

**LAPORAN PROYEK UJIAN AKHIR SEMESTER  
MATA KULIAH PEMBELAJARAN MESIN (PRAKTIKUM)**

**“PENGEMBANGAN SISTEM REKOMENDASI PILIHAN *GAME*  
BERBASIS *CONTENT BASED LEARNING (CBF)*”**



**Disusun Oleh :**

**KELOMPOK 2 – Kelas Praktikum C3**

434221034 | Anisa Maharani

434221037 | Salma Aida Yasmi

434221047 | Ratna Firdaus

**PROGRAM STUDI D4 TEKNIK INFORMATIKA**

**FAKULTAS VOKASI**

**UNIVERSITAS AIRLANGGA**

**SURABAYA**

**2024**

## DAFTAR ISI

DAFTAR ISI .....	I
DAFTAR TABEL .....	III
DAFTAR GAMBAR .....	IV
BAB 1 .....	1
PENDAHULUAN .....	1
1.1    LATAR BELAKANG .....	1
1.2    RUMUSAN MASALAH .....	1
1.3    TUJUAN .....	2
1.4    MANFAAT .....	2
1.5    BATASAN MASALAH .....	2
BAB 2 .....	3
TINJAUAN PUSTAKA .....	3
2.1    Game .....	3
2.2    Sistem Rekomendasi .....	4
2.3    Content Based Filtering (CBF) .....	5
2.4    Cosine Similarity .....	5
BAB 3 .....	7
METODOLOGI .....	7
3.1    Alur Pengembangan Sistem .....	7
3.2    Alat dan Bahan .....	9
BAB 4 .....	10
HASIL DAN PEMBAHASAN .....	10
4.1    Proses Preprocessing Data .....	10
4.2    Proses Implementasi Sistem .....	21
4.3    Hasil Implementasi Sistem .....	25

BAB 5 .....	28
KESIMPULAN.....	28
LAMPIRAN.....	29
DAFTAR PUSTAKA.....	30

## DAFTAR TABEL

Tabel 1. Feature Dataset.....	8
-------------------------------	---

## DAFTAR GAMBAR

Gambar 1. Game 2D dan 3D.....	3
Gambar 2. Tipe Sistem Rekomendasi .....	4
Gambar 3. Content Based Filtering (CBF) .....	5
Gambar 4. Rumus Cosine Similarity .....	6
Gambar 5. Alur Pengembangan Sistem .....	7
Gambar 6. Dataset.....	7
Gambar 7. Import Library.....	10
Gambar 8. Fungsi read_csv.....	10
Gambar 9. Fungsi index.....	11
Gambar 10. Fungsi df.shape().....	11
Gambar 11. Dimensi Dataset .....	11
Gambar 12. Fungsi isnull().sum().....	11
Gambar 13. Output Missing Value.....	12
Gambar 14. Fungsi dropna() .....	12
Gambar 15. Output Handling Missing Value.....	13
Gambar 16. Fungsi df.shape .....	13
Gambar 17. Dimensi Dataset .....	13
Gambar 18. Fungsi df.duplicate() .....	13
Gambar 19. Output Duplikasi Data .....	14
Gambar 20. Unique Value.....	14
Gambar 21. Output Unique Value.....	14
Gambar 22. Drop Kolom yang Tidak Relevan .....	14
Gambar 23. Kolom Rating "RP" .....	15
Gambar 24. Print Dimensi Baru.....	15
Gambar 25. Hasil Dimensi Baru.....	15
Gambar 26. Simpan Data Baru .....	15
Gambar 27. Dataset Baru.....	15
Gambar 28. Gambaran Dataset Baru .....	16
Gambar 29. Read Dataset Baru.....	16
Gambar 30. Konversi Kategorikal .....	16
Gambar 31. Normalisasi MinMaxScaler().....	17

Gambar 32. Hasil Normalisasi .....	17
Gambar 33. Simpan Hasil Normalisasi.....	17
Gambar 34. Dataset Hasil Normalisasi .....	18
Gambar 35. Cosine Similarity.....	18
Gambar 36. Informasi Cosine .....	18
Gambar 37. Hasil Cosine Similarity .....	19
Gambar 38. Fungsi Cosine Similarity.....	19
Gambar 39. Fungsi Rekomendasi Game .....	20
Gambar 40. Struktur Direktori Proyek.....	21
Gambar 41. Inisialisasi Flask dan Library .....	22
Gambar 42. Fungsi Pemilihan Gambar Acak .....	22
Gambar 43. Route Index .....	23
Gambar 44. Fungsi Rekomendasi Berdasarkan Nama Game.....	23
Gambar 45. Fungsi Rekomendasi Berdasarkan Kategori .....	24
Gambar 46. Running Sistem.....	24
Gambar 47. Halaman Beranda .....	25
Gambar 48. Halaman Form Rekomendasi.....	25
Gambar 49. Kategori Rekomendasi.....	26
Gambar 50. Recent Review .....	26
Gambar 51. Halaman Hasil Rekomendasi .....	27

# **BAB 1**

## **PENDAHULUAN**

### **1.1 LATAR BELAKANG**

Perkembangan teknologi digital telah membawa perubahan signifikan dalam cara manusia mengakses informasi dan hiburan. Salah satu bentuk hiburan yang paling populer di era modern ini adalah video game. Industri game telah berkembang pesat, mencakup berbagai genre, platform, dan gaya bermain yang beragam. Namun, dengan banyaknya pilihan game yang tersedia, pengguna sering kali menghadapi kesulitan dalam menemukan game yang sesuai dengan preferensi dan kebutuhan mereka.

Untuk mengatasi masalah ini, sistem rekomendasi telah menjadi solusi yang penting. Sistem rekomendasi membantu pengguna menyaring informasi yang relevan dari banyaknya opsi yang tersedia, sehingga mempermudah pengambilan keputusan. Salah satu metode yang banyak digunakan dalam pengembangan sistem rekomendasi adalah *Content-Based Filtering* (CBF), yang bekerja dengan menganalisis atribut dari item yang telah digunakan atau disukai oleh pengguna untuk merekomendasikan item lain yang memiliki kesamaan.

Proyek ini bertujuan untuk mengembangkan sistem rekomendasi game berbasis CBF dengan menggunakan metode *Cosine Similarity*. Sistem ini diimplementasikan menggunakan framework Flask untuk memberikan rekomendasi yang interaktif dan mudah diakses. Dengan adanya sistem ini, diharapkan pengguna dapat dengan mudah menemukan game yang relevan dan meningkatkan pengalaman bermain mereka.

### **1.2 RUMUSAN MASALAH**

Berdasarkan latar belakang di atas, dirumuskan beberapa masalah yang ingin diselesaikan dalam proyek ini:

1. Bagaimana cara mengembangkan sistem rekomendasi yang mampu memberikan saran game berdasarkan preferensi pengguna ?
2. Bagaimana mengimplementasikan metode *Content-Based Filtering* dengan menggunakan *Cosine Similarity* untuk mengukur kesamaan antar game ?
3. Bagaimana sistem dapat diimplementasikan dalam sebuah platform interaktif sehingga mudah digunakan oleh pengguna ?

### 1.3 TUJUAN

Proyek ini memiliki beberapa tujuan utama, yaitu:

1. Mengembangkan sistem rekomendasi game berbasis CBF yang efektif dan akurat, dengan menggunakan metode *Cosine Similarity* untuk menghitung tingkat kesamaan atribut antar game.
2. Mengimplementasikan sistem rekomendasi ini ke dalam sebuah aplikasi berbasis web menggunakan framework Flask.
3. Menyediakan solusi praktis bagi pengguna untuk menemukan game yang relevan dengan preferensi mereka.

### 1.4 MANFAAT

Hasil dari proyek ini diharapkan memberikan manfaat sebagai berikut:

1. Mempermudah dalam menemukan game yang sesuai dengan preferensi, sehingga dapat meningkatkan pengalaman bermain.
2. Memberikan wawasan baru mengenai pengembangan sistem rekomendasi berbasis CBF yang dapat diaplikasikan pada bidang lain.
3. Menjadi referensi dan bahan kajian lebih lanjut dalam pengembangan sistem rekomendasi dan aplikasi *machine learning*.

### 1.5 BATASAN MASALAH

Agar pengembangan sistem lebih terarah, terdapat beberapa batasan masalah dalam proyek ini, yaitu:

1. Sistem hanya menggunakan dataset “Video Game Sales with Ratings” yang diambil dari platform Kaggle.
2. Metode yang digunakan untuk sistem rekomendasi adalah *Content-Based Filtering* berbasis atribut game, sehingga rekomendasi terbatas pada kesamaan atribut.
3. Implementasi sistem dilakukan menggunakan framework Flask dengan antarmuka berbasis web, sehingga belum mendukung aplikasi mobile secara langsung.
4. Pengujian sistem dilakukan berdasarkan data dalam dataset tanpa melibatkan preferensi pengguna secara real-time.



