

PROPOSAL SKRIPSI

**PENGEMBANGAN SISTEM REKOMENDASI TEMPAT WISATA DI
KOTA SURABAYA MENGGUNAKAN NEURAL COLLABORATIVE
FILTERING DAN CONTENT-BASED FILTERING BERBASIS MOBILE**



Oleh:

Naufal Duta Maulana

21081010116

PROGRAM STUDI INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"

JAWA TIMUR

2024

BAB I

PENDAHULUAN

1.1. Latar Belakang

Surabaya merupakan salah satu kota terbesar kedua setelah kota Jakarta. Kota ini terletak di provinsi Jawa Timur dan juga sebagai kota metropolis. Kota ini terus berkembang dari tahun ke tahun bahkan kota ini juga sebagai pusat pertumbuhan ekonomi dan bisnis. Di Surabaya ini terdapat gedung-gedung pencakar langit yang banyak [1]. Surabaya merupakan ibu kota Provinsi Jawa Timur yang memiliki julukan sebagai Kota Pahlawan dan memiliki berbagai macam tempat destinasi wisata. Terdapat berbagai macam destinasi wisata yang begitu menarik mulai dari wisata alam, wisata air, dan berbagai wisata lainnya yang ada di Surabaya beberapa tempat wisata yang ada di Kota Surabaya diantaranya adalah Tugu Pahlawan, Wisata Atlantis Land Surabaya, kebun binatang Surabaya, Food Junction Pakuwon, Suroboyo Carnival, Mangrove, dan lain sebagainya (Mukaromah, 2019).

Tempat-tempat wisata di Surabaya memiliki makna dan sejarah yang khas, menjadikannya destinasi populer bagi sekolah dan rombongan study tour yang ingin menggabungkan liburan dengan pengetahuan. Namun, wisatawan biasanya memiliki pengetahuan yang terbatas tentang kota yang akan dikunjungi dan mereka tidak mengetahui tempat-tempat artistik, sosial, atau hiburan lokal. Seorang wisatawan dapat menemukan sejumlah besar informasi tentang kota tersebut, tetapi ia mungkin akan menghabiskan waktu lama untuk memilih kegiatan yang ia sukai dan mengaturnya untuk menghabiskan waktu satu hari secara efisien (Sebastia et al., 2008).

Sistem rekomendasi adalah suatu sistem yang menyarankan informasi yang berguna atau menduga apa yang akan dilakukan pelanggan untuk mencapai tujuannya, misalnya seperti memilih produk tertentu. Sehingga pelanggan memilih produk lebih efektif dalam menentukan produk yang diinginkannya (Devi Nurhayati & Widayani, 2021). Algoritma yang umum digunakan dapat dibagi menjadi dua kategori utama: metode *content-based* dan metode *collaborative filtering*. Metode *content-based* membangun potret pengguna dan item melalui analisis informasi tambahan, seperti konten dokumen, profil pengguna dan atribut item, untuk membuat rekomendasi (Zhang et al., 2018). Collaborative Filtering adalah pendekatan yang memberikan sebuah rekomendasi kepada pengguna berdasarkan preferensi pada profil (Nisha & Mohan, 2019). Hybrid Recommender System adalah pendekatan yang merupakan kombinasi dari dua pendekatan sebelumnya (Zhang et al., 2018).

Evolusi pengimplementasian metode recommender system berlanjut hingga saat ini. Deep learning yang merupakan salah satu penemuan dalam metode recommender system dapat mengatasi beberapa kekurangan dari teknologi lain dan merevolusi arsitektur rekomendasi untuk meningkatkan kinerja pemberian rekomendasi (Mah et al., 2023). Maka dari itu, penelitian ini dilakukan menggunakan metode Neural Collaborative Filtering dan Content-based filtering untuk mengatasi tantangan ini. Pendekatan dengan metode *hybrid* ini memungkinkan untuk menghasilkan penilaian dan akurasi yang lebih baik dan efektif, dengan kelebihan masing masing metode yang digunakan.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang dijelaskan sebelumnya, masalah yang akan dibahas dapat dirumuskan sebagai berikut:

1. Bagaimana mengimplementasikan Neural Collaborative Filtering dan Content-based Filtering pada aplikasi berbasis mobile?
2. Berapa hasil perhitungan yang diperoleh dari penerapan Neural Collaborative Filtering dan Content-based Filtering pada data uji yang digunakan?

1.3. Tujuan Penelitian

Berdasarkan rumusan masalah yang ada, maka tujuan dari penelitian yang dilakukan ini adalah:

1. Mengembangkan sistem rekomendasi tempat wisata berbasis mobile yang mengimplementasikan metode Neural Collaborative Filtering dan Content-Based Filtering untuk meningkatkan pengalaman pengguna dalam memilih tempat wisata di Surabaya.
2. Menghitung dan mengevaluasi efektivitas metode Neural Collaborative Filtering dan Content-Based Filtering dalam menghasilkan rekomendasi tempat wisata yang akurat dan sesuai dengan kebutuhan pengguna.

1.4. Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan kemudahan dan manfaat bagi pengguna dalam memilih tempat objek wisata di Kota Surabaya, antara lain:

1. Manfaat bagi Penulis

- a. Memperdalam pemahaman tentang teknik-teknik dalam pengembangan sistem rekomendasi, khususnya dalam konteks pariwisata dan teknologi informasi.
 - b. Hasil penelitian dapat berkontribusi pada literatur akademik mengenai sistem rekomendasi dan pariwisata, serta menjadi referensi bagi penelitian selanjutnya di bidang yang sama.
2. Manfaat bagi Pengguna
- a. Sistem rekomendasi yang dikembangkan dapat membantu wisatawan dalam menemukan tempat wisata yang sesuai dengan preferensi mereka, sehingga meningkatkan pengalaman berwisata di Surabaya.
 - b. Dengan meningkatnya jumlah pengunjung ke tempat-tempat wisata, diharapkan akan ada dampak positif terhadap ekonomi lokal melalui peningkatan pendapatan dari sektor pariwisata.

1.5. Batasan Masalah

Agar penelitian ini dapat dilakukan dengan efektif dan tidak terlalu meluas dari permasalahan yang dikaji, maka perlu dilakukan pembatasan masalah yang dibahas. Batasan yang diterapkan dalam penelitian ini sebagai berikut:

1. Penelitian ini hanya akan berfokus pada tempat wisata yang berada di Kota Surabaya. Tempat wisata di daerah lain tidak akan menjadi bagian dari analisis ini.
2. Sistem rekomendasi yang dikembangkan akan menggunakan dua metode utama, yaitu Neural Collaborative Filtering dan Content-Based Filtering.
3. Penelitian ini akan berfokus pada pengembangan aplikasi berbasis mobile. Implementasi sistem rekomendasi dalam bentuk website atau platform lainnya tidak akan menjadi fokus dalam penelitian ini.
4. Rekomendasi yang diberikan oleh sistem akan didasarkan pada preferensi pengguna yang diambil dari data interaksi sebelumnya dan karakteristik tempat wisata, tanpa mempertimbangkan faktor eksternal seperti cuaca atau event khusus yang mungkin mempengaruhi keputusan wisatawan.

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Berdasarkan penelitian sebelumnya, sistem rekomendasi dengan Neural Collaborative Filtering (NCF) telah menunjukkan efektivitas dalam memberikan rekomendasi yang akurat dengan menggunakan data interaksi antara pengguna dan item. Penelitian Ni'mah et al. (2023) menggunakan metode NCF pada sistem rekomendasi film berbasis dataset MovieLens. Penelitian ini menunjukkan bahwa penerapan NCF mampu menghasilkan prediksi dengan tingkat akurasi tinggi berdasarkan nilai RMSE. Hasil evaluasi menunjukkan bahwa sistem rekomendasi menggunakan metode Neural Collaborative Filtering (NCF) mencapai rata-rata nilai RMSE 0,91467. Model ini terbukti efektif dalam menangani masalah sparsity data dan memberikan prediksi yang akurat melalui penggunaan parameter optimal seperti learning rate sebesar 0,001 dan batch size 1024.

Berdasarkan penelitian terdahulu oleh Bambang Tri Wahyu Utomo et al. (2015), sistem rekomendasi paket wisata di Malang Raya menggunakan pendekatan hybrid yang menggabungkan metode Content-Based Filtering dan Collaborative-Based Filtering dengan algoritma Nearest Neighbor (NN). Metode hybrid ini dirancang untuk mengatasi kelemahan masing-masing metode, seperti ketergantungan pada data eksplisit pada Collaborative Filtering dan terbatasnya cakupan rekomendasi pada Content-Based Filtering. Didapatkan nilai hasil ketepatan untuk metode hybrid (content based-collaborative) 77% mendekati dengan hasil perhitungan sistem dan nilai hasil ketepatan. Penelitian ini menegaskan bahwa penggabungan metode mampu mengatasi keterbatasan masing-masing pendekatan individu dan menghasilkan rekomendasi yang relevan bagi pengguna.

2.2. Sistem Rekomendasi

Recommender system atau sistem rekomendasi adalah sebuah teknik untuk membantu seorang user dalam mencari sebuah item berdasarkan ketertarikan masing-masing dalam memberikan efisiensi waktu. Sejak tahun 1990, sistem pemberi rekomendasi diterima secara luas dalam mengatasi kesulitan penyaringan data yang paling relevan dari sekumpulan informasi dalam jumlah besar dan kompleks (CC & Mohan, 2019).

Terdapat tiga pendekatan umum pada recommender system yang telah banyak digunakan dalam penelitian hingga saat ini, yaitu pendekatan Content-based Filtering, pendekatan Collaborative Filtering, dan pendekatan Hybrid Recommendation System (Zhang et al., 2018). Content-based Filtering adalah pendekatan yang menggunakan kesamaan konten

item untuk memberikan rekomendasi kepada pengguna tanpa mengandalkan informasi sebelumnya (rating) (Zhang et al., 2018). Collaborative Filtering adalah pendekatan yang memberikan sebuah rekomendasi kepada pengguna berdasarkan preferensi pada profil (CC & Mohan, 2019). Hybrid Recommender System adalah pendekatan yang merupakan kombinasi dari dua pendekatan sebelumnya (Zhang et al., 2018).

2.3. Neural Collaborative Filtering

Neural Collaborative Filtering (NCF) adalah metode modern dalam sistem rekomendasi yang memanfaatkan jaringan saraf untuk memodelkan interaksi antara pengguna dan item. NCF mampu menangkap pola kompleks dalam data interaksi dengan menggunakan representasi latent vector untuk pengguna dan item. Metode ini mengatasi beberapa kelemahan dari pendekatan tradisional dengan memberikan prediksi yang lebih akurat melalui arsitektur jaringan saraf yang dalam. NCF juga dapat mengatasi masalah cold start dengan memanfaatkan informasi tambahan dari fitur item dan pengguna (Zhang et al., 2019).

2.4. Content-Based Filtering

Content-Based Filtering (CBF) adalah metode rekomendasi yang menganalisis karakteristik konten dari item untuk memberikan saran kepada pengguna. CBF membangun profil pengguna berdasarkan atribut atau fitur item yang telah disukai sebelumnya, seperti genre film atau jenis makanan. Dengan menggunakan teknik seperti TF-IDF atau analisis semantik, CBF dapat merekomendasikan item baru yang memiliki kesamaan dengan item favorit pengguna. Kelebihan CBF adalah kemampuannya untuk memberikan rekomendasi tanpa memerlukan data interaksi dari pengguna lain, sehingga sangat berguna dalam situasi cold start.

