

# Penerapan *Collaborative Filtering*, PCA dan *K-Means* dalam Pembangunan Sistem Rekomendasi *Ongoing* dan *Upcoming* Film Animasi Jepang

Mu'tashim Billah<sup>1</sup>, Muhammad Aidil Zartesyah<sup>2</sup>, Desy Komalasari, S Komp, M Kom.<sup>3</sup>,  
Ilmu Komputer

Jl. TB Simatupang, RT.3/RW.3, Cilandak Timur, Jakarta Selatan, DKI Jakarta 12560, Indonesia  
e-mail: m.billah@students.esqbs.ac.id, m.aidil.z@students.esqbs.ac.id, desy.komalasari@esqbs.ac.id

**Abstrak.** Penelitian ini bertujuan untuk mengembangkan sistem rekomendasi *anime* menggunakan kombinasi dari *Collaborative Filtering*, PCA, dan *K-Means*. Metode PCA diterapkan pada data agar waktu yang dibutuhkan saat proses *clustering* lebih cepat. Rata-rata kompleksitas waktu yang dihasilkan adalah 2.99602. Proses *clustering* akan menentukan karakteristik seorang *user* berdasarkan tingkat kemiripan dengan *user* lainnya. Didapatkan hasil  $k$  terbaik dari pengujian *Silhouette Coefficient* dan pengujian *Elbow* terletak pada  $k = 3$ . Rekomendasi yang dihasilkan kemudian dihitung dengan *Mean Reciprocal Rank* (MRR) untuk mengetahui tingkat ketepatan sebuah rekomendasi. Rata-rata MRR yang dihasilkan adalah 0.5619. Dari nilai tersebut dapat dikatakan rekomendasi yang dihasilkan cukup tepat.

**Kata Kunci:** *Collaborative Filtering*, *K-Means Clustering*, PCA, Sistem Rekomendasi

## 1 Pendahuluan

Teknologi informasi dan telekomunikasi semakin berkembang dan mengalami peningkatan yang sangat tinggi, dalam hal ini dapat diketahui banyak sekali kegiatan manusia yang membutuhkan teknologi informasi dan komunikasi untuk saat ini, tidak terkecuali dalam bidang musik maupun film. Film merupakan audio visual yang memiliki banyak *genre*, seperti *genre* komedi, drama, *horor*, *action*, dan masih banyak lagi.

Negara-negara maju sudah melebarkan sayap industri dalam bidang perfilman baik di negaranya sendiri maupun di mancanegara. Salah satu negara di Asia yang dikenal dengan industri filmnya di bidang animasi, yaitu Jepang. Film Animasi Jepang atau yang dikenal dengan sebutan *anime* tersebut memiliki banyak kategori diantaranya *genre*, jumlah episode, media tayang (layar lebar, TV, OVA, dan ONA), sumber cerita (Manga, Light Novel, Original), studio, musim rilis (Spring, Summer, Winter, dan Fall) serta komunitas.

Berdasarkan kategori yang telah disebutkan, memunculkan masalah baru bagi penikmat *anime* untuk menemukan *anime* mana yang selanjutnya akan ditonton, khususnya *anime* yang masih berstatus *ongoing* dan *upcoming*. Masalah ini dapat diatasi dengan menyampaikan informasi berupa daftar-daftar *anime* yang menjadi rekomendasi kepada penikmat *anime* tersebut berdasarkan preferensinya sendiri (*user*). Maka dari itu, perlu adanya sebuah sistem yang dapat memberikan rekomendasi daftar *anime* kepada *user*.

Dalam memberikan rekomendasi, sistem rekomendasi perlu mengetahui daftar item mana saja yang menjadi ciri dari *user* tersebut agar dapat mengenali dan memberikan rekomendasi terkait *item* yang disukainya tersebut. Pada penerapannya, sistem rekomendasi dibagi menjadi dua pendekatan antara lain *content-based filtering* dan *collaborative filtering*. Pada penelitian ini, metode yang digunakan adalah *collaborative filtering* yang melibatkan data *user* lain yang memiliki kemiripan dengan *user* yang akan diberikan rekomendasi.