## BAB 2

### TINJAUAN PUSTAKA

Pada bab ini akan dijelaskan mengenai tinjauan pustaka yang dapat mendukung pengembangan proyek ini.

#### 2.1 Game

Menurut Kamus Besar Bahasa Indonesia, game diartikan sebagai permainan. Permainan adalah kegiatan yang bertujuan untuk menghilangkan kepenatan melalui aktivitas yang melibatkan kecerdasan berpikir dan strategi, yang dapat dilakukan secara individu maupun bersama-sama. Aktivitas ini dirancang untuk memungkinkan interaksi dengan sistem dan konflik yang sengaja dibuat guna menciptakan pengalaman bermain yang seru dan menyenangkan, sehingga menjadikan aktivitas ini diminati oleh banyak orang (Andri Suryadi, 2017). *Game* memiliki berbagai macam genre, seperti aksi, simulasi, teka-teki, petualangan, balapan, *Role Playing Games* (RPG), strategi, olahraga, kartu, papan, edukasi, dan kata.

Berdasarkan representasi visualnya, *game* dapat dibagi menjadi dua jenis, yaitu *game* 2 dimensi (2D) dan 3 dimensi (3D). *Game* 2D secara matematis hanya melibatkan dua elemen koordinat kartesius, yaitu  $x$  dan  $y$ . Sebaliknya, *game* 3D melibatkan elemen  $x$ ,  $y$ , dan  $z$  dalam perhitungannya, sehingga konsep kamera pada *game* 3D lebih kompleks dan menyerupai kamera di dunia nyata.

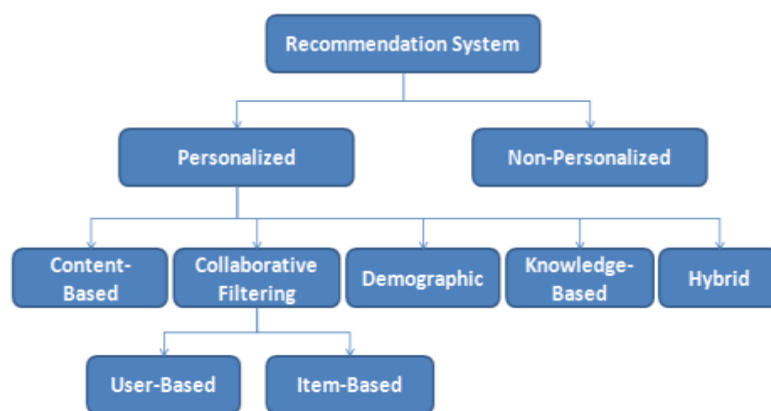


Gambar 1. Game 2D dan 3D

## 2.2 Sistem Rekomendasi

Sistem rekomendasi adalah sebuah metode atau perangkat lunak yang dirancang untuk memberikan rekomendasi item tertentu yang dianggap menarik atau relevan bagi pengguna. Rekomendasi diberikan berdasarkan berbagai proses pengambilan keputusan, seperti menentukan item apa yang sebaiknya dibeli, musik apa yang cocok didengarkan, makanan apa yang sebaiknya dikonsumsi, atau *game* apa yang menarik untuk dimainkan.

Pendekatan dalam sistem rekomendasi terbagi menjadi dua jenis, yaitu rekomendasi yang bersifat *personalized* dan *non-personalized* (Das, Sahoo, dan Datta, 2017).



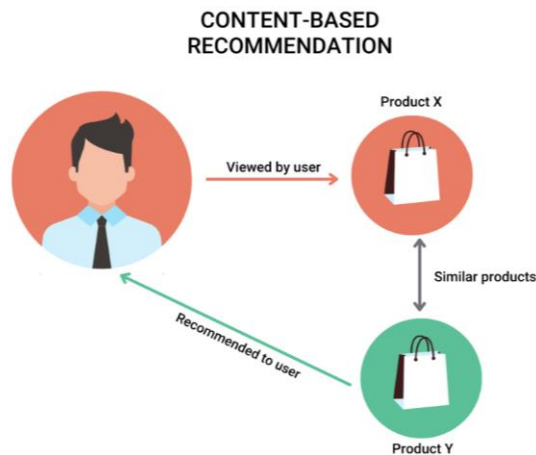
Gambar 2. Tipe Sistem Rekomendasi

Menurut Nagarnai dan Thomas (2023), sistem rekomendasi yang bersifat *non-personalized* memberikan rekomendasi dengan melihat semua item yang tersedia dalam sistem secara keseluruhan, tanpa mempertimbangkan preferensi spesifik pengguna. Contoh penerapannya seperti menampilkan item yang paling populer atau item yang baru saja rilis.

Sedangkan menurut Das, Sahoo, dan Datta (2017), sistem rekomendasi yang bersifat *personalized* memberikan rekomendasi berdasarkan preferensi individu pengguna. Untuk mengimplementasikan sistem rekomendasi yang bersifat *personalized*, terdapat tiga metode utama yang dapat digunakan adalah *Collaborative Filtering* (CF), *Content Based Filtering* (CBF) dan hybrid.

### 2.3 Content Based Filtering (CBF)

*Content-based Filtering* adalah metode dalam sistem rekomendasi yang merekomendasikan item berdasarkan kesamaan atribut, karakteristik, atau fitur dengan item yang pernah berinteraksi dengan pengguna.



Gambar 3. Content Based Filtering (CBF)

Misalnya, ketika pengguna telah membeli sepatu olahraga merek X, sistem akan merekomendasikan sepatu olahraga merek lain, sepatu dengan harga atau model yang mirip, atau aksesoris yang relevan seperti kaos kaki dan tas olahraga. Rekomendasi ini didasarkan pada kesamaan kategori, merek, atau gaya produk yang sesuai dengan preferensi pengguna.

Kelemahan dari metode *Content-based Filtering* adalah rekomendasi yang dihasilkan terbatas hanya pada item yang serupa, sehingga tidak ada kesempatan untuk mendapatkan item yang tidak terduga.

### 2.4 Cosine Similarity

Cosine similarity adalah metode yang sering digunakan untuk menghitung kemiripan antara dua item atau objek. Secara umum, fungsi similarity menerima dua buah objek yang diwakili sebagai vektor bilangan riil (misalnya 0 dan 1), dan menghasilkan nilai kemiripan dalam bentuk bilangan riil.

Cosine similarity mengukur kemiripan antara dua vektor dengan menghitung nilai cosinus dari sudut antara keduanya. Nilai ini berkisar antara -1 hingga 1, di mana nilai 1 menunjukkan kemiripan yang sangat tinggi (vektor tersebut sejajar), nilai 0 menunjukkan

tidak ada kemiripan, dan nilai -1 menunjukkan kemiripan negatif (vektor tersebut berlawanan arah). Fungsi *cosine similarity* antara item A dan item B dapat dihitung dengan persamaan berikut :

$$Sim(q, d_j) = \frac{q \times d_j}{|q| \times |d_j|} = \frac{\sum_{i=1}^n w_{i,q} \times w_{i,j}}{\sqrt{\sum_{i=1}^t (w_{i,q})^2} \sqrt{\sum_{i=1}^t (w_{i,j})^2}}$$

Gambar 4. Rumus Cosine Similarity

Keterangan:

$Sim(q, d_j)$  = similarity yang dimiliki oleh query dan dokumen

$|q|$  = jarak dari sebuah query

$|d_j|$  = jarak dari sebuah dokumen

$w_{i,j}$  = bobot yang dimiliki dokumen ke-i

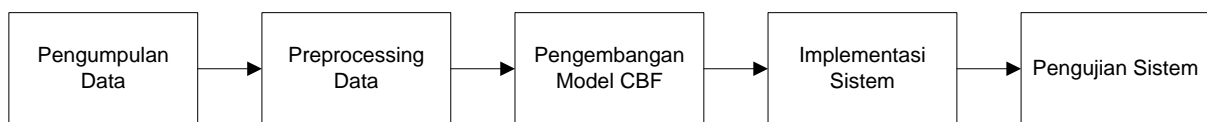
$w_{i,q}$  = bobot yang dimiliki query dokumen ke-i

## BAB 3

### METODOLOGI

Metodologi yang digunakan dalam pengembangan sistem rekomendasi game berbasis *Content Based Filtering* (CBF) ini terbagi menjadi sejumlah tahap yang digambarkan dalam bentuk *flow* alur proses pengembangan. Adapun rincian alur proses pengembangan yang digunakan diantaranya sebagai berikut :

