



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB II

### LANDASAN TEORI

#### 2.1 Sistem Rekomendasi

Sistem rekomendasi adalah perangkat lunak dan teknik yang menyediakan sugesti mengenai suatu *item* kepada *user* (Ricci dkk., 2011). Sistem rekomendasi telah terbukti bermanfaat bagi *online users* untuk mengatasi informasi yang berlebihan dan telah menjadi salah satu *tool* yang populer dan sangat *powerful* pada dunia *e-commerce* (Ricci dkk., 2011). Sistem rekomendasi ditujukan kepada individu yang kurang pengalaman atau kompetensi dalam mengevaluasi *item* dalam jumlah yang besar, yang ditawarkan oleh suatu *website* (Resnick dan Varian, 1997).

##### 2.1.1 Content-Based Filtering

Sistem menerapkan pendekatan *content-based filtering* untuk menganalisa dokumen atau deskripsi dari *items* yang sebelumnya telah di-*rating* oleh *user*, dan membuat sebuah model atau profil minat *user* berdasarkan fitur-fitur dari objek yang di-*rating* tersebut. Sistem rekomendasi yang menggunakan metode content-based akan memberikan rekomendasi *item* kepada pengguna berdasarkan deskripsi dari *item* dan profil dari minat pengguna (Pazzani dan Billsus, 2007).

Menurut Degemmis dkk. (2007), penerapan *content-based* pada sistem rekomendasi memiliki beberapa kelebihan, antara lain sebagai berikut.

- a. *User independence*: proses rekomendasi tidak bergantung pada *user* lain.
- b. *Transparansi*: penjelasan mengenai bagaimana sistem bekerja dapat dijabarkan secara eksplisit.

- c. *New item*: sistem dapat merekomendasikan *item* yang belum di-*rating*.

Masih menurut Degemmis dkk. (2007), selain memiliki kelebihan, metode *content-based* juga memiliki beberapa kelemahan, antara lain sebagai berikut.

- a. Konten terbatas: terdapat batasan alamiah pada jumlah dan tipe dari fitur yang diasosiasikan, baik secara otomatis maupun manual, terhadap objek yang direkomendasikan.
- b. *New user*: sistem tidak mampu merekomendasikan *item* kepada *user* baru, karena memerlukan *rating* dari *user* tersebut untuk memahami preferensinya.

### 2.1.2 Collaborative Filtering

*Collaborative Filtering* (CF) adalah metode yang memproduksi rekomendasi bagi *user* mengenai *item* berdasarkan pola dari *rating* atau pemakaian tanpa memerlukan informasi mengenai *item* ataupun *user*. Untuk menghasilkan rekomendasi, sistem rekomendasi yang menerapkan CF perlu untuk menghubungkan dua entitas utama yang berbeda, yaitu *items* dan *users* (Bell dan Koren, 2007).

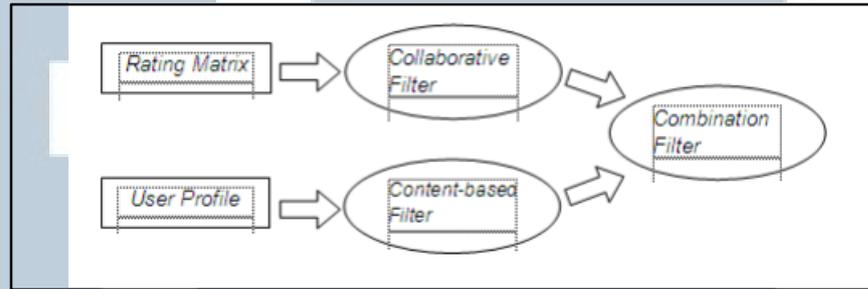
Ada dua pendekatan utama pada teknik CF, yaitu *the neighborhood approach* dan *latent factor models*. Pendekatan *neighborhood* fokus pada relasi antara *items* atau antara *users*, sedangkan *latent factor model* melakukan pendekatan alternatif dengan mengubah baik *items* maupun *users* menjadi *latent factor space*.

