BAB II
KAJIAN LITERATUR

2.1 Sistem Rekomendasi

Sistem rekomendasi merupakan program atau sistem penyaringan informasi yang dirancang untuk mengatasi permasalahan keterlebihan informasi dengan cara menyaring informasi yang relevan dari banyaknya informasi yang tersedia. Sistem ini bersifat dinamis karena mampu menyesuaikan rekomendasi berdasarkan preferensi, minat, atau perilaku pengguna terhadap suatu barang atau layanan [10].

Dalam implementasinya, sistem rekomendasi memanfaatkan data historis yang dianalisis menggunakan model rekomendasi untuk menghasilkan prediksi terhadap item yang berpotensi diminati oleh pengguna. Seiring dengan perkembangan internet dan perdagangan elektronik, sistem rekomendasi banyak diadopsi oleh perusahaan sebagai strategi untuk meningkatkan penjualan, dimana Sebagian besar algoritma yang digunakan berfokus pada kesesuaian rekomendasi dengan preferensi pengguna [11], [12].

Secara umum, sistem rekomendasi diklasifikasikan menjadi tiga pendekatan utama, yaitu:

1. Content-Based Filtering
2. Collaborative Filtering
3. Hybrid Recommendation System [13].

2.1.1 Content-Based Filtering

Content-Based Filtering (CBF) merupakan teori penting dalam sistem rekomendasi yang berfokus pada analisis konten dan atribut dari suatu item untuk menghasilkan rekomendasi yang dipersonalisasi. Pendekatan ini didasarkan pada asumsi bahwa item yang memiliki kesamaan fitur atau karakteristik dengan item yang disukai pengguna di masa lalu, cenderung akan disukai kembali oleh pengguna tersebut. Dalam penerapannya, item dideskripsikan menggunakan berbagai atribut, seperti nama item, kategori, atau bahan pembuatannya [14].

Metode ini mengevaluasi fitur dari item yang direkomendasikan—seperti contohnya pada spesifikasi sebuah hotel lalu mencocokkannya dengan riwayat pengguna. Sistem akan merekomendasikan item baru berdasarkan tingkat kesamaan dengan item yang telah diinteraksikan sebelumnya. Dengan demikian, *content-based filtering* dapat memberikan rekomendasi yang sangat relevan dengan preferensi individu pengguna secara mandiri, tanpa harus bergantung pada data interaksi dari pengguna lain [15].

Meskipun dapat memberikan rekomendasi secara mandiri, metode *content-based filtering* memiliki beberapa keterbatasan. Kelemahan utamanya adalah sistem memiliki kecenderungan untuk menyarankan item yang sangat mirip dengan apa yang sudah disukai pengguna sebelumnya. Hal ini membuat rekomendasi yang dihasilkan menjadi terlalu aman dan dapat diprediksi, serta membatasi munculnya variasi item baru (*limit novelty*) bagi pengguna. Selain itu, efektivitas metode ini sangat bergantung pada kualitas teks atributnya, sehingga sistem dapat mengabaikan karakteristik item yang tidak tercermin dengan baik dalam data tekstual tersebut [16].

2.1.2 Collaborative Filtering

Collaborative Filtering (CF) adalah metode sistem rekomendasi yang memberikan saran item berdasarkan pola preferensi pengguna lain yang memiliki kesamaan karakteristik atau perilaku. Metode ini bekerja dengan asumsi bahwa pengguna yang memiliki pola interaksi serupa pada masa lalu cenderung memiliki preferensi yang sama di masa mendatang [17].

Berbeda dengan content-based filtering yang berfokus pada atribut item, collaborative filtering hanya memanfaatkan data interaksi pengguna, seperti rating, histori pembelian, atau klik. Hal ini membuat metode ini sangat efektif dalam lingkungan dengan data interaksi yang cukup besar [12].

Collaborative Filtering terbagi menjadi dua pendekatan utama, yaitu:

1. User-Based Collaborative Filtering

   User-Based Collaborative Filtering adalah metode yang memberikan rekomendasi dengan cara mencari pengguna lain yang memiliki pola preferensi mirip dengan pengguna target. Setelah ditemukan pengguna yang memiliki tingkat kemiripan tinggi (nearest neighbors), sistem akan merekomendasikan item yang disukai oleh pengguna tersebut tetapi belum pernah diakses oleh pengguna target [12], [17].

