Indonesian Journal of Multidisciplinary on Social and Technology
Homepage: https://journal.ilmudata.co.id/index.php/ijmst
Vol. 4 No. 3 (2026) pp: 2811-2823
P-ISSN: 2986-6790, e-ISSN: 2986-6782

Sistem Rekomendasi Anime Menggunakan Pendekatan Hybrid Dengan Algoritma *Random Forest* Dan *K-Nearest Neighbor* (KNN)

Fauzan Nafazi Zul Hilmi[1], Ilham Saifudin[2*], Dudi Irawan[3]
[1,2,3]Program Studi Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Jember
[1]fanizmi26@gmail.com, [2]ilham.saifudin@unmuhjember.ac.id*, [3]dudi.irawan@unmuhjember.ac.id

Abstract

An anime recommendation system is needed to help users discover content matching their preferences amid the continuing growth in titles and genre diversity. This study aims to develop and evaluate an anime recommendation system using a weighted hybrid approach that combines Random Forest and K-Nearest Neighbor (KNN) based on cosine similarity. A secondary dataset obtained from Kaggle contains 24,905 anime records with name, genres, score, type, studios, and rating attributes. The research stages include handling missing values, transforming categorical attributes using Multi-Label Binarizer and Label Encoder, forming suitability labels based on the mean score, splitting the data into 70:30, 80:20, and 90:10 scenarios, training the models, combining the scores, and implementing a web-based application. Random Forest classifies anime suitability, whereas KNN calculates genre similarity. Both outputs are integrated using equal weights to generate recommendation rankings. The system is evaluated using precision, Normalized Discounted Cumulative Gain (NDCG), and computational time. Top-K testing produced precision values of 80.0%–85.0% and NDCG values of 89.5%–94.9%, with computational time below 2.5 seconds. Global evaluation produced precision values of 75.2%–75.6% and NDCG values of 97.3%–97.7%. These findings demonstrate that integrating Random Forest and KNN through a weighted hybrid approach can generate relevant, well-ranked, and computationally efficient anime recommendations for implementation on a web-based platform.

Keywords: Cosine Similarity, K-Nearest Neighbor, *Random Forest, Anime Recommendation System, Weighted Hybrid.*

Abstrak

Sistem rekomendasi anime diperlukan untuk membantu pengguna menemukan tayangan yang sesuai dengan preferensi di tengah jumlah judul dan keragaman genre yang terus meningkat. Penelitian ini bertujuan mengembangkan serta mengevaluasi sistem rekomendasi anime menggunakan pendekatan weighted hybrid yang menggabungkan algoritma Random Forest dan K-Nearest Neighbor (KNN) berbasis cosine similarity. Dataset sekunder diperoleh dari Kaggle dan terdiri atas 24.905 data anime dengan atribut name, genres, score, type, studios, dan rating. Tahapan penelitian meliputi pembersihan missing value, transformasi atribut kategorikal menggunakan Multi-Label Binarizer dan Label Encoder, pembentukan label kelayakan berdasarkan nilai rata-rata score, pembagian data dalam skenario 70:30, 80:20, dan 90:10, pelatihan model, penggabungan skor, serta implementasi aplikasi berbasis web. Random Forest digunakan untuk mengklasifikasikan kelayakan anime, sedangkan KNN menghitung kemiripan genre. Kedua keluaran digabungkan dengan bobot seimbang untuk menghasilkan peringkat rekomendasi. Evaluasi dilakukan menggunakan precision, Normalized Discounted Cumulative Gain (NDCG), dan waktu komputasi. Hasil pengujian Top-K menunjukkan precision sebesar 80,0%–85,0% dan NDCG sebesar 89,5%–94,9%, dengan waktu komputasi kurang dari 2,5 detik. Evaluasi global menghasilkan precision 75,2%–75,6% dan NDCG 97,3%–97,7%. Temuan ini menunjukkan bahwa integrasi Random Forest dan KNN melalui weighted hybrid mampu menghasilkan rekomendasi anime yang relevan, terurut dengan baik, dan efisien untuk diterapkan pada platform berbasis web.

Kata kunci: *Cosine Similarity*, *K-Nearest Neighbor*, *Random Forest*, Sistem Rekomendasi Anime, *Weighted Hybrid*.

1. Pendahuluan

Teknologi telah menjadi bagian yang tidak terpisahkan dari kehidupan manusia di era digital saat ini. Perkembangan teknologi informasi mendorong perubahan signifikan dalam cara masyarakat mengakses berbagai layanan, termasuk hiburan digital melalui *platform* daring. Kumala Sari (2024) menyatakan bahwa era digital telah membawa transformasi besar dalam aktivitas masyarakat, khususnya dalam pemanfaatan layanan berbasis internet seperti *e-commerce* dan *platform* digital lainnya. Kemudahan akses tersebut memberikan efisiensi, namun di sisi lain juga menghadirkan tantangan berupa melimpahnya pilihan konten yang tersedia, sehingga pengguna sering kali mengalami kesulitan dalam menentukan pilihan yang sesuai dengan preferensi mereka dan memerlukan mekanisme yang mampu membantu proses seleksi konten secara lebih efektif.

Salah satu bentuk hiburan digital yang mengalami pertumbuhan signifikan adalah anime. (Hasegawa & Masuda, 2024) Melaporkan bahwa industri anime terus mengalami peningkatan jumlah judul setiap tahunnya, sementara Gomez dkk. (2025) menekankan bahwa keberagaman genre dan tema anime turut memperluas jangkauan

pemanfaatannya, tidak hanya sebagai hiburan tetapi juga sebagai media edukasi. Meningkatnya jumlah judul anime dengan variasi genre, tipe penayangan, serta segmentasi usia membuat proses pencarian tontonan semakin kompleks. Dalam konteks ini, sistem rekomendasi menjadi solusi yang krusial untuk membantu pengguna menemukan konten yang relevan. Saifudin & Widiyaningtyas (2024) menjelaskan bahwa sistem rekomendasi berperan penting dalam menyaring informasi di tengah banyaknya pilihan data, sehingga pengguna dapat memperoleh konten yang sesuai dengan preferensi pribadi secara lebih efektif. Oleh karena itu, penerapan sistem rekomendasi pada konten anime menjadi sangat relevan mengingat karakteristiknya yang memiliki atribut beragam serta kombinasi genre yang kompleks, sebagaimana juga ditunjukkan oleh Setiawan dkk. (2023) dan Armenta Segura & Sidorov (2025) yang memodelkan popularitas dan tren genre anime menggunakan pendekatan pembelajaran mesin.

Berbagai penelitian menunjukkan bahwa efektivitas sistem rekomendasi dapat ditingkatkan melalui penerapan teknik machine learning. Michael Lauw dkk. (2023) menjelaskan bahwa algoritma *K-Nearest Neighbor* (KNN) efektif dalam mengukur tingkat kemiripan antaritem berdasarkan karakteristik yang dimiliki, sedangkan *random forest* mampu melakukan proses klasifikasi secara stabil pada data dengan atribut kategorikal yang beragam. Kedua algoritma tersebut memiliki karakteristik yang saling melengkapi, namun penggunaan metode tunggal masih memiliki keterbatasan karena hanya memanfaatkan satu pendekatan, yaitu klasifikasi atau perhitungan kemiripan, sehingga informasi yang diperoleh belum dimanfaatkan secara optimal untuk menghasilkan rekomendasi.

Beberapa penelitian sebelumnya telah menerapkan berbagai metode dalam pengembangan sistem rekomendasi. Alvina & Yamasari (2025) menunjukkan bahwa algoritma *random forest* memiliki performa yang lebih baik dibandingkan dengan *decision tree* dalam sistem rekomendasi lagu berbasis genre. Aini dkk. (2024) menerapkan pendekatan *hybrid recommendation* pada sistem rekomendasi mitra MSIB dan memperoleh hasil evaluasi yang baik dengan metrik *accuracy*, *precision*, *recall*, serta MAE. Fauzi dkk. (2025) mengembangkan sistem rekomendasi berbasis *content-based filtering* dan *random forest* yang mampu menghasilkan nilai *accuracy* sebesar 94,0%. Pada domain lain, Rachmaniar dkk. (2025) serta Yanisa Putri dkk. (2024) menunjukkan bahwa penggabungan *content-based filtering* dengan *collaborative filtering* mampu meningkatkan relevansi rekomendasi produk maupun rekomendasi *skincare*, sementara Mia dkk. (2025) menerapkan *hybrid filtering* pada sistem rekomendasi *make-up* berbasis web. Meskipun demikian, penelitian-penelitian tersebut masih menerapkan algoritma secara terpisah atau menggunakan kombinasi metode yang berbeda, sehingga integrasi antara *random forest* dan KNN dalam sistem rekomendasi anime masih belum banyak dikaji.