### 2.1.3 Hybrid Filtering

*Hybrid Filtering* adalah teknik pada sistem rekomendasi yang menggabungkan satu atau lebih teknik rekomendasi dengan tujuan untuk mengatasi kekurangan masing-masing pendekatan, sehingga menghasilkan

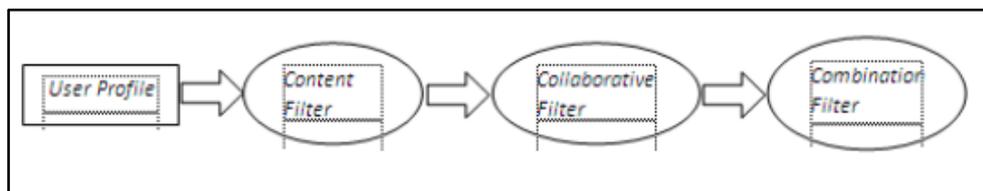
rekomendasi yang lebih baik (Wahyo dan Anggriawan, 2015). Terdapat beberapa cara penggabungan yang dapat dilakukan, yaitu sebagai berikut.

- a. Penggabungan secara linier (*Hybrid Linear Combination*): Teknik ini menggabungkan hasil prediksi (*rating*) dari metode *content-based* dan *collaborative*.



Gambar 2.1 Alur Proses Hybrid Linear Combination (Wahyo dan Anggriawan, 2015)

- b. Penggabungan secara *sekuensial* (*Hybrid Sequential Combination*): Penggabungan yang dilakukan dengan menghitung salah satu metode kemudian hasilnya digabungkan dengan metode lainnya.



Gambar 2.2 Alur Proses Hybrid Sequential Combination (Wahyo dan Anggriawan, 2015)

- c. Penggabungan *Item-based Clustering Hybrid Method* (ICHM): Penggabungan ini mengintegrasikan informasi *item* dan *rating* pengguna untuk menghitung kemiripan *item-item*.

Burk (2002) memperkenalkan taksonomi untuk sistem rekomendasi *hybrid*. Taksonomi ini diklasifikasikan ke dalam tujuh kategori, yaitu sebagai berikut (Burk, 2002).

- a. *Weighted Hybrid*: Nilai komponen dari sistem rekomendasi yang berbeda digabungkan secara numerik.
- b. *Switching Hybrid*: Sistem menjalankan salah satu metode, apabila hasilnya tidak memuaskan, sistem akan menggunakan metode lainnya.
- c. *Mixed Hybrid*: Rekomendasi dari berbagai metode direkomendasikan bersama.
- d. *Feature Combination*: Fitur-fitur dari berbagai sumber pengetahuan digabungkan dan diberikan algoritma rekomendasi.
- e. *Feature Augmentation*: Salah satu teknik yang digunakan untuk menghitung sebuah fitur yang kemudian dimasukkan ke teknik selanjutnya.
- f. *Cascade*: Rekomendasi yang memiliki prioritas tinggi sebagai solusi pemecahan masalah dalam perbaikan.
- g. *Meta-level*: Salah satu teknik rekomendasi yang diterapkan dan menghasilkan beberapa jenis model, yang kemudian dijadikan input oleh teknik lainnya.

## 2.2 Similarity Measurement

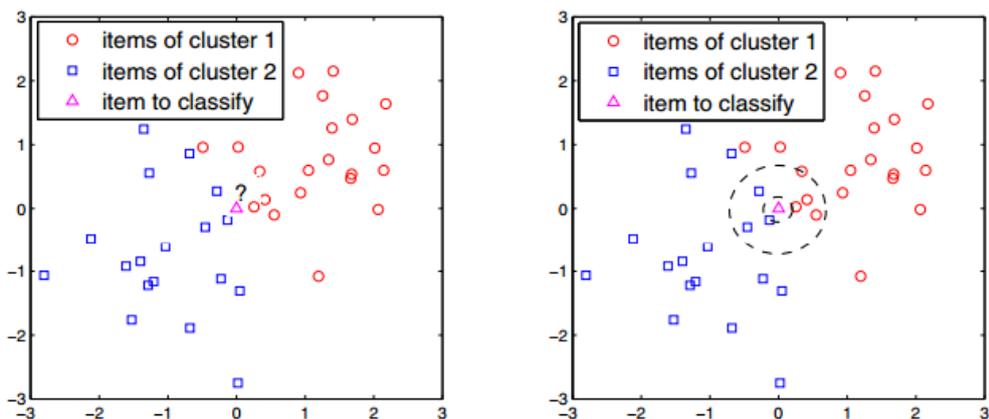
Metode yang digunakan untuk menghitung *similarity* atau kesamaan antara *items* ataupun antara *users*. Salah satu metode untuk mengukur tingkat kemiripan yang paling banyak digunakan adalah *cosine similarity*.

$$\cos(\theta) = \frac{A \cdot B}{|A| \cdot |B|} \dots(2.1)$$

*Similarity* antara *items* atau *users* perlu dicari dalam rangka untuk menentukan jarak antara *items* ataupun *users* sehingga proses rekomendasi dapat dilakukan.

