

Sistem Rekomendasi Seri Animasi Jepang (Anime) Menggunakan *User-Based Collaborative Filtering* dan *Spearman Rank Correlation Coefficient*

I Kadek Gowinda^{a1}, I Gede Santi Astawa^{a2}, I Gusti Ngurah Anom Cahyadi Putra^{a3}, Dr. Ngurah Agus Sanjaya^{a4}, Ida Bagus Gede Dwidasmara^{a5}, I Dewa Made Bayu Atmaja Darmawan^{a6}

^aProgram Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana Badung, Bali, Indonesia

¹gowinda338@gmail.com

²santi.astawa@unud.ac.id

³anom.cp@unud.ac.id

⁴agus_sanjaya@unud.ac.id

⁵dwidasmara@unud.ac.id

⁶dewabayu@unud.ac.id

Abstract

The number of existing anime is increasingly varied and more in line with the increasing number of enthusiasts. The surge in anime series among anime enthusiasts has become an obstacle to finding anime that matches their taste. This underlies the writer to create an anime recommendation system using User-Based Collaborative Filtering method. The research process consisted of several stages, namely data collection from the Kaggle website with 3 pieces of data uploaded, namely in the .csv format. Determination of users who have a correlation, using the Spearman Rank Correlation Coefficient method. Calculation of predictions using a weighted sum algorithm. The final stage is the implementation of the recommendations and evaluation of the recommendation system used to calculate the level of collaborative filtering using the Mean Absolute Error (MAE).. This research has output in the form of a website which has several components, namely Home Page, Login-Register, Search, Recommend, Result Page, Single View and Rating. Testing on the system uses MAE calculations which are carried out on 50 users with the most rating history. The results from the test show that the percentage of error obtained is 15.8% and the prediction accuracy results obtained are 84.12%. The smallest MAE value of the 50 profiles is 0.894933222 by Archaeon and the highest MAE value is 3.572438553 by Krunchyman.

Keywords: Recommendation System, Rating, Spearman Rank Correlation Coefficient, Collaborative Filtering, MAE, Accuration.

1. Pendahuluan

Dimasa globalisasi sekarang ini, Indonesia sudah tidak asing lagi dengan adaptasi kultur yang berasal dari luar daerah, bahkan luar negara. Terutama gen-Z yang sudah dari usia muda mendapatkan akses informasi terhadap budaya-budaya luar, akan cenderung untuk mengambil kultur tersebut yang biasanya menjadi tren atau biasa disebut kultur populer (*pop culture*). *Pop culture* adalah kultur yang diciptakan oleh masyarakat demi kepentingan sendiri kemudian menjadi tren yang disorot oleh media massa sehingga dapat dinikmati oleh lebih banyak kalangan dan penyebarannya pun menjadi lebih luas sebagai hiburan bagi masyarakat tersebut [1].

Seri Animasi Jepang (Anime) merupakan salah satu bentuk hiburan dalam dunia entertainen yang biasanya ditayangkan di televisi Jepang. Anime sendiri sudah masuk ke ranah hiburan di Indonesia dari tahun 2000-an awal di stasiun televisi Indonesia. Sekarang ini industri hiburan Jepang di dunia bahkan semakin meningkat pesat akibat perkembangan internet, termasuk di Indonesia. Anime memiliki jadwal penayangan yang mengikuti alur 4 musim yang ada di Jepang (summer, fall, winter, spring) [2], sehingga setiap tahunnya ada 4 musim penayangan anime baru yang tiap musimnya bisa memiliki hingga puluhan anime baru.

Sistem Rekomendasi Seri Animasi Jepang (Anime) Menggunakan *User-Based Collaborative Filtering* dan *Spearman Rank Correlation Coefficient*

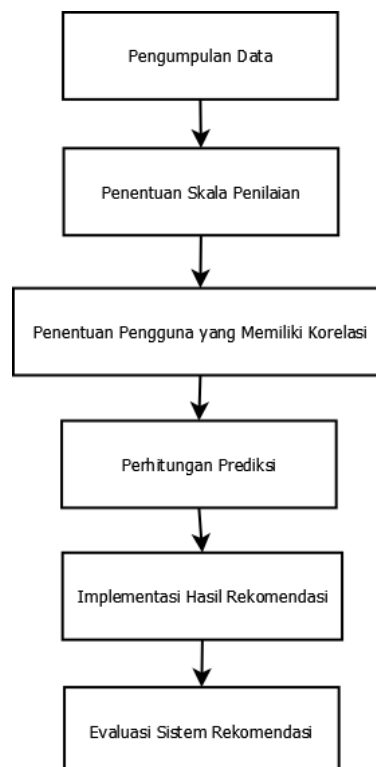
Schafer [3] memperkenalkan dua kelas berbeda dari nearest neighbor algoritma *collaborative filtering recommender system*: *user-based nearest neighbor* dan *item based nearest neighbor*. Dari metode yang diperkenalkan Schafer, telah banyak dilakukan penelitian tentang *recommender system* menggunakan *collaborative filtering*. Contohnya adalah penelitian oleh Moh Irfan [4], yang menggunakan metode item-based collaborative filtering untuk membuat sistem rekomendasi. Kendala data yang sedikit serta adanya *cold start problem* (data baru yang belum memiliki nilai rating) menyebabkan hasil prediksi sistem ini kurang baik. *Cold Start Problem*, yaitu memberikan rekomendasi kepada pengguna yang baru yang tidak memiliki referensi [5].

Sistem rekomendasi menggunakan *collaborative filtering* memiliki dua tahapan yaitu, pencarian *similar user* dan perhitungan prediksi. Dalam pencarian *similar user*, ada dua metode populer yang biasanya digunakan yaitu, *Spearman Correlation* dan *Pearson Correlation*. Kedua metode memiliki tujuan yang berbeda. Korelasi Spearman yang lebih mengutamakan kemonotonan data akan lebih tepat digunakan dalam data dengan variabilitas yang rendah namun tetap dengan kualitas korelasi yang tinggi daripada korelasi Pearson yang bersifat linear [6]. Karena hal ini, peneliti akan memakai Korelasi Spearman dalam pembuatan sistem. Lalu untuk perhitungan prediksi skor rating, akan dihitung menggunakan *weighted sum average*. Kedua metode ini nantinya akan diaplikasikan kedalam sistem rekomendasi lalu untuk menghitung tingkat akurasinya akan dilakukan evaluasi dengan menggunakan *mean absolute error* (MAE).

2. Metodologi Penelitian

2.1. Skema Penelitian

Berikut ini merupakan skema penelitian dalam diagram blok:



Gambar 1. Blok Diagram Skema Penelitian

2.2. Pengumpulan Data

Data yang digunakan merupakan dataset sekunder yang berasal dari Website MyAnimeList yang diambil sampelnya oleh Marlesson dan di upload pada website Kaggle. Data yang didapatkan berupa data website terhitung hingga tahun 2020, sehingga tidak akan menampilkan anime-anime yang tayang setelahnya.

