

## **BAB 2**

### **LANDASAN TEORI**

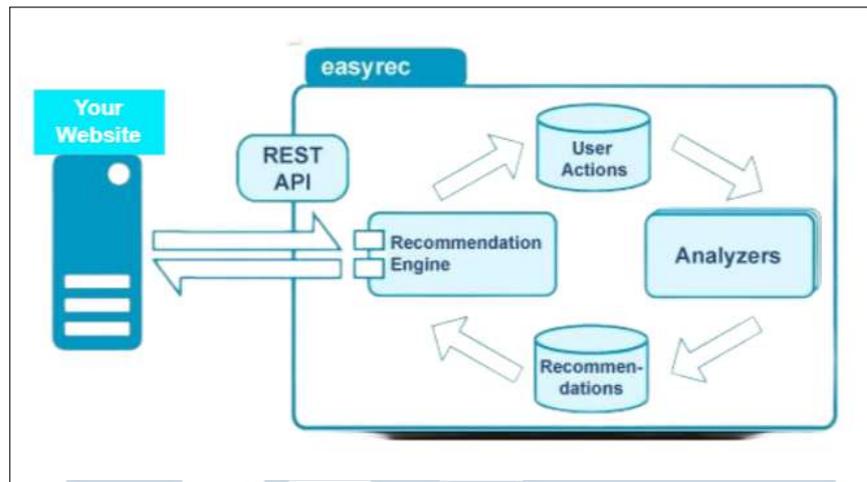
#### **2.1 Sistem Rekomendasi**

Sistem rekomendasi merupakan sebuah sistem yang dapat membantu pengguna untuk mengidentifikasi produk yang sesuai dengan kebutuhan, dan keinginan user berdasarkan preferensi dan perilaku pengguna pada masa lampau. Manfaat utama dari sistem rekomendasi adalah untuk menghindari fenomena *information overload* dengan menyaring fragmen informasi penting dari banyaknya informasi yang tersedia dengan menghasilkan kandidat secara dinamis sesuai dengan preferensi, minat, atau perilaku yang diamati pengguna terhadap item tersebut. Sistem rekomendasi memiliki kemampuan untuk memprediksi apakah seorang pengguna tertentu akan lebih memilih suatu item atau tidak berdasarkan profil pengguna yang tersedia

Sistem rekomendasi telah menjadi komponen yang krusial pada website saat ini, terutama website yang memiliki sumber daya informasi yang sangat luas karena sistem rekomendasi dapat memudahkan pengguna dalam menemukan produk yang relevan dari berbagai macam jenis produk [10]. Desain dari sistem rekomendasi bergantung pada domain dan karakteristik khusus dari data yang tersedia. Sistem ini dikembangkan dengan tujuan menutup kesenjangan antara pengumpulan informasi dan analisis dengan menyaring semua informasi yang tersedia untuk menyajikan apa yang dinilai paling berharga bagi pengguna [11].

#### **2.2 Content-Based Filtering**

*Content-Based Filtering* merupakan salah satu pendekatan yang digunakan dalam membangun sistem rekomendasi selain *collaborative Filtering*, metode ini mempertimbangkan preferensi pengguna berdasarkan karakteristik atau konten dari suatu item yang telah diinteraksi sebelumnya oleh pengguna tersebut [12]. Metode ini tidak membutuhkan profil pengguna lain dalam memberikan rekomendasinya. Deskripsi dari masing-masing item dan profil pengguna memiliki andil besar dalam keberhasilan metode sistem rekomendasi ini. Gambar 2.1 menggambarkan alur metode *Content-Based Filtering* dalam memberikan rekomendasi



Gambar 2.1. *Content-Based Filtering*

Sumber: [11]

