Pengembangan Model Rekomendasi Anime Dengan Metode Deep Learning: LSTM Neural Network

Fatimah Nur Alifiah[1], Yuni Yamasari[2]

[1,2] Jurusan Teknik Informatika/Teknik Informatika, Universitas Negeri Surabaya
[1]fatimah.21022@mhs.unesa.ac.id
[2]yuniyamasari@unesa.ac.id

***Abstrak*— Dalam era digital yang sarat dengan informasi dan pilihan konten yang melimpah, pengguna sering kali mengalami kesulitan dalam menemukan tontonan yang benar-benar sesuai dengan minat dan preferensi mereka. Permasalahan ini juga terjadi dalam konteks tontonan anime, di mana ratusan hingga ribuan judul tersedia setiap tahunnya. Untuk mengatasi tantangan tersebut, penelitian ini membangun model rekomendasi dengan algoritma deep learning LSTM (Long Short-Term Memory) yang mampu memahami urutan data, sehingga cocok digunakan untuk menganalisis pola interaksi pengguna dari waktu ke waktu yang nantinya cocok untuk model rekomendasi. Model ini dirancang dengan mempertimbangkan berbagai informasi penting, seperti identitas pengguna, genre anime, skor atau rating yang diberikan pengguna, serta tingkat popularitas dari masing-masing anime. Proses pelatihan dilakukan menggunakan teknik *k-fold cross validation* dimana hasil evaluasi menunjukkan performa model yang sangat baik. Model mampu menghasilkan akurasi prediksi sebesar 91,3%, yang mengindikasikan bahwa sebagian besar rekomendasi yang dihasilkan sesuai dengan nilai aktualnya. Selain itu, model juga menunjukkan nilai Mean Squared Error (MSE) yang sangat kecil, yaitu sebesar 0,0002. Nilai ini mengindikasikan bahwa hasil prediksi model sangat mendekati nilai sebenarnya, sehingga dapat dikatakan bahwa tingkat akurasi model tergolong tinggi. Di sisi lain, nilai Mean Absolute Error (MAE) juga menunjukkan hasil yang baik, yaitu sebesar 0,008, yang berarti rata-rata selisih antara nilai prediksi dan nilai aktual tergolong kecil, sehingga menunjukkan tingkat kesalahan yang rendah dalam prediksi. Dengan hasil ini, dapat disimpulkan bahwa model rekomendasi berbasis LSTM memiliki kemampuan yang baik dalam memahami preferensi pengguna dan memberikan saran anime yang sesuai dengan minat mereka.**

***Kata Kunci*— *Model Rekomendasi, LSTM, Deep Learning, Collaborative Filtering, Content-based Filtering, Anime*.**

I. PENDAHULUAN

Pada saat ini, perkembangan teknologi sangat pesat, yang tentunya berkaitan dengan lonjakan data informasi yang terus meningkat seiring bertambahnya tahun. Bahkan tercatat pada Statista.com, lonjakan data diperkirakan akan meningkat hingga 181 *zetabyte* pada tahun 2025. Di era internet yang terus berkembang seperti sekarang, kemudahan mengakses berbagai informasi menjadi hal yang sangat diinginkan oleh banyak orang. Hal ini tercermin dari data pengguna internet Indonesia yang mencapai 212,35 juta jiwa pada Maret 2021, menempatkan Indonesia di urutan ketiga dengan pengguna internet terbanyak di Asia. Selain itu, Data Digital 2021 Indonesia pada Januari menunjukkan bahwa 98,5% pengguna internet berusia 16 hingga 64 tahun menggunakan internet untuk aktivitas menonton video daring [1]. Namun, kemudahan akses ini memiliki dampak negatif, seperti kebingungan akibat terlalu banyaknya informasi yang bertebaran di internet, sehingga menyulitkan orang dalam menentukan pilihan suatu item atau variasi. Fenomena ini menimbulkan masalah baru, sehingga para peneliti menyadari pentingnya adanya suatu model yang dapat mempermudah pengguna dalam menentukan item atau variasi sesuai dengan kebutuhan dan preferensi mereka. Hal ini mendorong terciptanya model rekomendasi yang kini diterapkan di berbagai bidang, termasuk sektor hiburan. Namun pengembangan rekomendasi ini tentunya diperlukan model sebagai dasar utama.

Pembangunan model rekomendasi yang diterapkan pada platform-platform ini membantu pengguna menemukan hiburan sesuai preferensi mereka, sekaligus menciptakan ruang interaksi yang memperkaya pengalaman hiburan secara digital. Munculnya berbagai macam *platform streaming* dan aplikasi yang digunakan untuk *review* film atau daftar bacaan tentunya tidak lepas dari kebutuhan pengguna yang ingin mendapatkan hiburan secara mudah serta tempat untuk mengetahui hiburan yang sedang ramai diperbincangkan orang-orang. Salah satu hiburan yang saat ini sangat berkembang dan menyebar ke berbagai kalangan adalah anime. Anime adalah animasi yang dibuat dari negara Jepang dengan gaya penggambaran yang unik dibandingkan dengan animasi dari negara lain [2]. Bertambahnya tempat untuk menonton anime beriringan dengan munculnya bebagai *platform* yang menyediakan tempat *review* anime, pada *platform* tersebut pengguna dapat melihat berbagai macam jenis anime dan memberikan *rating* sesuai dengan preferensi pengguna. Pengguna juga bisa berinteraksi dengan pengguna lain dan bertukar opini tentang anime pada *platform* tersebut. Salah satu *platform* untuk *review* anime adalah MyAnimeList.

MyAnimeList sendiri merupakan *platform* untuk *review* anime yang banyak digunakan oleh para penggemar anime sendiri [3]. Meski begitu rekomendasi anime di *platform* tersebut tidak banyak menyesuaikan karakteristik pengguna masing-masing, sehingga rekomendasi anime yang diberikan masih cukup general dan tidak spesifik untuk tiap pengguna. Pengguna dapat berinteraksi dengan sesama pengguna lainnya dan saling memberikan rekomendasi anime yang menurut pengguna tersebut bagus, namun sayangnya dikarenakan jenis dan genre anime yang beragam tidak jarang rekomendasi anime dari pengguna lain dirasa tidak sesuai. Oleh karenanya, diperlukan digitalisasi dalam

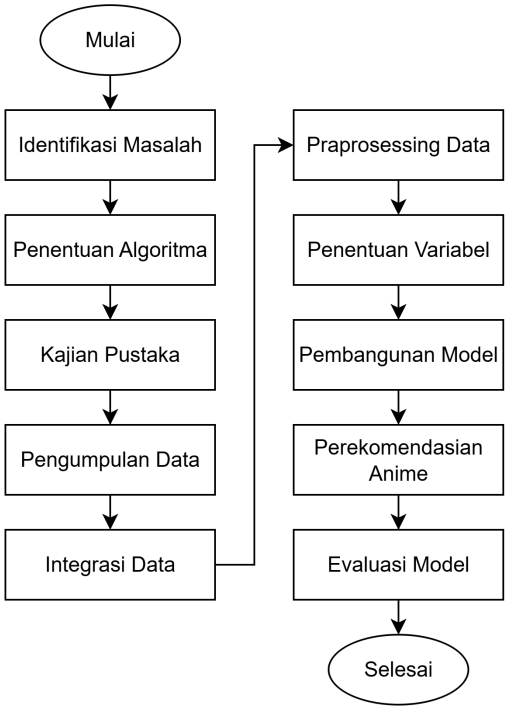
proses merekomendasikan suatu item atau variasi yang cocok dengan karakteristik pengguna, terutama dalam sektor hiburan, khususnya anime.