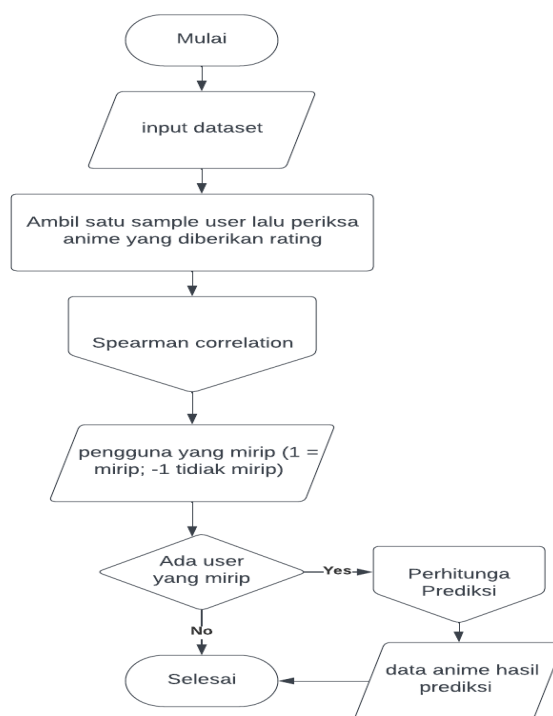
2.3. Penentuan Skala Penilaian

Website MyAnimeList memberikan pengguna kemungkinan untuk melakukan rating kepada suatu seri anime dalam rentang 1-10. Skala rating (rating scale) merupakan skala penilaian yang lebih fleksibel, skala penilaian tidak hanya untuk mengukur nilai tetapi dapat juga digunakan untuk mengukur persepsi pengguna terhadap suatu seri tertentu, apakah pengguna tersebut mendapatkan hiburan, ilmu, atau kepuasan dalam menonton anime tersebut. Penilaian rating scale merupakan keputusan yang menentukan skala mana yang digunakan untuk penilaian yang digunakan, apakah 1 sampai 3, 1 sampai 5, 1 sampai 7, 1 sampai 10, atau 1 sampai 100. Memilih rating scale merupakan preferensi dari peneliti itu sendiri, namun ada beberapa hal yang harus dipertimbangkan, yaitu:

1. Skala dengan nilai yang lebih tinggi akan memberikan perbedaan yang lebih detail terhadap hasil penelitian. Hal ini juga dapat membuat kesan bagi pengguna yang memberikan penilaian tersebut, dibandingkan dengan hanya mengatakan 'suka' atau 'tidak suka' terhadap suatu tayangan, pengguna mungkin lebih nyaman mengatakan jika mereka memberikan nilai menengah kebawah untuk tayangan yang tidak begitu disukai atau memberikan nilai menengah keatas bagi tayangan yang lumayan menarik perhatian mereka.
2. Laki-laki, perempuan, disemua usia, semua suku dan bangsa lebih menyukai penilaian 1-10. Hal ini dikarenakan skala 1-10 dapat digunakan secara universal.

2.4. Rekomendasi

Pada tahap ini, akan dilakukan proses perhitungan dan pengolahan data agar dapat digunakan sebagai rekomendasi. Berikut merupakan flowchart yang menunjukkan bagaimana alur sistem.



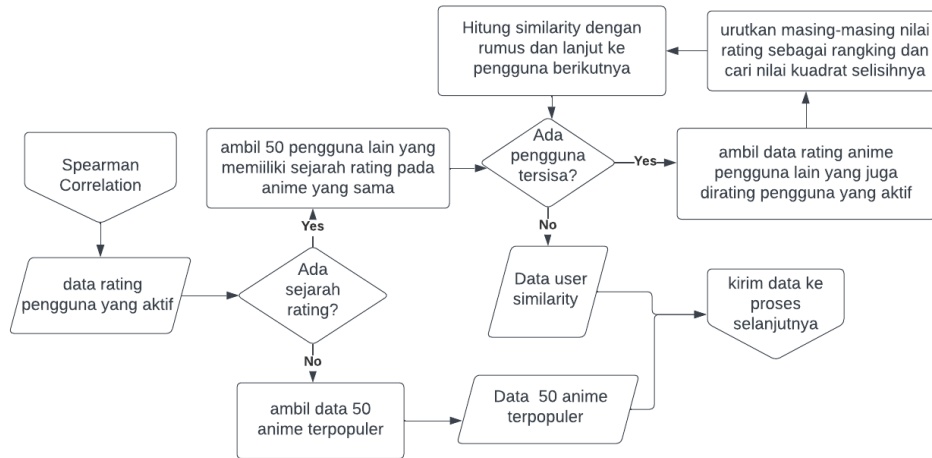
Gambar 2. Flowchart pengolahan data sistem rekomendasi

2.4.1 Penentuan Pengguna yang Memiliki Korelasi (user-corelation)

Penentuan kemiripan pengguna dilakukan untuk membuat sebuah kluster yang akan digunakan dalam perhitungan collaborative filtering. Penentuan ini akan dihitung menggunakan metode Spearman Rank Correlation Coefficient. Dalam beberapa penelitian sistem rekomendasi yang menggunakan metode item-based collaborative filtering banyak yang menggunakan adjusted cosine similarity untuk menghitung nilai kemiripan antar dua item. Adjusted cosine similarity sebenarnya bisa saja digunakan dalam penelitian ini, namun peneliti menemukan jika penggunaan metode Spearman Rank Correlation Coefficient akan lebih tepat digunakan karena sample data yang digunakan cukup sederhana dan hanya terfokus pada nilai rating.

Sistem Rekomendasi Seri Animasi Jepang (Anime) Menggunakan *User-Based Collaborative Filtering* dan *Spearman Rank Correlation Coefficient*

Untuk memahami kolerasi spearman, sangat penting untuk kita memahami sebelumnya tentang fungsi monotonic. Fungsi monotonic merupakan suatu fungsi yang tidak bertambah, begitupun berkurang sebagaimana variabel independennya berubah. Jadi untuk lebih sederhananya, jika fungsi cenderung naik maka nilai tidak akan turun begitupun sebaliknya. Untuk visualisasi bisa dilihat pada gambar dibawah.



Gambar 3. Flowchart Perhitungan Korelasi Spearman

2.4.2 Perhitungan Prediksi

Setelah nilai kemiripan didapatkan, maka langkah selanjutnya adalah proses perhitungan prediksi itu sendiri. Proses prediksi dilakukan menggunakan perkiraan nilai rating user terhadap suatu item yang belum pernah di-rating sebelumnya oleh *user*.

