

BAB II

TINJAUAN PUSTAKA

2.1 Sistem Rekomendasi

Sistem rekomendasi adalah sebuah sistem yang bekerja untuk melakukan pencarian dan rekomendasi berupa informasi, produk atau layanan yang bersifat personal (Sarwar, B. et al, 2001).

Rekomendasi yang diberikan adalah hasil dari observasi terhadap keadaan dan keinginan pelanggan. Sistem rekomendasi memanfaatkan opini seseorang pada suatu konten atau kategori tertentu, untuk membantu seseorang dalam memilih produk. Oleh karena itu sistem rekomendasi memerlukan model yang tepat untuk menghasilkan rekomendasi yang sesuai dengan keinginan pelanggan (McGinty & Smyth, 2006).

Keakuratan sistem rekomendasi bergantung pada algoritma yang digunakan. Sistem rekomendasi yang dapat mengenalkan item-item yang membuat *user* tertarik dan dapat meyakinkan *user* merupakan sistem yang efektif. Berikut beberapa ciri-ciri sistem rekomendasi yang efektif (Sebastia. et al. 2009) :

1. Memberikan rasa percaya pada *user*
2. Memberikan rincian item yang direkomendasikan seperti gambar dan penilaian dari *user* lainnya
3. Mampu menyarankan item yang belum pernah dilihat atau diketahui oleh *user*

2.2 *Content-based Filtering*

Content-based filtering (CBF) adalah algoritma yang bergantung pada penekanan analisis atribut untuk menghasilkan sebuah prediksi. Dalam CBF rekomendasi dibuat berdasarkan profil *user* yang menggunakan fitur yang diambil dari konten *item* yang telah dievaluasi oleh *user* di masa lalu (F. O. Isinkaye, et al, 2015).

Item yang sebagian besar terkait dengan *item* yang dinilai positif akan direkomendasikan ke *user*. CBF sendiri menggunakan berbagai jenis model untuk

menemukan kesamaan antar *item* untuk menghasilkan rekomendasi yang akurat. Pada CBF tidak memerlukan profil *user* lain karena tidak mempengaruhi rekomendasi. Jika ada perubahan pada profil *user* CBF masih dapat menyesuaikan rekomendasinya dengan waktu yang sangat singkat. Kekurangan utama dari CBF adalah kebutuhan untuk memiliki pengetahuan dan deskripsi yang mendalam tentang fitur *item* dalam profil *user* (F. O. Isinkaye, et al, 2015).

Cara kerja CBF adalah dengan mengumpulkan semua *item* yang ada pada riwayat *user*. Fitur set diwakili dengan $k \in \{\text{genre, tema, publisher, developer, platform}\}$. Nilai *rating* dari keseluruhan *user* diwakili dengan $r(u, i)$ u untuk *item* dan i untuk fitur. Jika *item* memiliki fitur j maka nilai menjadi 1 dan *rating* keseluruhan *user* akan berkontribusi pada jumlah tersebut. Jika *item* tidak memiliki fitur j maka nilai menjadi 0.

Untuk menghitung bobot setiap fitur dari *user* digunakan persamaan berikut:

$$wk(u, j) = \frac{1}{I_u^{train}} \times \sum x_{k,u}(i, j) \times r(u, i) \quad (2.1)$$

Keterangan :

u = Riwayat *user* u

j = Fitur set yang digunakan untuk menghasilkan rekomendasi

$x_{k,u}(i, j)$ = Nilai fitur dari *item* ke- i

$r(u, i)$ = Nilai *rating* keseluruhan *user* untuk *item* ke- i

I_u^{train} = Jumlah keseluruhan riwayat yang ada pada *user*

Setelah bobot semua fitur didapat selanjutnya *item* yang berkaitan akan dihitung berdasarkan elemen fitur yang ada pada *item* tersebut. Nilai bobot dari setiap *item* akan diurutkan dari yang terbesar ke terkecil. Setiap *item* yang memiliki nilai bobot terbesar akan diambil fitur yang ada pada *item* tersebut untuk digunakan sebagai atribut *filtering*.

2.3 Collaborative Filtering

Collaborative filtering (CF) merupakan sebuah algoritma rekomendasi yang digunakan untuk mengidentifikasi pelanggan sesuai dengan selera yang sama.