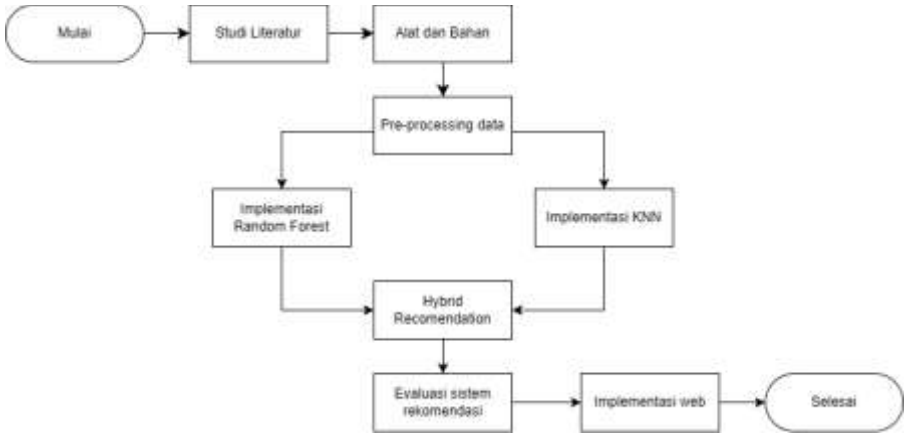
Berdasarkan celah penelitian tersebut, penelitian ini mengusulkan pendekatan *weighted hybrid* yang mengintegrasikan *random forest* dan KNN dalam sistem rekomendasi anime berbasis *content-based filtering*. Pendekatan ini dirancang untuk menggabungkan kemampuan *random fores*t dalam melakukan klasifikasi kelayakan konten dengan kemampuan KNN dalam menghitung kemiripan konten berdasarkan *cosine similarity*, sehingga diharapkan dapat meningkatkan akurasi dan relevansi rekomendasi. Penelitian ini bertujuan untuk menganalisis efektivitas pendekatan *weighted hybrid* tersebut dalam menghasilkan rekomendasi akhir, sekaligus mengevaluasi kualitas rekomendasi yang dihasilkan berdasarkan metrik *Normalized Discounted Cumulative Gain* (NDCG), *precision*, dan waktu komputasi. *Dataset* yang digunakan merupakan data sekunder yang diperoleh dari repositori publik Kaggle, berisi atribut nama, *genres*, *rating*, *type*, *studios*, dan *score* dengan jumlah 24.905 *records* pada tahun 2023. Atribut *score* digunakan sebagai dasar pembentukan label kelayakan konten, sedangkan proses rekomendasi didasarkan pada atribut konten seperti *genres*, *type*, *rating*, dan *studios*, tanpa melibatkan data riwayat interaksi pengguna. Dengan demikian, penelitian ini tidak hanya melanjutkan studi sebelumnya, tetapi juga menawarkan pendekatan yang lebih komprehensif dalam mengatasi kompleksitas pemilihan konten anime di era digital, sekaligus menjadi referensi bagi pengembangan sistem rekomendasi berbasis *content-based* maupun *hybrid recommendation* pada penelitian selanjutnya..

2. Metode Penelitian

Penelitian ini menggunakan metode pendekatan kuantitatif eksperimental untuk membangun sistem rekomendasi *hybrid*. Penelitian kuantitatif merupakan pendekatan ilmiah yang menekankan pada pengukuran objektif dan analisis data numerik untuk menjawab suatu permasalahan penelitian, sementara *experimental research* merupakan metode yang digunakan untuk menguji suatu perlakuan atau model secara sistematis guna mengukur pengaruhnya terhadap suatu variabel atau hasil tertentu. Pendekatan kuantitatif dipilih karena penelitian ini melibatkan pemrosesan data numerik dan pengukuran kinerja sistem menggunakan parameter statistik yang objektif, sedangkan metode eksperimental diterapkan melalui serangkaian uji coba model algoritma *random forest* dan *K-Nearest Neighbor* (KNN) untuk memvalidasi kualitas rekomendasi berdasarkan metrik *Normalized Discounted Cumulative Gain* (NDCG) dan *precision*.

DOI: https://doi.org/10.69693/ijmst.v4i2. 12782

Tahapan penelitian disusun secara sistematis mulai dari studi literatur, pengumpulan data, pra-pemrosesan, pemodelan algoritma, penggabungan *hybrid*, evaluasi, hingga implementasi antarmuka *website*, seperti ditunjukkan pada Gambar 1. Studi literatur dilakukan untuk menghimpun dan menelaah referensi terkait algoritma *random forest*, KNN, serta pendekatan *hybrid recommendation* yang relevan dengan objek penelitian, sebagai landasan teori dan bahan validasi untuk membandingkan kontribusi penelitian ini dengan penelitian terdahulu.



Gambar 1. Alur Penelitian

2.1. *Dataset*

Dataset yang digunakan merupakan data sekunder yang diperoleh dari Kaggle dan berisi informasi anime berupa *name*, *genres*, *score*, *type*, *studios*, dan *rating* sebanyak 24.905 data. Berikut Tabel 1 yang memuat variabel penelitian beserta penjelasannya.

Tabel 1. Variabel Penelitian

No	*Anime id*	*Name*	*Score*	*Genres*	*Type*	*Studios*	*Rating*
1	1	Cowboy Bebop	8.75	*Action, Award Winning, Sci-Fi*	TV	Sunrise	R - 17+ (*violence* & *profanity*)
2	5	Cowboy Bebop: Tengoku no Tobira	8.38	*Action, Sci-Fi*	*Movie*	Bones	R - 17+ (*violence* & profanity)
3	6	Trigun	8.22	*Action, Adventure, Sci-Fi*	TV	Madhouse	PG-13 - *Teens* 13 *or older*
4	7	Witch Hunter Robin	7.25	*Action, Drama, Mystery, Supernatural*	TV	*Sunrise*	PG-13 - *Teens* 13 *or older*
5	8	Bouken Ou Beet	6.94	*Adventure, Fantasy, Supernatural*	TV	*Toei Animation*	PG - *Children*
6	15	Eyeshield 21	7.92	*Sports*	TV	*Gallop*	PG-13 - *Teens* 13 *or older*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
24905	55734	Bokura no Saishuu Sensou	*UNKNOWN*	*UNKNOWN*	*Music*	*UNKNOWN*	PG-13 - *Teens* 13 *or older*
24906	55735	Shijuuku Nichi	*UNKNOWN*	*UNKNOWN*	*Music*	*UNKNOWN*	PG-13 - *Teens* 13 *or older*

2.2. *Pre-Processing*

Tahap pra-pemrosesan data dilakukan untuk mempersiapkan dataset anime agar berada dalam kondisi bersih, konsisten, dan siap digunakan dalam proses klasifikasi kelayakan serta perhitungan kemiripan. *Dataset* yang digunakan masih mengandung *missing value*, format tipe data yang belum seragam, serta atribut kategorikal yang perlu dikonversi menjadi numerik. Pada atribut numerik *score*, nilai yang tertulis sebagai *unknown* terlebih dahulu diubah menjadi *NaN*, kemudian dikonversi ke tipe numerik dan nilai yang kosong diisi menggunakan nilai rata-rata seluruh *score* dalam *dataset* agar baris data yang masih memiliki informasi pada atribut lain tetap dapat digunakan dalam proses pemodelan. Sementara itu, *missing value* pada atribut kategorikal *genres*, *type*, *studios*, dan *rating* ditangani dengan mengganti nilai kosong menjadi kategori *unknown*, sedangkan nilai kosong pada kolom gambar diganti dengan gambar *placeholder*.