2.5. Pariwisata di Kota Surabaya

Kota Surabaya merupakan salah satu destinasi wisata utama di Indonesia dengan berbagai atraksi budaya, sejarah, dan kuliner. Sebagai kota terbesar kedua di Indonesia, Surabaya menawarkan banyak tempat wisata yang menarik bagi wisatawan lokal maupun internasional. Penelitian mengenai sistem rekomendasi dalam konteks pariwisata di Surabaya penting untuk membantu wisatawan menemukan tempat-tempat menarik sesuai dengan preferensi mereka. Dengan mengembangkan sistem rekomendasi berbasis NCF dan CBF, penelitian ini bertujuan untuk meningkatkan pengalaman wisatawan dalam menjelajahi keindahan kota Surabaya serta mendukung pengembangan sektor pariwisata lokal

BAB III

METODOLOGI PENELITIAN

3.1. Pendekatan Penelitian

Pendekatan penelitian ini menggunakan metode kuantitatif dengan desain eksperimental. Penelitian ini bertujuan untuk mengembangkan dan mengevaluasi sistem rekomendasi tempat wisata di Kota Surabaya dengan mengintegrasikan Neural Collaborative Filtering (NCF) dan Content-Based Filtering (CBF). Data yang dikumpulkan akan dianalisis secara statistik untuk menilai akurasi dan efektivitas sistem rekomendasi yang dikembangkan.

3.2. Pengumpulan Data

Pengumpulan data pada penelitian ini dilakukan menggunakan teknik web scraping, yang merupakan proses otomatisasi pengambilan data dari situs web dengan struktur tertentu. Tujuannya adalah untuk mendapatkan informasi detail terkait tempat wisata di Kota Surabaya, yang menjadi objek penelitian ini. Situs web yang dijadikan sumber data merupakan Google Maps.

Selama proses scraping, perhatian khusus diberikan pada kepatuhan terhadap kebijakan penggunaan situs web (terms of service) untuk menghindari pelanggaran etika digital. Kecepatan scraping juga diatur menggunakan teknik rate-limiting agar tidak membebani server dari situs target. Data hasil scraping disimpan dalam format terstruktur seperti CSV atau JSON untuk kemudian digunakan dalam tahap preprocessing data, pengembangan model, dan evaluasi sistem rekomendasi. Dengan pendekatan ini, data yang dikumpulkan diharapkan mencerminkan preferensi aktual pengguna dan karakteristik tempat wisata yang relevan di Kota Surabaya.

3.3. Pengembangan Model

Pengembangan model dalam penelitian ini melibatkan proses membangun arsitektur sistem rekomendasi berdasarkan metode Neural Collaborative Filtering (NCF) dan Content-Based Filtering (CBF). Model NCF dikembangkan menggunakan pendekatan deep learning, dengan memanfaatkan embedding untuk merepresentasikan interaksi antara pengguna dan tempat wisata. Proses meliputi pembentukan matriks interaksi pengguna dan item, desain jaringan saraf tiruan dengan lapisan tersembunyi (hidden layers), serta pelatihan model menggunakan data rating pengguna terhadap tempat wisata. Untuk Content-Based Filtering, model dirancang menggunakan fitur-fitur tempat wisata seperti kategori, deskripsi, dan fasilitas. Penggunaan algoritma cosine similarity membantu mengukur kesesuaian antara profil

pengguna dan atribut tempat wisata. Model NCF dan CBF dikembangkan secara terpisah sebelum diintegrasikan untuk membentuk sistem rekomendasi yang komprehensif.

3.4. Integrasi Metode

Proses integrasi metode bertujuan untuk menggabungkan kelebihan dari kedua pendekatan, yaitu Neural Collaborative Filtering (NCF) dan Content-Based Filtering (CBF), dalam menghasilkan rekomendasi yang lebih akurat dan relevan. Hasil prediksi dari NCF yang berbasis interaksi pengguna digunakan sebagai input untuk menyempurnakan hasil rekomendasi dari CBF yang berbasis atribut tempat wisata. Metode hybrid ini dirancang untuk mengatasi kelemahan masing-masing pendekatan, seperti sparsity data pada NCF dan keterbatasan eksplorasi item baru pada CBF. Proses integrasi dilakukan dengan pendekatan weighted hybrid, di mana bobot untuk masing-masing metode ditentukan berdasarkan performa mereka dalam tahap validasi. Model hybrid ini kemudian diimplementasikan ke dalam aplikasi berbasis mobile untuk memberikan pengalaman rekomendasi yang optimal kepada pengguna.

3.5. Evaluasi Sistem

Evaluasi sistem dilakukan untuk menilai performa model rekomendasi dalam menghasilkan rekomendasi yang relevan dan akurat. Proses evaluasi melibatkan pengukuran metrik seperti Root Mean Square Error (RMSE) untuk menilai tingkat kesalahan prediksi, serta MAE untuk mengukur relevansi rekomendasi. Sistem diuji menggunakan data validasi dan data uji yang diperoleh dari proses split dataset. Selain evaluasi kuantitatif, dilakukan juga uji coba pada aplikasi mobile untuk memastikan sistem berjalan sesuai dengan ekspektasi pengguna dari segi kecepatan, tampilan, dan kemudahan penggunaan. Hasil evaluasi digunakan untuk menyempurnakan parameter model dan memastikan implementasi yang optimal sebelum sistem dirilis kepada pengguna.

DAFTAR PUSTAKA

- Devi Nurhayati, S., & Widayani, W. (2021). Sistem Rekomendasi Wisata Kuliner di Yogyakarta dengan Metode Item-Based Collaborative Filtering Yogyakarta Culinary Recommendation System with Item-Based Collaborative Filtering Method. In *JACIS : Journal Automation Computer Information System* (Vol. 1, Issue 2). <https://manganenakyog.my.id/>,
- Mah, N. ', Ayyiyah, K., Kusumaningrum, R., & Rismiyati, R. (2023). *FILM RECOMMENDER SYSTEM MENGGUNAKAN METODE NEURAL COLLABORATIVE FILTERING FILM RECOMMENDER SYSTEM USING NEURAL COLLABORATIVE FILTERING METHOD*. 10(3), 699–708. <https://doi.org/10.25126/jtiik.2023106616>
- Mukaromah, M. (2019). PENERAPAN METODE FUZZY SUGENO UNTUK MENENTUKAN JALUR TERBAIK MENUJU LOKASI WISATA DI SURABAYA. *Jurnal Matematika Sains Dan Teknologi*, 20(2), 95–101. <https://doi.org/10.33830/jmst.v20i2.187.2019>
- Nisha, C. C., & Mohan, A. (2019). A social recommender system using deep architecture and network embedding. *Applied Intelligence*, 49(5), 1937–1953. <https://doi.org/10.1007/s10489-018-1359-z>
- Sebastia, L., Garcia, I., Onaindia, E., & Guzman, C. (2008). e-Tourism : A tourist recommendation and planning application. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, 2, 89–96. <https://doi.org/10.1109/ICTAI.2008.18>
- Zhang, L., Luo, T., Zhang, F., & Wu, Y. (2018). A Recommendation Model Based on Deep Neural Network. *IEEE Access*, 6, 9454–9463. <https://doi.org/10.1109/ACCESS.2018.2789866>

FILM RECOMMENDER SYSTEM MENGGUNAKAN METODE NEURAL COLLABORATIVE FILTERING

Ni'mah Khoiriyah Ayyiyah¹, Retno Kusumaningrum^{*2}, Rismiyati Rismiyati³

^{1, 2, 3}Universitas Diponegoro, Semarang

Email: ¹nimahkazizah@students.undip.ac.id, ²retno@live.undip.ac.id, ³rismiyati@live.undip.ac.id,

^{*}Penulis Korespondensi

(Naskah masuk: 14 Oktober 2022, diterima untuk diterbitkan: 21 Juni 2023)

Abstrak

Pada saat ini media hiburan telah berkembang pesat dan tersedia secara digital. Hiburan khususnya dalam bentuk film semakin tersedia secara luas. Keinginan untuk menikmati hiburan dalam media digital mendorong pengguna internet lain untuk mengunjungi situs-situs yang menawarkan film tertentu, sehingga meningkatkan minat mereka terhadap *website* yang menawarkan hiburan digital. Tidak semua situs penyedia hiburan digital menyajikan item yang menjanjikan kepuasan pengguna. Sebuah item yang sama tidak tentu akan disukai oleh semua *user* dan terbatasnya informasi yang disediakan menjadi salah satu kendala bagi pengguna sehingga membutuhkan waktu untuk pengguna menemukan film yang sesuai. Oleh karena itu *recommender system* dibutuhkan dalam memberikan informasi berdasarkan kebutuhan pengguna. *Recommender system* akan membantu seorang *user* dalam mencari sebuah item yang berdasarkan ketertarikan masing-masing dengan memberikan prediksi beberapa item berdasarkan preferensi *user* yang berasal dari riwayat penilaian *user* terhadap item tersebut. *Recommender system* juga telah mengalami kemajuan dalam mengimplementasikan metode. *Deep learning* yang merupakan salah satu penemuan dalam metode *recommender system* dirancang untuk mengatasi beberapa kekurangan dari teknologi lain dan memberikan revolusi arsitektur rekomendasi dalam meningkatkan kinerja dalam pemberian prediksi. Penelitian ini menggunakan pendekatan prediksi *Collaborative Filtering* dengan mengimplementasikan *deep learning* berdasarkan teknologi *Neural Collaborative Filtering* pada dataset MovieLens. Evaluasi model dilakukan menggunakan metrik skor regresi *Root Mean Square Error* (RMSE). Hasil pada pengujian model menunjukkan hasil terbaik dengan nilai rata-rata *loss value* sebesar 0,1356 pada fase train dan sebesar 0,8898 pada fase val, dengan *learning rate* dan *batch size* memperoleh kinerja terbaik ketika *learning rate* bernilai 0,001 dan *batch size* dengan nilai 1024.

Kata kunci: film recommender system, deep learning, neural collaborative filtering.

FILM RECOMMENDER SYSTEM USING NEURAL COLLABORATIVE FILTERING METHOD

Abstract

At this time entertainment media has become available digitally. Entertainment especially in the form of movies is increasingly widely available. The desire to enjoy entertainment in digital media encourages other internet users to visit sites that offer certain movies, thus increasing interest in websites that offer digital entertainment. Not all digital entertainment provider sites present items that promise user satisfaction. The same item will not necessarily liked by all users and the limited information is one of the obstacles for users so that it takes time for users to find the right film. Therefore, a recommendation system is needed in providing information based on user needs. The recommendation system will help users find items based on their respective interests by providing predictions. The recommender system will help a user find an item based on their respective interests by providing predictions of several items based on user preferences derived from the user's assessment history of the item. The recommendation system has also made progress in implementing the method. Deep learning which is one of the discoveries in the recommender system method is designed to overcome some of the shortcomings of other technologies and provide a recommendation architecture revolution in improving performance in delivery. This study using a Collaborative Filtering prediction approach by implementing deep learning based on Neural Collaborative Filtering technology on the MovieLens dataset. The evaluation of the model was carried out using the Root Mean Square Error regression score metric. The results on the model test show the best results with can average loss value of 0,1356 on the train label and 0,8898 on the val label, with the learning rate and batch size getting the best performance when the learning rate is 0,001 and the batch size is 1024.

Keywords: film recommender system, deep learning, neural collaborative filtering

1. PENDAHULUAN

Perkembangan teknologi dan sistem informasi memiliki dampak terhadap kebutuhan informasi di era digital. Teknologi modern telah menyediakan berbagai bentuk kebutuhan informasi, salah satunya adalah media hiburan. Hiburan khususnya dalam bentuk film kini tersedia secara luas, film yang baru diproduksi biasanya menjadi tren di media sosial. Hal ini mendorong pengguna internet untuk mengikuti tren hiburan dengan mengunjungi *website* yang menawarkan film tertentu. Hal tersebut berdampak pada tingginya minat pengguna terhadap *website* yang menawarkan hiburan digital. Namun, tidak semua film yang menjadi tren disukai oleh semua pengguna, butuh waktu lama untuk menemukan film yang sesuai dengan minat pengguna karena banyaknya jumlah film dan terbatasnya informasi yang disediakan. Solusi dari permasalahan tersebut adalah dibutuhkan suatu sistem rekomendasi untuk memberikan informasi berdasarkan kebutuhan pengguna, atau yang lebih dikenal dengan sistem pemberi rekomendasi.