Umumnya, CF mengumpulkan nilai rating dari berbagai *item* dari *user* yang berbeda-beda. Kemudian memberikan rekomendasi berdasarkan penilaian tersebut kepada pelanggan tertentu. CF diwakili dengan matriks *rating*, di mana *user* diwakili sebagai baris dan *item* sebagai kolom. Nilai yang ada pada kolom mewakili penilaian yang diberikan oleh pelanggan tentang *item* (Raza, U. H., 2013).

CF sendiri dibagi menjadi 2 tipe yaitu, *user-based* dan *item-based*.

1. *User-based*

User-based CF merupakan algoritma yang bergantung penuh pada korelasi data setiap *user*, yang ditentukan oleh koefisien preferensi pribadi *user*. Algoritma akan memeriksa data preferensi *user* satu per satu, apakah saling berkaitan atau tidak. Produk yang didapat dari pencarian melalui preferensi tidak akan langsung dinilai. Produk akan dihitung skalabilitasnya melalui nilai agregasi setiap *user* yang memiliki preferensi sama.

Namun *user-based* CF hanya cocok digunakan untuk data *user* dengan skala kecil. Data *user* yang besar dapat memberikan beban yang lebih besar pada memori karena algoritma memeriksa data preferensi *user* satu per satu (Raza, U. H., 2013).

Cara kerja *user-based* CF adalah dengan mengumpulkan data riwayat *user* yang memiliki preferensi atau selera yang sama. Nilai *rating* yang diberikan *user* digunakan untuk mendapatkan nilai hasil *rating* rata-rata dari setiap *user* pada suatu *item*. Nilai *rating item* yang diberikan oleh *user* diwakili dengan $R_{u,k}$ dan nilai *rating* yang diberikan oleh *user* ke-N diwakili dengan $R_{u,l}$. Semua nilai *rating* dari *user* ke-N akan dikalkulasikan dengan nilai *rating* yang diberikan oleh *user*.

User-based CF dihitung dengan persamaan :

$$sim(k, l) = \frac{\sum_{u=1}^m (R_{u,k} - \bar{R}_k)(R_{u,l} - \bar{R}_l)}{\sqrt{\sum_{u=1}^m (R_{u,k} - \bar{R}_k)^2} \sqrt{\sum_{u=1}^m (R_{u,l} - \bar{R}_l)^2}} \quad (2.2)$$

Keterangan :

- m = Notasi total pelanggan yang me-rating produk
- u = User ke-N

$R_{u,k}$ = Rating yang diberikan oleh user ke-N pada produk k

$R_{u,l}$ = Rating yang diberikan oleh user ke-N pada produk l

\bar{R}_k = Rata-rata rating produk k

\bar{R}_l = Rata-rata rating produk l

2. *Item-based*

Item-based CF adalah algoritma yang didasarkan pada hubungan setiap *item* atau produk. *Item-based* CF hanya menggunakan kesamaan atribut *item* tanpa memperdulikan hubungan dengan *user* lainnya. Algoritma ini akan mengambil semua *item* yang sudah pernah dinilai walaupun memiliki preferensi yang berbeda dari *user*. Pendekatan berbasis *item* ini diambil dari langkah pertama dan menentukan seberapa mirip *item* tersebut dengan *item* lainnya. Setelah menemukan semua *item* yang dirasa mirip, maka rekomendasi ditentukan berdasarkan rata-rata nilai *rating user* pada *item* tersebut (Sarwar, B. et al, 2001).

Salah satu langkah paling penting dalam *item-based* CF adalah menghitung kesamaan antara *item* lalu memilih yang paling mirip. Data *item* akan dibuat kedalam matriks di mana baris matriks mewakili *user* dan kolom mewakili *item*.

Kesamaan antara dua *item* i dan j diukur dengan menghitung nilai korelasi dari keseluruhan *user*. Nilai *rating* yang diberikan *user* pada produk i diwakili dengan $R_{u,i}$ dan nilai *rating* yang diberikan oleh *user* pada produk j diwakili dengan $R_{u,j}$.