Mengacu kepada penikmat *anime* yang jumlahnya tidak sedikit, maka perlu adanya pengelompokan terlebih dahulu sebelum memberikan rekomendasi agar hasil daftar rekomendasi menjadi lebih akurat. Untuk mengelompokan *user* menjadi digunakan salah satu metode *clustering* yaitu *K-Means Clustering*. *User* akan terlebih dahulu dikelompokan berdasarkan daftar item yang disukai sebelum diberikan rekomendasi item. Namun, karena jumlah *anime* juga tidak sedikit dan mengakibatkan fitur yang dihasilkan semakin banyak, maka

pada penelitian ini digunakan metode *Principal Component Analysis* (PCA) guna mengurangi dimensi pada data namun tidak menghilangkan makna dari data tersebut.

## 2 Landasan Teori

Adapun teori-teori yang digunakan dalam penelitian ini adalah sebagai berikut:

### 2.1 Sistem Rekomendasi

Sistem rekomendasi merupakan sebuah sistem atau program yang dapat membuat keputusan bagi pengguna terkait item yang disukai dan diinginkannya [1]. Sistem rekomendasi dapat digambarkan sebagai daftar kebutuhan atau keinginan pengguna berdasarkan karakteristik dari pengguna itu sendiri [1]. Sistem rekomendasi memiliki output berupa daftar item yang diurutkan berdasarkan rating kemiripan tertinggi hingga terendah.

### 2.2 Collaborative Filtering

*Collaborative Filtering* merupakan suatu metode atau cara menyaring dan mengevaluasi suatu item berdasarkan opini user lain [2]. Collaborative Filtering dibagi menjadi 2 metode, yaitu *Item-Based Collaborative Filtering* (ICF) dan *User-Based Collaborative Filtering* (UCF). Pada metode ICF, sistem memberikan rekomendasi item yang mirip dengan profil pengguna [3]. Sedangkan UCF memberikan rekomendasi item kepada pengguna berdasarkan pendapat pengguna yang terdekat atau mirip yang memiliki pemikiran yang sama terkait item yang disukai [4]. Visualisasi perbedaan UCF dan ICF dalam memberikan rekomendasi kepada *user* dapat dilihat pada gambar 2 dan rumus UCF dalam memberikan rekomendasi *item* dapat dilihat dibawah ini.

$$S(x, y) = \sum_{z=1}^n R(z, y) \times Si(x, z) \quad (1)$$

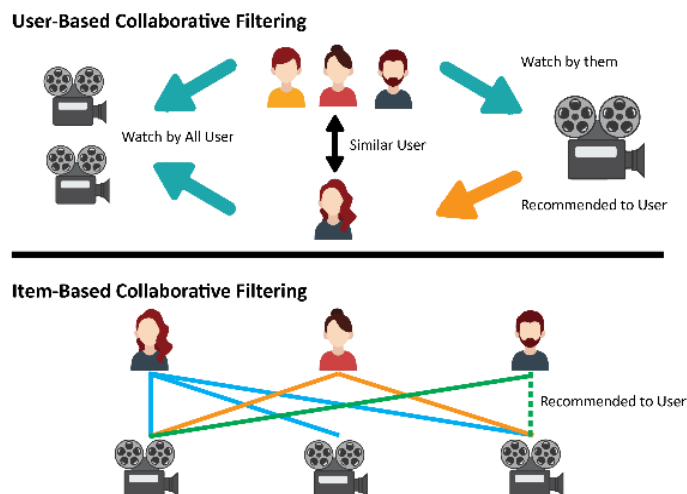
$$= \sum_{z=1}^n |Si(x, z)| n_y = 1$$

Dimana:

$S(x, y)$  = Nilai rekomendasi *user* ( $x$ ) terhadap *item* ( $y$ )

$R(z, y)$  = Besar *rating* atau tingkat kesukaan *user* ( $z$ ) terhadap *item* ( $y$ )

$Si(x, z)$  = Nilai *similarity* antara *user* ( $x$ ) dengan *user* ( $z$ )



**Gambar. 2.1.** Visualisasi User-Based dan Item-Based Collaborative Filtering

### 2.3 Principal Component Analysis

*Principal Component Analysis* (PCA) merupakan metode yang mampu melakukan reduksi dimensi pada suatu data, namun tetap menggambarkan dan mempertahankan pola dan tren data tersebut [5].

### 2.4 K-Means Clustering

*K-Means Clustering* merupakan metode yang mengoptimalkan data yang sensitif dan berdekatan terhadap pemilihan awal dan posisi tengah dari kumpulan data tersebut [6]. *K-Means* sering digunakan dalam proses pengelompokan data untuk menentukan label atau cirinya.

### 2.5 Silhouette Coefficient

*Silhouette Coefficient* dapat digunakan sebagai metode ukur dari hasil clustering. Metode ini juga dapat memilih jumlah  $k$  terbaik dalam model *K-Means Clustering*, sehingga model yang dibuat berdasarkan nilai *Silhouette Coefficient* tertinggi dapat menggambarkan struktur data yang telah dikelompokkan [7]. Adapun rumus untuk menghitung nilai *Silhouette Coefficient* adalah sebagai berikut:

$$s(a) = y - x \times \max(x, y) \quad (2)$$

Dimana:

$s(a)$  = Nilai *Silhouette Coefficient*

$x$  = Rata-rata nilai *intra cluster distance*

$y$  = Rata-rata nilai *inter cluster distance*

### 2.6 Elbow

*Elbow Method* berperan penting dalam proses pengujian jumlah  $k$  pada model *clustering*. Algoritma *K-Means Clustering* memiliki kelemahan saat menentukan jumlah  $k$  terbaik dari  $n$  percobaan [6]. Oleh karena itu, *Elbow Method* dapat mengatasi masalah tersebut sehingga model yang dihasilkan *K-Means* menjadi lebih baik. Berikut adalah rumus dari *Elbow Method*:

$$d = \sum (x_i - t_x) + (y_i - t_y) \quad (3)$$

Dimana:

$d$  = Nilai *Distortion*

$x_i$  = *Cluster (x)* pada perulangan ke ( $i$ )

$t_x$  = Titik tengah *cluster (x)*

$y_i$  = *Cluster (y)* pada perulangan ke ( $i$ )

$t_y$  = Titik tengah *cluster (y)*

### 2.7 Mean Reciprocal Rank

Saat proses rekomendasi item telah selesai diproses, alat ukur untuk menentukan tingkat akurasi dalam pemberian rekomendasi dapat dilakukan melalui penghitungan *Mean Reciprocal Rank* (MRR). MRR membandingkan dua atau lebih rekomendasi, dimana rekomendasi relevan yang pertama dibandingkan dengan rekomendasi selanjutnya hingga mendapatkan nilai *rank* untuk masing-masing item di rekomendasi yang berbeda [8]. Untuk mengukur nilai MRR pada sebanyak  $n$  rekomendasi dapat menggunakan rumus dibawah ini:

$$MRR = \frac{1}{|n|} \sum 1R|n|i = 1 \quad (4)$$

Dimana:

$MRR$  = Nilai *Mean Reciprocal Rank*

$n$  = Frekuensi data

$R$  = Ranking *item* dalam data

## 2.8 Penelitian Terdahulu

Penelitian ini mengacu kepada penelitian-penelitian sebelumnya, khususnya pada penelitian yang dilakukan oleh Ichwanto Hadi, Leo Willyanto Santoso, dan Alvin Nathaniel Tjondrowiguno yang berjudul “Sistem Rekomendasi Film menggunakan *User-based Collaborative Filtering* dan *K-modes Clustering*” yang memiliki kompleksitas waktu yang cukup besar dan memiliki nilai *Mean Reciprocal Rank* yang rendah. Penelitian ini menerapkan metode tambahan yaitu *Principal Component Analysis* (PCA) dengan tujuan agar proses *clustering* yang dilakukan dapat menjadi lebih cepat dan efisien.

## 3 Analisis dan Desain

Dari himpunan data yang sudah tersedia, perlu adanya proses analisis terlebih dahulu karena ada data-data yang tidak sesuai untuk dimasukkan kedalam model. Selain itu sistem juga memiliki beberapa proses yang terpisah, sehingga desain sistem yang dihasilkan menjadi beberapa bagian. Berikut adalah hasil analisis dan gambaran desain alur sistem yang diterapkan pada penelitian ini.

