

BAB II

TINJAUAN PUSTAKA

2.1 Sistem Rekomendasi

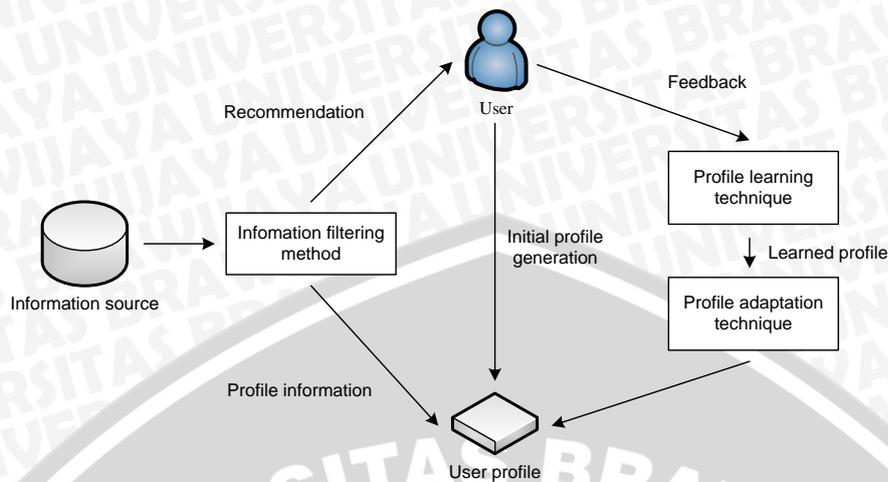
Sistem rekomendasi merupakan suatu sistem yang bertujuan memberikan saran kepada *user* tentang *item* yang akan mereka pilih. Rekomendasi dibuat oleh sistem yang dapat mengarahkan *user* dalam mendapatkan informasi berupa deskripsi *item* atau hal yang lain (Shinde, 2011).

Sistem rekomedasi membantu mengatasi kelebihan informasi dengan menyediakan personalisasi saran berdasarkan ketertarikan *user*. Perkiraan informasi ini bersifat personil yang didasarkan atas profil *user*. Profil *user* umumnya didasarkan menarik tidaknya suatu informasi yang dilihat oleh *user*.

Sistem rekomendasi sudah banyak digunakan pada berbagai *website* khususnya *website* toko *online*. Sistem rekomendasi dapat meningkatkan proses perdagangan pada toko *online* melalui tiga cara, yaitu : (Schafer, 2007)

1. Mengubah *user* dari hanya *browsing* menjadi pembeli. Sistem rekomendasi membantu *user* dalam menemukan *item* yang ingin diinginkan.
2. Jika rekomendasi yang diberikan dianggap cocok oleh *user* maka *volume* penjualan juga ikut meningkat, dimana sistem rekomendasi memberikan rekomendasi pembelian berdasarkan *item* yang telah dibeli oleh *user* sebelumnya.
3. Sistem rekomendasi dapat memberikan nilai tambah berupa hubungan antara pemilik *website* dengan *user*. *Website* memberikan rekomendasi berdasarkan profil *user* dan *user* memberikan *input* yang diperlukan. Hubungan yang telah terbentuk akan membuat *user* kembali berkunjung ke *website* yang telah memberikan rekomendasi kepadanya.

Menurut (Montainer, 2003), profil dari *user* memegang kunci penting pemberian rekomendasi. Profil *user* merupakan gambaran kebiasaan *user* ketika berinteraksi dengan sistem. Berikut ini merupakan gambar dari arsitektur sistem rekomendasi :



Gambar 2.1 Arsitektur Sistem Rekomendasi

Pada gambar diatas dapat dilihat bahwa *user* melakukan proses *initial profile generation* untuk membentuk *profile* dari *user*. Ketika ada sumber informasi maka informasi akan disaring dengan *information filtering method* yang digunakan oleh sistem rekomendasi yang berasal dari profil *user* dan setelah itu rekomendasi disampaikan ke *user*. Hasil rekomendasi yang diberikan ke *user* ditanggapi oleh *user* berupa *feedback* yang kemudian dipelajari dan diadaptasikan ke dalam profil *user*. Proses *profile generation and maintenance* berkaitan dengan pembangkitan dan pemeliharaan profil *user* sebagai komponen utama dalam memberikan rekomendasi. Pertama yang harus ditentukan adalah profil *user*, selanjutnya dilakukan *initial profile generation*. Ketika rekomendasi diberikan kepada *user*, *user* bisa memberikan *feedback* dan *feedback* tersebut dipelajari oleh sistem untuk digunakan rekomendasi selanjutnya (Montainer, 2003).

