

# Sistem Rekomendasi Film menggunakan User-based Collaborative Filtering dan K-modes Clustering

Ichwanto Hadi<sup>1</sup>, Leo Willyanto Santoso<sup>2</sup>, Alvin Nathaniel Tjondrowiguno<sup>3</sup>

Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Jl. Siwalankerto 121-131, Surabaya 60236

Telp (031) – 2983455, Fax. (031) - 8417658

iwan180697@gmail.com<sup>1</sup>, leow@petra.ac.id<sup>2</sup>, alvin.nathaniel@petra.ac.id<sup>3</sup>

## ABSTRAK

Film adalah salah satu media hiburan yang populer di masyarakat. Banyaknya judul-judul yang telah rilis membuat masyarakat kesulitan untuk menemukan film mana yang mereka ingin tonton. Untuk mengatasi masalah tersebut, perlu adanya informasi mengenai film yang akan memudahkan masyarakat untuk menemukan film yang cocok dengan preferensi *user*, oleh sebab itu *user* perlu sebuah sistem yang dapat memberikan rekomendasi film.

Sistem rekomendasi film menggunakan *User-based Collaborative Filtering* adalah salah satu metode yang mampu memberikan rekomendasi. *K-modes Clustering* juga dapat digunakan sebagai penambah ketepatan dari rekomendasi dengan mengelompokkan riwayat preferensi *user*.

Menurut hasil pengujian terhadap metode *k-modes clustering*, jumlah *cluster* terbaik untuk *K-Modes Clustering* untuk rekomendasi film yang didapatkan dengan menggunakan *Elbow Method* dan *Silhouette Coefficient* adalah sebesar  $k = 3$ . Dari hasil pengujian ketepatan rekomendasi dengan *Mean Reciprocal Rank* (MRR) dihasilkan rata-rata MRR sebesar 0.17092270381865 untuk rekomendasi film dengan rasio data *train* dan *test* sebesar 80% : 20% dan rata-rata MRR sebesar 0.15072658511145 untuk rekomendasi film dengan rasio data *train* dan *test* sebesar 60% : 40%. Dari hasil dua pengujian diatas dapat disimpulkan bahwa tingkat ketepatan dari rekomendasi film menurut MRR cukup karena MRR-nya mendekati 0.

**Kata Kunci:** Rekomendasi Film, K-Modes, User-based Collaborative Filtering, Elbow Method, Silhouette Coefficient, MRR.

## ABSTRACT

Film is one of the popular entertainment media in the community. The number of titles that have been released makes it difficult for people to find which movies they want to watch. To overcome this problem, it is necessary to have information about the film that will make it easier for the public to find films that fit the user's preferences, therefore the user needs a system that can provide movie recommendations.

The movie recommendation system using User-based Collaborative Filtering is one method that is able to provide recommendations. K-mode Clustering can also be used as an additional accuracy of recommendations by grouping user preferences history.

According to the results of the testing of the k-modes clustering method, the best number of clusters for K-Modes Clustering for film recommendations obtained using the Elbow Method and Silhouette Coefficient is  $k = 3$ . From the results of testing the accuracy of the recommendations with Mean Reciprocal Rank

(MRR) generated average MRR of 0.17092270381865 for film recommendations with a data train and test ratio of 80%: 20% and an average MRR of 0.15072658511145 for film recommendations with a data train and test ratio of 60%: 40%. From the results of the two tests above, it can be concluded that the level of accuracy of the film recommendations according to the MRR is sufficient because the MRR is close to 0.

**Keywords:** Movie Recommendation, K-Modes, User-based Collaborative Filtering, Silhouette Coefficient, MRR.

## 1. PENDAHULUAN

Film adalah salah satu media hiburan yang populer di masyarakat. Sampai Desember 2018, tercatat telah ada 5,980,614 judul film yang telah rilis [6]. Banyaknya judul-judul yang telah rilis membuat masyarakat kesulitan untuk menemukan film mana yang mereka ingin tonton. Untuk mengatasi masalah tersebut, perlu adanya informasi mengenai film yang akan memudahkan masyarakat untuk menemukan film yang cocok dengan preferensi *user*, oleh sebab itu *user* perlu sebuah sistem yang dapat memberikan rekomendasi film.