#### 3.1 Alur Pengembangan Sistem



Gambar 5. Alur Pengembangan Sistem

#### 1. Pengumpulan Data

Tahap pertama dalam pengembangan sistem ini adalah mengumpulkan data yang relevan dengan topik utama yaitu tentang Game. Data yang digunakan diambil dari website Kaggle dengan nama “Video Game Sales with Ratings”.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Name,Platform,Year_of_Release,Genre,Publisher,NA_Sales,EU_Sales,JP_Sales,Other_Sales,Global_Sales,Critic_Score,Critic_Count,User_Score,User_Count,Developer,Rating															
2	Wii Sports,Wii,2006,Sports,Nintendo,41.36,28.96,3.77,8.45,82.53,76.51,8.322,Nintendo,E															
3	Super Mario Bros.,NES,1985,Platform,Nintendo,29.08,3.58,6.81,0.77,40.24,.....															
4	Mario Kart Wii,Wii,2008,Racing,Nintendo,15.68,12.76,3.79,3.29,35.52,82.73,8.3,709,Nintendo,E															
5	Wii Sports Resort,Wii,2009,Sports,Nintendo,15.61,10.93,3.28,2.95,32.77,80.73,8.192,Nintendo,E															
6	Pokemon Red/Pokemon Blue,GB,1996,Role-Playing,Nintendo,11.27,8.89,10.22,1.31,37,.....															
7	Tetris,GB,1989,Puzzle,Nintendo,23.2,2.26,4.22,0.58,30.26,.....															
8	New Super Mario Bros.,DS,2006,Platform,Nintendo,11.28,9.14,6.5,2.88,29.8,89.65,8.5,431,Nintendo,E															
9	Wii Play,Wii,2006,Misc,Nintendo,13.96,9.18,2.93,2.84,28.92,58.41,6.6,129,Nintendo,E															
10	New Super Mario Bros. Wii,Wii,2009,Platform,Nintendo,14.44,6.94,4.7,2.24,28.32,87.80,8.4,594,Nintendo,E															
11	Duck Hunt,NES,1984,Shooter,Nintendo,26.93,0.63,0.28,0.47,28.31,.....															
12	Nintendogs,DS,2005,Simulation,Nintendo,9.05,10.95,1.93,2.74,24.67,.....															
13	Mario Kart DS,DS,2005,Racing,Nintendo,9.71,7.47,4.13,1.9,23.21,91.64,8.6,464,Nintendo,E															
14	Pokemon Gold/Pokemon Silver,GB,1999,Role-Playing,Nintendo,9.6,18.7,2.0,71.23,1,.....															
15	Wii Fit,Wii,2007,Sports,Nintendo,8.92,8.03,3.6,2.15,22.7,80.63,7.7,146,Nintendo,E															
16	Kinect Adventures!,X360,2010,Misc,Microsoft Game Studios,15.4,89.0,24.1,69.21,81.61,45.6,3,106,Good Science Studio,E															
17	Wii Fit Plus,Wii,2009,Sports,Nintendo,9.01,8.49,2.53,1.77,21.79,80.33,7.4,52,Nintendo,E															
18	Grand Theft Auto V,PS3,2013>Action,Take-Two Interactive,7.02,9.09,0.98,3.96,21.04,97.50,8.2,3994,Rockstar North,M															
19	Grand Theft Auto: San Andreas,PS2,2004>Action,Take-Two Interactive,9.43,0.4,0.41,10.57,20.81,95.80,9,1588,Rockstar North,M															

Gambar 6. Dataset

Dataset ini terdiri dari 16.719 record data dan 15 *feature* yang berkaitan dengan data penjualan video game dan peringkat yang sesuai dengan data yang terdapat pada situs Metacritic, yaitu situs web yang mengumpulkan dan merangkum ulasan dari game video, film, acara televisi, dan buku. Berikut adalah rincian dari *feature* yang ada pada dataset yang dapat dilihat pada tabel dibawah ini.

No	Feature	No	Feature
1	Name	10	Global_Sales
2	Platform	11	Critic_Score
3	Year_of_Release	12	Critic_Count
4	Genre	13	User_Score
5	Publisher	14	User_Count
6	NA_Sales	15	Developer,Rating
7	EU_Sales		
8	JP_Sales		
9	Other_Sales		

Tabel 1. Feature Dataset

## 2. Preprocessing Data

Pada tahap ini, data yang diperoleh akan diolah dengan tujuan untuk membersihkan data dari masalah atau kesalahan yang dapat menyebabkan error, dari tahap ini juga nantinya akan dihasilkan data yang lebih akurat dan dapat diandalkan untuk diproses menggunakan teknologi atau metode yang ada pada *machine learning*, untuk konteks pengembangan ini teknologi yang dimaksud yaitu sistem rekomendasi. Beberapa hal yang dilakukan pada tahap ini diantaranya yaitu melakukan pengecekan missing value dan duplikasi data, menghapus entri dan *feature* yang tidak relevan dengan pengembangan sistem, serta mengonversi *feature* yang bertipe kategorikal menjadi numerikal, agar memudahkan untuk pengolahan lebih lanjut menggunakan teknologi *machine learning*

## 3. Pengembangan Model CBF

Model *Content Based Filtering* (CBF) dikembangkan dengan menganalisis fitur-fitur dari game yang terdapat dalam dataset. Setiap game direpresentasikan kedalam bentuk vektor berdasarkan atributnya. Selanjutnya, kesamaan antar game dihitung menggunakan metrik *cosine similarity*. Model ini kemudian digunakan untuk memberikan rekomendasi berdasarkan kemiripan atribut sesuai dengan preferensi pengguna.

#### 4. Implementasi Sistem

Model yang telah dikembangkan sebelumnya selanjutnya akan diimplementasikan menggunakan bahasa pemrograman Python dengan menggunakan sejumlah library yang dapat membantu dalam analisa data dan model *machine learning*. Pengembangan dari model ini juga akan dikembangkan kedalam bentuk website menggunakan framework Flask untuk menyediakan sistem dengan antarmuka yang interaktif sehingga dapat memenuhi kebutuhan pengguna dalam menemukan rekomendasi game yang sesuai dengan preverensi mereka.

#### 5. Pengujian Sistem

Sistem yang telah dikembangkan akan di evaluasi dengan cara memasukkan inputan berupa data game yang telah tersedia dalam dataset. Selanjutnya, sistem akan memproses hasil dari inputan yang dimasukkan sehingga dapat menampilkan rekomendasi game yang sesuai. Hasil rekomendasi kemudian diperiksa dan dibandingkan dengan data yang ada dalam dataset untuk memastikan kesesuaian rekomendasi yang diberikan.

### 3.2 Alat dan Bahan

Pengembangan sistem rekomendasi pilihan game berbasis Content Based Filtering (CBF) memerlukan sejumlah alat dan bahan pendukung seperti laptop untuk menjalankan proses pengolahan data, pengembangan model, dan implementasi sistem. Selain itu penggunaan tools Visual Studio Code sebagai text editor untuk menuliskan, serta menjalankan kode program, sementara itu bahasa pemrograman yang digunakan pada pengembangan sistem ini yaitu menggunakan Python dengan didukung sejumlah library seperti scikit-learn, pandas, dan numpy. Sedangkan, bahan yang digunakan diantaranya yaitu dataset “Video Game Sales with Ratings” yang berisi informasi seperti nama game, genre, platform, dan *feature* lainnya yang tertera pada Tabel 1. Feature Dataset.

## BAB 4

### HASIL DAN PEMBAHASAN

Pada bab ini menjelaskan lebih rinci tentang bagaimana pengembangan sistem rekomendasi pilihan game menggunakan metode *Content Based Filtering* (CBF) mulai dari proses preprocessing data hingga implementasi sistem menggunakan framework Flask. Hasil yang diperoleh dari pengembangan sistem ini yaitu berupa sistem rekomendasi yang dapat memberikan rekomendasi game berdasarkan data yang diinputkan pengguna.

#### 4.1 Proses Preprocessing Data

##### a) Import Library

Dengan mengimpor *library*, kita dapat mengakses fungsi, kelas, dan variabel yang ada tanpa harus menulis ulang kode tersebut. Berikut sejumlah *library* yang digunakan untuk pengembangan proyek ini :

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics.pairwise import cosine_similarity
```

Gambar 7. Import Library

Berikut adalah penjelasan fungsi dari masing-masing *library*:

- **import pandas as pd** : Mengimpor pandas untuk manipulasi dan analisis data dalam format tabel (DataFrame).
- **from sklearn.preprocessing import MinMaxScaler**: Mengimpor MinMaxScaler untuk menormalisasi data ke rentang tertentu (biasanya 0-1).
- **from sklearn.metrics.pairwise import cosine\_similarity**: Mengimpor cosine\_similarity untuk menghitung kemiripan antara dua vektor berdasarkan sudut cosinus.

##### b) Pembacaan Data Asli

Fungsi `read_csv` digunakan untuk membaca file CSV `Video_Games.csv` dan memuatnya ke dalam sebuah DataFrame pandas yang disimpan dalam variabel `df`.