Proses kerja metode ini meliputi:

a) Menghitung kemiripan antar pengguna
b) Menentukan k pengguna dengan nilai kemiripan tertinggi
c) Mengambil item yang disukai oleh pengguna tersebut
d) Menghasilkan rekomendasi untuk pengguna target

Kualitas rekomendasi sangat dipengaruhi oleh jumlah dan kelengkapan data historis pengguna. Semakin banyak data interaksi yang tersedia, maka semakin akurat

perhitungan kemiripan yang dihasilkan [11].

2. Item-Based Collaborative Filtering

   Item-Based Collaborative Filtering memberikan rekomendasi berdasarkan kemiripan antar item. Jika seorang pengguna menyukai suatu item, maka sistem akan mencari item lain yang memiliki pola interaksi serupa berdasarkan pengguna lain [17]. Keunggulan metode ini adalah stabilitasnya ketika jumlah pengguna sangat besar, karena jumlah item biasanya lebih tetap dibandingkan jumlah pengguna. Oleh karena itu, pendekatan ini banyak digunakan dalam sistem rekomendasi e-commerce [12].

2.1.3 Hybrid Recommendation Sistem

Hybrid Recommendation Sistem merupakan pendekatan yang menggabungkan beberapa metode penyaringan sekaligus untuk mencapai tingkat akurasi rekomendasi yang lebih tinggi. Pendekatan hybrid terbukti sangat efektif dalam mengatasi berbagai kelemahan fundamental yang sering muncul pada algoritma tunggal, seperti masalah *data sparsity* (keterbatasan ketersediaan data interaksi) dan *cold start* (kondisi di mana pengguna baru belum memiliki riwayat aktivitas yang memadai). Sebagai contoh penerapannya, sebuah model rekomendasi hybrid dapat memadukan *Collaborative Filtering* dengan penyaringan berbasis sosial dan semantik memanfaatkan atribut seperti tingkat kepercayaan, jaringan pertemanan, serta kesamaan minat guna menghasilkan rekomendasi yang jauh lebih personal dan relevan bagi pengguna [18].

Dalam perkembangannya, sistem *hybrid filtering* dapat diimplementasikan melalui berbagai kombinasi algoritma. Salah satu bentuk yang efektif adalah integrasi antara *rule-based system* dan *user-based collaborative filtering*. *Rule-based system* berperan dalam menyaring hasil berdasarkan preferensi eksplisit pengguna seperti jarak, harga, dan jam operasional karena sifatnya yang sederhana namun fleksibel untuk diadaptasi pada berbagai masalah Sementara itu, *user-based collaborative filtering* digunakan untuk melengkapi kekurangan tersebut dengan menyarankan item berdasarkan kesamaan pola historis antar pengguna [19].

Hybrid system sangat efektif dalam menangani masalah *cold start* bagi pengguna maupun item baru karena tetap dapat memberikan rekomendasi berbasis konten saat data interaksi pengguna masih jarang. Penggunaan teknik seperti analisis sentimen dan penambangan ulasan multi-kriteria juga terbukti dapat meningkatkan kualitas rekomendasi, terutama dalam skenario dengan peringkat pengguna yang terbatas. Namun, terlepas dari

berbagai keunggulannya dalam memberikan personalisasi dan cakupan rekomendasi yang luas, implementasi *hybrid recommender systems* memiliki tantangan signifikan. Proses pengembangan sistem ini dapat menjadi sangat kompleks dan bersifat *resource-intensive* atau membutuhkan sumber daya yang besar dalam pengoperasiannya [20].

2.2 Neural Collaborative Filtering (NCF)

Neural Collaborative Filtering (NCF) merupakan pengembangan dari metode *Collaborative Filtering* tradisional dengan memanfaatkan arsitektur jaringan saraf tiruan (*Artifical Neural Network*). Pendekatan ini dirancang untuk memodelkan interaksi non-linear antara pengguna dan item secara lebih kompleks dibandingkan metode konvensional berbasis *dot product* atau *similarity* sederhana [8]. NCF menggantikan fungsi interaksi linear dengan *Multilayer Perceptron* (MLP), sehingga mampu menangkap hubungan laten yang lebih kompleks antara pengguna dan item [21].