Sistem rekomendasi adalah suatu cara yang dapat memberikan informasi atau rekomendasi yang sesuai dengan kesukaan pengguna berdasarkan informasi yang didapat dari pengguna tersebut. Dalam pembuatan sistem rekomendasi, ada dua metode rekomendasi yang bisa dipakai yaitu *collaborative filtering* dan *content-based filtering* [10]. Metode yang digunakan pada skripsi ini ialah *user-based collaborative filtering*. Metode ini memberikan rekomendasi kepada *user* dengan menentukan riwayat *user* yang mirip dengan riwayat *user* lain. Kelebihan dari metode ini ialah tidak perlu tahu isi dari *item* dan mampu menangkap perubahan dari preferensi *user*. Untuk menambah ketepatan dari rekomendasi maka pada skripsi ini digunakan *k-modes clustering* untuk mengelompokkan riwayat film yang ditonton *user*.

Dalam penelitian yang menggunakan *collaborative filtering* untuk merekomendasikan film, hasil yang didapatkan ialah sistem rekomendasi dapat menghasilkan rekomendasi film tetapi memiliki akurasi yang rendah karena peneliti menggunakan *item-based collaborative filtering* yang merekomendasikan film berdasarkan kemiripan film yang disukai dengan film lain [9]. *User* belum tentu menyukai film yang mirip dengan filmnya. Maka dari itu dalam skripsi ini akan digunakan metode *user-based collaborative filtering* dan *k-modes clustering* untuk merekomendasikan film sesuai dengan riwayat *user*.

## 2. LANDASAN TEORI

### 2.1. Sistem Rekomendasi

Sistem Rekomendasi merupakan sarana untuk mendukung pengguna untuk mencari atau menemukan dan memilih produk, layanan, atau informasi pada situs atau aplikasi tertentu. Ada 4

metode untuk membuat sistem rekomendasi yaitu *content based*, *collaborative based*, *demographic based*, *knowledge based* [10].

## 2.2. Collaborative Filtering

*Collaborative Filtering* merupakan metode yang sering digunakan dalam pembuatan sistem rekomendasi. *Collaborative filtering* memberikan rekomendasi terhadap *item* dengan mencari kemiripan *user* dengan *user* lain terhadap suatu *item* atau mencari kesamaan *item* dengan *item* lain.

*Collaborative Filtering* memiliki 2 metode yang berbeda yaitu *User-Based Collaborative Filtering* dan *Item-based Collaborative Filtering*. *User-Based Collaborative Filtering* merupakan suatu metode rekomendasi yang memberikan rekomendasi berdasarkan kemiripan antar *user*. *Item* yang direkomendasikan adalah *item* yang disukai oleh *user* lain yang preferensinya mirip dengan *user* utama. *Item-Based Collaborative Filtering* merupakan metode rekomendasi yang menggunakan *item* yang disukai pengguna sebagai parameternya. Kemudian akan dicari kemiripan *item* dengan *item* lain. *Item* yang mirip dengan *item* yang disukai pengguna yang nantinya akan dijadikan rekomendasi [5].

Kedua metode tersebut akan melakukan rekomendasi dengan melakukan *scoring* terhadap *item*. Metode yang digunakan dalam melakukan *scoring* rekomendasi adalah *weighted sum* [10]. Berikut ini adalah rumus *weighted sum* rekomendasi:

$$S_{(u,i)} = \frac{\sum_{v=1}^n R_{(v,i)} * Sim(u,v)}{\sum_{v=1}^n |Sim(u,v)|} \dots\dots\dots(1)$$

Dimana :

$S(u,i)$  = Score rekomendasi untuk user terhadap *item* i

$R(v,i)$  = Rating user v terhadap *item* i

$Sim(u,v)$  = Similarity antara user u dan user v.

## 2.3. K-modes Clustering

Metode *K-modes Clustering* merupakan pengembangan dari *K-means Clustering*. *K-modes Clustering* merupakan algoritma clustering yang khusus untuk menangani data yang nilainya berupa teks atau biner. Berikut ini merupakan langkah dari metode *K-modes Clustering*: [11]

- Tentukan jumlah *cluster* sebanyak k.
- Tentukan *centroid* awal dari masing-masing cluster.
- Mencari cluster terdekat dengan data yang ingin dikelompokkan dengan *simple matching method* atau *hamming distance*. *Hamming Distance* ditunjukkan pada Rumus sebagai berikut:

$$d(x, c) = \sum_{i=1}^n \delta(x_i, c_i) \dots\dots\dots(2)$$

$$\delta(x_i, c_i) = \begin{cases} 0 & \text{bila } x = c, \\ 1 & \text{bila } x \neq c \end{cases} \dots\dots\dots(3)$$

Dimana :

$d(x,c)$  = jarak antara data x ke centroid c

$x_i$  = atribut data x ke-i

$c_i$  = atribut centroid ke-I

- Masukkan data ke cluster terdekat.
- Hitung centroid dengan mencari modus dari setiap data yang ada di cluster.