2.2 Content-based

Content-based bekerja dengan cara mencari *item* lain yang mirip dengan *item* yang disukai oleh *user* berdasarkan informasi *content* atau tekstual dari setiap *item* (Mobasher, 2000). *Content-based* banyak digunakan untuk penyaringan informasi pada sistem berbasis teks, seperti fasilitas *related pages* dalam *search engine*. Pada sistem tersebut, sebuah algoritma untuk *query* pencarian diaplikasikan terhadap

sebuah matriks berisi pasangan dokumen *web* dan *keyword*. Oleh karena itu, metode *content-based* disebut juga dengan *search-based*.

Untuk *user* yang hanya memiliki sedikit *record* dalam catatan transaksi atau data rating, metode *content-based* berjalan dengan baik. Namun, ketika *user* telah memiliki banyak *record item* baik yang telah *di rating*, maka pencarian terhadap semua *item* yang mirip menjadi tidak praktis lagi. Secara umum, kualitas rekomendasi yang dihasilkan *content-based* relatif rendah (Mobasher, 2000). Pada kasus tertentu, rekomendasi bersifat terlalu umum (misalnya hanya judul film terlaris dalam kategori drama) atau terlalu khusus (hanya merekomendasikan buku dengan penulis yang sama). Kelemahan lain, pada domain seperti musik, film, atau gambar, *content-based* sulit diimplementasikan karena penyaringan informasi tekstual sulit dilakukan.

2.3 Collaborative Filtering

Collaborative filtering adalah suatu metode dalam membuat sebuah prediksi untuk memperkirakan ketertarikan atau selera seorang *user* terhadap suatu *item* dengan cara mengumpulkan informasi dari *user-user* yang lain yang direpresentasikan dalam bentuk nilai rating (Sarwar, 2001).

Secara umum, ada 2 proses yang dilakukan dalam *collaborative filtering*, yaitu:

1. Mencari *user* lain yang mempunyai kemiripan pola rating dengan *user* target (*user* yang akan diberikan prediksi).
2. Menggunakan nilai rating dari *user* lain yang didapat dari langkah 1 di atas untuk menghitung prediksi bagi *user* aktif.

Secara umum, metode *collaborative filtering* dikelompokkan menjadi 2 kategori (Sarwar, 2001), yaitu:

1. *Memory-based*

Algoritma *memory-based* mempergunakan seluruh data rating dalam membuat prediksi. Metode ini memakai teknik statistik dalam mencari sekumpulan *user*, disebut dengan *nearest-neighbor*, yang mempunyai kemiripan dengan *user* target. Berdasarkan kemiripan antar *user* tersebut, dibuat prediksi dan

rekomendasi *Top-N* untuk *user* target. Metode ini sering disebut juga dengan *nearest-neighbor* atau *user-based collaborative filtering*.

2. Model-based

Berbeda dengan *memory-based* yang mencari hubungan antar *user*, *model-based* mencari hubungan antar *item* berdasarkan tabel rating untuk membuat rekomendasi. Sehingga metode ini disebut juga dengan *item-based*. Sebelum menghasilkan rekomendasi, algoritma *model-based* membuat sebuah model korelasi antar *item* terlebih dahulu untuk mengetahui hubungan antar *item* berdasarkan nilai rating yang didapat. Proses pembuatan model dapat dilakukan dengan menggunakan berbagai teknik, seperti *association rule*, *classification*, atau *clustering*.

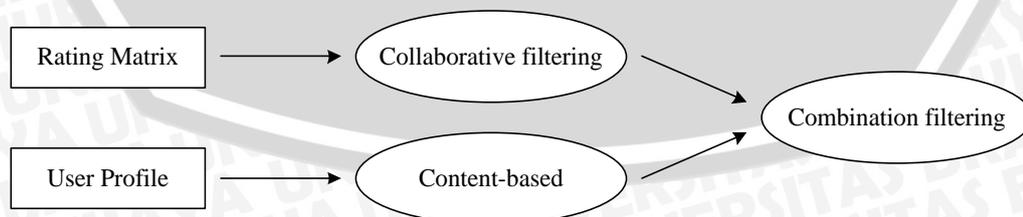
2.4 Metode Hybrid

Metode *hybrid* pada sistem rekomendasi digunakan untuk menggabungkan beberapa metode untuk menghasilkan rekomendasi (Melville, 2010). Menurut (Adiningsih, 2005) beberapa sistem rekomendasi menggunakan metode *hybrid* untuk menggabungkan metode *collaborative filtering* dan *content-based* untuk mengatasi kekurangan yang terdapat pada kedua metode tersebut.