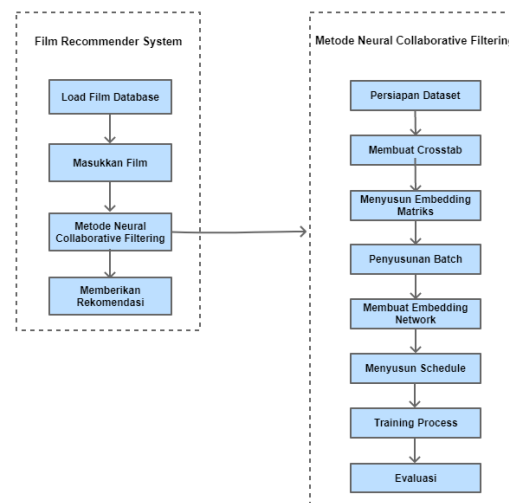
Recommender system atau sistem rekomendasi adalah sebuah teknik untuk membantu seorang *user* dalam mencari sebuah item berdasarkan ketertarikan masing-masing dalam memberikan efisiensi waktu. Sejak tahun 1990, sistem pemberi rekomendasi diterima secara luas dalam mengatasi kesulitan penyaringan data yang paling relevan dari sekumpulan informasi dalam jumlah besar dan kompleks (CC & Mohan, 2019). Kustomisasi teknologi pada sistem rekomendasi telah diadopsi secara luas oleh banyak perusahaan diberbagai bidang seperti Netflix, Amazon, Iflix, Youtube dan Spotify dalam mengembangkan bisnis mereka (Laksana, 2014).

Terdapat tiga pendekatan umum pada *recommender system* yang telah banyak digunakan dalam penelitian hingga saat ini, yaitu pendekatan *Content-based Filtering*, pendekatan *Collaborative Filtering*, dan pendekatan *Hybrid Recommendation System* (Zhang et al., 2018). *Content-based Filtering* adalah pendekatan yang menggunakan kesamaan konten item untuk memberikan rekomendasi kepada pengguna tanpa mengandalkan informasi sebelumnya (*rating*) (Zhang et al., 2018). *Collaborative Filtering* adalah pendekatan yang memberikan sebuah rekomendasi kepada pengguna berdasarkan preferensi pada profil (CC & Mohan, 2019). *Hybrid Recommender System* adalah pendekatan yang merupakan kombinasi dari dua pendekatan sebelumnya (Zhang et al., 2018).

Evolusi pengimplementasian metode *recommender system* berlanjut hingga saat ini. *Deep learning* yang merupakan salah satu penemuan dalam metode *recommender system* dapat mengatasi beberapa kekurangan dari teknologi lain dan merevolusi arsitektur rekomendasi untuk meningkatkan kinerja pemberian rekomendasi (Zhang et al., 2018). Oleh karena itu, penelitian yang

dilakukan menggunakan pendekatan prediksi *Collaborative Filtering* dengan mengimplementasikan *deep learning*, untuk menganalisis keakuratan prediksi yang dihasilkan berdasarkan teknologi *Neural Collaborative Filtering* pada dataset *MovieLens*.

2. METODE PENELITIAN



Gambar 1. Gambaran Umum Penelitian

Tahapan penelitian secara umum terbagi menjadi delapan tahapan, yaitu persiapan dataset, membuat *crosstab*, menyusun *embedding matrix*, menyusun *batch*, membentuk *embedding networks*, menyusun *schedule*, *training process*, dan evaluasi. Tahap persiapan dataset bertujuan untuk mempersiapkan dataset yang akan digunakan. Tahap membuat *crosstab*, bertujuan untuk membuat visualisasi dataset dengan mengubahnya ke dalam bentuk *crosstab* yang memperlihatkan hubungan antara *user* dan item. Tahap membentuk *embedding matrix* untuk mempersiapkan data ke dalam bentuk matrik data yang berisi hubungan vektor data berdimensi lebih rendah. Vektor data yang dihasilkan pada tahap sebelumnya kemudian dibagi menjadi beberapa bagian pada tahap menyusun *batch*, hal tersebut bertujuan untuk mengurangi ukuran data. Selanjutnya, tahap menyusun *embedding networks* untuk membangun sebuah jaringan *Neural Collaborative Filtering*. Tahap menyusun *schedule* dengan bantuan *Cyclical Learning Rate* (CLR) berfungsi untuk menerima parameter penjadwalan, bertujuan untuk membantu dalam menentukan *training epoch* berdasarkan hasil dari *learning rate*. Tahap *training* bertujuan untuk menghasilkan bobot dan performa model yang akan disimpan untuk tahap evaluasi. Tahap terakhir, yaitu evaluasi bertujuan untuk melihat rata-rata *error* terkecil perhitungan yang dihasilkan, parameter tersebut akan menunjukkan tingkat performa yang dihasilkan oleh metode yang telah digunakan.

2.1 Persiapan Dataset

Data yang digunakan untuk penelitian ini adalah data rating film dari situs MovieLens.org, yaitu sebuah laboratorium riset yang mempelajari mekanisme kajian dalam mesin rekomendasi yang dikembangkan oleh GroupLens. Dataset yang digunakan juga telah diimplementasikan untuk penelitian dalam bidang sistem rekomendasi lainnya.

Persiapan *dataset* dimulai dengan melakukan *input dataset rating* film dalam ekstensi atau format csv. ke dalam bentuk *dataframe*. Setelah dilakukan *input dataset* kemudian dilakukan konversi file .csv menggunakan fungsi *library pandas* untuk mengunggah data yang disimpan dalam variabel *df* dengan indeks dimulai dari 0. Struktur konversi file yang digunakan adalah *userId*, *movieId*, *rating* dan *timestamp*. Dataset yang digunakan memiliki atribut atau variabel yang meliputi konten item, *rating* item, *user* dan metadata sesuai dengan Tabel 1.

Tabel 1. Detail *Dataset MovieLens 100K*

No.	Nama Variabel	Jumlah Data
1.	ID Film	9066
2.	ID User	671
3.	Rating Film	100.004
4.	Timestamp	–

Data *MovieLens 100K* memiliki rincian data yang terdiri 671 pengguna, 9066 item (film), 100.004 jumlah *rating*, dan informasi konten item dalam bentuk *genre*. Kumpulan data berisi nilai dalam bentuk peringkat 1, 2, 3, 4, dan 5 dari 100.0004 *rating*. Beberapa contoh isi dari *dataset MovieLens 100K* dapat dilihat pada Tabel 2.

Tabel 2. Contoh Data *Rating MovieLens*

	UserId	MovieId	Rating	Timestamp
0.	1	31	2,5	1260759144
1.	1	1029	3,0	1260759179
2.	1	1061	3,0	1260759182
3.	1	1129	2,0	1260759185
4.	1	1172	4,0	1260759205
...
99999.	671	6268	2,5	1065579370
100000.	671	6269	4,0	1065149201
100001.	671	6365	4,0	1070940363
100002.	671	6385	2,5	1070979663
100003.	671	6565	3,5	1074784724

2.2 Membuat Crosstab

Tahap membuat *crosstab* bertujuan untuk memvisualisasikan *user* dan item dalam mendapatkan *rating* tertinggi *user* dan item, di mana baris mengidentifikasi sejumlah *user* yang memberikan jumlah *rating* untuk *movies* tertinggi atau disebut dengan *Top User* dan kolom mengidentifikasi sejumlah *movies* yang mendapatkan jumlah *rating* tertinggi atau disebut sebagai *Top Movies*. Pembuatan *crosstab*

mendefinisikan matrik interaksi item *user* dari umpan balik *user* (He et al., 2017).

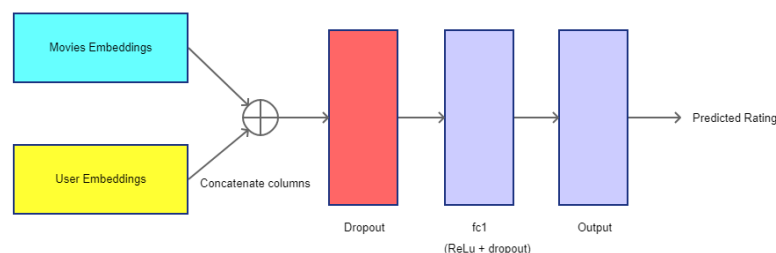
Proses yang dilakukan dalam membuat *crosstab* akan menampilkan 15 *rating* tertinggi dengan mengelompokkan sejumlah *userId* berdasarkan *rating*, kemudian dilakukan penyortiran dengan mengelompokkan *userId* berdasarkan *rating* menjadi *Top User*. Kemudian dilakukan pengelompokan sejumlah *movieId* berdasarkan *rating*, lalu dilakukan penyortiran dengan mengelompokkan *movieId* berdasarkan *rating* menjadi *Top Movie*. Setelah didapatkan *Top User* dan *Top Movie* selanjutnya dilakukan penyaringan informasi *rating* dengan menghitung top data tertinggi dengan menghubungkan *Top User* dan *Top Movie* berdasarkan masing-masing *userId* dan *movieId*, penyatuan tersebut divisualisasikan menjadi bentuk visualisasi *crosstab*.

2.3 Menyusun Embedding Matrix

Tahapan yang dilakukan pada penyusunan *embedding matrix* bertujuan untuk mengubah bentuk data yang awalnya berupa *crosstab* menjadi bentuk matrik. *Embedding matrix* merepresentasikan bentuk data menjadi model yang mudah dipahami dalam melihat hubungan vektor data dan merupakan representasi dimensi yang lebih rendah.

Tahap awal yang dilakukan adalah mengecek apakah *rating* pada *userId* yang masuk dalam data *Top Rating* tidak kosong. Jika data *Top Rating* tidak kosong, maka dilakukan pengelompokkan jumlah *userId* dan *rating*. Id *user* yang masuk dalam kategori pengelompokan tersebut bersifat unik. Data *userId* yang tidak kosong tersebut akan menjadi sebuah *Unique User*, kemudian *unique user* tersebut akan dijadikan bentuk data *New User* berdasarkan *rating* dan *userId*. Hal yang sama juga dilakukan dengan data *movieId* yang menjadi sebuah *Unique Movies* kemudian dijadikan bentuk data *New Movies* berdasarkan *rating* dan *movieId*.

User dan *Movie* yang bersifat unik akan diinisialisasikan menjadi *New User* dan *New Movies* dan memiliki *index* yang berawal dari 0. Melalui bantuan *pandas*, *dataframe* akan dibentuk *New User* dan *New Movies* yang membaca data berdasarkan *user_id* dan *movie_id* dan diinisialisasikan menjadi X. Sedangkan keseluruhan *rating* diinisialisasikan menjadi y. *Output* yang nantinya akan dikeluarkan adalah *user* dan *movie* dalam bentuk *embedding matrix*. Bentuk *dataset X*, yaitu sejumlah *rating* dengan *user* dan *movie* dengan bentuk matrik, dan target *rating* dengan bentuk dataset y. Hasil yang diberikan adalah matrik *dataset X* yang ditunjukkan sesuai dengan Tabel 3 dan matrik *rating* yang ditunjukkan sesuai dengan Tabel 4.

