

**LAPORAN PROJECT UAS**  
**“GAME RECOMMENDATIONS ON STEAM”**

**MATA KULIAH**  
**MACHINE LEARNING (PRAKTIKUM)**



**Disusun Oleh :**  
**MUHAMAD GHANDI NUR SETIAWAN | [43422101]**  
**PUTRI EGI HAPSA ANDARISTA | [434221015]**

**DIPLOMA TEKNIK INFORMATIKA**  
**FAKULTAS VOKASI**  
**UNIVERSITAS AIRLANGGA**  
**SURABAYA**  
**2024**

## DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>1</b>
<b>BAB I</b>	
<b>PENDAHULUAN.....</b>	<b>2</b>
1.1. LATAR BELAKANG.....	2
1.2. RUMUSAN MASALAH.....	3
1.3. TUJUAN.....	3
1.4. MANFAAT.....	3
<b>BAB II</b>	
<b>TINJAUAN PUSTAKA.....</b>	<b>4</b>
2.1. PENJELASAN DATASET.....	4
2.2. METODOLOGI.....	5
<b>BAB III</b>	
<b>PENYELESAIAN DAN PEMBAHASAN.....</b>	<b>7</b>
3.1. IMPLEMENTASI.....	7
3.2. PENGUJIAN.....	15
3.3. KESIMPULAN.....	22
<b>DAFTAR PUSTAKA.....</b>	<b>23</b>

# BAB I

## PENDAHULUAN

### 1.1. LATAR BELAKANG

Industri video game telah mengalami pertumbuhan yang signifikan dalam beberapa tahun terakhir, didorong oleh perkembangan teknologi digital dan adopsi luas platform distribusi online. Sebagai salah satu bentuk hiburan interaktif, video game kini menjadi bagian integral dari gaya hidup masyarakat modern, dengan pendapatan global mencapai \$91 miliar pada tahun 2016 dan diproyeksikan terus meningkat sebesar 6,3% setiap tahun hingga 2020 (Lin et al., 2019). Salah satu pendorong utama pertumbuhan ini adalah munculnya platform distribusi digital seperti Steam, yang memungkinkan pengguna untuk membeli, mengunduh, dan memainkan ribuan game tanpa batasan fisik.

Steam, sebagai platform distribusi digital terkemuka, menawarkan lebih dari 18.000 game kepada lebih dari 184 juta pengguna aktif (Lin et al., 2019; Lomanto et al., 2023). Namun, dengan katalog game yang sangat besar, pengguna sering kali menghadapi tantangan untuk menemukan game yang sesuai dengan preferensi dan kebutuhan mereka. Hal ini menciptakan fenomena yang dikenal sebagai "information overload," di mana pengguna kesulitan memilih informasi yang relevan di tengah banyaknya pilihan yang tersedia (Li et al., 2021). Dalam konteks ini, sistem rekomendasi menjadi solusi yang sangat diperlukan untuk mempermudah pengguna menemukan game yang relevan berdasarkan preferensi mereka.

Sistem rekomendasi adalah alat yang dirancang untuk memprediksi preferensi pengguna dan memberikan saran personal berdasarkan data historis. Secara umum, terdapat tiga pendekatan utama dalam sistem rekomendasi, yaitu berbasis konten (content-based), berbasis kolaborasi (collaborative filtering), dan pendekatan hybrid. Dari hasil penelitian (Lomanto et al., 2023), Collaborative Filtering (CF) adalah salah satu metode yang paling banyak digunakan karena kemampuannya untuk memanfaatkan data pengguna lain sebagai referensi dalam memberikan rekomendasi.

Penelitian ini menggunakan pendekatan Collaborative Filtering berbasis Singular Value Decomposition (SVD), sebuah teknik reduksi dimensi yang telah terbukti efektif dalam menangani masalah sparsitas data dan cold start yang sering muncul pada sistem rekomendasi berbasis kolaborasi. SVD bekerja dengan struktur matriks besar, seperti matriks rating pengguna dan item, menjadi representasi yang lebih kecil dan mudah diolah, sehingga memungkinkan perhitungan yang lebih efisien dan akurat (Lomanto et al., 2023). Teknik ini tidak hanya meninjau model, tetapi juga dapat memberikan rekomendasi yang relevan dan personal bagi pengguna platform seperti Steam.

Selain itu, penelitian terbaru menunjukkan bahwa mengubah umpan balik implisit, seperti waktu bermain dan aktivitas pengguna, menjadi peringkat eksplisit dapat meningkatkan akurasi sistem rekomendasi. Studi oleh Bunga et al. (2023) menggudari platform Steam untuk membangun sistem rekomendasi berbasis

Collaborative Filtering yang mampu mengubah data implisit menjadi eksplisit. Metode ini memungkinkan penggunaan informasi yang biasanya tersembunyi, seperti kebiasaan bermain game, untuk menciptakan rekomendasi yang lebih akurat.

Penelitian sebelumnya oleh Lin et al. (2018) menunjukkan bahwa ulasan pengguna dapat memberikan wawasan mendalam mengenai preferensi dan pengalaman bermain pengguna. Studi mereka mencatat bahwa mayoritas ulasan baik ulasan positif atau pun negatif tetep berfokus pada elemen desain game, seperti mekanika permainan dan alur cerita, yang merupakan informasi penting dalam membangun sistem rekomendasi yang lebih personal dan relevan.

Oleh karena itu, penelitian ini bertujuan untuk membangun sistem rekomendasi game yang memanfaatkan algoritma Collaborative Filtering berbasis SVD. Sistem ini tidak hanya dirancang untuk meningkatkan pengalaman pengguna, tetapi juga memberikan manfaat strategis bagi pengembang platform. Sebagai contoh, sistem rekomendasi yang efektif dapat meningkatkan tingkat keterlibatan pengguna, memperpanjang waktu bermain, dan mendorong pembelian game baru, sehingga secara keseluruhan berkontribusi pada peningkatan kepuasan pengguna dan pendapatan platform.

## **1.2. RUMUSAN MASALAH**

Bagaimana merancang dan mengimplementasikan sistem rekomendasi game berbasis Collaborative Filtering dengan menggunakan algoritma Singular Value Decomposition (SVD) yang dapat memberikan rekomendasi personal kepada pengguna berdasarkan histori ulasan dan preferensi mereka?

## **1.3. TUJUAN**

Tujuan dari penelitian ini adalah untuk mengembangkan sistem rekomendasi game berbasis Collaborative Filtering menggunakan algoritma Singular Value Decomposition (SVD) yang mampu memberikan rekomendasi personal berdasarkan data historis ulasan dan preferensi pengguna di platform Steam.

## **1.4. MANFAAT**

Sistem rekomendasi yang dikembangkan dari penelitian ini diharapkan memberikan beberapa manfaat, baik untuk pengguna maupun pengembang platform distribusi digital seperti Steam. Secara umum, manfaat dari penelitian ini meliputi:

1. Membantu menemukan game yang sesuai dengan preferensi individu secara lebih cepat dan efisien.
2. Meningkatkan kepuasan pengguna melalui rekomendasi game yang didasarkan pada histori ulasan dan preferensi individu.
3. Memungkinkan pengembang untuk mengenali tren atau preferensi pengguna tertentu berdasarkan data historis.
4. Mengurangi potensi game oversupply dengan memberikan visibilitas pada game-game yang sesuai dengan minat pengguna tertentu.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. PENJELASAN DATASET

Dataset yang digunakan dalam penelitian ini adalah dataset steam game review yang diambil dari Kaggle yang berasal dari (steam game review, 7 December 2024), yang terdiri dari dua file CSV utama, yaitu games.csv dan recommendations.csv. Dataset ini menyediakan informasi terkait dengan game yang tersedia di platform Steam serta ulasan dan rekomendasi dari pengguna.

##### 1. games.csv

app_id	title	release_date	developer	publisher	genres	positive_ratings	negative_ratings	price
514520	Sparky's Hunt	2016-08-18	Luke Crripps, Fellowplayer	Indie	6, 3	0, 99		
1012710	Endzeit	2019-04-03	rockyDev, RockyDev	Action; Early Access	0, 1, 7	1, 19		
279260	Richard & Alice	2014-06-05	Owl Cave	Owl Cave	Adventure; Indie	264, 99	4, 79	
220090	The Journey Down: Chapter One	2013-01-09	SkyGoblin	SkyGoblin	Adventure; Indie	902, 129	5, 99	
788870	In The Long Run The Game	2018-07-02	Zerstören Games	Zerstören Games	Action; Adventure; Indie; Simulation; Strategy; Early Access	6, 11	7, 19	
920780	Safe	2018-11-02	GraphixGames	Indie; Strategy	1, 0	2, 99		
625780	Magic Wand	2017-06-25	the catamites	harmony corp	Adventure; Indie; RPG	34, 2	2, 79	
339440	Critical Annihilation	2016-02-12	Devoga	Devoga	Action; Indie; Early Access	178, 16	4, 79	
448240	Hybrid Animals	2016-08-29	FamishedMammal	Abstract Software Inc.	Action; Casual; Indie; Simulation	196, 134	4, 99	
838110	HyperParasite	2019-04-03	Troglobytes Games	Troglobytes Games	Hound Picked Games	Action; Adventure; Indie; Early Access	30, 1	6, 19
444250	Crown and Council	2016-04-22	Mojang	Mojang	Casual; Indie; Strategy	450, 268	0, 0	
752550	Ripped Pants at Work	2018-01-21	Scott Ethington	Scott Ethington	Action; Indie	20, 4	2, 09	
848720	Z.W.L.	2018-05-15	S. Khasanov	S. Khasanov	Action; Indie	3, 1	0, 79	
925470	Endless Jade Sea - Midori no Umi	2019-01-23	Cabbit	SakuraGame	Adventure; Indie; RPG	92, 16	1, 69	

Dataset games.csv berisi informasi terkait setiap game yang ada di platform Steam. Kolom-kolom yang terdapat dalam dataset ini antara lain:

- app\_id: Identifier unik untuk setiap game di Steam.
- title: Nama game yang tercatat di Steam.
- release\_date: Tanggal rilis game.
- developer: Pengembang yang bertanggung jawab atas pembuatan game.
- publisher: Penerbit yang mendistribusikan game tersebut.
- genres: Genre atau kategori game (misalnya, aksi, petualangan, strategi, dll.).
- positive\_ratings: Jumlah ulasan positif yang diterima oleh game tersebut.
- negative\_ratings: Jumlah ulasan negatif yang diterima oleh game.
- price: Harga game pada platform Steam.