Item-based CF dihitung dengan persamaan :

$$sim(i, j) = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_i) (R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_i)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_j)^2}} \quad (2.3)$$

Keterangan :

u = *User*

$R_{u,i}$ = Rating yang diberikan oleh *user* pada produk i

$R_{u,j}$ = Rating yang diberikan oleh *user* pada produk j

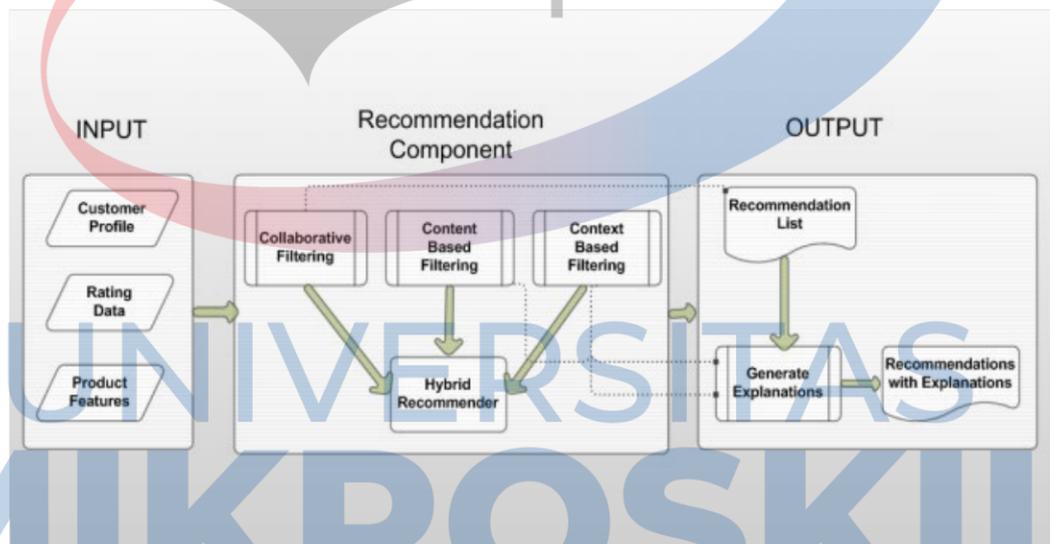
\bar{R}_i = Rata-rata rating produk i

$$\bar{R}_j = \text{Rata-rata rating produk } j$$

2.4 Hybrid Recommender

Hybrid Recommender merupakan teknik rekomendasi yang mengandalkan gabungan teknik A dan B untuk menggunakan keuntungan dari A dan memperbaiki kelemahan B. Misalnya *Collaborative filtering* yang tidak dapat merekomendasikan *item* yang tidak memiliki nilai *rating*. Kelemahan itu dapat ditutupi dengan menggabungkan algoritma *Content-based* dengan *Collaborative filtering*. *Hybrid* sendiri dapat menggabungkan lebih dari dua teknik untuk menciptakan sistem hibrida baru (Ricci F., et al, 2011).

Sistem *Hybrid Recommender* sendiri memiliki tiga komponen penting untuk dapat menghasilkan rekomendasi seperti yang ada pada gambar 2.1:



Gambar 2.1 Diagram model Hybrid Recommender

Hybrid Recommender sendiri memerlukan beberapa masukan untuk menghasilkan rekomendasi. Masukan dari sistem ini juga berisi ketertarikan *user*, data *user*, data *rating* dan data produk. Informasi riwayat *user* yang akurat memiliki peranan penting untuk mengintegrasikan teknik rekomendasi yang berbeda. Informasi riwayat *user* dapat dikumpulkan dengan berbagai cara contohnya dari riwayat pembelian sebelumnya, dari *feedback* yang diberikan *user* atau dari produk yang diberikan *rating* oleh *user* (Raza, U. H., 2013).

Hybrid Recommender terbagi kedalam beberapa jenis seperti di bawah ini:

1. *Switching*

Pada *switching* beberapa teknik rekomendasi tidak digabungkan melainkan memilih salah satu diantara banyak teknik yang dipakai. Pemilihan ini dilakukan sesuai situasi yang dihadapi. Kriteria pemilihannya bergantung pada nilai kepercayaan hasil dan kriteria eksternal yang komponennya mungkin berbeda tergantung situasi.