Lebih lanjut, Mulyana dan Rumaisa [7] menguraikan bahwa arsitektur komputasi NCF secara umum terdiri dari tiga bagian utama, yaitu *Embedding Layer*, *Hidden Layer* (MLP), dan *Output Layer*. Pada konteks sistem rekomendasi dengan umpan balik implisit, Proses perhitungan matematisnya didefinisikan sebagai berikut:

1. Embedding Layer dan Concatenation

   Identitas pengguna (u) dan item (i) direpresentasikan sebagai vektor padat (dense vector) melalui embedding. Dua vektor ini kemudian digabungkan (concatenation) untuk membentuk vektor input awal yang akan diteruskan ke jaringan saraf.

$$z_1 = (p_u, q_i)^T \tag{2.1}$$

   di mana:

   p_u = vektor embedding pengguna

   q_i = vektor embedding item

2. Hidden Layer (Multilayer Perceptron)

   Vektor gabungan diproses menggunakan beberapa lapisan tersembunyi pada jaringan MLP. Setiap lapisan mempelajari pola interaksi non-linear melalui matriks bobot (W), vektor bias (b), dan fungsi aktivasi non-linear seperti Rectified Linear Unit (ReLU). Perhitungan pada lapisan ke-l dapat dituliskan sebagai:

$$z_l = a_l\left(W_l^T z_{l-1} + b_l\right) \tag{2.2}$$

di mana:

z_l = output dari lapisan ke-l

W_l = matriks bobot

b_l = bias

a_l = fungsi aktivasi

3. Output Layer

Setelah melalui lapisan tersembunyi terakhir, representasi data diteruskan ke output layer untuk menghasilkan skor prediksi interaksi. Pada sistem dengan umpan balik implisit (implicit feedback), fungsi aktivasi sigmoid sering digunakan untuk memetakan nilai prediksi menjadi probabilitas antara 0 dan 1.

$$\hat{y}_{ui} = \sigma(h^T z_L) \quad (2.3)$$

di mana:

$\hat{y}_{ui}$ = nilai prediksi interaksi antara pengguna dan item

σ = fungsi sigmoid

h = vektor bobot pada lapisan output

4. Model Optimization (Loss Function)

Untuk mengoptimalkan parameter model selama proses pelatihan, NCF menggunakan fungsi Binary Cross-Entropy (BCE) untuk mengevaluasi kesalahan prediksi terhadap data interaksi aktual. Fungsi ini membantu meminimalkan perbedaan antara nilai prediksi dan data sebenarnya sehingga model dapat belajar secara lebih akurat.

$$L = -\sum \left(y_{ui} \, log \, (\widehat{y_{ui}}) + (1 - y_{ui}) \, log \, (1 - \widehat{y_{ui}}) \right) \quad (2.4)$$

di mana:

L = nilai loss

y_{ui} = nilai interaksi sebenarnya antara pengguna dan item

$\hat{y}_{ui}$ = nilai prediksi dari model

5. Binary Cross-Entropy (BCE)

Cross-entropy merupakan suatu ukuran diskrepansi atau perbedaan antara dua buah distribusi probabilitas. Kinerja dari suatu model dievaluasi menggunakan *cross-entropy loss*. Nilai probabilitas keluaran dari fungsi *cross-entropy loss* berada pada rentang antara 0 dan

1. Nilai ini akan meningkat ketika probabilitas prediksi dari model yang termasuk dalam kelas sebenarnya menjadi semakin tinggi.