Gambar 2. Skema *Embedding Networks*Tabel 3. Contoh Matrik *Dataset X*

Index	user_id	movie_id
0	0	0
1	0	1
2	0	2
3	0	3
4	0	4
...
99999	670	7005
100000	670	4771
100001	670	1329
100002	670	1331
100003	670	2946

Tabel 4 Contoh Matrik Target *y*

Index	Rating
0	2,5
1	3,0
2	3,0
3	2,0
4	4,0
...	...
99999	2,5
100000	4,0
100001	4,0
100002	2,5
100003	3,5

2.4 Penyusunan *Batch*

Tahap penyusunan *batch* bertujuan membagi *dataset* menjadi bagian kecil dengan melakukan *split dataset* (He et al., 2017). Pembagian *dataset* nantinya akan digunakan selama proses *training*. Proses dalam penyusunan *batch* dilakukan dengan membuat parameter untuk merepresentasikan objek yang akan dibuat, yaitu *X* atau *dataset* dalam bentuk *embedding* matrik, *y* atau target *rating*, jumlah data yang akan diambil pada setiap *batch* (*batch_size*) dengan ukuran 32, mengambil sejumlah data secara acak, dan mengubah bentuk *input X* dan *y* yang semula matrik menjadi sebuah *array*. Matrik *X* akan diacak sepanjang indeks pertamanya, *index X* atau *dataset* dalam bentuk *embedding* matrik adalah 0. Lalu dilakukan inisialisasi *self* terhadap *X*, *y*, *batch_size*, *shuffle*, dan jumlah ukuran *X* terhadap *batch*, langkah tersebut dilakukan dengan tujuan untuk membuat kelas yang menunjukkan iterasi pada *dataset* satu per satu, kemudian dilakukan pengulangan.

Nilai yang dihasilkan *batch* pada *X* adalah *torch.LongTensor* dan nilai yang dihasilkan *batch* pada *y* adalah *torch.FloatTensor*. Sedangkan kedua

batch yang akan dibentuk berdasarkan *input X*, *y*, dan *self.batch_size* yang bernilai 4. *Batch* yang dihasilkan berupa tensor *x_batch* dan *y_batch*.

2.5 Membuat *Embedding Network*

Tahap membuat *Embedding Network* bertujuan untuk membangun jaringan model *Neural Collaborative Filtering* (He et al., 2017). Proses awal dalam membuat *embedding networks*, yaitu membentuk *embedding layer* dengan melakukan inisialisasi fungsi yang dibutuhkan dalam pembuatan *embedding networks* melalui bantuan modul. Terlebih dahulu perlu dibentuk *dense network* dengan *embedding layer*, dimana *embedding layer* berfungsi sebagai pembawa informasi dari satu *layer* ke *layer* lainnya. Perlu digunakan *generator* untuk menghasilkan rangkaian *hidden layer* yang terletak di antara *input* dan *output* untuk mengaktifkan *dropout*. Fungsi *generator* ini menangkap nilai *hidden* dan *dropout* berdasarkan nilai pola tersembunyi. Pembentukan *hidden layer* menggunakan fungsi *generator* dengan menghasilkan persamaan dan *dropout layer* yang bergantung pada nilai dari hasil *hidden layer* menjadi *single* modul.

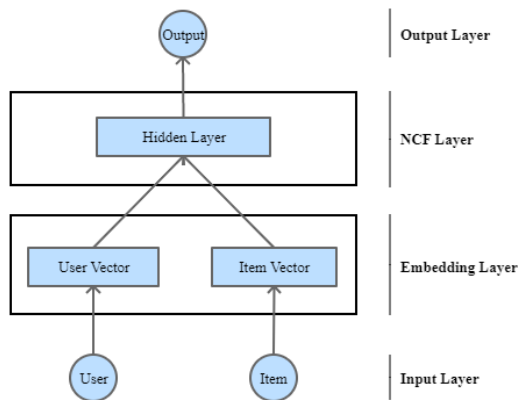
Proses selanjutnya adalah menempatkan *embeddings matrix* dan selanjutnya mengubah ID *integer* menjadi *array*, menyisipkan sekelompok *fully-connected layer* yang terdiri dari *hidden layer*, fungsi aktivasi, *output layer*, dan *loss function* dengan menggunakan MSE disertai *dropout* yang menghasilkan daftar *rating* yang diprediksi.

Pembuatan *neural networks* dapat membantu mengambil keputusan dalam melihat kemiripan antar *user* dan memprediksi *rating* tersebut dalam kumpulan data yang tersedia. Ilustrasi skema model *neural networks* yang terbentuk ditunjukkan sesuai pada Gambar 2.

Ilustrasi skema *embedding networks* menunjukkan setiap transaksi akan diekstraksi menjadi dua matrik atau bagian, yaitu vektor *user* dan vektor item. Vektor *user* dan vektor item kemudian direpresentasikan ke dalam *generalized matrix factorization* (GMF). Dua vektor *user* dan item digabungkan menjadi *multi layer perceptron* (MLP), lalu dibuat *embedding layer* dengan setiap *layernya* yang menggunakan *dropout*, dimana setiap *embedding* menggunakan fungsi aktivasi ReLU.

Selanjutnya, menggabungkan *generalized matrix factorization* dengan *output* dari *neural networks layer* yang menghasilkan model *training* dengan regresi linier dan *loss function* yang digunakan adalah *mean square error* (MSE).

Proses selanjutnya adalah menerapkan *embedding matrix* yang sudah dikonversikan menjadi bentuk *array* kemudian didapatkan *layer* yang terhubung secara lengkap dengan *dropout*, dan didapatkan daftar prediksi *rating*. Aktifitas transfer membawa *input* yang diaktifasi dengan *Rectifier Linear Unit* (ReLU). Terdapat contoh model arsitektur *network* yang ditunjukkan pada Gambar 3.



Gambar 3. Contoh Arsitektur *Neural Networks*

Arsitektur *Neural Network* yang terbentuk berasal dari *input* sejumlah *user* dan *item* (movie). *Layer* pertama yang terbentuk adalah *input layer* yang berisi masukan *user* sejumlah 671 dan *item* (movie) sejumlah 9066. *Layer* kedua adalah *embedding layer* yang berisi *embedding user* yang berdimensi 671×150 dan *embedding movie* yang berdimensi 9066×150 . Vektor *user* dan vektor *item* yang dihasilkan pada *layer* kedua akan dikonkatenasi menjadi *embedding matriks* dan akan menjadi *input* pada *layer* ketiga. *Layer* ketiga adalah *neural collaborative filtering* yang berisi *single hidden layer* dengan 300 *input* sampel, 100 *output* sampel, menggunakan *activation function* ReLU, dan *dropout* yang berisi parameter dengan nilai 0,5. *Layer* keempat adalah *output* yang berisi prediksi *rating*.

2.6 Menyusun Schedule

Tahap menyusun *schedule* bertujuan untuk mencari optima lokal atau optima global yang terbaik dengan bantuan *Cyclical Learning Rate* (CLR). *Cyclical Learning Rate* merupakan salah satu fitur pada *library fast.ai* dan menggunakan *learning rate cosine annealing* dengan teknik *restart* untuk membandingkan konfigurasi yang tersedia.

Proses menyusun *schedule* mengimplementasikan CLR dengan menggunakan *method* yang menghasilkan daftar *learning rate* berdasarkan *training epoch*. Semua *layer* akan

memiliki *learning rate* yang sama dan menghasilkan *single value*. Proses awal yang dilakukan dalam menyusun *schedule*, yaitu melakukan inisialisasi terhadap parameter CLR yang menunjukkan fungsi yang dapat dipanggil kembali. Fungsi tersebut akan menerima *training epoch* dan *learning rate* untuk mengembalikan tingkat *learning rate* baru. Kemudian proses selanjutnya adalah membuat fungsi yang menerima parameter dalam penjadwalan dan menghasilkan fungsi baru. Proses ini untuk mengurangi iterasi *learning rate* setelah beberapa *epoch* sehingga *optimizer* yang bekerja akan mencari optima lokal, global atau terbaik dan memperbarui bobot serta bias yang dihasilkan agar *error* berkurang.

2.7 Training Procces

Tahap *training procces* bertujuan untuk melihat kemiripan pola di antara *rating user* untuk mengisi *gap* dalam membuat sebuah prediksi pada data *rating* yang hilang. Persentase data yang digunakan sebagai set *training* dan *testing*, yaitu 80% data digunakan sebagai set *training* dan 20% data digunakan sebagai set *testing*. Pada perintah *train test split* diisi dengan parameter *array X* dan *y* yang menunjukkan *array* yang dipakai adalah *X* dan *y* dengan nilai *test_size* sebesar 0,2. Penelitian ini menggunakan 100.004 data sehingga 80.003 data digunakan sebagai data *training* dan 20.001 data digunakan sebagai data *testing*. Pada tahapan ini terdapat beberapa fase yang akan dilewati, yaitu fase *train* dan fase *val*.

Proses awal yang dilakukan pada tahap *training*, yaitu membagi *dataset* menggunakan fungsi *train test split* menjadi data *testing* dan data *training*. *Embedding network* yang telah dibuat pada proses sebelumnya diberi beberapa *input*, yaitu jumlah *user* dan *movies*, jumlah *factors* sebesar 150, tiga *hidden layer* dengan node 500, dan *embedding dropouts* sebesar 0,05.

Proses berikutnya, yaitu mengatur *training loop* menggunakan *Cyclical Learning Rate* (CLR) untuk menyimpan bobot terbaik model yang dihasilkan. Selanjutnya menerapkan beberapa parameter dengan menggunakan *optimizer* Adam, menerapkan parameter pada *learning rate* sebesar 0,001 dan 0,01 menggunakan *MSE loss* sebagai matrik untuk mengukur kualitas *networks* atau *loss function*. Kemudian menggunakan *learning rate cosine annealing* dengan teknik *restart* untuk membandingkan konfigurasi yang tersedia, yaitu ketika terdapat penurunan kecepatan pada setiap *batch* selama 2 *epoch*, maka kecepatan akan kembali ke nilai aslinya.

Proses berikutnya menjalankan dua fase pada tahap *training*, yaitu fase *train* dan *val* dalam *training loop*. Pertama disebut fase *train*, selama fase *train* bobot *network* akan diperbarui dan mengubah *learning rate*. Fase kedua disebut *val*, pada fase ini digunakan untuk menguji kinerja model. Saat bobot

pada nilai *loss* berkurang, maka parameter model akan disimpkan.

Setiap *batch* pada proses *training* data akan diacak secara *random* menggunakan hasil tensor *x_batch* dan *y_batch*. Saat sistem akan menghitung gradien, maka gradien akan di-*reset* kembali dengan tujuan meng-*update* model. Proses perhitungan gradien diaktifkan pada fase *train* yang menjadi mekanisme untuk mengaktifkan komputasi gradien, tujuannya adalah untuk menghitung *loss* yang dilewati oleh *output* pada *network*.

Pada fase *val* bobot dan *rate* tidak akan diperbarui. Di saat proses *training* gradien akan dihitung secara otomatis dan kemudian akan dilakukan perhitungan pada gradien. Fase *val* akan dilakukan sebuah operasi *back propagation* dengan memanggil fungsi *backward()* pada *loss* yang dihitung berdasarkan *input* dan *output*. Bobot *loss* selama proses berjalan akan dihasilkan dari jumlah nilai *loss* selama proses dan *loss.item*. Saat menghitung *loss batch* maka akan menghitung rata-rata *loss* dari *epoch* dengan cara memplot *loss* untuk setiap *epoch* dan menambahkan hasilnya ke *loss value* di setiap *epoch*. Rata-rata *loss* pada *batch* akan memberi perkiraan *epoch loss* yang berisi hilangnya seluruh *mini-batch* selama *training*.

Loss value didapatkan dengan menjumlahkannya dan menghitung rata-rata setelah *epoch* selesai. *Training loss* ini akan digunakan untuk melihat seberapa baik kinerja model yang dihasilkan pada *training dataset*. Setiap *epoch training* akan mengevaluasi performa model yang dilatih. Pada fase *val* tidak diperlukan parameter untuk diaktifkan dan parameter gradien tidak akan diperbarui bagian di tensor mana pun. Karena di fase *val*, *optimizer* akan bekerja untuk mengetahui parameter model mana saja yang akan di *update*. Fase ini juga akan menyimpan bobot model terbaik. Jika *epoch loss* bernilai lebih kecil dari *best_loss* maka *output* kenaikan *loss* akan dikeluarkan sesuai dengan *epoch* terakhir. Proses *training* berakhir setelah tidak ada peningkatan setelah 10 *training epoch* berturut-turut dan proses *looping* akan berhenti. Hasilnya bobot terbaik akan disimpan selama *training*, dan menggunakan data *testing* untuk validasi data dalam melihat performa model akhir.