Proses selanjutnya adalah transformasi data untuk mengubah atribut kategorikal agar dapat diproses secara matematis. Atribut tunggal seperti *type*, *studios*, dan *rating* dikonversi menjadi representasi numerik menggunakan teknik *label encoding*, sedangkan atribut *genres* yang memiliki banyak nilai ditransformasikan menggunakan teknik *one-hot encoding* melalui *multi-label binarizer*, sehingga setiap elemen genre dapat dikenali secara independen sebagai fitur vektor biner. Langkah terakhir adalah pembentukan label kelas (*class labeling*) untuk algoritma klasifikasi, di mana atribut *score* numerik diubah menjadi label kategori berdasarkan nilai rata-rata

DOI: https://doi.org/10.69693/ijmst.v4i2. 12782

dataset; anime dengan *score* di atas atau sama dengan nilai rata-rata diberi label Layak, sedangkan yang di bawah rata-rata diberi label Tidak Layak. Label ini berfungsi sebagai atribut target yang dipelajari oleh model *random forest*, sedangkan atribut *score* asli tidak digunakan sebagai fitur input untuk mencegah kebocoran data (*data leakage*). Setelah data dibersihkan, *dataset* dibagi menjadi data latih dan data uji dengan tiga skenario perbandingan, yaitu 70:30, 80:20, dan 90:10, untuk membandingkan performa model pada berbagai proporsi pembagian data. Alur tahapan pra-pemrosesan ditunjukkan pada Gambar 2.

Gambar 2. *Flowchart Pre-Processing*

2.3. Implementasi Algoritma *Random Forest*

Algoritma *random forest* diterapkan untuk memprediksi status kelayakan konten anime sebagai bagian dari mekanisme pembobotan dalam sistem *hybrid*. Berbeda dengan metode *filtering* murni, hasil prediksi dari tahap ini tidak digunakan untuk membuang data, melainkan dikonversi menjadi nilai bobot numerik yang selanjutnya dijumlahkan dengan skor kemiripan dari algoritma *K-Nearest Neighbor* (KNN). *Random forest* bekerja dengan membangun sekumpulan pohon keputusan (*decision tree*) dari data latih, di mana hasil prediksi akhir ditentukan melalui mekanisme *majority voting* dari seluruh pohon keputusan yang terbentuk. Pembentukan setiap pohon dilakukan melalui dua prinsip utama, yaitu *bootstrapping*, di mana setiap pohon dilatih menggunakan sampel data acak yang diambil dengan pengembalian dari data asli sehingga setiap pohon memiliki karakteristik yang unik, dan *random feature selection*, di mana pada setiap pemecahan simpul algoritma hanya mempertimbangkan sebagian fitur secara acak untuk mengurangi korelasi antarpohon dan mencegah *overfitting*.

Proses pemecahan simpul pada setiap pohon keputusan mengacu pada mekanisme seleksi atribut yang lazim digunakan dalam algoritma berbasis pohon keputusan, yaitu melalui perhitungan *entropy*, *gain*, *split information*, dan *gain ratio*. *Entropy* digunakan untuk mengukur tingkat ketidakmurnian (*impurity*) suatu atribut keputusan dalam sekumpulan data (Iddrus & Sari, 2023), dihitung menggunakan Persamaan (1) untuk satu atribut, serta Persamaan (2) apabila melibatkan tabel frekuensi dari dua atribut secara bersamaan.

$$Entropy(S) = \sum_{i=1}^{p} -p_i \; x \; log_2 \; p_i \tag{1}$$

dengan S adalah total kasus atau data yang sedang dihitung, n adalah jumlah partisi S, dan p_i adalah proporsi S_i terhadap S.

$$Entropy(T, X) = \sum_{c \in X} P(c)E(c) \tag{2}$$

dengan (T, X) menyatakan atribut T dan atribut X, P(c) adalah jumlah kelas yang terdapat dalam atribut target, dan E(c) adalah perbandingan jumlah data kelas ke-i terhadap total data pada S.

DOI: https://doi.org/10.69693/ijmst.v4i2. 12782

Setelah nilai *entropy* diperoleh, tahap berikutnya adalah menghitung nilai *gain*, yaitu ukuran efektivitas suatu atribut dalam memilah data pada setiap simpul pohon keputusan (Husaini & Jemakmun, 2023), yang dihitung menggunakan Persamaan (3).

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} \; x \; Entropy(S_i) \tag{3}$$

dengan S adalah himpunan kasus, A adalah atribut yang diuji, |S_i| adalah jumlah kasus pada partisi ke-i, |S| adalah jumlah total kasus pada himpunan S, n adalah jumlah partisi atribut A, Entropy(S) adalah nilai entropy dari himpunan kasus S sebelum dipartisi, dan Entropy(S_i) adalah nilai entropy dari partisi ke-i setelah himpunan dibagi berdasarkan atribut A.

Selain *gain*, pembentukan pohon keputusan turut memerlukan nilai *split information* untuk memperhitungkan proporsi ukuran serta jumlah partisi yang dihasilkan oleh suatu atribut (Setio dkk., 2020), sebagaimana dirumuskan pada Persamaan (4).

$$SplitInfo(S, A) = -\sum_{j=1}^{k} \frac{S_j}{S} \; x \; log_2 \; \frac{S_j}{S} \tag{4}$$

dengan S didefinisikan sebagai ruang sampel, A merupakan atribut yang sedang dihitung, S_j menunjukkan jumlah sampel pada atribut ke-j, dan k menunjukkan jumlah total partisi yang dihasilkan oleh atribut A.

Nilai *gain* dan *split information* selanjutnya digunakan untuk menghitung *gain ratio*, yaitu metode perbaikan seleksi atribut yang memanfaatkan informasi intrinsik dari atribut serta mampu menangani data yang tidak stabil (Nur dkk., 2022), sebagaimana dirumuskan pada Persamaan (5).

$$GainRatio(S, A) = \frac{Gain(S, A)}{SplitInfo(S, A)} \tag{5}$$

dengan Gain Ratio(S,A) adalah nilai rasio perolehan informasi untuk atribut A pada himpunan data S, Gain(S,A) adalah nilai information gain pada atribut tersebut, dan SplitInfo(S,A) adalah nilai split information dari atribut tersebut. Atribut dengan nilai *gain ratio* tertinggi pada setiap simpul dipilih sebagai kriteria pemecahan (*splitting criterion*) dalam pembentukan struktur pohon keputusan pada algoritma *random forest*.

Setiap pohon keputusan memprediksi apakah sebuah anime masuk kategori Layak atau Tidak Layak berdasarkan atribut *genres*, *type*, *studios*, dan *rating*, tanpa melibatkan atribut *score* secara langsung sebagai fitur *input*. Alvanof dkk. (2024) serta Suci Amaliah dkk. (2022) menunjukkan bahwa *random forest* mampu memberikan hasil klasifikasi yang stabil pada data dengan atribut kategorikal yang bervariasi, sedangkan Alvina & Yamasari (2025) menegaskan bahwa *random forest* umumnya menghasilkan performa klasifikasi yang lebih baik dibandingkan model pohon keputusan tunggal karena sifatnya yang berbasis ansambel (*ensemble*). Pada penelitian ini, model *random forest* dibangun menggunakan parameter *n_estimators*=100, yang berarti sistem membentuk 100 pohon keputusan untuk meningkatkan stabilitas dan akurasi prediksi, dengan parameter *random_state*=50 agar proses pelatihan menghasilkan *output* yang konsisten pada setiap percobaan. Anime yang diprediksi layak diberikan bobot bernilai 1, sedangkan anime yang tidak memenuhi kategori layak diberikan bobot bernilai 0. Bobot tersebut selanjutnya digunakan sebagai salah satu komponen dalam proses *weighted hybrid recommendation*.

2.4. Implementasi Algoritma *K-Nearest Neighbor* (KNN)

Setelah tahap klasifikasi menggunakan *random forest* selesai, hasil prediksi dikonversi menjadi bobot validasi. Jika *random forest* berfungsi sebagai pemberi bobot kelayakan, maka KNN berfungsi sebagai pengukur kemiripan konten, dan keduanya berjalan secara paralel untuk menghasilkan skor akhir. Penentuan tetangga terdekat pada penelitian ini tidak menggunakan jarak *euclidean*, melainkan menggunakan *cosine similarity*. Dzikri dkk. (2023) menunjukkan bahwa *cosine similarity* lebih sesuai digunakan pada data berdimensi tinggi, seperti representasi genre, karena orientasi vektor lebih merepresentasikan kemiripan konten dibandingkan dengan besaran jarak absolut.

Perhitungan *cosine similarity* dilakukan dengan mengukur nilai kosinus sudut antara dua vektor atribut untuk menentukan tingkat kemiripan antarkeduanya (Dzikri dkk., 2023), sebagaimana dirumuskan pada Persamaan (6).