- Ulangi langkah ke-3 sampai ke-5 sampai centroid tidak berubah terlalu banyak.

## 2.4. Jaccard Similarity

*Jaccard Similarity* atau *Jaccard Coefficient* dapat digunakan untuk mencari kemiripan antar suatu *item* dengan *item* lainnya. Dalam skripsi ini *item* yang dimaksud ialah *genre* film dan siapa saja yang pernah menonton dan memberikan rating terhadap film tersebut. Hasil dari *Jaccard Similarity* selalu diantara 0 sampai 1 [2]. *Jaccard Similarity* ditunjukkan pada rumus sebagai berikut.

$$j(A, B) = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \dots\dots\dots(4)$$

Dimana :

$j(A,B)$  = Kemiripan item A dan B

A = Jumlah atribut yang dimiliki item B

B = Jumlah atribut yang dimiliki item B

$A \cap B$  = Jumlah atribut item A yang sama dengan atribut item B

## 2.5. Data Preprocessing

*Data preprocessing* adalah salah satu proses di data mining. Banyak data yang kurang dikontrol sehingga terdapat value yang tidak memungkinkan (contoh *Age*: -2), kombinasi data yang tidak seharusnya (contoh., *Gender* : *Male*, *Pregnant* : *Yes*), data yang hilang, informasi yang redundant dan sebagainya. Data yang salah dapat berdampak buruk terhadap data yang akan dianalisa dan dapat menghasilkan hasil yang tidak akurat. Selain itu juga penggunaan data yang sudah dipreproses dapat mempermudah dan mempercepat mining process. Karena hal tersebut, data preprocessing menjadi salah satu proses yang sangat penting dan harus dilakukan pertama kali dalam *data mining* [8].

## 2.6. Elbow Method

Algoritma *K-modes* mensyaratkan jumlah *cluster* yang harus ditentukan sebelumnya. Salah satu metode populer untuk menentukan jumlah *cluster* adalah *elbow method*.

*Elbow method* hanya memerlukan melihat grafik garis yang (mudah-mudahan) menunjukkan lebih banyak *centroid* yang ditambahkan luasnya data di sekitar *centroid* berkurang. Dalam hal ini, luasnya data disebut *distortion* atau jumlah kesalahan kuadrat (SSE). *Distortion* dapat berkurang dengan cepat pada awalnya kemudian secara perlahan meratakan membentuk "siku" dalam grafik garis [4].

Berikut ini adalah cara mencari *distortion* atau sum of square.

$$\sum (Data X_i - Centroid X) + (Data Y_i - Centroid Y) \dots\dots(5)$$

Dimana :

Data  $X_i$  = data ke i dari cluster X

Centroid X = centroid dari cluster X

Data  $Y_i$  = data ke i dari cluster Y

Centroid Y = centroid dari cluster Y

## 2.7. Silhouette Coefficient

*Silhouette Coefficient* merupakan ukuran validitas dari suatu *cluster*. Berikut ini merupakan rumus dari *Silhouette Coefficient*.

$$Sil(i) = \frac{b-a}{\max(a,b)} \dots\dots\dots(6)$$

Dimana:

a = rata-rata dari *intra cluster distance*

b = rata-rata dari *inter cluster distance*

Sil(i) = *silhouette coefficient*

*Intra cluster distance* adalah jarak dari suatu objek dengan objek lainnya di dalam suatu *cluster*. Sedangkan *inter cluster distance* adalah jarak antara objek dengan objek lainnya di cluster yang berbeda. Semakin nilai *Silhouette Coefficient* mendekati nilai 1 maka objek yang dikelompokkan berada di dalam *cluster* yang tepat. Bila nilai *Silhouette Coefficient* mendekati nilai 0 maka objek-objek yang dikelompokkan berada di 2 *cluster* yang berdekatan [1].