Algoritma *Weighted Sum* mendapatkan nilai prediksi rating dengan mengitung total rating yang diberikan terhadap *item* oleh pengguna. Rumus dari perhitungan prediksi yaitu:

$$P_{u,i} = \frac{\sum_{all\ similar\ user, N}(S_{u,y} \times R_{y,i})}{\sum_{all\ similar\ user, N}(|S_{u,y}|)} \tag{1}$$

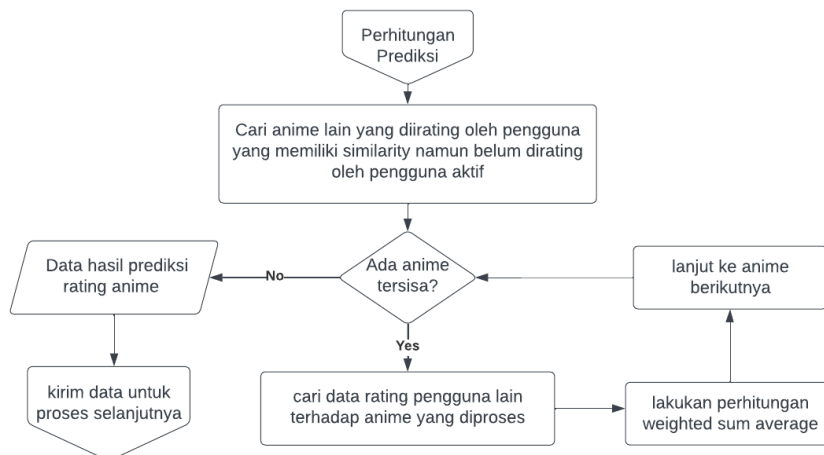
Keterangan:

$P_{u,i}$ adalah prediksi rating *user u* terhadap *item i*.

$S_{u,y}$ merupakan nilai *similarity* yang didapat antara *user u* dan *i*.

$R_{y,i}$ merupakan rating *user y* terhadap *item i*.

Perhitungan prediksi akan dilakukan pada setiap user dalam kluster yang sama (memiliki kemiripan yang tinggi). Setelah semua perhitungan dilakukan maka dapat ditentukan item mana yang kemungkinan besar dapat direkomendasikan pada user. Rekomendasi ini mengacu pada hasil prediksi yang didapatkan. Nantinya, tiap hasil prediksi ini akan dibandingkan dengan nilai rating yang sebenarnya dalam pengujian.



Gambar 4. Flowchart Perhitungan Prediksi

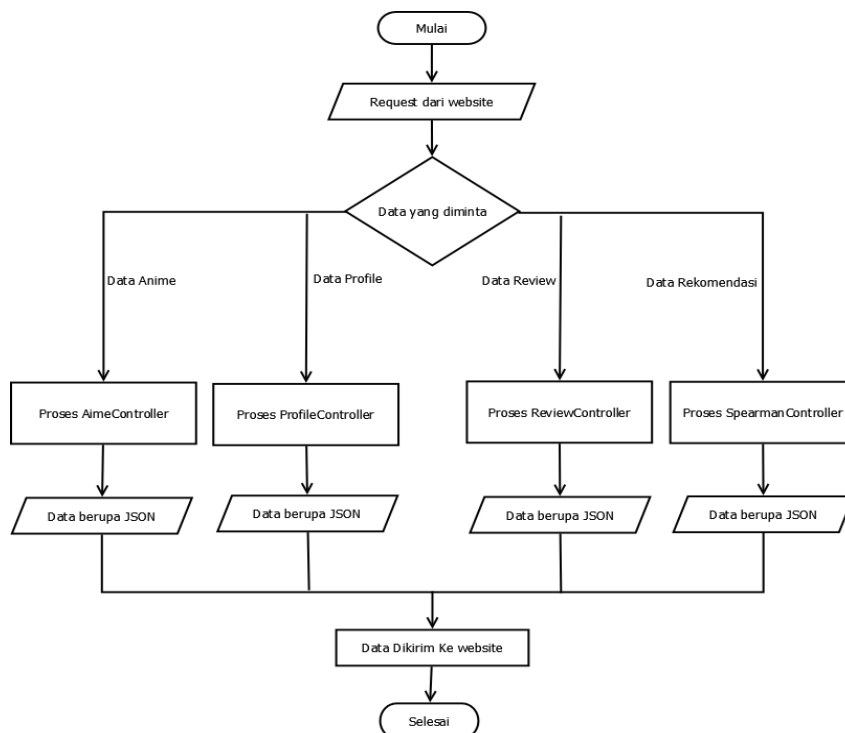
2.5. Rancangan Sistem

2.5.1. Rancangan API

Proses-proses pada rekomendasi telah dijabarkan pada poin 3.4. proses-proses tersebut nantinya akan dijalankan pada server backend yang akan dibuat menggunakan Laravel. Pada backend perhitungan-perhitungan yang menghasilkan data akan dijalankan saat ada permintaan dari pengguna.

Proses-proses diatas nantinya akan disimpan sebagai sebuah controller Laravel yang akan berjalan saat ada pemanggilan dari sistem. Server berjalan secara lokal menggunakan bantuan aplikasi XAMPP untuk menjalankan database. Server backend akan menangani permintaan atau request pemanggilan atau perubahan data. Sistem backend ini biasa disebut rest API.

Berikut ini adalah flowchart bagaimana alur data dari request hingga mendapatkan keluaran data berupa JSON.



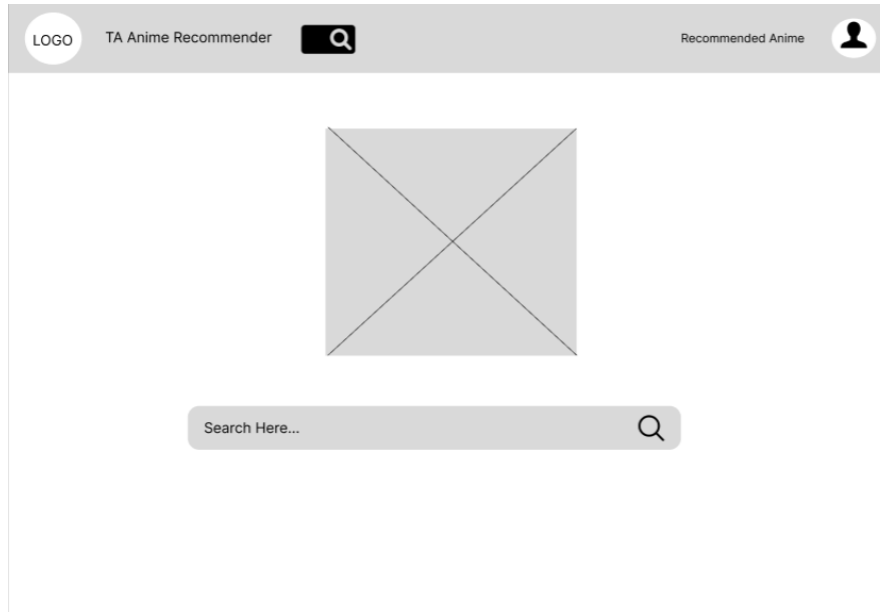
Gambar 5. Diagram Alur Data dari Request Hingga Terkirim