### 3.1 Analisis Data

Himpunan data dibagi menjadi dua himpunan antara lain himpunan data *anime* dan himpunan data *user rating*. Contoh beberapa data pada himpunan data *anime* dapat dilihat pada gambar 3.1 dan contoh himpunan data *user rating* dapat dilihat pada gambar 3.2.

	anime_id	name	genre	type	episodes	rating	members
0	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.37	200630
1	5114	Fullmetal Alchemist: Brotherhood	Action, Adventure, Drama, Fantasy, Magic, Mili...	TV	64	9.26	793665
2	28977	Gintama <sup>a</sup>	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9.25	114262

Shape: (12294, 7)

**Gambar. 3.1.** Contoh himpunan data *anime* dengan atribut *anime\_id*, *name*, *genre*, *type*, *episodes*, *rating*, dan *members*

	user_id	anime_id	rating
0	1	20	-1
1	1	24	-1
2	1	79	-1

Shape: (7813737, 3)

**Gambar. 3.2.** Contoh himpunan data *user rating* dengan atribut *user\_id*, *anime\_id*, dan *rating*

Pada himpunan data *user rating* terdapat data pada atribut *rating* yang tidak wajar, yaitu -1. Pada kasus ini, peneliti melakukan pengambilan rata-rata *rating user* terlebih dahulu kemudian mengambil kesimpulan melalui rata-rata *rating* yang dimiliki setiap *user* dari tiap-tiap *anime\_id* yang telah ia beri nilai. Hasil dari pengambilan rata-rata *rating* dapat dilihat pada gambar 3.3.

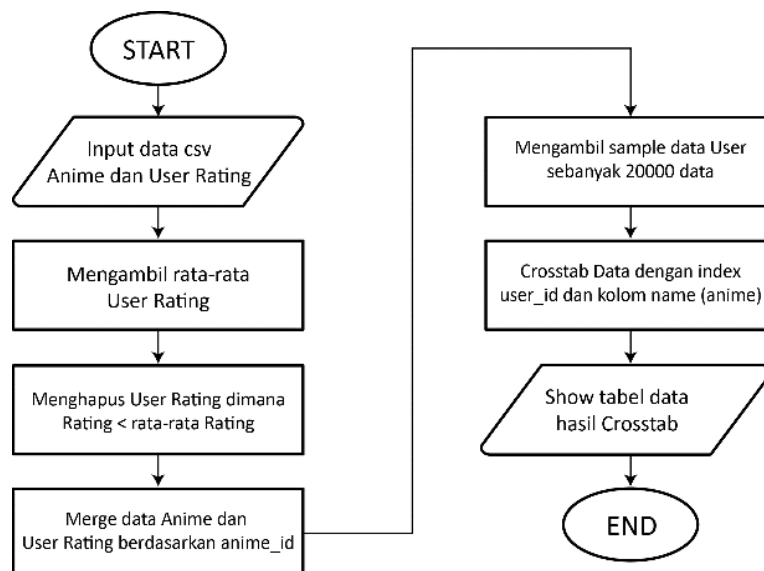
<code>user_id</code>	<code>mean_rating</code>
0	-0.712418
1	2.666667
2	7.382979

**Gambar. 3.3.** Hasil pengambilan rata-rata dari rating yang diberikan oleh masing-masing user

Setelah rata-rata rating setiap *user* diperoleh, maka selanjutnya melakukan penghapusan data *user rating* dimana atribut rating-nya lebih kecil daripada rata-rata rating. Hasil analisis ini menghasilkan *definition of like* dimana *anime* yang dinilai oleh *user* lebih kecil dibandingkan rata-rata *rating user* dianggap bahwa *user* tidak menyukainya. Sebaliknya, anime yang lebih tinggi rating-nya dibandingkan rata-rata *rating* dianggap bahwa *user* menyukai *anime* tersebut.