Proses ini dapat dilihat sebagai sebuah algoritma klasifikasi. *Binary cross-entropy* (BCE) adalah bentuk spesifik dari *cross-entropy*, di mana target prediksinya hanya bernilai 1 atau 0. *Cross-entropy* menjadi fungsi kerugian (*loss function*) standar (*default*) yang sering digunakan karena memiliki kaitan secara matematis dengan tingkat akurasi model.
Secara matematis, rumus *Log loss* (untuk menghitung BCE) didefinisikan sebagai berikut [22]:

$$L = -p \cdot log(\hat{p}) + (1 - p) \cdot log(1 - \hat{p}) \quad (2.5)$$

Penjelasan dari formulasi tersebut adalah sebagai berikut:

a. Variabel p dalam rumus ini merupakan probabilitas dari kelas 1.
b. Variabel $(1 - p)$ merupakan probabilitas dari kelas 0.
c. Ketika suatu data termasuk ke dalam kelas 1, maka bagian pertama dari rumus tersebut yang akan dilibatkan.
d. Sebaliknya, jika data tersebut termasuk ke dalam kelas 0, maka bagian kedua dari persamaan tersebut yang akan menjadi aktif. Melalui proses inilah nilai *Binary cross-entropy* dapat dihitung.

2.3 Data Implisit dan Eksplisit

Dalam pengembangan sistem rekomendasi, ketersediaan data interaksi pengguna merupakan faktor yang krusial. Secara umum, data interaksi dapat berupa umpan balik eksplisit maupun implisit. Umpan balik implisit merupakan data interaksi yang tidak diinput secara langsung atau sadar oleh pengguna [23]. Jenis data ini sangat mencerminkan perilaku alami pengguna saat berinteraksi dengan sebuah platform, sehingga menjadikannya jauh lebih mudah untuk dikumpulkan dibandingkan dengan penilaian eksplisit seperti pemberian rating [24]. Beberapa bentuk interaksi yang dikategorikan sebagai umpan balik implisit meliputi riwayat pembelian di toko online, histori klik pada situs web berita, hingga aktivitas pemutaran pada layanan radio online [24]. Keuntungan utama dari pemanfaatan pendekatan umpan balik implisit adalah ketersediaan datanya yang sangat melimpah karena sistem merekam setiap aktivitas pengguna secara otomatis [25].

Sebagai perbandingan, umpan balik eksplisit mengacu pada masukan langsung yang diberikan oleh pengguna untuk mengekspresikan preferensi mereka, seperti

menentukan kata kunci, memberikan peringkat (rating), atau menjawab pertanyaan tentang minat mereka. Umpan balik ini biasanya mewakili penilaian kualitas yang tidak ambigu, disengaja, dan dilakukan secara sadar oleh pengguna. Perilaku umpan balik eksplisit semuanya didukung oleh fitur platform tertentu, seperti menandai postingan dengan "Tidak tertarik", melaporkan, atau memblokir. Mekanisme umpan balik eksplisit ini memberikan transparansi, namun memiliki kelemahan karena menuntut upaya kognitif dan biaya pengguna yang lebih tinggi di luar perilaku interaksi normal mereka [26].

Dalam hal keunggulan penggunaannya, pilihan jenis umpan balik sangat terkait dengan tujuan spesifik pengguna di dalam platform. Umpan balik eksplisit dianggap lebih cepat dan lebih langsung untuk mengatasi masalah, sehingga menjadi pilihan utama untuk tujuan mengurangi konten yang tidak pantas, penuh iklan, atau mengganggu. Di sisi lain, keunggulan dari variasi data implisit saat ini mencakup umpan balik implisit yang disengaja (intentional implicit feedback). Pengguna secara sadar melakukan perilaku implisit, seperti memicu pencarian baru atau melewati (swipe) postingan secara cepat, yang terbukti sangat krusial dan efektif untuk meningkatkan keragaman konten (content diversity) serta menghindari konten yang homogen atau "kepompong informasi". Selain itu, dibandingkan dengan umpan balik eksplisit yang membutuhkan interupsi langsung, memandu rekomendasi melalui perilaku implisit yang disengaja ini membuat interaksi menjadi lebih mulus dan membebaskan sumber daya kognitif pengguna [26].

2.4 Evaluasi Sistem Rekomendasi

Untuk mengukur seberapa baik model *Neural Collaborative Filtering* (NCF) dalam memberikan rekomendasi yang relevan, penelitian ini menggunakan dua metrik evaluasi berbasis *Top-N recommendation*, yaitu *Normalized Discounted Cumulative Gain* (NDCG) dan *Hit Ratio* (HR) Sebelum evaluasi, data dibagi menggunakan strategi *Leave-One-Out*.