2.5.2. Rancangan *User Interface* website

Website nantinya akan memiliki 3 komponen utama yang akan menjadi halaman masing-masing yaitu halaman utama, halaman hasil, dan halaman review. Untuk detail desain dapat dilihat dibawah ini:

1. Halaman utama

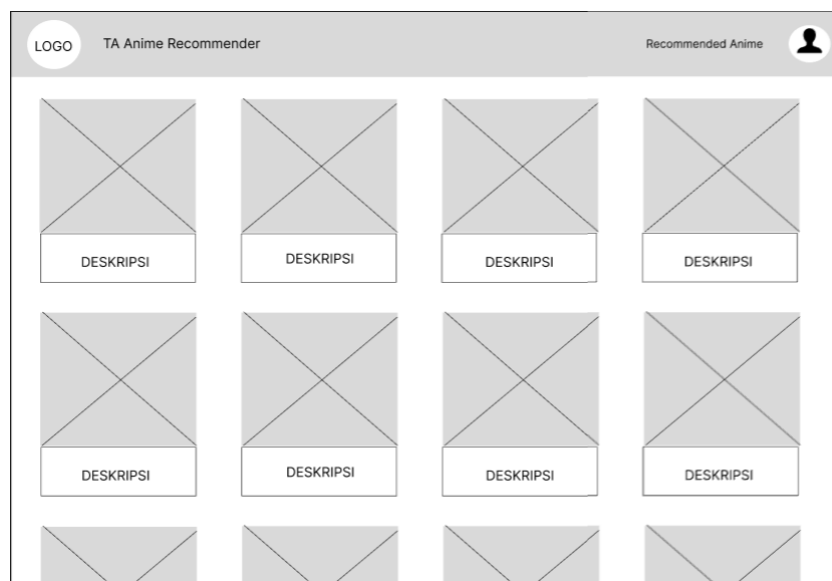
Pada halaman utaman akan berisikan beberapa komponen yang dapat berinteraksi dengan pengguna seperti logo dan nama website yang akan membawa pengguna untuk kembali ke halaman utama, field untuk pencarian yang memungkinkan pengguna untuk mencari anime yang diinginkan, tombol rekomendasi yang akan membawa pengguna ke halaman hasil untuk menampilkan hasil rekomendasi, lalu ada juga tombol profile yang dapat digunakan untuk login atau logout dari sistem.



Gambar 6. Mockup Desain Halaman Utama Website

2. Halaman Hasil

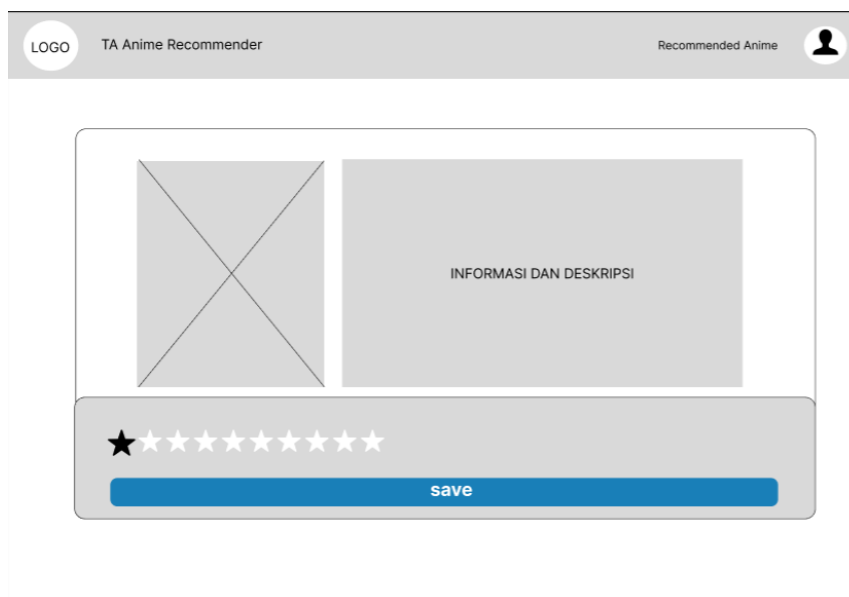
Pada halaman hasil, akan ditampilkan hasil pencarian ataupun hasil rekomendasi anime oleh pengguna. Hasil anime yang didapatkan akan ditampilkan ke dalam card-card yang menampung gambar serta deskripsi dari tiap-tiap anime. Dan saat di tekan, maka akan dialihkan ke halaman review yang akan menampilkan satu anime sesuai dengan yang ditekan pada halaman hasil ini.



Gambar 7. Mockup Desain Halaman Hasil Website

3. Halaman Review

Halaman review menampilkan satu anime yang telah dipilih sebelumnya pada halaman hasil. Pada halaman ini akan menjabarkan informasi dan deskripsi anime secara lebih lengkap. Selain deskripsi, ada juga fitur untuk memberikan review. Fitur ini ditampilkan dalam bentuk bintang-bintang yang merepresentasikan nilai 1 hingga 10. Saat pengguna telah memilih nilai yang ingin diberikan, tombol save akan menyimpan hasil review tersebut ke dalam database.



Gambar 8. Mockup Desain Halaman Review Website

2.6. Implementasi hasil Rekomendasi

Setelah perhitungan komputasi untuk mencari hasil prediksi nilai skor rating telah selesai, tentunya perhitungan tersebut perlu diaplikasikan dalam sistem. Dalam penelitian ini, sistem akan dibentuk dan dirangkum dalam sebuah website. Sistem website ini nantinya akan terdiri dari dua komponen utama yaitu backend dan frontend. Pada backend atau server, semua perhitungan dan data akan diolah dan disimpan, sedangkan pada frontend atau tampilan antarmuka, data akan ditampilkan kepada pengguna dalam bentuk yang bisa diobservasi secara langsung oleh pengguna.

Dalam implementasinya nanti, akan digunakan beberapa alat yang berupa perangkat lunak serta beberapa bahasa pemrograman yang akan digunakan dalam membangun sistem ini. Untuk alat-alat yang digunakan dan penjelasannya dapat dilihat pada tabel 1.