2.8. Evaluasi

Tahap evaluasi bertujuan untuk mengetahui seberapa besar akurasi yang dihasilkan oleh metode yang digunakan dalam penelitian dan sebagai landasan terhadap kelayakan hasil rekomendasi. Penelitian ini menggunakan RMSE sebagai evaluasi terhadap hasil prediksi dataset yang menerapkan metode *Neural Collaborative Filtering*. Setelah didapatkan hasil model performa yang berasal dari tahap *training* kemudian dilakukan estimasi *error* prediksi *rating* menggunakan rumus RMSE dengan menggunakan data *testing*. Metrik skor regresi *Root*

Mean Square Error (RMSE) akan menghitung rata-rata dari kuadrat perbedaan yang berasal dari hasil *rating* aktual dan prediksi. Semakin besar tingkat kesalahan yang dihasilkan akan berdampak pada nilai RMSE yang semakin besar, sehingga menunjukkan tingkat kesalahan yang tinggi. Rumus untuk setiap metrik skor seperti ditampilkan dalam Persamaan 1:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - x_i)^2}{n}} \quad (1)$$

Keterangan:

n : Jumlah item yang dirating pengguna

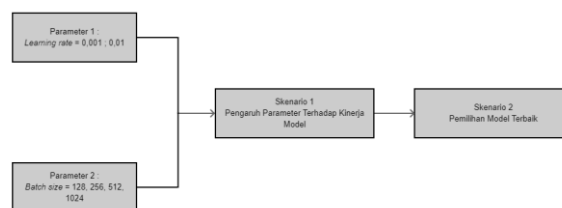
y : Rating prediksi

x : Rating sebenarnya

3. HASIL DAN PEMBAHASAN

3.1. Skenario Pengujian

Skenario pengujian membahas data pengujian dan skenario pengujian terhadap tingkat akurasi menggunakan *Neural Collaborative Filtering* (NCF) pada dataset *MovieLens* 100K. Gambaran umum skenario pengujian ditunjukkan pada Gambar 4.



Gambar 4. Skenario Pengujian

Nilai parameter yang diujikan dengan nilai *learning rate* adalah 0,001 dan 0,01, sedangkan nilai *batch size* adalah 128, 256, 512 dan 1024. Kedua parameter akan dikombinasikan untuk mendapatkan hasil terbaik. Kinerja dari setiap kombinasi dilakukan pada tahap *training* untuk mendapatkan nilai akurasi. Hasil perhitungan akurasi akan dibandingkan untuk mendapatkan hasil terbaik.

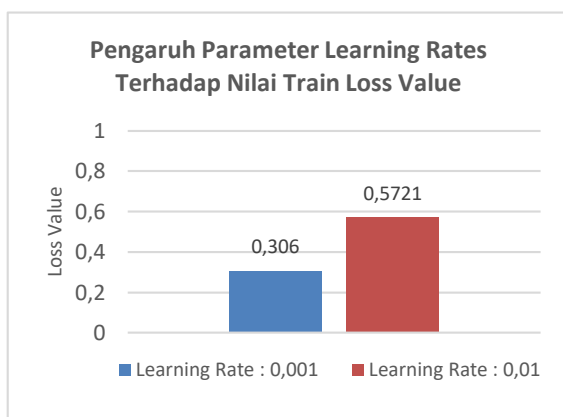
3.2. Hasil dan Analisa Skenario 1

Skenario 1 melakukan analisa berdasarkan kombinasi parameter *learning rate* dan *batch size*. Pengaruh dari kombinasi ini akan dihitung untuk mendapatkan bobot nilai. Pada Tabel 6 menunjukkan kombinasi parameter yang digunakan. Pada Gambar 5 menunjukkan hasil *loss value* pada parameter *learning rate*, sedangkan pada Gambar 6 menunjukkan hasil *loss value* pada parameter *batch sizes*.

Tabel 6. Hasil *Loss Value* Kombinasi Parameter yang Digunakan

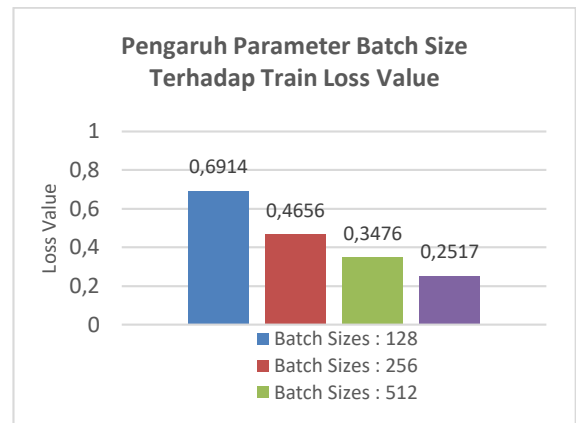
No	Kombinasi	Parameter		Fase	
		Learning Rate	Batch Size	Train	Val
1.	Kombinasi-1	0,001	128	0,5639	0,8425

No	Kombinasi	Parameter		Fase	
		Learning Rate	Batch Size	Train	Val
2.	Kombinasi-2	0,001	256	0,2902	0,9205
3.	Kombinasi-3	0,001	512	0,2116	0,9342
4.	Kombinasi-4	0,001	1024	0,1582	0,9019
5.	Kombinasi-5	0,01	128	0,8189	0,9035
6.	Kombinasi-6	0,01	256	0,6409	0,9138
7.	Kombinasi-7	0,01	512	0,4835	0,9600
8.	Kombinasi-8	0,01	1024	0,3452	0,9660

Gambar 5. Pengaruh *Learning Rate* Terhadap *Loss Value*

Parameter *learning rate* menunjukkan bagaimana perubahan bobot. Jika *learning rate* rendah akan berakibat pada proses *training* yang lebih lama dalam pencarian nilai *loss function*. Sedangkan nilai *learning rate* yang tinggi berakibat pada perubahan bobot yang terlalu besar. Pengaruh parameter *learning rate* terhadap bobot *loss value* dapat terlihat pada Gambar 5 dari grafik tersebut terlihat bahwa *learning rate* dengan nilai 0,001 menghasilkan rata-rata *loss value* lebih rendah daripada *learning rate* dengan nilai 0,01 yang memiliki *loss* lebih besar. Nilai *learning rate* 0,001 melakukan perubahan terhadap variabel secara kecil, hal tersebut lebih baik dibandingkan dengan nilai *learning rate* 0,01 sehingga memberi peluang pada model untuk mempelajari kumpulan bobot menjadi lebih optimal.

Batch size yang bernilai kecil akan berdampak pada waktu komputasi yang lebih lama. Sedangkan menggunakan *batch size* yang lebih besar akan mengurangi kemampuan model dalam memberikan hasil yang kurang sesuai. Pengaruh parameter *batch size* terhadap bobot *loss value* dapat terlihat pada Gambar 6., grafik tersebut terlihat *batch size* dengan nilai 1024 menghasilkan rata-rata nilai *loss value* yang lebih baik dari pada *batch size* dengan nilai 128, 256, dan 512.

Gambar 6. Pengaruh Parameter *Batch Size* Terhadap *Loss Value*

Semakin besar nilai *batch size* berdasarkan gambar maka nilai *loss* yang didapatkan semakin kecil. Walaupun nilai *batch size* tersebut lebih besar dari yang lainnya, dapat ditunjukkan hasil pada *batch size* 1024 yang paling optimal karena nilai *loss* yang paling mendekati 0, hal tersebut memungkinkan percepatan komputasi untuk mempelajari model dalam mencari nilai *loss value* yang lebih rendah karena menunjukkan jumlah sampel data yang disebarkan ke *neural networks* lebih besar daripada yang lainnya dan memberikan.

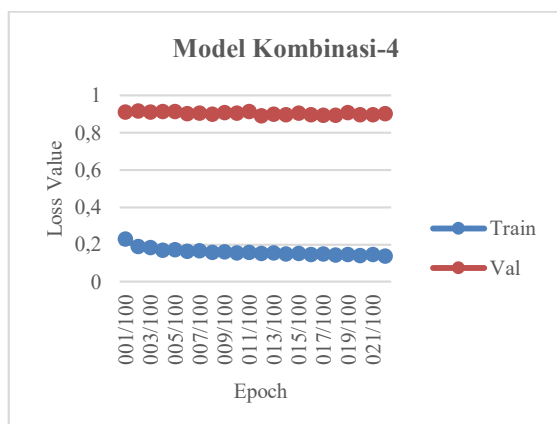
3.3. Hasil dan Analisa Skenario 2

Skenario 2 melakukan pemilihan model terbaik dari model yang telah diperoleh dari skenario 1 pada setiap pasang label. Parameter *learning rate* dengan nilai 0,001 memperoleh rata-rata *loss value* terendah. Parameter *batch size* dengan nilai 1024 memperoleh rata-rata *loss value* terendah. Dari kedua parameter tersebut dapat disimpulkan bahwa kombinasi parameter terbaik terdapat pada Kombinasi-4 dengan *learning rate* bernilai 0,001 dan *batch size* bernilai 1024. Tabel 7 menunjukkan *loss value* pada model Kombinasi-4 pada setiap model.

Tabel 7. *Loss Value* Model Pada Kombinasi Kombinasi-4

No	Epoch	Train	Val
1.	001/100	0,2271	0,9101
2.	002/100	0,1861	0,9144
3.	003/100	0,1824	0,9082
4.	004/100	0,1678	0,9126
5.	005/100	0,1701	0,9114
6.	006/100	0,1612	0,9002
7.	007/100	0,1637	0,9025
8.	008/100	0,1551	0,8989
9.	009/100	0,1583	0,9069
10.	010/100	0,1531	0,9046
11.	011/100	0,1571	0,9110
12.	012/100	0,1494	0,8898
13.	013/100	0,1535	0,8981
14.	014/100	0,1465	0,8954
15.	015/100	0,1505	0,9037
16.	016/100	0,1440	0,8942
17.	017/100	0,1476	0,8929
18.	018/100	0,1419	0,8927
19.	019/100	0,1453	0,9062
20.	020/100	0,1393	0,8942

No	Epoch	Train	Val
21.	021/100	0,1437	0,8948
22.	022/100	0,1356	0,8996



Gambar 7. Grafik Model Kombinasi-4

Berdasarkan Tabel 7, hasil bobot *loss value* pada *train* menunjukkan penurunan *loss* dan memperoleh hasil terbaik pada *epoch* ke 022/100 dengan *train loss* sebesar 0,1356 dan *val loss* sebesar 0,8896. Melalui penelitian yang telah dilakukan parameter yang memberikan hasil paling optimal adalah *learning rate* dengan nilai 0,001 dan *batch size* dengan nilai 1024. Berdasarkan hasil yang ditunjukkan berhubungan dengan nilai *learning rate* yang kecil sehingga perubahan bobot secara kecil memberikan kesempatan pada model untuk mempelajari kumpulan bobot agar menjadi lebih optimal. dan nilai *batch size* yang digunakan menyebarkan jumlah sampel data ke *neural network* yang lebih besar sehingga nilai *loss function* lebih rendah. Kemudian di dapatkan hasil RMSE dengan nilai 0,91467.

4. KESIMPULAN

Metode *Neural Collaborative Filtering* dapat diterapkan pada *Film Recommender System* melalui kombinasi pendekatan *Collaborative Filtering* dan metode *deep learning*, yaitu *neural networks*. Parameter *learning rate* dan *batch size* mempengaruhi kinerja model memperoleh kinerja terbaik ketika *learning rate* bernilai 0,001 dan *batch size* dengan nilai 1024 dengan hasil *train loss value* sebesar 0,1356 dan *val loss value* sebesar 0,8898. Model prediksi sistem rekomendasi menunjukkan hasil nilai RMSE sebesar 0,91467.

