Informasi adalah data yang telah diproses dan memiliki arti bagi pemakainya. Informasi merupakan sebuah data yang dikelola menjadi sesuatu yang lebih berguna dan lebih berarti bagi yang menerimanya untuk membuat suatu keputusan dan melakukan tindakan. Adapun ciri-ciri informasi yang berkualitas, yaitu:

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

1. Akurat, artinya informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan.
2. Relevan, artinya informasi tersebut mempunyai manfaat untuk pemakainya.
3. Tepat waktu, artinya informasi yang datang ke penerima tidak boleh terlambat agar tetap bernilai [6].

2.1.3 Sistem Informasi

Sistem informasi adalah sistem dalam suatu organisasi yang menyatukan kebutuhan pemrosesan transaksi sehari-hari, dukungan operasional, manajemen, dan kegiatan strategis suatu organisasi, dan menyediakannya kepada sejumlah pihak eksternal tertentu [7]. Terdapat beberapa komponen atau blok bangunan yang membentuk sistem informasi, yaitu:

1. Blok masukan yaitu input yang mewakili data yang masuk ke dalam sistem informasi.
2. Blok model yaitu blok yang terdiri dari kombinasi prosedur, logika, dan model matematik untuk memanipulasi data input dan data yang tersimpan agar menghasilkan keluaran yang diinginkan.
3. Blok keluaran yaitu produk atau hasil dari sistem informasi yang merupakan informasi berkualitas dan dokumen yang berguna untuk pemakai sistem.
4. Blok teknologi yaitu teknologi yang digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilk dan mengirimkan keluaran serta membantu pengendalian sistem secara kesuluruhan.
5. Blok basis data yaitu kumpulan data yang saling terhubung satu dengan yang lainnya dan tersimpan di perangkat keras yang digunakan untuk memanipulasinya.
6. Blok kendali yaitu pengendalian untuk mencegah faktor-faktor yang dapat merusak sistem informasi seperti bencana alam, api, suhu tinggi, air, debu, kecurangan, kejanggalan sistem, kesalahan sistem, sabotase, dan lain sebagainya [8].

2.2. Konsep Web

Website adalah merupakan kumpulan berbagai halaman yang menampilkan berbagai jenis informasi teks, gambar, animasi, dan elemen lainnya. Halaman-halaman dalam sebuah *website* bisa bersifat statis maupun dinamis, serta dapat saling terhubung membentuk jaringan yang terkoneksi. *Website* juga dapat diartikan sebagai kumpulan halaman dalam satu domain yang menyajikan berbagai informasi agar dapat diakses dan dibaca oleh pengguna internet melalui mesin pencari. Informasi yang ditampilkan dalam *website*

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

biasanya mencakup teks, gambar, ilustrasi, serta video yang disesuaikan dengan berbagai kebutuhan [9].

Website ialah serangkaian halaman di suatu domain di dalam internet yang dibuat dengan tujuan khusus, saling terhubung, dan dapat diakses secara umum melalui halaman depan menggunakan peramban dengan URL situs [10].

Dalam pengembangan aplikasi web modern, terdapat beberapa komponen utama yang saling terintegrasi:

1. *Frontend (Client-side)*

Frontend merupakan bagian dari aplikasi web yang berinteraksi langsung dengan pengguna. Komponen ini dibangun menggunakan HTML untuk struktur, CSS (Cascading Style Sheets) untuk tampilan, dan JavaScript untuk interaktivitas. Frontend bertanggung jawab menampilkan data dan mengumpulkan input dari pengguna.

2. *Backend (Server-side)*

Backend adalah bagian yang menangani logika bisnis, pemrosesan data, dan interaksi dengan *database*. *Backend* dapat dibangun menggunakan berbagai bahasa pemrograman seperti PHP, Python, Java, atau Node.js. Komponen ini bertanggung jawab untuk memproses permintaan dari frontend dan mengirimkan respons yang sesuai.

3. *Database*

Database berfungsi sebagai tempat penyimpanan data yang terstruktur. Dalam pengembangan web, *database* relasional seperti MySQL, PostgreSQL, atau database non-relasional seperti MongoDB sering digunakan untuk menyimpan dan mengelola data aplikasi.

Dalam konteks sistem rekomendasi diet berbasis web, arsitektur web memungkinkan pengguna untuk mengakses layanan rekomendasi melalui antarmuka yang *user-friendly*, sementara algoritma *collaborative filtering* dan logika bisnis lainnya dapat diproses di server dengan efisien.

2.3. Konsep Diet

Diet merupakan pengaturan pola makan yang terencana untuk mencapai tujuan kesehatan tertentu. Diet adalah pengaturan pola makan dan konsumsi makanan serta minuman yang di larang, dibatasi jumlahnya atau di perbolehkan dengan jumlah tertentu untuk tujuan terapi penyakit yang diderita, kesehatan, atau penurunan berat badan [11]. Diet tidak hanya terbatas pada pengurangan asupan makanan untuk menurunkan berat badan, tetapi juga mencakup perencanaan makan yang disesuaikan

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

dengan kebutuhan individu berdasarkan usia, jenis kelamin, tingkat aktivitas, dan kondisi kesehatan.

Dalam konteks ilmu gizi, diet memiliki beberapa komponen penting yang perlu diperhatikan. Komponen utama diet meliputi karbohidrat, protein, lemak, vitamin, mineral, dan air [12]. Setiap komponen memiliki peran spesifik dalam menjaga kesehatan tubuh:

1. Karbohidrat (45-65% dari total kalori harian)
    a. Berfungsi sebagai sumber energi utama tubuh
    b. Membantu mengatur metabolisme protein dan lemak
    c. Menjaga kesehatan sistem pencernaan
    d. Sumber: nasi, roti, kentang, pasta
2. Protein (10-35% dari total kalori harian)
    a. Berperan dalam pembentukan dan perbaikan jaringan tubuh
    b. Membantu pembentukan enzim dan hormon
    c. Mendukung sistem kekebalan tubuh
    d. Sumber: daging, ikan, telur, kacang-kacangan
3. Lemak (20-35% dari total kalori harian)
    a. Diperlukan untuk penyerapan vitamin larut lemak (A, D, E, K)
    b. Melindungi organ-organ vital
    c. Membantu menjaga suhu tubuh
    d. Sumber: minyak, kacang-kacangan, alpukat
4. Vitamin dan Mineral
    a. Mengatur berbagai proses metabolisme
    b. Mendukung pertumbuhan dan perkembangan
    c. Memelihara kesehatan tulang dan gigi
    d. Sumber: buah-buahan, sayuran, susu

Perkembangan teknologi dan penelitian telah mengungkapkan kompleksitas hubungan antara diet dan kesehatan. Studi longitudinal yang dilakukan oleh ahli menunjukkan bahwa pola makan memiliki dampak jangka panjang tidak hanya pada kesehatan fisik tetapi juga pada kesehatan mental dan kognitif [13]. Penelitian ini mengungkapkan bahwa diet seimbang dapat mengurangi risiko berbagai penyakit kronis seperti diabetes, penyakit jantung, dan beberapa jenis kanker hingga 60%.