Data ini memberikan gambaran mengenai karakteristik game yang ada di Steam, yang nantinya akan digunakan dalam sistem rekomendasi untuk mencocokkan game dengan preferensi pengguna.

##### 2. recommendations.csv

app_id	date	is_recommended	user_id	review_id
498240	2020-11-02	True	551574	17202162
359550	2021-12-07	False	5800984	1331080
570940	2022-07-31	True	9401278	17199753
230410	2017-01-25	True	8424375	15065522
235460	2022-07-23	True	9145094	26539023
49520	2022-04-09	True	1425172	12626780
289070	2022-02-09	True	9001558	21188514
4000	2020-02-12	True	7949698	5418356
438100	2022-06-09	True	947336	4963495
359550	2017-12-22	True	8086911	23112215
289070	2020-06-23	True	6269647	2159213
304390	2022-07-13	True	9211265	2134032
457140	2019-10-07	True	9963299	9607518
271590	2019-10-27	True	9515898	4759335

Dataset recommendations.csv berisi informasi tentang rekomendasi dan ulasan yang diberikan oleh pengguna Steam. Kolom-kolom utama dalam dataset ini adalah:

- app\_id: Identifier unik untuk game yang direkomendasikan atau diulas.
- rating: Tanggal ketika pengguna memberikan rating atau ulasan.
- is\_recommended: Indikator biner yang menunjukkan apakah game tersebut direkomendasikan oleh pengguna (True) atau tidak (False).
- user\_id: Identifier unik untuk pengguna yang memberikan rekomendasi atau ulasan.
- review\_id: Identifier unik untuk setiap ulasan atau penilaian yang diberikan oleh pengguna.

Dataset ini menyediakan data perilaku pengguna, yaitu apakah mereka menyukai atau tidak menyukai game yang mereka mainkan, yang digunakan untuk membangun model rekomendasi berdasarkan Collaborative Filtering.

## 2.2. METODOLOGI

Metodologi yang digunakan dalam penelitian ini berfokus pada penerapan algoritma Collaborative Filtering berbasis Singular Value Decomposition (SVD) untuk membangun sistem rekomendasi game. Berikut adalah langkah-langkah metodologi yang digunakan, yang didasarkan pada kode yang telah dikembangkan dan pendekatan yang terdapat dalam literatur terkait.

### 1. Pengumpulan Data

Data yang digunakan dalam penelitian ini berasal dari dua file CSV utama yang telah dijelaskan sebelumnya. Data ini mencakup informasi tentang game yang ada di Steam, ulasan dan rekomendasi yang diberikan oleh pengguna, serta informasi perilaku pengguna yang berharga untuk membangun sistem rekomendasi yang personal.

### 2. Pra-pemrosesan Data

Sebelum digunakan dalam model rekomendasi, data perlu diproses untuk mengurangi penggunaan memori dan menyiapkan data agar siap untuk analisis lebih lanjut. Beberapa tahapan utama dalam pra-pemrosesan data adalah:

- Pengurangan memori: Mengubah tipe data pada kolom-kolom tertentu (misalnya, mengonversi tipe data float64 ke float32 dan int64 ke int32) untuk menghemat ruang memori dan meningkatkan efisiensi pemrosesan data.
- Ekstraksi dan pengolahan informasi genre: Informasi genre pada dataset games.csv dipecah dan dihitung distribusinya untuk memahami dominasi genre dalam dataset.
- Penyusunan matriks rating: Menggunakan data ulasan dan rekomendasi dari recommendations.csv untuk membangun matriks preferensi pengguna terhadap game (berdasarkan apakah game tersebut direkomendasikan atau tidak).

### 3. Penerapan Algoritma Collaborative Filtering berbasis SVD

Pada tahap ini, dilakukan penerapan Singular Value Decomposition (SVD) untuk memfaktorkan matriks preferensi pengguna dan game. SVD adalah teknik reduksi dimensi yang mengurangi kompleksitas perhitungan dalam matriks besar dan meningkatkan akurasi dalam prediksi rekomendasi. Beberapa tahapan yang dilakukan adalah:

- Matrix Factorization dengan SVD: Matriks besar yang berisi data ulasan pengguna (apakah direkomendasikan atau tidak) diubah menjadi representasi yang lebih kecil menggunakan SVD.
- Rekomendasi game berdasarkan kesamaan pengguna: Menggunakan representasi hasil faktorisasi untuk menghitung kesamaan antara pengguna satu dengan pengguna lainnya dan merekomendasikan game berdasarkan kesamaan tersebut.

## BAB III

### PENYELESAIAN DAN PEMBAHASAN

#### 3.1. IMPLEMENTASI

- **Code app.py**

```
from flask import Flask, render_template, request  
from sistem_rekomendasi import get_recommendations_for_user  
import pandas as pd
```

- Import Modul

Kode ini mengimpor beberapa modul penting:

- Flask: Framework untuk membangun aplikasi web.
- render\_template: Untuk menghubungkan file HTML dengan aplikasi.
- request: Untuk mengambil data dari form yang diisi pengguna.
- pandas (pd): Untuk membaca dan mengolah file CSV.
- get\_recommendations\_for\_user: Fungsi dari file eksternal (sistem\_rekomendasi.py) yang digunakan untuk mendapatkan rekomendasi.

```
app = Flask(__name__)
```

- Membuat Objek Flask, membuat objek aplikasi Flask. Semua route (alamat halaman web) akan diatur melalui objek ini.

```
def reduce_memory(df):  
    for col in df.columns:  
        if df[col].dtype == 'float64':  
            df[col] = df[col].astype('float32')  
        if df[col].dtype == 'int64':  
            df[col] = df[col].astype('int32')  
    return df
```

- Fungsi reduce\_memory

Fungsi ini digunakan untuk mengurangi ukuran memori yang digunakan oleh DataFrame (recommendations\_df):

- float64 diubah menjadi float32.
- int64 diubah menjadi int32.

Tujuannya adalah agar file CSV yang berukuran besar bisa diolah dengan lebih efisien.

```
recommendations_df = reduce_memory(pd.read_csv('recommendations.csv'))
```

- Membaca Data Rekomendasi

- File recommendations.csv dibaca menggunakan pandas dan disimpan ke dalam variabel recommendations\_df.
- Fungsi reduce\_memory dipanggil untuk mengoptimalkan ukuran data.

```
@app.route('/')  
def index():  
    return render_template('index.html')
```

- Route Halaman Utama, route ini menentukan bahwa jika pengguna membuka halaman utama (/), aplikasi akan menampilkan file HTML bernama index.html.

```
@app.route('/form')
def form():
    # Mengambil ID user yang memiliki lebih dari 7 rekomendasi
    user_counts = recommendations_df['user_id'].value_counts()
    recommended_user_ids = user_counts[user_counts >= 7].index.tolist() # ID user yang memenuhi kriteria

    return render_template('form.html', recommended_user_ids=recommended_user_ids)
```

- Route Halaman Form
  - Route ini digunakan untuk menampilkan halaman form.
  - Fungsi ini menghitung jumlah rekomendasi untuk setiap user dari kolom user\_id di file CSV.
  - User yang memiliki minimal 7 rekomendasi akan disaring, dan ID mereka akan dikirim ke file HTML (form.html) sebagai daftar pilihan.

```
@app.route('/result', methods=['POST'])
def result():
    # Ambil ID user dari form
    user_id = int(request.form.get('user_id'))

    # Ambil rekomendasi game dari fungsi backend
    recommendations = get_recommendations_for_user(user_id, n_neighbors=6)

    # Render hasil ke halaman result.html
    return render_template('result.html', user_id=user_id, recommendations=recommendations)
```

- Route Halaman Hasil
  - Route ini menangani permintaan POST dari form.
  - ID user yang dipilih oleh pengguna diambil dari form dan diubah ke tipe int.
  - Fungsi get\_recommendations\_for\_user dipanggil untuk mendapatkan rekomendasi berdasarkan ID user tersebut.
  - Parameter n\_neighbors=6 menentukan jumlah rekomendasi (6 rekomendasi).
  - Hasil rekomendasi dikirim ke file HTML (result.html) untuk ditampilkan.

```
if __name__ == '__main__':
    app.run(debug=True)
```

- Menjalankan Aplikasi
  - Kode ini memastikan bahwa aplikasi berjalan dalam mode debug (untuk memudahkan pengembangan).
  - Aplikasi akan dijalankan di server lokal.