```
df = pd.read_csv('data/Video_Games.csv')
```

Gambar 8. Fungsi `read_csv`



Setelah itu, kolom 'Name' diubah menjadi index dengan menggunakan `set_index('Name', inplace=True)` untuk memudahkan pencarian dan pengelolaan data berdasarkan nama video game, sehingga akses data menjadi lebih cepat dan efisien.

```
df.set_index('Name', inplace=True)
```

Gambar 9. Fungsi index

Kode `print(f'Dimensi awal dataset: {df.shape}')` menampilkan dimensi awal dari dataset dalam bentuk jumlah baris dan kolom, yang memberikan gambaran tentang ukuran dataset yang sedang diproses.

```
print(f'Dimensi awal dataset: {df.shape}')
```

Gambar 10. Fungsi df.shape()

Berdasarkan output dibawah ini, menunjukkan bahwa dataset awal memiliki 16.719 baris dan 15 kolom.

```
Dimensi awal dataset: (16719, 15)
```

Gambar 11. Dimensi Dataset

## c) Pembersihan Data

### 1. Pengecekan dan Penanganan Missing Value

Pengecekan *missing value* dilakukan dengan menggunakan fungsi `isnull().sum()`.

```
print(f'Jumlah missing value sebelum dihapus:\n{df.isnull().sum()}')
```

Gambar 12. Fungsi isnull().sum()

Berikut adalah output pengecekan *missing value*:

```
Jumlah missing value sebelum dihapus:
Platform          0
Year_of_Release   269
Genre             2
Publisher         54
NA_Sales          0
EU_Sales          0
JP_Sales          0
Other_Sales       0
Global_Sales      0
Critic_Score      8582
Critic_Count      8582
User_Score        6704
User_Count        9129
Developer         6623
Rating            6769
dtype: int64
```

Gambar 13. Output Missing Value

Output menunjukkan jumlah *missing value* di setiap kolom dataset. Sebanyak enam kolom tidak memiliki *missing value*, sedangkan delapan kolom lainnya memiliki jumlah *missing value* yang cukup besar. Oleh karena itu, diperlukan penanganan menggunakan fungsi `dropna()` untuk menghapus data dengan nilai kosong.

```
df.dropna(inplace=True)
print(f"Jumlah missing value setelah dihapus:\n{df.isnull().sum()}")
```

Gambar 14. Fungsi `dropna()`

Selanjutnya, dilakukan pengecekan ulang *missing value* untuk memastikan bahwa data dengan nilai kosong pada kolom yang memiliki *missing value* telah terhapus. Berikut adalah outputnya :

```
Jumlah missing value setelah dihapus:
Platform          0
Year_of_Release   0
Genre             0
Publisher         0
NA_Sales          0
EU_Sales          0
JP_Sales          0
Other_Sales       0
Global_Sales      0
Critic_Score      0
Critic_Count      0
User_Score        0
User_Count        0
Developer         0
Rating            0
dtype: int64
```

Gambar 15. Output Handling Missing Value

Output menunjukkan bahwa setelah dilakukan penghapusan menggunakan fungsi `dropna()`, semua kolom dalam dataset tidak lagi memiliki *missing value* (di setiap kolom bernilai 0). Setelah itu, menampilkan kembali dimensi dataset setelah penghapusan *missing value*.

```
print(f"Dimensi dataset setelah penghapusan missing value: {df.shape}")
```

Gambar 16. Fungsi `df.shape`

Berikut adalah output dimensi datasetnya :

```
Dimensi dataset setelah penghapusan missing value: (6825, 15)
```

Gambar 17. Dimensi Dataset

Menampilkan dimensi bahwa dataset memiliki 6.825 baris dan 15 kolom.

## 2. Penghapusan Duplikasi Data

```
duplicate_count = df.duplicated().sum()
print(f"Jumlah duplikasi data: {duplicate_count}")
```

Gambar 18. Fungsi `df.duplicate()`

Kode tersebut digunakan untuk menghitung jumlah data duplikat dalam dataset. Fungsi `df.duplicated()` mengidentifikasi baris-baris yang memiliki nilai sama persis dengan baris lain di dataset, dan `.sum()` menjumlahkan total baris

duplikat yang ditemukan. Hasilnya disimpan dalam variabel `duplicate_count`. Berikut adalah output duplikasi data:

```
Jumlah duplikasi data: 0
```

Gambar 19. Output Duplikasi Data

Output menunjukkan bahwa tidak ada data duplikat dalam dataset, artinya setiap baris data adalah unik dan tidak ada entri yang sama persis dengan yang lainnya.

```
# lihat unique value pada kolom Genre
print("Value Kolom Genre : ")
print(df['Genre'].unique())

# lihat unique value pada kolom Platform
print("Value Kolom Platform : ")
print(df['Platform'].unique())

# lihat unique value pada kolom Rating
print("Value Kolom Rating : ")
print(df['Rating'].unique())
```

Gambar 20. Unique Value

Kode di atas digunakan untuk menampilkan nilai-nilai unik (*unique values*) pada tiga kolom dalam dataset, yaitu Genre, Platform, dan Rating. Berikut adalah outputnya:

```
Value Kolom Genre :
['Sports' 'Racing' 'Platform' 'Misc' 'Action' 'Puzzle' 'Shooter'
 'Fighting' 'Simulation' 'Role-Playing' 'Adventure' 'Strategy']
Value Kolom Platform :
['Wii' 'DS' 'X360' 'PS3' 'PS2' '3DS' 'PS4' 'PS' 'XB' 'PC' 'PSP' 'WiiU'
 'GC' 'GBA' 'XOne' 'PSV' 'DC']
Value Kolom Rating :
['E' 'M' 'T' 'E10+' 'AO' 'K-A' 'RP']
```

Gambar 21. Output Unique Value

### 3. Penghapusan Kolom yang Tidak Relevan

```
columns_to_drop = ['Year_of_Release', 'Publisher', 'Global_Sales', 'NA_Sales',
                   'EU_Sales', 'JP_Sales', 'Other_Sales', 'Critic_Count', 'User_Count', 'Developer']
df.drop(columns=columns_to_drop, axis=1, inplace=True)
```

Gambar 22. Drop Kolom yang Tidak Relevan

Kode tersebut digunakan untuk menghapus kolom-kolom yang dianggap tidak diperlukan dalam analisis dataset dengan menggunakan fungsi `df.drop()`.

Setelah kode ini dijalankan, dataset akan memiliki kolom yang lebih sedikit, hanya menyisakan informasi yang diperlukan.

#### 4. Penghapusan Baris dengan Rating 'RP'

```
df = df[df['Rating'] != 'RP']
```

Gambar 23. Kolom Rating "RP"

Kode tersebut digunakan untuk memfilter dataset sehingga hanya baris yang memiliki rating selain 'RP' yang akan dipertahankan. Nilai 'RP' biasanya mengindikasikan bahwa rating dari game tersebut belum ditentukan.

Setelah proses penghapusan kolom dan baris, tahap selanjutnya menampilkan dimensi terbaru dari dataset, yaitu jumlah baris dan kolom yang tersisa dengan menggunakan fungsi `df.shape`

```
print(f"Dimensi dataset setelah penghapusan kolom dan baris tertentu: {df.shape}")
```

Gambar 24. Print Dimensi Baru

Berikut adalah outputnya:

```
Dimensi dataset setelah penghapusan kolom dan baris tertentu: (6824, 5)
```

Gambar 25. Hasil Dimensi Baru

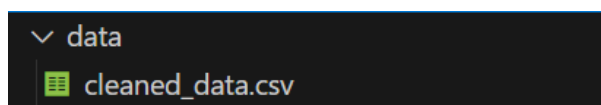
Menampilkan dimensi terbaru dari dataset yang terdiri dari 6.824 baris dan 5 kolom.

Setelah tahap pembersihan data, dataset yang telah dibersihkan akan disimpan ke dalam file CSV baru dengan nama `data/cleaned_data.csv`. Berikut adalah kodenya:

```
df.to_csv('data/cleaned_data.csv', index=True)
```

Gambar 26. Simpan Data Baru

Berikut adalah output penyimpanan dataset ke dalam file CSV baru dengan nama `data/cleaned_data.csv`.



Gambar 27. Dataset Baru

Name	Platform	Genre	Critic_Score	User_Score	Rating
Wii Sports	Wii	Sports	76.0	8	E
Mario Kart Wii	Wii	Racing	82.0	8.3	E
Wii Sports Resort	Wii	Sports	80.0	8	E
New Super Mario Bros.	DS	Platform	89.0	8.5	E
Wii Play	Wii	Misc	58.0	6.6	E
New Super Mario Bros. Wii	Wii	Platform	87.0	8.4	E
Mario Kart DS	DS	Racing	91.0	8.6	E
Wii Fit	Wii	Sports	80.0	7.7	E
Kinect Adventures!	X360	Misc	61.0	6.3	E
Wii Fit Plus	Wii	Sports	80.0	7.4	E
Grand Theft Auto V	PS3	Action	97.0	8.2	M
Grand Theft Auto: San Andreas	PS2	Action	95.0	9	M
Brain Age: Train Your Brain in Minutes a Day	DS	Misc	77.0	7.9	E
Grand Theft Auto V	X360	Action	97.0	8.1	M
Grand Theft Auto: Vice City	PS2	Action	95.0	8.7	M
Brain Age 2: More Training in Minutes a Day	DS	Puzzle	77.0	7.1	E

Gambar 28. Gambaran Dataset Baru

Dataset yang telah tersimpan dapat digunakan untuk tahap normalisasi.

#### d) Normalisasi Data

Normalisasi data merupakan mengubah rentang antar variabel atau atribut dalam data menjadi lebih menjadi lebih kecil. Normalisasi sangat penting untuk memastikan hasil prediksi yang akurat dan dapat diandalkan. Dataset yang digunakan sering memiliki skala yang berbeda, terutama pada atribut numerik. Perbedaan skala ini dapat mempengaruhi kinerja algoritma *machine learning*.

Adapun metode yang dapat digunakan untuk normalisasi data yaitu Min-Max normalization, Z-score Normalization, dan Feature Scaling. Dalam proyek ini, kami menggunakan salah satu metode normalisasi data yaitu Min-Max normalization. Tahapannya sebagai berikut:

##### 1) Membaca Kembali Dataset Cleaned

```
df_cleaned = pd.read_csv('data/cleaned_data.csv', index_col=0)
```

Gambar 29. Read Dataset Baru

Kode tersebut digunakan untuk memuat dataset yang telah dibersihkan dari file data/cleaned\_data.csv ke dalam DataFrame baru df\_cleaned, dengan kolom pertama sebagai indeks.

##### 2) Mengonversi Kolom Kategorikal menjadi Numerikal

```
df_cleaned['Platform'] = pd.Categorical(df_cleaned['Platform'])
df_cleaned['Genre'] = pd.Categorical(df_cleaned['Genre'])
df_cleaned['Rating'] = pd.Categorical(df_cleaned['Rating'])
df_cleaned['Platform'] = df_cleaned['Platform'].cat.codes
df_cleaned['Genre'] = df_cleaned['Genre'].cat.codes
df_cleaned['Rating'] = df_cleaned['Rating'].cat.codes
```

Gambar 30. Konversi Kategorikal

Kode tersebut digunakan untuk mengubah kolom 'Platform', 'Genre', dan 'Rating' dalam DataFrame `df_cleaned` menjadi tipe data kategori, kemudian mengonversinya menjadi angka kode. Setiap kategori yang ada pada kolom tersebut akan diberi angka unik (misalnya, 0, 1, 2, dan seterusnya).

### 3) Normalisasi menggunakan MinMaxScaler

```
scaler = MinMaxScaler()
df_normalized = pd.DataFrame(scaler.fit_transform(df_cleaned),
                             columns=df_cleaned.columns,
                             index=df_cleaned.index)
print("Contoh data setelah normalisasi:")
print(df_normalized.head())
```

Gambar 31. Normalisasi MinMaxScaler()

Kode tersebut melakukan normalisasi data pada `df_cleaned` menggunakan `MinMaxScaler`, yang mengubah nilai-nilai setiap fitur ke dalam rentang 0 hingga 1. Hasilnya disimpan dalam `df_normalized`, dan contoh data setelah normalisasi ditampilkan dengan `df_normalized.head()`. Berikut adalah outputnya:

Contoh data setelah normalisasi:					
	Platform	Genre	Critic_Score	User_Score	Rating
Name					
Wii Sports	0.750	0.909091	0.741176	0.824176	0.2
Mario Kart Wii	0.750	0.545455	0.811765	0.857143	0.2
Wii Sports Resort	0.750	0.909091	0.788235	0.824176	0.2
New Super Mario Bros.	0.125	0.363636	0.894118	0.879121	0.2
Wii Play	0.750	0.272727	0.529412	0.670330	0.2

Gambar 32. Hasil Normalisasi

Setelah tahap normalisasi, dataset yang telah dinormalisasi akan disimpan ke dalam file CSV baru dengan nama **data/normalized\_data.csv**. Berikut adalah kodenya:

```
df_normalized.to_csv('data/normalized_data.csv', index=True)
```

Gambar 33. Simpan Hasil Normalisasi

Berikut adalah output penyimpanan dataset ke dalam file CSV baru dengan nama **data/normalized\_data.csv**.

```

v data
  cleaned_data.csv
  normalized_data.csv
```

Name	Platform	Genre	Critic_Score	User_Score	Rating
Wii Sports	0.75	0.9090909090909092	0.7411764705882352	0.8241758241758241	0.2
Mario Kart Wii	0.75	0.5454545454545454	0.8117647058823529	0.8571428571428572	0.2
Wii Sports Resort	0.75	0.9090909090909092	0.788235294117647	0.8241758241758241	0.2
New Super Mario Bros.	0.125	0.3636363636363636	0.8941176470588236	0.8791208791208791	0.2
Wii Play	0.75	0.2727272727272727	0.5294117647058824	0.6703296703296703	0.2
New Super Mario Bros. Wii	0.75	0.3636363636363636	0.8705882352941176	0.8681318681318682	0.2
Mario Kart DS	0.125	0.5454545454545454	0.9176470588235294	0.8901098901098901	0.2
Wii Fit	0.75	0.9090909090909092	0.788235294117647	0.7912087912087912	0.2
Kinect Adventures!	0.875	0.2727272727272727	0.5647058823529412	0.6373626373626373	0.2
Wii Fit Plus	0.75	0.9090909090909092	0.788235294117647	0.7582417582417582	0.2
Grand Theft Auto V	0.5	0.0	0.988235294117647	0.846153846153846	0.8
Grand Theft Auto: San Andreas	0.4375	0.0	0.9647058823529412	0.9340659340659341	0.8
Brain Age: Train Your Brain in Minutes a Day	0.125	0.2727272727272727	0.7529411764705882	0.8131868131868132	0.2
Grand Theft Auto V	0.875	0.0	0.988235294117647	0.8351648351648351	0.8
Grand Theft Auto: Vice City	0.4375	0.0	0.9647058823529412	0.901098901098901	0.8
Brain Age 2: More Training in Minutes a Day	0.125	0.4545454545454546	0.7529411764705882	0.7252747252747251	0.2

Gambar 34. Dataset Hasil Normalisasi

Dataset yang telah tersimpan dapat digunakan untuk tahap selanjutnya.

### e) Pembangunan Matriks Cosine Similarity

Matriks cosine similarity digunakan untuk mengukur tingkat kesamaan antar data dalam suatu dataset berdasarkan nilai-nilai vektor yang telah dinormalisasi. Cosine similarity menghitung nilai kosinus dari sudut antara dua vektor, yang menghasilkan nilai antara 0 hingga 1 untuk kemiripan positif, atau -1 untuk kemiripan negatif. Nilai ini sering digunakan dalam sistem rekomendasi atau analisis kemiripan data. Berikut adalah kode *cosine similarity*:

```
cosine_sim = cosine_similarity(df_normalized)
cosine_sim_df = pd.DataFrame(cosine_sim, index=df_normalized.index, columns=df_normalized.index)
```

Gambar 35. Cosine Similarity

Kode tersebut menghitung matriks cosine similarity dari data yang telah dinormalisasi (`df_normalized`) untuk mengukur kemiripan antar baris. Hasilnya disimpan dalam `cosine_sim_df`.

Selanjutnya, beberapa informasi dari `cosine_sim_df`, seperti lima baris pertama, daftar kolom, dan indeks, ditampilkan untuk memberikan gambaran tentang struktur dan isi matriks tersebut. Berikut adalah kodenya:

```
print("Contoh data head:")
print(cosine_sim_df.head()) # Menampilkan 5 baris pertama
print("Contoh data kolom:")
print(cosine_sim_df.columns) # Menampilkan kolom atau indeks
print("Contoh data index:")
print(cosine_sim_df.index) # Menampilkan indeks DataFrame
```

Gambar 36. Informasi Cosine

Berikut adalah output lima baris pertama, daftar kolom, dan indeks dari matriks cosine similarity:



```

Contoh data kolom:
Index(['Wii Sports', 'Mario Kart Wii', 'Wii Sports Resort',
      'New Super Mario Bros.', 'Wii Play', 'New Super Mario Bros. Wii',
      'Mario Kart DS', 'Wii Fit', 'Kinect Adventures!', 'Wii Fit Plus',
      ...
      'Trine', 'Karnaaj Rally', 'Sébastien Loeb Rally Evo', 'Hospital Tycoon',
      'Fast Racing Neo ', 'E.T. The Extra-Terrestrial',
      'Mortal Kombat: Deadly Alliance', 'Metal Gear Solid V: Ground Zeroes',
      'Breach', 'STORM: Frontline Nation'],
      dtype='object', name='Name', length=6824)
Contoh data kolom:
Index(['Wii Sports', 'Mario Kart Wii', 'Wii Sports Resort',
      'New Super Mario Bros.', 'Wii Play', 'New Super Mario Bros. Wii',
      'Mario Kart DS', 'Wii Fit', 'Kinect Adventures!', 'Wii Fit Plus',
      ...
      'Trine', 'Karnaaj Rally', 'Sébastien Loeb Rally Evo', 'Hospital Tycoon',
      'Fast Racing Neo ', 'E.T. The Extra-Terrestrial',
      'Mortal Kombat: Deadly Alliance', 'Metal Gear Solid V: Ground Zeroes',
      'Breach', 'STORM: Frontline Nation'],
      dtype='object', name='Name', length=6824)
Contoh data index:
Index(['Wii Sports', 'Mario Kart Wii', 'Wii Sports Resort',
      'New Super Mario Bros.', 'Wii Play', 'New Super Mario Bros. Wii',
      'Mario Kart DS', 'Wii Fit', 'Kinect Adventures!', 'Wii Fit Plus',
      ...
      'Trine', 'Karnaaj Rally', 'Sébastien Loeb Rally Evo', 'Hospital Tycoon',
      'Fast Racing Neo ', 'E.T. The Extra-Terrestrial',
      'Mortal Kombat: Deadly Alliance', 'Metal Gear Solid V: Ground Zeroes',
      'Breach', 'STORM: Frontline Nation'],
      dtype='object', name='Name', length=6824)

```

Gambar 37. Hasil Cosine Similarity

Informasi ini memberikan konteks untuk melanjutkan proses pengembangan, seperti pembuatan fungsi rekomendasi permainan.

```

def CosineGameRecommended(gamename, recommended_games=5):
    # Validasi keberadaan gamename di index
    if gamename not in cosine_sim_df.index:
        return pd.DataFrame(columns=["Game", "Cosine Similarity"]) # Kembalikan DataFrame kosong

    # Ambil data similarity untuk gamename
    data = cosine_sim_df.loc[gamename]
    if isinstance(data, pd.DataFrame):
        data = data.iloc[0]

    if not isinstance(data, pd.Series):
        return pd.DataFrame(columns=["Game", "Cosine Similarity"]) # Kembalikan DataFrame kosong jika tipe data salah

    # Urutkan similarity scores
    sim_scores = data.sort_values(ascending=False)[1:recommended_games + 1]
    result_df = pd.DataFrame({
        "Game": sim_scores.index,
        "Cosine Similarity": sim_scores.values
    })

    return result_df

```

Gambar 38. Fungsi Cosine Similarity

Fungsi `CosineGameRecommended` dirancang untuk memanfaatkan matriks cosine similarity ini, sehingga dapat merekomendasikan sejumlah permainan yang memiliki kemiripan dengan permainan tertentu yang dipilih oleh pengguna. Berikut adalah penjelasan mengenai kode di atas :

Pertama, fungsi memvalidasi apakah nama permainan yang diminta ada dalam indeks matriks *cosine similarity* (`cosine_sim_df`). Jika nama permainan tidak ditemukan, fungsi akan mengembalikan DataFrame kosong.

Selanjutnya, fungsi mengambil data kemiripan untuk permainan tersebut dan memastikan tipe data yang sesuai. Kemudian, skor kemiripan diurutkan secara menurun, dan fungsi memilih sejumlah permainan dengan skor kemiripan tertinggi sesuai dengan parameter `recommended_games` (default 5). Hasil akhirnya adalah DataFrame yang berisi nama permainan yang direkomendasikan beserta skor *cosine similarity*-nya.

#### f) Fungsi Rekomendasi Berdasarkan Kategori Tertentu

```
def GameRecommended(platform: str, genre: str, rating: str, recommended_games: int = 5):
    # Validasi kategori dari data asli
    platform_categories = pd.Categorical(df_original['Platform']).categories
    genre_categories = pd.Categorical(df_original['Genre']).categories
    rating_categories = pd.Categorical(df_original['Rating']).categories

    if platform not in platform_categories:
        print(f"Platform '{platform}' tidak ditemukan dalam dataset.")
        return pd.DataFrame(columns=["Game", "Cosine Similarity"])

    if genre not in genre_categories:
        print(f"Genre '{genre}' tidak ditemukan dalam dataset.")
        return pd.DataFrame(columns=["Game", "Cosine Similarity"])

    if rating not in rating_categories:
        print(f"Rating '{rating}' tidak ditemukan dalam dataset.")
        return pd.DataFrame(columns=["Game", "Cosine Similarity"])

    # Konversi input ke kode numerik
    platform_code = platform_categories.get_loc(platform)
    genre_code = genre_categories.get_loc(genre)
    rating_code = rating_categories.get_loc(rating)

    # Filter game sesuai input
    filtered_games = df_cleaned[(df_cleaned['Platform'] == platform_code) &
                                (df_cleaned['Genre'] == genre_code) &
                                (df_cleaned['Rating'] == rating_code)]

    if filtered_games.empty:
        print(f"Tidak ada game yang ditemukan dengan kombinasi Platform '{platform}', Genre '{genre}', dan Rating '{rating}'.")
        return pd.DataFrame(columns=["Game", "Cosine Similarity"])

    # Hitung similarity untuk game yang difilter
    filtered_indices = filtered_games.index
    cosine_sim_filtered = cosine_sim_df.loc[filtered_indices, filtered_indices]
    recommended_games_list = cosine_sim_filtered.mean(axis=0).sort_values(ascending=False)[1:recommended_games + 1]

    # Membuat dataframe rekomendasi
    recommended = pd.DataFrame({"Game": recommended_games_list.index, "Cosine Similarity": recommended_games_list.values})
    return recommended.reset_index(drop=True)
```

Gambar 39. Fungsi Rekomendasi Game

Fungsi `GameRecommended` bertujuan untuk merekomendasikan permainan berdasarkan kategori yang dipilih pengguna, seperti platform, genre, dan rating. Berikut adalah penjelasan mengenai kode di atas:

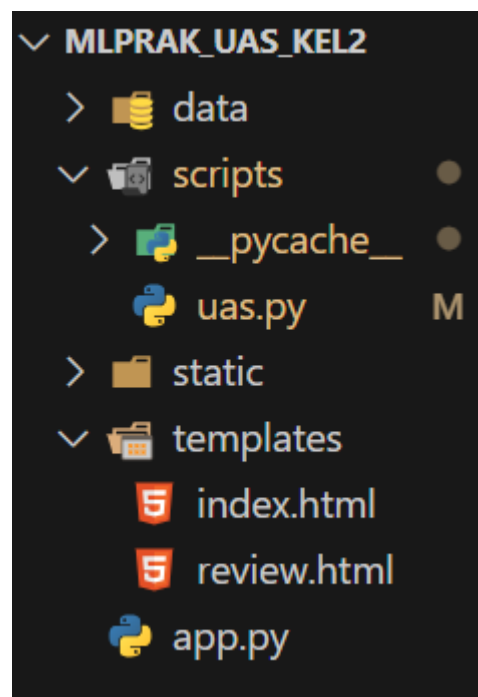
Fungsi ini dimulai dengan memvalidasi apakah kategori yang diberikan (platform, genre, dan rating) ada dalam dataset. Jika kategori tidak ditemukan, fungsi akan mengembalikan DataFrame kosong. Setelah validasi, kategori yang diberikan dikonversi menjadi kode numerik untuk mempermudah pencarian.

Kemudian, permainan yang sesuai dengan kombinasi platform, genre, dan rating difilter dari dataset. Jika tidak ada permainan yang cocok, fungsi akan mengembalikan DataFrame kosong.

Selanjutnya, fungsi menghitung kemiripan antar permainan yang telah difilter menggunakan matriks *cosine similarity* dan memilih permainan dengan skor kemiripan tertinggi. Hasil akhirnya adalah DataFrame yang berisi daftar permainan yang direkomendasikan beserta nilai *cosine similarity*-nya.

## 4.2 Proses Implementasi Sistem

### 1. Struktur Proyek



Gambar 40. Struktur Direktori Proyek

Gambar diatas merupakan struktur dari direktori pengembangan sistem rekomendasi game. Terdapat sejumlah direktori dan file utama diantaranya yaitu file **app.py** untuk implementasi Flask, direktori `templates/` yang berisi file HTML untuk tampilan sistem, direktori `static` untuk menyimpan file statis seperti CSS, JavaScript, dan Image, selain itu terdapat direktori `scripts` yang digunakan untuk menyimpan file pengolahan data.

## 2. Inisialisasi Flask dan Import Library