Aspek sosiokultural juga memainkan peran penting dalam praktik diet. Setiap budaya memiliki pendekatan unik terhadap makanan dan nutrisi yang telah berkembang selama berabad-abad. Seorang ahli menganalisis bagaimana faktor budaya, sosial, dan

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

ekonomi mempengaruhi pilihan makanan dan pola diet masyarakat [14]. Pemahaman tentang aspek sosiokultural ini penting dalam mengembangkan rekomendasi diet yang dapat diterima dan dilaksanakan oleh berbagai kelompok masyarakat.

Di era modern, tantangan diet semakin kompleks dengan adanya perubahan gaya hidup dan pola makan global. Urbanisasi, industrialisasi makanan, dan perubahan pola kerja telah menciptakan tantangan baru dalam mempertahankan pola makan sehat. Sekelompok ahli mengidentifikasi bahwa stress, waktu terbatas, dan aksesibilitas makanan cepat saji menjadi hambatan utama dalam menjalankan diet sehat. Kondisi ini memerlukan pendekatan yang lebih adaptif dalam memberikan rekomendasi diet.

Perkembangan teknologi informasi membuka peluang baru dalam manajemen diet. Sistem rekomendasi diet berbasis *artificial intelligence* dapat membantu individu membuat keputusan makanan yang lebih baik dengan mempertimbangkan berbagai faktor personal. Penelitian tersebut menunjukkan bahwa penggunaan aplikasi diet berbasis AI dapat meningkatkan kepatuhan terhadap rencana diet hingga 45% dibandingkan dengan metode konvensional [14].

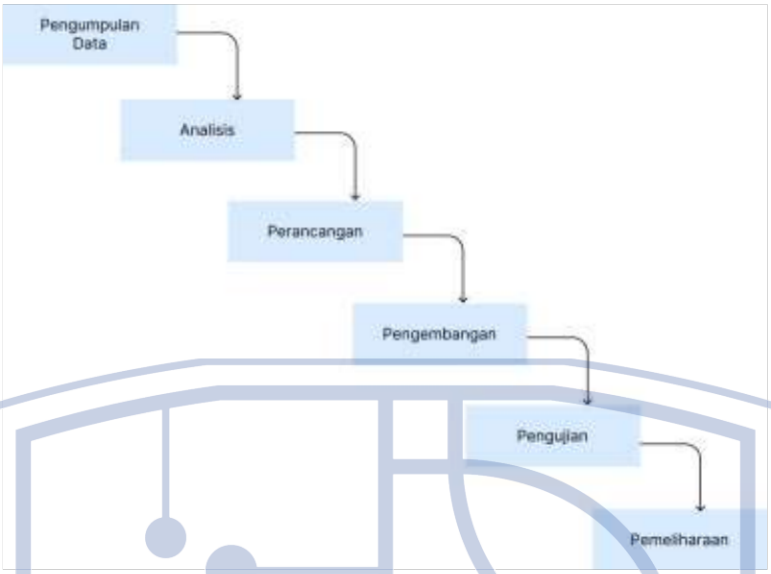
Masa depan diet semakin menjanjikan dengan berkembangnya nutrigenomik dan *personalized nutrition*. Pendekatan ini memungkinkan penyusunan rekomendasi diet yang sangat personal berdasarkan profil genetik individu. Prediksi menunjukkan bahwa dalam dekade mendatang, rekomendasi diet akan semakin terintegrasi dengan data kesehatan personal, termasuk informasi genetik, mikrobioma, dan biomarker lainnya [12].

2.4. Metodologi Pengembangan

2.4.1 Model Waterfall

Model Waterfall merupakan salah satu pendekatan paling klasik dan banyak digunakan dalam System Development Life Cycle (SDLC). Model ini menggunakan alur kerja yang sistematis dan berurutan, dimulai dari tahap perencanaan kebutuhan, analisis sistem, perancangan, pengkodean, pengujian, hingga pemeliharaan. Setiap tahapan harus diselesaikan secara menyeluruh sebelum melanjutkan ke tahap berikutnya, sehingga model ini sangat cocok diterapkan pada proyek-proyek yang kebutuhan sistemnya telah terdefinisi dengan jelas sejak awal. Waterfall dikenal sebagai model yang terstruktur dan mudah dikontrol karena setiap fase memiliki output yang terdokumentasi dengan baik [15]

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.



Gambar 2.1 Tahapan metode waterfall

Model Waterfall terdiri dari beberapa fase utama yang dilaksanakan secara berurutan, yaitu:

a. Tahap Pengumpulan Data

Tahap pengumpulan data, juga dikenal sebagai tahap analisis kebutuhan atau *requirements gathering*, merupakan fase awal dalam model *Waterfall*. Pada tahap ini, pengembang bekerja untuk mendapatkan pemahaman menyeluruh tentang kebutuhan sistem yang akan dikembangkan.

Pengumpulan data yang efektif sangat penting untuk keberhasilan proyek pengembangan perangkat lunak karena membentuk dasar untuk semua pekerjaan selanjutnya. Kesalahan atau kelalaian pada tahap ini dapat menyebabkan kesalahan desain yang mahal untuk diperbaiki pada tahap selanjutnya.

Teknik pengumpulan data dalam pengembangan sistem meliputi:

1. Wawancara, dilakukan dengan pengguna potensial dan pemangku kepentingan. Proses ini dimulai dengan mengidentifikasi tujuan wawancara, yaitu menentukan informasi apa yang ingin digali. Setelah itu, pilihlah responden yang relevan, seperti calon pengguna atau pihak-pihak yang berkepentingan. Susunlah daftar pertanyaan yang sesuai, baik dalam bentuk terstruktur maupun semi-terstruktur, lalu jadwalkan sesi wawancara pada waktu yang nyaman bagi responden. Selama wawancara berlangsung, gali informasi secara mendalam dan catat semua jawaban yang

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

diberikan. Setelah seluruh wawancara selesai, lakukan analisis terhadap hasil yang diperoleh untuk mengidentifikasi kebutuhan dan harapan mereka.

2. Survei dan Kuesioner, digunakan untuk mengumpulkan informasi dari lebih banyak pengguna potensial. Langkah pertama adalah menentukan tujuan dari survei tersebut. Kemudian, rancang kuesioner dengan pertanyaan yang jelas dan relevan, baik dalam bentuk pilihan ganda, skala, maupun isian. Tentukan siapa saja yang akan menjadi responden dan distribusikan survei melalui media yang sesuai, seperti online atau cetak. Setelah data terkumpul, lakukan analisis terhadap hasil survei untuk menemukan pola atau kebutuhan yang muncul, lalu simpulkan temuan tersebut sebagai bahan pertimbangan dalam pengambilan keputusan.
3. Observasi, dilakukan pengamatan terhadap proses kerja yang sedang berjalan. Tentukan terlebih dahulu fokus observasi, misalnya pada alur kerja tertentu atau aktivitas spesifik. Siapkan alat bantu seperti checklist atau catatan observasi untuk mendokumentasikan temuan secara sistematis. Lakukan observasi secara langsung tanpa mengganggu proses kerja, dan catat semua hal penting yang ditemukan, seperti hambatan atau peluang perbaikan. Setelah observasi selesai, analisis hasil yang didapat untuk mengidentifikasi area yang membutuhkan peningkatan, lalu buat laporan singkat yang berisi temuan dan rekomendasi perbaikan.