## 2.8. Mean Reciprocal Rank

*Mean Reciprocal Rank* (MRR) dikaitkan dengan *user model* di mana *user* hanya ingin melihat satu dokumen yang relevan. Dengan asumsi bahwa pengguna akan melihat ke bawah peringkat sampai dokumen yang relevan ditemukan, dan dokumen tersebut berada di peringkat n, maka ketepatan set mereka melihat 1 / n, yang juga merupakan ukuran *reciprocal rank*. MRR adalah ukuran yang tepat untuk *known item search*, tempat dimana pengguna berusaha menemukan dokumen yang dia pernah lihat sebelumnya atau sebelumnya tahu apabila ada. MRR ini bisa disebut pencarian navigasi dalam kasus pencarian web [3].

$$MRR = \frac{1}{|n|} \sum_{i=1}^{|n|} \frac{1}{Rank} \dots\dots\dots(7)$$

Dimana:

n = jumlah data

Rank = rank dari suatu objek dalam data

MRR = mean reciprocal rank

## 2.9. Tinjauan Studi

Dari penelitian-penelitian sebelumnya, terdapat judul penelitian yang sama dengan metode yang berbeda dan terdapat juga judul penelitian yang beda tetapi memiliki metode pendukung yang sama. Penelitian pertama yang berjudul “Movies Recommendation using Collaborative Filtering and K-Means” memiliki hasil rekomendasi yang akurasi lemah [9]. Perbedaannya dengan skripsi ini adalah metode yang digunakan adalah *item-based collaborative filtering* dan *k-means clustering*. Penelitian kedua yang berjudul “K-Modes Clustering untuk Mengetahui Jenis Masakan Daerah yang Populer pada Website Resep Online (Studi Kasus Masakan Banjar di Cookpad.com)” memiliki hasil jumlah *cluster* terbaik berdasarkan kaidah *elbow method* adalah k = 4 dan k = 8 [7]. Perbedaan penelitian tersebut dengan skripsi ini adalah tujuan dari skripsi, dimana peneliti hanya menguji apakah metode *k-modes* bisa digunakan untuk *clustering*.

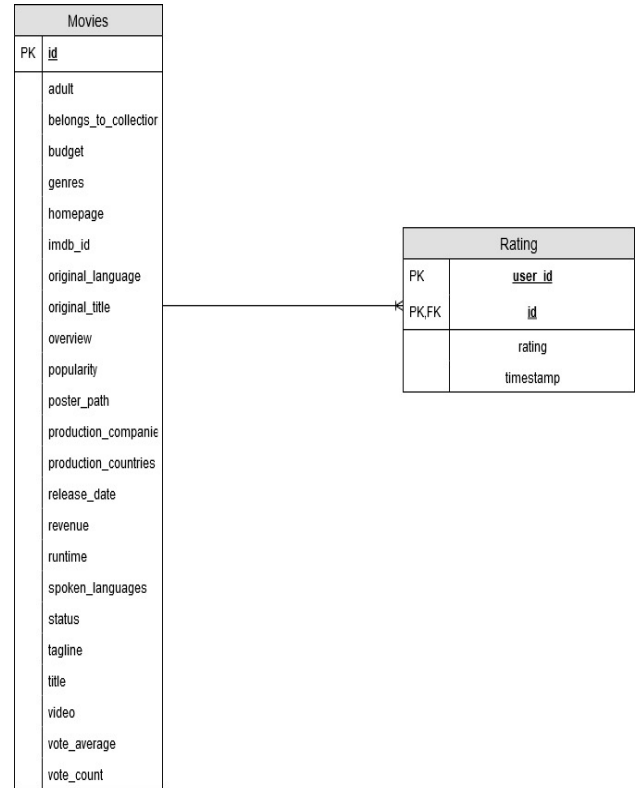
## 3. ANALISIS dan DESAIN SISTEM

### 3.1. Analisis Data

Analisis data dilakukan dengan tujuan untuk menganalisis data sehingga dapat ditentukan langkah yang akan diambil, dimana analisis data meliputi *entity relationship diagram* dan analisis contoh data.

*Entity Relationship Diagram* digunakan untuk menunjukkan relasi antar *table* di *database*. Berikut ini akan ditunjukkan *entity relationship diagram* dari *database* TMDb (*The Movies*

*Database*) yang digunakan pada penelitian ini. *Entity relationship diagram* tersebut bisa dilihat pada Gambar 1.