Pada gambar 2.1, sistem rekomendasi berbasis *content-based filtering* akan menganalisis aksi *user* yang tersedia untuk memberikan rekomendasi kepada pengguna *website* melalui API (*Application Programming Interface*). Metode ini akan merekomendasikan item berdasarkan perbandingan antara profil item dengan profil pengguna. Profil item biasanya direpresentasikan sebagai kumpulan deskriptor, istilah atau kata-kata yang muncul dalam sebuah dokumen, sedangkan profil pengguna direpresentasikan dengan istilah yang sama. Namun, dibangun dengan menganalisis dan menggabungkan masing-masing konten item yang telah diinteraksi oleh pengguna. Dalam proses penggabungan tersebut, dibutuhkan suatu pembobotan fitur agar sistem rekomendasi dapat turut mempertimbangkan berbagai fitur item yang diinteraksi *user* dengan kekuatan yang berbeda. Metode TF-IDF merupakan salah satu metode pembobotan fitur yang dapat merealisasikan hal tersebut [13].

### 2.3 TF-IDF

Konsep *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) sering digunakan dalam konteks *information retrieval* dan juga pada metode rekomendasi berbasis *content-based filtering*. Konsep ini digunakan untuk menentukan tingkat kepentingan dari suatu dokumen/artikel/berita/film, dll. TF adalah frekuensi sebuah kata dalam sebuah dokumen, sedangkan IDF adalah kebalikan dari frekuensi dokumen di seluruh korpus dokumen yang dihitung dengan menggunakan inversi logaritmik, Persamaan 2.1 merupakan persamaan yang digunakan untuk

perhitungan TF (*Term Frequency*):

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (2.1)$$

Dimana  $n_{i,j}$  mewakili jumlah kemunculan kata  $t_i$  dalam dokumen  $d_j$ ,  $\sum_k n_{k,j}$  mewakili jumlah total kemunculan kata dalam dokumen  $d_j$ , sedangkan persamaan 2.2 dan 2.3 merupakan persamaan yang digunakan untuk perhitungan DF (*Document Frequency*) dan IDF (*Inverse Document Frequency*)

$$DF_{i,j} = \frac{|d_j \in D : t_j \in d_j|}{|D|} \quad (2.2)$$

$$IDF_{i,j} = \log \frac{|D|}{|d_j \in D : t_j \in d_j|} \quad (2.3)$$

Dimana  $|D|$  mewakili jumlah total dokumen.  $d_j \in D : t_j \in d_j$  mewakili jumlah dokumen di mana kata kunci  $t_j$  muncul. IDF adalah kebalikan dari frekuensi dokumen di seluruh korpus dokumen yang dihitung dengan menggunakan inversi logaritmik untuk mengukur tingkat kepentingan kata dalam suatu kumpulan dokumen. Setelah berhasil mendapatkan nilai TF dan IDF maka perhitungan nilai TF-IDF dapat dilakukan dengan persamaan 2.4:

$$TF-IDF = TF \times IDF \quad (2.4)$$

Nilai TF-IDF meningkat ketika sebuah kata kunci spesifik memiliki frekuensi yang tinggi dalam sebuah dokumen dan frekuensi dokumen yang mengandung kata kunci tersebut di antara seluruh dokumen rendah. Prinsip ini dapat digunakan untuk menemukan kata kunci yang sering muncul dalam suatu teks dan menentukan tingkat kepentingan dari suatu kata kunci tersebut [13].

## 2.4 Cosine Similarity

Setelah menghitung skor TF-IDF, tahapan selanjutnya dalam *content-based filtering* adalah menentukan *item* mana yang lebih dekat satu sama lain dengan menggunakan model ruang vektor yang melakukan perhitungan kedekatan berdasarkan sudut antar vektor tersebut. *Cosine Similarity* merupakan suatu metrik yang digunakan untuk mengukur sejauh mana dua vektor berada dalam posisi searah atau sejajar satu sama lain. Pada model *content-based filtering*, setiap item

akan tersimpan sebagai vektor atribut dalam ruang berdimensi  $n$ , sudut antar vektor tersebut akan dihitung untuk ditentukan kemiripannya dengan *cosine similarity*. Dalam konteks sistem rekomendasi, *cosine similarity* memberikan ukuran sejauh mana preferensi pengguna cocok dengan atribut atau karakteristik item. Semakin tinggi nilai cosine similarity antara dua item maka kedua item tersebut semakin mirip. Oleh karena itu, item yang lebih mirip memiliki kesempatan lebih besar untuk direkomendasikan kepada pengguna. Berikut adalah rumus dari cosine-similarity:

$$\text{Cosine Similarity}(A, B) = \frac{A \cdot B}{\|A\| \cdot \|B\|} \quad (2.5)$$

$A$  merupakan vektor  $A$  dan  $B$  merupakan vektor  $B$ ,  $A \cdot B$  merupakan dot product antara vektor  $A$  dengan  $B$ , sedangkan  $\|A\|$  dan  $\|B\|$  merupakan norma euclidean dari vektor  $A$  dan  $B$ . Cosine similarity menghasilkan nilai antara minus satu dan satu. Nilai satu menunjukkan bahwa dua vektor sepenuhnya sejajar (sama persis), nilai nol menunjukkan bahwa vektor-vektor tersebut tegak lurus, dan nilai minus satu menunjukkan bahwa vektor-vektor tersebut sejajar tetapi berlawanan arah. Dalam penelitian ini, cosine similarity digunakan dalam *content-based filtering* di lapisan rekomendasi pertama untuk mengukur tingkat kesamaan antara profil item acara yang belum pernah dilihat *user* dengan profil *user* berdasarkan item acara yang telah dilihat, disimpan, dihadiri ataupun dievaluasi oleh *user* tersebut sebelumnya.

## 2.5 Haversine

Rumus haversine digunakan dalam menghitung jarak antara dua koordinat di bumi dalam garis lurus, tanpa memperhitungkan perbedaan ketinggian seperti bukit atau lembah. Metode ini menggunakan koordinat lintang dan bujur (Latitude dan Longitude) dari dua lokasi di bumi sebagai input. Hasil akhirnya adalah nilai jarak antara kedua lokasi tersebut [14]. Rumus ini akan digunakan pada tahap kedua sistem rekomendasi untuk mendapatkan nilai skor kedekatan lokasi antara kandidat acara dengan pengguna. Nilai ini kemudian akan dijumlahkan dengan skor kriteria lainnya untuk mendapatkan skor akhir pada acara yang akan direkomendasikan. Berikut adalah rumus untuk perhitungan jarak menggunakan haversine:

$$\text{distance} = R \times c \quad (2.6)$$

Dimana:

$$c = 2 \times \text{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$a = \sin^2\left(\frac{\Delta\text{lat}}{2}\right) + \cos(\text{lat1}) \times \cos(\text{lat2}) \times \sin^2\left(\frac{\Delta\text{lon}}{2}\right)$$

$$\Delta\text{lat} = \text{lat2} - \text{lat1}$$

$$\Delta\text{lon} = \text{lon2} - \text{lon1}$$

$$R = 6371.0$$

Distance mengacu pada jarak antara dua titik dalam satuan yang sama dengan R, yang biasanya diukur dalam kilometer. R adalah radius bumi, diukur dalam satuan yang sama dengan jarak (biasanya kilometer). Nilainya adalah sekitar 6371.0 kilometer, sedangkan  $c$  adalah sudut sentral antara dua titik, dihitung menggunakan fungsi arktan dalam radian. Variabel  $a$  dalam rumus Haversine merupakan komponen yang menggambarkan seberapa jauh kedua titik tersebut berjarak dari garis ekuator. Nilai  $a$  dihitung berdasarkan perbedaan lintang dan bujur antara kedua titik, serta digunakan dalam perhitungan sudut sentral  $c$ . Dalam perhitungan Haversine,  $\Delta\text{lon}$  mengacu pada perbedaan bujur antara dua titik, sedangkan  $\Delta\text{lat}$  adalah perbedaan lintang di antara keduanya. Variabel ini digunakan untuk menghitung *central angle*  $c$  antara dua titik.

## 2.6 Weighted Sum

Weighted sum merupakan salah satu metrik pengambilan keputusan terhadap sejumlah kandidat alternatif dengan kriteria yang beragam [15]. metode ini dapat diinterpretasikan sebagai cara untuk memberikan nilai bobot untuk masing-masing elemen kriteria dalam kontribusinya terhadap nilai total. Bobot atau yang disebut sebagai *weight* ini mengontrol sejauh mana suatu elemen mempengaruhi hasil akhir. Berikut adalah rumus dari weighted sum:

$$\text{Weighted Sum} = w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_n \cdot x_n \quad (2.7)$$

$w_1, w_2, \dots, w_n$  mewakili nilai bobot yang ditetapkan pada masing-masing elemen kriteria, sedangkan  $x_1, x_2, \dots, x_n$  mewakili nilai untuk masing-masing kriteria/element. Kriteria-kriteria tersebut akan dikalikan dengan nilai bobot tertentu dan dijumlahkan sehingga didapatkan skor akhir. Pada konteks sistem rekomendasi yang diajukan pada penelitian ini, perhitungan weighted sum akan dilakukan

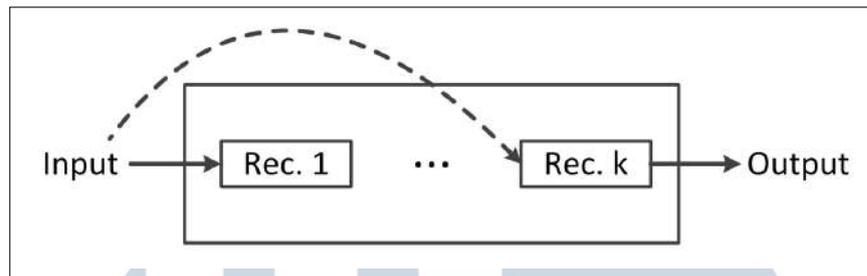
setelah *content-based filtering* berhasil memberikan kandidat rekomendasi item acara dengan tingkat kesamaan tertinggi. Metode ini akan mempertimbangkan kedekatan rentang waktu registrasi suatu acara dengan waktu saat ini, kedekatan lokasi geografis acara dengan pengguna, tingkat popularitas dari suatu acara dan nilai similaritas kandidat acara yang telah didapatkan dari tahap sebelumnya.

## 2.7 Sistem rekomendasi Hybrid

Sistem rekomendasi hybrid mengkombinasikan beberapa teknik rekomendasi untuk meningkatkan performa kinerja keseluruhan serta mengatasi keterbatasan masing-masing metode [16]. Kombinasi pendekatan ini tentunya menghadirkan berbagai keunggulan dari masing-masing pendekatan sistem rekomendasi yang digunakan. Metode hybrid ini terbagi lagi menjadi tiga jenis yaitu *monolithic hybridization* dimana metode ini menggunakan komponen rekomendasi tunggal yang menggabungkan karakteristik dan strategi rekomendasi berbeda dari berbagai pendekatan rekomendasi yang digunakan, lalu terdapat metode *parallel hybridization* dimana skor rekomendasi akhir diperoleh dengan menggabungkan skor rekomendasi dari berbagai pendekatan rekomendasi secara bersamaan dan yang terakhir adalah metode *pipelining hybridization* yang merupakan pendekatan rekomendasi yang disusun secara berurutan, membentuk pipa. [4].

## 2.8 Pipelining Hybridization

*Pipelining Hybridization* merupakan salah satu pendekatan *hybrid* yang digunakan dalam membangun sistem rekomendasi, setiap blok rekomendasi disusun secara berurutan menyerupai pipa seperti pada Gambar 2.2, dimana hasil dari sistem rekomendasi sebelumnya dapat menjadi bagian dari masukan untuk sistem rekomendasi setelahnya [4].



