

## **BAB 2**

### **LANDASAN TEORI**

Pada bagian ini dijabarkan teori-teori yang mendasari penelitian secara lengkap dan menyeluruh. Penyajiannya dapat berupa definisi dan model Matematika yang langsung berkaitan dengan ilmu atau masalah yang diteliti.

#### **2.1 Sistem Rekomendasi**

Sistem rekomendasi merupakan sistem penyaringan informasi yang secara dinamis menyeleksi dan menampilkan informasi paling relevan sesuai dengan preferensi, minat, dan perilaku pengguna [5]. Dengan keberadaan sebuah sistem rekomendasi, seseorang akan mendapatkan rekomendasi keputusan yang sesuai dengan preferensi dan kriteria yang diinginkan [6].

Sistem rekomendasi memiliki beragam metode yang dapat digunakan untuk menghasilkan rekomendasi yang sesuai dengan kebutuhan pengguna, antara lain yaitu Content-Based Filtering dan Association Rule

##### **2.1.1 Content-Based Filtering**

Metode *Content-Based Filtering* (CBF) merupakan salah satu pendekatan dalam sistem rekomendasi yang menghasilkan rekomendasi berdasarkan kesesuaian karakteristik item dengan preferensi pengguna. Prinsip utama dari metode ini adalah bahwa pengguna cenderung menyukai item yang memiliki kemiripan dengan item yang sebelumnya pernah disukai atau digunakan [7]. Oleh karena itu, rekomendasi dihasilkan tanpa bergantung pada data pengguna lain, melainkan fokus pada profil dan riwayat interaksi pengguna itu sendiri.

Dalam konteks rekomendasi video game, atribut yang digunakan dapat berupa *genre*, *kategori*, *developer*, atau *platform*. Sistem akan membangun profil preferensi pengguna berdasarkan game yang telah dimainkan atau disukai, kemudian mencocokkannya dengan karakteristik game lain yang tersedia. Pendekatan ini mengasumsikan bahwa kesamaan fitur mencerminkan kesamaan minat pengguna terhadap item tersebut.

Secara umum, proses dalam metode *Content-Based Filtering* dapat dijelaskan sebagai berikut:

1. Menentukan fitur atau atribut dari setiap game, seperti genre, kategori, atau tag.
2. Mengubah setiap game menjadi representasi vektor berdasarkan fitur yang dimilikinya.
3. Menghitung tingkat kesamaan antara dua game menggunakan ukuran kesamaan (*similarity measure*), salah satunya yaitu *Cosine Similarity*.

*Cosine Similarity* digunakan untuk mengukur tingkat kemiripan antara dua vektor berdasarkan sudut di antara keduanya, tanpa dipengaruhi oleh besar kecilnya nilai vektor. Pendekatan ini banyak digunakan pada data berdimensi tinggi dan bersifat jarang (*sparse*), seperti data fitur game, karena mampu menangkap pola kesamaan secara efektif [2]. Rumus 2.1 menunjukkan cara perhitungan *Cosine Similarity*.

$$\text{Similarity}(A, B) = \frac{A \cdot B}{\|A\| \times \|B\|} \quad (2.1)$$

di mana:

- $A$  dan  $B$  adalah vektor fitur dari dua game yang dibandingkan.
- $A \cdot B$  adalah hasil perkalian titik (*dot product*) dari kedua vektor.
- $\|A\|$  dan  $\|B\|$  adalah panjang (magnitude) masing-masing vektor.

Semakin tinggi nilai *similarity*, semakin mirip karakteristik dua game tersebut. Sebagai contoh, jika seorang pengguna sering memainkan game bergenre *Action RPG*, maka sistem akan merekomendasikan game lain yang memiliki genre dan fitur serupa.

Kelebihan utama metode *Content-Based Filtering* adalah kemampuannya dalam memberikan rekomendasi yang bersifat personal serta tidak bergantung pada preferensi pengguna lain, sehingga relatif tidak terpengaruh oleh permasalahan *cold-start* pada pengguna. Namun, metode ini juga memiliki keterbatasan, yaitu kecenderungan menghasilkan rekomendasi yang kurang beragam karena hanya berfokus pada item dengan karakteristik yang serupa dengan riwayat pengguna.

### 2.1.2 Association Rule

Metode Association Rule atau analisis asosiasi digunakan untuk menemukan hubungan keterkaitan antar item dalam suatu dataset transaksi. Dalam penelitian ini, proses pencarian pola asosiasi dilakukan menggunakan algoritma FP-Growth (*Frequent Pattern Growth*). Algoritma FP-Growth dirancang untuk menemukan *frequent itemset* secara efisien tanpa melalui tahap pembangkitan kandidat itemset seperti pada algoritma Apriori.