Gambar 1. *Entity relationship diagram*

Dari data yang diperlihatkan pada *entity relationship diagram*, atribut-atribut yang diperlukan ialah atribut *id*, *title*, *movie\_id*, *user\_id*, dan *rating*. Selanjutnya akan dilakukan analisis contoh data. Dalam analisis data yang ada di dalam *database* ditemukan data yang ada di dalam tabel *rating* tidak ada di dalam tabel *movies*. Tepatnya pada kolom *movieId* di tabel *rating* bila di cocokkan tidak ada di kolom *id* tabel *movies*. Contoh data yang tidak sesuai dapat dilihat pada Gambar 2 dan pengecekannya dilakukan di Gambar 3.

	userid	movieId	rating	timestamp
	0	1	31	2.5 1260759144
	1	1	1029	3.0 1260759179
	2	1	1061	3.0 1260759182
	3	1	1129	2.0 1260759185
	4	1	1172	4.0 1260759205
	...	...	...	...
	99999	671	6268	2.5 1065579370
	100000	671	6269	4.0 1065149201
	100001	671	6365	4.0 1070940363
	100002	671	6385	2.5 1070979663
	100003	671	6565	3.5 1074784724

Gambar 2. Contoh data *movieId* yang tidak ada di tabel *Movies*

```

g_cocok = ratings[~ratings.movieId.isin(movies.id)]
print(g_cocok.movieId)
print(g_cocok.movieId.isin(movies.id))

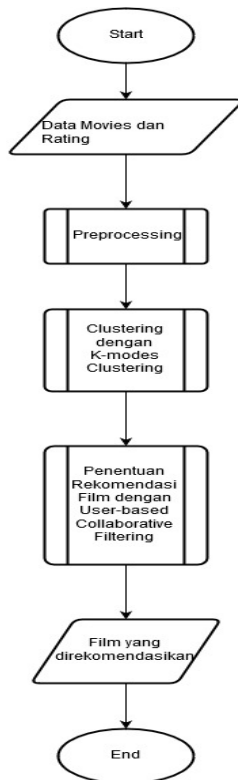
0      31
1     1029
2     1061
3     1129
4     1172
...
99999  6268
100000 6269
100001  6365
100002  6385
100003  6565
Name: movieId, Length: 100004, dtype: int64
0      False
1      False
2      False
3      False
4      False
...
99999  False
100000  False
100001  False
100002  False
100003  False
Name: movieId, Length: 100004, dtype: bool

```

Gambar 3. Pengecekan movieId dengan atribut id yang ada di tabel movies.

### 3.2. Desain Sistem

Desain sistem dilakukan dengan tujuan untuk menentukan arah pembuatan dan mengurangi kemungkinan terjadinya masalah untuk aliran proses pada program, dimana desain sistem meliputi gambaran secara umum, dan desain *flowchart* program. Berikut ini ialah gambaran sistem rekomendasi film secara umum.

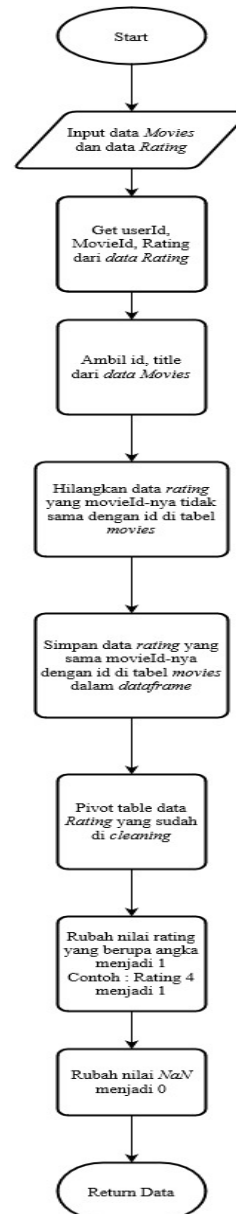


Gambar 4. Gambaran besar sistem rekomendasi film secara keseluruhan

Pada Gambar 4 ditunjukkan gambaran besar sistem rekomendasi film secara keseluruhan. Input berasal dari tabel *movies* dan *tabel rating*. Kemudian data-data tersebut akan diproses untuk menjadi sebuah rekomendasi untuk *user*. Pada Gambar 4 juga ditunjukkan proses-proses yang ada dalam pembuatan sistem rekomendasi film. Proses –proses tersebut meliputi proses *preprocessing*, *clustering* dengan *k-modes clustering*, dan penentuan rekomendasi dengan *user-based collaborative filtering*. Berikut ini dijelaskan masing-masing dari proses-proses tersebut.