Beberapa penelitian sebelumnya telah mengembangkan model rekomendasi anime dengan berbagai pendekatan. Model berbasis *collaborative filtering* menggunakan Pearson Correlation menunjukkan nilai MAE sebesar 2.94 [3]. Pendekatan *item-based* dengan *cosine similarity* terbukti lebih akurat dibandingkan *euclidean distance* [4]. Pada pendekatan konten, TF-IDF digunakan untuk mengekstrak informasi dari sinopsis, menghasilkan akurasi tinggi untuk pencarian tunggal [5]. Ada pula metode hybrid yang menggabungkan collaborative dan content-based filtering [6], serta model rekomendasi berbasis *user-based* dengan Spearman mencapai akurasi hingga 84% [7].

Dalam konteks model rekomendasi, LSTM mampu menganalisis pola interaksi antara pengguna dan genre secara mendalam. Keunggulan LSTM dalam memahami urutan data memungkinkan model rekomendasi untuk tidak hanya mempertimbangkan genre yang disukai, tetapi juga perubahan preferensi pengguna dari waktu ke waktu. Sehingga, pada penelitian kali ini penulis akan membangun model rekomendasi anime menggunakan metode deep learning dengan algoritma LSTM neural network. Penulis berharap melalui penelitian ini dapat memberikan model rekomendasi yang sesuai dengan keinginan dan nantinya bisa digunakan.

II. METODOLOGI PENELITIAN

Metodologi yang dilakukan pada penelitian untuk pengembangan model perekomendasian anime menggunakan deep learning ini dapat dilihat pada gbr. 1.



Gbr. 1 Rangkaian metodologi penelitian.

A. *Identifikasi Masalah*

Pengguna sering mengalami kebingungan memilih anime karena banyaknya pilihan yang tersedia secara daring. Perekomendasian anime yang ada seperti di MyAnimeList masih bersifat umum dan belum mampu menyesuaikan dengan preferensi pengguna.

B. *Penentuan Algoritma*

Penelitian ini menggunakan LSTM karena kemampuannya dalam mengenali pola pada data sekuensial, seperti interaksi pengguna dan genre anime. LSTM dinilai lebih bagus dan handal dalam menangkap preferensi pengguna dibanding metode lainnya. Keunggulan ini membuatnya cocok digunakan dalam pengembangan model rekomendasi.

C. *Kajian Pustaka*

1) Anime

Anime adalah sebuah animasi yang berasa dari jepang yang memiliki banyak *character* dengan berbagai warna, karakteristik, dan latar tempat yang berbeda. Dimana anime memiliki target penonton yang sangat beragam [8]

2) Collaborative Filtering

Collaborative filtering merupakan pendekatan yang merekomendasikan item berdasarkan kesamaan preferensi antar pengguna, teknik ini bekerja dengan cara membandingkan sejarah rating antar pengguna untuk menemukan kemiripan selera [9].

3) Content-based Filtering

Content-based filtering sendiri memanfaatkan informasi atau fitur dari suatu item, seperti genre, aktor, atau sinopsis dalam konteks film atau anime, untuk memprediksi keterkaitannya dengan preferensi pengguna. Sehingga model ini mampu untuk merekomendasikan item serupa yang disukai atau cenderung diberikan rating yang baik pada histori sebelumnya, berdasarkan kesamaan karakteristik konten tersebut dengan profil pengguna yang telah terbentuk [9].

4) Deep Learning

Deep learning adalah bagian dari machine learning yang dianggap sebagai teknologi revolusioner dengan dampak besar, layaknya internet. Teknologi ini sering digambarkan mampu membantu bahkan menggantikan peran manusia dalam berbagai pekerjaan. Hal ini dikarenakan kemampuan model program yang mempelajari pola yang kompleks dengan jaringan tiruan selayaknya jaringan kerja otak manusia [10].

5) LSTM Neural Network

LSTM dirancang untuk mengingat informasi jangka panjang dengan menggunakan arsitektur sel memori yang terdiri dari input gate, forget gate, dan output gate. Input gate memilih informasi yang akan diperbarui, forget gate melupakan informasi lama, dan output gate menentukan keluaran sel. LSTM menjaga aliran informasi melalui cell state yang menghubungkan kondisi sebelumnya dan saat ini, memungkinkan penyimpanan konteks penting. Gerbang sigmoid digunakan untuk mengatur apakah informasi diteruskan atau dihentikan, dengan output antara nol dan satu yang dikalikan dengan nilai lain untuk menentukan pengaruhnya pada proses selanjutnya [11].

D. *Pengumpulan Data*

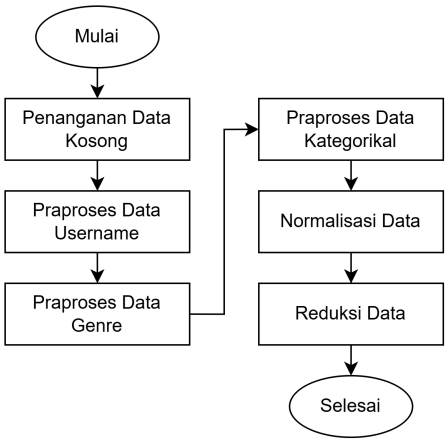
Pada penelitian ini, data yang digunakan berasal dataset publik yang didapatkan pada laman Kaggle, dimana dataset ini dikumpulkan dan dipublikasikan oleh Sajid. Dataset ini berjudul Anime Dataset 2023 dan di dalamnya memiliki anime dataset dengan *content* yang berbeda.