Untuk sistem rekomendasi produk makanan dan minuman, pengumpulan data melibatkan akuisisi informasi produk (seperti nama, kategori, kandungan nutrisi), data *rating* pengguna, dan preferensi diet. Sumber data dapat mencakup *database* makanan terbuka seperti *Open Food Facts*, platform e-commerce, dan survei pengguna.

b. Tahap Analisis

Tahap analisis dalam model *Waterfall* bertujuan untuk memahami masalah domain dan mengidentifikasi persyaratan spesifik untuk sistem yang akan dikembangkan [14]. Tahap ini mengubah kebutuhan mentah yang dikumpulkan selama tahap pengumpulan data menjadi spesifikasi kebutuhan yang terstruktur dan terperinci.

Analisis dalam pengembangan sistem umumnya terbagi menjadi dua kategori utama:

1. Analisis Proses, berfokus pada aspek dinamis sistem—bagaimana data diproses, alur kerja, dan urutan operasi. Teknik yang digunakan dalam analisis proses meliputi:
    a. Diagram Alir (*Flowchart*): Representasi grafis dari urutan operasi dalam proses

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

b. *Data Flow Diagram* (DFD): Menunjukkan aliran data melalui sistem dan pemrosesan yang dilakukan pada data tersebut.
c. *Business Process Modeling Notation* (BPMN): Standar untuk memodelkan proses bisnis

2. Analisis Kebutuhan, berfokus pada identifikasi, dokumentasi, dan verifikasi kebutuhan fungsional dan non-fungsional sistem. Analisis kebutuhan mencakup:
   a. Kebutuhan Fungsional: Mendeskripsikan apa yang harus dilakukan oleh sistem
   b. Kebutuhan Non-Fungsional: Menentukan kriteria yang dapat digunakan untuk menilai operasi sistem (seperti keamanan, kinerja, kegunaan)

Untuk sistem rekomendasi, analisis khususnya berfokus pada algoritma rekomendasi yang akan digunakan, desain antarmuka pengguna, dan integrasi dengan sumber data.

c. Tahap Perancangan

Tahap perancangan menerjemahkan spesifikasi kebutuhan menjadi blueprint untuk konstruksi sistem [13]. Perancangan adalah proses menentukan arsitektur, komponen, antarmuka, dan karakteristik lain dari sistem atau komponennya.

Perancangan dalam pengembangan sistem meliputi dua aspek utama:

1. Perancangan Tampilan (UI/UX Design): Berfokus pada antarmuka pengguna dan pengalaman pengguna sistem. Aspek perancangan tampilan meliputi:
   a. *Wireframing*: Membuat sketsa dasar antarmuka pengguna
   b. *Prototyping*: Mengembangkan model interaktif dari antarmuka pengguna
   c. Visual Design: Menentukan aspek estetika seperti skema warna, tipografi, dan ikonografi

Tools yang sering digunakan untuk perancangan tampilan di antaranya adalah Figma, Adobe XD, dan Sketch.

2. Perancangan Basis Data: Perancangan basis data adalah proses menghasilkan model data rinci yang mendukung operasi dan tujuan organisasi. Perancangan basis data meliputi:
   a. Perancangan Konseptual: Mengembangkan model data tingkat tinggi berdasarkan spesifikasi kebutuhan.
   b. Perancangan Logis: Memetakan model konseptual ke model logis seperti model relasional .

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

c. Perancangan Fisik: Mengimplementasikan model logis dalam sistem manajemen basis data tertentu

Tools yang sering digunakan untuk perancangan basis data meliputi MySQL Workbench, ERDPlus, dan draw.io.

Untuk sistem rekomendasi produk makanan dan minuman, perancangan melibatkan desain antarmuka yang intuitif untuk menampilkan rekomendasi, dan struktur basis data yang efisien untuk menyimpan data produk, *rating* pengguna, dan preferensi diet.

d. Tahap Pengembangan

Tahap pengembangan, juga dikenal sebagai tahap implementasi atau konstruksi, melibatkan konversi desain ke dalam kode yang dapat dieksekusi. Tahap ini adalah di mana sistem aktual dibangun—mengubah *blueprint* yang dibuat selama tahap perancangan menjadi produk yang berfungsi.

Pengembangan sistem modern umumnya mencakup dua komponen utama:

1. **Pengembangan *Front-End***: Berkaitan dengan implementasi antarmuka pengguna dan pengalaman pengguna. Teknologi front-end meliputi:
    a. HTML, CSS, dan JavaScript: Bahasa dasar untuk pengembangan web
    b. Framework Front-end: Seperti React, Angular, dan Vue.js yang menyediakan struktur dan komponen yang dapat digunakan kembali
    c. TypeScript: *Superset* dari JavaScript yang menyediakan pemeriksaan tipe statis
2. **Pengembangan *Back-End***: Berfokus pada server, aplikasi, dan basis data (Martin, 2020). Teknologi *back-end* meliputi:
    a. Bahasa Pemrograman Server-side: Seperti JavaScript (Node.js), Python, PHP, dan Java
    b. *Framework Back-end*: Seperti Express.js, Django, Laravel, dan Spring Boot
    c. Sistem Manajemen Basis Data: Seperti MySQL, PostgreSQL, MongoDB, dan SQL Server.

Untuk sistem rekomendasi produk makanan dan minuman, pengembangan melibatkan implementasi algoritma *collaborative filtering*, integrasi dengan sumber data produk, dan pengembangan antarmuka yang memungkinkan pengguna menjelajahi dan menerima rekomendasi.

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

e. Tahap Pengujian

Tahap pengujian adalah proses evaluasi sistematis untuk menentukan apakah sistem memenuhi persyaratan yang ditentukan dan mengidentifikasi perbedaan antara output yang diharapkan dan aktual. Pengujian adalah komponen kritis dari jaminan kualitas dan merupakan ulasan objektif dari program, aplikasi, atau sistem.

Strategi pengujian mencakup berbagai jenis pengujian:

1. Pengujian Unit: Menguji komponen individual atau modul kode
2. Pengujian Integrasi: Menguji interaksi antara komponen atau sistem yang terintegrasi.
3. Pengujian Sistem: Menguji sistem lengkap terhadap spesifikasi kebutuhan.
4. Pengujian Penerimaan: Menguji apakah sistem memenuhi kebutuhan bisnis dan dapat diterima oleh pengguna.
5. Pengujian Fungsional: Memverifikasi bahwa sistem melakukan fungsi yang ditentukan dalam spesifikasi kebutuhan
6. Pengujian Non-Fungsional: Menilai aspek sistem seperti kinerja, kegunaan, keandalan, dll.

Untuk sistem rekomendasi produk makanan dan minuman, pengujian melibatkan evaluasi akurasi rekomendasi, kinerja algoritma *collaborative filtering*, dan kegunaan antarmuka pengguna [16].

f. Tahap Pemeliharaan

Tahap pemeliharaan merupakan fase terakhir dan terpanjang dalam System Development Life Cycle (SDLC) model Waterfall. Pemeliharaan sistem didefinisikan sebagai serangkaian aktivitas yang dilakukan setelah penerapan sistem untuk memastikan sistem tetap berfungsi dengan baik, memperbaiki kesalahan, meningkatkan kinerja, dan mengadaptasi sistem terhadap perubahan kebutuhan atau lingkungan. Tahap ini sangat krusial karena sistem informasi yang telah diimplementasikan akan terus memerlukan penyesuaian sepanjang masa operasionalnya.