Tabel 1. Alat-alat Pendukung dalam Pembangunan Website

No	Alat/Tools	Keterangan
1	PC/Laptop	Media untuk menjalankan perangkat lunak yang akan digunakan dalam pembangunan sistem.
2	Browser	Digunakan untuk melihat pratinjau dari website selama masa pengembangannya
3	Visual Studio Code	Sebagai text editor yang akan digunakan untuk menulis kode-kode yang nantinya akan mengkonstruksi <i>backend</i> maupun <i>frontend</i>
4	Laravel dan Node.js	Merupakan <i>library</i> atau <i>framework</i> berbasis bahasa pemrograman PHP dan Javascript yang akan digunakan dalam membangun sebuah server yang akan berfungsi sebagai <i>REST API</i> .
5	React.js	Merupakan <i>Framework Javascript</i> yang akan memudahkan pembuatan tampilan antarmuka pada sistem.
6	XAMPP	Adalah sebuah aplikasi yang berperan sebagai server lokal yang dapat digunakan untuk menjalankan <i>backend</i> yang dibuat. XAMPP

juga berperan sebagai database dengan basis MySQL yang akan menyimpan data-data yang diperlukan.

7	Postman	Postman merupakan aplikasi yang dapat digunakan untuk mengambil data pada suatu server dengan melakukan sebuah <i>request</i> . Dengan bantuan aplikasi Postman, pengembangan server backend akan menjadi lebih mudah.
---	---------	--

2.7. Evaluasi Sistem Rekomendasi

Evaluasi merupakan sebuah kegiatan yang dilakukan untuk menilai efisiensi dan efektivitas suatu hal. Dalam penelitian ini, evaluasi digunakan untuk mengetahui seberapa baik hasil yang didapatkan oleh sistem rekomendasi.

Jenis evaluasi yang paling sering dipakai untuk menghitung baik tidaknya collaborative filtering adalah menggunakan *Mean Absolut Error* (MAE). MAE merupakan persamaan yang termasuk ke dalam jenis *statistical accuracy metrics* dimana MAE akan menghitung nilai rata-rata selisih antara nilai prediksi *rating* dengan nilai *rating* yang sebenarnya.

Selanjutnya untuk menentukan nilai akurasi maka akan kita lihat rentangan nilai *rating*. Pada kasus ini, nilai *rating* paling rendah adalah 1 dan paling tinggi adalah 10, maka kemungkinan error prediksi nilai *rating* paling tinggi adalah 9. Lalu semua nilai error dari sample pengguna yang dipakai akan dicari rata-ratanya dan dibagi dengan nilai 9. Untuk mendapatkan hasil dalam bentuk persentase, maka hasil perhitungan akan dikali dengan 100. Dari sini dapat kita cari nilai persentase error dengan rumus:

$$\text{persentase error} = \left(\frac{\text{rata-rata error}}{9} \right) \times 100 \quad (1)$$

Setelah didapatkan persentasi error, maka dapat dicari persentasi akurasi dengan mengurangi persentase error dari 100.

3. Hasil dan Pembahasan

3.1. Data Penelitian

Seperti yang disebutkan sebelumnya, data penelitian menggunakan data sekunder yang diambil dari website Kaggle. Data ini diunggah oleh seorang pengguna bernama Marlesson. Data yang diunggah berupa 3 buah data dengan format .csv. Tiga data tersebut dibagi menjadi:

1. Animes.csv

Data ini berjumlah total 16214, yang merupakan list dari data anime yang ada pada website MyAnimeList hingga tahun 2020. List data ini berisikan beberapa komponen diantaranya: uid, title, synopsis, genre, aired, episodes, members, popularity, ranked, skor, img_url, link.

Komponen penting dari data anime yang digunakan dalam penelitian adalah uid, title, img_url, synopsis, aired, rating, skor, episodes, dan link. Data ini akan digunakan pada implementasi frontend sistem untuk ditampilkan kepada pengguna.

2. Profiles.csv

Data profiles berisi 47854 list pengguna yang terdaftar pada website MyAnimeList hingga tahun 2020. Komponen yang ada pada file ini diantaranya: profileName, gender, birthday, favorites_anime, link.

Komponen penting yang digunakan dalam penelitian adalah profileName. Karena tidak mungkin untuk mendapatkan password milik pengguna, jadi peneliti membuat password dummy untuk semua pengguna. Serta membuat data email dummy untuk digunakan menggantikan profileName nantinya untuk pengguna baru.

3. Reviews.csv

Data reviews merupakan data paling penting yang akan digunakan dalam komputasi sistem. Dengan jumlah 128998 dapat menjadi kelebihan sendiri terhadap kualitas data, namun dilain sisi dengan data

sebanyak ini, sistem dapat dibebani. Komponen yang dimiliki file ini diantaranya: uid, profileName, animeUid, skor, skors, dan link.

profileName dan animeUid adalah korelasi dari list anime dan profiles, tentunya skor dari reviews yang akan berperan penting untuk proses komputasi.

Dari semua data yang didapatkan, peneliti menambahkan beberapa komponen data yang dapat berguna pada sistem yang dibuat atau berguna bagi perkembangannya nanti. Ketiga data csv ini akan dikonversi menjadi data relasional dalam bentuk sql. Kemudian akan digunakan dalam database MySQL dengan membuat tabel animes dan profiles berelasi pada table reviews.

3.2. Implementasi Sistem

3.2.1 Implementasi Komputasi

Proses komputasi rekomendasi anime dapat dibagi menjadi 2 bagian penting sesuai dengan proses collaborative filtering, yaitu proses perhitungan user similarity dan perhitungan prediksi. Peneliti menggunakan metode pemrograman functional untuk implementasi komputasi sistem.

a. Proses Perhitungan *User Similarity*

User similarity akan dicari dengan menggunakan metode spearman rank correlation coefficient. Data yang dibutuhkan untuk perhitungan spearman adalah data rating user terhadap anime. Yang akan menjadi titik pusat perbandingan nantinya tentu adalah pengguna yang memakai sistem, sehingga perlu dilakukan log-in terlebih dahulu.

Jika tidak ada korelasi sama sekali antar user yang dibandingkan percuma untuk dilakukan perhitungan. Sehingga kita perlu mengabaikan pengguna lain yang sama sekali tidak memiliki kesamaan dengan pengguna aktif. Artinya jika pengguna lain tidak melakukan rating pada anime yang sama dengan pengguna yang *login*, kita bisa abaikan pengguna tersebut.

