

IMPLEMENTASI SISTEM REKOMENDASI *GAME* MENGUNAKAN HYBRID FILTERING PADA PLATFROM DIGITAL STEAM

Oleh:

Zhulkhoir Rifat Rianda Raja Faqih¹

Farindika Metandi²

Noor Alam Hadiwijaya³

Politeknik Negeri Samarinda

Alamat: Jl. Cipto Mangun Kusumo, Sungai Keledang, Kec. Samarinda Seberang, Kota
Samarinda, Kalimantan Timur (75242).

Korespondensi Penulis: zhulkhoirrifat@gmail.com, farindika@polnes.ac.id,
alamhadiwijaya@polnes.ac.id.

Abstract. *A recommendation system is an essential feature in modern digital platforms, especially in the gaming industry, to help users discover content that matches their preferences. Unfortunately, inaccurate or irrelevant recommendations can reduce user satisfaction and engagement levels. To address this issue, this research develops a game recommendation system based on hybrid filtering, combining two main approaches: Neural Collaborative Filtering (NCF) and cosine similarity, using data from the Steam platform. NCF is utilized to learn user-game interaction patterns through deep learning techniques, while cosine similarity measures the similarity between games based on their content attributes. The system development process follows the Software Development Life Cycle (SDLC) Waterfall model, starting from requirement analysis, system design, implementation, to testing. The system functionality is validated using black-box testing, ensuring that it runs as expected. Furthermore, model accuracy evaluation is conducted using Binary Crossentropy Loss and Root Mean Squared Error (RMSE) metrics. The test results show that the NCF model achieves optimal performance with a loss value of 0.3602, RMSE of 0.2211, val_loss of 0.4828, and val_rmse of 0.2604. Meanwhile, the*

IMPLEMENTASI SISTEM REKOMENDASI *GAME* MENGUNAKAN HYBRID FILTERING PADA PLATFROM DIGITAL STEAM

cosine similarity method successfully provides relevant recommendations, such as the case where the input "Coral Island" resulted in the recommendation of the game "My Time at Sandrock" with a similarity score of 0.4888. The system is capable of providing 10 game recommendations tailored to user input preferences.

Keywords: *Steam, Recommender System, Hybrid Filtering, Neural Collaborative Filtering, Cosine Similarity.*

Abstrak. Sistem rekomendasi merupakan salah satu fitur penting dalam platform digital modern, terutama dalam industri *game*, untuk membantu pengguna menemukan konten yang relevan dengan preferensi mereka. Sayangnya, sistem rekomendasi yang tidak akurat atau kurang relevan dapat menyebabkan penurunan kepuasan pengguna dan mengurangi tingkat keterlibatan. Untuk mengatasi masalah tersebut, penelitian ini mengembangkan sistem rekomendasi *game* berbasis *hybrid filtering* dengan menggabungkan dua pendekatan utama, yaitu *Neural Collaborative Filtering* (NCF) dan *cosine similarity* pada dataset dari platform *Steam*. NCF digunakan untuk mempelajari pola interaksi antara pengguna dan *game* menggunakan teknik *deep learning*, sedangkan *cosine similarity* dimanfaatkan untuk mengukur tingkat kemiripan antar *game* berdasarkan atribut kontennya. Proses pengembangan sistem dilakukan menggunakan metodologi *Software Development Life Cycle (SDLC) Waterfall*, dimulai dari tahap analisis kebutuhan, desain sistem, implementasi, hingga pengujian. Pengujian dilakukan dengan metode *black-box testing* untuk memastikan fungsionalitas sistem berjalan sesuai harapan. Selain itu, evaluasi akurasi model dilakukan menggunakan metrik *Binary Crossentropy Loss* dan *Root Mean Squared Error* (RMSE). Hasil pengujian menunjukkan model NCF mencapai performa optimal dengan nilai *loss* 0,3602, RMSE 0,2211, *val_loss* 0,4828, dan *val_rmse* 0,2604. Sementara itu, metode *cosine similarity* berhasil memberikan rekomendasi yang relevan, seperti contoh kasus input "*Coral Island*" yang menghasilkan rekomendasi *game* "*My Time at Sandrock*" dengan skor similarity sebesar 0,4888. Sistem ini mampu memberikan 10 rekomendasi *game* yang disesuaikan dengan preferensi input pengguna.

Kata Kunci: *Steam, Sistem Rekomendasi, Hybrid Filtering, Neural Collaborative Filtering, Cosine Similarity.*

LATAR BELAKANG

Pada saat ini perkembangan teknologi yang sangat pesat telah membawa perubahan besar dalam berbagai aspek pada kehidupan manusia, termasuk dalam industri hiburan digital. Salah satu sektor yang sangat berpengaruh adalah industri *game*. Platform distribusi *game* digital seperti *Steam*, menjadi sangat populer di kalangan penggemar *game* di seluruh dunia karena diberikan kemudahan akses, berbagai pilihan *game*, dan fitur-fitur sosial yang dapat digunakan untuk berinteraksi antar pengguna. *Steam*, sebagai salah satu platform terbesar, menyediakan ribuan *game* dengan berbagai kategori yang dapat diunduh dan dimainkan para pengguna diseluruh dunia.

Namun, dengan jumlah *game* yang terus bertambah, pengguna sering kali mengalami kesulitan dalam menemukan *game* yang sesuai dengan preferensi mereka. Hal ini menjadi tantangan tersendiri bagi platform seperti *Steam* untuk menyajikan rekomendasi *game* yang relevan, personal, dan tepat sasaran.

Untuk mengatasi permasalahan tersebut, sistem rekomendasi menjadi solusi penting dalam menyaring informasi dan menyajikan konten yang relevan. Metode sistem rekomendasi secara umum terbagi menjadi tiga pendekatan utama, yaitu *collaborative filtering*, *content-based filtering*, dan *hybrid filtering*. *Collaborative filtering* menganalisis interaksi antar pengguna dan item untuk menghasilkan rekomendasi, sedangkan *content-based filtering* memanfaatkan kemiripan atribut antar item. Kombinasi keduanya membentuk pendekatan *hybrid* yang bertujuan meningkatkan relevansi rekomendasi.

Dalam penelitian ini, sistem rekomendasi dikembangkan dengan menggabungkan pendekatan *Neural Collaborative Filtering* (NCF) dan *content-based filtering* berbasis cosine similarity. NCF memanfaatkan kemampuan *deep learning* untuk menangkap relasi non-linier antara pengguna dan item melalui representasi vektor laten. Sementara itu, *content-based filtering* memanfaatkan informasi fitur seperti kategori, *tags*, dan pengembang dalam menentukan kemiripan antar *game*. Pendekatan hybrid ini diharapkan dapat menghasilkan rekomendasi yang lebih akurat dan personalisasi yang lebih baik bagi pengguna *Steam*.

Pengembangan sistem dilakukan dengan menggunakan model *Software Development Life Cycle* (SDLC) berbasis pendekatan *Waterfall*, yang mencakup tahapan analisis, perancangan, implementasi, pengujian, dan pemeliharaan. Implementasi sistem

IMPLEMENTASI SISTEM REKOMENDASI *GAME* MENGUNAKAN HYBRID FILTERING PADA PLATFROM DIGITAL STEAM

rekomendasi ini juga dikembangkan dalam bentuk aplikasi *desktop* menggunakan *Tkinter*, guna memberikan antarmuka pengguna yang sederhana, dan mudah diakses.

KAJIAN TEORITIS

Sistem Rekomendasi

Sistem rekomendasi adalah sebuah teknologi yang dirancang untuk memberikan saran atau rekomendasi mengenai item tertentu kepada pengguna. Tujuan utama dari sistem ini adalah untuk membantu pengguna dalam membuat keputusan yang lebih baik dan lebih terinformasi. Dengan menggunakan berbagai algoritma dan teknik, sistem rekomendasi dapat menganalisis preferensi dan perilaku pengguna untuk menyarankan produk, layanan, atau konten yang paling relevan dan menarik. Hal ini sangat berguna dalam berbagai konteks, seperti *e-commerce*, platform *streaming*, dan layanan media sosial, di mana pilihan yang tepat dapat meningkatkan kepuasan dan keterlibatan pengguna. (Februariyanti, Laksono, Wibowo, & Utomo, 2021)

Steam

Steam adalah platform distribusi *game* online yang sangat dikenal dan digunakan secara luas di seluruh dunia. Melalui *Steam*, pengguna memiliki akses untuk membeli, mengunduh, dan memainkan berbagai jenis *game*, mulai dari *indie* hingga *game* AAA. Selain itu, pengguna juga dapat memberikan ulasan atau evaluasi terhadap *game* yang telah dimainkan, yang sering digunakan oleh pemain lain sebagai referensi untuk memilih *game*. Fitur tambahan seperti komunitas pemain, forum diskusi, dan pembaruan otomatis menjadikan *Steam* sebagai salah satu platform *game* paling lengkap dan mendukung pengalaman bermain *game* secara menyeluruh. (Pangestu, Arifin, & Safitri, 2023)

Collaborative Filtering

Metode *collaborative filtering* adalah sebuah teknik yang digunakan untuk mengevaluasi dan menyaring item dengan memanfaatkan persepsi atau opini dari pengguna lain. Teknik ini melakukan proses penyaringan dengan menganalisis data dari semua pengguna untuk mendapatkan informasi yang relevan, sehingga dapat memberikan rekomendasi yang tepat. Dalam prosesnya, *collaborative filtering* bekerja berdasarkan kemiripan karakteristik pengguna, yang memungkinkan sistem untuk memberikan

informasi baru berdasarkan pola dalam kelompok pengguna yang memiliki kesamaan. (Februariyanti dkk., 2021)

Content-based Filtering

Content-based filtering adalah metode untuk mendapatkan hasil rekomendasi dengan mencari kesamaan bobot istilah dalam *bag-of-words* yang diperoleh dari pemrosesan deskripsi dan judul. Algoritma ini menganalisis konten untuk menentukan item yang memiliki kemiripan karakteristik. (Fajriansyah, Adikara, & Widodo, 2021)

Hybrid Filtering

Metode *hybrid filtering* yaitu metode menggabungkan *content-based filtering* dan *collaborative filtering*. Metode ini memanfaatkan kelebihan masing-masing teknik untuk memberikan rekomendasi yang lebih akurat dan relevan. *Collaborative filtering* berfokus pada preferensi dan perilaku pengguna lain yang mirip, sedangkan *content-based filtering* menganalisis atribut dan karakteristik item untuk memberikan rekomendasi yang sesuai dengan preferensi individu. (Mukti & Mardhiyah, 2022)

Deep Learning

Deep learning adalah cabang dari *machine learning* yang algoritmanya terinspirasi oleh cara kerja otak manusia, yang mejadikannya sangat efisien dalam mengolah data yang kompleks. Algoritma ini bekerja dengan meniru neuron-neuron di otak manusia, yang saling terhubung dan bekerja secara paralel untuk memproses informasi. Struktur ini biasanya dikenal sebagai Jaringan Syaraf Tiruan (JST) atau *Artificial Neural Network* (ANN). (Alfarizi, Al-farish, Taufiqurrahman, Ardiansah, & Muhamad Elgar, 2023)

Term Frequency-Inverse Document Frequency

TF-IDF (*Term Frequency-Inverse Document Frequency*) adalah metode yang digunakan untuk mengubah data teks menjadi format numerik atau vektor, sehingga dapat lebih mudah diproses dan dianalisis. Metode ini memberikan bobot pada setiap kata dalam sebuah dokumen berdasarkan seberapa sering kata tersebut muncul dan seberapa umum kata tersebut di seluruh *dataset*. Dengan menggunakan TF-IDF, kata-kata yang

IMPLEMENTASI SISTEM REKOMENDASI *GAME* MENGUNAKAN HYBRID FILTERING PADA PLATFROM DIGITAL STEAM

sering muncul dalam satu dokumen tetapi jarang muncul di dokumen lain mendapatkan bobot yang lebih tinggi. Proses ini membantu meningkatkan akurasi hasil analisis teks dan biasanya melibatkan dua komponen utama: Ketentuan Frekuensi (TF) yang mengukur seberapa sering sebuah kata muncul dalam dokumen, dan Dokumen Terbalik (IDF) yang mengukur seberapa jarang kata tersebut muncul di seluruh dokumen dalam *dataset*. (Al Rasyid & Ningsih, 2024)

Cosine Similarity

Cosine similarity adalah metode yang sering digunakan dalam bidang *information retrieval* dan ilmu terkait. Teknik ini merepresentasikan dokumen teks sebagai vektor yang terdiri dari istilah-istilah yang ada di dalamnya. Kesamaan antara dua dokumen kemudian diukur dengan menghitung nilai kosinus sudut yang terbentuk antara vektor kedua dokumen tersebut. Metode ini banyak digunakan dalam pemrosesan teks, analisis data, dan sistem rekomendasi untuk membandingkan dokumen, item, atau profil pengguna berdasarkan konten atau fitur yang relevan. (Al Rasyid & Ningsih, 2024)

Hyperparameter

Hyperparameter adalah parameter yang tidak bisa diubah oleh algoritma pembelajaran itu sendiri, melainkan harus diatur sebelum proses pelatihan dimulai. Menyusun *hyperparameter* atau biasa disebut *hyperparameter tuning* merupakan tahapan yang krusial dalam membangun model *machine learning*. *Tuning* dilakukan agar bisa mendapatkan model dengan peforma yang optimal. (Putra, Suprpto, & Bukhori, 2022)

1. *Epoch*

Epoch adalah *hyperparameter* yang menentukan berapa kali *neural network* melakukan proses pelatihan terhadap seluruh *dataset*. Satu *epoch* berarti seluruh *dataset* telah melalui proses pelatihan pada *neural network* dan akan dikembalikan ke awal. Proses pelatihan model tidak bisa dilakukan hanya dengan satu *epoch* karena *dataset* yang digunakan terbatas dan proses iteratif diperlukan untuk mengoptimalkan grafik *gradient descent*. Jumlah *epoch* yang diperlukan tergantung pada keberagaman *dataset* yang dimiliki. (Putra dkk., 2022)

2. *ReLU Activation*

Rectified Linear Activation (ReLU) merupakan *activation function* yang umum digunakan pada *hidden layer* karena mudah di

implementasikan dan efektif dalam mengatasi keterbatasan *activation function* lainnya seperti *sigmoid*. (Putra dkk., 2022)

3. ***Sigmoid Activation***

Sigmoid atau juga dikenal sebagai *logistic function*, digunakan didalam algoritma klasifikasi regresi. Fungsi ini menerima nilai real apapun sebagai nilai *input* dan menghasilkan *output* dalam rentang 0 dan 1. Semakin besar nilai input (semakin positif) maka semakin dekat nilai output ke 1. Sebaliknya, semakin kecil nilai input (semakin negatif) maka semakin dekat nilai *output* ke 0. (Putra dkk., 2022)

4. ***Binary Cross-Entropy***

Binary cross-entropy merupakan fungsi *loss* yang digunakan dalam permasalahan klasifikasi biner. Fungsi ini digunakan saat model harus memilih antara dua kemungkinan, seperti ya atau tidak, 0 atau 1, A atau B, atau pilihan dua arah seperti kiri atau kanan. (Putra dkk., 2022)

5. ***Mean Squared Error***

Mean Squared Error (MSE) merupakan salah satu metrik evaluasi yang sering digunakan dalam *machine learning* untuk mengukur tingkat kesalahan prediksi sebuah model. Nilai MSE diperoleh dari rata-rata kuadrat selisih antara nilai prediksi dengan nilai aktual. Semakin kecil nilai *Mean Squared Error*, maka semakin baik kemampuan model dalam merepresentasikan data aktual, yang menunjukkan performa prediksi yang tinggi. (Amansyah, Indra, Nurlaelasari, & Juwita, 2024)

6. ***Root Mean Squared Error***

Root Mean Squared Error (RMSE) adalah metrik yang dipakai untuk menilai seberapa baik model melakukan prediksi. Metrik ini diperoleh dengan menghitung akar kuadrat dari *Mean Squared Error (MSE)*. RMSE sering digunakan untuk mengevaluasi kinerja model prediksi karena memberikan ukuran yang intuitif atas besarnya kesalahan dalam prediksi. Nilai RMSE yang rendah atau mendekati nol menunjukkan bahwa prediksi model sangat mendekati data sebenarnya. (Putra dkk., 2022)

7. ***Adam***

IMPLEMENTASI SISTEM REKOMENDASI *GAME* MENGUNAKAN HYBRID FILTERING PADA PLATFROM DIGITAL STEAM

Adaptive Moment Estimation (Adam) adalah salah satu *optimizer* yang sangat populer digunakan. Cara kerja Adam adalah menggabungkan kelebihan dari dua ekstensi *stochastic gradient descent*, yaitu *adaptive gradient algorithm* dan *root mean square propagation*. Melalui kombinasi ini, Adam mampu memberikan optimasi yang efektif dalam menangani gradien yang jarang (*sparse gradients*) pada masalah dengan *noise* yang tinggi. Penggunaan *optimizer* yang dapat menurunkan gradien ini sangat efisien ketika diaplikasikan pada data dan parameter yang besar. (Witanto dkk., 2022)

Python

Python adalah bahasa pemrograman yang pertama kali dikembangkan pada akhir 1980-an oleh seorang peneliti dan pemrogram berkebangsaan Belanda, Guido van Rossum. *Python* mendukung berbagai teknik pemrograman, termasuk pemrograman berorientasi objek (OOP) dan pemrograman berbasis fungsi (*functional programming*). *Python* menyediakan banyak pustaka standar (*standard libraries*) secara bawaan, sehingga pengguna tidak perlu menulis kode dari awal untuk fungsi-fungsi umum yang sering digunakan dalam pembuatan program. (Budhi Gustiandi, 2023)

1. *NumPy*

NumPy adalah salah satu pustaka *Python* yang digunakan untuk mengimplementasikan *array* dan matriks multidimensi, serta menyediakan berbagai operasi matematika tingkat tinggi. Pustaka ini sangat efisien dalam hal performa dan memori, sehingga cocok untuk digunakan dalam komputasi numerik. Dengan *NumPy*, pengguna dapat melakukan operasi matematika kompleks, seperti aljabar linear dan statistik dengan lebih mudah dan cepat. (Amanda Muchsin Chalik, Bilal Abdul Qowy, Faiz Hanafi, & Ahlijati Nuraminah, 2021)

2. *Pandas*

Pandas adalah pustaka *Python* yang sangat populer untuk analisis data, menawarkan dukungan untuk struktur data yang cepat, fleksibel, dan ekspresif yang dirancang untuk bekerja dengan data "relasional" atau "berlabel". Saat ini, *Pandas* menjadi pustaka yang sangat diperlukan untuk

menyelesaikan analisis data dunia nyata yang praktis menggunakan *Python*. (Alfarizi dkk., 2023)

3. *Matplotlib*

Matplotlib adalah pustaka yang digunakan untuk visualisasi data dalam *Python*. Visualisasi data berperan penting dalam memahami data secara lebih mendalam sebelum melakukan pemrosesan data dan melatih model *machine learning*. Dengan visualisasi yang baik, kita dapat mengidentifikasi pola, anomali, dan trend dalam data, yang dapat membantu dalam pengambilan keputusan dan penyesuaian algoritma. Pustaka ini menyediakan berbagai jenis grafik dan diagram, seperti plot garis, histogram, *scatter plot*, dan banyak lagi, yang memungkinkan pengguna untuk menggambarkan data dengan cara yang mudah dimengerti. (Alfarizi dkk., 2023)

4. *Tkinter*

Modul *tkinter* adalah pustaka *Python* yang memungkinkan pembuatan aplikasi dengan antarmuka grafis pengguna (*GUI*) secara cepat dan mudah. *GUI* mengacu pada tampilan antarmuka yang memungkinkan pengguna berinteraksi melalui ikon dan elemen grafis lainnya. *Tkinter* sangat cocok untuk membuat tampilan aplikasi desktop yang memerlukan antarmuka visual yang intuitif. (Warnilah, Sutisna, Mulyana, Nuraeni, & Widiyanto, 2022)

5. *TensorFlow*

TensorFlow adalah pustaka populer untuk membuat model *machine learning*. Pustaka ini mendukung berbagai *toolkit* yang memungkinkan pembangunan model pada berbagai tingkat abstraksi. Dengan *TensorFlow*, pengguna dapat membuat model dari level dasar hingga tingkat tinggi, menggunakan *API* yang fleksibel dan beragam. (Alfarizi dkk., 2023)

6. *Scikit-learn*

Scikit-learn adalah pustaka yang menawarkan berbagai algoritma untuk pembelajaran mesin baik yang terawasi (*supervised*) maupun tidak terawasi (*unsupervised*). Pustaka ini dibangun di atas teknologi yang

IMPLEMENTASI SISTEM REKOMENDASI *GAME* MENGUNAKAN HYBRID FILTERING PADA PLATFROM DIGITAL STEAM

mungkin sudah familiar, seperti *NumPy*, *pandas*, dan *Matplotlib*. Dengan *scikit-learn*, pengguna dapat dengan mudah mengimplementasikan berbagai metode *machine learning*, seperti regresi, klasifikasi, klasterisasi, dan pengurangan dimensi. (Alfarizi dkk., 2023)

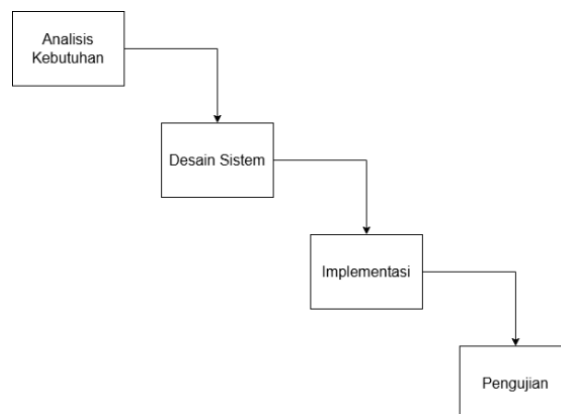
Software Development Life Cycle

SDLC adalah sebuah metode yang menggunakan pendekatan cara kerja perangkat lunak secara berurutan, dimulai dari tahap analisis, desain, implementasi, pengujian dan perawatan. (Darmawan & Geni, 2023)

METODE PENELITIAN

Penelitian ini menggunakan model *SDLC (Software Development Life Cycle) Waterfall*, yang terdiri dari beberapa tahapan yang dijalankan secara berurutan. Model ini menekankan pendekatan sistematis di mana setiap tahap harus diselesaikan sebelum melanjutkan ke tahap berikutnya. Adapun tahapan *SDLC* yang digunakan dalam penelitian ini hanya mencakup hingga tahap *testing* saja, tanpa melanjutkan ke tahap implementasi dan pemeliharaan, mengingat fokus penelitian ini terbatas pada pengembangan dan pengujian sistem.

Gambar 1. Tahapan Penelitian Model *Waterfall*



Analisis Kebutuhan

Penulis bertujuan untuk mengidentifikasi kebutuhan pengguna dan sistem rekomendasi yang akan dibangun.

1. Identifikasi Analisis Kebutuhan

Identifikasi kebutuhan pengguna seperti mengumpulkan informasi apa yang dibutuhkan pengguna didalam sistem rekomendasi.

2. Data Sistem Rekomendasi

IMPLEMENTASI SISTEM REKOMENDASI *GAME* MENGUNAKAN HYBRID FILTERING PADA PLATFROM DIGITAL STEAM

Identifikasi data yang dibutuhkan untuk membangun sistem rekomendasi seperti data *game* dan sampel data pengguna.

3. Alat

Menyiapkan alat yang dibutuhkan dalam membangun sistem rekomendasi.

Desain Sistem

Pada tahap ini diagram alir dan desain antarmuka pada aplikasi sistem rekomendasi akan dibuat.

Implementasi

Tahap ini melibatkan pengkodean dan pengembangan sistem rekomendasi sesuai dengan desain yang telah dibuat dari *pre-processing* data, membuat model sistem rekomendasi, dan membuat antarmuka pengguna.

Pengujian

Tahap ini bertujuan untuk memastikan bahwa sistem rekomendasi berfungsi sesuai harapan, dengan menggunakan metode *black-box testing* untuk menguji aplikasi yang telah dikembangkan.

HASIL DAN PEMBAHASAN

Pada tahap awal, dilakukan identifikasi dan pengumpulan berbagai kebutuhan yang diperlukan dalam pembangunan sistem rekomendasi. Ini mencakup kebutuhan pengguna, data pendukung, serta tools yang akan digunakan dalam proses pengembangan. Langkah ini bertujuan untuk memastikan perancangan dan pembuatan sistem rekomendasi berjalan sesuai dengan tujuan dari penelitian.

Analisis Kebutuhan

Pada tahap awal, dilakukan identifikasi dan pengumpulan berbagai kebutuhan yang diperlukan dalam pembangunan sistem rekomendasi. Ini mencakup kebutuhan pengguna, data pendukung, serta tools yang akan digunakan dalam proses

pengembangan. Langkah ini bertujuan untuk memastikan perancangan dan pembuatan sistem rekomendasi berjalan sesuai dengan tujuan dari penelitian.

1. Identifikasi Kebutuhan Pengguna Sistem

Kebutuhan pengguna terhadap sistem rekomendasi diidentifikasi melalui kuesioner. Hasilnya menunjukkan bahwa pengguna menginginkan sistem yang dapat merekomendasikan *game* berdasarkan preferensi seperti judul, *genre*, *developer*, *publisher*, kategori, *tag*, serta kemiripan dengan pengguna lain. Untuk itu, digunakan metode *hybrid filtering* yang menggabungkan *content-based* dan *collaborative filtering* guna menghasilkan rekomendasi yang lebih relevan.

2. Data Sistem Rekomendasi

Pengembangan sistem menggunakan dua jenis data:

a) Data *Game*

Data *game* diambil dari dataset publik *Kaggle*, mencakup atribut seperti AppID, nama *game*, *genre*, kategori, *tag*, *developer*, *publisher*, dan harga.

Tabel 1. Sampel Data *Game*

Appid	Name	Developers	Publishers	Categories	Genres	Tags	Price
4000	Garry's Mod	Facepunch Studios	Valve	Single-player, Multi-player, PvP	Indie, Simulation	Sandbox, Multiplayer, Funny, Moddable, Building	9.99
570	Dota 2	Valve	Valve	Multi-player, Co-op, Steam Trading Cards,	Action, Free to Play, Strategy	Free to Play, MOBA, Multiplayer, Strategy	3.99
550	Left 4 Dead	Valve	Valve	Single-player, Multi-player, PvP, Online PvP	Action	Zombies, Co-op, FPS, Multiplayer, Shooter	9.99

IMPLEMENTASI SISTEM REKOMENDASI *GAME* MENGUNAKAN HYBRID FILTERING PADA PLATFROM DIGITAL STEAM

b) Data Pengguna

Data pengguna diperoleh dari kuesioner yang disebarkan ke komunitas gamer Steam, mencakup nama, usia, username Steam, judul game yang dimainkan, durasi bermain, tingkat kepuasan, dan apakah game tersebut direkomendasikan.

Tabel 2. Sampel Data Pengguna

Nama lengkap	Usia	Jenis Kelamin	Steam username	Judul game	Waktu bermain	Apakah anda puas bermain game tersebut?	Apakah anda merekomendasikan kepada orang lain?	Apakah Anda setuju untuk memberikan data Anda untuk digunakan dalam penelitian ini dengan tetap menjaga kerahasiaan dan anonimitas?
Andreas Brian Aditya	25	Pria	brian_aditya	xxx	snowbrake	> 3 jam	4	Ya
Andreas Brian Aditya	25	Pria	brian_aditya	xxx	NFS heat	< 1 jam	3	Ya
IpTul	21	Pria	IpTul	Fishing Planet	321 jam	5	Ya	Ya
IpTul	21	Pria	IpTul	fifa	141 jam	5	Ya	Ya

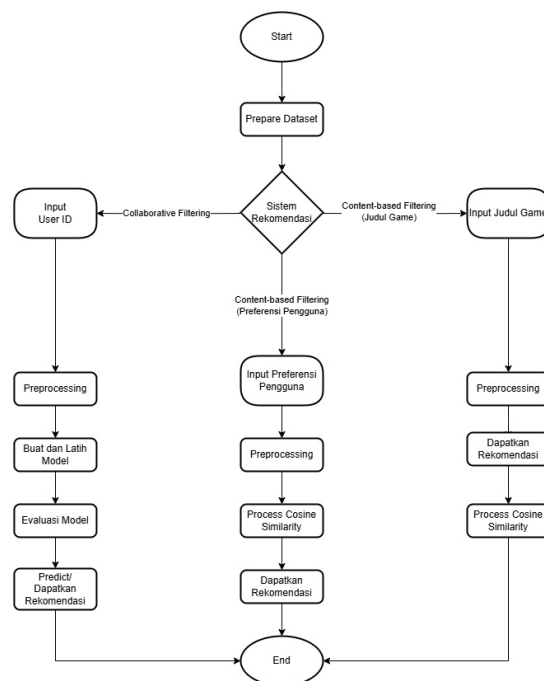
Desain Sistem

Tahap perancangan sistem merupakan bagian krusial sebelum masuk ke proses pengembangan aplikasi. Pada tahap ini, struktur dan komponen utama dari aplikasi mulai ditentukan guna memastikan bahwa sistem yang dibangun nantinya mampu menjawab kebutuhan serta mencapai tujuan yang telah dirumuskan sebelumnya.

1. Diagram Alir

Perancangan sistem dimulai dengan menyusun diagram alur aplikasi sebagai dasar visualisasi proses kerja sistem yang akan dibangun.

Gambar 2. Diagram Alir Sistem Rekomendasi

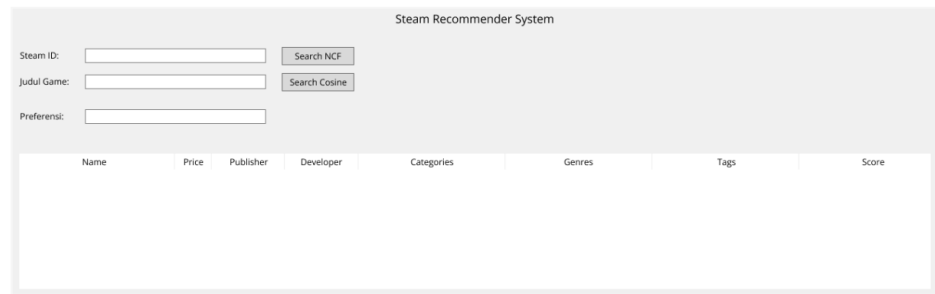


2. Desain Antarmuka

Antarmuka utama aplikasi dirancang sebagai halaman input yang memungkinkan pengguna memasukkan *SteamID*, judul *game*, serta preferensi pribadi mereka. Halaman ini menjadi titik awal interaksi antara pengguna dan sistem rekomendasi.

Gambar 3. Desain Antarmuka Halaman Utama

IMPLEMENTASI SISTEM REKOMENDASI *GAME* MENGGUNAKAN HYBRID FILTERING PADA PLATFROM DIGITAL STEAM



Implementasi

Bagian ini menguraikan tahapan implementasi dalam membangun model *machine learning* beserta aplikasi *desktop* sebagai antarmuka pengguna. Proses implementasi mencakup beberapa langkah penting, dimulai dari menyiapkan data yang relevan, melakukan *pre-processing* untuk membersihkan dan menyelaraskan data, membangun model sistem rekomendasi, hingga merancang dan mengembangkan aplikasi *desktop* sebagai media interaksi bagi pengguna.

1. Menyiapkan Data

Langkah awal dalam membangun model *machine learning* adalah menyiapkan data pengguna dan game. Data diimpor dalam format CSV menggunakan *Pandas*, lalu dipilih dan disesuaikan dengan kebutuhan sistem. Beberapa kolom diubah, seperti "judul *game*" menjadi *AppID*, "waktu bermain" menjadi *Playtime* dalam bentuk *float*, serta "kepuasan" dan "rekomendasi" pengguna diubah menjadi *Rating* dan *IsRecommended*. Username juga diubah menjadi *UserID* acak.

2. *Preprocessing*

Tahap *pre-processing* dilakukan untuk menyiapkan data sesuai kebutuhan model *hybrid filtering*, yaitu *Neural Collaborative Filtering* (NCF) dan *Cosine Similarity*.

a) *Neural Collaborative Filtering*

- i. *Label Encoding*: Mengubah kolom *UserID* dan *AppID* menjadi angka untuk representasi numerik.
- ii. Menyiapkan Fitur dan Label: Data dipisah menjadi fitur (*UserID*, *AppID*) dan label (*Rating*, *IsRecommended*) untuk pelatihan model.

- iii. *Split Data*: Data dibagi menjadi 80% untuk *training* dan 20% untuk *testing* agar model dapat dievaluasi secara adil.

b) *Cosine Similarity*

- i. Gabung Fitur: *Genre*, *tag*, *developer*, dan *publisher* digabung dalam satu kolom deskripsi.
- ii. TF-IDF *Vectorizer*: Kolom gabungan dikonversi menjadi matriks vektor untuk menghitung kemiripan antar *game*.

3. Membuat Model *Hybrid Filtering*

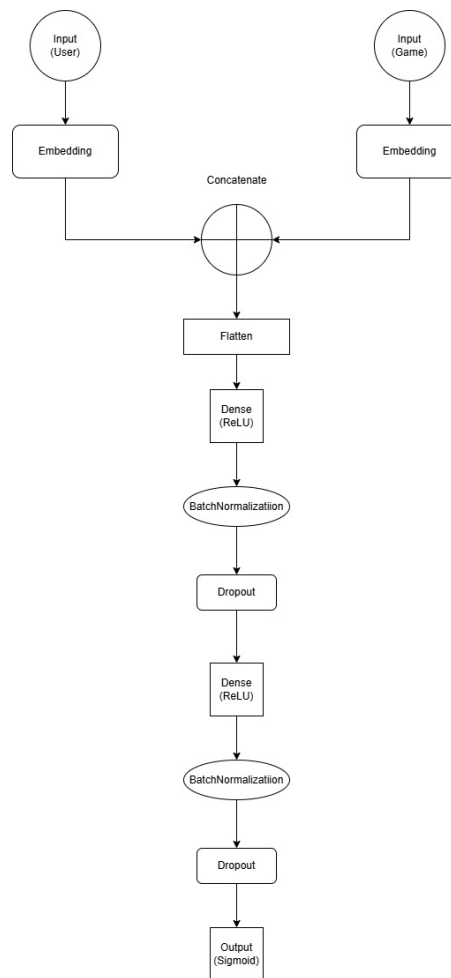
Tahap *pre-processing* dilakukan untuk menyiapkan data sesuai kebutuhan model *hybrid filtering*, yaitu *Neural Collaborative Filtering* (NCF) dan *Cosine Similarity*.

a) *Neural Collaborative Filtering*

Model dibangun menggunakan neural network yang mempelajari interaksi antara pengguna dan *game* melalui representasi vektor (*embedding*).

Gambar 4. Arsitektur Neural Collaborative Filtering

IMPLEMENTASI SISTEM REKOMENDASI *GAME* MENGUNAKAN HYBRID FILTERING PADA PLATFROM DIGITAL STEAM



Setelah arsitektur ditentukan, model dilatih dengan data interaksi yang telah diproses, menggunakan fungsi *loss* MSE dan *optimizer Adam*. Evaluasi dilakukan menggunakan RMSE, dan proses pelatihan dikendalikan dengan mekanisme *EarlyStopping* agar tidak *overfitting*.

b) *Cosine Similarity*

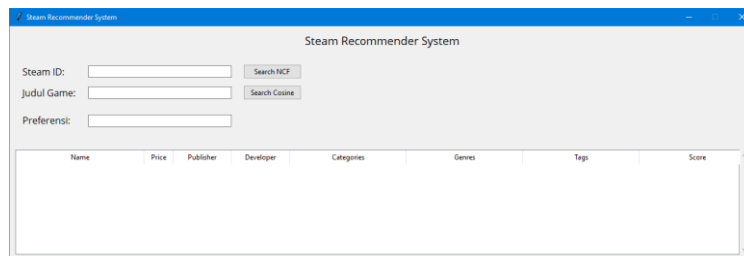
Menggunakan cosine similarity untuk menghitung tingkat kemiripan antar *game* berdasarkan fitur seperti *genre*, *tag*, dan *developer*. Data fitur digabung dan dikonversi menjadi vektor, kemudian dicari *game* yang paling mirip dengan preferensi pengguna.