DOI: https://doi.org/10.69693/ijmst.v4i2. 12782

$$CosineSimilarity(A,B) = \frac{\sum_{i=1}^{n} A_i \; x \; B_i}{\sqrt{\sum_{i=1}^{n} A_i{}^2} \; x \; \sqrt{\sum_{i=1}^{n} B_i{}^2}} \quad (6)$$

dengan A dan B merupakan vektor atribut item yang dibandingkan, misalnya genre atau fitur anime, A_i x B_i menunjukkan komponen nilai atribut ke-i dari masing-masing vektor A dan B, serta n adalah jumlah dimensi fitur yang diukur.

Sejalan dengan hasil klasifikasi *random forest* yang menempatkan atribut *genres* sebagai fitur terpenting, perhitungan kemiripan pada tahap ini difokuskan pada vektor genre yang telah direpresentasikan melalui *one-hot encoding*, di mana nilai 1 menunjukkan bahwa genre tersebut dimiliki oleh sebuah anime dan nilai 0 menunjukkan sebaliknya. Brema Daniel Ginting dkk. (2024) menunjukkan bahwa KNN efektif digunakan untuk mengukur kemiripan antardata kategorikal yang direpresentasikan dalam bentuk vektor biner. Proses pencarian kemiripan dilakukan terhadap seluruh kandidat anime, kemudian hasil perhitungan diurutkan dari nilai *similarity* tertinggi hingga terendah, dan sejumlah k anime teratas diambil sebagai kandidat rekomendasi. Pada penelitian ini, sistem menampilkan 10 rekomendasi teratas berdasarkan nilai kemiripan dan bobot kelayakan tertinggi kepada pengguna.

2.5. Penggabungan *Weighted Hybrid*

Tahap ini merupakan integrasi dari dua metode yang telah dilakukan sebelumnya. Pendekatan yang digunakan adalah *weighted hybrid*, di mana hasil *output* dari kedua algoritma digabungkan melalui penjumlahan bobot, sejalan dengan konsep yang dikemukakan oleh Sonule dkk. (2024) bahwa *weighted hybrid recommendation* menggabungkan beberapa skor prediksi menjadi satu nilai akhir melalui pembobotan, sehingga item yang diprediksi tidak layak akan mendapatkan skor akhir yang rendah dan posisinya turun dalam daftar rekomendasi. Secara sistematis, skor akhir rekomendasi untuk kandidat anime x terhadap anime target dirumuskan pada Persamaan (7).

$$HybridScore(x) = (w\_RF \times RF(x)) + (w\_KNN \times KNN(x, target)) \quad (7)$$

dengan RF(x) merupakan fungsi prediksi *random forest* yang bernilai 1 jika kandidat x diklasifikasikan layak dan 0 jika tidak layak, KNN(x, target) merupakan nilai *cosine similarity* antara vektor genre kandidat x dan anime target, w_RF dan w_KNN merupakan bobot yang diberikan pada masing-masing komponen.

Pada penelitian ini, kedua bobot ditetapkan sama besar, yaitu w_RF = 0,5 dan w_KNN = 0,5, sehingga kontribusi hasil klasifikasi kelayakan dan hasil pengukuran kemiripan konten terhadap skor akhir bersifat seimbang. Setelah seluruh kandidat memperoleh *HybridScore*, sistem mengurutkan hasil secara descending dan menampilkan sejumlah anime teratas sebagai hasil rekomendasi akhir kepada pengguna

2.6. Evaluasi Model

Evaluasi sistem dilakukan menggunakan metrik DCG, IDCG, NDCG, *precision*, dan waktu komputasi. NDCG digunakan untuk mengukur kualitas urutan rekomendasi, *precision* untuk mengukur ketepatan rekomendasi yang diberikan, sedangkan waktu komputasi digunakan untuk menilai efisiensi sistem.

Penentuan relevansi dilakukan menggunakan pendekatan *binary relevance* berdasarkan nilai rata-rata *score*. Anime dengan nilai *score* di atas atau sama dengan rata-rata diberi label relevan, sedangkan nilai di bawah rata-rata diberi label tidak relevan. Seluruh metrik dihitung menggunakan Rumus (8) hingga Rumus (11).

$$DCG@K = \sum_{i=1}^{K} \frac{rel_i}{log_2(i+1)} \quad (8)$$

dengan $DCG@K$ adalah nilai *Discounted Cumulative Gain* untuk jumlah item teratas yang dievaluasi, rel_i adalah nilai relevansi item pada posisi ke-i dalam daftar rekomendasi, i adalah urutan posisi item di dalam daftar hasil rekomendasi, K adalah batasan jumlah item teratas yang dievaluasi dari daftar rekomendasi tersebut (Kurniadi dkk., 2024).

$$IDCG@K = \sum_{i=1}^{K} \frac{rel_i^{ideal}}{log_2(i+1)} \quad (9)$$

dengan $IDCG@K$ adalah nilai *Ideal Discounted Cumulative Gain* untuk jumlah item teratas yang dievaluasi, rel_i^{ideal} adalah nilai relevansi item pada posisi ke-i dalam kondisi peringkat yang ideal, i adalah urutan posisi item di dalam daftar hasil rekomendasi, K adalah batasan jumlah item teratas yang dievaluasi dari daftar rekomendasi tersebut (Pratama dkk., 2025).

$$NDCG@K = \frac{DCG@K}{IDCG@K} \quad (10)$$

dengan $NDCG@K$ adalah nilai *Normalized Discounted Cumulative Gain* untuk jumlah item teratas yang dievaluasi, $DCG@K$ adalah nilai *Discounted Cumulative Gain* pada peringkat ke-K yaitu skor urutan rekomendasi aktual yang dihasilkan oleh sistem, $IDCG@K$ adalah nilai *Ideal Discounted Cumulative Gain* pada peringkat ke-

DOI: https://doi.org/10.69693/ijmst.v4i2. 12782

K yaitu skor urutan rekomendasi terbaik atau paling ideal yang mungkin dicapai, K adalah jumlah batasan item teratas yang dievaluasi dari daftar rekomendasi tersebut (Saifudin & Widiyaningtyas, 2024).

$$\text{Precision}(u) = \frac{|\text{Recommended}(u) \cap \text{Testing}(u)|}{|\text{Recommended}(u)|} \quad (11)$$

dengan *Recommended(u)* adalah himpunan item yang direkomendasikan sistem kepada pengguna u dan *Testing(u)*himpunan item yang relevan pada data pengujian untuk pengguna u.
Semakin tinggi nilai NDCG dan *precision*, semakin baik kualitas rekomendasi yang dihasilkan, sedangkan waktu komputasi yang lebih rendah menunjukkan sistem yang lebih efisien.

2.7. Implementasi Website

Tahap implementasi website merupakan proses penerjemahan model komputasi *hybrid* ke dalam bentuk antarmuka pengguna (*user interface*) berbasis web, dengan tujuan menyediakan *platform* interaktif yang memungkinkan pengguna mendapatkan rekomendasi anime tanpa perlu memahami kode pemrograman di baliknya. Pembangunan antarmuka sistem menggunakan *framework Flask* dipilih karena efisiensinya dalam mengonversi skrip data *Python* menjadi aplikasi web interaktif dengan cepat. Sistem web dirancang dengan arsitektur *client-server* sederhana, di mana pengguna memasukkan atau memilih judul anime yang disukai melalui fitur pencarian. Kemudian, sistem menerima input judul anime target dan algoritma *random forest* melakukan klasifikasi terhadap seluruh kandidat anime dalam *dataset* untuk memberikan bobot validasi. Secara paralel, sistem menghitung nilai *cosine similarity* antara anime target dan kandidat, kemudian menggabungkan kedua nilai tersebut untuk mendapatkan skor akhir dan menampilkan daftar rekomendasi yang telah diurutkan berdasarkan skor tertinggi.

Antarmuka *website* dirancang dengan prinsip minimalis untuk memudahkan navigasi pengguna, terdiri dari *header* yang menampilkan judul anime yang sedang dicari serta daftar hasil rekomendasi dalam bentuk kartu yang menampilkan atribut *type*, *studios*, *rating*, dan *genres* secara eksplisit, sehingga pengguna dapat memahami alasan sebuah anime direkomendasikan. Alur kerja sistem secara menyeluruh, mulai dari proses pemilihan anime hingga menghasilkan rekomendasi, ditunjukkan pada Gambar 3.

Gambar 3. Alur Sistem *Website*

3. Hasil dan Pembahasan

Pengujian dilakukan menggunakan *dataset* anime dari Kaggle dengan beberapa skenario pembagian data latih dan data uji. Kinerja sistem dievaluasi menggunakan metrik *Normalized Discounted Cumulative Gain* (NDCG), *precision*, dan waktu komputasi untuk menganalisis kualitas rekomendasi yang dihasilkan.

3.1. Hasil

Hasil penelitian mencakup proses *pre-processing data*, pelatihan model, evaluasi Top-K dan evaluasi global, serta implementasi antarmuka *website* sebagai media penggunaan sistem rekomendasi.

3.1.1 *Pre-Processing* dan Pelatihan Model

Tahap pra-pemrosesan data berhasil mengubah *dataset* mentah anime dari Kaggle menjadi data yang bersih, konsisten, dan siap digunakan untuk pemodelan. Penanganan *missing value* pada atribut *score* dilakukan dengan

DOI: https://doi.org/10.69693/ijmst.v4i2. 12782

mengonversi nilai *unknown* menjadi *NaN,* kemudian mengisinya dengan nilai rata-rata seluruh *score* dalam *dataset*. Sedangkan *missing value* pada atribut kategorikal *genres*, *type*, *studios*, dan *rating* ditangani dengan mengganti nilai kosong menjadi kategori *unknown* agar baris data tersebut tetap dapat dimanfaatkan dalam proses pemodelan. Salinan data dengan nilai asli berbentuk teks tetap disimpan secara terpisah untuk keperluan tampilan pada kartu hasil rekomendasi dan antarmuka *website*.

Setelah data bersih, dilakukan *feature selection* terhadap atribut *genres*, *type*, *studios*, dan *rating* sebagai fitur pemodelan, sedangkan atribut *score* tidak dimasukkan sebagai fitur *input* untuk mencegah *data leakage* karena atribut tersebut digunakan sebagai dasar pembentukan label kelayakan. Atribut *genres* yang memiliki lebih dari satu nilai per baris ditransformasikan menggunakan *multi-label binarizer* menjadi representasi vektor biner dari 22 kategori genre unik, sedangkan atribut *type*, *studios*, dan *rating* ditransformasikan menggunakan *label encoding*. Selanjutnya dilakukan *class labeling* menggunakan pendekatan *mean thresholding* berdasarkan nilai rata-rata *score*, sehingga terbentuk label target Layak dan Tidak Layak yang digunakan untuk pelatihan *random forest*. Model *random forest* dilatih menggunakan parameter *n_estimators*=100 dan *random_state*=50, sedangkan model *K-Nearest Neighbor* (KNN) dibangun menggunakan pustaka *NearestNeighbors* dengan parameter *metric='cosine'* yang berfokus pada representasi vektor genre. Nilai *cosine distance* kemudian dikonversi menjadi *cosine similarity* melalui rumus 1 - *distance* sehingga diperoleh nilai *similarity* pada rentang 0 hingga 1.

3.1.2 Hasil Evaluasi Top-k

Evaluasi Top-K dilakukan untuk mengukur performa sistem rekomendasi berdasarkan daftar rekomendasi teratas yang dihasilkan untuk setiap data uji. Pengujian ini menggunakan tiga skenario pembagian data (70:30, 80:20, dan 90:10) dengan nilai K yang ditetapkan sebesar 10, 20, dan 30. Hasil pengujian ditunjukkan pada Tabel 2.

Tabel 2. Hasil Evaluasi Top-k

Split data	NDCG Top-K			*Precission* Top-K			Waktu Komputasi (detik)		
	10	20	30	10	20	30	10	20	30
70:30	**0,949**	**0,945**	0,942	0,800	0,800	0,800	1,862	1,981	2,342
80:20	0,895	0,910	0,915	0,800	0,800	0,800	1,742	1,913	2,151
90:10	0,931	0,943	**0,941**	0,800	**0,850**	**0,833**	**1,684**	**1,712**	**1,942**

Berdasarkan Tabel 2, sistem menunjukkan performa pemeringkatan yang sangat baik dan konsisten pada seluruh skenario pengujian. Nilai *Normalized Discounted Cumulative Gain* (NDCG) berada pada rentang 0,895 hingga 0,949, dengan nilai tertinggi sebesar 0,949 yang diperoleh pada skenario pembagian data 70:30 untuk pengujian Top-10. Nilai NDCG yang mendekati 1 menunjukkan bahwa sistem mampu mengurutkan anime yang paling relevan pada posisi teratas daftar rekomendasi secara optimal. Sementara itu, nilai *precision* menunjukkan hasil yang stabil. Pada skenario pembagian data 70:30 dan 80:20, nilai *precision* tetap sebesar 0,800 untuk seluruh nilai K, sedangkan pada skenario 90:10 nilai *precision* meningkat dari 0,800 pada Top-10 menjadi 0,850 pada Top-20, kemudian sedikit menurun menjadi 0,833 pada Top-30. Hasil tersebut menunjukkan bahwa sebagian besar anime yang direkomendasikan oleh sistem relevan. Pada sisi efisiensi, sistem mampu menghasilkan rekomendasi Top-K dengan waktu komputasi yang tergolong cepat, yaitu seluruh skenario pengujian dapat diselesaikan dalam waktu kurang dari 2,5 detik. Waktu komputasi tercepat dicapai pada skenario 90:10 untuk Top-10 sebesar 1,684 detik, sedangkan waktu terlama terjadi pada skenario 70:30 untuk Top-30 sebesar 2,342 detik. Secara keseluruhan, hasil evaluasi Top-K menunjukkan bahwa model *weighted hybrid* yang dikembangkan mampu menghasilkan rekomendasi anime yang relevan, memiliki kualitas pemeringkatan yang baik, serta efisien untuk diterapkan pada sistem rekomendasi berbasis *website*.

3.1.3 Hasil Evaluasi Global

Evaluasi global dilakukan untuk mengukur performa sistem secara menyeluruh terhadap seluruh data uji pada masing-masing skenario pembagian data, dengan menjadikan setiap anime pada data uji sebagai *query* secara bergantian. Hasil pengujian global ditunjukkan pada Tabel 3.

Tabel 3. Hasil Evaluasi Global

Split Data	*Precision*	NDCG	Waktu Komputasi (detik)
70:30	**0,756**	**0,977**	24,183
80:20	0,754	0,976	12,253
90:10	0,752	0,973	**4,464**

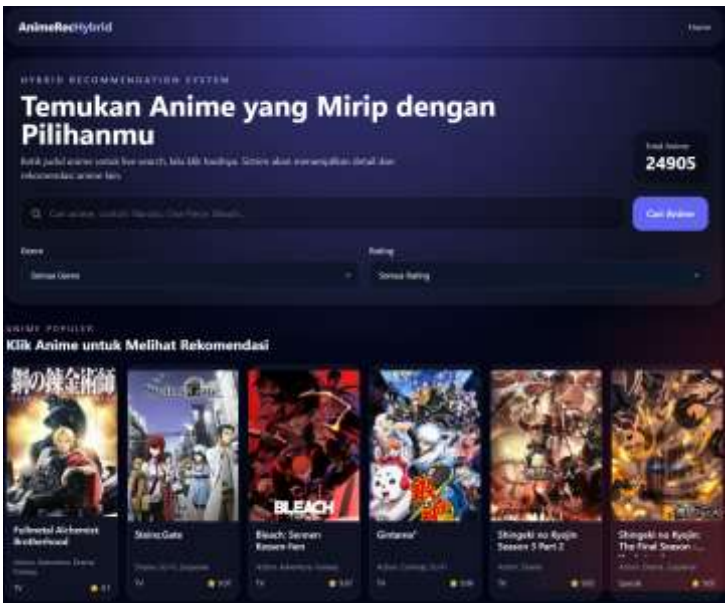
Berdasarkan Tabel 3, model menunjukkan performa yang sangat konsisten dan akurat dalam mengklasifikasikan serta mengurutkan tingkat relevansi anime pada skala besar. Nilai *precision* global berada pada rentang 0,752 hingga 0,756, dengan nilai tertinggi sebesar 0,756 yang diperoleh pada skenario pembagian data 70:30. Sementara itu, nilai NDCG global berada pada kisaran 0,973 hingga 0,977, dengan nilai tertinggi sebesar 0,977 pada skenario pembagian data 70:30. Nilai NDCG yang mendekati 1 menunjukkan bahwa sistem mampu mengurutkan anime berdasarkan tingkat relevansinya secara sangat optimal, meskipun evaluasi dilakukan terhadap seluruh data uji tanpa pembatasan jumlah item teratas. Dari sisi efisiensi, waktu komputasi pada evaluasi global menunjukkan

DOI: https://doi.org/10.69693/ijmst.v4i2. 12782

korelasi yang berbanding lurus dengan volume data uji yang diproses. Skenario pembagian data 70:30, yang memiliki proporsi data uji paling besar, membutuhkan waktu komputasi paling lama, yaitu 24,18 detik. Waktu tersebut kemudian menurun menjadi 12,253 detik pada skenario 80:20 dan mencapai waktu tercepat sebesar 4,464 detik pada skenario 90:10. Tren penurunan waktu komputasi ini menunjukkan bahwa semakin sedikit data uji yang diproses, semakin singkat waktu yang dibutuhkan sistem untuk menyelesaikan evaluasi global, sehingga model tetap memiliki efisiensi yang baik dalam menangani proses rekomendasi berskala besar.

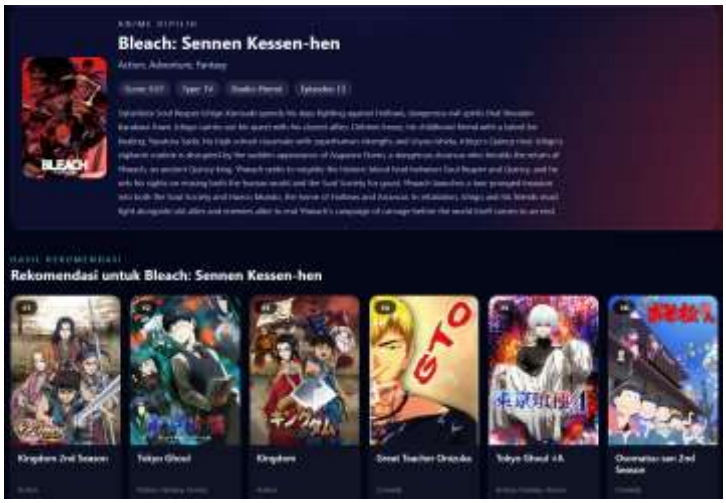
3.1.4 Implementasi Antarmuka *Website*

Sistem rekomendasi berbasis *website* dibangun menggunakan *framework Flask* berbasis *Python* dan *Tailwind* CSS dengan konsep modern dan responsif, sehingga pengguna dapat menggunakan sistem dengan nyaman melalui perangkat desktop maupun *mobile*. *Website* terdiri dari halaman utama (*home*), fitur pencarian anime, detail anime, serta halaman hasil rekomendasi Top-K. Gambar 4 menunjukkan tampilan halaman utama yang menjadi titik akses pertama bagi pengguna, dilengkapi fitur *live search* untuk mencari judul anime spesifik di antara 24.905 data yang tersedia, serta daftar anime populer dalam bentuk kartu yang berisi poster dan judul anime.



Gambar 4. Halaman Utama *Website*

Gambar 5 menunjukkan tampilan halaman hasil rekomendasi, yang menampilkan detail anime target berupa poster, judul, *genres*, *score*, *type*, studio, jumlah episode, dan sinopsis, diikuti daftar Top 10 rekomendasi dalam bentuk kartu yang disusun berdasarkan nilai *hybrid score* tertinggi. Semakin tinggi posisi anime pada daftar rekomendasi, semakin tinggi pula tingkat kemiripan dan relevansinya terhadap anime target yang dipilih pengguna.



Gambar 5. Halaman Hasil Rekomendasi

DOI: https://doi.org/10.69693/ijmst.v4i2. 12782

3.2 Pembahasan

Pembahasan dilakukan untuk menganalisis performa sistem berdasarkan hasil pengujian yang telah dilakukan serta membandingkannya dengan penelitian terdahulu yang memiliki keterkaitan metode maupun domain penelitian, sekaligus menganalisis pengaruh penggunaan pendekatan *hybrid* terhadap kualitas rekomendasi, efisiensi waktu komputasi, serta kelebihan dan hambatan yang ditemukan selama penelitian berlangsung.

3.2.1 Perbandingan dengan Penelitian Terdahulu

Penelitian ini dibandingkan dengan penelitian Alvina & Yamasari (2025) mengenai model rekomendasi lagu berbasis genre menggunakan algoritma *random forest* dan *decision tree*, yang menghasilkan nilai *precision* sebesar 0,56 tanpa proses *undersampling*. Perbandingan kedua penelitian ditunjukkan pada Tabel 4.

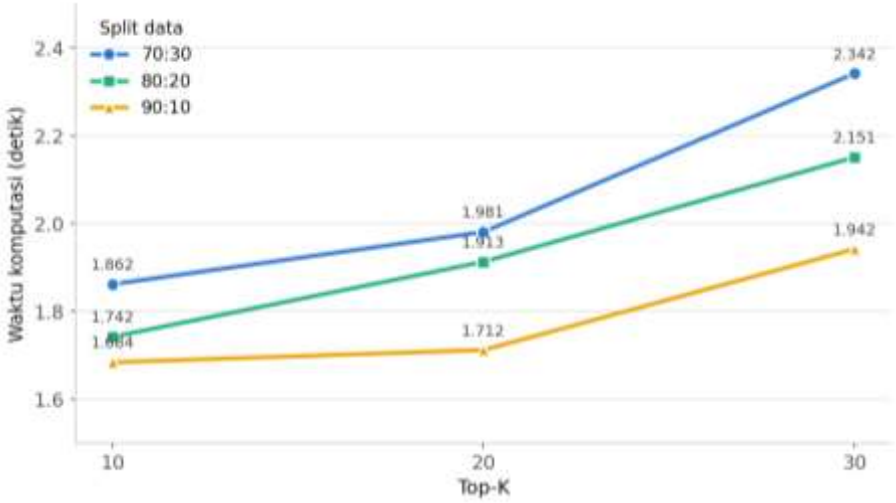
Tabel 4. Perbandingan dengan Penelitian Terdahulu

Aspek	Alvina & Yamasari (2025)	Penelitian Usulan
Objek penelitian	Lagu	Anime
Tujuan utama	Klasifikasi genre lagu	Peringkat rekomendasi anime
Algoritma	*Random Forest* dan *Decision Tree*	*Weighted Hybrid Random Forest* dan KNN
Pendekatan	Metode tunggal dan perbandingan algoritma	Penggabungan skor klasifikasi dan kemiripan
Metrik evaluasi	*Accuracy*, *Precision*, *Recall*, F1-*Score*	NDCG@K, *Precision*@K, waktu komputasi, evaluasi global
Hasil Precision	0,56	0,80-0,85 (Top-K) dan 0,752-0,756 (global)
Hasil NDCG	Tidak digunakan	0,895-0,949 (Top-K) dan 0,973-0,977 (global)

Meskipun nilai *precision* pada penelitian ini secara numerik lebih tinggi dibandingkan dengan penelitian Alvina & Yamasari (2025), kedua nilai tersebut tidak dapat dibandingkan secara langsung sebagai bukti keunggulan mutlak salah satu metode, karena kedua penelitian tersebut menggunakan objek, *dataset*, fitur, dan skenario pengujian yang berbeda. Perbedaan utama penelitian ini terletak pada penggunaan pendekatan *weighted hybrid* dan evaluasi berbasis peringkat menggunakan *Normalized Discounted Cumulative Gain* (NDCG), yang memungkinkan penelitian ini mengevaluasi tidak hanya jumlah anime yang relevan, tetapi juga posisi anime yang relevan dalam daftar rekomendasi. Dengan demikian, kontribusi penelitian ini bukan untuk membuktikan bahwa pendekatan *weighted hybrid* secara mutlak lebih baik daripada metode pada penelitian terdahulu, melainkan untuk menunjukkan bahwa integrasi *random forest* dan *K-Nearest Neighbor* (KNN) dapat diterapkan untuk menghasilkan rekomendasi anime berbasis peringkat yang dapat dievaluasi secara komprehensif menggunakan NDCG, *precision*, waktu komputasi, serta evaluasi global.