Penelitian terbaru menunjukkan bahwa tahap pemeliharaan masih menghabiskan sekitar 50% hingga 70% dari total biaya siklus hidup sistem [17]. Hal ini menegaskan pentingnya perencanaan dan pengelolaan pemeliharaan yang efektif dalam pengembangan sistem, termasuk sistem rekomendasi produk makanan dan minuman.

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

2.5. Teknik Pengembangan Sistem

2.5.1 Entity Relationship Diagram

Entity Relationship Diagram (ERD) adalah sebuah diagram struktural yang digunakan untuk merancang sebuah basis data. ERD akan mendeskripsikan data yang disimpan pada sebuah sistem maupun batasannya. ERD memiliki tiga konsep utama yaitu:

1. Entitas

Sebuah entitas dapat berupa orang, tempat, objek, atau kejadian yang dapat dianggap penting bagi sebuah organisasi atau perusahaan. Setiap entitas memiliki beberapa atribut yang mendeskripsikan karakteristik dari objek. Atribut yang ada dalam entitas harus disimpan dan dicatat dalam basis data. Entitas pada komponen ERD dapat dibedakan menjadi dua macam yaitu strong entity dan weak entity. Strong entity merupakan entitas yang tidak bergantung pada entitas lain atau entitas yang dapat berdiri sendiri. Sedangkan untuk weak entity merupakan entitas yang keberadaannya tergantung pada entitas lain.

2. Atribut

Setiap entitas memiliki karakteristik tertentu yang disebut dengan atribut. Atribut berfungsi untuk mendeskripsikan karakteristik yang ada pada entitas yang disimpan dalam basis data. Berdasarkan karakteristik sifatnya, atribut dapat dibedakan menjadi beberapa jenis yaitu simple attribute dan composite attribute, single valued attribute dan multi value attribute, derived attribute, key attribute. Primary key adalah nama untuk atribut yang digunakan dalam mengenali suatu entitas. Atribut dalam entitas yang merupakan primary key adalah kode identifikasi yang bersifat unik ditunjukkan berdasarkan masing-masing *record* pada sistem. Primary key bertujuan untuk memberitahu lokasi untuk tiap catatan pada suatu *file* tentang catatan-catatan yang sama.

3. Relasi

Relasi adalah sebuah hubungan antara dua atau lebih entitas yang saling berkaitan. Relasi pada ERD dapat digambarkan dengan menggunakan simbol belah ketupat (*diamond*). Relasi memiliki beberapa jenis relasi yaitu *unary, binary, ternary* [18].

Gambar 2.2 Bentuk entitas, relasi, dan atribut dalam ERD

2.5.1.2 Fungsi dan Manfaat ERD

ERD memiliki beberapa fungsi utama dalam proses perancangan *database*:

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

a. Visualisasi Struktur Data: ERD memberikan gambaran visual yang jelas mengenai struktur data dan relasi antar entitas, sehingga memudahkan komunikasi dan pemahaman antar anggota tim pengembang.
b. Dasar Perancangan Database: ERD menjadi acuan utama dalam pembuatan skema basis data yang efisien dan terstruktur sebelum tahap implementasi.
c. Mencegah Redundansi dan Anomali Data: Dengan pemodelan yang tepat, ERD membantu mengidentifikasi potensi duplikasi data dan anomali yang dapat terjadi dalam basis data, sehingga dapat dicegah sejak tahap desain.
d. Dokumentasi Sistem: ERD berperan sebagai dokumentasi penting yang dapat digunakan untuk pemeliharaan dan pengembangan sistem di masa mendatang [19].

2.5.1.3 Tantangan dan Kesalahan dalam Mendesain ERD

Meskipun ERD merupakan teknik yang esensial, banyak pengembang masih mengalami kesulitan dalam membuat ERD yang benar. Kesalahan umum yang sering terjadi antara lain:

a. Kesalahan Konseptual: Kurangnya pemahaman tentang komponen ERD seperti entitas, atribut, relasi, dan kardinalitas menyebabkan model yang dihasilkan tidak sesuai dengan kebutuhan sistem.
b. Kesalahan Prosedural: Tidak mengikuti langkah-langkah sistematis dalam mengidentifikasi entitas, atribut, dan relasi dapat menghasilkan ERD yang tidak konsisten.
c. Kesalahan Teknis: Kesalahan dalam penggunaan notasi atau penggambaran diagram, seperti bentuk simbol yang salah atau tidak adanya garis penghubung antar entitas dan relasi, dapat mengakibatkan kebingungan dalam interpretasi model.

Untuk mengatasi masalah ini, penting untuk memahami aturan-aturan dalam pembuatan ERD, seperti identifikasi entitas utama, penentuan atribut kunci, serta penetapan derajat kardinalitas pada setiap relasi. Selain itu, penggunaan pendekatan sistematis dan evaluasi berkala terhadap hasil desain sangat dianjurkan agar ERD yang dihasilkan benar-benar merepresentasikan kebutuhan sistem secara akurat [20].

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

2.5.1.4 Perkembangan dan Implementasi ERD

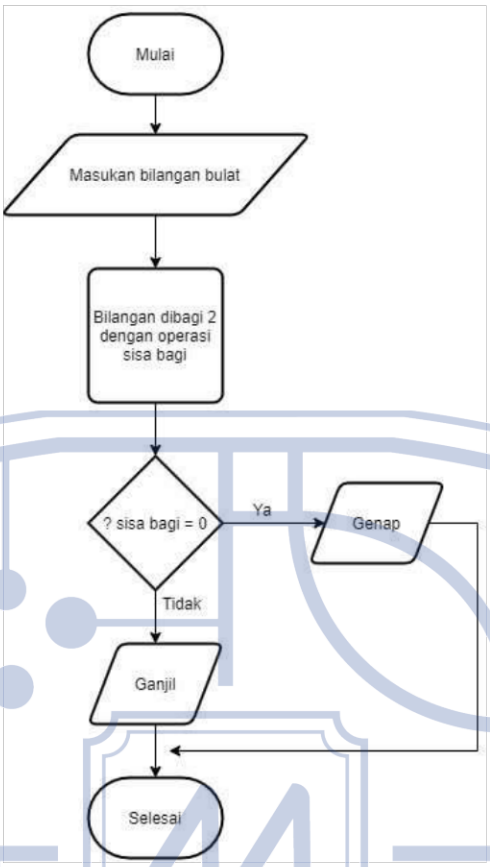
Seiring perkembangan teknologi, pembuatan ERD tidak hanya dilakukan secara manual, tetapi juga dapat diotomatisasi dengan bantuan perangkat lunak atau teknik seperti *Natural Language Processing* (NLP), yang dapat menghasilkan ERD langsung dari spesifikasi kebutuhan berbahasa alami. Hal ini mempercepat proses desain dan meminimalkan kesalahan manusia.

Dalam konteks pengembangan sistem informasi modern, ERD juga digunakan sebagai dasar dalam perancangan data *warehouse* dan sistem analitik, karena dapat membantu merancang struktur data yang optimal untuk pengambilan keputusan bisnis [21].