Menurut (Li, 2004) terdapat beberapa cara penggabungan yang dapat dilakukan dalam metode *hybrid* yaitu :

1. Linier Combination

Pendekatan ini menggabungkan hasil rating dari metode *content-based* dan *collaborative filtering*. Penggabungan ini dilakukan dengan cara pemberian rating. Berikut adalah gambar dari *linier combination* :



Gambar 2.2 Linier Combination

2. Sequential Combination

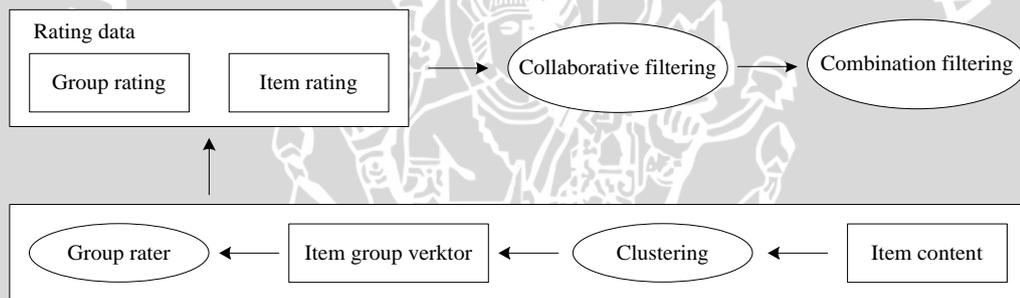
Pendekatan ini melakukan perhitungan pada salah satu metode (misalnya *content-based*) kemudian hasilnya dilakukan perhitungan metode lainnya (*collaborative filtering*). Berikut adalah gambar dari *sequential combination* :



Gambar 2.3 Sequential Combination

3. Item-based Clustering Hybrid Method (ICHM)

Pendekatan ini mengintegrasikan informasi *item* dan rating *user* untuk menghitung kemiripan *item-item*. Karena metode ini yang dipakai pada tulisan ini, maka tahap-tahapnya akan diuraikan secara detail pada subbab 2.5. Berikut adalah gambar dari *item-based clustering hybrid method* :



Gambar 2.4 Item-based Clustering Hybrid Method

2.5 Item-based Clustering Hybrid Method

Item-based clustering hybrid method (ICHM) adalah salah satu metode berbasis *hybrid* yang menggabungkan metode *collaborative filtering* dengan *content-based* (Li, 2004).

Item-based clustering hybrid method akan menghasilkan 2 bentuk *output*, yakni prediksi dan rekomendasi.

1. Prediksi adalah nilai yang menyatakan besarnya rating *item k* yang didapatkan dari *user u*.
2. Rekomendasi adalah sebuah daftar yang berisi *N item* yang mempunyai kemungkinan terbesar disukai oleh *user u*, dimana *N item* tersebut belum pernah di rating oleh *user u*. Bentuk ini disebut juga rekomendasi *Top-N*.

Sedangkan tahap-tahap dalam *item-based clustering hybrid method* adalah sebagai berikut :

2.5.1 Pembentukan Matriks

Pada proses pembentukan matriks akan di bentuk dua buah matriks, yaitu matriks *item-rating* dan matriks *group-rating*.

1. Matriks *item-rating*

Matriks *item-rating* dibangun dari rating yang ada pada *item* dan *user* yang memberikan rating pada *item* tersebut.

2. Matriks *group-rating*

Untuk membangun matriks *group-rating* terlebih dahulu dilakukan proses *clustering*. Algoritma yang digunakan dalam proses *clustering* adalah *K-Means clustering*. *K-means clustering* merupakan algoritma dasar yang mudah dan dapat dengan cepat melakukan pemusatan data (*convergen*) pada proses *clustering* (Adiningsih, 2007). *Input* yang digunakan dalam proses *clustering* adalah sejumlah *k cluster* dan konten dari *item*.

Tahap-tahap dalam proses *clustering* menggunakan algoritma *K-Means clustering* adalah sebagai berikut :

- a. Pemilihan secara acak *k*, *k* disini merupakan banyaknya *cluster* yang ingin dibentuk.
- b. Tentukan *cluster* awal (dipilih secara acak dari *record* pada *datasets*) sebagai titik tengah *cluster* atau biasa disebut dengan *centroid*.
- c. Masukkan tiap *item* ke dalam *cluster* yang paling mirip atau jarak dengan pusat *cluster* yang paling dekat. Perhitungan jarak tersebut menggunakan *Euclidian Distance* yang dirumuskan dengan :

$$D(x_2, x_1) = \sqrt{\sum_{u=1}^m (x_{2m} - x_{1m})^2} \quad (2.1)$$

dimana,

D = dimensi data

*x*₁ = pusat *cluster*

*x*₂ = data

m = jumlah total data

- d. Hitung kembali nilai tengah dari setiap *cluster*.
- e. Ulangi atau lakukan proses iterasi dari langkah c, sampai didapatkan nilai $\frac{BCV}{WCV}$ lebih kecil dari nilai $\frac{BCV}{WCV}$ pada proses iterasi sebelumnya (Larose, 2005).