E. *Integrasi Data*

Dataset anime-dataset-2023 berisi 24.905 data anime dengan 24 kolom, yang mencakup informasi seperti anime_id, Name, Genres, Synopsis, Score, Studios, Source, hingga Image URL. Dataset ini menyediakan deskripsi konten yang lengkap namun tidak memiliki riwayat interaksi pengguna. Sebaliknya, final_animedataset berisi 35 juta lebih data yang merekam interaksi antara pengguna dan anime, dengan 13 kolom seperti username, anime_id, my_score, title, genre, dan lainnya. Dataset ini memungkinkan analisis preferensi pengguna berdasarkan histori tontonan. Untuk keperluan penelitian, digunakan final_animedataset sebagai dataset utama karena mengandung informasi tentang perilaku pengguna terhadap anime tertentu. Dua kolom tambahan yaitu Synopsis dan Image URL diambil dari anime-dataset-2023 lalu ditambahkan ke final_animedataset guna memperluas informasi konten anime yang dianalisis.

F. *Praprosessing Data*

Tahapan ini penting dilakukan supaya dataset siap digunakan untuk melakukan pemodelan, berikut pada Gbr. 2, adalah beberapa tahapan yang diperlukan saat melakukan preprocessing data.



Gbr. 2 Rangkaian praprose data.

1) Penanganan data kosong

Pada tahap ini dilakukan pengecekan data yang kosong dan duplikat, karena missing value dapat memengaruhi akurasi analisis dan kinerja model. Ditemukan missing value pada kolom username (256), rank (751.970), genre (2.267), synopsis dan Image URL (masing-masing 34.001).

2) Praproses data username

Selanjutnya dilakukan encoding pada ‘username’ dengan mengubah data string menjadi nilai numerik agar dapat diproses oleh model deep learning. Misalnya, *Karthiga*, *Areq*, dan *Leyzo* diubah menjadi [0, 1, 2]. Hasil encoding ini digunakan sebagai input pada pemodelan model rekomendasi.

3) Praproses data genre

Tokenisasi genre dilakukan untuk mengubah data string menjadi format numerik agar dapat diproses oleh model deep learning. Proses ini melibatkan split genre dan pemberian token numerik. Karena panjang genre bervariasi, dilakukan penyeragaman panjang sequence sesuai nilai maksimum genre yang ada.

4) Praproses data kategorikal

Proses one-hot encoding dilakukan pada kolom type, source, dan gender karena merupakan data kategorikal. Kolom source berisi sumber cerita seperti Manga, Light novel, dan Original, sedangkan type mencakup jenis anime seperti TV, Movie, dan OVA. Nilai kategori tersebut diubah menjadi format numerik agar dapat digunakan dalam pemodelan.

5) Normalisasi data

Normalisasi dilakukan pada kolom my_score dan rank. Skor dinormalisasi menggunakan StandardScaler agar memiliki distribusi dengan mean 0 dan standar deviasi 1, sehingga meminimalkan perbedaan penilaian antar pengguna. Untuk kolom rank, digunakan pendekatan nilai

negatif agar anime dengan peringkat terbaik memiliki nilai normalisasi yang lebih tinggi.

6) Reduksi Data

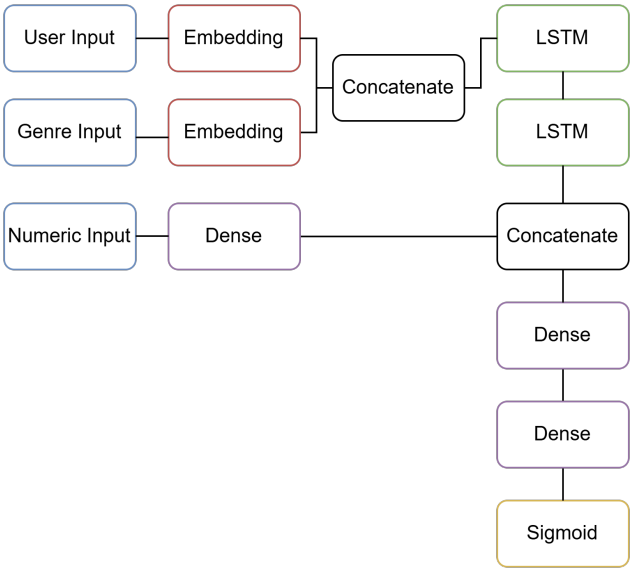
Dalam penelitian ini, penulis menggunakan 200 username pertama yang tersedia dalam dataset sebagai sampel data. Pemilihan ini menghasilkan total sebanyak 98.604 baris data dengan 15 kolom fitur yang akan dianalisis lebih lanjut dalam proses penelitian.

G. *Penentuan Variabel*

Penelitian ini menggunakan my_score sebagai label, yaitu skor dari pengguna terhadap anime dalam skala 1–10. Dalam penelitian ini, dilakukan pemetaan terhadap variabel-variabel yang dikelompokkan ke dalam dua pendekatan utama, yaitu *content-based filtering* dan *collaborative filtering*.

H. *Pembangunan Model*

Penelitian ini menggunakan metode deep learning dengan algoritma LSTM. Dengan arsitektur ini, model dapat menangkap pola dari aspek collaborative filtering maupun content-based filtering secara lebih menyeluruh. Adapun visualisasi struktur deep learning ini disajikan pada Gbr. 3.



Gbr. 3 Struktur arsitekur layer deep learning

1) Input Model

Input model pada penelitian ini menggunakan variabel yang termasuk content-based filtering dan collaborative filtering. Model ini menerima tiga jenis input utama yang mewakili berbagai aspek informasi dari pengguna dan anime. Pertama, user_input merepresentasikan username pengguna yang telah diubah ke dalam bentuk numerik melalui proses encoding. Kedua, genre_input berisi token angka dari genre anime yang mencerminkan karakteristik konten dari setiap anime. Ketiga, numeric_input mencakup fitur-fitur numerik seperti skor komunitas, peringkat, popularitas, scored_by, serta atribut kategori seperti type, source, dan genre, yang telah diproses sebelumnya menggunakan teknik one-hot encoding atau normalisasi.

2) Embedding

Setiap input kategorikal diubah menjadi representasi vektor melalui layer embedding. Representasi dari pengguna dan genre kemudian digabungkan menjadi satu sekuens vektor yang mencerminkan kombinasi antara karakteristik pengguna dari pendekatan collaborative filtering dan informasi konten dari pendekatan content-based filtering..

3) LSTM Layer

LSTM pertama memproses sekuens dan meneruskan seluruh hasilnya ke LSTM kedua, yang kemudian merangkum informasi tersebut menjadi satu representasi akhir. Representasi ini kemudian disesuaikan bentuknya agar dapat digabungkan dengan input lain. Melalui proses ini, model dapat memahami pola urutan antar genre serta hubungannya dengan karakteristik pengguna..