4. Membuat Model *Hybrid Filtering*

Setelah model selesai, antarmuka aplikasi dibuat menggunakan *Tkinter* dan *Python* di *Visual Studio Code*. Aplikasi ini memiliki satu

halaman utama untuk menampilkan rekomendasi berdasarkan preferensi pengguna. Fitur *Steam ID* dan *Search NCF* digunakan untuk *collaborative filtering*, sedangkan judul *game*, preferensi, dan *Search Cosine* digunakan untuk *content-based filtering*.

Gambar 5. Tampilan Aplikasi Sistem Rekomendasi



Pengujian

Tahap akhir penelitian ini melibatkan pengujian aplikasi dan evaluasi performa sistem rekomendasi berbasis *hybrid filtering*, yang menggabungkan *Neural Collaborative Filtering* (NCF) dan *cosine similarity*. Evaluasi dilakukan dengan mengukur *loss* dan RMSE untuk NCF, serta skor kemiripan pada model berbasis konten. Pengujian fungsional aplikasi dilakukan menggunakan metode *black-box* untuk memastikan semua fitur bekerja sesuai harapan.

1. Evaluasi NCF

Model NCF dievaluasi dengan metrik *loss* dan RMSE. Selama pelatihan, nilai *loss* dan RMSE mengalami penurunan signifikan yang menunjukkan bahwa model mampu mempelajari pola data dengan baik.

Gambar 6. Hasil Dari Pelatihan Model NCF Epoch 1-15

Epoch 1/25	
81/81	1s 14ms/step - loss: 0.1822 - rmse: 0.2023 - val_loss: 0.2460 - val_rmse: 0.2921
Epoch 2/25	
81/81	1s 13ms/step - loss: 0.1655 - rmse: 0.1643 - val_loss: 0.2391 - val_rmse: 0.2880
Epoch 3/25	
81/81	1s 13ms/step - loss: 0.1531 - rmse: 0.1370 - val_loss: 0.2314 - val_rmse: 0.2758
Epoch 4/25	
81/81	1s 13ms/step - loss: 0.1475 - rmse: 0.1450 - val_loss: 0.2248 - val_rmse: 0.2723
Epoch 5/25	
81/81	2s 17ms/step - loss: 0.1492 - rmse: 0.1614 - val_loss: 0.2298 - val_rmse: 0.2858
Epoch 6/25	
81/81	1s 8ms/step - loss: 0.1439 - rmse: 0.1451 - val_loss: 0.2160 - val_rmse: 0.2675
Epoch 7/25	
81/81	1s 6ms/step - loss: 0.1324 - rmse: 0.1340 - val_loss: 0.2359 - val_rmse: 0.3046
Epoch 8/25	
81/81	1s 7ms/step - loss: 0.1180 - rmse: 0.1035 - val_loss: 0.2120 - val_rmse: 0.2815
Epoch 9/25	
81/81	1s 3ms/step - loss: 0.1151 - rmse: 0.1129 - val_loss: 0.2142 - val_rmse: 0.2878
Epoch 10/25	
81/81	1s 3ms/step - loss: 0.1103 - rmse: 0.1016 - val_loss: 0.2175 - val_rmse: 0.2970
Epoch 11/25	
81/81	0s 3ms/step - loss: 0.1033 - rmse: 0.0982 - val_loss: 0.2046 - val_rmse: 0.2817
Epoch 12/25	
81/81	0s 3ms/step - loss: 0.0982 - rmse: 0.0896 - val_loss: 0.1980 - val_rmse: 0.2708
Epoch 13/25	
81/81	1s 3ms/step - loss: 0.0973 - rmse: 0.0954 - val_loss: 0.1919 - val_rmse: 0.2670
Epoch 14/25	
81/81	0s 3ms/step - loss: 0.0913 - rmse: 0.0935 - val_loss: 0.1991 - val_rmse: 0.2924
Epoch 15/25	
81/81	0s 3ms/step - loss: 0.0882 - rmse: 0.0767 - val_loss: 0.1924 - val_rmse: 0.2823

IMPLEMENTASI SISTEM REKOMENDASI *GAME* MENGGUNAKAN HYBRID FILTERING PADA PLATFROM DIGITAL STEAM

Nilai RMSE *training* dan validasi sempat fluktuatif setelah beberapa *epoch*, menandakan potensi *overfitting*, namun model terbaik berhasil diperoleh pada *epoch* ke-24. Pengujian menggunakan user ID tertentu menunjukkan model mampu memberikan rekomendasi *game* yang sesuai dengan preferensi pengguna.

Gambar 7. Hasil Dari Pelatihan Model NCF Epoch 16-25



Pengujian dilakukan menggunakan *user_id* = 67, dan sistem berhasil merekomendasikan *game* dengan skor relevansi tinggi, seperti, *Titanfall 2* dengan skor 0.99988, *Rainbow Six Siege* dengan skor 0.99987, *A Space From Unbound* dengan skor 0.99982.

2. Pengujian *Cosine Similarity*

Pengujian dilakukan sebanyak empat kali dengan input berbeda, untuk mengukur kemiripan konten berdasarkan preferensi pengguna. Berikut ringkasan hasilnya:

Tabel 3. Hasil Rekomendasi *Cosine Similarity*

Input	Game dengan Skor Tertinggi	Skor
Black Myth Wukong	God of War	0.41
Coral Island	My Time at Sandrock	0.488
RPG Square Enix	Kingdom Hearts	1.0
World War Single Player Strategy	The Great War: Western Front	1.0

3. Pengujian Fungsional Aplikasi

Untuk pengujian aplikasi, digunakan metode *black-box* guna memastikan bahwa fitur-fitur utama berjalan sesuai dengan fungsi yang diharapkan, seperti *input* menggunakan *Steam* ID untuk NCF, serta input berupa judul dan preferensi *game* untuk *cosine similarity*. Semua fungsi

berhasil menampilkan daftar rekomendasi berisi 10 *game* sesuai metode yang digunakan.

KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa sistem rekomendasi *game* berbasis *hybrid filtering* yang menggabungkan pendekatan *Neural Collaborative Filtering* (NCF) dan *cosine similarity* berhasil dibangun dan diuji secara fungsional melalui aplikasi antarmuka berbasis *Tkinter*. Hasil pengujian menunjukkan bahwa aplikasi mampu memberikan rekomendasi *game* yang sesuai dengan masukan pengguna, baik berdasarkan Steam ID, judul *game*, maupun preferensi konten seperti *genre*, *publisher*, dan kategori. Model NCF menunjukkan performa cukup baik dengan nilai *loss* dan RMSE yang menurun selama proses pelatihan, sementara model *cosine similarity* mampu memberikan rekomendasi konten yang relevan berdasarkan skor kemiripan.

Saran

Meskipun demikian, terdapat beberapa keterbatasan dalam penelitian ini, seperti jumlah data interaksi pengguna yang masih terbatas dan potensi *overfitting* pada model NCF setelah beberapa *epoch* pelatihan. Selain itu, sistem belum dilengkapi dengan validasi langsung dari pengguna akhir terkait kualitas hasil rekomendasi yang diberikan. Oleh karena itu, untuk penelitian selanjutnya disarankan agar data interaksi pengguna lebih diperkaya dan dilakukan evaluasi melalui *feedback* langsung pengguna. Pengembangan lebih lanjut juga dapat mencakup peningkatan antarmuka pengguna ke dalam platform *web* atau *mobile*, serta eksplorasi metode *hybrid* lain yang lebih kompleks untuk meningkatkan akurasi dan relevansi rekomendasi.

DAFTAR REFERENSI

Al Rasyid, R., & Ningsih, D. H. U. (2024). Penerapan Algoritma TF-IDF dan Cosine Similarity untuk Query Pencarian Pada Dataset Destinasi Wisata. *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, 8(1), 170–178. <https://doi.org/10.35870/jtik.v8i1.1416>

IMPLEMENTASI SISTEM REKOMENDASI *GAME* MENGUNAKAN HYBRID FILTERING PADA PLATFROM DIGITAL STEAM

- Alfarizi, M. R. sirfatullah, Al-farish, M. Z., Taufiqurrahman, M., Ardiansah, G., & Muhamad Elgar. (2023). PENGGUNAAN PYTHON SEBAGAI BAHASA PEMROGRAMAN UNTUK MACHINE LEARNING DAN DEEP LEARNING. *Karimah Tauhid*, 2(1), 1–6.
- Amanda Muchsin Chalik, Bilal Abdul Qowy, Faiz Hanafi, & Ahlijati Nuraminah. (2021). Mouse Tracking Tangan dengan Klasifikasi Gestur Menggunakan OpenCV dan Mediapipe. *Jurnal Ilmiah Teknik Informatika dan Komunikasi*, 1(2), 10–18. <https://doi.org/10.55606/juitik.v1i2.323>
- Amansyah, I., Indra, J., Nurlaelasari, E., & Juwita, A. R. (2024). Prediksi Penjualan Kendaraan Menggunakan RegresiLinear: Studi Kasus pada Industri Otomotif di Indonesia. *INNOVATIVE: Journal Of Social Science Research*, 4(4), 1199–1216. <https://doi.org/10.31004/innovative.v4i4.12735>
- Budhi Gustiandi. (2023). *Langkah Awal Menguasai Bahasa Pemrograman Python*. Penerbit BRIN. <https://doi.org/10.55981/brin.656>
- Darmawan, R., & Geni, B. Y. (2023). Perancangan dan Pengembangan Sistem Informasi Monitoring Sewa ATM Berbasis Web Menggunakan Metode SDLC. *Journal of Information System Research (JOSH)*, 4(4), 1109–1117. <https://doi.org/10.47065/josh.v4i4.3808>
- Fajriansyah, M., Adikara, P. P., & Widodo, A. W. (2021). Sistem Rekomendasi Film Menggunakan Content Based Filtering. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 5(6), Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer.
- Februariyanti, H., Laksono, A. D., Wibowo, J. S., & Utomo, M. S. (2021). IMPLEMENTASI METODE COLLABORATIVE FILTERING UNTUK SISTEM REKOMENDASI PENJUALAN PADA TOKO MEDEL. *Jurnal Khatulistiwa Informatika*, 9(1), 43–50. <https://doi.org/10.31294/jki.v9i1.9859>
- Mukti, K. T., & Mardhiyah, I. (2022). SISTEM REKOMENDASI PEMBELIAN LISENSI FILM MENGGUNAKAN PENDEKATAN HYBRID FILTERING (STUDI KASUS: FILM ANIMASI JEPANG). *JURSISTEKNI (Jurnal Sistem Informasi dan Teknologi Informasi)*, 4(3), 126–139. <https://doi.org/10.52005/jursistekni.v4i3.116>

- Pangestu, A., Arifin, Y. T., & Safitri, R. A. (2023). ANALISIS SENTIMEN REVIEW PUBLIK PENGGUNA GAME ONLINE PADA PLATFORM STEAM MENGGUNAKAN ALGORITMA NAÏVE BAYES. *Jurnal Mahasiswa Teknik Informatika*, 7(6), 3096–3113. <https://doi.org/10.36040/jati.v7i6.8829>
- Putra, T. I. Z. M., Suprpto, S., & Bukhori, A. F. (2022). Model Klasifikasi Berbasis Multiclass Classification dengan Kombinasi Indobert Embedding dan Long Short-Term Memory untuk Tweet Berbahasa Indonesia. *Jurnal Ilmu Siber dan Teknologi Digital*, 1(1), 1–28. <https://doi.org/10.35912/jisted.v1i1.1509>
- Warnilah, A. I., Sutisna, H., Mulyana, A. J.-, Nuraeni, F. S.-, & Widiyanto, T. A.-. (2022). Program Aplikasi Pendeteksi Masker Dengan Menggunakan Algoritma Haarcascade. *EVOLUSI: Jurnal Sains dan Manajemen*, 10(1). <https://doi.org/10.31294/evolusi.v10i1.12583>
- Witanto, K. S., Sanjaya Er, N. A., Karyawati, A. E., Kadyanan, I. G. A. G. A., Suhartana, I. K. G., & Astuti, L. G. (2022). Implementasi LSTM Pada Analisis Sentimen Review Film Menggunakan Adam Dan RMSprop Optimizer. *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, 10(4), 351. <https://doi.org/10.24843/JLK.2022.v10.i04.p05>