2.5.2 Flowchart

Flowchart atau bagan alir adalah representasi grafis dari algoritma atau proses yang menampilkan langkah-langkah dalam bentuk simbol-simbol grafis yang dihubungkan dengan panah untuk menunjukkan urutan operasi. *Flowchart* digunakan sebagai alat bantu komunikasi dan dokumentasi untuk membantu memvisualisasikan apa yang terjadi dalam program atau sistem tertentu. Tujuan utama *flowchart* adalah untuk menyajikan suatu tahapan penyelesaian masalah secara sederhana, terurai, rapi, dan jelas menggunakan simbol-simbol standar.

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.



Gambar 2.3 Contoh flowchart

2.5.2.1 Simbol-Simbol *Flowchart*

Flowchart menggunakan simbol-simbol standar yang telah ditetapkan oleh *American National Standard Institute* (ANSI) dan *International Organization for Standardization* (ISO). Beberapa simbol dasar yang umum digunakan dalam *flowchart*, antara lain:

1. Terminator (Oval), yaitu awal atau akhir konsep (prosedur)
2. Process (Persegi panjang), yaitu proses operasional
3. Document (Persegi panjang dengan sudut melengkung), yaitu dokumen atau laporan berupa print out
4. Decision (Belah ketupat), yaitu keputusan atau sub-point. Garis yang terhubung dengan bentuk decision merujuk pada situasi-situasi yang berbeda sesuai dengan keputusan yang digambarkan
5. Data (Jajar genjang), yaitu input dan output. Contohnya, Input: *feedback* dari pelanggan. Output: desain produk baru
6. *On-Page Reference/Connector* (Lingkaran), yaitu penghubung alur dalam halaman yang sama

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

7. *Off-Page Reference/Off-Page Connector* (Pentagon) yaitu penghubung alur dalam halaman yang berbeda

8. Flow (Panah) yaitu arah alur dalam konsep (prosedur)

Tabel 2.1 Simbol flowchart

No.	Simbol Flowchart	Nama	Arti Simbol Flowchart
1		Terminator	Awal atau akhir konsep (prosedur)
2		Process	Proses operasional
3		Document	Dokumen atau laporan berupa print out
4		Decision	Keputusan atau sub-point. Garis yang terhubung dengan bentuk decision merujuk pada situasi-situasi yang berbeda sesuai dengan keputusan yang digambarkan
5		Data	Input dan Output (Contohnya, Input: feedback dari pelanggan. Output: desain produk baru)
6		On-Page Reference/ Connector	Penghubung alur dalam halaman yang sama
7		Off-Page Reference/ Off-Page Connector	Penghubung alur dalam halaman yang berbeda
8		Flow	Arah alur dalam konsep (prosedur)

2.5.3 Use Case Scenario

Use case adalah teknik pemodelan untuk mendapatkan kebutuhan fungsional dari sebuah sistem. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara pengguna sebuah sistem (aktor) dengan sistemnya sendiri melalui sebuah cerita bagaimana sistem tersebut digunakan. *Use case* mengidentifikasi aktor yang terlibat dalam interaksi dan memberi nama tipe interaksi yang terjadi [13]. Setiap *use case* diasosiasikan dengan aktor, di mana aktor adalah entitas yang berinteraksi dengan sistem untuk melaksanakan tugas-tugas yang bermakna.

Use case berfungsi sebagai alat komunikasi yang efektif antara pengguna dan pengembang sistem. Dengan *use case*, analis sistem dapat memahami bagaimana sistem seharusnya berperilaku dari perspektif pengguna. *Use case* membantu dalam menangkap, mengorganisir dan mendokumentasikan kebutuhan sistem dalam konteks bisnis yang spesifik.

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

Gambar 2.4 Contoh use case diagram

2.5.4 Use Case Diagram

Use case diagram adalah diagram yang menggambarkan interaksi antara sistem dan aktor, serta menggambarkan fungsionalitas yang disediakan oleh sistem. *Use case* diagram merupakan salah satu diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk memodelkan aspek perilaku sistem.

Komponen utama dalam *use case* diagram meliputi:

1. Aktor

   Merepresentasikan peran yang dimainkan oleh pengguna sistem, perangkat keras, atau sistem lain yang berinteraksi dengan sistem yang sedang dimodelkan. Aktor digambarkan dengan simbol *stick figure*.

2. *Use Case*

   Menggambarkan fungsionalitas spesifik yang disediakan oleh sistem untuk aktor. *Use case* digambarkan dengan simbol oval yang berisi nama *use case* tersebut.

3. Sistem

   Menandai batasan sistem yang dimodelkan, biasanya digambarkan dengan kotak yang mengelilingi *use case*.

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

4. Hubungan (*Relationship*)

   Menggambarkan hubungan antara aktor dan *use case*, atau antar *use case*. Jenis-jenis hubungan meliputi:

   a. *Association*: Menunjukkan bahwa aktor berpartisipasi dalam *use case*
   b. *Include*: Menunjukkan bahwa suatu *use case* memasukkan fungsionalitas dari use case lain
   c. *Extend*: Menunjukkan bahwa suatu *use case* dapat diperpanjang oleh *use case* lain dalam kondisi tertentu
   d. *Generalization*: Menunjukkan hubungan umum-khusus antar aktor atau antar *use case*

Use case diagram memberikan gambaran visual tentang fungsi-fungsi sistem dan siapa saja yang berinteraksi dengan fungsi-fungsi tersebut, sehingga memudahkan komunikasi antara analis, pengembang, dan *stakeholder*.

2.5.5 Use Case Skenario/Narasi

Use case skenario atau narasi merupakan deskripsi tekstual yang menjelaskan secara detail bagaimana pengguna berinteraksi dengan sistem untuk mencapai tujuan tertentu. Berbeda dengan *use case* diagram yang memberikan gambaran visual pada tingkat yang lebih tinggi, skenario *use case* memberikan rincian yang lebih mendalam tentang interaksi pengguna-sistem.

Struktur umum dari sebuah *use case* skenario meliputi:

1. Nama *Use Case*

   Identifikasi unik dan deskriptif dari *use case*.
2. Aktor

   Pengguna atau sistem lain yang berinteraksi dengan sistem.
3. Tujuan

   Hasil yang ingin dicapai oleh aktor melalui use case ini.
4. *Precondition*

   Kondisi yang harus dipenuhi sebelum use case dapat dimulai.
5. *Postcondition*

   Kondisi yang harus dipenuhi setelah use case berhasil diselesaikan.
6. *Main Flow* (Alur Utama)

   Urutan langkah-langkah yang terjadi ketika segalanya berjalan normal (happy path).

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

7. *Alternative Flow* (Alur Alternatif)

   Variasi dari alur utama yang menangani situasi khusus atau pengecualian.

Use case skenario membantu pengembang memahami persyaratan dengan lebih terperinci dan memastikan bahwa sistem memenuhi kebutuhan pengguna dalam berbagai situasi.