### 3.2 Desain Alur Sistem

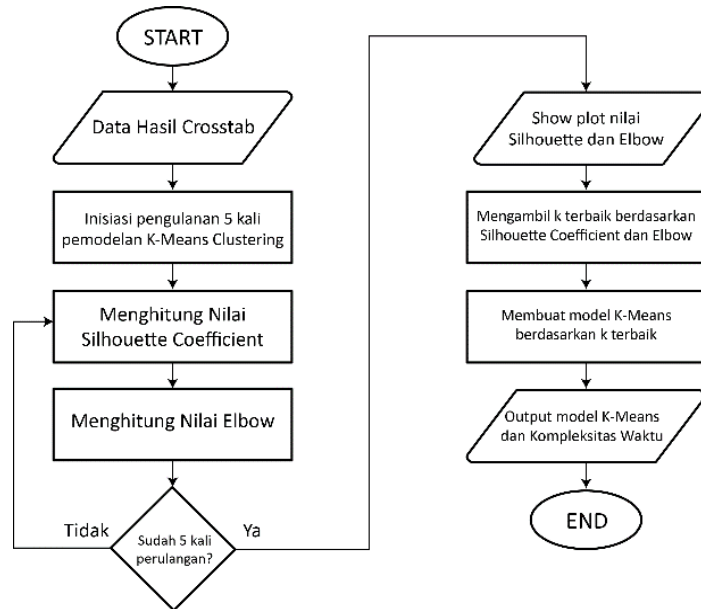
Pada pemrosesan data *anime* dan *user rating* yang akan menghasilkan himpunan rekomendasi kepada *user*, maka perlu dilakukan proses persiapan terlebih dahulu agar data yang digunakan menjadi data yang baik. Proses tersebut adalah menghilangkan nilai rating yang tidak wajar di himpunan data *user rating*. Hal ini juga menjadi penentu apakah *user* benar-benar menyukai *anime* tersebut atau tidak. Proses ini akan terlebih dahulu mendapatkan rata-rata rating tiap *user*, kemudian menghapus data *user rating* dimana *rating user*-nya lebih rendah daripada rata-rata *rating*-nya. Proses ini disebut *Definiton of Like* dimana sistem secara otomatis akan menilai apakah item tersebut benar-benar disukai oleh *user* atau tidak. Karena pada dasarnya ketika suatu item memiliki *rating* kecil, belum tentu *user* merekomendasikan item tersebut kepada orang lain. Selain itu, data juga perlu di migrasi berdasarkan atribut *anime id* dan data juga perlu dilakukan *crosstab* sehingga kita dapat melihat rekomendasi *anime* berdasarkan *anime* yang telah disukai oleh *user* lainnya. Alur proses pertama dapat dilihat pada gambar 3.4.



**Gambar. 3.4.** Desain alur kerja sistem dalam memberikan rekomendasi (Tahap 1)

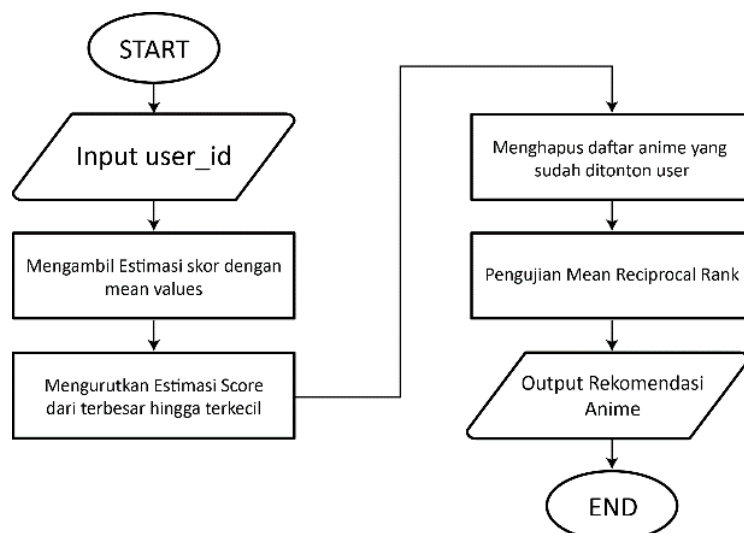
Setelah dilakukan proses persiapan, maka data siap melakukan proses *clustering*. Namun, sebelum proses *clustering* dilakukan, data akan terlebih dahulu melakukan reduksi dimensi dengan menggunakan metode *Principal Component Analysis*. Hal ini dilakukan agar kompleksitas waktu yang dibutuhkan dalam memproses

*clustering* data lebih cepat namun tidak menghilangkan makna yang sebenarnya dari data tersebut. Pada penelitian ini proses *clustering* terlebih dahulu dilakukan penilaian untuk mencari *k* terbaik agar hasil pengelompokan dapat menjadi lebih akurat. Pengujian dilakukan dengan membandingkan hasil nilai *Silhouette Coefficient* dan *Elbow*. Alur kerja kedua dalam penelitian ini dapat dilihat pada gambar 3.5.