- **Code sistem\_rekomendasi.py**

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.decomposition import TruncatedSVD
from sklearn.metrics.pairwise import cosine_similarity
from scipy.sparse import coo_matrix

```

- Import Library

Kode ini mengimpor berbagai library:

- pandas dan numpy: Untuk mengolah data.
- matplotlib dan seaborn: Untuk membuat visualisasi data.
- scikit-learn: Untuk metode SVD dan menghitung kemiripan antar pengguna.
- scipy: Untuk membuat matriks data dalam bentuk yang lebih efisien (sparse matrix).

```
# Fungsi untuk mengurangi penggunaan memori DataFrame
```

```

def reduce_memory(df):
    for col in df.columns:
        if df[col].dtype == 'float64':
            df[col] = df[col].astype('float32')
        if df[col].dtype == 'int64':
            df[col] = df[col].astype('int32')
    return df

```

- Fungsi reduce\_memory

Fungsi ini digunakan untuk mengurangi penggunaan memori dengan mengubah tipe data kolom dalam DataFrame:

- float64 diubah menjadi float32.
- int64 diubah menjadi int32.

```
# Memuat data
```

```

games_df = reduce_memory(pd.read_csv('games.csv')) # Data game
recommendations_df = reduce_memory(pd.read_csv('recommendations.csv')) # Data rekomendasi

```

- Membaca Data

- games\_df: Data tentang game, seperti judul, genre, harga, dll., diambil dari file games.csv.
- recommendations\_df: Data rekomendasi, yang mencakup ID pengguna dan informasi apakah game direkomendasikan, diambil dari file recommendations.csv.

```
# 1. Visualisasi Tren Rilis Game
```

```

games_df['release_year'] = pd.to_datetime(games_df['release_date']).dt.year # Mengambil tahun rilis
release_trend = games_df['release_year'].value_counts().sort_index()
plt.figure(figsize=(12, 6))
sns.lineplot(x=release_trend.index, y=release_trend.values, marker="o")
plt.title("Tren Rilis Game")
plt.xlabel("Tahun")
plt.ylabel("Jumlah Game yang Dirlis")
plt.grid()
plt.show()

```

- Analisis dan Visualisasi Data
  - Tren Rilis Game, menampilkan tren rilis game dari waktu ke waktu (jumlah game per tahun).

```
# 2. Visualisasi Distribusi Genre Game
genre_counts = games_df['genres'].str.split(';').explode().value_counts().head(10)
plt.figure(figsize=(12, 6))
sns.barplot(x=genre_counts.index, y=genre_counts.values, palette="viridis")
plt.title("Distribusi 10 Genre Teratas")
plt.xlabel("Genre")
plt.ylabel("Jumlah")
plt.xticks(rotation=45)
plt.show()
```

- Analisis dan Visualisasi Data
  - Distribusi Genre Game, membagi genre game (dipisahkan dengan ",") dan menampilkan 10 genre paling populer.

```
# 3. Visualisasi Distribusi Rasio Positif
games_df['positive_ratio'] = games_df['positive_ratings'] / (
    games_df['positive_ratings'] + games_df['negative_ratings']) # Rasio positif
plt.figure(figsize=(12, 6))
sns.histplot(games_df['positive_ratio'], bins=50, kde=True, color="blue")
plt.title("Distribusi Rasio Positif")
plt.xlabel("Rasio Positif (%)")
plt.ylabel("Jumlah")
plt.grid()
plt.show()
```

- Analisis dan Visualisasi Data
  - Distribusi Rasio Positif, menghitung rasio rating positif dari total rating dan memvisualisasikan distribusinya.

```
# 4. Visualisasi Distribusi Harga Game
plt.figure(figsize=(15, 6))
sns.histplot(data=games_df[games_df['price'] < 100], x='price', bins=50, color="blue")
plt.title("Distribusi Harga Game (< $100)")
plt.xlabel("Harga ($)")
plt.ylabel("Jumlah")
plt.grid()
plt.show()
```

- Analisis dan Visualisasi Data
  - Distribusi Harga Game, menampilkan distribusi harga game untuk game yang harganya di bawah \$100.

```
# Mapping setiap user dan item ke nilai numerik unik
user_ids = recommendations_df['user_id'].astype('category').cat.codes
item_ids = recommendations_df['app_id'].astype('category').cat.codes
```

- Mapping User dan Game, user\_ids dan item\_ids digunakan untuk mengubah ID pengguna dan ID game menjadi angka unik (kategori).

```
# Mendapatkan ID user dan game unik
unique_user_ids = recommendations_df['user_id'].astype('category').cat.categories
unique_item_ids = recommendations_df['app_id'].astype('category').cat.categories
```

- Daftar unik dari ID pengguna (user\_id) dan ID game (app\_id)
  - astype('category'): Mengubah kolom menjadi tipe kategori untuk efisiensi.
  - .cat.categories: Mengambil daftar nilai unik dari kategori tersebut.
  - unique\_user\_ids: Daftar ID pengguna unik.
  - unique\_item\_ids: Daftar ID game unik.

```
# Menggunakan 'is_recommended' sebagai preferensi (1 untuk direkomendasikan, 0 untuk tidak)
user_game_matrix = coo_matrix((recommendations_df['is_recommended'], (user_ids, item_ids)))
```

- Membuat Sparse Matrix, membuat matriks data pengguna dan game (sparse matrix) di mana nilai matriks adalah 1 jika game direkomendasikan untuk user.

```
# Melatih model Matrix Factorization menggunakan SVD (Singular Value Decomposition)
svd = TruncatedSVD(n_components=50)
user_matrix = svd.fit_transform(user_game_matrix)
item_matrix = svd.components_
```

- Matrix Factorization dengan SVD
  - user\_matrix: Representasi pengguna dalam dimensi yang lebih kecil.
  - item\_matrix: Representasi game dalam dimensi yang lebih kecil.

```
# Fungsi untuk mendapatkan user serupa menggunakan Matrix Factorization
def get_similar_users(user_id, user_matrix, n_neighbors=6):
    if user_id not in unique_user_ids:
        return []
    user_index = np.where(unique_user_ids == user_id)[0][0] # Index user
    user_vector = user_matrix[user_index].reshape(1, -1)
    cosine_similarities = cosine_similarity(user_vector, user_matrix)
    similar_users = cosine_similarities.argsort()[0][-n_neighbors-1:-1] # Top N user serupa
    return [unique_user_ids[i] for i in similar_users]
```

- Fungsi untuk Mendapatkan User Serupa
  - Fungsi ini mencari user serupa menggunakan kemiripan kosinus.
  - Mengembalikan N user paling mirip dengan user yang dimasukkan.

```
# Fungsi untuk mendapatkan game yang direkomendasikan berdasarkan user serupa
def recommend_games(user_id, n_neighbors=6):
    similar_users = get_similar_users(user_id, user_matrix, n_neighbors=n_neighbors)
    recommended_games = []
    for user in similar_users:
        user_games = recommendations_df[recommendations_df['user_id'] == user]['app_id'].unique()
        for game_id in user_games:
            game_info = games_df[games_df['app_id'] == game_id][['title', 'positive_ratings', 'genres', 'price']]
            recommended_games.extend(game_info.to_dict(orient='records'))
    return list({game['title']: game for game in recommended_games}.values()) # Menghapus duplikat berdasarkan judul game
```

- Fungsi untuk Mendapatkan Rekomendasi Game

- Mengambil user serupa.
- Mengumpulkan game yang direkomendasikan untuk user serupa.
- Menghapus game duplikat berdasarkan judul.

# Fungsi utama untuk digunakan di Flask

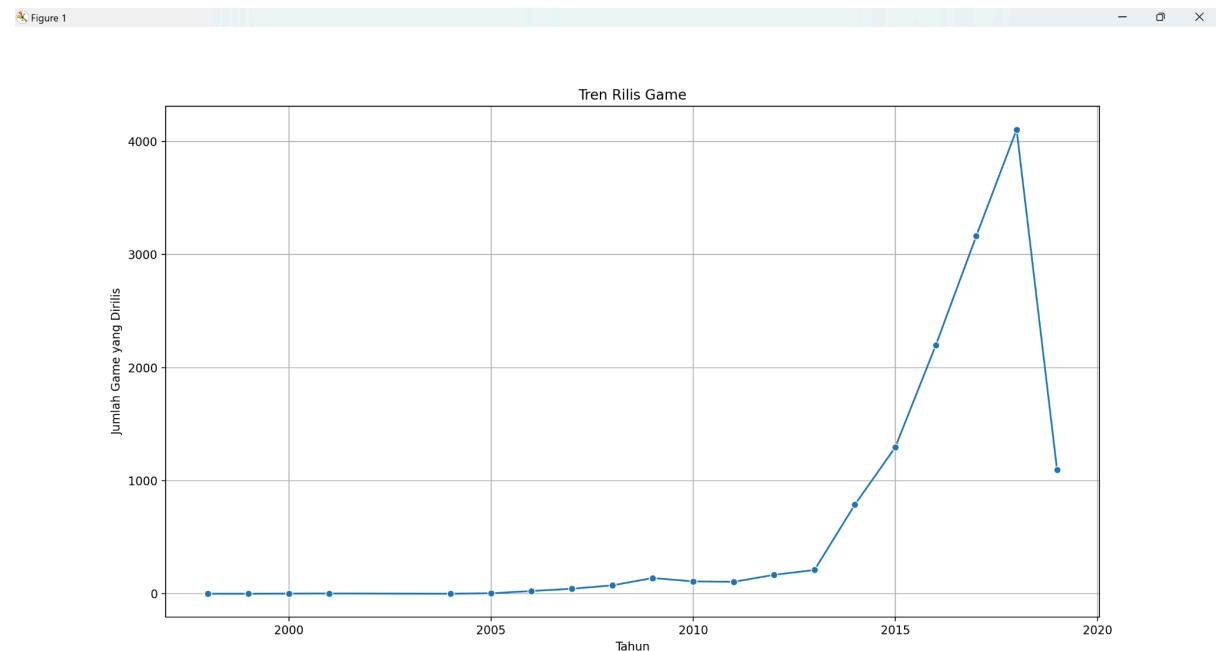
```
def get_recommendations_for_user(user_id, n_neighbors=6):
    try:
        user_id = int(user_id) # Pastikan user_id berupa integer
    except ValueError:
        return [] # Jika input tidak valid, kembalikan daftar kosong

    if user_id not in unique_user_ids:
        return [] # Jika user_id tidak ditemukan, kembalikan daftar kosong

    # Dapatkan rekomendasi
    recommendations = recommend_games(user_id, n_neighbors=n_neighbors)
    return recommendations
```

- Fungsi Utama untuk Flask
  - Memastikan ID user valid.
  - Jika valid, mengembalikan daftar rekomendasi game.

## ● Output

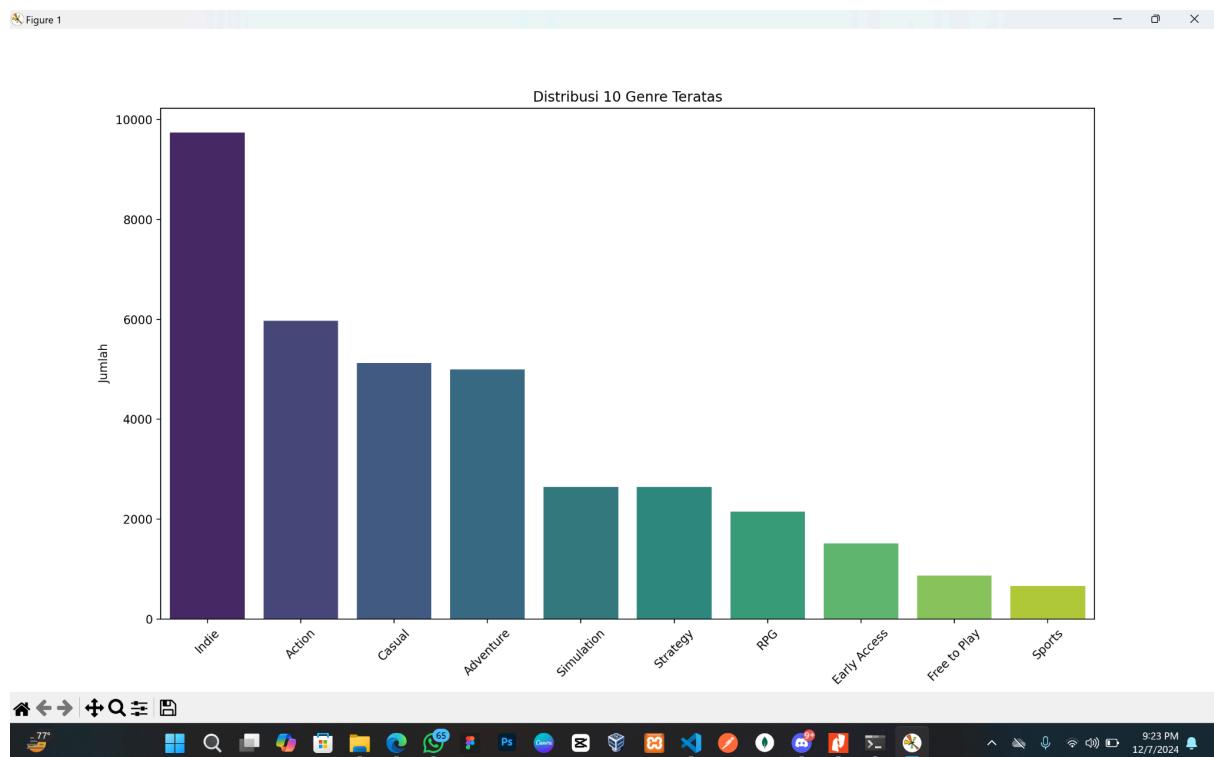


Gambar 1, Data Tren Rilis Game

Grafik garis (line chart) yang menunjukkan jumlah game yang dirilis setiap tahun berdasarkan data release\_year dari dataset games.csv. Grafik ini menggunakan sumbu X sebagai tahun rilis dan sumbu Y sebagai jumlah game yang dirilis pada tahun tersebut. Grafik ini memberikan gambaran bagaimana jumlah game yang dirilis mengalami perubahan dari tahun ke tahun. Pola ini bisa menunjukkan tren perkembangan industri game, misalnya:

- Lonjakan jumlah game rilis mengindikasikan periode dimana industri game sedang mengalami pertumbuhan pesat.
- Penurunan jumlah rilis bisa terjadi karena perubahan pasar atau teknologi.

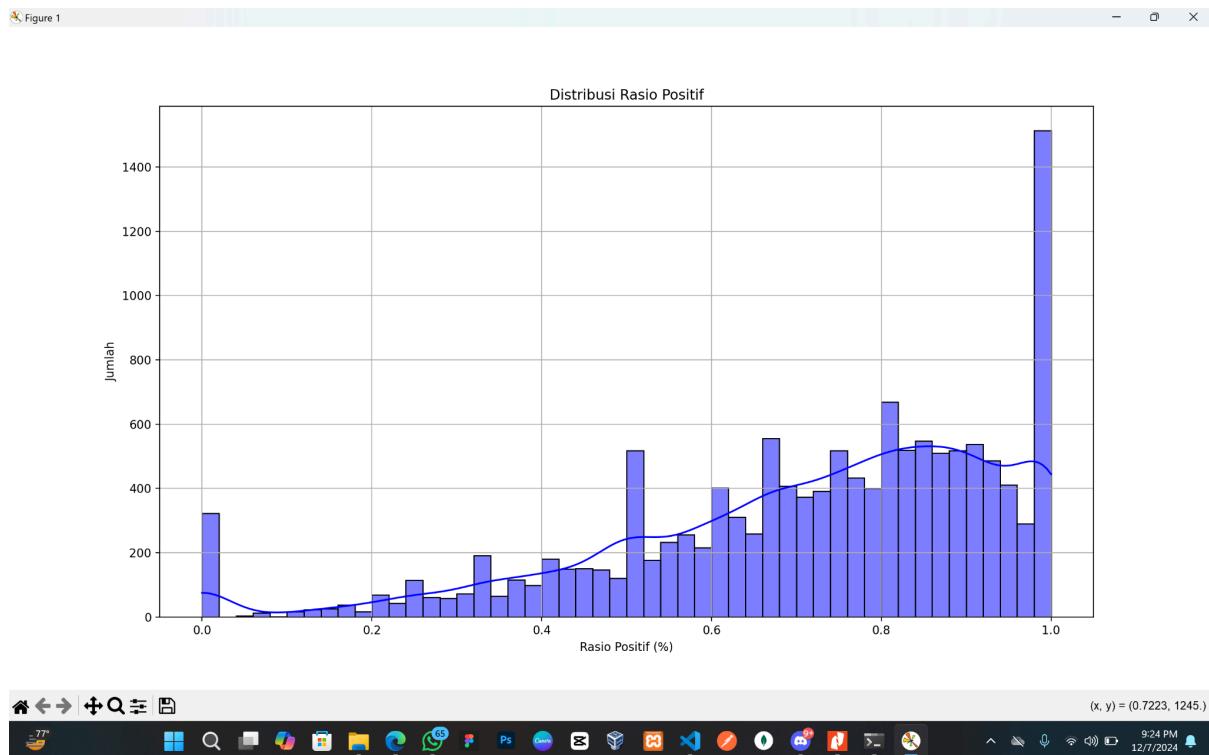
Grafik garis cocok untuk menunjukkan tren data yang berurutan (time series) dari tahun ke tahun. Bentuk garis yang naik turun mencerminkan fluktuasi rilis game berdasarkan periode tertentu. Grafik berwarna biru untuk garis utama karena warna ini memberikan kesan profesional dan mudah dibaca. Setiap titik pada garis menunjukkan jumlah game rilis di tahun tertentu untuk memperjelas perubahan secara visual.



Gambar 2, Data Distribusi 10 Genre Teratas

Diagram batang (bar chart) yang menampilkan 10 genre game terpopuler berdasarkan jumlah game yang termasuk dalam genre tersebut. Genre diambil dari kolom genres, yang dapat berisi beberapa genre per game. Diagram ini menunjukkan genre game yang paling sering muncul di dataset, memberikan wawasan tentang preferensi umum di industri game. Genre populer seperti "Action", "Adventure", atau "RPG" mungkin mendominasi grafik.

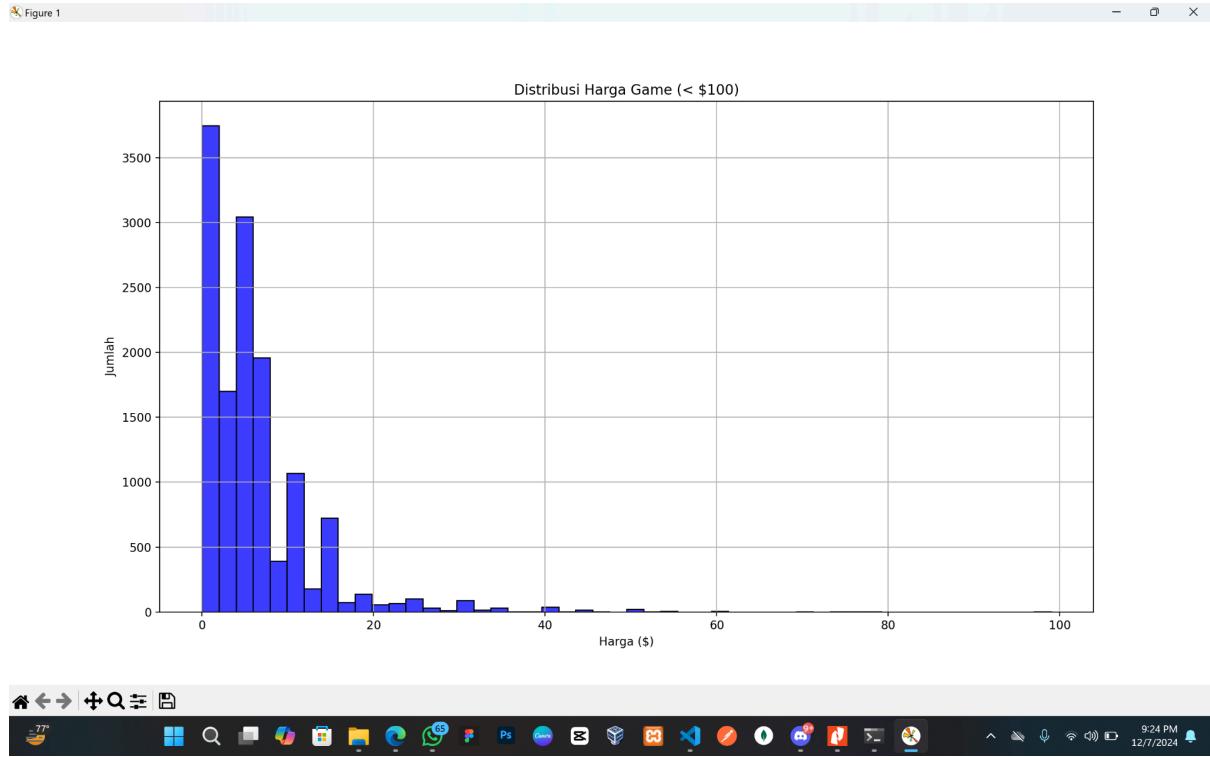
Diagram batang efektif dalam membandingkan jumlah data kategori (genre) secara visual. Setiap batang mewakili satu genre, dengan panjang batang menunjukkan jumlah game dalam genre tersebut. Palet warna seperti hijau hingga kuning (misalnya, palet viridis) untuk memberikan visualisasi yang menarik. Warna berbeda pada setiap batang membantu pengguna dengan mudah membedakan genre satu dengan yang lain.



Gambar 3, Data Distribusi Rasio Positif

Histogram yang memvisualisasikan distribusi rasio ulasan positif, dihitung dari `positive_ratings` dibandingkan dengan total ulasan (`positive_ratings + negative_ratings`). Sumbu X menunjukkan nilai rasio (dalam persentase), sedangkan sumbu Y menunjukkan jumlah game yang termasuk dalam kategori tersebut. Memberikan gambaran tingkat kepuasan pengguna terhadap game berdasarkan ulasan yang mereka berikan. Game dengan rasio positif tinggi (misalnya  $>80\%$ ) dianggap memiliki reputasi baik.

Histogram menunjukkan distribusi data (bukan tren atau perbandingan kategori). Bentuk grafik mencerminkan pola distribusi, di mana data bisa memusat (game dengan ulasan positif tinggi) atau tersebar (game dengan rasio beragam). Biasanya menggunakan satu warna (misalnya, biru) untuk batang histogram agar fokus tetap pada distribusi data. Penambahan garis KDE (Kernel Density Estimation) sering digunakan untuk menunjukkan pola distribusi yang lebih halus.



Gambar 4, Data Distribusi Harga Game (< \$100)

Histogram yang menunjukkan distribusi harga game dengan filter harga di bawah \$100. Sumbu X menunjukkan rentang harga, sedangkan sumbu Y menunjukkan jumlah game yang berada dalam rentang harga tersebut. Membantu memahami pola harga game yang tersedia di dataset. Grafik ini penting untuk mengetahui harga rata-rata game atau melihat apakah sebagian besar game berada di kisaran harga tertentu.

Histogram dipilih untuk menunjukkan distribusi harga dalam kelompok (binned data). Game dengan harga tertentu (misalnya \$10–\$20) lebih sering muncul mungkin karena merupakan kisaran harga yang populer atau terjangkau. Menggunakan warna biru untuk menyampaikan kesan netral dan fokus pada data distribusi. Pemilihan warna tunggal membantu mencegah gangguan visual pada interpretasi.

### 3.2. PENGUJIAN

- **Code form.html**

```
<form action="/result" method="POST">
```

- Struktur Formulir

- `action="/result"`: Form akan mengirimkan data ke URL /result.
- `method="POST"`: Data akan dikirim menggunakan metode POST (lebih aman untuk data sensitif).

```
<select class="form-control bg-transparent" id="user_id" name="user_id"
onchange="enableSubmit()>
    <option value="" disabled selected>Pilih ID User</option>
    {% for user_id in recommended_user_ids %}<option value={{user_id}}>{{user_id}}</option>
    {% endfor %}
</select>
```

```
        <option value="{{ user_id }}">{{ user_id }}</option>
    {% endfor %}
</select>
```

- Dropdown untuk Memilih ID User
  - <select>: Komponen dropdown yang memungkinkan pengguna memilih ID User.
    - id="user\_id": Memberikan identitas untuk elemen ini agar dapat digunakan oleh JavaScript atau CSS.
    - name="user\_id": Nama yang akan digunakan untuk mengirim data ke server.
    - onchange="enableSubmit()": Ketika pengguna memilih ID User, fungsi enableSubmit() akan dipanggil untuk mengaktifkan tombol kirim.
  - <option>: Elemen untuk menampilkan setiap pilihan dalam dropdown.
    - value="{{ user\_id }}": Nilai yang akan dikirim ke server adalah user\_id.
    - {% for user\_id in recommended\_user\_ids %}: Looping untuk menampilkan semua ID User dari daftar recommended\_user\_ids yang dikirim dari backend Flask.

```
<button type="submit" id="submit_btn" class="btn btn-primary w-100 py-3"
disabled>Kirim</button>
```

- Tombol Kirim
  - <button>: Tombol untuk mengirimkan form.
    - type="submit": Menentukan bahwa ini adalah tombol untuk submit form.
    - id="submit\_btn": Identitas tombol untuk digunakan oleh JavaScript.
    - disabled: Tombol dinonaktifkan secara default sampai pengguna memilih ID User di dropdown.
    - class="btn btn-primary w-100 py-3": Memberikan gaya tombol (menggunakan class Bootstrap).

- **Code result.html**

```
<div class="container">
```

- Struktur Utama (Container)
  - Elemen <div class="container"> adalah wadah utama yang menampung semua elemen di halaman.
  - Membuat tata letak lebih terorganisasi, biasanya dengan menggunakan Bootstrap.

```
<div class="row">
    <div class="col-12">
        <center>
            <h4>Games Recommendation Information</h4>
        </center>
    </div>
```

```
</div>
```

- Judul Halaman, menampilkan judul halaman "Games Recommendation Information" di tengah.

```
<div class="row mb-3">
```

```
    <div class="col-12">
```

```
        <p><strong>ID User:</strong> {{ user_id }}</p>
```

```
    </div>
```

```
</div>
```

- Menampilkan ID User

- Menampilkan ID User yang dipilih pengguna pada halaman sebelumnya.

- {{ user\_id }}: ID User yang dikirim dari backend Flask.

```
{% for game in recommendations %}
```

```
<div class="row mt-4">
```

```
    <h6>Recommendation {{ loop.index }}</h6>
```

```
    <div class="col-4"><strong>Games Name:</strong> {{ game['title'] }}</div>
```

```
    <div class="col-2"><strong>Ratio:</strong> {{ game['positive_ratings'] }}</div>
```

```
    <div class="col-4"><strong>Genre:</strong> {{ game['genres'] }}</div>
```

```
        <div class="col-2"><strong>Price:</strong> ${{ '%.2f' | format(game['price']) }}
```

```
></div>
```

```
</div>
```

```
{% endfor %}
```

- Daftar Rekomendasi Game

- {{% for game in recommendations %}}: Looping untuk menampilkan setiap rekomendasi game yang diberikan oleh backend Flask melalui variabel recommendations.

- game['title']: Nama game.

- game['positive\_ratings']: Rasio ulasan positif.

- game['genres']: Genre game.

- game['price']: Harga game, diformat hingga dua desimal menggunakan '%.2f'.

- Recommendation {{ loop.index }}: Menampilkan nomor urut rekomendasi.

```
<div class="row mt-4">
```

```
    <div class="col-12">
```

```
        <a href="{{ url_for('form') }}" class="btn btn-primary w-100 py-3">Back</a>
```

```
    </div>
```

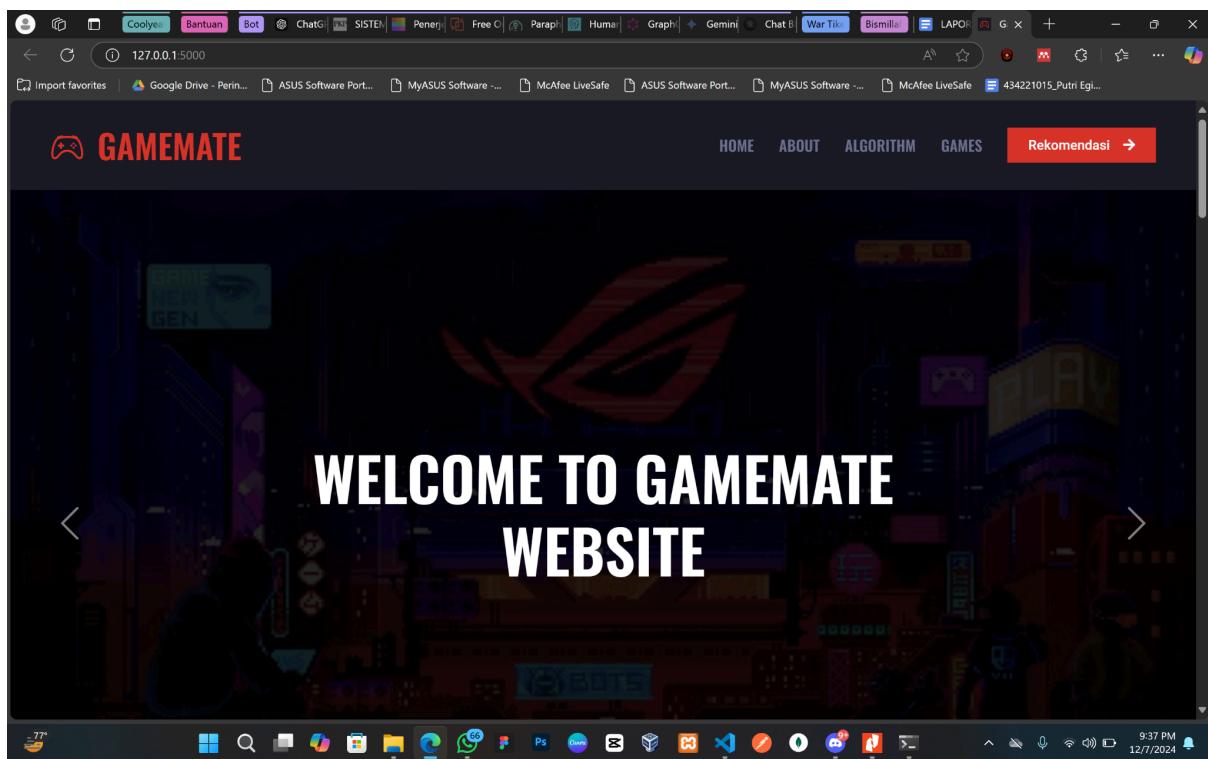
```
</div>
```

- Tombol Kembali

- Menambahkan tombol untuk kembali ke halaman form.

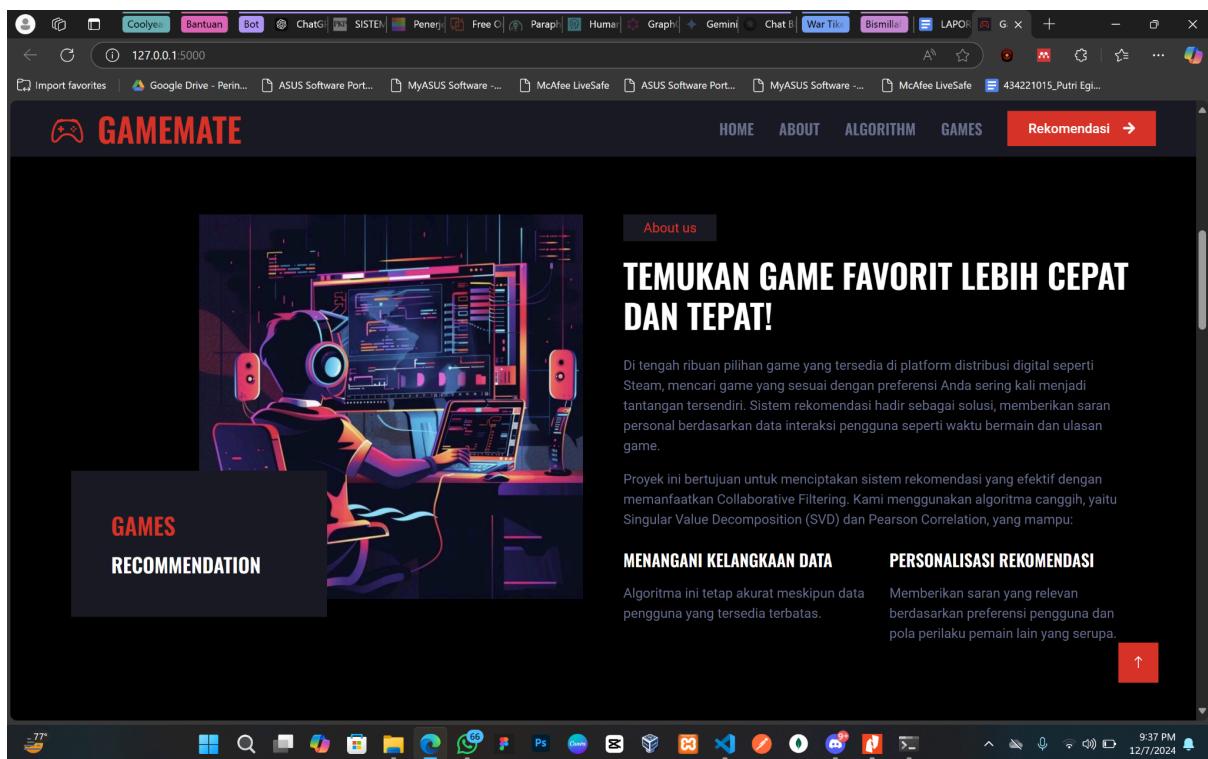
- url\_for('form'): Flask akan mengarahkan ke fungsi form, yaitu halaman untuk memilih ID User.

- Interface



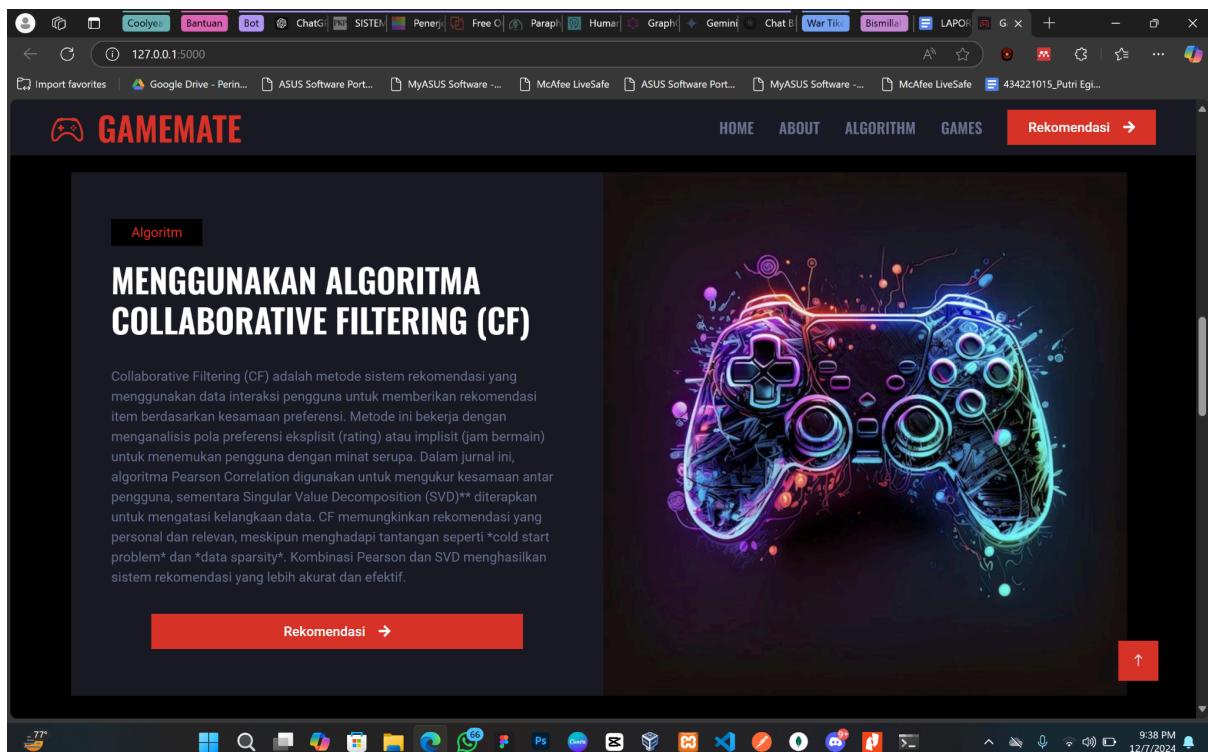
Gambar 5, Pages Header

Halaman ini adalah antarmuka awal yang berisi elemen header dengan nama proyek atau navigasi utama seperti Home, About Us, dan lainnya. Header ini dirancang untuk memberikan kesan profesional dan mempermudah pengguna mengakses fitur utama sistem rekomendasi game



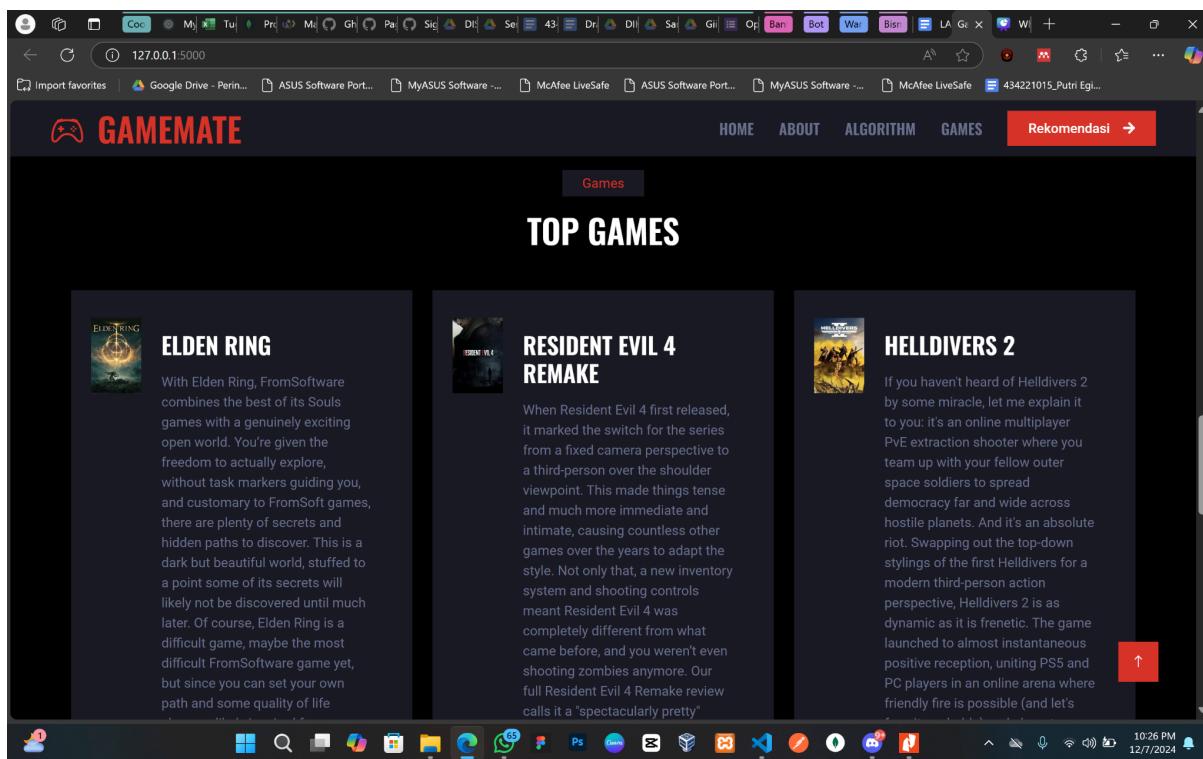
Gambar 6, Pages About Us

Halaman ini berfungsi memberikan informasi tentang pengembang aplikasi, seperti nama tim, tujuan, dan manfaat aplikasi. Tampilan ini membantu pengguna memahami latar belakang dan kredibilitas sistem yang mereka gunakan.

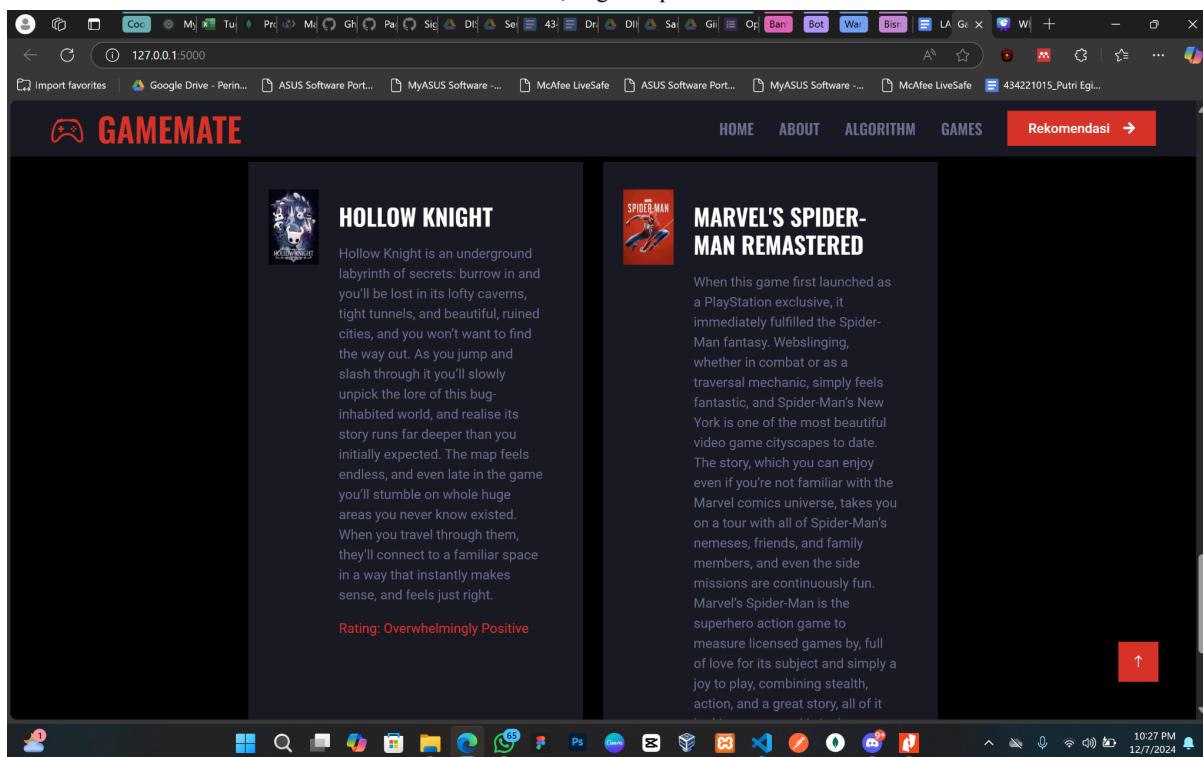


Gambar 7, Pages Algoritma Genetika

Halaman ini menjelaskan eksperimen menggunakan algoritma tertentu (misalnya, Algoritma Genetika). Meski mungkin tidak digunakan dalam rekomendasi utama, halaman ini memberikan wawasan tambahan untuk pengguna yang ingin mengetahui proses teknis atau eksperimen terkait.

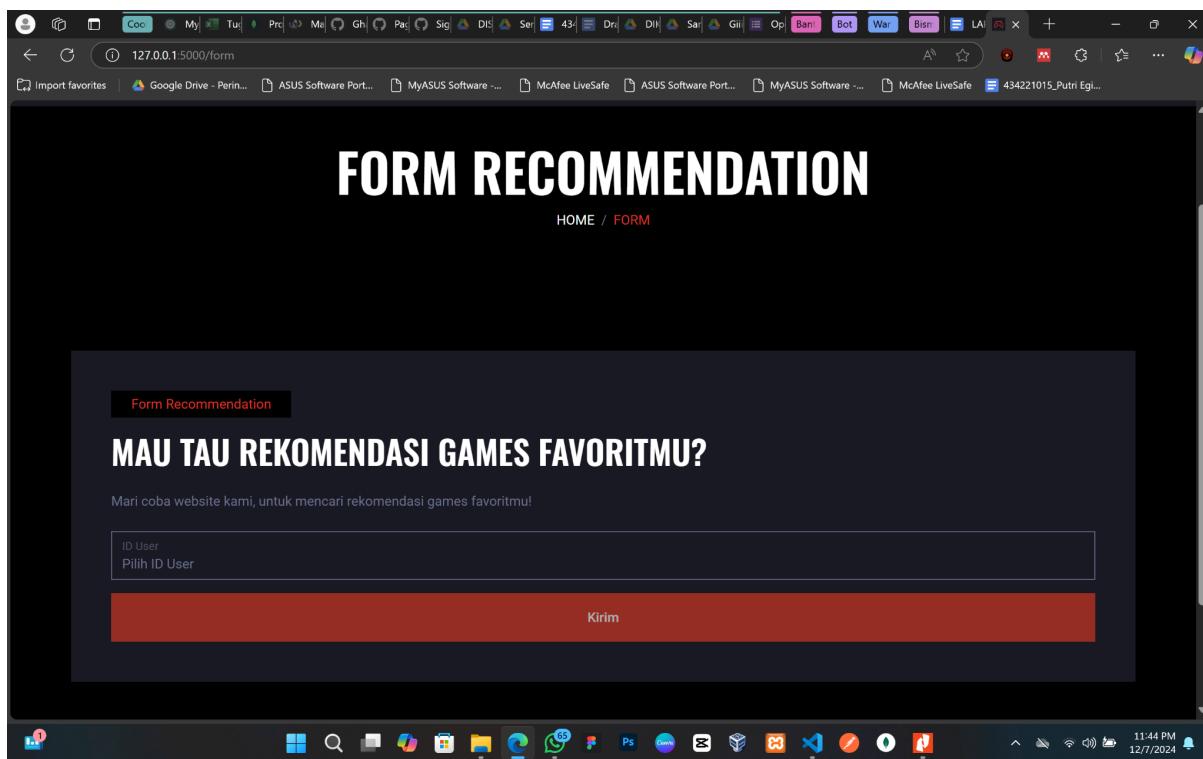


Gambar 8, Pages Top Games



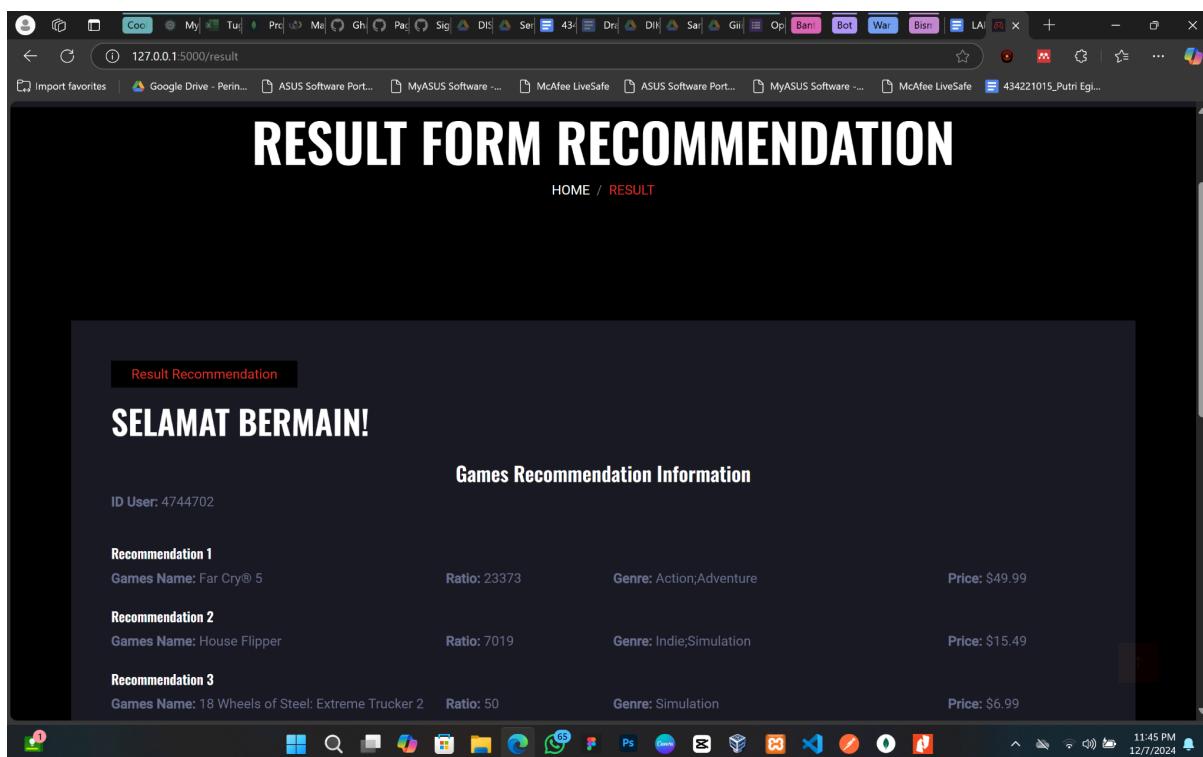
Gambar 9, Pages Top Games

Halaman ini menampilkan daftar game terbaik berdasarkan data tertentu, seperti ulasan positif atau popularitas. Gambar ini memperlihatkan daftar dalam format terstruktur, memudahkan pengguna menemukan game-game populer yang disarankan.



Gambar 10, Pages Form Recommendation

Halaman ini memungkinkan pengguna memasukkan ID mereka melalui form. Dengan tampilan yang sederhana dan intuitif, halaman ini menjadi titik awal untuk menghasilkan rekomendasi yang personal.



Gambar 11, Pages Result Recommendation

Halaman hasil ini menampilkan daftar rekomendasi game berdasarkan ID yang dimasukkan pengguna sebelumnya. Setiap rekomendasi mencakup detail seperti nama game, genre, harga, dan rasio ulasan positif, yang membantu pengguna memilih game yang sesuai dengan preferensi mereka.

### 3.3. KESIMPULAN

Penelitian ini berfokus pada pengembangan sistem rekomendasi game untuk mengatasi tantangan "information overload" yang sering dialami pengguna platform Steam. Dengan katalog lebih dari 18.000 game, pengguna membutuhkan alat yang dapat mempermudah mereka menemukan game yang sesuai dengan preferensi. Sistem rekomendasi berbasis Collaborative Filtering dengan algoritma Singular Value Decomposition (SVD) dipilih karena kemampuannya untuk memanfaatkan data pengguna lain sebagai referensi dalam memberikan rekomendasi yang relevan.