2.5.5.1 Manfaat Penggunaan *Use Case*

Penggunaan *use case* dalam pengembangan sistem informasi memberikan beberapa manfaat signifikan, antara lain:

1. Fokus pada Kebutuhan Pengguna

   Use case membantu tim pengembangan untuk tetap fokus pada kebutuhan dan tujuan pengguna, bukan pada solusi teknis.
2. Komunikasi yang Lebih Baik

   Menyediakan bahasa umum yang dapat dipahami oleh *stakeholder* teknis maupun non-teknis untuk mendiskusikan fungsi sistem.
3. Identifikasi Cakupan Sistem

   Membantu dalam mengidentifikasi dan membatasi cakupan sistem dengan jelas.
4. Alat Validasi

   Dapat digunakan untuk memvalidasi bahwa sistem yang dikembangkan memenuhi kebutuhan yang telah ditetapkan.
5. Basis untuk Pengujian

   Use case dapat digunakan sebagai dasar untuk membuat kasus uji (*test cases*), memastikan semua interaksi pengguna-sistem telah diuji dengan baik.
6. Manajemen Kompleksitas

   Membantu memecah sistem yang kompleks menjadi interaksi-interaksi yang lebih sederhana dan mudah dipahami.
7. Dukungan untuk Estimasi dan Perencanaan

   Use case dapat digunakan untuk mengestimasi upaya pengembangan dan merencanakan iterasi pengembangan.

2.6. Basis Data (*Database*)

Kumpulan data yang terorganisir yang disimpan di komputer sehingga aplikasi yang mengambil data dari basis data dapat mengaksesnya disebut basis data. Basis data secara

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

umum didefinisikan sebagai sistem penyimpanan data yang menampung sejumlah besar data input dalam satu sistem penyimpanan. Saat ini *database* digunakan dalam berbagai macam bisnis, tidak hanya teknologi. Bisnis kecil dan besar, institusi, kantor, supermarket, dan bahkan tempat tinggal menggunakan *database* [22].

Database memainkan peran penting dalam sistem rekomendasi diet dengan menyimpan, mengelola, dan mengorganisir data yang diperlukan untuk memberikan rekomendasi yang akurat dan personal. Berikut adalah beberapa peran utama *database* dalam sistem tersebut [23]:

1. Menyimpan Data Pengguna

   Database menyimpan informasi pengguna, seperti nama, usia, jenis kelamin, berat badan, tinggi badan, serta riwayat kesehatan (misalnya diabetes, hipertensi, atau alergi makanan). Data ini digunakan untuk menyesuaikan rekomendasi diet dengan kondisi kesehatan masing-masing individu.

2. Mengelola Informasi Makanan dan Nutrisi

   Sistem rekomendasi produk meyimpan merek, kategori, kalori, dan *Nutri-Score* dari makanan atau minuman. Dengan informasi ini, pengguna dapat mengetahui dengan jelas nutrisi yang ada pada produk makanan atau minuman yang mereka konsumsi.

3. Menyimpan Preferensi dan Riwayat Diet

   Agar rekomendasi semakin personal, sistem perlu menyimpan preferensi pengguna, seperti makanan yang disukai atau dihindari. Selain itu, riwayat diet yang telah diikuti sebelumnya juga dicatat untuk memberikan saran yang lebih sesuai berdasarkan pola makan sebelumnya.

4. Memproses Aturan dan Algoritma Rekomendasi

   Database juga berfungsi sebagai tempat penyimpanan aturan diet berdasarkan standar kesehatan dan pola makan tertentu, seperti diet rendah gula, diet tinggi protein, atau pola makan vegetarian. Dengan adanya aturan ini, sistem dapat menggunakan algoritma rekomendasi untuk menghasilkan saran makanan yang sesuai bagi pengguna.

Untuk memastikan sistem basis data beroperasi dengan efisien, beberapa teknik optimasi dapat diterapkan, antara lain:

1. *Normalisasi*: Proses ini bertujuan mengurangi redundansi data dan memastikan integritas informasi dalam basis data. Dengan normalisasi, struktur data disusun sedemikian rupa sehingga efisien dan terorganisir dengan baik.

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

2. *Indexing*: Indeks digunakan untuk meningkatkan kecepatan akses data dalam basis data dengan mengorganisir cara penyimpanan dan pencarian data. Dengan indeks yang tepat, pencarian data dapat dilakukan lebih cepat dan efisien.
3. *Caching*: Teknik ini melibatkan penyimpanan sementara hasil *query* atau data yang sering diakses dalam memori sementara. Dengan menggunakan *caching*, *database* tidak perlu melakukan *query* berulang kali untuk data yang sama, sehingga mengurangi beban kerja dan meningkatkan kecepatan respons.
4. *Partitioning*: *Partitioning* adalah teknik yang membagi basis data besar menjadi bagian-bagian yang lebih kecil untuk meningkatkan efisiensi dan kinerja. Dengan membagi data menjadi partisi-partisi, proses pencarian dan pengolahan data dapat dilakukan lebih cepat dan efisien.

2.6.1 Primary Key

Primary Key (PK) dan *Foreign Key* (FK) merupakan konsep fundamental dalam perancangan database relasional yang berperan penting dalam menjaga integritas data dan mendefinisikan hubungan antar tabel. Implementasi yang tepat dari PK dan FK merupakan dasar bagi desain *database* yang efisien dan terstruktur.

Primary Key adalah kolom atau sekumpulan kolom yang secara unik mengidentifikasi setiap baris dalam tabel *database*. *Primary Key* sebagai atribut atau kombinasi atribut yang memiliki nilai unik untuk setiap entitas dalam relasi dan tidak dapat bernilai NULL. *Primary Key* memiliki dua karakteristik esensial:

1. *Uniqueness*: Setiap nilai dalam kolom Primary Key harus unik di seluruh tabel
2. Non-null: *Primary Key* tidak boleh mengandung nilai NULL

Primary Key dapat dibentuk melalui dua pendekatan utama:

1. *Natural Key*: *Key* yang berasal dari atribut yang sudah ada dalam data, seperti nomor KTP atau nomor telepon
2. *Surrogate Key*: *Key* buatan yang tidak memiliki makna bisnis, biasanya berupa nilai integer *auto-increment* atau UUID

2.6.2 Foreign Key

Foreign Key adalah kolom atau sekumpulan kolom dalam satu tabel yang mengacu pada *Primary Key* di tabel lain. Foreign Key berfungsi sebagai penghubung antar tabel yang memungkinkan database mempertahankan integrasi referensial. *Foreign Key* membantu mencegah tindakan yang akan menghancurkan hubungan antar tabel.

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

Foreign Key memiliki karakteristik berikut:

1. *Foreign Key* dalam satu tabel harus cocok dengan *Primary* Key di tabel yang direferensikan, atau bernilai NULL (dalam beberapa kasus)
2. *Foreign Key* mendefinisikan *relationship* (hubungan) antar tabel
3. *Foreign Key* mendukung pemeliharaan integritas referensial melalui *constraint*

Dengan desain *database* yang terstruktur dan dioptimalkan, sistem rekomendasi diet dapat berfungsi lebih cepat, akurat, dan dapat menangani banyak pengguna sekaligus. Hal ini akan meningkatkan pengalaman pengguna dalam mendapatkan rekomendasi diet yang sesuai dengan kebutuhan kesehatan mereka.

2.7. Sistem Rekomendasi

Sistem rekomendasi adalah sistem yang dapat memberikan rekomendasi pada suatu item tertentu yang dapat digunakan untuk membantu pengguna sistem dalam mengambil suatu keputusan. Sistem rekomendasi akan memberikan saran-saran yang berhubungan dengan proses pengambilan keputusan, misalnya dalam menentukan item mana yang ingin dibeli atau dimakan [24].