#### 3.2.1 Preprocessing

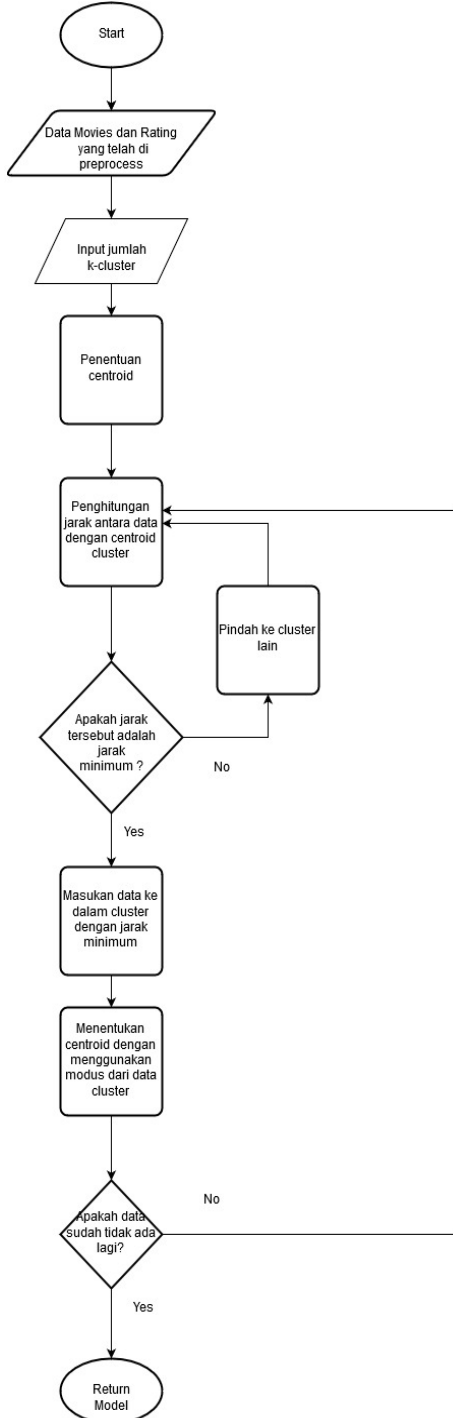
Proses pertama yang dilakukan adalah proses *preprocessing*. Proses ini bertujuan untuk mengolah data menjadi data yang siap dipakai oleh proses-proses selanjutnya. Yang dilakukan dalam proses *preprocessing* adalah *cleaning data*, *pivot table*, dan pemberian nilai 1 dan 0 pada isi data. Pada Gambar 5 akan ditunjukkan *flowchart* dari proses *preprocessing*.



Gambar 5. Desain *flowchart* proses *preprocessing*

### 3.2.2 Clustering dengan k-modes clustering

Proses kedua yang dilakukan oleh sistem rekomendasi film yang dibuat adalah proses *clustering*. *Clustering* yang dibuat menggunakan metode *k-modes* yang mengambil modus dari sebuah *cluster* sebagai titik tengah atau *centroid cluster* dan menggunakan *hamming distance* untuk mencari *cluster* terdekat. Pada Gambar 6 akan ditunjukkan *flowchart* dari proses *clustering*.



Gambar 6. Desain *flowchart* proses *clustering* dengan *k-modes clustering*.

### 3.2.3 Penentuan rekomendasi film dengan user-based collaborative filtering

Proses terakhir yang dilakukan oleh sistem rekomendasi film adalah proses penentuan rekomendasi film dengan *user-based collaborative filtering*. Yang dilakukan oleh proses ini ialah mencari *similarity* dan membuat *matrix user*, mencari beberapa *user* terdekat, mencari film yang belum ditonton oleh *user* dan sudah ditonton oleh *user* lain, dan penentuan rekomendasi. Pada Gambar 7 ditunjukkan desain *flowchart* dari proses penentuan rekomendasi film dengan *user-based collaborative filtering*.