Gambar 2.2. Metode Pipelining Hybridization

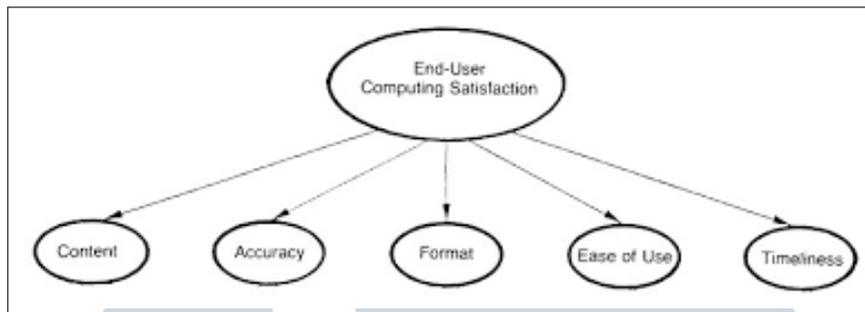
Sumber: Hybrid Recommendation PDF by Weike Pan

Pendekatan ini akan digunakan pada penelitian ini, dimana lapisan pertama akan digunakan metode *content-based filtering* yang menghasilkan total 20 kandidat rekomendasi dengan nilai similarity tertinggi yang dihitung berdasarkan profil item dan *user* yang tersedia, kandidat yang dihasilkan pada rekomendasi lapisan pertama akan diteruskan ke rekomendasi lapisan kedua untuk dilakukan *scoring / ranking* dengan metode *weighted sum* dengan hasil akhir 10 rekomendasi acara dengan skor tertinggi.

## 2.9 End User Computing Satisfaction (EUCS)

*End User Computing Satisfaction* (EUCS) merupakan suatu teknik untuk menilai sejauh mana pengguna sistem informasi merasa puas dengan pengalaman penggunaan sistem tersebut, dengan membandingkan ekspektasi mereka dengan realita yang mereka alami. Teknik ini melibatkan evaluasi keseluruhan dari pengguna sistem berdasarkan pengalaman mereka dalam menggunakan sistem tersebut. Gambar 2.3 merupakan Model evaluasi EUCS yang dirumuskan oleh Doll et al, model ini memperhitungkan kepuasan pengguna akhir terhadap berbagai aspek teknologi, yaitu konten, keakuratan, format, waktu, dan kemudahan penggunaan sistem yang dievaluasi [17].

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 2.3. End User Computing Satisfaction

Sumber: [17]

## 2.10 Skala Likert

Untuk mendapatkan hasil yang akurat berdasarkan persepsi pengguna pada survei End User Computing Satisfaction (EUCS), maka perhitungan Skala Likert akan digunakan. Skala Likert pertama kali dikembangkan oleh Rensis Likert pada tahun 1932 untuk mengukur sikap masyarakat. Skala Likert menggunakan skala ordinal mulai dari sangat positif sampai dengan sangat negatif, skala tersebut antara lain: Sangat Setuju(SS), Setuju(S), Netral(N), Tidak Setuju(TS) dan Sangat Tidak Setuju (STS). Urutan setuju atau tidak setuju dapat dibalik, mulai dari Sangat Tidak Setuju (STS) sampai dengan Sangat Setuju [18]. Ketika nilai dari kuesioner telah terkumpul, untuk pertanyaan yang bersifat positif, maka persentase rata-rata jawaban responden dapat dihitung dengan menggunakan rumus 2.8

$$\text{Hasil Persentase} = \frac{(SS \times 5) + (S \times 4) + (N \times 3) + (TS \times 2) + (STS \times 1)}{5 \times \text{Jumlah Responden}} \quad (2.8)$$

Untuk pertanyaan yang negatif, persentase rata-rata jawaban responden dapat dihitung dengan menggunakan rumus 2.9

$$\text{Hasil Persentase} = \frac{(SS \times 1) + (S \times 2) + (N \times 3) + (TS \times 4) + (STS \times 5)}{5 \times \text{Jumlah Responden}} \quad (2.9)$$