Tabel 2. Kode Program Seleksi Data Pengguna Lain

Penggalan Kode
<pre>\$sql = "SELECT profileName, count(profileName) from (SELECT profileName FROM reviews WHERE animeUid IN (SELECT animeUid FROM reviews WHERE profileName = '\$username') AND profileName <> '\$username') as hasil group by profileName HAVING count(profileName) ORDER BY count(profileName) DESC, profileName ASC LIMIT 50;"; \$getSimilarUser = mysqli_query(\$con, \$sql) or die("Bad Query!" . \$sql); \$similarUser = mysqli_fetch_all(\$getSimilarUser);</pre>

Kode diatas akan menghasilkan sebuah object berbentuk array yang disimpan pada variable \$similarUser. Mengingat jumlah pengguna yang banyak, akan kurang baik bagi sistem jika tidak dibatasi maka, query diatas memiliki LIMIT 50 dengan mengurutkannya dari jumlah anime yang sama yang dirating oleh pengguna dan pengguna lain.

Setelah itu, melalui perulangan, tiap pengguna yang masuk ke seleksi akan dihitung kemiripannya menggunakan metode spearman, namun sebelumnya kita harus mengetahui juga nilai rating dari pengguna lain terhadap anime-anime yang sama-sama dirating oleh kedua pengguna yang dibandingkan. Penggunaan metode konvensional dengan memakai matrix akan sangat membebani sistem mengingat tiap pengambilan nilai dari database akan memakan waktu dan jika dikalikan dengan 50 data user dan 50 data anime, maka pengambilan data akan dilakukan sebanyak 2500 kali. Dan bisa dibayangkan jika perhitungan tidak dibatasi dengan 50 data. Dengan

menganalisa potensi masalah dilakukan optimasi dengan melakukan hanya sekali pemanggilan dari database untuk tiap user, hingga waktu yang diperlukan dapat dipersingkat hingga 50 kali lipat.

Tabel 3. Kode Program untuk Pengambilan Data Rating Pengguna Lain

Penggalan Kode
<pre> foreach (\$similarUser as \$index => \$profileName) { \$sql = "SELECT u.animeUid as anime, similar.skor as lain, u.skor as pengguna from (SELECT * FROM reviews WHERE profileName = '\$profileName') as similar INNER JOIN (SELECT * FROM reviews WHERE profileName = '\$username') as u ON similar.animeUid = u.animeUid ORDER by pengguna desc;"; \$getSimilarUserSkor = mysqli_query(\$con, \$sql) or die("Bad Query!" . \$sql); \$SimilarUserSkor = mysqli_fetch_all(\$getSimilarUserSkor); } </pre>

Penggalan kode diatas juga dapat mengatasi masalah yang timbul saat ada beberapa anime yang tidak sama-sama di rating oleh kedua pengguna. Daripada memilah nanti, kode ini hanya mengambil data rating dari anime yang sama dari kedua pengguna.

Tabel 4. Kode untuk Menentukan Ranking Data Rating

Penggalan Kode	
<pre> function rankify(\$arr){ \$t = 0; for (\$i = 0; \$i < count(\$arr); \$i++) { \$rank = \$i + 1; \$count = 1; \$skor = 0; for (\$j = \$i + 1; \$j <= count(\$arr); \$j++) { if (!isset(\$arr[\$j][1])) { if (\$count > 1) { \$skor += \$j; \$rank = \$skor / \$count; \$t += ((pow(\$count, 3) - \$count)) / 12; } break; } } } } </pre>	<pre> else { if (\$arr[\$j - 1][1] == \$arr[\$j][1]) { \$count++; \$skor += \$j; } else { if (\$count > 1) { \$skor += \$j; \$rank = \$skor / \$count; \$t += ((pow(\$count, 3) - \$count)) / 12; } break; } } for (\$k = \$i; \$k < \$i + \$count; \$k++) { \$arr[\$k][3] = \$rank; } \$i += \$count - 1; } \$arr[0]['t'] = \$t; return \$arr; } </pre>

$$\rho = 1 - 6 \left[\frac{\sum D_i^2 + \sum T_i}{n(n^2 - 1)} \right] \quad (1)$$

T_i didapatkan dari data dengan ranking yang sama dimana $T_i = \frac{m_i^2 - m_i}{12}$ (2). Misalkan data ranking ke 3, 4, 5 memiliki rating yang sama, maka $m_i = 3$. Mencari rating dengan ranking yang sama di terjemahkan menggunakan perulangan pada kode. Lalu nilai T langsung dijumlahkan pada variable \$t untuk mempersingkat kode.

D_i yang merupakan selisih dari ranking rating pengguna terhadap pengguna lain di terjemahkan ke dalam kode pada fungsi berikutnya. Yang mana pada fungsi diatas, hanya akan diambil jumlah dari kedua ranking terlebih dahulu. Setelah kedua jumlah rating didapatkan, maka pada function selanjutnya akan dihitung selisih sekaligus dikuadratkan.

Tabel 5. Kode Program untuk Menghitung Kemiripan Pengguna

Penggalan Kode
<pre>function similarity(\$arr1, \$arr2, \$t1, \$t2){ \$square = 0; \$n = count(\$arr1); foreach (\$arr1 as \$index => \$a) { \$d = pow((\$arr1[\$index][3] - \$arr2[\$index][3]), 2); \$square += \$d; } if (\$n == 1) { return 1; } return 1 - (\$square + \$t1 + \$t2) / (\$n * (pow(\$n, 2) - 1)); }</pre>

b. Proses Perhitungan Prediksi

Tabel 5. Kode Program Perhitungan Nilai Akhir Prediksi

Penggalan Kode
<pre>function predict(\$skors, \$similarity, \$skorAwal){ \$sum = 0; \$divider = 0; foreach (\$skors as \$index => \$s) { \$profileName = \$s[1]; \$skor = \$s[0]; \$key = array_search(\$profileName, array_column(\$similarity, 'nama')); \$spearman = \$similarity[\$key]['spearman']; if (\$spearman > 0) { \$sum += \$skor * \$spearman; \$divider += \$spearman; } } if (\$sum === 0 \$divider === 0) { return \$skorAwal; } \$predictionSkor = \$sum / \$divider; return \$predictionSkor; }</pre>

Dalam perhitungan prediksi sebenarnya cukup sederhana, cara yang digunakan yaitu *weighted average* akan mencari rata-rata nilai dengan menggunakan semua aspek yang telah didapatkan seperti similarity dan rating pengguna lain.

Untuk mendapatkan hasil prediksi dan hasil rekomendasi, penulis menggunakan dua buah query yang berbeda untuk mengambil data animeUid. Pengujian akan dilakukan terhadap animeUid yang sebelumnya sudah pernah di-rating oleh pengguna, sedangkan untuk hasil rekomendasi data rating anime yang digunakan adalah rating yang dilakukan oleh pengguna lain yang memiliki kemiripan terhadap pengguna.