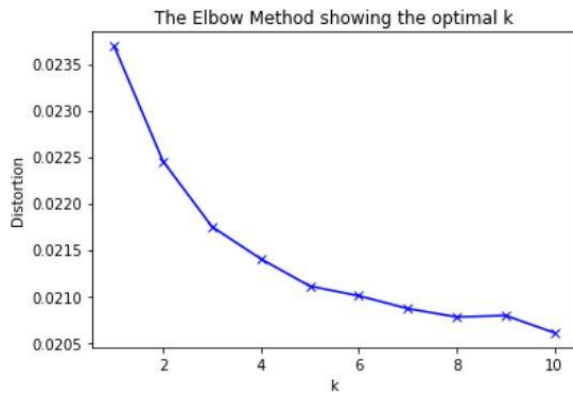
Gambar 7. Desain *flowchart* proses penentuan rekomendasi dengan *user-based collaborative filtering*.

## 4. PENGUJIAN SISTEM

Pengujian dilakukan pada laptop yang memiliki OS Windows 10 dengan CPU Intel Core i7-7700HQ @2,8GHz (8 CPU). Pengujian yang dilakukan meliputi pengujian *K-modes Clustering* dengan *Elbow Method*, perhitungan *Silhouette Coefficient* dan perhitungan *Mean Reciprocal Rank*.

### 4.1. Pengujian Elbow Method

Pengujian *Elbow Method* digunakan untuk mencari jumlah *cluster* terbaik. Parameter yang digunakan untuk *K-modes Clustering* ialah atribut *movieId*. Berikut ini adalah grafik dari hasil pengujian *cluster* dengan *Elbow Method*.



Gambar 8. Hasil pengujian *cluster* dengan *elbow method*.

Dari Gambar 8 dapat diketahui bahwa *K-modes Clustering* terbaik terdapat pada jumlah *cluster*  $k = 3$ . Jumlah *cluster*  $k = 3$  diambil karena berdasarkan grafik *distortion* menunjukkan penurunan yang curam di jumlah *cluster* sebelum 3 dan penurunan yang landai di jumlah *cluster* setelah 3.

### 4.2. Pengujian Silhouette Coefficient

Pada Tabel 3 ditunjukkan hasil pengujian *cluster* terbaik dengan *silhouette coefficient* dalam 4 kali percobaan. *Cluster* yang akan diuji coba ialah dari *cluster* 3 sampai 6.

Tabel 3. Pengujian jumlah *cluster* dengan *silhouette coefficient*.

Percobaan ke-	Jumlah K cluster	<i>Silhouette coefficient</i>
1	3	0.20789925069187895
2	4	0.17174491931489444
3	5	0.1285472574785336
4	6	0.018231365256793825

Dari hasil pengujian *cluster* dengan *silhouette coefficient*, didapatkan *cluster* 3 yang menjadi *cluster* terbaik karena memiliki *silhouette coefficient* yang paling besar. Semakin besar *silhouette coefficient* berarti data-data dikelompokkan dengan *cluster* yang sesuai.

### 4.3. Pengujian Mean Reciprocal Rank

Pengujian dilakukan dengan membandingkan rekomendasi data *rating user* yang dipisah menjadi 80% : 20% dan 60% : 40%.

data *train* dan *test*. Setelah data dipisah menjadi data *train* dan data *test*, maka dijalankan sistem rekomendasi film untuk kedua data tersebut dengan jumlah *cluster* = 3. Masing-masing memiliki hasil rekomendasi film dan kedua hasil tersebut diukur tingkat kebenarannya dengan *Mean Reciprocal Rank*.

Tabel 4. Mean reciprocal rank dengan 80% data *train* dan 20% data *test*.

User Id	Mean Reciprocal Rank
1	0.2313148367783838
2	0.1452188071767938
3	0.16326188512095546
4	0.17887969411658564
19	0.1834740693744758
23	0.22003029281027034
32	0.17999093383405504
40	0.17641092533012898
123	0.15239210357057287
154	0.07825349007431034
Rata-rata	0.17092270381865

Tabel 5. Mean reciprocal rank dengan 60% data *train* dan 40% data *test*.