2.7.1. *Content-Based Filtering*

Content-based filtering merupakan sistem rekomendasi yang didasari dengan korelasi antara item. Metode tersebut menggunakan informasi item yang direpresentasikan sebagai atribut untuk dihitung nilai kemiripan antar item. Dalam memberikan rekomendasi item serupa kepada suatu pengguna berdasarkan riwayat item yang disukai, sistem rekomendasi content-based membutuhkan suatu item-item yang merepresentasikan pengguna dan keseluruhan item. Content-based filtering menggunakan konsep kemiripan konten.

Pertama, pengguna dan item direpresentasikan pada ruang fitur yang sama. Kemudian, hitung nilai kemiripan antara pengguna dengan item-item. Pada akhirnya, rekomendasi berhasil dibuat berdasarkan kemiripan suatu pengguna terhadap semua item. Teknik content-based filtering yang digunakan merekomendasikan item baru dengan membandingkannya dengan item lain yang terdapat pada profil pengguna. Metode content-based filtering memiliki performa yang lebih baik ketika pengguna memiliki banyak aktivitas dalam sistem.

Kelebihan sistem rekomendasi dengan content-based filtering adalah mempunyai kemampuan untuk merekomendasikan item yang sifatnya baru bagi pengguna. Kelebihan

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

utama dari metode ini adalah kemampuan merekomendasikan item yang tidak terlihat sehingga mampu menyelesaikan keterbatasan cold start problem terkait dengan item. Metode content-based filtering untuk pembuatan rekomendasi dikatakan layak dan sangat efektif.

2.7.2. *Collaborative Filtering*

Collaborative Filtering (CF) merupakan salah satu teknik yang paling populer dalam sistem rekomendasi. *Metode Collaborative filtering* adalah suatu proses untuk kegiatan dalam mengevaluasi item atau penyaringan item dengan menggunakan persepsi atau opini orang lain. *Collaborative filtering* melakukan proses penyaringan terhadap semua pengguna untuk mendapatkan informasi pengguna dalam memberikan suatu rekomendasi. Dalam proses penyaringan data *collaborative filtering* bekerja berdasarkan kemiripan karakteristik pengguna yang nantinya akan mampu memberikan suatu informasi yang baru kepada pengguna. Hal ini dikarenakan sistem akan memberikan informasi didasarkan pada pola dalam satu kelompok pengguna yang hampir mirip atau sama [25].

Keunggulan utama *collaborative filtering* dibandingkan dengan pendekatan rekomendasi lainnya adalah kemampuannya untuk memberikan rekomendasi yang personal tanpa memerlukan pemahaman mendalam tentang konten item itu sendiri. Sistem hanya perlu menganalisis pola interaksi antara pengguna dan item, yang membuatnya sangat fleksibel dan dapat diterapkan di berbagai domain yang berbeda.

Dalam implementasinya, *collaborative filtering* dapat menggunakan berbagai jenis data interaksi pengguna. Data eksplisit seperti rating atau penilaian langsung dari pengguna merupakan bentuk yang paling umum. Namun, sistem juga dapat memanfaatkan data implisit seperti riwayat klik, waktu yang dihabiskan pada suatu item, atau frekuensi akses untuk memahami preferensi pengguna [26].

Secara umum, terdapat dua pendekatan utama dalam *collaborative filtering*:

1. *User-Based Collaborative Filtering*

Pendekatan ini mencari pengguna lain yang memiliki preferensi serupa dengan pengguna target (disebut juga tetangga atau *neighbor*). Sistem kemudian menggunakan preferensi dari tetangga-tetangga ini untuk memprediksi item yang mungkin disukai oleh pengguna target. Misalnya, jika pengguna A dan B memiliki riwayat preferensi diet yang mirip, maka menu diet yang disukai oleh pengguna A mungkin juga akan direkomendasikan kepada pengguna B.

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

Dalam implementasi *user-based* CF, sistem pertama-tama membangun profil pengguna berdasarkan interaksi mereka dengan berbagai item. Profil ini kemudian digunakan untuk menghitung kemiripan antar pengguna menggunakan berbagai metrik *similarity*. Setelah menemukan kelompok pengguna yang mirip, sistem dapat membuat prediksi preferensi dengan mengagregasi preferensi dari kelompok tersebut.

2. *Item-Based Collaborative Filtering*

Metode ini fokus pada hubungan antar item, bukan antar pengguna. Sistem mencari item yang memiliki pola penilaian serupa dari pengguna-pengguna yang sama. Pendekatan ini umumnya lebih stabil dan skalabel dibandingkan *user-based* CF karena preferensi terhadap item cenderung lebih stabil dibandingkan preferensi pengguna yang dapat berubah seiring waktu.

Item-based CF bekerja dengan membangun model kemiripan antar item berdasarkan pola penilaian pengguna. Model ini kemudian digunakan untuk mengidentifikasi item yang memiliki karakteristik serupa berdasarkan bagaimana pengguna berinteraksi dengan item-item tersebut. Pendekatan ini sangat efektif terutama ketika jumlah item lebih sedikit dibandingkan jumlah pengguna.

Dalam implementasi collaborative filtering, salah satu teknik yang paling umum digunakan untuk mengukur kemiripan antar item adalah cosine similarity. Cosine similarity mengukur sudut antara dua vektor dalam ruang multidimensi. Dalam konteks sistem rekomendasi, vektor tersebut mewakili pola rating dari item oleh berbagai pengguna.

Misalnya, dua produk akan direpresentasikan sebagai dua vektor berdasarkan rating yang diberikan oleh pengguna. Semakin kecil sudut antara kedua vektor tersebut (semakin dekat arah keduanya), maka semakin besar nilai cosine similarity-nya, yang berarti kedua produk dianggap mirip dalam persepsi pengguna.

Rumus Cosine Similarity adalah sebagai berikut:

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Keterangan:

- A dan B = dua buah item yang ingin dibandingkan
- A_i, B_i = nilai rating dari pengguna ke-i terhadap item A dan B
- A·B = hasil *dot product* dari dua vektor
- ‖A‖ dan ‖B‖ = panjang (norma) dari masing-masing vektor

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

Nilai cosine similarity berada dalam rentang 0 hingga 1 (untuk data rating non-negatif), di mana:

- 1 menunjukkan kemiripan sempurna (arah vektor sama)
- 0 menunjukkan tidak ada kemiripan sama sekali (arah tegak lurus)

Misalkan dua item makanan I1 dan I2 mendapat rating dari tiga pengguna sebagai berikut:

Tabel 2.2 Contoh data dua item makanan untuk perhitungan *similarity*

User	I1	I2
U1	5	4
U2	4	5
U3	2	2

Maka:

- Vektor A=[5,4,2]
- Vektor B=[4,5,2]

$$AA \cdot BB = (5 \times 4) + (4 \times 5) + (2 \times 2) = 20 + 20 + 4 = 44$$

$$\|AA\| = \sqrt{5^2 + 4^2 + 2^2} = \sqrt{25 + 16 + 4} = \sqrt{45} \approx 6.708$$

$$\|BB\| = \sqrt{4^2 + 5^2 + 2^2} = \sqrt{16 + 25 + 4} = \sqrt{45} \approx 6.708$$

$$ssssss(AA, BB) = \frac{44}{6.708 \times 6.708} = \frac{44}{45} \approx 0.978$$

Artinya, kedua item memiliki kemiripan yang sangat tinggi, sehingga bila seorang pengguna menyukai I1, maka kemungkinan besar ia juga akan menyukai I2.