4) Dense Layer

Vektor hasil penggabungan diproses lebih lanjut melalui beberapa layer Dense untuk memperdalam pemahaman model terhadap pola data. Sementara itu, fitur numerik diproses secara dengan aktivasi ReLU, disertai teknik normalisasi dan regularisasi untuk menjaga kestabilan dan mencegah overfitting. Setelah semua informasi diproses, hasil dari bagian LSTM dan fitur numerik digabungkan dan diproses kembali melalui layer tambahan.

5) Output Model

Output Model menghasilkan output berupa satu nilai dengan fungsi aktivasi sigmoid. Nilai ini merepresentasikan probabilitas apakah pengguna akan menyukai anime tertentu, dan digunakan sebagai dasar perekomendasian.

I. *Proses Rekomendasi Anime*

Pada penelitian ini dihasilkan output rekomendasi anime yang diberikan untuk username yang ada pada dataset, perekomendasian ini didasarkan atas histori preferensi pengguna pada anime tertentu dari data yang ada. model terlebih dahulu memeriksa data pengguna untuk mengidentifikasi anime yang sudah ditonton dan diberi rating. Dari data ini, dihitung genre yang cenderung disukai pengguna berdasarkan skor yang diberikan. Pada model proses ini dinamakan pembuatan profil pengguna [12]. Model lalu membandingkan daftar tersebut dengan anime yang belum ditonton untuk menyaring rekomendasi, yang termasuk dalam pendekatan content-based filtering.

$$avg_{score} = \frac{total_{score}}{count} \quad (1)$$

Dimana total_score adalah jumlah seluruh rating yang diberikan pengguna untuk anime dengan genre tertentu, sedangkan count adalah jumlah genre tersebut terhitung.

Melalui Tabel I, merupakan contoh sampel hasil perhitungan profil genre pengguna untuk user salah satu user dimana di penelitian ini disebut sebagai pengguna A. Dalam pembuatan profil genre pengguna ini, nantinya divisualisasikan perbandingan frekuensi atau kecenderungan genre yang ditonton oleh pengguna melalui wordcloud.

TABEL I
PERHITUNGAN PROFIL GENRE PENGGUNA A

Genre	Jumlah Anime	Score
Comedy	43	7.53
Romance	35	7.31
Drama	24	7.42
Achool	23	6.91
Shoujo	20	7.50
Shounen	12	8.17
Slice of Life	12	7.17
...	...	...
Kids	1	6.

Tahapan selanjutnya adalah memprediksi seberapa besar kemungkinan pengguna menyukai masing-masing anime tersebut, dengan digunakan model deep learning yang telah dilatih sebelumnya. Setelahnya, model menghasilkan nilai prediksi yang merepresentasikan tingkat kesukaan pengguna terhadap anime tersebut. Proses prediksi ini termasuk dalam collaborative filtering karena mempertimbangkan preferensi pengguna lain yang memiliki kesamaan. Setelah prediksi didapat, model mengurutkan anime berdasarkan beberapa kriteria sebelum dilakukan pemberian rekomendasi anime untuk pengguna, dimana kriterianya adalah sebagai berikut:

- Belum ditonton oleh pengguna
- Prediksi skor tinggi berdasarkan preferensi pengguna
- Rating umum yang baik *(score)*
- Peringkat yang bagus *(rank)*

J. *Evaluasi*

Penelitian ini menggunakan metode K-Fold Cross Validation untuk menguji performa model. Data dibagi menjadi K bagian, di mana setiap iterasi menggunakan satu fold sebagai data uji dan sisanya sebagai data latih. Dalam penelitian ini, Mean Squared Error (MSE) digunakan sebagai fungsi loss utama karena mampu mengukur seberapa besar perbedaan antara hasil prediksi model dan nilai sebenarnya. Selain itu, Mean Absolute Error (MAE) dan akurasi juga digunakan sebagai metrik tambahan untuk mengevaluasi performa model secara menyeluruh.

III. HASIL DAN PEMBAHASAN

Bagian ini membahas hasil dari serangkaian metode yang telah dijalankan dalam penelitian ini. Pembahasan mencakup hasil dari proses pelatihan, evaluasi, serta keluaran rekomendasi yang dihasilkan oleh model. Analisis dilakukan untuk memahami performa model serta relevansi rekomendasi yang diberikan kepada pengguna.

A. *Deskripsi Data*

Berdasarkan data yang digunakan pada penelitian ini, dilakukan pengklasifikasian untuk membedakan variabel mana saja yang termasuk *content-based filtering* dan *collaboratif filtering* [9]. Adapun pada data ini kolom-kolom yang termasuk dalam *content-based filtering*, seperti ‘anime_id’, ‘title’, ‘type’, ‘source’, ‘score’, ‘scored_by’, ‘rank’, ‘popularity’, ‘genre’, dan ‘sinopsys’, yang kemudian diolah untuk memahami kesesuaian antara karakteristik anime dan preferensi pengguna. Sedangkan kolom-kolom yang mendukung pendekatan *collaborative* antara lain adalah `username`, `user_id`, `my_score`, dan `gender`, karena kolom-kolom tersebut menggambarkan interaksi langsung antara pengguna dan item yang dinilai.

B. *Hasil Data Preprocessing*

Data dalam penelitian ini berasal dari gabungan dua dataset, sehingga perlu dilakukan proses *merge* terlebih dahulu. Setelahnya, dilakukan pengecekan dan penanganan *missing value*, kolom seperti genre dan sinopsis yang kosong diisi dengan "Unknown", sedangkan kolom peringkat (rank) yang kosong diisi dengan 0. Untuk kolom penting seperti username, jika ditemukan kosong maka baris terkait dihapus. Selanjutnya setelah menangani missing value ini kemudian penulis penulis menggunakan data 200 username pertama untuk penelitian ini sehingga total data yang digunakan pada penilian ini sebanyak 98604 baris dengan 15 kolom.

Normalisasi dilakukan pada dua aspek utama, yaitu skor penilaian pengguna (‘my_score’) dan peringkat popularitas anime (‘rank’). Skor dinormalisasi menggunakan StandardScaler agar memiliki distribusi dengan mean 0 dan standar deviasi 1, membantu model menangani perbedaan skala antar pengguna. Sementara itu, normalisasi rank dilakukan dengan pendekatan nilai negatif, sehingga anime dengan peringkat lebih baik (angka kecil) akan memiliki nilai normalisasi yang lebih tinggi.

Pada tahap selanjutnya, data genre yang berbentuk teks diubah menjadi format numerik agar dapat diproses oleh model. Setiap genre dipisahkan berdasarkan koma, kemudian diubah menjadi token angka menggunakan *Tokenizer* dari Keras. Daftar genre yang telah ditokenisasi lalu dikonversi menjadi urutan angka (sequence) sesuai penetapan. Karena jumlah genre tiap data berbeda, ditentukan panjang maksimum, lalu dilakukan *padding* agar semua urutan memiliki panjang yang sama dengan menambahkan nol di bagian akhir.

C. *Model*

Dalam eksperimen ini, model dilatih menggunaankan hyperparameter yang bervariasi. Kombinasi hyperparameter yang digunakan dapat dilihat pada Tabel II, dengan variasi jumlah epoch, dan nilai learning rate. Pemilihan konfigurasi dilakukan secara bertahap, untuk melihat pengaruhnya terhadap performa model. Optimizer yang digunakan adalah Adam, dan fungsi loss yang digunakan adalah Mean Squared Error (MSE). Seluruh proses pelatihan diimplementasikan menggunakan framework TensorFlow.

TABEL II
RANGKAIAN TRAINING MODEL

Uji Ke-	Fold	Epoch	Batch Size	Learning Rate
1	3	10	64	0.001
2	3	10	64	0.0001
3	3	10	64	0.00001
4	5	10	64	0.001
5	5	10	64	0.0001
6	5	10	64	0.00001
7	5	30	64	0.001
8	5	30	64	0.0001
9	5	30	64	0.00001
10	10	10	64	0.001
11	10	10	64	0.0001
12	10	30	64	0.001
13	10	30	64	0.0001

D. *Hasil Pembangunan Model LSTM*

Setelah konfigurasi model ditetapkan, tahap selanjutnya adalah melakukan proses pelatihan untuk masing-masing skenario kombinasi hyperparameter yang telah dirancang. Proses training ini bertujuan untuk mengamati performa model terhadap data validasi, serta mengevaluasi pengaruh dari setiap kombinasi epoch, fold, dan learning rate terhadap hasil prediksi. Hasil pelatihan dicatat dan dianalisis untuk mengetahui konfigurasi mana yang memberikan kinerja terbaik pada model rekomendasi yang dibangun. Adapun hasil yang dicantumkan pada hasil training ini merupakan rerata dari tiap eksperimen yang dilakukan, Waktu di sini adalah lama waktu yanng dibutuhkan untuk *running* tiap eksperimen.

1) Eksperimen 3 Fold dan 10 Epoch

Pengujian dilakukan dengan 3-Fold Cross Validation, 10 epoch per fold, dan batch size 64 pada data berukuran 98.604 ditampilkan pada Tabel III. Hasil terbaik diperoleh pada learning rate 0.001 dengan akurasi 0.917, F1-score 0.915, presisi 0.843, recall 1.0, serta nilai loss dan MAE terendah (0.0004 dan 0.0131), dengan waktu pelatihan tercepat yaitu 14 menit. Sebaliknya, learning rate 0.0001 dan 0.00001 menghasilkan performa lebih rendah, sehingga learning rate 0.001 dianggap paling optimal.

TABEL III
EKSPERIMEN 3-FOLD, 10 EPOCH PER FOLD, DAN BATCH SIZE 64

Learning Rate	Akurasi	Loss	Mae	Waktu (menit)
0.001	0.917	0.0004	0.013	14
0.0001	0.868	0.0026	0.0343	16
0.00001	0.874	0.019	0.101	17

2) Eksperimen 5 Fold dan 10 Epoch

Pada pengujian 5-Fold Cross Validation dengan 10 epoch per fold dan batch size 64 pada data sebesar 98.604 ditampilkan pada Tabel IV, learning rate 0.001 memberikan hasil terbaik dengan akurasi rata-rata 0.911, F1-score 0.91, presisi 0.834, recall 1.0, serta loss dan MAE terendah (0.0003 dan 0.0117), meskipun waktu pelatihan mencapai 44 menit. Sementara itu, learning rate 0.0001 dan 0.00001 menunjukkan performa lebih rendah, sehingga 0.001 tetap menjadi pilihan paling optimal.

TABEL IV
EKSPERIMEN 5-FOLD, 10 EPOCH PER FOLD, DAN BATCH SIZE 64

Learning Rate	Akurasi	Loss	Mae	Waktu (menit)
0.001	0.911	0.0003	0.011	44
0.0001	0.872	0.0021	0.031	49
0.00001	0.891	0.0147	0.086	33

3) Eksperimen 5 Fold dan 30 Epoch

Pada pengujian 5-Fold Cross Validation dengan 30 epoch per fold dan batch size 64 pada 98.604 data ditampilkan pada Tabel V, learning rate 0.001 memberikan hasil terbaik: akurasi 0.913, F1-score 0.911, dan loss serta MAE terendah (0.0002 dan 0.0082), walaupun durasi pelatihan paling lama (122 menit). Learning rate 0.0001 dan 0.00001 menunjukkan performa lebih rendah, sehingga 0.001 tetap optimal meskipun waktu training lebih lama.

TABEL V
EKSPERIMEN 5-FOLD, 30 EPOCH PER FOLD, DAN BATCH SIZE 64

Learning Rate	Akurasi	Loss	Mae	Waktu (menit)
0.001	0.913	0.0002	0.008	122
0.0001	0.872	0.0009	0.018	80
0.00001	0.865	0.0067	0.055	104

4) Eksperimen 10 Fold dan 10 Epoch

Pada pengujian 10-Fold Cross Validation dengan 10 epoch per fold dan batch size 64 pada data sebanyak 98.604 ditampilkan pada Tabel VI, baik learning rate 0.001 maupun 0.0001 menghasilkan performa identik (akurasi 0.873, F1-score 0.875, recall 1.0, loss 0.0019, MAE 0.0295). Namun, 0.001 jauh lebih efisien karena hanya membutuhkan 61 menit, dibandingkan 288 menit untuk 0.0001. Dengan demikian, learning rate 0.001 tetap menjadi pilihan yang lebih optimal.

TABEL VI
EKSPERIMEN 10-FOLD, 10 EPOCH PER FOLD, DAN BATCH SIZE 64

Learning Rate	Akurasi	Loss	Mae	Waktu (menit)
0.001	0.908	0.0003	0.0108	61
0.0001	0.873	0.0019	0.0295	288

5) Eksperimen 10 Fold dan 30 Epoch

Pada pengujian 10-Fold Cross Validation dengan 30 epoch per fold dan batch size 64 pada data sebanyak 98.604, model dengan learning rate 0.0001 menunjukkan performa terbaik. Hasil evaluasi menunjukkan akurasi sebesar 0.886, presisi 0.796 ditampilkan pada Tabel VII, recall sempurna sebesar 1.0, F1-score 0.887, serta nilai loss dan MAE

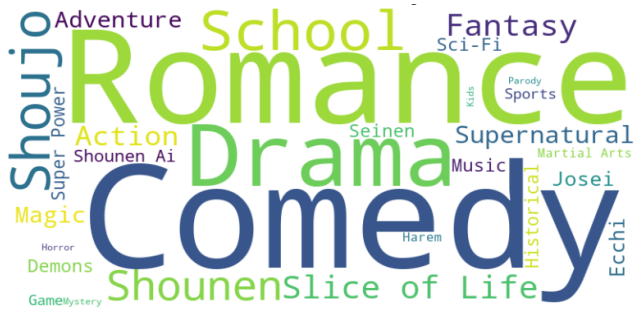
masing-masing 0.0008 dan 0.0178. Meskipun performanya unggul, proses pelatihan memerlukan waktu cukup lama, yaitu 310 menit.

TABEL VII
EKSPERIMEN 10-FOLD, 30 EPOCH PER FOLD, DAN BATCH SIZE 64

Learning Rate	Akurasi	Loss	Mae	Waktu (menit)
0.001	0.909	0.0002	0.0276	137
0.0001	0.886	0.0008	0.0178	310

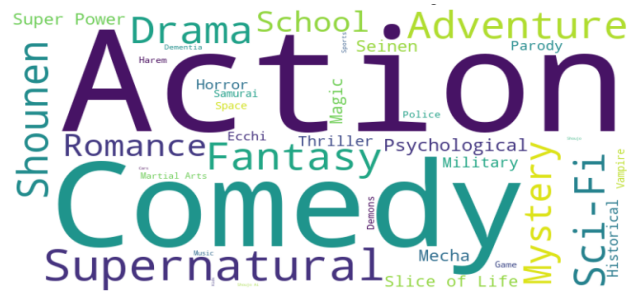
E. *Perekomendasian Anime*

Setelah model dinyatakan memiliki performa optimal, langkah selanjutnya adalah menerapkan model tersebut untuk memberikan rekomendasi anime kepada pengguna berdasarkan data histori yang ada. Pada tahap ini, penulis mengambil dua contoh pengguna dari dataset, yaitu pengguna A dan pengguna B, untuk melihat bagaimana model mengenali pola preferensi mereka. Visualisasi preferensi genre dari masing-masing pengguna ditampilkan pada Gambar 4.12 dan 4.13, yang menunjukkan genre-genre dominan yang cenderung disukai oleh masing-masing user



Gbr. 4 Representasi wordcloud pengguna A.

Melalui Gbr. 4, terdapat wordcloud yang merepresentasikan genre anime yang paling sering disukai oleh pengguna A.



Gbr. 6 Representasi wordcloud pengguna B.

Melalui Gbr. 5, terdapat wordcloud yang merepresentasikan genre anime yang paling sering disukai oleh pengguna B.

Perbedaan ini mencerminkan karakteristik pola preferensi kedua pengguna yang cukup berbeda, di mana pengguna A lebih menyukai genre slice-of-life dan romance, sedangkan pengguna B cenderung tertarik pada genre action, adventure, dan fantasi. Berdasarkan pola tersebut, model memberikan hasil rekomendasi anime yang sesuai preferensi masing-masing pengguna, yang ditampilkan pada bagian berikut.

TABEL VIII
REKOMENDASI ANIME UNTUK PENGGUNA A

#	Judul Anime	Skor Prediksi
1	Gintama.: Shirogane no Tamashii-hen	9.79
2	Gintama.	9.79
3	Shouwa Genroku Rakugo Shinjuu: Sukeroku Futatabi-hen	9.78
4	Natsume Yuujinchou Roku	9.78
5	Gintama.: Porori-hen	9.78

Pada Tabel VIII ditampilkan lima judul anime teratas yang direkomendasikan dari model untuk pengguna A. Judul-judul tersebut merupakan anime dengan skor prediksi tertinggi, yang menunjukkan tingkat kemungkinan tertinggi bahwa pengguna A akan menyukai anime tersebut. Semakin tinggi skor prediksi, semakin besar peluang anime tersebut sesuai dengan selera pengguna.

TABEL IX
REKOMENDASI ANIME UNTUK PENGGUNA B

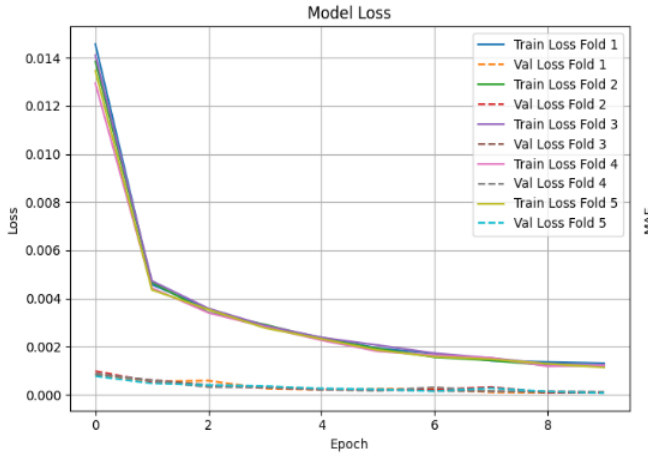
#	Judul Anime	Skor Prediksi
1	Shouwa Genroku Rakugo Shinjuu: Sukeroku Futatabi-hen	9.78
2	Natsume Yuujinchou Roku	9.78
3	Junjou Romantica Special	9.78
4	Natsume Yuujinchou Go	9.78
5	Natsume Yuujinchou Shi	9.78

Pada Tabel IX ditampilkan lima judul anime teratas yang direkomendasikan dari model untuk pengguna B. Judul-judul tersebut memperoleh skor prediksi tertinggi berdasarkan hasil perhitungan model, yang mencerminkan tingkat relevansi dan kesesuaian dengan preferensi pengguna. Skor prediksi yang tinggi ini menunjukkan bahwa model memperkirakan pengguna B akan sangat menyukai anime-anime tersebut.

F. *Pembahasan*

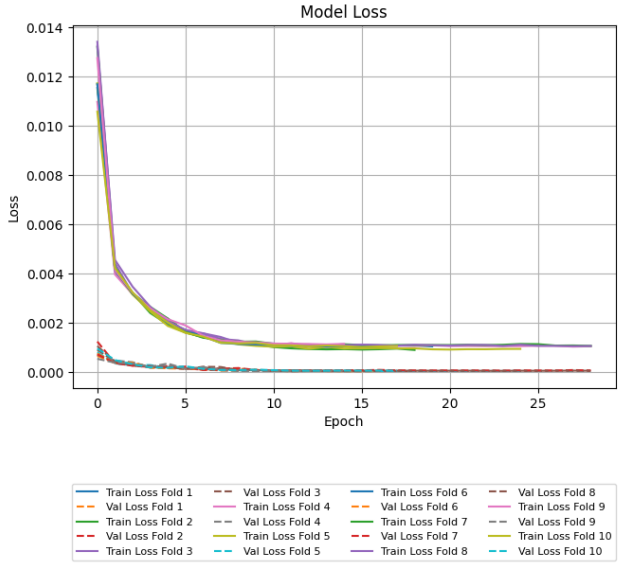
Berdasarkan hasil eksperimen yang dilakukan pada model ini, konfigurasi dengan kombinasi 5 Fold, 30 epoch per fold, batch size 64, dan learning rate 0.001. Jika dilihat dari metrik utama, nilai Mean Squared Error (MSE) yang sangat rendah yaitu 0.0002 menandakan bahwa selisih kuadrat antara nilai prediksi dan nilai aktual sangat kecil. Sementara itu, nilai Mean Absolute Error (MAE) sebesar 0.008 menunjukkan bahwa rata-rata kesalahan prediksi juga sangat minim.