### 2.3 Algoritma K-Nearest Neighbor

Algoritma K-Nearest Neighbor atau yang sering disebut K-NN adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut (Widiarsana, 2011). Metode ini termasuk dalam kategori *instance-based learning*, yaitu klasifikasi dilakukan dengan cara membandingkan langsung data uji dengan data pelatihan yang telah dikumpulkan (Lukito, 2016).



Gambar 2.3 Contoh Penerapan K-Nearest Neighbor (Ricci dkk., 2011).

Diberikan titik A, akan ditemukan sejumlah k objek atau titik *training* yang paling dekat dengan titik A. Algoritma K-NN sangatlah sederhana, bekerja berdasarkan jarak terpendek dari *query instance* ke *training sample* untuk menentukan K-NN-nya. *Training sample* diproyeksikan ke ruang berdimensi banyak, di mana masing-masing dimensi merepresentasikan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi *training sample*. Sebuah titik pada ruang ini ditandai kelas c jika kelas c merupakan klasifikasi yang paling

banyak ditemui pada  $k$  buah tetangga terdekat dari titik tersebut (Ricci dkk., 2011).

Isu atau tantangan pada *K-Nearest Neighbors* adalah bagaimana memilih *value* atau nilai dari  $k$ . Jika  $k$  terlalu kecil, maka *classifier* akan sensitif terhadap *noise points*, namun jika  $k$  terlalu besar, maka *neighborhood* akan mencakup terlalu banyak *point* atau titik dari kelas yang lain. Di sisi lain, metode KNN, meskipun *simple* dan intuitif, memiliki akurasi yang baik dan sangat mampu menerima perkembangan data. Faktanya, supremasi dari metode ini membuatnya secara *de facto* menjadi standar bagi rekomendasi dengan metode CF, dan hanya disaingi oleh pendekatan yang berdasarkan pengurangan dimensi (Amatriain dkk., 2009).

#### 2.4 The Movie Database

*The Movie Database* (TMDb) adalah komunitas *database movies* dan TV. Sejak berdiri dari tahun 2008 hingga saat ini, sudah ada lebih dari 100.000 *developers* dan perusahaan menggunakan *platform* ini. TMDb merupakan salah satu *platform movie database* internasional, yang mendukung 39 bahasa dan digunakan lebih dari 180 negara setiap harinya (TMDb, tanpa tahun).

Film yang disimpan di TMDb dikategorikan menjadi dua, yaitu *movie* dan TV *shows*. Pada kategori *movie* sendiri, film yang disimpan memiliki beberapa atribut, seperti *genre*, tahun rilis, durasi, dan sebagainya. Selain atribut-atribut tersebut, TMDb juga menyediakan informasi yang berkaitan dengan sebuah film, seperti aktor dan aktris yang berperan dalam suatu film.

TMDb adalah *platform* yang bersifat *open source*, yang berarti dapat diakses oleh setiap orang. Sebelum dapat mengakses data dari TMDb, sebelumnya

pengguna harus memenuhi ketentuan yang dibuat oleh TMDb, salah satunya adalah dengan membuat akun pada TMDb untuk mendapatkan *Application Program Interface (API) Key* yang diperlukan saat ingin mengakses data dari TMDb.

## 2.5 Skala Likert

Skala Likert adalah suatu skala psikometrik yang umum digunakan dalam kuesioner, dan merupakan skala yang paling banyak digunakan dalam riset berupa survei. Menurut Sugiyono (2012), Skala Likert adalah skala yang digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Dengan skala Likert, maka variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator tersebut dijadikan sebagai titik tolak ukur menyusun *item-item* instrumen yang dapat berupa pertanyaan atau pernyataan. Tabel 2.1 menunjukkan nilai dan interval yang digunakan pada skala Likert.