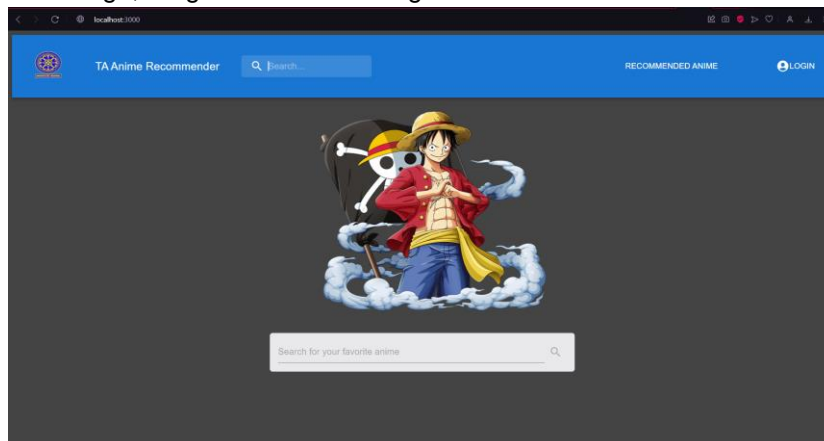
Tabel 5. Kode Program Query untuk Pengambilan Data Prediksi dan Rekomendasi

Penggalian Kode	
Query untuk Prediksi	Query untuk Rekomendasi
<pre> \$slq = "SELECT a.animeUid, a.skor, u.hitung FROM (SELECT * FROM reviews WHERE profileName = '\$username') as a INNER JOIN (SELECT animeUid, COUNT(animeUid) AS hitung from reviews GROUP by animeUid HAVING COUNT(animeUid) ORDER BY COUNT(animeUid) DESC) as u ON a.animeUid = u.animeUid ORDER BY u.hitung DESC LIMIT 50;"; </pre>	<pre> \$sql = "SELECT animeUid, COUNT(animeUid) FROM reviews WHERE profileName IN ('\$profileNames') AND animeUid NOT IN (SELECT animeUid FROM reviews WHERE profileName ='\$username') GROUP BY animeUid HAVING COUNT(animeUid) ORDER BY COUNT(animeUid) DESC LIMIT 50;"; </pre>

Pada penggalan kode disebelah kiri, query akan mengembalikan data berupa animeUid yang diurutkan berdasarkan seberapa banyak pengguna lain yang juga melakukan rating terhadap anime tersebut. Sedangkan dibagian kanan menunjukkan query yang akan mengembalikan data animeUid yang dirating oleh pengguna lain namun bukan oleh pengguna yang aktif.

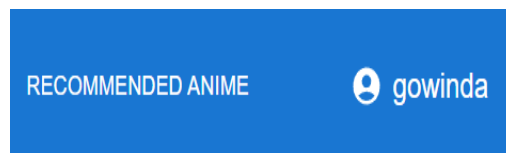
c. Implementasi Website

Sebuah sistem biasanya terdiri dari backend dan frontend yang disatukan menjadi satu kesatuan. Frontend atau tampilan antarmuka berperan sebagai jembatan antara pengguna dengan sistem yang berjalan dibelakang layar. Dalam sistem rekomendasi ini, penulis membuat sebuah website sederhana yang dapat menjalankan fungsi utama sistem yaitu memberikan rekomendasi kepada pengguna. Website ini memiliki beberapa komponen yaitu Home Page, Login-Register, Search, Recommend, Result Page, Single View dan Rating.

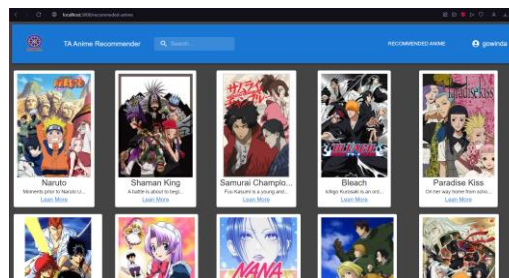
**Gambar 9.** Tampilan Home Page

Gambar 10. Tampilan *Login-Register*

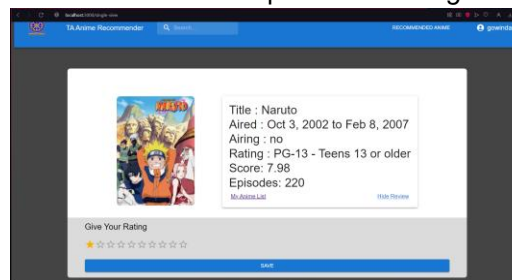
Gambar 11. Fitur Search



Gambar 12. Fitur Recommend



Gambar 13. Tampilan *Result Page*



Gambar 14. Tampilan *Single View dan Rating*

3.3 Pengujian

Pengujian pada sistem dilakukan menggunakan perhitungan mean absolute error (MAE). Pengujian akan dilakukan pada 50 pengguna yang memiliki riwayat rating paling banyak sehingga kecil kemungkinan data kosong dan pengujian yang dapat lebih akurat. Pengujian dilakukan menggunakan bantuan aplikasi Postman untuk melakukan request pada server. Namun sebelumnya harus ada beberapa modifikasi pada kode di backend. Perubahan hanya dilakukan untuk menampilkan hasil yang berbeda dari hasil rekomendasi. Seperti pada tabel 4.6, kode disebelah kiri akan diaktifkan sedangkan yang disebelah kanan akan di nonaktifkan.

Sistem Rekomendasi Seri Animasi Jepang (Anime) Menggunakan *User-Based Collaborative Filtering* dan *Spearman Rank Correlation Coefficient*

Dari Postman, nantinya data akan disalin secara manual pada tabel Excel dan akan dilakukan perhitungan MAE. Berikut ini adalah hasil akhir dari 50 pengguna yang diuji.

no	profilname	MAE
1	xinil	1.32752079
2	literaturenerd	2.094495312
3	ktulu007	2.108382822
4	BabyGirl06301	1.355164941
5	Zaku88	1.082795686
6	Archaeon	0.894933222
7	kajia	1.311209434
8	Krunchyman	3.572438553
9	Simonian	1.66241778
10	TrashDax	1.014748348
11	Veronin	1.068579222
12	Ranivus	1.354837666
13	papsoshea	1.675214356
14	SovietWeeb	1.371619573
15	mistah-manila	1.587956984
16	FAKEANIMEGIRL	1.050451584
17	Gonzo-lewd	1.001953366
18	Firechick12012	1.151623631
19	Injenss	1.956222585
20	LaLeLuLiLo	2.073773165
21	AnimeBW	1.562359625
22	Retro8bit	1.387328626
23	Nicholaevich	1.311466542
24	earl_of_sandvich	1.001053626
25	ImRingo	0.914789076
26	jet2r0cks	0.973267187
27	Venneh	1.047435141
28	ProfaneValkyrie	1.05242095
29	kokuborou	1.125687797
30	gwern	1.057150817
31	Sidewinder51	0.757609656
32	Stark700	1.175284885
33	LegendAqua	1.188668901
34	ggultra2764	1.14505729
35	PyraXadon	1.097216711
36	BanjoTheBear	1.28742513
37	Angelreview	1.279324715
38	Karhu	2.183493993
39	Eouama	0.486856516
40	TakamakiJoker	1.853670093
41	Mater10K	0.980556114
42	EggheadLuna	2.031954802
43	LeaOotori	1.3264048
44	Luquillo	1.705135271
45	ZephSilver	1.430259391
46	mercury1980	1.857674838
47	camay1997	1.770621327
48	Shura-shurato	1.661051153