3.2.2 Analisis Waktu Komputasi

Analisis waktu komputasi dilakukan untuk mengetahui efisiensi sistem rekomendasi anime yang dibangun. Pengukuran waktu mencakup proses pelatihan *random forest*, prediksi kelayakan anime, perhitungan *cosine similarity*, penggabungan skor *weighted hybrid*, pengurutan hasil rekomendasi, serta perhitungan nilai NDCG dan *precision*. Gambar 6 menunjukkan perbandingan waktu komputasi pada evaluasi Top-K, sedangkan Gambar 7 menunjukkan perbandingan waktu komputasi pada evaluasi global.

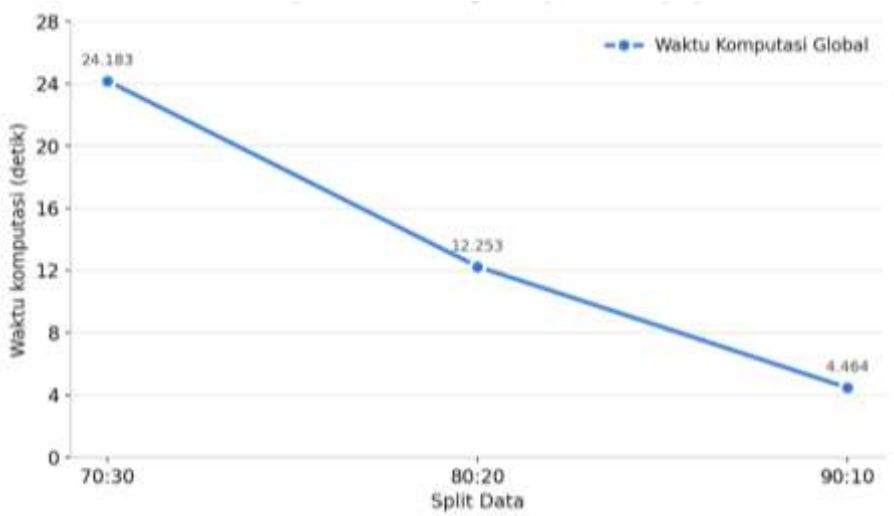


Gambar 6. Perbandingan Waktu Komputasi Top-K

Berdasarkan Gambar 6, terlihat adanya tren kenaikan waktu komputasi yang bersifat linier dan proporsional seiring bertambahnya nilai K pada seluruh skenario pengujian. Hal ini menunjukkan bahwa semakin banyak jumlah item

DOI: https://doi.org/10.69693/ijmst.v4i2. 12782

rekomendasi yang dihasilkan, semakin besar pula beban pemrosesan yang harus dilakukan oleh sistem, meskipun peningkatan waktu tersebut berlangsung secara stabil. Sementara itu, terdapat perbedaan waktu komputasi pada setiap skenario pembagian data. Skenario 70:30 mencatatkan waktu komputasi paling tinggi untuk seluruh nilai K, sedangkan skenario 90:10 menunjukkan efisiensi komputasi terbaik dengan waktu pemrosesan paling rendah. Meskipun waktu komputasi meningkat seiring bertambahnya nilai K, seluruh proses pada pengujian Top-K tetap dapat diselesaikan dalam waktu kurang dari 2,5 detik, sehingga menunjukkan bahwa model yang dikembangkan memiliki performa yang efisien dan responsif untuk diterapkan pada sistem rekomendasi secara *real-time*.



Gambar 7. Perbandingan Waktu Komputasi Global

Pada Gambar 7 terlihat tren penurunan waktu komputasi yang drastis seiring perubahan skenario pembagian data dari 70:30 hingga 90:10, karena semakin sedikit data uji yang diproses, semakin singkat durasi yang diperlukan sistem untuk menyelesaikan kalkulasi kemiripan secara menyeluruh. Ditinjau dari kompleksitas algoritmanya, proses klasifikasi *random forest* memiliki kompleksitas waktu rata-rata $O(N \times K \times \log N)$, di mana N menyatakan jumlah data anime dan K menyatakan jumlah pohon keputusan. Dengan demikian, peningkatan waktu bersifat logaritmik dan masih tergolong efisien. Sementara itu, perhitungan *cosine similarity* pada *K-Nearest Neighbor* (KNN) memiliki kompleksitas waktu yang lebih besar pada kondisi terburuk, yaitu $O(N^2 \times M)$, dengan M menyatakan jumlah fitur genre unik, karena setiap anime pada data uji harus dibandingkan kemiripannya dengan seluruh anime lain dalam *dataset* secara iteratif. Hasil analisis kompleksitas ini sejalan dengan hasil pengujian waktu komputasi, yang mengonfirmasi bahwa volume data uji memegang peranan krusial terhadap beban komputasi algoritma dalam menghitung skor *hybrid* secara global

3.2.3 Kelebihan dan Keterbatasan

Sistem rekomendasi anime menggunakan pendekatan *weighted hybrid* dengan algoritma *random forest* dan KNN memiliki kelebihan utama berupa kemampuannya mengintegrasikan hasil klasifikasi kelayakan dan perhitungan kemiripan konten secara simultan, yang dibuktikan dengan nilai *precision* mencapai 0,97 dan *Normalized Discounted Cumulative Gain* (NDCG) sebesar 0,94. Kelebihan lainnya adalah efisiensi waktu komputasi, karena sistem mampu menghasilkan rekomendasi dalam waktu kurang dari 25 detik bahkan pada evaluasi global, sehingga pendekatan ini sangat layak diterapkan pada *platform* berbasis *website* tanpa menyebabkan latensi yang mengganggu pengalaman pengguna.

Pada sisi lain, penelitian ini memiliki beberapa hambatan yang perlu diperhatikan. Penggunaan atribut yang terbatas pada *genres*, *type*, *studios*, dan *rating* menyebabkan representasi preferensi pengguna menjadi kurang optimal, karena sebagai sistem rekomendasi berbasis *content-based*, hasil rekomendasi sangat bergantung pada kemiripan atribut yang tersedia dalam *dataset*. Hambatan mendasar lainnya adalah ketiadaan data interaksi pengguna, seperti riwayat tontonan, *rating*, atau daftar favorit, sehingga sistem belum mampu mempelajari preferensi personal secara mendalam sebagaimana yang dapat dicapai oleh metode *collaborative filtering*, sebagaimana ditunjukkan oleh Rachmaniar dkk. (2025) dan Yanisa Putri dkk. (2024) yang menggabungkan *content-based filtering* dan *collaborative filtering* untuk meningkatkan personalisasi rekomendasi. Selain itu, ketergantungan pada layanan penerjemahan eksternal untuk sinopsis anime menjadi kendala teknis yang dapat menambah waktu proses saat sistem dioperasikan secara *daring*. Meskipun demikian, sistem yang dibangun telah terbukti mampu memberikan rekomendasi yang relevan dengan performa yang kompetitif.

DOI: https://doi.org/10.69693/ijmst.v4i2. 12782

4. Kesimpulan

Berdasarkan hasil implementasi dan pengujian sistem rekomendasi anime menggunakan pendekatan *weighted hybrid* dengan algoritma *random forest* dan KNN, dapat disimpulkan bahwa pendekatan yang menggabungkan skor klasifikasi dari algoritma *random forest* dan skor kemiripan dari *K-Nearest Neighbor* (KNN) dapat digunakan untuk menghasilkan rekomendasi anime. Penggabungan kedua metode tersebut menghasilkan daftar rekomendasi berdasarkan tingkat kemiripan anime serta bobot kelayakan yang diperoleh dari proses klasifikasi. Hasil evaluasi menunjukkan bahwa sistem mampu menghasilkan rekomendasi yang relevan dalam berbagai skenario pengujian. Pada pengujian Top-K, sistem memperoleh nilai *precision* tertinggi sebesar 85% dan nilai NDCG sebesar 94,9%. Sementara itu, pada evaluasi keseluruhan, sistem memperoleh nilai *precision* sebesar 75,6% dan NDCG sebesar 97,7%, yang menunjukkan bahwa urutan rekomendasi yang dihasilkan sudah mendekati urutan yang diharapkan. Pada sisi waktu pemrosesan, sistem dapat menghasilkan rekomendasi dalam waktu kurang dari 2,5 detik pada pengujian Top-K dan waktu tercepat sebesar 4,464 detik pada skenario pembagian data 90:10. Untuk pengembangan selanjutnya, sistem rekomendasi disarankan lebih berfokus pada integrasi data interaksi pengguna, seperti riwayat tontonan, daftar anime favorit, dan pemberian *rating* secara personal, agar sistem tidak hanya bergantung pada kemiripan atribut konten, tetapi juga mampu mempelajari preferensi unik setiap pengguna sehingga menghasilkan luaran yang lebih akurat, personal, dan efisien secara luas.