DAFTAR PUSTAKA

- AL AMIN, A. 2021. Mereduksi Error Prediksi Pada Sistem Rekomendasi Menggunakan Pendekatan Collaborative Filtering Berbasis Model Matrix Factorization. *EXPLORE*, 11(2), 8-14. <https://doi.org/10.35200/EXPLORE.V11I2.434>.
- BOBADILLA, J., ALONSO, S., & HERNANDO, A. 2020. Deep learning architecture for collaborative filtering recommender systems. *Applied Sciences*, 10(7), 2441. <https://doi.org/10.3390/app10072441>.
- CHAVARE, S. R., AWATI, C. J., & SHIRGAVE, S. K. 2021. Smart recommender system using deep learning. In 2021 6th International Conference on Inventive Computation Technologies. (ICICT) (pp. 590-594). IEEE. <https://doi.org/10.1109/ICICT50816.2021.9358580>.
- DWICAHYA, IMAM. 2018. Perbandingan Sistem Rekomendasi Film Metode User-based dan Item-based Collaborative Filtering. Universitas Sanata Dharma.
- CAKRANINGRAT, R. 2011. Sistem pendukung Keputusan untuk UMKM. [ebook]. UBX Press. Tersedia melalui: Perpustakaan Universitas BX <<http://perpustakaan.ubx.ac.id>> [Diakses 1 Juli 2021]
- GARANAYAK, M. dkk. 2019. Recommender system using item based collaborative filtering (CF) and K-means. *International Journal of Knowledge-based and Intelligent Engineering Systems*, 23(2), 93-101. <https://doi.org/10.3233/KES-190402>.
- GIRSANG, A. S., & WIBOWO, A. 2021. Neural collaborative for music recommendation system. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1071, No. 1, p. 012021). IOP Publishing. <https://iopscience.iop.org/article/10.1088/1757-899X/1071/1/012021>
- HE, X. dkk. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web* (pp. 173-182). <https://doi.org/10.1145/3038912.3052569>.
- LAKSANA, E. A. 2014. Collaborative Filtering dan Aplikasinya. *Jurnal Ilmiah Teknologi Infomasi Terapan*, 1(1). <https://doi.org/10.33197/jitter.vol1.iss1.2014.44>
- LIU, Y. dkk. 2018. A novel deep hybrid recommender system based on auto-encoder with neural collaborative filtering. *Big Data Mining and Analytics*, 1(3), 211-221. <https://doi.org/10.26599/BDMA.2018.9020019>
- MASRURI, F., & MAHMUDY, W. F. 2007. Personalisasi web e-commerce menggunakan recommender system dengan metode item-based collaborative filtering. *Jurnal Ilmiah Kursor*, 3(1).
- Grouplens.org. 1998. MovieLens 100K dataset. Tersedia di:

- <<https://grouplens.org/datasets/movielens/100k/>> [Diakses 3 September 2021].
- CC, N., & MOHAN, A. 2019. A social recommender system using deep architecture and network embedding. *Applied Intelligence*, 49(5), 1937-1953. <https://doi.org/10.1007/s10489-018-1359-z>
- RIZKY, M. I., ASROR, I., & MURTI, Y. R. 2020. Sistem Rekomendasi Program Studi Untuk Siswa Sma Sederajat Menggunakan Metode Hybrid Recommendation Dengan Content Based Filtering Dan Collaborative Filtering. *eProceedings of Engineering*, 7(1).
- ROCHMAWATI, N. dkk. 2021. Analisa Learning Rate dan Batch Size pada Klasifikasi Covid Menggunakan Deep Learning dengan Optimizer Adam. *JIEET (Journal of Information Engineering and Educational Technology)*, 5(2), 44-48. <https://doi.org/10.26740/jieet.v5n2.p44-48>
- PUTRA, A. I., & SANTIKA, R. R. 2020. Implementasi Machine Learning dalam Penentuan Rekomendasi Musik dengan Metode Content-Based Filtering. *Edumatic: Jurnal Pendidikan Informatika*, 4(1), 121-130. <https://doi.org/10.29408/edumatic.v4i1.2162>
- PRATAMA, Y. A. dkk. 2013. Digital Cakery Dengan Algoritma Collaborative Filtering, 14(1), 79-88.
- SAHOO, A. K. dkk. 2019. DeepReco: deep learning based health recommender system using collaborative filtering. *Computation*, 7(2), 25. <https://doi.org/10.3390/computation7020025>
- SARWAR, B. dkk. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web* (pp. 285-295). <https://doi.org/10.1145/371920.372071>
- SHAKIROVA, E. 2017. Collaborative filtering for music recommender system. In *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)* (pp. 548-550). IEEE. <https://doi.org/10.1109/EIConRus.2017.7910613>
- SMITH, L. N. 2017. Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)* (pp. 464-472). IEEE. <https://doi.org/10.1109/WACV.2017.58>
- YOSHUA, I., & BUNYAMIN, H. 2021. Pengimplementasian Sistem Rekomendasi Musik Dengan Metode Collaborative Filtering. *Jurnal STRATEGI-Jurnal Maranatha*, 3(1), 1-16.
- ZARZOUR, H., AL-SHARIF, Z. A., & JARARWEH, Y. 2019. RecDNNing: a recommender system using deep neural network with user and item embeddings. In *2019 10th International Conference on Information and Communication Systems (ICICS)* (pp. 99-103). IEEE. <https://doi.org/10.1109/IACS.2019.8809156>
- ZHANG, L. dkk. 2018. A recommendation model based on deep neural network. *IEEE Access*, 6, 9454-9463. <https://doi.org/10.1109/ACCESS.2018.2789866>
- ZHANG, S. dkk. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1), 1-38. <https://doi.org/10.1145/3285029>.

Halaman ini sengaja dikosongkan

SISTEM REKOMENDASI PAKET WISATA SE-MALANG RAYA MENGUNAKAN METODE HYBRID CONTENT BASED DAN COLLABORATIVE

Bambang Tri Wahyu U, Angga Widya Anggriawan
STMIK ASIA Malang

ABSTRAK

Sistem rekomendasi merupakan suatu aplikasi untuk menyediakan dan merekomendasikan suatu item dalam membuat suatu keputusan yang diinginkan oleh pengguna. Metode hybrid content dan collaborative based nearest neighbor dibuat untuk mengatasi kekurangan yang terdapat pada metode content dan collaborative based. Kota Malang merupakan objek yang akan digunakan dalam membangun sistem rekomendasi yang akan dibuat yaitu sistem rekomendasi paket wisata. Dengan menggunakan data history transaksi yang diproses dengan metode Content-based dan metode Collaborative, selanjutnya dilakukan proses hybrid dan menghasilkan hasil rekomendasi paket wisata yang sesuai bagi wisatawan.

Kata kunci: *Sistem Rekomendasi, Metode Hybrid, Content-based Filtering, Collaborative-based Filtering, nearest neighbor.*

ABSTRACT

Recommendation system is an application to provide and recommend an item in making a decision that is desired by the user. Hybrid method of content and collaborative based nearest neighbor made to overcome deficiencies in content and collaborative based methods. Malang is the object that will be used in building a system of recommendations that will be made of recommendation system package tours. Using transaction history data that are processed by the method of Content-based and Collaborative methods, then carried out the process of hybrid and produce recommendations for the appropriate package tours for tourists.

Keywords: *Recommendation System, Hybrid Recommendation, Collaborative-Content Based Filtering, nearest neighbor.*

PENDAHULUAN

Sistem rekomendasi telah banyak digunakan oleh hampir sebagian besar area bisnis dimana konsumen perlu membuat suatu keputusan atau rekomendasi pilihan dari informasi yang disediakan. Area pariwisata merupakan salah satu contoh bisnis area yang menerapkan sistem rekomendasi untuk membantu para wisatawan dalam membuat keputusan bagi perjalanan mereka. Internet dan world wide web menyediakan banyak informasi dibidang pariwisata karena pariwisata memiliki pengalaman menarik bagi para wisatawan namun sangat sulit untuk menemukan informasi paket wisata yang sesuai dengan keinginan para penggunanya. Maka dibuatlah suatu sistem rekomendasi bagi industri pariwisata atau perjalanan untuk menawarkan serta merekomendasikan paket tempat-tempat wisata di Malang dan sekitarnya kepada para wisatawan yang sesuai dengan keinginan mereka.

Sistem rekomendasi didefinisikan sebagai aplikasi pada website e-commerce untuk

mengusulkan informasi dan menyediakan fasilitas yang diinginkan pengguna dalam membuat suatu keputusan (Ricci, F., 2002). Sistem ini diasumsikan seperti penggambaran kebutuhan dan keinginan pengguna melalui pendekatan metode rekomendasi untuk mencari suatu item dengan menggunakan rating berdasarkan kemiripan dari karakteristik informasi pengguna.

Terdapat beberapa metode yang digunakan untuk mendukung cara kerja sistem rekomendasi dalam menghasilkan sebuah informasi diantaranya seperti demographic recommendation, collaborative recommendation, content-based recommendation dan knowledge based recommendation.

Metode collaborative dan content-based adalah metode yang sering digunakan pada sistem rekomendasi karena teknik ini menyaring informasi berdasarkan keinginan pengguna dan berdasarkan content yang disediakan (Sebastia, L et al., 2009). Namun terdapat beberapa kekurangan dalam kedua metode tersebut. Maka untuk menutupi kelemahan pada metode-metode

tersebut dapat dilakukan dengan menggunakan metode hybrid yaitu menggabungkan beberapa metode yang terdapat pada sistem rekomendasi untuk menghasilkan item rekomendasi yang sesuai dengan keinginan pengguna. Untuk mendukung cara kerja metode tersebut maka diperlukan algoritma classification yang akan mendukung kemampuan sistem rekomendasi dalam memberikan informasi yang sesuai. Salah satunya adalah algoritma Nearest Neighbor (NN).

Algoritma Nearest Neighbor (NN) merupakan algoritma pendekatan untuk mencari kasus dengan menghitung kedekatan antara kasus baru dengan kasus lama yaitu berdasarkan pencocokan bobot dari sejumlah atribut yang ada. Nearest Neighbor akan mengklasifikasikan hanya jika atribut dari kasus baru sesuai dengan salah satu atribut pada kasus lama (Ricci, F et al., 2010).

Oleh karena itu pada penulisan tugas akhir ini akan diteliti menggunakan metode hybrid (content-based dan collaborative) dengan objek paket wisata pada sistem rekomendasi menggunakan algoritma Nearest Neighbor (NN).

LANDASAN TEORI

1. Sistem Rekomendasi

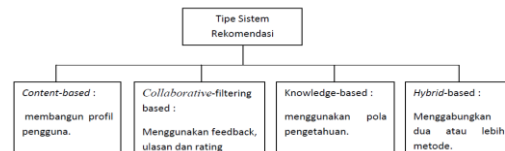
Konsep sistem rekomendasi telah digunakan secara luas oleh hampir semua area bisnis dimana seorang konsumen memerlukan informasi untuk membuat suatu keputusan. Sistem rekomendasi paket wisata atau perjalanan menggunakan konsep ini dapat menolong para wisatawan untuk memutuskan tujuan perjalanan mereka, akomodasi dan aktivitas di tempat tujuan.

Sistem rekomendasi merupakan model aplikasi dari hasil observasi terhadap keadaan dan keinginan pelanggan. Oleh karena itu sistem rekomendasi memerlukan model rekomendasi yang tepat agar yang direkomendasikan sesuai dengan keinginan pelanggan, serta mempermudah pelanggan mengambil keputusan yang tepat dalam menentukan produk yang akan digunakannya.

Menurut Sebastia, L et al (2009) sistem rekomendasi merupakan sebuah (web) alat personalisasi yang menyediakan pengguna sebuah informasi daftar item-item yang sesuai dengan keinginan masing-masing pengguna. Sistem rekomendasi menyimpulkan preferensi pengguna dengan menganalisa ketersediaan data pengguna, informasi tentang pengguna dan lingkungannya. Oleh karena itu Sebastia, L et al (2009) menyatakan sistem rekomendasi akan menawarkan kemungkinan dari penyaringan informasi personal sehingga hanya informasi yang sesuai dengan kebutuhan dan preferensi

pengguna yang akan ditampilkan di sistem dengan menggunakan sebuah teknik atau model rekomendasi.