```
from flask import Flask, request, render_template
from scripts.uas import CosineGameRecommended, GameRecommended
from typing import Optional
import os
import random

app = Flask(__name__, static_folder='static', template_folder='templates')

# List untuk menyimpan gambar yang sudah dipilih
used_images = []
```

Gambar 41. Inisialisasi Flask dan Library

Kode diatas merupakan implementasi sistem berbasis **Flask** untuk memberikan rekomendasi game berdasarkan input dari pengguna. Bagian pertama adalah inisialisasi proyek dengan mengimpor library yang dibutuhkan, seperti Flask, os, dan random. Selanjutnya, aplikasi dikonfigurasi dengan direktori `static_folder` dan `template_folder` yang berisi file statis dan template HTML yang digunakan untuk tampilan antarmuka pengguna.

## 3. Fungsi Pemilihan Gambar Acak tanpa Duplikasi

```
# Fungsi untuk mendapatkan gambar acak tanpa duplikasi
def get_random_image(folder):
    global used_images
    folder_path = os.path.join('static', folder)

    # Dapatkan semua gambar dalam folder
    images = [f for f in os.listdir(folder_path) if os.path.isfile(os.path.join(folder_path, f))]

    # Filter gambar yang belum dipilih
    available_images = [img for img in images if img not in used_images]

    if available_images:
        # Pilih gambar secara acak dari yang tersedia
        selected_image = random.choice(available_images)
        used_images.append(selected_image) # Simpan gambar yang sudah dipilih
        return f"{folder}/{selected_image}"
    else:
        # Jika semua gambar sudah digunakan, reset list dan mulai ulang
        used_images = []
        return get_random_image(folder)
```

Gambar 42. Fungsi Pemilihan Gambar Acak

Fungsi `get_random_image()` dibuat untuk memilih gambar acak dari folder tanpa mengulang gambar yang sama yang sebelumnya telah ditampilkan. Jika semua gambar telah digunakan, daftar akan direset untuk memulai dari awal.

#### 4. Route Sistem

##### a) Halaman Index

```
# Rute halaman utama
@app.route('/', methods=['GET'])
def index():
    return render_template('index.html')
```

Gambar 43. Route Index

Halaman utama diakses melalui rute /, yang mengembalikan tampilan dari file index.html. Ada dua fitur utama dalam aplikasi ini: rekomendasi berdasarkan nama game dan rekomendasi berdasarkan kategori (platform, genre, dan rating).

##### b) Rekomendasi Berdasarkan Nama Game dan Kategori Tertentu

```
# Rekomendasi berdasarkan nama game
@app.route('/recommend', methods=['POST'])
def recommend():
    game_name = request.form.get('game_name', '').strip()
    recommendations = None
    error = None
    images = []

    if game_name:
        num_recommendations = 6
        result = CosineGameRecommended(game_name, num_recommendations)
        if result.empty:
            error = f"Game '{game_name}' not found in dataset."
        else:
            recommendations = result.set_index("Game").to_dict()["Cosine Similarity"]
            recommendations = dict(list(recommendations.items())[:6])
            images = [get_random_image('img/review') for _ in range(len(recommendations))]
    else:
        error = "Please enter a valid game name."

    return render_template(
        'review.html',
        recommendations=recommendations,
        error=error,
        game_name=game_name,
        images=images
    )
```

Gambar 44. Fungsi Rekomendasi Berdasarkan Nama Game

Rekomendasi berdasarkan nama game, rute /recommend digunakan untuk menangkap input nama dari pengguna, lalu memanggil fungsi CosineGameRecommended() untuk menentukan rekomendasi game berdasarkan kemiripan skor. Jika input valid, hasil akan ditampilkan bersama dengan gambar acak yang dipilih dari direktori img/review.

```

# Rekomendasi berdasarkan kategori
@app.route('/recommend_by_category', methods=['POST'])
def recommend_by_category():
    platform = request.form.get('platform', '').strip()
    genre = request.form.get('genre', '').strip()
    rating = request.form.get('rating', '').strip()
    error = None
    recommendations = None
    images = []

    if platform and genre and rating:
        num_recommendations = 6
        result = GameRecommended(platform, genre, rating, num_recommendations)
        if result.empty():
            error = f"No games found for Platform '{platform}', Genre '{genre}', and Rating '{rating}'."
        else:
            recommendations = result.set_index("Game").to_dict()["Cosine Similarity"]
            recommendations = dict(list(recommendations.items())[:6])
            images = [get_random_image('img/review') for _ in range(len(recommendations))]
    else:
        error = "Please complete all filters (Platform, Genre, Rating)."

```

Gambar 45. Fungsi Rekomendasi Berdasarkan Kategori

Sedangkan untuk rekomendasi berdasarkan kategori, rute `/recommend_by_category` akan menangkap input platform, genre, dan rating dari formulir pengguna, lalu memanggil fungsi `GameRecommended()` untuk mendapatkan rekomendasi game berdasarkan kombinasi yang dipilih.

## 5. Running Flask