Dataset yang digunakan terdiri dari dua file utama, yaitu games.csv dan recommendations.csv. File games.csv berisi data tentang karakteristik game seperti nama, genre, ulasan, dan harga, sedangkan file recommendations.csv mencatat data rekomendasi dan ulasan pengguna. Metodologi penelitian melibatkan pengumpulan data, pra-pemrosesan untuk mengoptimalkan memori dan menyiapkan data, serta penerapan SVD untuk menghasilkan prediksi rekomendasi yang akurat.

Sistem ini diimplementasikan menggunakan Python dan framework Flask untuk antarmuka web. Model rekomendasi dibangun dengan memanfaatkan teknik Matrix Factorization menggunakan SVD, yang mampu menangani matriks besar dengan efisiensi tinggi. Sistem ini juga dilengkapi dengan fitur visualisasi data seperti tren rilis game, distribusi genre, rasio ulasan positif, dan harga game untuk memberikan wawasan tambahan.

Hasil pengujian menunjukkan bahwa sistem dapat memberikan rekomendasi personal yang relevan. Pengguna dapat memasukkan ID mereka melalui form, dan sistem akan menampilkan daftar game yang direkomendasikan berdasarkan kesamaan preferensi dengan pengguna lain. Setiap rekomendasi mencakup detail seperti nama game, genre, ulasan positif, dan harga, yang membantu pengguna memilih game yang sesuai.

Secara keseluruhan, sistem ini tidak hanya meningkatkan pengalaman pengguna dalam menemukan game yang relevan tetapi juga memberikan manfaat strategis bagi pengembang platform. Sistem ini dapat meningkatkan keterlibatan pengguna, memperpanjang durasi bermain, dan mendorong pembelian game baru, sehingga berkontribusi pada keberhasilan platform distribusi digital seperti Steam.

## **DAFTAR PUSTAKA**

Lin, D. et al. (2019) ‘An empirical study of game reviews on the Steam platform’, Empirical Software Engineering, 24(1), pp. 170–207. Available at:  
<https://doi.org/10.1007/S10664-018-9627-4/FIGURES/17>.

Li, Q. et al. (2021) ‘A Game Recommendation Method Based on Machine Learning’, 2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer, ICFTIC 2021, pp. 307–312. Available at:  
<https://doi.org/10.1109/ICFTIC54370.2021.9647367>.

Lomanto, W.E. et al. (2023) ‘Collaborative Filtering for Steam Games Recommendation’, 2023 5th International Conference on Cybernetics and Intelligent Systems, ICORIS 2023 [Preprint]. Available at: <https://doi.org/10.1109/ICORIS60118.2023.10352243>.

Bunga, R., Batista, F. and Ribeiro, R. (2021) ‘From implicit preferences to ratings: Video games recommendation based on collaborative filtering’, From implicit preferences to ratings: Video games recommendation based on collaborative filtering, 1, pp. 209–216. Available at: <https://doi.org/10.5220/0010655900003064>.

steam game review (no date). Available at:  
<https://www.kaggle.com/datasets/bayuabdurrosyidye/steam-game-review?select=games.csv> (Accessed: 7 December 2024).