49	ParaParaJMo	1.716353011
50	ItIsIDio	2.369690121
Mean		1.429072142
Persentase Error		15.87857936
Akurasi Prediksi		84.12142064

Dari kemungkinan nilai terkecil 1 dan terbesar 10, error prediksi paling tinggi adalah 9. Jadi, untuk mengitung persentase error digunakan rumus:

$$\text{persentase error} = \left(\frac{\text{rata-rata error}}{9} \right) \times 100 \quad (1)$$

Hasil yang dari pengujian menunjukkan jika persentase error yang didapat adalah sebesar 15,8% dan hasil akurasi prediksi yang didapat sebesar 84,12%. Nilai MAE terkecil dari ke 50 profile adalah 0,894933222 oleh Archaeon dan nilai MAE tertinggi adalah 3,572438553 oleh Krunchyman. Dalam kasus Krunchyman, peneliti mengamati jika skor rating yang diberikan oleh pengguna sangatlah drastis terhadap anime yang berbeda. Hal ini merupakan salah satu kekurangan yang dimiliki metode Spearman Rank Correlation Coefficient. Karena mengabaikan nilai absolut dari skor dan menyederhanakannya menjadi nilai ranking, perhitungan prediksi mungkin akan sedikit melenceng pada kasus ekstrem seperti ini. Namun dengan data yang memadai, error tidak akan begitu jauh dari yang seharusnya.

3.3.1 Masalah Penggunaan Metode Spearman Rank Correlation Coefficient

Dalam implementasinya, penggunaan metode Spearman dalam mencari anggota-anggota pengguna yang memiliki kesamaan memiliki masalah tersendiri bagi kasus-kasus tertentu. Penggunaan metode Spearman yang mengubah nilai sebenarnya dari skor rating dan menyederhanakannya kedalam bentuk rangking-rangking memang dapat meningkatkan efektifitas sistem namun dapat mempengaruhi hasil akhir dari proses prediksi.

Penilaian yang ekstrem ini bisa menjadi suatu masalah jika didapatkan adanya kemiripan oleh metode spearman namun pengguna lain lebih bermurah hati dalam memberikan skor rating sehingga nilai akhir prediksi dapat sedikit melenceng. Pada kasus pengguna Krunchyman, nilai akhir MAE yang didapatkan adalah 3.572439 atau sekitar 39,69% tingkat error.

4. Kesimpulan

Berdasarkan hasil yang telah diperoleh pada penelitian, maka dapat ditarik kesimpulan sebagai berikut:

1. Sistem rekomendasi yang dibuat dengan menggunakan Collaborative filtering memiliki dua tahapan proses, yang pertama adalah mencari anggota pengguna lain yang memiliki kemiripan dengan pengguna yang sedang aktif. Pada tahapan ini, peneliti menggunakan metode Spearman Rank Correlation Coefficient. Lalu tahapan kedua adalah mencari prediksi skor rating dengan metode weighted sum average pada penelitian. Dengan menggabungkan kedua tahapan ini, penelitian menghasilkan sistem dengan tingkat akurasi 84,12% yang dihitung dengan menggunakan MAE pada sampel 50 pengguna. Dengan rata-rata nilai MAE sebesar 1,429072142 terhadap 50 pengguna, didapatkan tingkat error sebesar 15,8% yang dapat dikatakan cukup rendah untuk sebuah hasil prediksi. Nilai MAE terkecil dari ke 50 pengguna adalah 0,894933222 oleh Archaeon dan nilai MAE tertinggi adalah 3,572438553 oleh Krunchyman.
2. Dalam sistem rekomendasi dengan *collaborative filtering*, dengan fokus membandingkan hasil rating pengguna, tentunya pola nilai rating yang akan menentukan hasil pencarian kesamaan. Hal ini dapat menyebabkan terjadinya kendala dalam kasus ekstrem saat pengguna memberikan nilai skor rating yang terlalu signifikan pada seri yang disukai atau tidak disukai. Contohnya ada pada pengguna dengan nama Krunchyman yang memberikan rating 1 pada sebagian besar anime yang kurang disukai dan nilai 10 ada anime yang disukai. Jika nanti didapati ada kemiripan terhadap urutan rangking skor namun nilai skornya sendiri jauh melenceng karena pengguna lain lebih bermurah hati dalam memberikan rangking, maka nilai akhir prediksi akan sedikit melenceng dan dalam kasus pengguna Krunchyman nilai MAE yang didapatkan adalah 3.572439 atau sekitar 39,69% tingkat error. Penulis menyarankan agar penelitian selanjutnya dapat menambahkan faktor lain dalam sistem rekomendasi selain perbandingan rating.

References

- [1] Safariani, P. (2017) 'Penyebaran Pop Culture Jepang Oleh Anime Festival Asia (AFA) Di Indonesia Tahun 2012-2016', *Ejournal.Hi.Fisip-Unmul.Ac.Id*, 5(3), pp. 729–744.
- [2] Schedule – Anime - MyAnimeList.net (2022). Available at: <https://myanimelist.net/anime/season/schedule> (Accessed: 22 September 2022).
- [3] Schafer, J. Ben et al. (2007) 'Collaborative Filtering Recommender Systems', pp. 291–324.
- [4] Moh. Irfan, Andharini Dwi C, Fika Hastarita R. (2014), 'SISTEM REKOMENDASI: BUKU ONLINE DENGAN METODE COLLABORATIVE FILTERING', 7(1), pp. 76–84.
- [5] Lam, X. N. (2008) 'Addressing Cold-Start Problem in Recommendation Systems', pp. 208–211.
- [6] Winter, J. C. F. De, Gosling, S. D. and Potter, J. (2016) 'Supplemental Material for Comparing the Pearson and Spearman Correlation Coefficients Across Distributions and Sample Sizes: A Tutorial Using Simulations and Empirical Data', *Psychological Methods*, 21(3), pp. 273–290. doi: 10.1037/met0000079.supp.