Sebagai perbandingan, algoritma Apriori dan FP-Growth memiliki tujuan yang sama, yaitu menemukan *frequent itemset* sebagai dasar pembentukan aturan asosiasi, namun berbeda dalam mekanisme pencariannya. Algoritma Apriori bekerja dengan membangkitkan kandidat itemset secara bertahap dan melakukan pemindaian basis data berulang kali untuk menghitung nilai *support*, sehingga berpotensi menghasilkan jumlah kandidat yang sangat besar. Sebaliknya, algoritma FP-Growth tidak melakukan pembangkitan kandidat itemset, melainkan menggunakan struktur *FP-Tree* untuk memadatkan data transaksi dan mengekstraksi pola asosiasi secara langsung. Perbedaan ini menjadikan FP-Growth lebih efisien dibandingkan Apriori, terutama pada dataset dengan jumlah transaksi dan kombinasi item yang relatif besar.

Proses kerja algoritma FP-Growth diawali dengan melakukan pemindaian terhadap dataset transaksi untuk menghitung frekuensi kemunculan setiap item. Item yang tidak memenuhi nilai minimum *support* akan dieliminasi. Selanjutnya, item-item yang tersisa disusun dalam urutan frekuensi tertentu dan direpresentasikan ke dalam sebuah struktur data berbentuk pohon yang disebut *FP-Tree*. Struktur ini berfungsi untuk memadatkan data transaksi sehingga pola keterkaitan antar item dapat direpresentasikan secara efisien.

Setelah *FP-Tree* terbentuk, algoritma FP-Growth melakukan proses penelusuran terhadap pohon tersebut untuk mengekstraksi kombinasi item yang sering muncul (*frequent itemset*). Proses ini dilakukan secara rekursif dengan membentuk *conditional pattern base* dan *conditional FP-Tree* untuk setiap item, hingga seluruh pola asosiasi yang memenuhi nilai minimum *support* berhasil ditemukan. Hasil dari proses ini berupa kumpulan *frequent itemset* yang kemudian digunakan sebagai dasar pembentukan aturan asosiasi.

Dalam konteks sistem rekomendasi video game, *frequent itemset* yang diperoleh mencerminkan pola kebiasaan bermain pengguna. Sebagai contoh, apabila banyak pengguna memainkan kombinasi game tertentu secara bersamaan,

maka kombinasi tersebut dapat dianggap sebagai pola yang kuat dan digunakan sebagai dasar rekomendasi.

Setelah *frequent itemset* diperoleh, aturan asosiasi dibentuk dan dievaluasi menggunakan ukuran-ukuran tertentu. Dua ukuran utama yang digunakan dalam *Association Rule* adalah:

1. *Support*, yaitu tingkat kemunculan kombinasi item dalam keseluruhan data, dihitung dengan Rumus 2.2:

$$\text{Support}(A \rightarrow B) = \frac{\text{Jumlah transaksi yang mengandung } A \text{ dan } B}{\text{Jumlah total transaksi}} \quad (2.2)$$

2. *Confidence*, yaitu tingkat kepercayaan bahwa item *B* muncul dalam transaksi yang mengandung item *A*, dihitung dengan Rumus 2.3:

$$\text{Confidence}(A \rightarrow B) = \frac{\text{Jumlah transaksi yang mengandung } A \text{ dan } B}{\text{Jumlah transaksi yang mengandung } A} \quad (2.3)$$

Dari data tersebut, dapat ditemukan pola bahwa pengguna yang memainkan game A cenderung juga memainkan game B dan C. Dengan demikian, sistem dapat merekomendasikan game B dan C kepada pengguna C karena pola asosiasi yang kuat antara game-game tersebut. Proses ini membantu sistem mengenali kecenderungan perilaku pengguna dan menghasilkan rekomendasi yang lebih akurat.

## 2.2 Evaluasi Sistem Rekomendasi

### 2.2.1 Precision, Recall, dan F1-Score

Evaluasi sistem rekomendasi bertujuan untuk mengukur sejauh mana rekomendasi yang dihasilkan relevan terhadap preferensi pengguna. Pada penelitian ini, evaluasi metode Content-Based Filtering dilakukan menggunakan metrik precision, recall, dan F1-score.