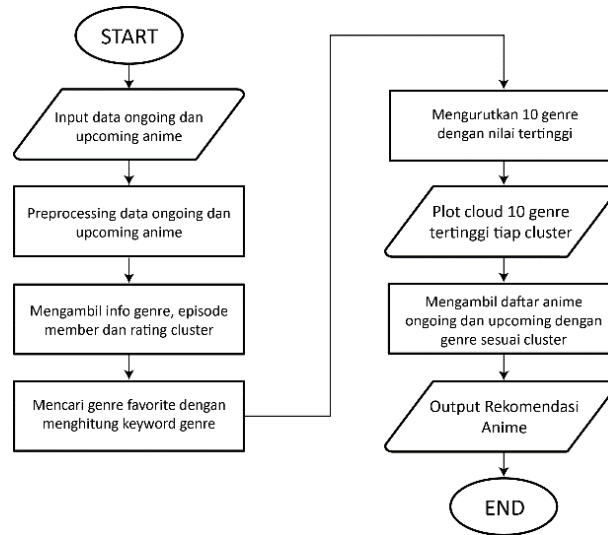
**Gambar. 3.5.** Desain alur kerja sistem dalam memberikan rekomendasi (Tahap 2)

Proses *clustering* akan menentukan karakteristik seorang *user* berdasarkan tingkat kemiripan dengan *user* lainnya. Sehingga proses selanjutnya adalah memberikan rekomendasi berdasarkan kemiripan *user* pada *cluster*-nya. *User* akan diberikan 15 rekomendasi *anime* dengan *mean values* tertinggi berdasarkan *anime* yang belum pernah ditonton sebelumnya. Pada proses ini juga akan mengukur seberapa besar tingkat keakuratan hasil rekomendasi dengan menggunakan pengujian *Mean Reciprocal Rank*. Alur kerja sistem tersebut dapat dilihat pada gambar 3.6.



**Gambar. 3.6.** Desain alur kerja sistem dalam memberikan rekomendasi (Tahap 3)

Setelah sistem dapat memproses rekomendasi *anime* berdasarkan riwayat *anime* yang belum pernah ditonton sebelumnya. Maka pada penelitian ini, sistem akan dapat memberikan rekomendasi *anime* yang sedang tayang (*ongoing*) dan yang akan tayang di musim selanjutnya (*upcoming*). Rekomendasi tersebut berdasarkan 10 *genre* yang paling diminati oleh *user* dan *member*. Alur proses tersebut dapat dilihat pada gambar 3.7.



**Gambar. 3.7.** Desain alur kerja sistem dalam memberikan rekomendasi (Tahap 4)

## 4 Hasil Pengujian

Penelitian ini diuji dengan pengujian *Silhouette Coefficient* dan *Elbow* untuk menemukan jumlah *cluster* terbaik, serta pengujian *Mean Reciprocal Rank* untuk menguji ketepatan hasil rekomendasi. Pengujian dilakukan pada laptop dengan rincian sistem Windows 10, CPU Intel Core i5-8250 CPU @1.60 GHz (8 CPUs) dan pada kondisi malam hari.

### 4.1 Pengujian Kompleksitas Waktu

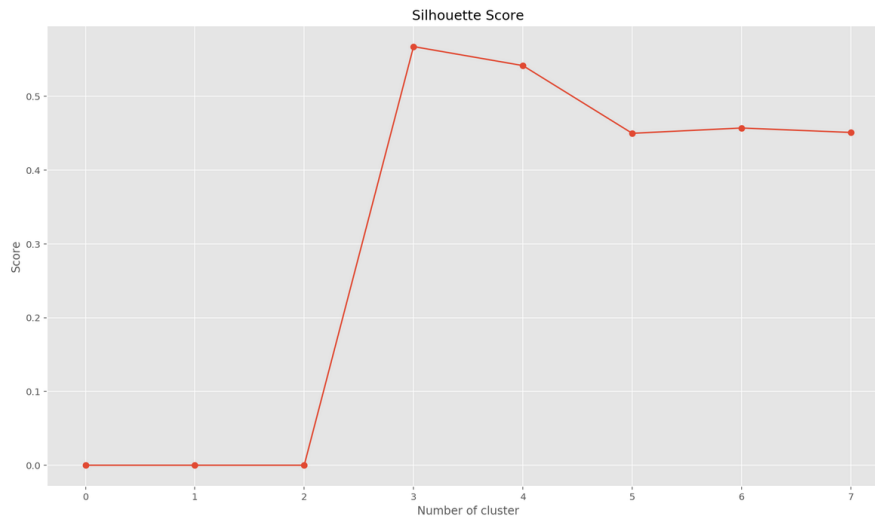
Pengujian kompleksitas waktu yang dihasilkan dalam 5 kali iterasi dalam melakukan *Clustering* dengan metode *K-Means* yang sebelumnya dilakukan reduksi dimensi menggunakan *Principal Component Analysis*. Hasil yang didapatkan adalah kompleksitas waktu paling cepat terletak pada  $k = 5$  dengan kompleksitas waktu sebesar 2.92932, sedangkan kompleksitas tertinggi terletak pada  $k = 3$  yaitu sebesar 3.12003. Sehingga dihasilkan rata-rata dari kelima perulangan tersebut sebesar 2.99602. Hasil ini dapat dikatakan bahwa kompleksitas waktu yang dibutuhkan tergolong cepat dan singkat. Rincian hasil pengujian kompleksitas waktu dapat dilihat pada tabel 4.1.

**Tabel 4.1.** Hasil pengujian Kompleksitas Waktu

Jumlah Cluster	Waktu Proses
3	3.12003
4	2.9705
5	2.92932
6	3.01981
7	2.94044
Rata-rata	2.99602

#### 4.2 Pengujian *Silhouette Coefficient*

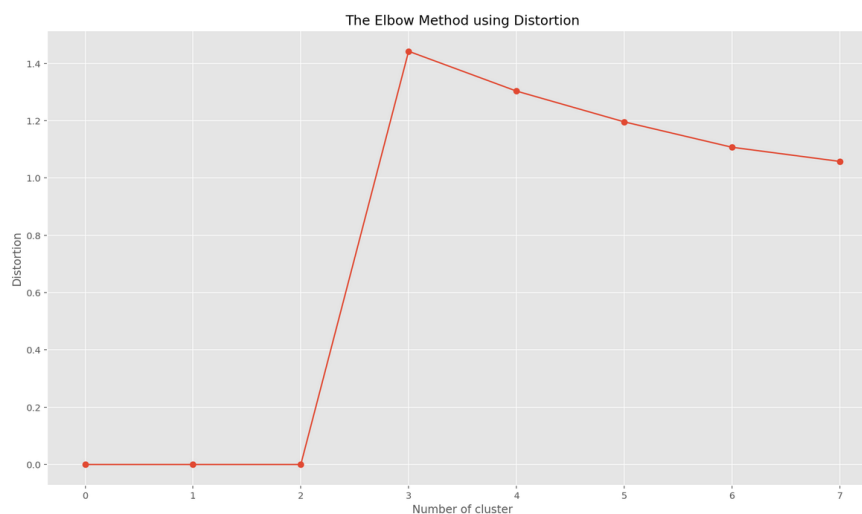
Pengujian *Silhouette Coefficient* dilakukan sebanyak 5 kali, mulai dari  $k = 3$  hingga 7. Hasil pengujian tersebut mendapatkan *cluster* terbaik berada pada  $k = 3$ . Hasil ini didapatkan karena semakin tinggi hasil *Silhouette Coefficient* maka data-data yang berada pada *cluster-cluster* tersebut juga sudah sesuai. Grafik perbandingan nilai *Silhouette Coefficient* tiap  $k$ -nya dapat dilihat pada gambar 4.1.



**Gambar. 4.1.** Hasil Pengujian *Silhouette Coefficient*

#### 4.3 Pengujian *Elbow*

Metode kedua dalam menentukan jumlah  $k$  terbaik untuk melakukan proses clustering adalah dengan menguji nilai *Elbow*. Dari hasil yang sudah diperoleh, didapatkan bahwa jumlah *cluster* terbaik terletak pada  $k = 3$ . Hasil ini diperoleh berdasarkan grafik pada gambar 4.2 yang menunjukkan penurunan yang landai dan signifikan pada jumlah *cluster* setelah 4.



**Gambar. 4.2.** Hasil Pengujian *Elbow Method* menggunakan *Distortion*



#### 4.4 Pengujian *Mean Reciprocal Rank*

Pengujian *Mean Reciprocal Rank* melibatkan seluruh data himpunan 15 tertinggi dari rekomendasi tiap-tiap *user*. Tabel dibawah ini merupakan rincian dari hasil *Mean Reciprocal Rank* tiap-tiap *user*, kemudian menghasilkan rata-rata yaitu sebesar 0.5619. Berdasarkan kaidah dari metode pengujian *Mean Reciprocal Rank*, apabila hasilnya berjarak dari 1 hingga 0.5 dapat dikatakan rekomendasi sudah tepat. Namun, apabila nilainya berada dijarak 0.5 sampai dengan 0, maka rekomendasi yang dihasilkan dapat dikatakan kurang tepat. Sehingga, dilihat dari hasil yang diperoleh dari penelitian ini, hasil rekomendasi *anime* kepada *user* dapat dikatakan cukup tepat. Pada tabel 4.2 dapat dilihat beberapa rincian pengujian *Mean Reciprocal Rank* dari beberapa *user*.

**Tabel 4.2.** Hasil pengujian *Mean Reciprocal Rank*

userId	<i>Mean Reciprocal Rank</i>
1	0.2
2	1
3	0.00
4	0.00
5	0.3333333333333333
10	0.2
50	0.00
100	0.2
500	0.06666666666666667
1000	0.00
5000	0.00
10000	0.06666666666666667
Rata-rata keseluruhan	0.5619

## 5 Penutup

### 5.1 Simpulan

Penelitian ini telah menghasilkan sebuah sistem rekomendasi *anime* dengan menggunakan algoritma *K-Means Clustering* dan *User-Based Collaborative Filtering* yang telah dikembangkan lagi dengan menggunakan metode *Principal Component Analysis*. Kompleksitas waktu yang dihasilkan setelah dilakukan reduksi menggunakan *Principal Component Analysis* yaitu sebesar 2.99602. Tingkat akurasi pada hasil rekomendasi telah dihitung dengan menggunakan nilai MMR. Nilai rata-rata MMR dari tersebut adalah sebesar 0.5619 yang disimpulkan bahwa rekomendasi yang dihasilkan cukup tepat.

### 5.2 Saran Pengembangan Lanjutan

Penelitian ini dapat dikembangkan lebih lanjut agar nilai yang dihasilkan dapat lebih baik dan lebih akurat lagi. Peneliti memberikan saran agar menerapkan beberapa *Preprocessing* tambahan dan menerapkan *GridSearchCV* agar dapat menemukan jumlah *cluster* yang lebih baik lagi karena menggunakan *hyper parameter* yang lebih lengkap dan rinci.

## Referensi

- [1] B. T. W. Utomo and A. W. Anggriawan, "Sistem Rekomendasi Paket Wisata Se-Malang Raya Menggunakan Metode *Hybrid Content Based Dan Collaborative*," J. Ilm. Teknol. Inf. Asia, vol. 9, no. 1, pp. 6–13, 2015.
- [2] M. Nilashi, K. Bagherifard, O. Ibrahim, H. Alizadeh, L. A. Nojeem, and N. Roozegar, "Collaborative filtering recommender systems," Res. J. Appl. Sci. Eng. Technol., vol. 5, no. 16, pp. 4168–4182,
- [3] F. XUE, X. HE, X. WANG, J. XU, K. LIU, and R. HONG, "Deep item-based collaborative filtering for top-n recommendation," arXiv, vol. 37, no. 3, 2018.
- [4] Z. Tan and L. He, "An Efficient Similarity Measure for User-Based Collaborative Filtering Recommender Systems Inspired by the Physical Resonance Principle," IEEE Access, vol. 5, pp. 27211–27228, 2017, doi: 10.1109/ACCESS.2017.2778424.
- [5] J. Lever, M. Krzywinski, and N. Altman, "Points of Significance: Principal component analysis," Nat. Methods, vol. 14, no. 7, pp. 641–642, 2017, doi: 10.1038/nmeth.4346.
- [6] M. A. Syakur, B. K. Khotimah, E. M. S. Rochman, and B. D. Satoto, "Integration K-Means Clustering Method and Elbow Method for Identification of the Best Customer Profile Cluster," IOP Conf. Ser. Mater. Sci. Eng., vol. 336, no. 1, 2018, doi: 10.1088/1757-899X/336/1/012017.
- [7] R. Lletí, M. C. Ortiz, L. A. Sarabia, and M. S. Sánchez, "Selecting variables for k-means cluster analysis by using a genetic algorithm that optimises the silhouettes," Anal. Chim. Acta, vol. 515, no. 1, pp. 87–100, 2004, doi: 10.1016/j.aca.2003.12.020.
- [8] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, N. Oliver, and A. Hanjalic, "CLiMF," p. 139, 2012, doi: 10.1145/2365952.2365981.