```

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

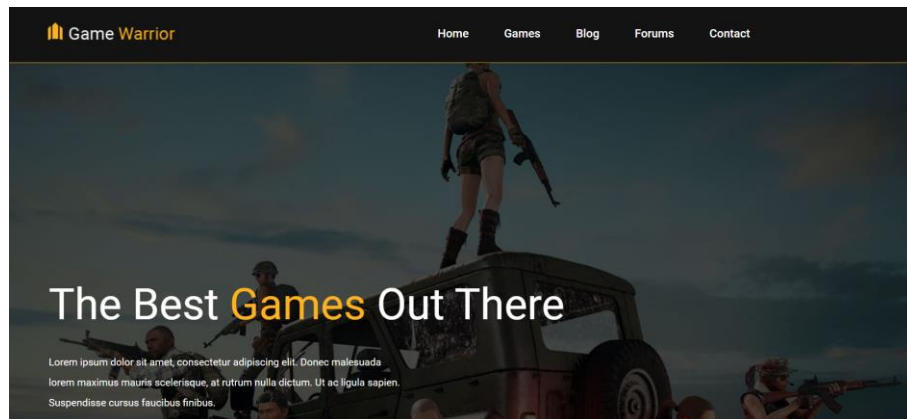
```

Gambar 46. Running Sistem

Aplikasi dijalankan menggunakan `app.run(debug=True)`. Ini membuat server Flask berjalan dalam mode debug, yang memudahkan pengembang dalam memantau perubahan kode dan memperbaiki kesalahan tanpa perlu me-restart server setiap kali ada perubahan kode.

## 4.3 Hasil Implementasi Sistem

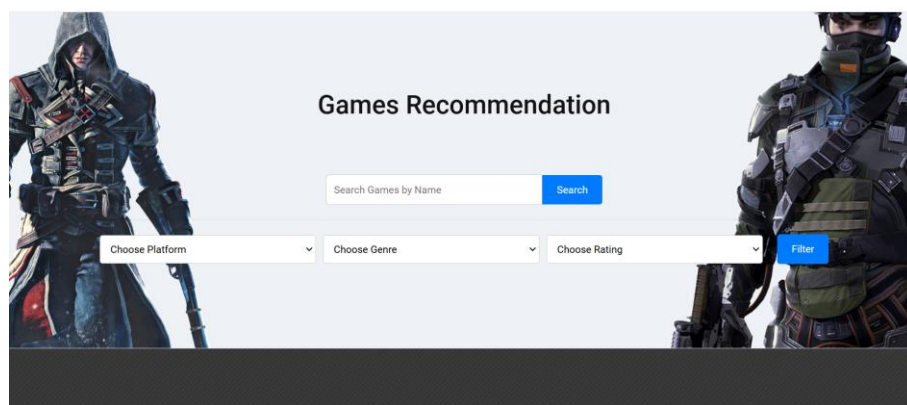
### 1. Halaman Beranda



Gambar 47. Halaman Beranda

Halaman beranda merupakan tampilan awal yang akan muncul ketika pengguna mengakses website ini. Tampilan ini berisi teks “The Best Games Out There” dan sejumlah menu.

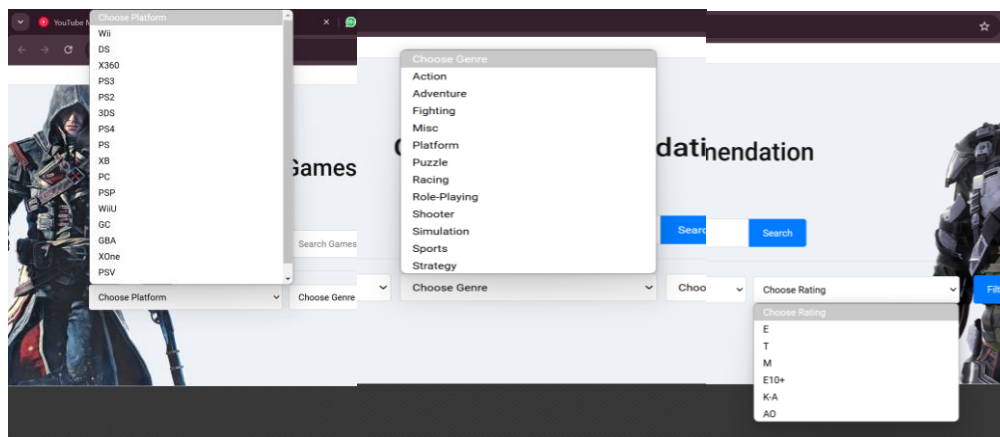
### 2. Halaman Form Rekomendasi



Gambar 48. Halaman Form Rekomendasi

Halaman form rekomendasi berisi formulir atau form yang digunakan untuk memasukkan data untuk menentukan rekomendasi game. Pada halaman ini, pengguna dapat mencari rekomendasi game menggunakan dua pilihan, yaitu berdasarkan nama game atau berdasarkan kategori tertentu, yang pada form ini kategori yang dimaksud terdiri dari Platform, Genre, dan Rating dari game yang diinginkan.

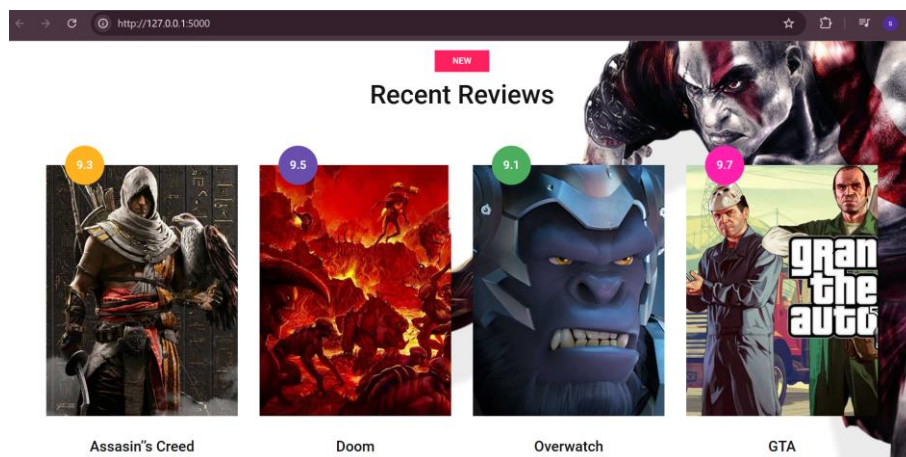
### 3. Halaman Kategori



Gambar 49. Kategori Rekomendasi

Gambar diatas merupakan rincian daftar kategori yang dapat dipilih oleh pengguna untuk mencari rekomendasi game tertentu. Rincian kategori tersebut diambil dari *feature* yang ada di dataset.

### 4. Halaman Recent Review

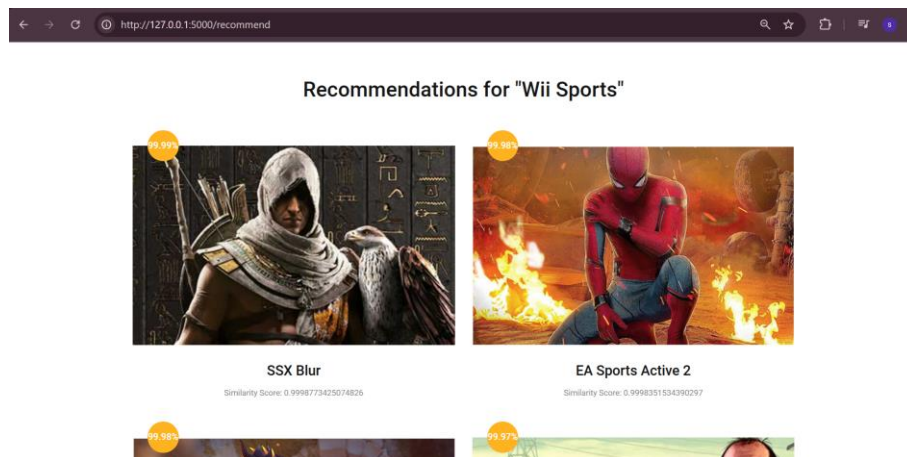


Gambar 50. Recent Review

Halaman *Recent Review* dirancang sebagai fitur penting dalam sistem rekomendasi ini, yang berfungsi untuk memberikan informasi terkini kepada pengguna mengenai game-game terbaru yang telah direview oleh para pengguna. Hal ini memungkinkan pengguna untuk melihat rating dari pengguna lainnya sebelum memutuskan untuk mencoba atau memainkan suatu game. Dengan fitur ini, pengalaman pengguna dalam menemukan rekomendasi akan semakin informatif dan berbasis pengalaman nyata dari pengguna lain.



## 5. Halaman Hasil Rekomendasi



Gambar 51. Halaman Hasil Rekomendasi

Halaman *Rekomendasi* adalah fitur utama dalam sistem ini yang dirancang untuk membantu pengguna menemukan game yang sesuai dengan preferensi mereka berdasarkan analisis data. Halaman ini akan menampilkan daftar game yang direkomendasikan kepada pengguna dengan metode Content-Based Filtering (CBF) menggunakan *Cosine Similarity*.

## **BAB 5**

### **KESIMPULAN**

Proyek ini berhasil mengembangkan sistem rekomendasi pilihan game berbasis *Content-Based Filtering* dengan memanfaatkan metode *Cosine Similarity* untuk menghitung kesamaan antar atribut game. Data yang digunakan telah melalui tahapan preprocessing, termasuk penanganan missing value, normalisasi, dan transformasi data. Sistem yang diimplementasikan menggunakan framework Flask mampu memberikan rekomendasi berdasarkan preferensi pengguna melalui input nama game maupun kategori tertentu seperti platform, genre, dan rating. Hasil pengujian menunjukkan bahwa sistem dapat memberikan rekomendasi yang relevan dan akurat sesuai dengan data yang tersedia. Dengan demikian, sistem ini berpotensi meningkatkan pengalaman pengguna dalam memilih game yang sesuai dengan preferensi mereka.

## **LAMPIRAN**

Link Github Implementasi Sistem :

[https://github.com/salmaaida0504/PROYEK-UAS\\_ML-PRAKTIKUM\\_KEL-2.git](https://github.com/salmaaida0504/PROYEK-UAS_ML-PRAKTIKUM_KEL-2.git)

## DAFTAR PUSTAKA

- Arfisko, Hilmi Hidayat, and Agung Toto Wibowo. 2022. "Sistem Rekomendasi Film Menggunakan Metode Hybrid Collaborative Filtering Dan Content-Based Filtering." *E-Proceeding of Engineering* 9(3):2149–59.
- Mondi, Rhesa Havilah, Ardhi Wijayanto, and Winarno. 2019. "Recommendation System with Content-Based Filtering Method for Culinary Tourism in Mangan Application." *ITSMART: Jurnal Ilmiah Teknologi Dan Informasi* 8(2):65–72.
- Hellyana, Corie Mei, Nuzul Imam Fadlilah, Ragil Wijianto, and Khoerul Amin Ade Putra. 2023. "Game Edukasi 'Perjalanan Si Koko' Sebagai Media Pembelajaran." *Informatics and Computer Engineering Journal* 3(1):88–96. doi: 10.31294/icej.v3i1.1784.
- Putri, Dewa Ayu, Diah Pramesti, Dan I, and Wayan Santiyasa. 2022. "Penerapan Metode Content-Based Filtering Dalam Sistem Rekomendasi Video Game." *Jnatia* 1(1):229–34.