Keberhasilan implementasi *collaborative filtering* dalam sistem rekomendasi diet sangat bergantung pada kualitas dan kuantitas data yang tersedia. Semakin banyak interaksi pengguna yang tercatat, semakin baik sistem dapat memahami pola preferensi dan memberikan rekomendasi yang relevan. Oleh karena itu, perancangan sistem perlu mempertimbangkan cara untuk mendorong *engagement* pengguna dan mengumpulkan feedback yang berkualitas [30].

Pendekatan *item based collaborative filtering* memberikan rekomendasi berdasarkan kemiripan antar item sedangkan pada pendekatan *user based collaborative filtering* sistem memberikan rekomendasi kepada user item yang disukai atau dinilai oleh user lain yang memiliki kemiripan. Kelebihan dari pendekatan *user based collaborative filtering* adalah dapat menghasilkan rekomendasi yang berkualitas baik. Sedangkan kekurangannya adalah kompleksitas perhitungan akan semakin bertambah seiring

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

dengan bertambahnya *user* sistem, semakin banyak user yang menggunakan sistem maka proses rekomendasi akan semakin lama.

2.7.3. *Hybrid Recommendation*

Hybrid Recommendation merupakan pendekatan dalam sistem rekomendasi yang menggabungkan dua atau lebih teknik rekomendasi, seperti collaborative filtering dan content-based filtering, untuk memperoleh hasil rekomendasi yang lebih akurat dan relevan. Pendekatan ini muncul sebagai solusi atas keterbatasan masing-masing metode, seperti masalah cold start, data sparsity, dan over-specialization, yang sering kali mengurangi kualitas rekomendasi jika hanya menggunakan satu teknik saja [27].

Dalam konteks sistem rekomendasi makanan dan minuman, hybrid recommendation dapat mengombinasikan data rating pengguna (seperti pada collaborative filtering) dengan informasi konten produk, seperti kandungan nutrisi, kategori makanan, atau preferensi diet tertentu (seperti pada content-based filtering) [28]. Sebagai contoh, sistem dapat merekomendasikan makanan berdasarkan rating dari pengguna lain yang memiliki selera serupa, namun sekaligus mempertimbangkan kebutuhan diet pengguna seperti rendah kalori atau tinggi protein.

Terdapat beberapa pendekatan umum dalam hybrid recommendation, di antaranya:

1. Weighted Hybrid: Masing-masing metode menghasilkan skor rekomendasi yang kemudian digabungkan dengan bobot tertentu.
2. Switching Hybrid: Sistem memilih salah satu metode tergantung pada kondisi tertentu, misalnya memilih content-based filtering jika data pengguna masih sedikit.
3. Mixed Hybrid: Sistem menampilkan rekomendasi dari berbagai metode secara bersamaan.
4. Feature Combination: Fitur dari metode yang berbeda digabungkan ke dalam satu model pembelajaran.
5. Cascade Hybrid: Hasil dari satu metode menjadi input untuk metode berikutnya, misalnya collaborative filtering digunakan terlebih dahulu, lalu content-based filtering menyaring hasilnya.

Keunggulan dari pendekatan hybrid ini adalah kemampuannya dalam menghasilkan rekomendasi yang lebih personal dan tahan terhadap kelemahan dari masing-masing metode. Selain itu, hybrid system mampu menangani pengguna baru dengan menggunakan data konten, sekaligus memanfaatkan informasi dari komunitas pengguna melalui teknik kolaboratif [28].

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

Namun, penerapan sistem hybrid juga memiliki tantangan tersendiri, seperti kompleksitas sistem yang lebih tinggi, kebutuhan akan integrasi antar model, serta peningkatan beban komputasi. Oleh karena itu, dalam merancang sistem rekomendasi berbasis hybrid, diperlukan analisis yang cermat terhadap data pengguna, jenis produk, serta tujuan rekomendasi agar penggabungan metode dapat dilakukan secara optimal.

2.7.4. *Deep Learning Based Recommendation*

Sistem rekomendasi berbasis *Deep Learning* adalah pendekatan modern yang memanfaatkan jaringan saraf tiruan (*neural networks*) untuk membangun model rekomendasi yang mampu mempelajari pola-pola kompleks dari data pengguna dan item. Pendekatan ini berkembang seiring meningkatnya kebutuhan akan akurasi rekomendasi yang lebih tinggi dan kemampuan untuk menangani data dalam jumlah besar dan beragam, seperti teks, gambar, suara, hingga interaksi pengguna [29].

Berbeda dengan metode konvensional seperti collaborative filtering atau content-based filtering yang biasanya menggunakan pendekatan statistik sederhana atau berbasis kemiripan, sistem rekomendasi berbasis deep learning mampu melakukan pembelajaran representasi (*representation learning*) [30]. Artinya, model tidak hanya mengandalkan fitur eksplisit seperti rating atau atribut produk, tetapi juga mampu mengekstraksi fitur laten secara otomatis dari data mentah.

Beberapa arsitektur deep learning yang umum digunakan dalam sistem rekomendasi antara lain:

1. Neural Collaborative Filtering (NCF): Merupakan pengembangan dari collaborative filtering yang mengganti perhitungan similarity tradisional dengan jaringan saraf untuk mempelajari interaksi non-linear antara pengguna dan item.
2. Autoencoder: Digunakan untuk menangani data yang bersifat sparse (jarang) seperti matriks rating, dengan cara merekonstruksi input dan mempelajari representasi tersembunyi dari preferensi pengguna.
3. Recurrent Neural Network (RNN): Cocok digunakan untuk sistem rekomendasi berbasis urutan (*sequential recommendation*), misalnya untuk menganalisis pola konsumsi makanan dari waktu ke waktu.
4. Convolutional Neural Network (CNN): Umumnya digunakan ketika sistem rekomendasi melibatkan data visual, seperti gambar makanan, untuk mengenali fitur visual yang relevan terhadap preferensi pengguna.

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.

5. Attention Mechanism dan Transformer: Digunakan untuk menangkap hubungan kompleks antar item dan pengguna, dan banyak digunakan dalam sistem rekomendasi modern seperti YouTube dan Amazon.

Dalam sistem rekomendasi makanan dan minuman, deep learning dapat diterapkan untuk mengolah data ulasan teks, deskripsi produk, dan bahkan informasi gambar kemasan makanan. Sebagai contoh, sistem dapat mempelajari bahwa seorang pengguna menyukai makanan tinggi protein berdasarkan pola konsumsi sebelumnya, lalu secara otomatis merekomendasikan produk lain yang memiliki kandungan protein serupa tanpa harus secara eksplisit didefinisikan.

Keunggulan sistem rekomendasi berbasis deep learning adalah kemampuannya dalam menggeneralisasi pola, menangani data besar, serta memberikan hasil rekomendasi yang lebih personal dan kontekstual [29]. Namun demikian, pendekatan ini juga memiliki tantangan, seperti kebutuhan akan data dalam jumlah besar, waktu pelatihan yang lama, dan kesulitan interpretasi hasil karena sifat model yang *black-box*.

Oleh karena itu, dalam pengembangan sistem rekomendasi yang berbasis deep learning, diperlukan infrastruktur komputasi yang memadai, dataset yang kaya, serta evaluasi berkala untuk memastikan bahwa model tetap memberikan rekomendasi yang relevan dan bermanfaat bagi pengguna.

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.