2.4.1 Leave-One-Out Cross-Validation

Leave-One-Out Cross-Validation (LOO-CV) merupakan pendekatan yang populer digunakan untuk mengestimasi performa prediktif dari sebuah model. Dalam metode ini, setiap observasi tunggal dari data digunakan sebagai satu set validasi (*validation set*) secara bergantian. Sementara itu, observasi data lainnya yang tersisa digunakan untuk melatih dan membangun model tersebut.

Strategi ini sangat berguna ketika data uji (*test data*) yang independen tidak tersedia. Melalui LOO-CV, sistem dapat mengevaluasi seberapa baik performa model

terhadap data yang belum pernah dilihat sebelumnya (*unseen data*) dengan cara menggunakan kembali sejumlah observasi yang ada sebagai proksi untuk merepresentasikan distribusi data di masa depan [27].

2.4.2 Ground Truth

Kinerja suatu model prediktif diukur berdasarkan kemampuannya dalam merepresentasikan kondisi dunia nyata. Evaluasi menggunakan metrik berbasis *ground truth* dilakukan untuk mengukur seberapa presisi prediksi yang dihasilkan oleh model dengan status aktual yang benar-benar diobservasi. Dalam konteks sistem rekomendasi, penggunaan titik data *ground truth* yang independen dan dipisahkan dari proses pelatihan model ini sangat esensial untuk memastikan penilaian akurasi prediksi menu berjalan secara objektif dan tidak bias [28].

2.4.3 Normalized Discounted Cumulative Gain (NDCG)

Normalized Discounted Cumulative Gain (NDCG@k) digunakan untuk mengukur seberapa baik sistem menempatkan item (menu) yang paling relevan di posisi teratas dalam daftar rekomendasi dengan panjang k. Metrik ini memperhitungkan relevansi dan posisi item, sehingga rekomendasi menu dengan relevansi tinggi yang muncul lebih awal akan mendapatkan bobot yang lebih besar.

Perhitungan NDCG@k dilakukan dalam dua tahapan. Pertama, menghitung *Discounted Cumulative Gain* (DCG@k):

$$DCGk = \sum_{i=1}^{k} \frac{2^{rel_i - 1}}{\log_2(i+1)} \tag{2.6}$$

dengan ($rel\ i$) adalah relevansi item pada posisi (i). Kedua, DCG@k dinormalisasi dengan Ideal DCG(IDCG@k), yaitu DCG dari urutan ideal:

$$NDCGk = \frac{DCG@k}{IDCG@k} \tag{2.7}$$

Nilai NDCG@k berada pada rentang 0 hingga 1, di mana nilai yang semakin mendekati 1 menandakan bahwa urutan daftar rekomendasi menu dari model hampir sama baiknya dengan urutan ideal. Pemilihan metrik ini sangat penting karena mencerminkan pengalaman pengguna secara langsung, di mana relevansi dan posisi rekomendasi menu

yang tepat akan sangat memengaruhi kepuasan pelanggan terhadap sistem yang diimplementasikan [8].

2.4.4 Hit Ratio (HR)

Hit Ratio merupakan metrik evaluasi berbasis *Top-N recommendation* yang digunakan untuk mengukur keberhasilan sistem dalam merekomendasikan item yang relevan kepada pengguna. Metrik ini banyak digunakan dalam penelitian sistem rekomendasi modern karena mampu memberikan gambaran sederhana mengenai kemampuan sistem dalam menemukan item yang relevan [29], [30].

Hit Ratio berfokus pada apakah item yang benar-benar relevan (*ground truth*) berhasil muncul dalam daftar rekomendasi Top-K yang dihasilkan oleh sistem. Jika item relevan tersebut muncul dalam daftar rekomendasi, maka dihitung sebagai *hit* (bernilai 1), sedangkan jika tidak muncul maka dihitung sebagai *miss* (bernilai 0) [31].

$$HR = \frac{\text{Jumlah hit}}{\text{Jumlah pengguna}} \tag{2.8}$$

Penjelasan:

a. Hit: Bernilai 1 jika item yang relevan (misalnya item yang benar-benar disukai user) muncul dalam daftar Top-K rekomendasi, dan 0 jika tidak.
b. Jumlah pengguna: Total user yang diuji dalam evaluasi.