User Id	Mean Reciprocal Rank
1	0.042366462167172145
2	0.1806846048256106
3	0.1328419795661225
4	0.22589378598747753
19	0.16562828207213054
23	0.25857877086969944
32	0.1474431179732747
40	0.08997393961965533
123	0.16761344156533056
154	0.09624146646806789
Rata-rata	0.15072658511145

Pada Tabel 4 dan Tabel 5 ditunjukkan hasil dari MRR dari rekomendasi film. Semakin besar nilai MRR semakin tepat film yang direkomendasikan. Dari hasil pengujian MRR pada Tabel 5.5 dan Tabel 5.6 dihasilkan rata-rata MRR sebesar 0.17092270381865 untuk rekomendasi film dengan rasio data *train* dan *test* sebesar 80% : 20% dan rata-rata MRR sebesar 0.15072658511145 untuk rekomendasi film dengan rasio data *train* dan *test* sebesar 60% : 40%. Dari hasil dua pengujian diatas dapat disimpulkan bahwa tingkat ketepatan dari rekomendasi film menurut *Mean Reciprocal Rank* cukup rendah, karena berdasarkan kaidah *Mean Reciprocal Rank* rekomendasi dikatakan kurang tepat apabila nilainya mendekati 0 dengan batas sebesar 0.5 dan dikatakan tepat apabila nilainya diatas 0.5 dan mendekati 1.

## 5. KESIMPULAN

Dari hasil perancangan dan pembuatan sistem dan aplikasi, dapat diambil kesimpulan antara lain:

- Dari hasil pengujian jumlah cluster terbaik yang digunakan untuk rekomendasi menggunakan *Elbow Method* dan *Silhouette Coefficient*, didapatkan jumlah cluster sebesar 3 yang merupakan jumlah cluster terbaik.
- Dari hasil pengujian ketepatan rekomendasi dengan *Mean Reciprocal Rank* (MRR) dihasilkan rata-rata MRR sebesar 0.17092270381865 untuk rekomendasi film dengan rasio data *train* dan *test* sebesar 80% : 20% dan rata-rata MRR sebesar 0.15072658511145 untuk rekomendasi film dengan rasio data *train* dan *test* sebesar 60% : 40%. Dari hasil dua pengujian diatas dapat disimpulkan bahwa tingkat ketepatan dari rekomendasi film menurut MRR cukup rendah, karena berdasarkan kaidah MRR rekomendasi dikatakan kurang tepat apabila nilainya mendekati 0 dengan batas sebesar 0.5 dan dikatakan tepat apabila nilainya diatas 0.5 dan mendekati 1.

## 6. DAFTAR PUSTAKA

- [1] Aranganayagi, S., & Thangavel, K. 2007. *Clustering Categorical Data Using Silhouette Coefficient as a Relocating Measure*. <https://ieeexplore.ieee.org/abstract/document/4426662>
- [2] Arnaboldi, V., Passarella, A., Conti, M., & Dunbar, R. I. 2015. *Online Social Networks : Human Cognitive Constraints in Facebook and Twitter Personal Graphs*. Amsterdam, Netherlands: Elsevier Inc.
- [3] Craswell, N. 2009. *Mean Reciprocal Rank*. [https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-39940-9\\_488](https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-39940-9_488)
- [4] Guaracci, C. 2016. *Easily understand K-means clustering*. <https://avidml.wordpress.com/2016/10/29/easily-understand-k-means-clustering/>
- [5] Hu, Q. Y., Zhao, Z. L., Wang, C. D., & Lai, J. H. 2017. An item orientated recommendation algorithm from the multi-view perspective. In *Neuro Computing*, 261-272.
- [6] IMDb. n.d. *IMDb Statistics*. [www.imdb.com/pressroom/stats](http://www.imdb.com/pressroom/stats)
- [7] Indriani, F., & Budiman, I. 2017. K-Modes Clustering untuk Mengetahui Jenis Masakan Daerah yang Populer pada Website Resep Online (Studi Kasus Masakan Banjar di Cookpad.Com). In *Jurnal Teknologi Informasi dan Ilmu Komputer*, 4 (4). 290-296. DOI: 10.25126/jtiik.201744542.
- [8] Han, J., Kamber, K., & Pei, J. 2013. *Data Mining Concept and Techniques (3rd Edition)*. Waltham, USA: Morgan Kauffman.
- [9] Phorasim, P., & Yu, L. 2017. Movies Recommendation Using Collaborative Filtering and K-Means. In *International Journal of Advanced Computer Research*, 55-58.
- [10] Ricci, F., Rokach, L., & Shaphira, B. 2015. *Introduction and Challenges*. In F. Ricci, L. Rokach, & B. Shaphira, *Recommender Systems* (pp. 11-14). Boston: Springer.
- [11] Sharma, N., & Gaud, N. 2013. K-Modes Algorithm for Categorical Data. In *International Journal of Computer Application*, 127(17),1-6.