Tabel 2.1 Nilai dan Interval Skala Likert

Interval	Kategori	Nilai
Skor $\geq$ 80%	Sangat Setuju	5
80% > Skor $\geq$ 60%	Setuju	4
60% > Skor $\geq$ 40%	Netral	3
40% > Skor $\geq$ 20%	Tidak Setuju	2
Skor < 20%	Sangat Tidak Setuju	1

Untuk menghitung persentase skor dari kuesioner yang telah diisi, maka dapat menggunakan rumus yang diberikan oleh Sugiyono (2012).

$$\text{Persentase Skor} = ((\text{Sangat Tidak Setuju} * 1) + (\text{Tidak Setuju} * 2) + (\text{Netral} * 3) + (\text{Setuju} * 4) + (\text{Sangat Setuju} * 5) / 5 * \text{Jumlah Responden}) * 100\% \quad \dots(2.2)$$

## 2.6 Cronbach Alpha

Cronbach Alpha digunakan untuk mengukur keandalan indikator-indikator yang digunakan dalam kuesioner penelitian (McDaniel dan Gates, 2013). Untuk menghitung realibilitas jawaban responden dapat menggunakan rumus *alpha*, sebagai berikut.

$$r_{11} = \left[ \frac{k}{k-1} \right] \left[ 1 - \frac{\sum S_i}{S_t} \right] \quad \dots(2.3)$$

Dimana:

$r_{11}$  = nilai realibilitas

$\sum S_i$  = jumlah varians skor tiap-tiap *item*

$S_t$  = varians total

$k$  = jumlah *item*

Menurut Hair et al. (2007), nilai tingkat keandalan Cronbach Alpha dapat ditunjukkan seperti pada Tabel 2.1.

Tabel 2.2 Tingkat Keandalan Cronbach Alpha

Nilai Cronbach Alpha	Tingkat Keandalan
0.0 – 0.20	Kurang Andal
>0.20 – 0.40	Agak Andal
>0.40 – 0.60	Cukup Andal
>0.60 – 0.80	Andal
>0.80 – 1.00	Sangat Andal

## 2.7 Uji Validitas

Uji Validitas merupakan salah satu uji instrumen yang digunakan untuk mengetahui kelayakan butir-butir dalam suatu daftar pertanyaan dalam mendefinisikan suatu variabel (Sujarweni, 2015). Validitas sebuah pertanyaan dapat dihitung dengan menggunakan rumus korelasi Pearson, sebagai berikut.

$$r_{xy} = \frac{n\sum XY - (\sum X)(\sum Y)}{\sqrt{(n\sum X^2 - (\sum X)^2)(n\sum Y^2 - (\sum Y)^2)}} \quad \dots(2.4)$$

Dimana:

$r_{xy}$  = koefisien korelasi

$n$  = jumlah responden uji coba

$X$  = skor tiap item

$Y$  = skor seluruh item responden uji coba

Setelah mendapatkan nilai korelasi dari setiap butir pertanyaan, maka selanjutnya nilai korelasi yang telah didapatkan dibandingkan dengan tabel  $r$ . Sebelum melakukan proses perbandingan, diperlukan perhitungan nilai *degree of freedom* (df) dengan rumus sebagai berikut.

$$df = N - k \quad \dots(2.5)$$

Dimana:

df = *deegre of freedom* / derajat kebebasan

$N$  = jumlah responden

$k$  = variabel yang diperhitungkan

Tabel 2.3 menunjukkan tabel nilai r Pearson dengan nilai signifikansi sebesar 0.05. Tabel ini digunakan untuk menentukan validitas sebuah pertanyaan dalam kuesioner yang digunakan dalam penelitian.

Tabel 2.3 Tabel r ( $p = 0.05$ )

N	R	N	R	N	R
3	0.997	27	0.381	55	0.266
4	0.950	28	0.374	60	0.254
5	0.878	29	0.367	65	0.244
6	0.811	30	0.361	70	0.235
7	0.754	31	0.355	75	0.227
8	0.707	32	0.349	80	0.220
9	0.666	33	0.344	85	0.213
10	0.632	34	0.339	90	0.207
11	0.602	35	0.334	95	0.202
12	0.576	36	0.329	100	0.195
13	0.553	37	0.325	125	0.176
14	0.532	38	0.320	150	0.159
15	0.514	39	0.316	175	0.148
16	0.497	40	0.312	200	0.138
17	0.482	41	0.308	300	0.113
18	0.468	42	0.304	400	0.098
19	0.456	43	0.301	500	0.088
20	0.444	44	0.297	600	0.080
21	0.433	45	0.294	700	0.074
22	0.423	46	0.291	800	0.070
23	0.413	47	0.288	900	0.065
24	0.404	48	0.284	1000	0.062
25	0.396	49	0.281	-	-
26	0.388	50	0.279	-	-

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA