

# Pengembangan Model k-NN Collaborative Filtering pada Pendekatan User-Based untuk Sistem Rekomendasi

## Improvement of KNN Collaborative Filtering Model in User-based Approach on Anime Recommendation System

<sup>1</sup>Rezky Putratama Raharjo, <sup>2</sup>Vynska Amalia Permadi\*, <sup>3</sup>Nur Heri Cahyana

<sup>1,2,3</sup>Informatika, Fakultas Teknik Industri, Universitas Pembangunan Nasional "Veteran" Yogyakarta

\*e-mail: [vynspermadi@upnyk.ac.id](mailto:vynspermadi@upnyk.ac.id)

(received: 10 Desember 2022, revised: 18 Maret 2023, accepted: 27 April 2023)

### Abstrak

Penelitian ini bertujuan untuk mengatasi masalah pemberian rekomendasi yang sesuai dengan preferensi *user*. Untuk pengujian model, digunakan *dataset* MyAnimeList yang dapat diakses dari situs Kaggle. Hasil dari penelitian ini adalah rancangan sistem rekomendasi berdasarkan preferensi *user* lainnya (*user-based model*). Metode yang diusulkan menggunakan model *collaborative filtering* dengan menggunakan algoritma KNN dan penambahan bobot pada *dataset*. Data yang digunakan pada penelitian ini berjumlah 193.272 data yang berisi *rating user* terhadap *anime*, dan terdiri dari atribut: *username*, *anime\_id*, *my\_score*, dan *my\_status*. Sebagai pengembangan model KNN *collaborative filtering*, dilakukan pembobotan pada nilai *rating* berdasarkan status dari *user*. Penentuan nilai pembobotan didasarkan oleh kuisioner yang telah diisi oleh 105 responden. Selain pembobotan, nilai *my\_score* dan *my\_status* akan digabungkan dan dinormalisasi menggunakan *MinMaxNormalization*. Berbagai nilai parameter *k* digunakan pada algoritma KNN di penelitian ini, yaitu: 3, 5, 9, 15, 23, 33, dan 45. Variasi parameter digunakan untuk mencari nilai *k* terbaik untuk diimplementasikan pada KNN yang menggunakan matriks *pearson similarity* dalam mencari nilai similaritas antar *user*. Hasil pengujian menunjukkan bahwa nilai *Mean Absolute Error* terbaik adalah 0,14726 dan nilai *Root Mean Square Error* terbaik sebesar 0,19855, pada parameter *k* = 5. Hasil dari model modifikasi ini juga menunjukkan bahwa KNN *collaborative filtering* yang ditambahkan pembobotan dapat memprediksi *rating* dengan nilai *error* yang stabil dan relatif rendah pada semua nilai *k*.

**Kata kunci:** Collaborative Filtering, Sistem rekomendasi, User-Based.

### Abstract

This research aims to resolve the challenge of finding the list of recommendations that correspond to user preferences. The MyAnimeList dataset is utilized for model evaluation, accessible via Kaggle website. The outcome of this study is the development of a recommendation system based on the preferences of other users (*user-based model*). The suggested solution employs a collaborative filtering model based on the KNN algorithm and weighted attribute. The dataset consisted of 193,272 user ratings on anime, with the following attributes: *username*, *anime\_id*, *my\_score*, and *my\_status*. As an extension of the KNN collaborative filtering paradigm, the rating value is weighted based on the user's status. The determination of the weight is based on the responses of 105 respondents to a questionnaire. *my\_score* and *my\_status* values will be combined and adjusted using *MinMaxNormalization* in addition to being weighted. This work implemented the KNN algorithm with the following *k* parameter values: 3, 5, 9, 15, 23, 33, and 45. Variations in parameters are utilized to determine the optimal *k* value to employ in KNN, which uses the Pearson similarity matrix to calculate user similarity values. The model evaluation indicate that the optimal Mean Absolute Error and Root Mean Square Error values at parameter *k* = 5 are 0.14726 and 0.19855, respectively. This improved model's findings further demonstrate that KNN collaborative filtering with an additional weighted parameter can predict ratings with stable and generally low error values for all *k* values.

**Keywords:** Collaborative Filtering, Sistem rekomendasi, User-Based.

## 1 Pendahuluan

Dengan perkembangan teknologi, bermacam-macam bentuk hiburan dalam bentuk film, musik maupun hiburan lainnya mulai bervariasi, bahkan dapat diakses bebas melalui berbagai situs penyedia layanan hiburan daring. Dilihat dari sisi pengguna, ekspektasi *user* yang menggunakan layanan hiburan tersebut adalah dapat diberikannya rekomendasi film atau hiburan yang sesuai dengan preferensi *user*. Berbagai algoritma dan metode telah dirancang untuk menyelesaikan *case* tersebut, salah satunya adalah dengan menggunakan sistem pendukung keputusan. Namun, metode tersebut memiliki kelemahan, yaitu hasil dari sistem tersebut tidak selalu sesuai dengan preferensi *user*. Metode lain yang sering digunakan dikenal sebagai sistem rekomendasi. Sistem rekomendasi dapat menganalisis data dari suatu item maupun *user* yang kemudian akan memberikan suatu rekomendasi terhadap *user* sesuai dengan *item* yang pernah berinteraksi dengan *user*, maupun sesuai dengan preferensi *user*. Sistem rekomendasi ini telah digunakan pada berbagai tempat diantaranya pada toko daring “Amazon”, lalu layanan streaming lagu “Spotify”, hingga layanan penyedia film “Netflix”.

Sistem rekomendasi adalah alat perangkat lunak maupun teknik untuk mencari rekomendasi suatu item terhadap suatu pengguna [1]. Sistem rekomendasi telah banyak digunakan di berbagai bidang, di antara lain memberikan rekomendasi daftar putar layanan video dan musik, rekomendasi produk pada toko *online*, hingga rekomendasi konten pada *platform* media sosial. Terdapat beberapa metode yang sering digunakan untuk membuat sistem rekomendasi, yaitu *content-based filtering*, *collaborative filtering* dan *hybrid*. *Content-based filtering* merekomendasikan berdasarkan kemiripan antar item, sedangkan *collaborative filtering* memberikan rekomendasi berdasarkan tingkat kemiripan preferensi antar pengguna.

Pada *content-based filtering* terdapat kelemahan yang terjadi menyebabkan hasil rekomendasi yang diberikan kurang beragam atau *overspecialization* karena rekomendasinya hanya bergantung pada kemiripan fitur dari item. *Collaborative filtering* adalah solusi yang efektif untuk mengatasi masalah dalam informasi yang berlebihan [2], [3]. Tahapan dalam *collaborative filtering* yakni mencari nilai kemiripan antar user yang nantinya akan digunakan sebagai patokan untuk memprediksi rating dari *user* terhadap suatu item. Untuk mencari nilai similaritas antar *user* terdapat banyak metode yang digunakan di antara lain adalah *mean-squared difference*, *pearson similarity*, *cosine similarity*, *adjusted cosine similarity*, dan *spearman correlation* [4]. Menurut Al-Bakri & Hashim [5], metode *pearson similarity* adalah metode yang paling tepat digunakan untuk menghitung jarak terdekat antara *user* berdasarkan *rating*.

Perancangan sistem rekomendasi dengan *collaborative filtering* pada *anime* pernah dilakukan [6] dengan menggunakan algoritma *Alternating Least Square* (ALS). Dengan model tersebut didapatkan hasil yakni nilai *Root Mean Square Error* sebesar 2,537. Penelitian lainnya juga dilakukan [7] dengan menggunakan gambar atau poster dari *anime* sebagai parameter tambahan untuk sistem rekomendasi. Gambar akan diekstraksi fiturnya dengan menggunakan LASSO dan *rating* dari *user* diprediksi menggunakan ALS. Dari kedua fitur dari gambar dan *rating* kemudian digabungkan sehingga dapat memprediksi *rating* bagi *user*. Hasil yang didapatkan adalah nilai RMSE sebesar 1,149.

Dari kedua penelitian di atas, dapat disimpulkan bahwa dengan menambahkan fitur-fitur tambahan untuk ditambahkan pada sistem rekomendasi dapat meningkatkan kualitas sistem rekomendasi dengan baik sehingga pada penelitian ini digunakan metode *collaborative filtering* dengan penambahan pembobotan untuk membangun sistem rekomendasi. Alasan penggunaan metode ini dikarenakan pada penelitian yang telah dilakukan sebelumnya pernah dilakukan penambahan nilai preferensi *user* sebagai parameter [8], [9]. Sehingga pada penelitian ini akan membuat sistem rekomendasi *anime* dengan penambahan bobot pada *collaborative filtering* untuk menghasilkan prediksi *rating* terhadap suatu *anime* dengan mempertimbangkan tingkat similaritas antar pengguna dan pembobotan status *anime* yang ditonton oleh *user*.

## 2 Tinjauan Literatur

Salah satu metode sistem rekomendasi yang sering digunakan dalam perumusan rekomendasi adalah *collaborative filtering* yang merekomendasikan suatu *item* berdasarkan perilaku pengguna. Metode *collaborative filtering* juga dilakukan pada penelitian oleh Girsang dkk. [6] dengan objek *anime*. Penelitian pertama – tama dilakukan dengan normalisasi rata – rata dari *rating anime*, membuat matrik similaritas dari *user*, dan kemudian merekomendasikan *anime* teratas. Untuk

<http://sistemasi.ftik.unisi.ac.id>

membuat matriks similaritas, peneliti menggunakan *tools* SimRank karena efektif dalam mengurangi masalah *sparsity* pada data. Data yang digunakan adalah data dari “Kaggle” yang bernama *anime recommendation database*. Data tersebut berisi relasi antara 73.516 user dan 12.294 *anime*. Dalam penelitian, model yang digunakan adalah model *Alternating Least Square* (ALS) untuk *training* data dan evaluasi menggunakan nilai RMSE. Hasil yang didapatkan adalah nilai RMSE sebesar 2,537 pada penelitian ini.

Metode KNN juga digunakan pada penelitian yang dilakukan oleh Dharma dkk. [10] untuk sistem rekomendasi pada konten artikel berita. Penelitian ini menggabungkan dari algoritma KNN dengan SlopeOne untuk menghasilkan prediksi rating untuk rekomendasi. Dari hasil penelitian menunjukkan bahwa algoritma gabungan KNN dan SO mendapatkan nilai RMSE sebesar 0,5894, nilai RMSE KNN sebesar 0,6533 dan nilai RMSE SO sebesar 0,6218. Penelitian yang dilakukan oleh Gusti dkk. [11] juga menggunakan metode KNN untuk membuat sistem rekomendasi pemilihan mobil. Hasil yang diperoleh pada penelitian tersebut adalah nilai akurasi terbaik ketika nilai  $K = 10$  yakni sebesar 95,15%.

Untuk pembobotan pada dataset dilakukan pada penelitian Bourkoku dan Achbarou [12] yakni menggunakan metode *collaborative filtering* dan model *learning style*. Penelitian ini menggunakan history dari akses *user* sebagai *input learning* dan melakukan pembobotan pada *data learning*. Pembobotan dilakukan dengan menggabungkan nilai rating eksplisit, nilai implisit dan nilai dimensi sosial. Setelah pembobotan dilakukan maka nilai similaritas dihitung menggunakan *pearson similarity* yang akan digunakan dalam prediksi rating. Hasil yang didapatkan yakni nilai MAE berkisar diantara 0,6 hingga 0,8.

Berdasarkan penelitian sebelumnya [6], [10]–[12], penelitian ini memiliki kesamaan dalam hal tema penelitian yakni sistem rekomendasi dengan menggunakan *collaborative filtering*. Secara lebih rinci, terdapat kesamaan objek penelitian, yakni *anime*, dengan penelitian yang dilakukan oleh Girsang dkk. [6]. Namun, pada penelitian tersebut model *collaborative filtering* yang dibangun menggunakan algoritma ALS. Perbedaan algoritma pada model *collaborative filtering* yang dibangun juga ditemui pada penelitian [10] dimana pada penelitian tersebut digunakan algoritma KNN dan SlopeOne sedangkan pada penelitian ini model *collaborative filtering* yang dibangun menggunakan algoritma KNN serta menambahkan bobot dari status pada nilai *rating*. Selain itu, implementasi pembobotan dataset juga dilakukan pada penelitian [12], dimana pembobotan dilakukan pada data sebelum diproses selanjutnya untuk diprediksi. Perbedaan dengan penelitian ini terdapat pada model yang digunakan untuk pembobotan. Penelitian [12] menggunakan model *learning style Felder-and Silverman* dan menggunakan data implisit sebagai bobot pada data, sedangkan pada penelitian ini menggunakan KNN pada model dan menggunakan status dari *user* sebagai bobot. Terkait evaluasi model, hampir semua penelitian sebelumnya maupun penelitian ini menggunakan nilai MAE ataupun RMSE. Hanya beberapa penelitian yang menggunakan nilai metrik yang berbeda yakni pada penelitian [11] yang menggunakan nilai akurasi.

### 3 Metode Penelitian

Metode penelitian yang digunakan pada penelitian ini menggunakan metode kuantitatif berawal dengan pengumpulan data, pra-proses data, pembuatan model KNN *collaborative filtering*, hingga pengujian model KNN *collaborative filtering*. Alur penelitian dimulai dari pengumpulan data, pra-proses data, dan pembuatan model KNN *collaborative filtering*.

#### 3.1 Pengumpulan Data dan Pembobotan

Pada tahap pengumpulan dilakukan kuisioner untuk pembobotan dan pengumpulan *dataset* yang dikutip melalui situs *Kaggle* dan milik Matěj Račinský [13] dengan berisi data *rating* sebesar 20 juta baris. Parameter yang digunakan terdiri dari *username*, *anime\_id*, *my\_score*, dan *my\_status* untuk digunakan pada KNN *collaborative filtering*. Detail dari tiap parameter dapat dilihat pada Error! Reference source not found..

Tabel 1. Parameter yang akan digunakan

No	Nama Tabel	Deskripsi	Keterangan
1	<i>username</i>	Nama dari <i>user</i> yang memberikan <i>rating</i>	Berupa <i>string</i> kombinasi

<http://sistemasi.ftik.unisi.ac.id>

		kepada suatu <i>anime</i>	angka dan huruf
2	<i>anime_id</i>	Id pengenalan untuk suatu <i>anime</i>	Berupa <i>string</i> yang berisikan angka
3	<i>my_score</i>	<i>Rating</i> yang diberikan oleh <i>user</i> kepada suatu <i>anime</i>	Berupa <i>integer</i> dengan nilai antara 1 hingga 10
4	<i>my_status</i>	Status dari <i>user</i> terhadap suatu <i>anime</i> . Masing-masing nilai mempunyai maksud tersendiri yakni 1: <i>User</i> masih menonton <i>anime</i> tersebut 2: <i>User</i> telah menonton dan menyelesaikan <i>anime</i> tersebut	Berupa <i>integer</i> dengan nilai 1,2,3,4 dan 6

Selanjutnya pengumpulan data untuk penentuan bobot dari *my\_status* dengan melakukan kuisisioner. Kuisisioner dilakukan dengan memberi 5 pernyataan berdasarkan status pada *my\_status* dan diberikan 4 pilihan jawaban dari Tidak Setuju hingga Sangat Setuju. Untuk menghitung skor maksimum pada setiap jawaban, skor maksimum dari masing-masing jawaban akan dikalikan dengan jumlah responden. Rancangan Nilai skor maksimum dapat dilihat pada Tabel 2.

**Tabel 2. Rancangan Skor Maksimum Jawaban**

Jawaban	Skor	Skor Maksimum (Skor * Jumlah Responden)
Sangat Setuju	4	
Setuju	3	
Tidak Setuju	2	
Sangat Tidak Setuju	1	

Untuk pertanyaan yang diberikan pada kuisisioner, dapat dilihat pada Tabel 3 berikut.

**Tabel 3. Pernyataan pada Kuisisioner**

Pernyataan ke -	Pernyataan
1	Anda lebih percaya rekomendasi orang yang sudah selesai menonton <i>anime</i> tersebut.
2	Anda lebih percaya rekomendasi orang yang sudah selesai menonton <i>anime</i> tersebut.
3	Anda lebih percaya rekomendasi orang yang sudah selesai menonton <i>anime</i> tersebut.
4	Anda lebih percaya rekomendasi orang yang sudah selesai menonton <i>anime</i> tersebut.
5	Anda lebih percaya rekomendasi orang yang sudah selesai menonton <i>anime</i> tersebut.

Data yang didapat pada proses pengumpulan data masih perlu dilakukan pra-proses yang akan digunakan pada pembuatan model. Data terlebih dahulu diolah sebelum data tersebut dapat digunakan untuk pembuatan dan pelatihan model. Beberapa pra-proses yang akan dilakukan meliputi : *filtering* data, modifikasi kolom dan *splitting data*. Proses terakhir yang dilakukan adalah membagi keseluruhan data menjadi 2 bagian, 80% untuk data training dan 20% untuk *data testing*.

Proses pertama yakni memodifikasi kolom yang akan digunakan. Data yang telah diambil dan akan digunakan, tidak semua kolom akan digunakan untuk kebutuhan dalam pembuatan model maka perlu dilakukan proses modifikasi kolom. Modifikasi kolom yang akan digunakan meliputi membuang kolom yang tidak digunakan, membuang rating user yang berisi 0, lalu mengambil data *anime* yang bersifat TV Series, dan mengambil 1% dari total data.

Sebagai tahapan pertama dalam pembuatan model KNN *collaborative filtering* adalah membobotkan status dari *user* terhadap *rating*. Status pada dataset adalah status dari user tersebut terhadap suatu *anime*. Nilai status diantara lain adalah 1, 2, 3, 4 dan 6 yang dimana tiap angka tersebut mempunyai arti sendiri dan sudah dijelaskan pada poin 3.3. Dari nilai tersebut, dibobotkan setiap masing – masing status. Dengan pembobotan ini, maka user yang menyelesaikan atau selesai menonton dari *anime* tersebut memiliki dampak lebih tinggi dibandingkan *user* yang berhenti

menonton. Pembobotan akan berdasarkan dari kuisioner yang telah dilakukan pada tahap pengumpulan data. Setelah nilai status diberi bobot maka bobot dari status tersebut akan dikalikan dengan rating pada *database* nya. *Rating* yang sudah dikalikan dengan bobot status maka akan dimasukkan pada kolom baru yakni kolom *scoreXstatus*. *Rating* yang sudah diberi bobot dan dikalikan akan membuat jarak data yang besar dari nilai 1 hingga 45, maka dari itu sebaiknya dinormalisasi dengan metode normalisasi *MinMax Normalization* membagi setiap nilai dengan nilai bobot maksimum sehingga didapatkan nilai diantara 0-1 yang sudah dibobotkan. Adapun untuk rumus normalisasi *MinMax* dapat dilihat pada Persamaan (1).

$$v_i' = \frac{v_i - \min_A}{\max_A - \min_A} \quad (1)$$

### 3.2 Pemodelan KNN Collaborative Filtering dan Pengujian

Setelah model KNN *collaborative filtering* telah dilatih untuk memprediksi *rating* dari *anime* untuk seorang *user* pada data latih sebesar 80% dari total data, selanjutnya akan dilakukan pengujian terhadap model untuk mengetahui seberapa besar kesalahan pada model dalam memprediksi *rating*. Pengujian akan dilakukan pada data uji sebesar 20% dari total data. Pengujian dari KNN *collaborative filtering* ini menggunakan nilai *Mean Absolute Error* (MAE) dan *Root Mean Square Error* (RMSE). Nilai MAE dan RMSE ini akan membandingkan *rating* hasil prediksi pada *data test* dengan *rating* sebenarnya dari data test tersebut. Semakin rendah nilai MAE atau RMSE, maka tingkat *error* dari model juga semakin rendah [14]–[16]. Pengujian akan dilakukan pada masing – masing *data testing* yang didapat pada *pre-processing* data sebelumnya. Pengujian juga dilakukan untuk melihat parameter terbaik yang bisa digunakan dalam *filtering*, yaitu nilai *k* terbaik. Rumus dari MAE dan RMSE dapat dilihat pada Persamaan (2) dan (3)

$$MAE = \sum \frac{|y' - y|}{n} \quad (2)$$

$$RMSE = \sqrt{\sum \frac{(Y' - Y)^2}{n}} \quad (3)$$

## 4 Hasil dan Pembahasan

### 4.1 Pengumpulan Data dan Pembobotan

Pengumpulan data pada penelitian ini dengan mengambil *dataset* dari *website* Kaggle oleh Matěj Račinský [13]. Data yang disediakan diantara lain data dari *anime*, data dari *user*, serta data *rating* dari *user* terhadap suatu *anime*. Selanjutnya dilakukan penyebaran kuisioner dan didapatkan data sebanyak 105 responden sebagai penentu bobot dari pembobotan. Detail skor setiap jawaban terdapat pada Tabel 4.

Tabel 4. Skor Maksimum Jawaban

Jawaban	Skor	Skor Maksimum (Skor * Jumlah Responden)
Sangat Setuju	4	420
Setuju	3	315
Tidak Setuju	2	210
Sangat Tidak Setuju	1	105

#### Pernyataan 1

Anda lebih percaya rekomendasi orang yang sudah selesai menonton *anime* tersebut. Persentase didapatkan dari nilai jumlah skor dikali dengan skor maksimum pada Tabel 4 sehingga

<http://sistemasi.ftik.unisi.ac.id>

didapatkan nilai 371 dibagi dengan 420 dikali 100% yang hasilnya adalah 88,33%. Hasil kuisioner pernyataan pertama dapat dilihat pada Tabel 5.

**Tabel 5. Hasil Kuisioner Pernyataan Pertama**

Jawaban	Skor	Responden	Jumlah Skor	Persentase
Sangat Setuju	4	62	248	(371 / 420) * 100% = 88,33%
Setuju	3	37	111	
Tidak Setuju	2	6	12	
Sangat Tidak Setuju	1	0	0	
<b>Jumlah</b>		105	371	

### Pernyataan 2

Anda lebih percaya rekomendasi orang yang sedang menonton *anime* tersebut. Persentase didapatkan dari nilai jumlah skor dikali dengan skor maksimum pada Tabel 4 sehingga didapatkan nilai 314 dibagi dengan 420 dikali 100% yang hasilnya adalah 74,76%. Hasil kuisioner pernyataan kedua dapat dilihat pada Tabel 6.

**Tabel 6. Hasil Kuisioner Pernyataan Kedua**

Jawaban	Skor	Responden	Jumlah Skor	Persentase
Sangat Setuju	4	12	48	(314 / 420) * 100% = 74,76 %
Setuju	3	81	243	
Tidak Setuju	2	11	22	
Sangat Tidak Setuju	1	1	1	
<b>Jumlah</b>		105	314	

### Pernyataan 3

Anda lebih percaya rekomendasi orang yang ingin menonton *anime* tersebut. Persentase didapatkan dari nilai jumlah skor dikali dengan skor maksimum pada Tabel 4 sehingga didapatkan nilai 239 dibagi dengan 420 dikali 100% yang hasilnya adalah 56,90%. Hasil kuisioner pernyataan ketiga dapat dilihat pada **Error! Not a valid bookmark self-reference..**

**Tabel 7. Hasil Kuisioner Pernyataan Ketiga**

Jawaban	Skor	Responden	Jumlah Skor	Persentase
Sangat Setuju	4	4	16	(239 / 420) * 100% = 56,90 %
Setuju	3	33	99	
Tidak Setuju	2	56	112	
Sangat Tidak Setuju	1	12	12	
<b>Jumlah</b>		105	239	

### Pernyataan 4

Anda lebih percaya rekomendasi orang yang berhenti menonton *anime* tersebut. Persentase didapatkan dari nilai jumlah skor dikali dengan skor maksimum pada Tabel 4 sehingga didapatkan nilai 237 dibagi dengan 420 dikali 100% yang hasilnya adalah 56,43%. Hasil kuisioner pernyataan ketiga dapat dilihat pada Tabel 8.

**Tabel 8. Hasil Kuisioner Pernyataan Keempat**

Jawaban	Skor	Responden	Jumlah Skor	Persentase
Sangat Setuju	4	10	40	(237 / 420) * 100% = 56,43 %
Setuju	3	29	87	
Tidak Setuju	2	44	88	
Sangat Tidak Setuju	1	22	22	
<b>Jumlah</b>		105	237	

## Pernyataan 5

Anda lebih percaya rekomendasi orang yang hiatus menonton *anime* tersebut. Persentase didapatkan dari nilai jumlah skor dikali dengan skor maksimum pada Tabel 4 sehingga didapatkan nilai 209 dibagi dengan 420 dikali 100% yang hasilnya adalah 49,76%. Hasil kuisisioner pernyataan ketiga dapat dilihat pada Tabel 9.

**Tabel 9. Hasil Kuisisioner Pernyataan Keenam**

Jawaban	Skor	Responden	Jumlah Skor	Persentase
Sangat Setuju	4	1	4	(209 / 420) * 100% = 49,76 %
Setuju	3	23	69	
Tidak Setuju	2	55	110	
Sangat Tidak Setuju	1	26	26	
<b>Jumlah</b>		105	209	

Dari tabel hasil kuisisioner diatas, dilakukan penentuan bobot berdasarkan tingkat dari setuju atau tidak pada tiap kolom pada tabel. Untuk sangat setuju diberikan skor nilai 4, setuju diberikan skor nilai 3, tidak setuju diberi skor nilai 2 dan sangat tidak setuju diberi skor nilai 1. Masing-masing skor tersebut akan dikalikan dengan hasil kuisisioner Penilaian bobot diberikan nilai dengan jarak 1 hingga 5 berdasarkan urutan dari persentase terbesar. Untuk status selesai menonton akan diberi bobot 5, untuk sedang menonton diberi bobot 3, untuk berencana untuk menonton dan berhenti diberikan nilai 2 karena persentase skala tidak berbeda jauh, sedangkan untuk status sedang menahan / tidak menonton untuk sementara diberikan bobot 1. Untuk bobot bernilai 4 tidak diberikan pada status karena sudah ada skala yang sama dan memberikan bobot yang lebih untuk telah selesai menonton. Pembobotan pada status dapat disimpulkan pada tabel Tabel 10 berikut.

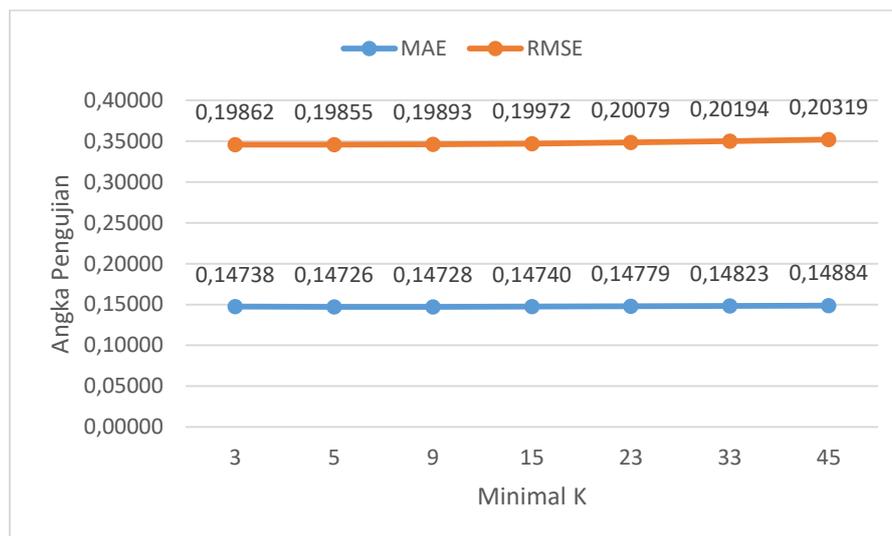
**Tabel 10. Pembobotan Status**

Status	Bobot
Sedang menonton	3
Telah selesai menonton	5
Sedang menahan / tidak menonton untuk sementara	1
Berencana untuk menonton	2
Berhenti menonton	2

## 4.2 Pemodelan KNN Collaborative Filtering dan Pengujian

Tahapan selanjutnya adalah pengujian model KNN *collaborative filtering* dengan pembobotan. Pengujian dilakukan dengan menghitung *Mean Absolute Error* (MAE) dan *Root Mean Square Error* (RMSE) dari model KNN *collaborative filtering* yang diberi pembobotan dengan yang tidak. Parameter pengujian menggunakan nilai K yang terdiri dari 3, 5, 9, 15, 23, 33, dan 45 yang bertujuan untuk mengetahui evaluasi dari penambahan bobot pada algoritma KNN *collaborative filtering*. Hasil dari pengujian model KNN *collaborative filtering* dengan pembobotan dapat dilihat pada Gambar 1. Grafik MAE dan RMSE

Dari hasil grafik pada Gambar 2, model KNN *collaborative filtering* dengan pembobotan menunjukkan nilai parameter *min\_k* yang optimal adalah 5. Nilai error juga semakin meningkat seiring bertambahnya parameter nilai *k* pada model. Hal ini dikarenakan bahwa semakin banyak nilai *k*, maka semakin banyak data yang diambil sehingga dalam mengklasifikasi semakin banyak data yang tidak relevan dan mempengaruhi nilai pada prediksi. Tetapi untuk peningkatan nilai error tidak meningkat secara signifikan melainkan meningkat secara stabil pada semua nilai *k*.



Gambar 1. Grafik MAE dan RMSE

## 5 Kesimpulan

Penelitian ini berhasil membangun sistem rekomendasi *anime* dengan menggunakan metode KNN *collaborative filtering* dengan pembobotan dalam melakukan rekomendasi *anime* berdasarkan *rating user* lainnya dengan hasil pengujian nilai MAE sebesar 0,14726 dan nilai RMSE sebesar 0,19855 untuk nilai  $k$  sebesar 5. Penelitian dilakukan dengan data sebesar 193.272 yang dibagi menjadi 80% data *training* dan 20% data *testing*. Model KNN *collaborative filtering* dengan pembobotan ini dapat memprediksi *rating* yang sesuai dengan *dataset* yang dimana *user* memberi *rating* yang cukup berbeda dengan *rata-rata user* lainnya. Jika *rating user* berbeda signifikan dengan *rating rata-rata user* lainnya, maka akan terjadi kesalahan dalam prediksi *rating*. Adapun saran yang dapat diambil pada penelitian ini untuk pengembangan pada penelitian selanjutnya adalah model yang digunakan menggunakan *model-based* seperti *matrix factorization* untuk mengurangi penggunaan memori dan memudahkan pembaharuan model. Selain itu perlunya pencarian parameter yang optimal secara otomatis seperti menggunakan proses *hyperparameter tuning* untuk mendapatkan nilai parameter yang optimal. Serta penambahan pembobotan pada prediksi perlu untuk dikaji ulang agar dapat meningkatkan kualitas rekomendasi.

## Referensi

- [1] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*. Boston, MA: Springer US, 2011. doi: 10.1007/978-0-387-85820-3.
- [2] Y. Hu, F. Xiong, D. Lu, X. Wang, X. Xiong, and H. Chen, "Movie collaborative filtering with multiplex implicit feedbacks," *Neurocomputing*, vol. 398, pp. 485–494, Jul. 2020, doi: 10.1016/j.neucom.2019.03.098.
- [3] J.-H. Su, W.-Y. Chang, and V. S. Tseng, "Effective social content-based collaborative filtering for music recommendation," *Intelligent Data Analysis*, vol. 21, pp. S195–S216, Apr. 2017, doi: 10.3233/IDA-170878.
- [4] Z. Fayyaz, M. Ebrahimian, D. Nawara, A. Ibrahim, and R. Kashef, "Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities," *Applied Sciences*, vol. 10, no. 21, p. 7748, Nov. 2020, doi: 10.3390/app10217748.
- [5] N. F. Al-Bakri and S. H. Hashim, "Reducing Data Sparsity in Recommender Systems," *Journal of Al-Nahrain University Science*, vol. 21, no. 2, pp. 138–147, Jun. 2018, doi: 10.22401/JNUS.21.2.20.
- [6] A. S. Girsang, B. Al Faruq, H. R. Herlianto, and S. Simbolon, "Collaborative Recommendation System in Users of Anime Films," *J Phys Conf Ser*, vol. 1566, no. 1, p. 012057, Jun. 2020, doi: 10.1088/1742-6596/1566/1/012057.

- [7] J.-J. Vie *et al.*, “Using Posters to Recommend Anime and Mangas in a Cold-Start Scenario,” in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, IEEE, Nov. 2017, pp. 21–26. doi: 10.1109/ICDAR.2017.287.
- [8] B. Guo, S. Xu, D. Liu, L. Niu, F. Tan, and Y. Zhang, “Collaborative filtering recommendation model with user similarity filling,” in *2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC)*, IEEE, Oct. 2017, pp. 1151–1154. doi: 10.1109/ITOEC.2017.8122536.
- [9] S. Suriati, M. Dwiastuti, and T. Tulus, “Weighted hybrid technique for recommender system,” *J Phys Conf Ser*, vol. 930, p. 012050, Dec. 2017, doi: 10.1088/1742-6596/930/1/012050.
- [10] A. S. Dharma, R. B. B. A. Hutasoit, and R. R. Pangaribuan, “Sistem Rekomendasi Menggunakan Item-based Collaborative Filtering pada Konten Artikel Berita,” *Jurnal Nasional Teknik Informatika dan Elektro (JURNALTIO)*, vol. 2, no. 1, 2021.
- [11] I. G. Gusti, M. Nasrun, and R. A. Nugrahaeni, “Rekomendasi Sistem Pemilihan Mobil Menggunakan K-Nearest Neighbor (KNN) Collaborative Filtering,” *TEKTRIKA*, vol. 4, no. 1, 2019.
- [12] O. Bourkhouk and O. Achbarou, “Weighting based approach for learning resources recommendations,” *JOIV: International Journal on Informatics Visualization*, vol. 2, no. 3, p. 104, Apr. 2018, doi: 10.30630/joiv.2.3.124.
- [13] Matěj Račinský, “MyAnimeList Dataset,” <https://www.kaggle.com/azathoth42/myanimelist>, 2018.
- [14] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, “A collaborative filtering approach to mitigate the new user cold start problem,” *Knowl Based Syst*, vol. 26, pp. 225–238, Feb. 2012, doi: 10.1016/j.knosys.2011.07.021.
- [15] B. M. Kim, Q. Li, C. S. Park, S. G. Kim, and J. Y. Kim, “A new approach for combining content-based and collaborative filters,” *J Intell Inf Syst*, vol. 27, no. 1, pp. 79–91, Jul. 2006, doi: 10.1007/s10844-006-8771-2.
- [16] Q. Li, S. H. Myaeng, and B. M. Kim, “A probabilistic music recommender considering user opinions and audio features,” *Inf Process Manag*, vol. 43, no. 2, pp. 473–487, Mar. 2007, doi: 10.1016/j.ipm.2006.07.005.