Visualisasi grafik dengan konfigurasi 5 Fold, 30 epoch per fold, batch size 64, dan learning rate 0.001 pada eksperimen ini dapat dilihat pada Gbr. 6. Akurasi model pada konfigurasi ini juga paling tinggi, yakni mencapai 0.913, lebih unggul dibandingkan konfigurasi dengan learning rate 0.0001 (0.872) dan 0.00001 (0.865), yang keduanya juga memiliki MAE dan MSE lebih tinggi



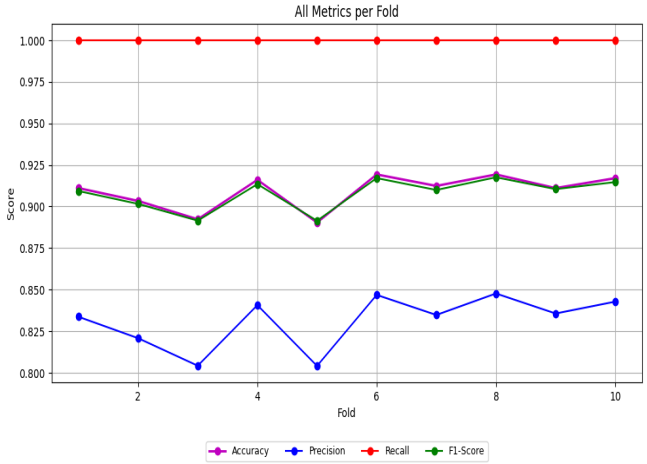
Gbr. 6 grafik konfigurasi 5 Fold, 30 epoch per fold, batch size 64, dan learning rate 0.001.

Selain unggul dari segi performa, konfigurasi 5 Fold, 30 epoch per fold, batch size 64, dan learning rate 0.001 ini juga cukup efisien dari segi waktu pelatihan, dengan total waktu hanya 122 menit, masih lebih cepat dibandingkan konfigurasi 10 Fold dengan learning rate 0.0001 yang membutuhkan waktu hingga 310 menit pada eksperimen ini dapat dilihat pada Gbr. 7. Performa konfigurasi 10 Fold dengan learning rate 0.001 sebenarnya juga cukup tinggi dengan akurasi 0.909 dan MSE serupa (0.0002), namun MAE-nya lebih besar yaitu 0.0276. Dengan mempertimbangkan keseimbangan antara akurasi, kesalahan prediksi, dan efisiensi waktu pelatihan, konfigurasi 5 Fold dengan learning rate 0.001 dapat disimpulkan sebagai pilihan paling optimal pada eksperimen ini. Perbedaan performa ini terlihat cukup signifikan pada grafik akurasi dan loss yang dihasilkan selama proses pelatihan.



Gbr. 7 grafik konfigurasi 10 Fold, 30 epoch per fold, batch size 64, dan learning rate 0.001

Konfigurasi 5-Fold dengan 30 epoch dan learning rate sebesar 0.001 menunjukkan performa paling optimal dalam eksperimen ini, dengan menghasilkan akurasi tertinggi sebesar 0.913, nilai loss terkecil 0.0002, dan Mean Absolute Error (MAE) yang sangat rendah yaitu 0.008. Jika dibandingkan dengan konfigurasi 10-Fold pada jumlah epoch dan learning rate yang sama, akurasinya sedikit menurun menjadi 0.909, MAE meningkat menjadi 0.0276, serta membutuhkan waktu pelatihan yang lebih lama. Meskipun keduanya menghasilkan nilai loss yang sangat kecil, hal ini menunjukkan bahwa konfigurasi 5-Fold lebih efisien dalam proses pelatihan dan mampu menghasilkan prediksi yang lebih presisi. Visualisasi grafik hasil pelatihan dengan konfigurasi 10-Fold, 30 epoch per fold, batch size 64, dan learning rate 0.001 dapat dilihat pada Gbr. 8. Grafik tersebut memperlihatkan fluktuasi akurasi dan loss yang lebih besar dibandingkan konfigurasi 5-Fold. Selain itu, pola konvergensi pada konfigurasi 10-Fold tampak lebih lambat dan tidak sehalus konfigurasi 5-Fold.



Gbr. 8 grafik konfigurasi 10 Fold, 30 epoch per fold, batch size 64, dan learning rate 0.001 representasi tambahan metriks

Selain menunjukkan performa yang baik secara evaluasi, model ini juga mampu memberikan rekomendasi anime dengan skor prediksi yang tinggi kepada pengguna. Rekomendasi yang dihasilkan didominasi oleh judul-judul dengan kualitas baik dimana dapat dilihat melalui rating anime yang tinggi dan skor prediksi yang diberikan, yang selaras dengan preferensi umum komunitas. Hal ini menunjukkan bahwa penggabungan variabel content based dan collaborative filtering yang digunakan cukup efektif dalam menangkap pola kesukaan pengguna, meskipun belum mempertimbangkan aspek naratif atau genre secara mendalam. Hal ini terlihat dari kemunculan judul-judul anime dengan genre dan seri yang saling berkaitan, seperti Gintama, Natsume Yuujinchou, dan Shouwa Genroku Rakugo Shinjuu. Skor prediksi yang tinggi dan hampir seragam, yaitu sekitar 9.78 hingga 9.79, hasil tersebut mengindikasikan tingkat konsistensi model dalam memberikan rekomendasi kepada pengguna. Dengan mempertimbangkan performa evaluasi dan kualitas hasil rekomendasi yang dihasilkan, model menunjukkan potensi

yang bagus untuk diterapkan untuk perekomendasian anime kepada pengguna.

IV. KESIMPULAN

Penelitian ini bertujuan mengetahui kemampuan LSTM dalam merekomendasikan anime secara lebih personal untuk tiap pengguna yang tentunya memiliki karakteristik preferensi berbeda tiap pengguna, dengan memanfaatkan data preferensi pengguna dan konten anime. Melalui beberapa eksperimen, model diuji untuk melihat sejauh mana pola-pola tersebut dapat dipelajari dan dimanfaatkan dalam model rekomendasi.