Nilai $\frac{BCV}{WCV}$ dihitung dengan menggunakan rumus berikut :

$$\frac{BCV}{WCV} = \frac{d(m_1, m_2)}{SSE} \quad (2.2)$$

dimana,

$\frac{BCV}{WCV}$ = nilai rasio antar *cluster*

$d(m_1, m_2)$ = jarak titik pusat *cluster*

SSE = *Sum Squared Error*, dihitung dengan rumus,

$$SSE = \sum_{i=1}^k \sum_{p \in C_i} d(p, m_i)^2 \quad (2.3)$$

dimana,

SSE = *Sum Squared Error*

$d(p, m_i)$ = jarak terdekat pada *cluster*

Setelah didapatkan *cluster* atau kelompok, dihitung keterkaitan atau peluang antar *item* terhadap *cluster* yang digunakan untuk membangun matriks *group-rating* dengan menggunakan rumus :

$$Pro(j, k) = 1 - \frac{CS(j, k)}{\max CS(i, k)} \quad (2.4)$$

dimana,

$Pro(j, k)$ = peluang *item j* menjadi bagian dari *cluster k*

$CS(j, k)$ = *counter-similarity* antara *item j* dan *cluster k*,
dihitung dengan *Euclidean Distance*

$\max CS(i, k)$ = nilai *counter-similarity* terbesar antara sebuah *item i*
dan *cluster k*

Output yang dihasilkan dari proses diatas adalah sejumlah *k cluster* dan nilai kemungkinan tiap *item* terhadap *cluster*.

2.5.2 Perhitungan Nilai *Similarity*

Hitung *similarity* dari matriks *item-rating* (*similarity item-rating*) dan matriks *group-rating* (*similarity group-rating*), kemudian hasil dari perhitungan kedua matriks tersebut digabungkan dengan menggunakan metode *linier combination of similarity* untuk digunakan pada perhitungan prediksi rating.

1. Untuk perhitungan *similarity item-rating* digunakan metode *pearson correlation-based*. Nilai dari metode *pearson correlation-based* adalah dari -1 sampai dengan 1 (Jannach, 2010). Nilai 1 diartikan bahwa dua *item* saling berkorelasi, sedangkan nilai -1 diartikan bahwa dua *item* tidak saling berkorelasi.

Rumus dari metode *pearson correlation-based* adalah sebagai berikut :

$$sim(k,l)_{item} = \frac{\sum_{u=1}^m (R_{u,k} - \bar{R}_k)(R_{u,l} - \bar{R}_l)}{\sqrt{\sum_{u=1}^m (R_{u,k} - \bar{R}_k)^2} \sqrt{\sum_{u=1}^m (R_{u,l} - \bar{R}_l)^2}} \quad (2.5)$$

dimana,

$sim(k,l)$ = nilai *similarity* antara *item k* dan *l*

m = jumlah total *user*

\bar{R}_k, \bar{R}_l = rata-rata *rating item k* dan *l*

$R_{u,k}, R_{u,l}$ = *rating* dari *user u* pada *item k* dan *l*

2. Sedangkan untuk perhitungan *similarity group-rating* menggunakan metode *adjusted-cosine*. Nilai dari metode *adjusted-cosine* adalah dari 0 sampai dengan 1 (Jannach, 2010). Nilai 1 diartikan bahwa dua *item* saling berkorelasi, sedangkan nilai 0 diartikan bahwa dua *item* tidak saling berkorelasi. Rumus dari metode *adjusted-cosine* adalah sebagai berikut :

$$sim(k,l)_{group} = \frac{\sum_{u=1}^m (R_{u,k} - \bar{R}_u)(R_{u,l} - \bar{R}_u)}{\sqrt{\sum_{u=1}^m (R_{u,k} - \bar{R}_u)^2} \sqrt{\sum_{u=1}^m (R_{u,l} - \bar{R}_u)^2}} \quad (2.6)$$