Berdasarkan metode rekomendasi yang sering digunakan, sistem rekomendasi dibagi dalam beberapa klasifikasi yaitu: content-based recommendation, collaborative-based recommendation dan hybrid-based recommendation dan beberapa peneliti menambahkan metode knowledge based recommendation.



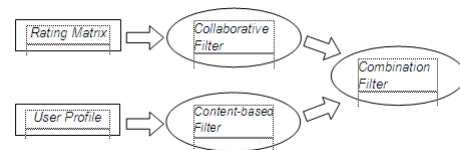
Gambar 1 Klasifikasi Sistem Rekomendasi

2. Hybrid Recommender System

Secara umum pendekatan *hybrid recommender system* adalah menggabungkan lebih dari satu pendekatan sistem rekomendasi dengan tujuan untuk mengatasi kekurangan masing-masing pendekatan, sehingga menghasilkan sebuah rekomendasi yang baik. Terdapat beberapa cara penggabungan yang dapat dilakukan yaitu :

a. Penggabungan secara linier (*Hybrid Linear Combination*)

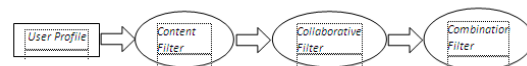
Penggabungan ini menggabungkan hasil prediksi (rating) dari metode *content-based* dan *collaborative*. Penggabungan ini dilakukan dengan cara pemberian ranking atau rating. Penggabungan ini digambarkan pada gambar 2.3 berikut :



Gambar 1 Hybrid Linear Combination

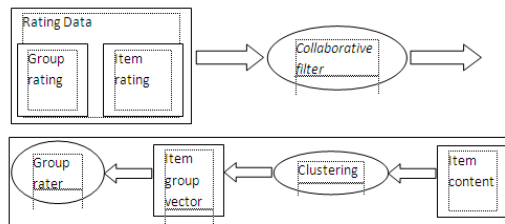
b. Penggabungan secara sekuensial (*Sequential Combination*)

Penggabungan ini adalah melakukan perhitungan pada salah satu metode (misalkan *content-based*) kemudian hasilnya digabungkan dengan metode lainnya (misalkan *collaborative*). Penggabungan ini digambarkan pada gambar 2.4 berikut :



Gambar 2 Hybrid Sequential Combination

Penggabungan *Item-based Clustering Hybrid Method (ICHM)* Penggabungan ini mengintegrasikan informasi item dan rating pengguna untuk menghitung kemiripan item-item. *Item-based clustering Hybrid Method (ICHM)* merupakan sebuah metode yang menerapkan penggabungan *hybrid recommendersystem* dengan tujuan untuk meningkatkan akurasi prediksi pendekatan *collaborative filtering* dan menangani masalah item baru yang belum dirating (*cold-star problem*).



Gambar 3 *Item-based Clustering Hybrid Method*

Burk, R. (2002) memperkenalkan taxonomi untuk sistem rekomendasi *hybrid*. Taxonomi ini diklasifikasikan ke dalam tujuh kategori, yaitu:

- Weighted hybrid* : Nilai komponen dari sistem rekomendasi yang berbeda digabungkan secara numerik atau menggunakan algoritma linier.
- Switching hybrid* : Sistem memilih komponen-komponen dari setiap rekomendasi dan menerapkan komponen yang dipilih.
- Mixed hybrid* : Rekomendasi dari berbagai sistem rekomendasi disajikan bersama.
- Feature Combination* : Fitur-fitur yang berasal dari berbagai sumber pengetahuan digabungkan dan diberikan algoritma rekomendasi.
- Feature Augmentation* : Merupakan salah satu teknik rekomendasi yang digunakan untuk menghitung sebuah fitur atau sekumpulan fitur yang kemudian menjadi bagian yang dimasukkan ke teknik berikutnya.
- Cascade* : Merupakan rekomendasi yang memiliki prioritas tinggi sebagai solusi pemecahan masalah dalam melakukan perbaikan
- Meta-level* : merupakan salah satu teknik rekomendasi yang diterapkan dan menghasilkan beberapa jenis model, yang kemudian digunakan sebagai input oleh teknik berikutnya

3. Algoritma Nearest Neighbor

Algoritma nearest neighbor adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut (Widiarsana, et al., 2011).

Algoritma nearest neighbor juga disebut dengan *lazy learner* yang mudah menyimpan data di dalam memori dan dapat mengklasifikasikan item baru dengan membandingkan item tersebut dengan item yang telah tersimpan dengan menggunakan fungsi kesamaan (*similarity function*)

Menurut Kusriani dan Emha (2009) algoritma nearest neighbor adalah pendekatan untuk mencari kasus dengan menghitung kedekatan antara kasus baru dengan kasus lama dengan berdasarkan pada pencocokan bobot dari sejumlah fitur yang ada yang memiliki kesamaan (*similarity*). Tujuan dari algoritma ini untuk mengklasifikasikan objek baru berdasarkan atribut dan *training sample*. *Classifier* tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori. Adapun rumus untuk menghitung algoritma ini yaitu :

$$Similarity(T, S) = \frac{\sum_{i=1}^n f(T_i, S_i) * w_i}{\sum w_i}$$

Keterangan :

T = kasus baru

S = kasus yang ada dalam memori (penyimpanan)

n = jumlah atribut dalam setiap kasus

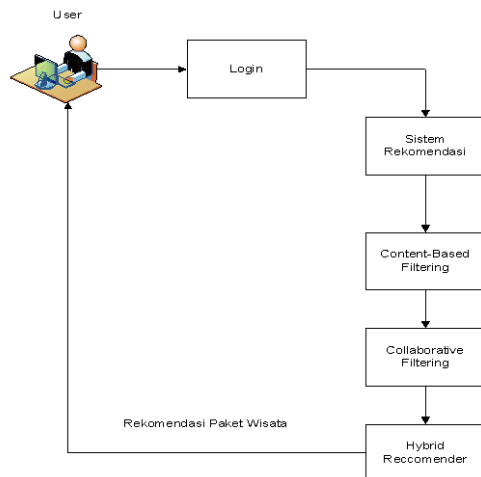
i = atribut individu antara 1 s/d n

f = fungsi *similarity* atribut i antara kasus T dan kasus S

w = bobot yang diberikan pada atribut ke-i

PEMBAHASAN

1. Blog Diagram Sistem yang Akan Dibangun



Gambar 4 Blog Diagram Sistem yang akan dibangun

Berdasarkan sistem yang akan dibangun ini user dapat menggunakan bantuan sistem agar dapat memberi keputusan mengenai paket wisata yang akan dipilih yang sesuai dengan keinginannya dan akan lebih mudah dalam melakukan pencarian.

2. Analisa Masalah

Banyaknya permasalahan yang telah dialami dalam menentukan suatu keputusan memilih paket wisata, membuat wisatawan sering mengalami kesulitan dalam mencari maupun memilih paket wisata yang sesuai dengan apa yang di inginkan. Sehingga wisatawan terkadang membutuhkan waktu yang lama dalam penyeleksian paket wisata yang sesuai dengan keinginannya dan memilih salah satu tempat wisata yang akan dikunjungi. Dengan adanya permasalahan-permasalahan tersebut mengakibatkan menurunnya daya tarik wisatawan terhadap tempat wisata yang ada di Kota Malang Raya.

Melihat permasalahan diatas maka dibutuhkan sebuah solusi baru untuk dapat menyelesaikannya, dan sebuah sistem rekomendasi merupakan salah satu solusi untuk permasalahan yang sering dihadapi oleh wisatawan. Dimana sistem akan membantu wisatawan dalam mencari paket wisata dan dapat membantu merekomendasikan paket wisata yang sesuai dengan keinginan dan kebutuhan. Dengan adanya sistem ini diharapkan dapat membantu wisatawan dalam memutuskan tempat wisata yang akan dikunjungi, sehingga segala permasalahan wisatawan dapat teratasi dengan efektif dan

efisien. Tidak menutup kemungkinan dengan sebuah solusi baru ini akan dapat meningkatkan daya tarik wisatawan logistik maupun asing untuk mengunjungi tempat wisata yang ada di Kota Malang Raya.

Dalam penyaringan informasi pada sistem rekomendasi ini dengan menggunakan metode Hybrid (Content Based - Collaborative filtering) dan menggunakan algoritma Nearest Neighbor. Dengan menggabungkan beberapa metode ini maka akan dapat memberikan hasil rekomendasi yang sesuai dengan keinginan wisatawan. Karena dalam penyaringan informasi untuk metode Hybrid (Content Based-Collaborative filtering) menggunakan profil user dan aktifitas rating terhadap tempat wisata. Sehingga wisatawan tidak akan kesulitan dalam menentukan, dan memutuskan paket wisata yang akan dipilih berdasarkan keinginannya.

3. Studi Kasus

Untuk melakukan perhitungan nearest neighbor pada sistem ini, misalkan pada database sistem telah terdapat 10 kasus user yang memilih rekomendasi paket wisata berdasarkan preferensi sebagai berikut:

Tabel 1 Atribut Pengguna Baru

No	Variabel	Bobot
1	Aktivitas	Refreshing
2	Harga	Sedang
3	Jenis wisata	Wisata Alam
4	Lama perjalanan	2 Hari

Untuk melakukan perhitungan nearest neighbor data user baru akan dibandingkan dengan data user lama yang terdapat pada tabel. Kemudian dilakukan langkah-langkah sebagai berikut untuk menentukan similarity user baru dengan user sebelumnya:

Tabel 2 Contoh Atribut Rekomendasi Pengguna Lama

No	Aktivitas	Harga
1	refreshing	MURAH
2	refreshing	MURAH
3	honeymoon	SEDANG
4	adventure	MAHAL
5	refreshing	SEDANG
6	honeymoon	MAHAL
7	refreshing	SEDANG
8	refreshing	MAHAL
9	adventure	MAHAL

10 honeymoon MURAH

Berdasarkan data atribut rekomendasi pengguna lama tersebut, maka akan dilihat bagaimana kedekatan atau kesamaan antara pengguna baru dengan pengguna lama berdasarkan kedekatan atribut atau karakter dari paket wisata yang di sukai oleh pengguna.

Langkah berikutnya adalah menghitung nilai *similarity* antara *user* baru dengan *user* lama. Maka akan ditentukan terlebih dahulu nilai dari kedekatan setiap atribut dan bobotnya. Menghitung kedekatan *user* baru dengan *user* no. 1 :

- Kedekatan nilai atribut aktivitas : refreshing - refreshing = 1
- Bobot atribut aktivitas : 1
- Kedekatan nilai atribut Harga: sedang - murah = 0.75
- Bobot atribut Harga: 0.75
- Kedekatan nilai atribut jenis wisata: wisata alam - wisata alam = 1
- Bobot atribut jenis wisata: 0.5
- Kedekatan nilai atribut lama perjalanan: 2 hari - 1 hari = 0.75
- Bobot atribut lama perjalanan: 0.25

Tabel 3Contoh Kedekatan Antar Pengguna

No	Aktivitas	Harga
1	1	0,75
2	1	0,75
3	0,25	1
4	0,5	0,75
5	1	1
6	0,25	0,75
7	1	1
8	1	0,75
9	0,5	0,75
10	0,25	0,75

Setelah diketahui hasil kedekatan antar pengguna lama dengan pengguna baru, kemudian dihitung nilai kedekatannya seperti berikut.

Similarity (t,s)

$$= \frac{(a*b)+(c*d)+(e*f)+(g*h)}{b+d+f+h}$$

$$= \frac{(1*1)+(0.75*0.75)+(1*0.5)+(0.75*0.25)}{1+0.75+0.5+0.25}$$

$$= \frac{2.25}{2.5}$$

$$= 0.9$$

Di atas adalah nilai hasil kedekatan dari antara pengguna baru dengan pengguna dengan Id-user 1, untuk melihat hasil kedekatan pengguna yang lain dapat dilihat tabel berikut.