Model LSTM berhasil diterapkan dalam model rekomendasi untuk menangkap pola preferensi pengguna berdasarkan data yang diberikan. Keberhasilan ini tidak hanya terlihat dari performa model dalam menghasilkan prediksi, tetapi juga dari hasil stabil yang diperoleh saat diuji dengan data yang bervariasi melalui fold validasi silang. Kemampuan adaptasi model terhadap perubahan data menunjukkan potensi LSTM sebagai pendekatan yang baik dalam membangun model rekomendasi berbasis konten dan karakteristik preferensi pengguna.

Selain keberhasilan dalam memahami pola data, performa akhir model juga sangat dipengaruhi oleh berbagai faktor teknis seperti jenis input yang digunakan, struktur arsitektur LSTM, serta konfigurasi pelatihan seperti jumlah epoch, jumlah fold, dan learning rate. Penyesuaian parameter ini menjadi aspek penting dalam proses pelatihan model agar mampu menghasilkan hasil yang optimal. Secara keseluruhan, penerapan metode deep learning dengan LSTM terbukti efektif dalam memberikan rekomendasi yang mencerminkan karakteristik pengguna. Hal ini ditunjukkan melalui pencapaian nilai loss paling rendah sebesar 0.0002 dan akurasi pelatihan tertinggi yang mencapai 91,3%, yang menandakan bahwa model tidak hanya akurat tetapi juga efisien dalam mempelajari dan memprediksi preferensi pengguna.

V. SARAN

Adapun dari penelitian ini, masih terdapat berbagai aspek yang dapat ditingkatkan dan dieksplorasi lebih lanjut. Oleh karena itu, berikut beberapa saran yang dapat menjadi acuan untuk pengembangan dan penelitian lanjutan ke depannya:

- Model dapat dikembangkan lebih lanjut dengan menambahkan fitur tambahan seperti ulasan pengguna untuk menghasilkan rekomendasi yang lebih personal sesuai preferensi pengguna.
- Model rekomendasi dapat ditingkatkan dengan mempertimbangkan perubahan preferensi pengguna yang mungkin berkembang seiring berjalannya waktu sehingga model dapat terus memberikan rekomendasi yang relevan dalam jangka panjang..
- Pengembangan selanjutnya dapat difokuskan pada pembuatan aplikasi berbasis web atau mobile dengan antarmuka yang intuitif dan mudah digunakan oleh pengguna.
- Masalah *cold start*, terutama pada pengguna baru yang belum memiliki riwayat interaksi, perlu ditangani dengan pendekatan seperti *hybrid recommendation* atau penyusunan profil awal berdasarkan preferensi yang diberikan di awal penggunaan.

UCAPAN TERIMA KASIH

Puji syukur kepada Tuhan Yang Masa Esa, karenaNya penelitian ini dapat diselesaikan dengan baik. Serta ucapan terima kasih kepada semua rekan-rekan yang terlibat pada penelitian ini, kepada dosen pembimbing yang telah membimbing saya dari awal hingga selesai, dan terakhir kepada orang tua yang selalu mendukung tanpa pamrih.

REFERENSI

[1] K. T. Mukti and I. Mardhiyah, “Sistem Rekomendasi Pembelian Lisensi Film Menggunakan Pendekatan Hybrid Filtering (Studi Kasus : Film Animasi Jepang),” *JURSISTEKNI (Jurnal Sist. Inf. dan Teknol. Informasi)*, vol. 4, no. 3, pp. 127–139, 2022.

[2] N. M. Roziqiin and M. Faisal, “Sistem Rekomendasi Pemilihan Anime Menggunakan User-Based Collaborative Filtering,” *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 9, no. 1, pp. 299–306, 2024, doi: 10.29100/jipi.v9i1.4222.

[3] M. D. A. Putra, D. A. Dewi, W. T. H. Putri, and H. T. Y. Achsan, “Efficient web mining on my animelist: A concurrency-driven approach using the go programming language,” *J. Appl. Data Sci.*, vol. 5, no. 3, pp. 1472–1481, 2024, doi: 10.47738/jads.v5i3.352.

[4] M. A. Ma’ruf and A. Qoiriah, “Perbandingan Algoritma Cosine Similarity dan Euclidean Distance pada Sistem Rekomendasi Film dengan Metode Item-Based Collaborative Filtering,” *Journal of Informatics and Computer Science (JINACS)*. pp. 160–168, 2022. doi: 10.26740/jinacs.v4n02.p160-168.

[5] M. Fajriansyah, P. P. Adikara, and A. W. Widodo, “Sistem Rekomendasi Film Menggunakan Content Based Filtering,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 6. pp. 2188–2199, 2021. [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/9163

[6] N. F. E. Putra and R. E. Putra, “Penerapan Metode Long Short-Term Memory Dalam Sistem Rekomendasi Tim Fantasy Premier League,” *J. Informatics Comput. Sci.*, vol. 5, no. 04, pp. 655–666, May 2024, doi: 10.26740/jinacs.v5n04.p655-666.

[7] I. K. Gowinda, I. G. S. Astawa, I. G. N. A. Cahyadi Putra, N. Agus Sanjaya, I. B. G. Dwidasmara, and I. D. M. B. Atmaja Darmawan, “Sistem Rekomendasi Seri Animasi Jepang (Anime) Menggunakan User-Based Collaborative Filtering dan Spearman Rank Correlation Coefficient oleh I Kadek Gowindaa et. al (2023).pdf,” *JELIKU (Jurnal Elektron. Ilmu Komput. Udayana)*, vol. 12, no. 2, p. 469, Oct. 2023, doi: 10.24843/JLK.2023.v12.i02.p26.

[8] M. C. Aghnia, “Perancangan Anime Community Center,” *J. Tingkat Sarj. Bid. Senirupa dan Desain*, vol. 1, no. 1, pp. 1–6, 2012.

[9] F. Ricci, L. Rokach, B. Shapira, P. B. Kantor, and F. Ricci, *Recommender Systems Handbook*. 2011. doi: 10.1007/978-0-387-85820-3.

[10] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[11] H. Chung and K. S. Shin, “Genetic algorithm-optimized long short-term memory network for stock market prediction,” *Sustain.*, vol. 10, no. 10, 2018, doi: 10.3390/su10103765.

[12] H. H. Arfisko and A. T. Wibowo, “Sistem Rekomendasi Film Menggunakan Metode Hybrid Collaborative Filtering dan Content-Based Filtering,” *e-Proceeding Eng.*, vol. 9, no. 3, pp. 2149–2159, 2022.