Ucapan Terimakasih

Penulis menyampaikan terima kasih kepada Bapak Ilham Saifudin, S.Kom., M.Kom., selaku dosen pembimbing yang telah memberikan arahan, bimbingan, dan masukan selama proses penelitian. Ucapan terima kasih juga disampaikan kepada Program Studi Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Jember atas dukungan dan fasilitas yang telah disediakan

Reference

Aini, N., Arif, M., Binti Toyibah, Z., & Tri Agustin, I. (2024). Implementasi Metode Hybrid Recommendation untuk Sistem Rekomendasi Mitra MSIB di Prodi Pendidikan Informatika. *Jurnal Ilmiah Edutic : Pendidikan Dan Informatika*, *10*(2), 171–180. https://doi.org/10.21107/edutic.v10i2.23920

Alvina, P., & Yamasari, Y. (2025). Model Rekomendasi Lagu Berbasis Genre Menggunakan Metode Random Forest Dan Decision Tree. *Journal of Informatics and Computer Science*, *07*. https://doi.org/10.26740/jinacs.v7n02.p267-275

Armenta-Segura, J., & Sidorov, G. (2025). Anime popularity prediction before huge investments: a multimodal approach using deep learning. *PeerJ Computer Science*, *11*. https://doi.org/10.7717/peerj-cs.2715.

Brema Daniel Ginting, Yusfrizal Yusfrizal, & Lina Arliana Nur Kadim. (2024). Penerapan Algoritma K-Nearest Neighbor untuk Klasifikasi Usaha Masyarakat Berdasarkan Jenis Izin Usaha. *Modem : Jurnal Informatika Dan Sains Teknologi.*, *2*(4), 92–101. https://doi.org/10.62951/modem.v2i4.233.

Dzikri, M., Ilyasa, H., & Yamasari, Y. (2023). Perbandingan Cosine Similarity Dan Euclidean Distance Pada Model Rekomendasi Buku dengan Metode Item-Based Collaborative Filtering. *Journal of Informatics and Computer Science*, *04*. https://doi.org/10.26740/jinacs.v4n03.p264-274

Fauzi, M. R., Sanjaya, I., & Kharisma, I. L. (2025). Genre-Based Anime Recommendation System Using KNN with Fanbase Bias Detection. *Bit-Tech*, *8*(2), 1386–1396. https://doi.org/10.32877/bt.v8i2.2917.

Gomez, R. G., Tringali, B., Baker, C., & Stone, K. (2025). The Power of Anime: Using Anime for Education and Outreach in STEM. *Frontiers in Education*, *10*. https://doi.org/10.3389/feduc.2025.1707055.

Hasegawa, M., & Masuda, H. (2024). *Anime Industry Report 2023*. Industry Profile. The Association of Japanese Animations. https://aja.gr.jp/english/japan-anime-data/

Husaini, B. Q., & Jemakmun, J. (2023). Penerapan Algoritma Decision Tree C45 untuk Klasifikasi Penjurusan Siswa. *Jurnal Teknologi Informatika Dan Komputer*, *9*(1), 455–470. https://doi.org/10.37012/jtik.v9i1.1512

Iddrus, & Sari, D. W. (2023). Penerapan Data Mining Menggunakan Algoritma Decision Tree C4.5 Untuk Memprediksi Mahasiswa Drop Out Di Universitas Wiraraja. *Jurnal Advance Research Informatika*, (2), 1–7. https://doi.org/10.24929/jars.v1i02.2684

Kumala Sari, V. (2024). Dampak E-commerce Terhadap Perkembangan Digital. *Jurnal Akademik Ekonomi Dan Manajemen*, *1*(4), 18–24. https://doi.org/10.61722/jaem.v1i4.3113.

Kurniadi, D., Sutedi, A., Nursyaban, D., & Mulyani, A. (2024). Sistem Rekomendasi Pemilihan Pengepul Limbah di PT. Pituku Cordova International Menggunakan Algoritma *Haversine*. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *11*(1), 155–166. https://doi.org/10.25126/jtiik.202411117694.

Mahendra Alvanof, M., & Kesuma Dinata, R. (2024). Penerapan Algoritma Random Forest dalam Deteksi dan Klasifikasi Ransomware. *Jurnal Elektronika dan Teknologi Informasi*, *5*(2). https://doi.org/10.5201/jet.v5i2.488

Mia, P., Utami, S., Prayana Trisna, N., & Vihikan, W. O. (2025). Web-Based Makeup Recommendation System Using Hybrid Filtering. *Journal of Applied Informatics and Computing (JAIC)*, *9*(3). https://doi.org/10.30871/jaic.v9i3.9339

Michael Lauw, C., Hairani, H., Saifuddin, I., Ximenes Guterres, J., Maariful Huda, M., & Mayadi, M. (2023). Combination of Smote and Random Forest Methods for Lung Cancer Classification. *International Journal of Engineering and Computer Science Applications (IJECSA)*, *2*(2), 59–64. https://doi.org/10.30812/ijecsa.v2i2.3333.

Nur, F., Ahsan, M., & Harianto, W. (2022). Komparasi Tingkat Akurasi Information Gain Dan Gain Ratio Pada Metode K-Nearest Neighbor. *Jurnal Mahasiswa Teknik Informatika*, *6*(1). https://doi.org/10.36040/jati.v6i1.4694

Pratama, R. A., Safi'i, Y., Nugraha, M. A., Sobihah, A. S., & Ifada, N. (2025). Perbandingan User-Based dan Item-Based pada Sistem Rekomendasi Film Kombinasi Teknik Reduksi Dimensi dan Clustering. *Jurnal Tekno Insentif*, *19*(1), 1–14. https://doi.org/10.36787/jti.v19i1.1662.

Rachmaniar, A., Widayati, S., & Rokoyah, K. (2025). Sistem Rekomendasi Produk E-Commerce Menggunakan Collaborative Filtering Dan Content-Based Filtering. *Journal of Information System, Informatics and Computing Issue Period*, *9*(1), 40–54. https://doi.org/10.52362/jisicom.v9i1.1904

DOI: https://doi.org/10.69693/ijmst.v4i2. 12782

Saifudin, I., & Widiyaningtyas, T. (2024). Systematic Literature Review on Recommender System: Approach, Problem, Evaluation Techniques, Datasets. *IEEE Access*, *12*, 19827–19847. https://doi.org/10.1109/ACCESS.2024.3359274

Setiawan, N., Hendra, J. M., Wijaya, B., Qomariyah, N. N., & Astriani, M. S. (2023). Time Series Model to Predict Future Popular Animes Genres in 2025. *E3S Web of Conferences*, *388*. https://doi.org/10.1051/e3sconf/202338802002.

Setio, P. B. N., Saputro, D. R. S., & Winarno, B. (2020). *PRISMA, Prosiding Seminar Nasional Matematika: Klasifikasi dengan Pohon Keputusan Berbasis Algoritme C4.5. 3*, 64–71. https://journal.unnes.ac.id/sju/index.php/prisma/article/view/37650

Sonule, A., Jagtap, H., & Mendhe, V. (2024). Weighted Hybrid Recommendation System. *International Journal of Research and Analytical Reviews*, 402. https://doi.org/10.56975/ijrar.v11i1.284204

Suci Amaliah, Nusrang, M., & Aswi, A. (2022). Penerapan Metode Random Forest Untuk Klasifikasi Varian Minuman Kopi di Kedai Kopi Konijiwa Bantaeng. *VARIANSI: Journal of Statistics and Its Application on Teaching and Research*, *4*(3), 121–127. https://doi.org/10.35580/variansiunm31.

Yanisa Putri, K. S., I Made Agus Dwi Suarjaya, & Wayan Oger Vihikan. (2024). Sistem Rekomendasi Skincare Menggunakan Metode Content Based Filtering dan Collaborative Filtering. *Decode: Jurnal Pendidikan Teknologi Informasi*, *4*(3), 764–774. https://doi.org/10.51454/decode.v4i3.601

DOI: https://doi.org/10.69693/ijmst.v4i2. 12782