2. *Mixed*

Teknik ini didasarkan pada penggabungan dan penyajian beberapa hasil rekomendasi dari teknik berbeda menjadi satu. Setiap teknik yang ada pada teknik hibrida ini harus dapat menghasilkan daftar rekomendasi dan dapat digabungkan menjadi satu daftar tunggal dari banyak hasil rekomendasi dengan teknik yang berbeda.

3. *Cascade*

Teknik hibrida ini melibatkan proses yang bertahap. Pada teknik ini, satu teknik rekomendasi digunakan untuk menghasilkan daftar *item* yang akan direkomendasikan, kemudian teknik kedua digunakan untuk memperbaiki hasil dari rekomendasi sebelumnya.

4. *Weighted Sum*

Merupakan sebuah *Hybrid Recommender* di mana skor *item* yang direkomendasikan dihitung berdasarkan hasil dari semua teknik yang ada pada hibrida. *Weighted sum* menghitung prediksi pada *item* i untuk *user* u dengan menghitung nilai *rating* yang diberikan oleh *user* pada *item* yang mirip dengan *item* i . Setiap nilai *rating* yang diberikan oleh *user* u akan dilakukan pembobotan. Penghitungan nilai bobot setiap *rating* diambil dari nilai kesamaan *item* yang dihasilkan dari algoritma *Collaborative filtering* (Sarwar, B. et al, 2001).

Untuk mendapatkan hasil yang akurat maka *weighted sum* menggabungkan nilai kesamaan produk i dengan nilai *rating* produk i dan dihitung menggunakan persamaan seperti di bawah ini:

$$(2.4)$$

$$P_{u,i} = \frac{\sum \text{Allsimilaritems}, N(S_{i,N} * R_{u,N})}{\sum \text{Allsimilaritems}, N(| S_{i,N} |)}$$

Keterangan :

$S_{i,N}$ = Hasil kesamaan produk i dari produk set N

$R_{u,N}$ = Hasil rating dari *user* u pada set N

$| S_{i,N} |$ = Hasil absolute dari $S_{i,N}$

2.5 Precision

Precision merupakan cara ukur yang menghitung proporsi jumlah data yang ditemukan dan dianggap relevan untuk kebutuhan si pencari informasi, *precision* berkaitan dengan kemampuan sistem untuk tidak memanggil data yang tidak relevan (Hasugian, 2006).

Precision biasanya dipakai pada data *mining* untuk menunjukkan bahwa proporsi prediksi benar-benar positif. Langkah dan kombinasi *precision* hanya terfokus pada data positif dan mengabaikan data negatif yang tidak berkaitan dari kebutuhan si pencari informasi (Powers, D.M.W., 2011).

Untuk menghitung nilai *precision* digunakan rumus:

$$Precision = \frac{\text{hitCount}}{N} \quad (2.5)$$

Keterangan:

hitcount = Jumlah *game* relevan yang terambil

N = Jumlah *game* relevan dalam database

2.6 Recall

Recall adalah cara ukur untuk menghitung proporsi jumlah dokumen yang dapat ditemukan-kembali oleh sebuah proses pencarian, perolehan *recall* berhubungan dengan kemampuan sistem untuk memanggil dokumen yang relevan (Hasugian, 2006).

Recall menggunakan proporsi *real* positif yang ada pada prediksi positif untuk mengukur cakupan data. Hasil yang diinginkan adalah cerminan dari fitur

yang relevan tanpa memperdulikan berapa banyak dokumen relevan yang ada (Powers, D.M.W., 2011).

Untuk menghitung nilai *recall* digunakan rumus:

$$Recall = \frac{hitCount}{I_u^{test}} \quad (2.6)$$

Keterangan:

hitcount = Jumlah *game* relevan yang terambil

I_u^{test} = Jumlah *game* terambil dalam pencarian

2.7 *F – Measure*

F-Measure merupakan cara pengukuran yang berfokus pada nilai kombinasi dari contoh positif dan prediksi. *F-Measure* membutuhkan nilai pengukuran yang dihasilkan oleh *precision* dan *recall* yang juga disebut sebagai *harmonic mean*. *F-Measure* sendiri tidak menghitung nilai *true* negatif, yang dilakukan *F-Measure* hanya menormalkan pengukuran menjadi sebuah nilai ideal (Powers, D.M.W., 2011).

F-Measure dihitung dengan menggunakan rumus:

$$F - Measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.7)$$

UNIVERSITAS
MIKROSKIL