Tabel 4Hasil Nilai kedekatan

No	Hasil
1	0,9
2	0,825
3	0,6
4	0,7
5	1
6	0,5
7	0,925
8	0,875
9	0,6
10	0,6

Dari langkah 1-10 diketahui bahwa nilai tertinggi terdapat pada user 5 dengan nilai jarak 1. Berarti user 5 memiliki kemiripan dengan user baru, sehingga hasil rekomendasi user 5 dapat direkomendasikan kepada user baru. Dari tabel diatas akan di ambil dari 5 nilai kedekatan tertinggi, kedekatan tertinggi antara user baru dengan user lama di antaranya adalah user 5, user 7, user 1, user 8, dan user 2. Berdasarkan hasil kedekatan antar user tersebut akan di ambil data history paket wisata yang telah dilakukan oleh user dan akan dijadikan kandidat paket wisata yang akan direkomendasikan, kemudian di cari hasil peratingan dengan menggunakan metode collaborative filtering. Berikut adalah data history paket wisata user dari hasil perhitungan dengan metode content based filtering.

Tabel 5History Pembelian Paket Wisata User Rekomendasi

No	Paket Wisata
1	Paket Bromo Sempu 2 Hari
2	Paket Malang Batu-Blitar 3 Hari
3	Paket Gunung Bromo Tour
4	Paket Kawah Ijen Bromo Malang Batu 4D3N
5	Paket Surabaya Tour 2 Hari

Berikut ini adalah data paket wisata yang akan direkomendasikan menggunakan proses perhitungan metode collaborative filtering dengan total yang didapat dari data transaksi yang berjumlah 500.

Tabel 6 Paket Wisata Rekomendasi

Paket Wisata	Total
Paket Bromo Sempu 2 Hari	23
Paket Malang Batu-Blitar 3 Hari	20
Paket Gunung Bromo Tour	32
Paket Kawah Ijen Bromo Malang Batu 4D3N	14
Paket Surabaya Tour 2 Hari	18

Proses hitung paket wisata rekomendasi berdasarkan rating untuk setiap item yang akan direkomendasikan :

$$\begin{aligned} \text{Paket Bromo Sempu 2 Hari} &= \left(\frac{4 \times 100}{5 \times 1} \right) \\ &= \frac{400}{5} \\ &= 80 \end{aligned}$$

Di atas merupakan proses perhitungan nilai rekomendasi Paket Bromo Sempu 2 Hari, untuk paket wisata rekomendasi lainnya dapat dilihat pada tabel 3.15.

Tabel 7 Hasil Rekomendasi

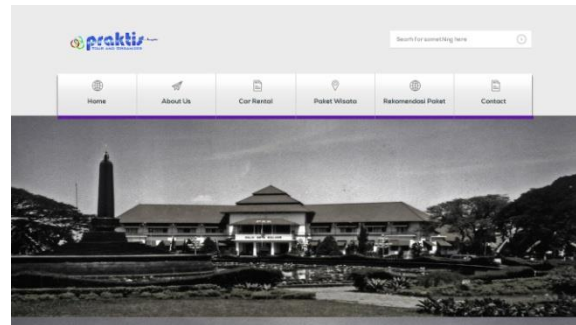
Paket Wisata	NR
Paket Gunung Bromo Tour	100
Paket Bromo Sempu 2 Hari	80
Paket Malang Batu-Blitar 3 Hari	80
Paket Kawah Ijen Bromo Malang Batu 4D3N	60
Paket Surabaya Tour 2 Hari	60

Berdasarkan hasil rekomendasi dengan nilai 3 tertinggi akan dijadikan rekomendasi kepada user baru dan ditampilkan pada halaman rekomendasi. Dari tabel 3.15 paket wisata yang akan di rekomendasikan adalah Paket Gunung Bromo Tour, Paket Bromo Sempu 2 Hari, dan Paket Malang Batu-Blitar 3 Hari.

IMPLEMENTASI DAN PENGUJIAN

1. Implementasi Halaman Utama Untuk User (Customer)

Pada halaman depan ini terdapat beberapa menu utama yang dapat digunakan oleh user, terdapat menu home, about us, car rental, paket wisata, rekomendasi paket, dan contact. Berikut ini merupakan tampilan dari halaman utama untuk user.



Gambar 5 Halaman Utama

2. Implementasi Menu Rekomendasi

Pada menu rekomendasi ini dapat digunakan ketika user tersebut sudah mengisi form rekomendasi. Tujuan menu ini adalah mempermudah dalam memilih paket wisata dan menentukan paket wisata mana yang akan dituju sesuai dengan atribut rekomendasi setiap user. Proses perhitungan pada menu rekomendasi ini menggunakan metode content based filtering dan collaborative filtering. Menu Count adalah menu untuk melihat proses perhitungan sistem berdasarkan hasil rekomendasi tersebut. Berikut ini adalah tampilan menu rekomendasi.

Gambar 6 Menu Rekomendasi

3. Pengujian Beta (Kuesioner)

Pengujian beta dengan menggunakan kuesioner terhadap pihak terkait. Tujuan dari pengujian beta dengan kuesioner kepada pengguna, apakah aplikasi yang dibuat dapat diterapkan untuk membantu dalam penyelesaian masalah dan menentukan atau memilih paket

wisata yang akan direkomendasikan kepada pengguna.

Dalam pengujian kuesioner sistem ini menggunakan 30 data pengguna baru yang memiliki kesamaan atribut rekomendasi, di mana setiap pengguna baru tersebut akan menjalankan sistem yang telah dibuat, Kemudian melakukan pemilihan paket wisata dari total semua paket wisata yang telah diproses untuk di jadikan rekomendasi.

Hasil kecocokan tiap pengguna baru akan dijumlahkan dengan hasil kecocokan pengguna lainnya, total hasil kecocokan semua paket wisata yang di rekomendasikan kepada pengguna baru akan di bagii dengan jumlah nilai korespondensi paket wisata rekomendasikan. Kemudian hasil tersebut akan dikalikan dengan 100%, maka akan didapat persentase hasil pengujian kuesioner. Hasil kuesioner terhadap sistem yang dijalankan oleh pengguna baru dapat ditunjukkan pada tabel 4.2.

Tabel 8 Hasil Kuesioner Dengan Metode Hybrid(Content based – Collaborative)

Nama	BKI 2D	BS2 H	KI2 D	KIB F2D
Annas	v	v		v
Patria	v	v	v	
Buchory	v		v	v
Nur	v	v		v
Rudianto		v	v	v
Resa	v	v	v	
Yevi	v	v		v
Ria	v		v	v
Rendik	v	v	v	
David	v	v	v	v
Aan		v		v
Chris	v	v	v	v
Asti		v	v	v
Ni Made		v	v	v
Yudha	v	v	v	
Agung D		v	v	v
JUMLAH	25	23	23	22
TOTAL	93			

BKI2D=Paket Wisata Bromo Kawah Ijen Tour 2 Hari

BS2H = Paket Wisata Bromo Sempu 2 Hari

KI2D = Paket Wisata Kawah Ijen Tour 2 Hari

KIBF2D = Paket Wisata Kawah Ijen Blue Fire Tour 2H

BKI2D = Paket Wisata Bromo Kawah Ijen Tour 2 Hari

Hasil kuesioner sistem pada metode hybrid(content based-collaborative) dapat dihitung dengan cara menghitung sebagai berikut :

$$\begin{aligned}\text{Hybrid} &= (\text{Total Kecocokan} / \text{Total Koresponden}) * 100\% \\ &= (93 / 120) * 100\% \\ &= 0.77 * 100\% \\ &= 77 \%\end{aligned}$$

PENUTUP

1. Kesimpulan

Berdasarkan implementasi dan pengujian yang telah dilakukan maka dapat diambil kesimpulan sebagai berikut :

- Telah berhasil dibuat sistem rekomendasi dengan metode hybrid dengan menggunakan algoritma nearest neighbor pada paket wisata.
- Pada pengujian Beta (kuesioner) didapatkan nilai hasil ketepatan untuk metode hybrid(content based-collaborative) 77% mendekati dengan hasil perhitungan sistem dan nilai hasil ketepatan.

2. Saran

Penelitian ini menggunakan metode hybrid (content-based dan collaborative) dengan algoritma nearest neighbor, oleh karena itu dapat dikembangkan dan diteliti lebih lanjut dengan penggabungan metode serta algoritma lainnya pada ruang lingkup sistem rekomendasi. Sehingga dapat menghasilkan rekomendasi yang lebih baik.

DAFTAR PUSTAKA

- Adomavicius, G dan Tuzhilin, A. Toward the Next Generation of Recommender Systems: A survey of the state-of-the-art and possible extensions. IEEE transaction on knowledge and data engineering 17(6): hal.734-749. 2005.
- Anonymous. "Keputusan Dirjen Pariwisata Nomor (Kep-06/U/IV/1992) tentang Pelaksanaan Ketentuan Usaha Konvensi, Perjalanan Intensif dan Pameran", Dirjen Pariwisata Republik Indonesia. 1992.
- Baudisch, P. Joining collaborative and content-based filtering. In Integrated Publication and Information Systems Institute IPSI: hal. 1-5. 1999.
- Belka, T dan Plöbning, M. Designing recommender systems for tourism. Cairo. In Proceedings of ENTER. 2004.

- [5] Bogers, T dan Bosch, A. V. D. Comparing and evaluating information retrieval algorithms for news recommendation. In the Proceedings of the 2007 ACM Conference, Minneapolis, MN, USA. RecSys'07. ACM, New York, pp 141-144. 2007.
- [6] Burk, R. Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction, Vol. 12(4), 331-370. 2002.
- [7] Burk, R. Hybrid recommender systems. In Proceedings of The Adaptive Web, pp 377-408. 2007.
- [8] Fathansyah. Basis Data. Informatika Bandung, Bandung. 1999.
- [9] Jogyanto.HM. Analisis & Desain Sistem Informasi. Yogyakarta: Andi Offset. 1990.
- [10] Kadir, Abdul. Dasar Pemrograman Web Dinamis Menggunakan PHP. Yogyakarta: Andi Offset. 2002.
- [11] Karyono, Hari. Kepariwisata. Jakarta: Gramedia Widia Sarana Indonesia. 1997.
- [12] Kusriani & Emha. Algoritma Data Mining. Edisi ke-1. Yogyakarta: Andi Offset. 2009.
- [13] Li, Qing dan Kim, Byeong Man. An Approach for Combining Content-based and Collaborative Filters. Departement of Computer Science, Kumoh National Institute of Technology. 2002.
- [14] Lundberg, Donald D. Ekonomi Pariwisata. Jakarta: PT Gramedia Pustaka Utama. 1997.
- [15] Pendit, Nyoman S. Ilmu Pariwisata Sebuah Pengantar Perdana. Jakarta: PT Pradnya Paramita. 1990.
- [16] Ricci, F. Travel recommender system. IEEE Intelligent Systems 17(6): 55-57. 2002.
- [17] Ricci, F., Rokach, L., Shapira, B., Kantor, B, P. Recommender System Handbook. London: Springer New York Dordrecht Heidelberg. 2010.
- [18] Sebastia, L., Garcia, I., Onaindia, E., Guzman, C. e-Tourism: A tourist recommendation and planning application. International Journal on Artificial Intelligence Tools 18(5): 717-738. 2009.
- [19] Smith, Valene L. Tourism Alternative. USA : The University of Ttennsylvania Press. 1991.
- [20] Suwanto, Gamal. SH, Dasar-dasar Pariwisata. Yogyakarta: Andi Offset. 1997.
- [21] Widiarsana, O., Putra, N.W., Budiya, P.G.I., Bismantara, A.N.I., Mahajaya, S.N. Data Mining: Metode Clasification K-Nearest Neighbor (KNN). Bali: Program Studi Teknologi Informasi Universitas Udayana. 2011.