dimana,

$sim(k,l)$ = nilai *similarity* antara *item k* dan *l*

m = jumlah total *user*

\bar{R}_u = rata-rata *rating* dari *user u*

$R_{u,k}, R_{u,l}$ = *rating* dari *user u* pada *item k* dan *l*

3. Setelah didapatkan nilai kedua *similarity* diatas (*similarity item-rating* dan *similarity group-rating*), kemudian dilakukan perhitungan terhadap keduanya dengan menggunakan metode *linier combination of similiraty*. Nilai dari metode *linier combination of similiraty* adalah dari 0 sampai dengan 1 (Yarlett, 2008). Koefisien *c* menandakan persentase nilai *similarity* dari matriks *group-rating* yang digunakan. Metode *linier combination of similiraty* dirumuskan dengan :

$$sim(k,l) = sim(k,l)_{item} \times (1 - c) + sim(k,l)_{group} \times c \quad (2.7)$$

dimana,

$sim(k,l)$ = nilai *similarity* antara *item k* dan *l*

c = *combination coefficient*, nilainya antara 0 - 1

$sim(k,l)_{item}$ = nilai *similarity* antara *item k* dan *l*, didapat dari perhitungan *similarity item-rating*

$sim(k,l)_{group}$ = nilai *similarity* antara *item k* dan *l*, didapat dari perhitungan *similarity group-rating*

2.5.3 Perhitungan Prediksi Rating

Dalam menghitung prediksi rating suatu *item* dibagi menjadi dua perhitungan, yaitu untuk *item* yang sudah pernah di-rating oleh user (*non cold-start problem*) dan pada *item* yang belum pernah di rating oleh user (*cold-start problem*).

1. Prediksi pada *item* yang pernah di rating.

Untuk menghitung prediksi rating pada *item k* yang telah di rating, digunakan metode *weighted average of deviation* yang didapat dari rata-rata *item* yang telah di rating, yang dirumuskan dengan :

$$P_{u,k} = \bar{R}_k + \frac{\sum_{i=1}^n (R_{u,i} - \bar{R}_i) \times sim(k,i)}{\sum_{u=1}^n |sim(k,i)|} \quad (2.8)$$

dimana,

$P_{u,k}$ = prediksi rating user *u* pada *item k*

n = jumlah *neighbour item k*

$R_{u,i}$ = rating dari user *u* pada *item i*

- \bar{R}_k = rata-rata rating pada *item k*
 $sim(k,i)$ = nilai *similarity* antara *item k* dan *neighbour item i*
 \bar{R}_i = rata-rata rating pada *item i*

2. Prediksi pada *item* yang belum pernah di rating.

Karena *item* belum pernah *di rating* sehingga nilai rata-rata pada *item k* akan bernilai nol, maka metode *weighted sum* digunakan untuk prediksi rating pada kasus *item* baru. Metode *weighted sum* dirumuskan dengan :

$$P_{u,k} = \frac{\sum_{i=1}^n R_{u,i} \times sim(k,i)}{\sum_{i=1}^n |sim(k,i)|} \quad (2.9)$$

dimana,

- $P_{u,k}$ = prediksi rating *user u* pada *item k*
 n = jumlah *neighbour item k*
 $R_{u,i}$ = rating dari *user u* pada *item i*
 $sim(k,i)$ = nilai *similarity* antara *item k* dan *neighbour item i*

2.6 Pengukuran Tingkat Akurasi

Dalam menghitung tingkat akurasi dari rekomendasi yang dihasilkan, digunakan suatu metode yang bernama *Mean Absolute Error (MAE)*. *MAE* diperoleh dengan menghitung *error absolut* dari N pasang rating yang sebenarnya (p) dengan rating hasil prediksi (q). Dimana semakin kecil nilai *MAE* yang didapat, maka prediksi yang dihasilkan semakin akurat (Sarwar, 2001). *MAE* dirumuskan dengan :

$$MAE = \frac{\sum_{u=1}^N |P_{u,i} - R_{u,i}|}{N} \quad (2.10)$$

dimana,

- $P_{u,i}$ = prediksi rating dari *user u* pada *item i*
 $R_{u,i}$ = rating yang sebenarnya dari *user u* pada *item i*
 N = jumlah data

Berdasarkan beberapa penelitian tentang sistem rekomendasi yang telah dilakukan menggunakan metode-metode lain dengan menggunakan dataset yang sama, pada metode *collaborative filtering* yaitu dengan *pearson correlation* mendapatkan nilai *MAE* sebesar 0,845. *Spearman correlation* mendapatkan nilai *MAE* sebesar 0,85 (Sania, 2013). Pada *item-based* mendapatkan nilai *MAE* sebesar 0,749 (Sarwar, 2001). Dan *semantic similarity* mendapatkan nilai *MAE* sebesar 0,77 (Maharani, 2008).