Precision merupakan rasio antara jumlah item relevan yang berhasil direkomendasikan dengan total item yang direkomendasikan oleh sistem. Precision menunjukkan tingkat ketepatan rekomendasi yang diberikan kepada pengguna, yang dirumuskan sebagai berikut, dengan  $R$  menyatakan himpunan item yang direkomendasikan oleh sistem dan  $G$  menyatakan himpunan item relevan (*ground truth*). Rumus 2.4 menunjukkan cara perhitungan *Precision*.

$$Precision = \frac{|R \cap G|}{|R|}, \quad (2.4)$$

Recall merupakan rasio antara jumlah item relevan yang berhasil direkomendasikan dengan total item relevan yang seharusnya direkomendasikan. Recall menggambarkan kemampuan sistem dalam menemukan item yang relevan. Rumus 2.5 menunjukkan cara perhitungan *Recall*.

$$Recall = \frac{|R \cap G|}{|G|} \quad (2.5)$$

F1-score merupakan nilai harmonisasi antara precision dan recall, yang digunakan untuk memberikan keseimbangan antara kedua metrik tersebut. Rumus 2.6 menunjukkan cara perhitungan *F1-score*.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.6)$$

dengan  $R$  menyatakan himpunan item yang direkomendasikan oleh sistem, dan  $G$  menyatakan himpunan item relevan atau *ground truth*.

### 2.2.2 Uji Lift Ratio

Association Rule Mining digunakan untuk menemukan pola keterkaitan antar item berdasarkan data transaksi pengguna. Untuk mengevaluasi kekuatan dan kualitas aturan asosiasi yang dihasilkan, penelitian ini menggunakan uji lift ratio sebagai metrik evaluasi.

Lift ratio mengukur tingkat ketergantungan antara item antecedent dan item consequent dibandingkan dengan kondisi apabila kedua item tersebut bersifat independen. Lift ratio dihitung dengan membandingkan nilai confidence suatu aturan asosiasi terhadap nilai support dari consequent, yang dirumuskan pada

Rumus 2.7.

$$Lift(A \rightarrow B) = \frac{Confidence(A \rightarrow B)}{Support(B)} \quad (2.7)$$

Nilai lift ratio yang lebih besar dari 1 menunjukkan adanya hubungan positif antara antecedent dan consequent, yang berarti kemunculan item antecedent meningkatkan probabilitas kemunculan item consequent [8]. Semakin besar nilai lift ratio, maka semakin kuat hubungan asosiasi yang terbentuk antar item.

Pada penelitian ini, evaluasi sistem rekomendasi berbasis Association Rule dilakukan dengan menghitung rata-rata nilai lift ratio dari rekomendasi yang dihasilkan untuk setiap pengguna. Pendekatan ini digunakan untuk menilai sejauh mana rekomendasi yang diberikan memiliki hubungan asosiasi yang kuat dan relevan berdasarkan pola interaksi pengguna.

### 2.2.3 End User Computing Satisfaction (EUCS)

End User Computing Satisfaction (EUCS) merupakan metode yang digunakan untuk mengukur tingkat kepuasan pengguna terhadap suatu sistem informasi atau aplikasi. Kepuasan pengguna diukur berdasarkan sejauh mana sistem mampu memenuhi harapan dan kebutuhan pengguna dalam menjalankan tugas-tugasnya. Menurut teori kepuasan pengguna, semakin tinggi kesesuaian antara harapan dan kenyataan yang dirasakan, maka semakin tinggi pula tingkat kepuasan pengguna terhadap sistem tersebut.

Menurut penelitian sebelumnya, "*End User Computing Satisfaction (EUCS) adalah metode untuk mengukur tingkat kepuasan dari pengguna suatu sistem aplikasi dengan membandingkan antara harapan dan kenyataan dari sebuah sistem informasi*". EUCS berperan penting dalam menilai efektivitas dan penerimaan sistem informasi dari sudut pandang pengguna akhir, bukan hanya dari sisi teknis pengembang[9].

Model EUCS dikembangkan oleh Doll dan Torkzadeh (1988) yang menetapkan lima variabel utama dalam pengukuran kepuasan pengguna, yaitu *content (isi)*, *accuracy (keakuratan)*, *format (bentuk)*, *ease of use (kemudahan)*, dan *timeliness (ketepatan waktu)*.

1. **Isi (Content)** mengukur sejauh mana informasi yang disediakan sistem relevan dan bermanfaat bagi pengguna.



2. **Keakuratan (Accuracy)** menilai ketepatan dan keandalan informasi yang dihasilkan.
3. **Bentuk (Format)** berkaitan dengan tampilan dan penyajian informasi dalam sistem.
4. **Kemudahan (Ease of Use)** mencerminkan seberapa mudah sistem digunakan oleh pengguna.
5. **Ketepatan Waktu (Timeliness)** mengukur kecepatan sistem dalam menyediakan informasi yang dibutuhkan tepat waktu.

EUCS menjadi instrumen yang sistematis dan terukur dalam mengevaluasi kualitas interaksi pengguna dengan sistem. Dengan demikian, penerapan EUCS dapat membantu pengembang maupun organisasi untuk mengetahui tingkat penerimaan dan kepuasan pengguna terhadap sistem yang dikembangkan.

